8TRAN XVM UTILITY MANUAL

DEC-XV-UTRNA-A-D

XVM Systems

digital

8TRAN XVM UTILITY MANUAL

DEC-XV-UTRNA-A-D

digital equipment corporation · maynard. massachusetts

First Printing, December 1975

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

CONTENTS

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

The operation and use of the XVM/DOS Utility Program 8TRAN are described in this manual.

It was assumed in the preparation of this manual that the reader is familiar with the operation of the PDP-8 and XVM computer systems and with their assembly languages.

CHAPTER 1
INTRODUCTION

8TRAN is a relocatable program which assists in the translation of PDP-8 programs to XVM programs. 8TRAN accepts source coding in PAL-III, PAL-D, or MACRO-8 assembly languages and produces source code in MACRO XVM (MACRO) assembly language. Its main functions are:

Reformatting of statements

Translation of mnemonics

Insertion of flags to indicate either that instructions have been translated or that translation is impossible.

NOTE

EAE coding and any input/output instructions other than those for the reader/punch and Teletype® must be modified. Floating point instructions are translated to calls to floating point subroutines.

Symbols used in this manual are defined as follows.

Symbol	Meaning
↵	Carriage RETURN
→|	Horizontal tab
␣	Space
[ ]	Optional command element
{ }	One of the enclosed command elements must be chosen.

® Teletype is a registered trademark of the Teletype Corporation.

CHAPTER 2
INSTRUCTION SETS

2.1 MEMORY REFERENCE INSTRUCTIONS

Table 2-1 shows the correspondence between the PDP-8 and XVM instruction sets.

Table 2-1
Correspondence Between Memory Reference Instructions

PDP-8		XVM	
Mnemonic	Octal	Mnemonic	Octal
AND	0000	AND	500000
TAD	1000	TAD	340000
ISZ	2000	ISZ	440000
JMS	4000	JMS	100000
JMP	5000	JMP	600000
DCA	3000	DAC	040000
		CLA	
		CAL	000000
		DZM	140000
		LAC	200000
		XOR	240000
		ADD	300000
		XCT	400000
		SAD	540000

2.2 INDIRECT REFERENCES AND AUTO-INDEX REGISTERS

Single level indirect addressing is identical on both machines.

	PDP-8		XVM
I	0400	*	020000

The Auto-Index Registers (10-17) also operate identically. However, the PDP-8 has one set of Auto-Index Registers for each 8K memory field, while the XVM has only one set of Auto-Index Registers in Page 0 of Bank 0.

2.3 OPERATE INSTRUCTIONS

A single group of Operate instructions in XVM corresponds to the two groups in PDP-8, as shown in Table 2-2.

Table 2-2
Correspondence Between Operate Instructions

PDP-8			XVM	
Group 1	NOP or OPR	7000	NOP or OPR	740000
	CLA	7200	CLA	750000
	CLL	7100	CLL	744000
	CMA	7040	CMA	740001
	CML	7020	CML	740002
	RAR	7010	RAR	740020
	RAL	7004	RAL	740010
	RTR	7012	RTR	472020
	RTL	7006	RTL	742010
	IAC	7001	IAC	740030
Group 2	CLA	7600	CLA	750000
	SMA	7500	SMA	740100
	SZA	7440	SZA	740200
	SNL	7420	SNL	740400
	SPA	7510	SPA	741100
	SNA	7450	SNA	741200
	SZL	7430	SZL	741400
	SKP	7410	SKP	741000
	OSR	7404	OAS	740004
	HLT	7402	HLT	740040

2.4 LAW INSTRUCTION

The LAW instruction in XVM has no equivalent in PDP-8. The mnemonic LAW N has an octal value of 760000 + N.

2.5 PAGING

The address portion of Memory Reference Instructions consists of 8 bits on the PDP-8 and 12 bits on the XVM, permitting direct addressing of 4K on the XVM.

Using indirect address references, the address size is 12 bits in PDP-8 and up to 17 bits in XVM, so that data and instruction fields are not required in XVM.

CHAPTER 3
ASSEMBLERS

The symbolic programs acceptable to PAL III/MACRO-8 and to MACRO XVM (MACRO) are similar in most respects. The important differences which do exist are discussed in this chapter.

3.1 FORMAT

The MACRO Assembler is field-oriented, which means that the interpretation of a statement depends on the field in which each element of the statement lies. There are four fields:

LABEL	(Field delimiter)
OPERATION	(Field delimiter)
ADDRESS	(Field delimiter)
COMMENT	(Statement delimiter)

A field delimiter is either a space or a tab. A statement delimiter is either a Carriage RETURN or a semicolon. In this document, tabs are indicated with the symbol →|, and carriage returns are indicated with ⤸.

In MACRO there is no field for an indirect reference because indirect addressing is indicated by an asterisk (*) immediately following the mnemonic operator in the operation field.

Examples:

```
              PDP-8          |          XVM
      TAG,           TAD A   |  TAG   →|TAD       →|A
                     ISZ I B |        →|ISZ*      →|B
                     JMP C   |        →|JMP       →|C
```

Tabs are normally preferred to spaces as field delimiters. MACRO does not require commas to terminate labels. Labels appearing on successive lines without any code are given consecutive addresses by MACRO, not the same address; an important difference from the PDP-8 assembler.

3.2 SYMBOLS

Symbols in MACRO may use period (.) and percent sign (%) in addition to letters and numbers. The initial character must be a letter, period (.), or percent sign (%).

3.3 EXPRESSION OPERATORS

MACRO has a more extensive set of operators than does PAL-III/MACRO-8. Addition (+), Subtraction (-), AND (&), and Inclusive OR (!) are supplemented by Exclusive OR (\), Multiplication (*), and Division (/).

3.4 NUMBERS

Octal and decimal numbers in the range $\pm 2^{18}-1$ for unsigned integers and $\pm 2^{17}-1$ for signed integers are available in MACRO. Double precision and floating-point constants (DUBL and FLTG pseudo-ops of MACRO-8) are not permitted in MACRO.

3.5 LOCATION COUNTER

The Location Counter may be referenced by period (.) in both systems. It is set as follows:

PDP-8	XVM
* n	.LOC n
PAGE n	Ignored Paging is redundant in XVM

The Location Counter is advanced as follows:

PDP-8	XVM
* . + n	.BLOCK n

The .BLOCK pseudo-op makes the operation more explicit in XVM.

3.6 TEXT HANDLING

Pseudo-ops to perform packing of 6-bit trimmed ASCII characters are available in both systems.

PDP-8	XVM
TEXT	.SIXBT

The 18-bit word in the XVM permits 3 characters per word, as compared with 2 characters per word in PDP-8.

An additional pseudo-op which performs the packing of five 7-bit ASCII characters in two words (.ASCII) is available in the XVM.

3.7 TERMINATING PSEUDO-OPS

The correspondence between the two systems is:

PDP-8	XVM
PAUSE	.EOT
$	.END

3.8 LITERALS

Page zero literals have no meaning in XVM. The correspondence is:

PDP-8	XVM
[	(
(	(
]	)
)	)

It should be noted that neither nested literals, e.g., TAD (TAD (20)), nor 8-bit ASCII characters, e.g., TAD ("A), are allowed in MACRO.

3.9 SYMBOL TABLE

The Symbol Table of MACRO cannot be deleted, hence EXPUNGE is not a valid pseudo-op.

Since the permanent Symbol Table of MACRO is searched for octal matching and not symbol matching the pseudo-ops FIXMRI and FIXTAB are redundant. A memory reference instruction is defined by a parameter assignment, e.g., IDX=ISZ, or IDX=440000.

3.10 USER MACROS

User Macros can be more elaborate in MACRO; the correspondence between the sets is:

PDP-8	XVM
DEFINE	.DEFIN
<	Not used
>	.ENDM

Example:

PDP-8	XVM	
DEFINE SUB A B	.DEFIN	SUB A,B
< CLA	LAC	B
TAD B	CMA	
CIA	TAD	(1
TAD A>	TAD	A
	.ENDM	

CHAPTER 4
INPUT/OUTPUT

The most important difference between the PDP-8 and the XVM in I/O lies in the fact than an extra 3 bits (12-14) are available in the XVM instruction word.

Bits 12 and 13 are used for subdevice selection, and Bit 14, when set, clears the AC at event time 1; i.e., prior to transfers to/from the AC.

4.1 FLAGS

The XVM has two instructions, not available on PDP-8, which read and clear flags.

IORS	Input/Output Read Status
CAF	Clear All Flags

4.2 INTERRUPT

The interrupt structure is identical on both machines. When the interrupt function is enabled (ION), the setting of a flag will cause an interrupt; i.e., an effective JMS to location 0 of field 0.

The XVM has an 18-bit word so that the full 15-bit address of the interrupt location is stored, eliminating the need for the 6-bit Interrupt Buffer in the PDP-8.

The remaining 3 bits of the XVM word are used to store the state of the Link, Bank/Page Mode, and Memory Protect.

XVM

L	BPM	MP	ADDRESS (15-Bits)
0	1	2	3-17

4.3 TELETYPE

There are two differences in the IOT instructions between PDP-8 and XVM.

KCC does not exist on the XVM. The Keyboard Flag is cleared by issuing a KRB, which also reads the buffer. All other Teletype IOTs are identical.

PDP-8	XVM
KCC	None (use KRB)
KRS	None (use KRB)
---	KRS (Keyboard Reader Select)

4.4 READER/PUNCH

All reader/punch functions available on the PDP-8 are also available on the XVM. The PDP-8 operates in Alphanumeric mode only, while the XVM operates in Binary mode as well as Alphanumeric mode. (Binary permits the reading of an 18-bit word from 3 lines of tape and the punching of a single line in the format of the binary read.)

PDP-8	XVM
RSF	RSF
RRB	RRB
RFC	RSA
--	RSB
PLS	PSA
--	PSB
PCF	PCF
PSF	PSF

4.5 DECTAPE

The DECtape controllers, TC01 on PDP-8 and TC02 on XVM, work in an identical manner.

The standard format of data on the tape differs significantly (see Table 4-1). There is, however, no incompatibility of control word format; therefore, a standard block of 129 words written by the PDP-8 can be read as a block of 86 words by the XVM.

Table 4-1
Tape Standard Data Format

	PDP-8	XVM
Words per Block	129_{10} (12 bits)	256_{10} (18 bits)
Blocks	$0\text{-}2701_8$	$0\text{-}1101_8$
Address of Word Count	7754	30
Address of Current Address	7755	31

The instructions to the two controllers differ in only one respect. The single IOT DTSF (skip if error flag or DECtape control flag=1) in the TC01 is replaced by the two IOTs, DTDF (skip if DECtape control flag=1) and DTEF (skip if DECtape error flag=1), on the TC02.

The two Status Registers, A and B, are 12 bits on both machines (ACØ-AC11 on the XVM).

4.6 IOTs IN PERMANENT SYMBOL TABLE

MACRO assumes that input/output will be handled by I/O System Macros using device handlers. Device-dependent IOTs are therefore not included in the Symbol Table. The P (parameter) option in MACRO provides the facility for adding to the Symbol Table prior to an assembly; IOTs may always be assigned in this manner if required.

CHAPTER 5
EXTENDED ARITHMETIC ELEMENT

Any EAE coding for the PDP-8 should be rewritten for the XVM, since:

a. The word length on the two machines is different.

b. The XVM EAE is considerably more powerful than its PDP-8 counterpart.

Table 5-1 indicates the correspondence between the two instruction sets.

Table 5-1
Correspondence Between Instruction Sets

PDP-8	XVM
DVI	DVI
NMI	NORM
SHL N	LLS+N+1
ASR N	LRSS+N+1
LSR N	LRS+N+1
MQL	LMQ
MUY	MUL
MQA	OMQ
CAM	CLQ
SCA	OSC

5.1 MULTIPLICATION AND DIVISION

Signed operations (MULS and DIVS) are available on XVM. The full number of shifts is always performed in multiplication and division on PDP-8. The number of shifts in XVM is programmable (contained in bits 12-17 of the instruction word) to reduce execution time where numbers are less than 18 bits in magnitude.

In division, the high order part only or the lower order part only of the dividend may be used in the operation as alternatives to the usual double word operation.

(IDIV, IDIVS, FRDIV, FRDIVS)

5.2 SHIFTING

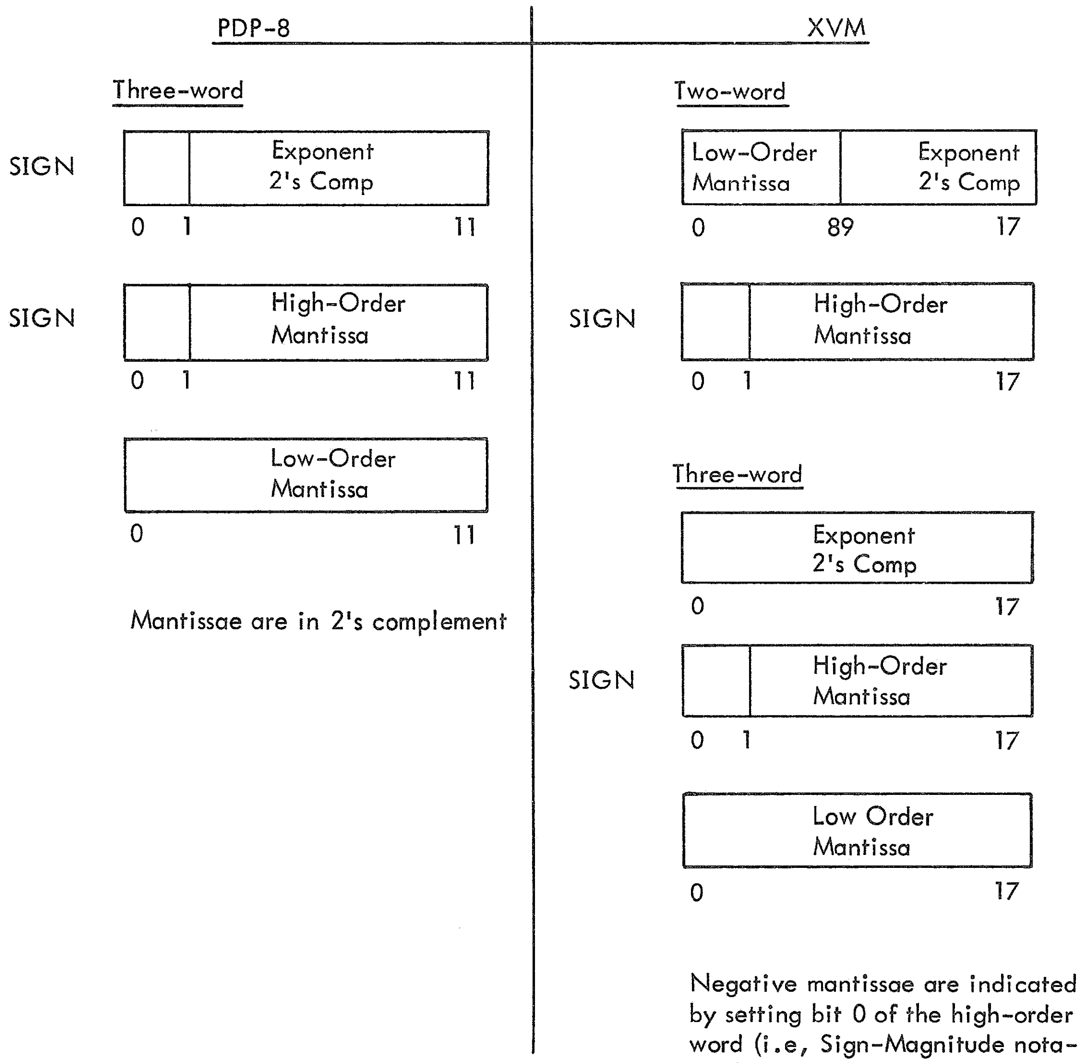
Three additional shifting operations are available on XVM. They are:

a. Signed left shift (LLSS)
b. Accumulator left shift (ALS)
c. Signed accumulator left shift (ALSS).

CHAPTER 6
FLOATING-POINT ARITHMETIC

Both the format and the treatment of floating-point numbers are different on the two machines.

6.1 FORMAT



Negative mantissae are indicated by setting bit 0 of the high-order word (i.e, Sign-Magnitude notation)

6.2 ARITHMETIC PACKAGE

The PDP-8 performs floating-point arithmetic by use of an interpreter. 8TRAN translates the PDP-8 floating-point instructions into subroutine calls with the address of the argument in the location following the JMS. The user may either provide the necessary floating-point arithmetic subroutines, or translate the subroutine calls to hardware FPP instructions if FPP hardware is available. Indirect addresses are indicated by setting bit 0 of the word following the JMS (i.e., XCT).

The correspondence between the two systems is shown in Table 6-1.

Table 6-1
Correspondence Between Arithmetic Packages

PDP-8	XVM
JMS 17 (Enter Interpreter)	None (not interpretive)
FADD A	JMS →|FAD A
FSUB I B	JMS →|FSUB XCT →|B
FMPY	JMS →|FMPY
FDIV	JMS →|FDVD
FGET	JMS →|FLAC
FPUT	JMS →|FDAC
FNOR	JMS →|%FNOR
FEXT	None (not interpreted)

6.3 INPUT/OUTPUT

On the PDP-8, numbers are input to and output from the floating accumulator.

8TRAN translates the PDP-8 I/O calls to subroutine calls followed by an argument. On input, bit 0 of the argument indicates the input device (other bits are ignored). On output, bit 0 indicates the output device, the other bits indicate the number of digits in the mantissa output.[1]

[1] COMPACT-15 used bit 0 to indicate the input or the output device. The important thing to note is that PDP-8 I/O requests are translated into subroutine calls.

PDP-8	XVM
JMS I 5	JMS →|FLIP 0
JMS I 6	JMS →|FLOP 6

Floating-point constants must be changed, and where FLTG is used in MACRO-8 the appropriate octal numbers must be inserted.

Examples:

Decimal	PDP-8	XVM
0.1	7775 3146 3146	460775 314631
-7.0	0003 4400 0000	000003 740000
π	0002 3110 3755	550002 311037
5.0	FLTG 5.0	000003 240000

CHAPTER 7
TRANSLATOR FUNCTIONS

The Translator program accepts a symbolic source tape (or file) written in either PAL III, PAL D, or MACRO-8, and translates it to MACRO assembly language, within the limits described below.

7.1 FORMATS

Each statement is reformatted in MACRO format. The Translator inserts a tab after a label field and after the operation field, so that the MACRO listing appears in columns as shown in the examples below.

PDP-8	XVM		
TAG,CLA CMA	TAG	CLA!CMA	
TAD I Z AUTO		TAD*	AUTO

[The optional page zero indicator (Z) is ignored.]

7.2 TRANSLATION

The following translations are performed. (Mnemonics processed by the Translator, and their translation where appropriate, are listed in Appendix A.)

7.2.1 Memory Reference Instructions

PDP-8	XVM	
DCA LOC	DAC	→| LOC
	CLA	

Where DCA is followed by TAD, the CLA in the above translation may sometimes be eliminated. The Translator treats the situation in one of three ways:

Translator Functions

PDP-8	XVM		
DCA LOC TAD LOC		DAC	LOC
DCA LOC1 TAD LOC2		DAC LAC	LOC1 LOC2
DCA LOC TAG,TAD LOC	 TAG	DAC CLA TAD	LOC LOC

In the last example above, the CLA is not eliminated because the TAD is labeled.

Where successive DCAs are translated, DZM replaces each DCA after the first, terminating the last DZM with a CLA. If a DCA is tagged, however, DZM is not generated.

PDP-8	XVM		
DCA LOC1		DAC	LOC1
DCA LOC2		DZM	LOC2
DCA LOC3		DZM	LOC3
TAG,DCA LOC4		CLA	
JMP LOC5	TAG	DAC	LOC4
		CLA	
		JMP	LOC5

All other MRIs are transferred without change.

7.2.2 Input/Output Instructions

PDP-8	XVM
RFC	RSA
PLS	PSA

All other Teletype and reader/punch instructions are transferred without change. IOT instructions for other devices are declared undefined symbols (see Section 7.3.1.4).

7.2.3 Operate Instructions

PDP-8	XVM
IAC	IAC
CLA IAC	CLA!IAC
CIA	CMA!IAC
CLL RTL	CLL RTL
OSR	OAS
CLA OSR	LAS

Double-rotate instructions are separated from other instructions because of conflicting event times. Single-rotate instructions are also separated when they conflict.

7.2.4 Pseudo-Operations

7.2.4.1 Miscellaneous

PDP-8	XVM
DECIMAL	.DEC
OCTAL	.OCT
PAUSE	.EOT
$	.END

The PDP-8 pseudo-ops PAGE and * for setting the location counter are not translated.

7.2.4.2 Text Handling

PDP-8	XVM
TEXT	.SIXBT

Both .SIXBT in MACRO and TEXT in PDP-8 assemblers treat text strings as trimmed ASCII. MACRO stores three characters per 18-bit word.

7.2.4.3 Macro Defining - The left angle bracket (<) used in MACRO-8 is ignored. However, instructions within a macro are translated.

Translator Functions

PDP-8	XVM
DEFINE	.DEFIN
<	ignored
>	.ENDM

7.2.5 Literals

PDP-8	XVM
[	(
]	)
("N	(nnn

Page 0 literals are translated as ordinary literals. Instructions within literals are translated. ASCII characters are translated into their 7-bit octal equivalent.

7.2.6 Floating Point

PDP-8	XVM
FEXT	Ignored
FADD	JMS →| FAD
FSUB	JMS →| FSUB
FMPY	JMS →| FMPY
FDIV	JMS →| FDVD
FGET	JMS →| FLAC
FPUT	JMS →| FDAC
FNOR	JMS →| %FNOR
I	XCT
JMS I 5	JMS →| FLIP 0
JMS I 6	JMS →| FLOP 6
JMS I 7	Ignored

The calls to the Input/Output routines (JMS I 5 and JMS I 6) and the entry to the interpreter (JMS I 7) are frequently given other names by parameter assignment. Where these instructions occur in a parameter assignment, the following translations occur.

Translator Functions

PDP-8	XVM
JMS I 5	JMS →| FLIP
JMS I 6	JMS →| FLOP
JMS I 7	NOP

The arguments of JMS FLIP and JMS FLOP are dropped and any entry to the interpreter is effectively ignored. However, the input and output calls must be followed by arguments, so insertion would have to be made at appropriate points in the program.

Floating-point variable storage can be reduced by one location for each variable, but no program error will occur if this is not done.

7.3 FLAGS

Flags are inserted to signal translations (or the absence of a translation) which may result in incorrect operation when assembling on the XVM. A flag occurs as a comment after the relevant instruction, starting on a new line, and is always preceded by

/**-

followed by a brief message, as defined below and summarized in Appendix B. There are two types of flags, optional and mandatory.

7.3.1 Optional Flags

Optional flags may be suppressed by the user because the appearance of these flags does not necessarily mean that changes have to be made to the program. (The method of suppressing optional flags is explained in Chapter 8.) The Translator may insert optional flags for the conditions described below.

7.3.1.1 Additional Code - When the Translator generates extra lines of code, a flag may be raised. The two situations are:

DCA	A DCA instruction was translated into DAC followed by CLA.
SMI	(Segmented Micro-Instruction) - A segmented microinstruction was encountered which could be translated only by splitting it into two instructions.

7.3.1.2 Relative Addresses - Since extra lines of code are generated, any relative address which occurs in the program may be in error.

REL	An address followed by ±n (where n is a number) was encountered.

7.3.1.3 Rotate Instructions - Since the word length of the two machines is different, the use of rotate instructions may result in incorrect operation.

ROT	A single or double rotate instruction was encountered.

7.3.1.4 Undefined Symbols - Since the Translator contains tables of pseudo-operations, Memory Reference, Operate, and Teletype and reader/punch I/O instructions only, any other symbol encountered in the operation field is undefined. Provision is therefore included to read a User Symbol Table prior to translation, so that a symbol is not declared undefined until a search of all user symbols has been made.

US	An undefined symbol was found in the operation field.

NOTE

1) If a User Symbol Table is to be retained for a number of tapes, each tape except the last must be terminated by a PAUSE pseudo-op.
2) If a User Symbol Table is not read prior to translation, it will be built during translation; hence only forward references in the operation field will be flagged.

7.3.2 Mandatory Flags

The flags in this group are always printed for a reason (see sections 7.3.2.1 through 7.3.2.6).

7.3.2.1 Location Counter Settings - The setting of the location counter by PAGE and * is ignored, and the code is simply printed as a comment.

LOC	A location counter setting was ignored.

NOTE

Each translation is preceded by .ABS and .LOC 100.

7.3.2.2 Illegal Characters - When an illegal character is encountered, its octal value in 7-bit ASCII is output. The character is ignored.

IC nnn	Illegal character.

7.3.2.3 Literals - Instructions within literals are translated only if the translation comprises a single line of code. Nested literals are illegal in MACRO.

LIT	A literal was nested, or, if translated, would have produced extra code.

7.3.2.4 Multiple Precision Constants - In XVM, the representation of double precision integers (DUBL) and floating-point constants (FLTG) is different from their representation in PDP-8. They are not translated.

MPC	A DUBL or FLTG pseudo-operation was encountered and ignored.

7.3.2.5 Skip Instructions - If the instruction following a Skip instruction is segmented, the Skip will cause an error at execution:

SKP	A segmented microinstruction followed a Skip instruction.

7.3.2.6 Symbol Table Overflow - The symbol table capacity is all of the unused core between .SCOM+2 and .SCOM+3. If the symbol table is filled, no further symbols will be stored and the following message is given:

SE	Symbol Table Exceeded.

CHAPTER 8
OPERATION

Appendix C provides a typical example of 8TRAN operation in the XVM/DOS environment.

8.1 .DAT SLOT ASSIGNMENTS

The following .DAT slots are used:

-15	Input
-14	Output
-3	Messages
-2	Command String

NOTE

8TRAN uses the system Macro .FSTAT. Any file-oriented device handler used must recognize .FSTAT, otherwise an IOPS6 error will occur.

8.2 CALLING PROCEDURE

After the Monitor's $, call 8TRAN by typing 8TRAN followed by a Carriage RETURN.

When loaded,

```
8TRAN XVM Vnxnnn              (where Vnxnnn is the current version)
>
```

is printed on the teleprinter and the program waits for your command string.

8.3 COMMAND STRING

The command string has the format : options␣filename␣ext␣terminator

Options	filename␣ext	terminator
S = Read a symbol table prior to translation	Name of symbol table or file to be translated. Default for extension (ext) is SRC.	ALT MODE = Return to Monitor after current operation
F = Suppress output of optional flags. When this option is selected, 8TRAN types: FLAGS-		↵ = Restart 8TRAN after current operation

Any or all of the Optional Flags (see Table 8-1) are printed in any order separated by a space or comma and terminated by a Carriage RETURN.

Table 8-1
Optional and Mandatory Flags

OPTIONAL	
Flag	Meaning
DCA	A DCA instruction was translated.
IC nnn	An illegal character was encountered and ignored (nnn is ≠ the character's octal value in 7-bit ASCII).
REL	A relative address was encountered.
ROT	A rotate microinstruction was encountered.
SMI	A segmented microinstruction was encountered.
MANDATORY	
Flag	Meaning
LIT	A literal either was nested or, if translated, would have produced extra code.
LOC	A location counter setting was ignored.
MPC	A DUBL or FLTG pseudo-op was found and ignored.
SE	The symbol table was exceeded.
SKP	An SMI followed a SKP instruction.

8.4 OPERATING PROCEDURE

The program to be translated and its symbol table, if required, must be ready on the appropriate device before the command string is typed. If the input is on a directoried device, both the symbol table and the program to be translated must be on the same disk UIC, DECtape, or magtape.

If the S option is selected, it must be followed by the name of the symbol table file. This file is the symbol table output from either PAL III or MACRO-8.

Example:

```
>S←SYMTAB↵
```

After the symbol table has been read, 8TRAN prints:

```
SYMBOL TABLE READ
>
```

and waits for the name of the file to be translated.

Example:

```
>←PRGNAM↵
```

If a PAUSE pseudo-op is encountered, the symbol table is preserved for the translation of other programs. 8TRAN prints:

```
PAUSE
>
```

and waits for another command string.

8.5 ERROR CONDITIONS

Table 8-2 provides the error conditions which may occur and their meanings.

Table 8-2
Error Conditions and Their Meanings

Error Condition	Meaning
??	Bad command string - retype
INPUT FILE NOT ON DEVICE	8TRAN cannot find the file named in the command string - retype
OUTPUT FILE ALREADY ON DEVICE TYPE ↑P TO RESTART OR CR TO OVERWRITE	The program named in the command string (with SRC extension) is already on the output device
IOPS 4	I/O device not ready - type CTRL R when ready.
IOPS 0-77	See Appendix C of the XVM/DOS Keyboard Command Guide

Examples:

(User responses are underlined.)

a. To translate a program and symbol table on paper tape:

```
8TRAN XVM Vnxnnn
S←↵
SYMBOL TABLE READ
>←↵
```

b. To translate a program with no symbol table and the ROT and SMI options suppressed using directoried I/O:

```
8TRAN XVM Vnxnnn
>F←PRGNAM PAL↵
FLAGS-ROT,SMI↵
```

APPENDIX A
SYMBOL TRANSLATIONS

PDP-8	XVM
Memory Reference	
AND	AND
TAD	TAD
ISZ	ISZ
DCA	(DAC (CLA
JMS	JMS
JMP	JMP
I	*
Z	Ignored
Operate	
NOP	NOP
OPR	OPR
IAC	IAC
RAL	RAL
RTL	RTL
RAR	RAR
RTR	RTR
CML	CML
CMA	CMA
CLL	CLL
CLA	CLA
HLT	HLT
OSR	OAS
SKP	SKP
SNL	SNL

PDP-8	XVM
Operate (Cont)	
SZL	SZL
SZA	SZA
SNA	SNA
SMA	SMA
SPA	SPA
Combined Operate	
CIA	CMA!IAC
STL	STL
GLK	GLK
STA	CLC
LAS	LAS
Input/Output	
IOF	IOF
ION	ION
IOT	IOT
KSF	KSF
KRB	KRB
TCF	TCF
TSF	TSF
TLS	TLS
RFC	RSA
RSF	RSF
RRB	RRB
PLS	PSA

Symbol Translations

PDP-8	XVM
Input/Output (Cont)	
PCF	PCF
PSF	PSF
Floating Point	
FEXT	Ignored
FADD	JMS FAD
FSUB	JMS FSUB
FMPY	JMS FMPY
FDIV	JMS FDVD
FGET	JMS FLAC
FPUT	JMS FDAC
FNOR	JMS%FNOR
I	XCT
JMS I 5	JMS FLIP 0
JMS I 6	JMS FLOP 6
JMS I 7	Ignored

PDP-8	XVM
Pseudo Operators	
PAGE FIELD *	Ignored
DECIMAL	.DEC
OCTAL	.OCT
DUBL FLTG	Ignored
TEXT	.SIXBT
PAUSE	.EOT
$	.END
DEFINE	.DEFIN
<	Ignored
>	.ENDM
EXPUNGE FIXMRI FIXTAB	Ignored

APPENDIX B
FLAGS

Group	Descriptor	Meaning
Optional	DCA	A DCA instruction was translated.
	REL	A relative address was encountered.
	ROT	A rotate microinstruction was encountered.
	SMI	Segmented microinstruction.
	US	An undefined symbol occurred in the operation field.
Mandatory	IC nnn	An illegal character was encountered and ignored.
	LIT	A literal was nested, or, if translated, would have produced extra code.
	LOC	A location counter setting was ignored.
	MPC	A DUBL or FLTG pseudo-operation was encountered and ignored.
	SE	Symbol Table Exceeded.
	SKP	An SMI followed a Skip instruction.

APPENDIX C
8TRAN DEMONSTRATION

This appendix is a demonstration of 8TRAN operation in the XVM/DOS environment. The listing in the left column is the MACRO-8 source program to be translated. (For the convenience of the reader, additional spaces have been inserted in the body of the listing to align the code.) The column on the right of the page is an 8TRAN output listing showing the results of translation.

NOTE

Summaries of symbol translations and diagnostic flags are provided in Appendices A and B respectively.

MACRO-8 Program

```
/DIGIT OCTAL SQUARE CONVERSATIONAL
/PROGRAM
*200
START,          CLA  CLL
                TLS
                JMS          CRLF
                JMS          LISN
                TAD          M260
                RAL  CLL
                RTL
                DCA          NUMBER
                JMS          LISN
                TAD          M260
                TAD          NUMBER
                DCA          NUMBER
```

Translation

```
                .ABS
                .LOC        100
/DIGIT OCTAL SQUARE CONVERSATIONAL
/PROGRAM
/** -LOC       *200
START           CLA!CLL
                TLS
                JMS         CRLF
                JMS         LISN
                TAD         M260
                CLL!RAL
/** -ROT
                RTL
/** -ROT
                DAC         NUMBER
/** -DCA
                CLA
                JMS         LISN
                TAD         M260
                TAD         NUMBER
                DAC         NUMBER
/** -DCA
                CLA
```

8TRAN Demonstration

MACRO-8 Program

```
MULT,     TAD       NUMBER
          CIA
          DCA       TALLY
          TAD       NUMBER
          ISZ       TALLY
          JMP       .-2
          DCA       NUMSQR


TYPSQU,   TAD       MESAG1
          DCA       POINTR
          TAD       M1Ø
          DCA       FNDCHK
          JMS       MESAGE


TYPANS,   TAD       M4
          DCA       DIGCTR
          DCA       STORE
          TAD       NUMSQR
          CLL  RAL

UNPACK,   TAD       STORE
          RAL
          RTL
          DCA       STORE
          TAD       STORE
          AND       K7
          TAD       K26Ø
          JMS       TYPE
          ISZ       DIGCTR
          JMP       UNPACK

TYPOCT,   TAD       MESAG2
          DCA       POINTR
          TAD       M7
          DCA       ENDCHK
          JMS       MESAGE
          JMS       CRLF
          JMP       START+2
```

Translation

```
MULT       TAD        NUMBER
           CMA!IAC
           DAC        TALLY
/** -DCA
           LAC        NUMBER
           ISZ        TALLY
           JMP        .-2
/** -REL
           DAC        NUMSQR
/** -DCA
           CLA
TYPSQU     TAD        MESAG1
           DAC        POINTR
/** -DCA
           LAC        M1Ø
           DAC        FNDCHK
/** -DCA
           CLA
           JMS        MESAGE
TYPANS     TAD        M4
           DAC        DIGCTR
/** -DCA
           DZM        STORE
           LAC        NUMSQR
           CLL!RAL
/** -ROT
UNPACK     TAD        STORE
           RAL
/** -ROT
           RTL
/** -ROT
           DAC        STORE
/** -DCA
           AND        K7
           TAD        K26Ø
           JMS        TYPE
           ISZ        DIGCTR
           JMP        UNPACK
TYPOCT     TAD        MESAG2
           DAC        POINTR
/** -DCA
           LAC        M7
           DAC        FNDCHK
/** -DCA
           CLA
           JMS        MESAGE
           JMS        CRLF
           JMP        START+2
/** -REL
```

8TRAN Demonstration

MACRO-8 Program

```
TYPE,    Ø
         TSF
         JMP        .-1
         TLS
         CLA
         JMS  I     TYPE
CRLF,    Ø
         TAD        K215
         JMS        TYPE
         TAD        K212
         JMS        TYPE
         JMP  I     CRLF
LISN,    Ø
         KSF
         JMP        .-1
         KRB
         TLS
         JMP  I     LISN

MESAGE,  Ø
         TAD  I     POINTR
         JMS        TYPE
         ISZ        POINTR
         ISZ        ENDCHK
         JMP        .-4
         JMP  I     MESAGE
NUMBER,  Ø
M26Ø,    -26Ø
TALLY,   Ø
NUMSQR,  Ø
MESAG1,  START1
POINTR,  Ø
M1Ø,     -1Ø
ENDCHK,  Ø
STORE,   Ø
M4,      -4
DIGCTR,  Ø
K7,      7
M7,      -7
K26Ø,    26Ø
K212,    212
K215,    215
MESAG2,  START2
```

Translation

```
TYPE      Ø
          TSF
          JMP       .-1
/** -REL
          TLS
          CLA
          JMS*      TYPE
CRLF      Ø
          TAD       K215
          JMS       TYPE
          TAD       K212
          JMS       TYPE
          JMP*      CRLF
LISN      Ø
          KSF
          JMP       .-1
/** -REL
          KRB
          TLS
          JMP*      LISN
MESAGE    Ø
          TAD*      POINTR
          JMS       TYPE
          ISZ       POINTR
          ISZ       ENDCHK
          JMP       .-4
/** -REL
          JMP*      MESAGE
NUMBER    Ø
M26Ø      -26Ø
TALLY     Ø
NUMSQR    Ø
MESAG1    START1
/** -US
POINTR    Ø
M1Ø       -1Ø
ENDCHK    Ø
STORE     Ø
M4        -4
DIGCTR    Ø
K7        7
M7        -7
K26Ø      26Ø
K212      212
K215      215
MESAG2    START2
/** -US
```

8TRAN Demonstration

MACRO-8 Program

```
START1,   323        /S
          321        /Q
          325        /U
          3Ø1        /A
          322        /R
          3Ø5        /E
          3Ø4        /D
          275        /=
START2,   24Ø        /SPACE
          317        /O
          3Ø3        /C
          324        /T
          3Ø1        /A
          314        /L
          256        /PERIOD
$
```

Translation

```
START1    323        /S
          321        /Q
          325        /U
          3Ø1        /A
          322        /R
          3Ø5        /E
          3Ø4        /D
          275        /=
START2    24Ø        /SPACE
          317        /O
          3Ø3        /C
          324        /T
          3Ø1        /A
          314        /L
          256        /PERIOD
          .END
```

INDEX

8TRAN XVM
Utility Manual
DEC-XV-UTRNA-A-D

READER'S COMMENTS

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

FIRST CLASS
PERMIT NO. 33
MAYNARD, MASS.

BUSINESS REPLY MAIL
NO POSTAGE STAMP NECESSARY IF MAILED IN THE UNITED STATES

Postage will be paid by:

digital

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

digital

digital equipment corporation

PRINTED IN U.S.A.