

**8TRAN XVM
UTILITY MANUAL**

DEC-XV-UTRNA-A-D



**XVM
Systems
digital**

**8TRAN XVM
UTILITY MANUAL**

DEC-XV-UTRNA-A-D

First Printing, December 1975

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

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.

Copyright (C) 1975 by Digital Equipment Corporation

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

CONTENTS

| | Page |
|--|------|
| PREFACE | vii |
| CHAPTER 1 INTRODUCTION | 1-1 |
| CHAPTER 2 INSTRUCTION SETS | 2-1 |
| 2.1 MEMORY REFERENCE INSTRUCTIONS | 2-1 |
| 2.2 INDIRECT REFERENCES AND AUTO-INDEX REGISTERS | 2-1 |
| 2.3 OPERATE INSTRUCTIONS | 2-2 |
| 2.4 LAW INSTRUCTION | 2-2 |
| 2.5 PAGING | 2-2 |
| CHAPTER 3 ASSEMBLERS | 3-1 |
| 3.1 FORMAT | 3-1 |
| 3.2 SYMBOLS | 3-2 |
| 3.3 EXPRESSION OPERATORS | 3-2 |
| 3.4 NUMBERS | 3-2 |
| 3.5 LOCATION COUNTER | 3-2 |
| 3.6 TEXT HANDLING | 3-2 |
| 3.7 TERMINATING PSEUDO-OPS | 3-3 |
| 3.8 LITERALS | 3-3 |
| 3.9 SYMBOL TABLE | 3-3 |
| 3.10 USER MACROS | 3-3 |
| CHAPTER 4 INPUT/OUTPUT | 4-1 |
| 4.1 FLAGS | 4-1 |
| 4.2 INTERRUPT | 4-1 |

| | | Page |
|-----------|--------------------------------|------|
| 4.3 | TELETYPE | 4-2 |
| 4.4 | READER/PUNCH | 4-2 |
| 4.5 | DECTAPE | 4-2 |
| 4.6 | IOTs IN PERMANENT SYMBOL TABLE | 4-3 |
| CHAPTER 5 | EXTENDED ARITHMETIC ELEMENT | 5-1 |
| 5.1 | MULTIPLICATION AND DIVISION | 5-1 |
| 5.2 | SHIFTING | 5-2 |
| CHAPTER 6 | FLOATING-POINT ARITHMETIC | 6-1 |
| 6.1 | FORMAT | 6-1 |
| 6.2 | ARITHMETIC PACKAGE | 6-2 |
| 6.3 | INPUT/OUTPUT | 6-2 |
| CHAPTER 7 | TRANSLATOR FUNCTIONS | 7-1 |
| 7.1 | FORMATS | 7-1 |
| 7.2 | TRANSLATION | 7-1 |
| 7.2.1 | Memory Reference Instructions | 7-1 |
| 7.2.2 | Input/Output Instructions | 7-2 |
| 7.2.3 | Operate Instructions | 7-3 |
| 7.2.4 | Pseudo-Operations | 7-3 |
| 7.2.4.1 | Miscellaneous | 7-3 |
| 7.2.4.2 | Text Handling | 7-3 |
| 7.2.4.3 | Macro Defining | 7-3 |
| 7.2.5 | Literals | 7-4 |
| 7.2.6 | Floating Point | 7-4 |
| 7.3 | FLAGS | 7-5 |
| 7.3.1 | Optional Flags | 7-5 |
| 7.3.1.1 | Additional Code | 7-5 |
| 7.3.1.2 | Relative Addresses | 7-5 |
| 7.3.1.3 | Rotate Instructions | 7-6 |
| 7.3.1.4 | Undefined Symbols | 7-6 |
| 7.3.2 | Mandatory Flags | 7-6 |
| 7.3.2.1 | Location Counter Settings | 7-6 |
| 7.3.2.2 | Illegal Characters | 7-6 |
| 7.3.2.3 | Literals | 7-7 |
| 7.3.2.4 | Multiple Precision Constants | 7-7 |
| 7.3.2.5 | Skip Instructions | 7-7 |
| 7.3.2.6 | Symbol Table Overflow | 7-7 |

| | | Page |
|------------|-----------------------|---------|
| CHAPTER 8 | OPERATION | 8-1 |
| 8.1 | .DAT SLOT ASSIGNMENTS | 8-1 |
| 8.2 | CALLING PROCEDURE | 8-1 |
| 8.3 | COMMAND STRING | 8-2 |
| 8.4 | OPERATING PROCEDURE | 8-3 |
| 8.5 | ERROR CONDITIONS | 8-3 |
| APPENDIX A | SYMBOL TRANSLATIONS | A-1 |
| APPENDIX B | FLAGS | B-1 |
| APPENDIX C | 8TRAN DEMONSTRATION | C-1 |
| INDEX | | Index-1 |

TABLES

| Number | | Page |
|--------|--|------|
| 2-1 | Correspondence Between Memory Reference Instructions | 2-1 |
| 2-2 | Correspondence Between Operate Instructions | 2-2 |
| 4-1 | Tape Standard Data Format | 4-3 |
| 5-1 | Correspondence Between Instruction Sets | 5-1 |
| 6-1 | Correspondence Between Arithmetic Packages | 6-2 |
| 8-1 | Optional and Mandatory Flags | 8-2 |
| 8-2 | Error Conditions and Their Meanings | 8-4 |

)

9

9

)

)

)

9

9

)

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 |
| 8STRAN 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 VIA 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 |

)

6

8

)

)

)

9

2

)

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.

)

6

8

)

)

)

4

11

)

CHAPTER I

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.

)

2

2

)

)

)

2

4

)

CHAPTER 2
INSTRUCTION SETS

2.1 MEMORY REFERENCE INSTRUCTIONS

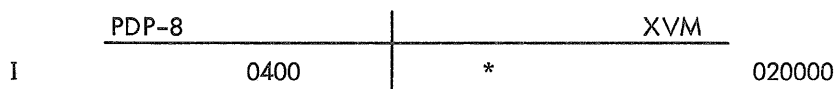
Table 2-1 shows the correspondence between the PDP-8 and XVM instruction sets.

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.



Instruction Sets

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 \rightarrow , and carriage returns are indicated with \downarrow .

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 \rightarrow TAD | \rightarrow A |
| | ISZ I B | | \rightarrow ISZ* | \rightarrow B |
| | JMP C | | \rightarrow JMP | \rightarrow 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.

Assemblers

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 |

Assemblers

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 |

Assemblers

Example:

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

CHAPTER 4 INPUT/OUTPUT

The most important difference between the PDP-8 and the XVM in I/O lies in the fact that 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.

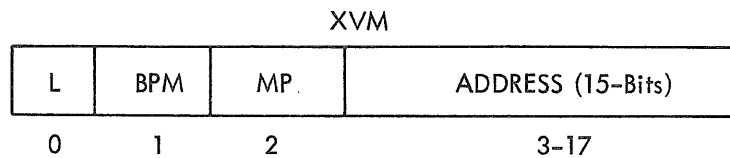
| | |
|------|--------------------------|
| IORS | Input/Output Read Status |
| CAF | Clear All Flags |

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.



Input/Output

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.

| <u>PDP-8</u> | <u>XVM</u> |
|--------------|------------------------------|
| 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.)

| <u>PDP-8</u> | <u>XVM</u> |
|--------------|------------|
| 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.

Input/Output

Table 4-1
Tape Standard Data Format

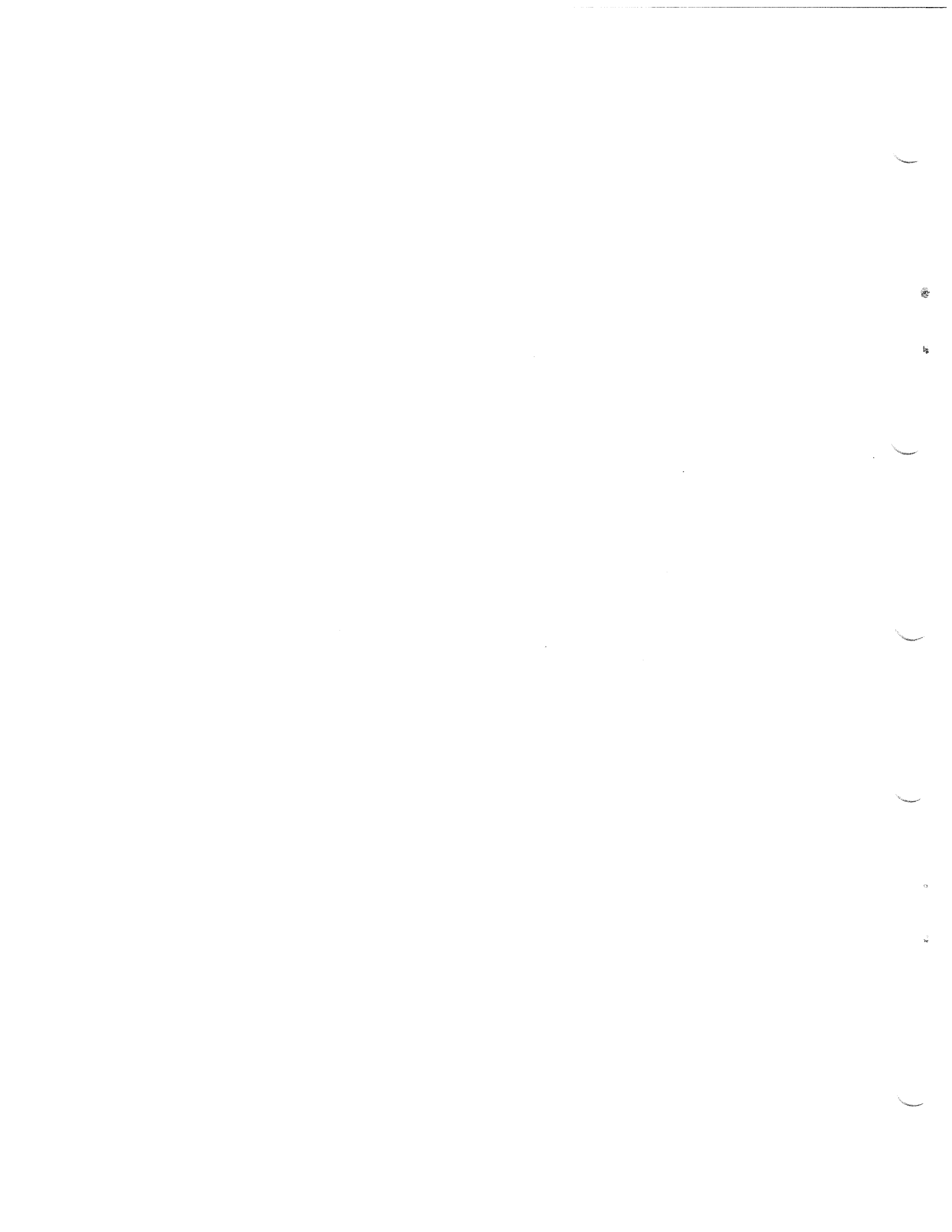
| | PDP-8 | XVM |
|-------------------------------|--------------------------------|--------------------------------|
| Words per Block | 129 ₁₀ (12 bits) | 256 ₁₀ (18 bits) |
| Blocks | 0-270 ₈ | 0-110 ₈ |
| Address of Word Count | 7754 | 30 |
| Address of Current Address | 7755 | 31 |

The instructions to the two controllers differ in only one respect. The single IOT DTSF (skip if error flag or DEC-tape 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 (AC0-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.

Extended Arithmetic Element

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

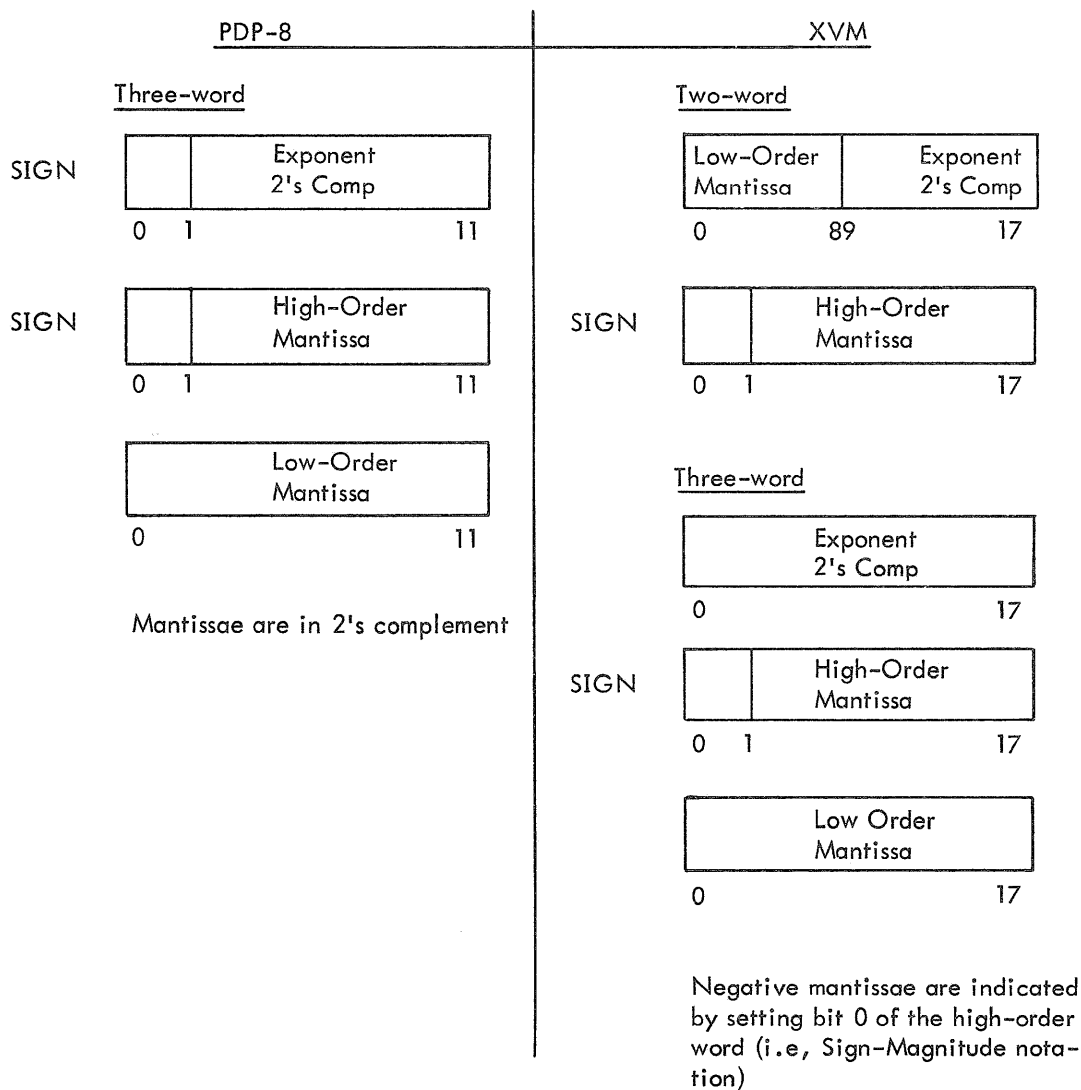
Three additional shifting operations are available on XVM. They are:

- 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



Floating-Point Arithmetic

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

¹COMPACT-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.

Floating-Point Arithmetic

| | |
|---------|-------------------|
| 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 |

)

8

9

)

)

)

8

9

)

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 CLA } → LOC |

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 | 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 | 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).

Translator Functions

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.

Translator Functions

REL An address followed by $\pm n$ (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.

Translator Functions

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.

)

8

9

)

)

)

10

11

)

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.

Operation

8.3 COMMAND STRING

The command string has the format : optionsfilenameextterminator

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

FLAGS-

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. |

Operation

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.

Operation

Table 8-2
Error Conditions and Their Meanings

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

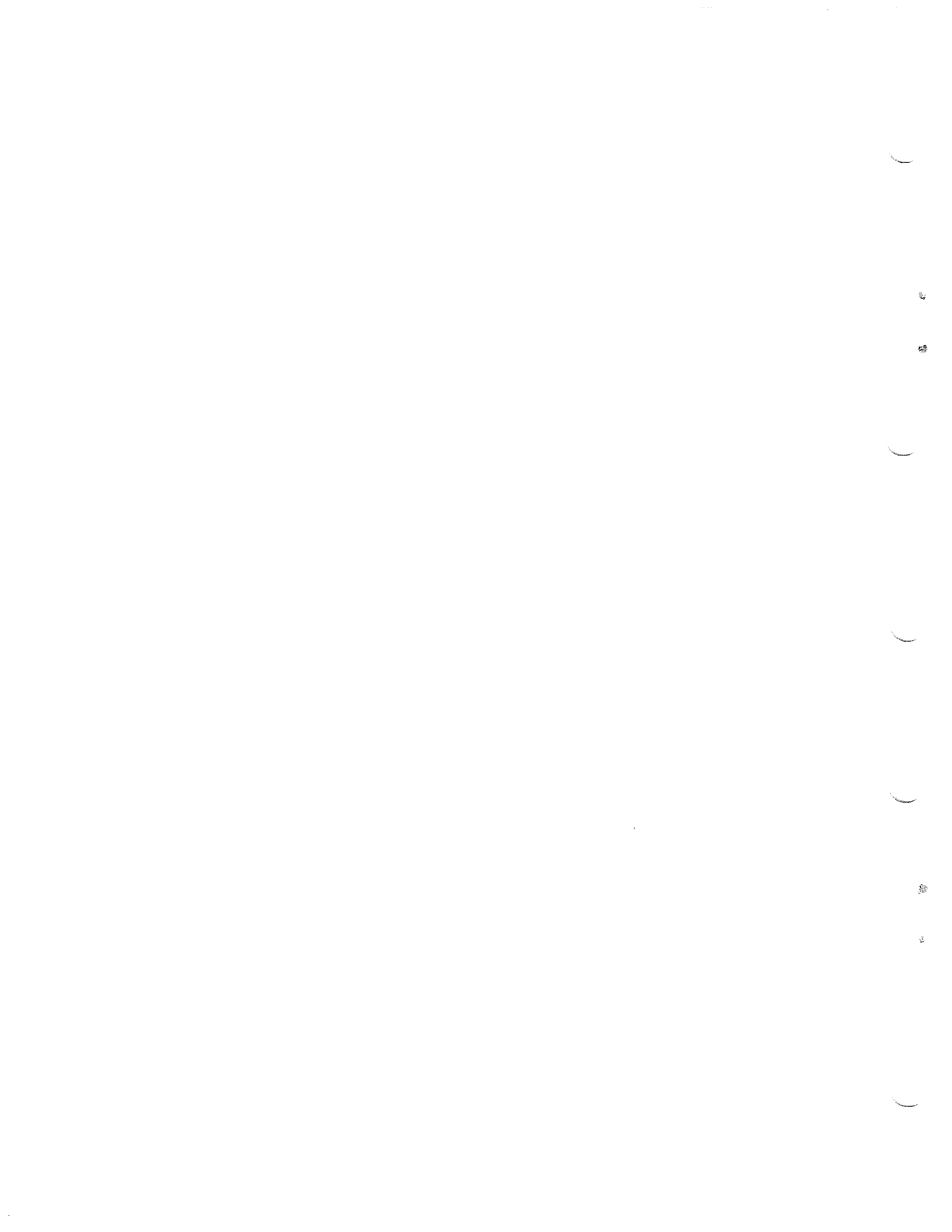
| <u>PDP-8</u> | <u>XVM</u> | <u>PDP-8</u> | <u>XVM</u> |
|--------------|-------------------------|--------------|-------------------------|
| | <u>Memory Reference</u> | | <u>Operate (Cont)</u> |
| AND | AND | SZL | SZL |
| TAD | TAD | SZA | SZA |
| ISZ | ISZ | SNA | SNA |
| DCA | (DAC (CLA | SMA | SMA |
| JMS | JMS | SPA | SPA |
| JMP | JMP | | |
| I | * | | <u>Combined Operate</u> |
| Z | Ignored | CIA | CMAIIAC |
| | <u>Operate</u> | STL | STL |
| NOP | NOP | GLK | GLK |
| OPR | OPR | STA | CLC |
| IAC | IAC | LAS | LAS |
| RAL | RAL | | <u>Input/Output</u> |
| RTL | RTL | IOF | IOF |
| RAR | RAR | ION | ION |
| RTR | RTR | IOT | IOT |
| CML | CML | KSF | KSF |
| CMA | CMA | KRB | KRB |
| CLL | CLL | TCF | TCF |
| CLA | CLA | TSF | TSF |
| HLT | HLT | TLS | TLS |
| OSR | OAS | RFC | RSA |
| SKP | SKP | RSF | RSF |
| SNL | SNL | RRB | RRB |
| | | PLS | PSA |

Symbol Translations

| <u>PDP-8</u> | <u>XVM</u> | <u>PDP-8</u> | <u>XVM</u> |
|----------------------------|-----------------------|-----------------------------------|------------|
| <u>Input/Output (Cont)</u> | | <u>Pseudo Operators</u> | |
| PCF | PCF | PAGE } FIELD } | Ignored |
| PSF | PSF | * | |
| | <u>Floating Point</u> | DECIMAL | .DEC |
| FEXT | Ignored | OCTAL | .OCT |
| FADD | JMS FAD | DUBL } FLTG } | Ignored |
| FSUB | JMS FSUB | TEXT | .SIXBT |
| FMPY | JMS FMPY | PAUSE | .EOT |
| FDIV | JMS FDVD | \$ | .END |
| FGET | JMS FLAC | DEFINE | .DEFIN |
| FPUT | JMS FDAC | < | Ignored |
| FNOR | JMS%FNOR | > | .ENDM |
| I | XCT | EXPUNGE } FIXMRI } FIXTAB } | Ignored |
| JMS I 5 | { JMS FLIP 0 | | |
| JMS I 6 | { JMS FLOP 6 | | |
| JMS I 7 | 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.

| <u>MACRO-8 Program</u> | <u>Translation</u> |
|---|--|
| <pre> /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 </pre> | <pre> .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 </pre> |

8TRAN Demonstration

| <u>MACRO-8 Program</u> | | | | <u>Translation</u> | | | |
|------------------------|---------|---------|---|--------------------|----------|---------|--|
| | | | | MULT | TAD | NUMBER | |
| | | | | | CMA: IAC | | |
| | | | | | DAC | TALLY | |
| MULT, | TAD | NUMBER | } | /** -DCA | LAC | NUMBER | |
| | CIA | | | | ISZ | TALLY | |
| | DCA | TALLY | | | JMP | .-2 | |
| | TAD | NUMBER | | /** -REL | | | |
| | ISZ | TALLY | | | DAC | NUMSQR | |
| | JMP | .-2 | | /** -DCA | | | |
| | DCA | NUMSQR | | | CLASQU | | |
| | | | | TAD | MESAG1 | | |
| | | | | DAC | POINTR | | |
| TYPESQU, | TAD | MESAG1 | } | /** -DCA | LAC | M10 | |
| | DCA | POINTR | | | DAC | FNDCHK | |
| | TAD | M10 | | /** -DCA | | | |
| | DCA | FNDCHK | | | CLASQU | | |
| | JMS | MESSAGE | | JMS | MESSAGE | | |
| | | | | TAD | M4 | | |
| | | | | DAC | DIGCTR | | |
| TYPANS, | TAD | M4 | } | /** -DCA | DZM | STORE | |
| | DCA | DIGCTR | | | LAC | NUMSQR | |
| | DCA | STORE | | | CLL:RAL | | |
| | TAD | NUMSQR | | /** -ROT | | | |
| | CLL RAL | | | UNPACK | TAD | STORE | |
| | | | | | RAL | | |
| UNPACK, | TAD | STORE | } | /** -ROT | RTL | | |
| | RAL | | | | /** -ROT | | |
| | RTL | | | | DAC | STORE | |
| | DCA | STORE | | /** -DCA | | | |
| | TAD | STORE | | | AND | K7 | |
| | AND | K7 | | | TAD | K260 | |
| | TAD | K260 | | | JMS | TYPE | |
| | JMS | TYPE | | | ISZ | DIGCTR | |
| | ISZ | DIGCTR | | | JMP | UNPACK | |
| | JMP | UNPACK | | | TAD | MESAG2 | |
| | | | | DAC | POINTR | | |
| TYPACT, | TAD | MESAG2 | } | /** -DCA | LAC | M7 | |
| | DCA | POINTR | | | DAC | FNDCHK | |
| | TAD | M7 | | /** -DCA | | | |
| | DCA | ENDCHK | | | CLASQU | | |
| | JMS | MESSAGE | | | JMS | MESSAGE | |
| | JMS | CRLF | | | JMS | CRLF | |
| | JMP | START+2 | | | JMP | START+2 | |
| | | | | /** -REL | | | |

8TRAN Demonstration

| <u>MACRO-8 Program</u> | | | <u>Translation</u> |
|------------------------|--------|---------|--------------------|
| TYPE, | Ø | | TYPE |
| | TSF | | Ø |
| | JMP | .-1 | TSF |
| | TLS | | JMP |
| | CLA | | .-1 |
| | JMS I | TYPE | TLS |
| CRLF, | Ø | | CLA |
| | TAD | K215 | JMS* |
| | JMS | TYPE | TYPE |
| | TAD | K212 | Ø |
| | JMS | TYPE | TAD |
| | JMP I | CRLF | K215 |
| LISN, | Ø | | JMS |
| | KSF | | TAD |
| | JMP | .-1 | JMS |
| | KRR | | JMP* |
| | TLS | | CRLF |
| | JMP I | LISN | LISN |
| | | | Ø |
| | | | KSF |
| | | | JMP |
| | | | .-1 |
| | | | /** -REL |
| | | | KRR |
| | | | TLS |
| | | | JMP* |
| | | | LISN |
| MESSAGE, | Ø | | MESSAGE |
| | TAD I | POINTR | Ø |
| | JMS | TYPE | TAD* |
| | ISZ | POINTR | JMS |
| | ISZ | ENDCHK | ISZ |
| | JMP | .-4 | ISZ |
| | JMP I | MESSAGE | JMP |
| | | | .-4 |
| | | | /** -REL |
| | | | JMP* |
| | | | MESSAGE |
| NUMBER, | Ø | | NUMBER |
| M26Ø, | -26Ø | | Ø |
| TALLY, | Ø | | M26Ø |
| NUMSQR, | Ø | | -26Ø |
| MESAG1, | START1 | | TALLY |
| POINTR, | Ø | | Ø |
| M1Ø, | -1Ø | | NUMSQR |
| ENDCHK, | Ø | | Ø |
| STORE, | Ø | | MESAG1 |
| M4, | -4 | | START1 |
| DIGCTR, | Ø | | /** -US |
| K7, | 7 | | POINTR |
| M7, | -7 | | Ø |
| K26Ø, | 26Ø | | M1Ø |
| K212, | 212 | | -1Ø |
| K215, | 215 | | ENDCHK |
| MESAG2, | START2 | | 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

Translation

```
START1, 323 /S  
        321 /Q  
        325 /U  
        301 /A  
        322 /R  
        305 /E  
        304 /D  
        275 /=  
START2, 240 /SPACE  
        317 /O  
        303 /C  
        324 /T  
        301 /A  
        314 /L  
        256 /PERIOD  
$
```

```
START1 323 /S  
        321 /Q  
        325 /U  
        301 /A  
        322 /R  
        305 /E  
        304 /D  
        275 /=  
START2 240 /SPACE  
        317 /O  
        303 /C  
        324 /T  
        301 /A  
        314 /L  
        256 /PERIOD  
.END
```

INDEX

- Addition (+) operator, 3-2
- Additional code, generation of, 7-5
- Addressing,
 - direct, 2-2
 - indirect, 2-2
- Alphanumeric mode, 4-2
- AND (&) operator, 3-2
- Arithmetic package, 6-2
- ASCII character,
 - packing, 3-2, 3-3
 - translation, 7-4
- Assemblers, 3-1
- Assembly language, 1-1
- Asterisk (*) usage in MACRO, 3-1
- Auto-Index Registers, 2-2

- Bank/Page mode, 4-1
- Binary mode, 4-2
- .BLOCK pseudo-op, 3-2

- Calling procedure, 8-1
- Carriage RETURN, 1-1
- Character packing, 3-2, 3-3
- Characters, illegal, 7-6
- Choice of elements in command string, 1-1
- Code, generation of additional, 7-5
- Command element, optional, 1-1
- Command string, 8-2
 - symbols, 1-1
- COMPACT-15, 6-2

- .DAT slots, 8-1
- DCA instruction, 7-5
- Decimal numbers, 3-2
- DECtape controllers, 4-2
- Delimiters, 3-1
- Demonstration program, C-1
- Device, directoried, 8-3
- Device handlers, 4-3
- Direct addressing, 2-2
- Directoried device, 8-3
- Division and multiplication, 5-1
- Division (/) operator, 3-2
- Double precision constants, 3-2
- Double precision integers, 7-7

- EAE coding, 1-1, 5-1
- 8-bit ASCII characters, 3-3
- Error Conditions, 8-3, 8-4
- Exclusive OR (\) operator, 3-2
- Expression operators, 3-2
- EXPUNGE pseudo-op, 3-3
- Extended arithmetic element, 1-1, 5-1

- Field delimiter, 3-1
- Fields, 3-1
- FIXMRI pseudo-op, 3-3
- FIXTAB pseudo-op, 3-3
- Flags, 1-1, 4-1, 7-5
 - optional and mandatory, 8-2
 - summary, B-1
- Floating-point,
 - arithmetic, 6-1, 6-2
 - constants, 3-2, 6-3, 7-7
 - instructions, 1-1
 - pseudo-ops, 7-4
 - variable storage, 7-5

- Horizontal tab, 1-1

- Illegal characters, 7-6
- Inclusive OR (!) operator, 3-2
- Indirect addressing, 2-1, 2-2, 6-2
- Input/output, 4-1
 - in floating point, 6-2
 - instructions, 7-2
 - system macros, 4-3
- Instructions,
 - input/output, 7-2
 - memory reference, 7-1
 - operate, 7-3
 - reader/punch, 7-2
 - rotate, 7-6
 - Teletype, 7-2
- Instruction sets, 2-1
- Integers, signed and unsigned, 3-2
- Interrupt structure, 4-1

- Keyboard Flag, 4-2

- Labels, 3-1
- LAW instruction, 2-2
- Left angle bracket (<), 7-3
- Letters in MACRO symbols, 3-2
- Link status, 4-1
- Literals, 7-7
 - page zero, 3-3
 - pseudo-ops, 7-4
- Location Counter, 3-2
 - settings, 7-6

- Macro defining pseudo-ops, 7-3
- MACRO-8, 1-1
- MACRO XVM, 1-1
- Mandatory flags, 7-6, 8-2
- Memory protect, 4-1
- Memory reference instructions, 2-1, 7-1
- Mnemonics, translation of, 1-1
- Multiple precision constants, 7-7
- Multiplication and division, 5-1
- Multiplication (*) operator, 3-2

- Nested literals, 3-3, 7-7
- Numbers in MACRO symbols, 3-2

- Octal numbers, 3-2
- Operate instructions, 2-2, 7-3
- Operating procedure, 8-3
- Operation of 8TRAN, 8-1
- Operators, expression, 3-2
- Optional command element, 1-1
- Optional flags, 7-5, 8-2
- Overflow of Symbol Table, 7-7

- Page zero literals, 3-3, 7-4
- Paging, 2-2
- PAL-D, 1-1
- PAL-III, 1-1
- PAUSE pseudo-op, 8-3
- Percent sign (%) symbol, 3-2
- Period (.) symbol, 3-2
- Permanent Symbol Table of MACRO, 3-3
- Programs, symbolic, 3-1
- Pseudo-operations, 3-2, 7-3

- Reader/punch,
 - functions, 4-2
 - input/output, 1-1
 - instructions, 7-2
- Relative addresses, 7-5
- Rotate instructions, 7-6

- 7-bit ASCII characters, 3-3
- Shifting operations, 5-2
- Signed integers, 3-2
- 6-bit trimmed ASCII characters, 3-2
- Skip instructions, 7-7
- SMI (segmented microinstruction), 7-5
- Source code, 1-1
- Spaces used as field delimiters, 3-1
- Space () symbol, 1-1
- Statement delimiter, 3-1
- Statement reformatting, 1-1
- Status registers, 4-3
- Storage, floating-point variable, 7-5
- Subtraction (-) operator, 3-2
- Suppressing optional flags, 7-5
- Symbolic programs, 3-1
- Symbols in MACRO, 3-2
- Symbols, undefined, 7-6
- Symbol table, 3-3, 4-3
 - file, 8-3
 - overflow, 7-7
- Symbol translations summary, A-1

- Tab, horizontal, 1-1, 3-1
- Tape standard data format, 4-3
- Teletype,
 - input/output, 1-1
 - instructions, 4-2, 7-2
- Terminating pseudo-ops, 3-3
- Text handling pseudo-ops, 3-2, 7-3
- Translator between PDP-8 and XVM symbols, A-1
- Translator functions, 7-1

- Undefined symbols, 7-6
- Unsigned integers, 3-2
- User macros, 3-3
- User Symbol Table, 7-6

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 -----

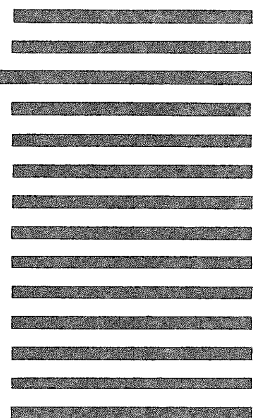
FIRST CLASS
PERMIT NO. 33
MAYNARD, MASS.

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