IBM Technical Newsletter

File Number	1130-25
Re: Form No.	Y27-7174-0
This Newsletter No.	Y27-7183
Date	October 31, 1968
Previous Newsletter Nos.	None

IBM 1130/2250 Graphic Subroutine Package for FORTRAN IV

The attached pages include corrections to and clarifications of the above publication. Changes are indicated by a vertical line adjacent to the affected text and a bullet (•) at the left of the title of a changed illustration. Pages to be inserted and/or removed are listed below.

5-7
15-18
25-26
29-30
33-34
39-42
45-48
57-58
65-68
85-86
91-96
101-102
117-120
143-144.2
153-154

Please file this cover letter at the back of the manual to provide a record of changes.

Restricted Distribution

IBM Corporation, Programming Publications, Dept. 637, Neighborhood Road, Kingston, N.Y. 12401

PRINTED IN U.S.A.

ILLUSTRATIONS

FIGURES

TABLES

CHARTS

The 1130/2250 Graphic Subroutine Package for Basic FORTRAN IV (GSP) is a set of subroutines, control blocks, and communication areas which are used in conjunction with the IBM 2250 Display Unit Model 4 attached to an 1130 Computing System. The GSP gives the FORTRAN or Assembler-language programmer the capability of displaying images in the form of lines, points, and characters on the screen of the 2250. The GSP also provides for communication between the 2250 operator and the program.

GENERAL CHARACTERISTICS OF THE GSP

The GSP may be divided into two groups of subroutines: external and internal. The external subroutines are those subroutines which are called by the user's program. The internal subroutines are transparent to the programmer and are not entered from the user's program. The internal programs are normally called by other GSP subroutines. To support the interaction between the user's program, the external subroutines, and the internal subroutines, the GSP also uses several control blocks and communication areas (see Figure 1).

The GSP uses a system display, a skeleton display program containing no displayable data but constructed to permit the attachment of other display programs. The display programs (graphic elements) which are attached to the system display may be built by the GSP as a result of CAII statements in the user's program or may permanently reside within certain GSP subroutines.

User-defined display programs are built through calls to the external image management and image generation subroutines. These subroutines process the data passed in the calling arguments, and call upon internal subroutines to do some further processing. From the data passed, display programs are built. The main processing program may then invoke GSP subroutines which connect the newly built display program to the system display and incorporate it into the system display's regeneration cycle. These subroutines may also be called upon to disconnect display programs from the system display.

Display programs which permanently reside within the GSP are brought into the regeneration cycle in a similar manner. However, image management and image generation functions are unnecessary since these display programs are permanently defined and need no modification.

To handle communications between the program and the 2250 operator, the GSP provides a subroutine which services interrupts from the light pen and from the keyboards. This subroutine may pass control to other internal subroutines to modify the display or build new display programs as the light pen or keyboards are being used.

GSP/DISK MONITOR SYSTEM INTERFACE

The GSP operates under the direction and control of the Disk Monitor System, Version 2 (DMV2). Figure 2 illustrates the actions taken by the DMV2 beginning with recognition of the *G2250 Supervisor Control Record through the passing of control to the mainline program. In general, the DMV2 supports the GSP by (1) enabling the user to declare his intention to use the GSP, (2) converting GSP subroutines from Disk System Format to Core Image Format, and (3) loading any GSP subroutines which must reside in core in order for the GSP to operate.

The GSP is initially accessed through the use of the *G2250 Supervisor Control Record. The *G2250 card causes GCOM, the GSP communication area, to be loaded as the first in-core subroutine after the mainline program. The option of requesting other internal subroutines to be loaded is also provided by the *G2250 card.

When an XEQ Monitor Control Record, which indicates that a Supervisor Control Record(s) is to follow, is detected in the job stream, control passes to the Supervisor Control Record Analyzer. This subroutine examines the *G2250 record and, after converting the information contained on it into name code, places this information in the Supervisor Control Record Area (SCRA). The information is now available to the Core Load Builder. When the Core Load Builder constructs the Load Table (4), the first names entered after the mainline program are those of GSP subroutines found in the SCRA.

The purpose of having the GSP subroutines as the initial entries in the Load Table is to insure that these subroutines reside in the lower 8K of main storage. Residence below location 8192 is necessary

The purpose of having the GSP subroutines as the initial entries in the Load Table is to insure that these subroutines reside in the lower 8K of main storage. Residence below location 8192 is necessary because the Graphic Short Branch order (GSB) is restricted to referencing locations within the lower 8K of storage. The GSB is used in GCOM, in the optional character generation subroutine (GCHAR or GUPER), and the optional Verification Direct Entry subroutine (GSP12).

The Load Table is used by the Core Load Builder to build a core load. Each name in the Load Table is searched for in the Location Equivalence Table (LET) to obtain the address of the specific subroutine on the disk. As each subroutine is found it is converted from disk system format to core image format and stored in the Core Image Buffer (CIB). The CIB is an area on the disk which is used as a temporary storage area for the portion of the core load which is to reside below location 4096.

Immediately after the time the Core Load Builder converts the word which is to occupy location 4097, the remaining subroutines are stored directly into core. As

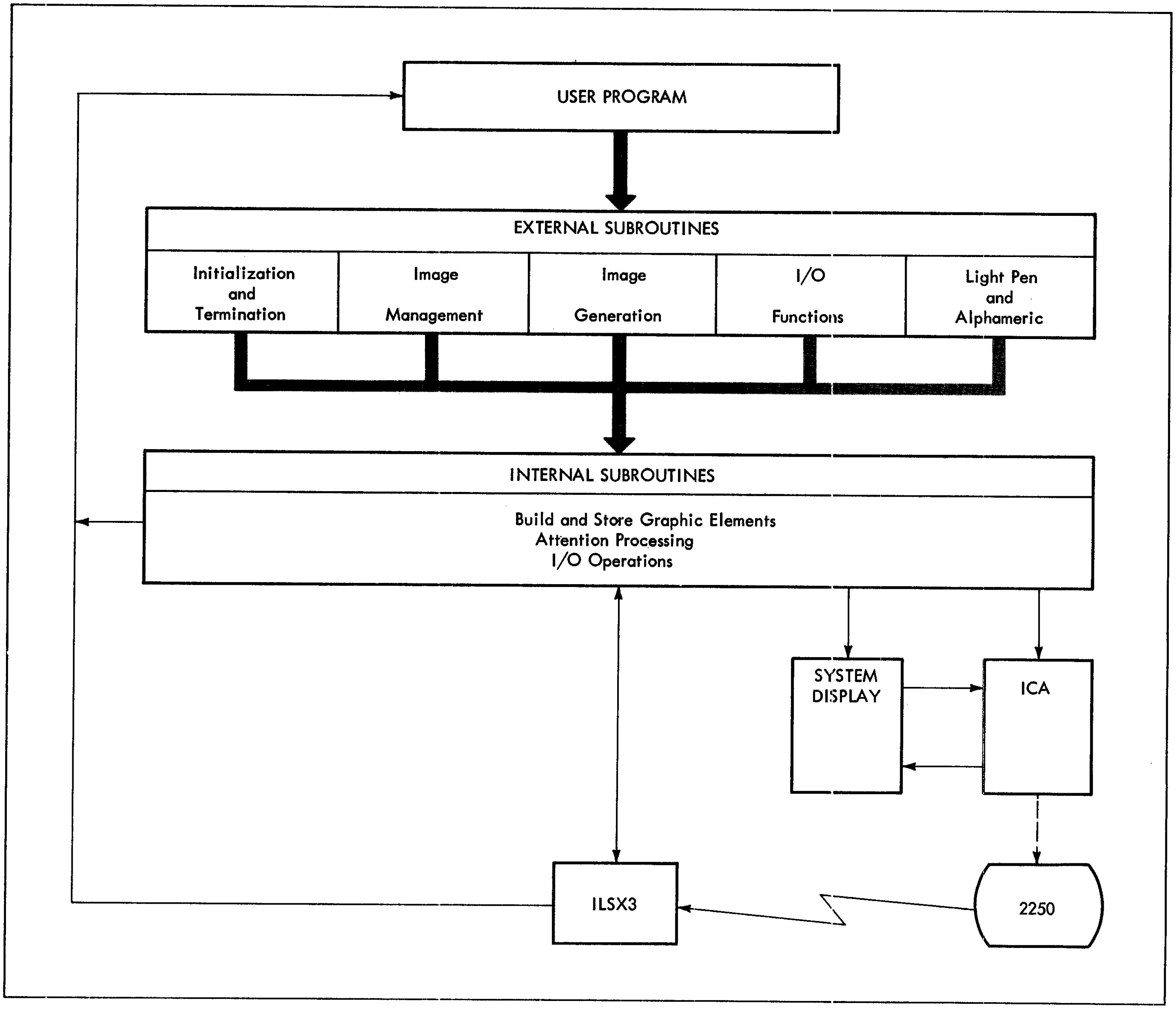


Figure 1. GSP Overview

ICA preface, sets the pointers to the specified addresses, and sets the indicators to standard values. An element correlation control block (ECCB) for the ICA is built using the assigned correlation value, and the ICA is indicated as active.

The Generation Control Area (GCA) may be completely initialized by GCAIN, or initialized in part by any of the following subroutines: SSCAL, which sets the scaling values; SINDX, which sets the index values for accessing data arrays; SINCR, which sets the increment values; SDATM, which sets the x- and y-input data mode; SGRAM, which sets the output graphic mode; and SSCIS, which sets the scissor option. The internal subroutine GSP10 is used by GCAIN and SSCAL for scaling data, and by GCAIN and SINCR for storing data in the GCA.

The subroutine GSPTM terminates the use of the GSP.

IMAGE MANAGEMENT

The image management subroutines provide a means of defining, modifying, and deleting graphic elements within an image construction area (ICA), or creating and including in an order program graphic elements outside of an ICA. Figure 6 shows the subroutines called by the program to perform unique functions. The following internal subroutines perform common functions for the image management subroutines.

GSP01 saves the status of the calling program upon entering a subroutine, stores any arguments in save areas provided in GCOM, and provides a mechanism for returning to the next sequential instruction in the calling program upon completion of the subroutine, or upon encountering an error condition.

GSP02 stores overhead orders for an element within an ICA.

GSP03 builds an element correlation control block (ECCB) for a specified element, searches for a previously defined ECCB, initializes the generation control fields in GCOM, or transfers data between the ICA preface and GCOM for update functions.

GSP04 (Push-down) moves non-deleted elements to lower core positions within an ICA, overlaying elements that have been indicated as deleted, to make more area available within the ICA for the creation of new elements or for the extension of existing elements.

GSP05 (Push-up) moves all elements following a specified element to higher core positions within an ICA so that the specified element may be extended. GSP05 calls GSP04 to provide more area if the ICA is full but there are elements indicated as deleted.

IMAGE GENERATION

The image generation subroutines cause the generation of graphic orders necessary to position and display an image on the 2250 screen. Figure 7 shows the subroutines called by the program to perform unique functions.

The internal subroutines perform common functions for the creation of the graphic orders within elements in the ICA. In addition to the function described in Image Management, GSP02 can convert absolute data to incremental graphic orders, can generate graphic short branch orders to character stroke subroutines, and can generate absolute and incremental graphic orders. The internal subroutine GSP06 scissors portions of an image that fall outside of specified grid boundaries so that only that part of an image within the boundaries is displayed.

I/O OPERATIONS OF THE 2250

I/O commands to the 2250 are issued by DSPYN, the GSP I/O subroutine. This subroutine is used by other GSP subroutines to do I/O functions and can be directly referenced by Assembler-language programs. Interrupts originating from the 2250 keyboards and light pen also cause I/O functions to be performed. This topic is covered in the section "Attention Processing." Figure 8 shows subroutines that use DSPYN to perform I/O operations, and the flow of control and data.

GSPIN, the initialization subroutine, prepares the GSP to display data by calling upon DSPYN to start regeneration of the system display. The associated I/O Control Command (IOCC) which specifies the address of the system display, is sent to the channel interface section of the 2250. The 2250 then begins to access and decode the order program.

EXEC enters the address of the image entity in the system display, resets the no-operation bit in the branch order to the image entity, and calls DSPYN to start regeneration. The IOCC then directs the 2250 to begin accessing orders for the display section of the 2250.

TMDSP resets the no-operation bit in the graphic branch order, causing the ICA to be bypassed during the regeneration cycle. DSPYN is again called to start the system display.

SPFKL lights the programmed function keyboard (PFKB) lights and (optionally) enables those keys that have been lighted. This subroutine constructs two words within DSPYN from an array specified in the calling sequence. The bit configuration in the two words represents the PFKB indicators which are to be lighted. DSPYN issues the command to Set Programmed Function Indicators and the two words are transferred to the 2250 to light the indicators.

GSPTM terminates use of the GSP. DSPYN turns off (bit configuration of zeros) the PFKB lights and issues the command which stops regeneration. Regeneration ceases and all registers and values are returned to their normal status.

The Assembler-language programmer can call DSPYN directly and request each I/O function. The individual I/O functions are included in the description of the DSPYN subroutine.

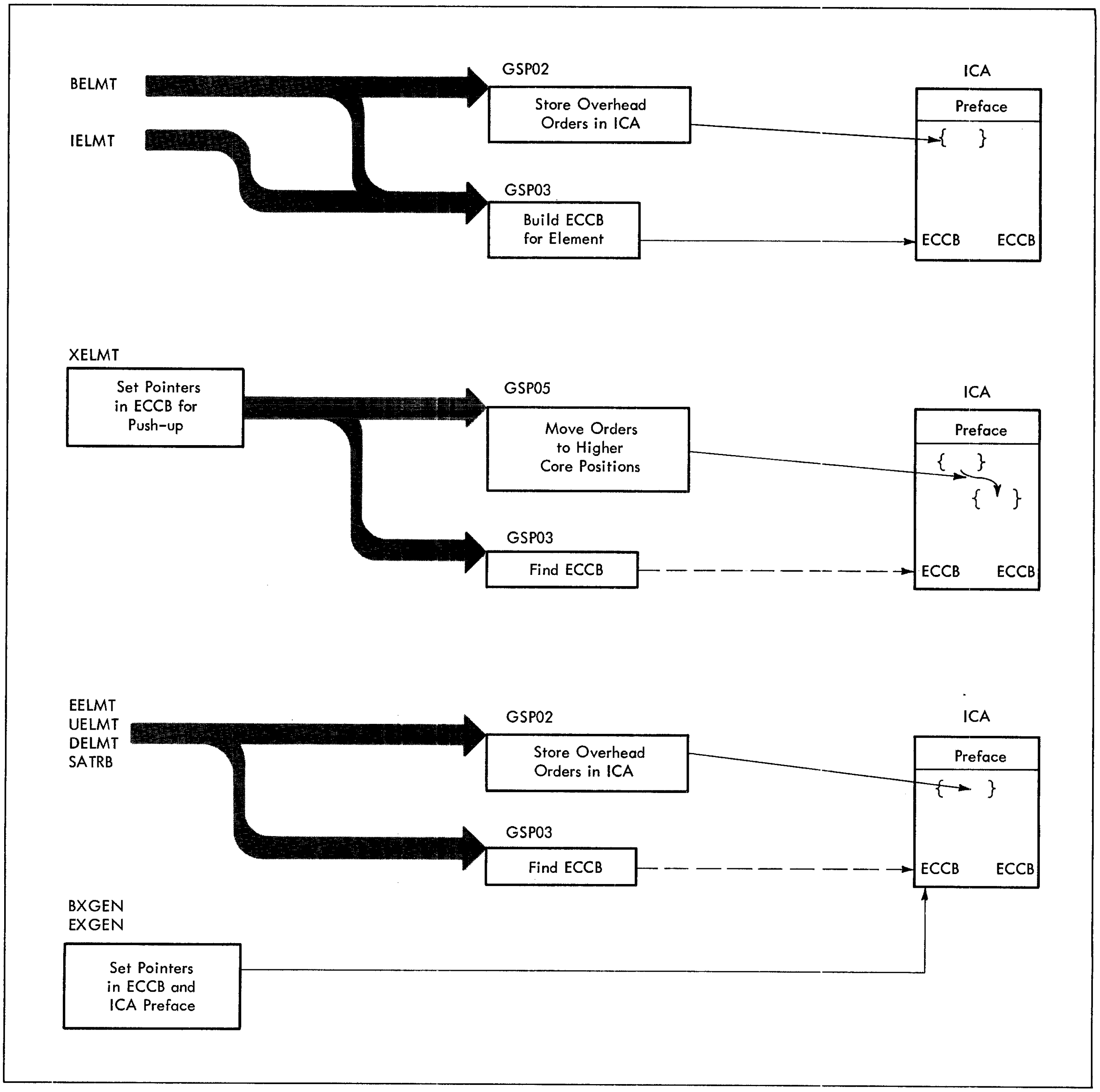


Figure 6. Image Management

LIGHT PEN FACILITIES

The light pen subroutines are those which facilitate communication between the 2250 operator and the program through the use of the light pen. These subroutines provide the services of locating positions on the 2250 screen and dynamically building graphic elements into the ICA through light pen tracking.

LOCATING POSITIONS

To locate a position on the screen, one of three subroutines may be called: LOCPN, LOCND, or LCPOS. Figure 9 shows the subroutines used for locating positions on the screen. If the operator desires to determine the coordinates of a specific point on the screen, he may signal the program to call LOCPN or LOCND. A call to these subroutines activates a branch in the system display to the scanning pattern. A call to LOCPN immediately activates the branch, while a call to LOCND does not activate the branch until the light pen switch is closed. Once the scanning pattern is displayed, a normal light pen attention may be generated to put the coordinates of the position of the light pen in an array specified by RQATN. A call to the LCPOS subroutine positions the tracking symbol at the specified coordinates by activating a branch in the system display to the tracking symbol and entering the x and y coordinates into an order in GSP09 which positions the tracking symbol.

LIGHT PEN TRACKING

The light pen tracking feature of the GSP permits the operator to dynamically build graphic elements through manipulation of the tracking symbol. As the tracking symbol is moved about the 2250 screen, points may be fixed and used as reference locations by graphic orders which are concurrently being built into a tracking entity. Figure 10 shows the interaction of the light pen tracking subroutines and the sequence of events when an interrupt occurs on the tracking symbol.

A call to the TRACK subroutine provides the environment necessary for light pen tracking by initiating the display of the tracking symbol and having an ECCB built for a tracking entity. The graphic orders

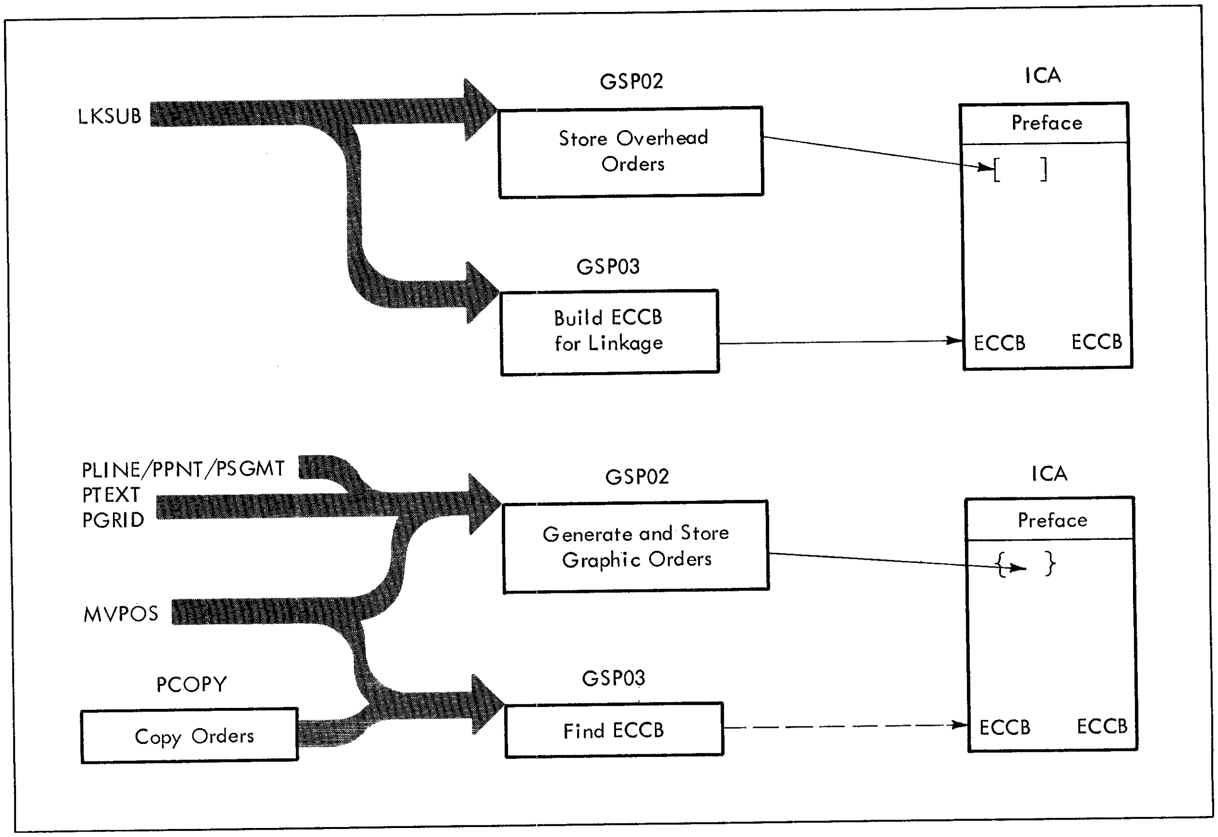
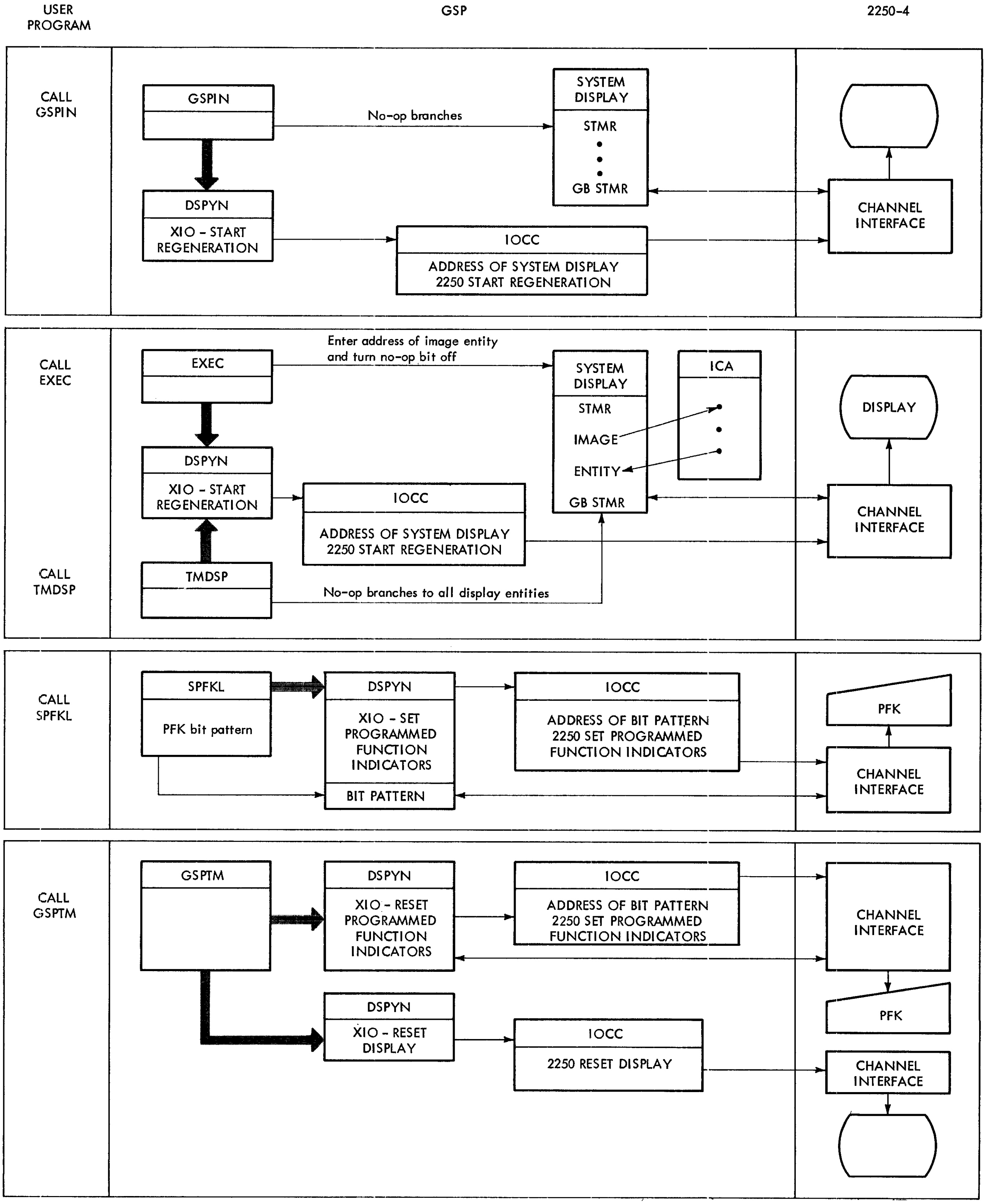


Figure 7. Image Generation



• Figure 8. GSP I/O Operations

error processing subroutine gains control. Upon return from the error processing subroutine, control is passed to the return function of GSP01. If no error processing subroutine has been specified, two WAIT instructions allow error information to be displayed in the accumulator and extension lights. The error processing subroutine IERRS will cause a print-out of error information on the console printer.

Although the existence of an error condition within a subroutine will cause the termination of that subroutine, GSP01 causes control to be passed to the next sequential instruction in the external calling program so that the program will not be interrupted. The location to which a subroutine returns (see Figure 15) upon encountering an error condition is determined by the usage of the save function of GSP01 which, when used by a called subroutine, establishes the address of the next sequential instruction (NSI) in the calling program. If the save function of GSP01 is not called by a subroutine and an error condition is encountered, control is passed to the address of the last NSI established by a called subroutine.

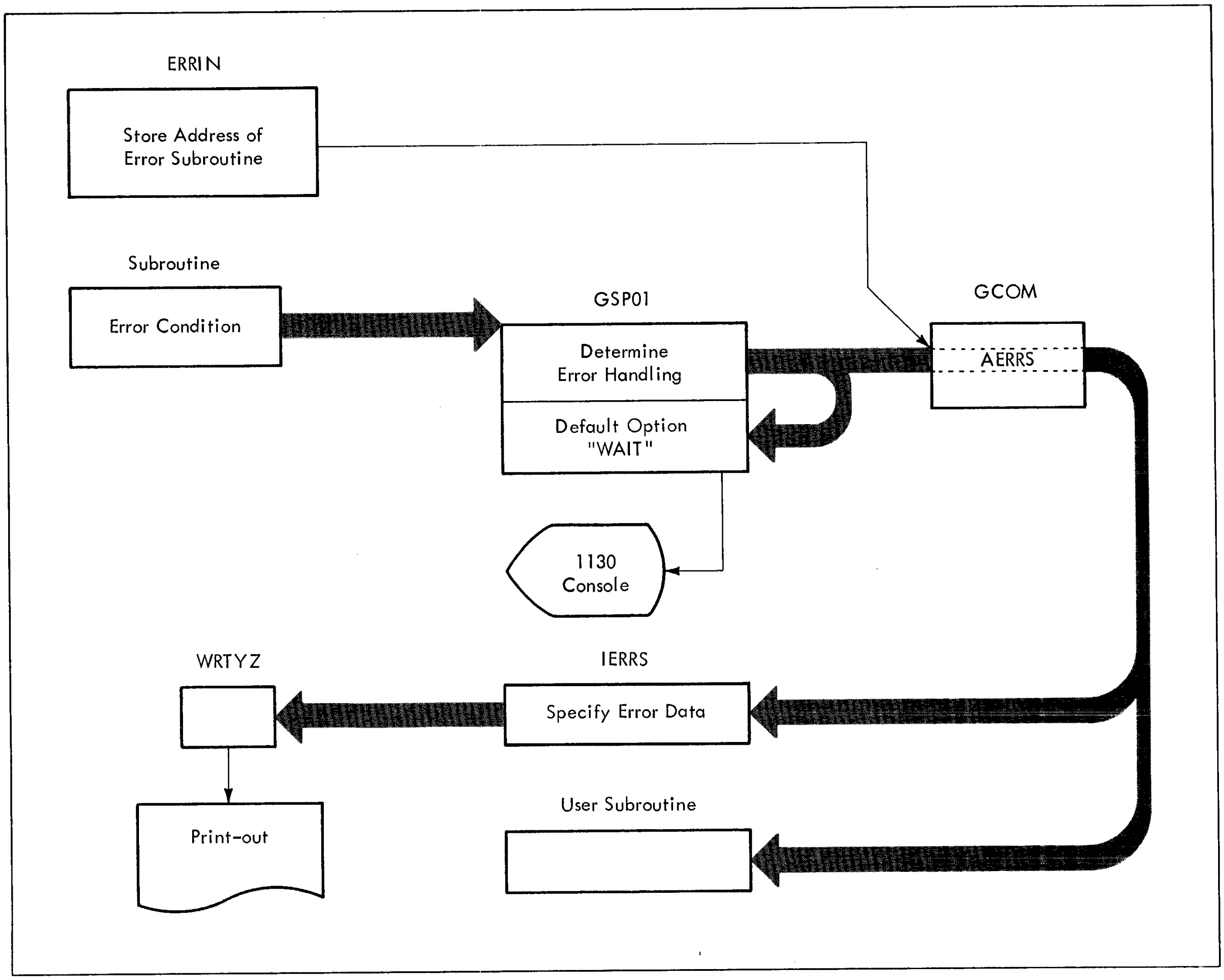


Figure 14. Error Handling

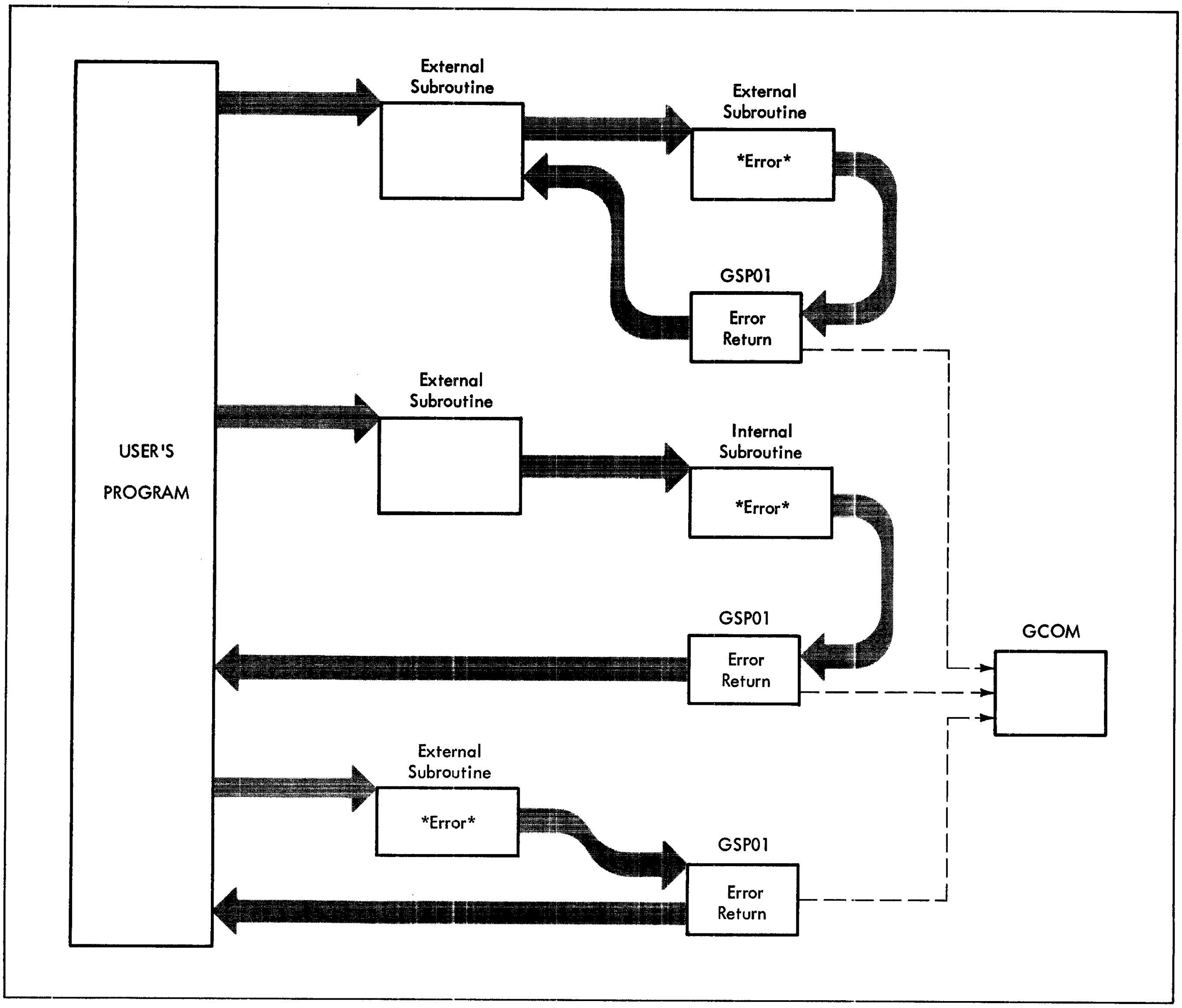


Figure 15. Error Return Linkage

GLLX,GLLY = integer 0

GURX,GURY = integer 1023

XMSCL,YMSCL = real extended precision 1

XASCL,YASCL = real extended precision 0

XSINC,YSINC,XEINC,YEINC = integer 0

XSIND,YSIND,XEIND,YEIND = integer 1

XIPMD,YIPMD = integer 1

OPMD = integer 1

SCIS = integer 2

Refer to Appendix A, Table 8 for a description of the GCA format.

The real value 1 stored in XMSCL and YMSCL is obtained by having FLOAT convert an integer 1. The storing is performed ty GSP10.

Errors: None.

SSCAL -- Set Scaling Information

Chart: AD

Function: To set scaling information in a GCA.

Entry: SSCAL

Exit: Returns through GSP01 to the next sequential instruction in the calling program.

Input:
Argument 1 - address of the GCA.
Argument 2 - address of the array containing the scaling data.

Output: See "Operation."

Operation: The real array whose address is contained in argument 2 is composed of the following elements:

SXL = screen lower-left x-coordinate

SYL = screen lower-left y-coordinate

SXU = screen upper-right x-coordinate

SYU = screen upper-right y-coordinate

GXL = grid lower-left x-coordinate

GYL = grid lower-left y-coordinate

GXU = grid upper-right x-coordinate

GYU = grid upper-right y-coordinate

DXL = data lower-left x-coordinate

DYL = data lower-left y-coordinate

DXU = data upper-right x-coordinate

DYU = data upper-right y-coordinate

SSCAL establishes an indexing value for accessing the array elements according to the program's precision for real values. GSP10 is used to access the array elements and perform calculations in extended precision. The fields in the GCA set by SSCAL and the equations used to calculate the values for each field are as follows:

Field	Description
GLLX	grid lower-left x-coordinate in raster units (integer)
GLLY	grid lower-left y-coordinate in raster units (integer)
GURX	grid upper-right x-coordinate in raster units (integer)
GURY	grid upper-right y-coordinate in raster units (integer)
XMSCL	x-coordinate multiplication scaling factor (real)
XASCL	x-coordinate addition scaling factor (real)
YMSCL	y-coordinate multiplication scaling factor (real)
YASCL	y-coordinate addition scaling factor (real)

Equations

1. $$GLLX = 1023 * \frac{(GXL-SXL)}{(SXU-SXL)}$$

2. $$GLLY = 1023 * \frac{(GYL-SYL)}{(SYU-SYL)}$$

3. $$GURX = 1023 * \frac{(GXU-SXL)}{(SXU-SXL)}$$

4. $$GURY = 1023 * \frac{(GYU-SYL)}{(SYU-SYL)}$$

5. $$XMSCL = 1023 * \frac{(GURX-GLLX)}{(DXU-DXL)}$$

6. $$XASCL = GLLX - \frac{DXL(GURX-GLLX)}{(DXU-DXL)}$$

7. $$YMSCL = 1023 * \frac{(GURY-GLLY)}{(DYU-DYL)}$$

8. $$YASCL = GLLY - \frac{DYL(GURY-GLLY)}{(DYU-DYL)}$$

IFIX is used to convert the results of equations 1 through 4 to integer values. GSP10 is used to store all results in the GCA.

Errors:
Code 1 - An element in the scaling array is invalid because it does not meet the following requirements:

$$SXU \geq GXU > GXL \geq SXL$$

or

$$SYU \geq GYU > GYL \geq SYL$$

SSCIS -- Set Scissoring Option

Chart: AE

Function: To set the scissoring option in a GCA so that scissoring occurs at either the grid or screen boundaries.

Entry: SSCIS

Exit: Returns through GSP01 to the next sequential instruction in the calling program.

Input:
Argument 1 - address of the GCA.
Argument 2 - address of the value indicating the scissoring option.

Output: See "Operation."

Operation: SSCIS places a value (1 or 2) for the option in the SCIS field of the GCA.

Errors:
Code 1 - The value specified for the scissoring option is not 1 or 2.

SINDX -- Set Index Values

Chart: AE

Function: To set values in the GCA that are used to access data from input arrays.

Entry: SINDX

Exit: Returns through GSP01 to the next sequential instruction in the calling program.

Input:
Argument 1 - address of the GCA.
Argument 2 - address of the array containing the index values.

Output: See "Operation."

Operation: The INPRE (integer precision) field in GCOM is examined in order to correctly access the array containing the index values. If the index values are all positive integers, they are stored in the start and end index fields of the GCA.

Errors:
Code 1 - One or more elements in the index array contain zeros or negative values.

SINCR -- Set Increment Values

Chart: AE

Function: To set values in the GCA that are used as increments for generating coordinates for points, lines, or line segments.

Entry: SINCR

Exit: Returns through GSP01 to the next sequential instruction in the calling program.

Input:
Argument 1 - address of the GCA.
Argument 2 - address of the array containing the increment values.

Output: See "Operation."

Operation: The RLPRE (real precision) field in GCOM is examined in order to properly access the array containing the increment values. The increment values are then stored in the start and end increment fields in the GCA.

Errors: None.

SDATM -- Set Input Data Mode

Chart: AE

Function: To set the input data mode in a GCA for accessing and using data from the user's input arrays.

Entry: SDATM

Exit: Returns through GSP01 to the next sequential instruction in the calling program.

Input:
Argument 1 - address of the GCA.
Argument 2 - address of the value for setting the x-coordinate mode.
Argument 3 - address of the value for setting the y-coordinate mode.

Operation: EELMT finds the ECCB for the element currently being processed (CCB in GCOM) and, if it is a controlled, level-controlled, index-controlled, subroutine, or image entity, EELMT loads overhead orders which are stored in the ICA by GSP02. The element is then indicated as ended, and its start address is compared with the address of the element specified to be ended. If the ended element is not the specified element, the next outer element is found by the search function of GSP03 and processed as above. When the specified element is ended, the GSP's mode is determined by examining GENIN in GCOM. If the mode is update and there is unused area remaining in the element, the build function of GSP03 builds an ECCB with a correlation value of /FFFF for this area. GSP02 stores overhead orders in the unused area, and DLCNT is incremented by one. The element pointers (NAGAP, LAGAP, and CCB) are updated, the mode indicator GENIN is set to normal, and the processing message displayed by UELMT is discontinued. If the GSP's mode is normal, the current control block pointer (CCB) in the ICA preface is updated.

For example, given this incomplete image structure:

```
1 2 3
{ ( (
```

The statement CALL EELMT (1) produces:

```
{  ( ( ) ) }
```

Whereas the statement CALL EELMT (3) produces:

```
{ ( ( )
```

See the "Overview" of IBM 1130/2250 Graphic Subroutine Package for Basic FORTRAN IV (C27-6934) for an explanation of the symbols used above.

An element is ended by storing certain overhead orders at the end of the element and updating the element end address (ELEND) and element indicator (INDIC) in the element's ECCB. The selection of overhead orders depends on the element type, as follows:

Element	Orders	Reason
Uncontrolled entity	None	
Controlled entity and Index-controlled entity	SPM	Disable detects.
Level-controlled entity	GB I	Return linkage to right bracket of the verification subroutine in GSP12.
Subroutine entity	GB I	Return linkage to main order program.
Image entity	SGMV	Sets vector mode.
	MBA	Positions beam.
	GB N	Branch to tracking entity.
	GB N	Branch to current rubberband vector in GSP09.
	GB I	Return linkage to system display.

Errors:

Code 2 - The correlation value is not in the range 1 to 32767.

Code 3 - The correlation value does not refer to an element begun by BELMT.

Code 9 - EELMT has been called while the GSP was in external generation mode.

Code 15 - The element is not in the limits specified by ELSTA and ELEND of the current update ECCB.

Code 16 - The element has already been ended.

DELMT -- Delete Element

Chart: BC

Function: To delete one or more defined elements.

Entry: DELMT

Exit: Returns through GSP01 to the next sequential instruction in the calling subroutine or program.

Input: The address of the correlation value of the element to be deleted.

Output: None.

Operation: GSP03 searches for the ECCB of the element to be deleted. INDIC in the ECCB is tested to determine the type of element. If the element is an included element (a subroutine identified by IELMT), the ECCB for this element is placed on the available chain and its correlation value set to zero.

If the element is a subroutine entity, the start address is decremented by two to include the overhead orders preceding the element. Any type of element, other than an included element, must have been ended before it may be deleted.

When an element has been deleted, its correlation value in the ECCB is set to hexadecimal FFFF, and the delete count (DLCNT) in the ICA preface is incremented by one. Any ECCBs which point to elements nested within the deleted area are put on the available chain. The correlation values in them are set to zero to indicate that they may be reused.

Errors:

Code 2 - The correlation value is not in the range 1-32767.
Code 3 - The ECCB is not found by GSP03.
Code 6 - The element is not ended (not applicable to an included element).
Code 9 - The generation mode is external.

UELMT -- Update Element

Charts: BD,BE

Function: To modify an element and its ECCB so that new orders may be stored in the element and any elements nested within it.

Entry: UELMT

Exit: Returns through GSP01 to the next sequential instruction in the calling subroutine or program.

Input:

Argument 1 - address of the correlation value.
Argument 2 - address of the element type.

Output: None.

Operation: UELMT tests the validity of the element type and correlation value, and tests ACICA in GCOM to ensure that there is an active ICA. GSP03 searches for the element specified by argument 1 and further tests are made to ensure that the element is ended and that GSP is not in update or external generation mode.

If there is an active display, it is replaced by a processing message, and DELMT is entered. Upon return, the generation mode is set to update, and GSP03 transfers generation control data from GCOM to the ICA. The pointers in the ICA are modified, and DLCNT (which had been incremented by one in DELMT) is decremented by one. NAGAP and LAGAP in GCOM are set to reflect ELSTA and ELEND in the ECCB. INDIC is changed to show the new element type, and new overhead orders are stored in the element. GNOP orders are stored in the remainder of the element.

If the forward pointer of the ECCB for the specified element, or any ECCB whose element is nested within the specified element, is zero, NCBAU in the ICA preface is set to zero. Otherwise, NCBAU is loaded with the address of the next ECCB after the ECCB for the specified element.

Note: The update function is ended by a call to EELMT in the problem program.

Errors:

Code 1 - The element type is not in the range 1-6.
Code 2 - The correlation value is not in the range 1-32767.
Code 3 - The ECCB is not found by GSP03.
Code 6 - The element found is not ended.
Code 7 - There is no active ICA.
Code 9 - The GSP is already in update or external generation mode.

XELMT -- Extend Element

Chart: BF

Function: To indicate an element that is to be extended.

Entry: XELMT

Exit: Returns through GSP01 to the next sequential instruction in the calling program.

Input:

Argument 1 - address of the correlation value.
Argument 2 - address of the value estimating the number of lines, points, or line segments by which the element is to be extended.

Output: None.

Operation: XELMT tests ACICA in GCOM to ensure that there is an active ICA and tests the validity of the two arguments, storing the estimated number (Argument 2) in ICNTR. GSP03 then searches for the ECCB specified by argument one. Tests are made to ensure that the element type is valid, that the element is ended, and that the generation mode is not update or external. GENIN in GCOM is set to indicate update generation mode, and GSP03 transfers generation control data from GCOM to the ICA preface and initializes GCOM with standard values. NAGAP, LAGAP, and CCB are saved in NAVWD, EAVA, and SCCB, respectively, and the ECCB address is stored in CUCB and CCB. LAGAP and NAGAP are modified to point to the end of the element area referred to by the ECCB and the beginning of the area to which graphic orders will be moved.

Output: See "Operation."

Operation: If the correlation value is zero, an SGMV order is placed, if necessary, in the ICA by the store function of GSP02. The generation function of GSP02 then generates positioning orders for the coordinates specified by arguments 2 and 3. If the correlation value is non-zero, the search function of GSP03 attempts to find the ECCB for the element. If this is a new element, the build function of GSP03 creates an ECCB for an origin entity, and, using the store function of GSP02, places an SGMV order in the ICA. The generation function of GSP02 generates positioning orders using the specified coordinates.

If the search function of GSP03 finds the ECCB for an origin entity and it is not three words in length, or if the element is three words long and the output mode is incremental, UELMT updates the element, and an SGMV order is placed in the ICA by the store function of GSP02. The generation function of GSP02 generates positioning orders for the specified coordinates, and EELMT ends the update function. The ECCB for the element is then modified to indicate that the element is an origin entity (see "Comments" below).

If the element is three words long and the output mode is not incremental, NAGAP and LAGAP are stored, and the generation function of GSP02 updates the origin entity. NAGAP and LAGAP are restored, and GSP03 restores the normal mode by transferring generation control data from the ICA preface to GCOM.

Errors:
Code 2 - The correlation value is not in the range 1 to 32767.
Code 4 - The ECCB found by GSP03 is not for an origin entity.
Code 9 - MVPOS is called to redefine an origin entity while in update mode.

Comments: When UELMT updates an origin entity, it treats the element as an uncontrolled entity. MVPOS modifies the INDIC field of the ECCB to indicate that it is an origin entity.

PLINE/PPNT/PSGMT -- Plot Lines, Plot Points, Plot Line Segments

Chart: CB

Function: To generate orders for plotting lines, points, or line segments.

Entry:
PLINE - for Plot Lines
PPNT - for Plot Points
PSGMT - for Plot Line Segments

Exit: Returns through GSP01 to the next sequential instruction in the calling subroutine.

Input:
For PLINE and PPNT
Argument 1 - address of the GCA.
Argument 2 - address of the x-coordinate array.
Argument 3 - address of the y-coordinate array.
Argument 4 - address of the count.

For PSGMT
Argument 1 - address of the GCA.
Argument 2 - address of the x-starting coordinate array.
Argument 3 - address of the y-starting coordinate array.
Argument 4 - address of the x-end coordinate array.
Argument 5 - address of the y-end coordinate array.
Argument 6 - address of the count.

Output: See "Operation."

Operation: The subroutine determines from the MDLST field in GCOM whether or not a graphic mode order (SGMV or SGMP) is needed. If so, the appropriate order is stored in the ICA by the store function of GSP02, and MDLST is updated to reflect the new graphic mode. Generation of orders is then performed by the generation function of GSP02. The ICNTR field in GCOM is used to indicate to GSP02 the number of lines, points, or line segments to be generated.

Comments: At each entry point an indexing value is loaded into index register 2, and control is passed to common coding. The common coding uses indexing to access those constants that are appropriate for line, point, or line segment plotting. The constants are arranged to facilitate this technique.

Errors:
Code 1 - The count is not in the range 1-32767.

PTEXT -- Plot Text

Chart: CC

Function: To generate graphic data to produce characters and symbols.

Entry: PTEXT

Exit: Returns through GSP01 to the next sequential instruction in the calling program.

for the boundary point (PT4 or PT6) are stored in NGXI and NGYI.

For the second type of vector (outside to inside), positioning may be required. The coordinates for the boundary point (PT4 or PT6) are stored in NPXI and NPYI, and the current computed coordinates (PT2) are stored in NGXI and NGYI.

For the last type of vector (outside through to outside), a computation is made to determine the relative proximity of the two boundary points to the last computed coordinate. The coordinates of the nearest boundary point (PT4 or PT6) are stored in NPXI and NPYI, and the coordinates of the other boundary point (PT3 or PT5) are stored in NGXI and NGYI.

Errors: None.

2250 I/O SUBROUTINES

The 2250 I/O subroutines are those which issue I/O commands to the 2250 and are called by the FORTRAN program to do I/O operations. The following is a list of I/O subroutines and their functions:

- EXEC, which identifies an image entity and attaches it to the system display.
- TMDSP, which disconnects the image entity from the system display.
- SPFKL, which identifies the programmed function keyboard lights which are to be lighted.
- DSPYN, which issues I/O commands to the 2250 and handles interrupts from the 2250.

EXEC -- Execute Display

Chart: DA

Function: To start regeneration of an image entity on the 2250 display unit.

Entry: EXEC

Exit: Returns through GSP01 to the next sequential instruction in the calling program.

Input:
Argument 1 - address of the correlation value of the ICA.
Argument 2 - address of the correlation value of an image entity; or (Assembler language only) the address of the first order of an order program.
Argument 3 - address of the option code defining the type of Argument 2 and the address at which regeneration is to start as follows:

0=correlation value, and the use of the switch trap in system display is desired.
1=correlation value, and no switch trap is desired.
2=arguments 1 and 2 are disregarded; start at the next order address.
3=address of an order program, and the switch trap is desired.
4=address of an order program, and the switch trap is not desired.

Output: See "Operation."

Operation: DSPYN resets the 2250, and EXEC examines the option code (argument 3) to determine the type of data passed in argument 2. If a correlation value is specified, the switch trap is set dependent upon the option specified, and the address of SD1 or STMR1 is stored in ASD in GCOM. If the ICA specified in argument 1 is the active ICA, GSP03 finds the specified image entity. If it is not the active ICA, EXEC finds the specified ICA and performs its own search to find the specified image entity.

The address of the found image entity is compared with the IMECB field in GCOM. If they are the same, DSPYN starts regeneration of the system display and return is made to the next sequential instruction in the calling program. If they are not the same, DSPIN in GCOM is set to indicate execute status, the address of the image entity is stored in the graphic branch (SD5AD) in the system display, and the branch is made operational. DSPYN starts regeneration of the system display and control is returned to the calling program.

If the Assembler-language programmer specifies an address of an order program in argument 2, the switch trap and ASD are set dependent upon the option code (argument 3). The address of the order program is stored in the graphic branch (SD5AD) in the system display, and the branch is made operational. DSPIN is set to indicate execute status and DSPYN starts regeneration of the system display.

If argument 3 specifies that the display is to restart at the next order address, the address is obtained from the field NOADR in GCOM and stored in a graphic branch (BRNCH) in an order program in EXEC.

The graphic order program has the following format:

```
OPT2      STMR

GRAMD     SGMP

MBA       DC  0

          DC  0

          SPM /A

MODCH     GNOP

BRNCH     GB  *-*
```

The x- and y-coordinates (ATNX and ATNY) for the move beam absolute order are found in GCOM and stored in location MBA. The order stream prior to the next order address is searched for a character or graphic mode order. A found character mode order is stored in MODCH, and a found graphic mode order is stored in GRAMD. The address of this order program is stored in the parameter list of the LIBF to DSPYN. This order program ensures that the display resulting from the order series starting at the next order address will have the attributes of the image entity within which it is nested. DSPYN starts regeneration of the order program and return is made to the calling program.

Note: The Graphic Branch Indirect order at the end of an image entity is not modified.

Errors:
Code 1 - Argument 3 (option code) is not in the range 0-4.
Code 2 - The correlation value for the ICA or the image entity is not in the range 1-32767.
Code 3 - The ECCB is not found by GSP03, or by EXEC.
Code 4 - The ECCB found is not for an image entity.
Code 5 - The image entity is not ended.
Code 9 - The GSP is in update mode.

TMDSP -- Terminate Display

Chart: DB

Function: To stop the regeneration of an image entity.

Entry: TMDSP

Exit: Returns through GSP01 to the next sequential instruction in the calling program.

Input: None.

Output: See "Operation."

Operation: TMDSP sets the following graphic branch orders in the system display to perform no-operation:

- Branch to the processing message subroutine (SD2)
- Branch to the scan subroutine (SD3)
- Branch to the tracking symbol subroutine (SD4)
- Branch to the image entity (SD5)

DSPYN is then used to restart the system display. The DSPIN field in GCOM is set to indicate that the display has been terminated. The ICACB field in GCOM is set to zero.

Errors:
Code 9 - The display has already been terminated.

SPFKL -- Set Programmed Function Keyboard Lights

Chart: DC

Function: To light the specified programmed function lights and (if requested) selectively enable those keys which are lighted.

Entry: SPFKL

Exit: Returns through GSP01 to the next sequential instruction in the calling program.

Input:
Argument 1 - address of the selective enable code.
Argument 2 - address of the array indicating the lights to be lit.
Argument 3 - address of the count of the array elements.

Output: None.

Operation: The arguments are tested to ensure their validity. The array indicating the lights to be turned on is accessed to set the bit pattern for the two word parameter passed to DSPYN. DSPYN is then called to set the programmed function keyboard lights specified. The selective enable code is examined, and, if it is zero, the SATEN field specifies the enable status of the programmed function keyboard. If enabled, a hexadecimal FFFF is stored in PFKL1 and PFKL2 in GCOM. If disabled, zeros are stored in PFKL1 and PFKL2. If the enable code is one, and the programmed function keyboard is enabled, the parameter designating the enabled lights is stored in PFKL1 and PFKL2. For all conditions DSPYN is called to start regeneration.

Errors:
Code 1 - The selective enable option is not 0 or 1; or
the count of the array elements is not in the range -1 to +32; or
the value indicating the lights to be set is not in the range 0 to 31; or
the count of array elements is 32, but the elements contain values other than 0 or 1.
Code 12 - The selective enable option is 1, but the programmed function keyboard is not enabled.

DSPYN -- 2250 I/O Subroutine

Charts: DD,DE,DF,DG,DH,DJ

Function: To provide interrupt services (i.e., the 2250 ISS) for the 2250 Display Unit, Model 4; to provide I/O functions for the Assembler-language programmer and the GSP; or to store the addresses of several direct entry subroutines in GCOM.

Entry:
1. DSPYN for I/O operations.
2. ISSDP for attention processing operations.

Exit: Returns to the next sequential instruction in the calling program through the address stored in the entry word.

Input:
1. Address of the I/O function code
2. Address of the I/O area or address of direct entry subroutine
3. Address of error routine

Output:
1. Specified I/O function
2. Information stored in the attention data array

Operation: DSPYN operation is divided into the following areas:

1. Input/Output operations
   a. Start Regeneration
   b. Set Programmed Function Lights
   c. Read Status
   d. No-operation
   e. Reset Display
   f. Sense Device Status Word
2. Store direct entry subroutine address
3. 2250 attention-handling operations
   a. CANCEL key attention
   b. END key attention
   c. Alphameric keyboard attention
   d. Programmed function keyboard attention
   e. Order-controlled attention
   f. Light pen attention
4. Error handling

INPUT/OUTPUT OPERATIONS: The first function of DSPYN is to give input/output capabilities to the Assembler-language programmer. Upon entry from the Assembler-language program, DSPYN examines the function argument to determine the proper I/O function to be executed. Table 1 contains the I/O operations, function codes, and their associated entry points (the last two characters of the entry point are the hexadecimal value for the function code).

Start Regeneration: This section of DSPYN stores the address of the first order of the display program in the I/O control command used to start execution of the display program. A branch is made to DPN70 (Sense Device) to obtain the Device Status Word (DSW). If the DSW indicates that a display is not currently regenerating, the first order in the display program is examined to determine if it is an unconditional Graphic Interrupt (GI). A GI order causes a branch to the DSPYN error routine; otherwise a branch is made to DPN50 to issue the XIO-Start Regeneration. If the 2250 is busy, a branch is made to DPN48 to stop regeneration of the image. Once regeneration has ceased, the first order is inspected to see if it is a GI and normal execution continues. Two attempts are made to start regeneration of the new display. After the second attempt, if the DSW does not indicate that the 2250 is busy, control is passed to the DSPYN error routine.

Table 1. I/O Operations, Function Code, and Entry Points

Operation	Entry Point and Function Code
Start Regeneration	DPN50
Set Programmed Function Lights	DPN58
Read Status	DPN60
No-operation	DPN40
Reset Display	DPN48
Sense Device	DPN70

Set Programmed Function Lights: This section of DSPYN puts the address of the bit pattern to be displayed on the PFKB in the two indicator words found at the address in the IOCC and issues the XIO-Set Programmed Function Lights.

Read Status: This section of DSPYN moves six words containing the status of the 2250 at the time of the attention from ASM60 to the area specified by the user. ASM60 is an area within DSPYN which, at the time of an attention, receives 6 words of data from GCOM indicating the status of the 2250. When the data has been transferred to the user's area, the first word of ASM60 is made negative to indicate that at the time of the next attention, ASM60 is free to receive new status information.

No Operation: This section of DSPYN returns control to the calling program without performing any operations.

Reset Display: This section of DSPYN issues the XIO command which stops regeneration of the display and turns off the PFKB lights. When control returns, a branch is made to access the DSW. The DSW is tested to determine if the 2250 is still busy. If the 2250 is busy, a second attempt is made to stop regeneration by issuing the command again. A second unsuccessful attempt causes a branch to the error routine specified by the user. If no error routine is specified, or if the user's error routine returns control with the error code still in the accumulator, control passes to the standard error routine. (See "Error Handling"). If the 2250 is not busy, control returns to the calling program.

Sense Device: Upon entering the Sense Device portion of DSPYN, the accumulator is set to zero. An XIO -- Sense Device control command is issued to place the DSW in the accumulator. When control returns, a test is made to determine if the request has been successful. If the accumulator is not zero, a return is made to the calling program. If the accumulator is zero, a second attempt is made to get the DSW. Failure to complete the operation on the second attempt causes control to pass to a user-defined or the GSP error routine.

STORE DIRECT ENTRY SUBROUTINE ADDRESS: When DSPYN is entered, the function code is examined. A code of /194n (n=1-7) causes a branch to DPN4X. The function code is again examined resulting in the direct entry subroutine address being stored in the appropriate field in GCOM. An address of zero stored in GCOM indicates, at the time of an attention, that no direct entry routine is specified for that type of attention. If the function code is /1947, the entire direct entry subroutine branch table in GCOM is set to zero.

2250 ATTENTION HANDLING SUBROUTINE: When an attention occurs, the 2250 attention processing subroutine is entered via ILSX3. Upon entry, an XIO-Read Status is executed and the result placed in GCOM, the GSP communications control block. The XIO-Read Status puts 6 words of data describing the 2250 at the time of the interrupt into GCOM. (If GSP is not being used, the 6 words of status information are placed in an alternate area (ASM60) within DSPYN.) A series of tests is made to insure that the Read Status produced valid information. Invalid status information causes a branch to the error routine. Valid 2250 status information is then placed in ASM60, an area within DSPYN from which it may be accessed by the user. Before the status information is moved to ASM60, a test is made to determine that the status information associated with the last processed attention has been transferred to the user's area. The new data is transferred only if the previous data has been transferred. The specific types of attentions are then processed as follows:

CANCEL key attention: An attention produced by the CANCEL key causes an indicator (CNIND) within DSPYN to be set to 1. A special routine is then entered which saves the ILSX3 return address. The ILSX3 return address is modified to the address of a wait routine and control passes to the wait routine. An XIO is issued to start regeneration at an address which displays the CANCEL frame. The CANCEL frame display contains 3 options, one of which the operator may choose to determine the course his program will take. The format of the CANCEL frame is shown in Figure 17.

Each selection is composed of a set of orders which include one Graphic Interrupt Conditional order. When a detect is made on one of these selections and the light pen switch is closed, the order controlled attention causes control to pass to the appropriate processing routine with in DSPYN. As each attention occurs, the CNIND indicator is examined to determine whether the CANCEL frame is being displayed. If it is being displayed, control returns through the ILS to the wait routine. Order controlled attentions are examined to determine which selection in the CANCEL frame the operator detected on. An attention on * TERMINATE causes the program to terminate (via a CALL EXIT) without a storage dump. An attention on * DUMP results in program termination with a dump of main storage. An attention on * RESUME restores the data stored from the original CANCEL attention, restarts the display, and returns control to be interrupted program.

- ICURS, which establishes message collection mode for entering data from the alphameric keyboard.

- RCURS, which ends message collection mode.

- BCNV, which converts real or integer values to EBCDIC, or vice versa.

GCHAR -- Character Strokes

Chart: None.

Function: To create strokes required by GSP subroutines PTEXT, DFMSG, GSP07, and GSP02 (text function).

Entry: GCHAR

Exit: Not applicable.

Input: Not applicable.

Output: Not applicable.

Operation: This subroutine is selected for loading by leaving column 13 on the *G2250 supervisor record card blank. The address of GCHAR is indicated to other GSP subroutines by GSPIN.

The entry point GCHAR is the base for relatively addressing a 258 word table of Graphic Short Branch (GSB) orders to character stroke subroutines. This table corresponds to 256 EBCDIC values with decimal values 0-255 and hexadecimal values 00-FF. Providing character strokes for 92 standard characters (upper- and lower-case letters, digits 0-9, and special characters), GCHAR is indexed by the EBCDIC value to reference the appropriate GSB (i.e., GCHAR+226 references upper case S). GCHAR+256 is a GSB to a special null for the cursor. GCHAR+257 is a GSB to the cursor.

For the remaining 164 values for which no standard character is assigned, the table references a "not available" character appearing as a superscript N and a subscript A in the same grid. GCHAR may be updated to include non-standard characters by replacing the GSB of a value to the "not available" character with a GSB to the new character stroke subroutine.

Since the GSP provides for translating light pen attentions on characters and translating GSB orders in message entities to the correct EBCDIC values, the format of all character stroke subroutines in GCHAR is fixed. The following example of uppercase A illustrates this format:

```
     DC         .A     EBCDIC value

     CS         4,0    Superscript Entry

     CS         1,0    Subscript Entry

A    DBS        3,7    Normal (aligned) Entry

     DBS        6,0

     MBS        5,2

     DBS    R   1,2
```

The GSB in the table at displacement 193 is a GSB to label A. GSP implementation will subtract 1 from the address in the GSB order to produce a subscripted character and 2 to produce a superscripted character.

For the translation function, the GSP will test the order at the modified address to see if it is a control stroke order (CS). If so, it will further test to see if it is a subscript control stroke order. Thus, depending upon the stroke order specified by the GSB address, the GSP will subtract 1, 2, or 3 to obtain the EBCDIC value for the stroke subroutine.

Note: Strokes which would not normally use subscripting or superscripting (space, null, etc.) will have this same format for consistency in implementation. Therefore, if any new characters are added to GCHAR, they must also follow this format.

GUPER -- Upper Case Character Strokes

Chart: None.

Function: This subroutine provides the branching table capability of GCHAR except that when the table is indexed to a lower case letter, the GSB addresses the upper case of that letter. Lower case letters are not available.
GUPER is selected for loading by punching a "U" in column 13 of the *G2250 card.

Entry: GUPER

Note: For further information, see the similar subroutine, GCHAR.

DFMSG -- Define Message Entity

Chart: GA

Function: To create or reinitialize a message entity.

Entry: DFMSG

Exit: Returns through GSP01 to the next sequential instruction in the calling program.

Input:

- Argument 1 - address of the correlaticn value.
- Argument 2 - address of the size, in characters, of the element.
- Argument 3 - address of the character size option (basic cr large).
- Argument 4 - address of the initialization character option (blank or null).

Output: A new or modified message entity and ccrresponding ECCB.

Operation: DFMSG tests the validity of the four arguments, and GSP03 searches for the ECCB specified by argument 1 (correlaticn value). If GSP03 returns a zero, indicating that the ECCB is not found, arguments 2, 3, and 4 are retested to ensure non-zero values, and GSP03 builds an ECCB for a message entity. GSP02 stores in the message entity a Set Character Mode order (SCMB or SCML) and a Graphic Short Branch (GSB) to the special cursor null. The text function of GSP02 initializes the message entity with GSBs to the null or blank character as specified by argument 4. GSP02 stores a GNOP at the end of the element to terminate character mode, and ELEND in the message entity's ECCB is set to reflect the end address of the element. The exit phase is then entered.

The initialization character option value determines whether the message entity is to remain unchanged (value=0), if it is to be filled with GSBs to the null character (value=1), or if it is to be filled with GSBs to the blank character (value=2). If a message entity is found by GSP03, a new Set Character Mode order is stored, if necessary, and the length of the element is compared with the length specified by argument 2.

If the initialization value is not zero, an update phase is entered regardless cf the relative length of the element to that specified. In the update phase, DSPYN removes the address of GSP07 from GCOM, through which GSP07 is entered, thus preventing any possible movement of the cursor. The ECCB is set to indicate an uncontrolled entity, and UELMT resets the pointers in the element's ECCB and in GCOM, and adjusts the length of the element if necessary. DFMSG then proceeds as though an ECCB for a new message entity had been built.

If the initialization value is zero and the element length is the same as that specified, no further processing occurs. If the initialization value is zero and the element is longer than that specified, a test is made to determine if the cursor or cursor null is within the limits of the new message entity. If it is not, it is placed in the last position in the new element and CURAD in GCOM is updated. The update phase is entered and, upon return, a GNOP is stored in the end of the element to terminate character mode. The exit phase is then entered.

If the initialization value is zero and the present length of the element is shorter than that specified, ELSTA in the ECCB is set to reflect the address of the GNOP order at the end of the element and ICNTR is set to indicate the additional space needed. The update phase is entered and, upon return, the extension is filled with GSBs to the blank character by the text function of GSP02. A GNOP is stored at the end of the element, ELEND is set, and the exit phase is entered.

The exit phase checks to determine if the update phase has been entered. If not, no further processing occurs. If the update function has been entered, EELMT ends the update phase, and DSPYN is used to restore the address of GSP07 to ANDE in GCOM.

Errors:

- Code 1 - The size of the element is not in the range 0-32767; or the character size option is not 0, 1, or 2; or the initialization character option is not 0, 1, or 2.
- Code 2 - The correlation value is not in the range 1-32767.
- Code 3 - Argument 2, 3, or 4 is 0 and GSP03 does not find an ECCB for the correlation value specified.
- Code 4 - The ECCB fcund by GSP03 is not for a message entity.
- Code 8 - GCHAR or GUPER is not loaded.
- Code 9 - DFMSG is called to redefine an origin entity while in update mode.

MSGIN -- Message Entity Initialization

Chart: GB

Function: To store graphic short branch orders to character stroke subroutines in a message entity.

Entry: MSGIN

Exit: Returns through GSP01 to the next sequential instruction in the calling program.

Input:

- Argument 1 - address of the correlation value.
- Argument 2 - address of the data array.
- Argument 3 - address of the position within the element where the first GSB order is to be stored.

Function: To store in GCOM the address cf the error handling subroutine that will process non-I/O error conditions.

Entry: ERRIN

Exit: Returns through GSP01 to the next sequential instruction in the calling program.

Input: Address of the error handling subroutine.

Output: See "Operation."

Operation: ERRIN stores the address of the error handling subroutine in the ERADR field in GCOM.

Errors: None.

IERRS -- Identify Errors

Chart: JA

Function: To produce one line of print on the console typewriter, consisting of a subroutine ID, error return code, cumulative error variable, and a count indicating the number of times external subroutines have been entered. The maximum count is +32767. If subroutines have been entered more than 32767 times, the count remains at 32767.

Entry: IERRS

Exit: Returns through the address stored in the entry word to the next sequential instruction in GSP01.

Input: Error variables defined by GSPIN and fields in GCOM.

Output: One line of print in the following format:

"Routine ID=a, Error Code=b, Cumulative=c, Count=d."

where
a=two hexadecimal digits denoting the ID of the last GSP subroutine entered via a user's call statement.
b=an error code in the range 0-16 set by the GSP.
c=sixteen digits corresponding to the error indicators set in the cumulative error variable.
d=five decimal digits, the count in GCOM.

Operation: The subroutine ID, cumulative error variable, error return variable, and count are converted to the EBCDIC character equivalents and stored in their respective locations in the message to be printed. IERRS passes control to WRTYZ (the console print routine) and the message is printed.

Errors: None.

INTERNAL SERVICE SUBROUTINES

The following subroutines provide common functions for several of the GSP subroutines:

- GSP01, which saves the status of index registers upon entering a subroutine, and establishes linkage for exit from the subroutine for normal and error conditions.
- GSP03, which builds ECCBs for specified elements, finds ECCBs for specified elements, transfers generation control data between the ICA preface and GCOM, or initializes the generation control fields in GCOM.
- GSP10, which provides floating point arithmetic functions.

GSP01 -- GSP Inner Subroutine 1

Chart: KA

Function: To provide one or more of the following services required by most of the GSP subroutines called by the mainline program:

- Save function -- Saves registers and performs other housekeeping.
- Error function -- Sets error variables in GCOM.
- Return function -- Restores registers and provides return linkage to the mainline program.

Entry: GSP01

Exit: For the save function, returns to the next sequential instruction in the calling subroutine. For the error and return functions, returns to the next sequential instruction in the mainline program.

Input:
1. An indicator word in the calling sequence.
2. For the error function: an error code in the accumulator.

Output: See "Operation."

Operation: GSP01 examines an indicator word in the calling sequence to determine what service is requested. The calling sequence and an explanation of the indicator word are as follows:

```
LIBF  GSP01

DC    /n1 n2 n3 n4
```

n1 indicates the service to be provided by GSP01 as follows:

0 = save function

1 = error function

2 = return function

n2 and n3 are reserved.

n4 For save function indicates the number of arguments in the mainline calling sequence.

For error function indicates source (1=GSPIN; 0=others).

Save function -- Index registers are stored in fields REG1, REG2, and REG3 in GCOM. The address of the next sequential instruction in the mainline program is computed and stored in field MLRET in GCOM. This computation is done by adding to the first word of the external subroutine (the contents of the IAR are stored there by the CALL statement) the lower three digits (n2, n3, n4) of the indicator word which indicate the number of one-word parameters in the calling sequence. The result is the next sequential instruction in the calling program. GSP01 then stores the arguments in the mainline calling sequence in GCOM, beginning at field ARG1. The number of arguments stored is determined by n4 of the indicator word. The return error variable field in GCOM (RETEV) is set to zero, the address of GCOM is loaded into register 1, the address of the active ICA is loaded into register 2 (if there is an active ICA), and control is returned to the calling subroutine.

Error function -- GSP01 stores the error code passed in the accumulator in the field indicated by RETEV in GCOM and ORs the code into the cumulative error variable field indicated by CUMEV in GCOM. If a direct entry subroutine for error processing is specified (see ERRIN) control passes to that subroutine. Upon return from the error processing subroutine, GSP01 then returns to the mainline program by means of the return function. If a direct entry subroutine is not specified, a WAIT instruction causes a hexadecimal FFFF to be displayed in the accumulator lights, and the contents of the field indicated by RETEV and the ID of the last external subroutine to be displayed in the extension lights. Depression of the program start key causes another WAIT statement to be encountered with the contents of the field indicated by CUMEV displayed in the accumulator lights and the count of external GSP subroutines entered (GSPEC) displayed in the extension lights. Depressing the Program Start key causes the return function of GSP01 to be entered.

Return function -- GSP01 restores index register 1, 2, and 3, and returns to the mainline program by means of the address in the MLRET field in GCOM. When the return function is entered from GSPTM, $GCOM is set to zero before returning to the mainline program.

Errors: Code 11 -- GSPIN has not been called.

GSP03 -- GSP Inner Subroutine 3

Charts: KB,KC

Function: To perform one of the following functions required by other GSP subroutines.

- Build an ECCB in the active ICA.
- Search for an ECCB in the active ICA.
- Transfer control data needed for coordinate generation between GCOM and the active ICA's preface.
- Initialize fields in GCOM with control data needed for coordinate generation.

Entry: GSP03

Exit:
Normal - returns to the next sequential instruction in the calling subroutine.
Error - returns through GSP01 to the mainline program.

Input:
1. A 1-word indicator in the calling sequence.
2. An address or correlation value in the accumulator.

Output:
1. For the build function, a 4-word ECCB in the active ICA and its address in the accumulator.
2. For the search function, an ECCB address in the accumulator.
3. For the transfer function, data transferred between GCOM and the ICA preface.
4. For the initialization function, generation control data in GCOM for coordinate generation initialized with standard values.

Operation: GSP03 first saves the index registers and the accumulator. An indicator word is then analyzed to determine what service is required. The calling sequence and an explanation of the indicator word are as follows:

```
  LIBF  GSP03

  DC    /n1 n2 n3 n4
```

n1 indicates the service to be provided by GSP03 as follows:

0 = build an ECCB as further defined by n4

1 = search for an ECCB as further defined by n3 and n4

2 = transfer control data between GCOM and the active ICA as further defined by n4

3 = initialize fields in GCOM with standard values for coordinate generation

n2 is reserved

n3 further defines the search function by indicating the type of argument being passed in the accumulator, as follows:

0 = correlation value

1 = address of the ECCB

2 = an address within an element

n4(A) further defines the search function by indicating the type of search requested, as follows:

0 = find the address of the ECCB

1 = find the address of the preceding ECCB on the chain

2 = find the address of the ECCB for the next outer element

or (B) further defines the build function by indicating the type of element for which an ECCB is to be built, as shown below. The accumulator contains the correlation value to be associated with the ECCB. (A value of /FFFF in the accumulator indicates that the ECCB is for an area in the ICA that was left after an element was updated.)

0 = image entity
1 = controlled entity
2 = uncontrolled entity
3 = subroutine entity
4 = origin entity
5 = message entity
6 = linkage entity
7 = tracking entity
8 = level-controlled entity
9 = indexed entity
A = reserved
B = subroutine entity requested by IELMT (Assembler language)
C through F = reserved

or (C) further defines the transfer function by indicating the direction of data transfer, as follows (the accumulator contains the ICA address):

0 = transfer generation control data from GCOM to the ICA preface
1 = transfer generation control data from the ICA preface to GCOM

The services described above are tested for in the order of presentation below. Requests for a service other than initialization require that an ICA be active.

Initialization function -- Fields IGXI through LCYI in GCOM are set to 512. Fields MDLST and OPLST in GCOM are set to zero.

Build function -- GSP03 performs a series of tests to determine whether or not the rules for nesting have been followed. If so, the ECCB is built using an available ECCB on the chain, if there is one, or by using the last four contiguous words in the ICA generation area. Should no words be available in the ICA, GSP04 (Push-Down) is called to make area available if there are elements present that are indicated as deleted. If area is made available by GSP04, the ECCB is built as above. IAGAP, ECB, FAVCB, and CCB are updated, and the address of the new ECCB is put in the accumulator.

Search function -- Given a correlation value, the address of the ECCB is found by searching once from the beginning of the chain of ECCBs to its end or until the correlation value is found. Given an address within an element, however, requires that the chain always be completely searched, since the address may be within a nest of elements. The address of the ECCB for the innermost element whose boundaries include the specified address is the one returned by GSP03. Requesting the address of an ECCB when the argument passed in the accumulator is also the address of an ECCB (n3=1 and n4=0) is an error condition.

If the address of the preceding ECCB on the chain was requested, the chain is searched again until the ECCB whose forward pointer (word 3) points to the ECCB specified by the argument in the accumulator. If the address of the ECCB for the next outer element was requested, the end and start address fields of preceding ECCBs are compared with those of the ECCB specified by the argument in the accumulator. This search ends when the ECCB is found or the beginning of the chain is reached.

For each type of search (address of the ECCB, of the preceding ECCB, or of the ECCB for the next outer element), a find results in the ECCB's address being returned in the accumulator; an unsuccessful search results in zero being returned in the accumulator.

Transfer function -- Fields LGXI through OPLST in GCOM (words 32 through 43) are moved to fields GENX through LSTOP in the ICA preface (words 10 through 19) or vice versa, depending on the value of n4 in the indicator word passed to GSP03.

Comments: ECCBs are built in descending core locations from the highest position within the ICA. There may be two chains of ECCBs in the ICA. The first is a chain of "active" ECCBs for elements that are or will be used in graphic displays. The addresses of the first and last ECCBs on the active chain are found in the SCB and ECB fields respectively in the ICA preface. An existing ECCB on the active chain may be modified to reflect an increase or decrease in the length of the element, or to indicate that the element is "deleted" (no longer desired for display). ECCBs for deleted elements remain on the active chain with a correlation value of hexadecimal FFFF.

The second chain consists of "available" ECCBs for elements whose area is overlayed by other elements. The address of the first available ECCB is found in the FAVCB field in the ICA preface. The only ECCBs placed on the available chain are for those elements that were nested within elements indicated as deleted, and ECCBs for elements that were indicated as deleted and have been overlayed by GSP04 (Push-down) or GSP05 (Push-up). ECCBs on the available chain may be redefined and reused for any type of element. Chaining is accomplished by storing an index value in the FORPT field in the preceding ECCB on the chain. This index is used in an algorithm to determine the absolute address of the ECCB in core as follows:

$$\frac{SCB-ECCB+4}{4}$$

Errors:

Code 4 - There is invalid nesting.
Code 9 - There is no active ICA.
Code 5 - There are no words available in the ICA.

GSP10 -- Floating Point Functions

Chart: None.

Function: To provide several floating point functions required by the GSP. (Equivalent functions available in the 1130 subroutine library are not usable by the GSP.) These functions and the identifying codes used in the calling sequence are:

Function	Code
Extended Floating Point Load (ELD)	0
Extended Floating Point Store (ESTO)	1
Extended Floating Point Add (EADD)	2
Extended Floating Point Subtract (ESUB)	3
Extended Floating Point Multiply (EMPY)	4
Extended Floating Point Divide (EDIV)	5
Floating Point Load (FLD)	6
FLOATING Point Store (FSTO)	7

Entry: GSP10

Exit: Returns to the next sequential instruction in the calling subroutine via the address stored in the entry word + 2.

Input: An argument address, a function code (see above), and in some instances a value in the floating accumulator. The calling sequence is as follows:

```
LIBF  GSP10
DC    address of the argument
DC    address of the function code
```

Output: A value placed in the argument address or in the floating accumulator, depending on the function.

Operation: GSP10 examines the function code and executes a branch to perform the requested function. The eight functions are duplicates of the corresponding floating point functions described in IBM 1130 Subroutine Library.

Errors: None.

• Chart AB. ICAIN - ICA Initialization

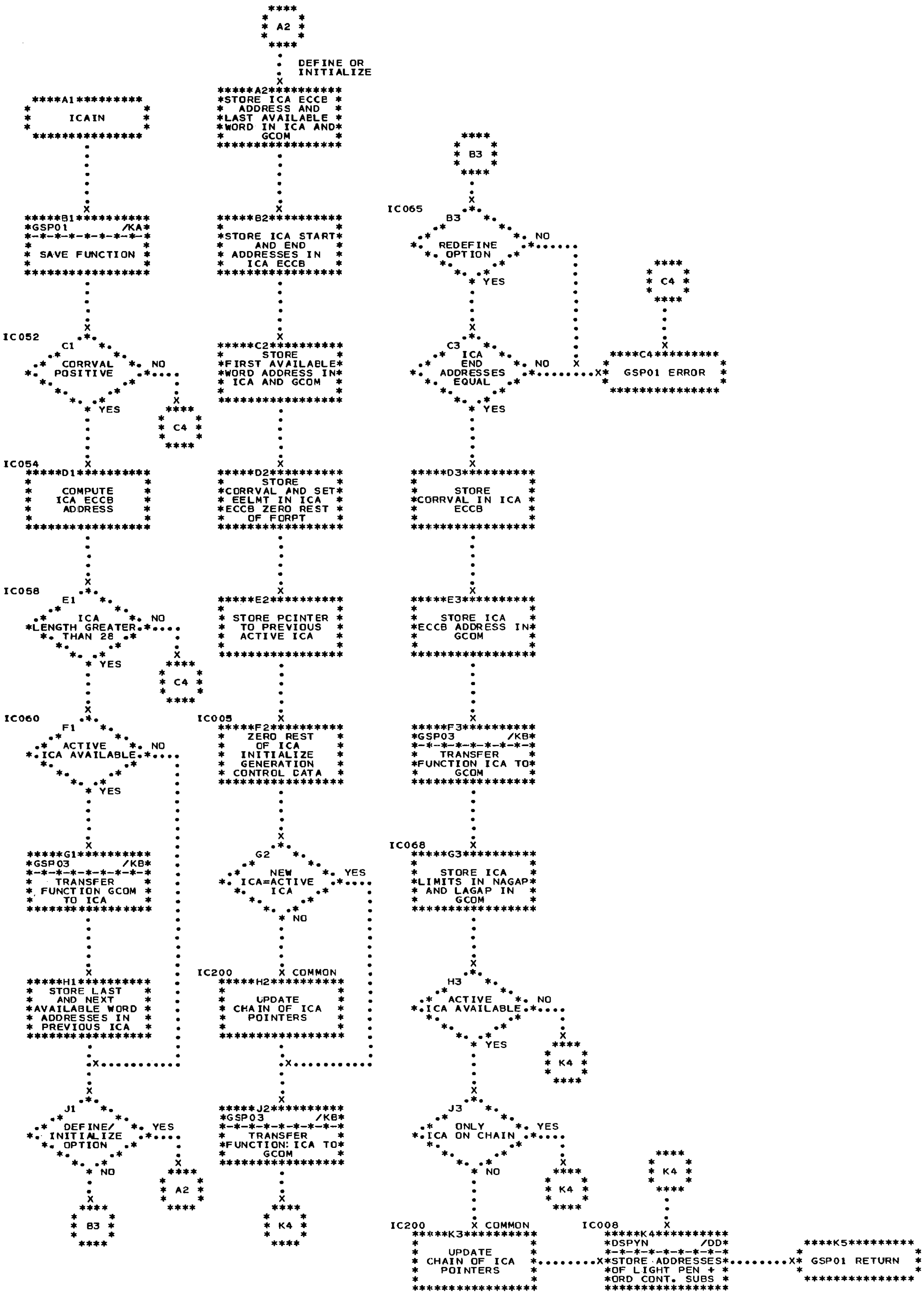
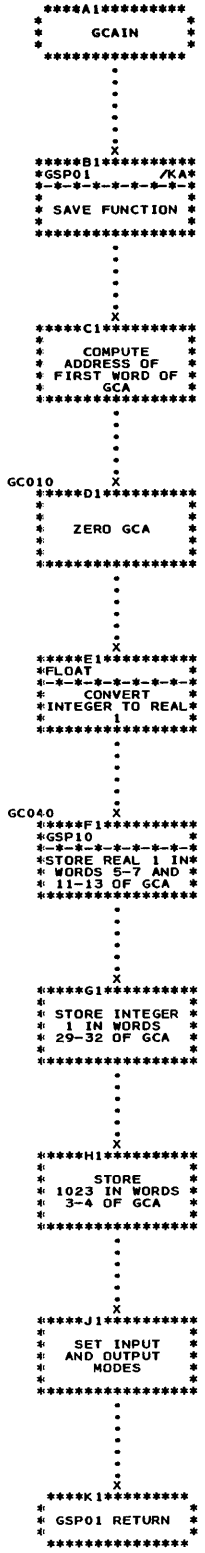

A2
DEFINE OR INITIALIZE
A2 STORE ICA ECCB ADDRESS AND LAST AVAILABLE WORD IN ICA AND GCOM
A1 ICAIN
B1 GSP01 /KA SAVE FUNCTION
B2 STORE ICA START AND END ADDRESSES IN ICA ECCB
B3
IC065 B3 REDEFINE OPTION NO YES
C4
IC052 C1 CORRVAL POSITIVE NO YES
C4
C2 STORE FIRST AVAILABLE WORD ADDRESS IN ICA AND GCOM
C3 ICA END ADDRESSES EQUAL NO YES
C4 GSP01 ERROR
IC054 D1 COMPUTE ICA ECCB ADDRESS
D2 STORE CORRVAL AND SET EELMT IN ICA ECCB ZERO REST OF FORPT
D3 STORE CORRVAL IN ICA ECCB
IC058 E1 ICA LENGTH GREATER THAN 28 NO YES
C4
E2 STORE POINTER TO PREVIOUS ACTIVE ICA
E3 STORE ICA ECCB ADDRESS IN GCOM
IC060 F1 ACTIVE ICA AVAILABLE NO YES
IC005 F2 ZERO REST OF ICA INITIALIZE GENERATION CONTROL DATA
F3 GSP03 /KB TRANSFER FUNCTION ICA TO GCOM
G1 GSP03 /KB TRANSFER FUNCTION GCOM TO ICA
G2 NEW ICA=ACTIVE ICA YES NO
IC068 G3 STORE ICA LIMITS IN NAGAP AND LAGAP IN GCOM
H1 STORE LAST AND NEXT AVAILABLE WORD ADDRESSES IN PREVIOUS ICA
IC200 COMMON H2 UPDATE CHAIN OF ICA POINTERS
H3 ACTIVE ICA AVAILABLE NO YES
K4
J1 DEFINE/ INITIALIZE OPTION YES NO
A2
B3
J2 GSP03 /KB TRANSFER FUNCTION ICA TO GCOM
K4
J3 ONLY ICA ON CHAIN YES NO
K4
K4
IC200 COMMON K3 UPDATE CHAIN OF ICA POINTERS
IC008 K4 DSPYN /DD STORE ADDRESSES OF LIGHT PEN + ORD CONT. SUBS
K5 GSP01 RETURN

Chart AC. GCAIN - GCA Initialization


A1
GCAIN
B1
GSP01 /KA
SAVE FUNCTION
C1
COMPUTE ADDRESS OF FIRST WORD OF GCA
GC010
D1
ZERO GCA
E1
FLOAT
CONVERT INTEGER TO REAL 1
GC040
F1
GSP10
STORE REAL 1 IN WORDS 5-7 AND 11-13 OF GCA
G1
STORE INTEGER 1 IN WORDS 29-32 OF GCA
H1
STORE 1023 IN WORDS 3-4 OF GCA
J1
SET INPUT AND OUTPUT MODES
K1
GSP01 RETURN

• Chart BB. EELMT - End Element

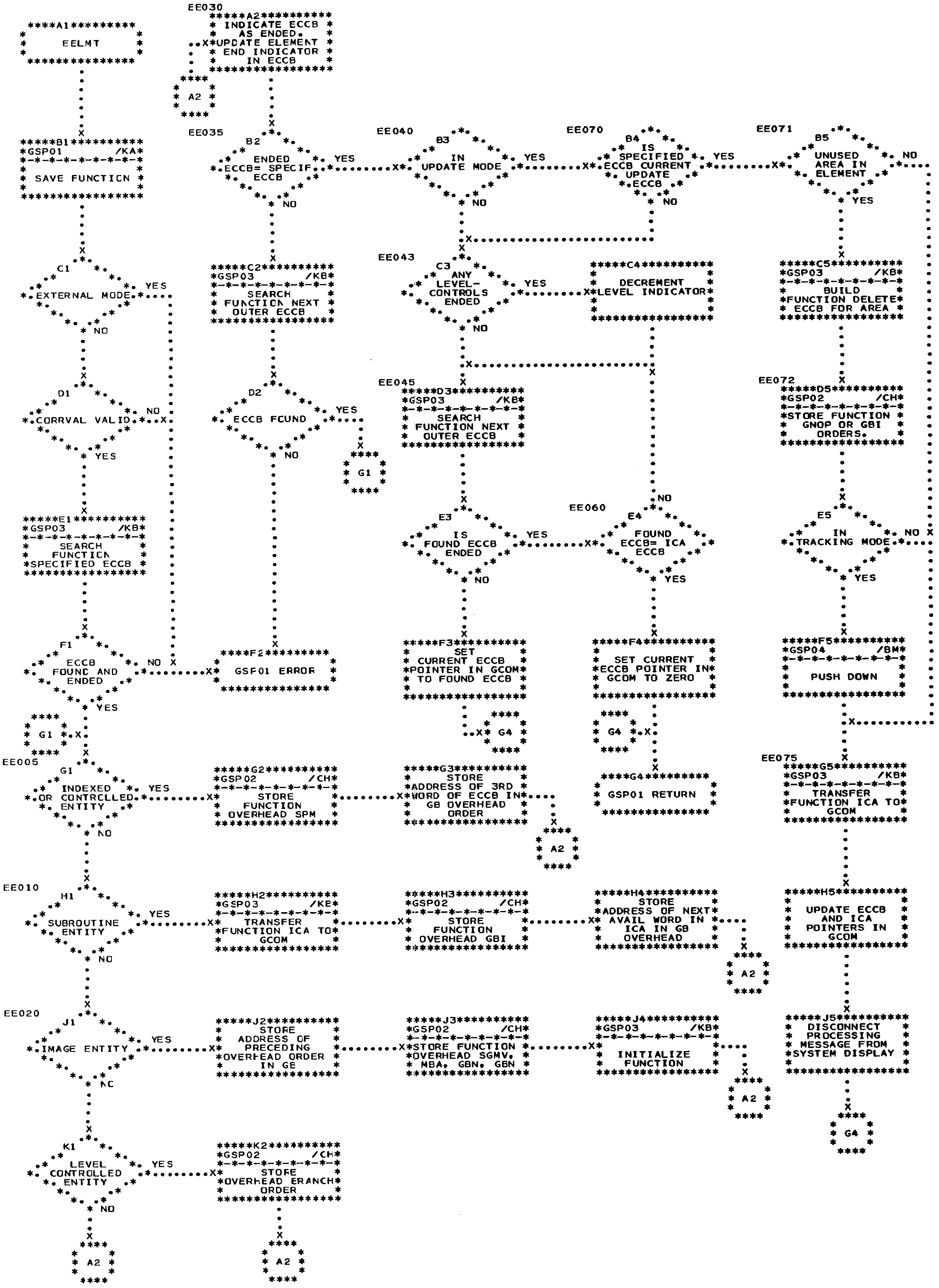
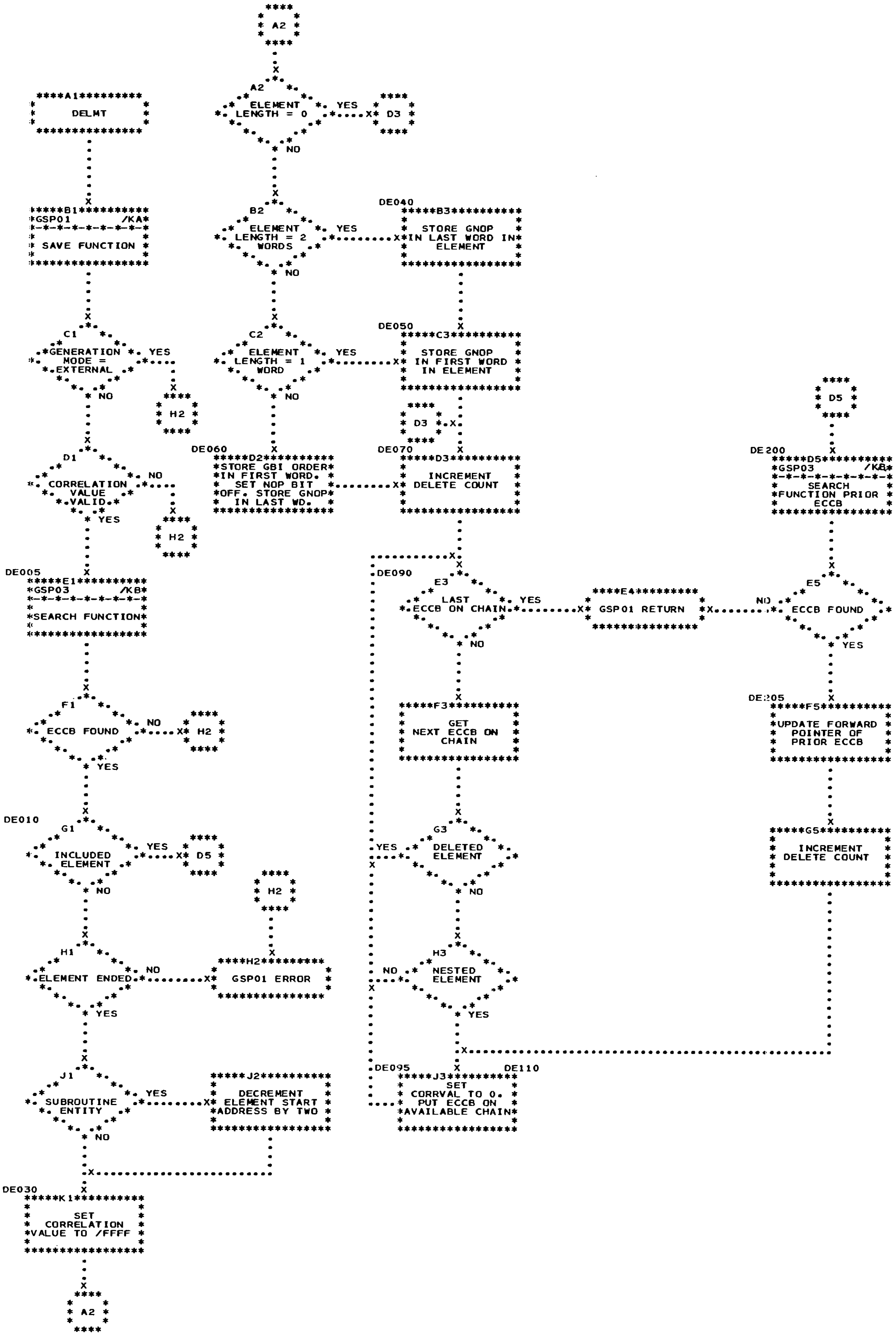

A1
EELMT
B1
GSP01 /KA
SAVE FUNCTICN
C1
EXTERNAL MODE
YES
NO
D1
CORRVAL VALID
NO
YES
E1
GSP03 /KB
SEARCH FUNCTICN SPECIFIED ECCB
F1
ECCB FOUND AND ENDED
NO
YES
G1
EE005
G1
INDEXED OR CONTRCLLED ENTITY
YES
NO
EE010
H1
SUBROUTINE ENTITY
YES
NO
EE020
J1
IMAGE ENTITY
YES
NC
K1
LEVEL CONTROLLED ENTITY
YES
NO
A2
EE030
A2
INDICATE ECCB AS ENDED. UPDATE ELEMENT END INDICATOR IN ECCB
A2
EE035
B2
ENDED ECCB= SPECIF. ECCB
YES
NO
C2
GSP03 /KB
SEARCH FUNCTICN NEXT OUTER ECCB
D2
ECCB FCUND
YES
NO
G1
F2
GSP01 ERROR
G2
GSP02 /CH
STORE FUNCTION OVERHEAD SPM
H2
GSP03 /KE
TRANSFER FUNCTION ICA TO GCOM
J2
STORE ADDRESS OF PRECEDING OVERHEAD ORDER IN GE
K2
GSP02 /CH
STORE OVERHEAD BRANCH ORDER
A2
EE040
B3
IN UPDATE MODE
YES
NO
EE043
C3
ANY LEVEL-CONTROLS ENDED
YES
NO
EE045
D3
GSP03 /KB
SEARCH FUNCTION NEXT OUTER ECCB
E3
IS FOUND ECCB ENDED
YES
NO
F3
SET CURRENT ECCB POINTER IN GCOM TO FOUND ECCB
G4
G3
STORE ADDRESS OF 3RD WORD OF ECCB IN GB OVERHEAD ORDER
A2
H3
GSP02 /CH
STORE FUNCTION OVERHEAD GBI
J3
GSP02 /CH
STORE FUNCTION OVERHEAD SGMV, MBA, GBN, GBN
EE070
B4
IS SPECIFIED ECCB CURRENT UPDATE ECCB
YES
NO
C4
DECREMENT LEVEL INDICATOR
EE060
E4
FOUND ECCB= ICA ECCB
NO
YES
F4
SET CURRENT ECCB POINTER IN GCOM TO ZERO
G4
G4
GSP01 RETURN
H4
STORE ADDRESS OF NEXT AVAIL WORD IN ICA IN GB OVERHEAD
A2
J4
GSP03 /KB
INITIALIZE FUNCTION
A2
EE071
B5
UNUSED AREA IN ELEMENT
NO
YES
C5
GSP03 /KB
BUILD FUNCTION DELETE ECCB FOR AREA
EE072
D5
GSP02 /CH
STORE FUNCTION GNOP OR GBI ORDERS.
E5
IN TRACKING MODE
NO
YES
F5
GSP04 /BM
PUSH DOWN
EE075
G5
GSP03 /KB
TRANSFER FUNCTION ICA TO GCOM
H5
UPDATE ECCB AND ICA POINTERS IN GCOM
J5
DISCONNECT PROCESSING MESSAGE FROM SYSTEM DISPLAY
G4

Chart BC. DELMT - Delete Element



• Chart BD. UELMT - Update Element

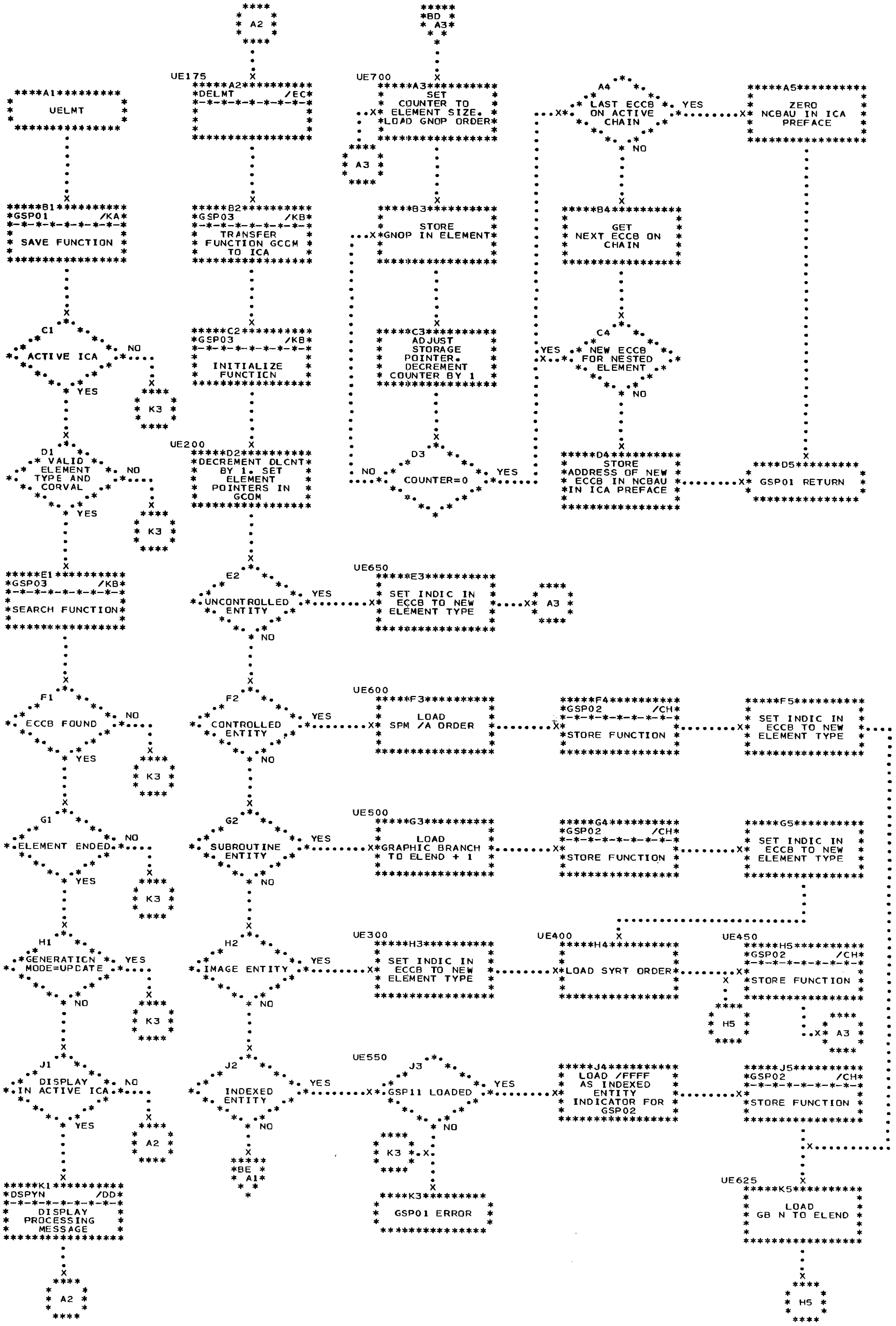
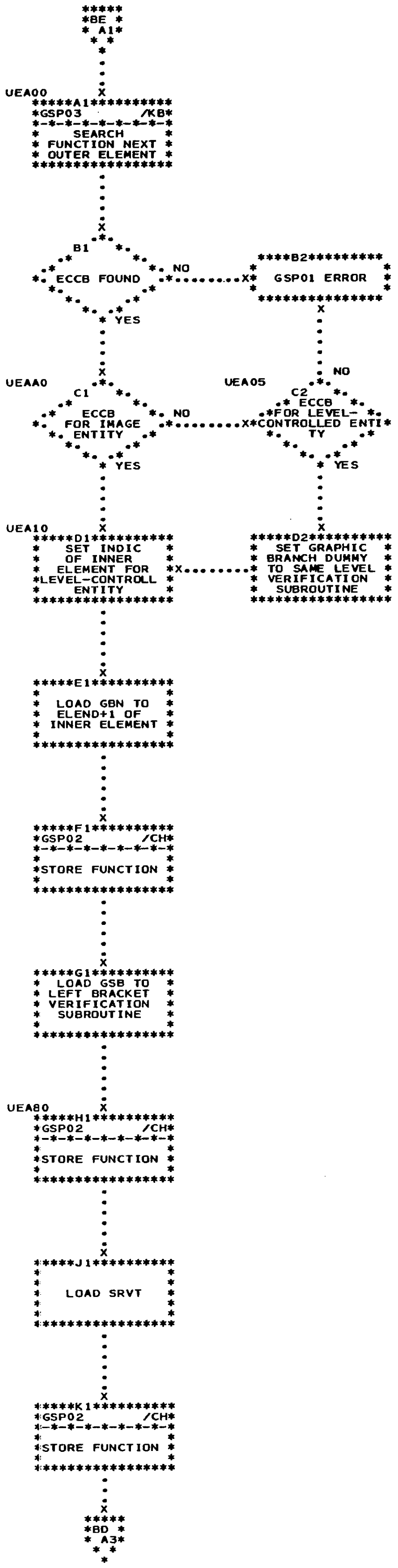


Chart BE. UELMT - Update Element (continued)


BE A1
UEA00
A1 GSP03 /KB
SEARCH FUNCTION NEXT OUTER ELEMENT
B1
ECCB FOUND
NO
B2
GSP01 ERROR
YES
UEAA0
C1
ECCB FOR IMAGE ENTITY
NO
UEA05
NO
C2
ECCB FOR LEVEL-CONTROLLED ENTITY
YES
YES
UEA10
D1
SET INDIC OF INNER ELEMENT FOR LEVEL-CONTROLL ENTITY
D2
SET GRAPHIC BRANCH DUMMY TO SAME LEVEL VERIFICATION SUBROUTINE
E1
LOAD GBN TO ELEND+1 OF INNER ELEMENT
F1 GSP02 /CH
STORE FUNCTION
G1
LOAD GSB TO LEFT BRACKET VERIFICATION SUBROUTINE
UEA80
H1 GSP02 /CH
STORE FUNCTION
J1
LOAD SRVT
K1 GSP02 /CH
STORE FUNCTION
BD A3

• Chart BF. XELMT - Extend Element

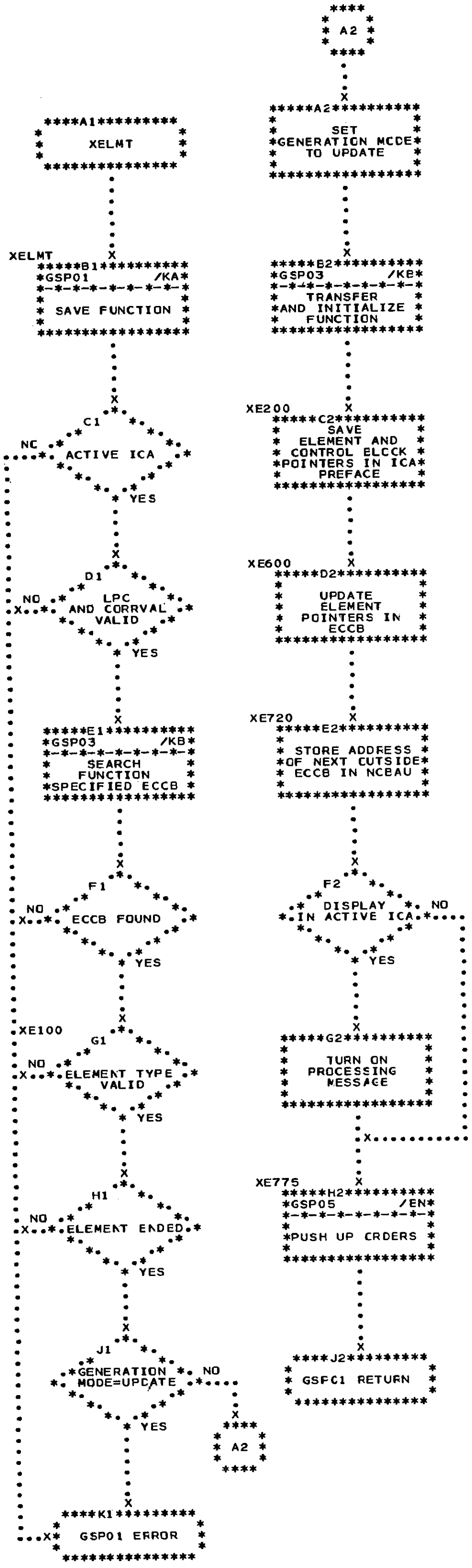


Chart BG. IELMT - Include Element

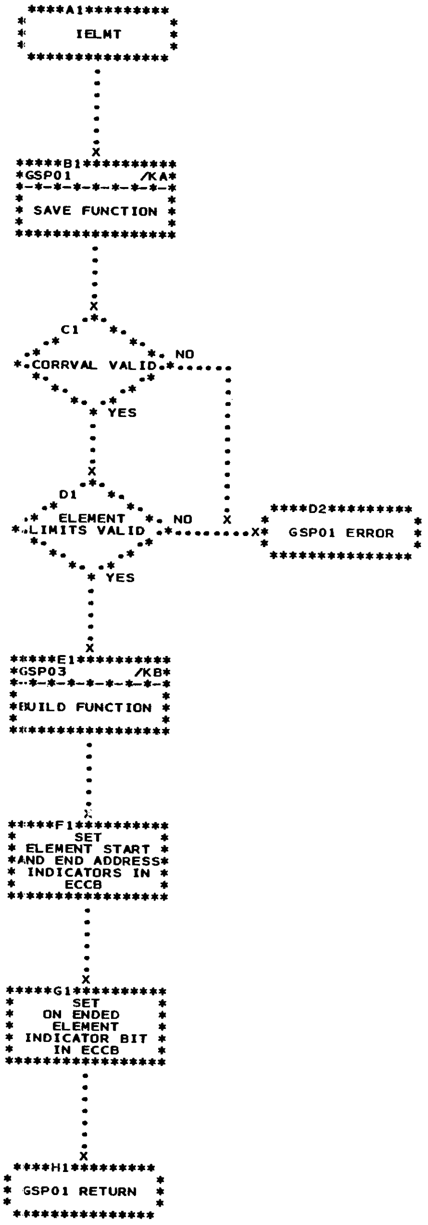
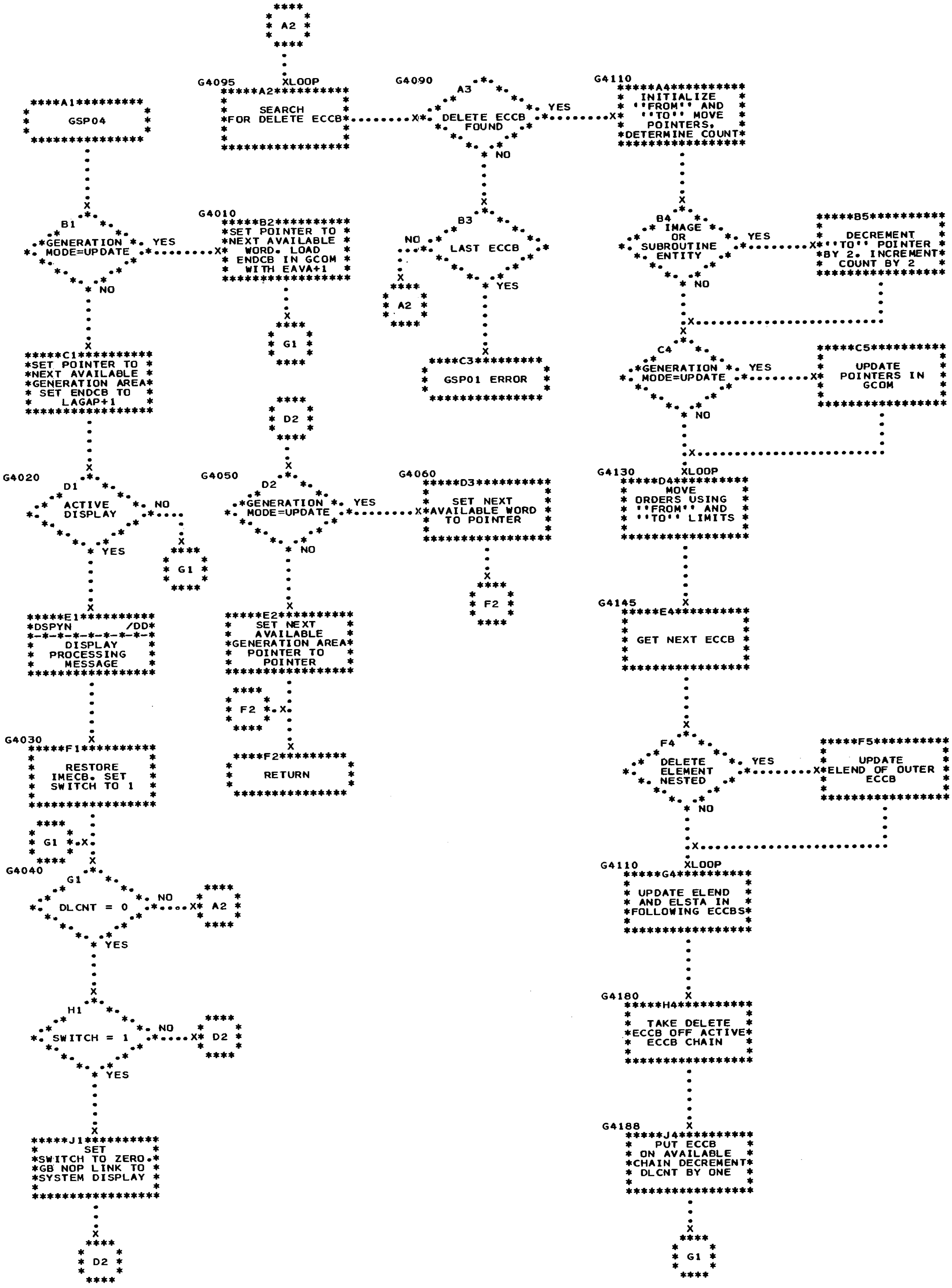
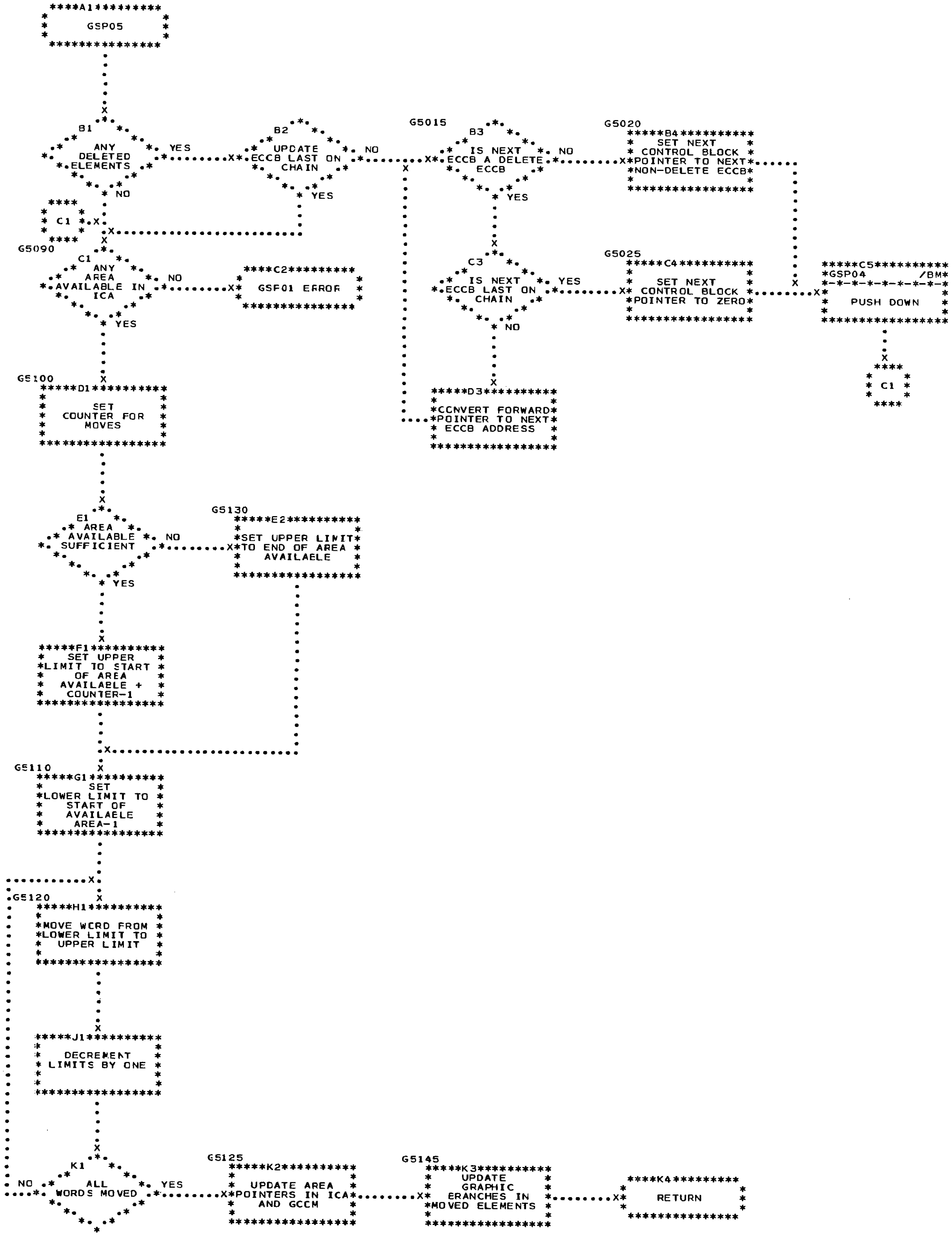


Chart BM. GSP04 - Push Down


A2
G4095 XLOOP A2
SEARCH FOR DELETE ECCB
G4090 A3
DELETE ECCB FOUND
YES
NO
G4110 A4
INITIALIZE ''FROM'' AND ''TO'' MOVE POINTERS. DETERMINE COUNT
A1
GSP04
B1
GENERATION MODE=UPDATE
YES
NO
G4010 B2
SET POINTER TO NEXT AVAILABLE WORD. LOAD ENDCB IN GCOM WITH EAVA+1
G1
B3
LAST ECCB
NO
YES
A2
C3
GSP01 ERROR
B4
IMAGE OR SUBROUTINE ENTITY
YES
NO
B5
DECREMENT ''TO'' POINTER BY 2. INCREMENT COUNT BY 2
C1
SET POINTER TO NEXT AVAILABLE GENERATION AREA SET ENDCB TO LAGAP+1
C4
GENERATION MODE=UPDATE
YES
NO
C5
UPDATE POINTERS IN GCOM
D2
G4020 D1
ACTIVE DISPLAY
NO
YES
G1
G4050 D2
GENERATION MODE=UPDATE
YES
NO
G4060 D3
SET NEXT AVAILABLE WORD TO POINTER
F2
G4130 XLOOP D4
MOVE ORDERS USING ''FROM'' AND ''TO'' LIMITS
E1
DSPYN /DD
DISPLAY PROCESSING MESSAGE
E2
SET NEXT AVAILABLE GENERATION AREA POINTER TO POINTER
G4145 E4
GET NEXT ECCB
F2
F2
RETURN
G4030 F1
RESTORE IMECB. SET SWITCH TO 1
F4
DELETE ELEMENT NESTED
YES
NO
F5
UPDATE ELEND OF OUTER ECCB
G1
G4040 G1
DLCNT = 0
NO
A2
YES
G4110 XLOOP G4
UPDATE ELEND AND ELSTA IN FOLLOWING ECCBS
H1
SWITCH = 1
NO
D2
YES
G4180 H4
TAKE DELETE ECCB OFF ACTIVE ECCB CHAIN
J1
SET SWITCH TO ZERO. GB NOP LINK TO SYSTEM DISPLAY
G4188 J4
PUT ECCB ON AVAILABLE CHAIN DECREMENT DLCNT BY ONE
D2
G1

• Chart BN. GSP05 - Push Up


A1
GSP05
B1
ANY DELETED ELEMENTS
YES
NO
B2
UPDATE ECCB LAST ON CHAIN
NO
YES
G5015
B3
IS NEXT ECCB A DELETE ECCB
NO
YES
G5020
B4
SET NEXT CONTROL BLOCK POINTER TO NEXT NON-DELETE ECCB
C1
G5090
C1
ANY AREA AVAILABLE IN ICA
NO
YES
C2
GSP01 ERROR
C3
IS NEXT ECCB LAST ON CHAIN
YES
NO
G5025
C4
SET NEXT CONTROL BLOCK POINTER TO ZERO
C5
GSP04 /BM
PUSH DOWN
C1
G5100
D1
SET COUNTER FOR MOVES
D3
CONVERT FORWARD POINTER TO NEXT ECCB ADDRESS
E1
AREA AVAILABLE SUFFICIENT
NO
YES
G5130
E2
SET UPPER LIMIT TO END OF AREA AVAILAELE
F1
SET UPPER LIMIT TO START OF AREA AVAILAELE + COUNTER-1
G5110
G1
SET LOWER LIMIT TO START OF AVAILAELE AREA-1
G5120
H1
MOVE WORD FROM LOWER LIMIT TO UPPER LIMIT
J1
DECREMENT LIMITS BY ONE
K1
ALL WORDS MOVED
NO
YES
G5125
K2
UPDATE AREA POINTERS IN ICA AND GCCM
G5145
K3
UPDATE GRAPHIC BRANCHES IN MOVED ELEMENTS
K4
RETURN

• Chart DA. EXEC - Execute Display

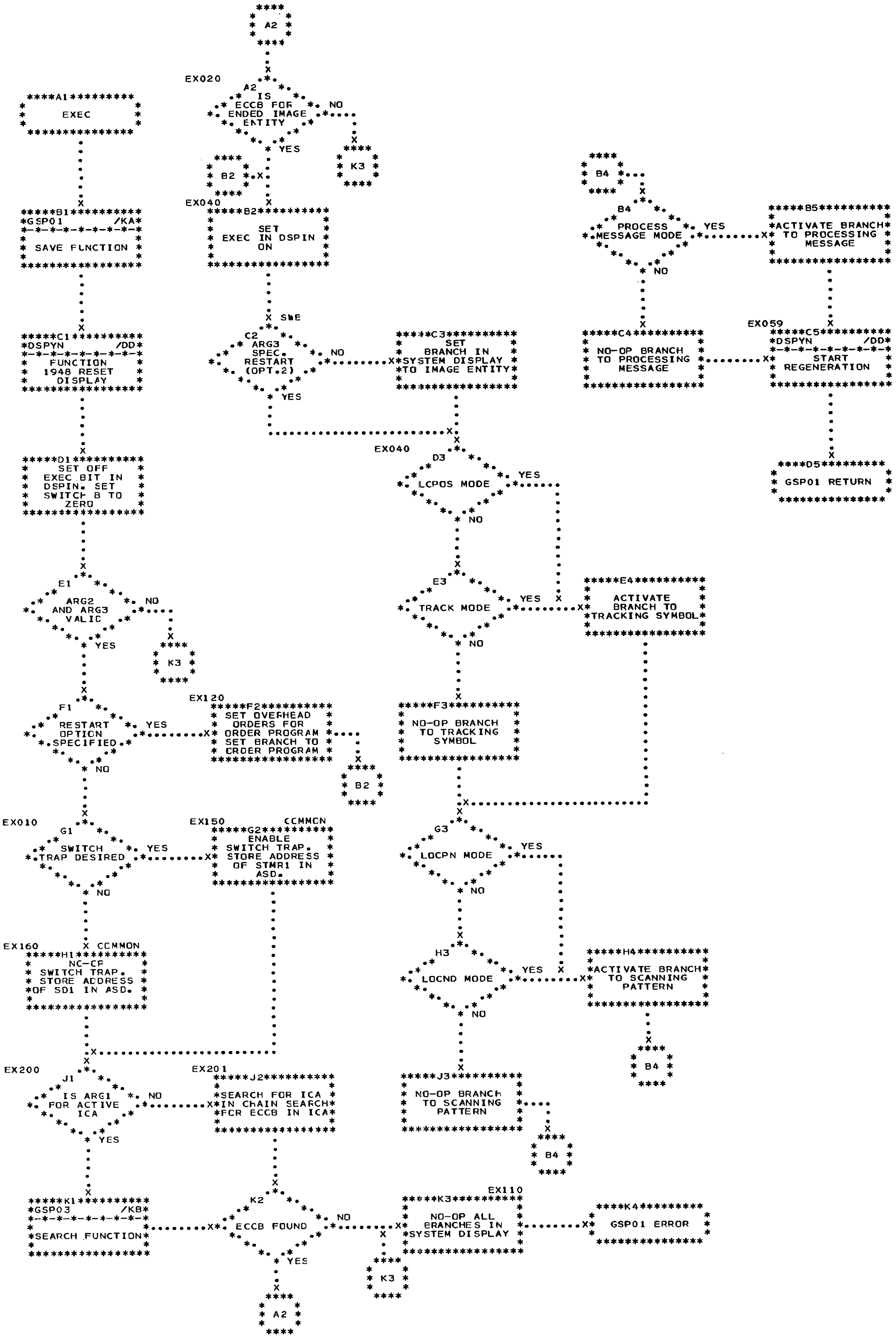

A2
A1
EXEC
EX020
A2
IS ECCB FOR ENDED IMAGE ENTITY
NO
YES
K3
B2
EX040
B2
SET EXEC IN DSPIN ON
B1
GSP01 /KA
SAVE FUNCTION
B4
B4
PROCESS MESSAGE MODE
YES
NO
B5
ACTIVATE BRANCH TO PROCESSING MESSAGE
SWE
C2
ARG3 SPEC. RESTART (OPT.2)
NO
YES
C3
SET BRANCH IN SYSTEM DISPLAY TO IMAGE ENTITY
C1
DSPYN /DD
FUNCTION 1948 RESET DISPLAY
C4
NO-OP BRANCH TO PROCESSING MESSAGE
EX059
C5
DSPYN /DD
START REGENERATION
EX040
D3
LCPOS MODE
YES
NO
D1
SET OFF EXEC BIT IN DSPIN. SET SWITCH B TO ZERO
D5
GSP01 RETURN
E1
ARG2 AND ARG3 VALID
NO
YES
K3
E3
TRACK MODE
YES
NO
E4
ACTIVATE BRANCH TO TRACKING SYMBOL
F1
RESTART OPTION SPECIFIED
YES
NO
EX120
F2
SET OVERHEAD ORDERS FOR ORDER PROGRAM SET BRANCH TO ORDER PROGRAM
B2
F3
NO-OP BRANCH TO TRACKING SYMBOL
EX010
G1
SWITCH TRAP DESIRED
YES
NO
EX150
COMMON
G2
ENABLE SWITCH TRAP. STORE ADDRESS OF STMR1 IN ASD.
G3
LOCPN MODE
YES
NO
EX160
COMMON
H1
NO-OP SWITCH TRAP. STORE ADDRESS OF SD1 IN ASD.
H3
LOCND MODE
YES
NO
H4
ACTIVATE BRANCH TO SCANNING PATTERN
B4
EX200
J1
IS ARG1 FOR ACTIVE ICA
NO
YES
EX201
J2
SEARCH FOR ICA IN CHAIN SEARCH FOR ECCB IN ICA
J3
NO-OP BRANCH TO SCANNING PATTERN
B4
K1
GSP03 /KB
SEARCH FUNCTION
K2
ECCB FOUND
NO
YES
A2
EX110
K3
NO-OP ALL BRANCHES IN SYSTEM DISPLAY
K3
K4
GSP01 ERROR

Chart DB. TMDSP - Terminate Display

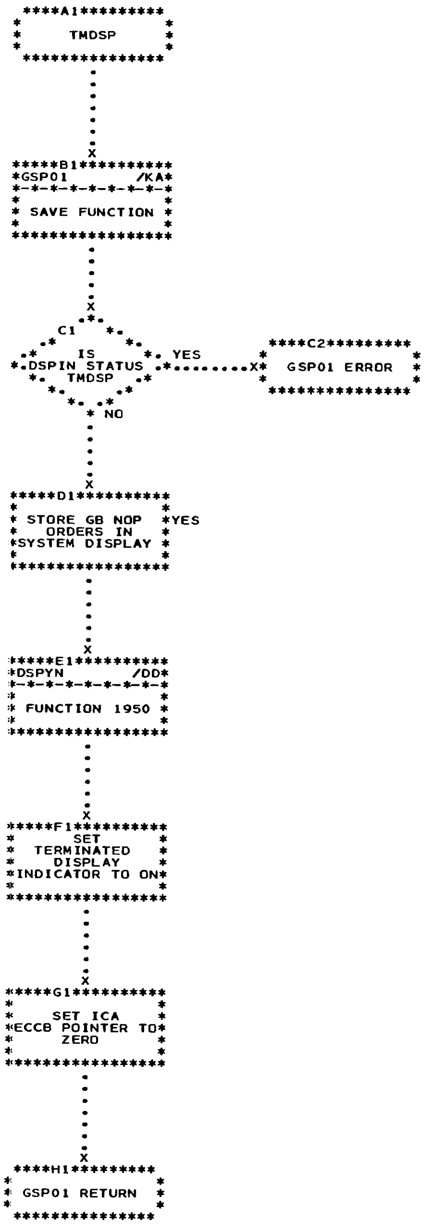
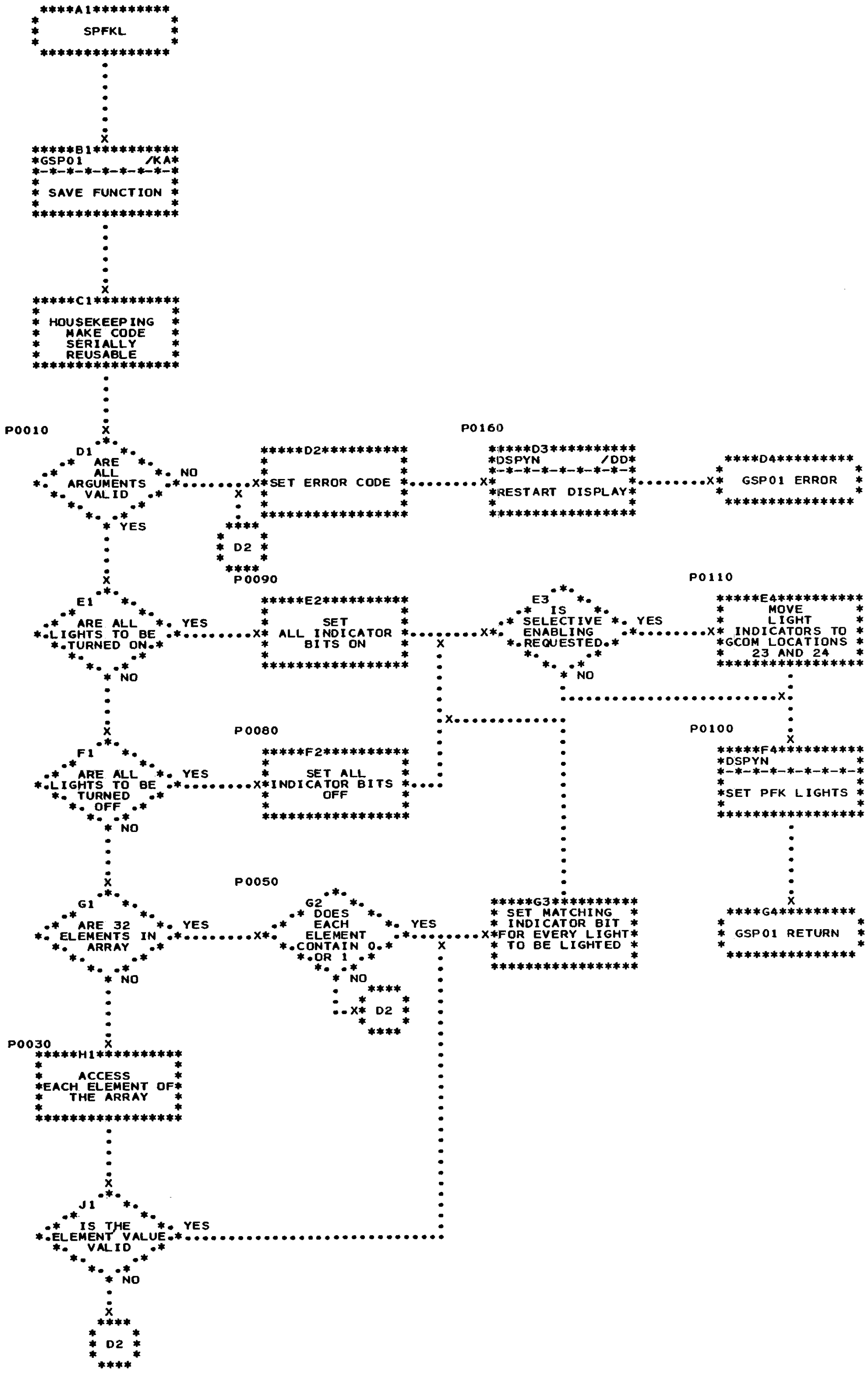
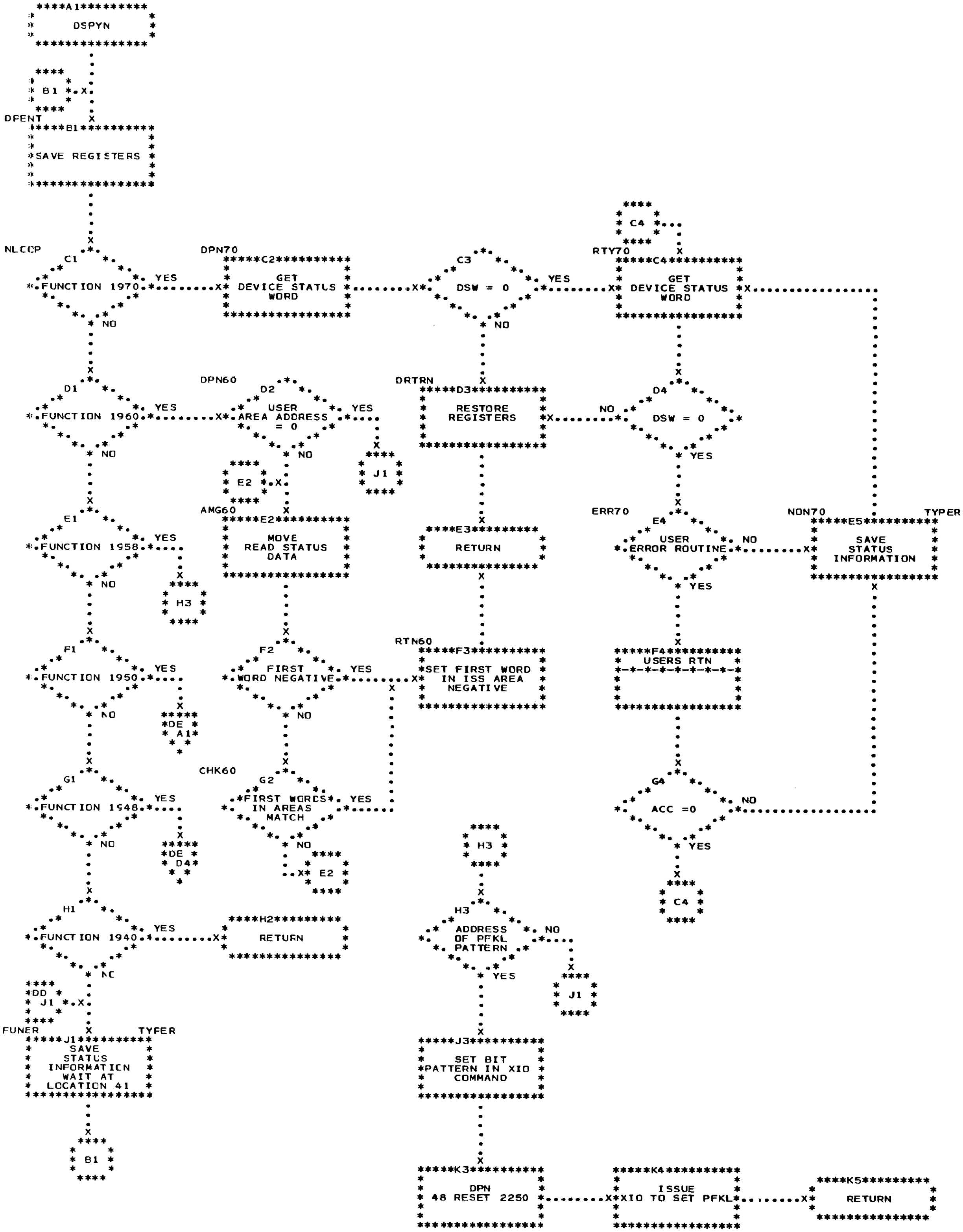


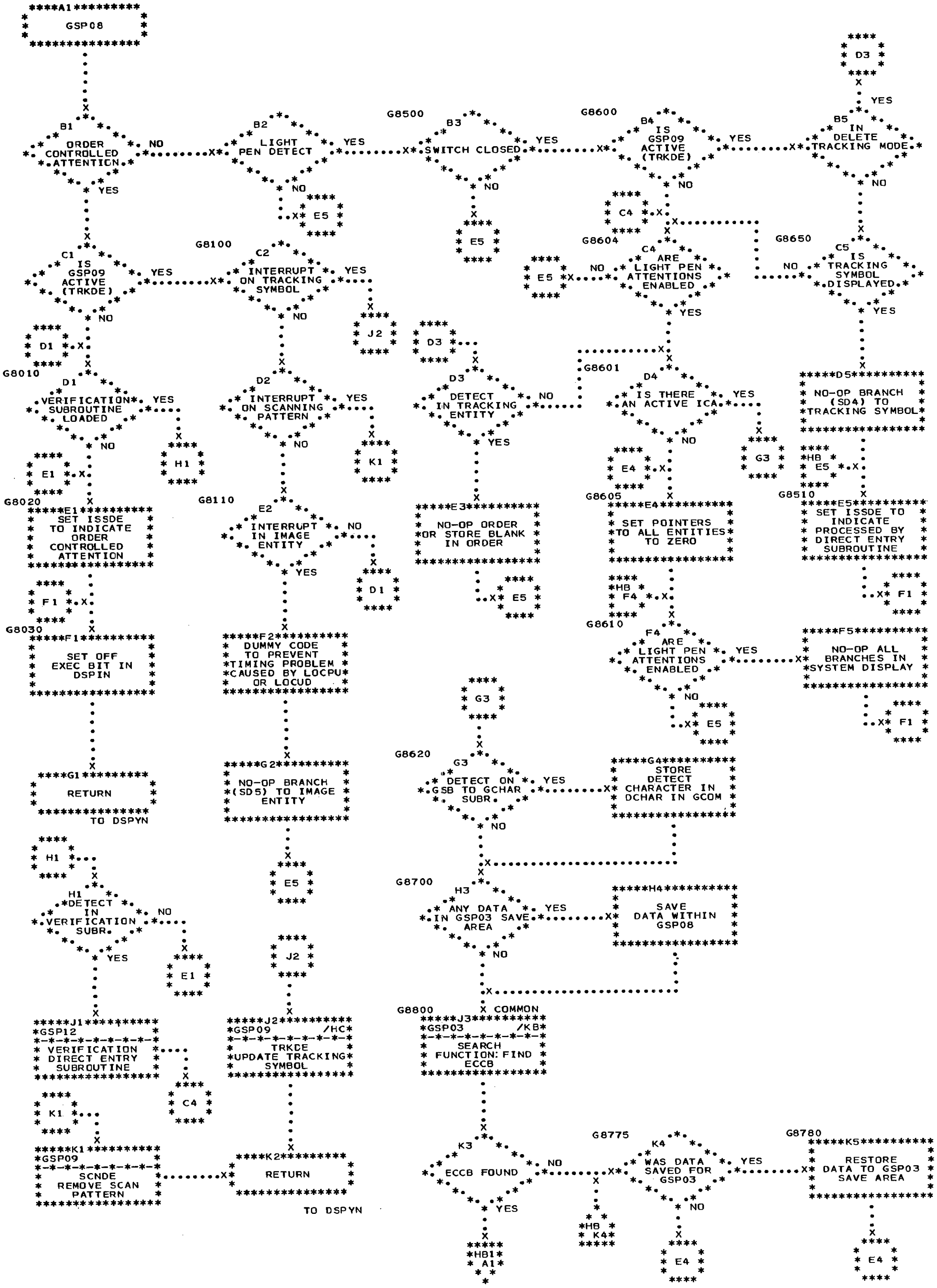
Chart DC. SPFKL - Set Programmed Function Keyboard Lights


A1
SPFKL
B1
GSP01 /KA
SAVE FUNCTION
C1
HOUSEKEEPING MAKE CODE SERIALLY REUSABLE
P0010
D1
ARE ALL ARGUMENTS VALID
NO
YES
D2
SET ERROR CODE
P0160
D3
DSPYN /DD
RESTART DISPLAY
D4
GSP01 ERROR
D2
P0090
E1
ARE ALL LIGHTS TO BE TURNED ON
YES
NO
E2
SET ALL INDICATOR BITS ON
E3
IS SELECTIVE ENABLING REQUESTED
YES
NO
P0110
E4
MOVE LIGHT INDICATORS TO GCOM LOCATIONS 23 AND 24
P0080
F1
ARE ALL LIGHTS TO BE TURNED OFF
YES
NO
F2
SET ALL INDICATOR BITS OFF
P0100
F4
DSPYN
SET PFK LIGHTS
P0050
G1
ARE 32 ELEMENTS IN ARRAY
YES
NO
G2
DOES EACH ELEMENT CONTAIN 0 OR 1
YES
NO
D2
G3
SET MATCHING INDICATOR BIT FOR EVERY LIGHT TO BE LIGHTED
G4
GSP01 RETURN
P0030
H1
ACCESS EACH ELEMENT OF THE ARRAY
J1
IS THE ELEMENT VALUE VALID
YES
NO
D2

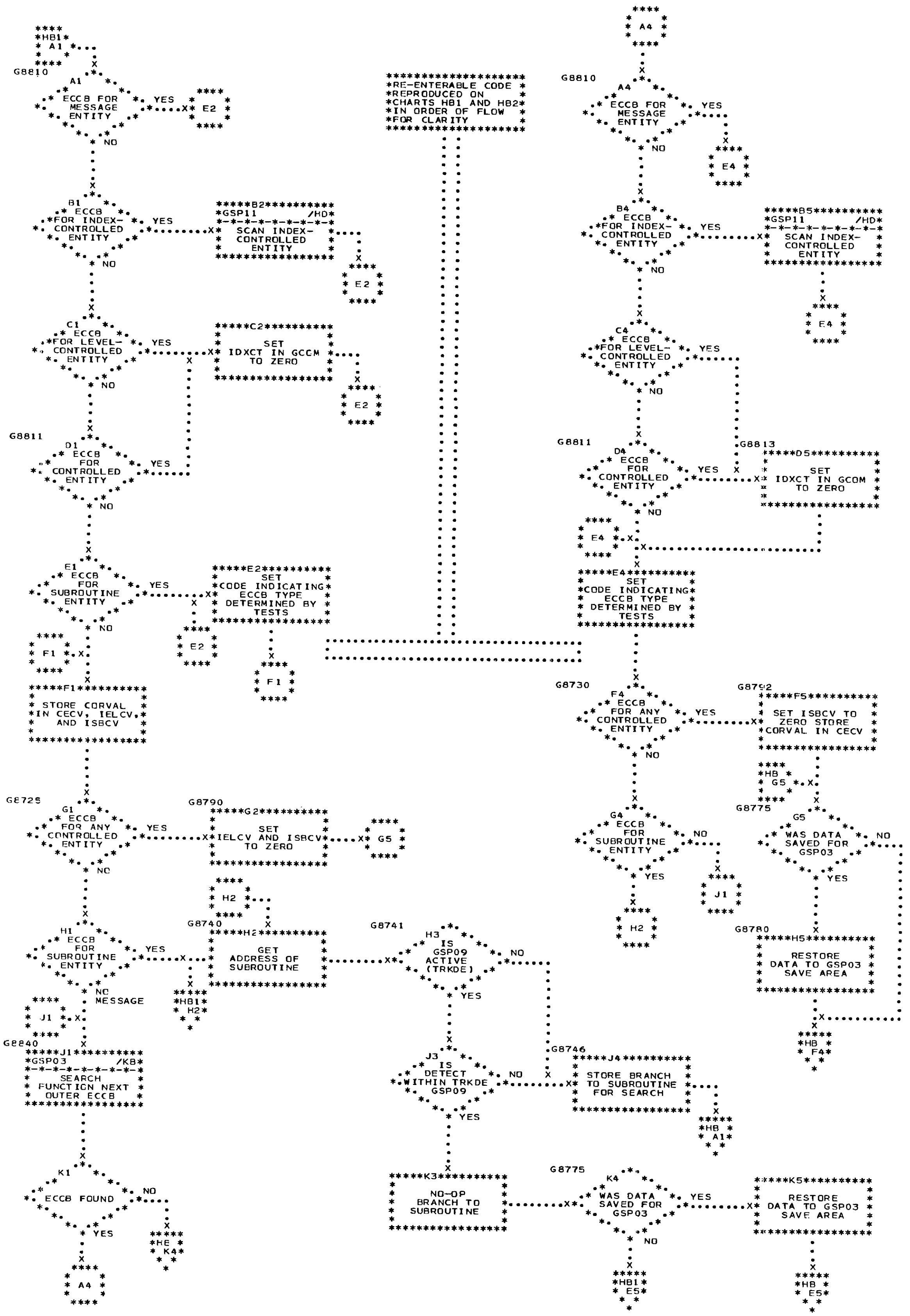
• Chart DD. DSPYN - 2250 I/O Subroutine



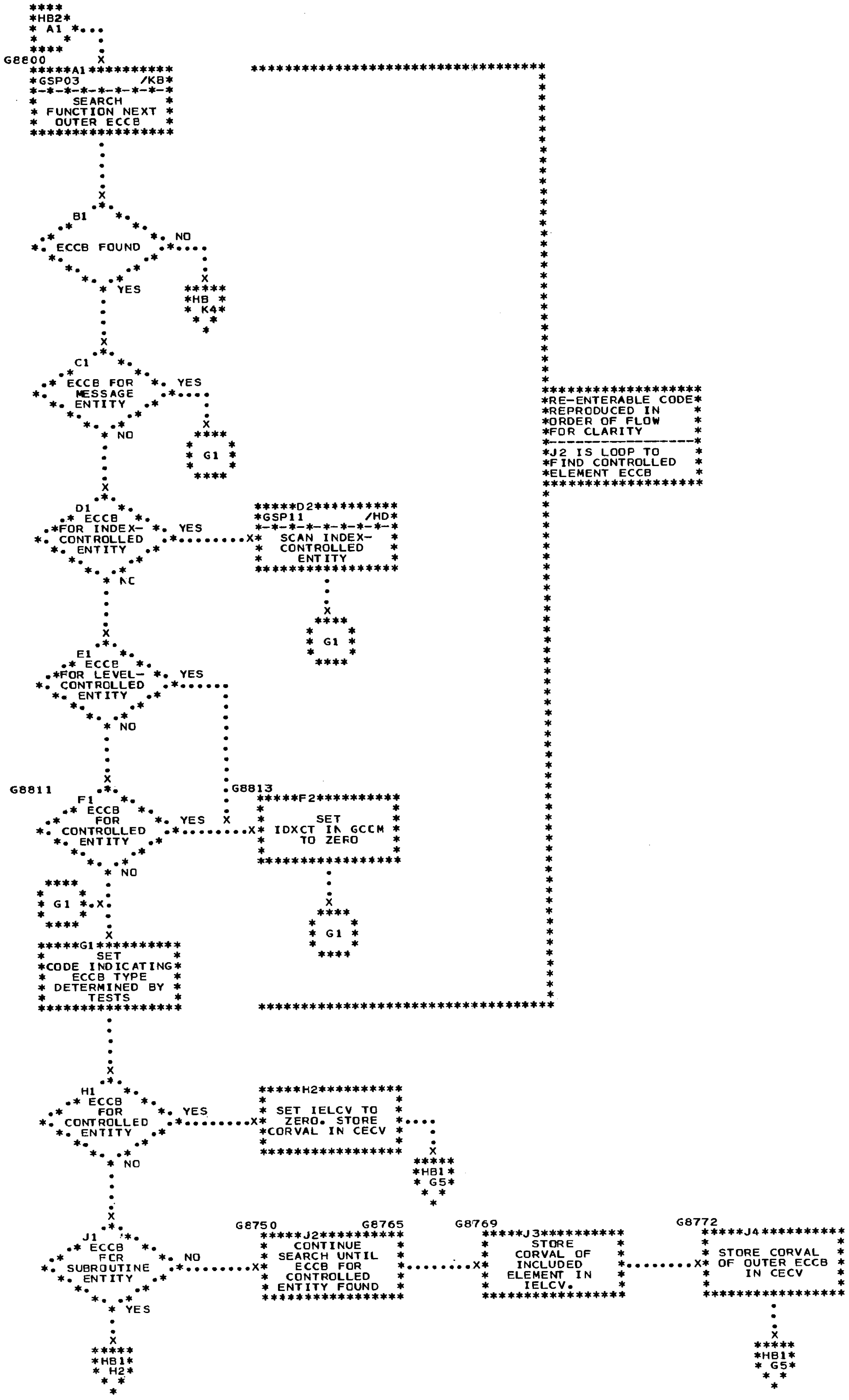
• Chart HB. GSP08 - Light Pen and Order Controlled Direct Entry Subroutine


A1 GSP08
B1 ORDER CONTROLLED ATTENTION
NO
YES
G8100 B2 LIGHT PEN DETECT
YES
NO
E5
G8500 B3 SWITCH CLOSED
YES
NO
E5
G8600 B4 IS GSP09 ACTIVE (TRKDE)
YES
NO
D3
YES
B5 IN DELETE TRACKING MODE
NO
C1 IS GSP09 ACTIVE (TRKDE)
YES
NO
C2 INTERRUPT ON TRACKING SYMBOL
YES
NO
J2
C4
G8604 C4 ARE LIGHT PEN ATTENTIONS ENABLED
NO
YES
E5
G8650 C5 IS TRACKING SYMBOL DISPLAYED
NO
YES
D1
G8010 D1 VERIFICATION SUBROUTINE LOADED
YES
NO
H1
D2 INTERRUPT ON SCANNING PATTERN
YES
NO
K1
D3
D3 DETECT IN TRACKING ENTITY
NO
YES
G8601 D4 IS THERE AN ACTIVE ICA
YES
NO
G3
D5 NO-OP BRANCH (SD4) TO TRACKING SYMBOL
HB E5
E1
G8020 E1 SET ISSDE TO INDICATE ORDER CONTROLLED ATTENTION
G8110 E2 INTERRUPT IN IMAGE ENTITY
NO
YES
D1
E3 NO-OP ORDER OR STORE BLANK IN ORDER
E5
E4
G8605 E4 SET POINTERS TO ALL ENTITIES TO ZERO
G8510 E5 SET ISSDE TO INDICATE PROCESSED BY DIRECT ENTRY SUBROUTINE
F1
F1
G8030 F1 SET OFF EXEC BIT IN DSPIN
F2 DUMMY CODE TO PREVENT TIMING PROBLEM CAUSED BY LOCPU OR LOCUD
HB F4
G8610 F4 ARE LIGHT PEN ATTENTIONS ENABLED
YES
NO
E5
F5 NO-OP ALL BRANCHES IN SYSTEM DISPLAY
F1
G1 RETURN
TO DSPYN
G2 NO-OP BRANCH (SD5) TO IMAGE ENTITY
G3
G8620 G3 DETECT ON GSB TO GCHAR SUBR.
YES
NO
G4 STORE DETECT CHARACTER IN DCHAR IN GCOM
H1
H1 DETECT IN VERIFICATION SUBR.
NO
YES
E1
E5
G8700 H3 ANY DATA IN GSP03 SAVE AREA
YES
NO
H4 SAVE DATA WITHIN GSP08
J2
J1 GSP12 VERIFICATION DIRECT ENTRY SUBROUTINE
C4
J2 GSP09 /HC TRKDE UPDATE TRACKING SYMBOL
G8800 COMMON
J3 GSP03 /KB SEARCH FUNCTION: FIND ECCB
K1
K1 GSP09 SCNDE REMOVE SCAN PATTERN
K2 RETURN
TO DSPYN
K3 ECCB FOUND
NO
YES
HB1 A1
HB K4
G8775 K4 WAS DATA SAVED FOR GSP03
YES
NO
E4
G8780 K5 RESTORE DATA TO GSP03 SAVE AREA
E4

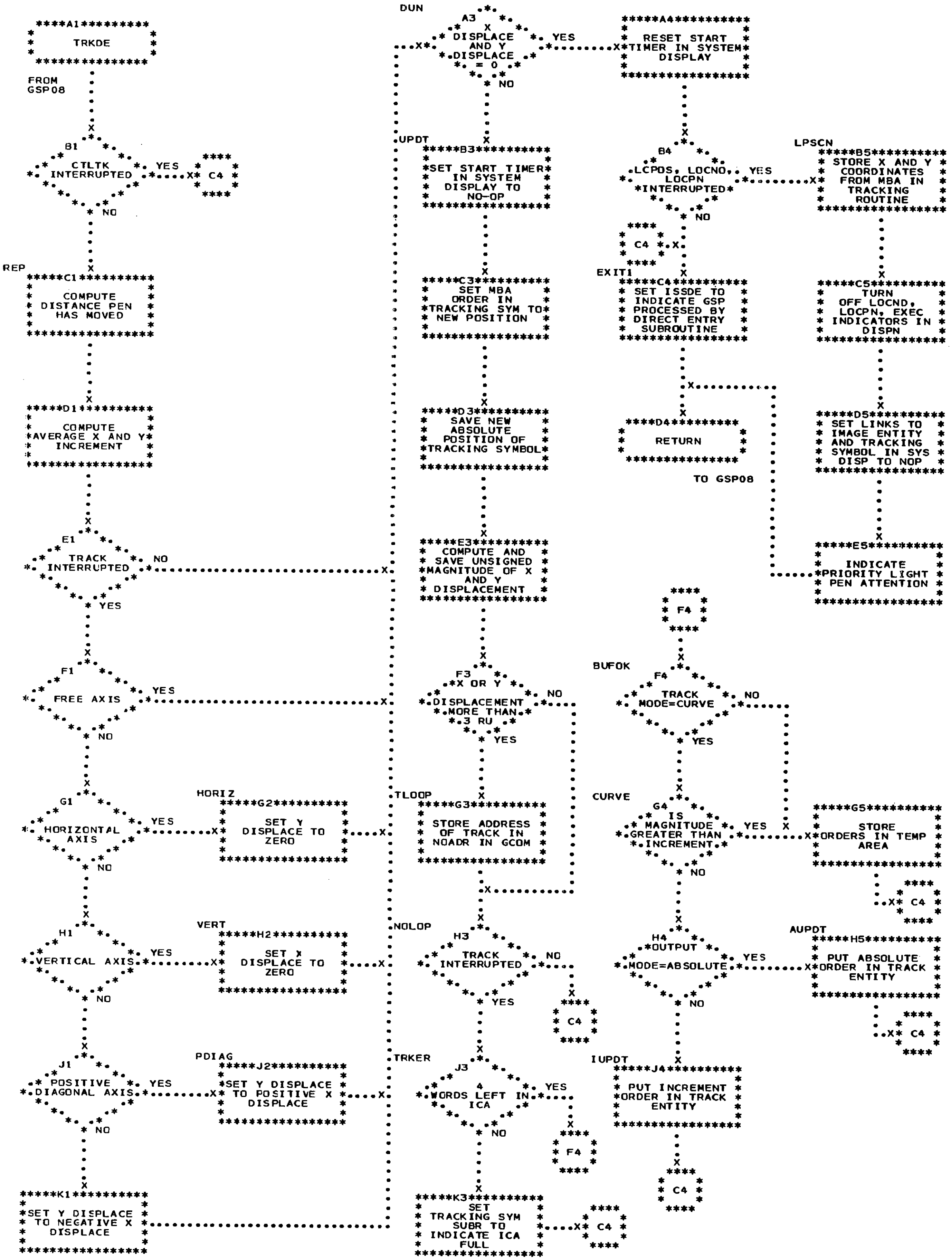
• Chart HB1. GSP08 - Light Pen and Order Controlled Direct Entry Subroutine (continued)


HB1 A1
G8810
A1
ECCB FOR MESSAGE ENTITY
YES
E2
NO
B1
ECCB FOR INDEX-CONTROLLED ENTITY
YES
B2
GSP11 /HD
SCAN INDEX-CONTROLLED ENTITY
E2
NO
C1
ECCB FOR LEVEL-CONTROLLED ENTITY
YES
C2
SET IDXCT IN GCCM TO ZERO
E2
NO
G8811
D1
ECCB FOR CONTROLLED ENTITY
YES
NO
E1
ECCB FOR SUBROUTINE ENTITY
YES
E2
SET CODE INDICATING ECCB TYPE DETERMINED BY TESTS
E2
F1
NO
F1
F1
STORE CORVAL IN CECV, IELCV, AND ISBCV
G8725
G1
ECCB FOR ANY CONTROLLED ENTITY
YES
G8790
G2
SET IELCV AND ISBCV TO ZERO
G5
NO
H2
G8740
H2
GET ADDRESS OF SUBROUTINE
HB1 H2
H1
ECCB FOR SUBROUTINE ENTITY
YES
NO MESSAGE
J1
G8840
J1
GSP03 /KB
SEARCH FUNCTION NEXT OUTER ECCB
K1
ECCB FOUND
NO
HE K4
YES
A4
RE-ENTERABLE CODE REPRODUCED ON CHARTS HB1 AND HB2 IN ORDER OF FLOW FOR CLARITY
A4
G8810
A4
ECCB FOR MESSAGE ENTITY
YES
E4
NO
B4
ECCB FOR INDEX-CONTROLLED ENTITY
YES
B5
GSP11 /HD
SCAN INDEX-CONTROLLED ENTITY
E4
NO
C4
ECCB FOR LEVEL-CONTROLLED ENTITY
YES
G8811
D4
ECCB FOR CONTROLLED ENTITY
YES
G8813
D5
SET IDXCT IN GCOM TO ZERO
NO
E4
E4
SET CODE INDICATING ECCB TYPE DETERMINED BY TESTS
G8730
F4
ECCB FOR ANY CONTROLLED ENTITY
YES
G8792
F5
SET ISBCV TO ZERO STORE CORVAL IN CECV
NO
HB G5
G4
ECCB FOR SUBROUTINE ENTITY
NO
J1
YES
H2
G8775
G5
WAS DATA SAVED FOR GSP03
NO
YES
G8780
H5
RESTORE DATA TO GSP03 SAVE AREA
HB F4
G8741
H3
IS GSP09 ACTIVE (TRKDE)
NO
YES
J3
IS DETECT WITHIN TRKDE GSP09
NO
G8746
J4
STORE BRANCH TO SUBROUTINE FOR SEARCH
HB A1
YES
K3
NO-OP BRANCH TO SUBROUTINE
G8775
K4
WAS DATA SAVED FOR GSP03
YES
K5
RESTORE DATA TO GSP03 SAVE AREA
NO
HB1 E5
HB E5

• Chart HB2. GSP08 - Light Pen and Order Controlled Direct Entry Subroutine (continued)


HB2 A1
G8800
A1 GSP03 /KB
SEARCH FUNCTION NEXT OUTER ECCB
B1
ECCB FOUND
NO
HB K4
YES
C1
ECCB FOR MESSAGE ENTITY
YES
G1
NO
D1
ECCB FOR INDEX- CONTROLLED ENTITY
YES
D2 GSP11 /HD
SCAN INDEX- CONTROLLED ENTITY
G1
NC
E1
ECCB FOR LEVEL- CONTROLLED ENTITY
YES
NO
G8811
F1
ECCB FOR CONTROLLED ENTITY
YES
G8813
F2
SET IDXCT IN GCCM TO ZERO
G1
NO
G1
G1
SET CODE INDICATING ECCB TYPE DETERMINED BY TESTS
RE-ENTERABLE CODE REPRODUCED IN ORDER OF FLOW FOR CLARITY
J2 IS LOOP TO FIND CONTROLLED ELEMENT ECCB
H1
ECCB FOR CONTROLLED ENTITY
YES
H2
SET IELCV TO ZERO. STORE CORVAL IN CECV
HB1 G5
NO
J1
ECCB FOR SUBROUTINE ENTITY
NO
G8750
G8765
J2
CONTINUE SEARCH UNTIL ECCB FOR CONTROLLED ENTITY FOUND
G8769
J3
STORE CORVAL OF INCLUDED ELEMENT IN IELCV.
G8772
J4
STORE CORVAL OF OUTER ECCB IN CECV
HB1 G5
YES
HB1 H2

• Chart HC. GSP09 - Scanning Pattern and Tracking Symbol Direct Entry Subroutine


A1
TRKDE
FROM GSP08
B1
CTLTK INTERRUPTED
YES
C4
NO
REP
C1
COMPUTE DISTANCE PEN HAS MOVED
D1
COMPUTE AVERAGE X AND Y INCREMENT
E1
TRACK INTERRUPTED
NO
YES
F1
FREE AXIS
YES
NO
G1
HORIZONTAL AXIS
YES
NO
HORIZ
G2
SET Y DISPLACE TO ZERO
H1
VERTICAL AXIS
YES
NO
VERT
H2
SET X DISPLACE TO ZERO
J1
POSITIVE DIAGONAL AXIS
YES
NO
PDIAG
J2
SET Y DISPLACE TO POSITIVE X DISPLACE
K1
SET Y DISPLACE TO NEGATIVE X DISPLACE
DUN
A3
X DISPLACE AND Y DISPLACE = 0
YES
NO
UPDT
B3
SET START TIMER IN SYSTEM DISPLAY TO NO-OP
C3
SET MBA ORDER IN TRACKING SYM TO NEW POSITION
D3
SAVE NEW ABSOLUTE POSITION OF TRACKING SYMBOL
E3
COMPUTE AND SAVE UNSIGNED MAGNITUDE OF X AND Y DISPLACEMENT
F3
X OR Y DISPLACEMENT MORE THAN 3 RU
NO
YES
TLOOP
G3
STORE ADDRESS OF TRACK IN NOADR IN GCOM
NOLOP
H3
TRACK INTERRUPTED
NO
YES
C4
TRKER
J3
4 WORDS LEFT IN ICA
YES
NO
F4
K3
SET TRACKING SYM SUBR TO INDICATE ICA FULL
C4
A4
RESET START TIMER IN SYSTEM DISPLAY
B4
LCPOS, LOCNO, LOCPN INTERRUPTED
YES
NO
C4
EXIT1
C4
SET ISSDE TO INDICATE GSP PROCESSED BY DIRECT ENTRY SUBROUTINE
D4
RETURN
TO GSP08
F4
BUFOK
F4
TRACK MODE=CURVE
NO
YES
CURVE
G4
IS MAGNITUDE GREATER THAN INCREMENT
YES
NO
H4
OUTPUT MODE=ABSOLUTE
YES
NO
IUPDT
J4
PUT INCREMENT ORDER IN TRACK ENTITY
C4
LPSCN
B5
STORE X AND Y COORDINATES FROM MBA IN TRACKING ROUTINE
C5
TURN OFF LOCND, LOCPN, EXEC INDICATORS IN DISPN
D5
SET LINKS TO IMAGE ENTITY AND TRACKING SYMBOL IN SYS DISP TO NOP
E5
INDICATE PRIORITY LIGHT PEN ATTENTION
G5
STORE ORDERS IN TEMP AREA
C4
AUPDT
H5
PUT ABSOLUTE ORDER IN TRACK ENTITY
C4

```
************************
* Base Address of GCOM *
************************
*****************   ATTENTION HANDLING FIELDS *************************

LPDE            0    Address of light pen direct entry subroutine    0000
                     (set by ICAIN)

OCDE            1    Address of order controlled direct entry        0000
                     subroutine (set by ICAIN)

ANDE            2    Address of alphameric keyboard direct entry     0000
                     subroutine (set by ICURS)

ENDDE           3    Address of END key direct entry subroutine      0000
                     (user provided)

CANDE           4    Address of CANCEL key direct entry              0000

PFDE            5    Address of program function keyboard direct     0000
                     entry subroutine (user provided)

ATNDT        6-11    Attention data set by XIO Read Status and used
                     by DSPYN

ATNAD           6    Address of attention order                      0000

ATNTP           7    Attention type (bits 0-10)                      0000

                     bit pattern fcr bits 0-10
                       0-order controlled attention
                       1-alphameric keyboard
                       2-detect attention, light pen enabled
                       4-detect status, light pen disabled
                       7-light  pen  switch closed when Start  Timer
                         instruction entered
                       8-busy
                       9-character mode (1=character; 0=graphic)
                      10-graphic mode (1=point; 0=vector)

ATNDS           7    Attention order address displacement (bits 14-15)

ATNX            8    Attention x ccordinate (bits 6-15)              0000

ATNY            9    Attention y coordinate (bits 6-15)              0000

ATNPF          10    Attention program function keyboard             0000
                     (bits 3-7)

ATNOV          10    Attention program function keyboard overlay     0000
                     (bits 8-15)

ATNAN          11    Attention alphameric key (bits 3-7)             0000

DCHAR          11    Detected character on light pen attention       0000
                     (set by GSP08)

ISSDE          12    Internal service subroutine  indicator          0000
                     (bits 12-15)

                     bit pattern fcr bits 12-15
                       0000 not processed by direct entry subroutine
                       0001 priority light pen attention
                       0010 light pen attention for program
                       0100 order controlled for program
                       1000 processed by direct entry subroutine

ICACB          13    Address of ICA ECCB                             0000
```

IMECB	14	Address of image entity ECCB	0000
OADDR	15	Current order address	0000
NOADR	16	Next order address (to be generated)	0000
ICACV	17	Correlation value for ICA ECCB	0000
IECV	18	Correlation value for image entity	0000
CECV	19	Correlation value for controlled entity	0000
IELCV	20	Correlation value for inner element	0000
SUBCV	21	Correlation value for subroutine or tracking entity	0000
ISBCV	22	Correlation value for inner element of subroutine or tracking entity	0000
PFKI1	23	Indicate which program function keyboard lights are enabled.	0000
PFKI2	24	Bits 0-31 correspond to lights 0-31	
SATEN	25	Word to indicate attention status (bits 11-15) bit pattern for bits 11-15 00000 all attentions disabled 00001 order controlled interrupts enabled 00010 light pen attentions enabled 00100 END key enabled 01000 alphameric keyboard enabled 10000 program function keyboard enabled	
RQNDA	26	Address of attention queue in DSPYN	0000
CMECB	27	Address of the ECCB for the message entity in which there is a GB to the cursor	0000
DSPIN	28	Display word indicator (bits 0-6) bit pattern for bits 0-6 1000000 LCPOS 0100000 TRACK 0010000 LOCPN 0001000 LOCND 0000100 PROCESSING 0000010 TMDSP 0000001 EXEC	0000

******************* IMAGE GENERATION FIELDS ****************************

NAGAP	29	Address of next available word in ICA for storing graphic order	0000
LAGAP	30	Address of last available word in ICA for storing graphic order	0000
GENIN	31	Generation mode indicator word (bits 14-15) bit pattern for bits 14-15 00 normal 01 update 10 external	0000
LGXI	32	Last generated x integer	0200