XVM/DOS V1A SYSTEM INSTALLATION GUIDE

DEC-XV-ODS1A-A-D

XVM Systems

digital

XVM/DOS V1A SYSTEM INSTALLATION GUIDE

DEC-XV-ODSIA-A-D

digital equipment corporation · maynard. massachusetts

First Printing, January 1976

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.

The software described in this document is furnished under a license and may be used or copied only in accordance with the terms of such license.

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.

The following are trademarks of Digital Equipment Corporation:

DIGITAL
DEC
PDP
DECUS
UNIBUS
COMPUTER LABS
COMTEX
DDT
DECCOMM
DECsystem-10
DECtape
DIBOL
EDUSYSTEM
FLIP CHIP
FOCAL
INDAC
LAB-8
DECsystem-20
MASSBUS
OMNIBUS
OS/8
PHA
RSTS
RSX
TYPESET-8
TYPESET-10
TYPESET-11

CONTENTS

Contents (Cont)

Page

Contents (Cont)

Contents (Cont)

Contents (Cont)

Contents (Cont)

FIGURES

TABLES

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 READER'S 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

In the preparation of this manual, it was assumed the reader is familiar with the contents of the following manuals:

EDIT/EDITVP/EDITVT XVM UTILITY MANUAL
MAC11 XVM ASSEMBLER LANGUAGE MANUAL
MACRO XVM ASSEMBLER LANGUAGE MANUAL
PATCH XVM UTILITY MANUAL
PIP XVM UTILITY MANUAL
SGEN XVM UTILITY MANUAL
UPDATE XVM UTILITY MANUAL
XVM/DOS KEYBOARD COMMAND GUIDE
XVM/DOS USERS MANUAL
XVM UNICHANNEL SOFTWARE MANUAL

CHAPTER 1
INTRODUCTION

This document provides information concerning XVM/DOS in the following four general areas.

1. It describes XVM/DOS.
2. It provides a step-by-step system build procedure.
3. It indicates DOS-15 V3B problems which now have been resolved.
4. It provides a list of assembly parameters for future reference.

The user who just received an XVM/DOS kit and wants to "GET ON THE AIR" should carefully read the entire document, then follow the step-by-step instructions found in Section 2.2.

The user installing the LV11 raster scan plotter will be interested in Section 2.8.

The user interested in new features which are available in XVM/DOS should review Appendices A and B.

The user who has already installed XVM/DOS using the instructions in Section 2.2 and who wishes to reassemble a particular portion of the system should review Appendices C and D for assembly parameter information.

The remainder of this chapter contains an overview of XVM/DOS and its configuration requirements.

1.1 OVERVIEW OF XVM/DOS

Prime features of the XVM/DOS system are the following:

- Support of 128K Wide-Addressing of the XVM hardware.
- Extensive system reliability with respect to:
    1. Disk corruption
    2. Error detection

3. GET/PUT processing.

.Batch operation (BOSS) includes:

1. Wide-Addressing.
2. MAC11 assembly.
3. Single file F4 compile and go capability.

.Optional resource utilization including:

1. User specified ↑Q area sizes (including "NONE").
2. DOSSAV optional ↑Q area save with no retention of spooler area information.
3. Monitor memory allocation on an as-required basis.
4. Configuration verification.
5. Assembly of larger MACRO XVM programs through use of a garbage collecting algorithm.

.New system installation process (Section 4.1) including:

1. Source file release.
2. Initial system construction program (BUILD).
3. "MCLOAD", a MAC11 installation aid.
4. Skip entry in SGEN.
5. Handerless device capability.

.Multiple device command batching.

1.2 XVM/DOS CONFIGURATION

The following table describes the basic XVM/DOS configuration capabilities and provides that of the DOS-15 V3BØØØ configuration for a basis of comparison.

Table 1-1
XVM/DOS Configuration

	XVM/DOS[1]	DOS-15 V3BØØØ
Minimum Memory Size	24K words	16K words
Maximum Memory Size	128K words	32K words
Extra 4K Support	No	YES
Minimum Disk Support	512K words	256K words
XM-15 Support	YES	NO
PDP-15 Support	YES	YES
PDP-9 Support	NO	YES
Real Time Clock	Recommended[2]	Optional
Automatic Priority Interrupt	Recommended[3]	Optional
LV11 Raster Scan Plotter	Optional	Not Supported
DL11 10/XVM Communication	Optional	Not Supported

1.3 KIT CONTENTS

This section describes the XVM/DOS V1AØØØ kit contents.

1.3.1 There are three XVM/DOS kits:

1. DECTAPE
2. 7-TRACK MAGTAPE
3. 9-TRACK MAGTAPE

The two magtape kits are essentially identical. The contents of the DECtape and magtape kits are:

[1] A card reader (CR11, CR15, or CR03B) and a line printer (LP11, LP15, LS11, LV11 (used as a printer)) are required to use BOSS XVM.

[2] The real time clock is required for these systems with UNICHANNEL-15, or, those systems using BOSS XVM. In addition the real time clock is recommended for all other systems to provide timing capability.

[3] The Automatic Priority Interrupt (API) option is required for these systems with more than 4 UNICHANNEL-15 devices. In addition, system reliability, and, throughput is enhanced on those systems with API.

DECTAPE		MAGTAPE	
No. of Tapes	Description	No. of Tapes	Description
2	RK Skeleton DOSSAV	1	RK Skeleton DOSSAV
2	RP Skeleton DOSSAV	1	RP Skeleton DOSSAV
2	RF Skeleton DOSSAV	1	RF Skeleton DOSSAV
20	Source Tapes	5	Source Tapes
1	DOSSAV Paper Tape	1	DOSSAV Paper Tape
1	PIREX Paper Tape	1	PIREX Paper Tape
1	ABSL11 Paper Tape	1	ABSL11 Paper Tape
1	RKBOOT Paper Tape	1	RKBOOT Paper Tape
1	RPBOOT Paper Tape	1	RPBOOT Paper Tape
1	RFBOOT Paper Tape	1	RFBOOT Paper Tape
25	Documents	25	Documents

The user should familiarize himself with the kit contents.

1.3.2 Each Skeleton DOSSAV contains:

System Programs:

EDIT - Editor
PIP - File Manipulation Utility
UPDATE - Library Manipulation Utility
PATCH - System Program Installation
SGEN - Online Configuration
MACRO - Assembler

Under the <IOS> uic:

RKA. or DPA. or DKA. disk handler; DTA. or MTA. distribution media handler; and the PPA paper tape punch handler.

Under the <BNK> uic:

EXECUT BIN
MACDEF MAC
INSTRC BIN
INSALL SRC
INSERR SRC

Under the <LIB> uic:

.LIBF BIN Floating Point Library
.LIBNF BIN Non-Floating Point Library

Under the <BLD> uic:

M11.8	IMG	8K MAC11 Image
M11.12	IMG	12K MAC11 Image
QDMP	IMG	QDMP XVM Utility Image
UDMP	IMG	UDMP XVM Utility Image
BUILT	XCT/XCU	Build Question/Answer Routine
BUILD	BAT	Build Batch File
CHECK	XCT/XCU	Tape Selection Verification Routine
AAA	PRM	Various MAC11 parameter files (i.e. URK PRM)

1.3.3 The skeleton DOSSAV should not be used for any other purpose than to "BUILD" the desired local configuration. In particular, PATCH, UPDATE, SGEN, and MACRO should NOT be used until after the automatic BUILD process completes.

There is only one exception to this rule. If XVM/DOS will only be used as the vehicle by which XVM/RSX is built and/or bootstrapped, the skeleton DOSSAV may be used without any modification. However, it is recommended that all users of XVM/RSX "BUILD" XVM/DOS so that it matches the installation's hardware configuration.

CHAPTER 2
BUILDING THE SYSTEM

2.1 INTRODUCTION

This chapter contains step-by-step instructions for generating an XVM/DOS V1AØØØ system. The object of the XVM/DOS build procedure (BUILD) is to create an XVM/DOS system tailored to the user's local hardware configuration and performance requirements. This section provides an outline of the stages involved in performing system construction. Specific details are found in Section 2.2.1.

The user is supplied with a DECtape or magtape distribution kit containing among other items a DOSSAVed version of a working skeleton Disk Operating System. After going through a DOSSAV and bootstrap procedure to get this skeleton system running, the user initiates a batch stream named BUILD which in turn EXECUTES a program named BUILD. The BUILD program creates the full Disk Operating System by asking the user questions about desired configuration options, device handlers, and system programs and creating multiple batch files which actually incorporate the indicated items into the system. This is done as an automatic process and the user need only mount or label the tapes requested by the BUILD program. After the automatic build procedure is completed, the user may additionally modify system parameters not handled by BUILD. Finally, the user should save the full system for future use and backup.

The BUILD process implies variability and choice. Three categories of choices are available:

1. Configuration Options
2. Handler Options
3. System Program Options

A system is tailored to local needs by selecting options from these categories via BUILD.

2.2 STEP-BY-STEP BUILD PROCEDURE

The system construction process can be viewed as having major milestones; the particular set of user actions and system output in achieving a milestone is considered a phase of BUILD. The milestones are listed below and shown in Figure 2-1.

1. Creating the skeleton system disk.
2. Answering the configuration, handler and system program questions.
3. The automatic transfer, assembly, and installation of the various system components.
4. Manually adjusting various system parameters.
5. Saving the completed system for future use.

The third phase of the BUILD process is directed by a disk batch file produced by the second phase of the BUILD process.

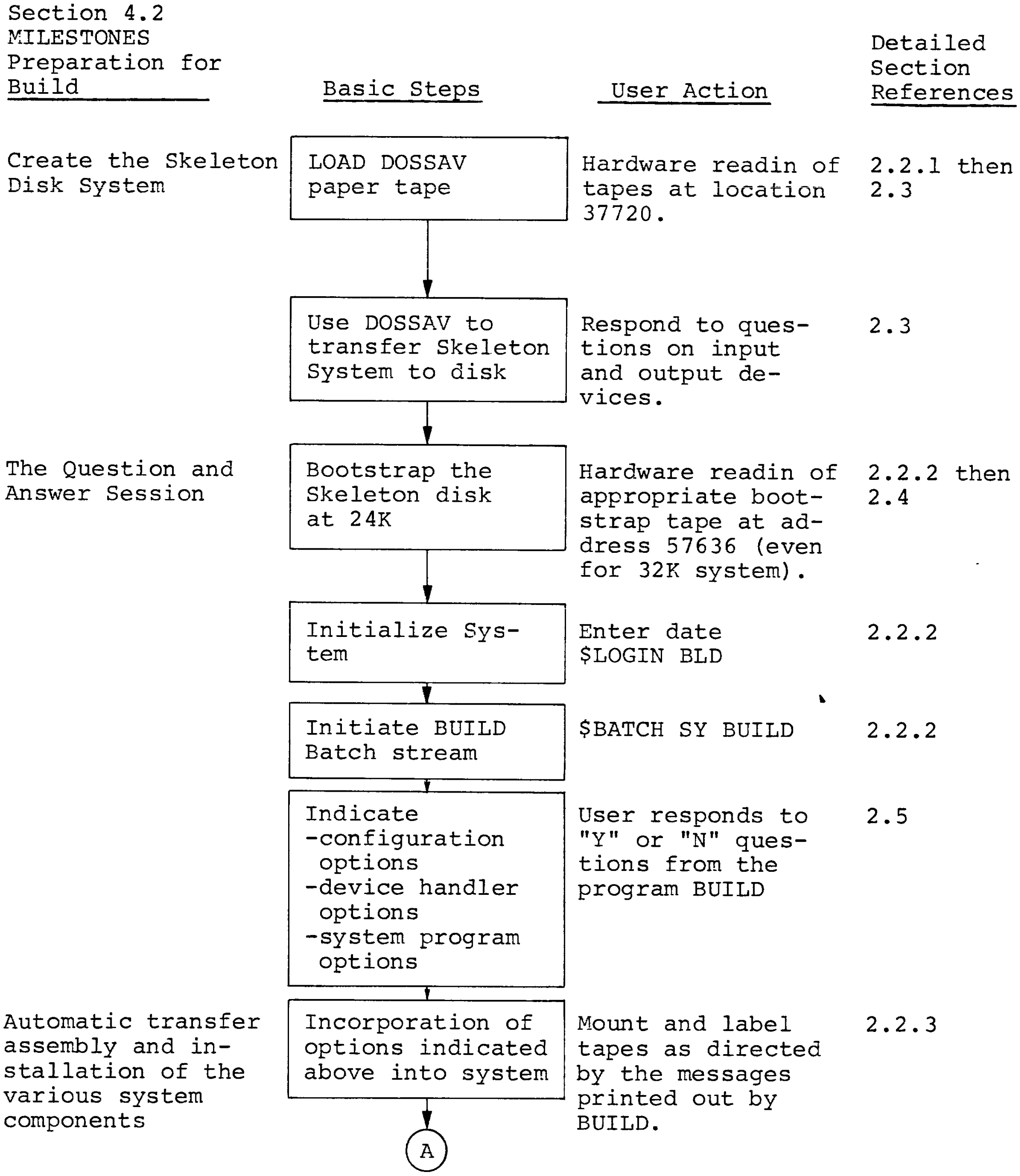
2.2.1 Startup Creating The Skeleton System Disk

The procedure to be followed by the user to create the skeleton system disk and prepare to run the BUILD automated process is:

1. Read this document in its entirety, very thoroughly.
2. While reading this chapter, determine the answers to the relevant questions in Section 2.6. If your system has an LV11 Electrostatic Printer/Plotter, refer to the information in Section 2.8 before proceeding to the next step.
3. Check the kit contents as described in Section 1.3 for completeness.
4. Determine the type of system disk to be constructed and select the corresponding skeleton DOSSAV tape.
5. If your system has a UNICHANNEL-15 peripheral processor (even if RKØ5 is not your system disk), load and start the PIREX Peripheral Processor Execution according to the instructions in Section 2.3.1.
6. Use the DOSSAV paper tape to load your DOSSAV tape for the Skeleton Disk System according to Section 2.3.2.

NOTE

The user is advised not to work solely from this diagram. Do not attempt build procedure until all of Chapter 4 has been read.



Section 4.2 MILESTONES Preparation for Build	Basic Steps	User Action	Detailed Section References
Create the Skeleton Disk System	LOAD DOSSAV paper tape	Hardware readin of tapes at location 37720.	2.2.1 then 2.3
	Use DOSSAV to transfer Skeleton System to disk	Respond to questions on input and output devices.	2.3
The Question and Answer Session	Bootstrap the Skeleton disk at 24K	Hardware readin of appropriate bootstrap tape at address 57636 (even for 32K system).	2.2.2 then 2.4
	Initialize System	Enter date $LOGIN BLD	2.2.2
	Initiate BUILD Batch stream	$BATCH SY BUILD	2.2.2
	Indicate -configuration options -device handler options -system program options	User responds to "Y" or "N" questions from the program BUILD	2.5
Automatic transfer assembly and installation of the various system components	Incorporation of options indicated above into system	Mount and label tapes as directed by the messages printed out by BUILD.	2.2.3

Figure 2-1

Non-Detailed Summary of the Build Procedures

Section 4.2 MILESTONES Preparation for Build	Basic Steps	User Action	Detailed Section References
	A		
Manual Adjustment of various system parameters	Install new PIREX in UNICHANNEL system	Use ABSL11 to install new version of PIREX created in previous section.	2.2.4
	Bootstrap system	Hardware readin of appropriate bootstrap tape at address 57637 for a 24K system or 77637 for a 32K or greater system.	2.2.2 then 2.4
	MICLOG SYS		2.2.4.1
	Use SGEN		2.2.4.1
Saving the Completed System	DOSSAV the system		2.2.5 and 2.3

[1] For systems with a UNICHANNEL peripheral processor.

Figure 2-1 (Cont)
Non-Detailed Summary of the Build Procedures

2.2.2 The Question And Answer Session

Once the skeleton system has been written on disk via DOSSAV, the user can proceed with the next phase of the procedures which is:

1. Place the source tapes, found in the XVM/DOS kit in assending order - to aid errorless loading.
2. Load the Paper Tape Punch with a FULL box of paper tape. This is very important because there is no recovery from an out-of-paper tape condition during the following procedure.
3. Bootstrap the skeleton disk at 24K (starting address of 57637_8). The 24K bootstrap is required, even on larger systems, since, BUILD can only execute with a 24K bootstrap. See Section 2.4 for bootstrap information.
4. Enter the date as "MM/DD/YY" when requested.
5. "LOGIN" to the <BLD> uic ("LOGIN BLD").
6. The skeleton system is configured to use an LA30 DECwriter. The terminal characteristic parameter settings are:

   "TAB ON" and "FILL ON"

   Users with different console terminals may wish to alter the default terminal characteristics by using the "TAB" and "FILL" keyboard commands. The following are recommended settings for the various supported console terminals:

Terminal	TAB	FILL
LA30 DECwriter	ON	ON
LA36 DECwriter II	ON	OFF
LT33 Teletype[1]	ON	OFF
LT35 Teletype[1]	OFF	OFF

   For example, the user with a LT35 Teletype[1] console terminal would type:

   ```
   $ FILL OFF
   $ TAB OFF
   ```

7. Start the question and answer session by typing "BATCH SY BUILD". This will "BATCH" from the system device ("SY") using the BATCH file "BUILD BAT".

[1] Copyright of Teletype Corporation.

8. The execution of the BATCH file "BUILD BAT" causes the execution of the "BUILD" program (Figure 2-2). The "BUILD" program asks questions, and based upon the answers, it creates the multiple BATCH files that actually constructs the XVM/ DOS system.

9. The user should reply to each question with either "Y" or "N". See Figure 2-2 and Section 2.5 for details of the question and answer session.

10. At the completion of the question and answer session (indicated by a "STOP ØØØ message), the BATCH file "BUILD BAT" BATCHes to the first of the newly created system construction BATCH files "MACRO BAT".

NOTE

> BATCHing from a BATCH file is in effect a CHAINing process - with NO return capability.

2.2.3 The Automatic Construction Phase

The Question and Answer Phase automatically starts execution of the BATCH files created by "BUILD". (See Section 2.1 for an example of an automatic BATCH BUILD Phase of system construction.) During this execution, the user is required to:

1. Mount tapes in response to:

   "$PAUSE: PLEASE MOUNT ON UNIT #Ø "SOURCE $\left\{\begin{matrix}\text{DECTAPE}\\\text{MAGTAPE}\end{matrix}\right\}$ #n"

   where DECTAPE or MAGTAPE is typed as appropriate, and, "n" is the source tape number.

   The user mounts the appropriate tape and types ↑R (control "R").

2. Check the paper tape punch for sufficient paper tape in response to:

   ```
   "$PAUSE: CHECK THAT THE PUNCH HAS AT LEAST 3/4 BOX OF
     P' TAPE"
   ```

   The user refills the paper tape punch if needed and types ↑R (control "R").

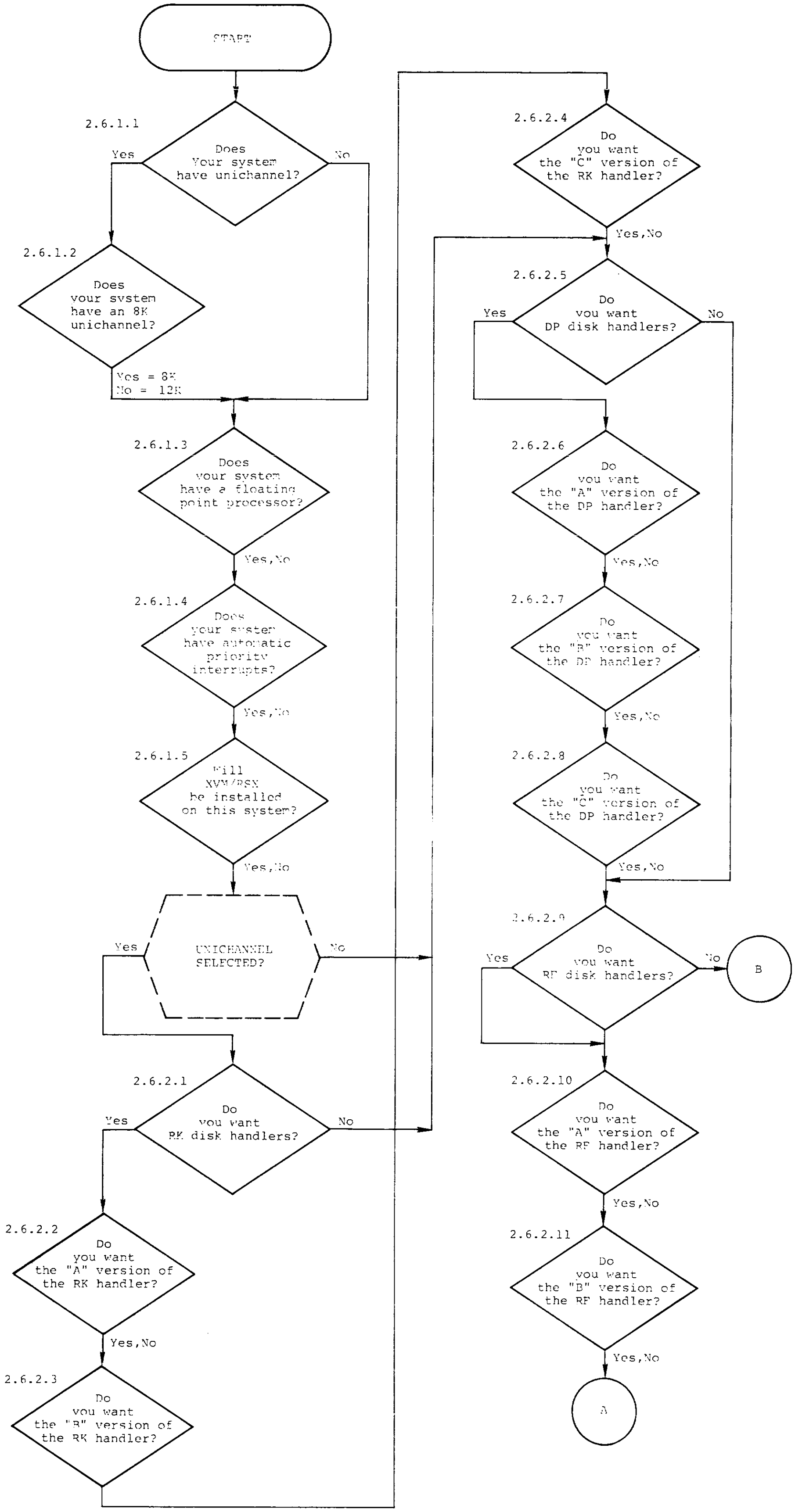

START
2.6.1.1
Does your system have unichannel?
Yes
No
2.6.1.2
Does your system have an 8K unichannel?
Yes = 8K
No = 12K
2.6.1.3
Does your system have a floating point processor?
Yes,No
2.6.1.4
Does your system have automatic priority interrupts?
Yes,No
2.6.1.5
Will XVM/RSX be installed on this system?
Yes,No
UNICHANNEL SELECTED?
Yes
No
2.6.2.1
Do you want RK disk handlers?
Yes
No
2.6.2.2
Do you want the "A" version of the RK handler?
Yes,No
2.6.2.3
Do you want the "B" version of the RK handler?
2.6.2.4
Do you want the "C" version of the RK handler?
Yes,No
2.6.2.5
Do you want DP disk handlers?
Yes
No
2.6.2.6
Do you want the "A" version of the DP handler?
Yes,No
2.6.2.7
Do you want the "B" version of the DP handler?
Yes,No
2.6.2.8
Do you want the "C" version of the DP handler?
Yes,No
2.6.2.9
Do you want RF disk handlers?
Yes
No
B
2.6.2.10
Do you want the "A" version of the RF handler?
Yes,No
2.6.2.11
Do you want the "B" version of the RF handler?
Yes,No
A

Figure 2-2

XVM/DOS System Build Flow Diagram

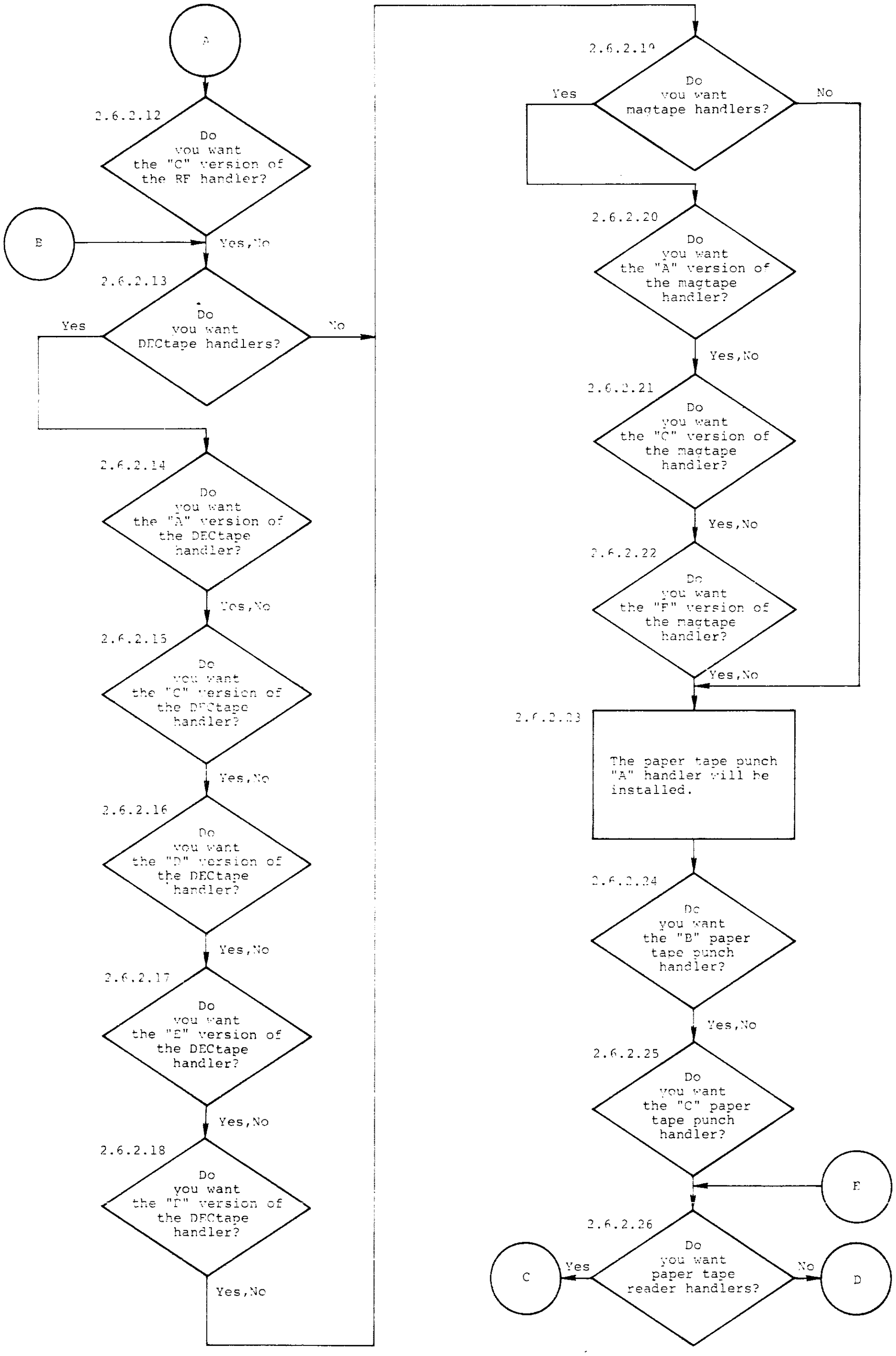

A
2.6.2.12
Do you want the "C" version of the RF handler?
B
Yes,No
2.6.2.13
Do you want DECtape handlers?
Yes
No
2.6.2.14
Do you want the "A" version of the DECtape handler?
Yes,No
2.6.2.15
Do you want the "C" version of the DECtape handler?
Yes,No
2.6.2.16
Do you want the "D" version of the DECtape handler?
Yes,No
2.6.2.17
Do you want the "E" version of the DECtape handler?
Yes,No
2.6.2.18
Do you want the "F" version of the DECtape handler?
Yes,No
2.6.2.19
Yes
Do you want magtape handlers?
No
2.6.2.20
Do you want the "A" version of the magtape handler?
Yes,No
2.6.2.21
Do you want the "C" version of the magtape handler?
Yes,No
2.6.2.22
Do you want the "F" version of the magtape handler?
Yes,No
2.6.2.23
The paper tape punch "A" handler will be installed.
2.6.2.24
Do you want the "B" paper tape punch handler?
Yes,No
2.6.2.25
Do you want the "C" paper tape punch handler?
E
2.6.2.26
C
Yes
Do you want paper tape reader handlers?
No
D

Figure 2-2

XVM/DOS System Build Flow Diagram (cont)

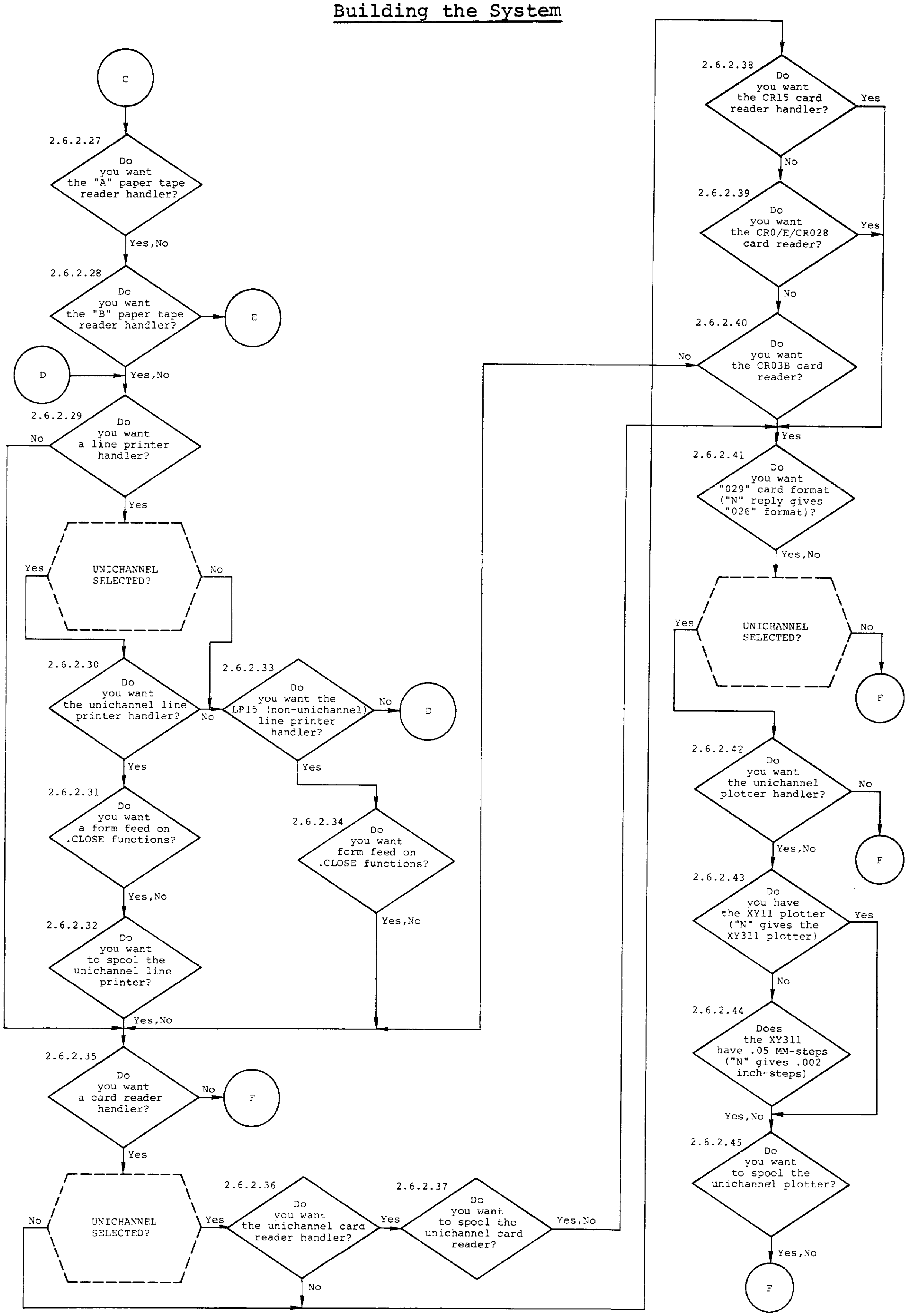

C
2.6.2.27
Do you want the "A" paper tape reader handler?
Yes,No
2.6.2.28
Do you want the "B" paper tape reader handler?
E
D
Yes,No
2.6.2.29
Do you want a line printer handler?
No
Yes
UNICHANNEL SELECTED?
Yes
No
2.6.2.30
Do you want the unichannel line printer handler?
No
Yes
2.6.2.31
Do you want a form feed on .CLOSE functions?
Yes,No
2.6.2.32
Do you want to spool the unichannel line printer?
Yes,No
2.6.2.33
Do you want the LP15 (non-unichannel) line printer handler?
No
D
Yes
2.6.2.34
Do you want form feed on .CLOSE functions?
Yes,No
2.6.2.35
Do you want a card reader handler?
No
F
Yes
UNICHANNEL SELECTED?
No
Yes
2.6.2.36
Do you want the unichannel card reader handler?
Yes
No
2.6.2.37
Do you want to spool the unichannel card reader?
Yes,No
2.6.2.38
Do you want the CR15 card reader handler?
Yes
No
2.6.2.39
Do you want the CR0/E/CR028 card reader?
Yes
No
2.6.2.40
Do you want the CR03B card reader?
No
Yes
2.6.2.41
Do you want "029" card format ("N" reply gives "026" format)?
Yes,No
UNICHANNEL SELECTED?
Yes
No
F
2.6.2.42
Do you want the unichannel plotter handler?
No
F
Yes,No
2.6.2.43
Do you have the XY11 plotter ("N" gives the XY311 plotter)
Yes
No
2.6.2.44
Does the XY311 have .05 MM-steps ("N" gives .002 inch-steps)
Yes,No
2.6.2.45
Do you want to spool the unichannel plotter?
Yes,No
F

Figure 2-2

XVM/DOS System Build Flow Diagram (cont)

]

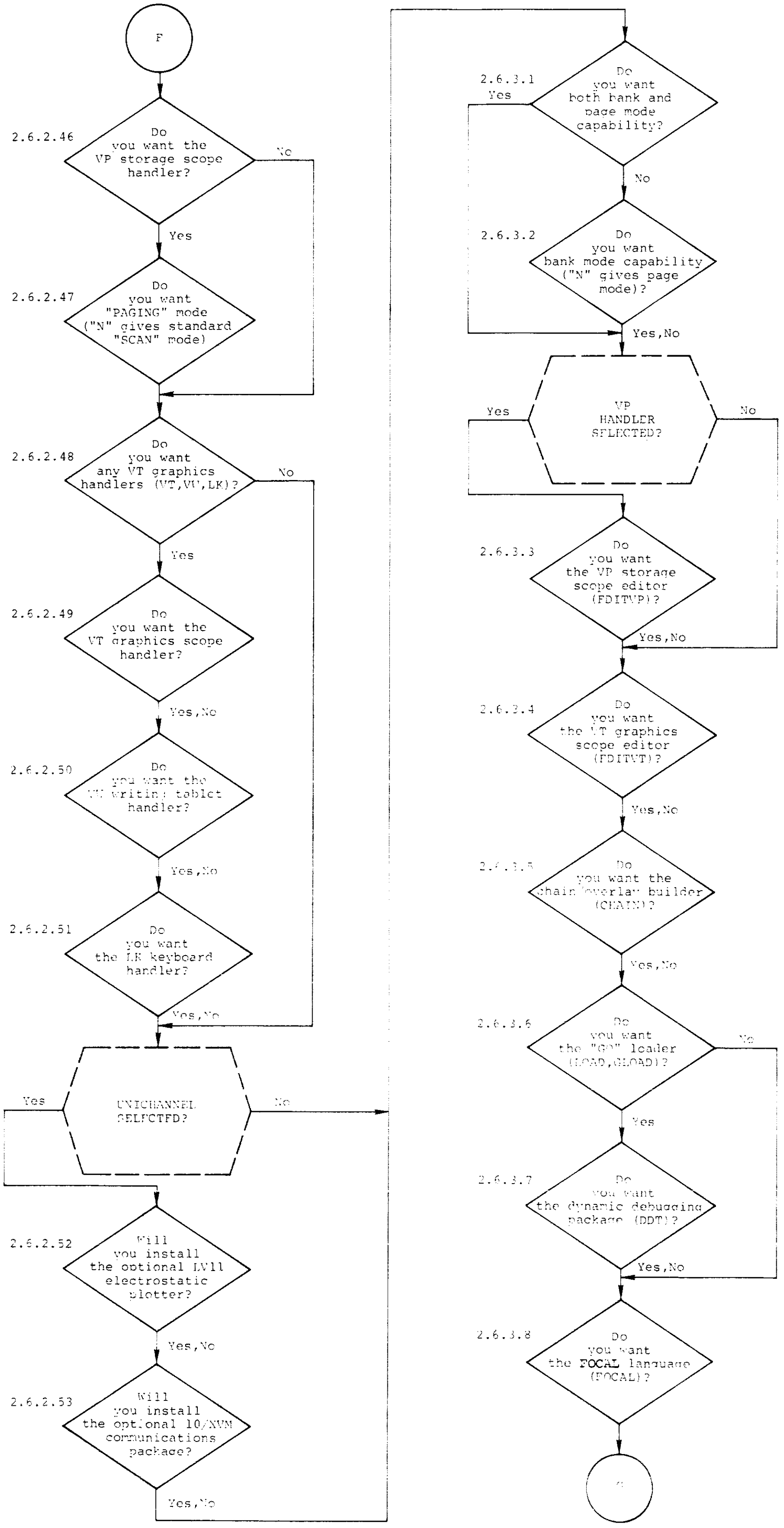

F
2.6.2.46
Do you want the VP storage scope handler?
No
Yes
2.6.2.47
Do you want "PAGING" mode ("N" gives standard "SCAN" mode)
2.6.2.48
Do you want any VT graphics handlers (VT,VW,LK)?
No
Yes
2.6.2.49
Do you want the VT graphics scope handler?
Yes,No
2.6.2.50
Do you want the VW writing tablet handler?
Yes,No
2.6.2.51
Do you want the LK keyboard handler?
Yes,No
Yes
UNICHANNEL SELECTED?
No
2.6.2.52
Will you install the optional LV11 electrostatic plotter?
Yes,No
2.6.2.53
Will you install the optional 10/XVM communications package?
Yes,No
2.6.3.1
Yes
Do you want both bank and page mode capability?
No
2.6.3.2
Do you want bank mode capability ("N" gives page mode)?
Yes,No
Yes
VP HANDLER SELECTED?
No
2.6.3.3
Do you want the VP storage scope editor (EDITVP)?
Yes,No
2.6.3.4
Do you want the VT graphics scope editor (EDITVT)?
Yes,No
2.6.3.5
Do you want the chain/overlay builder (CHAIN)?
Yes,No
2.6.3.6
Do you want the "GO" loader (LOAD,GLOAD)?
No
Yes
2.6.3.7
Do you want the dynamic debugging package (DDT)?
Yes,No
2.6.3.8
Do you want the FOCAL language (FOCAL)?
G

Figure 2-2
XVM/DOS System Build Flow Diagram (cont)

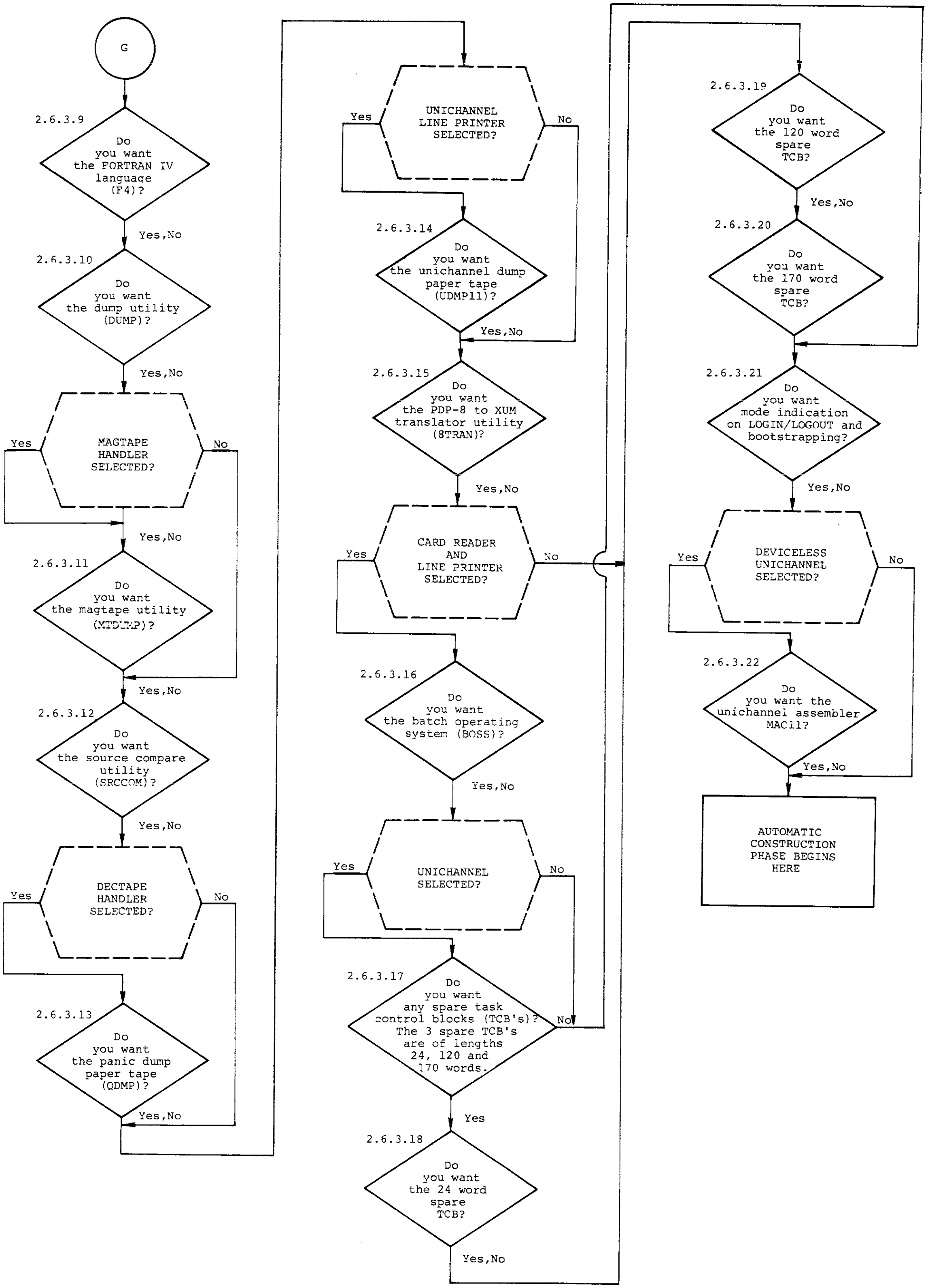

G
2.6.3.9
Do you want the FORTRAN IV language (F4)?
Yes,No
2.6.3.10
Do you want the dump utility (DUMP)?
Yes,No
MAGTAPE HANDLER SELECTED?
Yes
No
Yes,No
2.6.3.11
Do you want the magtape utility (MTDUMP)?
Yes,No
2.6.3.12
Do you want the source compare utility (SRCCOM)?
Yes,No
DECTAPE HANDLER SELECTED?
Yes
No
2.6.3.13
Do you want the panic dump paper tape (QDMP)?
Yes,No
UNICHANNEL LINE PRINTER SELECTED?
Yes
No
2.6.3.14
Do you want the unichannel dump paper tape (UDMP11)?
Yes,No
2.6.3.15
Do you want the PDP-8 to XUM translator utility (8TRAN)?
Yes,No
CARD READER AND LINE PRINTER SELECTED?
Yes
No
2.6.3.16
Do you want the batch operating system (BOSS)?
Yes,No
UNICHANNEL SELECTED?
Yes
No
2.6.3.17
Do you want any spare task control blocks (TCB's)? The 3 spare TCB's are of lengths 24, 120 and 170 words.
No
Yes
2.6.3.18
Do you want the 24 word spare TCB?
Yes,No
2.6.3.19
Do you want the 120 word spare TCB?
Yes,No
2.6.3.20
Do you want the 170 word spare TCB?
2.6.3.21
Do you want mode indication on LOGIN/LOGOUT and bootstrapping?
Yes,No
DEVICELESS UNICHANNEL SELECTED?
Yes
No
2.6.3.22
Do you want the unichannel assembler MAC11?
Yes,No
AUTOMATIC CONSTRUCTION PHASE BEGINS HERE

Figure 2-2
XVM/DOS System Build Flow Diagram (cont)

3. Label various paper tapes punched by the BUILD process in response to:

   "$PAUSE: PLEASE LABEL THE NEW PAPER TAPE "NAME""

   The user labels the paper tape and then types ↑R (control "R").

4. Load into the paper tape reader, indicated ("NAME") paper tapes in response to:

   "$PAUSE: PLEASE LOAD THE "NAME" PAPER TAPE INTO THE READER"

   The user loads the paper tape carefully, then types ↑R (control "R").

5. During the automatic execution of "PATCH" several "ILLEGAL COMMAND" error messages may occur. These should be ignored as they do not indicate actual errors.

6. When the Automatic Process completes, as indicated by the "$EXIT" message, the user is required to begin the manual portion of the BUILD procedure.

2.2.4 Manual Adjustment of Assembly Parameters

At the completion of the Automatic Phase of the system construction, the user should determine the specific system parameter settings most useful for the local configuration. (Refer to the SGEN XVM Utility Manual for specific details of the available parameter settings and for information on how to do it.) Parameter decisions include:

1. Terminal Characteristics ("FILL"/"TAB") state
2. Number/Size of buffers ("BUFFS") state
3. Default UNICHANNEL/POLLER status ("UC15"/"POLLER") state
4. BANK/PAGE default mode ("BANK"/"PAGE") state
5. XVM default status ("XVM") state
6. System Memory size ("MEMSIZ") state
7. Device Assignment Table (.DAT) Entries
8. Special Device Handler entries
9. Special Skip Chain ordering
10. Clock Frequency

The procedure to follow to complete the system construction process and to set the relevant system parameters is:

Building the System

1. Install the new PIREX paper tape if one was created (see Section 2.3.1).
2. Bootstrap the system (see Section 2.4).
3. 'MICLOG" into the protected system uic using "SYS" as the password. ("MICLOG SYS")
4. Use "PIP" to delete the <BLD> uic.

```
        >N  SY <BLD>  (K)
```

   This is done because a BATCH file cannot be deleted while being batched from, and, <BLD> contains the last BATCH file which is no longer needed.

5. Run "SGEN" to:
   a. Select specific system parameters not determined by the BUILD process (i.e., default buffer size).
   b. Add any special device handlers - specific to your installation (i.e., an Analog-to-Digital Converter).
   c. Change the skip chain order if necessary, to reflect installation specific needs.
   d. Set up the default .DAT table - as the default supplied with the Skeleton is useful only for the build process.
   e. Allocate a ↑Q area if desired.
   f. Install your installation specific system programs.
6. Install the optional LV11 Plotter (see Section 2.7).

2.2.5 Saving the Completed System

Once the system parameter customization phase has been completed, the user should save the system for future use. The process is as follows:

1. Mount a blank tape on the DOSSAV output unit.
2. DOSSAV the system using the procedure described in Section 2.3.
3. Once the DOSSAV is completed, the user should re-bootstrap the system (see Section 2.4) using as a starting address:

   a. 77637 for 32K or larger systems

   b. 57637 for 24K or larger systems

4. The system is now ready for use.

2.3 DOSSAV XVM

NOTE

> For systems with UNICHANNEL hardware, PIREX must be loaded and running[1] before using DOSSAV with disk cartridges.

This section provides detailed instructions for the DOSSAV process. DOSSAV XVM (DOSSAV) is the save/restore program for XVM/DOS and XVM/RSX. DOSSAV is initially used in restoring the user's system. Once a working system is built, DOSSAV is useful in backing up systems as a safeguard.

2.3.1 Loading PIREX

To load PIREX, halt the PDP-11 by pressing the ENABLE/HALT console switch down. Stop the XVM by pressing the STOP/RESET toggles simultaneously. On both machines, the run lights should be off. Set the XVM's address switches to 17700_8. (Front of rocker switch is "UP" if the switch is "ON".) Place the ABSL11 paper tape in the XVM's paper tape reader. Press both the STOP and RESET switches simultaneously and then the READIN switch on the XVM console. The paper tape should read in, and the XVM should halt (RUN light off).

Place 140000 (switch numbers 14 and 15 are up and all others down) in the address switches of the PDP-11. (A bit is "ON" when the corresponding switch is up.) Press down the LOAD-ADDRESS switch. Move the ENABLE-HALT switch to ENABLE (up). Press down the START switch. At this point the PDP-11 should start (RUN light should come on).

Place the PIREX paper tape in the XVM's paper tape reader and then press the CONTINUE toggle on the XVM. The PIREX tape should read in and the XVM should halt. The PIREX monitor in the PDP-11 should start, indicated by: RUN light on and bit Ø of the address lights on. (This is the characteristic pattern of the PIREX NULL task.)

[1] Spooling should be disabled during all DOSSAV operations.

2.3.2 Loading DOSSAV

Before loading DOSSAV to be used with disk cartridges, PIREX must be running in the PDP-11 (see above). DOSSAV is a stand-along program which is loaded from the paper tape reader using the XVM's Hardware Readin Mode. To load DOSSAV:

1. Make sure that the input, output and paper tape devices are powered on and on-line.
2. Place the DOSSAV paper tape in the paper tape reader so that it is ready to be read.
3. Set the XVM's console address switches to 3772Ø$_8$ (restart address = 54ØØØ$_8$).[1]
4. Simultaneously press the STOP and RESET switches on the XVM's console.
5. Finally press the READIN switch. (The DOSSAV paper tape should pass through the paper tape reader.)
6. When loaded, DOSSAV identifies itself as follows:

```
DOSSAV XVM Vnxnnn
INPUT DEVICE?
```

2.3.3 Using DOSSAV

Once DOSSAV has been loaded and has identified itself, as above, the user must answer several questions. The questions asked by DOSSAV and the legal user responses are listed in Table 2-1. Each user response must be terminated by a Carriage RETURN. It should also be noted that some of the questions shown are typed out only when applicable (i.e., no questions are asked about Disk Pack unit number or Magtape density, or number of channels if these devices are not to be used). Legal input and output device combinations are shown in Table 2-2. When the requested operation is complete, DOSSAV restarts and identifies itself as shown above.

[1] DOSSAV may be loaded at any of the following locations: 1772Ø$_8$, 3772Ø$_8$, 5772Ø$_8$ or 7772Ø$_8$. A binary loader precedes DOSSAV on the paper tape and this loader moves DOSSAV to bank 2 where DOSSAV actually runs. The restart address must be 54ØØØ$_8$.

Table 2-1

DOSSAV Commands

Query	User Responses
INPUT DEVICE?	DT = DECtape, MT[1] = Magtape, DK = DECdisk, DP = Disk Pack, RK = Disk Cartridge
UNIT #?	Legal unit numbers are 0-7.
SAVE QAREA[4] (Y OR N)?	Y = SAVE THE CTRL/Q AREA, N = DO NOT SAVE THE CTRL/Q AREA. Asked only if the input device is a disk and it has a CTRL/Q AREA.
TRACK (7 OR 9)?	Magtape track or channel number. (If 9 is specified, density is assumed to be 800 BPI.) Asked only if specified device is Magtape.
DENSITY[2,3](2,5,8)?	Magtape recording density: 2 = 200 Bits Per Inch (BPI), 5 = 556 BPI, 8 = 800 BPI. Asked only if specified device is Magtape.
OUTPUT DEVICE?	(See INPUT DEVICE above.)
UNIT #?	(See UNIT NO. above.)
TRACK (7 OR 9)?	(See TRACK above.)
DENSITY (2,5,8)?	(See DENSITY above.)
DATE CREATED:	The date that the restore tape was created is typed out by DOSSAV.
TAPE DONE MOUNT ANOTHER	A multi-tape DOSSAV requires the next tape to be mounted. Mount the tape and type the unit number followed by a Carriage RETURN.

1. Magtapes with even parity cannot be saved or restored by DOSSAV.
2. All XVM/DOS Magtapes distributed by DEC are 800 BPI, Odd Parity.
3. This question is only asked when using 7-Track Magtape.
4. Asked only when saving disks which have a SYSBLK and a ↑Q Area of non-zero length.

Table 2-2

Legal DOSSAV I/O Device Combinations

Output Device Input Device	DEC-disk (DK)	Disk Pack (DP)	DEC-tape (DT)	Mag-tape[1] (MT)	Disk Cartridge (RK)
DECtape (DT)	OK	OK	Illegal[2]	Illegal	OK
Magtape (MT)	OK	OK	Illegal	Illegal	OK
DECdisk (DK)	Illegal	OK	OK	OK	OK
Disk Pack (DP)	OK	Illegal[2]	OK	OK	Illegal
Disk Cartridge	Illegal	OK	OK	OK	Illegal[2]

[1] All XVM/DOS Magtapes distributed by DEC are 800 BPI, Odd Parity.

[2] Use PIP Utility program copy command with the (H) switch option for this operation.

2.3.4 Restarting DOSSAV

The following restart procedures should be used to terminate the current operation prematurely or to reinitialize DOSSAV if it fails to start up automatically after an error.

1. Press the XVM console switches STOP and RESET simultaneously.
2. Set the ADDRESS switches to 54000_8
3. Press the START console switch.

DOSSAV should then identify itself as it does when originally loaded, if not, it must be reloaded as described above in LOADING DOSSAV.

2.3.5 Error Conditions

2.3.5.1 Recoverable Errors During Command String Decoding - If a question is answered incorrectly, DOSSAV outputs an appropriate error message and then repeats the question. These error messages are:

ILLEGAL DEVICE — An illegal device mnemonic was typed (something other than DP, DK, RK, DT, or MT) or an illegal combination of devices was typed (i.e., input = DT and output = MT).

BAD TRACK — Something other than 7 or 9 was typed.

BAD DENSITY — Something other than 2 (200), 5 (556), or 8 (800) was typed.

2.3.5.2 Recoverable Errors During Operations - If it is possible to recover from an error, DOSSAV attempts to do it. The error message is output to the console. After the problem has been corrected, any character typed on the keyboard followed by a Carriage RETURN resumes operation.

TAPE NOT READY — The DECtape or magtape unit is off line or not write enabled.

DISK NOT READY — DECdisk is write locked.

DISK PACK NOT READY — The Disk Pack or Disk Cartridge unit is not ready.

2.3.5.3 Unrecoverable Errors - These are primarily hardware errors from which DOSSAV cannot recover. After the error message has been output, DOSSAV restarts. DOSSAV retries five times on parity error, before issuing an unrecoverable error message.

DECTAPE ERROR
MAGTAPE ERROR
DISK ERROR
DISK PACK ERROR
ATTEMPT TO RESTORE SYSTEM TO WRONG DISK

To protect users who have access to more than one type of disk and who may have several sets of restore tapes, all restore tapes are created with the mnemonic of the disk type in the first SAT. DOSSAV checks this code against the output device code. If they differ, this message is output.

BLK 1775 OCCUPIED. NO 2ND SAT CREATED

A DECdisk system created for 4 or fewer platters is restored to a machine with 5 or more platters and block 1775 is already used. Therefore, no second SAT is created. A master tape was not used to make the restore.

XX ERR IGN — where XX = DK or DP or RK.
This error is typed on the console, and the XVM halts. This reports that "READ/Write check" errors occurred more than 12_8 times during a save or restore process. The bad block number is present in the XVM's AC. Users can continue the save or restore process by pressing the continue switch on the console of the machine.

Examples illustrating typical DOSSAV save/restores are included in Appendix A of the SGEN XVM Utility Manual and in Chapter 7 of the XVM/DOS User's Manual.

2.4 BOOTSTRAPPING

This section provides information concerning bootstrapping the XVM/DOS System. Use the following procedure to start the XVM/DOS System.

1. Perform this step only for UNICHANNEL configurations, otherwise go on to step 2. If the Peripheral Executive Software System (PIREX) is loaded and running in the UNICHANNEL's PDP-11 peripheral processor, then go on to step 2.

   To load PIREX, place the ABSL11 paper tape in the XVM's paper tape reader. Place the ENABLE/HALT switch on the PDP-11 in the HALT position. Press the STOP and RESET switches on the XVM simultaneously. Set the ADDRESS switches on the XVM to 17700. Press the READIN switch on the XVM. When the readin operation is completed and the XVM has halted, set the PDP-11 switch register to 140000 and depress the PDP-11 LOAD ADDR switch, then set the ENABLE/HALT switch on the PDP-11 to ENABLE, and finally press the PDP-11 START switch.

   Remove ABSL11 from the paper tape reader, and reload the reader with the PIREX paper tape. Press CONTINUE on the XVM. This will cause the ABSL11 program (which has two segments: A PDP-11 segment, and a XVM segment) to read in PIREX (which is a PDP-11 absolute binary tape) via the XVM segment and load it into PDP-11 lower memory via the PDP-11 segment.

   When the PIREX paper tape has been read in, the XVM will halt, and the PDP-11 will be running PIREX. Remove the PIREX paper tape from the reader. At this point the UNICHANNEL Peripheral Processor has been loaded and is waiting for an I/O request from XVM/DOS.

2. Load the proper XVM/DOS bootstrap for the type of system disk you are using in the paper tape reader. Set the XVM ADDRESS switches as shown below:

57637	for 24K words of memory on the XVM (always use this address to start the "BUILD" procedures.)
77637	for 32K or greater words of memory on the XVM (use this at the termination of the build procedure for a 32K or greater system).

   Press STOP and RESET on the XVM simultaneously, then press READIN. This will load the XVM/DOS system bootstrap, which will in turn load the XVM/DOS monitor system without further intervention.

3. When the XVM/DOS monitor system is loaded into memory either by initial system loading (step 2) or manual restarting (step 4) the system identifies itself and requests the current date, as follows:

```
XVM/DOS Vnxnnn
ENTER DATE (MM/DD/YY)--
```

After the date is typed in the format shown, (i.e.,8/24/75 for August 24, 1975) followed by a carriage return, the monitor will output a one or two line mode specification message. The format and content of this message are included in the MODE description under Special Function Commands (in the XVM/DOS USER's Manual). After the mode specification message has been output, the monitor prompts the user for command line input, as follows:

```
$
```

The user responds by typing a monitor command or the name of a system program which the user desires to execute to the right of the '$' prompt character. Each monitor command line (System Program Loader Commands or Special Function Commands) must be terminated by either a CARRIAGE RETURN or ALTMODE. After each Special Function Command has been processed by the monitor, the '$' prompt character is once again typed. Typing a System Program Load command will cause the named system program to be loaded and started. It will then prompt for dialog as described in the corresponding section of the XVM/DOS Keyboard Command Guide or appropriate Utility Manual.

4. Once the bootstrap is in memory and PIREX is loaded (in UNICHANNEL configuration), it is possible to manually reload the XVM/DOS monitor after power-downs, user program loops, system program errors, etc., usually without reloading the bootstrap or PIREX (i.e., UNICHANNEL configurations). The restart procedures are:

PIREX (if the UNICHANNEL has stopped) - follow step 1.

Bootstrap - If the XVM is looping, halted or hung, and the system does not respond to typing ↑C, set the ADDRESS switches as follows:

57646 for 24K memory on the XVM
77646 for 32K or greater memory on the XVM

and press STOP and RESET simultaneously. Then depress START to cause the XVM/DOS monitor system to be loaded.

If the above procedures fail, return to 2 to reload the Bootstrap program.

2.5 DETAILS OF THE AUTOMATIC BUILD PROCESS

This section consists of detailed information concerning the Automatic Phase of the BUILD Process. The actual step-by-step procedure to follow is provided in the preceeding Section 2.2.

2.5.1 Operational Information

The question and answer phase of the BUILD process consists of:

a. Questions
b. "Installed" messages
c. Error Recovery

2.5.1.1 Questions - Each question is of the form:

```
"TEXT OF QUESTION"
REPLY "Y" or "N"?
```

The user should determine the correct reply and then type Y↵ for a YES reply or N↵ for a NO reply. A reply other than "Y" or "N" will not be accepted, and, BUILD will retype:

```
"REPLY "Y" or "N"?"
```

The user must then enter a valid reply.

In order to minimize terminal interaction, BUILD will ask qualifier questions before asking any long series of questions about a particular device type. For instance, BUILD will ask:

```
"DO YOU WANT RP DISK HANDLERS?"[1]
```

[1] This particular question is asked only if the system disk is not an RP15.

a "Y" reply will cause associated questions to be asked about the DPA., DPB., and DPC. versions of the RP handler. A "N" reply will cause all RP handler questions to be skipped. A "Y" reply to a qualifier question implies that a "Y" reply to at least one associated question will occur. If all associated questions are answered negatively, an error is assumed and the message[1]:

```
"NO XX HANDLER SELECTED!!!"
"PLEASE WAIT ONE MOMENT"
```

is typed and after a few seconds the qualifier question is repeated.

Qualifier questions are not asked for devices requiring only one reply for complete specification.

2.5.1.2 "INSTALLED" Messages - At various times during the question and answer session, the BUILD program determines that certain modules are to be installed in the target system, even though they have not been explicitly requested by the user.

An example of this implied selection is the automatic selection of the system disk type A handler. For an RP system, this selection is indicated by the message:

```
"-THE "A" VERSION OF THE RP HANDLER WILL BE INSTALLED-"
```

"INSTALLED" messages are used to indicate handler, system program and library routine installation.

If the user does not want a module that is automatically installed, he should wait until the completion of the entire BUILD process and then use either PIP or SGEN (where appropriate) to remove the module.

CAUTION

Under NO circumstances should the user attempt to interrupt the BUILD process before completion.

[1] XX is the device code from the respective qualifier question.

2.5.1.3 Inadvertent Errors - There are two types of inadvertent errors:

1. An incorrect choice during the question and answer phase. For example, not choosing a necessary device handler.

2. A hardware or operator error during the construction phase. For example a disk malfunction.

An incorrect choice during the question and answer phase can be voided by typing ↑C (control C) at any point before the "STOP ØØØ message indicating the end of the question and answer phase. The user must then type "BATCH␣SY␣<BLD>␣BUILD" to restart the question and answer phase.

> CAUTION
>
> Under NO circumstances should the user type ↑C after the question and answer phase has completed; this will abort the BUILD process.

A fatal hardware or operator error can be recovered by:

1. Aborting the BATCH run if necessary by typing a ↑C (control C)

2. Correcting the malfunction.

3. Determining the name of the current BATCH file by locating the last "BATCH␣SY␣<BLD>␣XXXXX" on the console listing where "XXXXX" is the name of a BATCH file, XXXXX␣BAT.

4. Editing the BATCH file XXXXX BAT, found in uic <BLD>, to remove those operations already performed.

5. Insuring that the BATCH file still contains the correct "ASSIGN", "KEEP", "BUFFS", and "UC15" parameter setting to complete execution of the file.

6. Typing "MICLOG␣SYS"

7. Then typing BATCH␣SY␣<BLD>␣XXXXX

This should resume the build process. If this procedure is not successful, the user must begin the BATCH process from the skeleton restore step (see Section 2.2.1). This recovery may fail if:

a. The incorrect .DAT assignments are specified.

b. The user incorrectly edits the current BATCH file.

c. A required file has already been deleted.

d. The user discovers the error after the automatic build process has deleted the BATCH file under which it occurred.

2.5.1.4 BUILD Time Requirements - The BUILD process time requirements differ depending upon configurations, selections, and media:

a. DECTAPE BUILDS require between 3.5 and 4.5 hours.

b. MAGTAPE BUILDS require between 4.5 and 6 hours. Magtape BUILDS require no operator action between the question and answer session and the mount of tape #2. Operator action is required thereafter. The user should plan to devote the maximum amount of time, in one contiguous session, to the BUILD process.

2.6 QUESTION AND ANSWER DETAILS

Figure 2-2 provides the questions asked by the BUILD program in the order in which they will be asked. The balance of this section provides explanations of all questions and the implication of certain responses are described. An example of the Automatic BUILD Phase is also provided.

2.6.1 Configuration Questions

The session begins with the following being output on your console:

"THIS SECTION ASKS QUESTIONS ABOUT YOUR CONFIGURATION"

Explanation: Configuration questions that modify the remainder of the question and answer section will be asked.

2.6.1.1 DOES YOUR SYSTEM HAVE UNICHANNEL?

Implication: A "N" reply will cause all UNICHANNEL related device and software questions to be eliminated from the "BUILD" question stream. A "Y" reply causes the UC15 and POLLER to be turned "ON" by the "BUILD" procedures, a "N" reply will have the opposite effect.

NOTE

This question is not asked if the system is being built on an RK disk skeleton - a "Y" reply is assumed.

2.6.1.2 DOES YOUR SYSTEM HAVE AN 8K UNICHANNEL ("N" GIVES "12K")?

Explanation: This question asks which of the supported UNICHANNEL local memory sizes (8K, 12K) is on your configuration.

Implication: There are two possible MAC11 versions for 8K and 12K UNICHANNELS.

2.6.1.3 DOES YOUR SYSTEM HAVE A FLOATING POINT PROCESSOR?

Implication: A "Y" reply will, if libraries are required, cause the floating point versions of the libraries to be selected. In addition, the non-floating point libraries will, if required by FOCAL, be retained as .LIBNF BIN under the <BNK> and/or <PAG> uics.

A "N" reply will, if libraries are required, cause the non-floating point libraries to be selected. If FORTRAN is desired, this question determines which version, floating point vs. non-floating point is to be generated.

2.6.1.4 DOES YOUR SYSTEM HAVE AUTOMATIC PRIORITY INTERRUPTS?

Implication: The automatic SGEN performed during the BUILD process will set your default API mode based on the reply to this section.

2.6.1.5 WILL XVM/RSX BE INSTALLED ON THIS SYSTEM?

Explanation: This question asks if you intend to utilize the system as a base for XVM/RSX.

Implication: The disk handler interleave factor is set to 1 if a "Y" reply is given. This provides optimal storage allocation for XVM/RSX, but, is not optimal for XVM/DOS systems. A "N" reply should be given if XVM/DOS will be run on this system most of the time.

2.6.2 THIS SECTION ASKS QUESTIONS ABOUT HANDLERS

Explanation: Device handler questions are asked in this section. Selection of a handler implies the presence of the device.

2.6.2.1 DO YOU WANT RK DISK HANDLERS?

Explanation: This question asks if you want <u>any</u> RKØ5 disk handlers. A "Y" reply will result in additional RK disk handler questions. A "N" reply will eliminate further RK disk handler questions.

Implications: Spooling on UNICHANNEL systems requires the presence of an RKØ5 disk. Spooling questions will not be asked if no RKØ5 handler is selected.

NOTE

This question will not be asked if the system disk is an RKØ5. The answer will be assumed to be "Y".

2.6.2.2 DO YOU WANT THE "A" VERSION OF THE RK HANDLER?

Explanation: This question asks if you want the RKA. handler. The "A" version is the most general of all handler versions. It is also the largest.

NOTE

This question will not be asked if the system disk is on RKØ5. The answer will be assumed to be "Y".

2.6.2.3 DO YOU WANT THE "B" VERSION OF THE RK HANDLER?

Explanation: This question asks if you want the RKB. handler. The "B" version is capable of only serial file structured Input/Output. It is 20% smaller than the "A" version of the RK handler.

2.6.2.4 DO YOU WANT THE "C" VERSION OF THE RK HANDLER?

Explanation: This question asks if you want the RKC. handler. The "C" version is capable of only serial file structured Input. It is 50% smaller than the "A" version of the RK handler.

2.6.2.5 DO YOU WANT RP DISK HANDLERS?

Explanation: This question asks if you want any RPØ2 disk handlers. A "Y" reply will result in additional RP disk handler questions. A "N" reply will eliminate further RP disk handler questions.

NOTE

This question will not be asked if the system disk is an RPØ2. The answer will be assumed to be "Y".

2.6.2.6 DO YOU WANT THE "A" VERSION OF THE RP HANDLER?

Explanation: This question asks if you want the DPA. handler. The "A" version is the most general of all handler versions. It is also the largest.

NOTE

This question will not be asked if the system disk is on RPØ2. The answer will be assumed to be "Y".

2.6.2.7 DO YOU WANT THE "B" VERSION OF THE RP HANDLER?

Explanation: This question asks if you want the DPB. handler. The "B" version is capable of only serial file structured Input/Output. It is 20% smaller than the "A" version of the RP handler.

2.6.2.8 DO YOU WANT THE "C" VERSION OF THE RP HANDLER?

Explanation: This question asks if you want the DPC. disk handler version. The "C" version is capable of only serial file structured Input. It is 50% smaller than the "A" version of the RP handler.

2.6.2.9 DO YOU WANT RF DISK HANDLERS?

Explanation: This question asks if you want any RF15 disk handlers. A "Y" reply will result in additional RF disk handler questions. A "N" reply will eliminate further RF disk handler questions.

NOTE

This question will not be asked if the system disk is an RF15. The answer will be assumed to be "Y".

2.6.2.10 DO YOU WANT THE "A" VERSION OF THE RF HANDLER?

Explanation: This question asks if you want the DKA. disk handler version. The "A" version is the most general of all handler versions. It is also the largest.

NOTE

This question will not be asked if the system disk is on RF15. The answer will be assumed to be "Y".

2.6.2.11 DO YOU WANT THE "B" VERSION OF THE RF HANDLER?

Explanation: This question asks if you want the DKB. disk handler version. The "B" version is capable of only serial file-structured Input/Output. It is 20% smaller than the "A" version of the RF handler.

2.6.2.12 DO YOU WANT THE "C" VERSION OF THE RF HANDLER?

Explanation: This question asks if you want the DKC. disk handler version. The "C" version is capable of only serial file-structured Input. It is 50% smaller than the "A" version of the RF handler.

2.6.2.13 DO YOU WANT DECTAPE HANDLERS?

Explanation: This question asks if you want any DECtape handlers. A "Y" reply will result in additional DECtape handler questions. A "N" reply will eliminate further DECtape handler questions.

Implications: DECtape is required to utilize the QDMP XVM panic DUMP papertape. A "N" reply will eliminate the QDMP XVM question from the BUILD process.

NOTE

This question will not be asked for systems supplied on DECtape media. A "Y" reply will be assumed.

2.6.2.14 DO YOU WANT THE "A" VERSION OF THE DECTAPE HANDLER?

Explanation: This question asks if you want the DTA. handler. The "A" version is the most general in capability; it is also the largest.

NOTE

This question will not be asked for systems supplied on DECtape media. A "Y" reply will be assumed.

2.6.2.15 DO YOU WANT THE "C" VERSION OF THE DECTAPE HANDLER?

Explanation: This question asks if you want the DTC. handler. The "C" version is an Input only file-structured handler. It is 50% smaller than the "A" version.

2.6.2.16 DO YOU WANT THE "D" VERSION OF THE DECTAPE HANDLER?

Explanation: This question asks if you want the DTD. handler. The "D" version is a single file-structured Input/Output handler with "MTAPE" capability. It is as large as the "A" version.

2.6.2.17 DO YOU WANT THE "E" VERSION OF THE DECTAPE HANDLER?

Explanation: This question asks if you want the DTE. handler. The "E" version is identical to the "D" version with the exception of no "MTAPE" capability. It is 10% smaller than the "A" version.

2.6.2.18 DO YOU WANT THE "F" VERSION OF THE DECTAPE HANDLER?

Explanation: This question asks if you want the DTF. handler. The "F" version is a multiunit non-file structured "MTAPE" capability handler. It is 60% smaller than the "A" version.

2.6.2.19 DO YOU WANT MAGTAPE HANDLERS?

Explanation: This question asks if you want any MAGTAPE handlers. A "Y" reply will result in additional MAGTAPE handler questions. A "N" reply will eliminate further MAGTAPE handler questions.

Implications: A "N" reply will eliminate the ability to run the MTDUMP program. The MTDUMP question will not be asked in this case.

NOTE

This question will not be asked for systems supplied on MAGTAPE media. A "Y" reply will be assumed.

2.6.2.20 DO YOU WANT THE "A" VERSION OF THE MAGTAPE HANDLER?

Explanation: This question asks if you want the MTA. handler. The "A" version is the most general version with both directoried and true MAGTAPE file capabilities. It is the largest version.

NOTE

This question will not be asked for systems supplied on MAGTAPE media. A "Y" reply will be assumed.

2.6.2.21 DO YOU WANT THE "C" VERSION OF THE MAGTAPE HANDLER?

Explanation: This question asks if you want the MTC. handler. The "C" version is a read only directoried file structure handler with single unit capability. It is 75% smaller than the "A" version.

2.6.2.22 DO YOU WANT THE "F" VERSION OF THE MAGTAPE HANDLER?

Explanation: This question asks if you want the MTF. handler. The "F" version is a multi-unit MAGTAPE file structure handler. It is 60% smaller than the "A" version.

2.6.2.23 THE PAPER TAPE PUNCH "A" HANDLER WILL BE INSTALLED -

Explanation: The PPA. handler is required to punch the various paper tapes that are a part of the BUILD process. The "A" version is always installed in the skeleton. This version is the most general supporting all data modes.

2.6.2.24 DO YOU WANT THE "B" PAPER TAPE PUNCH HANDLER?

Explanation: This question asks if you want the PPB. handler. The "B" version supports all data modes except IOPS ASCII. It is 25% smaller than the "A" version.

2.6.2.25 DO YOU WANT THE "C" PAPER TAPE PUNCH HANDLER?

Explanation: This question asks if you want the PPC. handler. The "C" version has only IOPS Binary data mode capability. It is 40% smaller than the "A" version.

2.6.2.26 DO YOU WANT PAPER TAPE READER HANDLERS?

Explanation: This question asks if you want any paper tape reader handlers. A "Y" reply will result in additional Paper Tape Reader handler questions. A "N" reply will eliminate further Paper Tape Reader questions.

2.6.2.27 DO YOU WANT THE "A" PAPER TAPE READER HANDLER?

Explanation: This question asks if you want the PRA. handler. The "A" version is the most general supporting all data modes. It is the largest version.

2.6.2.28 DO YOU WANT THE "B" PAPER TAPE READER HANDLER?

Explanation: This question asks if you want the PRB. handler. The "B" version supports only IOPS ASCII data mode input. It is 30% smaller than the "A" version.

2.6.2.29 DO YOU WANT A LINE PRINTER HANDLER?

Explanation: This question asks if you want any line printer handler. A "Y" reply will result in additional line printer handler questions. A "N" reply will eliminate further line printer questions.

Implications: A line printer handler is required to utilize BOSS XVM. A "N" reply will eliminate the BOSS question from the BUILD process.

2.6.2.30 DO YOU WANT THE UNICHANNEL LINE PRINTER HANDLER?

Explanation: This question asks if you want the LPA. handler. A "Y" reply will result in additional UNICHANNEL line printer questions. A "N" reply will eliminate them. A "Y" reply to this question will eliminate the LP15 line printer question.

2.6.2.31 DO YOU WANT A FORM FEED ON .CLOSE FUNCTION?

Explanation: This question asks if you want a form feed to occur on every .CLOSE to the line printer handler.

Implications: A "N" reply will result in less paper utilization, but the output format will be less highly structured.

2.6.2.32 DO YOU WANT TO SPOOL THE UNICHANNEL LINE PRINTER?

Explanation: This question asks if you want Line Printer SPOOLER to be generated for your configuration.

Implications: The UNICHANNEL version of LPA. is assembled to allow spooling, in addition the SPOOLER Line Printer module is assembled and installed.

NOTE

This question will be asked only if an RKØ5 is in your configuration.

It is the user's reponsibility in insure that if the configuration has an 8K UNICHANNEL, no more than two devices are spooled. Three devices may be spooled on 12K UNICHANNEL configurations.

2.6.2.33 DO YOU WANT THE LP15 (NON-UNICHANNEL) LINE PRINTER HANDLER?

Explanation: This question asks if you want the LP15 version of the LPA. handler.

NOTE

This question is asked if there is no UNICHANNEL-15 Line Printer handler selected.

2.6.2.34 DO YOU WANT A FORM FEED ON .CLOSE FUNCTIONS?

Explanation: See 2.6.2.31 above.

Implications: See 2.6.2.31 above.

2.6.2.35 DO YOU WANT A CARD READER HANDLER?

Explanation: This question asks if you want any card reader handler. A "Y" reply will result in additional card reader questions. A "N" reply will eliminate further card reader questions.

Implications: A card reader handler is required to utilize BOSS XVM. A "N" reply will eliminate the BOSS question from the BUILD process.

2.6.2.36 DO YOU WANT THE UNICHANNEL CARD READER HANDLER?

Explanation: This question asks if you want the CR11 UNICHANNEL handler (CDB.). A "Y" reply will result in additional UNICHANNEL card reader questions. A "N" reply will eliminate them.

Implications: A "Y" reply to this question will allocate a TASK CONTROL BLOCK (TCB) in the resident monitor for the UNICHANNEL card reader. A "N" reply will not allocate a TCB for the card reader. A "Y" reply will eliminate the Non-UNICHANNEL card reader questions.

2.6.2.37 DO YOU WANT TO SPOOL THE UNICHANNEL CARD READER?

Explanation: See 2.6.2.32 for relevant explanation

Implications: See 2.6.2.32 for relevant explanation

NOTE

See 2.6.2.32 for relevant note.

2.6.2.38 DO YOU WANT THE CR15 CARD READER HANDLER?

Explanation: This question asks if you want the CR15 version of the CDB. handler.

Implications: A "Y" reply will eliminate further card reader model options.

2.6.2.39 DO YOU WANT THE CRØ1E/CRØ2B CARD READER?

Explanation: This question asks if you want the CRØ1E or CRØ2B versions of the CDB. handler.

Implications: A "Y" reply will eliminate further card reader model options.

2.6.2.40 DO YOU WANT THE CR03B CARD READER?

Explanation: This question asks if you want the CR03B version of the CDB. handler.

2.6.2.41 DO YOU WANT "Ø29" CARD FORMAT ("N" REPLY GIVES "Ø26" FORMAT)?

Explanation: The reply to this question conditions the card reader handler to accept either DECØ29 keypunch code or DECØ26 keypunch code.

2.6.2.42 DO YOU WANT THE UNICHANNEL PLOTTER HANDLER?

Explanation: This question asks if you want the XY11/XY311 UNICHANNEL plotter handler XYA.

Implications: A "Y" reply to this question will allocate a Task Control Block (TCB) in the resident monitor for the UNICHANNEL plotter.

2.6.2.43 DO YOU HAVE THE XY11 PLOTTER ("N" GIVES THE XY311 PLOTTER)?

Explanation: This question asks which model of the UNICHANNEL plotter do you have.

2.6.2.44 DOES THE XY311 HAVE .Ø5MM STEPS ("N" GIVES .ØØ2 INCH STEPS)?

Explanation: This question asks whether the XY311 is a metric (.Ø5mm steps) or English (.ØØ2 inch steps) model.

2.6.2.45 DO YOU WANT TO SPOOL THE UNICHANNEL PLOTTER?

Explanation: See 2.6.2.32 for relevant explanation

Implications: See 2.6.2.32 for relevant implications

NOTE

See 2.6.2.32 for relevant note.

2.6.2.46 DO YOU WANT THE VP STORAGE SCOPE HANDLER?

Explanation: This question asks if you want the VPA. handler for the VP15A storage CRT.

Implications: A "Y" reply will cause the VP15A routines "NUVAL", "FORT" and "VECTOR" to be installed in the <PER> UFD, and the EDITVP question to be asked.

2.6.2.47 DO YOU WANT "PAGING" MODE ("N" GIVES STANDARD "SCAN" MODE)?

Explanation: This question asks if you want the automatic erase/refill SCAN mode or the manual pushbutton PAGING mode for the VPA. handler.

Implications: This mode is effective for all VPA. handler operations including EDITVP.

2.6.2.48 DO YOU WANT ANY VT GRAPHICS HANDLERS (VT,VW,LK)?

Explanation: This question asks if you want any of the handlers associated with the VT Graphics System.

Implication: A "Y" reply will cause the Graphic's handler questions to be asked. A "N" reply will eliminate these questions.

2.6.2.49 DO YOU WANT THE VT GRAPHICS SCOPE HANDLER?

Explanation: This question asks if you want a handler for the VT15 Graphics Scope (VTA)

Implications: A "Y" reply will cause the VT15 routines "VTPRIM", "DYLDR", "LTORPB", "TRACK", "ROTATE", and "CIRCLE" to be installed in the <PER> UFD.

2.6.2.50 DO YOU WANT THE VW WRITING TABLET HANDLER?

Explanation: This question asks if you want a handler (VWA) for the VW writing Tablet attachment to the VT15 Graphics system.

2.6.2.51 DO YOU WANT THE LK KEYBOARD HANDLER?

Explanation: This question asks if you want a handler (LKA) for the LK Graphics Console keyboard.

2.6.2.52 WILL YOU INSTALL THE OPTIONAL LV11 ELECTROSTATIC PLOTTER?

Explanation: This question asks if you have an LV11 and if you will be using it as a plotter. If the LV11 is to be used only instead of a line printer, reply "N" to this question should be given (see Appendix A for more details.)

Implication: The LVA handler will be installed and a LV plotter TCB will be allocated in the resident monitor and the LV driver will be assembled into PIREX, if a "Y" reply is given.

NOTE

The optional "QUICKSCAN" package requires the LVA plotter handler.

2.6.2.53 WILL YOU INSTALL THE OPTIONAL 10/XVM COMMUNICATIONS PACKAGE?

Explanation: If you intend to install the "TENLINK" PDP-10/XVM communications package then a "Y" reply is necessary. Otherwise, a "N" reply should be given.

Implication: An 10/XVM communications TCB will be allocated in the resident monitor if a "Y" reply is given.

2.6.3 THIS SECTION ASKS QUESTIONS ABOUT SYSTEM PROGRAMS

Explanation: The choice of system programs, libraries, and selected paper tapes is made in this section.

2.6.3.1 DO YOU WANT BOTH BANK AND PAGE MODE CAPABILITY?

Explanation: This question asks if you will utilize both 8K bank and 4K Page index addressing modes. The actual effect of a "Y" reply is to create both <BNK> and <PAG> uic's and their associated files.

Implication: Libraries, EXECUT, DDT, FOCAL, .LOAD, MACDEF, INSALL, INSERR, and INSTRC will be installed in both <BNK> and <PAG> uic's if a "Y" reply is given. A "N" reply will result in your being given a choice between Bank and Page modes. The "Y" reply eliminates the next question.

2.6.3.2 DO YOU WANT BANK MODE CAPABILITY ("N" GIVES PAGE MODE)?

Explanation: A choice between retaining the <BNK> uic or the <PAG> uic is given.

Implication: Libraries, DDT, FOCAL, .LOAD, MACDEF, EXECUT, INSALL, INSERR and INSTRC will be installed in the selected uic (<BNK> or <PAG>). The other uic will contain no files.

2.6.3.3 DO YOU WANT THE VP STORAGE SCOPE EDITOR (EDITVP)?

NOTE

This question is asked only if a VP handler is selected above.

2.6.3.4 DO YOU WANT THE VT GRAPHICS SCOPE EDITOR (EDITVT)?

NOTE

This question is always asked since the VT handler is not used by EDITVT.

2.6.3.5 DO YOU WANT THE CHAIN/OVERLAY BUILDER (CHAIN)?

Explanation: This question determines if CHAIN, the Execute File and Overlay Builder will be installed in the system.

2.6.3.6 DO YOU WANT THE "GO" LOADER (LOAD, GLOAD)?

Explanation: This question asks if you want the immediate binary file loader. This loader has no overlay capability, but does have the ability to be utilized with DDT the Dynamic Debugging Technique.

Implication: A "Y" reply will enable the DDT question to be asked. as the "GO" loader is necessary for DDT operation. A "N" reply will eliminate the DDT question.

2.6.3.7 DO YOU WANT THE DYNAMIC DEBUGGING PACKAGE (DDT)?

Explanation: This question asks if you want the symbolic debugging aid DDT installed in the <BNK> and/or <PAG> uic(s).

2.6.3.8 DO YOU WANT THE FOCAL LANGUAGE (FOCAL)?

Explanation: This question asks if you want the FOCAL interactive language installed in the <BNK> and/or <PAG> uic(s).

Implication: FOCAL utilizes the Non-Floating Point library. If the system has the floating point processor then the floating point library is installed as the standard library, .LIBR BIN in <BNK> and/or <PAG> uic(s). In this case, if FOCAL is selected, then the non-floating point library is retained as .LIBNF BIN under the <BNK> and/or <PAG> uic(s).

2.6.3.9 DO YOU WANT THE FORTRAN IV LANGUAGE (F4)?

Implications: A "N" reply to both this question and the "FOCAL" question will result in no library being installed in the system.

NOTE

The version of the Fortran Compiler floating point or non-floating point is determined by the answer to the floating point hardware question.

2.6.3.10 DO YOU WANT THE DUMP UTILITY (DUMP)?

Explanation: This question asks if you want the mass storage block dumping and QAREA dumping program.

2.6.3.11 DO YOU WANT THE MAGTAPE UTILITY (MTDUMP)?

Explanation: This question asks if you want the Magnetic tape utility program installed in your system. MTDUMP is used to initialize tape directories, dump tape records, and compare and copy complete tapes.

NOTE

This question is asked only if a Magtape handler is selected.

2.6.3.12 DO YOU WANT THE SOURCE COMPARE UTILITY (SRCCOM)?

Explanation: This question asks if you want the SRCCOM program to compare two source files for differences.

2.6.3.13 DO YOU WANT THE PANIC DUMP PAPER TAPE (QDMP)?

Explanation: This question asks if you want the panic XVM/PDP-15 Memory Dump paper tape. This tape, when loaded, dumps memory to a DECtape. The Dump utility can then be used to list the DECtape's contents.

NOTE

This question is asked only is a DECtape handler is selected.

2.6.3.14 DO YOU WANT THE UNICHANNEL DUMP PAPER TAPE (UDMP11)?

Explanation: This question asks if you want the UNICHANNEL Core Dump paper tape. UDUMP11 dumps PDP-11 memory directly to the PDP-11 line printer.

2.6.3.15 DO YOU WANT THE PDP-8 TO XVM TRANSLATOR UTILITY (8TRAN)?

Explanation: This question asks if you want the program that translates PDP-8 assembly language into XVM assembly language - 8TRAN.

2.6.3.16 DO YOU WANT THE BATCH OPERATING SYSTEM (BOSS)?

Explanation: This question asks if you want to install the card oriented BATCH OPERATING SYSTEM.

Implication: The various components of the Batch Operating System; NRBOSS- the Non-Resident Monitor; B.PRE - The Batch Editor; and the BOSS Procedure files will be installed with a "Y" reply.

NOTE

This question is asked only if the system has both line printer and card reader handlers.

2.6.3.17 DO YOU WANT ANY SPARE TASK CONTROL BLOCKS (TCB'S)?
THE 3 SPARE TCB's ARE OF LENGTHS 24, 120, and 170 WORDS

Explanation: This question asks if you need any special task control blocks for your own programming purposes. These TCB's are not required by any standard DEC software.

Implication: Additional Task Control Blocks will utilize memory space even when not used (as long as the UNICHANNEL is enabled). Spare TCB's should not be selected unless they will definitely be used. A "N" reply will cause the remaining TCB questions to be eliminated.

NOTE

This question is asked only if the UNICHANNEL is selected.

2.6.3.18 DO YOU WANT THE 24 WORD SPARE TCB?

Explanation: This question asks if you want the smallest spare Task Control Block.

2.6.2.19 DO YOU WANT THE 120 WORD SPARE TCB?

Explanation: This question asks if you want the medium size spare task control block.

2.6.3.20 DO YOU WANT THE 170 WORD SPARE TCB?

Explanation: This question asks if you want the largest spare task control block.

2.6.3.21 DO YOU WANT MODE INDICATION ON LOGIN/LOGOUT AND BOOTSTRAPPING?

Explanation: This question asks if you want the non-resident monitor to automatically type the mode information (see Appendix A) after each LOGIN/LOGOUT and BOOTSTRAP.

2.6.3.22 DO YOU WANT THE UNICHANNEL ASSEMBLER MAC11?

Explanation: This question asks if you want the PDP-11 UNICHANNEL assembler MAC11 installed on your configuration.

Implication: MAC11 is required to modify all UNICHANNEL software.

NOTE

If MAC11 is required by the BUILD process, this question is assumed to have a "Y" reply.

2.6.4 "PLEASE WAIT ONE MOMENT"

Explanation: The "BUILD" question and answer session is over. "BUILD" is now completing construction of the BATCH files that actually create the finished system.

2.6.5 "STOP ØØØ"

Explanation: "BUILD" has created the BATCH files and BATCHING will begin automatically.

2.7 EXAMPLE OF THE AUTOMATIC BUILD PHASE

This chapter contains an annotated example of the Automatic Phase of system construction. This is a typical example, but, as most users will not have the selected configuration, the actual automatic build process will differ somewhat from this example.

This example assumes:

a. That a skeleton DECtape DOSSAV has been restored to an RKØ5 system disk.

b. That the user has logged into the <BLD> UIC.

NOTE

> User responses shown in this example are underlined and terminated with Carriage RETURNS (↲). System responses are not underlined. Also, when the user types CTRL R (Control R) the system uses this character as ↑R.

```
>^C

XVM/DOS V1A000

$B SY BUILD↲ - - - - - - - - - - - -   BATCH The File BUILD BAT

XVM/DOS V1A000

$$JOB BUILD INITIAL XVM/DOS SYSTEM
$BUFFS 12

$E BUILD       - - - - - - - - - - - -   BEGIN The Question and Answer Session
EXECUTE XVM V1A000
```

```
XVM/DOS INITIAL SYSTEM BUILDER

THIS SECTION ASKS QUESTIONS ABOUT YOUR CONFIGURATION

DOES YOUR SYSTEM HAVE AN 8K UNICHANNEL ("N" GIVES "12K")?
REPLY  Y OR N ?
Y)
DOES YOUR SYSTEM HAVE A FLOATING POINT PROCESSOR?
REPLY  Y OR N ?      Y)
DOES YOUR SYSTEM HAVE AUTOMATIC PRIORITY INTERRUPTS?
REPLY  Y OR N ?      Y)
WILL XVM/RSX BE INSTALLED ON THIS SYSTEM?
REPLY  Y OR N ?      N)

THIS SECTION ASKS QUESTIONS ABOUT HANDLERS

-THE "A" VERSION OF THE RK HANDLER WILL BE INSTALLED- - - An RK System is
DO YOU WANT THE "B" VERSION OF THE RK HANDLER?             being built
REPLY  Y OR N ?      Y)
DO YOU WANT THE "C" VERSION OF THE RK HANDLER?
REPLY  Y OR N ?      Y)
DO YOU WANT RP DISK HANDLERS?
REPLY  Y OR N ?      N)
DO YOU WANT RF DISK HANDLERS?
REPLY  Y OR N ?      N)
-THE DECTAPE "A" HANDLER WILL BE INSTALLED- - - - - - -  The media is DECtape
DO YOU WANT THE "C" VERSION OF THE DECTAPE HANDLER?
REPLY  Y OR N ?      Y)
DO YOU WANT THE "D" VERSION OF THE DECTAPE HANDLER?
REPLY  Y OR N ?      Y)
DO YOU WANT THE "E" VERSION OF THE DECTAPE HANDLER?
REPLY  Y OR N ?      Y)
DO YOU WANT THE "F" VERSION OF THE DECTAPE HANDLER?
REPLY  Y OR N ?      Y)
DO YOU WANT MAGTAPE HANDLERS?
REPLY  Y OR N ?      Y)
DO YOU WANT THE "A" VERSION OF THE MAGTAPE HANDLER?
REPLY  Y OR N ?      Y)
DO YOU WANT THE "C" VERSION OF THE MAGTAPE HANDLER?
REPLY  Y OR N ?      Y)
DO YOU WANT THE "F" VERSION OF THE MAGTAPE HANDLER?
REPLY  Y OR N ?      Y)
-THE PAPER TAPE PUNCH "A" HANDLER WILL BE INSTALLED-
DO YOU WANT THE "B" PAPER TAPE PUNCH HANDLER?
REPLY  Y OR N ?      Y)
DO YOU WANT THE "C" PAPER TAPE PUNCH HANDLER?
REPLY  Y OR N ?      Y)
DO YOU WANT PAPER TAPE READER HANDLERS?
REPLY  Y OR N ?      Y)
DO YOU WANT THE "A" PAPER TAPE READER HANDLER?
REPLY  Y OR N ?      Y)
DO YOU WANT THE "B" PAPER TAPE READER HANDLER?
REPLY  Y OR N ?      Y)
DO YOU WANT A LINE PRINTER HANDLER?
REPLY  Y OR N ?      Y)
DO YOU WANT THE UNICHANNEL LINE PRINTER HANDLER?
REPLY  Y OR N ?      Y)
DO YOU WANT A FORM FEED ON .CLOSE FUNCTIONS?
REPLY  Y OR N ?      N)
DO YOU WANT TO SPOOL THE UNICHANNEL LINE PRINTER?
REPLY  Y OR N ?      Y)
DO YOU WANT A CARD READER HANDLER?
REPLY  Y OR N ?      Y
DO YOU WANT THE UNICHANNEL CARD READER HANDLER?
REPLY  Y OR N ?      Y)
DO YOU WANT TO SPOOL THE UNICHANNEL CARD READER?
```

```
REPLY  Y OR N ?    Y
DO YOU WANT "029" CARD FORMAT ("N" REPLY GIVES "026" FORMAT)
REPLY  Y OR N ?    Y
DO YOU WANT THE UNICHANNEL PLOTTER HANDLER?
REPLY  Y OR N ?    Y
DO YOU HAVE THE XY11 PLOTTER ("N" GIVES THE XY311 PLOTTER)
REPLY  Y OR N ?    Y
DO YOU WANT TO SPOOL THE UNICHANNEL PLOTTER?
REPLY  Y OR N ?    Y
DO YOU WANT THE VP STORAGE SCOPE HANDLER?
REPLY  Y OR N ?    Y
DO YOU WANT "PAGING" MODE ("N" GIVES STANDARD "SCAN" MODE)?
REPLY  Y OR N ?    N
DO YOU WANT ANY VT GRAPHICS HANDLERS (VT VW LK)?
REPLY  Y OR N ?    Y
DO YOU WANT THE VT GRAPHICS SCOPE HANDLER?
REPLY  Y OR N ?    Y
DO YOU WANT THE VW WRITING TABLET HANDLER?
REPLY  Y OR N ?    Y
DO YOU WANT THE LK KEYBOARD HANDLER?
REPLY  Y OR N ?    Y
WILL YOU INSTALL THE OPTIONAL LV11 ELECTROSTATIC PLOTTER?
REPLY  Y OR N ?    N
WILL YOU INSTALL THE OPTIONAL 10/15 COMMUNICATIONS PACKAGE?
REPLY  Y OR N ?    N

THIS SECTION ASKS QUESTIONS ABOUT SYSTEM PROGRAMS

DO YOU WANT BOTH BANK AND PAGE MODE CAPABILITY?
REPLY  Y OR N ?    Y
-THE VP ROUTINE "NUVAL" WILL BE INSTALLED IN <PER>-
-THE VP ROUTINE "FORT" WILL BE INSTALLED IN <PER>-
-THE VP ROUTINE "VECTOR" WILL BE INSTALLED IN <PER>-
```

The user selected the VPA. handler

```
DO YOU WANT THE VP STORAGE SCOPE EDITOR (EDITVP)?
REPLY  Y OR N ?    Y
DO YOU WANT THE VT GRAPHICS SCOPE EDITOR (EDITVT)?
REPLY  Y OR N ?    Y
-THE VT ROUTINE "VTPRIM" WILL BE INSTALLED IN <PER>-
-THE VT ROUTINE "DYLDR" WILL BE INSTALLED IN <PER>-
-THE VT ROUTINE "LTORPB" WILL BE INSTALLED IN <PER>-
-THE VT ROUTINE "TRACK" WILL BE INSTALLED IN <PER>-
-THE VT ROUTINE "ROTATE" WILL BE INSTALLED IN <PER>-
-THE VT ROUTINE "CIRCLE" WILL BE INSTALLED IN <PER>-
```

The user selected the VTA. handler

```
DO YOU WANT THE CHAIN/OVERLAY BUILDER (CHAIN)?
REPLY  Y OR N ?    Y
DO YOU WANT THE "GO" LOADER (LOAD,GLOAD)?
REPLY  Y OR N ?    Y
DO YOU WANT THE DYNAMIC DEBUGGING PACKAGE (DDT)?
REPLY  Y OR N ?    Y
DO YOU WANT THE FOCAL LANGUAGE (FOCAL)?
REPLY  Y OR N ?    Y
DO YOU WANT THE FORTRAN IV LANGUAGE (F4)?
REPLY  Y OR N ?    Y
DO YOU WANT THE DUMP UTILITY (DUMP)?
REPLY  Y OR N ?    Y
DO YOU WANT THE MAGTAPE UTILITY (MTDUMP)?
REPLY  Y OR N ?    Y
DO YOU WANT THE SOURCE COMPARE UTILITY (SRCCOM)?
REPLY  Y OR N ?    Y
DO YOU WANT THE PANIC DUMP PAPER TAPE (QDMP)?
REPLY  Y OR N ?    Y
DO YOU WANT THE UNICHANNEL DUMP PAPER TAPE (UDMP11)?
REPLY  Y OR N ?    Y
DO YOU WANT THE PDP-8 TO XVM TRANSLATOR UTILITY (8TRAN)?
REPLY  Y OR N ?    Y
DO YOU WANT THE BATCH OPERATING SYSTEM (BOSS)?
```

```
REPLY  Y OR N ?     Y
DO YOU WANT ANY SPARE TASK CONTROL BLOCKS (TCB'S)?
REPLY  Y OR N ?     N
DO YOU WANT MODE INDICATION ON LOGIN/LOGOUT AND BOOTSTRAPINC
REPLY  Y OR N ?     N
-THE UNICHANNEL ASSEMBLER "MAC11" WILL BE INSTALLED-       }    RK systems require
-THE MAC11 INSTALLATION AID "MCLOAD" WILL BE INSTALLED-    }    MACII to be built
-THE SPOOL INSTALLATION AID "SPLOAD" WILL BE INSTALLED-           }
-THE UNICHANNEL SPOOLER "SPOOL" WILL BE INSTALLED-                } The user asked for
-THE SPOOLER DISK AREA ALLOCATOR "SPLGEN" WILL BE INSTALLED-      } a spooled device
PLEASE WAIT ONE MOMENT
STOP  000000


XVM/DOS V1A000

$$JOB RUN BATCH FILES - - - - - - The automatic BATCH run
$BUFFS 4                           begins here

$K ON

$BATCH SY <BLD> MACRO - - - - - - The first BATCH file is
XVM/DOS V1A000                    MACRO BAT

$$JOB MACRO
$MICLOG SYS

$TIME 010000

$A CM -10

$K ON

$A SY <BLD> 1,2,-11

$A SY <IOS> -13

$$PAUSE : PLEASE MOUNT ON UNIT #0 "SOURCE DECTAPE #1"
^R
^R
E CHECK               )          The DECtape is checked to insure it
EXECUTE XVM V1A000    ( - - -     is source DECtape #1
DOSRFA159  1          )
XVM/DOS V1A000        )

$PIP
PIP XVM V1A000

>D SY <BLD> M11.12 IMG                              )_ _Excess files are removed
                                                    }    from the skeleton
>D SY <BLD> CODE SRC,BUILD XCT,BUILD XCU            )

>T SY <BLD>_DT DOSRFA 159

>
XVM/DOS V1A000   - - - - -  Handler assemblies begin here

$$JOB
$MACRO
BMACRO XVM V1A000
>BP_DOSRFA 159
%RK05=0
%RP02=0
%BVERS=0
$END
```

```
 EOT
 END OF PASS 1
SIZE=04047     NO ERROR LINES
BMACRO XVM V1A000
>
XVM/DOS V1A000

$$JOB
$PIP
PIP XVM V1A000

>R SY <IOS> RKB. BIN_SY <IOS> DOSRFA BIN

>
XVM/DOS V1A000

$$JOB
$MACRO
BMACRO XVM V1A000
>BP_DOSRFA 159
%RK05=0
%RP02=0
%CVERS=0
$END
 EOT
 END OF PASS 1
SIZE=02350     NO ERROR LINES
BMACRO XVM V1A000
>
XVM/DOS V1A000

$$JOB
$PIP
PIP XVM V1A000

>R SY <IOS> RKC. BIN_SY <IOS> DOSRFA BIN

>
XVM/DOS V1A000

$$JOB
$PIP
PIP XVM V1A000

>D SY <BLD> DOSRFA 159

>T SY <BLD>_DT DTC. 103

>
XVM/DOS V1A000

$$JOB
$MACRO
BMACRO XVM V1A000
>B_DTC. 103
 END OF PASS 1
SIZE=01271  .  NO ERROR LINES
BMACRO XVM V1A000
>
XVM/DOS V1A000

$$JOB
$PIP
PIP XVM V1A000

>D SY <BLD> DTC. 103
```

```
>T SY <BLD>_DT DTD. 108

>
XVM/DOS V1A000

$$JOB
$MACRO
BMACRO XVM V1A000
>B_DTD. 108
 END OF PASS 1
SIZE=03121      NO ERROR LINES
BMACRO XVM V1A000
>
XVM/DOS V1A000

$$JOB
$PIP
PIP XVM V1A000

>D SY <BLD> DTD. 108

>
XVM/DOS V1A000

$$JOB CHANGE DECTAPES
$$PAUSE : PLEASE MOUNT ON UNIT #0 "SOURCE DECTAPE #2"
^R
^R
E CHECK              }     Source tape #2 is checked
EXECUTE XVM V1A000   }
MTA.  108  2
XVM/DOS V1A000

$PIP
PIP XVM V1A000

>T SY <BLD>_DT DTE. 108

>
XVM/DOS V1A000

$$JOB
$MACRO
BMACRO XVM V1A000
>B_DTE. 108
 END OF PASS 1
SIZE=02724      NO ERROR LINES
BMACRO XVM V1A000
>
XVM/DOS V1A000

$$JOB
$PIP
PIP XVM V1A000

>D SY <BLD> DTE. 108

>T SY <BLD>_DT DTF. 108

>
XVM/DOS V1A000

$$JOB
$MACRO
BMACRO XVM V1A000
```

```
>B_DTF. 108
 END OF PASS 1
SIZE=01157      NO ERROR LINES
BMACRO XVM V1A000
>
XVM/DOS V1A000

$$JOB
$PIP
PIP XVM V1A000

>D SY <BLD> DTF. 108

>T SY <BLD>_DT MTA. 108

>
XVM/DOS V1A000

$$JOB
$MACRO
BMACRO XVM V1A000
>B_MTA. 108
 END OF PASS 1
SIZE=04730      NO ERROR LINES
BMACRO XVM V1A000
>
XVM/DOS V1A000

$$JOB
$PIP
PIP XVM V1A000

>D SY <BLD> MTA. 108

>T SY <BLD>_DT MTC. 101

>
XVM/DOS V1A000

$$JOB
$MACRO
BMACRO XVM V1A000
>B_MTC. 101
 END OF PASS 1
SIZE=01264      NO ERROR LINES
BMACRO XVM V1A000
>
XVM/DOS V1A000

$$JOB
$PIP
PIP XVM V1A000

>D SY <BLD> MTC. 101

>T SY <BLD>_DT MTF. 114

>
XVM/DOS V1A000

$$JOB
$MACRO
BMACRO XVM V1A000
>B_MTF. 114
 END OF PASS 1
SIZE=01340      NO ERROR LINES
```

```
BMACRO XVM V1A000
>
XVM/DOS V1A000

$$JOB
$PIP
PIP XVM V1A000

>D SY <BLD> MTF. 114

>T SY <BLD>_DT PPB. 101

>
XVM/DOS V1A000

$$JOB
$MACRO
BMACRO XVM V1A000
>B_PPB. 101
 END OF PASS 1
SIZE=00430      NO ERROR LINES
BMACRO XVM V1A000
>
XVM/DOS V1A000

$$JOB
$PIP
PIP XVM V1A000

>D SY <BLD> PPB. 101

>T SY <BLD>_DT PPC. 101

>
XVM/DOS V1A000

$$JOB
$MACRO
BMACRO XVM V1A000
>B_PPC. 101
 END OF PASS 1
SIZE=00327      NO ERROR LINES
BMACRO XVM V1A000
>
XVM/DOS V1A000

$$JOB
$PIP
PIP XVM V1A000

>D SY <BLD> PPC. 101

>T SY <BLD>_DT PRA. 100

>
XVM/DOS V1A000

$$JOB
$MACRO
BMACRO XVM V1A000
>B_PRA. 100
 END OF PASS 1
SIZE=00711      NO ERROR LINES
BMACRO XVM V1A000
>
XVM/DOS V1A000
```

```
$$JOB
$PIP
PIP XVM V1A000

>D SY <BLD> PRA. 100

>T SY <BLD>_DT PRB. 100

>
XVM/DOS V1A000

$$JOB
$MACRO
BMACRO XVM V1A000
>B_PRB. 100
 END OF PASS 1
SIZE=00453      NO ERROR LINES
BMACRO XVM V1A000
>
XVM/DOS V1A000

$$JOB
$PIP
PIP XVM V1A000

>D SY <BLD> PRB. 100

>T SY <BLD>_DT LPU. 124

>
XVM/DOS V1A000

$$JOB
$MACRO
BMACRO XVM V1A000
>BP_LPU. 124
NOFF=0
$END
 EOT
 FND OF PASS 1
SIZE=00672      NO ERROR LINES
BMACRO XVM V1A000
>
XVM/DOS V1A000

$$JOB
$PIP
PIP XVM V1A000

>D SY <BLD> LPU. 124

>
XVM/DOS V1A000

$$JOB CHANGE DECTAPES
$$PAUSE : PLEASE MOUNT ON UNIT #0 "SOURCE DECTAPE #3"
^R
^R
E CHECK            }
EXECUTE XVM V1A000 }   Source tape #3 is checked
EDIT15141  3
XVM/DOS V1A000

$PIP
PIP XVM V1A000
```

```
>T SY <BLD>_DT CD.DOS 134

>
XVM/DOS V1A000

$$JOB
$MACRO
BMACRO XVM V1A000
>BP_CD.DOS 134
UC15=0
$END
 EOT
 END OF PASS 1
SIZE=00646      NO ERROR LINES
BMACRO XVM V1A000
>
XVM/DOS V1A000

$$JOB
$PIP
PIP XVM V1A000

>D SY <BLD> CD.DOS 134

>T SY <BLD>_DT XYU. 036

>
XVM/DOS V1A000

$$JOB
$MACRO
BMACRO XVM V1A000
>BP_XYU. 036
$END
 EOT
 END OF PASS 1
SIZE=01176      NO ERROR LINES
BMACRO XVM V1A000
>
XVM/DOS V1A000

$$JOB
$PIP
PIP XVM V1A000

>D SY <BLD> XYU. 036

>T SY <BLD>_DT VPA. 113

>
XVM/DOS V1A000

$$JOB
$MACRO
BMACRO XVM V1A000
>BP_VPA. 113
$END
 EOT
 END OF PASS 1
SIZE=01266      NO ERROR LINES
BMACRO XVM V1A000
>
XVM/DOS V1A000

$$JOB
```

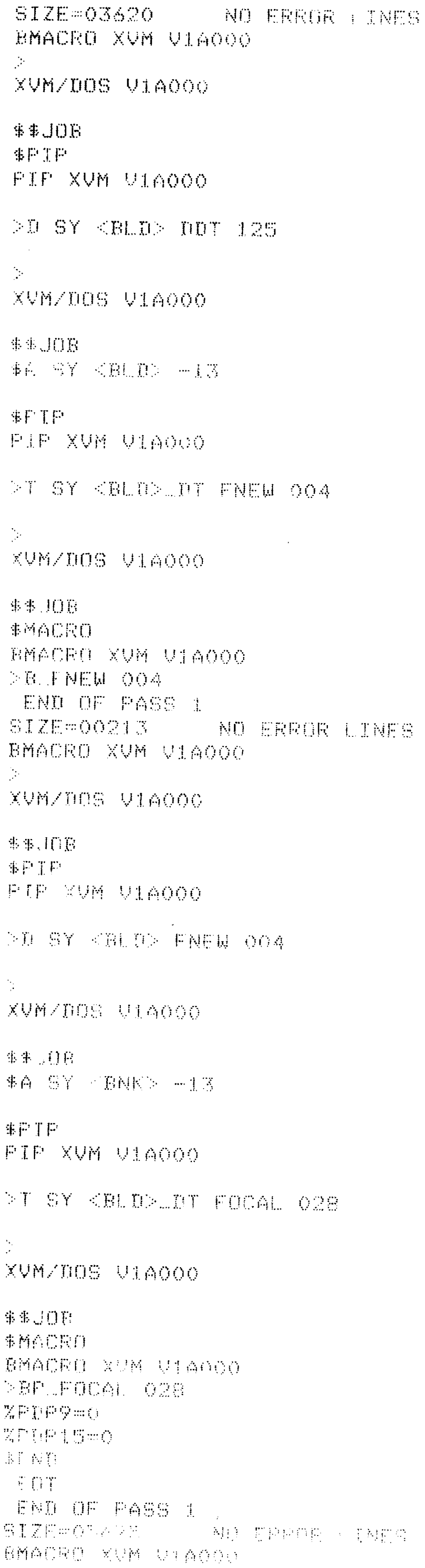
```
$PIP
PIP XVM V1A000

>D SY <BLD> VPA. 113

>T SY <BLD>_DT VTA. 006

>
XVM/DOS V1A000

$$JOB
$MACRO
BMACRO XVM V1A000
>B_VTA. 006
 END OF PASS 1
SIZE=00743      NO ERROR LINES
BMACRO XVM V1A000
>
XVM/DOS V1A000

$$JOB
$PIP
PIP XVM V1A000

>D SY <BLD> VTA. 006

>T SY <BLD>_DT VWA. 005

>
XVM/DOS V1A000

$$JOB
$MACRO
BMACRO XVM V1A000
>B_VWA. 005
 END OF PASS 1
SIZE=00325      NO ERROR LINES
BMACRO XVM V1A000
>
XVM/DOS V1A000

$$JOB
$PIP
PIP XVM V1A000

>D SY <BLD> VWA. 005

>T SY <BLD>_DT LKA. 002

>
XVM/DOS V1A000

$$JOB
$MACRO
BMACRO XVM V1A000
>B_LKA. 002
 END OF PASS 1
SIZE=00740      NO ERROR LINES
BMACRO XVM V1A000
>
XVM/DOS V1A000

$$JOB
$PIP
PIP XVM V1A000
```

```
>D SY <BLD> IKA. 002

>
XVM/DOS V1A000  - - - - End handler assemblies

$$JOB
$PIP
PIP XVM V1A000

>C SY <PAG>_SY <BNK> - -Page mode was requested - the <PAG>
                         UIC is created
>
XVM/DOS V1A000  - - - - System program assembly and library
                         construction begins here
$$JOB
$A SY <PER> -13

$PIP
PIP XVM V1A000

>T SY <BLD> , _DT NUVAL 001,FORT 003

>
XVM/DOS V1A000

$$JOB
$MACRO
BMACRO XVM V1A000
>B_NUVAL 001
 END OF PASS 1
SIZE=00022      NO ERROR LINES
BMACRO XVM V1A000
>B_FORT 003
 END OF PASS 1
SIZE=00201      NO ERROR LINES
BMACRO XVM V1A000
>
XVM/DOS V1A000

$$JOB
$PIP
PIP XVM V1A000

>D SY <BLD> NUVAL 001,FORT 003

>T SY <PER>_DT VECTOR SRC

>
XVM/DOS V1A000

$$JOB
$A SY <BLD> -13

$PIP
PIP XVM V1A000

>T SY <BLD>_DT EDIT15 141

>
XVM/DOS V1A000

$$JOB
$MACRO
BMACRO XVM V1A000
>BP_EDIT15 14
EDITVP=0
```

```
$END
 EOT
 END OF PASS 1
SIZE=10373      NO ERROR LINES
BMACRO XVM V1A000
>
XVM/DOS V1A000

$$JOB
$PIP
PIP XVM V1A000

>R SY <BLD> EDITVP BIN_SY <BLD> EDIT15 BIN

>
XVM/DOS V1A000

$$JOB
$A SY <BLD> -13

$MACRO
BMACRO XVM V1A000
>BP_EDIT15 141
EDITVT=0
$END
 EOT
 END OF PASS 1
SIZE=10370      NO ERROR LINES
BMACRO XVM V1A000
>
XVM/DOS V1A000

$$JOB
$PIP
PIP XVM V1A000

>R SY <BLD> EDITVT BIN_SY <BLD> EDIT15 BIN

>D SY <BLD> EDIT15 141

>
XVM/DOS V1A000

$$JOB
$A SY <PER> -13

$$PAUSE : PLEASE MOUNT ON UNIT #0 "SOURCE DECTAPE #4"
^R
^R                          }
E CHECK                     }   Source tape #4 is checked
EXECUTE XVM V1A000          }
CHAIN 179  4
XVM/DOS V1A000

$PIP
PIP XVM V1A000

>T SY <BLD> , _DT VTPRIM 136,DYLDR 002

>T SY <BLD> , _DT LTORPB 003,TRACK 007

>
XVM/DOS V1A000

$$JOB
$MACRO
```

```
BMACRO XVM V1A000
>B_VTPRIM 136
 END OF PASS 1
SIZE=01556      NO ERROR LINES
BMACRO XVM V1A000
>B_DYLDR 002
 END OF PASS 1
SIZE=00370      NO ERROR LINES
BMACRO XVM V1A000
>B_LTORPB 003
 END OF PASS 1
SIZE=00112      NO ERROR LINES
BMACRO XVM V1A000
>B_TRACK 007
 END OF PASS 1
SIZE=00513      NO ERROR LINES
BMACRO XVM V1A000
>
XVM/DOS V1A000

$$JOB
$PIP
PIP XVM V1A000

>D SY <BLD> VTPRIM 136,DYLDR 002,LTORPB 003,TRACK 007

>T SY <PER> _ DT ROTATE SRC

>T SY <PER> _ DT CIRCLE SRC

>
XVM/DOS V1A000

$$JOB
$A SY <BLD> -13

$PIP
PIP XVM V1A000

>T SY <BLD>_DT CHAIN 179

>
XVM/DOS V1A000

$$JOB
$MACRO
BMACRO XVM V1A000
>B_CHAIN 179
 END OF PASS 1
SIZE=12106      NO ERROR LINES
BMACRO XVM V1A000
>
XVM/DOS V1A000

$$JOB
$PIP
PIP XVM V1A000

>D SY <BLD> CHAIN 179

>
XVM/DOS V1A000

$$JOB
$$PAUSE : PLEASE MOUNT ON UNIT #0 "SOURCE DECTAPE #5"
^R
```

```
^R
E CHECK
EXECUTE XVM V1A000          Source tape #5 is checked
.LOAD 144  5
XVM/DOS V1A000

$A SY <BNK> -13

$PIP
PIP XVM V1A000

>T SY <BLD>_DT .LOAD 144

>
XVM/DOS V1A000

$$JOB
$MACRO
BMACRO XVM V1A000
>BP_.LOAD 144
%PDP9=0
$END
 EOT
 END OF PASS 1
SIZE=03670      NO ERROR LINES
BMACRO XVM V1A000
>
XVM/DOS V1A000

$$JOB
$A SY <PAG> -13

$MACRO
BMACRO XVM V1A000
>B_.LOAD 144
 END OF PASS 1
SIZE=03721      NO ERROR LINES
BMACRO XVM V1A000
>
XVM/DOS V1A000

$$JOB
$PIP
PIP XVM V1A000

>D SY <BLD> .LOAD 144

>
XVM/DOS V1A000

$$JOB
$A SY <BLD> -13

$PIP
PIP XVM V1A000

>T SY <BLD>_DT DDT 125

>
XVM/DOS V1A000

$$JOB
$MACRO
BMACRO XVM V1A000
>B_DDT 125
 END OF PASS 1
```



```
SIZE=03620      NO ERROR LINES
BMACRO XVM V1A000
>
XVM/DOS V1A000

$$JOB
$PIP
PIP XVM V1A000

>D SY <BLD> DDT 125

>
XVM/DOS V1A000

$$JOB
$A SY <BLD> -13

$PIP
PIP XVM V1A000

>T SY <BLD>_DT FNEW 004

>
XVM/DOS V1A000

$$JOB
$MACRO
BMACRO XVM V1A000
>B_FNEW 004
 END OF PASS 1
SIZE=00213      NO ERROR LINES
BMACRO XVM V1A000
>
XVM/DOS V1A000

$$JOB
$PIP
PIP XVM V1A000

>D SY <BLD> FNEW 004

>
XVM/DOS V1A000

$$JOB
$A SY <BNK> -13

$PIP
PIP XVM V1A000

>T SY <BLD>_DT FOCAL 028

>
XVM/DOS V1A000

$$JOB
$MACRO
BMACRO XVM V1A000
>BP_FOCAL 028
%PDP9=0
%PDP15=0
.END
 EOT
 END OF PASS 1
SIZE=0[illegible]      NO ERROR LINES
BMACRO XVM V1A000
```

```
>
XVM/DOS V1A000

$$JOB
$A SY <PAG> -13

$MACRO
BMACRO XVM V1A000
>B_FOCAL 028
 END OF PASS 1
SIZE=05675      NO ERROR LINES
BMACRO XVM V1A000
>
XVM/DOS V1A000

$$JOB
$PIP
PIP XVM V1A000

>D SY <BLD> FOCAL 028

>
XVM/DOS V1A000

$$JOB
$A DT -11

$A SY <BLD> -14

$$PAUSE : PLEASE MOUNT ON UNIT #0 "SOURCE DECTAPE #6"
^R
^R
E CHECK              }
EXECUTE XVM V1A000   }     Source tape #6 is checked
F4MP1 069  6         }
XVM/DOS V1A000

$PIP
PIP XVM V1A000

>T SY <BLD>_DT F4MP1 069

>
XVM/DOS V1A000

$$JOB
$$PAUSE : PLEASE MOUNT ON UNIT #0 "SOURCE DECTAPE #7"
^R
^R
E CHECK              }
EXECUTE XVM V1A000   }     Source tape #7 is checked
F4MP2 069  7
XVM/DOS V1A000

$A SY <BLD> -13

$MACRO
BMACRO XVM V1A000
>BPFZ_F4MP1 069,F4MP2 069
%FPP=0
$END
 EOT
 END OF PASS 1
SI7E=17064      NO ERROR LINES
BMACRO XVM V1A000
>
```

```
XVM/DOS V1A000

$$JOB
$A SY <BLD> -11

$PIP
PIP XVM V1A000

>D SY <BLD> F4MP1 069

>
XVM/DOS V1A000

$$JOB
$A SY <BLD> -13

$PIP
PIP XVM V1A000

>T SY <BLD>_DT DUMP 022

>
XVM/DOS V1A000

$$JOB
$MACRO
BMACRO XVM V1A000
>B_DUMP 022
 END OF PASS 1
SIZE=02251      NO ERROR LINES
BMACRO XVM V1A000
>
XVM/DOS V1A000

$$JOB
$PIP
PIP XVM V1A000

>D SY <BLD> DUMP 022

>
XVM/DOS V1A000

$$JOB CHANGE DECTAPES
$$PAUSE ↕ PLEASE MOUNT ON UNIT #0 "SOURCE DECTAPE #8"
^R
^R
E CHECK               }
EXECUTE XVM V1A000    }    Source tape #8 is checked
SRCCOM012  8          }
XVM/DOS V1A000

$PIP
PIP XVM V1A000

>T SY <BLD>_DT MTDUMP 114

>
XVM/DOS V1A000

$$JOB
$MACRO
BMACRO XVM V1A000
>B_MTDUMP 114
 END OF PASS 1
SIZE=05174      NO ERROR LINES
```

```
BMACRO XVM V1A000
>
XVM/DOS V1A000

$$JOB
$PIP
PIP XVM V1A000

>D SY <BLD> MTDUMP 114

>T SY <BLD>_DT SRCCOM 012

>
XVM/DOS V1A000

$$JOB
$MACRO
BMACRO XVM V1A000
>B_SRCCOM 012
 END OF PASS 1
SIZE=04647      NO ERROR LINES
BMACRO XVM V1A000
>
XVM/DOS V1A000

$$JOB
$PIP
PIP XVM V1A000

>D SY <BLD> SRCCOM 012

>T SY <BLD>_DT 8TRAN 003

>
XVM/DOS V1A000

$$JOB
$MACRO
BMACRO XVM V1A000
>B_8TRAN 003
 END OF PASS 1
SIZE=03717      NO ERROR LINES
BMACRO XVM V1A000
>
XVM/DOS V1A000

$$JOB
$PIP
PIP XVM V1A000

>D SY <BLD> 8TRAN 003

>
XVM/DOS V1A000

$$JOB CHANGE DECTAPES
$$PAUSE : PLEASE MOUNT ON UNIT #0 "SOURCE DECTAPE #9"
^R
^R
E CHECK              }
EXECUTE XVM V1A000   }   Source tape #9 is checked
NRBOSS061  9
XVM/DOS V1A000

$PIP
PIP XVM V1A000
```

```
>T SY <CTP>_DT DLG PRC (A)
>T SY <CTP>_DT FIL PRC (A)
>T SY <CTP>_DT FOR PRC (A)
>T SY <CTP>_DT MIC PRC (A)
>T SY <CTP>_DT ASG PRC (A)
 T SY <CTP>_DT LOG PRC (A)
>T SY <CTP>_DT DIR PRC (A)
>T SY <CTP>_DT JOB PRC (A)
>T SY <CTP>_DT BUF PRC (A)
>T SY <CTP>_DT CHN PRC (A)
>T SY <CTP>_DT MNT PRC (A)
>T SY <CTP>_DT BNK PRC (A)
>T SY <CTP>_DT OVL PRC (A)
>T SY <CTP>_DT DOS PRC (A)
>T SY <CTP>_DT PRT PRC (A)
>T SY <CTP>_DT ASM PRC (A)
>T SY <CTP>_DT QDP PRC (A)
>T SY <CTP>_DT MSG PRC (A)
>T SY <CTP>_DT XCT PRC (A)
>T SY <CTP>_DT LST PRC (A)
>T SY <CTP>_DT LIB PRC (A)
>T SY <CTP>_DT LCM PRC (A)
>T SY <CTP>_DT DMP PRC (A)
>T SY <CTP>_DT MSW PRC (A)
>T SY <CTP>_DT LNK PRC (A)
>T SY <CTP>_DT NDR PRC (A)
>T SY <CTP>_DT MAP PRC (A)
>T SY <CTP>_DT CMP PRC (A)
>T SY <CTP>_DT MAC PRC (A)
>T SY <CTP>_DT KEP PRC (A)
>T SY <CTP>_DT RUN PRC (A)
>T SY <CTP>_DT PAG PRC (A)
>T SY <CTP>_DT XVM PRC (A)
```

- The BOSS procedure files are copied into the <CTP> UIC

```
>T SY <BLD> _ DT NRBOSS 061

>T SY <BLD> _ DT B.PRE 022

>
XVM/DOS V1A000

$$JOB
$MACRO
BMACRO XVM V1A000
>B_B.PRE 022
 END OF PASS 1
SIZE=01260      NO ERROR LINES
BMACRO XVM V1A000
>B_NRBOSS 061
 END OF PASS 1
SIZE=06310      NO ERROR LINES
BMACRO XVM V1A000
>
XVM/DOS V1A000

$$JOB
$PIP
PIP XVM V1A000

>D SY <BLD> NRBOSS 061,B.PRE 022

>
XVM/DOS V1A000

$$JOB
$A DT -11

$A SY <BLD> -14

$$PAUSE : PLEASE MOUNT ON UNIT #0 "SOURCE DECTAPE #10"
^R
^R
E CHECK                )
EXECUTE XVM V1A000     (    Source tape #10 is checked
RESMNA210 10           )
XVM/DOS V1A000

$PIP
PIP XVM V1A000

>T SY <BLD> _DT RESMNA 210

>
XVM/DOS V1A000

$$JOB
$$PAUSE : PLEASE MOUNT ON UNIT #0 "SOURCE DECTAPE #11"
^R
^R
E CHECK                )
EXECUTE XVM V1A000     (    Source tape #11 is checked
RESMNB210 11           )
XVM/DOS V1A000

$MACRO
BMACRO XVM V1A000
>BPFZ_RESMNA 210,RESMNB 210
CDTCB=0
PLTCB=0
```

```
$END
 EOT
 END OF PASS 1
SIZE=15332      NO ERROR LINES
BMACRO XVM V1A000
>
XVM/DOS V1A000

$$JOB
$A SY <BLD> -11

$PIP
PIP XVM V1A000

>D SY <BLD> RESMNA 210

>
XVM/DOS V1A000

$$JOB
$$PAUSE : PLEASE MOUNT ON UNIT #0 "SOURCE DECTAPE #12"
^R
^R
E CHECK
EXECUTE XVM V1A000  }   Source tape #12 is checked
DOSNRM108 12
XVM/DOS V1A000

$PIP
PIP XVM V1A000

>T SY <BLD>_DT DOSNRM 108

>
XVM/DOS V1A000

$$JOB
$MACRO
BMACRO XVM V1A000
>BP_DOSNRM 108
NOMODE=0
$END
 EOT
 END OF PASS 1
SIZE=12157      NO ERROR LINES
BMACRO XVM V1A000
>
XVM/DOS V1A000

$$JOB
$PIP
PIP XVM V1A000

>D SY <BLD> DOSNRM 108

>
XVM/DOS V1A000     -    Paper tapes are punched during this phase

$$JOB
$$PAUSE : CHECK THAT THE PUNCH HAS AT LEAST 3/4 BOX OF P'TAPE
^R
^R
PIP
PIP XVM V1A000

>T PP_SY <BLD> QDMP IMG (L)
```

```
>D SY <BLD> QDMP IMG

>
XVM/DOS V1A000

$$JOB
$$PAUSE : PLEASE LABEL THE NEW PAPER TAPE "QDMP XVM V1A"
^R
^R
$JOB
$$PAUSE : CHECK THAT THE PUNCH HAS AT LEAST 3/4 BOX OF P'TAPE
^R
^R
PIP
PIP XVM V1A000

>T PP_SY <BLD> UDMP IMG (I)

>D SY <BLD> UDMP IMG

>
XVM/DOS V1A000

$$JOB
$$PAUSE : PLEASE LABEL THE NEW PAPER TAPE "UDMP11 XVM V1A"
^R
^R
$JOB
$$PAUSE : PLEASE MOUNT ON UNIT #0 "SOURCE DECTAPE #13"
^R
^R
C CHECK
EXECUTE XVM V1A000
FIREX 154 13
-WRONG TAPE MOUNTED
-PLEASE MOUNT TAPE #13
-TYPE "Y",CR WHEN READY
Y
-PLEASE MOUNT TAPE #13
-TYPE "Y",CR WHEN READY
Y
```

Source tape #13 is checked and found invalid, twice. The correct tape is finally mounted

```
XVM/DOS V1A000

$$JOB
$PIP
PIP XVM V1A000

>T SY <BLD>_DT MACIMG 010

>T SY <BLD>_DT MACINT 021

>
XVM/DOS V1A000

$$JOB
$MACRO
BMACRO XVM V1A000
>B_MACIMG 010
 END OF PASS 1
SIZE=01207      NO ERROR LINES
BMACRO XVM V1A000
>B_MACINT 021
 END OF PASS 1
SIZE=17635      NO ERROR LINES
```

```
BMACRO XVM V1A000
>
XVM/DOS V1A000

$$JOB
$PIP
PIP XVM V1A000

>D SY <BLD> MACIMG 010,MACINT 021

>T SY <BLD>_DT SPLIMG 012

>T SY <BLD>_DT SPOL15 163

>T SY <BLD>_DT SPLGEN 017

>
XVM/DOS V1A000

$$JOB
$MACRO
BMACRO XVM V1A000
>B_SPLIMG 012
 END OF PASS 1
SIZE=01026      NO ERROR LINES
BMACRO XVM V1A000
>B_SPOL15 163
 END OF PASS 1
SIZE=04641      NO ERROR LINES
BMACRO XVM V1A000
>B_SPLGEN 017
 END OF PASS 1
SIZE=05245      NO ERROR LINES
BMACRO XVM V1A000
>
XVM/DOS V1A000

$$JOB
$PIP
PIP XVM V1A000

>D SY <BLD> SPLIMG 012,SPOL15 163,SPLGEN 017

>
XVM/DOS V1A000   - - - -  End assembly phase

$$JOB
$BATCH SY <BLD> PIP2 - - The next BATCH file is linked to.
XVM/DOS V1A000            It is PIP2 BAT.

$$JOB PIP2    - - - - - - This phase renames handlers and deletes
$MICLOG SYS               unneeded files.

$PIP
PIP XVM V1A000

>D SY <BLD> MACRO BAT

>R SY <IOS> LPA. BIN_SY <IOS> LPU. BIN

>R SY <IOS> CDB. BIN_SY <IOS> CD.DOS BIN

>R SY <IOS> XYA. BIN_SY <IOS> XYU. BIN

>
XVM/DOS V1A000 - - - -  End PIP2 phase
```

Building the System

```
$$JOB
$BATCH SY <BLD> UPDATE - -The next BATCH file is linked to.
XVM/DOS V1A000              It is UPDATE BAT.

$$JOB UPDATE  - - - - - - This phase renames local symbols from
$MICLOG SYS                 the handlers, and, compacts the
                            resulting binaries on the disk.
$A SY <IOS> -14

$A SY <IOX> -15

$K ON

$UPDATE
UPDATE XVM V1A000
>US_RKB.
>C
UPDATE XVM V1A000
>US_RKC.
>C
UPDATE XVM V1A000
>US_DTC.
>C
UPDATE XVM V1A000
>US_DTD.
>C
UPDATE XVM V1A000
>US_DTE.
>C
UPDATE XVM V1A000
>US_DTF.
>C
UPDATE XVM V1A000
>US_MTA.
>C
UPDATE XVM V1A000
>US_MTC.
>C
UPDATE XVM V1A000
>US_MTF.
>C
UPDATE XVM V1A000
>US_PPB.
>C
UPDATE XVM V1A000
>US_PPC.
>C
UPDATE XVM V1A000
>US_PRA.
>C
UPDATE XVM V1A000
>US_PRB.
>C
UPDATE XVM V1A000
>US_LPA.
>C
UPDATE XVM V1A000
>US_CDB.
>C
UPDATE XVM V1A000
>US_XYA.
>C
UPDATE XVM V1A000
>US_VPA.
>C
```

Building the System

```
UPDATE XVM V1A000
>US_VTA.
>C
UPDATE XVM V1A000
>US_VWA.
>C
UPDATE XVM V1A000
>US_LKA.
>C
UPDATE XVM V1A000
>
XVM/DOS V1A000

$$JOB
$PIP
PIP XVM V1A000

>T SY <IOX>_SY <IOS> RKA. BIN

>T SY <IOX>_SY <IOS> PPA. BIN          Transfer skeleton handlers

>T SY <IOX>_SY <IOS> DTA. BIN

>N SY <IOS>

>C SY <IOS>_SY <IOX>             Compact the disk

>N SY <IOX> (K)

>
XVM/DOS V1A000 - - - End update phase

$$JOB
$BATCH SY <BLD> SGEN - - The next BATCH file is linked to.
XVM/DOS V1A000              It is SGEN BAT.

$$JOB SGEN - -  This phase - deletes unused handlers from
$$JOB            "SGNBLK" and from the skip chain; sets
$MICLOG SYS      several system parameters; and enters
                 the selected system programs into "SYSBLK".
$SGEN

SGEN XVM V1A000

        SYSTEM UPDATE ON RK0

A. ALTER SYSTEM PARAMETERS? (N)      Y
        API? (N)      Y
        TAB SIMULATION? (Y) _
        FILL CHARACTERS? (Y) _
        MIC[SYS] _
        DEFAULT # BUFFERS[3]      4
        # WORDS/BUFFER[500] _
        UC15 CONFIG? (Y)      Y
        POLLER? (Y)      Y
        DEFAULT FILES PROTECTION CODE[2] _
        RESIDENT PATCH AREA SIZE[0] _
        PAGE MODE SYSTEM? (N) _
        XVM MODE SYSTEM? (N) _
        SYSTEM MEMORY SIZE  [ 24K] _
        60 CPS? (Y) _

B. ALTER I/O DEVICES OR HANDLERS? (N)      Y
        DELETE DISCARDED HANDLERS? (Y)
        TO BE KEPT:
```

```
PR? ($)       Y
PRA? (Y) _
PRB? (Y) _
NEW HANDLERS:

>_
RSF=700101? (Y) _
NEW SKIPS:

>_
PP? ($)       Y
PPA? (Y) _
PPB? (Y) _
PPC? (Y) _
NEW HANDLERS:

>_
PSF=700201? (Y) _
NEW SKIPS:

>_
DT? ($)       Y
DTA? (Y) _
DTC? (Y) _
DTD? (Y) _
DTE? (Y) _
DTF? (Y) _
NEW HANDLERS:

>_
DTDF=707601? (Y) _
DTEF=707561? (Y) _
NEW SKIPS:

>_
DK? ($)       N
DP? ($)       N
MT? ($)       Y
MTA? (Y) _
MTC? (Y) _
MTF? (Y) _
NEW HANDLERS:

>_
MTSF=707341? (Y) _
NEW SKIPS:

>_
LP? ($) _
CD? ($) _
VP? ($) _
VT? ($) _
VW? ($) _
LK? ($) _
RK? ($)       Y
RKB? (Y) _
RKC? (Y) _
NEW HANDLERS:

>_
RKSF=706101? (Y) _
NEW SKIPS:

>_
XY? ($) _
HANDLERLESS DEVICE? ($) _
```

```
C. ADD NEW DEVICE? (N) _

D. CHANGE SKIP CHAIN? (N) _

E. ALTER DEVICE PARAMETERS? (N) _

F. ALTER .DAT SLOTS? (N)      Y
        # OF POSITIVE .DAT SLOTS[20] _
        DISPLAY .DAT SLOTS? (Y)      N
        NEW ASSIGNMENTS:

        >  A CDB 5/LPA 6
        >_

G. CHANGE SYS FILES? (N) _

H. ADD SYS PROG? (N)      Y
        PROG NAME[]      EDITVP
        # OF BLOCKS[]     26
        OVERLAY NAME[] _
        BUFFS[0]      4
        .DAT SLOTS:

        >    -15,-14,-13,-12,-10,10
        >_

H. ADD SYS PROG? (N)      Y
        PROG NAME[]      EDITVT
        # OF BLOCKS[]     26
        OVERLAY NAME[] _
        BUFFS[0]      4
        .DAT SLOTS:

        > -15,-14,-13,-12,-10
        >_

H. ADD SYS PROG? (N)      Y
        PROG NAME[] CHAIN
        # OF BLOCKS[]     30
        OVERLAY NAME[] _
        BUFFS[0]      4
        .DAT SLOTS:

        >     -6,-5,-4,-1
        >_

H. ADD SYS PROG? (N)      Y
        PROG NAME[]     F4
        # OF BLOCKS[]     40
        OVERLAY NAME[] _
        BUFFS[0]      3
        .DAT SLOTS:

        >     -13,-12,-11
        >_

H. ADD SYS PROG? (N)      Y
        PROG NAME[]  DUMP
        # OF BLOCKS[]      5
        OVERLAY NAME[] _
        BUFFS[0]      2
        .DAT SLOTS:

        >    -12,-14
        >_
```

```
H. ADD SYS PROG? (N)      Y
        PROG NAME[]     MTDUMP
        # OF BLOCKS[]     13
        OVERLAY NAME[] _
        BUFFS[0]      2
        .DAT SLOTS:

        >  1,3
        >_

H. ADD SYS PROG? (N)      Y
        PROG NAME[]     SRCCOM
        # OF BLOCKS[]     13
        OVERLAY NAME[] _
        BUFFS[0]      3
        .DAT SLOTS:

        >    -15,-14,-12
        >_

H. ADD SYS PROG? (N)      Y
        PROG NAME[] 8TRAN
        # OF BLOCKS[]     13
        OVERLAY NAME[] _
        BUFFS[0]      2
        .DAT SLOTS:

        >        -15,-14
        >_

H. ADD SYS PROG? (N)      Y
        PROG NAME[]  BOSS
        # OF BLOCKS[]     15
        OVERLAY NAME[] _
        BUFFS[0]      3
        .DAT SLOTS:

        >-15,-14,-11,5,6
        >_

H. ADD SYS PROG? (N)      Y
        PROG NAME[] R.PRE
        # OF BLOCKS[]      3
        OVERLAY NAME[] _
        BUFFS[0]      2
        .DAT SLOTS:

        >    -15,-14,5,6
        >_

H. ADD SYS PROG? (N)      Y
        PROG NAME[] MAC11
        # OF BLOCKS[]     40
        OVERLAY NAME[] _
        BUFFS[0]      2
        .DAT SLOTS:

        >   -11,-12
        >_

H. ADD SYS PROG? (N)      Y
        PROG NAME[]     MCLOAD
        # OF BLOCKS[]     5
        OVERLAY NAME[] _
        BUFFS[0]      1
```

```
         .DAT SLOTS:

         >  -14
         >_

H. ADD SYS PROG? (N)      Y
         PROG NAME[]  SPOOL
         # OF BLOCKS[]     55
         OVERLAY NAME[] _
         BUFFS[0]      1
         .DAT SLOTS:

         >  -14
         >_

H. ADD SYS PROG? (N)      Y
         PROG NAME[]     SPLGEN
         # OF BLOCKS[]     25
         OVERLAY NAME[] _
         BUFFS[0]      1
         .DAT SLOTS:

         >  -14
         >_

H. ADD SYS PROG? (N)      Y
         PROG NAME[]     SPLOAD
         # OF BLOCKS[]      5
         OVERLAY NAME[] _
         BUFFS[0]      1
         .DAT SLOTS:

         >  -14
         >_

H. ADD SYS PROG? (N)      N

                 MODIFYING SYSTEM(^P,^C IGNORED)

                 SGEN COMPLETE

XVM/DOS V1A000 - - - - - - -  End SGEN phase

$$JOB
$BATCH SY <BLD> PATCH  - - -  The next BATCH file is linked to.
XVM/DOS V1A000                 It is PATCH BAT

$$JOB PATCH   - - - - - - -   This phase actually installs the
$MICLOG SYS                    system programs

$BANK ON

$BUFFS 4

$A SY -14

$A SY <BLD> -10

$K ON

$PATCH
PATCH XVM V1A000
>EDITVP
>READR EDITVP
>EDITVT
>READR EDITVT
```

```
>CHAIN
>READR CHAIN
>F4
>READR F4MP2
>DUMP
>READR DUMP
>MTDUMP
>READR MTDUMP
>SRCCOM
>READR SRCCOM
>8TRAN
>READR 8TRAN
>BOSS
>FA
>00221/070417>20
 00222/251455>6400
 00223/774607>420
 00224/025620>M  - - - - Note that this is not a real error
ILLEGAL COMMAND
>READ NRBOSS
>B.PRE
>READR 16077 B.PRE
>RESMON
>READ RESMNB
>DOS15
>READR 16077 DOSNRM
>MCLOAD
>READR MACIMG
>MAC11
>FA
>00237/623147>1
 00240/271466>17635
 00241/054642>17500
 00242/150314>M  - - - - Note that this is not a real error
ILLEGAL COMMAND
>READ MACINT
>SPLOAD
>READR SPLIMG
>SPOOL
>READR SPOL15
>SPLGEN
>READR SPLGEN
>
XVM/DOS V1A000   - - - - End PATCH phase

$$JOB
$BATCH SY <BLD> PIP3- - The next BATCH file is linked to.
XVM/DOS V1A000           It is PIP3 BAT

$$JOB PIP3  - - - - - - This phase removes unneeded binaries;
$MICLOG SYS              copies library - completes the
                         construction of the <PAG> and <BNK>
$PIP                     UIC's; and, loads the UNICHANNEL sources.
PIP XVM V1A000

>D SY <BLD> PIP2 BAT,UPDATE BAT

>D SY <BLD> SGEN BAT,PATCH BAT

>D SY <BLD> EDITVP BIN

>D SY <BLD> EDITVT BIN

>D SY <BLD> CHAIN BIN

>T SY <BNK> DDT BIN,SY <BLD> DDT BIN
```

```
>T SY <PAG> DDT BIN_SY <BLD> DDT BIN

>D SY <BLD> DDT BIN

>T SY <BNK>_SY <BLD> FNEW BIN

>T SY <BNK> _ SY <LIB> .LIBNF BIN

>T SY <PAG>_SY <BLD> FNEW BIN

>T SY <PAG>_SY <LIB> .LIBNF BIN

>D SY <BLD> FNEW BIN

>D SY <BLD> F4MP2 BIN

>T SY <BNK> .LIBR BIN_SY <LIB> .LIBF BIN

>T SY <PAG> .LIBR BIN_SY <LIB> .LIBF BIN

>N SY <LIB> (K)

>D SY <BLD> DUMP BIN

>D SY <BLD> MTDUMP BIN

>D SY <BLD> SRCCOM BIN

>D SY <BLD> STRAN BIN

>D SY <BLD> B.PRE BIN

>D SY <BLD> NRBOSS ABS

>D SY <BLD> RESMNB ABS

>D SY <BLD> DOSNRM BIN

>D SY <BLD> MACIMG BIN

>D SY <BLD> MACINT ABS

>T SY <BLD>_DT PIREX 154

>
XVM/DOS V1A000

$$JOB
$$PAUSE : PLEASE MOUNT ON UNIT #0 "SOURCE DECTAPE #14"
^R
^R
E CHECK
EXECUTE XVM V1A000
SPOL11145 14
XVM/DOS V1A000

$PIP
PIP XVM V1A000

>D SY <BLD> SPLIMG BIN

>D SY <BLD> SPOL15 BIN

>D SY <BLD> SPLGEN BIN
```

Source tape #14 is checked

```
>T SY <BLD> _ DT SPOL11 145

>
XVM/DOS V1A000 - - - - -  End PIP phase

$$JOB
$BATCH SY <BLD> UPDAT2- - The next BATCH file is linked to.
XVM/DOS V1A000             It is UPDATE BAT

$$JOB UPDAT2  - - - - - - This phase is used to delete local
$$JOB                      symbols from binary programs found
$MICLOG SYS                in the <PAG> and <BNK> UIC's.

$A SY <PAG> -14,-15

$UPDATE
UPDATE XVM V1A000
>US_.LOAD
>C
NAMED FILE IS ALREADY ON OUTPUT DEVICE
DO YOU WISH TO CONTINUE (Y/N)
>Y
UPDATE XVM V1A000
>US_DDT
>C
NAMED FILE IS ALREADY ON OUTPUT DEVICE
DO YOU WISH TO CONTINUE (Y/N)
>Y
UPDATE XVM V1A000
>US_FOCAL
>C
NAMED FILE IS ALREADY ON OUTPUT DEVICE
DO YOU WISH TO CONTINUE (Y/N)
>Y
UPDATE XVM V1A000
>US_FNEW
>C
NAMED FILE IS ALREADY ON OUTPUT DEVICE
DO YOU WISH TO CONTINUE (Y/N)
>Y
UPDATE XVM V1A000
>
XVM/DOS V1A000

$$JOB
$A SY <BNK> -14,-15

$UPDATE
UPDATE XVM V1A000
>US_.LOAD
>C
NAMED FILE IS ALREADY ON OUTPUT DEVICE
DO YOU WISH TO CONTINUE (Y/N)
>Y
UPDATE XVM V1A000
>US_DDT
>C
NAMED FILE IS ALREADY ON OUTPUT DEVICE
DO YOU WISH TO CONTINUE (Y/N)
>Y
UPDATE XVM V1A000
>US_FOCAL
>C
NAMED FILE IS ALREADY ON OUTPUT DEVICE
DO YOU WISH TO CONTINUE (Y/N)
>Y
```

```
UPDATE XVM V1A000
>US_FNEW
>C
NAMED FILE IS ALREADY ON OUTPUT DEVICE
DO YOU WISH TO CONTINUE (Y/N)
>Y
UPDATE XVM V1A000
>
XVM/DOS V1A000    - - - - - The UPDATE phase is completed

$$JOB
$BATCH SY <BLD> MAC11- -   The next BATCH file is linked to.
XVM/DOS V1A000              It is MAC11 BAT

$$JOB MAC11   - - - - - -  This phase installs MAC11; assembles
$A SY <BLD> -11             PIREX and the SPOOLER: and completes
                            the system construction process.
$K ON

$PIP
PIP XVM V1A000

>D SY <BLD> PIP3 BAT,UPDAT2 BAT

>T SY <BLD> PIREX SRC_SY <BLD> ULP PRM,PIREX 154 (AW)- Note that the
                                                         next 21 lines
>D SY <BLD> PIREX 154                                    cause various
                                                         assembly para-
>R SY <BLD> PIREX 154_SY <BLD> PIREX SRC                 meters to be
                                                         appended to the
>T SY <BLD> SPOL11 SRC_SY <BLD> ULP PRM,SPOL11 145 (AW)front of the
                                                         PIREX and SPOL11
>D SY <BLD> SPOL11 145                                   files.

>R SY <BLD> SPOL11 145_SY <BLD> SPOL11 SRC

>T SY <BLD> PIREX SRC_SY <BLD> UCD PRM,PIREX 154 (AW)

>D SY <BLD> PIREX 154

>R SY <BLD> PIREX 154_SY <BLD> PIREX SRC

>T SY <BLD> SPOL11 SRC_SY <BLD> UCD PRM,SPOL11 145 (AW)

>D SY <BLD> SPOL11 145

>R SY <BLD> SPOL11 145_SY <BLD> SPOL11 SRC

>T SY <BLD> SPOL11 SRC_SY <BLD> UXY PRM,SPOL11 145 (AW)

>D SY <BLD> SPOL11 145

>R SY <BLD> SPOL11 145_SY <BLD> SPOL11 SRC

>T SY <BLD> PIREX SRC_SY <BLD> UXY PRM,PIREX 154 (AW)

>D SY <BLD> PIREX 154

>R SY <BLD> PIREX 154_SY <BLD> PIREX SRC

>T SY <BLD> PIREX SRC_SY <BLD> URK PRM,PIREX 154 (AW)

>D SY <BLD> PIREX 154

>R SY <BLD> PIREX 154_SY <BLD> PIREX SRC
```

```
>
XVM/DOS V1A000 - - - MAC11 is punched

$$JOB PUNCH UNICHANNEL COMPONENTS
$$PAUSE : CHECK THAT THE PUNCH HAS AT LEAST 3/4 BOX OF P'TAPE
^R
^R
PIP
PIP XVM V1A000

>T PP_SY <BLD> M11.8 IMG (I)

>D SY <BLD> M11.8 IMG

>
XVM/DOS V1A000

$$JOB
$$PAUSE : PLEASE LABEL THE NEW PAPER TAPE "MAC11 V1A 8K"
^R
^R
$PAUSE : PLEASE LOAD THE MAC11 PAPER TAPE INTO THE READER
^R
^R
MCLOAD - - MAC11 is installed
MCLOAD XVM V1A000
LOADING MAC11 FOR 12K UNICHANNEL
DONE
XVM/DOS V1A000

$$PAUSE : CHECK THAT THE PUNCH HAS AT LEAST 1/2 BOX OF P'TAPE
^R
^R
UC15 ON

$MAC11 - - PIREX is assembled producing a paper tape binary

MAC11 XVM V1A000

>B_    PIREX 154

END OF PASS 1


ERRORS DETECTED:  0


MAC11 XVM V1A000

>
XVM/DOS V1A000

$$JOB
$$PAUSE : PLEASE LABEL THE NEW PAPER TAPE "PIREX 154"
^R
^R
PIP
PIP XVM V1A000

>D SY <BLD> PIREX 154

>
XVM/DOS V1A000

$$JOB
$$PAUSE : CHECK THAT THE PUNCH HAS AT LEAST 1/2 BOX OF P'TAPE
```

```
^R
^R
UC15 ON

$MAC11  - -  SPOOL is assembled producing a paper tape binary

MAC11 XVM V1A000

>B_   SPOL11 145

END OF PASS 1


ERRORS DETECTED:   0


MAC11 XVM V1A000

>
XVM/DOS V1A000

$$JOB
$$PAUSE : PLEASE LABEL THE NEW PAPER TAPE "SPOL11 145"
^R
^R
$PAUSE : PLEASE LOAD THE SPOL11 PAPER TAPE INTO THE READER
^R
^R
SPLOAD - - -SPOOL is installed
SPLOAD XVM V1A000
DONE
XVM/DOS V1A000

$PIP
PIP XVM V1A000

>D SY <BLD> SPOL11 145

>
XVM/DOS V1A000

$$JOB
$TIME
TIME 043034

$$JOB
$$EXIT  - -  The automatic phase of system construction
              has completed
XVM/DOS V1A000

$
```

2.8 INSTALLATION OF LV11 RASTER SCAN PLOTTER

If your LV Printer-Plotter hardware is attached to the UNICHANNEL Line Printer register address 177514, the LV is run as a line printer. In this case, the user should specify the UNICHANNEL Line Printer and not the LV11 Printer/Plotter in the XVM/DOS BUILD Procedure Question and Answer Session. If your LV Printer/Plotter hardware is attached to the UNICHANNEL LV register address 164ØØØ, the user should specify an LV11 Printer/Plotter to the XVM/DOS BUILD Procedure. These users should follow the instructions below for LV handler installation and optional plot installation. Users specifying the LV11 as a UNICHANNEL Line Printer should ignore this section.

The optional LV RASTER SCAN PACKAGE consists of 9 source modules in addition to this documentation file. These source modules are used in the XVM. The PDP-11 support is already part of the standard PIREX source. The source LVU.Ø3Ø is the XVM/DOS I/O Handler source for the LV. The other 8 sources make up the raster scan package. Sources with an extension SRC are FORTRAN; other extensions signify MACRO sources.

The XVM DOS I/O Handler for the LV is installed like any other I/O Handler, with one exception. The skip IOT for the UNICHANNEL devices is a variable. Typically, the four UNICHANNEL skips are assigned, in order, to RK, CD, LP, and XY. If your system has all four of these devices, go to Appendix C. The LV Handler is going to steal a skip not used by a missing device. The following table shows the correspondence among devices, API levels, and skips.

Device	Level	Skip
RK	Ø	7Ø61Ø1
CD	1	7Ø6121
LP	2	7Ø6141
XY	3	7Ø6161

When assembling LVU.Ø3Ø, specify APILVL=x at assembly time, where x is the level in the table above. If no specification is made, the level defaults to 3. When running SGEN to install the handler, the device skip is the corresponding skip in the table; the handler name is LVA. Also, remember to change the handler binary name to LVA. BIN when PIPing to <IOS> .

If you intend to use the LV as a printer only, you are now done. If you wish to use the plot capability of the LV, continue with the next paragraph.

The RASTER SCAN modules are assembled or compiled, as appropriate. It may be convenient to collect the binaries into a library with UPDATE. RASTER Ø73 should be assembled with the parameter NONXVM=Ø for non-XVM systems, i.e., V3A, V3B. The modules are LOADed or CHAINed with the user's code.

APPENDIX A
NEW FEATURES

This appendix provides information for those who are upgrading from DOS-15 to an XVM/DOS System.

A.1 EXTENDED ADDRESSING

XVM/DOS has 128K Addressing Capability. Common blocks may now reside both above and below the traditional DOS 32K boundary, code must reside below 32K.

A.2 XVM MODE

There is a new operational mode called XVM mode. XVM mode makes use of the new XM15 hardware option to provide 17-bit indirect addressing. When XVM mode is enabled, user programs run in user-mode rather than exec-mode and the following flags are set in XM15 hardware registers: Relocate-disable, 17-bit G-Mode Enable, and IOT Protect Disable. When XVM mode is disabled, user programs run in exec-mode identically as before.

Although user programs run in XVM mode, the XVM/DOS monitor, I/O drivers and most DEC-supplied system programs do not. The only legitimate entries to both the monitor and the I/O drivers are via interrupts and the CAL instruction, both of which take the CPU out of XVM mode. The DBR (debreak-and-restore) instruction returns the CPU to the user programs and restores XVM (user) mode if it was enabled. Previously supported transfers of control via .SCOM locations may still be used. These will result in CAL entries to the resident monitor. DEC-supplied system programs are loaded by the System Loader, which does not initialize XVM mode. Linking Loader and Execute initialize, if requested, XVM mode when loading user programs. User-written system programs loaded by the System Loader must initialize XVM mode themselves if 17-bit indirect addressing is desired.

A.3 CONFIGURATION VERIFICATION

A.3.1 System Parameter Verification

During monitor initialization XVM/DOS verifies that the actual hardware configuration is sufficient for the system parameters specified. If insufficient hardware is available, the initialization routine will alter the parameter settings to reflect the hardware configuration and print an appropriate message on the console printer. The only parameter settings altered will be current settings kept in core; default values kept on disk will remain untouched. This same process will occur whenever a monitor command is used to change the setting of any of these parameters.

A.3.2 Monitor Parameter Informational Messages

A number of informational messages are output, if appropriate, to inform the user of unavailable hardware. These messages include: XVM NOT AVAILABLE, nnnK NOT AVAILABLE -- MEMORY SIZE=mmmK, API NOT AVAILABLE, CLK NOT AVAILABLE. With the exception of the memory size message, these messages are self-explanatory. In the memory size message, the value "nnnK" is the requested memory size and "mmmK" is the actual (physical) memory size. If this message is output, it indicates that the actual (physical) memory size has become the system memory size. If the XM15 hardware option is absent from the system, the user can not specify a "MEMSIZ" greater than the bootstrap location. This message also occurs if an attempt is made to set MEMSIZ to a value lower than the bootstrap location.

These messages are output, if the requested hardware is not available, after any of the following events:

1. A system cold-start.
2. A login or logout.
3. A system parameter change via a monitor command.

If a message is output, the appropriate system parameters are changed to reflect the available hardware. In actuality, the system parameters are continually being verified against the available hardware - this process occurs every time the non-resident monitor is entered or exits. If this verification ever fails, an appropriate message is output, and the corresponding system parameter is adjusted. After this adjustment has been performed once, the verification always succeeds, so no further messages or adjustments occur (until after the next change to the parameters). System parameters may be explicitly changed with a monitor command: Cold-starts, logins, and logouts modify system parameters by restoring system defaults.

A.4 MONITOR DIALOGUE CHANGES

A.4.1 XVM ON/OFF

A command to turn XVM mode on or off has been added. XVM mode requires the presence of the XM15 hardware option. If the required hardware is not present, XVM mode is not enabled; instead, an appropriate informational message is output. Setting this parameter does not effect the "MEMSIZ" parameter. Enabling XVM mode allows the user to use 17-bit indirect addressing. Disabling XVM mode restricts the user to 15-bit indirect addressing.

When XVM mode is disabled, the effective memory size (that which the user can use) is the bootstrap address. When XVM mode is enabled, the user can use the entire logical system memory size (MEMSIZ). Actual allocation is performed by the Linking Loader and CHAIN/EXECUTE.

A.4.2 MEMSIZ numK

A command has been added to set the logical system memory size. The "num" is a decimal number in the range of 24K through 128K in 8K increments. This command attempts to set the system memory size to the entered value. The memory size may not be set lower than the bootstrap address. If "MEMSIZ" is set lower than the bootstrap address, an appropriate error message is displayed (see Chapter 8 of XVM/DOS Users Manual), and "MEMSIZ" is reset to the bootstrap address. If insufficient memory is available, the memory size is set to the actual (physical) memory size on systems with an XM15 option or to the bootstrap location on systems without the XM15 hardware and an appropriate informational message will be output.

A.4.3 UC15 ON/OFF

A command has been provided to enable and disable use of the UC15 peripheral processor. The UC15 peripheral processor may, of course, only be enabled if it is present. If UC15 OFF is entered and the UC15 is not present, it is not enabled and an appropriate informational message is output. The POLLER ON/OFF command has no effect when the UC15 is disabled; its effect is realized, however, if the UC15 is subsequently enabled. The POLLER checks both the "UC15" and "POLLER" switches to determine if it should run. The UC15 on an RK resident system can not be disabled.

A.4.4 X4K Command Deleted

Extra 4K is no longer supported, therefore the X4K command has been deleted.

A.4.5 SCOM, REQUEST and INSTRUCT Command Changes

The SCOM, REQUEST, and INSTRUCT command printouts have been modified to reflect XVM/DOS changes.

A.4.6 MODE Specification Message

In certain appropriate circumstances, the monitor outputs a one-line mode specification message. This mode specification message has the following form:

$$\left\{\begin{array}{l}\text{PAGE MODE}\\ \text{BANK MODE}\end{array}\right\} \text{nnnK} \left[\left\{\begin{array}{l}\text{API ON}\\ \text{API OFF}\end{array}\right\} \left\{\begin{array}{l}\text{XVM ON}\\ \text{XVM OFF}\end{array}\right\} \left\{\begin{array}{l}\text{UC15 ON}\\ \text{UC15 OFF}\end{array}\right\} \left\{\begin{array}{l}\text{POLLER ON}\\ \text{POLLER OFF}\end{array}\right\}\right] \text{UIC}$$

This mode specification message includes all of the applicable specifications following:

nnnK   This specification always appears. The "nnn" is the current system memory size ("MEMSIZ").

BANK   This specification appears whenever the system is in bank mode.

PAGE   This specification appears whenver the system is in page mode.

API ON   This specification appears whenever the API hardware is present and enabled.

API OFF   This specification appears whenever the API hardware is present but disabled.

XVM ON — This specification appears whenever the XM15 hardware option is present and wide addressing mode is enabled.

XVM OFF — This specification appears whenever the XM15 hardware option is present and wide addressing mode is disabled.

UC15 ON — This specification appears whenever the UC15 peripheral processor is present and its use is enabled.

UC15 OFF — This specification appears whenever the UC15 peripheral processor is present and its use is disabled.

POLLER ON — This specification appears whenever the UC15 peripheral processor is present and the POLLER is enabled.

POLLER OFF — This specification appears whenever the UC15 peripheral processor is present and the POLLER is disabled.

UIC — The current UIC.

The mode specification code is output:

1. Immediately after the date is entered during system start-up.
2. After invalid parameter entries.
3. After LOGIN/LOGOUTs.
4. In response to the following new command

MODE

Automatic Mode specification output (1,2,3 above) may be suppressed via the NOMODE assembly parameter to the non-resident monitor or through a "N" reply to the BUILD Procedure's "MODE MESSAGE" question.

A.4.7 LA30 and 33TTY Commands Deleted and TAB ON/OFF and FILL ON/OFF Added

The LA30 and 33TTY commands have been deleted and replaced by:

TAB ON/OFF
FILL ON/OFF

The FILL command enables (ON) and disables (OFF) fill character insertion (useful on LA30-type devices).

The TAB command specifies whether or not the system should simulate the tab character; specifying TAB ON conditions the XVM/DOS monitor to simulate tabs with the appropriate number of spaces. The TAB and FILL commands do not interact with each other in any manner whatsoever.

A.4.8 Monitor Batching

An enhanced BATCH facility allows batching from the following devices:

RP02	(DP any unit, any UIC)
RF15	(DK, any UIC)
RK05	(RK, any unit, any UIC)
System Disk	(SY, any unit, any UIC of the disk type on which DOS resides)
TC15	(DT, any unit)
TC59	(MT, any unit)
PC15	(PR)
CR11 CR03B CR15	(CD)

BATCH uses the standard XVM/DOS handlers found in the <IOS> UIC. The type of card reader in use for BATCH is determined by the card reader handler in <IOS>. The format of the BATCH command to the non-resident monitor is:

$$\$\left\{\begin{array}{l}\text{BATCH}\\ \text{B}\end{array}\right\}\sqcup\text{dv}\left[\left[\text{unit}\right]\sqcup\left[<\text{UIC}>\right]\sqcup\text{filnam}\right]$$

Where:

dv: is the two or three character device name from above.

unit: is optional for all device codes except PR and CD

<UIC>: is optional for DP, RK, and DK and is the UIC in which the BATCH file resides. Note that the BATCH UIC is independent of the logged in UIC and .DAT UIC assignments.

filnam: is a one to six character name of the BATCH file and is required on all devices but PR and CD. The BATCH file has a required extension of 'BAT'.

A.4.9 Error Messages

The following are new error messages and their meanings:

Error Message	Meaning
ILLEGAL UNIT NUMBER	A non-blank character other than 0-7 follows the device name.
BAD BATCH DEV	Either a five or six character device name specified or a three character device name specified with a non-numeric fourth character.
FILENAME MISSING	A filename was not specified for a file-oriented BATCH device.
BAD UIC	An expected '>' was missing in a UIC specification.
BAD DEV	The specified device is not in this system.

A.4.10 Batch Mode Commands

The following are new BATCH Mode Commands:

$JOB This command is optional (new with XVM) and serves to delimit "JOBs". When read by the nonresident monitor, the command is output on the operator's console, and the next command is read. When read by the user program, the job is terminated and the nonresident monitor is recalled.

$END This command is optional and serves to delimit data in BATCH streams. When read by the nonresident monitor, the command is typed and the next command is read. When read by a user program, the EOF header word pair is returned.

$PAUSE — This command is optional and serves to "halt" BATCH processing until the operator intervenes. The command is output on the console. The terminal bell is rung and a ↑R prompt is issued. Then no further action is taken until the operator types ↑R on the console.

$DATA — This command is optional and serves to notify BATCH that user data cards follow which should not be echoed on the console. The command is not typed. The next $ card restores printing.

$EXIT — No matter how this command is read, it is output and the monitor reverts to interactive mode from BATCH Mode.

When XVM/DOS is in BATCH Mode, IOPS errors no longer cause automatic job termination. IOPS errors are handled as in interactive mode.

A.4.11 New Device Mnemonics

Two new device mnemonics are available and can be assigned with 'ASSIGN':

SY: — This device mnemonic is mapped to the 'A' version of the disk handler on which XVM/DOS resides.

CM: — If in BATCH Mode, input is from the BATCH device; if not in BATCH Mode, input is from the teletype. If running BOSS, output goes to the line printer; otherwise, output goes to the teletype.

	Batch	BOSS	Others
Input	Batch Device	TT	TT
Output	TT	LP	TT

A.5 SGEN CHANGES

1. The question "EXTRA 4K" has been deleted, since XVM/DOS does not support extra 4K.

2. The question "↑Q AREA SIZE? (NONE, nnK) [nnn]" is asked for all systems, not just RF systems as before. The range of allowable ↑Q area sizes has been expanded to zero through 128K inclusive in 8K increments. A ↑Q area size of zero is indicated by the response "NONE" or by the response "ØK". ("NONE" is allowed to maintain compatability.)

   This question allows the user to set the ↑Q area size on the system disk. The ↑Q area size should be equal to the amount of memory available in the user's configuration. Users who lack sufficient system disk space to accommodate a large ↑Q area may set the ↑Q area to a size smaller than available memory space. The following table indicates what effect the ↑Q area has on the ↑Q, DUMP, GET and PUT operations.

		Maximum Amount Transferred		
↑Q AREA	↑Q	GET	PUT	DUMP
NONE	NONE	NONE	NONE	NONE
↑QAREA<MEMSIZ	↑QArea	↑QArea*	↑QArea*	↑QArea
↑QAREA=MEMSIZ	MEMSIZ	MEMSIZ	MEMSIZ	MEMSIZ
↑QAREA>MEMSIZ	MEMSIZ	MEMSIZ	MEMSIZ	MEMSIZ

*These will occur only if the Bootstrap is contained within the image.

3. A new question has been added to the section for altering system parameters. This question is:

   "XVM MODE? (Y)"

   If the reply to this question is "Y", the XVM mode is enabled by default. If the reply is "N", XVM mode is disabled by default.

4. A new question has been added to the section for altering system parameters. The question is:

   "SYSTEM MEMORY SIZE nnnK"

   The reply to this question is used as the default for system memory size. Legitimate memory sizes are in the range 24K through 128K inclusive, in 8K increments.

5. SGEN performs the following tests to determine if a skip is legitimate.

   a. Is the skip an IOT? (i.e., must be of the form 70nnnn.)

   b. Is bit 14 zero? (The skip may not clear the accumulator.)

      Errors detected by these tests display the "ILLEGAL IOT" message and the IOT is ignored.

   c. The low order octal digit must be a 1; it must be a skip IOT and not any other type of IOT. This skip check may be overridden by a new feature. The message:

      "NOT A STANDARD SKIP IOT"
      "DO YOU WISH IT ACCEPTED (Y/N)? (N)"

      will be displayed. If the user replies "Y", the skip is accepted; any other reply causes the skip to be ignored.

6. The 33TTY and LA3Ø questions have been deleted and replaced with the following two questions:

   FILL CHARACTERS? (N)
   TAB SIMULATION? (N)

   The answers to these questions are used to set the defaults for the FILL and TAB parameters, respectively. If the fill character question is answered with a yes, fill characters (for LA3Ø type devices), are inserted at appropriate points in console terminal output; if the answer is no, fill characters are not issued. If the TAB question is answered with a yes, the XVM/DOS monitor handler simulates tabs with spaces, if the answer is no, tabing is left to the terminal.

7. SGEN now allows handlerless devices. A handlerless device is a set of skips without any associated handlers. There may be many handlerless devices, though all handlerless skips can be accommodated by one handlerless device entry.

   In Section B of SGEN (device retention question) the handlerless device is printed as "HANDLERLESS DEVICE (Y)". The only options are complete deletion or skip addition/ deletion.

In Section C of SGEN a "new" handlerless device may be added by entering:

a. A two-character dummy device name (not retained) to the "NEW DEVICE" question of SGEN.

b. Then a "carriage return" is entered to the "HANDLER NAME" question.

c. Then the skips are entered in the reply to the "NEW SKIPS" question.

The system is supplied with an existing handlerless device used to provide LT15/LT19 skips to prevent unexpected interrupt from these devices.

A.6 DEVICE DRIVER CHANGES

1. The .TRAN function uses buffer addresses of 17 bits allowing transfer to and from memory regions above 32K.
2. The .TRAN function uses word counts of up to 131,072 words.
3. The .DLETE, .ENTER, .FSTAT, .RAND, .RENAM, and .SEEK functions require that the "NAMPTR" parameter (pointer to a .SIXBT filename) not point to filenames located above 32K. This restriction means that filenames stored in common blocks located above 32K must not be used for arguments in I/O macros before they are copied below 32K. It is the user's responsibility to abide by this restriction.
4. The .INIT function requires that the "RESTRT" (restart address) parameter be restricted to a 15-bit value -- an address within the lower 32K addressing region. It is the user's responsibility to abide by this restriction.
5. The .WAITR function requires that the "WAITAD" parameter (address to which control is returned if I/O is incomplete) be restricted to a 15-bit value -- an address within the lower 32K addressing region. It is the user's responsibility to abide by this restriction.
6. The .READ, RTRAN, and .WRITE functions require that the "BUFFAD" (buffer address) parameter be restricted to a 15-bit value -- an address within the lower 32K addressing region. This restriction has no effect on FORTRAN programs, because the FORTRAN OTS uses an internal buffer (located

below 32K) before transferring information into common blocks during I/O. MACRO users must allocate an I/O buffer below 32K for use when transferring data into common blocks located above 32K; if they do not, a fatal IOPS 67 error occurs.

7. The .READ and .WRITE functions require that the "WDC" (word count) parameter be restricted to a maximum of 32,768 15-bit words. This parameter is used only for DUMP mode transfers, which are not used by the FORTRAN OTS. This restriction has little effect on the user. MACRO coded transfers to areas located above 32K must use intermediate buffers below 32K. Violation of this requirement for DUMP mode transfers will cause a fatal IOPS 67 error. The IOPS 67 error will also be output on any attempt to extend a DUMP mode transfer beyond the 32K limit.

8. The user is responsible for insuring that no portion of an I/O buffer used by .RTRAN, .READ or .WRITE extends beyond the 32K boundary for non-DUMP mode transfers.

   An example of this error will illustrate the point:

   ```
   .READ 1,4,777ØØ,256
   ```

   In this function, both the buffer address 777ØØ (octal) and the word count 256 (decimal) are valid according to 6 and 7 above, but the buffer actually extends above 32K. The results of such an operation are undefined.

9. The IOPS 67 error message definition has been extended to include the violation of placing the start of a buffer above 32K for the .READ, .WRITE, .RTRAN functions and attempting a DUMP mode transfer into the region above 32K.

10. The .CLEAR directive no longer reserves a ↑Q area on RPØ2 type disk packs. The ↑Q area is now reserved by use of SGEN. Users with "old" (non XVM-initialized) disk packs lose 32K of disk area (which they never actually had). Since "system" disk packs must be restored from the XVM-release DOSSAV tapes, there will be no loss of 32K on them. Reinitialization of "old" packs "recover" the missing blocks.

11. The RK driver tests for the UC15 hardware being enabled during any I/O operations. An IOPS 12 (terminal device

error) is issued if UC15 hardware is not enabled. The error message definition has been expanded to include this case.

12. All UNICHANNEL device drivers now detect the absence of UC15 via the SC.MOD bit 16 "UC15 enable flag" and issue an IOPS 12 if UC15 is not enabled.

13. The UC15 Line Printer Handler responds to the "NOFF" assembly parameter by NOT issuing a form feed on .CLOSE (form feed on .INIT will remain enabled).

A.7 GET, PUT, AND ↑Q CHANGES

A.7.1 .GET and GET

The GET command and the .GET MACRO have been altered to test, via QFILE, the image size of the file as stored in the SC.MOD image of that file. Hardware compatability of the image, and the presence of open mass storage files is also checked.

GET (or .GET) are successful only if:

1. The Image MEMSIZ (SC.MSZ) is less than or equal to the current MEMSIZ (SC.MSZ) and the image contains the image's Bootstrap.

2. The image fits into the ↑Q Area and logical memory, as defined by MEMSIZ.

3. No devices are enabled (via the images SC.MOD) that are not present in the current system (SC.NMF).

4. Optionally (via "Y/N" reply) there may be no open files to mass storage in the image.

   The GET command syntax has been changed to the following:

$$\text{GET} \begin{bmatrix}\text{P}\\\text{T}\\\text{S}\end{bmatrix} \begin{bmatrix}\text{␣filnam ext}\end{bmatrix}$$

   The .GET MACRO syntax has been changed to the following:

   .GET␣f, namptr

   The "namptr" now points to a location within the first 32K of memory. The user must abide with this restriction.

   The elimination of the unit number field should cause no incompatabilities for those users with preexisting unit number fields of zero in object code.

A.7.2 PUT and .PUT

The PUT command and .PUT MACRO have been altered to output the lesser of the current system memory size or the ↑Q Area size. This value is determined by the system initialization routine and is stored in .SCOM. This value is also stored in the file (as part of .SCOM in the core image) for use by .GET and GET.

The syntax of the PUT command is changed to:

```
PUT␣filnam␣ext
```

The .PUT MACRO syntax is changed to:

```
.PUT␣f, namptr
```

The "namptr" field should follow the same restrictions as in .GET.

The elimination of the unit number field should cause no incompatabilities for those users with preexisting unit number fields of zero in object code.

A PUT is allowed:

1. If the ↑Q area exists (RESMON check).
2. If the Min (↑Q area, MEMSIZ) contains the bootstrap - (.PUT - Resmon Test) (PUT - Qfile Test).

A.7.3 ↑Q Function

The ↑Q function has been modified to DUMP to the ↑Q area the lesser of the ↑Q area size or the current memory size. This value is determined at system bootstrap and initialization time. Only the system device (unit Ø) is allowed as the DUMPing device, removing the unit number ↑Q<u>n</u> option for disk packs and disk cartridges.

A.8 DUMP CHANGES

The following changes have been made to DUMP.

The ALL command lists the contents of the ↑Q area from location 5 to the highest memory location, where the highest memory location is defined as:

1. ↑Q Area on disk devices: the highest memory location dumped to the ↑Q area as determined by the minimum of SC.MSZ in the image <u>on the ↑Q Area</u>, or the size of the ↑Q area. The image's SC.MSZ must contain a valid MEMSIZ or the ↑Q area size is used.

2. ↑Q area on DECtape is the current memory size as determined by SC.MSZ in core. To DUMP other DECtape ↑Q area sizes, use the ALL (nn) option described in A.7.2.

The "ALL(nn)" command specifies that the DECtape ↑Q area size (nn=1,2,3, ...128) is different than that of the current system memory size. This option is valid only if .DAT-14 is assigned to DECtape. This option allows the flexibility to list QDMP-XVM dumps of XVM/RSX.

The "a - b" command lists the contents of the core locations between absolute addresses a and b (in octal) of the ↑Q areas on the .DAT-14 device where "a" is greater than or equal to 5 and "b" is less than or equal to:

1. ↑Q area on disk: the highest memory location (defined above).
2. ↑Q area on DECtape: 377777_8 (131Ø71 decimal). This option allows the flexibility to list RSX QDMP-XVM tapes.

A.9 DOSSAV CHANGES

The spooler disk area on RK cartridges is no longer saved/restored.

The ↑Q Area is saved/restored only at the user's option. DOSSAV will ask the following question during a save disk operation:

"SAVE ↑Q AREA (Y/N)?"

The user must reply Y↙ or N↙. Any other reply is considered to be equivalent to "Y↙".

This question is asked only if an ↑Q Area is present on the disk being saved. This implies that the question will not be asked for non-system resident disks.

The previous changes decrease the size of the DOSSAV output/input and consequently the time required to perform a DOSSAV.

A.10 PIREX CHANGES

The RK DRIVER has been changed to allow transfers of up to 128K words. The format of the RK task control block has been changed to allow the extra 64K transfer to be requested. (DOS-15 V3B PIREX allowed transfers up to a maximum of 64K.)

The revised TCB format is:

<table>
<tr><td colspan="2">15 8</td><td colspan="2">7 ∅</td><td></td></tr>
<tr><td colspan="2">ATA</td><td colspan="2">ALV</td><td>0</td></tr>
<tr><td colspan="2">FCN</td><td colspan="2">2∅2</td><td>1</td></tr>
<tr><td colspan="4">REV</td><td>2</td></tr>
<tr><td colspan="4">BLOCK NUMBER
15 2 1 0</td><td>3</td></tr>
<tr><td>R
E
L</td><td>unused</td><td>6
4
K</td><td>MSMA</td><td>4</td></tr>
<tr><td colspan="4">LSMA</td><td>5</td></tr>
<tr><td colspan="4">WORD COUNT
15 11 10 9 8 7 0</td><td>6</td></tr>
<tr><td colspan="2">UNUSED</td><td>UNIT</td><td>FUNCTION</td><td>7</td></tr>
<tr><td colspan="4">RKCS</td><td>10</td></tr>
<tr><td colspan="4">RKER</td><td>11</td></tr>
<tr><td colspan="4">RKDS</td><td>12</td></tr>
</table>

Where:

"REL": Word 4, Bit 15. If set (1), request comes from PDP-11 task. If ∅, then from the XVM.

"64K": Word 4, Bit 2. When set causes an additional 64K words to be transferred.

NOTE

A zero in the word count field (word 6) causes a 64K transfer. The 64K bit in word 4 is to be used for an extra 64K transfer.

PIREX now allocates memory to connected tasks via a memory allocation table. This allows PIREX to properly recover contiguous memory segments regardless of order of deallocation.

PIREX now monitors the spooling disk to detect the offline condition and disconnects the SPOOLER if this condition is detected.

PIREX now detects TCB's that are sent from an area beyond the 28K memory window available to the UNICHANNEL and issues an error message via the POLLER mechanism.

A.11 BOOTSTRAP CHANGES

A.11.1 RKBOOT

The RK disk bootstrap has been modified to support transfers of greater than 64K words. This is done in conjunction with the PIREX modifications. The bootstrap is used for the ↑Q DUMP operation. Since ↑Q dumps of greater than 64K are now possible, this modification is necessary.

A.11.2 RPBOOT/RFBOOT

The special UNICHANNEL version of these bootstraps has been eliminated. The stop task's directive is now issued by the resident monitor.

A.12 QDMP XVM

The QDMP utility papertape has been modified to allow the user to specify up to 17 bits of upper address limit in the switch register. This allows panic memory dumps of up to 128K.

A.13 LINKING LOADER CHANGES

The format of the loader map output has been altered. Values representing the relocated addresses of programs, common blocks, and global symbols, are output as 6 octal digits instead of 5.

In XVM mode, the manner in which a set of programs is allocated memory has been altered so that it is noticable to the user only by the load addresses typed out in the loader map. If the system is in XVM mode, the Linking Loader attempts to allocate all uninitialized common blocks above the bootstrap. Common blocks are initialized by block data subprograms in FORTRAN and the .CBS/.CBC/.CBE psuedo-ops in MACRO. Initialized common blocks and all programs are loaded below the bootstrap. Uninitialized common blocks may be allocated below the bootstrap if insufficient memory is available above the bootstrap. In the extreme case of no memory above the bootstrap, this method defaults to the previous allocation strategy. If the system is not in XVM mode, the Linking Loader will use the previous memory allocation strategy; i.e., it will allocate everything below the bootstrap. If the system is in XVM mode, the Linking Loader will initialize XVM mode before starting the user program.

New Features

A new error code, .LOAD 6, has been added. This code indicates that an attempt was made to either allocate a common block with a length of zero or greater than 32K-1 words, or to lengthen a common block beyond the size at which it has already been initialized. A loader map line of the form:

```
C blknam  length
```

is output on the console printer immediately prior to the line containing the error code. In this loader map line "blknam" is the common block and "length" is the attempted length. The program which made the offending common block reference is the most recent program name typed out via the "P" (program load address) loader map option. The loader map line output for this error is unique in that the numeric value output is the length of the associated common block rather than its load address.

A new error code, .LOAD 7, has been added. This code indicates that an attempt was made to define a global symbol with a relocated value of zero or greater than 32K-1. A loader map line of the form:

```
G symnam  value
```

is output on the console printer immediately prior to the line containing the error code. In this loader mapline "symnam" will be global symbol name and "value" will be its attempted value. The program which made the offending definition will be the most recent program name typed out via the "P" (program load address) loader map option. The occurrence of this error implies that the global symbol value lies outside the limits of the program which defined the global symbol.

A new data code, 34_8, has been added. It directs the loader to enter or leave common block initialization mode, depending upon the preceding symbol (codes Ø7 and 1Ø). If the symbol is not blank, the loader will initialize the common block identified by that symbol. If the symbol is blank, the loader will leave common block initialization mode. This code may be immediately preceded by code Ø1, common block size, when entering common block initialization mode. Code 34_8 must always be followed by code Ø2, the common block load address, when entering common block initialization mode, and the program load address when leaving common block initialization mode. If entering common block initialization mode, the common block is then initialized via codes Ø3, Ø4, and Ø5. In common block initialization mode, all relocation is done with respect to the base of the common block. The data word associated with code 34_8 is reserved for future use.

A new error code, .LOAD 8, has been added. This code will indicate that an attempt was made to initialize a location outside the range of the current program or common block.

The option switch "C" has been altered to output the common block name followed by the abbreviation "INIT" when a common block is initialized after it has been loaded.

The option switch "P" has been altered to output the block data name followed by the abbreviation "B DATA", since block data is no longer necessarily loaded at the time of the block data declaration.

A new error warning code, .LOAD 9, has been added. If two handlers for the same device (the first two letters of the filename being identical) are requested, the message

.LOAD 9 xx

where: xx = the first two letters of the duplicated handler name

will be output on the console printer. The loading process will not terminate and every handler requested will be loaded.

The system loader also detects an attempted load of two different handlers for one device. However, the system loader's message appears as:

.SYSLD 9 xx

where: xx = the first two letters of the duplicated handler name. The error will not terminate program loading.

Two new additional option switches have been added:

UL:xxxxxx

where: xxxxxx is the User Library name overrding the default .LIBR5. The User Library must be in the UFD associated with .DAT-5.

SL:yyyyyy

where: YYYYYY is the System Library name overriding the default .LIBR. The System Library must be in the UFD associated with .DAT-1 (either BNK or PAG).

These options must be separated from any preceding options by a comma.

The extension is assumed to be 'BIN'.

There is a new loading sequence. The new loading sequence (XVM/DOS V1A) is as follows:

1. Load Handlers and Scan User Library (if present): the symbol table is scanned for all .DAT references and the associated handlers loaded. One pass is then made through the user library to resolve .GLOBLs in the symbol table. If unresolved .GLOBLs remain after the scan, proceed to step 2 which follows. Otherwise, loading is complete.
2. Possibly Scan User Library Again: if at least one .GLOBL was resolved from the user library, return to step 1. Otherwise, go to step 3.
3. Scan System Library: one pass is made through the system library to resolve unresolved .GLOBLs. If at the end of a scan unresolved .GLOBLs remain, go to step 4. Otherwise, loading is complete.
4. Possibly Scan System Library Again: if at least one .GLOBL was resolved from the system library, control returns to step 3. Otherwise, go to step 5.
5. Scan Again or Give Error Report: if no GLOBLs were resolved by the last scan of either the user or the system library, attempt to match unresolved .GLOBLs against COMMON BLOCK names. If unresolved .GLOBLs still remain, declare .LOAD 3 error. However, if at least one .GLOBL was resolved from one of the libraries, return to step 1.

A.14 DDT CHANGES

Quantities used as addresses are not truncated to 15 bits. Addresses are evaluated as 18-bit quantities and the full 18 bits used to access a memory location, independent of whether or not the system is running in XVM mode. Symbols passed to DDT by the Linking Loader only take on 15-bit values and the $J transfer vector sequencing command produces a 15-bit value. Therefore, an 18-bit address normally occurs only if the user types an 18-bit octal constant as part of an address expression.

New Features

The transfer vector output mode ($V) has been modified so that if the high order three bits of the word being output are zero, the following is output:

address

where "address" is the contents of the low order 15 bits of the word typed in the current address output mode. This is identical to the method of operation with DOS-15 V3B000 and earlier. If the high order three bits of the word being output are non-zero, the following is output:

n00000 + address

where "n" is the octal representation of the high order three bits of the word and "address" is defined above. In DOS-15 V3B000 and earlier the high order 3 bits of the word are always ignored (treated as zero).

The transfer vector retype command (":") will be modified to invoke the new transfer vector output mode in Section A.14.

The address output modes have been modified as follows: The absolute address output mode, $A, outputs 6 octal digits instead of 5. The other address output modes, symbolic ($R) and floating ($F) have been modified to output an offset of up to 6 octal digits when appropriate.

DDT allows the user to modify memory locations located above DDT. A new special symbol, TP#, is provided to indicate the exact location of the top of DDT. DDT will allow only core between P# and C# (inclusive) and above TP# to be modified. The location defined by TP# may be modified. The value of TP# is formally defined to be the address of the first word above the bootstrap, rather than the last word of DDT.

The effective address search (#E) ignores the high order three bits of the address when checking for a match.

Two new output modes are available. These modes are SIXBIT and RAD50.

To select permanent SIXBIT display mode, enter $$S.
To select permanent RAD50 display mode, enter $$R.
To display the contents of Q# in SIXBIT, enter $?.
To display the contents of Q# in RAD50, enter $$?.

A.15 CHAIN AND EXECUTE CHANGES

The format of the load map has been altered. Addresses are output as 6 octal digits instead of 5.

A new option specification has been added to CHAIN to allow the user to specify the bootstrap location for which the overlay system is to be built. The format of this option is:

BOO: nnK

where: nnK is one of the following: 24K or 32K. This option allows the user to specify the bank in which the bootstrap will reside when the overlay system is executing. CHAIN uses this information to avoid overwriting the bootstrap with the overlay system. XVM/DOS always locates the bootstrap at 24K or 32K. If this option is not specified, the default bootstrap location is the current system's bootstrap location.

A new option, XVM, has been added to CHAIN. This option is specified for overlay systems run in XVM mode. Specifying this option causes EXECUTE to verify that XVM mode is enabled (at the time the overlay system is run) and to produce a fatal error if XVM mode is not enabled. This option (or requirement) is automatically assumed whenever the user explicitly states a memory size parameter larger than 32K.

A new option, NOX, has been added to CHAIN. This option is specified when the system on which CHAIN is being run has XVM enabled and the desired overlay system is not to be run in XVM mode.

Additional memory size parameters have been provided. Any parameter of the form:

numK

is accepted as a memory size specification, where "num" is a decimal number between 24 and 128 inclusive. If "num" is not a multiple of 8, a warning message 'QUESTIONABLE MEMORY SIZE' will be displayed, and the value accepted.

If XVM is specified or defaulted to, the current system size ("MEMSIZ") is the default memory size for the overlay system.

If XVM mode is disabled, the current bootstrap location is used for the overlay system's memory size.

The memory allocation strategy for resident code has been altered. This change is only noticeable to the user by the load addresses typed out in the load map. If the overlay system is being generated for a memory size greater than the bootstrap location, CHAIN will attempt to allocate all uninitialized resident common blocks above the bootstrap. Common blocks are initialized by block data subprograms in FORTRAN and the .CBS/.CBC/.CBE pseudo-ops in MACRO. Initialized common blocks, all programs, and all overlaid common blocks are allocated below the bootstrap. Uninitialized resident common blocks may also be allocated below the bootstrap if insufficient memory is available above the bootstrap. In the extreme case of no memory above the bootstrap, this defaults to the current memory allocation strategy. Blank common is a resident common block.

A new unrecoverable error has been added to CHAIN. This error indicates that an attempt has been made to allocate a common block with a length greater than 32K-1 words. The error message is:

```
COMMON BLK TOO BIG -- blknam
```

where "blknam" is the common block name. The program which made the offending common block reference is the most recent program listed in the load map.

A new unrecoverable error has been added to CHAIN. This error indicates that an attempt has been made to define a global symbol with a value greater than 32K-1. The error message is:

```
GLOBAL DEF TOO BIG -- symnam
```

where "symnam" is the global symbol name. The program which made the offending definition will be the most recent program listed in the load map. Occurrence of this error implies that the global symbol value lies outside the limits of the program which defines the global symbol.

The CHAIN error messages have been altered to the following:

```
ILLEGAL LOADER CODE
LABELED COMMON BLK SIZE ERROR -- blknam
ABS PROG
```

```
DUPLICATE GLOBAL DEF -- symnam
COMMON BLK TOO BIG -- blknam
GLOBAL DEF TOO BIG -- symnam
```

Two additional situations produce fatal errors from EXECUTE. These are:

1. The overlay system requires XVM mode, yet, it is not currently enabled.
2. The overlay system was built for a bootstrap location different from the current bootstrap location.

A new option specification has been added to CHAIN. This option specifies a switch which applies to common blocks in much the same manner as the VTC option applies to common blocks. This option is called RES and has two alternative specifications. The option is specified as follows:

```
...,RES,...
```

where the RES switch is applied to all common blocks (including blank common). If, on the other hand, the option is specified as follows:

```
...,RES/comnm1, comnm2, ../....
```

the RES switch is only applied to the named common blocks. Blank common may be specified by the name ".XX". (This is exactly the syntax of the VTC option.) The option specification may be repeated any number of times, and its effects are cumulative (as are the effects of the VTC option). Applying the RES option to a common block affects the common block in the following manner:

1. The common block becomes part of the resident code, if it is not so already.
2. The common block is forced to reside above the bootstrap. If insufficient memory is available above the bootstrap, the error "CORE OVERFLOW" results even if sufficient memory is available below the bootstrap.

NOTE

> 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.

3. The setting of the SAC option switch is ignored for this common block, as it is for all resident common blocks.

The RES switch is only applied to uninitialized common blocks. Common blocks are initialized by block data subprograms in FORTRAN and the .CBS/.CBC/.CBE pseudo-ops in MACRO. The user must also ensure that the first link which references a common block to which the RES option has been applied defines the full length of the common block. Common blocks to which the RES option has been applied defines the full length of the common block. Common blocks to which the RES switch has been applied are allocated memory after the first link which references them is allocated; an attempt by a subsequently allocated link to lengthen such a common block causes an error.

A new entry is added to the load map to describe common blocks to which the RES switch is applied. At the end of the load map, the heading "RESIDENT COMMON BLOCKS" is output for each common block which is global to the overlay structure being built.

A new unrecoverable error has been added to CHAIN. This error indicates that an illegal attempt has been made to initialize a common block. Common blocks are initialized by block data subprograms in FORTRAN and the .CBS/.CBC/.CBE pseudo-ops in MACRO. The error message is:

```
ILLEGAL ATTEMPT TO INITIALIZE COMMON BLK -- blknam
```

where "blknam" is the common block name. The program which made the offending initialization attempt is the most recent program listed in the load map. This error can be caused by the following situations:

1. An attempt to initialize a common block which is part of another link.
2. An attempt to initialize a common block which has been declared with the RES option.

EXECUTE initializes XVM mode when control is passed to the user program if XVM mode is enabled.

The FGD and BGD options of DOS-15 CHAIN are deleted. The FGD:x option has been renamed BOT:nnnnnn. Its purpose is to force bottom up memory allocation starting at location nnnnnn.

New Features

The FASTEX version of EXECUTE introduced in DOS-15 V3BØØØ is now the only supported version of EXECUTE. FASTEX has been renamed EXECUTE.

Two new options, UL: and SL: are provided to allow the user to override the default user and system libraries.

Execute files created by DOS-15 V3BØØØ are correctly processed by EXECUTE.

The new method for loading in XVM/DOS V1A is as follows:

1. Scan User Library (if present): one pass is made through the user library resolving .GLOBLs in the symbol table. If unresolved .GLOBLs remain after the scan, control goes to step 2. Otherwise, loading is complete.
2. Scan User Library Again: if at least one .GLOBL was resolved from the user library, control returns to step 1. Otherwise, go to step 3.
3. Scan System Library: one pass is made through the system library to resolve unresolved .GLOBLs. If at the end of a scan, unresolved .GLOBLs remain, go to step 4. Otherwise, loading is complete.
4. Possibly Scan System Library Again: if at least one .GLOBL was resolved from the system library, control returns to step 3. Otherwise, go to step 5.
5. Scan Again or Give Error Report: if no .GLOBLs were resolved by the last scan of either the user or the system library, declare UNRESOLVED GLOBL(s) error. However, if at least one .GLOBL was resolved from one of the libraries, return to step 1.

A.16 EDITOR CHANGES

The LIST command has been altered to issue a .CLOSE at the end of each operation. In addition, the Editor initializes (.INIT) the line printer handler with the no form-feed on close flag set.

Two new commands have been added to the Editor (PUT filename ext, PCLOSE) and an old command has been deleted (ICLOSE). The new commands are:

New Features

1. PUT filnam ext.

   This allows the user to duplicate the disk output of the Editor onto another file on .DAT-13. If no filename is given, the current input filename specified in the OPEN is assumed. The PUT command is legal only after an OPEN command. The following message can be output by the PUT command.

   ```
   NO FILENAME GIVEN
   FILE:  filname ext IS ASSUMED
   ```

   The user has issued a PUT with no filenam. The filnam ext given is the assumed filename.

   ```
   FILE:  filnam ext IS ALREADY PRESENT ON .DAT-13
   ```

   The filnam ext given in the PUT command already exists. Reissue the PUT command with a new, unique filename.

2. PCLOSE

   This command closes the PUT file. The range of the PUT command is from the current line at the time of the PUT to the line before the current line at the time of the PCLOSE. PCLOSE is simulated by the following commands:

   a. TOP

   b. CLOSE

   c. SCLOSE

   d. EXIT

   All remaining data in the edit file will be placed in the PUT file prior to closing the PUT file.

3. The ICLOSE command has been deimplemented due to the inconsistencies in operation it forces on the Editor. The functions of ICLOSE can be effectively simulated by the PUT, PCLOSE, and GET commands.

A.17 XVM/DOS GRAPHICS CONVERSION

There is one major incompatability between DOS-15 and XVM/DOS graphics, arising from the mechanism of changing existing code in a display file. If code is to be changed, an extra argument is provided at the time of the generation of this code. The previous format of the returned information in this argument is 3 bits of count, followed by 15 bits of address. In XVM/DOS, the 15 bits of address must be extended to 17, so that the count field is lost. The count is derived by examination of the display file. This approach leads to four restrictions upon the user.

1. The REPLOT command can only be used on the code generated by VTPRIM. Foreign code can not be deciphered since the count is to be derived from an examination of the display file.

   NOTE

   A "PLOT (2" or "REPLOT (2" refers to a PLOT or REPLOT call that places parameter instructions in the display file.

2. An examination of a display file during a REPLOT operation cannot distinguish two one-word PLOT (2 Graphics Calls from one two-word PLOT (2 Graphics Call. The solution is to follow a PLOT (2,,,,,CNAME) with a special graphics no-op. If the following call to add code to this file is not a PRAMTR or PLOT (2, Call, the no-op is written over. If a PLOT (2 terminates the display file, the file is another location longer than previously. No immediate recoding is requested of the user but he should beware of possible increase in display file length.

3. In REPLOT (2, where the previous information was parameter instructions from a PLOT (2, the proper spacing already exists in the file. A leading graphics no-op must be placed where a REPLOT (2 is used to replace non-parameter instructions if parameter instructions follow. The REPLOT (2 may fail for a lack of space to place these no-ops.

4. The fourth restriction concerns users who do not have the arbitrary vector hardware option. For these users, the software approximates the arbitrary vector by basic vectors. Previously, if more than seven vectors were placed, the count resided in the display file rather than the count field. In XVM/DOS, the count resides in the display file if two or more vectors are placed.

   NOTE

   There are three kinds of no-ops. Free space in the display file is represented by "2ØØØØØ", end-of-group, "2ØØØØ1", may be reclaimed for space at any time, and beginning-of-group, "2ØØØØ4", may be reclaimed at REPLOT time only if it is the first word of a group.

The 24K DYSET-DYLINK bug has been fixed.

5. A new VWA. handler has been written to replace the previous inadequate version. This new handler is totally compatable with the old specifications.

A.18 FORTRAN CHANGES

The FORTRAN system has been modified to allow compiled programs and the OTS to run in XVM mode in the XVM/DOS environment. There are two types of changes:

1. Any uses of bits 0-2 in transfer vectors generated by the compiler or present in the OTS have been removed.
2. Data in common blocks can now be accessed via an 18-bit transfer vector, rather than a 15-bit one.

Neither of these changes have any FORTRAN interface implications. A MACRO-15 user, however, who calls the FORTRAN OTS, must be aware of the fact that bits 0-2 of some arguments passed to OTS routines no longer contain mode information, but are simply high order bits of addresses.

All previous releases of the FORTRAN OTS have translated the first character of all output records to an internal carriage control code. In DOS-15 V2A and earlier systems, on input, these control codes were stripped before the FORMAT conversions began. In DOS-15 V3AØØØ, this action was modified to convert leading line feed to a space, but drop all other control codes. This method created incompatabilities between DOS V2A and V3A as well as some serious inconsistencies in DOS V3A. Table A-1 illustrates the difference between the various versions of FORTRAN carriage control.

In XVM/DOS, the OTS is modified to provide carriage control characters translation on output to TTY, LP, XY, and VP ('printer type' devices) only. On input, leading control characters are unconditionally stripped. Therefore, for all devices capable of both writing and reading (PT, DT, MT, RK, DP, and RF) identical FORMAT statements may be used for both writing and reading.

New Features

Due to the increase in performance, the OTS system is distributed assembled for double buffering. If this feature is not desired by the user, he may reassemble the affected OTS modules to omit this feature.

The compiler has been modified to print a heading at the top of each page of output of the compilation listing and to perform pagination every 56 lines of output. The heading is of the form:

```
          PAGE xxx     FILNAM SRC     DD-MON-YY     HH:MM
                 (cont.)     compiler hello message
```

so that the hello message is printed following the time on the same line.

The compiler has been modified to output one of two messages at the end of compilation.

```
                    SIZE = nnnnn NO ERRORS
                            (or)
                        xxxx ERRORS
```

depending on whether errors were detected.

A.19 MACRO CHANGES

A.19.1 MACRO Definition File

The MACRO definition table, previously a part of MACRO-15 proper, has been removed and made into a separate file. This is done to provide space inside the assembler for bug fixes and new features. It also has the desirable side effect of enabling easy expansion of system MACROs.

The file, named MACDEF MAC, resides in the system UIC's PAG and BNK. It is read from the system disk via .DAT-1, which must be assigned to the system disk (default). If the file is not present, the message "MACDEF MAC NOT PRESENT ON .DAT-1" is issued and control returns to the monitor.

The assembler now works as documented; users are not allowed to duplicate user or system MACRO names. This has always been true of user MACROs, but system MACRO names used in the label field were not flagged as errors.

The file, in IOPS ASCII, contains MACRO definitions for all the system MACROs currently supported by XVM/DOS. It may be easily modified by the Editor.

A.19.2 New Pseudo-Ops

A.19.2.1 A new pseudo-op has been added:

```
.CBS name [, size]
```

where: name is any valid name (maximum length of 6 characters);
(optional) size is any absolute expression.

This pseudo-op places the assembler in common block initialization mode, enabling selected portions of the identified common block to be initialized. The name parameter identifies the common block. The size parameter specifies the minimum size of the common block.

Unlike .CBD or .CBDR, .CBS does not generate a transfer vector; thus, a symbol in the label field is meaningless and will cause the line to be ignored, as well as being flagged with the L error indicator. Between a .CBS and the following .CBE, only the .CBC pseudo-op, comments, conditionals, and MACRO calls may occur. The MACRO calls must generate statements of the aforementioned forms. The occurrence of any code-generating statement or any other pseudo-op is an error.

Example:

```
        .CBS ABCD,6
```

indicates that locations within the common block ABCD, which has a minimum size of 6, will be initialized in statements following.

A.19.2.2 A new pseudo-op:

```
        .CBC addr, value
```

where: addr and value are any absolute expression has been added.

This pseudo-op is allowed only when the assembler is in common block initialization mode (i.e., it is only allowed between a .CBS and the following .CBE). The location at relative offset addr in the common block identified by the preceding .CBS pseudo-op is initialized to value. A symbol in the label field is meaningless: the line will be ignored and flagged with the L error indicator.

Example:

```
        .CBC 2,4
```

initializes the second element of the common block named in the preceding .CBS statement to the value 4.

A new pseudo-op, .CBE, is allowed only when the assembler is in common block initialization mode (i.e., following a .CBS). The assembler is taken out of common block initialization mode and restored to normal operation.

NOTE

1. A common block initialization section (consisting of a .CBS, one or more .CBC's and a .CBE) may occur anywhere within a program module. The same common block may be initialized any number of times by any number of modules.

2. All the restrictions which affect block data programs and common block allocation apply to this method of common block initialization. In an overlay system, it will usually prove desirable to place initialized common blocks in the resident section.

CBDR, previously available only in RSX MACRO, is now implemented in this assembler.

Additional system MACROs have been added to the MACRO definition file. Two of these provide MACRO expansion for functional capabilities which already exist. These are the .GET and .PUT MACROs, which were neither recognized nor expanded by MACRO-15. The format of these MACROs has been revised to conform with the new format for .GET and .PUT.

.SYSID is provided to aid system version identification. It performs two functions:

1. It defines several parameters, which might be used for conditional assembly control.
2. It provides a convenient method for inserting a text string (identifying the system version) into an assembly statement.

In providing the first capability, the MACRO defines the following parameters:

```
%RELES = 1  /indicates release A (in V1A)
%XVM =   1  /indicates XVM hardware
%VERSN = 1  /indicates version 1 (in V1A)
```

These parameters are not altered (by this MACRO) if they are already defined. These parameters cannot be defined by this MACRO until after this MACRO is invoked. These parameters are defined only the first time the MACRO is invoked; they will not be redefined if the MACRO is invoked many times nor will they be redefined during pass 2. The definitions of these parameters are not included in the assembly listing.

Besides possibly defining the above parameters, this MACRO is used to insert a system identification string into an assembler statement. The format of the MACRO call and its expansion are as follows:

New Features

Format:

```
→|.SYSID frnt, back
```

Expansion:

```
frnt@XVM VIA@back
```

The use of this MACRO is best illustrated by the following examples:

1. Generate CUSP hello message.

   MACRO Call:

   ```
   .SYSID<→|.ASCII␣/MACRO␣>,<ØØØ/<Ø15>>
   ```

   MACRO Expansion:

   ```
   .ASCII   /MACRO␣XVM␣V1AØØØ/<Ø15>
   ```

2. Generate .TITLE statement.

   MACRO Call:

   ```
   .SYSID<→|.TITLE␣XVM/DOS␣>,<ØØØ␣MACRO␣ASSEMBLER>
   ```

   MACRO Expansion:

   ```
   .TITLE␣XVM/DOS␣XVM V1AØØØ MACRO ASSEMBLER
   ```

3. Define parameters (only).

   MACRO Call:

   ```
   SYSID</DEFINE␣PARAMETERS␣FOR␣MACRO␣>
   ```

   MACRO Expansion:

   ```
   /DEFINE PARAMETERS FOR MACRO XVM V1A
   ```

The .SYSID MACRO will be defined as follows:

```
.DEFIN  .SYSID  FRNT,BACK
```

```
     .IFUND %XVM
%XVM = 1
     .ENDC
     .IFUND %VERSN
%VERSN = 1
     .ENDC
     .IFUND %RELES
%RELES = 1
     .ENDC
     .DEFIN .SYSID FR,BK
FR@XVM V1A@BK
     .ENDM
     .SYSID<FRNT>,<BACK>
     .ENDM
```

The others are:

.MTRAN - supplies access to the monitor's .TRAN routine; and

.XVMON, .XVMOFF - Turn XVM mode on and off.

In the case of a symbol table overflow, an attempt is made to recover memory in the MACRO definition table. The structure of that table is such that all entries have 2 or 3 control words.

1. MACRO definitions have 3 control words:

   Word 1 - positive, non-zero is the size of an active entry;
   - negative is the twos complement size of an inactive entry;
   - Ø flags a partial entry (i.e., one currently being processed).

   Word 2 and 3 - the MACRO names in Radix 5Ø.

2. Real argument lists for MACRO expansions have 2 control words:

   Word 1 - same as word 1 above.
   Word 2 - Ø

3. Local symbol table has 2 words per total entry (note that this is never left after the LOCAL switch has been turned off):

   Word 1 - same as word 1 above.
   Word 2 - Ø

   If the collect is unsuccessful, the TABLE OVERFLOW message is given.

A.19.3 New Error Messages

Y - A .CBS was given with no closing .CBE; a .DEFIN has no corresponding .ENDM and/or a .IF conditional has no closing .ENDC. This error is output on the .END statement at the end of PASS 1.

C - A .ENDC or a .CBE is given with no opening conditional. The error appears on the offending line.

Space allocated for MACROs which have been redefined is reclaimed.

A.20 BOSS CHANGE

BOSS has become a standard component of the XVM system and the "BOSS" command has been added to the standard repetoire of XVM/DOS commands.

The following BOSS commands have been changed or added:

A.20.1 $FOR Card

FUNCTION: The $FOR card calls the FORTRAN IV (F4M) compiler and the Line Editor.

FORMAT:

```
$FOR; options  filename-1:device-1(UIC-1)
              ;filename-2:device-2(UIC-2)
                         ;:device-3(UIC-3)
                         ;:device-4(UIC-4)
```

Line Edit Card/s, or
Source Card/s, or
BCL Card

ARGUMENTS: Options - (Default-B,L) A six-character option field. The FORTRAN options follow:

B = Binary output
L = Source listing
O = Object listing
S = Symbol map

filename-1 = (Default-FILTMP) This ten-character field specifies the SOURCE filename. The source extension is assumed to be SRC. Device-1 (UIC-1) - Standard format.

filename-2 = (Default-filename-1 or FILTMP) A ten-character field used to specify the name of the original source file to be edited. The name is required to edit an existing file, create a new "edited file and compile the newly created file. Device-2 (UIC-2) - Standard format.

device-3 = (Default - Line Printer) This field is used to specify the desired listing output device.

UIC-3 = (Default-current UIC) This field is used if device-3 specifies the disk as the listing output device.

device-4 = (Default-system disk) This field is used to specify the desired binary output device.

UIC-4 = (Default-current UIC) This field is used if device-4 specifies disk as the output device.

A.20.2 New $RUN Card

FUNCTION: The $RUN command takes a file that is located on the disk, does a FORTRAN compile of it, and begins to execute the program. This is an efficient combination of the $FOR, $MAP, and $XCT commands.

FORMAT: $RUN Filename-1;Device-1(UIC-1)

ARGUMENTS: Filename-1 = This is a FORTRAN source coded program located on Device-1.

Device-1 = (Default-System Device) This field is used to specify the desired source file input device.

UIC-1 = (Default-current UIC) This field is used if device-3 specifies disk as the input device.

Once a program has been compiled and runs satisfactorily, the only other command needed to rerun the same program is $XCT.

```
     First Run             Next Run

   $JOB TEST;BLR         $JOB TEST;BLR
   $RUN TEST             $XCT TEST
      DATA                  DATA          /This is much faster
                                           for the second run
   $END                  $END
```

The system device is used for all intermediate files. The resulting binary and XCT/XCU files are resident there.

A.20.3 New $MAC11 Card

FUNCTION: The $MAC11 card calls the PDP-11 MAC11 assembler and Line Editor.

FORMAT:

```
$MAC11; options filename-1:device-1(UIC-1)
      ; filename-2:device-2(UIC-2)
      ;:device-3
      Line-Edit cards
      Source Cards or
      Next BCL command
```

ARGUMENTS: Options - (Default - B,L) A six-character option. The options with their associated actions are given below:

B = Generate binary paper tape

L = Generate a listing on the requested output device.

filename-1 = Source file (Edit output/new filename)

device-2(UIC-2) = Standard format

filename-2 = Edit input (Old filename)

device-2(UIC-2) = Listing device-must not be file oriented.

The $MAC11 command provides the user with the symbolic programming capabilities of the MAC11 assembler which includes a MACRO instruction generator.

New Features

If the Line Editor is used, the following rules must be followed:

1. DECdisk, disk pack, or disk cartridge must be used for input/output, or the following error message is printed:

   'NON-DISK I/O DEVICE'

   and an exit to the next control card occurs.

2. The same device/unit must be assigned to device-1 and device-2. If not, the following message is printed and an exit to the next control card occurs.

   'DIFFERENT DEVICES ASSIGNED TC .DAT-14 & -15'

3. The $MAC11 command can be used to:

   a. Assemble a file already existing. The $MAC11 card must specify filename-1, and the next control card should follow the $MAC11 command.

   b. Create a new file and assemble it. The $MAC11 card must specify filename-1 and the "source" deck should follow this command.

   c. Assemble without creating a file. No filename should be specified and the "source" deck should follow the $MACII command.

   d. Edit an already existing file and assemble it. The filename-1 must be specified, and line-edit commands should follow the $MAC11 command.

   e. Edit an already existing file, create a new "edited" file and assemble the newly created file. The original source filename is specified in filename-2. The new source filename is specified in filename-1, and the $MAC11 commands should be followed by the Line-Edit commands.

   Depending on which function the $MAC11 command specifies, the command can be followed by:

   A MAC11 source deck,
   Line-Edit commands,
   The next Control Card.

New Features

A.20.3.1 New $XVM card - this card should be used only on systems with the XM15 hardware option.

FUNCTION: The $XVM card enables and disables XVM mode.

FORMAT: $XVM ON
OFF

ARGUMENTS: ON = Enables XVM mode for user programs.
OFF = Disables XVM mode for user program.

A.20.3.2 BOSS does not time-out a JOB at the expiration of its allotted time - even if an "IOPS 4" recoverable error is pending. This was implemented to allow pending JOBS to be given an opportunity to execute during an unattended period. The $MNT and $MSGW commands do <u>not</u> allow the time-out to occur, and function as previously specified.

↑C will now abort BOSS execution and return control to the XVM/DOS monitor.

A.21 ADDITIONS TO XVM/DOS MONITOR TCB AREA

A new TCB for the 10/15 Communication project has been added to the current monitor TCB area. The new TCB is 44 (decimal) words in length. This is conditionally assembled with the use of the "DLTCB" assembly parameter, or may be selected by a "Y" reply to the BUILD procedure "10/XVM Communication" question.

A new TCB has been added for the LV Graphics project. The new TCB is 80 (decimal) words in length. This is conditionally assembled with the use of the LVTCB assembly parameter, or, may be selected by a "Y" reply to the BUILD procedure "LV11 Plotter" question.

A.22 DTCOPY/LPA.09/LP.647/89TRAN/DOSBCD/LTX

DTCOPY has been decommitted since PIP XVM can perform all of DTCOPY's functions.

The following handlers are no longer supported (sources will be available through DECUS):

LPA.09 (PDP-9 not supported by XVM software)
LP.647 (PDP-9 not supported by XVM software)

The following are decommitted:

89TRAN	(PDP-9 not supported by XVM software)
DOSBCD	(not required by new monitor)
LTX	(not required by new monitor)

A.23 PIP CHANGES

A new switch, V, has been added to PIP. The V switch, standing for Vertical Forms control, is used on transfer commands to translate the first character of each record to a special character, which, when interpreted by the line printer handler, will produce forms control functions. The translation occurs according to the FORTRAN conventions below:

Character Found	Translated To	Meaning to LP Handler
'1'	FF, 14_8	Skip to top of form
'+'	DLE, 20_8	Overprint
'0'	DC1, 21_8	Double Space
' ' (space)	LF, 12_8	Single Space
anything else	LF, 12_8	Single Space

PIP now recognizes two new pseudo device mnemonics:

SY: signifies a system resident disk. The unit number and UIC are optional.

CM: signifies that the input should come from either the BATCH file or the console, depending on whether the system is in BATCH Mode or not.

The UPDATE command has been modified as follows.

1. If the block number given in the U command is occupied, the standard warning message is given,

"WARNING*BLOCK OCCUPIED: n"

where: n is the block number.

However, the block is no longer put into the BAT nor counted at this point. The user should delete the file containing that block and redo the UPDATE command.

2. If the block number given is already in the BAT, the message:

"BLOCK ALREADY IN BAT: n"

where: n is the block number

is output and no further action is taken.

A.24 MAC11 CHANGES

1. The hello is changed to "MAC11 XVM" from MACRO-11.
2. The extraneous "A" errors have been eliminated.
3. The "FREE CORE" message is no longer printed.
4. The system program "MCLOAD" has been added to ease installation of MAC11 papertapes. To install MAC11 simply:
   a. Install "MAC11" via SGEN
   b. Patch MACINT into "MAC11"
   c. Load papertape of "MAC11" (8K or 12K)
   d. Type "MCLOAD"

A.25 .SCOM TABLE

The existing .SCOM locations described below have been revised as indicated:

Table A-1

System Communication (.SCOM) Table

Location	Mnemonic	Description
100	SC.COD	First free register below bootstrap; the highest location useable for code.
101	SC.RMS	Resident Monitor size; first free register above Resident Monitor.
102 103	SC.FRL SC.FRL+1	Free memory in low core (below the bootstrap). SC.FRL contains the address of the first free location. SC.FRL+1 contains the address of the last free location.
104	SC.MOD	Operating mode bit register:
	SC.API = 400000_8	1 → API enabled
	SC.TAB = 100000_8	1 → Simulate tabs with spaces
	SC.NRM = 40000_8	1 → Non-resident monitor in core
	SC.UB2 = 20000_8	Reserved for customer use.
	SC.UB1 = 1000_8	Reserved for customer use.
	SC.9CH = 4000_8	1 → 9-channel magtapes assumed.
	SC.FIL = 200_8	1 → Insert fill characters.
	SC.BNK = 100_8	1 → Bank mode operation.
	SC.LPSZ = 60_8	Line printer line length (in characters): 0 → zero 40 → 120 20 → 80 60 → 132
	SC.PLR = 4_8	1 → Poller enabled.
	SC.UC15 = 2_8	1 → UC15 enabled.
	SC.XVM = 1_8	1 → XVM mode enabled.
105	SC.SST	Core image system program starting address.
106	SC.UST	User program starting address/↑S address.
	SC.DDT = 400000_8	1 → DDT in core.
	SC.GLD = 200000_8	1 → Linking Loader invoked via GLOAD.
	SC.DNS = 100000_8	1 → DDT invoked without symbol table (via DDTNS).
	Bits 3-17 contain the user program starting address, which is also the address of the ↑S control character routine.	

Table A-1 (Cont)

System Communication (.SCOM) Table

Location	Mnemonic	Description
107	SC.FNM	Execute Filename, .GET,.PUT Filename, or linking loader handler indices.
110	SC.FNM+1	
111	SC.FNM+2	
112	SC.LV4	API level 4 software interrupt transfer vector.
113	SC.LV5	API level 5 software interrupt transfer vector.
114	SC.LV6	API level 6 software interrupt transfer vector.
115	SC.LV7	API level 7 software interrupt transfer vector.
116	SC.TTP	Saved PC on control character interrupts.
117	SC.TTA	Reserved.
120	SC.MSZ	System memory size, as set via MEMSIZ command.
121	SC.MTS	Magtape status register.
122	SC.AMS	Actual (physical) memory size.
123	SC.DAT	Address of .DAT table.
124	SC.SLT	Number of positive .DAT slots.
125	SC.UFD	Address of .UFD table.
126	SC.BNM	Number of file buffers.
127	SC.BLN	Number of words per file buffer.
130	SC.BTB	Address of file buffer transfer vector table.
131	SC.OTB	Address of overlay table or zero.
132	SC.BBN	Bad block number for IOPS20 and IOPS21.

Table A-1 (Cont)

System Communication (.SCOM) Table

Location	Mnemonic	Description
133	SC.VTF	VT ON flag register;
	SC.HFN = 400000_8	1 → Half size VT15 buffer (HALF ON).
	SC.VTN = 100000_8	1 → VT ON issued.
	SC.DMN = 1_8	1 → Display mode on.
134	SC.ETS	Elapsed time in seconds.
135	SC.CTT	Instruction to clear teletype busy switch.
136	SC.ACT	Number of active .DAT slots; number of entries in mass storage busy table.
137	SC.EEP	Expanded error processor entry point.
140	SC.EEP+1	JMP to expanded error processor.
141	SC.UIC	Current UIC
142	SC.NMF	Non-resident monitor flag register;
	SC.MIC = 400000_8	1 → MICLOG successful
	SC.NRE = 200000_8	1 → Non-resident monitor .EXIT
	SC.NRO = 100000_8	1 → Non-resident monitor .OVRLA
	SC.LPON = 40000_8	1 → LP ON 0 → LP OFF
	SC.DMP = 20000_8	1 → ↑Q dump on IOPS errors.
	SC.HLT = 10000_8	1 → Halt on IOPS errors.
	SC.TMM = 4000_8	1 → Mode message should be typed.
	SC.PVT = 1000_8	1 → VT15 present.
	SC.PCLK = 400_8	1 → Real time clock present.
	SC.PAPI = 200_8	1 → API present.
	SC.PUC15 = 100_8	1 → UC15 present.
	SC.PXVM = 40_8	1 → XVM present.
	SC.DT6 = 10_8	1 → Set up .DAT+6 (for BOSS).
	SC.DT7 = 4_8	1 → Set up .DAT-7 (for BOSS or Batch)
	SC.KPN = 2_8	1 → KEEP ON 0 → KEEP OFF
	SC.BCH = 1_8	1 → Batch or BOSS mode active.

Table A-1 (Cont)

System Communication (.SCOM) Table

Location	Mnemonic	Description
143	SC.SPN	.SIXBT name of system program to to loaded.
144	SC.SPN+1	
145	SC.NMN	.SIXBT name of the non-resident monitor ("DOS15").
146	SC.NMN+1	
147	SC.DAY	Today's date, formatted as MMDDYY.
150	SC.TIM	The current time, formatted as HHMMSS.
151	SC.ETT	Elapsed time in ticks.
152	SC.BOS	BOSS bit register:
	SC.BMD = 400000_8	1 → BOSS mode active.
	SC.BCR = 200000_8	1 → Control card read by user.
	SC.BEOF = 100000_8	1 → EOF reached on run time file.
	SC.BTM = 40000_8	1 → Time estimate exceeded.
	SC.BTT = 20000_8	1 → I/O CAL to go to TTA.
	SC.BIO = 10000_8	1 → Terminal IOPS error.
	SC.BDMP = 4000_8	1 → Give user ↑Q dump on IOPS errors.
	SC.BOA = 2000_8	1 → Operator abort (↑T)
	SC.BJA = 1000_8	1 → Job active.
	SC.BXT = 400_8	1 → Exit from BOSS mode.
	SC.BPT = 200_8	1 → User tried to do a .PUT
	SC.BGT = 100_8	1 → User tried to do a .GET
	SC.BERR = 16_8	.SYSLD error number.
	SC.BAB = 1_8	1 → Job abort.
153	SC.VTR	VT ON display file restart address or zero if display file not set up.
154	SC.PRC	Default file protection code.
155	SC.TRN	Reserved.
156	SC.TLM	Two's complement of time limit in seconds or zero if no time limit.
157	SC.SDV	Handler index of system device for Linking Loader.

Table A-1 (Cont)

System Communication (.SCOM) Table

Location	Mnemonic	Description
160	SC.TMT	Two's complement of number of clock ticks until .TIMER interrupt.
161	SC.TMA	.TIMER interrupt routine entry point address.
162	SC.BTA	Address of mass storage busy table.
163	SC.BTL	Number of words per mass storage busy table entry.
164		Reserved
165	SC.CQF	.GET/.PUT flag register (for communication with QFILE):
	SC.QFLG = 400000_8	1 → Call QFILE 0 → Return from .PUT
	SC.QNF = 2000_8	1 → Skip file transfer operation.
	SC.QNRM = 1000_8	1 → Exit to non-resident monitor when QFILE completes.
	SC.QPUT = 400_8	1 → Transfer ↑Q area to file 0 → Transfer file to ↑Q area
	SC.QRTN = 7_8	.PUT/.GET return code.
166	SC.CQB	.MTRAN parameter block for ↑Q area contains:
167	SC.CQB+1	First block of ↑Q area First address -1 in core Two's complement transfer length
170	SC.CQB+2	The transfer length is the minimum of the ↑Q area size and the current system memory size.
171	SC.TDT	Tomorrow's date, formatted MMDDYY.
172		Reserved.
173	SC.TMR	Two's complement of the number of ticks left in this second.
174	SC.LFR	Two's complement of the number of ticks per second.
175	SC.RTF	Indicates the current position within the batch stream (batch mode) or the run time file (BOSS mode).

Table A-1 (Cont)

System Communication (.SCOM) Table

Location	Mnemonic	Description
176	SC.FRH	Free memory in high core (above the bootstrap). SC.FRH contains the address of the first free location. SC.FRH+1 contains the address of the last free location.
177	SC.FRH+1	
200	SC.TCB	Address of Task Control Block transfer vector table.
201	SC.U01	Reserved for customer use.
202	SC.U02	Reserved for customer use.
203	SC.U03	Reserved for customer use.
204	SC.U04	Reserved for customer use.
205	SC.U05	Reserved for customer use.
206	SC.BFNM	.SIXBT batch stream filename or zero if batch device is non-file oriented.
207	SC.BFNM+1	
210	SC.BFXT	.SIXBT batch stream file extension.
211	SC.BUIC	.SIXBT batch stream file UIC code.
212	SC.BDEV	.SIXBT batch stream device mnemonic.
213	SC.BUNT	Batch stream device unit number (high 3 bits).

APPENDIX B
SPRs RESOLVED SINCE DOS-15 V3BØØØ

B.1 BATCH

The CR15 is now a legal batch device. Other new batch devices exist also. See Appendix A concerning new XVM features.

B.2 BOSS

B.2.1 A $RUN command is now available which compiles a FORTRAN program and executes it. See Appendix A concerning new XVM features.

B.2.2 It is now possible to assign the FORTRAN binary output file.

B.2.3 BOSS wastes less paper. The typing of extraneous characters on the console has been corrected.

B.3 CARD READER HANDLER

The size constraint on the table sizes has been removed to make it possible to add new card codes to the handler.

B.4 DOSSAV

It is now possible to restore to the RF disk a system which has a BAT (bad block allocation table).

B.5 DECTAPE HANDLER (DTA.)

DTA. now types out its name and the unit number as part of the IOPS error messages.

B.6 EDITOR

B.6.1 The Editor no longer accepts more than six characters for filenames or more than three characters for extensions.

B.6.2 The Editor now checks for the existence of the original files during CALL RENAME and DELETE processing.

B.6.3 The Editor now checks for the existence of the new filename during CALL RENAME processing.

B.7 FORTRAN

B.7.1 The handling of carriage control characters has been modified. See Appendix A concerning new XVM features.

B.7.2 The "END= " and "ERR= " options now function with papertape.

B.7.3 The use of random access files on units other than zero now function properly.

B.8 MACRO

B.8.1 Character strings in macro definitions are no longer truncated at six characters.

B.8.2 A problem which caused multiple symbol definition errors when using .ETC in a .DEFIN is corrected.

B.8.3 An interaction problem between .DEFIN and .GLOBL, which caused .GLOBLs to not be defined, has been corrected.

B.8.4 MACRO now correctly expands macros with angle brackets.

B.9 MAGTAPE HANDLERS

B.9.1 MTA. and MTF. have been modified to put the magtape status register in .SCOM+21.

B.9.2 MTF. now ignores .SEEK and .ENTER instead of giving an IOPS6 error.

B.10 MONITOR

B.10.1 When VT is ON, turning API ON and OFF repeatedly no longer causes a system crash.

B.10.2 "TIME" and "DATE" now interact via a calendar routine to increment the "DATE" at midnight.

B.10.3 A "TIME" of ØØØØ is now legal.

B.10.4 Various monitor related disk corruption problems have been eliminated under API.

B.11 PIP

B.11.1 Two extra words are no longer appended to each record when using the T switch.

B.11.2 The C switch, used to convert tabs to space, now functions as specified in the manual.

B.11.3 The "H" mode copy command may now be used to copy to the system disk.

B.12 SGEN

B.12.1 It is now possible to enter illegal skips.

B.12.2 Handerless devices are now allowed. See Appendix A concerning new XVM features.

B.13 VECTOR

Two incorrect variable names have been fixed.

B.14 VP15A HANDLER (VPA.)

Interrupts are now enabled properly.

B.15 VWA. HANDLER

The VWA. handler has been rewritten to conform with the existing documentation.

APPENDIX C
THE FIVE DEVICE PROBLEM

When your system contains RK, CD, LP, and XY, as UNICHANNEL devices, an API priority level must be "shared" with the LV. You MUST run with API to accomplish this. All five devices have separate API traps. For example, if we are to share with the LP, edit the LVU source, changing the line:

```
     LSSF=APILVL*20+706101
```

to

```
     LSSF=703421     OR ANY OTHER UNUSED SKIP
```

Assemble with APILVL=2. SGEN, giving LVSF=703421. Note, 703421 is a skip for the second VT15, not supported under DOS. Other non-standard devices could possibly be using this skip. If so, choose any unused skip. This procedure is a solution to what is known as "the five device problem". There is nothing magic about the LV; the same procedure could be applied to some other added fifth device. Further, it is possible to "pair-up" more than one level for a sixth device (find another skip).

NOTE

> The device with the bogus skip cannot be used for the MAC11 listing device, because MAC11 forces API off.

APPENDIX D

INTERNAL DOCUMENTATION FOR XVM/DOS BUILD PROGRAM

D.1 INTRODUCTION

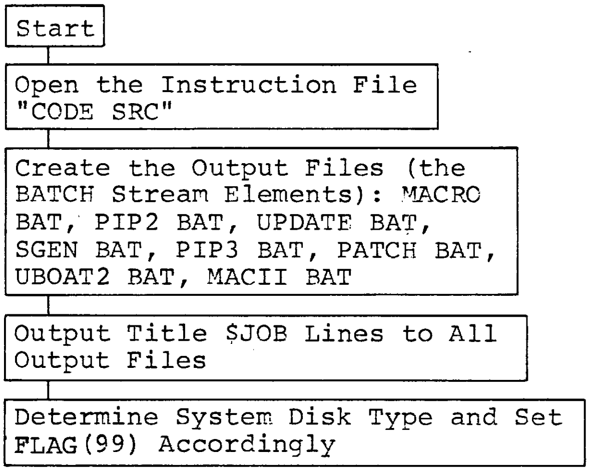
The "BUILD" program is a FORTRAN routine that is used to ask questions, and, based upon the reply, produce a system construction BATCH stream (a "stream" is a series of BATCH files that batch from one to another).

D.2 OVERVIEW

In more general terms the "BUILD" program is an interpreter that inputs instructions (the "CODE" file), and, based upon these instructions, it: outputs text to a file, types a message to the console, asks a question, tests a condition, etc.

The following information is specific to the internal operation of XVM/DOS system construction and is included in this document only for illustrative purposes. The typical user need not examine this material.

The gross flow of control of the BUILD program is:



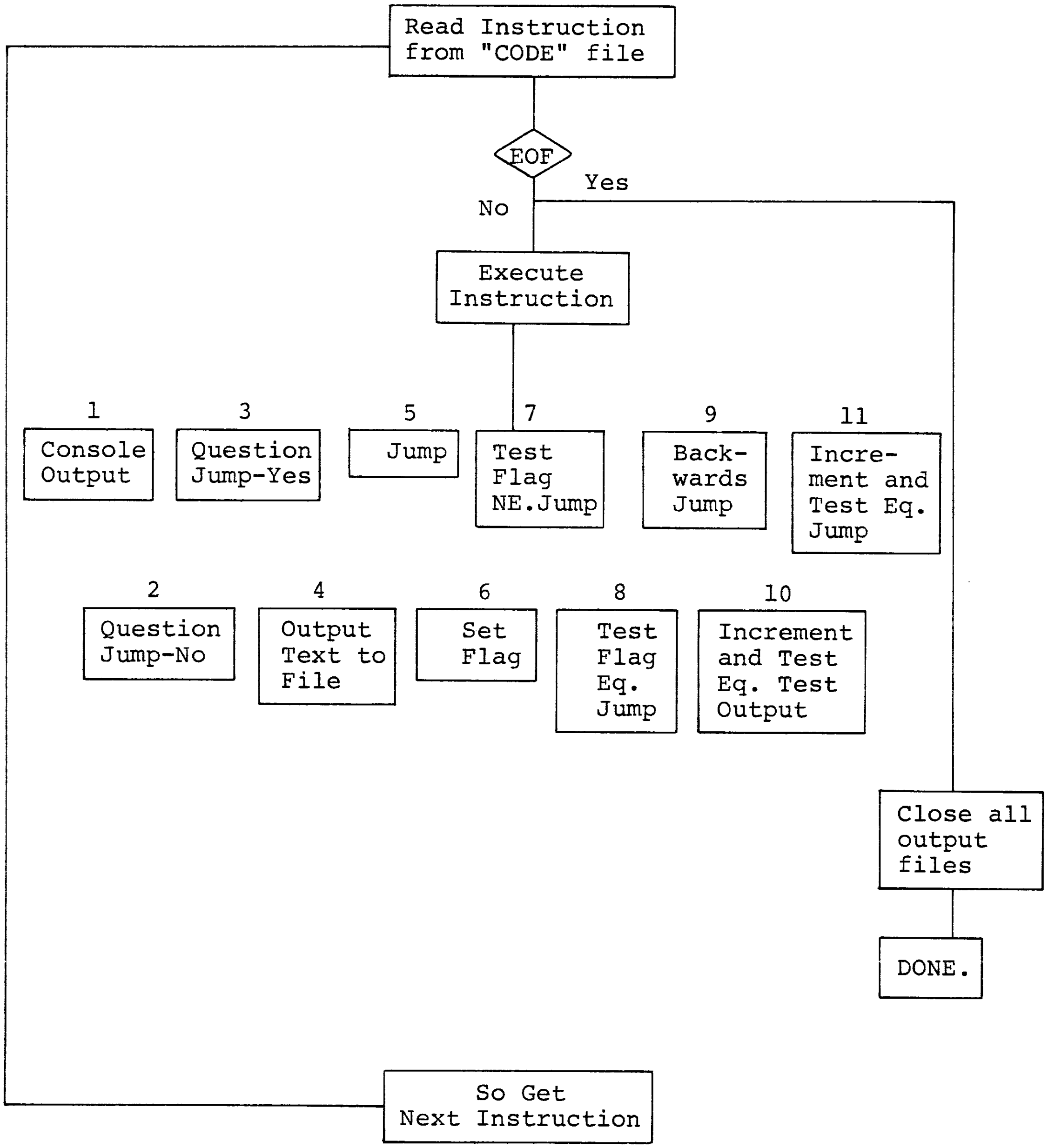
Start

Open the Instruction File "CODE SRC"

Create the Output Files (the BATCH Stream Elements): MACRO BAT, PIP2 BAT, UPDATE BAT, SGEN BAT, PIP3 BAT, PATCH BAT, UBOAT2 BAT, MACII BAT

Output Title $JOB Lines to All Output Files

Determine System Disk Type and Set FLAG(99) Accordingly



D.3 FLAGS

The BUILD program maintains an array of "FLAG" variables. This array "FLAG(99)" is the active storage for the various instructions in the code file. Instructions have the ability to: set, test, and increment these locations. In addition the location "FLAG(99)" is set, by BUILD, to indicate the system disk type (RK=1, RP=2, RF=3).

The various "FLAG" locations are, for XVM/DOS construction, given the following meanings when NON-ZERO:

Internal Documentation for XVM/DOS Build Program

FLAG	VALUE	MEANING
1	1	System has UC15
2	1	System will use RSX
3	1	SPOOLER Required
4	1	BANK MODE Required
5	1	PAGE MODE Required
6	1	8K MACII Required
-	2	12K MACII Required
7	1	RKA. Required
8	1	RKB. Required
9	1	RKC. Required
1Ø	1	RP15 DISK Required
11	1	RF15 DISK Required
12	1	RK11 DISK Required
13	1	LP11 Required
-	2	LP15 Required
14	1	CR11 Required
-	2	CR15 Required
-	3	CRØ1E/CRØ2B Required
-	4	CRØ3B Required
15	1	XY11/XY311 Required
16	1	LV11 PLOTTER TCB Required
17	1	1Ø/15 Communication TCB Required
18	1	VT15 Required
19	1	VP15A Required
2Ø	1	DECtape Required
21	1	MAGTAPE Required
22	1	PAPERTAPE PUNCH Required
23	1	PAPERTAPE READER Required
24	1	VWA. Required
25	1	LKA. Required

FLAG	VALUE	MEANING
26	1	SYSTEM has Floating Point Hardware
27	1	DPA. Required
28	1	DPB. Required
29	1	DPC. Required
3Ø	1	CHAIN Required
31	1	.LOAD Required
32	1	DDT Required
33	1	DUMP Required
34	1	MTDUMP Required
35	1	SRCCOM Required
36	1	8TRAN Required
37	1	FOCAL Required
38	1	FNEW Required
39	1	F4 Required
4Ø	1	F4-OTS Required
41	1	FORT SRC Required
42	1	NUVAL Required
43	1	VECTOR Required
44	1	EDITVP Required
45	1	VTPRIM Required
46	1	DYLDR Required
47	1	LTORPB Required
48	1	TRACK Required
49	1	ROTATE Required
5Ø	1	EDITVT Required
51	1	CIRCLE Required
52	1	MACINT Required
53	1	MCLOAD Required
54	1	PIREX Required
55	1	SPOL15 Required

Internal Documentation for XVM/DOS Build Program

FLAG	VALUE	MEANING
56	1	SPLOAD Required
57	1	SPLGEN Required
58	1	SPOL11 Required
59	1	RESMON Required
6Ø	1	DOSNRM Required
61	1	NRBOSS Required
62	1	B.PRE Required
63	1	BOSS Procedure Files Required
64	1	UDMP Required
65	1	QDMP Required
66	1	DKA. Required
67	1	DKB. Required
68	1	DKC. Required
69	-	UNUSED
7Ø	1	System has API
71	NON-ZERO	Mount Message, Tape 1, Issued
72	NON-ZERO	Mount Message, Tape 2, Issued
73	NON-ZERO	Mount Message, Tape 3, Issued
74	NON-ZERO	Mount Message, Tape 4, Issued
75	NON-ZERO	Mount Message, Tape 5, Issued
76	NON-ZERO	Mount Message, Tape 6, Issued
77	NON-ZERO	Mount Message, Tape 7, Issued
78	NON-ZERO	Mount Message, Tape 8, Issued
79	NON-ZERO	Mount Message, Tape 9, Issued
8Ø	NON-ZERO	Mount Message, Tape 1Ø, Issued
81	NON-ZERO	Mount Message, Tape 11, Issued
82	NON-ZERO	Mount Message, Tape 12, Issued
83	NON-ZERO	Mount Message, Tape 13, Issued
84	NON-ZERO	Mount Message, Tape 14, Issued
85	NON-ZERO	Mount Message, Tape 15, Issued

FLAG	VALUE	MEANING
86	1	PRA. Required
87	1	PRB. Required
88	1	PPB. Required
89	1	PPC. Required
9Ø	1	DTA. Required
91	1	DTC. Required
92	1	DTD. Required
93	1	DTE. Required
94	1	DTF. Required
95	1	MTA. Required
96	1	MTC. Required
97	1	MTF. Required
98	-	UNUSED
99	1	RK11 is System Disk
-	2	RP15 is System Disk
-	3	RF15 is System Disk

In the above table, the term "Required" indicates that the BUILD process may need to produce BATCH file commands to install the associated item. It should be noted that several flags, though given meanings, are not currently used by the XVM/DOS VIAØØØ BUILD process. These flags may be utilized in future releases and are thus listed with the currently used ones.

The FLAGS (71-85) with NON-ZERO values are accessed by Increment And Test instructions. These flags are used to insure that a given "MOUNT" message is output only once into the BATCH stream being created.

D.4 .DAT ASSIGNMENTS FOR THE BUILD PROGRAM

.DAT	USE
1	Input Device for Code File
2 thru 11	Output Device for BATCH Files <BLD> UIC
12 thru 13	Console Device

D.5 INSTRUCTIONS

The "CODE SRC"file contains the instruction for BUILD to interpret. These instructions are in a fixed format:

```
1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18-77
|LABEL| MODE| JUMP   |  FILE  |  FLAG  |␣ TEXT |
```

FIELD	TYPE	WIDTH	JUSTIFICATION	USE
LABEL	Decimal	3	Right	Label for Jumps - Note Labels may be now-unique. Jumps are to the next occurrence of the label in the code file.
MODE	Decimal	3	Right	Instruction OP CODE
JUMP	Decimal	4	Right	Either the label to Jump TO, or, a file I.D. for the Incrementation Test Instruction.
FILE	Decimal	3	Right	Either a file I.D. or a FLAG value.
FLAG	Decimal	3	Right	The FLAG to use
␣	Space	1	Right	Delimiter
TEXT	Alpha-numeric	6Ø	Left	Text for output to console or to BATCH file.

Note: Unused fields should contain Ø for numeric fields and blank for alphanumeric's fields.

D.5.1 "PRINT ON CONSOLE" Instruction

Format:

FIELD	USE
LABEL	Optional
MODE	1
JUMP	Unused
FILE	Unused
FLAG	Unused
TEXT	Printed on Console

Operation: This instruction types the message "TEXT" onto the console.

D.5.2 "QUESTION-JUMP ON NO" Instruction

Format:

FIELD	USE
LABEL	Optional
MODE	2
JUMP	Label to Jump to if the Reply to question is "N"
FILE	Unused
FLAG	Unused
TEXT	Text of Question

Operation: This instruction asks a question in the form:

```
TEXT
REPLY "Y" or "N"?
```

A "Y" reply causes the next sequential instruction to be accessed, a "N" reply causes a jump to the "JUMP" label in the CODE file.

D.5.3 "QUESTION-JUMP ON "YES" Instruction

Format:

FIELD	USE
LABEL	Optional
MODE	3
JUMP	Label to Jump to if the Reply to question is "Y"
FILE	Unused
FLAG	Unused
TEXT	Text of Question

Operation: This instruction asks a question in the form:

```
TEXT
REPLY "Y" or "N"?
```

A "N" reply causes the next sequential instruction to be accessed, a "N" reply causes a jump to the "JUMP" label in the code file.

D.5.4 "OUTPUT TEXT TO FILE" Instruction

Format:

FIELD	USE
LABEL	Optional
MODE	4
JUMP	Unused
FILE	Target BATCH File
FLAG	Unused
TEXT	Text to be Output to BATCH file.

Operation: This instruction outputs the text to BATCH file "FILE"

The values of "FILE" and the corresponding filenames are:

VALUE	FILENAME
2	UPDAT2 BAT
3	PIP2 BAT
4	PIP3 BAT
5	MACRO BAT
6	MACII BAT
7	PATCH BAT
8	UPDATE BAT
9	SGEN BAT

These filenames are of limited significance, indicating only the primary function of these files. The order in which these files link to one another is determined only by the output of respective code file.

D.5.5 "JUMP TO LABEL" Instruction

Format:

FIELD	USE
LABEL	Optional
MODE	5
JUMP	Label to Jump to
FILE	Unused
FLAG	Unused
TEXT	Unused

Operation: This instruction transfers control to the next sequential occurrence of label "JUMP" in the CODE file.

D.5.6 "SET FLAG WITH VALUE" Instruction

Format:

FIELD	USE
LABEL	Optional
MODE	6
JUMP	Unused
FILE	Contains three digit value to be inserted into flag "FLAG"
FLAG	Indicates which flag is set
TEXT	Unused

Operation: This instruction stores the value "FILE" into flag "FLAG".

D.5.7 "TEST FLAG-JUMP NOT EQUAL" Instruction

Format:

FIELD	USE
LABEL	Optional
MODE	7
JUMP	Label to jump to, if contents of "FLAG" not equal to "FILE"

FIELD	USE
FILE	Contains a three digit value to be compared with the content of "FLAG"
FLAG	Indicates which flag is to be tested
TEXT	Unused

Operation: This instruction compares the value "FILE" against the contents of the flag "FLAG" and jumps to label "JUMP" if not equal. If equal the next sequential instruction is executed.

D.5.8 "TEST FLAG-JUMP EQUAL" Instruction

Format:

FIELD	USE
LABEL	Optional
MODE	8
JUMP	Label to jump to, if contents of "FLAG" are equal to "FILE"
FILE	Contains a three digit value to be compared with the contents of "FLAG"
FLAG	Indicates which flag is to be tested
TEXT	Unused

Operation: This instruction compares the value "FILE" against the contents of the flag "FLAG" and jumps to the label "JUMP" if equal. If not equal the next sequential instruction is executed.

D.5.9 "BACKWARDS JUMP" Instruction

Format:

FIELD	USE
LABEL	Optional
MODE	9
JUMP	Label to jump to
FILE	Unused

FIELD	USE
FLAG	Unused
TEXT	Unused

Operation: This instruction causes the "CODE SRC" file to rewind and a jump to the first occurrence of the label "JUMP" to occur. This instruction is useful for repeating sections of the CODE file.

D.5.10 "INCREMENT AND TEST-OUTPUT TO FILE IF EQUAL" Instruction

Format:

FIELD	USE
LABEL	Optional
MODE	1∅
JUMP	Target BATCH file
FILE	Contains a three digit value to be compared with the contents of "FLAG"
FLAG	Indicates which flag is to be tested
TEXT	Text to be output to BATCH file

Operation: This instruction operates as follows:

a. Increment flag "FLAG"

b. Compare the contents of flag "FLAG" with the value of "FILE"

c. If the compare is equal then the text "TEXT" is output to BATCH file "JUMP"

d. The next sequential instruction is accessed

D.5.11 "INCREMENT AND TEST-JUMP IF EQUAL" Instruction

Format:

FIELD	USE
LABEL	Optional
MODE	11
JUMP	Label to jump to

FIELD	USE
FILE	Contains a three digit value to be compared with the contents of "FLAG"
FLAG	Indicates which flag is to be tested
TEXT	Unused

Operation: This instruction operates as follows:

a. Increment flag "FLAG"

b. Compare the contents of flag "FLAG" with the value of "FILE"

c. If the compare is equal then jump to label "JUMP"

d. If compare is not equal then access the next sequential instruction

D.6 BATCH FILE HINTS

While normal XVM/DOS commands may be entered into the constructed BATCH file, several "Tricks" may be useful:

D.6.1 Chaining

The BATCH line:

```
            "BATCH SY FILE1"
```

When encountered in a BATCH file will cause transfer of control to the BATCH file "FILE1" on device "SY".

This method is useful for linking separate batch files on mass storage devices.

D.6.2 Batching "SGEN"

The output of the BUILD program is somewhat incompatible with the required input of SGEN. When using BUILD to produce "SGEN" batch files the user should substitute ← (backarrow) for the default replys. In addition, all non-default replys must contain a multiple of five characters in the reply with the last character being non-blank. Note that leading blanks are valid and may be used to "pad" replys.

D.6.3 Batching "PATCH"

When adding "SYSBLK" values for "ABS" files being PATCHED into the system, the terminating "ALTMODE" is not output by "BUILD". Any INVALID command may be substituted for the "ALTMODE". While this will produce an error message "INVALID COMMAND", the result is the same as if an altmode had been entered.

D.6.4 Batching "MAC11"

MAC11 requires the same multiple of five command line that SGEN does. See B.6.2 for further information.

D.6.5 Batching MACRO XVM

When using MACRO XVM with the "P" (assembly parameters) option, the user will want to terminate the list of assembly parameters. To do this a $END command is used. This causes an .EOT indication to be passed to MACRO XVM.

The $END simulates ↑D.

D.6.6 Terminating Repetitive SYSTEM Programs Under Batching

When using system programs that normally request new command lines after completing a given task (such as PIP or MACRO XVM), a $JOB command can be used to cause control to be passed back to the BATCH stream. This is most useful for the BUILD process, since BUILD normally does not output ALTMODES (the only other way to terminate a repetitive system program).

The $JOB simulates ↑C. (But is not fatal to the current BATCH stream.)

D.7 "BUILD" COMPONENTS

The following is a list of source components of the XVM/DOS VIAØØØ build system.

Component		Language	Use
BUILD	SRC	F4	Main Routine Build Interpreter
REPLY	SRC	F4	Yes/No Reply Subroutine
TTOUT	SRC	F4	Console Output Subroutine
SYSDEV	ØØØ	MACRO XVM	System Device Determination Subroutine
CODE	DTA	BUILD CODE	Code File for DECtape Builds
CODE	MTA	BUILD CODE	Code File for MAGTAPE Builds
CHECK	ØØØ	MACRO XVM	Tape Correctness Checker
BUILD	BAT	XVM/DOS Commands	BATCH File to Run Build and Start BATCH Stream
URK	PRM	MAC11	RK Disk Assembly Parameter
ULP	PRM	MAC11	LP11 Assembly Parameter
UCD	PRM	MAC11	CR11 Assembly Parameter
ULV	PRM	MAC11	LV11 Assembly Parameter
UXY	PRM	MAC11	XY Plotter Assembly Parameter
U311	PRM	MAC11	XY311 Assembly Parameter
UMM	PRM	MAC11	Metric Plotter Assembly Parameter

The BUILD execute file is composed of:

BUILD	SRC
REPLY	SRC
TTOUT	SRC
SYSDEV	ØØØ

The <BLD> UIC contains:

a. The BUILD execute file

b. The CHECK execute file

c. Either CODE DTA or CODE MTA renamed to CODE SRC

d. BUILD BAT

e. All of the MAC11 Assembly Parameter files (XXXX PRM).

The .DAT assignments used throughout the BUILD process are:

.DAT	UIC	Device
-4	BLD	SYA
1 thru 11	Current	SYA
12 thru 13	Current	TTA.
16	Current	PPA.
17 thru 20	Current	Either DTA. on DECtape release systems or MTA. on MAGTAPE release systems.

APPENDIX E
ASSEMBLY PARAMETERS

E.1 ASSEMBLY PARAMETERS FOR XVM/DOS V1A000

Table E-1 provides the Assembly Parameters for XVM/DOS V1A000.

E.2 ASSEMBLY PARAMETERS FOR UNICHANNEL COMPONENTS OF XVM/DOS V1A000

Table E-2 provides the Assembly Parameters for UNICHANNEL Components of XVM/DOS V1A000.

E.3 ASSEMBLY PARAMETERS FOR FORTRAN OTS COMPONENTS OF XVM/DOS V1A000

Table E-3 consists of a list of all routines of the XVM FORTRAN Object Time System, along with relevant conditional assembly parameters. All routines are listed for both RSX and DOS, and both hardware floating point (FP15), or non-hardware floating point systems. Routines are marked by some standard notes as to their applicability to various systems.

Any assembly parameters not described in the table are not used in standard XVM systems and are not supported by DEC.

Some standard assembly modifications (i.e., non-double buffered, etc.) appear at the end of Table E-3.

The libraries being used for BANK and PAGE mode are now identical, so only two libraries FP15 and non-FP15 are distributed with each system.

To create an OTS for greater than 20_8 .DAT slots; .FLTB, FIOPS and STOP must be reassembled.

To create an OTS for greater than 4 random access files open simultaneously; DEFINE must be reassembled.

To create an OTS which doesn't use double-buffered I/O; AUXIO and FIOPS must be reassembled.

Assembly Parameters

To create an OTS with DOS-15 V2A- style or DOS-15 V3A- style carriage control character usage; BCDIO and RBCDIO must be reassembled.

As an identification convention dummy routines are inserted at the head of the library (using UPDATE). These dummy programs are macro routines of the form:

```
START   Ø
        .END    START
```

These routines have a name and extension which conveys information about the type of library and when it was last modified and by whom.

The first dummy gives the mode (whether floating point hardware or not) in the filename and the system (whether DOS or RSX) in the extension. For example:

```
NFPP   DOS
```

Means the non-FP15 library for XVM/DOS. Following this dummy entry are entries which date the last time the library was updated and by whom. For example:

```
Ø5DC75   ABC
```

means that the library was last modified on 05-DEC-75 by someone with the initials "ABC". Each time the library is modified, the new date file is inserted after the mode dummy using the UPDATE command "INSERT". In this way, the complete update history is maintained in the library. This convention is assumed to be followed by users as well as DEC personnel. The abbreviations to be used for months are:

January	JA	July	JL
February	FE	August	AU
March	MR	September	SE
April	AP	October	OC
May	MY	November	NV
June	JN	December	DC

Following, in Figures E-1 and E-2, are UPDATE listings of the DOS libraries distributed with XVM/DOS V1A000.

Table E-1

Assembly Parameters for XVM/DOS V1A000

Source File Name	Binary Name	Version	Assembly Parameters	Type of Binary	READR Argument (PATCH)	No. of Blocks (SGEN)	No. of Buffers (SGEN)	.DAT Assignments (SGEN)	Notes (at end of table)
ABSL11	N/A	N/A	N/A	HRM	N/A	N/A	N/A	N/A	1.
B.PRE	B.PRE	N/A	N/A	BIN	16077	3	2	-14,-15,5,6	3.
CD.DOS	CDB.	CRØ3B model	N/A	BIN	N/A	N/A	N/A	N/A	2.
	CDB.	CR15 model	CR15=Ø	BIN	N/A	N/A	N/A	N/A	2.
	CDB.	CRØ1E/CRØ2B model	NOTGDI=Ø	BIN	N/A	N/A	N/A	N/A	2.
	CDB.	CR11 model	UC15=Ø	BIN	N/A	N/A	N/A	N/A	2.
	CDB.	Ø29 card code	N/A	BIN	N/A	N/A	N/A	N/A	2.
	CDB.	Ø26 card code	DECØ26=Ø	BIN	N/A	N/A	N/A	N/A	2.
	CDB.	Spooled (CR11 only)	N/A	BIN	N/A	N/A	N/A	N/A	2.
	CDB.	Unspooled (CR11 only)	NOSPL=Ø	BIN	N/A	N/A	N/A	N/A	2.
CHAIN	CHAIN	N/A	N/A	BIN	N/A	3Ø	4	-1,-4,-5,-6	3.
CIRCLE	CIRCLE	N/A	Compile with FORTRAN	BIN	N/A	N/A	N/A	N/A	7.

Table E-1

Assembly Parameters for XVM/DOS V1A000 (cont)

Source File Name	Binary Name	Version	Assembly Parameters	Type of Binary	READR Argument (PATCH)	No. of Blocks (SGEN)	No. of Buffers (SGEN)	.DAT Assignments (SGEN)	Notes (at end of table)
CREF	CREF	N/A	N/A	BIN	N/A	N/A	N/A	N/A	4.
DDT	DDT	N/A	N/A	BIN	N/A	N/A	N/A	N/A	4.
DOSNRM	DOS15	Automatic mode message	N/A	BIN	16Ø77	N/A	N/A	N/A	3.
		No automatic mode message	NOMODE=Ø	BIN	16Ø77	N/A	N/A	N/A	3.
DOSFRA	DKA.	DKA.	N/A	BIN	N/A	N/A	N/A	N/A	2.
	DKB.	DKB.	%BVERS=Ø	BIN	N/A	N/A	N/A	N/A	2.
	DKC.	DKC.	$CVERS=Ø	BIN	N/A	N/A	N/A	N/A	2.
	DKL.	DKL.	%LVERS=Ø	ABS	N/A	N/A	N/A	N/A	5.
	DPA.	DPA.	%RPØ2=Ø	BIN	N/A	N/A	N/A	N/A	2.
	DPB	DPB.	%RPØ2=Ø %BVERS=Ø	BIN	N/A	N/A	N/A	N/A	2.
	DPC	DPC	%RPØ2=Ø %CVERS=Ø	BIN	N/A	N/A	N/A	N/A	2.
	DPL.	DPL.	%RPØ2=Ø %LVERS=Ø	ABS	N/A	N/A	N/A	N/A	5.
	RKA.	RKA.	%RPØ2=Ø %RKØ5=Ø	BIN	N/A	N/A	N/A	N/A	2.

Table E-1

Assembly Parameters for XVM/DOS V1A000 (cont)

Source File Name	Binary Name	Version	Assembly Parameters	Type of Binary	READR Argument (PATCH)	No. of Blocks (SGEN)	No. of Buffers (SGEN)	.DAT Assignments (SGEN)	Notes (at end of table)
	RKB.	RKB.	%RPØZ=Ø %RKØ5=Ø %BVERS=Ø	BIN	N/A	N/A	N/A	N/A	2.
	RKC.	RKC.	%RPØ2=Ø %RKØ5=Ø %CVERS=Ø	BIN	N/A	N/A	N/A	N/A	2.
	RKL.	RKL.	%RPØ2=Ø %RKØ5=Ø %LVERS=Ø	ABS	N/A	N/A	N/A	N/A	5.
DOSSAV	N/A	N/A	N/A	HRM	N/A	N/A	N/A	N/A	6.
DTA.	DTA.	N/A	N/A	BIN	N/A	N/A	N/A	N/A	2.
DTC.	DTC.	N/A	N/A	BIN	N/A	N/A	N/A	N/A	2.
DTD.	DTD.	N/A	N/A	BIN	N/A	N/A	N/A	N/A	2.
DTE.	DTE.	N/A	N/A	BIN	N/A	N/A	N/A	N/A	2.
DTF.	DTF.	N/A	N/A	BIN	N/A	N/A	N/A	N/A	2.
DUMP	DUMP	N/A	N/A	BIN	N/A	5	2	-12,-14	3.
DYLDR	DYLDR	N/A	N/A	BIN	N/A	N/A	N/A	N/A	7.

Table E-1

Assembly Parameters for XVM/DOS V1A000 (cont)

Source File Name	Binary Name	Version	Assembly Parameters	Type of Binary	READR Argument (PATCH)	No. of Blocks (SGEN)	No. of Buffers (SGEN)	.DAT Assignments (SGEN)	Notes (at end of table)
EDIT15	EDIT	EDIT	N/A	BIN	N/A	26	4	-15,-14,-13 -12,-10	3.
	EDITVP	EDITVP	EDITVP=Ø	BIN	N/A	26	4	-15,-14,-13 -12,-10,10	3.
	EDITVT	EDITVT	EDITVT=Ø	BIN	N/A	26	4	-15,-14,-13 -12,-10	3.
EXECUT	EXECUT	N/A	N/A	BIN	N/A	N/A	N/A	N/A	4.
FNEW	FNEW	N/A	N/A	BIN	N/A	N/A	N/A	N/A	4.
FOCAL	FOCAL	PAGE MODE	N/A	BIN	N/A	N/A	N/A	N/A	4.
	FOCAL	BANK MODE	%PDP9=Ø %PDP15=Ø	BIN	N/A	N/A	N/A	N/A	4.
FORT	FORT	N/A	N/A	BIN	N/A	N/A	N/A	N/A	7.
FORTRAN OTS	FORTRAN OTS	N/A	N/A	BIN	N/A	N/A	N/A	N/A	8.
F4MP1 F4MP2	F4	Non-floating point hardware	N/A	FIN	N/A	37	3	-11,-12, -13	3,9.
	F4	Floating Point hardware	%FPP=Ø	BIN	N/A	40	3	-11,-12, -13	3,9.

Table E-1

Assembly Parameters for XVM/DOS V1A000 (cont)

Source File Name	Binary Name	Version	Assembly Parameters	Type of Binary	READR Argument (PATCH)	No. of Blocks (SGEN)	No. of Buffers (SGEN)	.DAT Assignments (SGEN)	Notes (at end of table)
INSALL	INSALL SRC	N/A	N/A	N/A	N/A	N/A	N/A	N/A	4.
INSERR	INSERR SRC	N/A	N/A	N/A	N/A	N/A	N/A	N/A	4.
INSTRC	INSTRC	N/A	N/A	BIN	N/A	N/A	N/A	N/A	4.
LKA.	LKA.	N/A	N/A	BIN	N/A	N/A	N/A	N/A	2.
.LOAD	.LOAD	PAGE MODE	N/A	BIN	N/A	N/A	N/A	N/A	4.
	.LOAD	BANK MODE	PDP9=Ø	BIN	N/A	N/A	N/A	N/A	4.
LPA.15	LPA.	FORM FEED ON .CLOSE	N/A	BIN	N/A	N/A	N/A	N/A	2.
	LPA.	NO FORM FEED ON .CLOSE	NOFF=1	BIN	N/A	N/A	N/A	N/A	2.
LPU.	LPA.	FORM FEED ON .CLOSE	N/A	BIN	N/A	N/A	N/A	N/A	2.
	LPA.	NO FORM FEED ON .CLOSE	NOFF=Ø	BIN	N/A	N/A	N/A	N/A	2.
	LPA.	LINE COUNT OTHER THAN 58.	FFCNT= OCTAL #	BIN	N/A	N/A	N/A	N/A	2.
	LPA.	SPOOLED when SPOOLER enabled	N/A	BIN	N/A	N/A	N/A	N/A	2.
	LPA.	UNSPOOLED always	NOSPL=Ø	BIN	N/A	N/A	N/A	N/A	2.
LTORPB	LTORPB	N/A	N/A	BIN	N/A	N/A	N/A	N/A	7.
MACIMG	MCLOAD	N/A	N/A	BIN	N/A	5	1	-14	3.

Table E-1

Assembly Parameters for XVM/DOS V1A000 (cont)

Source File Name	Binary Name	Version	Assembly Parameters	Type of Binary	READR Argument (PATCH)	No. of Blocks (SGEN)	No. of Buffers (SGEN)	.DAT Assignments (SGEN)	Notes (at end of table)
MACINT	MAC11	N/A	N/A	ABS	N/A	4Ø	2	-11,-12	1Ø.
MACR15	MACRO	N/A	N/A	BIN	N/A	N/A	N/A	N/A	3.
MTA.	MTA.	N/A	N/A	BIN	N/A	N/A	N/A	N/A	2.
MTC.	MTC.	N/A	N/A	BIN	N/A	N/A	N/A	N/A	2.
MTDUMP	MTDUMP	N/A	N/A	BIN	N/A	13	2	1,3	3.
MTF.	MTF.	N/A	N/A	BIN	N/A	N/A	N/A	N/A	2.
NRBOSS	BOSS	N/A	N/A	ABS	N/A	15	3	-15,-14, -11,5,6	1Ø.
NUVAL	NUVAL	N/A	N/A	BIN	N/A	N/A	N/A	N/A	7.
PATCH	PATCH	N/A	N/A	ABS	N/A	N/A	N/A	N/A	1Ø.
PIP	PIP	N/A	N/A	BIN	N/A	N/A	N/A	N/A	3.
PPA.	PPA.	N/A	N/A	BIN	N/A	N/A	N/A	N/A	2.
PPB.	PPB.	N/A	N/A	BIN	N/A	N/A	N/A	N/A	2.
PPC.	PPC.	N/A	N/A	BIN	N/A	N/A	N/A	N/A	2.
PRA.	PRA.	N/A	N/A	BIN	N/A	N/A	N/A	N/A	2.
PRB.	PRB.	N/A	N/A	BIN	N/A	N/A	N/A	N/A	2.
QFILE	QFILE	N/A	N/A	BIN	17477	N/A	N/A	N/A	3.

Table E-1

Assembly Parameters for XVM/DOS V1A000 (cont)

Source File Name	Binary Name	Version	Assembly Parameters	Type of Binary	READR Argument (PATCH)	No. of Blocks (SGEN)	No. of Buffers (SGEN)	.DAT Assignments (SGEN)	Notes (at end of table)
RESMNA RESMNB	RESMON	Include Mini-ODT	%ODT=Ø	ABS	N/A	N/A	N/A	N/A	9.,10.
	RESMON	Include all TCB's	ALLTCB=Ø	ABS	N/A	N/A	N/A	N/A	9.,10.
	RESMON	Include card reader TCB	CDTCB=Ø	ABS	N/A	N/A	N/A	N/A	9.,10.
	RESMON	Include XY plotter TCB	PLTCB=Ø	ABS	N/A	N/A	N/A	N/A	9.,10.
	RESMON	Include spare TCB #1	TCB1=Ø	ABS	N/A	N/A	N/A	N/A	9.,10.
	RESMON	Include spare TCB #2	TCB2=Ø	ABS	N/A	N/A	N/A	N/A	9.,10.
	RESMON	Include spare TCB #3	TCB3=Ø	ABS	N/A	N/A	N/A	N/A	9.,10.
	RESMON	Include LV printer/plotter TCB	LVTCB=Ø	ABS	N/A	N/A	N/A	N/A	9.,10.
	RESMON	Include communications TCB	PLTCB=Ø	ABS	N/A	N/A	N/A	N/A	9.,10.
RFBOOT	N/A	N/A	N/A	HRM	N/A	N/A	N/A	N/A	6.
RKBOOT	N/A	N/A	N/A	HRM	N/A	N/A	N/A	N/A	6.
ROTATE	ROTATE SRC	N/A	Compile with FORTRAN	BIN	N/A	N/A	N/A	N/A	7.
RFBOOT	N/A	N/A	N/A	HRM	N/A	N/A	N/A	N/A	6.
SGEN	SGEN	N/A	N/A	BIN	16122	N/A	N/A	N/A	3.

Table E-1

Assembly Parameters for XVM/DOS V1A000 (cont)

Source File Name	Binary Name	Version	Assembly Parameters	Type of Binary	READR Argument (PATCH)	No. of Blocks (SGEN)	No. of Buffers (SGEN)	.DAT Assignments (SGEN)	Notes (at end of table)
SGNBLK	SGNBLK	RF SYSTEM	N/A	ABS	N/A	N/A	N/A	N/A	11.
	SGNBLK	RK SYSTEM	RKØ5=Ø	ABS	N/A	N/A	N/A	N/A	11.
	SGNBLK	RP SYSTEM	RPØ2=Ø	ABS	N/A	N/A	N/A	N/A	11.
SPLGEN	SPLGEN	N/A	N/A	BIN	N/A	25	1	-14	3.
SPLIMG	SPLOAD	N/A	N/A	BIN	N/A	5	1	-14	3.
SPOL15	SPOOL	N/A	N/A	BIN	N/A	55	1	-14	3.
SRCCOM	SRCCOM	N/A	N/A	BIN	N/A	13	2	-14,-15	3.
SYSBLK	SYSBLK	RF SYSTEM	N/A	ABS	N/A	N/A	N/A	N/A	12.
	SYSBLK	RK SYSTEM	RKØ5=Ø	ABS	N/A	N/A	N/A	N/A	12.
	SYSBLK	RP SYSTEM	RPØ2=Ø	ABS	N/A	N/A	N/A	N/A	12.
.SYSLD	.SYSLD	N/A	N/A	ABS	N/A	N/A	N/A	N/A	10.
TRACK	TRACK	N/A	N/A	BIN	N/A	N/A	N/A	N/A	7.
UPDATE	UPDATE	N/A	N/A	ABS	N/A	N/A	N/A	N/A	10.
VECTOR	VECTOR SRC	N/A	COMPILE WITH FORTRAN	BIN	N/A	N/A	N/A	N/A	7.
VPA.	VPA.	SCAN MODE	N/A	BIN	N/A	N/A	N/A	N/A	2.
	VPA	PAGING MODE	VPA.5=Ø	BIN	N/A	N/A	N/A	N/A	2.
VTA.	VTA.	N/A	N/A	BIN	N/A	N/A	N/A	N/A	2.
VTPRIM	VTPRIM	N/A	N/A	BIN	N/A	N/A	N/A	N/A	7.

Table E-1

Assembly Parameters for XVM/DOS V1A000 (cont)

Source File Name	Binary Name	Version	Assembly Parameters	Type of Binary	READR Argument (PATCH)	No. of Blocks (SGEN)	No. of Buffers (SGEN)	.DAT Assignments (SGEN)	Notes (at end of table)
VWA.	VWA.	N/A	N/A	BIN	N/A	N/A	N/A	N/A	2.
XYU.	XYA.	XY11 PLOTTER	N/A	BIN	N/A	N/A	N/A	N/A	2.
	XYA.	XY311 PLOTTER	PL311=Ø	BIN	N/A	N/A	N/A	N/A	2.
	XYA.	SPOOLED WHEN SPOOLER ENABLED	N/A	BIN	N/A	N/A	N/A	N/A	2.
	XYA.	UNSPOOLED ALWAYS	NOSPL=Ø	BIN	N/A	N/A	N/A	N/A	2.
8TRAN	8TRAN	N/A	N/A	BIN	N/A	13	2	-14,-15	3.

NOTES: ON ASSEMBLY PARAMETERS

1. The ABSL11 loader cannot be assembled with the version of MACRO XVM supplied to customers. It requires a special version of MACRO XVM, called MACRO-D, used to generate XVM Diagnostic programs. The difference is in the load addresses for HRM tapes: 1772Ø for MACRO XVM and 177ØØ for MACRO-D.

2. Nearly all relocatable binary I/O handlers should be transferred, using PIP, to the IOS directory, e.g.,

```
>T␣DK␣<IOS>←DK␣<SCR>␣CDB.␣BIN↵
```

3. This System Program, assembled to BIN extension is installed using the PATCH READR command. Where indicated, the READR command has a parameter to be used when installing the program.

NOTES: (continued)

4. The binary of this routine should be transferred to the <BNK> and <PAG> directories.

5. The absolute version of the system disk handler is installed using the PATCH READ command into .SYSLD.

6. HARDWARE READIN MODE paper tape binary.

7. The binary of this routine should be transferred to the <PER> directory.

8. Assembly instructions for the FORTRAN OBJECT TIME SYSTEM routines are given in an appendix of the XVM/DOS FORTRAN IV OPERATING ENVIRONMENT manual.

9. This is a two part source file and is assembled using the BPFZ options.

10. This system program, assembled to ABS extension, is installed using the PATCH READ command.

11. The absolute table SGNBLK is installed using the PATCH READ command into RESMON.

12. SYSBLK is built into the basic skelton provided with the XVM/DOS kit.

Table E-2
Assembly Parameters for Unichannel Components of XVM/DOS V1A000

UNICHANNEL ASSEMBLY PARAMETERS

Source File Name	Devices	Assembly Parameters	Notes
PIREX	NO DEVICES		1
	RKØ5 DISK	$RK=1ØØØØØ	1
	CR11 CARD READER	$CD=2ØØØØ	1
	LP11/LS11/LV11 (Printer)	$LP=4ØØØØ	1
	LV11	$LV=Ø	1
	XY11 PLOTTER	$PL=1ØØØØ	1
	XY311 PLOTTER with inches scale	$PL=1ØØØØ $PL311=Ø	1
	XY311 PLOTTER with metric scale	$PL=1ØØØØ $PL311=Ø $UNITS=Ø	1
	DISABLE XYLL OFFLINE SWITCH and SPOOLER HALT-AT-END-OF-FILE switches	$NOSW=Ø	1,2
	DECTAPE (unsupported)	$DT=Ø	3
SPOL11	CARD READER	$CD=2ØØØØ	1
	LINE PRINTER	$LP=4ØØØØ	1
	XY PLOTTER	$PL=1ØØØØ	1

NOTES: ON ASSEMBLY PARAMETERS FOR UNICHANNEL COMPONENTS OF XVM/DOS V1A000

1. MAC11 Assembly Parameters must be a part of the main source file.

2. <u>OFFLINE SWITCH</u> - The CALCOMP XY11 plotter does not have an offline switch. This leads to difficulty in stopping the plotter to refill pen, take off plots, etc. Bit 2 switch of the PDP-11 console switches is to be used as the offline switch for the CALCOMP. If you do <u>NOT</u> wish this feature define $NOSW=Ø.

3. This option is provided as an example of a complex device driver, and is not intended to produce a working driver.

Table E-3
Assembly Parameters for FORTRAN OTS Components of XVM/DOS V1A000

Source	Standard Notes	Additional Parameters	
		Parameter	Function
.BB ØØ4	None	N/A	N/A
.BC Ø1Ø	1	N/A	N/A
.BD Ø1Ø	1	N/A	N/A
.BE ØØ6	1	N/A	N/A
.BF ØØ5	1	N/A	N/A
.BG ØØ8	1	N/A	N/A
.BH ØØ5	1	N/A	N/A
.BI ØØ4	1	N/A	N/A
.BP ØØØ	5	N/A	N/A
.CAT1 ØØ1	1	N/A	N/A
.CB ØØ4	None	N/A	N/A
.DA Ø13	4	N/A	N/A
.DB ØØ4	1	N/A	N/A
.DC ØØ1	1	N/A	N/A
.DE ØØ3	1	N/A	N/A
.DF ØØ1	1	N/A	N/A
.DJ ØØØ	None	N/A	N/A

Table E-3

Assembly Parameters for FORTRAN OTS Components of XVM/DOS V1A000 (cont)

Source	Standard Notes	Additional Parameters	
		Parameter	Function
.EB ØØ4	1	N/A	N/A
.EC ØØ1	1	N/A	N/A
.EE ØØ2	1	N/A	N/A
.EF ØØ8	1	N/A	N/A
.FLTB ØØ4	6	FLTBSZ=n	To increase size of .DAT slot table to "n". Default is 20_8
.FP ØØØ	5	N/A	N/A
.FPP Ø18	2	N/A	N/A
.SS ØØ9	None	N/A	N/A
ABS ØØ2	1	N/A	N/A
AINT ØØ2	1	N/A	N/A
ALOG ØØ4	1	N/A	N/A
ALOG1Ø ØØ4	1	N/A	N/A
AMOD ØØ3	1	N/A	N/A
ATAN ØØ2	1	N/A	N/A
ATAN2 ØØ8	1	N/A	N/A
AUX10 Ø19	4	%NODBL=Ø	To use single buffered I/O. % applies only to DOS systems.

Table E-3

Assembly Parameters for FORTRAN OTS Components of XVM/DOS V1A000 (cont)

.Source		Standard Notes	Additional Parameters	
			Parameter	Function

BCDIO	Ø52	1,4	MSCC=Ø
			LFTOSP=Ø
				To cause the first character of each formatted output record to a non-printer device to be translated to an internal carriage control character.
				To cause an initial line feed character to converted to a space character on formatted input records.

BINIO	Ø2Ø	4	N/A	N/A
COS	ØØ3	1	N/A	N/A
DABS	ØØ1	1	N/A	N/A
DATAN	ØØ1	1	N/A	N/A
DBLE	ØØ1	1	N/A	N/A
DBLINT	ØØ7	3	N/A	N/A
DCOS	ØØ2	1	N/A	N/A
DDIO	Ø19	1,4	N/A	N/A
DEFINE	Ø2Ø	4	%TBSIZ=n	The maximum number of Random Access files which may be open at a time. Default is 4.
DEXP	ØØ1	1	N/A	N/A
DIM	ØØ2	1	N/A	N/A
DLOG	ØØ5	1	N/A	N/A
DLOG1Ø	ØØ3	1	N/A	N/A

Table E-3

Assembly Parameters for FORTRAN OTS Components of XVM/DOS V1A000 (cont)

Source	Standard Notes	Additional Parameters	
		Parameter	Function
DMOD ØØ4	1	N/A	N/A
DMNMX Ø11	1,4	N/A	N/A
DSIGN ØØ4	1	N/A	N/A
DSIN ØØ1	1	N/A	N/A
DSQRT ØØ7	1	N/A	N/A
DOUBLE ØØ4	3	N/A	N/A
EDCODE ØØ3	None	N/A	N/A
EOF ØØØ	6	N/A	N/A
ERRSET ØØØ	None	N/A	N/A
EXP ØØ2	1	N/A	N/A
FILE Ø1Ø	6	N/A	N/A
FIOPS Ø42	4	%NODBL=Ø	To use single buffered I/O. %NODBL applies only to DOS Systems.
		DKTBSZ=n	To increase size of .DAT slot (DOS) or LUN (RSK) table to "n". Default size is 2Ø$_8$(DOS) or 77$_8$(RSX).
		STTBSZ=n	To increase size of .DAT slot (DOS) or LUN (RSX) table to "n". Default size is 2Ø$_8$(DOS) or 77$_8$(RSX).
FLOAT ØØ2	None	N/A	N/A
FLOATJ ØØ1	1	N/A	N/A

Table E-3

Assembly Parameters for FORTRAN OTS Components of XVM/DOS V1A000 (cont)

Source	Standard Notes	Additional Parameters	
		Parameter	Function
GOTO ØØ3	None	N/A	N/A
IABS ØØØ	None	N/A	N/A
IDIM ØØØ	None	N/A	N/A
IDINT ØØØ	1	N/A	N/A
IDZERO ØØ1	None	N/A	N/A
IFIX ØØ2	1	N/A	N/A
IFLOW ØØ1	None	N/A	N/A
IMNMX ØØ9	4	N/A	N/A
INT ØØ2	1	N/A	N/A
INTEAE ØØ9	None	N/A	N/A
IOERR ØØ2	6	N/A	N/A
ISENSW ØØ1	None	N/A	N/A
ISIGN ØØØ	None	N/A	N/A
ISNGL ØØ2	1	N/A	N/A
JABS ØØ1	1	N/A	N/A
JDBLE ØØ1	1	N/A	N/A
JDFIX ØØ1	1	N/A	N/A

Table E-3

Assembly Parameters for FORTRAN OTS Components of XVM/DOS V1A000 (cont)

Source	Standard Notes	Additional Parameters	
		Parameter	Function
JDIM ØØ1	1	N/A	N/A
JFIX ØØ1	1	N/A	N/A
JMNMX ØØ5	1,4	N/A	N/A
JMOD ØØ3	1	N/A	N/A
JSIGN ØØ4	1	N/A	N/A
MOD ØØØ	None	N/A	N/A
OTSER Ø15	1,4	N/A	N/A
PARTWD ØØ3	1	N/A	N/A
PAUSE ØØ6	4	N/A	N/A
RANCOM Ø14	4	N/A	N/A
RBCDIO ØØ9	4	MSCC=Ø	To cause the first character of each formatted output record to a non-printer device to be translated to an internal carriage control character.
RBINIO ØØ6	4	N/A	N/A
RELEAE Ø11	3	N/A	N/A
RMNMX Ø14	1,4	N/A	N/A
SIGN ØØ4	1	N/A	N/A
SIN ØØ3	1	N/A	N/A

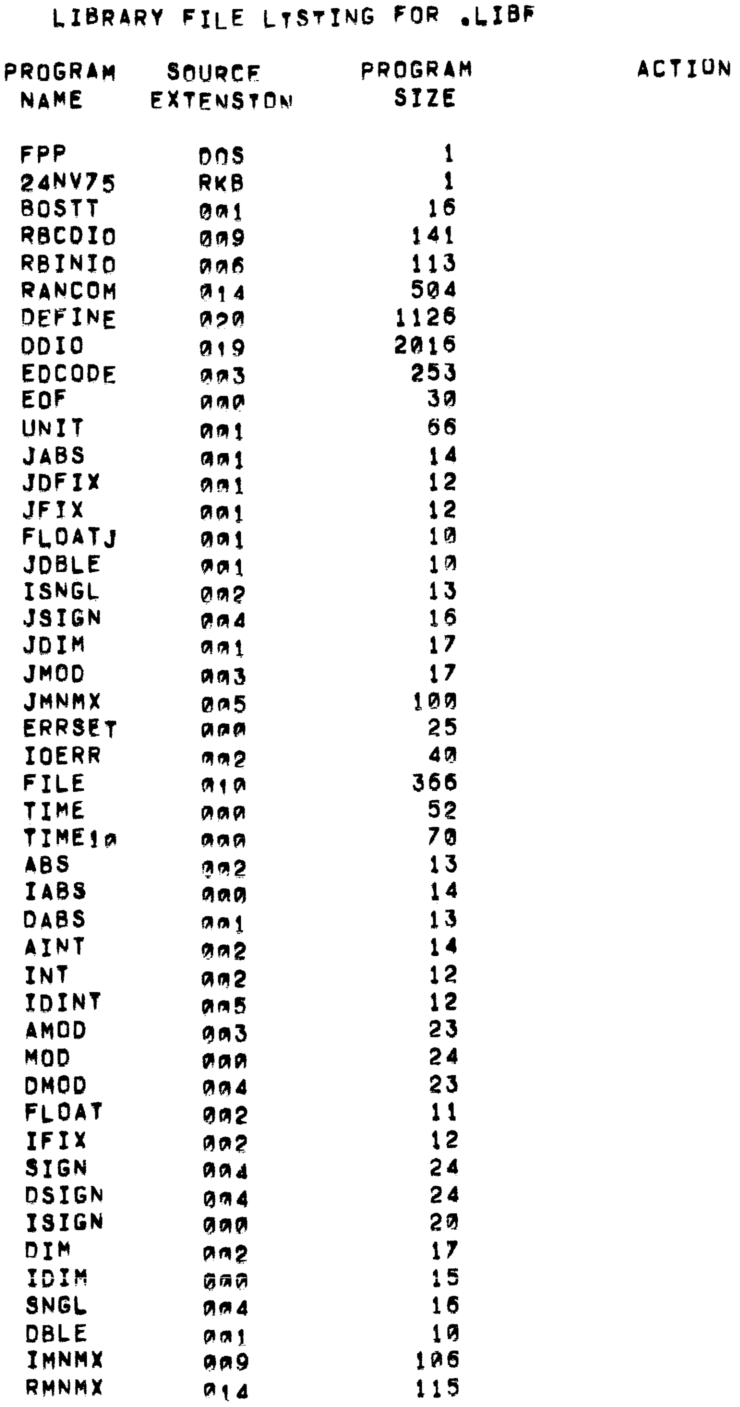
Table E-3

Assembly Parameters for FORTRAN OTS Components of XVM/DOS V1A000 (cont)

Source	Standard Notes	Additional Parameters	
		Parameter	Function
SNGL ØØ4	1	N/A	N/A
SPMSG Ø12	4	N/A	N/A
SQRT ØØ8	1	N/A	N/A
STOP ØØ8	4	STTBSZ=n	To increase size of .DAT slot (DOS) or LUN (RSX) table to "n". Default size is 2Ø$_8$(DOS or 77$_8$(RSX).
TANH ØØ4	1	N/A	N/A
TIME ØØØ	6	N/A	N/A
TIME1Ø ØØØ	6	N/A	N/A
UNIT ØØ1	6	N/A	N/A

NOTES:

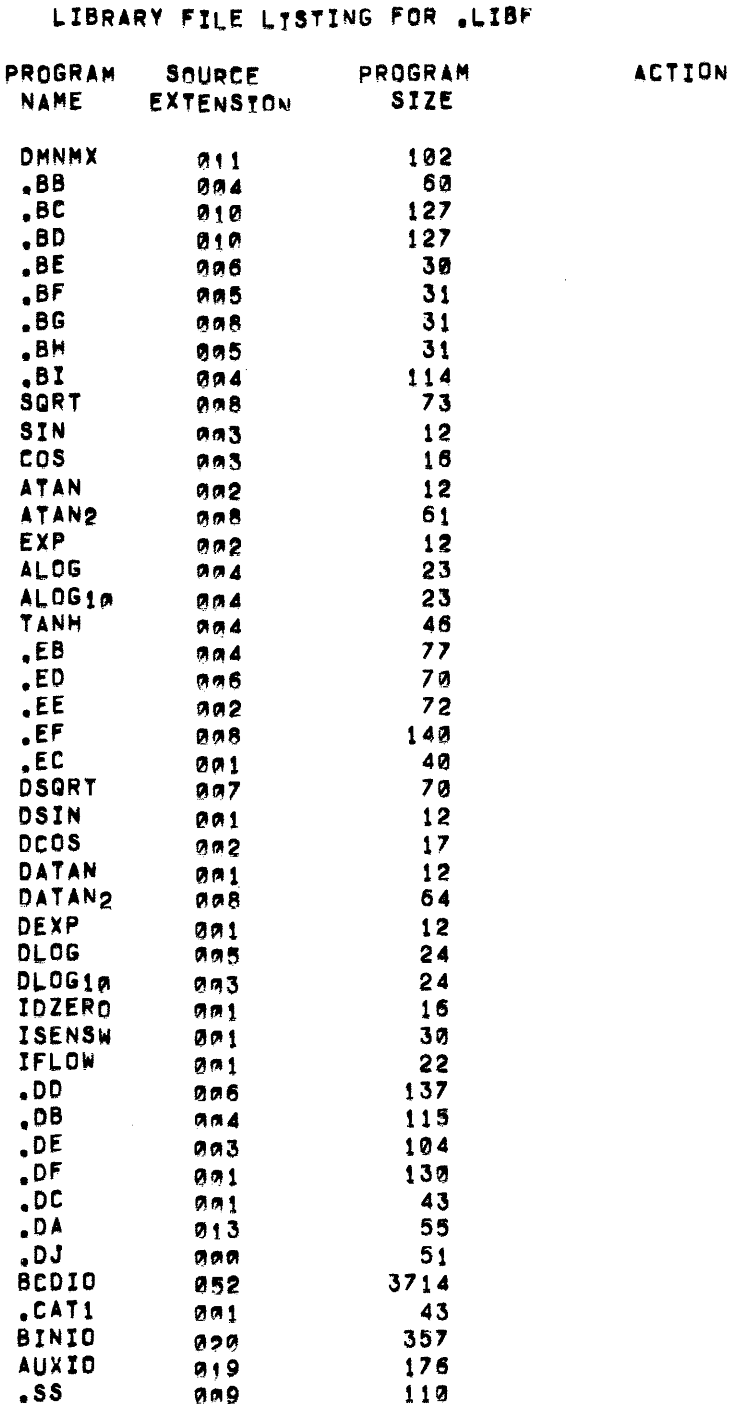
1. Used for both FP15 or non-FP15 systems. For FP15 systems, define %FPP=Ø.
2. Used only in FP15 systems. %FPP conditional is not used.
3. Used only in non-FP15 systems.
4. Used in both DOS and RSX. For RSX, define RSX=Ø.
5. Used only in RSX. The RSX conditional is not used.
6. Used only in DOS.



```
   LIBRARY FILE LISTING FOR .LIBF

PROGRAM   SOURCE      PROGRAM            ACTION
 NAME    EXTENSION     SIZE

 FPP        DOS           1
 24NV75     RKB           1
 BOSTT      001          16
 RBCDIO     009         141
 RBINIO     006         113
 RANCOM     014         504
 DEFINE     020        1126
 DDIO       019        2016
 EDCODE     003         253
 EOF        000          30
 UNIT       001          66
 JABS       001          14
 JDFIX      001          12
 JFIX       001          12
 FLOATJ     001          10
 JDBLE      001          10
 ISNGL      002          13
 JSIGN      004          16
 JDIM       001          17
 JMOD       003          17
 JMNMX      005         100
 ERRSET     000          25
 IOERR      002          40
 FILE       010         366
 TIME       000          52
 TIME10     000          70
 ABS        002          13
 IABS       000          14
 DABS       001          13
 AINT       002          14
 INT        002          12
 IDINT      005          12
 AMOD       003          23
 MOD        000          24
 DMOD       004          23
 FLOAT      002          11
 IFIX       002          12
 SIGN       004          24
 DSIGN      004          24
 ISIGN      000          20
 DIM        002          17
 IDIM       000          15
 SNGL       004          16
 DBLE       001          10
 IMNMX      009         106
 RMNMX      014         115
```

Figure E-1

UPDATE Listing for .LIBF of DOS Libraries

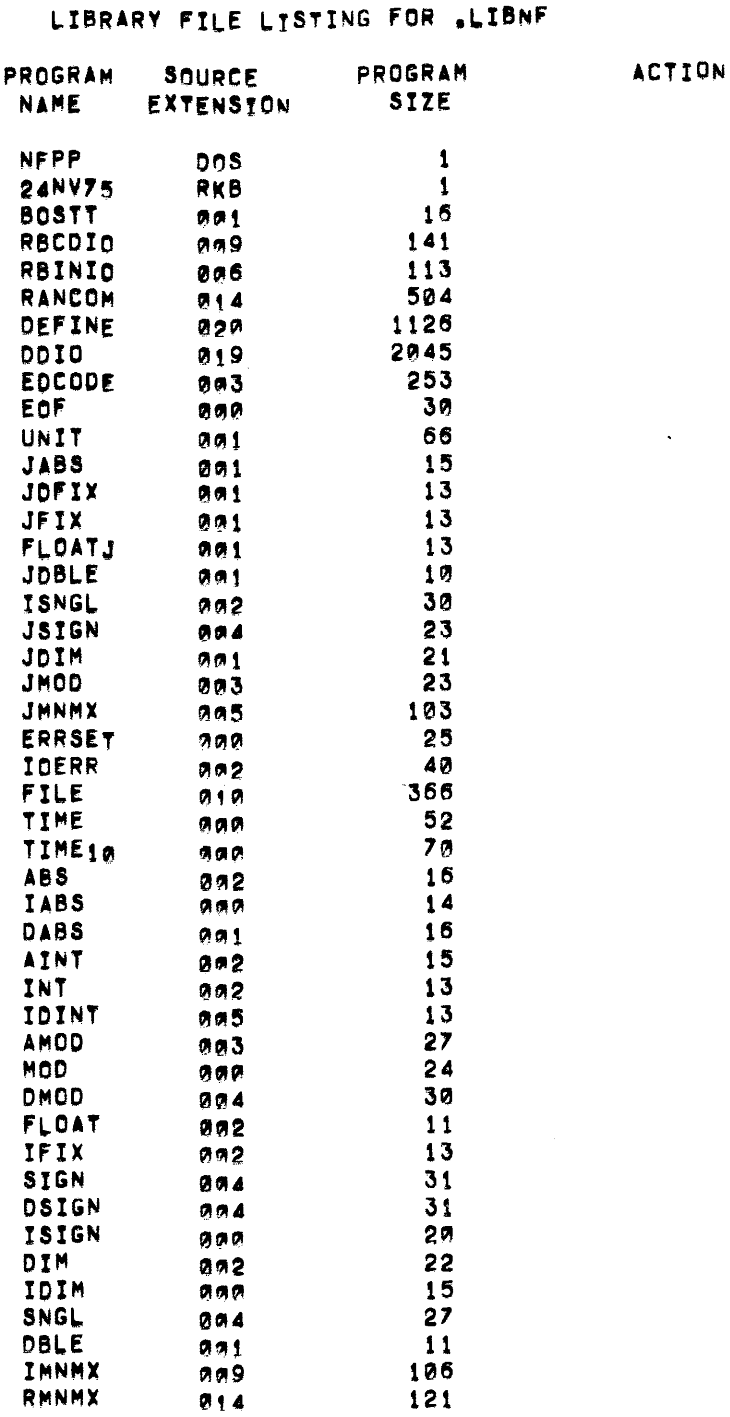


```
   LIBRARY FILE LISTING FOR .LIBF

PROGRAM   SOURCE       PROGRAM           ACTION
 NAME    EXTENSION      SIZE

 DMNMX      011          102
 .BB        004           60
 .BC        010          127
 .BD        010          127
 .BE        006           30
 .BF        005           31
 .BG        008           31
 .BH        005           31
 .BI        004          114
 SQRT       008           73
 SIN        003           12
 COS        003           16
 ATAN       002           12
 ATAN2      008           61
 EXP        002           12
 ALOG       004           23
 ALOG10     004           23
 TANH       004           46
 .EB        004           77
 .ED        006           70
 .EE        002           72
 .EF        008          140
 .EC        001           40
 DSQRT      007           70
 DSIN       001           12
 DCOS       002           17
 DATAN      001           12
 DATAN2     008           64
 DEXP       001           12
 DLOG       005           24
 DLOG10     003           24
 IDZERO     001           16
 ISENSW     001           30
 IFLOW      001           22
 .DD        006          137
 .DB        004          115
 .DE        003          104
 .DF        001          130
 .DC        001           43
 .DA        013           55
 .DJ        000           51
 BCDIO      052         3714
 .CAT1      001           43
 BINIO      020          357
 AUXIO      019          176
 .SS        009          110
```

Figure E-1
UPDATE Listing for .LIBF of DOS Libraries (cont)

```
   LIBRARY FILE LISTING FOR .LIBF

PROGRAM   SOURCE       PROGRAM            ACTION
 NAME    EXTENSION      SIZE

 GOTO       ØØ3           26
 STOP       ØØ8           61
 PAUSE      ØØ6           14
 SPMSG      Ø12          117
 .FLTB      ØØ4          266
 FIOPS      Ø42         1221
 PARTWD     ØØ3          146
 INTEAE     ØØ9          131
 .FPP       Ø18          44Ø
 OTSER      Ø15          213
 .CB        ØØ4           22              C
```

Figure E-1
UPDATE Listing for .LIBF of DOS Libraries (cont)



```
  LIBRARY FILE LISTING FOR .LIBNF

PROGRAM   SOURCE      PROGRAM               ACTION
 NAME    EXTENSION     SIZE

 NFPP       DOS           1
 24NV75     RKB           1
 BOSTT      ØØ1          16
 RBCDIO     ØØ9         141
 RBINIO     ØØ6         113
 RANCOM     Ø14         5Ø4
 DEFINE     Ø2Ø        1126
 DDIO       Ø19        2Ø45
 EDCODE     ØØ3         253
 EOF        ØØØ          3Ø
 UNIT       ØØ1          66
 JABS       ØØ1          15
 JDFIX      ØØ1          13
 JFIX       ØØ1          13
 FLOATJ     ØØ1          13
 JDBLE      ØØ1          1Ø
 ISNGL      ØØ2          3Ø
 JSIGN      ØØ4          23
 JDIM       ØØ1          21
 JMOD       ØØ3          23
 JMNMX      ØØ5         1Ø3
 ERRSET     ØØØ          25
 IOERR      ØØ2          4Ø
 FILE       Ø1Ø         366
 TIME       ØØØ          52
 TIME1Ø     ØØØ          7Ø
 ABS        ØØ2          16
 IABS       ØØØ          14
 DABS       ØØ1          16
 AINT       ØØ2          15
 INT        ØØ2          13
 IDINT      ØØ5          13
 AMOD       ØØ3          27
 MOD        ØØØ          24
 DMOD       ØØ4          3Ø
 FLOAT      ØØ2          11
 IFIX       ØØ2          13
 SIGN       ØØ4          31
 DSIGN      ØØ4          31
 ISIGN      ØØØ          2Ø
 DIM        ØØ2          22
 IDIM       ØØØ          15
 SNGL       ØØ4          27
 DBLE       ØØ1          11
 IMNMX      ØØ9         1Ø6
 RMNMX      Ø14         121
```

Figure E-2
UPDATE Listing for .LIBNF of DOS Libraries

```
LIBRARY FILE LISTING FOR .LIBNF

PROGRAM   SOURCE       PROGRAM          ACTION
NAME      EXTENSION     SIZE

DMNMX      Ø11          1Ø6
.BB        ØØ4           6Ø
.BC        Ø1Ø          133
.BD        Ø1Ø          133
.BE        ØØ6           33
.BF        ØØ5           34
.BG        ØØ8           35
.BH        ØØ5           34
.BI        ØØ4          121
SQRT       ØØ8           73
SIN        ØØ3           13
COS        ØØ3           2Ø
ATAN       ØØ2           13
ATAN2      ØØ8           7Ø
EXP        ØØ2           13
ALOG       ØØ4           25
ALOG1Ø     ØØ4           25
TANH       ØØ4           47
.EB        ØØ4          1Ø2
.ED        ØØ6           67
.EE        ØØ2           71
.EF        ØØ8          143
.EC        ØØ1           44
DSQRT      ØØ7           71
DSIN       ØØ1           13
DCOS       ØØ2           21
DATAN      ØØ1           13
DATAN2     ØØ8           73
DEXP       ØØ1           13
DLOG       ØØ5           26
DLOG1Ø     ØØ3           26
IDZERO     ØØ1           16
ISENSW     ØØ1           3Ø
IFLOW      ØØ1           22
.DD        ØØ6          146
.DB        ØØ4          12Ø
.DE        ØØ3          1Ø1
.DF        ØØ1          137
.DC        ØØ1           47
.DA        Ø13           55
.DJ        ØØØ           51
BCDIO      Ø52         4ØØ6
.CAT1      ØØ1           45
BINIO      Ø2Ø          357
AUXIO      Ø19          176
.SS        ØØ9          11Ø
```

Figure E-2
UPDATE Listing for .LIBNF of DOS Libraries (cont)

```
   LIBRARY FILE LISTING FOR .LIBNF

PROGRAM   SOURCE      PROGRAM           ACTION
 NAME    EXTENSION     SIZE

 GOTO       003         26
 STOP       008         61
 PAUSE      006         14
 SPMSG      012        117
 .FLTB      004        266
 FIOPS      042       1221
 PARTWD     003        140
 DBLINT     007        377
 INTEAE     009        131
 DOUBLE     004        203
 RELEAE     011       1077
 OTSER      015        213
 .CB        004         22              C
```

Figure E-2
UPDATE Listing for .LIBNF of DOS Libraries (cont)

INDEX

INDEX (Cont.)

XVM/DOS V1A System Installation Guide
DEC-XV-ODSIA-A-D

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. □

Please cut along this line.

---------- 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.