DEC-11-GGTMA-B-D

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

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

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.

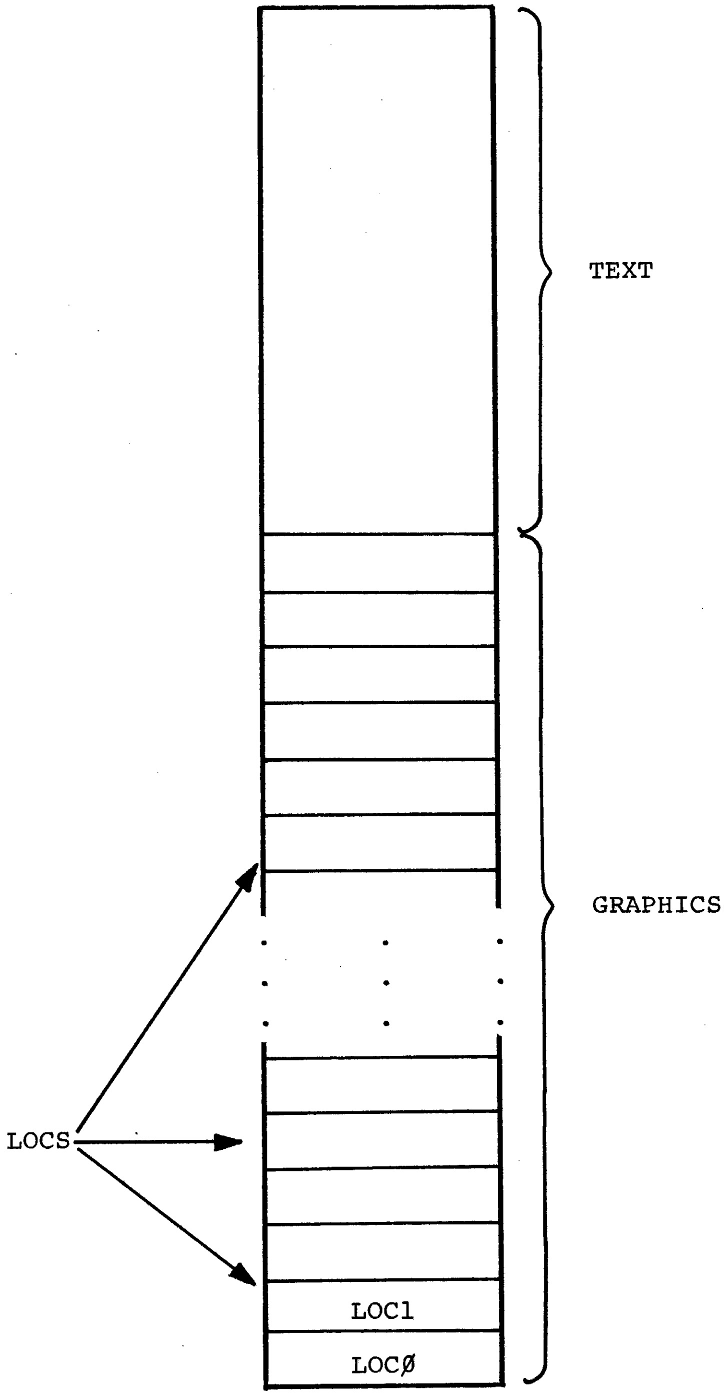


Figure 1-1   The Display File

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. FTXT 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 FTXT 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=FTXT(LOC,0N,05,0W,0BL,0L,05,0T,0T,05,0R,58)
     1.3  ASK "LETTER" L
     1.4  X  FTXT(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 sensitivite 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(chanl[,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 asychronous 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.

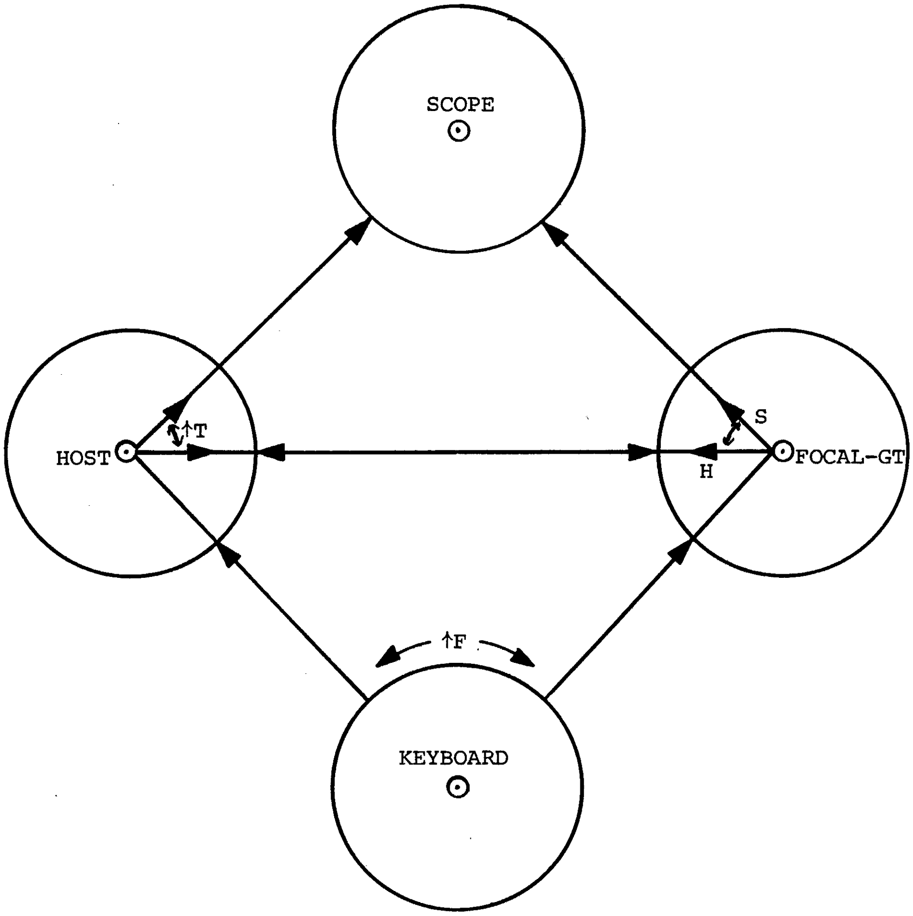


Figure 3-1 Flow of Data

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 commmand, the output from the TYPE commmand 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 compute to buffer command input since FOCAL outputs these commands as soon as it reaches the TYPE commmand. 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
```

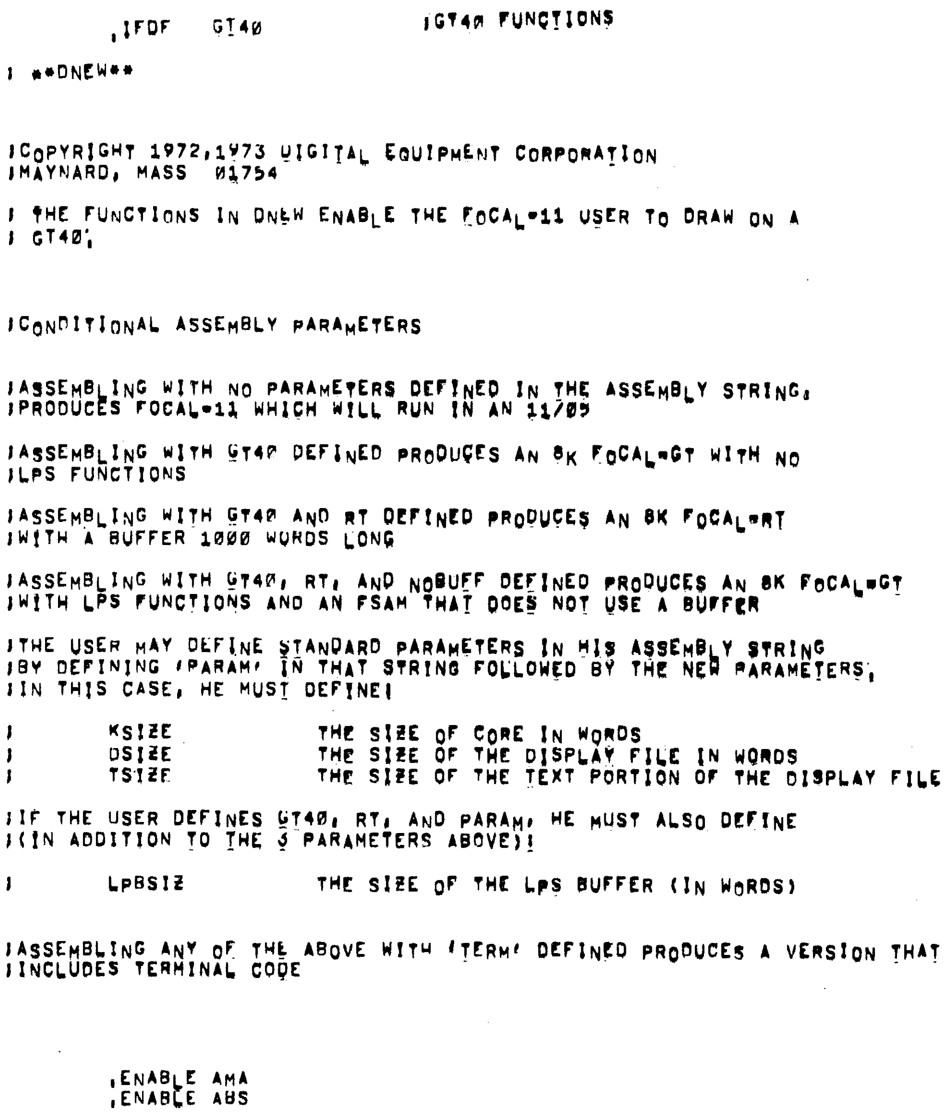
This produces FOCAL-11 which does not make use of the display.

```
FOCGT.BIN/I/EQ:GT40←FOCGT.002
```

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

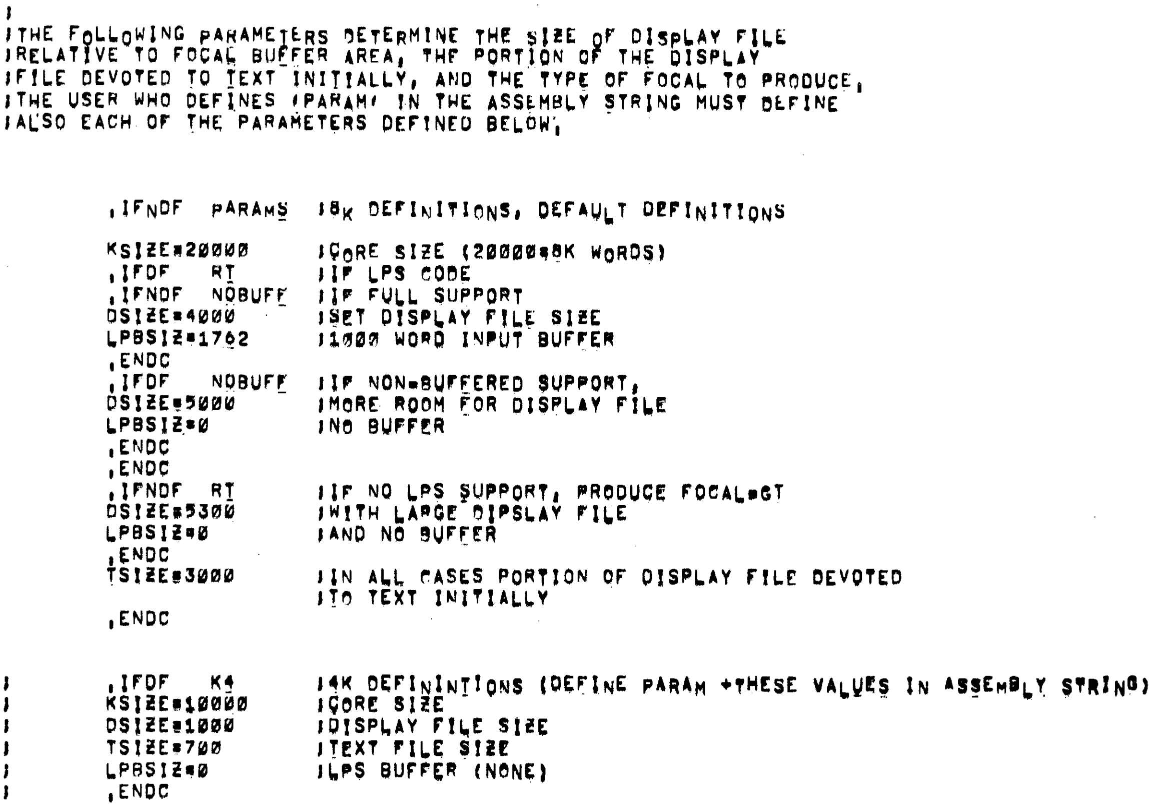
```
FOCGT.BIN/I/EQ:GT40:TERM:PARAM←TTY:,DSK:FOCGT.002

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 inital text portion 700 (octal) words long.



```
        .IFDF   GT40            ;GT40 FUNCTIONS

; **DNEW**


;COPYRIGHT 1972,1973 DIGITAL EQUIPMENT CORPORATION
;MAYNARD, MASS  01754

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



;CONDITIONAL ASSEMBLY PARAMETERS


;ASSEMBLING WITH NO PARAMETERS DEFINED IN THE ASSEMBLY STRING,
;PRODUCES FOCAL-11 WHICH WILL RUN IN AN 11/05

;ASSEMBLING WITH GT40 DEFINED PRODUCES AN 8K FOCAL-GT WITH NO
;LPS FUNCTIONS

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

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

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

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

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

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


;ASSEMBLING ANY OF THE ABOVE WITH 'TERM' DEFINED PRODUCES A VERSION THAT
;INCLUDES 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  ;8K 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=1762     ;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 DIPSLAY FILE
        LPBSIZ=0        ;AND NO BUFFER
        .ENDC
        TSIZE=3000      ;IN ALL CASES PORTION OF DISPLAY FILE DEVOTED
                        ;TO TEXT INITIALLY
        .ENDC

;       .IFDF   K4      ;4K DEFININTIONS (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 DECsystem-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 STACKO+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.

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	/	9Ø	Z
5 (ØE)	E	48	Ø	91	[
6 (ØF)	F	49	1	92	\
7 (ØG)	G	5Ø	2	93	]
8 (ØH)	H	51	3	94	↑
9 (ØI)	I	52	4	95	←
1Ø (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	1ØØ	d
15 (ØO)	O	58	:	1Ø1	e
16 (ØP)	P	59	;	1Ø2	f
17 (ØQ)	Q	6Ø	<	1Ø3	g
18 (ØR)	R	61	=	1Ø4	h
19 (ØS)	S	62	>	1Ø5	i
2Ø (ØT)	T	63	?	1Ø6	j
21 (ØU)	U	64	@	1Ø7	k
22 (ØV)	V	65	A	1Ø8	l
23 (ØW)	W	66	B	1Ø9	m
24 (ØX)	X	67	C	11Ø	n
25 (ØY)	Y	68	D	111	o
26 (ØZ)	Z	69	E	112	p
27	[	7Ø	F	113	q
28	\	71	G	114	r
29	]	72	H	115	s
3Ø	↑	73	I	116	t
31	←	74	J	117	u
32 (ØBL)	␣	75	K	118	v
33	!	76	L	119	w
34	"	77	M	12Ø	x
35	#	78	N	121	y
36	$	79	O	122	z
37	%	8Ø	P	123	{
38	&	81	Q	124	|
39	'	82	R	125	}
4Ø	(	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. FTXT 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.

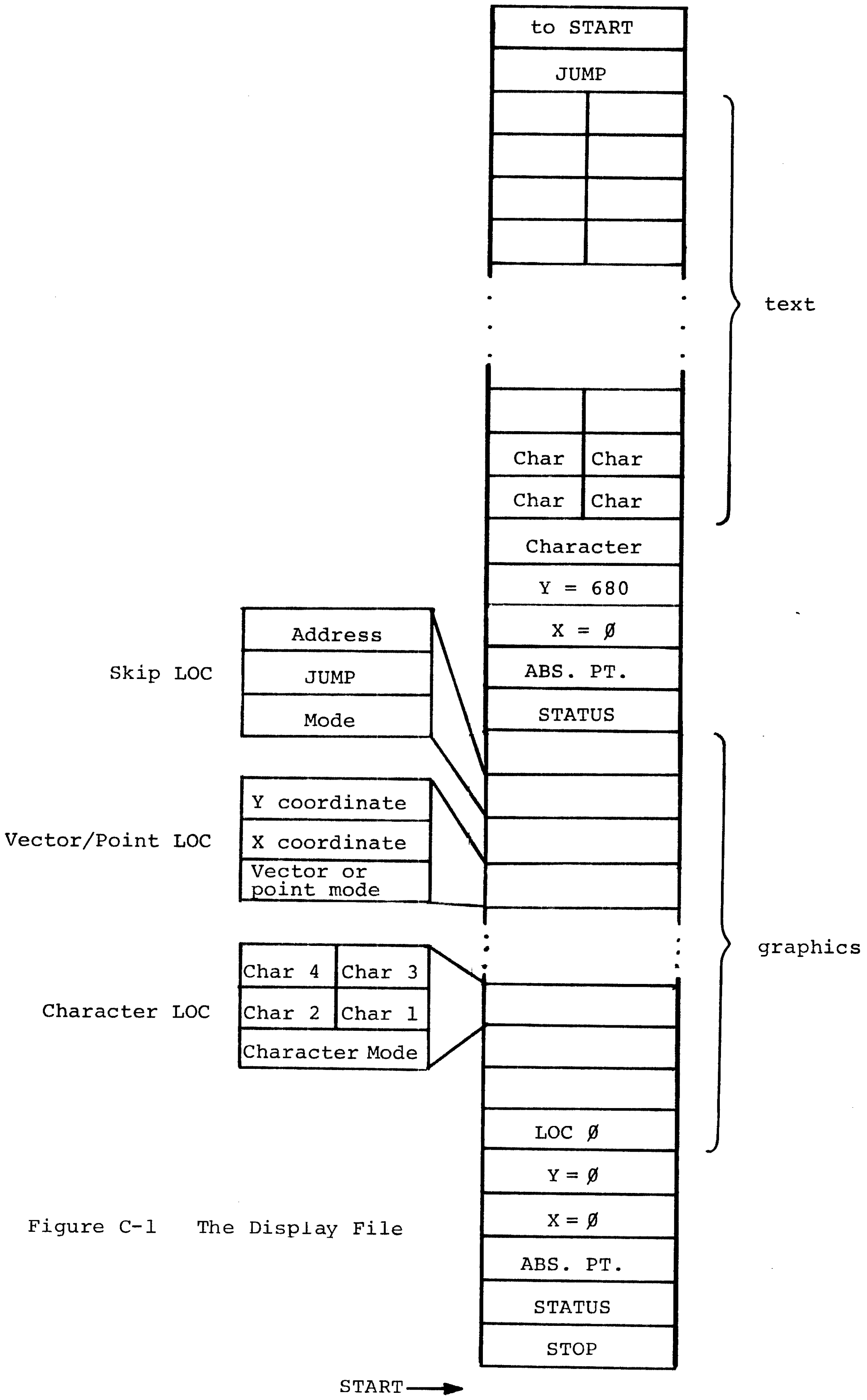


Figure C-1 The Display File

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 commmand, 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	C	COMMENT	Ignore the remainder of the line.
CONTINUE	C		
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	Abbre-viation	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  3Ø-APR-75  13:42  PAGE 66-3
FOCGT.ØØ2          SYMBOL TABLE

ABSFF     ØØ6556     AC      =%ØØØØØ1     ADCS    = 17677Ø     ADOB    = 176772
ADDCHR    Ø25114     ADDFF     ØØ623Ø     ADIN    = 17Ø4ØØ     AFIX      ØØ5344
AGO       ØØ5372     AH      =%ØØØØØ4     AL      =%ØØØØØ5     ALGZA     ØØ65Ø6
ALGZR     ØØ6522     ALIGNA    ØØ627Ø     ALIGNB    Ø26256     ALIST   = ØØ1433
ALL     = ØØØØØ1     ALTAKA    ØØ6324     ARGNXT    Ø24476     ARO       ØØ533Ø
ASCII     Ø2342Ø     ASK       ØØ5244     ASPACE    Ø2534Ø     ATAKE     ØØ53Ø4
ATLIST  = ØØ1344     AXIN      ØØ1642     AXOUT   =%ØØØØØ3     BDIVX     ØØ6476
BE        ØØ1616     BEGIN   = ØØ11ØØ     BELL    = ØØØØØ7     BELOW     Ø24262
BH      =%ØØØØØ2     BL      =%ØØØØØ3     BOTTOM    ØØ1652 023014  BUFBEG  = Ø1133Ø
BUFFUL    ØØ22Ø2     BUFR      ØØ1644     CCFLG     Ø216Ø1     CFRS      ØØ1654
CHAR    =%ØØØØØ4     CHIN      ØØ271Ø     CHINX     Ø22756     CINT      Ø1Ø256
CLCU      Ø1Ø21Ø     CLKADR  = ØØØ134     CLKIN   = 17Ø4Ø4     CLKT      Ø1Ø274
CLOSE   = Ø22626     CLRLOP    Ø24426     COCO      Ø24Ø1Ø     COMBUF  = ØØ1461
COMLST  = ØØ125Ø     COPYLN  = 1Ø463Ø     COPYLX    Ø24Ø4Ø     CR      = ØØØØ15
CRLF    = ØØ5315     DBOT    = Ø23Ø16     DCHR    = 17ØØØØ     DDSIZE  = Ø12576
DEBG      ØØ161Ø     DECONV    ØØ77Ø2     DECON1    ØØ7722     DECON2    Ø1ØØØ2
DECOY     ØØ7764     DELETE  = ØØØØØ4     DETN      Ø1Ø7Ø6     DFILE     Ø25172
DIFL      ØØ66ØØ     DIGTSA    ØØ2654     DIGTST  = 1Ø4644     DIRECT  = ØØØØØØ
DIVFF     ØØ6566     DIVZER    ØØ6642     DJMP    = 16ØØØØ     DNOP    = 164ØØØ
DO        ØØ4Ø66     DOCONT    ØØ42Ø4     DOER      Ø2422Ø     DOGR      ØØ41Ø6
DOGRP1    ØØ4124     DOGRP2    ØØ412Ø     DOXIT     Ø2421Ø     DPT     = 114ØØØ
DSIZE   = ØØ1625     DSTP    = 1734ØØ     DUNASC    Ø23454     DVEC    = 11ØØØØ
DXIT      ØØ754Ø     E       =%ØØØØØ2     ECHOLS  = Ø2145Ø     ECOLOG    ØØ3252
EDITIT    Ø24714     EFUN      ØØ4614     EFUN3     Ø24656     ELPAR     ØØ45Ø4
EMDIR     ØØ5672     EMIME     ØØ57Ø4     EMIND     Ø25726     EMREL     ØØ5744
EMT     = ØØ7ØØØ     EMTHR     ØØ5736     EMTO      Ø25716     ENAB    = ØØ3124
ENDFLG    Ø2467Ø     ENIT2     ØØ3Ø7Ø     ENUM      Ø24574     ENUM2     ØØ46Ø4
EPAR      ØØ446Ø     EPAR2     ØØ451Ø     EPURE     Ø24436     ERASE     ØØ3744
ERASET  = 1Ø464Ø     ERASEV  = 1Ø4636     ERG       Ø24Ø1Ø     ERL       ØØ4ØØ2
ERROR   = 1Ø44ØØ     ERRORC    ØØ35Ø2     ERR2      ØØ1752     ERT       ØØ377Ø
ERTX      ØØ445Ø     ERV       ØØ3764     ERVC      Ø24Ø34     ERVX      ØØ51ØØ
ETERM     ØØ4374     ETERMN    ØØ4566     ETERM1    ØØ4544     ETERM2    ØØ44Ø6
EVAL      ØØ452Ø     EVALU   = 1Ø466Ø     EVALUX    ØØ4664     EVAL.X  = 1Ø466Ø
F       =%ØØØØØ3     FCODE   = ØØ72ØØ     FCONT     Ø243Ø6     FCONT2    ØØ4312
FCOS      Ø1Ø426     FCOS2     Ø1Ø466     FCOS4     Ø1Ø526     FEMT      ØØ5472
FERRO     ØØ6732     FIGOE     ØØ7654     FIGO1     ØØ7566     FIGO2     ØØ7636
FIGO3     ØØ7664     FIGO4     ØØ7674     FINCR     ØØ4254     FINDLN  = 1Ø4624
FINDN     ØØ2574     FINDO     ØØ2622     FINDX     ØØ2566     FINERR    ØØ4252
FINFIN    ØØ4266     FISW      ØØ1626     FLARG     ØØ1612     FLIMIT    ØØ43ØØ
FLIST1    ØØ1336     FLIST2    ØØ133Ø     FLOAT     Ø2434Ø     FLOSGN    ØØ7Ø34
FLOUT     ØØ745Ø     FLTDO     ØØ5556     FLTDO1    Ø25544     FLTONE    Ø1Ø56Ø
FLTX      ØØ5624     FLTZER    ØØ353Ø     FNADDR    ØØ6146     FNTABL  = ØØ1166
FOGO1     ØØ7Ø42     FOGO2     ØØ7Ø72     FOGO3     ØØ711Ø     FOGO4     ØØ7124
FOR       ØØ4222     FPABS   = ØØ7Ø73     FPADD   = Ø27Ø1Ø     FPDIV   = ØØ7Ø3Ø
FPGET   = ØØ7ØØØ     FPINT   = ØØ7Ø71     FPMUL   = Ø27Ø4Ø     FPNEG   = ØØ7Ø74
FPNOR   = ØØ7Ø7Ø     FPPOW   = ØØ7Ø5Ø     FPPUT   = ØØ7Ø6Ø     FPRINT  = ØØ7Ø76
FPRNT     ØØ7356     FPRNTP    ØØ7376     FPSGN   = ØØ7Ø72     FPSUB   = ØØ7Ø2Ø
FPURE     ØØ5522     FP1       Ø24332     FP6       Ø23576     FREAD   = ØØ7Ø75
FROM    = ØØØØØ3     FSIN      Ø1Ø416     FSIZE   = ØØ2154     FTRY      ØØ7434
FZER    = ØØ7Ø77     GALL      ØØ2554     GALL1     ØØ2522     GEND    = Ø24144
GETARG    ØØ4674     GETC    = 1Ø4614     GETCH   = 124614     GETF      ØØ6114
GETLN   = 1Ø4622     GETLNX    ØØ2454     GETVAR    Ø247Ø4     GETX      ØØ3232
GGROUP    ØØ256Ø     GOFOC     Ø23112     GONE      ØØ2146     GOTO      ØØ3436
GRALL     Ø2332Ø     GROOVY  = 1Ø465Ø     GROVX     ØØ2672     GROVZ     ØØ265Ø
```

```
.MAIN.  MACDLX 622(622)-1  3Ø-APR-73  13:42  PAGE 66-4
FOCGT.ØØ2       SYMBOL TABLE

GSERCH    ØØ473Ø      GSIZE  = ØØ2ØØ4      GSWIP     ØØ5Ø6Ø      GSØ       ØØ4764
GS1       ØØ4776      GTAKE    ØØ517Ø      GTEST     ØØ516Ø      GTESTN    ØØ2562
GTESTW    ØØ2512      GTPC   = 172ØØØ      GTRY      ØØ5134      GTRY2     ØØ5144
GTSTAT =  172ØØ2      GTX    = 172ØØ4      GTY    =  172ØØ6      GT4Ø   =  ØØØØØØ
HIBITS    Ø2466Ø      HORD     ØØ162Ø      IF        ØØ3374      IGNOR     ØØ2114
IGNORE    Ø23152      IMMED  = ØØØØØ5      INADDR    ØØ5662      INCH   =  1Ø4612
INDEV     ØØ1632      INFILE   Ø2367Ø      INIT      Ø11Ø3Ø      INITD     Ø23166
INIT2     ØØ3Ø72      INLIST = ØØ1366      INPTR     Ø24672      INPUTN    ØØ22Ø4
INTF      ØØ633Ø      INTG     ØØ634Ø      INTO   =  ØØØØØ3      INTX      ØØ6362
INTY      ØØ6366      INTZ     ØØ6356      INV       Ø23314      INVØ      Ø24562
IOFIX     Ø1Ø3Ø4      IOGO   = Ø1Ø362      IOLIST =  Ø1Ø352      IOPATC =  Ø1Ø4ØØ
IOQ       Ø1Ø334      IPTR   = ØØØØØ1      JMS    =  123Ø14      JUSTØ     Ø23456
K         Ø23Ø16      KIN      ØØ1636      KINT      Ø1Ø11Ø      KINT1     Ø1Ø132
KINT2     Ø1Ø136      KINT3    Ø1Ø136      KINT4     Ø1Ø142      KITH      Ø1ØØ74
KKSIZE =  Ø3777Ø      KSIZE  = Ø2ØØØØ      LASTL     Ø25ØØ6      LEAST     Ø23634
LEDOUT =  17Ø4Ø2      LF     = ØØØØ12      LFCHAR    Ø25Ø64      LFS       Ø24674
LINCNT    ØØ163Ø      LINENO   ØØ1624      LINERR    ØØ2314      LISTGO =  ØØ14Ø6
LIST3  =  ØØ1456      LIST6  = ØØ1453      LOC      =%ØØØØØØ      LOCGET    Ø242Ø4
LOCGO     Ø24176      LOCSET   Ø24164      LOCØ      Ø25ØØ2      LORD      ØØ1622
LOSEØ     ØØ3422      LOSE1    ØØ3426      LOSE2     ØØ3414      LOST      ØØ337Ø
LPADR  =  ØØØ324      LPBSIZ = ØØØØØØ      LPINT     Ø24526      LPLOC     Ø24656
LPS    =  177514      LPSSIZ = ØØØØØØ      LSPR      ØØ16Ø2      LTZERO    ØØ7414
MAKSUR    Ø2344Ø      MDPØ     ØØ6434      MODE      Ø24662      MODIFY    ØØ3552
MORNOP    Ø24156      MORTXT   Ø2337Ø      MULFF     ØØ6426      MULZ      ØØ6166
NALPHA =  ØØØØ2Ø      NEGFF    ØØ6562      NEGX      ØØ647Ø      NEWLOC    Ø242ØØ
NOBLI     Ø23752      NOEKO    Ø231Ø6      NOLP      Ø2377Ø      NOLPCO    Ø24614
NOLPS     Ø232Ø4      NOP      ØØ6ØØØ      NOPLOP    Ø2415Ø      NORESP    Ø23146
NORF      ØØ6Ø36      NORMD    ØØ6Ø72      NORMZ     ØØ6172      NORM2     ØØ6Ø46
NORX      ØØ62Ø4      NOS      Ø1ØØ7Ø      NOSEND    Ø24774      NOSHOW    Ø24734
NOTBAK    Ø23612      NOTEXT   Ø236Ø6      NOTF      Ø23Ø44      NOTFUL    Ø25144
NOTOUT    Ø2365Ø      NOTT     Ø23Ø66      NOTYET    Ø23432      NO18      Ø24476
NUMF   =  ØØØØ54      ONE    = ØØØ2ØØ      ONPIC     Ø2461Ø      OPADDR    ØØ6126
OPEN   =  Ø24646      OPERR    ØØ4572      OPNEXT    Ø24Ø36      OUT       ØØ3ØØ2
OUTCH  =  1Ø461Ø      OUTCO    Ø24Ø56      OUTDEV    ØØ1634      OUTW      ØØ2764
OUTX      ØØ3Ø64      OUTY     ØØ3Ø42      OUTZ      ØØ3Ø12      P        =%ØØØØØØ
PACKC  =  1Ø4616      PACKX    ØØ3136      PARTSA    ØØ52Ø6      PARTST =  1Ø4646
PBAR      ØØ3176      PC      =%ØØØØØ7      PCF       ØØ16Ø4      PC1       ØØ3524
PC2       ØØ35Ø4      PERR     ØØ4672      PLUGCO    Ø23344      POPJ   =  ØØØ2Ø7
POSCO     Ø243ØØ      POSOUT   Ø24Ø5Ø      POWDO     ØØ67ØØ      POWF      ØØ665Ø
POWF1     ØØ6644      POWS     ØØ667Ø      PPS    =  177554      PRESET =  17Ø4Ø6
PRINT  =  1Ø44ØØ      PRINTA   ØØ2316      PRINTC =  1Ø46Ø4      PRINT2 =  1Ø4642
PRIN2A    ØØ2626      PRNTF    ØØ7Ø12      PRNTLN =  1Ø4626      PRNTP     ØØ75Ø6
PRNTS     ØØ7474      PRNT1    ØØ7532      PRNT2     ØØ7522      PROC      ØØ3454
PROCES    ØØ3452      PROGIO   Ø1Ø336      PRS    =  17755Ø      PSCAN     ØØ3444
PTEN      ØØ7464      PTR     =%ØØØØØ2      PUTCO     Ø24264      PUTF      ØØ563Ø
PUTTXT    Ø234Ø2      PWRDWN   ØØ5752      PWREGS    ØØ6Ø2Ø      PWRON     ØØ6Ø34
PWRUP     ØØ6ØØ2      READC  = 1Ø46Ø6      READF     ØØ7542      REL    =  ØØØØØ6
RESETD    Ø245Ø4      RETLOC   Ø2411Ø      RETNEW    Ø24312      RETNOW    Ø24Ø76
RETURN    ØØ3516      ROOTGO   Ø1Ø244      RTPAR  =  ØØØ211      RUBIT     ØØ32ØØ
RUBX      ØØ321Ø      RUBX2    ØØ3214      R5       =%ØØØØØ5      SCHAR     ØØ3572
SCONL     ØØ3566      SCONT    ØØ3564      SERR      ØØ3624      SET    =  ØØ4222
SFIND     ØØ361Ø      SFOUND   ØØ362Ø      SGNF      ØØ6532      SGN1      ØØ6536
SIGN      ØØ6734      SIGND    ØØ646Ø      SILENT    Ø25Ø26      SKPLPR =  1Ø4652
SKPNON =  1Ø4654      SKPNOX   Ø12372      SORTB     ØØ24Ø6      SORTC  =  1Ø46Ø2
```

```
.MAIN.  MACDLX 622(622)-1  30-APR-73  13:42  PAGE 66-5
FOCGT.002          SYMBOL TABLE

SORTD      022440         SORTJ  = 104600         SOUND      022424         SOX        002420
SP       =%000006         SPECIA = 001445         SPNOR  = 104634         SPNORX     003126
SPNXT      003124         SQUEND   010252         SQX      010050         SRETN      002226
SRNLST = 001400           STACK  = 000000         STACKO   021706         START  = 104632
STARTV     001646         STARTX   002034         STATUS = 177776         STILON     024170
STLIN      003626         STOP     001704         STOPIT   024650         STPADR = 000320
STPFLG     024664         STPINT   024634         STRING = 020030         SUBEND = 024676
SUBFF      006222         SWITCH = 001631         TASK   = 104656         TASKX      005412
TASK4      005410         TCLEAR   024362         TCRLF    025404         TCRLF2     005464
TDUMP      010564         TDUMP1   010570         TDUMP2   010612         TDUMP3     010600
TDUMP4     010656         TEMP   =%000000         TEN      007102         TENADR = 000300
TENIN  = 175610           TENINT   023116         TENOUT = 175614         TERM   = 000000
TERMS  = 001412           TERSIZ = 000150         TESTC  = 174620         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     023156         TOOBIG   024242         TOP      001650         TOR        007310
TOS        007324         TOT      007276         TPR      003534         TPR1       003540
TPS    = 177564           TQUOT    005442         TRAPH    021714         TSIZE  = 001000
TSTERR     024472         TSTTXT   024676         TTSIZE = 026000         TXTDIS     024720
TXTDUN     025156         TXTHAN   025000         TXTJMP   023642         TXTPTR     024666
TYPE       005224         UTX      003220         VERR     004670         VIS        023304
WAYOUT     023654         WGO      003742         WHIPV    004770         WHOOPS     001640
WRED       003730         WRIG     003724         WRITE    023646         WRITEL     003670
WRITE2     003654         WTESTG   003712         XABS     006376         XADC       011070
XCHARG     010772         XCHMO    010756         XCHR     010760         XCHR1      011004
XCLR       024130         XDELET   003272         XDIS     023700         XDOX       003336
XDTHIS     003360         XD3      003276         XECUTE   023044         XEL        011322
XEM        011210         XER      011316         XERR     011026         XET        011250
XEX        011134         XEXIT    011302         XEX2     011152         XEX2A      011214
XEX3       011226         XEX4     011240         XFCLK    010672         XFSBR      011026
XITR       006406         XI33     010022         XI33X    010044         XLP        024064
XMOV       023254         XOUT     010052         XPRNTL   023400         XPT        023264
XPTR   = 000002           XRAN     010704         XROL     010726         XSET       023274
XSGN       006402         XSKP     023514         XSQT     010454         XTSTLP     002636
XTXT       023364         XVEC     023244         XXCO     024400         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,ØØ2        SYMBOL TABLE

ABAD      Ø25376        ABSFF     ØØ6556        AC       =%ØØØØØ1       ADCS     = 17677Ø
ADDB    = 176772        ADDCHR    Ø27724        ADDFF      ØØ625Ø       ADIN     = 1704ØØ
AFIX      ØØ5344        AGO       ØØ5372        AH       =%ØØØØØ4       AL       =%ØØØØØ5
ALGZA     ØØ65Ø6        ALGZB     ØØ6522        ALIGNA     ØØ627Ø       ALIGNB     ØØ6256
ALIST   = ØØ1433        ALL     = ØØØØØ1        ALTAKA     ØØ6324       ARGNXT     ØØ4476
ARO       ØØ533Ø        ASCII     Ø2623Ø        ASK        ØØ5244       ASPACE     ØØ534Ø
ATAKE     ØØ53Ø4        ATLIST  = ØØ1344        AXIN       ØØ1642       AXOUT    =%ØØØØØ3
BDIVX     ØØ6476        BE        ØØ1616        BEGIN    = ØØ11ØØ       BELL     = ØØØØØ7
BELOW     Ø27Ø72        BH      =%ØØØØØ2        BL       =%ØØØØØ3       BOTTOM     ØØ1652
BUFBEG  = Ø1133Ø        BUFFUL    ØØ22Ø2        BUFR       ØØ1644       CCFLG      ØØ16Ø1
CFRS      ØØ1654        CHAR    =%ØØØØØ4        CHIN       ØØ271Ø       CHINX      ØØ2756
CINT      Ø1Ø256        CLCU      Ø1Ø21Ø        CLKADR   = ØØØ134       CLKIN    = 1704Ø4
CLKINT    Ø2557Ø        CLKLOP    Ø25552        CLKT       Ø1Ø274       CLOSE    = Ø22626
CLRLOP    Ø27236        COCO      Ø2662Ø        COMBUF   = ØØ1461       COMLST   = ØØ125Ø
COPYLN  = 1Ø463Ø        COPYLX    ØØ4Ø4Ø        CR       = ØØØØ15       CRLF     = ØØ5Ø15
DBOT    = Ø21174        DCHR    = 1ØØØØØ        DDSIZE   = ØØ7774       DEBG       ØØ161Ø
DECONV    ØØ77Ø2        DECON1    ØØ7722        DECON2     Ø1ØØØ2       DECOY      ØØ7764
DELETE  = ØØØØØ4        DETN      Ø1ØØØ6        DFILE      Ø27774       DIFL       ØØ66ØØ
DIGTSA    ØØ2654        DIGTST  = 1Ø4644        DIRECT   = ØØØØØØ       DIVFF      ØØ6566
DIVZER    ØØ6642        DJMP    = 16ØØØØ        DNOP     = 164ØØØ       DO         ØØ4Ø66
DOCONT    ØØ42Ø4        DOER      ØØ422Ø        DOGR       ØØ41Ø6       DOGRP1     ØØ4124
DOGRP2    ØØ412Ø        DOXIT     ØØ421Ø        DPT      = 114ØØØ       DSIZF    = ØØ1252
DSTP    = 1734ØØ        DTIME     Ø256Ø4        DUNAD      Ø25356       DUNASC     Ø26264
DVEC    = 11ØØØØ        DXIT      ØØ754Ø        E        =%ØØØØØ2       ECHOLS   = ØØ145Ø
ECOLOG    ØØ3252        EDITIT    Ø27524        EFUN       ØØ4614       EFUNS      ØØ4656
ELPAR     ØØ45Ø4        EMDIR     ØØ5672        EMIME      ØØ57Ø4       EMIND      ØØ5726
EMREL     ØØ5744        EMT     = ØØ7ØØØ        EMTHR      ØØ5736       EMTO       ØØ5716
ENAB    = ØØ3124        ENDFLG    Ø275ØØ        ENIT2      ØØ3Ø7Ø       ENUM       ØØ4574
ENUM2     ØØ46Ø4        EPAR      ØØ446Ø        EPAR2      ØØ451Ø       EPURE      ØØ4436
ERASE     ØØ3744        ERASET  = 1Ø464Ø        ERASEV   = 1Ø4636       ERG        ØØ4Ø1Ø
ERL       ØØ4ØØ2        ERROR   = 1Ø44ØØ        ERRORC     ØØ35Ø2       ERR2       ØØ1752
ERT       ØØ377Ø        ERTX      ØØ4Ø5Ø        ERV        ØØ3764       ERVC       ØØ4Ø34
ERVX      ØØ51ØØ        ETERM     ØØ4374        ETERMN     ØØ4566       ETERM1     ØØ4544
ETERM2    ØØ44Ø6        EVAL      ØØ452Ø        EVALU    = 1Ø466Ø       EVALUX     ØØ4664
EVAL,X  = 1Ø466Ø        F       =%ØØØØØ3        FCODE    = ØØ72ØØ       FCONT      ØØ43Ø6
FCONT2    ØØ4312        FCOS      Ø1Ø426        FCOS2      Ø1Ø466       FCOS4      Ø1Ø526
FEMT      ØØ5472        FERRO     ØØ6732        FIGOE      ØØ7654       FIGO1      ØØ7566
FIGO2     ØØ7636        FIGO3     ØØ7664        FIGO4      ØØ7674       FINCR      ØØ4254
FINDLN  = 1Ø4624        FINDN     ØØ2574        FINDO      ØØ2622       FINDX      ØØ2566
FINERR    ØØ4252        FINFIN    ØØ4266        FISW       ØØ1626       FLARG      ØØ1612
FLIMIT    ØØ43ØØ        FLIST1    ØØ1336        FLIST2     ØØ135Ø       FLOAT      Ø2715Ø
FLOSGN    ØØ7Ø34        FLOUT     ØØ745Ø        FLTDO      ØØ5556       FLTDO1     ØØ5544
FLTONE    Ø1Ø56Ø        FLTX      ØØ5624        FLTZER     ØØ355Ø       FNADDR     ØØ6146
FNTABL  = ØØ1166        FOGO1     ØØ7Ø42        FOGO2      ØØ7Ø72       FOGO3      ØØ711Ø
FOGO4     ØØ7124        FOR       ØØ4222        FPABS    = ØØ7Ø73       FPADD    = ØØ7Ø1Ø
FPDIV   = ØØ7Ø3Ø        FPGET   = ØØ7ØØØ        FPINT    = ØØ7Ø71       FPMUL    = ØØ7Ø4Ø
FPNEG   = ØØ7Ø74        FPNOR   = ØØ7Ø7Ø        FPPOW    = ØØ7Ø5Ø       FPPUT    = ØØ7Ø6Ø
FPRINT  = ØØ7Ø76        FPRNT     ØØ7356        FPRNTP     ØØ7376       FPSGN    = ØØ7Ø72
FPSUB   = ØØ7Ø2Ø        FPURE     ØØ5522        FP1        Ø27142       FP1Ø       Ø25444
FP6       Ø264Ø6        FREAD   = ØØ7Ø75        FROM     = ØØØØØ3       FSIN       Ø1Ø416
FSIZE   = ØØ66ØØ        FTRY      ØØ7434        FZER     = ØØ7Ø77       GALL       ØØ2554
GALL1     ØØ2522        GEND    = Ø26754        GETARG     ØØ4674       GETC     = 1Ø4614
GETCH   = 1Ø4614        GETF      ØØ6114        GETLN    = 1Ø4622       GETLNX     ØØ2454
GETVAR    ØØ47Ø4        GETX      ØØ3232        GGROUP     ØØ256Ø       GOFOC      Ø257Ø4
```

```
,MAIN,  MACDLX 622(622)-1  3Ø-APR-73  13:4Ø  PAGE 66-4
FOCGT,ØØ2         SYMBOL TABLE

GONE        ØØ2146          GOTO        ØØ3436          GRALL       Ø2613Ø          GROOVY   =  1Ø465Ø
GROVX       ØØ2672          GROVZ       ØØ265Ø          GSERCH      ØØ473Ø          GSIZE    =  ØØ2Ø14
GSWIP       ØØ5Ø6Ø          GSØ         ØØ4764          GS1         ØØ4776          GTAKF       ØØ517Ø
GTEST       ØØ516Ø          GTESTN      ØØ2562          GTESTW      ØØ2512          GTPC     =  172ØØØ
GTRY        ØØ5134          GTRY2       ØØ5144          GTSTAT   =  172ØØ2          GTX      =  172ØØ4
GTY      =  172ØØ6          GT4Ø     =  ØØØØØØ          HIBITS      Ø2747Ø          HORD        ØØ162Ø
IF          ØØ3374          IGNOR       ØØ2114          IGNORE      Ø25744          IMMED    =  ØØØØØ5
INADDR      ØØ5662          INCH     =  1Ø4612          INDEV       ØØ1632          INFILE      Ø265ØØ
INIT        Ø1133Ø          INITD       Ø2576Ø          INIT2       ØØ3Ø72          INLIST   =  ØØ1366
INPTR       Ø275Ø2          INPUTN      ØØ22Ø4          INTF        ØØ633Ø          INTG        ØØ634Ø
INTO     =  ØØØØØ3          INTX        ØØ6362          INTY        ØØ6366          INTZ        ØØ6356
INV         Ø26124          INVØ        Ø27372          IOFIX       Ø1Ø3Ø4          IOGO     =  Ø1Ø362
IOLIST   =  Ø1Ø352          IOPATC   =  Ø1Ø4ØØ          IOQ         Ø1Ø334          IPTR     =  ØØØØØ1
JMS      =  1Ø3414          JUSTØ       Ø26266          K           Ø2561Ø          KIN         ØØ1636
KINT        Ø1Ø11Ø          KINT1       Ø1Ø132          KINT2       Ø1Ø136          KINT3       Ø1Ø136
KINT4       Ø1Ø142          KITH        Ø1ØØ74          KKSIZE   =  Ø3777Ø          KSIZE    =  Ø2ØØØØ
LASTL       Ø27716          LEAST       Ø26444          LEDOUT   =  17Ø4Ø2          LF       =  ØØØØ12
LFCHAR      Ø27674          LFS         Ø275Ø4          LINCNT      ØØ163Ø          LINENO      ØØ1624
LINERR      ØØ2314          LISTGO   =  ØØ14Ø6          LIST3    =  ØØ1456          LIST6    =  ØØ1453
LOC      =%ØØØØØØ           LOCGET      Ø27Ø14          LOCGO       Ø27ØØ6          LOCSET      Ø26774
LOCØ        Ø3ØØØ4          LORD        ØØ1622          LOSEØ       ØØ3422          LOSE1       ØØ3426
LOSE2       ØØ3414          LOST        ØØ337Ø          LPADR    =  ØØØ324          LPBSIZ   =  ØØ1762
LPINT       Ø27336          LPLOC       Ø27466          LPREG       Ø254Ø2          LPS      =  177514
LPSBUF   =  Ø21174          LPSSIZ   =  ØØ4414          LSPR        ØØ16Ø2          LTZERO      ØØ7414
MAKSUR      Ø2625Ø          MDPØ        ØØ6434          MODE        Ø27472          MODIFY      ØØ3552
MORNOP      Ø26766          MORTXT      Ø262ØØ          MULFF       ØØ6426          MULZ        ØØ6166
NALPHA   =  ØØØØ2Ø          NEGFF       ØØ6562          NEGNUM      Ø25426          NEGX        ØØ647Ø
NEWLOC      Ø27Ø1Ø          NOAD        Ø25312          NOAROV      Ø256Ø2          NOBLI       Ø26562
NOEKO       Ø257ØØ          NOLP        Ø266ØØ          NOLPCO      Ø27424          NOLPS       Ø26Ø14
NOP         ØØ6ØØØ          NOPLOP      Ø2676Ø          NORESP      Ø2574Ø          NORF        ØØ6Ø36
NORMD       ØØ6Ø72          NORMZ       ØØ6172          NORM2       ØØ6Ø46          NORX        ØØ62Ø4
NOS         Ø1ØØ7Ø          NOSEND      Ø276Ø4          NOSET       Ø25544          NOSH        Ø25242
NOSHOW      Ø27544          NOTBAK      Ø26422          NOTEXT      Ø26416          NOTF        Ø25636
NOTFUL      Ø27754          NOTOUT      Ø2646Ø          NOTT        Ø2566Ø          NOTYET      Ø26242
NO18        Ø273Ø6          NUMF     =  ØØØØ7Ø          NUMLOP      Ø25436          ONE      =  ØØØ2ØØ
ONPIC       Ø2742Ø          OPADDR      ØØ6126          OPEN     =  Ø24646          OPERR       ØØ4572
OPNEXT      ØØ4536          OUT         ØØ3ØØ2          OUTCH    =  1Ø461Ø          OUTCO       Ø26666
OUTDEV      ØØ1634          OUTW        ØØ2764          OUTX        ØØ3Ø64          OUTY        ØØ3Ø42
OUTZ        ØØ3Ø12          P        =%ØØØØØØ           PACKC    =  1Ø4616          PACKX       ØØ3136
PARTSA      ØØ52Ø6          PARTST   =  1Ø4646          PBAR        ØØ3176          PC       =%ØØØØØ7
PCF         ØØ16Ø4          PC1         ØØ3524          PC2         ØØ35Ø4          PERR        ØØ4672
PLUGCO      Ø26154          POPJ     =  ØØØ2Ø7          POSCO       Ø2711Ø          POSOUT      Ø2666Ø
POWDO       ØØ67ØØ          POWF        ØØ665Ø          POWF1       ØØ6644          POWS        ØØ667Ø
PPS      =  177554          PRESET   =  17Ø4Ø6          PRINT    =  1Ø44ØØ          PRINTA      ØØ2316
PRINTC   =  1Ø46Ø4          PRINT2   =  1Ø4642          PRIN2A      ØØ2626          PRNTF       ØØ7Ø12
PRNTLN   =  1Ø4626          PRNTP       ØØ75Ø6          PRNTS       ØØ7474          PRNT1       ØØ7532
PRNT2       ØØ7522          PROC        ØØ3454          PROCES      ØØ3452          PROGIO      Ø1Ø336
PRS      =  17755Ø          PSCAN       ØØ3444          PTEN        ØØ7Ø64          PTR      =%ØØØØØ2
PUTCO       Ø27Ø74          PUTF        ØØ563Ø          PUTTXT      Ø26212          PWRDWN      ØØ5752
PWREGS      ØØ6Ø2Ø          PWRON       ØØ6Ø34          PWRUP       ØØ6ØØ2          READC    =  1Ø46Ø6
READF       ØØ7542          REL      =  ØØØØØ6          RESETD      Ø27314          RETDAT      Ø25366
RETLOC      Ø2672Ø          RETNEW      Ø27122          RETNOW      Ø267Ø6          RETURN      ØØ3516
ROOTGO      Ø1Ø244          RT       =  ØØØØØØ          RTPAR    =  ØØØ211          RUBIT       ØØ32ØØ
RUBX        ØØ321Ø          RUBX2       ØØ3214          R5       =%ØØØØØ5           SAMLOP      Ø253Ø4
```

```
,MAIN,  MACDLX 622(622)-1  3Ø-APR-73  13:4Ø  PAGE 66-5
FOCGT,ØØ2        SYMBOL TABLE

SCHAR     ØØ3572     SCONL     ØØ3566     SCONT     ØØ3564     SERR      ØØ3624
SET     = ØØ4222     SFIND     ØØ361Ø     SFOUND    ØØ362Ø     SGNF      ØØ6532
SGN1      ØØ6536     SIGN      ØØ6734     SIGND     ØØ64ØØ     SILENT    Ø27636
SIZOK     Ø25254     SKPLPR  = 1Ø4652     SKPNON  = 1Ø4654     SKPNOX    ØØ2372
SORTB     ØØ24Ø6     SORTC   = 1Ø46Ø2     SORTD     ØØ244Ø     SORTJ   = 1Ø46ØØ
SOUND     ØØ2424     SOX       ØØ242Ø     SP      =XØØØØØ6     SPECIA  = ØØ1445
SPNOR   = 1Ø4634     SPNORX    ØØ3126     SPNXT     ØØ3124     SQUEND    Ø1Ø252
SQX       Ø1Ø25Ø     SRETN     ØØ2226     SRNLST  = ØØ14ØØ     STACK   = ØØØØØØ
STACKO    ØØ17Ø6     START   = 1Ø4632     STARTV    ØØ1646     STARTX    ØØ2Ø34
STATUS  = 177776     STILON    Ø27ØØØ     STLIN     ØØ3626     STOP      ØØ17Ø4
STOPIT    Ø2746Ø     STPADR  = ØØØ32Ø     STPFLG    Ø27474     STPINT    Ø27444
STRING  = ØØØØ3Ø     SUBENU  = Ø275Ø6     SUBFF     ØØ6222     SWITCH  = ØØ1631
TASK    = 1Ø4656     TASKX     ØØ5412     TASK4     ØØ541Ø     TCLEAR    Ø27172
TCRLF     ØØ54Ø4     TCRLF2    ØØ5464     TDUMP     Ø1Ø564     TDUMP1    Ø1Ø57Ø
TDUMP2    Ø1Ø612     TDUMP3    Ø1Ø6ØØ     TDUMP4    Ø1Ø656     TEMP    =XØØØØØØ
TEN       ØØ71Ø2     TENADR  = ØØØ3ØØ     TENIN   = 17561Ø     TENINT    Ø2571Ø
TENOUT  = 175614     TERM    = ØØØØØØ     TERMS   = ØØ1412     TERSIZ  = ØØØ15Ø
TESTC   = 1Ø462Ø     TESTX     ØØ2334     TEST1Ø    Ø256Ø6     TEX       ØØ2366
TEXT      Ø31772     THISLN    ØØ16Ø6     THROUG  = ØØØØØ4     TINTR     ØØ543Ø
TJUMPS    Ø26754     TKS     = 17756Ø     TLIST   = ØØ1441     TOF       ØØ726Ø
TOFOCL    Ø25751     TOG       ØØ7246     TOHOST    Ø2575Ø     TOOBIG    Ø27Ø52
TOP       ØØ165Ø     TOR       ØØ731Ø     TOS       ØØ7324     TOT       ØØ7276
TPR       ØØ3534     TPR1      ØØ354Ø     TPS     = 177564     TQUOT     ØØ5442
TRAPH     ØØ1714     TSIZE   = ØØ1ØØØ     TSTERR    Ø273Ø2     TSTTXT    Ø275Ø6
TTSIZE  = ØØ6ØØØ     TXTDIS    Ø2753Ø     TXTDUN    Ø27766     TXTHAN    Ø2761Ø
TXTJMP    Ø26452     TXTPTR    Ø27476     TYPE      ØØ5224     UTX       ØØ322Ø
VERR      ØØ467Ø     VIS       Ø26114     WAYOUT    Ø26464     WGO       ØØ3742
WHIPV     ØØ477Ø     WHOOPS    ØØ164Ø     WRED      ØØ373Ø     WRIG      ØØ3724
WRITE     ØØ3646     WRITEL    ØØ367Ø     WRITE2    ØØ3654     WTESTG    ØØ3712
XABS      ØØ6376     XADC      Ø11Ø7Ø     XCHARG    Ø1Ø772     XCHMO     Ø1Ø756
XCHR      Ø1Ø76Ø     XCHR1     Ø11ØØ4     XCLR      Ø2674Ø     XDELET    ØØ3272
XDIS      Ø2651Ø     XDOX      ØØ3336     XDTHIS    ØØ336Ø     XD3       ØØ3276
XECUTE    ØØ3544     XEL       Ø11322     XEM       Ø1121Ø     XER       Ø11316
XERR      Ø11326     XET       Ø1125Ø     XEX       Ø11134     XEXIT     Ø113Ø2
XEX2      Ø11152     XEX2A     Ø11214     XEX3      Ø11226     XEX4      Ø1124Ø
XFCLK     Ø1Ø272     XFSBR     Ø11Ø26     XITR      ØØ64Ø6     XI33      Ø1ØØ22
XI33X     Ø1ØØ44     XLED      Ø254Ø4     XLP       Ø26674     XMOV      Ø26Ø64
XOUT      Ø1ØØ52     XPRNTL    ØØ31ØØ     XPT       Ø26Ø74     XPTR    = ØØØØØ2
XRAN      Ø1Ø7Ø4     XROL      Ø1Ø726     XSAM      Ø25142     XSET      Ø261Ø4
XSGN      ØØ64Ø2     XSKP      Ø26324     XSQT      Ø1Ø154     XTIC      Ø25516
XTSTLP    ØØ2636     XTXT      Ø26174     XVEC      Ø26Ø54     XXCO      Ø2661Ø
XYCO      Ø26614     Z       = Ø2Ø724     ZEROOM    ØØ637Ø     .       = Ø25752


ERRORS DETECTED:  Ø
```

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

DEC-11-GGTMA-B-D
FOCAL-GT/RT USER'S MANUAL

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