

FOCAL-GT/RT

user's manual

For additional copies, order No. DEC-11-GGTMA-B-D
from Software Distribution Center, Digital Equipment
Corporation, Maynard, Mass.

digital equipment corporation • maynard. massachusetts

First Printing January, 1973

Revised June, 1973

Copyright © 1973 by Digital Equipment Corporation

The following are trademarks of Digital Equipment Corporation,
Maynard, Massachusetts:

CDP	DIGITAL	KAl0	PS/8
COMPUTER LAB	DNC	LAB-8	QUICKPOINT
COMTEX	EDGRIN	LAB-8/e	RAD-8
COMSYST	EDUSYSTEM	LAB-K	RSTS
DDT	FLIP CHIP	OMNIBUS	RSX
DEC	FOCAL	OS/8	RTM
DECCOMM	GLC-8	OS/11	SABR
DECTAPE	IDAC	PDP	TYPESET 8
DIBOL	IDACS	PHA	UNIBUS
	INDAC		

PREFACE

The user of this manual is expected to be familiar with FOCAL programming and the GT40 graphics system. The following documents are prerequisites to using FOCAL-GT/RT:

FOCAL-11 USER'S MANUAL (DEC-11-LFOCA-D-D)
GT40 USER'S GUIDE (DEC-11-HGTGA-A-D)

The "HOW TO OBTAIN SOFTWARE INFORMATION" page, located at the back of this document, explains the various services available to DIGITAL software users.

The postage prepaid "READER'S COMMENTS" form on the last page of this document requests the user's critical evaluation. All comments received are acknowledged and will be considered when subsequent documents are prepared.

The software described in this document is furnished to the purchaser under a license for use on a single computer system and can be copied (with inclusion of DIGITAL's copyright notice) only for use in such system, except as may otherwise be provided in writing by DIGITAL.

The material in this document is for information purposes only and is subject to change without notice. DIGITAL assumes no responsibility for the use or reliability of software and equipment which is not supplied by it. DIGITAL assumes no responsibility for any errors which may appear in this document.

CONTENTS

	<u>Page</u>
CHAPTER 1 FOCAL-GT	1-1
1.1 THE DISPLAY	1-1
1.2 THE NEW FUNCTIONS	1-1
1.3 THE GRAPHICS FUNCTIONS	1-2
1.3.1 Lines, Points, and Characters	1-4
1.3.2 Skips	1-5
1.3.3 Use of Negative LOCS	1-6
1.3.4 Graphic Modes	1-6
1.3.5 LOC Coordinates	1-7
1.3.6 Light Pen Operation	1-7
1.3.7 Light Pen Hits	1-9
1.3.8 Erasures	1-9
CHAPTER 2 FOCAL-RT	2-1
2.1 LPS FUNCTIONS	2-1
2.1.1 FLED(ARG)	2-1
2.1.2 FTIC(ARG)	2-1
2.1.3 FSAM(chan1[,num,chan2])	2-2
CHAPTER 3 FOCAL-GT/RT CHARACTERISTICS	3-1
3.1 FOCAL-GT/RT AS A TERMINAL	3-2
3.2 FLOW OF DATA	3-2
3.3 SAVING AND RESTORING FOCAL PROGRAMS	3-4
3.4 PROGRAMMED MONITOR COMMANDS	3-5
CHAPTER 4 ASSEMBLING, LOADING, AND OPERATING FOCAL-GT/RT	4-1
4.1 ASSEMBLY PROCEDURES	4-1
4.2 LOADING PROCEDURES	4-4
4.3 NEW ADDRESSES	4-4
4.4 RESTART	4-4
4.5 ADDING NEW FUNCTIONS TO FOCAL-GT/RT	4-5
APPENDIX A FUNCTIONS AND SWITCHES	A-1
A.1 FUNCTIONS	A-1
A.1.1 Graphic Functions	A-1
A.1.2 LPS Functions	A-2
A.2 SWITCHES	A-2
APPENDIX B DECIMAL-CHARACTER EQUIVALENTS	B-1
APPENDIX C THE DISPLAY FILE	C-1
APPENDIX D BAUD RATES GREATER THAN 300	D-1
APPENDIX E FOCAL-11 ERROR DIAGNOSTICS	E-1
APPENDIX F FOCAL-11 COMMAND AND FUNCTION SUMMARY	F-1
F.1 COMMANDS	F-1
F.2 FUNCTIONS	F-4
APPENDIX G FOCAL-GT SYMBOL TABLE	G-1
APPENDIX H FOCAL-RT SYMBOL TABLE	H-1

CHAPTER 1

FOCAL-GT

FOCAL-GT is a version of FOCAL-11 that includes additional functions to handle a GT40 and additional code to enable the user to communicate with a host computer. The GT40 may include either a keyboard or a Teletype. FOCAL-GT may be conditionally assembled to include a Laboratory Peripheral System (see Chapter 4). FOCAL-RT, described in Chapter 2, performs all the operations described for FOCAL-GT in addition to the LPS functions.

FOCAL text is displayed on the GT40's screen and optionally on the Teletype if the system includes one. The user can create and run any FOCAL program that could be run using FOCAL-11 on a PDP-11/05. In addition, he can use the new functions to create graphic representations on the GT40's scope.

When used as a terminal, FOCAL-GT displays all communications with the host computer on the GT40 scope and optionally on the Teletype. The user can perform any Monitor operations he could perform from a Teletype. He can also save FOCAL programs and output on any storage device accessible to the host computer and read FOCAL programs previously saved on such storage devices.

1.1 THE DISPLAY

All text displays on the screen, 72 upright characters per line and a maximum of 32 lines. FOCAL-GT I/O displays as all upper case characters; terminal I/O may include upper and lower case characters. The user may dynamically vary the size of his text file.

When the text file is full, or when 32 lines have been displayed, FOCAL-GT clears the screen of text and resets a solid rectangular cursor to the upper left hand corner of the screen where additional input will be displayed. The FOCAL-GT cursor does not respond to cursor control characters nor to FORM FEED or TAB characters.

Graphics done by FOCAL consist of lines, points, and italic characters displayed in any of the modes available on the GT40. These modes include solid, dotdash, longdash, or shortdash lines, blinking or steady display, any of 8 intensities, and light-pen sensitive or insensitive. Any line, point, or character may have its own mode or may take on the mode of the previous data in the graphics file. The graphics file starts with solid, non-blinking, non-light pen sensitive graphics of intensity 4.

1.2 THE NEW FUNCTIONS

The new code added to FOCAL-11 is in the form of an FNEW as described in Section 6.3 of the FOCAL-11 USER'S MANUAL. This FNEW, unlike user-written FNEW's, constitutes a part of the FOCAL-11 source which can be conditionally assembled. The new FOCAL-11 also contains two additional operate options: S and H. FNEW consists of 11 graphic functions, code to handle terminal operations, and a display file.

The display file consists of two contiguous parts (see Appendix C), the graphics file, and the text file. The graphics file contains all graphic data that the user has displayed using the graphics functions. The text file contains all the I/O currently displayed on the screen. All I/O enters this file and displays on the screen; the user may cause output to echo on the Teletype by using the OPERATE T (O T) command and can reset to only screen echo by using OPERATE S (O S), which is the default setting. I/O output under the T option displays considerably more slowly than under the S option.

The graphics file consists of small blocks called LOCS which contain the graphic data. Each LOC has assigned to it a number corresponding to its offset from the start of the graphics file. The first LOC has the number 0 and starts at coordinates (0,0), the lower left hand corner of the screen. Graphic functions which add data to the graphics file use the LOC number as one of their arguments. The graphic data enters the LOC specified, overlaying any data currently in that LOC. This makes it possible, for instance, to continuously alter the length or angle of a vector or change its mode.

The user may use the FSKP function to increase the size of his graphics file while decreasing his text file. The text file is reset to its original size when the user types two CTRL/C's (typing C while holding down the CTRL key).

1.3 THE GRAPHICS FUNCTIONS

The graphics functions add data to the display file, alter its configuration, return data from the file, and handle the light pen.

Each of the functions that add data to the file includes the LOC offset as its first argument. Figure 1-1 shows the display file. It consists of a graphics file divided into LOCS and a text file. Data added to the graphics file enters the LOC specified in the function argument list. The function then returns the value of the next LOC in the file. If the user specifies a negative LOC, FOCAL-GT uses its absolute value, returning the value specified plus 1. If the user specifies a LOC beyond the end of the graphics file, FOCAL-GT returns the negative of this LOC +1 and places the data in the last LOC of the graphics file.

The GT40 starts executing the data in the graphics file at LOC 0. It sets the display beam to coordinates (0,0), the lower left hand corner of the screen, then moves the beam according to the information it encounters in each succeeding LOC, ignoring LOCS that do not contain any information. When the GT40 has completed its execution of the graphics part of the display file, it sets the display beam to the upper left hand corner of the screen and begins execution of the text file. At the end of the text file, it resets the beam to (0,0) and restarts the graphics file.

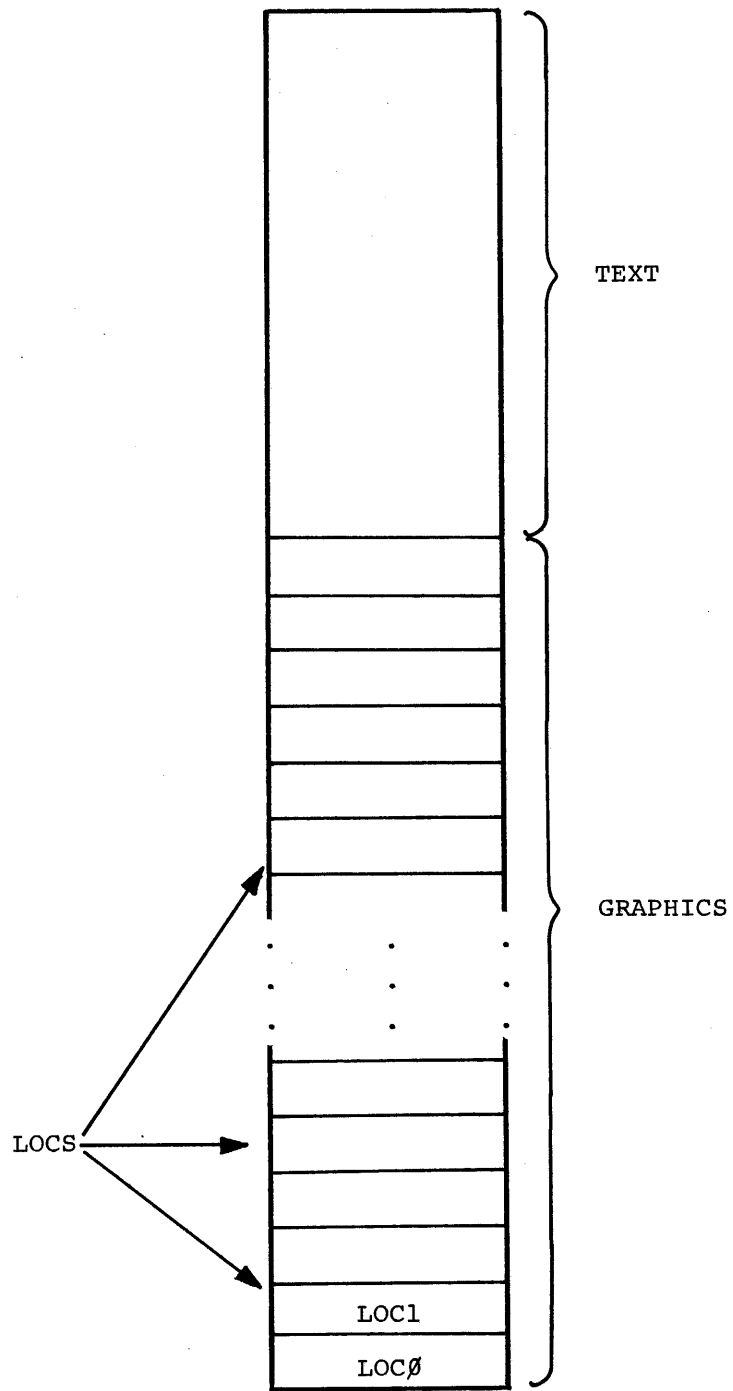


Figure 1-1 The Display File

1.3.1 Lines, Points, and Characters

FVEC(LOC,X,Y)

FVEC puts the vector (X,Y) into the graphics file. This vector starts where the last datum in the graphics file ended and overlays any datum currently in the LOC specified in its argument list.

Example:

```
F I=1,100;X FVEC(I,10,FSQT(I)*20)
```

This draws a parabola consisting of line segments in LOCS 1 to 100.

FMOV(LOC,X,Y)

FMOV moves the location of the display beam (X,Y). It is the equivalent of an invisible vector added to the graphics file.

Example:

```
1.1 S LOC=1
1.2 S LOC=FVEC(LOC,20,20);S LOC=FVEC(LOC,20,-20)
1.3 S LOC=FVEC(LOC,-40,0)
1.4 F I=1,500;X FMOV(0,I,I)
```

This draws a triangle then gradually moves it diagonally across the screen.

FPT(LOC,X,Y)

FPT adds a point at absolute location (X,Y) on the screen regardless of the previous location of the display beam. Any vector added to the file in the next LOC will start at this absolute location.

Example:

```
X FPT(1,512,390)
```

This draws a point at the center of the screen.

FSET(LOC,X,Y)

FSET sets the display beam to an absolute location on the screen. It is the equivalent of an invisible absolute point.

Example:

```
F I=1,2,10;X FSET(I,I*20,I*20);X FVEC(I+1,100,0)
```

This draws horizontal vectors at 5 points along a 45-degree slope.

FTXT(LOC,A,B,C,...)

FTXT inserts italic characters in the graphics file. Each LOC in the file can contain a maximum of four characters; the origin of these

characters (the lower left hand corner of the first character) starts at the current position of the display beam. FTEXT inserts characters starting at the LOC specified in the first argument and continues to insert characters until it reaches the last argument, going on to the next LOC in the graphics file whenever one fills. It returns the value of the last LOC it has used plus 1.

The argument of FTEXT take the form of numeric values that correspond to FOCAL character values or to ASCII values. For instance, FOCAL-GT interprets the argument 0D as the letter D, but it also interprets the argument 68 as the letter D. Appendix B contains a complete list of FOCAL-GT's character set and the corresponding decimal values. For further explanation of FOCAL's treatment of characters used as numeric values, see the FOCAL-11 USER'S MANUAL.

EXAMPLE:

```
1.1 X FMOV(1,500,500);S LOC=2
1.2 S LOC=FTEXT(LOC,0N,05,0W,0BL,0L,05,0T,0T,05,0R,58)
1.3 ASK "LETTER" L
1.4 X FTEXT(LOC,L);G 1.3
```

This displays NEW LETTER at the center of the screen, then accepts input and displays the value of the input. Note that 0BL has the value 32 which corresponds to a blank.

1.3.2 Skips

FSKP(LOC[,L2])

FSKP serves a number of purposes depending upon the values of its arguments. Basically, it places in the LOC specified in the first argument, a jump to the LOC specified in the second argument. A jump serves to skip all data in the intervening LOCS. The second argument, L2, is shown in square brackets to indicate that it is optional.

If L2 is beyond the entire graphics file, FOCAL-GT increases the length of the graphics file at the expense of the text file so that L2 becomes the last LOC. In so doing, FOCAL-GT clears the text file, resetting the cursor to the upper left hand corner of the screen and inserts, in LOC, a jump to the start of the text file. If L2 specifies an offset that surpasses the entire text file, FOCAL-GT decreases the size of the text file so that it displays only one line at a time and inserts, in LOC, a jump to the start of the text file.

If L2 is negative, FOCAL-GT inserts a jump, in LOC, to the beginning of the graphics file. Thereafter, FOCAL-GT never reaches the part of the display file that displays text -- so the text does not display. If the user wished to type in additional data, he would have to do so without an echo on the screen.

If the argument L2 does not appear in the argument list of FSKP, as in XE FSKP(LOC), FOCAL-GT inserts a jump to the start of the text file and clears the text file. Thus, the function may be employed to clear all text display from the screen.

Examples:

For a graphics file 100 LOCS long,

```
X FSKP(10,50)
```

skips around all data in LOCS 11 to 49;

```
X FSKP(10,150)
```

makes the graphics file 150 LOCS long and clears the text file;

```
X FSKP(10,10000)
```

makes the text file one line long, devoting the remainder of the display file to graphics;

```
X FSKP(10,-1)
```

enters a skip to the start of the graphics file so that any text in the text file does not display;

```
X FSKP(10)
```

enters a skip to the start of the text file and clears the text file.

1.3.3 Use of Negative LOCS

As mentioned previously, FOCAL-GT interprets a negative value in the LOC argument of a graphics function as its absolute value but returns the negative value plus 1. FOCAL-GT also returns the negative of a LOC argument that exceeds the end of the graphics file. The user can use these characteristics to draw in sequentially lower LOCS and to automatically reverse when graphics input reaches the end of the graphics file.

Example:

```
1.1 X FSKP(1);S LOC=1
```

```
1.2 F I=0,.1,1000;S LOC=FVEC(LOC,1,FSIN(I)*10+1)
```

This draws a sinuous wave at an upward angle across the screen from left to right. Then, when the graphics file fills, it draws the wave in the opposite direction. When the left hand edge of the screen is reached, it will start again.

1.3.4 Graphic Modes

FDIS(TYPE,INT,BLINK,LP)

Each datum that enters the display file has associated with it a set of 4 modes. These modes are established with an FDIS command and apply to all subsequent graphics functions until a new FDIS command is issued. The original modes of the graphics file are solid line, intensity 3, steady display, and non-light pen sensitive.

The arguments of FDIS have the following significance:

TYPE	-Line Type =0 means solid lines =1 means longdash lines =2 means shortdash lines =3 means dotdash lines
INT	-Intensity A number between 0 (lowest intensity) and 7 (highest intensity)
BLINK	-Blinking or steady display =0 means steady =1 means blinking
LP	-Light pen sensitivity =0 means non-light pen sensitive graphics =1 means light pen sensitive graphics

Example:

```
1.1 XE FDIS(3,1,1,0);XE FVEC(0,0,0)
1.2 XE FDIS(0,7,0,1);XE FMOV(1,50,50)
1.3 XE FVEC(2,-100,0);XE FVEC(3,0,-100)
1.4 XE FVEC(4,100,0);XE FVEC(5,0,100)
```

This draws a bright, light pen sensitive square whose center moves to the point on its circumference hit by the light pen, and a blinking, dotdash line from the lower left hand corner of the screen to the center of the square.

1.3.5 LOC Coordinates

FXCO(LOC) and FYCO(LOC)

FXCO and FYCO return the X and Y coordinates, respectively, of the LOC specified in the argument. If the LOC contains no data or if it contains a skip, FXCO and FYCO return zero values.

Example:

```
X FVEC(10,FXCO(10)+100,FYCO(10)+100)
```

This extends the vector in LOC 10.

1.3.6 Light Pen Operation

When FOCAL-GT senses a light pen hit on graphics specified as light pen sensitive, it saves the value of the LOC containing the line, point, or character hit and puts the coordinates of the hit into LOC 0. Since FOCAL-GT allows only vectors and points in LOC 0, the vector or point in LOC 0 takes on the coordinates of the light pen hit. For example, an invisible vector in LOC 0 always extends from the lower left hand corner of the screen to the location of the last light pen hit. The user may take advantage of this property of LOC 0 in several ways.

For example, FXCO(0) and FYCO(0) can be used to return the coordinates of the last light pen hit. Also, LOC 0 can be used to implement tracking by drawing a light pen sensitive tracking cross with its center starting in LOC 1, and an invisible vector in LOC 0. Whenever FOCAL-GT gets a hit on this cross, it will put the coordinates of the hit into the invisible vector in LOC 0 causing the center of the cross to move to the location of the hit. Thus the cross, once drawn, will track the light pen without program control. The user usually inserts an absolute point in the LOC following the last LOC used by the tracking cross so that any graphics following this point will not move.

The user may connect individual pictures he has created by means of the light pen. This is done by inserting a skip in LOC 1 to the start of the picture to be connected. If the picture consists entirely of light pen sensitive vectors, its origin moves to wherever the light pen hits it. When the picture is in position, the user need only determine the coordinates of its new origin using FXCO(0) and FYCO(0) and insert an absolute point with these coordinates in the LOC preceding the origin of the picture. He may then remove the skip from LOC 1 and the picture will remain where he left it.

Example:

```

1.1 D 9
1.2 S LOC=FSET(LOC,0,0);S X1=0;S Y1=0
2.2 ASK "DRAW"D;S X=FXCO(0);S Y=FYCO(0)
2.3 S LOC=FVEC(LOC,X-X1,Y-Y1);S X1=X;S Y1=Y;G 2.2
9.05 XE FMOV(0,500,400)
9.07 S T=40;S R=.7*T;S S=.5*T
9.1 X FDIS(0,5,0,1)
9.2 X FVEC(1,R,S)
9.3 X FVEC(2,-R,-S-T);X FVEC(3,-R,S+T)
9.4 X FVEC(4,2*R,-2*S)
9.5 X FVEC(5,-R,S+T);X FVEC(6,-R,-S-T)
9.6 X FVEC(7,2*R,2*S)
9.7 X FVEC(8,-2*R,0)
9.8 X FVEC(9,2*R,-2*S)
9.9 X FVEC(10,-2*R,0);X FVEC(11,R,S)
9.95 S LOC=12

```

This displays a tracking cross which can be used to draw lines by hitting the RETURN key. The RETURN key draws a line from the previous location of the cross to its new location. The cross starts and ends at its center; consists of 13 lines, all visible and of considerable length; and registers light pen hits in any direction that the light pen moves.

The user may construct other tracking crosses responsive to the needs of the individual program.

Example:

```
F I=50,100;XE FCLR(I,0,0)
```

This clears LOCS 50 through 100.

1.3.7 Light Pen Hits

FLP (ARG)

FLP returns the LOC of the data which caused the last light pen hit. If ARG=0, FLP waits for the next light pen hit before returning to the caller. If ARG does not equal zero, FLP returns immediately. FLP allows the light pen to be used to initiate program operations as well as to determine where in the graphics file a light pen hit occurred. This property can be used, for instance, to implement a menu, using the LOC of the letters hit to determine which operations to perform.

Example:

```
1.1 XE FPT(1,0,0);X FMOV(2,0,500);X FDIS(0,6,0,1)
1.2 X FTXT(3,0U,0P);X FMOV(4,0,-100)
1.3 X FTXT(5,0D,0O,0W,0N)
1.4 S X=500;S Y=400
1.5 X FPT(6,X,Y)
1.6 I (FLP(0)-4)1.7,1.7,1.8
1.7 S Y=Y+20;G 1.5
1.8 S Y=Y-20;G 1.5
```

This displays a point near the center of the screen and a menu consisting of the two words UP and DOWN. When the light pen hits UP, the point moves up; when it hits DOWN, the point moves down.

When a light pen hit occurs near an edge of the screen on a vector that displays partly off the screen, FOCAL-GT may insert erroneous coordinates in LOC 0. Since the hardware may not inform FOCAL-GT of the light pen hit until the vector has already gone off the screen, FOCAL-GT may put the wraparound coordinates of the vector in LOC 0. For instance, a vector drawn off the screen to the left may return an X-coordinate of 1023 rather than 1 where the hit actually occurred. FOCAL-GT never returns coordinates not within the screen's viewing area, so a tracking cross never disappears, although it may jump from one edge of the screen to the opposite edge.

1.3.8 Erasures

FCLR (ARG)

FCLR clears the entire graphics file, without altering the text file or the size of the graphics file. It returns a value equal to the ARG+1 though the argument has no effect on its operation.

The user may clear individual LOCS by inserting 0 length vectors.

Example:

```
X FCLR()
```

This clears the graphics file, leaving the text file undisturbed.

CHAPTER 2

FOCAL-RT

FOCAL-RT, a new FOCAL for use with the GT40 and LPS (Laboratory Peripheral System), extends and modifies FOCAL-GT. The user can obtain two separate standard binaries: FOCAL-GT for use with a GT40 without LPS and FOCAL-RT for use with a GT40/LPS combination.

In addition, the user may obtain the FOCAL-GT/RT source and assemble it to produce a variety of nonstandard configurations. Chapter 4 contains the assembly instructions for FOCAL-GT/RT.

2.1 LPS FUNCTIONS

The following three LPS functions allow the user to sample any of the 8 A/D channels, to start the clock and use it as a timer for FOCAL operations, and to put an integer value into the numeric display on the face of the LPS.

2.1.1 FLED(ARG)

The FLED function loads the integer part of the argument into the numeric display on the face of the LPS (called the LEDS). FOCAL-RT only uses the 5 least significant digits of the integer part of the argument and returns the value of the remaining digits.

Example:

```
1.1 X FLED(FSAM(0)-2048);G 1.1
```

This loads the LEDS with the value in channel 0 less 2048, a value between -2048 and +2048.

2.1.2 FTIC(ARG)

The FTIC function allows the user to set and read the timer, a 16-bit register that the LPS clock increments each time it ticks. The timer maintains a count of the number of ticks that have occurred since the clock started. Its 16 bits can count up to 32767 before it clears to zero and continues incrementing.

Calling FTIC with an argument greater than zero sets the timer to zero and starts the clock ticking every ARG/10000th of a second. It returns the current value of the timer, which is 0.

Any error except Error 0.00 stops the clock, if the timer is non-zero at the time of the error, thereby leaving the timer at its value at the time of the error. Error 0.00 clears the timer to zero.

Example:

```
1.1 X FTIC(1000)
1.2 T FTIC(0);G 1.2
```

This continually displays the value of the clock which ticks once every 1/10 of a second.

Example:

```
1.1 X FTIC(100)
1.2 X FTIC(-100);T "TIC",!;G 1.2
```

This displays "TIC" once a second.

2.1.3 FSAM(chan1[,num,chan2])

The FSAM function may be used as follows to sample the channel specified in the argument and return its value as an integer between 0 and 4096.

Example:

```
1.1 X FVEC(1,FSAM(0)/4,FSAM(1)/4)
1.2 G 1.1
```

This continuously draws a vector with X and Y coordinates determined by the values in channels 0 and 1.

Optionally, the FSAM function may be used to sample up to 5000 points per second and place the values into a 1000-word buffer. It may also use Schmitt trigger 1 to initiate sampling. The optional FSAM arguments take the following form:

FSAM(+A[,+B,+C])

One Argument: For A positive, sample channel A.
For A negative, return Ath to last sample from the buffer.

Two Arguments: A must be positive.
For B positive, sample B points from channel A.
For B negative, wait for Schmitt pulse, then sample B points from channel A.

Three Arguments: Same as for two arguments except that it allows a dual sample from channels A and C. Every time the clock ticks, the value from A will enter the buffer followed, within 30 us., by the value from C.

To use these optional arguments, the user first starts the clock ticking with the FTIC function as described above. The user must ensure that the clock does not tick faster than the highest sample rate: 2/10000 for single sampling, 3/10000 for dual sampling. FTIC sets the clock ticking in intervals determined by its argument over 10,000 parts of a second. FSAM may then sample all 8 A/D channels, two at a time if desired, and use Schmitt trigger 1 to initiate sampling. It may also read at random from the LPS input buffer.

Example:

```
1.1 S A=FSAM(2,100)
```

This reads 100 values from channel 2, one for each clock tick, into the LPS buffer. A is set equal to the last value.

Example:

```
1.1 S A=FSAM(0,100,2)
```

This reads a total of 100 values, two at each clock tick, the first from channel 0 and the second from channel 2 with 20 to 30 us. between members of the pair.

NOTE

Dual sampling can sample up to 6667 points per second (3333 dual samples per second).

Example:

```
1.1 X FSAM(7,-10)
```

This stops the clock, restarts it on A Schmitt pulse, then reads 10 samples, one on each clock tick.

Example:

```
1.1 S A=FSAM(2,1)
```

This sets A equal to the value in channel 1 as soon as the clock ticks.

Example:

```
1.1 S A=FSAM(-1)
```

This sets A equal to the last sample.

Example:

```
1.1 S A=FSAM(-5)
```

This sets A equal to the fourth to last sample.

If the user wishes to display his samples as points on the screen, he may do so by reading their values from the LPS input buffer and putting them into the graphics file.

Example:

```
1.1 X FSAM(0,500)
1.2 S LOC=1
1.3 F I=1,10,1000;S LOC=FPT(LOC,I,FSAM(-I))
```

This displays the first 100 points sampled along the X axis, incrementing 10 units per point.

CHAPTER 3

FOCAL-GT/RT CHARACTERISTICS

The user may create FOCAL programs and execute command lines under FOCAL-GT/RT the same as under FOCAL-11. FOCAL-GT/RT prompts with a # sign rather than an *. A FOCAL program that contains none of the new functions runs exactly the same under FOCAL-GT/RT as under FOCAL-11 except that I/O is displayed on the screen and only displays on the Teletype after an O T instruction. The user may execute all graphic and LPS functions either within a program or directly as shown in the previous examples. Once a graphics program has started, typing CTRL/C twice (holding down the CTRL key while typing the C key) stops it, clears the display file, and resets the size of the graphics and text files to their original values. To stop a graphics program without erasing its graphics, hit any two characters other than CTRL/C at some point in the program's execution. The program halts with an 18 error, leaving its graphics on the screen and the display file unchanged. The display can be altered with direct commands or with another program.

An unusual and sometimes useful phenomenon the user may observe when overlaying one picture with another that uses the same LOCS is the movement of the old picture as the new one is created. This occurs because all vectors extend relative to the previous vector in the graphics file. Altering some of the vectors that began a picture moves the remaining vectors in the picture around the screen.

Example:

```
F J=1,10;F I=1,10;X FVEC(I,10,J)
```

This draws a line composed of line segments in LOCS 1 to 100, then redraws it at greater slopes causing the line displayed to bend.

The user may also find that certain figures he has drawn, e.g., a circle composed of line segments, do not seem to close when the figure completes. The fact that all vectors display with integer lengths whereas the arguments to the vector functions may have non-integer parts causes this rounding error. To minimize this error, add .5 to all non-integer arguments so that errors cancel.

Example:

```
1.1 X FSKP(1,400);X FSET(0,800,400);X FVEC(1,-800,0);S LOC=2
1.2 F I=.1,.2,12.56;S LOC=FVEC(LOC,10,FSIN(I)*10+.5)
```

This draws a cosine wave consisting of 62 line segments above a tangent. The peaks of the cosine wave touch the tangent as they should. Neglecting to add the rounding factor, .5, produces a wave whose peaks dip below the tangent.

When text or skip data continually enters the graphics file, the display blinks somewhat; when the data has stopped entering the file, the display stops blinking and grows brighter. FOCAL-GT/RT must stop the display every time it adds a piece of such data to the graphics file; this causes the display to blink.

3.1 FOCAL-GT/RT AS A TERMINAL

If the GT40 is connected via an asynchronous line of baud rate 300 or lower to a host computer, it can communicate with that computer displaying all communication on the screen of the GT40, and optionally, if the GT40 has a Teletype, on the Teletype. The GT40 can be used in the same way as a Teletype to do all Monitor operations normally done on the host computer. However, any commands not normally handled by a Teletype cannot be used since FOCAL-GT/RT does not interpret command characters not normally understood by a Teletype. The text display displays up to 40 lines of I/O and then clears.

FOCAL-GT/RT also makes it possible to send FOCAL output, including the FOCAL program itself, to the host computer and allows FOCAL to listen to input, including a FOCAL program, sent to it from the host computer. The user switches between the various modes of operations using two control characters which FOCAL-GT/RT traps and one additional operate option.

3.2 FLOW OF DATA

Figure 3-1 portrays a GT40 system consisting of FOCAL, a keyboard, a scope, and a host computer. The arrows indicate the possible flow of output from each device. The state of three switches determines each direction. Output flows from its origin along the arrows indicated by the setting of the switches until it reaches the scope. Typing CTRL/F causes the direction of keyboard output to change. Typing CTRL/T causes the direction of host output to change. The operate options S, for screen, and H, for host, determine the direction of FOCAL output.

As Figure 3-1 indicates, output from the keyboard can go either to FOCAL or to the host computer, depending on the state of the CTRL/F switch. Output from the host computer, including its echo of the keyboard or FOCAL input, can go either to the host or to the screen depending on the state of the CTRL/T switch. Output from FOCAL, including its echo of host or keyboard input, can go either to the host or to the screen depending on the state of the operate option. For example, with the CTRL/F switch pointing to the host and the CTRL/T switch pointing to the screen, notice on the diagram that keyboard output goes to the host which echoes it to the screen. If the CTRL/T switch had pointed to FOCAL, and FOCAL had the S option operative, keyboard output would have gone first to the Monitor, then to FOCAL, then to the screen. Both the host Monitor and FOCAL would process all command lines resulting in an error. Usually two prompting characters on a single line, one from FOCAL, the # sign, and one from the host Monitor, indicate that this problem is occurring. Change the direction of the CTRL/T switch to solve this problem. The CTRL/T that discontinues output to the host computer echoes an underline character. This character indicates that the current direction of the host computer is to the screen and also serves to erase any characters that the host computer may have left on the current line so that FOCAL will ignore them.

The operate character H (host) sends FOCAL output to the host computer. For example, O H;W A;O S writes a FOCAL program to the host computer, then returns the screen to the output device.

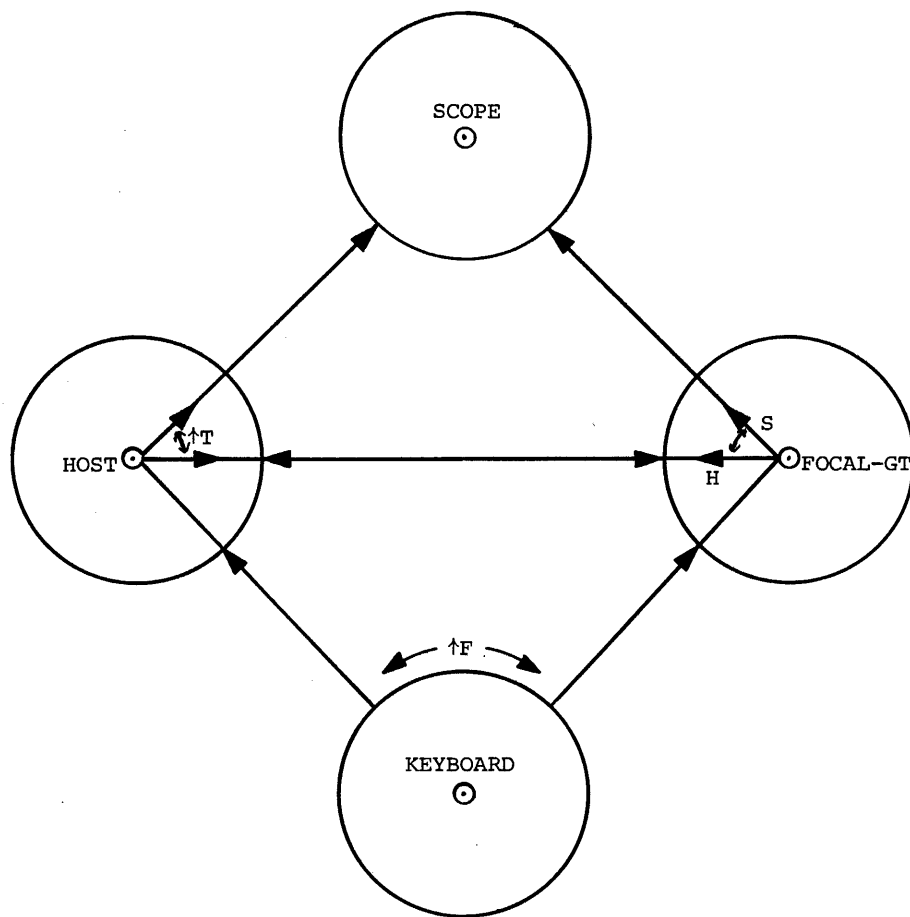


Figure 3-1 Flow of Data

3.3 SAVING AND RESTORING FOCAL PROGRAMS

The various switches that control I/O allow the user to use the host computer for storage of FOCAL programs. To save a program on the host computer's storage device, for example, type CTRL/F to begin communications with the host computer. Then create a file, giving it the name of the FOCAL program. Begin insertion from the Teletype, then type CTRL/F to resume communication with FOCAL. Once in FOCAL, use the H option and the WRITE command to write the program to the file on the host computer. When FOCAL-GT/RT has finished writing the file, type CTRL/F to resume communications with the host computer and close the file which now contains the FOCAL program. In the same way, a file of FOCAL output can be created on the host computer by inserting an O H command in the FOCAL program before output commences.

A FOCAL program can be read from the host computer by instructing the host computer to print the program and, before this output begins, typing CTRL/T. The output displays on the screen, but FOCAL-GT/RT receives this output as though it had come from the keyboard.

FOCAL may also read data files from the host computer as follows. Start the FOCAL program running. When it prints a colon to indicate that it wants input, type CTRL/F and instruct the host computer to print a file, typing CTRL/T before output begins. If the loop in which FOCAL asks for information is fast enough to keep up with the rate of output from the host computer, FOCAL will be able to read the file.

NOTE

When CTRL/F is typed for the first time, control transfers to LOADVT which is still in core and the system responds with:

>R*

Type CTRL/C to return to the host Monitor. All subsequent CTRL/F's will cause a response of . or #.

Example:

The following assumes that a user has created a FOCAL program and has a DECsystem-10 as host computer. Characters printed by the system are underlined, control characters typed by the user but not echoed on the screen appear within angle brackets. Originally, the user is talking to FOCAL.

```
#<↑F><CR>
.MAKE PROG.FOC<CR>
*I<↑F><CR>
#O H;W A;O S<CR>
C: FOCAL
1.1 S A=5
1.2 T A
#<↑F><CR>
$EX$$
.TYPE PROG.FOC
C: FOCAL
1.1 S A=5
```

```

1.2 T A
.<↑F><CR>
#

```

NOTE

At baud rates higher than 300 on a heavily loaded system, TECO may not be able to keep up with the input from a WRITE ALL statement. This makes saving a program which consists of many lines impossible using the WRITE ALL statement. Under these circumstances, the user may save a program a line at a time. For example, O H;W 1.1;O S sends line 1.1 to TECO.

This transfers the contents of PROG.FOC to FOCAL and saves a FOCAL program in a file called PROG.FOC.

Example:

At some later time, the user wishes to restore his program.

```

#<↑F><CR>
.TYPE PROG.FOC<↑T><CR>
#C: FOCAL
#1.1 S A=5
#1.2 T A
.<↑T>+<↑F><CR>
#W A
C: FOCAL
1.1 S A=5
1.2 T A
#

```

For saving and storing FOCAL programs on paper tape, refer to the FOCAL-11 USER'S MANUAL, Section 3.4.1.1.

3.4 PROGRAMMED MONITOR COMMANDS

A FOCAL program can also issue host Monitor commands via the TYPE command. If an O H precedes a TYPE command, the output from the TYPE command goes to the host computer. If this output is in the format of a host computer Monitor command, the host computer interprets it as if it had come from a terminal. However, programming FOCAL to execute a series of Monitor commands depends on the ability of the host computer to buffer command input since FOCAL outputs these commands as soon as it reaches the TYPE command. The user may use a clock or a long FOR loop to slow down FOCAL's output but, in any case, if the input to the host computer exceeds its ability to buffer input, the host computer responds as if the excess input had come from any other terminal device.

Example:

```
1.1  O H;T "MAKE OUTPUT.FOC",!, "I"  
1.15 F I=1,100;S A=5  
1.2  F I=1,10;T FSAM(0),!;F J=1,100;S A=5  
1.3  T "$EX$$";O S
```

This creates a file called OUTPUT.FOC and prints 10 A/D values into it, using FOR loops to leave time between each value output. It then closes the new file. Note that FOCAL-GT/RT echoes \$ when it receives an ALTMODE as in line 1.3.

CHAPTER 4

ASSEMBLING, LOADING, AND OPERATING FOCAL-GT/RT

The user may obtain the FOCAL-GT/RT source from the Software Distribution Center. This source is provided on a PDP-10 formatted DECTape (DEC-11-GGRSA-A-UA). The user assembles this source on DECsystem-10 using MACDLX, the PDP-11 assembler for the PDP-10. The parameters defined in the assembly process determine the nature of the binary produced. For example:

```
.R MACDLX
*FOCRT.BIN/I/EQ:GT40:TERM:RT+FOCGT.002
```

This produces the standard binary of FOCAL-RT.

4.1 ASSEMBLY PROCEDURES

Assemble FOCAL-GT or FOCAL-RT with MACDLX on the DECsystem-10. Figure 4-1 depicts the code in the source of FOCAL-GT/RT that defines the size of the display file and the parameters that determine the nature of the binary produced. The user defines these parameters at assembly time.

Examples:

```
FOCAL.BIN/I+FOCGT.002
```

This produces FOCAL-11 which does not make use of the display.

```
FOCGT.BIN/I/EQ:GT40+FOCGT.002
```

This produces an 8K FOCAL-GT with no terminal code.

```
FOCGT.BIN/I/EQ:GT40:TERM:PARAM+TTY:,DSK:FOCGT.002
```

```
READY
KSIZE=10000
DSIZE=1000
TSIZE=700
↑Z
READY
↑Z
```

This produces a 4K FOCAL-GT with a display file 1000 (octal) words long and an initial text portion 700 (octal) words long.

```

      ,IFDF   GT40          IGT40 FUNCTIONS
I **DNEW**

ICOPYRIGHT 1972,1973 DIGITAL EQUIPMENT CORPORATION
IMAYNARD, MASS 01754

I THE FUNCTIONS IN DNEW ENABLE THE FOCAL-11 USER TO DRAW ON A
I GT40;

ICONDITIONAL ASSEMBLY PARAMETERS

IASSEMBLING WITH NO PARAMETERS DEFINED IN THE ASSEMBLY STRING,
IPRODUCES FOCAL-11 WHICH WILL RUN IN AN 11/09

IASSEMBLING WITH GT40 DEFINED PRODUCES AN 8K FOCAL-GT WITH NO
ILPS FUNCTIONS

IASSEMBLING WITH GT40 AND RT DEFINED PRODUCES AN 8K FOCAL-RT
IWITH A BUFFER 1000 WORDS LONG

IASSEMBLING WITH GT40, RT, AND NOBUFF DEFINED PRODUCES AN 8K FOCAL-GT
IWITH LPS FUNCTIONS AND AN FSAM THAT DOES NOT USE A BUFFER

I THE USER MAY DEFINE STANDARD PARAMETERS IN HIS ASSEMBLY STRING
IBY DEFINING 'PARAM' IN THAT STRING FOLLOWED BY THE NEW PARAMETERS,
IIN THIS CASE, HE MUST DEFINE:

I      KSIZE      THE SIZE OF CORE IN WORDS
I      DSIZE      THE SIZE OF THE DISPLAY FILE IN WORDS
I      TSIZE      THE SIZE OF THE TEXT PORTION OF THE DISPLAY FILE

IIF THE USER DEFINES GT40, RT, AND PARAM, HE MUST ALSO DEFINE
I(IN ADDITION TO THE 3 PARAMETERS ABOVE):

I      LPBSIZ     THE SIZE OF THE LPS BUFFER (IN WORDS)

IASSEMBLING ANY OF THE ABOVE WITH 'TERM' DEFINED PRODUCES A VERSION THAT
IINCLUDES TERMINAL CODE

,ENABLE AMA
,ENABLE ABS

```

Figure 4-1 Display File Code

```

;
;THE FOLLOWING PARAMETERS DETERMINE THE SIZE OF DISPLAY FILE
;RELATIVE TO FOCAL BUFFER AREA, THE PORTION OF THE DISPLAY
;FILE DEVOTED TO TEXT INITIALLY, AND THE TYPE OF FOCAL TO PRODUCE,
;THE USER WHO DEFINES (PARAM) IN THE ASSEMBLY STRING MUST DEFINE
;ALSO EACH OF THE PARAMETERS DEFINED BELOW,

```

```

;IFNDF PARAMS 18K DEFINITIONS, DEFAULT DEFINITIONS

KSIZE=20000  ;CORE SIZE (20000*8K WORDS)
;IFDF RT      ;IF LPS CODE
;IFNDF NOBUFF ;IF FULL SUPPORT
DSIZE=4000    ;SET DISPLAY FILE SIZE
LPBSIZ=1702   ;1000 WORD INPUT BUFFER
;ENDC
;IFDF NOBUFF  ;IF NON-BUFFERED SUPPORT,
DSIZE=5000    ;MORE ROOM FOR DISPLAY FILE
LPBSIZ=0      ;NO BUFFER
;ENDC
;ENDC
;IFNDF RT      ;IF NO LPS SUPPORT, PRODUCE FOCAL=GT
DSIZE=5300    ;WITH LARGE DISPLAY FILE
LPBSIZ=0      ;AND NO BUFFER
;ENDC
TSIZE=3000    ;IN ALL CASES PORTION OF DISPLAY FILE DEVOTED
;            ;TO TEXT INITIALLY
;ENDC

;
;IFDF K4       ;4K DEFINITIONS (DEFINE PARAM +THESE VALUES IN ASSEMBLY STRING)
;KSIZE=10000   ;CORE SIZE
;DSIZE=1000    ;DISPLAY FILE SIZE
;TSIZE=700     ;TEXT FILE SIZE
;LPBSIZ=0      ;LPS BUFFER (NONE)
;ENDC

```

Figure 4-1 (Cont.) Display File Code

4.2 LOADING PROCEDURES

The user may load his FOCAL binary via the paper tape reader or over an asynchronous line connected to a host computer on whose storage area both the binary of FOCAL and the loader program reside. To load from a host computer:

1. LOAD ADDRESS 166000 AND START.

This starts the hardware bootstrap which will display I/O on the screen and from which the user may perform all Monitor commands. To reset the screen when output reaches the screen's bottom, hit START again.

2. START THE LOADER PROGRAM.

To the DECSys-10 Monitor, type RU LOADVT.

3. Give the loader the name of the proper bootstrap (BOOT4.BIN for 4K; BOOT8.BIN for 8K). (The binary tape supplied requires the 8K bootstrap.)
4. When the prompting character appears again, give the loader the name of the FOCAL binary.
5. Wait until the FOCAL error message 00 appears on the screen.

Example:

```
.RU LOADVT<CR>
*←FOCGT.BIN<CR>
?00 AT 0.00
#
```

The procedure for loading FOCAL-GT/RT from paper tape requires loading the Bootstrap Loader and the Absolute Loader, followed by the FOCAL-GT/RT paper tape. For more detailed instructions, refer to Chapter 6 of the PDP-11 PAPER TAPE SOFTWARE PROGRAMMING HANDBOOK (DEC-11-XPTSA-A-D). To make more efficient use of core, FOCAL-GT/RT uses the core where the loaders resided when it runs.

4.3 NEW ADDRESSES

To eliminate the terminal code from FOCAL-GT, enter the address INITD-2 in location BOTTOM (1652).

For the purposes of adding new functions, for both FOCAL-GT and FOCAL-RT, consider the address in BOTTOM as the top of core.

4.4 RESTART

Operation of FOCAL-GT/RT differs from that of FOCAL-11 only in that on restart and power-fail, the user must type CTRL/C twice to FOCAL to restart the display. In general, if the display disappears for any reason, two CTRL/C's sent to FOCAL (not to the host Monitor) will bring it up again.

4.5 ADDING NEW FUNCTIONS TO FOCAL-GT/RT

Chapter 6 of the FOCAL-11 USER'S MANUAL describes the procedure for adding an FNEW to FOCAL-11. FOCAL-GT/RT already contains such an FNEW called DNEW. DNEW consists of the graphic and LPS functions, the display file, and the terminal hook-up code. The user may add additional code in the area just below the start of DNEW and additional function names in the function list starting where the new functions leave off. In other words, he may add an additional function to FOCAL-GT/RT in the same way he would extend an FNEW he had written himself.

For both FOCAL-GT and FOCAL-RT, the address in location BOTTOM is the highest free address in core. The highest free address in the function list is 1012 for FOCAL-RT and 1026 for FOCAL-GT. The user who has obtained the source of FOCAL-11 may determine these addresses directly from the listing for his particular binary.

Referring to the FNEW example in Section 6.3.5 of the FOCAL USER'S MANUAL, FQUAD may be added to the 8K standard binary of FOCAL-GT/RT by altering the example as follows: make its starting address equal to the address contained in location BOTTOM-144. Then, in location STACK0+2, change '1100-4' to '1012-4' for RT or '1026-4' for GT and change the absolute address setting '.=1100+2' to '.=1012+2' for RT or '.=1026+2' for GT. The user who intends to add an FNEW must, however, have a paper tape reader attached to his GT40 or have access to the source of FOCAL-GT/RT because the GT40's loading program cannot load overlays from the DECsystem-10 (host computer).

To add an FNEW to FOCAL-GT/RT and load the new code from the PDP-10, the user must delete the .END statement from FOCAL-GT/RT and combine its source with the source of his overlay when assembling, then load the single new binary produced.

Example:

Delete the .END statement from FOCGT.002, then:

```
.R MACDLX
*FOCNEW.BIN+FOCGT.002,FNEW.SRC
```

This will produce a binary that includes the new functions. An FNEW used in this manner need not redefine variables such as FGET already defined in the FOCAL-GT/RT source.

APPENDIX A FUNCTIONS AND SWITCHES

A.1 FUNCTIONS

A.1.1 Graphic Functions

FVEC(LOC,X,Y)	Puts vector (X,Y) in LOC. Returns LOC+1.																					
FMOV(LOC,X,Y)	Puts invisible vector (X,Y) in LOC.																					
FPT(LOC(X,Y)	Puts the absolute point (X,Y) in LOC. Returns LOC+1.																					
FSET(LOC,X,Y)	Puts the invisible absolute point (X,Y) in LOC. Returns LOC+1.																					
FTXT(LOC,A,B,C,...)	Adds character of numeric value A,B,C... starting in LOC, 4 characters per LOC. Returns last LOC used +1.																					
FSKP(LOC[,L2])	For L2 less than end of graphics file, puts a skip to L2 in LOC. For L2 greater than end of graphics file but less than end of display file, increases size of graphics file to L2 LOCs, clears the text file, and enters a skip to start of text in LOC. For L2 greater than end of display file, increases size of graphics file until text file contains one line of text, maximum, clears text file, and enters a skip to its start in LOC. For L2 not included, clears the text file and enters a skip to its start in LOC. All the above return LOC+1.																					
FDIS(TYPE,INT,BLINK,LP)	Sets the mode of the next LOC entered in the graphics file. <table><tr><td>Type</td><td>0</td><td>Solid</td></tr><tr><td></td><td>1</td><td>Longdash lines</td></tr><tr><td></td><td>2</td><td>Shortdash lines</td></tr><tr><td></td><td>3</td><td>Dotdash lines</td></tr><tr><td>Intensity</td><td colspan="2">0(Dimmest) to 7(Brighest)</td></tr><tr><td>Blink</td><td colspan="2">0(Off) or 1(On)</td></tr><tr><td>Light Pen Sensitivity</td><td colspan="2">(LP) 0(Off) or 1(On)</td></tr></table>	Type	0	Solid		1	Longdash lines		2	Shortdash lines		3	Dotdash lines	Intensity	0(Dimmest) to 7(Brighest)		Blink	0(Off) or 1(On)		Light Pen Sensitivity	(LP) 0(Off) or 1(On)	
Type	0	Solid																				
	1	Longdash lines																				
	2	Shortdash lines																				
	3	Dotdash lines																				
Intensity	0(Dimmest) to 7(Brighest)																					
Blink	0(Off) or 1(On)																					
Light Pen Sensitivity	(LP) 0(Off) or 1(On)																					
FXCO(LOC)	Returns X-coordinate of datum in LOC.																					

FYCO (LOC)	Returns Y-coordinate of datum in LOC.
FLP (ARG)	For ARG equal to 0, waits for next light pen hit and returns LOC hit. For ARG not equal to 0, returns LOC last hit.
FCLR (ARG)	Clears graphics file. Returns ARG+1.

A.1.2 LPS Functions

FSAM (CHAN)	Returns value in A/D channel CHAN.
FLED (ARG)	Loads LEDS with integer part of ARG, up to 5 digits. Returns value of digits over 5 digits.
FTIC (ARG)	For ARG greater than 0, sets timer to 0 and starts clock ticking at ARG/10000 of a second. Returns value of timer, 0. For ARG=0, returns value of timer. For ARG less than 0, sets time to the value of ARG, returns 0 value when timer reaches 0.

A.2 SWITCHES

CTRL/F	Changes direction of keyboard output, either to screen or to host Monitor.
CTRL/T	Changes direction of host output, either to FOCAL or to screen.
O H	Sends FOCAL output to host.
O S	Sends FOCAL output to screen.
O T	Echoes FOCAL output on Teletype.

APPENDIX B
DECIMAL-CHARACTER EQUIVALENTS

Value*	Character	Value	Character	Value	Character
Ø	@	43	+	86	V
1 (ØA)	A	44	,	87	W
2 (ØB)	B	45	-	88	X
3 (ØC)	C	46	.	89	Y
4 (ØD)	D	47	/	90	Z
5 (ØE)	E	48	Ø	91	[
6 (ØF)	F	49	1	92	\
7 (ØG)	G	50	2	93	]
8 (ØH)	H	51	3	94	†
9 (ØI)	I	52	4	95	+
10 (ØJ)	J	53	5	96	,
11 (ØK)	K	54	6	97	a
12 (ØL)	L	55	7	98	b
13 (ØM)	M	56	8	99	c
14 (ØN)	N	57	9	100	d
15 (ØO)	O	58	:	101	e
16 (ØP)	P	59	;	102	f
17 (ØQ)	Q	60	<	103	g
18 (ØR)	R	61	=	104	h
19 (ØS)	S	62	>	105	i
20 (ØT)	T	63	?	106	j
21 (ØU)	U	64	@	107	k
22 (ØV)	V	65	A	108	l
23 (ØW)	W	66	B	109	m
24 (ØX)	X	67	C	110	n
25 (ØY)	Y	68	D	111	o
26 (ØZ)	Z	69	E	112	p
27	[	70	F	113	q
28	\	71	G	114	r
29	]	72	H	115	s
30	†	73	I	116	t
31	+	74	J	117	u
32 (ØBL)	!	75	K	118	v
33	"	76	L	119	w
34	#	77	M	120	x
35	\$	78	N	121	y
36	%	79	O	122	z
37	&	80	P	123	{
38	'	81	Q	124	}
39	(	82	R	125	~
40	)	83	S	126	■
41	*	84	T	127	
42		85	U		

*All values Modulo 128.

APPENDIX C

THE DISPLAY FILE

The display file is a contiguous area of PDP-11 core consisting of two parts: the graphics file and the text file, the boundary between them being adjustable. Figure C-1 shows the display file. The graphics file consists of 3-word cells called LOCS into which instructions that the GT40 understands are placed by user functions. The text file consists only of characters at two characters per word. The display file begins with a display stop instruction followed by a status word and an absolute point at (0,0). (See the GT40 USER'S GUIDE for a description of GT40 instructions.) The text file starts with a status word followed by an absolute point at (0,680) and then a word indicating character mode. The display file ends with a display jump to the start of the display file where the display stop resides.

Each time the display stops, FOCAL-GT/RT restarts it at the status word that follows the stop. When the display file is clear, each LOC in the graphics file consists of three display NOPS while the text file consists of null characters. The four vector and point functions place a mode word, either vector or absolute point, in word 1 of the LOC specified in the ARG list and the X and Y coordinates in words 2 and 3 respectively. FTEXT places a character mode word in word 1 of the LOC and two characters each in words 2 and 3. In all these cases, the mode word determines the type of graphics in words 2 and 3 as well as their mode if specified in FDIS.

FSKP also places a mode word in word 1 of the LOC as well as a display jump in word 2 and a jump address in word 3. In the case of FSKP, the mode word determines mode only, not the type of graphics in the LOC.

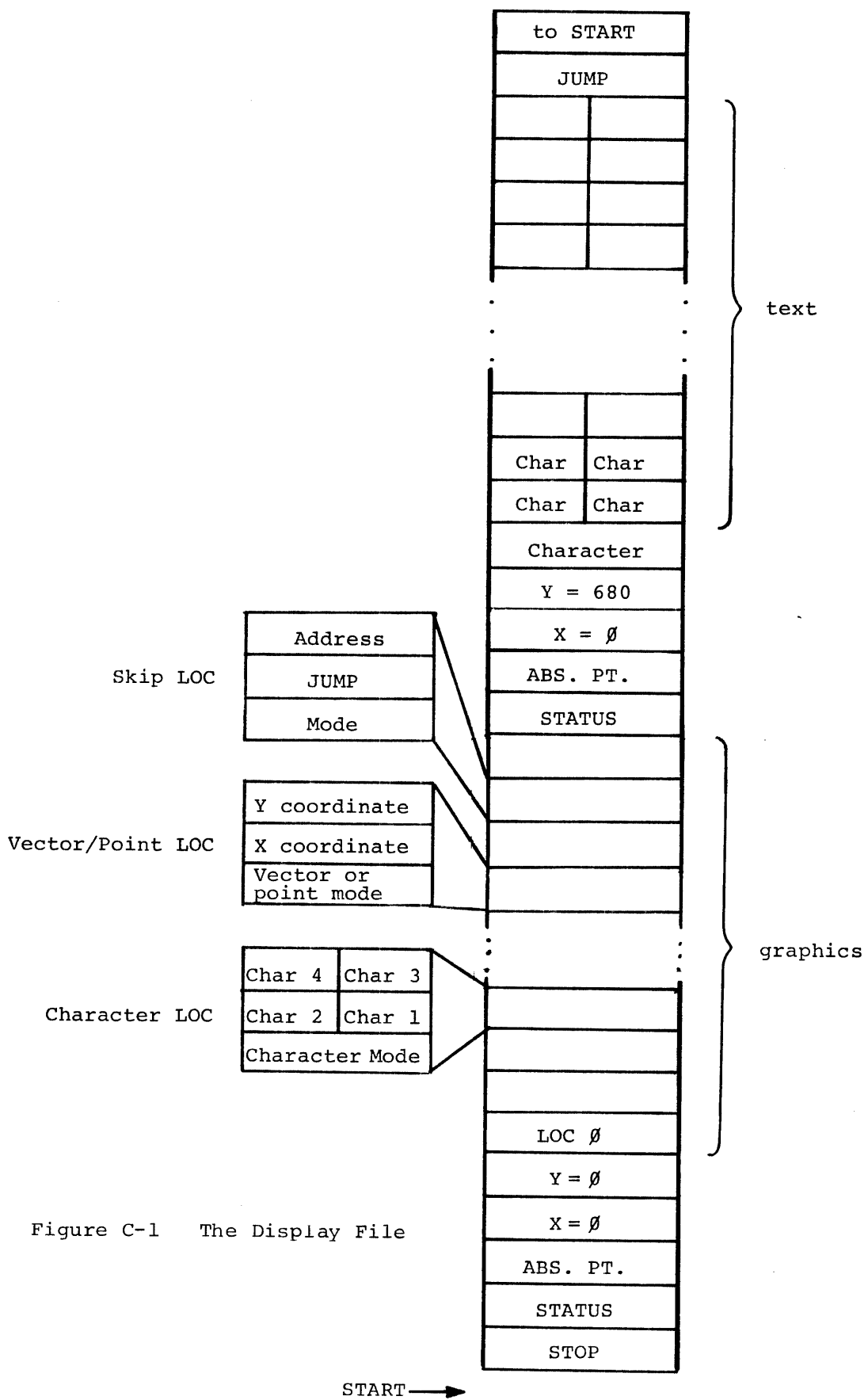


Figure C-1 The Display File

APPENDIX D

BAUD RATES GREATER THAN 300

FOCAL-GT/RT can do most I/O at Baud rates up to 2400. It can act as a terminal at these rates, and it can output data, as in the WRITE ALL command, at these rates. However, inputting a program at 2400 Baud may cause a buffer overflow, error 18, between lines. For example, if the user types:

```
.TYPE PROG.FOC
```

followed by CTRL/T and a carriage return, an error may occur. FOCAL-GT/RT will not receive the entire program. To run FOCAL-GT/RT at 2400 Baud, do FOCAL input by allowing additional time between output of each line, thus giving FOCAL-GT/RT time to process each line before beginning the next one.

Example:

```
.TECO PROG.FOC  
100<100<L-L>TL>$$
```

NOTE

In the above command line, the angle brackets constitute TECO commands. The '\$'s in the above example are echoed when the user types the ALTMODE key. After typing ALTMODE, the user types CTRL/T.

This outputs up to 100 lines of PROG.FOC, leaving time between output of each line while TECO executes an instruction that effectively does nothing.

APPENDIX E
FOCAL-11 ERROR DIAGNOSTICS

Code	Explanation
?00	manual restart from location 0 or by CTRL/C. (r)
?01	illegal line number.
?02	illegal variable or function name.
?03	unmatching parentheses.
?04	illegal command.
?05	nonexistent line number.
?06	nonexistent group or line number in DO.
?07	illegal format in SET or FOR.
?08	double or missing operators in expression.
?09	stack overflow or nonexistent device.
?10	core filled by text or command line too long. (o)
?11	core filled by variables or no room for variables. (o)
?12	exponent range greater than E+38. (o)
?13	disallowed bus address in "FX". (o)
?14	division by zero attempted. (r)
?15	attempt to exponentiate to a negative power or power too large. (r)
?16	too many characters in input data. (r)
?17	square root of negative number. (r)
?18	input buffer overflow.
(o)	operational error
(r)	a run-time error

APPENDIX F

FOCAL-11 COMMAND AND FUNCTION SUMMARY

F.1 COMMANDS

Command	Abbreviation	Example of Form	Action
ASK	A	ASK M	Request input from the current input device.
		ASK "AGE",A	Output text (AGE) and store input as variable A.
COMMENT CONTINUE	C C	COMMENT	Ignore the remainder of the line.
DO	D	DO 4.1	Execute line 4.1; return to command following DO command.
		DO 4	Execute all group 4 lines; upon completion, return to command following DO command or when a RETURN is encountered.
	D A	DO ALL	Execute entire indirect text as a subroutine.
	D v	DO var	Execute the line or group of lines defined by the variable (var).
ERASE	E	ERASE	Erase the symbol table.
		ERASE 2	Erase all group 2 lines.
		ERASE 2.1	Erase line 2.1.
	E A	ERASE ALL	Erase the entire program and clear all variables.
	E T	ERASE TEXT	Erase text only; do not erase symbol table.
FOR	F	FOR I=X,Y,Z; (commands) FOR I=X,Z; (commands)	Where the command(s) is executed at each new value of I. X=initial value of I. Y=value added to I until I is greater than Z. Y assumed=1 if omitted.
GO	G	GO	Starts indirect program at lowest numbered line number.
		GO 3.4	Transfers control to line 3.4.

Command	Abbreviation	Example of Form	Action
IF	I	IF(X)L1,L2,L3 IF(X)L1,L2;(commands) IF(X)L1;(commands)	Where X is a defined variable, a value, or an expression, followed by one to three line numbers. If X is less than zero, transfer control to the line number L1; if X is equal to zero, transfer control to the second line number, L2. If X is greater than zero, transfer control to L3. If the line number is not specified, proceed to the next sequential command.
KILL	K	KILL	Stop all I/O and reset I/O devices. Error code ?09 is printed.
MODIFY	M	MODIFY 1.15	Enable editing of line 1.15.
OPERATE			Selects the input and/or output device for such commands as TYPE and ASK.
	O T	OPERATE T	Select Teletype printer.
	O K	OPERATE K	Select Teletype keyboard for input.
	O P	OPERATE P	Select high-speed paper tape punch for output.
	OR	OPERATE R	Select high-speed paper tape reader for input.
	O RP	OPERATE RP	Select both high-speed reader and punch for I/O.
	O TK	OPERATE TK	Select both Teletype keyboard and printer for I/O.
	O L	OPERATE L	Select line printer for output.
QUIT	Q	QUIT	Return control to the user (command mode).
RETURN	R	RETURN	Terminate DO subroutines, returning to the original sequence.
SET	S	SET A=5/B*C	Perform arithmetic assignment. The variable on the left side of "=" is set equal to the value of the expression on the right.
TYPE	T	TYPE A+B-C	Evaluate expression and type "=" followed by result in current output format.

Command	Abbreviation	Example of Form	Action
		TYPE A-B,C/E	Compute each expression and type the resultant values.
		TYPE "TEXT STRING"	Type text. May be followed by ! to generate carriage return/line feed, or # to generate carriage return.
WRITE	W	WRITE	Type out the entire indirect program.
	W A	WRITE ALL	
		WRITE 1	Type out all group 1 lines.
		WRITE 1.1	Type out line 1.1.
XECUTE	X	XECUTE FSBR(5,ARG)	Call functions without need for a dummy SET statement.
?(TRACE)		GO?	Starts at lowest numbered line and traces entire indirect program until another ? or an error is encountered, or until completion of program.

F.2 FUNCTIONS

Function	Form	Action
FABS	FABS (expression)	Returns absolute (positive value of expression.
FADC	FADC (gain, channel)	Provides access to A/D channels.
FCHR	FCHR (arg)	Accepts and/or prints ASCII codes.
FCLK	FCLK ()	Returns the value of the time elapsed.
FCOS	FCOS (angle)	Calculates the cosine of a specified angle in radians.
FITR	FITR (expression)	Provides the integer part of a number.
FRAN	FRAN () FRAN (1)	Generates a random value between -1 and 1.
FSBR	FSBR (group, arg)	Calls program group specified as a subroutine.
FSGN	FSGN (expression)	Returns -1 if expression <0, 0 if expression =0, 1 if expression >0.
FSIN	FSIN (angle)	Calculates the sine of the specified angle in radians.
FSQT	FSQT (expression)	Computes square root of expression.
FX	FX (func, UNIBUS-address, data)	Controls additional device options or nonstandard peripherals or references to core storage.

APPENDIX G FOCAL-GT SYMBOL TABLE

MAIN: MACDLX 622(622)-1 30-APP-73 13:42 PAGE 06-3
FOCGT, 002 SYMBOL TABLE

ABSFF	006556	AC	=%0000001	ADCS	= 176770	ADDB	= 176772
ADDOHR	225114	ADOFF	006230	ADIN	= 170400	AFIX	005344
AGO	005372	AH	=%0000004	AL	=%0000005	ALGZA	006506
ALGZR	006522	ALIGNA	006270	ALIGNB	006256	ALIST	= 001433
ALL	= 200001	ALTAKA	006324	ARGNXT	004476	ARO	= 005330
ASCII	223427	ASK	005244	ASPACE	005340	ATAKE	005304
ATLIST	= 001344	AXIN	006162	AXOUT	=%000003	BDIVX	006476
BE	001616	BEGIN	= 001100	BELL	= 000207	BELOW	002426
BH	=%000002	BL	=%0000003	BOTTOM	001652	RUFBEQ	= 001330
BUFFUL	002202	BUFR	006164	CCFLG	001601	CFRS	001654
CHAR	=%000004	CHIN	002710	CHINX	002756	CINT	002256
CLCU	010210	CLKADR	= 000134	CLKIN	= 100404	CLKT	010274
CLOSE	= 222626	CLRLOP	004426	COCO	004412	COMBUF	= 001461
COMLST	= 001250	COPYLN	= 104630	COPYLX	004640	CR	= 000015
CRLF	= 005315	DBOT	= 003016	DCHR	= 100000	DDSIZE	= 012576
DEBG	001610	DECONV	007702	DECON1	007722	DECON2	010002
DECOY	007764	DELETE	= 000004	DETN	000006	DFILE	005172
DIFL	006600	DIGTSA	002654	DIGTST	= 104444	DIRECT	= 000000
DIVFF	006564	DIVZER	006642	DJMP	= 160000	DNOP	= 160000
DO	004066	DOCONT	004204	DOER	004220	DOGR	004106
DOGRP1	004124	DOGRP2	004120	DOXIT	004210	DPT	= 114000
DSIZE	= 001625	DSTP	= 173400	DUNASC	003454	DVEC	= 110000
DXIT	007540	E	=%0000002	ECHOLS	= 001450	ECOLOG	003252
EDITIT	224714	EFUN	004614	EFUN3	004656	ELPAR	004504
EMDIR	005672	EMIME	005704	EMIND	005726	FMREL	005744
EMT	= 007000	EMTHR	005736	EMTO	005716	FNAB	= 003124
ENDFLG	004670	ENIT2	003070	ENUM	004474	ENUM2	004604
EPAR	004460	EPAR2	004510	EPURE	004436	FRASE	003744
ERASET	= 104642	ERASEV	= 104636	ERUR	004010	ERL	004002
ERROR	= 104400	ERRORC	003502	ERR2	001752	ERT	003770
ERTX	004454	ERV	003764	ERVC	004034	ERVX	005100
ETERM	004374	ETERMN	004566	ETERM1	004544	ETERM2	004406
EVAL	004520	EVALU	= 104660	EVALUX	004664	EVALX	= 104660
F	=%0000003	FCDCE	= 007200	FCONT	004306	FCONT2	004312
FCOS	010424	FCOS2	010466	FCOS4	010526	FEMT	005472
FERR0	006732	FIGOE	007654	FIGO1	007566	FIGO2	007636
FIGO3	007664	FIGO4	007674	FINGR	004254	FINDLN	= 104624
FINDN	002574	FINDO	002622	FINDX	002566	FINERR	004252
FINFIN	004266	FISW	001626	FLARG	001012	FLIMIT	004300
FLIST1	001336	FLIST2	001330	FLOAT	004340	FLOSGN	007034
FLOUT	007450	FLTD0	005556	FLTD01	005544	FLTONE	010560
FLTX	005624	FLTZER	003530	FNADDR	006146	FNTABL	= 001166
FOG01	007042	FOG02	007072	FOG03	007110	FOG04	007124
FOR	004222	FPABS	= 007073	FPADD	= 007010	FPDIV	= 007030
FPGET	= 007000	FPINT	= 007071	FPMUL	= 007040	FPNEG	= 007074
FPNOR	= 007070	FPPOW	= 007050	FPPUT	= 007060	FPRINT	= 007076
FPRNT	007356	FPRNTP	007376	FPSGN	= 007072	FPSUR	= 007020
FPURE	005522	FP1	004332	FP6	003576	FREAD	= 007075
FROM	= 000003	FSIN	010416	FSIZE	= 002154	FTRY	007434
FZER	= 007077	GALL	002554	GALL1	002522	GEND	= 004144
GETARG	004674	GETC	= 104614	GETCH	= 104614	GETF	006114
GETLN	= 104622	GETLNK	002454	GETVAR	004704	GETX	003232
GGROUP	002560	GOF0C	003112	GONE	002146	GOTO	003436
GRALL	002320	GROOVY	= 104650	GROVX	002172	GROVZ	002650

MAIN, MACDLX 622(622)-1 30-APR-73 13142 PAGE 66-4
FOCGT,002 SYMBOL TABLE

GSERCH	004730	GSI	004776	GTESTW	002512	GTSTAT	= 172002	HIBITS	024660	IGNORE	023152	INDEV	001632	INIT2	003072	INTF	006332	INTY	006366	IOFIX	010304	IOO	010334	K	023016	KINT2	010136	KKSIZE	= 037770	LEDOUT	= 170402	LINCNT	001630	LIST3	= 001456	LOC60	024176	LOSE	003422	LPADR	= 000324	LPS	= 177514	MAKSUR	023440	MORNOP	024156	NALPHA	= 000020	NOBLI	023752	NOLPS	023204	NORF	006036	NORX	006204	NOTBAK	023612	NOTOUT	023650	NUMF	= 000054	OPEN	= 024646	OUTCH	= 104610	OUTX	003064	PACKC	= 104616	PBAR	003176	PC2	003504	POSCO	024300	POWF1	006644	PRINT	= 104400	PRIN2A	002626	PRNTS	007474	PROCES	003452	PTEN	007164	PUTTXT	023402	PWRUP	006002	RESET0	024504	RETURN	003516	RUBX	003210	SCONL	003566	SFINO	003610	SIGN	006734	SKPNON	= 104654	GSI	= 002004	GTAKE	005170	GTPC	= 172000	GTX	= 172004	HORD	001620	IMMED	= 000005	INFILE	023670	INLIST	= 001366	INTG	006340	INTZ	006356	LOGO	= 010362	IPTR	= 000001	KIN	001636	KINT3	010136	KSIZE	= 020000	LF	= 000012	LINENO	001624	LIST6	= 001453	LOCSET	024164	LOSE1	003426	LPBSIZ	= 000000	LPSSIZ	= 000000	MDP0	006434	MORTXT	023370	NEGFF	006562	NOEKO	003106	NOP	006000	NORMD	006072	NOS	010070	NOTEXT	023606	NOTT	023066	ONE	= 000200	OPERR	004572	OUTCO	024056	OUTY	003042	PACKX	003136	PC	= 000200	PERR	004672	POSOUT	024050	POWS	006670	PRINTA	002316	PRNTF	007012	PRNT1	007532	PROGIO	010336	PTR	= 000002	PWRDWN	005752	READC	= 104606	RETLOC	024110	ROOTGO	010244	RUBX2	003214	SCONT	003564	SFOUND	003620	SIGND	006460	SKPNOX	012372	GSWIP	005060	GTEST	005160	GTRY	005134	GT	= 172006	IF	003074	INADDR	005662	INIT	011030	INPTR	004072	INTO	= 000003	INV	003014	IOLIST	= 010052	JMS	= 103014	KINT	010110	KINT4	010042	LASTL	005006	LFCHAR	005064	LINERR	002314	LOC	= 000000	LOC0	005002	LOSE2	003014	LPINT	004526	LSPR	001602	MODE	004062	MULFF	006426	NEGX	006470	NOLP	003770	NOPLOP	004150	NORMZ	006072	NOSEND	004074	NOTF	003044	NOTYET	003032	ONPIC	004010	OPNEXT	004036	OUTDEV	001634	OUTZ	003012	PARTSA	005006	PCF	001604	PLUGCO	003344	POWDO	006700	PPS	= 177554	PRINTC	= 104004	PRNTLN	= 104026	PRNT2	007522	PRS	= 177550	PUTCO	004264	PWREGS	006020	READF	007542	RETNEW	004012	RTPAR	= 000211	R5	= 000005	SERR	003624	SGNF	006032	SILENT	005026	SORTR	002006	GS0	004764	GTESTN	002562	GTRY2	005144	GT40	= 000000	IGNOR	002114	INCH	= 104612	INITD	023166	INPUTN	002204	INTX	006362	INV0	024562	IOPATC	= 010400	JUST0	023456	KINT1	010132	KITH	010074	LEAST	023634	LFS	024674	LISTGO	= 001406	LOCGET	024204	LORD	001622	LOST	003370	LPLOC	024656	LTEERO	007414	MODIFY	003552	MULZ	006166	NEWLOC	024200	NOLPCO	024614	NORESP	023146	NORM2	006046	NOSHOW	024734	NOTFUL	025144	NO18	024476	OPADDR	006126	OUT	003002	OUTW	002764	P	= 000000	PARTST	= 104646	PC1	003524	POPJ	= 000207	POWF	006650	PRESET	= 170406	PRINT2	= 104642	PRNTP	007506	PROC	003454	PSCAN	003444	PUTF	005630	PWRON	006034	REL	= 000006	REYNOW	024076	RUBIT	003200	SCHAR	003572	SET	= 004222	SGN1	006536	SKPLPR	= 104652	SORTC	= 104602
--------	--------	-----	--------	--------	--------	--------	----------	--------	--------	--------	--------	-------	--------	-------	--------	------	--------	------	--------	-------	--------	-----	--------	---	--------	-------	--------	--------	----------	--------	----------	--------	--------	-------	----------	-------	--------	------	--------	-------	----------	-----	----------	--------	--------	--------	--------	--------	----------	-------	--------	-------	--------	------	--------	------	--------	--------	--------	--------	--------	------	----------	------	----------	-------	----------	------	--------	-------	----------	------	--------	-----	--------	-------	--------	-------	--------	-------	----------	--------	--------	-------	--------	--------	--------	------	--------	--------	--------	-------	--------	--------	--------	--------	--------	------	--------	-------	--------	-------	--------	------	--------	--------	----------	-----	----------	-------	--------	------	----------	-----	----------	------	--------	-------	----------	--------	--------	--------	----------	------	--------	------	--------	------	----------	------	----------	-----	--------	-------	--------	-------	----------	----	----------	--------	--------	-------	----------	--------	--------	-------	--------	--------	----------	--------	----------	------	--------	--------	--------	-------	--------	-------	--------	-----	--------	-------	--------	-----	--------	--------	--------	------	--------	-----	----------	-------	--------	-------	--------	------	--------	-------	--------	----	----------	------	--------	--------	--------	------	--------	--------	--------	-------	--------	-------	--------	--------	--------	-----	----------	--------	--------	-------	----------	--------	--------	--------	--------	-------	--------	-------	--------	--------	--------	-------	--------	--------	--------	-------	--------	-------	--------	------	--------	----	----------	----	--------	--------	--------	------	--------	-------	--------	------	----------	-----	--------	--------	----------	-----	----------	------	--------	-------	--------	-------	--------	--------	--------	--------	--------	-----	----------	------	--------	-------	--------	-------	--------	------	--------	------	--------	-------	--------	------	--------	------	--------	--------	--------	-------	--------	--------	--------	------	--------	--------	--------	-------	--------	--------	--------	--------	--------	------	--------	--------	--------	-----	--------	--------	--------	-------	--------	-----	----------	--------	----------	--------	----------	-------	--------	-----	----------	-------	--------	--------	--------	-------	--------	--------	--------	-------	----------	----	----------	------	--------	------	--------	--------	--------	-------	--------	-----	--------	--------	--------	-------	--------	------	----------	-------	--------	------	----------	-------	--------	--------	--------	------	--------	------	--------	--------	----------	-------	--------	-------	--------	------	--------	-------	--------	-----	--------	--------	----------	--------	--------	------	--------	------	--------	-------	--------	--------	--------	--------	--------	------	--------	--------	--------	--------	--------	--------	--------	-------	--------	--------	--------	--------	--------	------	--------	--------	--------	-----	--------	------	--------	---	----------	--------	----------	-----	--------	------	----------	------	--------	--------	----------	--------	----------	-------	--------	------	--------	-------	--------	------	--------	-------	--------	-----	----------	--------	--------	-------	--------	-------	--------	-----	----------	------	--------	--------	----------	-------	----------

SORTD	222444	SORTJ	= 14600	SOUND	022424	SOX	002420
SP	= 002000A	SPECIA	= 001445	SPNOR	= 124434	SPNORX	003126
SPNXT	203124	SQUEND	014252	SQX	010450	SRETN	002226
SRNLST	= 001400	STACK	= 002000	STACKO	021706	START	= 104632
STARTV	001644	STARTX	002034	STATUS	= 177576	STILON	024170
STLIN	003626	STOP	001704	STOPIT	024650	STPADR	= 000320
STPFLG	224664	STPINT	224634	STRING	= 020030	SUREND	= 024676
SURFF	006222	SWITCH	= 001631	TASK	= 104656	TASKX	005412
TASK4	005414	TCLEAR	024362	TCRLF	025404	TCRLF2	025464
TDUMP	010564	TDUMP1	010570	TDUMP2	010612	TDUMP3	010600
TDUMP4	010654	TEMP	= 007000	TEN	007102	TENADR	= 000300
TENIN	= 175610	TENINT	023116	TENOUT	= 175614	TERM	= 000000
TERMS	= 001412	TERSIZ	= 000150	TESTC	= 124620	TESTX	002334
TEST10	023074	TEX	002366	TEXT	011772	THISLN	001606
THROUG	= 000004	TINTR	005430	TJUMPS	024144	TKS	= 177560
TLIST	= 001441	TOF	007260	TOFOCL	023457	TOG	007246
TOHOST	023154	TOOBIG	024242	TOP	001450	TOR	007310
TOS	007324	TOT	007276	TPR	003434	TPR1	003540
TPS	= 177564	TQUOT	005442	TRAPH	021714	TSIZE	= 001000
TSTERR	024472	TSTXT	024676	TTSIZE	= 026000	TXDIS	024720
TXTDUN	025156	TXTHAN	025000	TXJMP	023442	TXTPTR	024666
TYPE	005224	UTX	003220	VEHR	004670	VIS	023304
WAYOUT	023654	WGO	003742	WHIPV	004770	WHOOPS	001640
WRED	003734	WRIG	003724	WRITE	003446	WRITEL	003670
WRITE2	003654	WTESTG	003712	XABS	006476	XADC	011070
XCHARG	010772	XCHMO	010756	XCHR	010760	XCHR1	011004
XCLR	024130	XDELET	003272	XDIS	023400	XDOX	003336
XDTHIS	003360	XD3	003276	XECUTE	023444	XEL	011322
XEM	011210	XER	011316	XERR	011426	XET	011250
XEX	011134	XEXIT	011302	XEX2	011452	XEX2A	011214
XEX3	011226	XEX4	011240	XFCLK	010472	XFSBR	011026
XITR	006406	XI33	010022	XI33X	010444	XLP	024064
XMOV	023254	XOUT	010052	XPRNTL	023400	XPT	023264
XPTR	= 000002	XRAN	010704	XROL	010726	XSET	023274
XSGN	006402	XSKP	023514	XSQT	010454	XYSTLP	002636
XTXT	223364	XVEC	023244	XXCO	024000	XYCO	024004
Z	= 020724	ZERODM	006370		= 023160		

ERRORS DETECTED: 0

APPENDIX H FOCAL-RT SYMBOL TABLE

,MAIN, MACDLX 622(622)=1 30-APR-73 13:40 PAGE 66-3
FOCGT,002 SYMBOL TABLE

ABAD	025376	ABSFF	006556	AC	=%000001	ADCS	= 176770
ADDB	= 176772	ADGHR	027724	ADFF	006230	ADIN	= 170400
AFIX	005344	AGO	005372	AH	=%000004	AL	=%000005
ALGZA	006506	ALGZH	006522	ALIGNA	006270	ALIGNB	006256
ALIST	= 021433	ALL	= 000001	ALTAKA	006324	ARGNXT	004476
ARO	005330	ASCII	026230	ASK	005244	ASPACE	005340
ATAKE	005304	ATLIST	= 001344	AXIN	001642	AXOUT	=%000003
BDIVX	026476	BE	001616	BEGIN	= 001100	BELL	= 000007
BELOW	027072	BH	=%000002	BL	=%000003	BOYOM	001652
BUFFEG	= 011330	BUFFUL	002202	HUFR	001644	CCFLG	001601
CFRS	001654	CHAR	=%000004	CHIN	002710	CHINX	002756
CINI	010256	CLCU	010210	CLKADR	= 000134	CLKIN	= 170404
CLKINT	025570	CLKLOP	025552	CLKT	010274	CLOSE	= 022626
CLRLOP	027236	COCO	026620	COMBUF	= 001461	COMLST	= 001250
COPYLN	= 104630	COPYLX	004040	CR	= 000015	CRLF	= 005015
DBOT	= 021174	DCHR	= 100000	DDSIZE	= 007774	DEBG	001610
DECONV	007702	DECON1	007722	DECON2	010002	DECOY	007764
DELETE	= 000104	DETN	010006	DFILE	027774	DIFL	006600
DIGTSA	002654	DIGIST	= 104644	DIRECT	= 000000	DIVFF	006566
DIVZER	006642	DJMP	= 100000	DNOP	= 164000	DO	004066
DOCONT	004204	DOEN	004220	DOGR	004106	DOGRP1	004124
DOGRP2	004120	DOXIT	004210	DPT	= 114000	DSIXF	= 001252
DSTP	= 173400	DTIME	025604	DUNAD	025356	DUNASC	026264
DVEC	= 110000	DXIT	007540	E	=%000002	EGHLS	= 001450
EGOLOG	003252	EDITIT	027524	EFUN	004614	FFUNS	004656
ELPAR	004504	EMDIR	005672	EMIME	005704	FMIND	005726
EMREL	005744	EMT	= 007000	EMTHR	005736	FMT0	005716
ENAB	= 003124	ENDFLG	027500	ENIT2	003070	FNUM	004974
ENUM2	004604	EPAR	004460	EPAR2	004510	FPUKE	004436
ERASE	003744	ERASET	= 104640	ERASEV	= 104636	FRG	004010
ERL	004002	ERROR	= 104400	ERRORC	003502	FRR2	001752
ERT	003770	ERTX	004050	ERV	003764	FRVC	004034
ERVX	005100	ETERM	004374	ETERMN	004566	ETERM1	004544
ETERM2	004406	EVAL	004520	EVALU	= 104600	EVALUX	004664
EVAL,X	= 104660	F	=%000003	FCODE	= 007200	FCONT	004306
FCONT2	004312	FCOS	010426	FCOS2	010466	FCOS4	010526
FEMT	005472	FERR0	006732	FIGOE	007654	FIG01	007566
FIG02	007636	FIG03	007664	FIG04	007674	FINGR	004254
FINDLN	= 104624	FINDN	002574	FIND0	002622	FINDX	002566
FINEER	004252	FINFIN	004266	FISW	001626	FLARG	001612
FLIMIT	004300	FLIST1	001336	FLIST2	001350	FLOAT	027150
FLOSGN	007034	FLOUT	007450	FLTD0	005556	FLTD01	005544
FLTONE	010560	FLTX	005624	FLTZER	003530	FNADDR	006146
FNTABL	= 001166	FOG01	007042	FOG02	007072	FOG03	007110
FOG04	007124	FOR	004222	FPABS	= 007073	FPA00	= 007010
FPDIV	= 007030	FPGET	= 007000	FPINT	= 007071	FPMUL	= 007040
FPNEG	= 007074	FPNOR	= 007070	FPPOW	= 007050	FPPUT	= 007060
FPRINT	= 007076	FPRNT	007356	FPRNTP	007376	FPSGN	= 007072
FPSUB	= 007020	FPUKE	005522	FP1	027142	FP10	025444
FP6	026406	FREAD	= 007075	FROM	= 000003	FSIN	010416
FSIZE	= 006600	FTRY	007434	FZER	= 007077	GALL	002554
GALL1	002522	GEN0	= 026754	GETARG	004674	GETC	= 104614
GETCH	= 104614	GETF	006114	GETLN	= 104622	GETLNx	002454
GETVAR	004704	GETX	003232	GGROUP	002560	GOF0C	025704

GONE	002146	GOTO	003436	GRALL	026130	GROOVY	= 104050
GROVX	002672	GROVE	002650	GSECH	004730	GSIKE	= 002014
GSWIP	005060	GS0	004764	GS1	004776	GTAKF	005170
GTEST	005160	GTESTN	002562	GTESTW	002512	GTPC	= 172000
GTRY	005134	GTRY2	005144	GTESTAT	= 172002	GTX	= 172004
GTY	= 172006	GT40	= 000000	HIBITS	027470	HORO	001020
IF	003374	IGNOR	002114	IGNORE	025744	IMMED	= 000003
INADDR	005662	INCH	= 104612	INDEV	001632	INFILE	026500
INIT	011330	INITD	025760	INIT2	003072	INLIST	= 001366
INPTR	027502	INPUTN	002204	INTF	006330	INTG	006340
INTO	= 000003	INTX	006362	INTY	006366	INTZ	006356
INV	026124	INV0	027372	IOFIX	010304	IOGO	= 010362
IOLIST	= 010352	IOPATC	= 010400	IOO	010334	IPTR	= 000001
JMS	= 103414	JUST0	026266	K	025610	KIN	001636
KINT	010110	KINT1	010132	KINT2	010136	KINT3	010136
KINT4	010142	KITH	010074	KKSIZE	= 037770	KSIZE	= 020000
LASTL	027716	LEAST	026444	LEDOUT	= 170402	LF	= 000012
LFCHAR	027674	LFS	027504	LINCNT	001630	LINENO	001024
LINERR	002314	LISTGO	= 001406	LIST3	= 001456	LIST6	= 001453
LOC	= %000000	LOCGET	027014	LOGGO	027006	LOCSET	026774
LOC0	030004	LORD	001622	LOSE0	003422	LOSE1	003426
LOSE2	003414	LOST	003370	LPADR	= 000324	LPSIZE	= 001762
LPINT	027336	LPLOC	027466	LPREG	025402	LPS	= 177514
LPSBUF	= 021174	LPSSIZ	= 004414	LSPR	001602	LYZERO	007414
MAKSUR	026250	MOP0	006434	MODE	027472	MODIFY	003552
MORNOP	026766	MORTXT	026200	MULFF	006426	MULZ	006166
NALPHA	= 000000	NEGFF	000562	NEGNUM	025426	NEGX	006470
NEWLOC	027010	NOAD	025312	NOAROV	025602	NORLI	026562
NOEKO	025700	NOLP	026600	NOLPCO	027424	NOLPS	026014
NOP	006000	NOPLOP	026760	NORESP	025740	NORF	006036
NORMD	006072	NORMZ	006172	NORM2	006046	NORX	006204
NOS	010070	NOSEND	027604	NOSET	025544	NQSH	025242
NOSHOW	027544	NOTBAK	026422	NOTEXT	026416	NOTF	025036
NOTFUL	027754	NOTOUT	026460	NOTT	025660	NOTYET	026242
NO18	027306	NUMF	= 000070	NUMLOP	025436	ONE	= 000200
ONPIC	027420	OPADUR	006126	OPEN	= 024646	OPERR	004572
OPNEXT	004536	OUT	003002	OUTCH	= 104610	OUTCO	026666
OUTDEV	001634	OUTW	002764	OUTX	003064	OUTY	003042
OUTZ	003012	P	= %000000	PACKC	= 104616	PACKX	003136
PARTSA	005206	PARIST	= 104646	PBAR	003176	PC	= %000007
PCF	001604	PC1	003524	PC2	003504	PERR	004672
PLUGCO	026154	POPJ	= 000207	POSCO	027110	POSOYT	026660
POWDO	006700	POWF	006650	POWF1	006644	POWS	006670
PPS	= 177554	PRESET	= 170406	PRINT	= 104400	PRINTA	002316
PRINTC	= 104604	PRINT2	= 104642	PRIN2A	002626	PRNTF	007012
PRNTLN	= 104626	PRNTP	007506	PRNTS	007474	PRNT1	007532
PRNT2	007522	PROC	003454	PROCES	003452	PROGIO	010336
PRS	= 177550	PSCAN	003444	PTEN	007064	PTR	= %000002
PUTCO	027074	PUTF	005630	PUTTXT	026212	PWRDWN	005752
PWREGS	006020	PWRUN	006034	PWRUP	006002	PEADC	= 104006
READF	007542	REL	= 000006	RESETO	027314	REUAT	025366
RETLOC	026720	RETNEW	027122	REYNOW	026706	RETURN	003516
ROOTGO	010244	RT	= 000000	RTPAR	= 000211	RUBIT	003200
RUBX	003210	RUBX2	003214	R5	= %000005	SAMLOP	025304

SCHAR	003572	SCONL	003566	SCONT	003564	SERN	003624
SET	= 004222	SFIND	003610	SFOUND	003620	SGNF	006532
SGN1	006536	SIGN	006734	SIGND	006460	SILENT	027636
SIZOK	025254	SKPLPR	= 104652	SKPNON	= 104654	SKPNON	002372
SORTB	002406	SORTC	= 104602	SORTD	002440	SORTJ	= 104600
SOUND	002424	SOX	002420	SP	=X000006	SPECIA	= 001445
SPNOR	= 104634	SPNORX	003126	SPNXT	003124	SQUEND	010252
SQX	010250	SRETN	002226	SRNLST	= 001400	STACK	= 000000
STACKO	001706	START	= 104632	STARTV	001646	STARTX	002034
STATUS	= 177776	STILON	027000	STLIN	003626	STOP	001704
STOPT	027460	STPADR	= 000320	STPFLG	027474	STPINT	027444
STRING	= 000030	SUBEND	= 027506	STUFF	006222	SWITCH	= 001631
TASK	= 104656	TASKX	005412	TASK4	005410	TCLEAR	027172
TCRLF	005404	TCRLF2	005464	TDUMP	010594	TDUMP1	010570
TDUMP2	010612	TDUMP3	010600	TDUMP4	010656	TEMP	=X000000
TEN	007102	TENADR	= 000300	TENIN	= 175610	TENINT	025710
TENOUT	= 175614	TERM	= 000000	TERMS	= 001412	TERSIZ	= 000150
TESTC	= 104620	TESTX	002334	TEST10	025606	TEX	002366
TEXT	031772	THISLN	001606	THROUG	= 000004	TINTR	005430
TJUMPS	026754	TKS	= 177560	TLIST	= 001441	TOF	007260
TOFOCL	025751	TOG	007246	TOHOST	025750	TOOBIG	027052
TOP	001650	TOR	007310	TOS	007324	TOT	007270
TPR	003534	TPR1	003540	TPS	= 177564	TQUOT	005442
TRAPH	001714	TSEE	= 001000	TSTERR	027302	TSTXT	027506
TYSIZE	= 006000	TXDIS	027530	TXDUN	027706	TXTHAN	027610
TXJMP	026452	TXTPTR	027476	TYPE	005224	UTX	003220
VERR	004670	VIS	026114	WAYOUT	026464	WGO	003742
WHIPV	004770	WHOOPS	001640	WRED	003730	WRIG	003724
WRITE	003646	WRITEL	003670	WRITE2	003654	WTFSTG	003712
XABS	006376	XADC	011070	XCHARG	010772	XCHMO	010756
XCHR	010760	XCHR1	011004	XCLR	026740	XDELET	003272
XDIS	026510	XDOX	003336	XDTHIS	003360	XDS	003276
XECUTE	003544	XEL	011322	XEM	011210	XER	011316
XERR	011326	XET	011250	XEX	011134	XEXIT	011302
XEX2	011152	XEX2A	011214	XEX3	011226	XEX4	011240
XFCLK	010272	XFSBR	011026	XITH	006406	XI33	010022
XI33X	010044	XLED	025404	XLP	026674	XMOV	026064
XOUT	010052	XPRNTL	003100	XPT	026074	XPTR	= 000002
XKAN	010704	XROL	010726	XSAM	025142	XSET	026104
XSGN	006402	XSKP	026324	XSET	010154	XTIC	025516
XSTLP	002636	XTXT	026174	XVEC	026054	XXCO	026610
XYCO	026614	Z	= 020724	ZERODM	006370	.	= 025732

ERRORS DETECTED: 0

HOW TO OBTAIN SOFTWARE INFORMATION

Announcements for new and revised software, as well as programming notes, software problems, and documentation corrections, are published by Software Information Service in the following newsletters.

DIGITAL Software News for the PDP-8 and PDP-12
DIGITAL Software News for the PDP-11
DIGITAL Software News for 18-bit Computers

These newsletters contain information applicable to software available from DIGITAL'S Software Distribution Center. Articles in DIGITAL Software News update the cumulative Software Performance Summary which is included in each basic kit of system software for new computers. To assure that the monthly DIGITAL Software News is sent to the appropriate software contact at your installation, please check with the Software Specialist or Sales Engineer at your nearest DIGITAL office.

Questions or problems concerning DIGITAL'S software should be reported to the Software Specialist. If no Software Specialist is available, please send a Software Performance Report form with details of the problems to:

Digital Equipment Corporation
Software Information Service
Software Engineering and Services
Maynard, Massachusetts 01754

These forms, which are provided in the software kit, should be fully completed and accompanied by terminal output as well as listings or tapes of the user program to facilitate a complete investigation. An answer will be sent to the individual, and appropriate topics of general interest will be printed in the newsletter.

Orders for new and revised software manuals, additional Software Performance Report forms, and software price lists should be directed to the nearest DIGITAL field office or representative. USA customers may order directly from the Software Distribution Center in Maynard. When ordering, include the code number and a brief description of the software requested.

Digital Equipment Computer Users Society (DECUS) maintains a user library and publishes a catalog of programs as well as the DECUSCOPE magazine for its members and non-members who request it. For further information, please write to:

Digital Equipment Corporation
DECUS
Software Engineering and Services
Maynard, Massachusetts 01754

READER'S COMMENTS

Digital Equipment Corporation maintains a continuous effort to improve the quality and usefulness of its publications. To do this effectively we need user feedback--your critical evaluation of this document.

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

How can this document be improved?

How does this document compare with other technical documents you have read?

Job Title _____ Date: _____

Name: _____ Organization: _____

Street: _____ Department: _____

City: _____ State: _____ Zip or Country _____

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

Digital Equipment Corporation
Software Information Service
Software Engineering and Services
Maynard, Massachusetts 01754

