CHAIN XVM/EXECUTE XVM UTILITY MANUAL

DEC-XV-UCHNA-A-D

XVM Systems

digital

CHAIN XVM/EXECUTE XVM UTILITY MANUAL

DEC-XV-UCHNA-A-D

digital equipment corporation · maynard. massachusetts

First Printing, December 1975

The information in this document is subject to change without notice and should not be construed as a commitment by Digital Equipment Corporation. Digital Equipment Corporation assumes no responsibility for any errors that may appear in this document.

Digital Equipment Corporation assumes no responsibility for the use or reliability of its software on equipment that is not supplied by DIGITAL.

The postage prepaid READER'S COMMENTS form on the last page of this document requests the user's critical evaluation to assist us in preparing future documentation.

CONTENTS

CONTENTS (Cont.)

LIST OF ALL XVM MANUALS

The following is a list of all XVM manuals and their DEC numbers, including the latest version available. Within this manual, other XVM manuals are referenced by title only. Refer to this list for the DEC numbers of these referenced manuals.

BOSS XVM USER'S MANUAL	DEC-XV-OBUAA-A-D
CHAIN XVM/EXECUTE XVM UTILITY MANUAL	DEC-XV-UCHNA-A-D
DDT XVM UTILITY MANUAL	DEC-XV-UDDTA-A-D
EDIT/EDITVP/EDITVT XVM UTILITY MANUAL	DEC-XV-UETUA-A-D
8TRAN XVM UTILITY MANUAL	DEC-XV-UTRNA-A-D
FOCAL XVM LANGUAGE MANUAL	DEC-XV-LFLGA-A-D
FORTRAN IV XVM LANGUAGE MANUAL	DEC-XV-LF4MA-A-D
FORTRAN IV XVM OPERATING ENVIRONMENT MANUAL	DEC-XV-LF4EA-A-D
LINKING LOADER XVM UTILITY MANUAL	DEC-XV-ULLUA-A-D
MAC11 XVM ASSEMBLER LANGUAGE MANUAL	DEC-XV-LMLAA-A-D
MACRO XVM ASSEMBLER LANGUAGE MANUAL	DEC-XV-LMALA-A-D
MTDUMP XVM UTILITY MANUAL	DEC-XV-UMTUA-A-D
PATCH XVM UTILITY MANUAL	DEC-XV-UPUMA-A-D
PIP XVM UTILITY MANUAL	DEC-XV-UPPUA-A-D
SGEN XVM UTILITY MANUAL	DEC-XV-USUTA-A-D
SRCCOM XVM UTILITY MANUAL	DEC-XV-USRCA-A-D
UPDATE XVM UTILITY MANUAL	DEC-XV-UUPDA-A-D
VP15A XVM GRAPHICS SOFTWARE MANUAL	DEC-XV-GVPAA-A-D
VT15 XVM GRAPHICS SOFTWARE MANUAL	DEC-XV-GVTAA-A-D
XVM/DOS KEYBOARD COMMAND GUIDE	DEC-XV-ODKBA-A-D
XVM/DOS READERS GUIDE AND MASTER INDEX	DEC-XV-ODGIA-A-D
XVM/DOS SYSTEM MANUAL	DEC-XV-ODSAA-A-D
XVM/DOS USERS MANUAL	DEC-XV-ODMAA-A-D
XVM/DOS V1A SYSTEM INSTALLATION GUIDE	DEC-XV-ODSIA-A-D
XVM/RSX SYSTEM MANUAL	DEC-XV-IRSMA-A-D
XVM UNICHANNEL SOFTWARE MANUAL	DEC-XV-XUSMA-A-D

PREFACE

The operation and use of the CHAIN XVM/EXECUTE XVM Utility Program is described in this manual. It was assumed in the preparation of this manual, that the reader is familiar with the operation of the XVM equipment and the contents of the following manuals:

XVM/DOS System Manual
XVM/DOS Users Manual

CHAPTER 1
INTRODUCTION

The programs CHAIN XVM (CHAIN) and EXECUTE XVM (EXECUTE) facilitate a user generated system of core overlays in an XVM/DOS (DOS) environment. This system of overlays consists of a resident main program, and may include other indicated resident routines, a resident blank COMMON storage area, and a set of subroutines which overlay each other as directed by the user. These subroutines are grouped into units called LINKS. Many or all LINKS may overlay each other, and several LINKS may overlay a larger LINK without overlaying each other. Cascading of sub-overlays is not limited.

A LINK is loaded into core when a subroutine within the LINK is called and remains resident until overlaid. A LINK's core image is not recorded or "swapped out" when it is overlaid. The same image is brought into core each time a LINK is loaded.

Subroutines are called and return control to the calling routine in the normal fashion, except when a calling routine will be overlaid by a called routine. In the latter case, no arguments can be conveyed via the call, and subroutine exit must be accomplished by calling another routine.

Within links, there is no imposed order in which routines must be called nor is there restriction of the routines callable by any routine.

The Overlay System, when recorded, is called an XCT file[1].

The program CHAIN is used to build an XCT file and the program EXECUTE supervises core residency during the execution of a CHAIN-built Overlay System.

[1] An "XCT file" is actually two files (see Appendix B).

CHAPTER 2
THE OVERLAY SYSTEM

CHAIN-built Overlay Systems consist of 1) resident and non-resident code, 2) COMMON storage, and 3) a Link Table, all of which are described below.

The Overlay System is built onto an I/O device (XCT file) and the core that will be required to run it need not be available (or existent) to build it.

I/O handlers are not included in an XCT file, but are loaded prior to execution per .DAT slot assignments and remain in core throughout a run.

2.1 RESIDENT ROUTINES

CHAIN requests a list of resident routines. All routines listed and any library routines they may call remain in core throughout a run. The first resident routine is the program to which initial control of the execution of the Overlay Systems is given by EXECUTE. All other routines in the Overlay System are either subroutines or co-routines. Resident subroutines may be called by any routine (resident or LINK component); therefore, the names assigned to resident routines must be unique throughout the Overlay System.

2.2 EXTERNAL LINK COMPONENTS

Each LINK consists of one or more subroutines, whose calling will result in the loading of the LINK when non-resident. These subroutines may be called by any routine, resident, or LINK component, and are called External Link Components. The names of External Link Components must also be unique throughout the Overlay System.

2.3 INTERNAL LINK COMPONENTS

A LINK may also contain subroutines which are only to be called from within the LINK. These subroutines are called Internal Link Components, and may be used as internal components of other LINKS. The names of Internal Link Components must be unique only within their LINK.

2.4 BLANK COMMON

Blank COMMON is universal COMMON. It is resident throughout a run and is available to all routines.

2.5 LABELED COMMON

Labeled COMMON blocks declared by resident code are also resident and available to all routines. (This feature has been included to allow a resident initialized block to be available to all routines.)

A labeled COMMON block declared within a LINK with a block name that does not match a resident COMMON block name, is internal to the LINK. It is initialized (if BLOCK DATA or .CBS) or cleared when the LINK is loaded, and overlaid when the LINK is overlaid. The same COMMON block may be initialized by more than one routine as long as all initializing routines are contained within the same link. Using the same name for labeled COMMON blocks in different LINKS does not force an equivalence (see Example 3, page 7-12).

CHAIN provides a command option which causes LABELED COMMON BLOCKS of the same name to be allocated core only once, and to allow elements of LABELED COMMON to be referenced from any co-resident LINK. (Without this option, LABELED COMMON is considered internal to LINKS.) The option abbreviation accepted by CHAIN is "SAC" (Single Allocation of Commons). With the SAC option selected, a LABELED COMMON BLOCK is made a part of the first LINK listed in the Overlay Description that contains a declaration for the COMMON BLOCK.

This option facilitates an overlayable inter-LINK communication area. For example, SUB2, SUB3, and SUB4 each contains a

```
COMMON/COMBLK/A,B,C,
```

statement. With the SAC option and the following Overlay Description, the LABELED COMMON BLOCK "COMBLK" would reside in SUB2, and would be available for communication between SUB3 and SUB4 (which overlay each other).

```
SUB1:SUB2,SUB3
SUB3:SUB4
```

CAUTION: Since there can be references to elements of a LABELED COMMON BLOCK from outside its LINK, a COMMON BLOCK may be overlaid while references to it still exist in other LINKS.

2.6 RESTRICTING COMMON AREAS

CHAIN option "VTC" allows the user to restrict COMMON areas to bank boundaries. This is useful to the VT15 user who builds display files in COMMON, since the VT15 cannot cross bank boundaries directly (i.e., 13-bit addressing). There are two forms of the option and the giving of one cancels out the other form if it was given previously.

"VTC" option without names restricts all common areas to bank boundaries. The "VTC" option is delimited by being the first option in the command string or by a comma on the left, and by a comma or altmode on the right.

"VTC/NAM1,NAM2,...,NAM3/" option with names restricts to bank boundaries only those COMMON areas named (Note: Blank common is .XX). More than one VTC option with names may be given in the command string and all names specified will be restricted. The option is delimited by being the first option in the command string or by a comma on the left, and a slash on the right. The name field within the option is delimited by a slash right after the "VTC" and the slash that terminates the option. Names in the list are separated by commas.

The VTC option will not restrict COMMON areas declared in BLOCK DATA SUBPROGRAMS.

The COMMON area is restricted to bank boundaries even if CHAIN is running in page relocation mode.

The RES option has the same format as the VTC option but it restricts Labeled Common Blocks to Extended Memory (above the bootstrap).

2.7 THE LINK TABLE

Also resident throughout a run is a table with an eleven-word entry for each External Link Component, a one-word buffer, and a transfer vector to an entry point to EXECUTE. This table is called the Link Table. Calls to External Link Components are loaded as JMS's to the appropriate Link Table entry. Transfer of control from the table is dependent on LINK residency (see Appendix A.)

CHAPTER 3
SOURCE OF RELOCATABLE BINARY UNITS

When describing an Overlay System, the names of files are listed as containing either resident code or LINK components. The named files are read from the "User Program Device" (see Chapter 6). Library Routine names may also be used when describing resident code. The Library Routine names used are distinguished from file names by a "Library Indicator". The indicator is a pound sign (#) which may precede, follow, or appear within the name. The indicator is useful when it is desirable to force a Library Routine, which is not called by a resident routine to be included as resident code (for example, to "factor" a Library Routine out of all or many LINKS).

CHAIN accepts library indicators (#) on both internal and external components. The name given must correspond to the GLOBAL name of the routine desired in the library.

CHAPTER 4

LINK ENTRY POINTS AND FILE NAMES

Each External Link Component has only one entry point whose calling will result in the loading of the LINK, if non-resident. This entry point name (GLOBAL symbol definition) must be the same as the filename of the file containing the binary unit of the External Component. The names of files containing either resident code or Internal LINK Components need not correspond to routine names (although normally they would).

The following must be considered to ensure the correct naming of files:

1. For External Components written in FORTRAN, the name of the source file should be the same as the subroutine name in the subroutine statement.

2. For External Components written in assembly language, the name of the source file should be the same as the label on the major entry point (the entry point should be defined as a GLOBAL symbol).

CHAPTER 5
THE RELOCATION PROCESS

Each LINK is relocated and output as a unit in the format described in Appendix C. The resident code, although not formally a LINK, is also relocated and output in LINK format, and is denoted LINK #000.[1]

CHAIN relocates into the XCT file rather than into core to avoid requiring sufficient core for itself and a LINK while it is being constructed. This results in the incorrect recording of 1) transfer vectors to routines not yet relocated (except External Link Components), 2) transfer vectors for elements of COMMON blocks not yet defined, and 3) string code address corrections (Standard Loader Codes 20 and 21). As this information is developed, it is stored in a core resident "Patch Table" and is recorded after all routines of the LINK or resident code have been relocated and output (see Appendix C).

A core resident "Symbol Table" is used for global symbols, COMMON blocks, and COMMON symbols. Before any relocation, a GLOBAL symbol definition entry is made for each External Link Component such that calls to these routines will result in a transfer to the appropriate Link Table entry.

When relocating the resident code, a second Symbol Table entry is made for all names flagged with library indicators (i.e., #). This entry is a dummy GLOBAL symbol reference which is only recognized when searching for unresolved GLOBAL symbol references. It is used to ensure the inclusion of library routines not called by a resident routine.

Whenever a PGR (page) or BKR (bank) command option is specified, CHAIN performs a .USER macro for the <PAG> or <BNK> UIC respectively in .UFDT-1 to ensure that the correct library is searched during relocation.

[1] This standardization is imposed to minimize the size of EXECUTE. The Link Table is also recorded in LINK format and is denoted LINK #777 (actually #377777).

The Relocation Process

After the relocation of each LINK, the Symbol Table is trimmed back to contain only entries made while relocating the resident code.

The Patch Table is constructed in increasing core from the first available register.[1] The Symbol Table is constructed in decreasing core writing over the lower third of CHAIN's code (which is no longer needed). The overlapping of the Symbol and Patch Tables results in a terminal error.

The image of the resident code and each LINK is recorded in the XCT file by relocating and outputting the routines from the indicated files on the user program device. If unresolved GLOBAL symbol references (references to yet unrelocated routine) exist, the user's Library (if existent) and the System Library are scanned, relocating and outputting any routines which contain a GLOBAL symbol definition that matches an unresolved GLOBAL symbol reference, thus resolving them. This Library search continues until all GLOBAL symbol references have been resolved or the libraries have been exhausted. The user's library is searched repeatedly until no GLOBAL symbol references are resolved. Then, the system library is scanned repeatedly until no GLOBALS are resolved. If unresolved GLOBALs still remain, the library search starts again with the user's library. The library search terminates either when no unresolved GLOBALs remain or when a search through both libraries results in no resolved GLOBALs. After each routine has been relocated and output, and if a load map has not been suppressed (NM option), a line will be typed out containing the routine name (unless GM option) and the limits of core the routine will occupy.

CHAIN calculates the number of 400_8 word-blocks needed to store the overlay system, by LINKs, as a core image. This information is stored in the environment indicator in bits Ø through 11 as a right-justified octal number. The number calculated does not include the LINK table (LINK 377777) or the resident code (LINK Ø).

[1] Determined from .SCOM+2

CHAPTER 6
I/O DEVICE ASSIGNMENTS

CHAIN's I/O operations are accomplished via six .DAT slots. The slots and the functions assigned each are:

.DAT -1 System Library - The system library (default .LIBR) is scanned to satisfy unresolved GLOBAL references.

.DAT -2 Command Input - Normally the console terminal, but may be the batch device under Batch Mode.

.DAT -3 Typed output device - Must be the console terminal.

.DAT -4 User program device - All routines listed during command input, which are not indicated as residing in a library, are relocated from this device.

.DAT -5 User's Library - The current user library (default .LIBR5) is scanned to satisfy unresolved GLOBAL symbol references. If a user's Library does not exist, .DAT -5 must be assigned "NONE".

.DAT -6 Relocated output device - The XCT file is written on this device.

CHAIN never has more than two files open at the same time. Whenever two files are open, one is open for input and the other for output. The handler functions are limited to: .INIT, .ENTER, .SEEK, .READ, .WRITE, .WAIT, and .CLOSE (i.e., "B" type handlers may be used).

CHAPTER 7
BUILDING AN OVERLAY STRUCTURE

Before calling CHAIN, the user should be sure that the proper .DAT slot assignments have been made (see 7.8).

CHAIN is called by typing "CHAIN" following the monitor's $ request. When loaded, CHAIN will type its name and version number and make the following requests:

```
NAME XCT FILE
LIST OPTIONS & PARAMETERS
DEFINE RESIDENT CODE
DESCRIBE LINKS & STRUCTURE
```

A response to each request via the command input device is necessary.

7.1 COMMAND INPUT

CHAIN reads commands via the console Teletype or, in Batch mode, via the current Batch device. All input is accepted in logical lines, which consist of one or more physical lines. A carriage return (or card column 81) is used to continue a logical line onto the next physical line. An ALTMODE is used to terminate a logical line. A line (logical) consists of names (file, routine, option, parameter), library indicators (resident code definition only), and break characters (name terminators). Blanks are ignored. Names consist of up to 6 "RAD50" (A to Z, 0 to 9, ".", and "%") characters and a library indicator ("#") which may appear anywhere within or around the name

The characters "=", ",", ":", and "/" are all considered to be "break" characters. Each individual break character is only valid in the appropriate context (see below for each case).

When an error is detected, the entire logical line containing the error is rejected. IOPS ASCII editing (RUBOUT and↑U) apply only to physical lines. A ↑P issued prior to the end of the overlay structure definition will restart CHAIN and will reset all defaults as well as erasing any effects the command input has had. A ↑P issued after the completion of the overlay structure definition will cause control to be returned to the XVM/DOS monitor.

An angle bracket (">") is typed to indicate the beginning of a line. A hyphen ("-") indicates that a line is being continued across more than one physical line.

7.2 CONDITIONAL MESSAGES

CHAIN's error messages attempt to indicate the source of any error. If recovery is possible, either by retyping a line (logical) or by typing ↑P and restarting CHAIN, CHAIN will type an angle bracket at the left margin to indicate that input is requested. If recovery is not possible, CHAIN will exit to XVM/DOS after typing the appropriate error message. A list of CHAIN's messages, their meaning and possible recovery procedures, is contained in Appendix F.

7.3 NAME XCT FILE

A 1- to 6-character filename terminated by an ALTMODE is required. This name is used when requesting EXECUTE to run the overlay system being built.

7.4 LIST OPTIONS AND PARAMETERS

Options and environmental parameters are listed on a (logical) line, separated by commas. An exception to this rule are the RES and VTC switches with arguments, in these cases no comma may follow the "/". A zero length line (ALTMODE only) indicates no options or parameters specified. The following options and parameter abbreviations are recognized by CHAIN.

7.4.1 BKR

Components of the overlay structure being built will be relocated within BANKS (8K) of core. Instructions which reference memory will be assumed to have 13 bits of direct addressability.

7.4.2 BOO:nnK

The system for which the overlay structure is being built will have a bootstrap at nnK (where nn is 24 or 32).

7.4.3 BOT:nnnnn

The overlay structure will be built upward from address nnnnn toward the current bootstrap address. Allocation of extended memory (see nnnK) will be from the first word following the bootstrap toward the current value of the memory size (nnnK).

7.4.4 GM

If a load map is output, names are to be file and global symbol rather than program names.

7.4.5 nnnK

The overlay structure being built will be run on a machine with at least nnnK words of memory. If the value of nnn is greater than 32 then the XVM parameter (see below) will be turned on.

7.4.6 NM

No memory map is to be output.

7.4.7 NOX

The overlay structure being built will be run on a machine without the XM15 option or with the XM15 option disabled (monitor command XVM OFF).

7.4.8 PAL

CHAIN will pause after each link is relocated. This includes the resident code (link 0) and the link table (link 377777). To continue the user must enter ↑P.

7.4.9 PAR

CHAIN will pause after the resident code is relocated. To continue the user must enter ↑P.

7.4.10 PGR

Components of the overlay structure being built will be relocated within PAGES (4K) of core. Instructions which reference memory will be assumed to have 12 bits of direct addressability.

7.4.11 SAC

Named common blocks may be referenced from any link.

7.4.12 RES or RES/naml,nam2,...,namN/

If the RES switch is used without arguments, all uninitialized common blocks will be forced to reside in extended memory. Extended memory is any memory above the current bootstrap address. If the RES switch is used with arguments only those common blocks named in the argument list, plus any uninitialized common blocks specified in the resident code, will be forced to reside in extended memory. All common blocks which are forced to reside in extended memory by the RES switch become part of the resident code. If extended memory is filled and a common block has the RES switch applied to it, CHAIN will terminate rather than place the specified block in low memory. This restriction may be avoided by referencing, perhaps by a dummy reference, the common blocks from the resident code and not using the RES option.

7.4.13 SL:aaaaaa

CHAIN will search the library named "aaaaaa" on .DAT -1 to resolve global references.

7.4.14 SZ

The size of the link table, common blocks, and routines are to be included in the load map. The size is listed following the core limits.

7.4.15 UL:aaaaaa

CHAIN will search the library named "aaaaaa" on .DAT -5 to resolve global references.

7.4.16 VTC or VTC/naml,nam2,...,namN/

If the VTC switch is used without arguments, all common blocks will be forced to reside within a bank (8K) of memory. If the VTC switch is used with arguments, then only those common blocks named will be forced to reside within a bank. This switch is independent of both the SAC and RES switches.

7.4.17 XVM

The overlay structure being built will be run on a machine with the XM15 option enabled (monitor command XVM ON).

7.4.18 Mode Values and Locations

The relocation mode (BKR, PGR), bootstrap address (BOO), and the XVM mode (NOX, XVM) take values from the current state of .SCOM. The initial core allocation mode is top down (BOT changes this). The current memory size (nnnK) is that of the current machine. SAC, RES, and VTC are off. The default user and system libraries are .LIBR5 and .LIBR respectively.

7.5 RESIDENT CODE DEFINITION

The names of files containing relocatable binary units of routines to be resident throughout a run and the names of Library Routines (flagged by library indicators (#) to be resident throughout a run are listed on a line, separated by commas (,). EXECUTE transfers initial control to the entry point of the first resident routine relocated, i.e., the first routine of the first file listed, unless resident code is exclusively Library Routines. The response to "DEFINE RESIDENT CODE" must be at least one name.

7.6 LINK AND STRUCTURE DEFINITION

The overlay structure is described in terms of LINK names. When a LINK is to consist of only one external component, the name of the file containing the external component may be used as the LINK name. However, when a LINK is to consist of more than one external component, the LINK must be named and defined.

7.6.1 Link Definitions

Each LINK definition requires one line of command input consisting of the LINK's name followed by an equal sign (=) followed by the LINK definition.

Building an Overlay Structure

A LINK definition is a list of the names of files which contain the relocatable binary units that comprise the LINK components. The individual filenames listed are separated by commas (,); the two types of LINK components which may be used (external and internal) are separated within the definition by a slash (/). All external LINK component names must be listed before (to the left of) the slash separator; all internal LINK components must be listed after (to the right of) the slash. External LINK components are accepted only from files with names which match the external component name (i.e., GLOBAL symbol definition). In DOS external and internal link components can be retrieved from a library by the use of #, but the GLOBAL name must be used in the command input.

Example:

```
ABC=SUB1,SUB2/SUB3,SUB4
```

In the above example, SUB1 and SUB2 are external components of LINK ABC, and SUB3 and SUB4 are internal components of LINK ABC.

Rules for defining a LINK:

1. A LINK may not be a component of another LINK.
2. The names of the components of a LINK may not be used as LINK names.[1]
3. A filename used in the resident code description cannot be used in a LINK definition.
4. A filename preceding a slash may be used only once.
5. A filename following a slash may be used in other LINK definitions (following a slash).

7.6.2 Overlay Structure Description

An overlay structure is described using the names of defined LINKS, or the names of files containing LINK components and the operators colon (:) and comma (,), under the following set of rules:

[1] When a LINK consists of only one component, the component's filename may be used as the LINK name in the "overlay structure description", but not in a LINK definition; i.e., it is not necessary to define a single component LINK but, if defined, the LINK name cannot be the component name.

Building an Overlay Structure

1. A line is an independent statement processed from left to right.

2. A colon is read "is overlaid by".[1]

3. A comma is read "and".

Example:

```
SUB1:SUB2
SUB2:SUB3,SUB4
```

is interpreted as -- SUB1 is overlaid by (uses the same core as) SUB2, SUB2 is overlaid by SUB3 and SUB4, but SUB3 and SUB4 do not overlay each other.

4. A colon operator may not be used in a line after a comma has been used. This restriction prevents the following ambiguity:

```
SUB2:SUB3,SUB4:SUB5
```

The above line is rejected by CHAIN because it is not clear whether SUB5 overlays SUB3 or SUB4 or both. All four of the following examples are acceptable:

SUB2:SUB3,SUB4 SUB4:SUB5	SUB5 uses the same core as SUB4 but not the same core as SUB3.
SUB2:SUB3,SUB4 SUB3:SUB5	SUB5 uses the same core as SUB3 but not the same core as SUB4.
SUB2:SUB5:SUB3,SUB4	SUB5 uses the same core as SUB3 and SUB4. SUB3 and SUB4 are loaded individually (if non-resident) as called.
LINK=SUB3,SUB4 SUB2:LINK:SUB5	SUB5 uses the same core as SUB3 and SUB4. Both SUB3 and SUB4 are loaded (if non-resident) whenever either is called.

5. A LINK name may appear only once preceding a colon and only once following another colon.

6. If a LINK name is used twice, it must be used following a colon before being used preceding another colon.

7. Several LINKS overlaying each other may be defined in one statement.

[1] A loading order is not implied; just core mapping.

Example:

```
SUB1:SUB2:SUB3,SUB4
```
[1]

NOTE

This is a short method of defining the same overlay structure as in the first example, under rule 3.

Rules 5 and 6, although they may appear restrictive, do not limit the user's description of an overlay structure, but do prevent multiple description of the position of a LINK in an overlay structure. A LINK may be both overlaid and overlaying, and it may not be possible or convenient to describe both conditions by using the LINK name only once, as follows:

```
SUB1:SUB2:SUB3        SUB2 is overlaying SUB1 and is
                      overlaid by SUB3
```

Therefore, when a LINK is both overlaying and overlaid, its LINK name may be used twice, but the LINK(s) overlaid by it must be described before the LINK(s) by which it is overlaid.

Example:

```
SUB2:SUB3,SUB4        SUB3 overlays SUB2
SUB3:SUB5             SUB3 is overlaid by SUB5
```

NOTE

The description of an overlay structure only defines a desired core mapping; i.e., stating that SUB1 is overlaid by SUB2 means that both are to be relocated to the same core and cannot co-reside, but does not imply that SUB1 must be called before SUB2. There is no imposed order in which routines must be called, nor is there restriction of the routines callable by any routine.

7.7 TERMINATION OF COMMAND INPUT

When the last line of overlay structure description has been input,

[1] A loading order is not implied; just core mapping.

command input is terminated by a zero length line (ALTMODE only). At this time, relocation of the resident code will begin and ↑P restart will no longer be possible because the command input code will be written over by the Symbol Table.

Command input may also be terminated after describing only resident code by typing an ALTMODE in response to the DESCRIBE LINKS & STRUCTURE request. This allows CHAIN & EXECUTE to be a useful alternative to the LINKING LOADER when the routines to be loaded and the loader cannot fit in core together, or when a job is to be run often and it is desirable to be able to load it with a simple command. Viz., E JOBNAM.

7.8 EXAMPLES

The following source code is used in examples 1 through 6.

```
C    FILE: MAIN
C    MAIN PROGRAM
C
     COMMON A,B
     CALL SUB1(4,P)
     CALL SUB2(2,0)
     IF (P-Q) 40,10,40
10   IF (A-P) 40,20,40
20   IF (B-Q) 40,30,40
30   IF (P+Q) 50,40,50
40   PAUSE
50   STOP
     END
C    FILE: SUB1
C
     SUBROUTINE SUB1 (N,X)
     DIMENSION C(4)
     COMMON A/XXX/C
     X=C(N)
     A=C(N)
     RETURN
     END
C    FILE: SUB2
C
     SUBROUTINE SUB2(N,X)
     DIMENSION C(4)
     COMMON A,B/XXX/C
     X=C(N)*C(N)
     B=C(N)*C(N)
     RETURN
     END
C    FILE: BDATA
C
     BLOCK DATA
     DIMENSION C(4)
     COMMON /XXX/C
     DATA C/1.0.2.0.3.0.4.0/
     END
```

NOTE

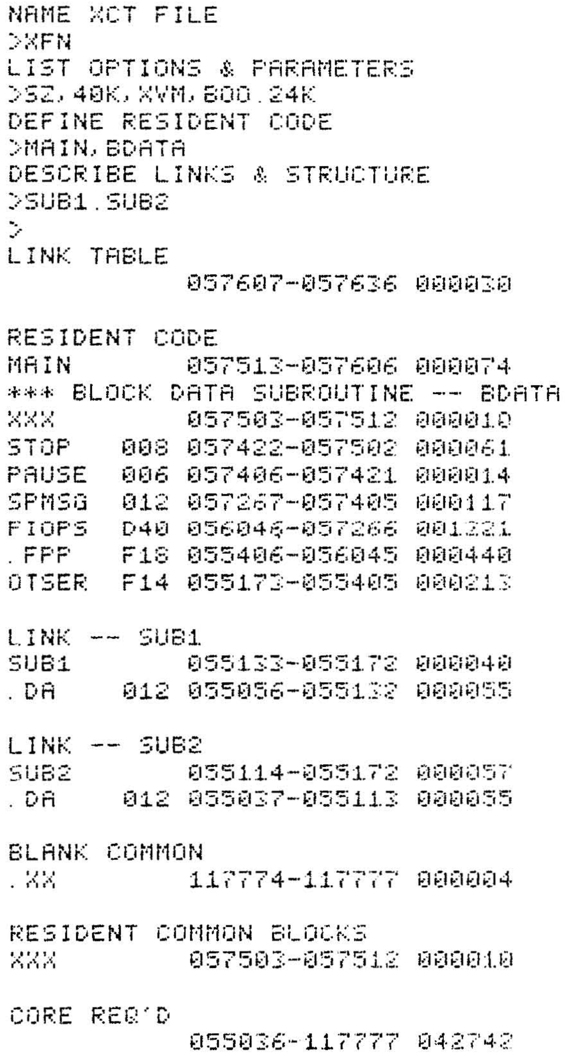
A complete list of error messages produced by CHAIN is given in Appendix F.

Building an Overlay Structure

EXAMPLE 1

The angle bracket (">") in the left margin indicates the beginning of a logical line of command input; the remainder of the line is keyed in by the user and terminated by an ALTMODE. ALTMODE is a non-printing character. Both the minimum and maximum core locations occupied by the link table, a routine, or a common block, are output in the load map as octal constants. Also note that all common blocks associated with the resident code are listed both in the link defining the particular common block and in the section of the load map named "RESIDENT COMMON BLOCKS".

Common block "XXX" is initialized via the block data subroutine "BDATA", and is available to both SUB1 and SUB2.



```
NAME XCT FILE
>XFN
LIST OPTIONS & PARAMETERS
>SZ,40K,XVM,BOO.24K
DEFINE RESIDENT CODE
>MAIN,BDATA
DESCRIBE LINKS & STRUCTURE
>SUB1.SUB2
>
LINK TABLE
                057607-057636 000030

RESIDENT CODE
MAIN            057513-057606 000074
*** BLOCK DATA SUBROUTINE -- BDATA
XXX             057503-057512 000010
STOP      008 057422-057502 000061
PAUSE     006 057406-057421 000014
SPMSG     012 057267-057405 000117
FIOPS     D40 056046-057266 001221
.FPP      F18 055406-056045 000440
OTSER     F14 055173-055405 000213

LINK -- SUB1
SUB1            055133-055172 000040
.DA       012 055056-055132 000055

LINK -- SUB2
SUB2            055114-055172 000057
.DA       012 055037-055113 000055

BLANK COMMON
.XX             117774-117777 000004

RESIDENT COMMON BLOCKS
XXX             057503-057512 000010

CORE REQ'D
                055036-117777 042742
```

EXAMPLE 2

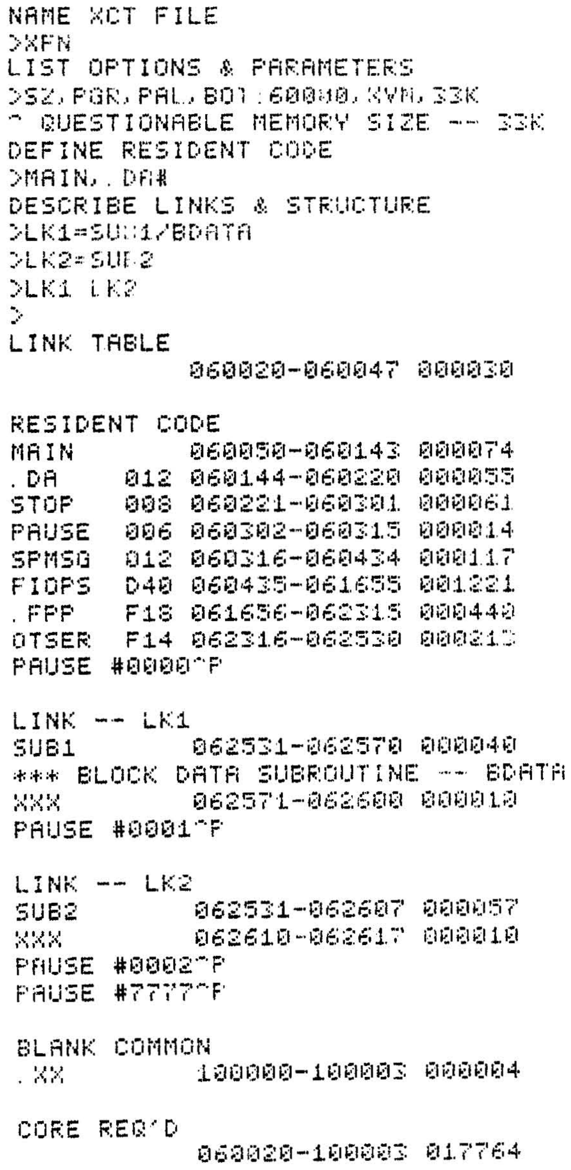
Both links LK1 and LK2 contain the block data subroutine BDATA. Note that the error message produced by the memory size "28K" is warning only, and that CHAIN built the overlay structure properly.

```
NAME XCT FILE
>XFN
LIST OPTIONS & PARAMETERS
>SZ,BOO.24K,NOX,28K,PGR
^ QUESTIONABLE MEMORY SIZE -- 28K
DEFINE RESIDENT CODE
>MAIN
DESCRIBE LINKS & STRUCTURE
>LK1=SUB1/BDATA
>LK2=SUB2/BDATA
>LK1:LK2
>
LINK TABLE
            057607-057636 000030

RESIDENT CODE
MAIN        057513-057606 000074
STOP    006 057432-057512 000061
PAUSE   006 057416-057431 000014
SPMSG   012 057277-057415 000117
FIOPS   D40 056056-057276 001221
.FPP    F18 055416-056055 000440
OTSER   F14 055203-055415 000213

LINK -- LK1
SUB1        055143-055202 000040
*** BLOCK DATA SUBROUTINE -- BDATA
XXX         055133-055142 000010
.DA     012 055056-055132 000055

LINK -- LK2
SUB2        055124-055202 000057
*** BLOCK DATA SUBROUTINE -- BDATA
XXX         055114-055123 000010
.DA     012 055037-055113 000055

BLANK COMMON
.XX         067774-067777 000004

CORE REQ'D
            055036-067777 012742
```

EXAMPLE 3

The overlay system being built will be relocated from location 60000 upward. Note the use of odd memory sizes and the use of the library indicator on global .DA to factor that routine out of the non-resident portion of the overlay structure.

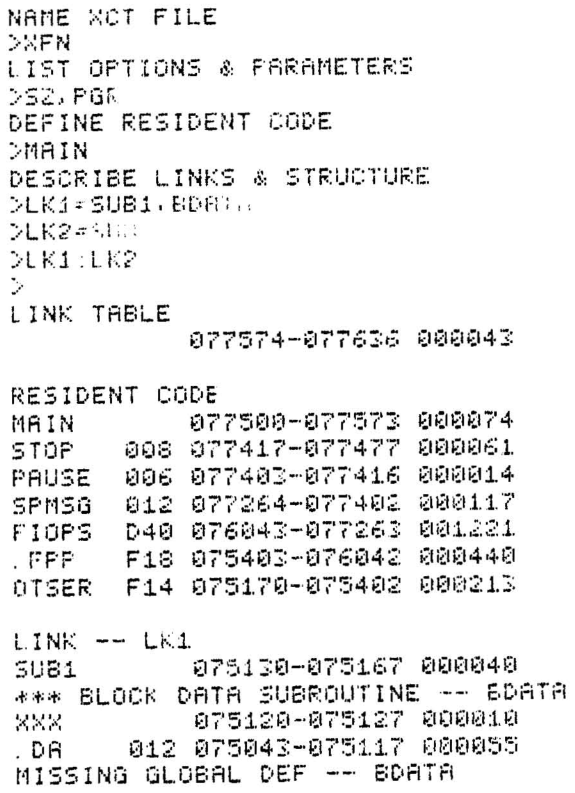
Only link LK1 has an initialized labeled common block XXX, the COMMON declaration in SUB2 has resulted in the allocation of core within link LK2 for an uninitialized labeled common block XXX.



```
NAME XCT FILE
>XFN
LIST OPTIONS & PARAMETERS
>SZ,PGR,PAL,BOT:60000,XVM,33K
^ QUESTIONABLE MEMORY SIZE -- 33K
DEFINE RESIDENT CODE
>MAIN,.DA#
DESCRIBE LINKS & STRUCTURE
>LK1=SUB1/BDATA
>LK2=SUB2
>LK1 LK2
>
LINK TABLE
                 060020-060047 000030

RESIDENT CODE
MAIN             060050-060143 000074
.DA        012   060144-060220 000055
STOP       008   060221-060301 000061
PAUSE      006   060302-060315 000014
SPMSG      012   060316-060434 000117
FIOPS      D40   060435-061655 001221
.FPP       F18   061656-062315 000440
OTSER      F14   062316-062530 000213
PAUSE #0000^P

LINK -- LK1
SUB1             062531-062570 000040
*** BLOCK DATA SUBROUTINE -- BDATA
XXX              062571-062600 000010
PAUSE #0001^P

LINK -- LK2
SUB2             062531-062607 000057
XXX              062610-062617 000010
PAUSE #0002^P
PAUSE #7777^P

BLANK COMMON
.XX              100000-100003 000004

CORE REQ'D
                 060020-100003 017764
```

EXAMPLE 4

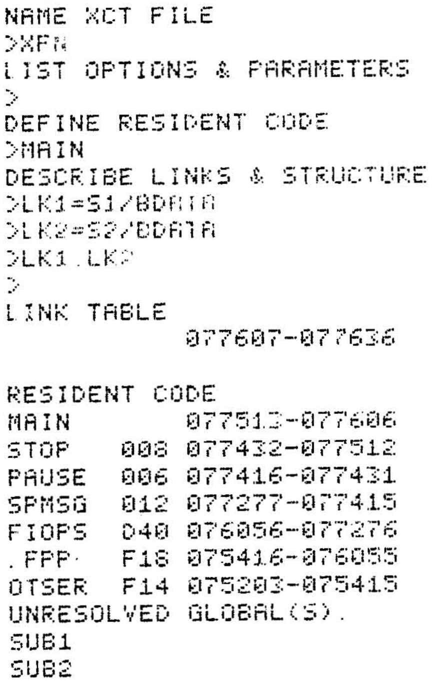
Block Data subroutines do not have an entry point (filename defined as a global symbol), and therefore cannot be an external link component.



```
NAME XCT FILE
>XFN
LIST OPTIONS & PARAMETERS
>S2,PGR
DEFINE RESIDENT CODE
>MAIN
DESCRIBE LINKS & STRUCTURE
>LK1=SUB1,BDATA
>LK2=XXX
>LK1:LK2
>
LINK TABLE
              077574-077636 000043

RESIDENT CODE
MAIN          077500-077573 000074
STOP     008 077417-077477 000061
PAUSE    006 077403-077416 000014
SPMSG    012 077264-077402 000117
FIOPS    D40 076043-077263 001221
.FPP     F18 075403-076042 000440
OTSER    F14 075170-075402 000213

LINK -- LK1
SUB1          075130-075167 000040
*** BLOCK DATA SUBROUTINE -- BDATA
XXX           075120-075127 000010
.DA      012 075043-075117 000055
MISSING GLOBAL DEF -- BDATA
```

EXAMPLE 5

SUB1 and SUB2 have been misspelled in the link definitions. Since SUB1 and SUB2 are called by the main program, global references for the symbols SUB1 and SUB2 are made. Neither SUB1 nor SUB2 are defined in a library, nor do they appear as link components. Therefore, the attempt to resolve the global references fails.



```
NAME XCT FILE
>XFR
LIST OPTIONS & PARAMETERS
>
DEFINE RESIDENT CODE
>MAIN
DESCRIBE LINKS & STRUCTURE
>LK1=S1/BDATA
>LK2=S2/BDATA
>LK1.LK2
>
LINK TABLE
               077607-077636

RESIDENT CODE
MAIN           077513-077606
STOP     008   077432-077512
PAUSE    006   077416-077431
SPMSG    012   077277-077415
FIOPS    040   076056-077276
.FPP     F18   075416-076055
OTSER    F14   075203-075415
UNRESOLVED GLOBAL(S).
SUB1
SUB2
```

EXAMPLE 6

Miscellaneous recoverable errors.

```
NAME XCT FILE
>XFN45
^ NAME LENGTH ERR
>XFN
LIST OPTIONS & PARAMETERS
>YL.USERL.BKR..1
^ IMPROPER BREAK CHAR -- .
>UL.USERL,BKR,XVN
^ UNRECOGNIZED SYMBOL -- XVN
>US.USERL,BKR,XVM
^ IMPROPER BREAK CHAR -- .
>UL.USERL,BKR,XVM
DEFINE RESIDENT CODE
>MAIN/BDATA
^ IMPROPER BREAK CHAR --
>MAIN,BDATA
DESCRIBE LINKS & STRUCTURE
>LK1#=SUB1/BDATA
^ LIB IND ON LINK NAME -- LK1
>LK1=SUB1/BDATA
^ INTERNAL NAME USED PRV -- BDATA
>LK1=SUB1/LBDATA
>LK2=SUB1/LBDATA
^ EXTERNAL NAME USED PRV -- SUB1
>LD2=SUB2/LBDATA
>LK1.LD2:MAIN
^ RES ROUTINE NAME USED AS LINK NAME -- MAIN
>LK1.LD2
>LK1.SUB3
^ NAME USED LEFT OF COLON TWICE -- LK1
>LD2.SUB3
>
LINK TABLE
             077574-077636

RESIDENT CODE
MAIN         077500-077573
*** BLOCK DATA SUBROUTINE -- BDATA
XXX          077470-077477
  .               .                    .

  .               .                    .

  .               .                    .
```

CHAPTER 8
EXECUTION OF AN OVERLAY SYSTEM

The program EXECUTE controls the execution of an overlay structure built with CHAIN.

Before calling EXECUTE, the user should be sure that the "XCT" file is on the device assigned to .DAT slot -4, and that the .DAT slots required by the overlay system are properly assigned.

EXECUTE is called by typing "EXECUTE" or "E" followed by the XCT filename, in response to XVM/DOS's $ request. The command and filename must be separated by at least one space.

The system loader (.SYSLD) will open and process the XCT file to determine the overlay system's .DAT slot requirements, load EXECUTE and any required I/O handlers, and transfer control to EXECUTE.

EXECUTE will announce itself and then open the XCT file named in the original "E" command, load the Link Table (if any) and the resident code, and transfer control to the main program. The order in which the links are loaded is a function of the overlay system being executed, a link is loaded whenever control is transferred and the link is non-resident.

EXECUTE will detect and indicate the following errors:

1. CAN'T FIT

   The overlay system will not fit in the available core.

2. CAN'T RUN

   The overlay system was built with "BANK" addressing and "PAGE" is on or vice versa.

3. WRONG MODE

   The overlay system was built with "XVM" off and "XVM" is on or vice versa.

4. WRONG BOOT

   The overlay system was built for a bootstrap address of 32K and the current bootstrap address is 24K or vice versa.

5. READ ERR

   A portion of the XCT file is unreadable.

EXECUTE reads the XCT file, from .DAT -4, once prior to entering the main code in order to set up the Link Tran Table. The Link Tran Table consists of a two word entry for each link in the overlay structure. The first word is the block number within which the link begins and the second word is the offset within the block to the beginning of the link. After this pass is made, the XCT file is closed and not used again.

The XCT file is not allowed to reside on a non-block device, i.e., paper tape, magtape, cards, etc.

APPENDIX A
THE LINK TABLE

The Link Table is the mechanism by which EXECUTE is able to field calls to non-resident links. The Link Table consists of a transfer vector to and entry point in EXECUTE (EXUTV), a one-word buffer (BUF) for the AC, and an eleven word entry for each link component.

When CHAIN built the Link Table it created a JMS* to the first word of the Link Table for each external link component. At the time the Link Table is loaded by EXECUTE the proper word in the Link Table is loaded with the address of the entry point to EXECUTE.

Non-resident Link Table Entry Format

```
0                   /ENTRY POINT
DAC BUF             /SAVE AC
JMS* EXUTV          /CALL EXECUTE
DAC* .+3            /STORE RETURN ADDRESS
LAC BUF             /RESTORE AC.
JMP* .+2            /ENTER CALLED ROUTINE
ENTRY               /CALLED ROUTINE ENTRY ADDRESS
ENTRY+1             /CALLED ROUTINE ENTRY ADDRESS+1
LINK NUMBER         /IDENTIFIER FOR EXECUTE
BASE ADDRESS        /LOWEST ADDRESS OF LINK
TOP ADDRESS         /HIGHEST ADDRESS OF LINK.
```

Thus, calling a non-resident subroutine transfers control to EXECUTE pointing to the fourth word of the Link Table entry for the external link component called.

EXECUTE then performs the following functions:

1. Fetches the link number from the link table, clears memory from the lowest address to the highest address of the new link and loads the link containing the called routine.

2. Places a LAC .-2 instruction in the third word of the Link Table entry for each component of the link just loaded.

3. Places a JMS* EXUTV instruction in the third word of the Link Table entry for each component of each link overlayed by the link just loaded.

4. Transfers control to the third word of the Link Table entry for the called routine, which simulates the original call.

While the link is resident, further calls of external components result in a transfer from the Link Table directly to the subroutine.

APPENDIX B
THE XCT FILE

The XCT file is actually two files, one with an extension of "XCT" and the other with an extension of "XCU". Both have the filename given by the user in response to CHAIN's "NAME XCT FILE" request.

CHAIN writes the XCU file first. It contains the resident code (link 0) and all of the actual links numbered sequentially from one. The last information output is the XCT file which contains the Link Table, parameters required by EXECUTE, and the .IODEV information required by .SYSLD. The XCT file is written in the standard link format (see next appendix) and is designated link 377777 (octal). This allows EXECUTE to load the Link Table by simply loading link 377777.

The XCT file is opened and read by .SYSLD (to retrieve the .IODEV information) and by EXECUTE before the XCU file is opened. Thus, by not including the XCT file information at the end of the XCU file, two passes to the end of a file, each time EXECUTE is loaded, have been eliminated.

APPENDIX C
XCT FILE RECORD FORMATS

C.1 TYPE 1 RECORDS

C.1.1 Link ID (Other than Link No. 377777)

IOPS Record Length (word 1)
IOPS Checksum (word 2)
"1" (octal) In the higher order 3 bits of the word
(1xxxxx). xxxxx is the link number (word 3)

C.1.2 Link No. 377777

One special case of a type 1 record exists. Since certain information must be conveyed to both .SYSLD (for allocation and loading of I/O handlers) and EXECUTE (for proper placement of the links and common blocks) the type 1 record with a link number of 377777 (octal) has been reserved for this type of information. The record format for this particular type of ID record is as follows. Each word is described and, when necessary, the bits of the word are also described.

IOPS Record length (word 1)
IOPS Checksum (word 2)
Record Type (1xxxxx (octal)) (word 3)
Link Number (377777 (octal)) (word 4)
First Address of Blank Common (word 5)
Resident Code Entry Point (word 6)
First Address of the Link Table (word 7)
Last Address of the Link Table (word 8)
Lowest Address Used by the Overlay Structure (word 9)
Highest Address Used by the Overlay Structure (word 10)
Operating Environment Indicators (word 11)

Bits 0-11 -- number of blocks in the XCU file.

Bit 12 -- unused

Bit 13 -- Operating system indicator

0 implies that the overlay structure was built under DOS-15.

1 implies that the overlay structure was built under XVM/DOS.

4. Bit 14 -- Bootstrap location.

   Bit 14=0 implies that the bootstrap resides at 57636 (octal).

   Bit 14=1 implies that the bootstrap resides at 77636 (octal).

5. Bit 15 -- XVM mode bit

   Bit 15=0 implies that the XM15 option should be disabled to run the overlay structure.

   Bit 15=1 implies that the XM15 option should be enabled to run the overlay structure.

6. Bit 16 -- Addressing mode

   Bit 16=0 implies that the overlay structure will run in PAGE mode.

   Bit 16=1 implies that the overlay structure will run in BANK mode.

7. Bit 17 -- reserved (currently unused)

.DAT Slots -18 through -1 (word 12)
.DAT Slots -1 through -17 (word 13)
.DAT Slots -18 through -35 (word 14)
.DAT Slots -36 through -53 (word 15)

Extended Memory Limits (word 16):

> If the contents of this word are positive and non-zero then .SCOM+76 (the lower limit of extended memory) is set to this value.
>
> If the contents of this word are zero then .SCOM+76 and .SCOM+77 (the lower and higher limits of extended memory) are left alone and not initialized.
>
> If the contents of this word are negative then .SCOM+76 (the lower limit of extended memory) is set to the two's complement of this word.

RESERVED -- Currently unused (word 17)

NOTE

> When link 377777 (octal) is loaded by EXECUTE, the Link Table is loaded into memory. When link 000000 is loaded the resident code portion of the overlay structure is loaded into memory.

C.2 TYPE 2 RECORDS (CONTIGUOUS MEMORY INITIALIZATION)

IOPS Record Length (word 1)

IOPS Checksum (word 2)

Record Identification - (2) in the high order three bits (0 to 2) of the word, followed by the number of contiguous words to be initialized. The word format is "2XXXXX" where 2 is the record type and XXXXX is the number. (word 3)

Starting Address of the Memory Area to be Initialized (word 4)

Number of Words to be Stored in Memory - XXXXX number of words to be stored in memory starting at the specified address given in the previous word and incrementing that address by one for each word stored. (word 5)

C.3 TYPE 3 RECORD (PATCH ADDRESS)

IOPS Record Length (word 1)

IOPS Checksum (word 2)

Record type identifier (3) in the high order three bits (0 to 2) of the word followed by the total number of patches to be made for this link of the overlay structure. For example, "3XXXXX" where 3 is the record type and XXXXX is the number of patches. (word 3)

Number of patches contained in the current input record. (word 4)

Patch records at two words per record - Patch descriptor in the high order three bits (0 to 2) of the word followed by the address, i.e., "NXXXXX" where XXXXX is the address to be patched and N is the type of patch to perform. (word 5)

N=0 implies the second word of the patch record is to be stored at address XXXXX.

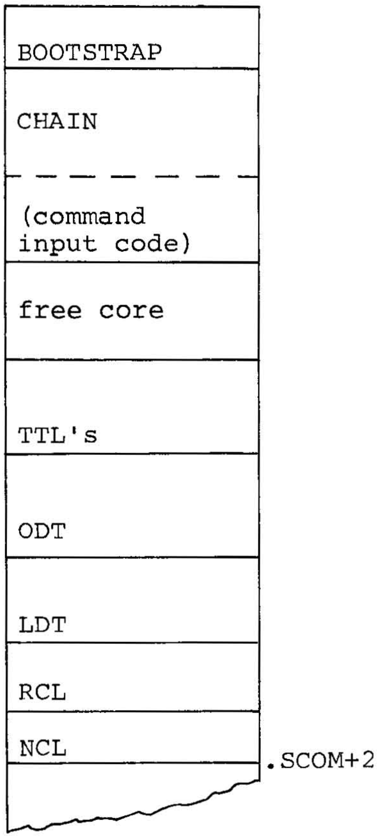
N=1 implies that the second word of the patch record is to be stored in the address field (12 or 13 bits depending on the type of addressing) of the instruction located at XXXXX.

N=2 implies that the base address of blank common is to be added to the second word of the patch record and the result stored at XXXXX.

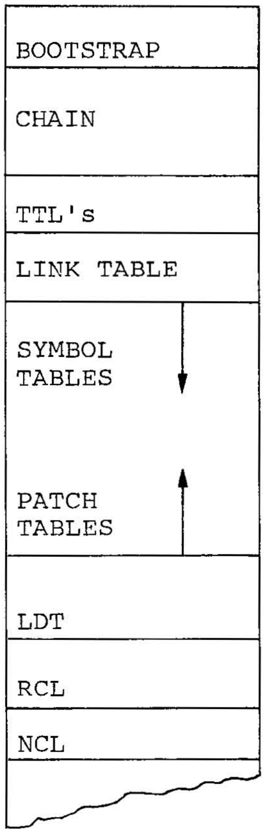
APPENDIX D
CORE ALLOCATION DURING CHAIN

Core map #1 is representative of CHAIN during the link and overlay description entry, prior to the beginning of relocation of the links. Core map #2 is representative of CHAIN during the relocation of the resident code and the links.

#1



#2

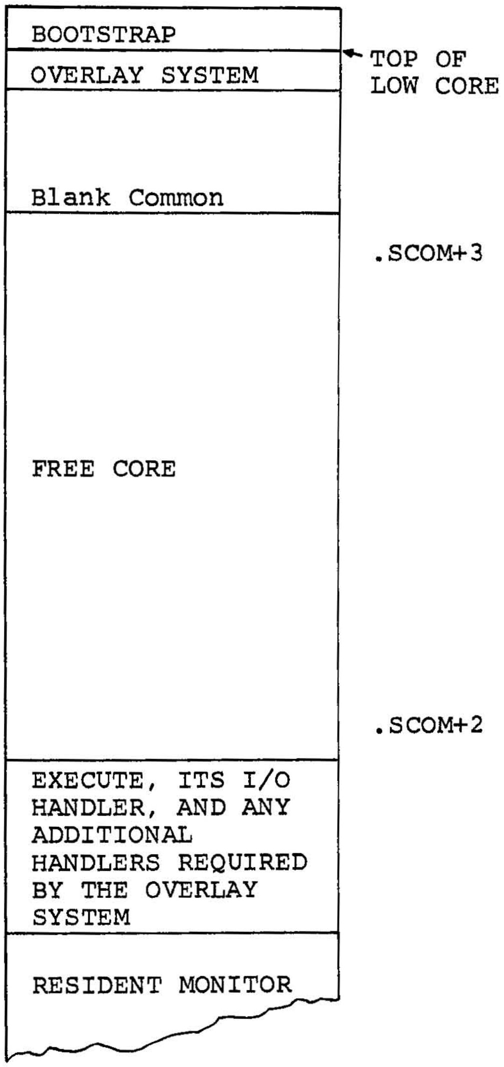


The core used the NCL (Named Common List, generated by the use of the VTC and RES options), the RCL (Resident Code List), and the LDT (Link Definition Table), and ODT (Overlay Description Table), and the TTL's (Trunk-to-Twig Lists) is shown here to help when reading the program listing. This information is not necessary for the proper usage of CHAIN.

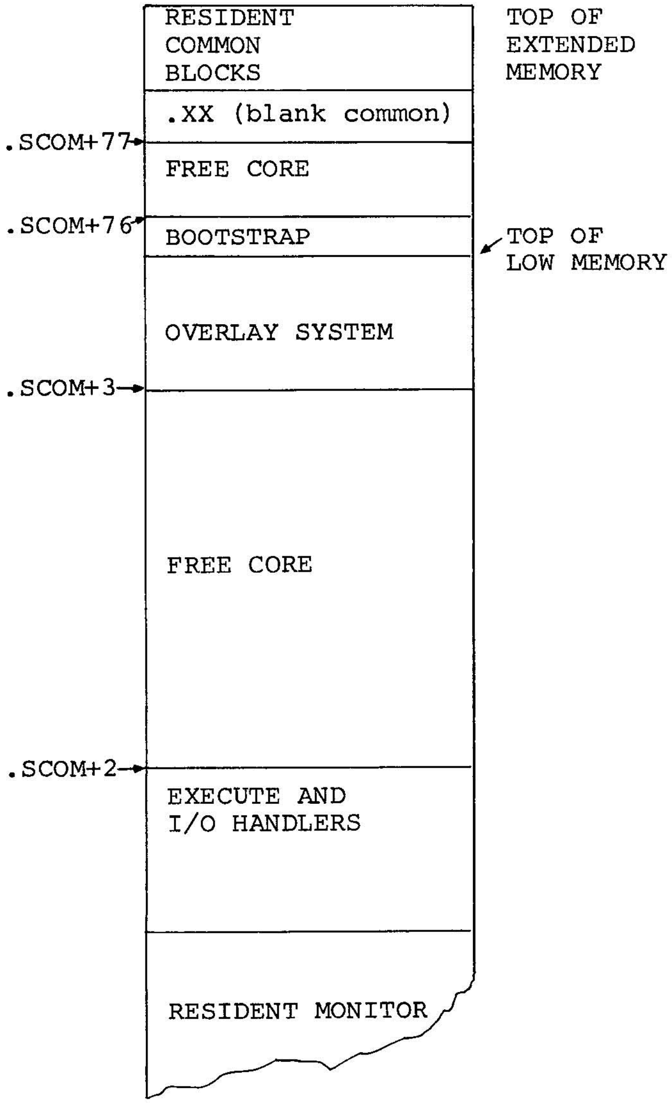
APPENDIX E

CORE ALLOCATION DURING EXECUTE

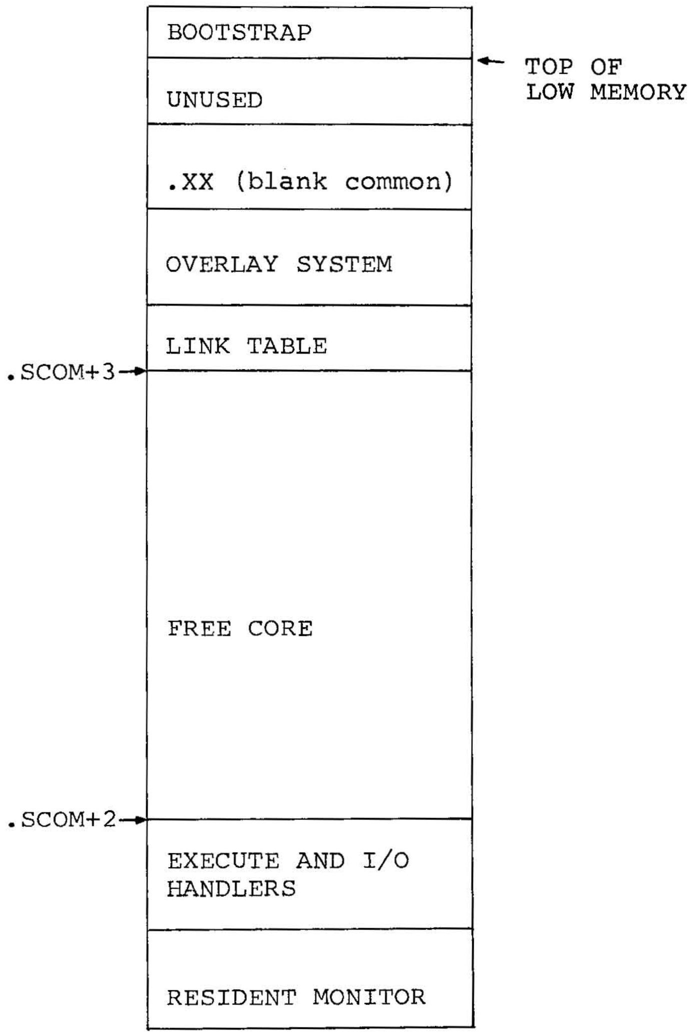
E.1 TOP-DOWN ALLOCATION WITH NO EXTENDED MEMORY



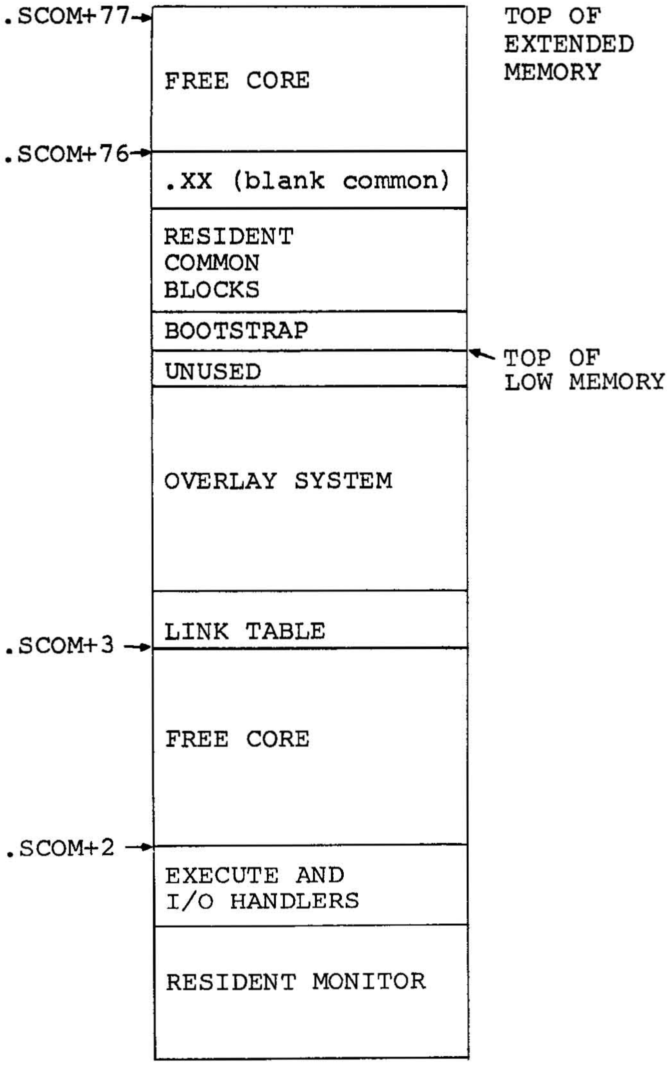
E.2 TOP-DOWN ALLOCATION WITH EXTENDED MEMORY



E.3 BOTTOM-UP ALLOCATION WITH NO EXTENDED MEMORY



E.4 BOTTOM-UP ALLOCATION WITH EXTENDED MEMORY



APPENDIX F

ALPHABETICAL LIST OF CHAIN MESSAGES

The following contractions and symbols are used throughout this appendix:

RES	-- Resident
PRV	-- Previous
DEF	-- Definition
LIB	-- Library
IND	-- Indicator
BLK	-- Block
ERR	-- Error
ABS	-- Absolute
PROG	-- Program
rtnnam	-- Routine Name
blknam	-- Common Block Name
glbsym	-- External Global Symbol
symbol	-- Any of the above
brkchr	-- Break Character
lnknam	-- Link Name

CHAIN Error Messages

ABS PROG

MEANING: An element of code being CHAINed is not a relocatable binary module. CHAIN is terminated and control is returned to XVM/DOS.

RECOVERY: Reassemble the module in question as relocatable binary and rerun CHAIN.

ABSOLUTE LOAD ADDRESS

MEANING: A module has been read which has an absolute starting address. CHAIN is unable to relocate the module and control is returned to XVM/DOS.

RECOVERY: Reassemble the module in question and rerun CHAIN.

BLANK COMMON

MEANING: CHAIN is about to print the top and bottom limits of Blank Common.

***BLOCK DATA SUBROUTINE -- rtnnam

MEANING: CHAIN has encountered a block data subroutine with the name "rtnnam".

CHAIN XVM VnAxxx

MEANING: CHAIN is announcing its presence prior to accepting user commands. This message will appear every time ↑P is used to restart CHAIN during the options processing section.

↑COLON MUST FOLLOW FIRST LINK NAME

MEANING: The user has entered an overlay description in which the first break character is not a ":". The current input line is ignored.

RECOVERY: Reenter the line correctly.

COMMON BLOCK TOO BIG -- blknam

MEANING: The user has attempted to allocate a common block larger than 77777 (octal, 32K-1) in length. The name of the block in question appears in the message. CHAIN terminates and control is returned to XVM/DOS.

RECOVERY: Correct the error in the appropriate module, recompile or reassemble it, and rerun CHAIN.

↑COMPONENT NAME USED AS LINK NAME

MEANING: An element of an overlay description not specified in the link description has the same name as a previously defined link. The current input line is ignored.

RECOVERY: Correct the error by either redefining the previously entered links to avoid using the component name or rename and recompile the appropriate overlay component to avoid using the same link name.

CORE REQ'D

MEANING: CHAIN is about to list the upper and lower core limits of the overlay structure just built.

DEFINE RESIDENT CODE

MEANING: CHAIN is requesting the user to enter those modules which are to be permanently resident throughout the running of the overlay structure to be built. The input format has been discussed in previous sections.

DESCRIBE LINKS AND STRUCTURE

MEANING: CHAIN is requesting the user to enter the link descriptions and the overlay structure to be built. Tht link descriptions are entered first followed by the overlay descriptions. The format of the link and overlay descriptions has been discussed earlier.

DUPLICATE GLOBAL DEF -- glbsym

MEANING: The symbol printed with the message has appeared at least once within the link as an internal global symbol name. CHAIN terminates and control is returned to XVM/DOS.

RECOVERY: The user must correct the error in the appropriate module, reassemble or recompile the module, and rerun CHAIN.

↑EXTERNAL NAME USED PRV

MEANING: An external module name has appeared in more than one link. The current input line is ignored.

RECOVERY: Enter the corrected line and continue with the definition of links and overlays.

Alphabetical List of Chain Messages

GLOBAL SYMBOL TOO BIG -- symbol

MEANING: The user has attempted to reference an area beyond the scope of the given symbol. This error usually occurs when utilizing the common block definition facilities of MACRO XVM. The module within which the error occurs will be the last module displayed in the memory map.

RECOVERY: Correct the appropriate module, reassemble or recompile the module, and rerun CHAIN.

HIGH CORE OVERFLOW -- symbol

MEANING: The given symbol would not fit in high memory (above 32K) and was required to be there, i.e., it was the target of the RES option. CHAIN terminates and control is returned to XVM/DOS.

RECOVERY: Rerun CHAIN with the appropriate common block specified without the RES option being applied.

ILLEGAL ATTEMPT TO INITIALIZE COMMON BLOCK -- blknam

MEANING: The user has attempted to perform one of the following actions:

1. Attempted to initialize a common block with the RES switch applied to it.
2. Attempted to initialize a common block associated with the "resident" code from a link. Common blocks become part of the resident code if either the SAC option or the RES option applies to them.

CHAIN terminates and control is returned to XVM/DOS.

RECOVERY: Alter the appropriate module to either not initialize the common block in question or initialize a different one. Reassemble or recompile the module and rerun CHAIN.

Alphabetical List of Chain Messages

ILLEGAL LOADER CODE

MEANING: CHAIN has detected an unrecognizable loader code. CHAIN terminates and control is returned to XVM/DOS.

RECOVERY: Verify that all modules are binary modules. If all modules are good then reassemble appropriate modules and rerun CHAIN.

↑IMPROPER BREAK CHARACTER -- brkchr

MEANING: The character displayed is not a valid character within the context of the option selected, i.e., using a colon inside a RES or VTC list. The current line is ignored.

RECOVERY: Enter the corrected line and proceed with CHAIN.

↑INTERNAL NAME REPEATED IN LINE

MEANING: The user has used an internal name as a symbol in a previous part of the internal specification of the link. The current line is ignored.

RECOVERY: Enter the correct line and continue with CHAIN.

↑INTERNAL NAME USED PRV

MEANING: The user has entered an internal component name which is the same as a previosuly defined external component. The current input line is ignored.

RECOVERY: Enter the correct line or restart CHAIN and redefine the link and overlay descriptions properly.

LABELED COMMON BLK SIZE ERR -- blknam

MEANING: The user has attempted to change the size of the specified common block after it has been defined. The size change attempted was to make it larger. CHAIN terminates and control is returned to XVM/DOS.

RECOVERY: Change the appropriate common block definition, reassemble or recompile the changed module, and rerun CHAIN.

Alphabetical List of Chain Messages

↑LIB IND ON LINK NAME

MEANING: The user has incorrectly applied the library indicator (#) to a link name. The current input line is ignored.

RECOVERY: Enter the correct line and continue with CHAIN.

LINK -- lnknam

MEANING: CHAIN is announcing the beginning of the relocation of a new link.

↑LINK DEF WITHIN OVERLAY DESCRIPTION

MEANING: The user has entered a new link definition after the first overlay description has been processed. The current input line is ignored.

RECOVERY: If the link definition is unnecessary the user may simply continue with the overlay descriptions. If the link definition is necessary then the user must restart CHAIN and reenter all previous data along with the new link definition.

↑LINK NAME USED PRV

MEANING: The user has used a link name which was used previously. The current input line is ignored.

RECOVERY: Enter the line with a new link name.

LINK TABLE

MEANING: CHAIN is about to print the lower and upper bounds of the link table necessary to the proper execution of the overlay structure to be built.

LIST OPTIONS AND PARAMETERS

MEANING: CHAIN is requesting the user to enter a list of processing options which will define the environment in which the overlay structure will run.

LOW CORE OVERFLOW

MEANING: The executable code portion, including initialized common blocks, of the overlay structure defined by the user will not fit into the available core. CHAIN terminates and control is returned to XVM/DOS.

RECOVERY: The user has two options. The first is to use extended memory (above 32K), if possible. The second is to redesign the overlay structure in an attempt to reduce the core limits necessary to run the program.

MISSING GLOBAL DEF -- glbsym

MEANING: The symbol "glbsym" was specified as a link component but was not resolved during the relocation process. CHAIN terminates and control is returned to XVM/DOS.

RECOVERY: Make sure that the appropriate module has the correct internal .GLOBL specifications. Modify the module, reassemble or recompile it, and rerun CHAIN.

↑MORE THAN ONE LINK OVERLAYED

MEANING: The user has entered an overlay description which requires that one link be overlayed by more than one link. The current input line is ignored.

RECOVERY: The user may either correct the line if it is in error or restart CHAIN and redefine the link and overlay descriptions in order to avoid the error.

↑NAME LENGTH ERR

MEANING: The user has entered a name which is either null (consisting of no characters) or too long (consisting of more than six characters). The current input line is ignored.

RECOVERY: Enter the correct line and continue with CHAIN.

Alphabetical List of Chain Messages

↑NAME RIGHT OF COLON USED PRV

MEANING: The user has entered an overlay description which requires one link to overlay more than one other link at a time. For example, "L1:L2" and "L3:L2" would cause the error. The current input line is ignored.

RECOVERY: Enter the correct line or restart CHAIN and redefine the link properly.

↑NAME USED LEFT OF COLON TWICE

MEANING: The user has entered an overlay description which would require one link to be overlayed by multiple links. The current input line is ignored.

RECOVERY: Enter the correct line if possible or restart CHAIN and redefine the link and overlay descriptions appropriately.

↑NAME USED MORE THAN ONCE

MEANING: The user has requested the same module twice when defining the resident code. The current input line is ignored.

RECOVERY: Enter the correct resident code definition and continue with CHAIN.

↑NAME USED MORE THAN TWICE

MEANING: The user has entered an overlay description in which the following construct has been detected "...:L1:..." and the link L1 also appears elsewhere in the overlay description. The current input line is ignored.

RECOVERY: Enter the correct line if possible or restart CHAIN and redefine the links and overlays so as to avoid the error.

NAME XCT FILE

MEANING: CHAIN is requesting the user to enter an output file name. The file named will become the XCT and XCU files output by CHAIN.

Alphabetical List of Chain Messages

NON-FILE ORIENTED INPUT DEVICE

MEANING: One of CHAIN's input .DAT slots is assigned to a non-directoried device. CHAIN terminates and control is returned to XVM/DOS.

RECOVERY: Reassign the appropriate .DAT slot and rerun CHAIN.

PAUSE

MEANING: The user has entered either the PAR or PAL options.

RECOVERY: To allow CHAIN to continue processing enter ↑P.

↑QUESTIONABLE MEMORY SIZE

MEANING: The user has entered a core size (nnnK) which is not a multiple of 8. The message is only a warning and CHAIN will correctly build the overlay for the desired memory size.

READ ERROR

MEANING: An unrecoverable I/O error was detected during the processing of input files. CHAIN terminates and control is returned to XVM/DOS.

RECOVERY: Reassemble the appropriate file and rerun CHAIN. If the problem persists and cannot be accounted for call your DEC field service office.

↑RES ROUTINE NAME USED AS LINK NAME

MEANING: The last link name used is the same as one of the routines defined to be an element of the resident code. The current input line is ignored.

RECOVERY: Choose another link name and reenter the link definition rejected by CHAIN.

Alphabetical List of Chain Messages

↑RES ROUTINE REQ'D

MEANING: The user has attempted to build an overlay structure with no resident code section. The current input line is ignored.

RECOVERY: Define a set of resident code before continuing with CHAIN.

RESIDENT CODE

MEANING: CHAIN is about to print the memory map of the resident code section.

RESIDENT COMMON BLOCKS

MEANING: CHAIN is about to print a list of all common blocks associated with the resident code section of the overlay structure. Blank common is excluded from this list and is printed separately.

TABLE OVERLAP

MEANING: The internal tables used by CHAIN have run out of free core. CHAIN terminates and control is returned to XVM/DOS.

RECOVERY: Build the overlay structure in a machine with more core if possible (32K instead of 24K). Otherwise, redesign the overlay structure to reduce the number of symbols kept in CHAIN's internal tables.

↑UNRECOGNIZED SYMBOL

MEANING: The user entered an option to CHAIN which was not recognized. The current input line is ignored.

RECOVERY: Enter the correct option and continue with CHAIN.

UNRESOLVED GLOBALS:

MEANING: CHAIN is about to print a list of global symbols which were not resolved during the relocation process. CHAIN terminates after printing the symbols and control is returned to XVM/DOS.

RECOVERY: Include the appropriate modules in the proper link to define the missing global symbols.

↑ZERO LENGTH LIBRARY NAME

MEANING: The user has entered a null (consisting of zero characters) name to the UL or SL options. The current input line is ignored.

RECOVERY: Enter the correct library name and continue with chain.

INDEX

CHAIN XVM/EXECUTE XVM
Utility Manual
DEC-XV-UCHNA-A-D

Please cut along this line.

READER'S COMMENTS

NOTE: This form is for document comments only. Problems with software should be reported on a Software Problem Repcrt (SPR) form.

Did you find errors in this manual? If so, specify by page.

____

Did you find this manual understandable, usable, and well-organized? Please make suggestions for improvement.

____

Is there sufficient documentation on associated system programs required for use of the software described in this manual? If not, what material is missing and where should it be placed?

____

Please indicate the type of user/reader that you most nearly represent.

- □ Assembly language programmer
- □ Higher-level language programmer
- □ Occasional programmer (experienced)
- □ User with little programming experience
- □ Student programmer
- □ Non-programmer interested in computer concepts and capabilities

Name ____________________ Date ____________

Organization ____________________________

Street ____________________________

City ______________ State ________ Zip Code or Country __________

If you require a written reply, please check here. □

-- Fold Here --

-- Do Not Tear - Fold Here and Staple --

digital

Software Communications
P. O. Box F
Maynard, Massachusetts 01754

digital

digital equipment corporation

PRINTED IN U.S.A.