DIGIGRAPHIC

MAN↔MACHINE

SYSTEMS

DIGIGRAPHIC SYSTEM 270 FUNCTION CONTROL PROGRAM SPECIFICATIONS

SUPPLEMENT 1

APPLICATION INTERFACE-FORTRAN SUBROUTINE SPECIFICATIONS

CONTROL DATA

CORPORATION

270-FCP32-465-7
SUPPLEMENT 1

DIGIGRAPHIC SYSTEM 270

FUNCTION CONTROL PROGRAM SPECIFICATIONS

SUPPLEMENT 1

APPLICATION INTERFACE-FORTRAN SUBROUTINE SPECIFICATIONS

June 1965

Disclaimer: This document is a working paper only, and does not necessarily represent any official intent on the part of Control Data.

CONTROL DATA CORPORATION
Digigraphic Laboratories
Northwest Industrial Park - 3rd Avenue
Burlington, Massachusetts

TABLE OF CONTENTS

TABLE OF CONTENTS (Continued)

Section Page

*These FCP routines are directly callable by application programs.

PREFACE

This document is a supplement to Section 6 of the Digigraphic System 270 Function Control Program Specifications dated April, 1965 and contains the subroutine specifications for the Digigraphic System Application Interface.

Although published for information purposes only, these subroutine specifications are sufficiently detailed to support design development and coding of application programs.

6.1 Application Interface Library

The Application Interface Library provides for full Digigraphic List processing, simulation of operator actions, modification of system controls, and creation of user designed controls. In all cases, application interface routines are called as FORTRAN subroutines.

Call statements with BCD words and numerical parameters establish the conditions for proper execution of the routines.

Output parameters must always be defined as variables.

Several variations of input parameters can be used:

a. Those input parameters specified as BCD constants can also be integer variables provided those variables have been preset with the required BCD constants.

b. Those input parameters which can be ignored must be present in the call statement as dummy parameters; normally zero.

EXAMPLE

```
CALL MENTI (0, IERR, 3HCNN, ICAT, 0, 0)
```

The options on the first parameter (0) are: DI = Display, ND = Non Display, ER = ERASE. The value zero indicates that none of the options are requested, thus no change. The second parameter (IERR) is the location of an error flag. The third parameter (3HCNN) indicates that a change is to be made on the Category Field (character C) and that the last two parameters (characters NN) will be ignored. ICAT contains the value for the Category Field. The two remaining zeros are dummy parameters.

6.1.1 Parameters

6.1.1.1 Entity Type Codes (ET)

(Used only with subroutine CENTI)

6.1.1.2 Entity Type Modifiers (ETM)

(Used only with subroutine CENTI)

6.1.1.3 Light Button Mnemonics. The following list contains the FCP Light Button BCD mnemonics for display purposes and for use by application interface routines.

NOTE

Primary Light Buttons have leading blanks and Secondaries have trailing blanks. The following FORTRAN feature can be used to identify Primary Light Button Mnemonics; 2RGF. This results in the BCD constant 00GF.

Light Button Mnemonics

Light Button Name	Type	Codes for FCP Light Buttons
Graphic Form	Primary	00GF
Dot	Secondary	DE
Line	Secondary	LE
Circle	Secondary	CE
Circular Arc	Secondary	CA
Polystring	Secondary	PE
Point Line Classification	Primary	00PL
Center	Secondary	CP

6.1.1.3 (Continued)

Light Button Mnemonics (cont'd)

Light Button Name	Type	Codes for FCP Light Buttons
Group Control	Primary	00GC
Collect	Secondary	CG
Remove	Secondary	RG
Add	Secondary	AG
Break	Secondary	BG
Level 1	Secondary	1G
Level 2	Secondary	2G
Level 4	Secondary	4G
Top	Secondary	TG
Utility Control	Primary	00UC
Redisplay	Secondary	RD
Non-display	Secondary	ND
Call Application Program	Secondary	AP
Terminate Application	Secondary	TA
Sign-Off	Secondary	SO
Drawing Control	Primary	00DC
Identify	Secondary	ID
Store	Secondary	ST
Select	Secondary	SE
Alphanumeric Control	Primary	00AC
Label	Secondary	LA
Note	Secondary	NT
Text	Secondary	TX
Value	Secondary	VL
Enter	Secondary	EN
Double	Primary	00DP
Halve	Primary	00HP

6.1.1.4 Keyboard Assignment Table (KAT) Mnemonics. The following table shows the relationship between a physical Keyboard button and its logical assignment. Modifier buttons have a character flag address (FCPFUNCT) in COMMON. Verb buttons have a transfer address in table KVERB. Use of the intermediate table allows reassignment of physical buttons without affecting the logical function of the buttons.

Physical-Logical Keyboard Button Number Assignment Relationships Currently Implemented by FCP

KAT No.*	Keyboard Button	FCP Function No.	KVERB No.	Mnemonic
1	Reject		1	R
2	Accept		2	A
3	Unassigned			
4				
5	↕			
6				
7	Unassigned			
8	Clear Pick Table		5	CP
9	Transform Copy		3	TC
10	Transform		4	T
11	Unassigned			
12	Normal	8		N
13	Erase		6	E
14	Lock	4		L
15	Override	9		O
16	Center	6		C
17	Graphic Entity	7		B
18	Point	3		G
19	Unassigned	2		P
20	↕			
21	Unassigned			
22	Light Pen	1		LP
23	Unassigned			
24	End Point	5		EP

* For illustration of Keyboard button number assignments see Figure 2-1.

6.1.1.5 Flag Word. Presently, all error flags or error signals returned to the application program are minus values, for example minus one (77777776). Minus zero will not be used as an error flag.

6.1.1.6 FCP Defined Register Call Mnemonics

Name	Mnemonic	Length in BCD Characters	Mode	3200 Words Needed for Mode Value
X_1 Coordinate	X1	0	Flt Pt	2
Y_1 Coordinate	Y1	0	Flt Pt	2
Length	LE	0	Flt Pt	2
Message Register	MR	24	BCD	6
Input Register	IR	24	BCD	6
Category	CA	2	BCD	1
Save Register 1	S1	12	BCD	3
Save Register 2	S2	12	BCD	3
Reference Field	RF	4	BCD	1
Angle	AR	0	Flt Pt	2

6.1.1.7 Display Image Control. Display images are projected on the 20 inch diameter viewing area of a flat faced cathode ray tube (CRT). The digital display-addressing scheme employs a 4096 x 4096 bit Display Grid within which the CRT is inscribed. The variable window for viewing graphic and alphanumeric information is defined by a displayed frame on the CRT. This frame is a key factor in Digigraphic display processing. To the viewer, the frame outline separates the Working Surface within the frame from the Control Surface outside the frame. Graphic operations such as graphic inputs and manipulation using the Light Pen are restricted to the Working Surface within the frame.

To the application programmer, the frame parameters form the basis for searching the Digigraphic List to determine which entities should be processed for displaying any particular view. The resulting Frame Grid which encompasses the framed entities is based upon the same distance values as the Display Grid but with a very important distinction. Namely, that Frame Grid units are expressed in terms of Construction Grid coordinates with the center of the Frame defined by a pair of Construction Grid coordinates whereas the Display Grid center is always zero. Framing is the action of defining the entities encompassed within the frame area, then transforming these entities from Frame Grid coordinates to Display Grid coordinates.

A Zoom Index is used to define the relationship between the size of an image presented to a viewer on the Display Grid and the geometric description of that image on the Construction Grid. This section describes the parameters and the mathematical relationships involved in display image control.

Each time either the Zoom Index or the positioning parameters are changed, the displayed image is completely regenerated. This digital image regeneration-on-change procedure enables continuous view manipulation without cumulative analog distortion.

Programming for the Digigraphic 270 system is based on a 12-bit display coordinate grid. Digital circuitry in the 271 Controller processes the full 12-bit X and Y coordinates. However, the last two bits in these coordinates have only a minor effect on beam positioning on the 273 Digigraphic Console display.

Display image control is accomplished by manipulating the four following display image control parameters:

KZOOMI defines the ratio between the displayed frame and its projection on the construction grid.

KCGC defines the frame center on the construction grid.

KSGD defines the frame size in display units. (1 unit = .005")

KSGC defines the frame center on the Display Grid (CRT).

The sequence involved in going from the Digigraphic List to a viewed image includes translation of Construction Grid coordinates as recorded in the displayable entities into the Frame projection, then performing a centro- affine transformation to establish an interim Frame Grid and finally translating this into a display grid image.

KZOOMI

KZOOMI is a signed two digit decimal integer (6-bits). To use this parameter, first compute the distance from the center of the Frame Grid to the point being processed, then either use the Zoom Index (KZOOMI) to map from the Construction Grid to the Display Grid, or use the complement of KZOOMI to map from the Display Grid to the Construction Grid.

During this mapping process, the Construction Grid coordinates are shifted left or right as designated by the Zoom Index. When the Zoom Index is zero, there is no shift. The maximum Zoom Index values are -22 and +11, representing a shift of 22 places to the right and 11 places to the left respectively. Beyond these values, the system does not discriminate changes. The most probable range of useful Zoom Index values lies between -13 and +4. These two limits represent the displaying of the entire Construction Grid within an area 2.56 x 2.56 inches (ZI = -13) on the CRT or as a viewing ratio such that a Display Grid measuring 167,772.16 x 167,777.16 inches (ZI = +4) would be required to display the entire Construction Grid. The least significant bits of the Construction Grid coordinates would be equated to 6 x 10^{-7} inches (ZI = -13) and 0.08 inches (ZI = +4) respectively.

KCGC

The location of the frame on the Construction Grid is controlled by a two word FCP COMMON parameter array called KCGC. KCGC contains 22-magnitude bits (bits 0 through 21). Bit-23 is a sign bit and bit-22 is a sign extension bit. KCGC(1) is the X value, KCGC(2) is the Y value.

KSGD

The frame dimensions are defined by the two word FCP COMMON parameter array KSGD. KSGD(1) contains the value in Display Grid units for the X dimension from the center to the side of the frame (one Display Grid unit = 0.005-inch). KSGD(2) contains the value in Display Grid units for the Y dimension from the center to the top of the frame.

EXAMPLE

To produce a frame measuring 17-inches across by 11 inches high, KSGD would have the following values:

a. KSGD(1) = 8.5 inches center-to-side by 200 Display Grid units per inch.
= 1700 Display Grid units.

b. KSGD(2) = 5.5 inches center-to-top by 200 Display Grid units per inch.
= 1100 Display Grid units.

KSGC

KSGC is a two-word parameter array in FCP COMMON which defines the frame center in Display Grid coordinates. KSGC(1) and (2) contain 11 magnitude bits (0 through 10) with sign bit in bit-23 and sign extension in bit-11 through 22. When KSGC(1) and KSGC(2) = 0, the frame is positioned in the center of the Display Grid. KSGC(1) is the X value, and KSGC(2) is the Y value.

6.1.1.8 Coordinate Transformations. In order to determine if a point (point P) on the Construction Grid lies within the display frame, the distance from point P to the frame center is scaled to Display Grid units and the magnitude compared with the frame dimensions. Scaling is done by multiplying the X and Y coordinates of point P by the value 2 raised to the Zoom Index power. This multiplication is equivalent to shifting the Construction Grid coordinates right or left, depending on the sign of the Zoom Index. If point P lies within the display frame, it must be translated to the frame reference system by adding the coordinates of the frame center on the Display Grid to the point P coordinates (Construction Grid) for display in the correct position on the CRT.

EXAMPLE

The following equations represent the process of determining position of point P on the CRT.

Px, Py are X and Y coordinates of point P in Construction Grid units.

$P'x$, $P'y$ are coordinates of point P in Display Grid units.

To determine if point P lies within the displayed frame:

$P'x = \left| (Px\text{-}KCGC(1)*2^{KZOOMI} \right| KSGD(1)$

$P'y = \left| (Py\text{-}KCGC(2)*2^{KZOOMI} \right| KSGD(2)$

To determine the Display Coordinates of point P:

$P'x + KSGC(1)$ = X display coordinate of P.

$P'y + KSGC(2)$ = Y display coordinate of P.

6.1.2 General Subroutine Structure. Application interface subroutines in general are structured to perform three sequential functions; namely, collection of data from pertinent calling sequences, implementation of FCP processing sequence, and dispersion of resultant data to the application program as required.

6.1.3 Control Routines. The application interface control routines allow the user to redefine a specified range of Digigraphic System controls according to his requirements. These control functions include:

a. The ability to restrict specific Console controls to application program use only.

b. Definition or redefinition of Keyboard functions.

c. Creation of both primary and secondary Light Pen control elements.

d. Restoration of Console control to the operator.

e. Termination of application-defined Keyboard functions and Light Pen control elements.

f. Termination of the application program and transfer of control to FCP and the Console operator.

Subroutine Ident. LOCKI

1. Subroutine Name:

Lockout Selected Functions Through the Interface.

2. References:

Digigraphic System 270 Function Control Program Specifications, Section 6.

3. Abstract:

This subroutine provides the application program with the ability to suspend operator use of specific console controls.

4. Functional Description:

LOCKI suspends operator use of specified console controls. Operator control will be reinstated only when a reset command (through a call to subroutine RSETI) is given by the application program.

CALL STATEMENT

CALL LOCKI (2HC$_1$C$_2$, IARRAY1, IARRAY2, IERR)

5. Input/Output Data:

2HC$_1$C$_2$

Hollerith control characters which specify the processing to be performed by LOCKI. The specific codes are listed below.

C$_1$ = A

Lock out all Light Button controls (IARRAYl field will be ignored).

C$_1$ = D

Lock out selected controls listed in IARRAY1. "D" indicates that IARRAYl contains DL addresses of control.

C$_1$ = B

Lock out selected controls listed in IARRAY1. "B" indicates that IARRAYl contains BCD mnemonics of primary and/or secondary Light Button control to be locked out.

5. (Continued)

C_1 = N

Do not lock out any Light Button control (IARRAY1 field will be ignored).

C_2 = A

Lock out all Console Keyboard buttons (IARRAY2 field will be ignored).

C_2 = S

Lock out selectively the Console Keyboard buttons in IARRAY2.

C_2 = N

Do not lock out any Console Keyboard buttons (IARRAY2 field will be ignored).

IARRAY1

List of DL Addresses or BCD mnemonics of the Light Button controls to be locked out. When BCD mnemonic codes are used, a four character comparison is made against the contents of DL record. FCP primary Light Button mnemonics are right justified and secondary Light Buttons are left justified (See paragraph 6.1.1.3). 3200 FORTRAN inserts leading zero fill and trailing blank fill (octal 60) into words with less than four characters. As an example, to lockout the FCP primary button Graphic Form (GF), and the secondary button Non-Display (ND), the following coding sequence is used:

```
IARRAY1 (1) = 2
IARRAY1 (2) = 2RGF
IARRAY1 (3) = 2HND
CALL LOCKI  (2HBN, IARRAY1, 0, IERR)
    .
    .
IARRAY1 (1) is the count of the Light Buttons to
            be locked out.
IARRAY1 (2)
 through
IARRAY1 (N) is the DL addresses or BCD mnemonics
            of Light Buttons.
```

5. (Continued)

IARRAY2

List of Console Keyboard button numbers (one per word to be selectively locked out.

IARRAY2 (1) = count of items.
IARRAY2 (2) = first entry
⋮
IARRAY2 (25) = last possible entry.

IERR

Location of an error flag returned by LOCKI. Error values and their significance are listed as follows:

IERR = -1	C_1 and C_2 = NN
= -2	C_1 is illegal.
= -3	C_2 is illegal.
= -4	Specified Light Button entity is not found in DL (FETCI error when DL address was given in IARRAY1).
= -5	Light Button entity was found in DL but could not be returned to DL.
= -6	Specified Light Button entity is not found in DL (SEARI error - end of DL encountered before finding Light Button with proper BCD code).
= -7	DL search error.
= -8	IARRAY2 contains button number not in range 1 to 24.

FCP-32 Application Interface - FORTRAN
Subroutine Specification

Subroutine Ident. RSETI

1. Subroutine Name:

Reset Locked Out Functions Through the Interface.

2. References:

Digigraphic System 270 Function Control Program Specifications, Section 6.

3. Abstract:

RSETI re-establishes or unlocks previously locked-out Keyboard and Light Button controls, thus restoring these controls for console operator use.

4. Functional Description:

RSETI follows after a call to LOCKI and returns use of the specified controls to the console operator.

CALL STATEMENT

CALL RSETI (2HC$_1$ C$_2$, IARRAY1, IARRAY2, IERR)

5. Input/Output Data:

2HC$_1$C$_2$

Hollerith control characters which specify the processing to be performed by RSETI. The specific codes are listed below:

C$_1$ = A

Reset all light controls to unlocked state (IARRAY1 field will be ignored).

C$_1$ = D

Reset selected controls listed in IARRAY1. "D" indicates that IARRAY1 contains DL addresses of primary and/or secondary Light Button controls to be reset.

C$_1$ = B

Reset selected controls listed in IARRAY1. "B" indicates that IARRAY1 contains BCD mnemonics of primary and/or secondary Light Button controls to be reset.

C$_1$ = N

No Light Button controls to be reset (IARRAY1 field will be ignored).

5. (Continued)

C_2 = A

Reset all Console Keyboard buttons to unlocked state (IARRAY2 field will be ignored).

C_2 = S

Reset selected Console Keyboard buttons listed in IARRAY2.

C_2 = N

No Keyboard buttons to be reset - (IARRAY2 field will be ignored).

IARRAY1

List of DL addresses or BCD mnemonics of the Light Button controls to be reset. When BCD mnemonic codes are used, a four character comparison is made against the contents of DL record. FCP primary Light Button mnemonics are right justified and secondary Light Buttons are left justified (See paragraph 6.1.1.3). 3200 FORTRAN inserts leading zero fill and trailing blank fill (octal 60) into words with less than four characters. As an example, to restore the FCP primary button Graphic Form (GF), and the secondary button Non-Display (ND), the following coding sequence is used:

```
IARRAY1(1)  = 2
IARRAY1(2)  = 2RGF
IARRAY1(3)  = 2HND
CALL RSETI (2HBN, IARRAY1, O, IERR)
:
:
```

IARRAY1(1) is the count of the Light Buttons to be reset.

IARRAY1(2) through IARRAY1(N) is the DL addresses or BCD mnemonics of Light Buttons.

5. (Continued)

IARRAY2

List of Console Keyboard button numbers (one per word to be selectively locked out).

IARRAY2 (1) = count of items.

IARRAY2 (2) = first entry.

⋮

IARRAY2 (25) = last possible entry.

IERR

Location of an error flag returned by RSETI. Error values and their significance are listed as follows:

IERR	= -1	C_1 and C_2 = NN.
	= -2	C_1 is illegal.
	= -3	C_2 is illegal.
	= -4	Light Button entity not found.
	= -5	Reset Light Button cannot be replaced in DL.
	= -6	Cannot find all of Light Button entities in DL identified by BCD mnemonics.
	= -7	DL search error.
	= -8	IARRAY2 contains Keyboard Button number not in range 1 to 24.

FCP-32 Application Interface - FORTRAN
Subroutine Specification

Subroutine Ident. CLBCI

1. Subroutine Name:

Create Light Button Control Through the Interface.

2. References:

Digigraphic System 270 Function Control Program Specifications, Section 6.

3. Abstract:

This subroutine allows an application program to create both primary and secondary Light Button control elements.

4. Functional Description:

CLBCI gives the application program the ability to create its own Light Button control elements. These elements fall into one of two classifications:

a. Primary Controls

These controls normally specify no direct action (upon Light Pen activation) in themselves other than the generation of a core display of an associated set of secondaries. In some cases a primary control may have no secondary controls, but rather is in itself a control. In this situation, Light Pen activation of this primary causes a jump to a specified subroutine.

b. Secondary Controls

These controls when picked, cause a transfer of control to a specified subroutine. The core display of secondary controls terminates upon release of the Light Pen switch. A maximum of 16 secondary controls can be established for a specified primary.

When called, CLBCI creates DL entities for the primary and secondary controls specified in the calling parameters and generates a byte stream for the display of the primary control. It then returns to the calling program with the DL address of the primary control's DL entity.

CALL STATEMENT

CALL CLBCI (2HC_1 C_2, KNT, IVERT, IXY, MN, ICAT, IRFW, IENTRY, IDLADD, IERR).

5. Input/Output Data:

2HC_1C_2

Hollerith control characters which specify the nature of input data contained in the call statement. The specific codes are listed below: (where n is the secondary element number)

C_1 = C

The coordinates in the array IXY are Construction Grid coordinates. These coordinates are contained in IXY(2, n) and IXY (3, n) with IXY (1, n) = 0.

C_1 = D

The coordinates in the array IXY are Display Grid coordinates. These coordinates are contained in IXY (1, n) with IXY (2, n) = IXY (3, n) = 0.

C_1 = B

The coordinates in each entry of the array IXY are:
IXY (1, n) = Display Grid coordinates
IXY (2, n) = X Construction Grid coordinate
IXY (3, n) = Y Construction Grid coordinate.

C_2 = F, Blank (60 octal), or zero

The Light Button controls specified in the calling parameters are FCP defined.

C_2 = A

The Light Button controls specified in the calling parameters are application program defined.

KNT

Count of the secondary elements associated with the primary specified. If KNT = 0, no secondaries are associated with the specified primary and the first entry of the array IENTRY contains the address of a subroutine.

5. (Continued)

If KNT ≠ 0 the first entry of IENTRY is ignored. The value of KNT cannot be greater than 16.

IVERT

Vertical display order of the specified primary (1 to N). This value is used to position the secondaries in relation to their primary. If IVERT = "0", the coordinate values in IXY are used for positioning the associated secondaries (see figure CLBCI - 1).

IXY

A three by (KNT+1) array. Each entry of the array contains the following items: (where n is the secondary element number)

If C_1 = C, IXY (1, n) = 0, and IXY (2, n) and IXY (3, n) contain Construction Grid coordinates (see figure CLBCI - 2).

If C_1 = D, IXY (1, n) = Display Grid coordinates and IXY (2, n) = IXY (3, n) = 0 (see figure CLBCI - 2).

If C_1 = B, IXY (l, n) contains Display Grid coordinates and IXY (2, n) and IXY (3, n) contain Construction Grid coordinates. If IVERT is not zero, only the primary is positioned from IXY and all the associated secondaries are positioned by using the vertical display order of the primary.

If C_1 = B or D and IVERT is not zero and the Display Grid coordinates in the first entry of the array IXY are 0, 0 the Display Grid coordinates for the primary are assigned as follows:

If C_2 = F, $X_0 = 1400_8$, $Y_0 = 5500_8$ + ((IVERT-1) Y)

If C_2 = A, $X_0 = 3400_8$, $Y_0 = 0600_8$ + ((IVERT-1) Y)

MN

An array (KNT+1) of BCD mnemonics (maximum of four characters each). The first entry is for the primary control, the remaining entries for its associated secondaries.

5. (Continued)

ICAT

An array (KNT+1) of categories with the first entry for the primary and the remaining entries for its associated secondaries.

IRFW

An array (KNT + 1) of reference words with the first entry for the primary and the remaining entries for its associated secondaries.

IENTRY

An array (KNT+1) of subroutine entry point addresses. If KNT is not "0" the first entry of IENTRY is "0".

IDLADD

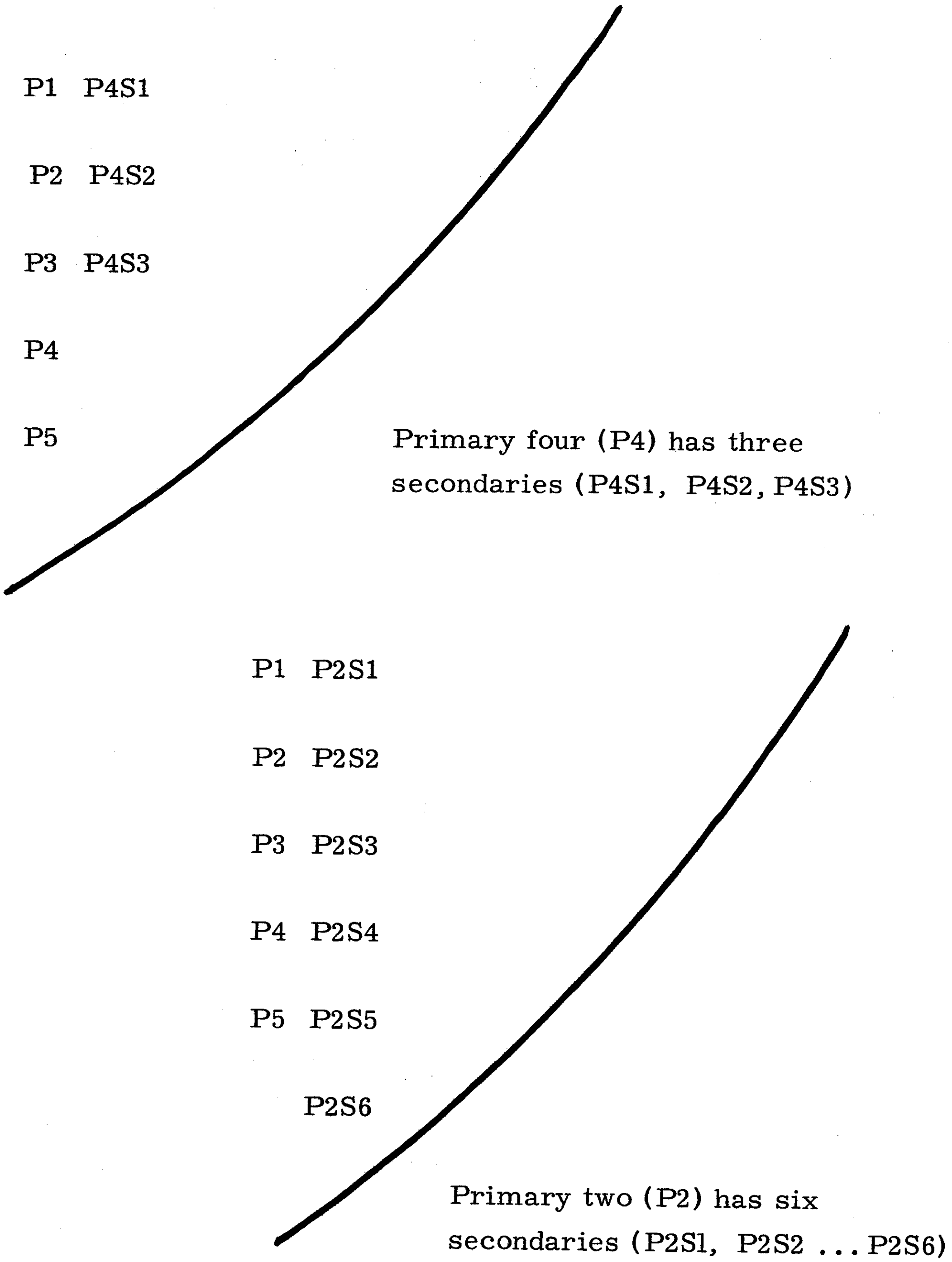
On return from CLBCI, IDLADD contains the DL address of the primary control's DL entity.

IERR

Location of an error flag returned by CLBCI. Error values and their significance are listed below:

IERR = -1	C_1 is illegal.
= -2	C_2 is illegal.
= -3	Unable to enter primary or secondary.
= -4	Unable to display primary.
= -5	KNT is negative or greater than 16.
= -6	IVERT is negative or greater than 63.

5. (Continued)



Primary four (P4) has three secondaries (P4S1, P4S2, P4S3)

Primary two (P2) has six secondaries (P2S1, P2S2 ... P2S6)

Figure CLBCI - 1
Positioning of Secondaries on the Working Surface
Using the Vertical Display Order of Primaries

5. (Continued)

	$C_1 = C$	$C_1 = D$	$C_1 = B$
IXY (1, 1)	Zero	X, Y Display (Primary)	X, Y Display (Primary)
(2, 1)	X Construction (Primary)	Zero	X Construction (Primary)
(3, 1)	Y Construction (Primary)	Zero	Y Construction (Primary)
(1, 2)	Zero	X, Y Display (First Secondary)	X, Y Display (First Secondary)
(2, 2)	X Construction (First Secondary)	Zero	X Construction (First Secondary)
(3, 2)	Y Construction (First Secondary)	Zero	Y Construction (First Secondary)
(1, 3)	Zero	X, Y Display (Second Secondary	X, Y Display (Second Secondary)
(2, 3)	X Construction (Second Secondary)	Zero	X Construction (Second Secondary)
(3, 3)	Y Construction (Second Secondary)	Zero	Y Construction (Second Secondary)
(1, 4)	Zero	X, Y Display (Third Secondary)	X, Y Display (Third Secondary)
(2, 4)	X Construction (Third Secondary)	Zero	X Construction (Third Secondary)
(3, 4)	Y Construction (Third Secondary)	Zero	Y Construction (Third Secondary)
⋮		⋮	⋮
(1, KNTH)	Zero	X, Y Display (Last Secondary)	X, Y Display (Last Secondary)
(2, KNTH)	X Construction (Last Secondary)	Zero	X Construction (Last Secondary)
(3, KNTH)	Y Construction (Last Secondary)	Zero	Y Construction (Last Secondary)

Figure CLBCI - 2

IXY, Input X and Y Coordinates for Construction Grid, Display Grid, or Both

Subroutine Ident. KYBDI

1. Subroutine Name:

Keyboard Assignment Through the Interface.

2. References:

Digigraphic System 270 Function Control Program Specifications, Section 6.

3. Abstract:

This subroutine allows an application program to define or change Keyboard functions.

4. Functional Description:

KYBDI allows the application program to define or redefine the functions of the console Keyboard. Functionally, the application program may define a Keyboard button to operate in either of the two following modes:

a. Verb Control.

In this mode, FCP transfers control to the defined subroutine when the specified Keyboard button is pressed. When interrupt occurs, FCP inhibits processing of other verb controls for the particular console. The application program may enable verb controls while in the interrupt subroutine, but caution must be exercised at this point to prevent losing the return path to the main program.

b. Modifier Control.

In this mode, FCP sets an application defined character flag to '1' when the specified button has been pressed and resets this flag to '0' when the button is released.

Specific Keyboard buttons are defined by FCP for its own use. The remainder are unused and are completely free for application definition. Those buttons defined by FCP may be redefined by the application. However, upon completion of a specific Keyboard use, the application program must reinstate original FCP button definitions.

CALL STATEMENT

CALL KYBDI (NH$C_1C_2C_3$, NUMBER, NTABLE, IX, IERR)

5. Input/Output Data:

NH$C_1C_2C_3$

Hollerith control characters which specify both the processing to be performed by KYBDI and the nature of the input data contained in the call statement. The specific codes are listed below:

C_1 = S

Return a status map of the Keyboard Assignment Table. NUMBER may be '0' in which case NTABLE is a 24-word integer array to which the entire Keyboard Assignment Table will be transmitted. NUMBER may also be an integer in the range of '1' through '24' and NTABLE an integer variable to which the specified table entry is to be transmitted. C_2 and C_3 are ignored.

C_1 = R

Return status map to the Keyboard Assignment Table. NUMBER is an integer constant or variable specifying the number of the button to be returned.
NUMBER must be in the range '1' through '24' or '0' to return 24 words to the Keyboard Assignment Table.

C_1 = D

Define a specified button according to C_2 and C_3.

C_2 = F

A button defined for FCP. NUMBER is an integer constant or variable specifying the number of the button to be assigned. NTABLE is a Hollerith constant or an integer variable containing a Hollerith constant specifying one of the FCP defined button mnemonics. In this case C_3 will be ignored.

C_2 = A

A button defined for an application. C_3 must be present, with NUMBER an integer constant or variable specifying the number of the button to be reassigned.

5. (Continued)

C_3 = V

A button defined as verb control by an application. NTABLE is the name of a subroutine (entry point) in the application main section to which control will be transferred when the specified button has been pressed. Console verb control lockout is imposed by FCP upon interrupt. The application program may enable the verb controls, but it must also assume nested interrupt responsibilities at this time.

C_3 = M

A button defined as a modifier control by an application.

IX

The number of the character in NTABLE that will be set to one by FCP when the specified Keyboard button is pressed, and set to zero when the button is released. Only the low order bit of the character is affected by pressing or releasing the related Keyboard button.

NUMBER

An integer constant or variable containing a value in the range 1 to 24 identifying the button to be assigned, reassigned, or returned to the application program. If zero, all buttons will be copied.

NTABLE

A 24-character array whose content is dependent on NUMBER and C_1, C_2, or C_3. NTABLE can have the following forms:

A 24 word integer array when C_1 = S and NUMBER = 0.

An integer variable when C_1 = S and NUMBER is in range 1 to 24.

A Hollerith constant or integer variable containing Hollerith constant button mnemonics when C_1 = D, and C_2 = F.

A subroutine entry point when C_1 = D, C_2 = F, C_3 = V.
A core address of a character array when C_1 = D, C_2 = A, and C_3 = M.

IERR

Location of an error flag returned by KYBDI. Error values and their significance are listed as follows:

IERR = -1 C_1 is illegal.

= -2 Button specified for definition is currently depressed.

= -3 Illegal NUMBER inserted.

= -4 Button to be restored to Keyboard table has on/off status bit out of synchronization with current status.

= -5 C_2 is illegal.

= -6 C_3 is illegal.

= -7 Mnemonics for FCP button do not compare with legal FCP codes.

Subroutine Ident. TERMI

1. Subroutine Name:

Terminate Current Application Through the Interface.

2. References:

Digigraphic System 270 Function Control Program Specifications, Section 6.

3. Abstract:

This subroutine provides for termination of application-defined Keyboard functions and Light Button control elements.

4. Functional Description:

TERMI terminates the application program, reinstates the FCP defined hardware Keyboard and reinstates FCP Light Button control elements. There is no return to the calling application program from TERMI. Other application programs may not be initiated.

CALL STATEMENT

```
CALL TERMI
```

6.1.4 Fetch Routines. The purpose of the fetch routines is to allow an application program to scan the DL for specific data and retrieve that data when found. Functionally these routines provide for a search of the DL for an entity through use of Key Field, Category Field, and Reference Field masks. When a comparison is found between masked values and a given parameter, the desired entity is returned to the application program. If a match is not found, an indication of this condition is returned.

FCP-32 Application Interface - FORTRAN
Subroutine Specification

Subroutine Ident. SEARI

1. Subroutine Name:

   Searchthe Digigraphic List Through the Interface.

2. References:

   Digigraphic System 270 Function Control Program Specifications, Section 6.

3. Abstract:

   This subroutine provides for a search of the DL for an entity through use of the Key Field, Category Field, and Reference Field.

4. Functional Description:

   A call to this routine causes FCP to search the DL for the entity which satisfies the constraining application parameters. The masks for the Key, Category, and Reference Fields are set up by the application program. When an entity is found whose Key, Category, and Reference Field matches the constraining masks, the search routine returns the DL address of this entity to an application defined integer variable and lists the DL entity in the application defined integer array. If no entity is found, an error flag is returned in the defined integer variable IERR. The application program also has the option of specifying a continue-search operation. In this mode the last DL address returned to FCP is specified as the continue search location.

   CALL STATEMENT

   CALL SEARI (IDLAD, IARRAY, MASK1, $2HC_1$, MASK2, $2HC_2$, IERR)

5. Input/Output Data:

   IDLAD

   An integer variable or constant specifying the DL address at which the search will begin. If IDLAD is zero, the search will begin at the first entity of the DL. If IDLAD contains a DL address, the search will begin at the entity following the DL address. This variable will also receive the DL address of an entity which satisfies the search descriptions.

5. (Continued)

If the programmer would like to continue searching, he can use the IDLAD returned by the search as the starting point for the next search. In this way he can scan the entire DL. When the end of the DL is reached, an indicator is returned to IDLAD (IDLAD = -1).

IARRAY

The name of the integer array which is to contain the entity matching the constraining parameters. If IARRAY(1) is set to zero by the calling program, SEARI will assume the array to be large enough to contain the entire entity, and will return the entire entity.

If IARRAY(1) is set to a positive value by the calling program, SEARI will use the value as a constraint and will not overflow the space provided.

IARRAY(1) is set by SEARI to the number of words in the entity. If the array was not large enough for the entire entity, IERR will contain a special flag (IERR = -2).

IARRAY(2) is the first word of the entity.

MASK1

A two word or four word integer array containing: comparison code (value) and mask word for Key and Category Field search. (+0 mask = no Key Field search). MASK1(1) and (3) = value of comparison. MASK1(2) and (4) = mask for comparison. If C_1 = IR or OR, then MASK1 contains four words. There are two sets of masks and comparison codes to define limits of a search.

$2HC_1$

Hollerith characters which control the use of the array MASK1. The specific codes are listed below:

C_1 = EQ

The Key Field is equal to the masked comparison code.

5. (Continued)

C_1 = NE

The Key Field is not equal to the masked comparison code.

C_1 = LT

The Key Field is less than the masked comparison code.

C_1 = GT

The Key Field is greater than the masked comparison code.

C_1 = GE

The Key Field is greater or equal to the masked comparison code.

C_1 = LE

The Key Field is less than or equal to the masked comparison code.

C_1 = IR

The Key Field lies in the range of masked comparison codes.

C_1 = OR

The Key Field lies outside the range of masked comparison codes.

MASK2

A two or four word integer array containing comparison code (value) and mask for a Reference Field search. (+0 mask = no Reference Field search). There are two sets of masks and comparison codes to define limits of a search.

MASK2(1) and (3) = value of comparison.

MASK2(2) and (4) = mask word for comparison.

If C_2 = IR or OR, then MASK2 contains 4 words.

$2HC_2$

Hollerith characters which control the use of the array MASK2. The specific codes are listed below:

C_2 = EQ, NE, LT, GT, GE, LE, IR, OR.

5. (Continued)

IERR

Location of an error flag returned by SEARI. Error values and their significance are listed below:

IERR	= -1	Indicates an illegal DL address.
	= -2	IARRAY is not large enough for the entity.
	= -3	C_1 or C_2 are illegal character values.

NOTE

1. When the mask code for the Key and Category Fields and Reference Field MASK1(2) and MASK2(2) are both set to zero by the calling program, SEARI will ignore the character codes C_1 and C_2 and will return the next entity to IARRAY. This may be used as a "fetch next entity" call.

2. When using SEARI in a "DO-loop" or in the continue search mode, the application programmer is cautioned to reset the value in IARRAY(1) and to check the returned value in IERR for negative. IARRAY(1) is updated by SEARI to the total size of the entity which matched the input parameters. This value may be larger or smaller than the buffer provided. If the entity exceeded the size of the buffer, only the number of words specified in IARRAY(1) will be returned, to avoid inadvertent destruction of the data in memory cells adjacent to IARRAY. When this occurs, IERR will be set to -2 and IARRAY(1) will tell the calling program how large the entity was. The calling program is assumed to have stored the value input to IARRAY(1) so that it can compare it with the returned value to derive the number of words not returned. IDLAD is set to a value of -1 when the end of the DL is encountered.

FCP-32 Application Interface - FORTRAN
Subroutine Specification

Subroutine Ident. FETCI

1. Subroutine Name:

   Fetch a Digigraphic List Entity Through the Interface.

2. References:

   Digigraphic System 270 Function Control Program Specifications, Section 6.

3. Abstract:

   This subroutine, when called, retrieves a known DL entity.

4. Functional Description:

   A call to this routine transfers the DL entry of a specified entity to the application program. The call statement specifies the DL address of the entity to be transferred and also gives the integer array to which the entry must be sent. This routine provides the application programmer with a direct call for a DL entity.

CALL STATEMENT

```
CALL FETCI (IADD, IARRAY, IERR)
```

5. Input/Output Data:

IADD

An integer variable or constant which gives the DL address of an entity which is to be listed.

IARRAY

The name of the integer array which is to contain the entity. If IARRAY(1) is set to zero by the calling program, FETCI will assume the array to be large enough to contain the entire entity, and will return the entire entity. If IARRAY(1) is set to a positive value by the calling program, FETCI will use the value as a constraint and will not return the specified number of words. IARRAY(1) is set by FETCI to the number of words in the entity and IARRAY(2) is the first word of the entity. If the array was not large enough for the entire entity, IERR will contain a special flag.

5. (Continued)

IERR

Location of an error flag returned by FETCI. Error values and their significance are listed below:

IERR	= -1	The DL address is illegal.
	= -2	IARRAY not large enough for entity (bits 8 - 23).
	= -3	Illegal DL address (bits 0 - 7).

6.1.5 Pick Routines. The application interface pick routines permit an application program to perform the following functions:

a. Request an operator to use the Light Pen to pick a specified parameter.

b. Define a particular entity or a point on an entity as a Pick Table entry.

c. Transfer the content of the Pick Table to memory locations controlled by the application program.

FCP-32 Application Interface - FORTRAN
Subroutine Specification

Subroutine Ident. IPICI

1. Subroutine Name:

   Instruct a Pick Through the Interface.

2. References:

   Digigraphic System 270 Function Control Program Specifications, Section 6.

3. Abstract:

   Through this subroutine an application can request a console operator to use the Light Pen to pick a specified parameter.

4. Functional Description:

   Using IPICI, the application program requests the operator to pick a parameter using the Light Pen. Instructions to the operator are sent through the Message Register. The pertinent Pick Table entry is sent to a specified array in the application program. Set-up of the proper length array is the responsibility of the application programmer. The ability to lock a pick in the Pick Table is also provided.

CALL STATEMENT

CALL IPICI (MESS, 1HC$_1$, NTRY, IERR)

5. Input/Output Data:

MESS

An integer array, seven words or less, which contains the information to be sent to the message register. MESS(1) = number of characters. MESS(2) - (7) = characters.

1HC$_1$

A Hollerith control character which specifies the processing to be performed by IPICI.

C_1 = L

Lock the entry in the Pick Table.

C_1 = N

Do not lock the entry.

5. (Continued)

NTRY

An integer array for storage of the Pick Table entry. NTRY is dimensioned for the largest Pick Table entry expected (four words + NTRY(1)). Upon return from IPICI to the calling program NTRY will appear as follows:

NTRY(1) is the count of the number of words transmitted (two or four).

NTRY(2) contains the Pick Descriptor.

NTRY(3) contains the DL address.

NTRY(4) contains the Construction Grid X coordinate.

NTRY(5) contains the Construction Grid Y coordinate (when the operator has picked the writing point, or a point on an entity).

IERR

Location of an error flag returned by IPICI. Error values and their significance are listed as follows:

IERR = -1 Unable to send message to console operator.
= -2 Unable to recognize Pick Table entry.
= -3 C_1 is illegal.

FCP-32 Application Interface - FORTRAN
Subroutine Specification

Subroutine Ident. DPICI

1. Subroutine Name:

Define a Picked Parameter Through the Interface.

2. References:

Digigraphic System 270 Function Control Program Specifications, Section 6.

3. Abstract:

With this subroutine, an application program can define a particular entity or a point on an entity as a Pick Table entry.

4. Functional Description:

The DPICI routine allows an application programmer to define a parameter as a Pick Table entry for later use. The function of this routine is similar to IPICI except Pick Table entries are application defined. At the option of the application programmer, any of the defined Pick Table entries can be locked in the table, or any specified entry can be cleared from the Pick Table, whether locked or not. DPICI will also clear unlocked, or clear all Pick Table entries at the option of the calling program.

CALL STATEMENT

CALL DPICI (1HC$_1$, IARRAY, IERR)

5. Input/Output Data:

1HC$_1$

A Hollerith control character which specifies the processing to be performed by DPICI. Specific codes are listed below:

C_1 = E

Enter the Pick Table entry contained in IARRAY.

C_1 = A

Clear all locked and unlocked Pick Table entries. IARRAY is ignored.

C_1 = C

Clear the specified Pick Table entry, locked or unlocked, contained in IARRAY.

C_1 = U

Clear all unlocked Pick Table entries. IARRAY is ignored.

5. (Continued)

IARRAY

The name of integer array containing the Pick Table entry. Upon return to the calling program IARRAY will appear as follows:

IARRAY(1) contains count of the array length.
IARRAY(2) is the Pick Descriptor.
IARRAY(3) is the DL address or zero if entry is a point.
IARRAY(4) is the second Pick Descriptor, of X coordinate.
IARRAY(5) is second DL address or Y coordinate.
IARRAY(n) as required, n = 91 maximum.

IERR

Location of an error flag returned by DPICI. Error values and their significance are listed as follows:

IERR = -1 No rrom in Pick Table for entry.
= -2 Pick error.
= -3 Function requested (C_1 = C) is not available.
= -4 C_1 is illegal.

FCP-32 Application Interface - FORTRAN
Subroutine Specification

Subroutine Ident. LPICI

1. Subroutine Name:

List the Pick Table Through the Interface.

2. References:

Digigraphic System 270 Function Control Program Specifications, Section 6.

3. Abstract:

This subroutine provides for transfer of the content of the Pick Table to memory location controlled by the application program.

4. Functional Description:

LPICI transmits the contents of the Pick Table to an application defined array. This routine does not allow an insertion into the Pick Table, nor do the contents of the Pick Table change.

CALL STATEMENT

CALL LPICI (NTABLE)

5. Input/Output Data:

NTABLE

The integer array to which Pick Table entries are transmitted. The maximum size of the array is 91 words. Upon return from LPICI to the calling program, NTABLE will appear as follows:

NTABLE(1), Word count (zero if Pick Table is empty).

NTABLE(2), Pick Descriptor of last entry placed in Pick Table.

NTABLE(3), DL address of last entry placed in Pick Table or zero if a point entry.

NTABLE(4), Pick Descriptor of next to last entry placed in Pick Table or the Y coordinate if point is current entry.

NTABLE(5), DL address of next to last entry placed in Pick Table or the Y coordinate if point is current entry.

NTABLE(n), As required, n = 91 maximum.

Subroutine Ident. PICEXTR

1. Subroutine Name:

   Extract Pick Table Information Through the Interface.

2. References:

   Digigraphic System 270 Function Control Program Specifications, Section 6.

3. Abstract:

   This subroutine will extract information from the Pick Table pertaining to picked entries.

4. Functional Description:

   (See Abstract).

CALL STATEMENT

CALL PICEXTR (NAME, ITYPE, INUM, IADD, IBUFF, IERR)

5. Input/Output Data:

NAME

An integer variable indicating the class of entry to be extracted from the Pick Table.

NAME = 1

Extract a point entry.

NAME = 2

Extract a line entry.

NAME = 3

Extract a non-group non-register entry.

NAME = 4

Extract a group entry.

NAME = 5

Extract a register entry.

ITYPE

An integer variable indicating the type of point, line, entity or register to be extracted from the Pick Table.

5. (Continued)

For a point and line:

ITYPE = 1

Extract a center point or center line entry.

For an Entity:

ITYPE = 10000000B

Extract an Alphanumeric entity.

ITYPE = 11000000B

Extract a Dot entity.

ITYPE = 14000000B

Extract a Line entity.

ITYPE = 15000000B

Extract a Circle entity.

ITYPE = 16000000B

Extract a Circular Arc entity.

ITYPE = 23000000B

Extract a Polystring entity.

For a Register:

ITYPE = 2

Extract the INPUT register.

ITYPE = 3

Extract the SAVE 1 register.

ITYPE = 4

Extract the SAVE 2 register.

ITYPE = 5

Extract the ANGLE register.

ITYPE = 6

Extract the LENGTH register.

ITYPE = 7

Extract the X register.

ITYPE = 8

Extract the Y register.

5. (Continued)

ITYPE = 9

Extract the CATEGORY register.

ITYPE = 10

Extract the REFERENCE WORD register.

ITYPE = 11

Extract the APPLICATION register.

INUM

An integer variable indicating the number of items to be extracted. Note that if INUM > 1, NAME and ITYPE apply to all items extracted.

IADD

An integer variable denoting the address of the next unsearched Pick Descriptor. When this variable = zero, the Pick Table is searched from the bottom up.

IBUFF

A single dimensioned array where the extracted items are stored.

IERR

Location of an error flag returned by PICEXTR. Error values and their significance are listed below:

IERR = -1	The table was searched but the requested number of items was not found. INUM will contain the number of items found.
= -2	PICEXTR failed to recognize Bits 5 to 1 in the Pick Descriptor. INUM will contain the number of items extracted before failure.
= -3	Illegal starting address for the search.

FCP-32 Application Interface - FORTRAN
Subroutine Specification

Subroutine Ident. LASTPIC

1. Subroutine Name:

   Extract the Last "N" Pick Table Entries Through the Interface.

2. References:

   Digigraphic System 270 Function Control Program Specifications, Section 6.

3. Abstract:

   This subroutine will extract the last "N" entries from the Pick Table.

4. Functional Description:

   (See Abstract and example below).

CALL STATEMENT

CALL LASTPIC (N, IADD, IBUFF)

5. Input/Output Data:

N

An integer variable denoting the number of entries to be extracted.

IADD

An integer variable denoting the starting address of the search in the Pick Table. If IADD = zero, the extraction will start from the bottom of the Pick Table. Upon exiting, IADD contains the next unsearched address in the Pick Table. If IADD = zero upon return from this routine, the entire Pick Table has been searched.

IBUFF

A single dimensioned buffer where extracted information is stored starting in IBUFF(2). IBUFF(1) contains an integer denoting the type of the first Pick Table entry extracted.

5. (Continued)

IBUFF(1) = 1

The first Pick Table entry to be extracted is a Point.

IBUFF(1) = 2

The first Pick Table entry to be extracted is a Line.

IBUFF(1) = 3

The first Pick Table entry to be extracted is an Entity.

IBUFF(1) = 4

The first Pick Table entry to be extracted is a Group.

IBUFF(1) = 5

The first Pick Table entry to be extracted is a Register.

IBUFF(1) = 0

The routine can not recognize BITS 5 - 1 in the first Pick Descriptor encountered.

IBUFF(1) < 0

The routine can not recognize BITS 5 - 1 in the first Pick Descriptor encountered after finding |IBUFF(1)| Pick Table entries and storing them in IBUFF.

Example

To extract eight words from the Pick Table, an application program would make the following call:

IADD = 0

CALL LASTPIC (2, IADD, IRAY).

The internal Pick Table is as follows:

PICK TABLE DL Address	Entity Entry
+1 Pick Descriptor	
+2 Y	Point Entry
+3 X	
+4 DL Address	
+5 Pick Descriptor	
+6 DL Address	Line Entry
+7 Pick Descriptor	

5. (Continued)

Upon return to the application program:

IADD = 2

IRAY(1) = 2

$$\begin{array}{ll}
\left.\begin{array}{l}\text{IRAY(2) = Pick Descriptor}\\ \text{(3) = DL Address}\end{array}\right\} & \text{Line Entry}\\
\left.\begin{array}{l}\text{(4) = Pick Descriptor}\\ \text{(5) = DL Address}\\ \text{(6) = X}\\ \text{(7) = Y}\end{array}\right\} & \text{Point Entry}
\end{array}$$

6.1.6 Entity Manipulation Routines. Through use of the entity manipulation routines, an application program can create, enter, modify, and transform a DL entity according to a set of specified rules. The specific entity manipulations permitted through the application interface are as follows:

a. Create an entity using application defined parameters and store the created entity in the DL. This function essentially duplicates creation of entities by the operator at the console.

b. Enter and delete existing entities in the DL.

c. Modify the Descriptor of an entity in the DL.

d. Geometrically transform a graphic entity in the DL.

e. Display a DL.

FCP-32 Application Interface - FORTRAN
Subroutine Specification

Subroutine Ident. EENTI

1. Subroutine Name:

Enter an Entity Through the Interface.

2. References:

Digigraphic System 270 Function Control Program Specifications, Section 6.

3. Abstract:

Through this subroutine, an application program can enter any legal entity (with the exception of a Group entity) into the Digigraphic List.

4. Functional Description:

EENTI allows the application program to assemble an entity and enter this entity into the DL. Proper entity formating is the responsibility of the application programmer and any improper entities may result in erroneous data. Entity formats and Key Field structure are FCP defined. EENTI returns the DL address of a successful DL entry or returns a flag if the entry is not acceptable. This subroutine may not be used to enter groups (see subroutine specifications for EGNTI). The entered entity will be displayed if the entity is displayable and the display bit is set.

CALL STATEMENT

CALL EENTI (IDLAD, IARRAY, IERR)

5. Input/Output Data:

IDLAD

The integer variable where DL address is to be stored.

IARRAY

The name of the integer array which contains the entity to be entered.

IARRAY(1) gives a count of number of words in the entity.

IARRAY(2) is the first word (DL Control word) of the entity.

IARRAY(n), as required.

5. (Continued)

IERR

Location of an error flag returned by EENTI. The error value and significance is listed below:

IERR = -1 Not enough room in the DL for the specified entity.

Subroutine Ident. RPNTI

1. Subroutine Name:

Replace an Entity Through the Interface.

2. References:

Digigraphic System 270 Function Control Program Specifications, Section 6.

3. Abstract:

Through this subroutine, an application program can replace an entity with a changed version of the same entity.

4. Functional Description:

RPNTI allows an application program to replace a specified entity in the DL with a changed entity and still maintain the same DL address. Although group membership may not be changed on a replace, the size of the entity may be changed.

CALL STATEMENT

CALL RPNTI (IDLAD, IARRAY, IERR)

5. Input/Output Data:

IDLAD

The integer variable where the DL address of the entity to be replaced is stored.

IARRAY

The name of the integer array which contain entity to be stored.

IARRAY(1) gives a count of number of words in the entity.

IARRAY(2) is the first word of the entity.

IERR

Location of an error flag returned by RPNTI. Error values and their significance are listed as follows:

IERR = -1 Indicates an illegal DL address.

= -2 Indicates an erroneous parental pointer.

= -3 Indicates DL saturation, or no room in the DL for the entity being replaced. A "no room" return will occur only on variable size entities such as group, polystring, alphanumeric, and register entities.

FCP-32 Application Interface - FORTRAN
Subroutine Specification

Subroutine Ident. DENTI

1. Subroutine Name:

   Delete a Specified Entity Through the Interface.

2. References:

   Digigraphic System 270 Function Control Program Specification, Section 6.

3. Abstract:

   Through this routine, an application program can delete a specified entity from the Digigraphic List.

4. Functional Description:

   DENTI allows an application program to erase a specified entity from the DL. When the entity is a group, the group structure will be broken in the DL, (i.e., the parental pointer will be removed from all members of the group). If an entity is being displayed, it will be erased from the CRT.

CALL STATEMENT

CALL DENTI (IADD, IERR)

5. Input/Output Data:

IADD

An integer variable or constant containing the DL address of the entity to be erased.

IERR

Location of an error flag returned by DENTI. Error values and their significance are listed below:

IERR	= -1	Illegal DL address (bits 8 to 23).
	= -2	Illegal DL address (bits 0 to 7).

FCP-32 Application Interface - FORTRAN
Subroutine Specification

Subroutine Ident. CENTI

1. Subroutine Name:

Create an Entity Through the Interface.

2. References:

Digigraphic System 270 Function Control Program Specification, Section 6.

3. Abstract:

This subroutine creates an entity using application program-defined parameters and stores the created entity in the DL. This function essentially duplicates creation of entities by the operator at the Console.

4. Functional Description:

Through CENTI, an application programmer defines an entity in a manner similar to operator action. Before calling this routine, all of the necessary parameters must be entered in the proper registers and in the Pick Table. If this operation has been successful, the DL address of the entry is returned to the application and the Pick Table is cleared of all unlocked entries. If the operation is not successful, a special flag is returned to indicate failure.

CALL STATEMENT

CALL CENTI (2HC_1C_2, 3HET, NHETM, IRW, ICAT, IVAR1, IVAR2, IERR)

5. Input/Output Data:

2HC_1C_2

Hollerith control characters specifying the processing to be performed by CENTI; C_1 and C_2 cannot be blank. The specific codes are listed below:

C_1 = D

Display this entity.

C_1 = N

Do not display this entity.

5. (Continued)

C_2 = A

The entity is to be added to a group whose DL address is located in IVAR2.

C_2 = N

The entity is not to be grouped and IVAR2 will be ignored.

3HET

Three Hollerith characters specifying the entity type being created. The specific codes are listed below:

ET = DOT

Dot entity to be created.

ET = LIN

Line entity to be created.

ET = CIR

Circle entity to be created.

ET = CAR

Circular arc entity to be created.

ET = PLS

Polystring entity to be created.

ET = GRP

Group entity to be created.

ET = ANU

Alphanumeric entity to be created.

NHETM

Hollerith characters specifying the Entity Type Modifier code (ETM). N, the number of Hollerith characters in ETM, currently must = 3. The specific codes for ETM are listed below:

ETM = SOS

Solid line, standard weight.

5. (Continued)

ETM = HIS

Hidden line, standard weight.

ETM = CES

Center line, standard weight,

ETM = BOS

Boundary line, standard weight.

IRW

An integer variable or constant containing the 24-bit value for the Reference Field.

ICAT

An integer variable or constant which gives a two-digit octal number for the category type. If no category type is desired, a '0' must be entered.

IVAR1

An integer variable where the DL address of the created entity is stored.

IVAR2

When C_2 = A, IVAR2 contains the integer variable which is the DL address of the parent entity. When C_2 = N, then IVAR2 will be ignored, but a dummy parameter must be provided.

IERR

Location of error flag returned by CENTI. Error values and their significance are listed below:

IERR = -1	Illegal entity type.
= -2	The Pick Table does not contain parameters necessary for entity creation.
= -3	Illegal entity type modifier.
= -4	No room in DL for the entity.
= -5	A group is requested but no parent entity is given in IVAR2.
= -6	A group is requested but the parent entity is not found in DL.

5. (Continued)

NOTE

When creating a group, its members which were previously grouped will have the created group as a new parent. Old group structures will be broken so that these entities will be able to have the newly created group for a parent.

FCP-32 Application Interface - FORTRAN Subroutine Specification

Subroutine Ident. MENTI

1. Subroutine Name:

   Modify an Entity Through the Interface.

2. References:

   Digigraphic System 270 Function Control Program Specifications, Section 6.

3. Abstract:

   This subroutine performs modifications to the Descriptor of an existing entity according to prescribed rules. Entity Descriptors are FCP defined.

4. Functional Description:

   MENTI modifies the Descriptor of an existing entity. Prior to the call, the entity to be modified must have been placed in the Pick Table. If the operation is successful, the Pick Table will be cleared of all unlocked entries. An error flag will be returned to the application program in the case of failure.

CALL STATEMENT

CALL MENTI ($2HC_1C_2$, IERR, $3HC_1C_2C_3$, ICAT, IETYPE, IRW)

5. Input/Output Data:

$2HC_1C_2$

Hollerith control characters which indicate the modifications to be made to the specified entity; a full zero field means no change. The specific codes are listed below:

C_1C_2 = DI

Display the specified entity.

C_1C_2 = ND

Nondisplay the specified entity.

C_1C_2 = ER

Erase the specified entity from DL.

5. (Continued)

IERR

Location of an error flag returned by MENTI. Error values and their significance are listed below:

IERR = -1	Illegal code for display, non-display, erase.
= -2	No picked entity.
= -3	Entity not found in DL.
= -4	Erase group error, cannot break group.
= -5	Cannot erase or cannot delete entity from DL.
= -6	Function requested not coded.
= -7	Entity requested for display not displayable.
= -8	Illegal character codes for Category Field, Reference Field, or modifier parameters.
= -9	Cannot replace entity in DL.

$3HC_1C_2C_3$

Hollerith control characters which specify the data contained in the call statement. The specific codes are listed below:

C_1 = C

A category entry will be found in the integer variable ICAT.

C_1 = N

Ignore ICAT.

C_2 = E

An entity form modifier will be found in the integer variable IETYPE.

C_2 = N

Ignore IETYPE.

C_3 = R

A Reference Field modifier will be found in the integer variable IRW.

C_3 = N

Ignore IRW.

5. (Continued)

ICAT

An integer constant or variable which gives category value to be entered in the specified entity.

IETYPE

An entity type modifier to be entered into Key Field of the specified entity. The entry must be an integer variable or a constant. Bits zero through four of IETYPE will replace bits 12 through 16 of the entity's Key Field.

IRW

An integer constant or variable which gives the Reference Field to be entered in the specified entity.

FCP-32 Application Interface - FORTRAN
Subroutine Specification

Subroutine Ident. TRANI

1. Subroutine Name:

   Transform an Entity Through the Interface.

2. References:

   Digigraphic System 270 Function Control Program Specification, Section 6.

3. Abstract:

   This subroutine performs application intitiated modifications of the data portions of a graphic entity to allow the rotation, reflection and transformation of graphic data, and of an alphanumeric entity, to allow the transformation of the starting point.

4. Functional Description:

   TRANI performs the rotate, reflect, and translate functions (with or without a copy) on a specified entity. If the copy option is requested, a new entity is created and its DL address is returned to the application program. The copy may or may not be given the same parental pointer (grouped) as the original, at the option of the application program. If no copy is requested, the transformed entity retains the original DL address. Prior to the call, all of the necessary parameters are entered into the proper registers and the Pick Table.

   The following is required in the Pick Table for Reflect:

   a. The entity to be reflected.
   b. The line designated as the axis of reflection (the center point-line control bit in the Pick Descriptor set to 1).

   The following is required in the Pick Table for Rotation:

   a. The entity to be rotated.
   b. The number of degrees of rotation set in the Angle Register.
   c. A point specifying the axis of rotation.

4. (Continued)

The following is required in the Pick Table for Translation:

a. The entity to be translated.

b. Either the delta X and delta Y for translation obtained from X, Y registers, a point on the entity and a second point, or two points where the translation will be from the first point picked to the second point.

If the original is displayed, the transformed and copied entities will also be displayed. If the operation is successful, the Pick Table will be cleared of unlocked entries.

Alphanumeric entities may be translated using the TRANI routine. If an attempt is made to rotate or reflect alphanumeric entities, either directly or as a member of a group being transformed, only the starting point will move. The standard rules of alphanumerics apply, (i.e., left-to-right, horizontal).

CALL STATEMENT

CALL TRANI(nHC_1C_2, IVAR, IERR)

5. Input/Output Data:

nHC_1C_2

Hollerith control characters which specify the processing to be performed by TRANI; n must = 1 or 2. The specific codes are listed below:

C_1 = N

Indicates that a copy is not desired. C_2 must be omitted.

C_1 = C

Perform the transformation with a copy. C_2 must be omitted when grouping is not desired.

C_1C_2 = CG

Perform the transformation with a copy and assign the parental pointer of the original to the copy. If the parental pointer is a non-zero value, this code will add

5. (Continued)

the copy to the group entity containing the original. If the parental pointer is zero, indicating the original is not in the group, the zero value will be transferred to the copy.

IVAR

An integer variable where the DL address of copy is stored when copy option is requested.

IERR

Location of an error flag returned by TRANI.

NOTE

To ensure that erroneous parameters are not left in the Pick Table from a previous operation, the application program must clear locked and unlocked entries in the Pick Table. The Pick Table contents can be saved and restored by the following call sequence:

```
CALL LPICI(-----)
CALL DPICI(lHA,----)
CALL instruct operator to either pick or enter parameters
     through DPICI(1HE,------)
CALL TRANI(-----)
CALL DPICI (return array obtained through LPICI).
```

FCP-32 Application Interface - FORTRAN
Subroutine Specification

Subroutine Ident. RDNTI

1. Subroutine Name:

   Redisplay all Displayable Entities Through the Interface.

2. References:

   Digigraphic System 270 Function Control Program Specifications, Section 6.

3. Abstract:

   RDNTI redisplays all non-displayed graphic entities within the current frame.

4. Functional Description:

   RDNTI sets the display/non-display bit in all graphic entities to the display state and calls the subroutine DISPDL which regenerates the display byte stream on Buffer Memory according to the current frame and zoom parameters.

CALL STATEMENT

```
CALL RDNTI
```

FCP-32 Application Interface - FORTRAN
Subroutine Specification

Subroutine Ident. DISPDL

1. Subroutine Name:

   Display the Digigraphic List Through the Interface.

2. References:

   Digigraphic System 270 Function Control Program Specifications, Section 6.

3. Abstract:

   DISPDL regnerates the display byte stream for Buffer Memory based on current contents of the Digigraphic List.

4. Functional Description:

   DISPDL examines the entire Digigraphic List, decodes the entity control word, and if a graphic entity and the display bit is set, calls the display byte stream generator for the specific entity type. DISPDL adds the I.D. bytes to the byte stream and writes the new byte stream on Buffer Memory.

CALL STATEMENT

```
CALL DISPDL
```

FCP-32 Application Interface - FORTRAN

Subroutine Specification

Subroutine Ident. <u>EGNTI</u>

1. Subroutine Name:

   Enter a Group Entity Through the Interface.

2. References:

   Digigraphic System 270 Function Control Program Specifications, Section 6.

3. Abstract:

   Through this subroutine, an application program can enter a group entity into the Digigraphic List.

4. Functional Description:

   This routine enters the group entity in the DL, inserts the group's DL address in the Parental Pointer Field of all members of the group and exits with the group DL address in the application integer variable IDLAD. Null groups may be entered, (i. e. groups with no members).

CALL STATEMENT

CALL EGNTI (IDLAD, IARRAY, IERR)

5. Input/Output Data:

IDLAD

The integer variable where DL address of the group is to be stored.

IARRAY

The name of the integer array which contains the group entity to be entered.

IARRAY(1) gives a count of words in the entity.

IARRAY(2) is the DL control word of the entity.

IARRAY(3) is reserved for the Parental Pointer.

IARRAY(4) through (n) contains the DL addresses of the group members.

5. (Continued)

IERR

Location of an error flag returned by EGNTI. Error values and their significance are listed below:

IERR	= -1	Not enough room in the DL for the specified entity.
	= -2	Error in one of the group members DL address. IDLAD will be complemented.

FCP-32 Application Interface - FORTRAN
Subroutine Specification

Subroutine Ident. ADGRI

1. Subroutine Name:

Add an Entity to a Group Through the Interface.

2. References:

Digigraphic System 270 Function Control Program Specifications, Section 6.

3. Abstract:

Through this subroutine, an application program can modify the Parental Pointer Fields of entities.

4. Functional Description:

This subroutine allows an application program to add specified entities to a specified group by placing the required pointers in the pertinent entities (a parental pointer in child and a child pointer in the parent).

CALL STATEMENT

```
CALL ADGRI (ICADD, IPADD, IERR)
```

5. Input/Output Data:

ICADD

An integer array where the DL addresses of the entities to be added to group are stored (children).

ICADD(1) contains a count of the entities to be added to the group.

ICADD(2) through (n) contains the DL addresses.

IPADD

An integer variable where DL address of group entity is stored (parent).

IERR

Location of an error flag returned by ADGRI. Error values and their significance are listed below:

IERR = -1	Illegal DL address of member (ICADD).
= -2	Illegal DL address of group (IPADD).
= -3	Not enough room in the DL to expand the group.
= -4	Child already grouped.

FCP-32 Application Interface - FORTRAN
Subroutine Specification

Subroutine Ident. RMGRI

1. Subroutine Name:

Remove an Entity from a Group Through the Interface.

2. References:

Digigraphic System 270 Function Control Program Specification, Section 6.

3. Abstract:

Through this subroutine, an application program can remove specified entities from their related group.

4. Functional Description:

This subroutine will allow application to remove specified entities from a specified group, by removing the required pointers in both entities (a parental pointer in the child and a child pointer in the parent). Note that if all the entities of a group have been removed, the group is deleted.

CALL STATEMENT

CALL RMGRI (ICADD, IPADD, IERR)

5. Input/Output Data:

ICADD

An integer array where DL addresses of entities to be removed from group is stored.

ICADD(1) contains the count of entities to be removed.

ICADD(2) through (n) contains the DL addresses of the entities to be removed.

IPADD

An integer variable where DL address of group is stored (parent).

IERR

Location of an error flag returned by RMGRI.

Error values and their significance are listed below:

IERR	= -1	Illegal address of child entity.
	= -2	Illegal address of parent.

FCP-32 Application Interface - FORTRAN
Subroutine Specification

Subroutine Ident. BRGRI

1. Subroutine Name:

   Break a Group Through the Interface.

2. References:

   Digigraphic System 270 Function Control Program Specification, Section 6.

3. Abstract:

   Through this routine an application program can break up a complete group structure.

4. Functional Description:

   This subroutine allows application to break group relationships. This is accomplished by deleting a group and removing all parent pointers in the members of the group. This also will remove a specified group from its parent group.

CALL STATEMENT

```
CALL BRGRI (IDLAD, IERR)
```

5. Input/Output Data:

IDLAD

An integer variable where address of group entity to be broken is stored.

IERR

Location of an error flag returned by BRGRI. The error value and its significance is listed below:

IERR = -1	Illegal DL address.

6.1.7 Control Surface Routines. Through the control surface routines, specific data and control manipulations can be performed by an application program on the Digigraphic Control Surface and Working Surface. These include:

a. Entrance and display of application defined parameters in specific Control Surface registers.

b. Retrieval by the application program of data from specified Control Surface registers.

c. Repositioning of the Tracking Cross on any part of the Working Surface according to coordinates provided by the application program.

d. Redefinition of the size and position of the Working Surface Frame on the CRT, and its size and position on the Construction Grid.

FCP-32 Application Interface - FORTRAN
Subroutine Specification

Subroutine Ident. EREGI

1. Subroutine Name:

   Enter a Register Through the Interface.

2. References:

   Digigraphic System 270 Function Control Program Specification, Section 6.

3. Abstract:

   This subroutine enters and displays application defined parameters in specified Control Surface Registers.

4. Functional Description:

   EREGI enters and forces display of contents of the specified register. Information to be entered must agree in format with the format defined for the register. Previous register contents are destroyed. The content of numeric registers is stored as a binary integer or a floating point number. Registers are addressed by either their mnemonic code or by their DL address. Integer and floating point registers are displayed as right justified BCD strings, and BCD registers are displayed as left justified BCD strings. Once a value is placed in a register, it remains unchanged until the next value is entered. To clear a register, KNT is set to zero. Registers defined by FCP are set initially to zero.

CALL STATEMENT

CALL EREGI (3HC$_1$C$_2$C$_3$, R, KNT, NTRY, IERR)

5. Input/Output Data:

3HC$_1$C$_2$C$_3$

Hollerith control characters which specify both the processing to be performed by EREGI and the nature of the input data contained in the call statement. The specific codes are listed below:

C$_1$ = M

A Hollerith register mnemonic follows in R.

5. (Continued)

C_1 = A

The register entity's DL address follows in R.

C_2 = C

Display the register contents from core.

C_2 = D

Display the register contents from drum.

C_2 = Zero

Display the register contents from drum.

C_2 = Blank (60 octal)

Display the register contents from drum.

C_3 = L

Lock the register contents.

C_3 = U

Unlock the register contents.

C_3 = Zero

Unlock the register contents.

C_3 = Blank (60 octal)

Unlock the register contents.

R

If C_1 = M, R contains one through four Hollerith characters which is the register's mnemonic code.

If C_1 = A, R contains the register entity DL address.

KNT

An integer variable or constant specifying either the number of characters in NTRY including the end-of-message character (octal 14) if the register is alphanumeric, a 1 for an integer register, or a 2 for a floating point register.

NTRY

The name of array which holds the value to be placed in the specified register. BCD codes start in the left most character position of NTRY(1). Entries for BCD registers must end with the end-of-message character (octal 14).

5. (Continued)

IERR

Location of an error flag returned by EREGI. Error values and their significance are listed below:

IERR = -1 R parameter is neither A nor M.

= -2 Error from SEARI subroutine.

= -3 Error from FETCI subroutine.

= -4 Register requested is null.

= -5 Error from the display program when KNT = 0 and the register is empty.

= -6 Error from RPNTI when KNT = 0 and the register is displayed and empty.

= -7 Error from DISPDL when KNT = 0 and the register is displayed and empty.

= -8 Error from RPNTI when KNT = 0 and the register is not empty and not displayed.

= -9 Error from the display program when KNT = 0 and the register is not empty and not displayed.

= -10 Error from RPNTI when register is displayed and KNT is positive.

= -11 Error from DISPDL when register is displayed and KNT is positive.

= -12 Error from the display program when KNT is positive.

= -13 Error from RPNTI when KNT is positive.

= -14 Floating point KNT is negative.

= -15 Floating point KNT is not 2.

= -16 KNT for a BCD register is greater than 13.

= -17 KNT for a BCD register is negative.

= -18 KNT for an integer register is negative.

FCP-32 Application Interface - FORTRAN
Subroutine Specification

Subroutine Ident. RREGI

1. Subroutine Name:

   Read a Register Through the Interface.

2. References:

   Digigraphic System 270 Function Control Program Specifications, Section 6.

3. Abstract:

   Through this subroutine, an application program can obtain the contents of a specified Control Surface register.

4. Functional Description:

   RREGI reads the contents of the specified register and stores the results in an application defined array. The contents of the register remains unchanged after this operation. The application program must set up the proper array lengths for each register to be read. The contents of alphanumeric registers are left justified in the array, so that the highest register character is in the first character position, the next highest in the second character position, and so on.

CALL STATEMENT

CALL RREGI (1HC, R, KNT, NTRY, IERR)

5. Input/Output Data:

1HC

A Hollerith control character which specifies the nature of the input contained in the call statement. The specific codes are listed below:

C = A

The register entity's DL address follows in R.

C = M

A Hollerith register mnemonic follows in R.

R

If C = M, R contains one to four Hollerith characters which is the register's mnemonic code.

If C = A, R contains the register entities' DL address.

5. (Continued)

KNT

An integer variable which the RREGI subroutine sets to the number of characters read from an alphanumeric register or the number of 24-bit words read from a numeric register.

NTRY

The name of array into which the contents of the specified register is read. Note that NTRY may be integer or real depending on register format.

IERR

Location of an error flag returned by RREGI. Error values and their significance are listed below:

IERR = -1 C is an illegal character.

= -2 The register entity identified by the DL address in R cannot be found.

= -3 The register entity identified by the mnemonic code in R cannot be found.

Subroutine Ident. FRAMI

1. Subroutine Name:

   Change the Frame Through the Interface.

2. References:

   Digigraphic System 270 Function Control Program Specifications, Section 6.

3. Abstract:

   This routine allows an application program to change the frame size, shape, position, and/or Zoom Index value.

4. Functional Description:

   FRAMI allows the calling program to redefine the frame shape and location on the CRT, move the frame projection to a new position on the Construction Grid, or to change the size of the frame projection on the Construction Grid (Zoom in or out on an image).

   Before returning to the calling program, FRAMI reviews the Digigraphic List to display all entities bounded by the new frame.

CALL STATEMENT

CALL FRAMI (4H$C_1C_2C_3C_4$, MODFY, IDGC, ICXY, IXYD, IZI, IERR)

5. Input/Output Data:

4H$C_1C_2C_3C_4$

Four Hollerith control characters which specify the processing to be performed by FRAMI. The specific codes are listed below:

C_1 = C

Change the frame center of the Display Grid.

C_1 = N

Do not change the frame center of the Display Grid.

C_2 = C

Change the frame center of the Construction Grid.

C_2 = N

Do not change the frame center of the Construction Grid.

C_3 = C

Change the frame XY dimension.

5. (Continued)

C_3 = N

Do not change the frame XY dimension.

C_4 = C

Change the Zoom Index value.

C_4 = N

Do not change the Zoom Index value.

MODFY

The frame shape Modifier: 0 = no change, 1 = rectangle.

IDGC

A two word integer array containing Display Grid Coordinates for the center of the frame. IDGC(1) contains the X coordinate and IDGC(2) contains the Y coordinate. Only the low order 12 bits of each word are used, and sign extension is required in both words.

ICXY

A two word integer array containing Construction Grid Coordinate for the center of the frame. ICXY(1) contains the X coordinate, and ICXY(2) contains the Y coordinate. Sign extension in bit 22 is required.

IXYD

A two word integer array containing the X and Y dimensions for the distance from the center of the frame to the side and top of the frame. IXYD(1) contains the distance from center to the edge (X dimension). The value for a 17-inch width is 1700 decimal. IXYD(2) contains the distance from the center to the top (Y dimension). The value for an 11-inch height is 1100 decimal.

IZI

An integer variable containing a new Zoom Index value. A Zoom Index value must be in the range -22 to +11 decimal.

5. (Continued)

IERR

Location of an error flag returned by FRAMI. Error values and their significance are listed below:

IERR = -1	C_1 is illegal.
= -2	C_2 is illegal.
= -3	C_3 is illegal.
= -4	MODFY is negative or greater than 1.
= -5	Error from DISPDL.
= -6	A coordinate is illegal or the frame is off the CRT.

NOTE

When C_1, C_2, or C_3 = N, the corresponding parameter will be ignored. However, a dummy parameter must be inserted in its place in the call statement to achieve proper transmission of parameters.

FCP-32 Application Interface - FORTRAN
Subroutine Specification

Subroutine Ident. RPTCI

1. Subroutine Name:

   Reposition the Tracking Cross Through the Interface.

2. References:

   Digigraphic System 270 Function Control Program Specifications, Section 6.

3. Abstract:

   Through this subroutine, an application program can reposition the tracking cross on any part of the Working Surface.

4. Functional Description:

   This subroutine repositions the tracking cross to the coordinates specified by this application program.

CALL STATEMENT

CALL RPTCI (ICXY, IERR)

5. Input/Output Data:

ICXY

Application specified Construction Grid coordinates for the tracking cross:

ICXY(1) = X Coordinate

ICXY(2) = Y Coordinate.

IERR

Location of an error flag returned by RPTCI. Error values and their significance are listed below:

IERR = -1 The Construction Grid coordinates are outside the display frame.

= -2 There is a tracking process currently in operation; the call must be reinitiated until successfully executed or until another action is initiated.

CONTROL DATA MAIN OFFICE 8100 34th AVENUE SOUTH, MINNEAPOLIS 20, MINNESOTA

SALES OFFICES
ALAMOGORDO • ALBUQUERQUE • ATLANTA • BEVERLY HILLS • BOSTON
CAPE CANAVERAL • CHICAGO • CLEVELAND • COLORADO SPRINGS • DALLAS
DAYTON • DENVER • DETROIT • DOWNEY, CALIF. • HONOLULU • HOUSTON
HUNTSVILLE • ITHACA • KANSAS CITY, KAN. • MINNEAPOLIS • NEWARK
NEW ORLEANS • NEW YORK CITY • OAKLAND • OMAHA • PHILADELPHIA
PHOENIX • PITTSBURGH • SACRAMENTO • SALT LAKE CITY • SAN BERNARDINO
SAN DIEGO • SAN FRANCISCO • SEATTLE • WASHINGTON, D.C.

INTERNATIONAL OFFICES
FRANKFURT, GERMANY • HAMBURG, GERMANY • STUTTGART, GERMANY
ZURICH, SWITZERLAND • MELBOURNE, AUSTRALIA • SYDNEY, AUSTRALIA
CANBERRA, AUSTRALIA • ATHENS, GREECE • LONDON, ENGLAND • OSLO, NORWAY
PARIS, FRANCE • STOCKHOLM, SWEDEN • MEXICO CITY, MEXICO, (REGAL
ELECTRONICA DE MEXICO, S.A.) • OTTAWA, CANADA, (COMPUTING DEVICES OF
CANADA, LIMITED) • TOKYO, JAPAN, (C. ITOH ELECTRONIC COMPUTING
SERVICE CO., LTD.)

PUB. NO. 601-46500

DIGIGRAPHIC LABORATORIES CONTROL DATA
CORPORATION

NORTHWEST INDUSTRIAL PARK, 3RD AVE. • BURLINGTON, MASSACHUSETTS 01804