File No. 1130-25
Form Y27-7174-0

Program Logic

IBM 1130/2250 Graphic Subroutine Package For Basic FORTRAN IV

Program Number 1130-LM-008

This publication describes the internal logic of the 1130/2250 Graphic Subroutine Package (GSP) for Basic FORTRAN IV. The GSP is a group of subroutines that the FORTRAN programmer uses to write graphics programs for the IBM 2250 Display Unit, Model 4, attached to an 1130 Computing System. The GSP is not an extension of FORTRAN, and it may also be used by Assembler-language programs.

This program logic manual is intended for use by persons involved in program maintenance and system programmers who are altering the program design. Its primary purpose is to serve as a guide to the program listings with which it is to be used. Since program logic information is not necessary for using the program, distribution of this publication is limited to those with the aforementioned requirements.

Restricted Distribution

PREFACE

The purpose of this publication is to assist those persons concerned with adjusting or re-designing the logic of the subroutines that comprise the 1130/2250 Graphic Subroutine Package for FORTRAN IV. The program logic manual is primarily a guide to the listings, and should properly be used with the listings. It should not be used in place of the listings when an understanding of the logic of the program at the coding level is necessary.

This manual is divided into six areas: (1) an introduction consisting of an overview of the Graphic Subroutine Package and its relation to the 1130 Disk Monitor System, Version 2; (2) a theory of operation describing the logic and design of the Graphic Subroutine Package and how communication and control is effected; (3) GSP subroutines, a discussion of each subroutine in detail; (4) diagnostic aids; (5) flowcharts; and (6) appendixes.

The reader is assumed to have an understanding of the external operations of the IBM 1130 Computing System, the Graphic Subroutine Package for FORTRAN IV, and the functional characteristics of the IBM 2250 Display Unit, Model 4.

PREREQUISITE PUBLICATIONS

It is assumed that the reader is familiar with the following publications:

<u>IBM 1130 Computing System Component Description: IBM 2250 Display Unit Model 4</u>, Form A27-2723
<u>IBM 1130/2250 Graphic Subroutine Package for Basic FORTRAN IV</u>, Form C27-6934

The following publications are recommended for reference:

<u>IBM 1130 Disk Monitor System Version 2, Programming and Operator's Guide</u>, Form C26-3717
<u>IBM 1130 Disk Monitor Programming System, Version 2, Program Logic Manual</u>, Form Y26-3714

RESTRICTED DISTRIBUTION: This publication is intended primarily for use by IBM personnel involved in program design and maintenance. It may not be made available to others without the approval of local IBM management.

First Edition (July 1968)

Significant changes or additions to the specifications contained in this publication are continually being made. When using this publication in connection with the operation of IBM equipment, check the latest SRL Newsletter for revisions or contact the local IBM branch office.

This publication was prepared for production using an IBM computer to update the text and to control the page and line format. Page impressions for photo-offset printing were obtained from an IBM 1403 Printer using a special print chain.

Copies of this and other IBM publications can be obtained through IBM branch offices.

A form for reader's comments appears at the back of this publication. Address any additional comments concerning the contents of this publication to: IBM Corporation, Programming Publications, Department 637, Kingston, New York 12401

CONTENTS

INTRODUCTION 7
General Characteristics of the GSP 7
GSP/Disk Monitor System Interface 7
Format of Subroutine Descriptions 10
Operation Diagram Conventions 10

THEORY OF OPERATION 11
Linkage Mechanisms 11
Communications Areas 11
System Display 12
Initialization and Termination 13
Image Management 15
Image Generation 15
I/O Operations of the 2250 15
Light Pen Facilities 17
Locating Positions 17
Light Pen Tracking 17
Alphameric Keyboard and Text Generation Subroutines 21
Attention Handling 21
Attention Enabling and Information Retrieval 22
Attention Processing 23
Light Pen and Order Controlled Attentions 24
Programmed Function Keyboard Attentions 24
Alphameric Keyboard Attentions 24
Error Handling 24

GSP SUBROUTINES 27
Initialization and Termination Subroutines 27
GSPIN -- Graphic Subroutine Package Initialization 27
ICAIN -- Image Construction Area Initialization 28
GCAIN -- Generation Control Area Initialization 28
SSCAL -- Set Scaling Information 29
SSCIS -- Set Scissoring Option 30
SINDX -- Set Index Values 30
SINCR -- Set Increment Values 30
SDATM -- Set Input Data Mode 30
SGRAM -- Set Output Graphic Mode 31
GSPTM -- Graphic Subroutine Package Termination 31
Image Management Subroutines 31
BELMT -- Begin Element 32
EELMT -- End Element 32
DELMT -- Delete Element 33
UELMT -- Update Element 34
XELMT -- Extend Element 34
IELMT -- Include Element 35
BXGEN -- Begin External Generation 35
EXGEN -- End External Generation 35
SATRB -- Set Controlled Entity Attributes 36
STPVS -- Step Verification Setting 36
IDPOS -- Indicate Element Position 36
RDPOS -- Read Position 37
GSP04 -- Push Down 37
GSP05 -- Push Up 38
Image Generation Subroutines 38
MVPOS -- Move Element to a Position 38
PLINE/PPNT/PSGMT -- Plot Lines, Plot Points, Plot Line Segments . 39
PTEXT -- Plot Text 39
LKSUB -- Linkage to a Subroutine 40
PGRID -- Plot Grid Outline 41
PCOPY -- Plot Copy of an Element 41
GSP02 -- Generation Support 41
GSP06 -- Scissor 43

2250 I/O Subroutines 45
EXEC -- Execute Display 45
TMDSP -- Terminate Display 46
SPFKL -- Set Programmed Function Keyboard Lights 46
DSPYN -- 2250 I/O Subroutine 47
Attention Enabling and Information Retrieval 51
SATNS -- Set Attention Status 51
RQATN -- Request Attention Information 51
ROCOR -- Return Outer Correlation Value 53
Light Pen Subroutines 53
LOCPN/LOCND -- Locate Position of the Light Pen and Locate Position of the Light Pen on No Detect 53
LCPOS -- Locate a Position with the Tracking Symbol 54
TRACK -- Track Position of Light Pen 54
CTLTK -- Control Light Pen Tracking 54
CVTTD -- Convert Tracking Data 56
Alphameric and Text Generation Subroutines 56
GCHAR -- Character Strokes 57
GUPER -- Upper Case Character Strokes 57
DFMSG -- Define Message Entity 57
MSGIN -- Message Entity Initialization 58
ICURS -- Insert Cursor 59
RCURS -- Remove Cursor 59
TLMSG -- Translate Message Entity 59
BCNV -- Basic Conversion 60
Attention Processing 61
GSP07 -- Cursor Direct Entry 61
GSP08 -- Light Pen and Order Controlled Direct Entry Subroutine . 62
GSP09 -- Scanning Pattern and Tracking Symbol Direct Entry Subroutine 62
GSP11 -- Scan Index-Controlled Entity 63
GSP12 -- Verification Direct Entry 64
Error Handling Subroutines 64
ERRIN -- Error Initialization 64
IERRS -- Identify Errors 65
Internal Service Subroutines 65
GSP01 -- GSP Inner Subroutine 1 65
GSP03 -- GSP Inner Subroutine 3 66
GSP10 -- Floating Point Functions 68

DIAGNOSTIC AIDS 69
PAUSE 0001 72
PAUSE 0002 73
PAUSE 0003 (First Execution) 76
PAUSE 0003 (Second Execution) 77
Graphic Order Reference Table 81

FLOWCHARTS 83

APPENDIX A: CONTROL BLOCK FORMATS 149
GCOM Cross Reference 149
GCOM Format 150
GCA Format 158
ICA Preface Format 160
ECCB Format 162

APPENDIX B: SUBROUTINE CROSS REFERENCE 163

APPENDIX C: GRAPHICS INITIALIZATION PROGRAM (GRIN) 167

INDEX 169

FIGURES

Figure 1. GSP Overview 8
Figure 2. GSP Interface with the DMV2 9
Figure 3. Linkage Mechanisms 12
Figure 4. System Display 13
Figure 5. Intialization 14
Figure 6. Image Management 16
Figure 7. Image Generation 17
Figure 8. GSP I/O Operations 18
Figure 9. Locating Positions with the Light Pen 19
Figure 10. Light Pen Tracking 20
Figure 11. Alphameric and Text Generation Subroutines 21
Figure 12. Attention Enabling and Information Retrieval 22
Figure 13. Attention Processing 23
Figure 14. Error Handling 25
Figure 15. Error Return Linkage 26
Figure 16. Scissoring to Screen or Grid Boundaries 44
Figure 17. Cancel Frame 49
Figure 18. Verification Subroutine Usage 64
Figure 19. 1130/2250 Main Storage Map 70
Figure 20. Sample Program 71
Figure 21. PAUSE 0001 72
Figure 22. PAUSE 0002 74
Figure 23. PAUSE 0003 79
Figure 24. PAUSE 0003 (second execution) 80
Figure 25. SELECT OPTION Frame 167
Figure 26. EXECUTE 1130 PROGRAM Frame 167

TABLES

Table 1. I/O Operations, Function Code, and Entry Points 47
Table 2. Attention Data Array 50
Table 3. User's Attention Array for RQATN 52
Table 4. Alphameric Data Format 60
Table 5. Graphic Order Reference Table 81
Table 6. Cross Reference Table for GCOM Mnemonics 149
Table 7. GSP Communication Area Format (GCOM) 150
Table 8. Generation Control Area Format (GCA) 158
Table 9. ICA Preface Format .160
Table 10. Element Correlation Control Block Format 162
Table 11. Subroutine Cross Reference Table 164

CHARTS

Chart AA. GSPIN - GSP Initialization 84
Chart AB. ICAIN - ICA Initialization 85
Chart AC. GCAIN - GCA Initialization 86
Chart AD. SSCAL - Set Scaling Values 87
Chart AE. SSCIS - Set Scissoring Option 88
SINDX - Set Index Values 88
SINCR - Set Increment Values 88
SGRAM - Set Output Graphic Mode 88
SDATM - Set Input Data Mode 88
Chart AF. GSPTM - GSP Termination 89

Chart BA. BELMT - Begin Element 90
Chart BB. EELMT - End Element 91
Chart BC. DELMT - Delete Element 92
Chart BD. UELMT - Update Element 93
Chart BE. UELMT - Update Element 94
Chart BF. XELMT - Extend Element 95
Chart BG. IELMT - Include Element 96
Chart BH. BXGEN - Begin External Generation 97
Chart BJ. EXGEN - End External Generation 98
Chart BK. SATRB - Set Controlled Entity Attributes 99
Chart BL. STPVS - Step Verification Setting 100
Chart BM. GSP04 - Push Down .101
Chart BN. GSP05 - Push Up 102
Chart CA. MVPOS - Move Element to a Position 103
Chart CB. PLINE/PPNT/PSGMT - Plot Lines, Points, Segments 104
Chart CC. PTEXT - Plot Text .105
Chart CD. LKSUB - Linkage to a Subroutine 106
Chart CE. LKSUB - Linkage to a Subroutine 107
Chart CF. PGRID - Plot Grid Outline 108
Chart CG. PCOPY - Plot Copy of an Element 109
Chart CH. GSP02 - Generation Support 110
Chart CJ. GSP02 - Generation Support 111
Chart CK. GSP02 - Generation Support 112
Chart CL. GSP02 - Generation Support 113
Chart CM. GSP02 - Generation Support 114
Chart CN. GSP02 - Generation Support 115
Chart CO. GSP06 - Scissor Subroutine 116
Chart DA. EXEC - Execute Display 117
Chart DB. TMDSP - Terminate Display 118
Chart DC. SPFKL - Set Programmed Function Keyboard Lights 119
Chart DD. DSPYN - 2250 I/O Subroutine 120
Chart DE. DSPYN - 2250 I/O Subroutine 121
Chart DF. DSPYN - 2250 I/O Subroutine 122
Chart DG. DSPYN - 2250 I/O Subroutine 123
Chart DH. DSPYN - 2250 I/O Subroutine 124
Chart DJ. DSPYN - 2250 I/O Subroutine 125
Chart EA. SATNS - Set Attention Status 126
Chart EB. RQATN - Request Attention Information 127
Chart EC. ROCOR - Return Outer Correlation Value 128
Chart FA. LOCPN - Locate Position of the Light Pen 129
LOCND - Locate Position of the Light Pen with
No Detect 129
Chart FB. LCPOS - Locate a Position with the Tracking Symbol 130
Chart FC. TRACK - Track Posision of Light Pen 131
Chart FD. CTLTK - Control Light Pen Tracking 132
Chart FE. CTLTK - Control Light Pen Tracking 133
Chart FF. CVTTD - Convert Tracking Data 134
Chart GA. DFMSG - Define Message Entity 135
Chart GB. MSGIN - Message Entity Initialization 136
Chart GC. ICURS - Insert Cursor 137
Chart GD. RCURS - Remove Cursor 138
Chart GE. TLMSG - Translate Message Entity 139
Chart GF. TLMSG - Translate Message Entity 140
Chart GG. BCNV - Basic Conversion Subroutine 141
Chart HA. GSP07 - Cursor Direct Entry Subroutine 142
Chart HB. GSP08 - Light Pen and Order Controlled Direct Entry
Subroutine 143
Chart HC. GSP09 - Scanning Pattern and Tracking Symbol Direct
Entry Subroutine 144
Chart HD. GSP11 - Scan Index-Controlled Entity 145
Chart JA. IERRS - Identify Errors 146
Chart KA. GSP01 - GSP Inner Subroutine 1 147
Chart KB. GSP03 - GSP Inner Subroutine 3 148
Chart KC. GSP03 - GSP Inner Subroutine 3 149

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 CALL 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. In addition to loading these subroutines, the *G2250 card causes certain fields in COMMA to be initialized for use by the GSP initialization subroutine.

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 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

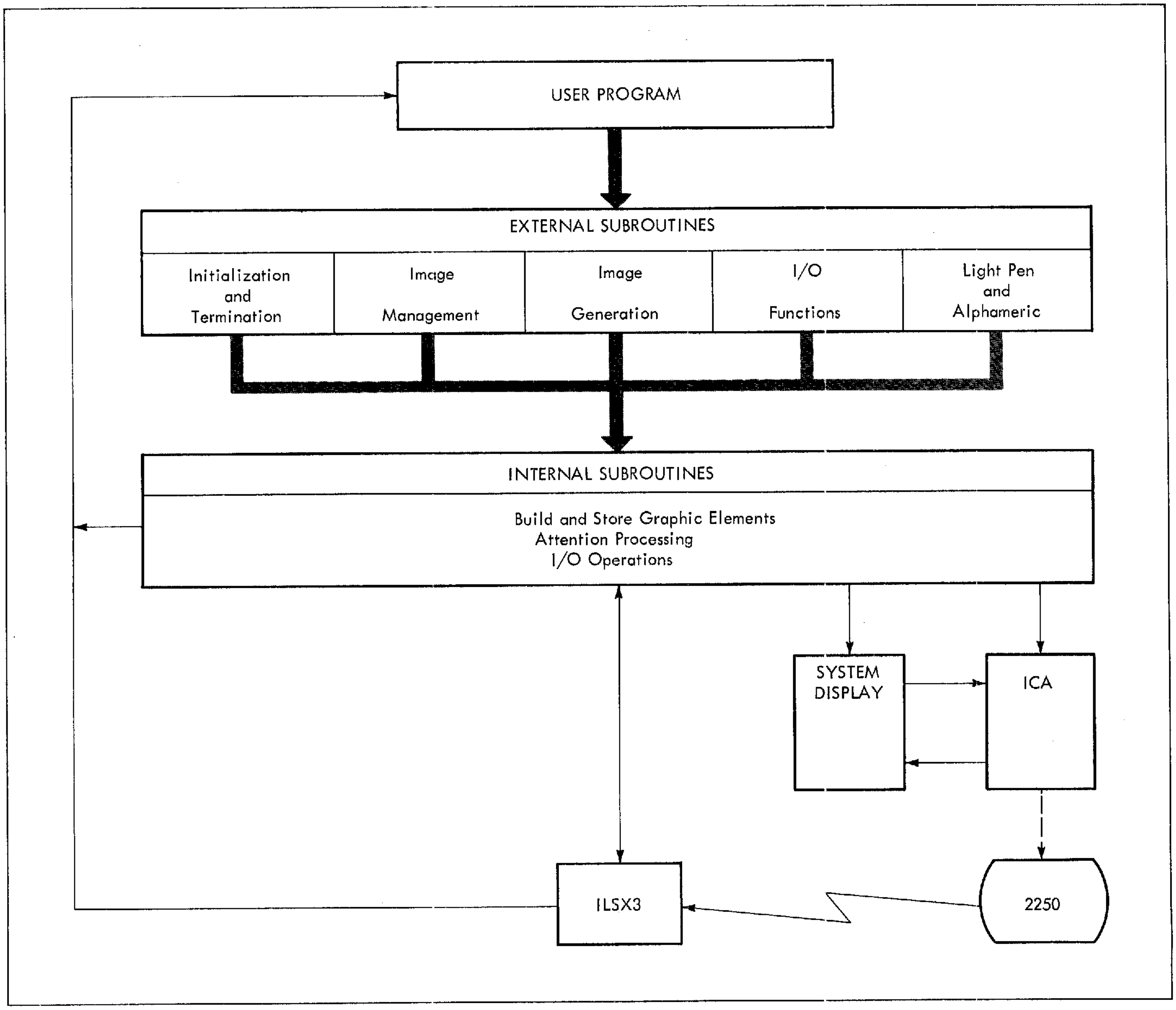


Figure 1. GSP Overview

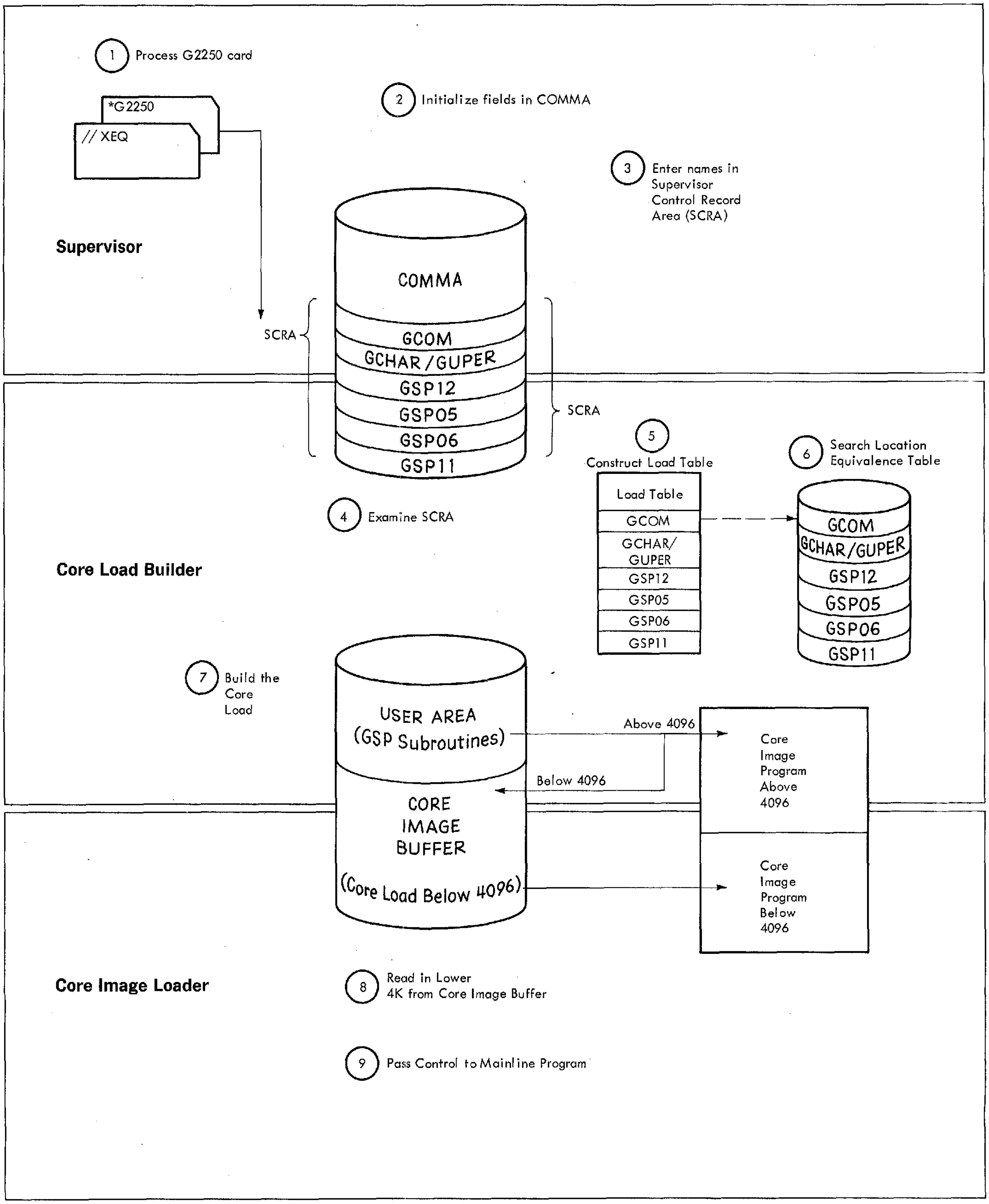


Figure 2. GSP Interface with the DMV2

the core load is being built, the Core Load Builder examines each subroutine to (1) ensure that any subroutines which are referenced by a CALL or LIBF statement are incorporated into the core load and (2) reserve areas within the core load for the largest LOCAL and/or SOCAL.

When the last word of the in-core load has been converted and stored, control passes to the Core Image Loader. The Core Image Loader reads the contents of the CIB into the lower 4K of main storage (8). The core load now overlays the Core Load Builder which has completed its processing. The Core Image Loader passes control to the mainline program to begin execution.

FORMAT OF SUBROUTINE DESCRIPTIONS

Detailed descriptions of all GSP subroutines are presented in this publication in the following format:

MNEMONIC -- Name is an abbreviated symbol and the full name of the subroutine.

Chart is the two-letter symbol assigned to that subroutine's flowchart(s).

Function describes the services provided by the subroutine.

Entry is the mnemonic designating the point at which the subroutine is entered. The type of referencing statement used is found in Appendix B.

Exit is the manner in which and the location to which exit is made.

Input is the information in the calling sequence passed by the calling program to the subroutine.

Output is the result of the subroutine's processing.

Operation is the general flow of internal logic from the entry point of the subroutine, and describes the manner in which processing is performed.

Errors is a description of possible error conditions within the subroutine and the resultant error codes.

OPERATION DIAGRAM CONVENTIONS

The figures used in this publication use the following conventions:

➡ = flow of control

░░▶ = flow of control in display program

——▶ = data movement

– – –▶ = data reference

↯ = an attention (2250 interrupt)

This section describes the communications areas and the general flow of logic and data in the following functional areas of the GSP:

- Initialization and termination
- Image Management
- Image Generation
- Alphameric and Text Generation
- Light Pen
- I/O Operations of the 2250
- Attention Handling
- Error Handling

The figures associated with the text are supplementary and show only the principal paths for the function. A description in greater detail is given in the respective sections of "GSP SUBROUTINES."

LINKAGE MECHANISMS

The GSP uses several means for establishing linkage between the mainline program and the GSP subroutines, and among the subroutines themselves. CALL or LIBF instructions are used to access the GSP subroutines depending upon the type and subtype of the subroutine being accessed (see Appendix B). Also, a Branch and Store IAR Indirect instruction may be used to access a subroutine whose address is stored in a known location (this is used in a manner similar to a LIBF instruction). Although the Assembler-language programmer can use both CALL and LIBF statements, thus making all GSP subroutines available, the FORTRAN programmer can access only those subroutines referenced by a CALL statement. The subroutines not accessed by CALL statements are internal service subroutines that perform common functions for other GSP subroutines.

The GSP subroutine GSP01 (see Figure 3) performs several functions associated with control transfer. The save function stores index registers, the address of the next sequential instruction in the calling program, and any arguments in the calling sequence in unique fields in GCOM. These arguments may be referenced by the called subroutine for processing. The return function of GSP01 restores the index registers to the status existing prior to entering the called subroutine, and causes a branch to the next sequential instruction in the calling program.

The error function of GSP01 is discussed under "Error Handling".

Return to a calling program may also be made by branching indirectly through the operand field of the entry word of the called subroutine. The address of the next sequential instruction in the calling subroutine is stored in this operand field by a CALL statement. Similarly, a LIBF statement stores in the third word of the called subroutine the address of the next word after the LIBF statement in the calling subroutine. This also may be used to compute the next sequential instruction in the calling subroutine if there are parameters immediately following the LIBF statement.

COMMUNICATIONS AREAS

The GSP uses four communications areas (GCOM, GCA, ICA preface and ECCB) to assist the subroutines in processing data. The format of each is shown in Appendix A.

The GSP Communication Area (GCOM) is loaded as a result of the *G2250 Monitor Supervisor Control Record, and initialized by GSPIN (GSP Initialization). GCOM is a communication area used by many of the GSP subroutines as pointers, indicators, and constants. Included in GCOM are the graphic orders to display a processing message and a scanning pattern, and a series of graphic orders comprising the system display (see below). Several other areas are of particular importance. These include the attention handling fields, the image generation fields, and the fields used by the internal subroutines GSP02 and GSP06.

The Generation Control Area (GCA) is defined by the FORTRAN programmer. It contains values and options used by the GSP image generation subroutines for scaling, scissoring, converting, and indexing data. There may be more than one GCA, but each must be initialized and fully defined before being used.

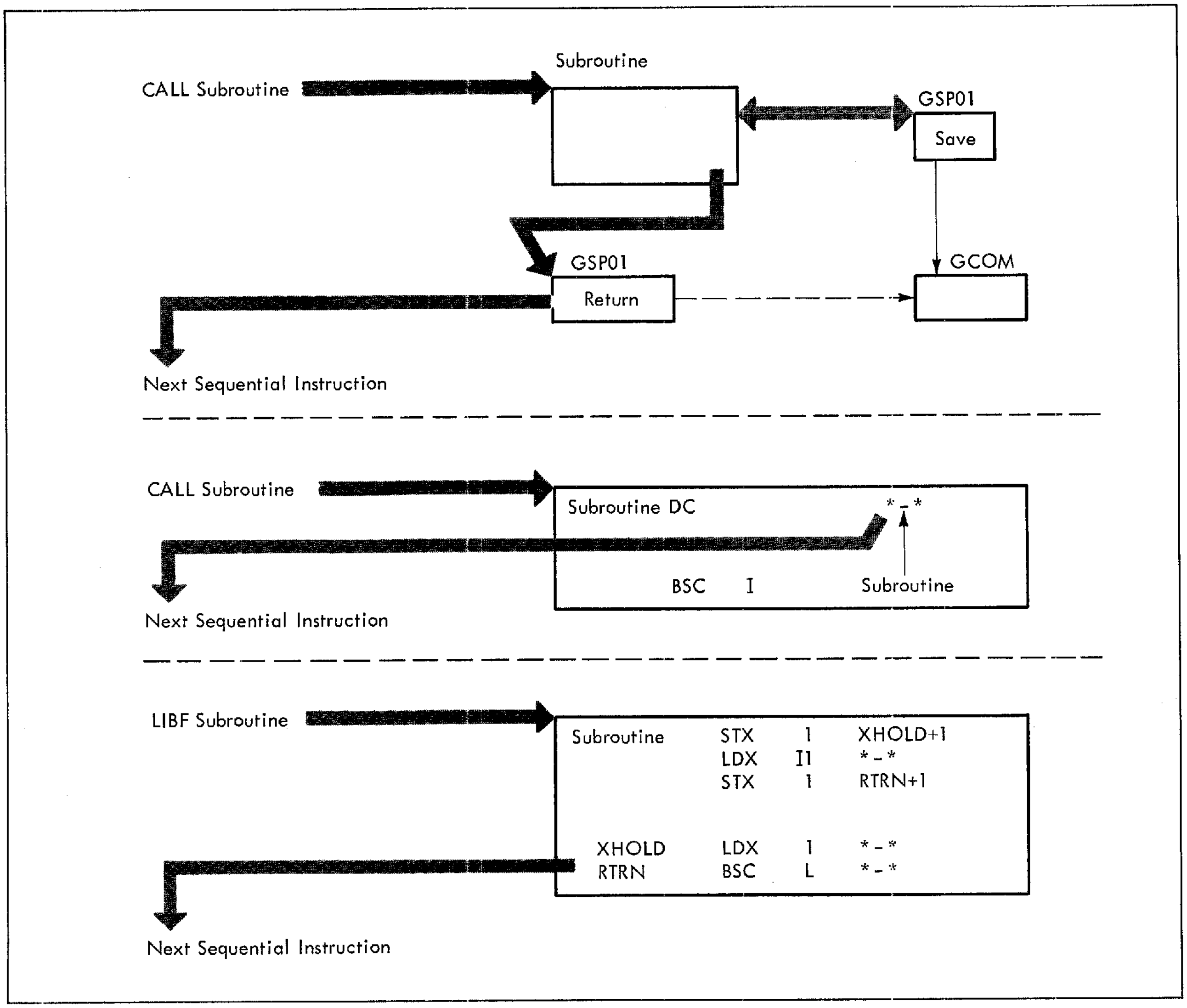
The Image Construction Area (ICA) preface is a twenty-four word communications area created in the lower core positions of an ICA by the subroutine ICAIN. The preface contains pointers, counters, and indicators used by the image management subroutines in creating and modifying graphic elements within the ICA.

An Element Correlation Control Block (ECCB) is a four-word area in the higher core positions of the ICA. It contains pointers to and indicators for a specified named graphic element within the ICA.

SYSTEM DISPLAY

The system display, which resides in GCOM, is the "mainline" order program of the GSP. Any user- or GSP-defined set of graphic orders is executed by connecting them to the system display. When an order program is connected to the system display, it becomes incorporated into the regeneration cycle of the system display until some action is taken to disconnect it. The system display, through the Start Timer order, also permits keyboard interrupts and initiates testing of the light pen switch. Figure 4 shows the skeleton coding of the system display.

The first two orders in the system display form a "switch trap", a mechanism used to prohibit the rest of the system display from executing. The switch trap is comprised of a Start Timer order followed by a Graphic Branch Conditional order to the Start Timer order. The branch is executed when the light pen switch is closed if the user requests this service in a call to the EXEC subroutine. As long as the light pen switch remains closed, only the switch trap may execute.


CALL Subroutine
Subroutine
GSP01
Save
GSP01
Return
GCOM
Next Sequential Instruction
CALL Subroutine
Next Sequential Instruction
LIBF Subroutine
Next Sequential Instruction

```
Subroutine DC              *-*

           BSC    I        Subroutine
```

```
Subroutine   STX    1      XHOLD+1
             LDX    I1     *-*
             STX    1      RTRN+1

  XHOLD      LDX    1      *-*
  RTRN       BSC    L      *-*
```

Figure 3. Linkage Mechanisms

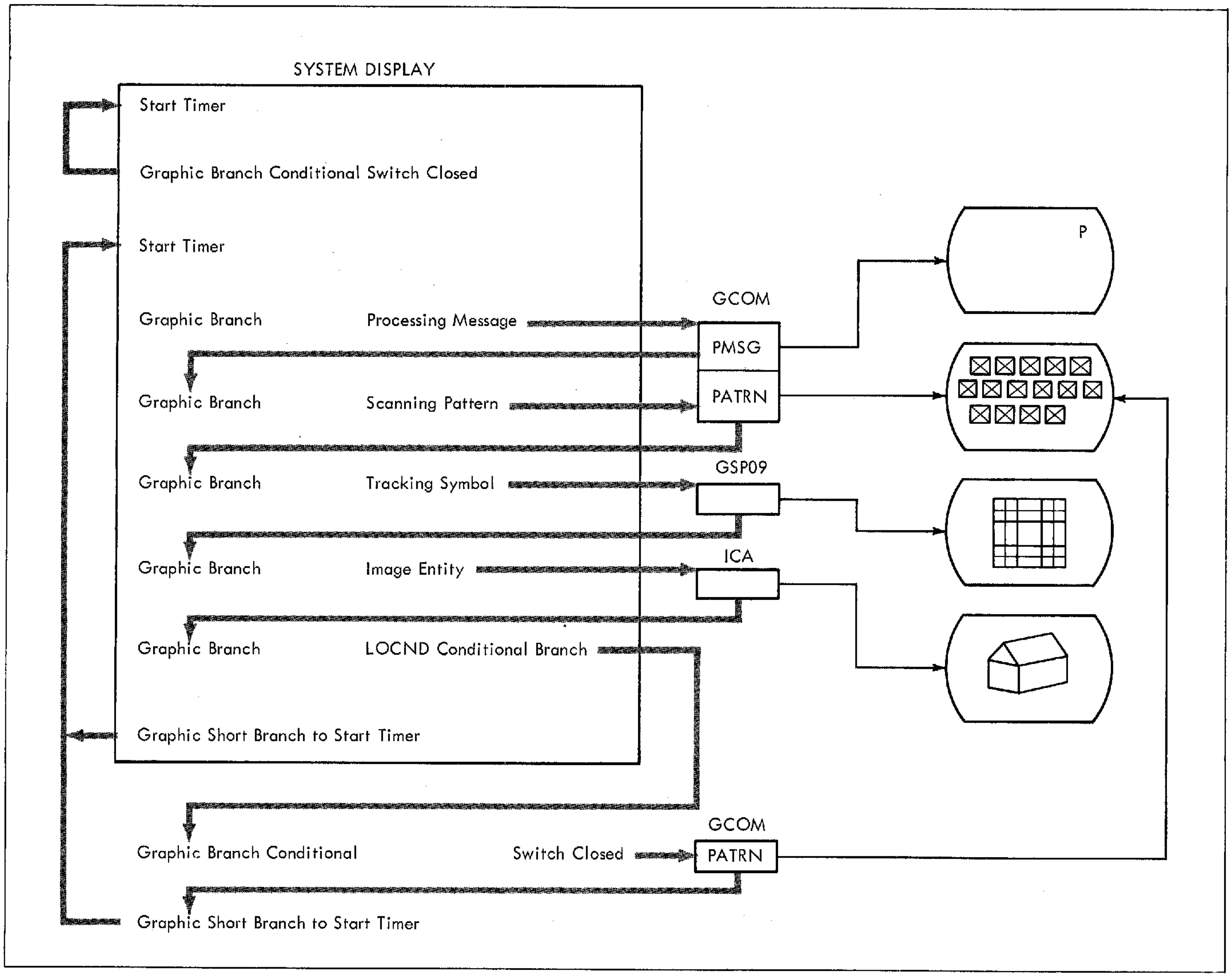


Figure 4. System Display

Following the switch trap is a second Start Timer order, a series of Graphic Branch orders, and a Graphic Short Branch to the second Start Timer order to form a continous loop. It contains no displayable data. To connect a set of graphic orders to the system display, the address of the order program is placed in the operand field of a Graphic Branch order and the no-operation bit in the order is turned off. The order program is disconnected by setting the no-operation bit on.

The system display contains Graphic Branches to four order programs: (1) the processing message, (2) the tracking symbol, (3) the scanning pattern, and (4) the user's image entity. The addresses of the orders for the processing message and the scanning pattern are permanently set at load time in the branch orders. The addresses of the orders for the tracking symbol and the image entity are placed in the orders in response to a CALL statement in the user's program.

The system display is initialized by GSPIN, the GSP initialization subroutine. GSPIN insures that all the Graphic Branch orders are set to perform no-operation. GSPIN then gives control to the 2250 I/O subroutine, DSPYN, to issue the XIO command to start regeneration of the system display. Regeneration of the system display is stopped and restarted when attentions occur.

INITIALIZATION AND TERMINATION

The purpose of the initialization subroutines is to prepare the GSP for creating, displaying, and modifying graphic elements.

To prepare the GSP for processing, the initialization subroutines set the communications areas to standard values, or modify them to non-standard values for specific functions.

The first subroutine called by the program must be GSPIN (see Figure 5) which establishes the relative addressing for the GSP Communication Area (GCOM) using the Transfer Vector Table and COMMA. GCOM is then set to standard generation control values and precision specifications, and error return variables are set. GSPIN uses GSP03 to initialize the generation control fields in GCOM, and DSPYN to start regeneration of the system display before returning to the mainline program.

The Image Construction Area (ICA) is initialized by ICAIN. This establishes an

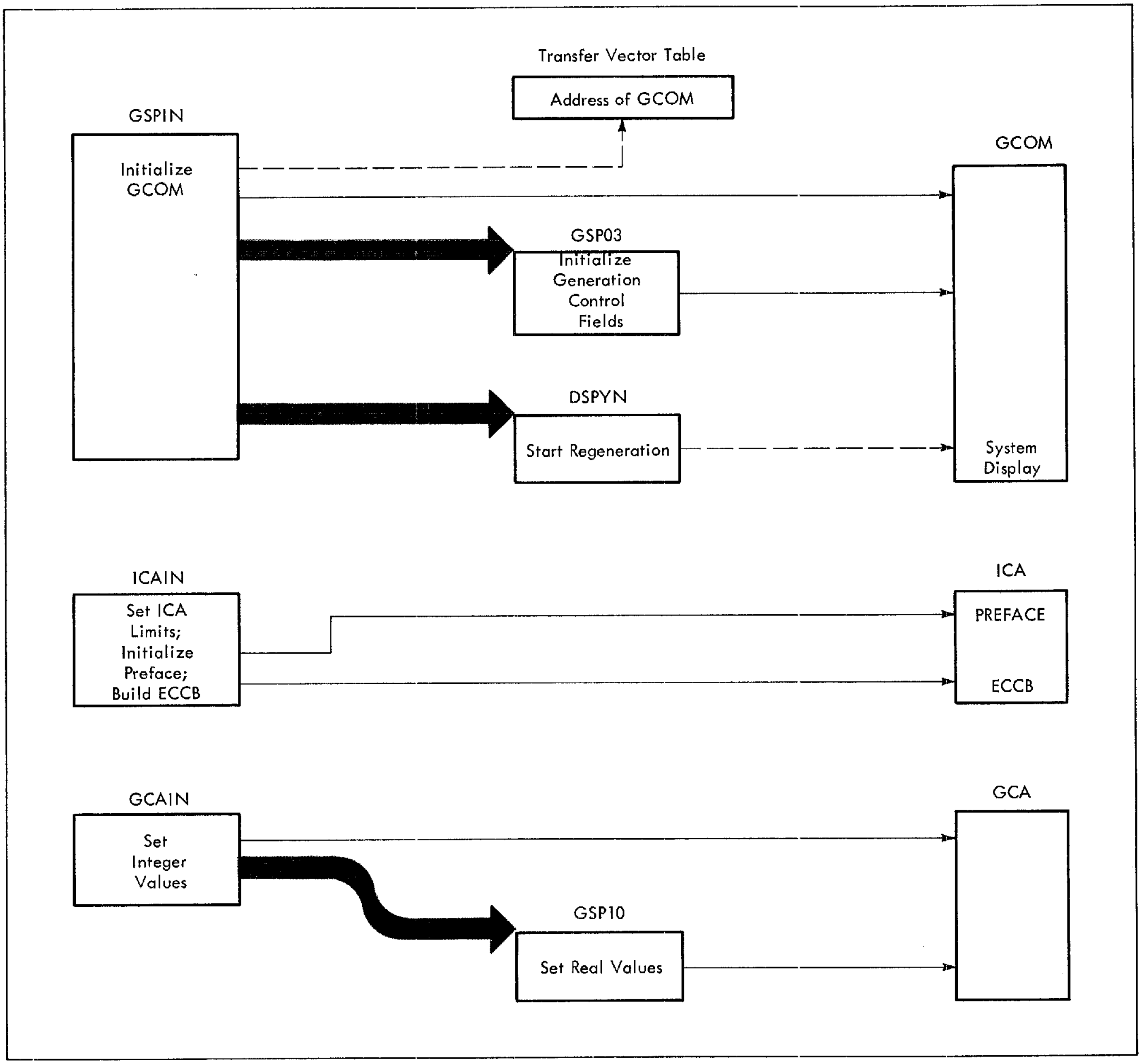


Figure 5. Intialization

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 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.

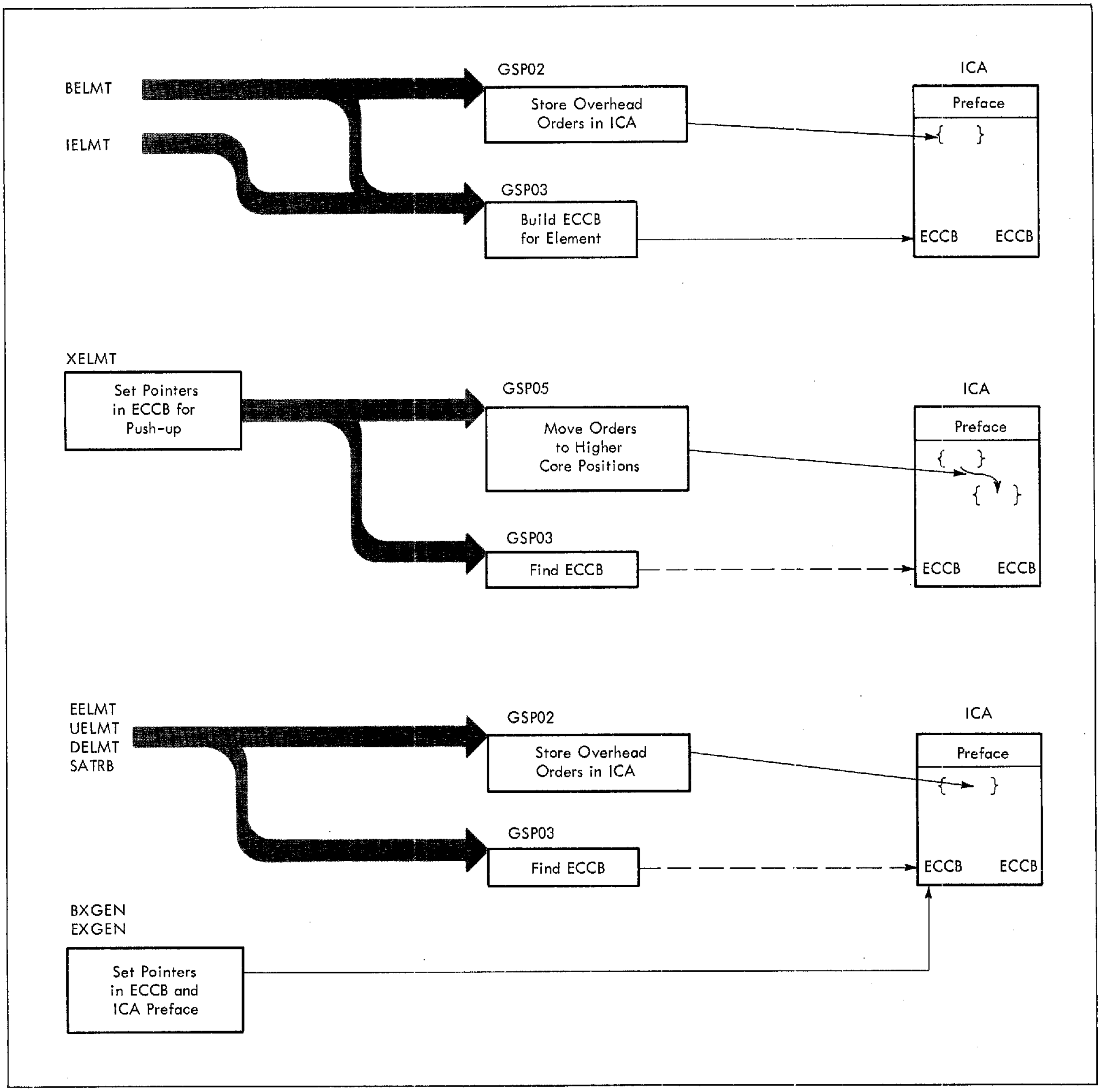


Figure 6. Image Management

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

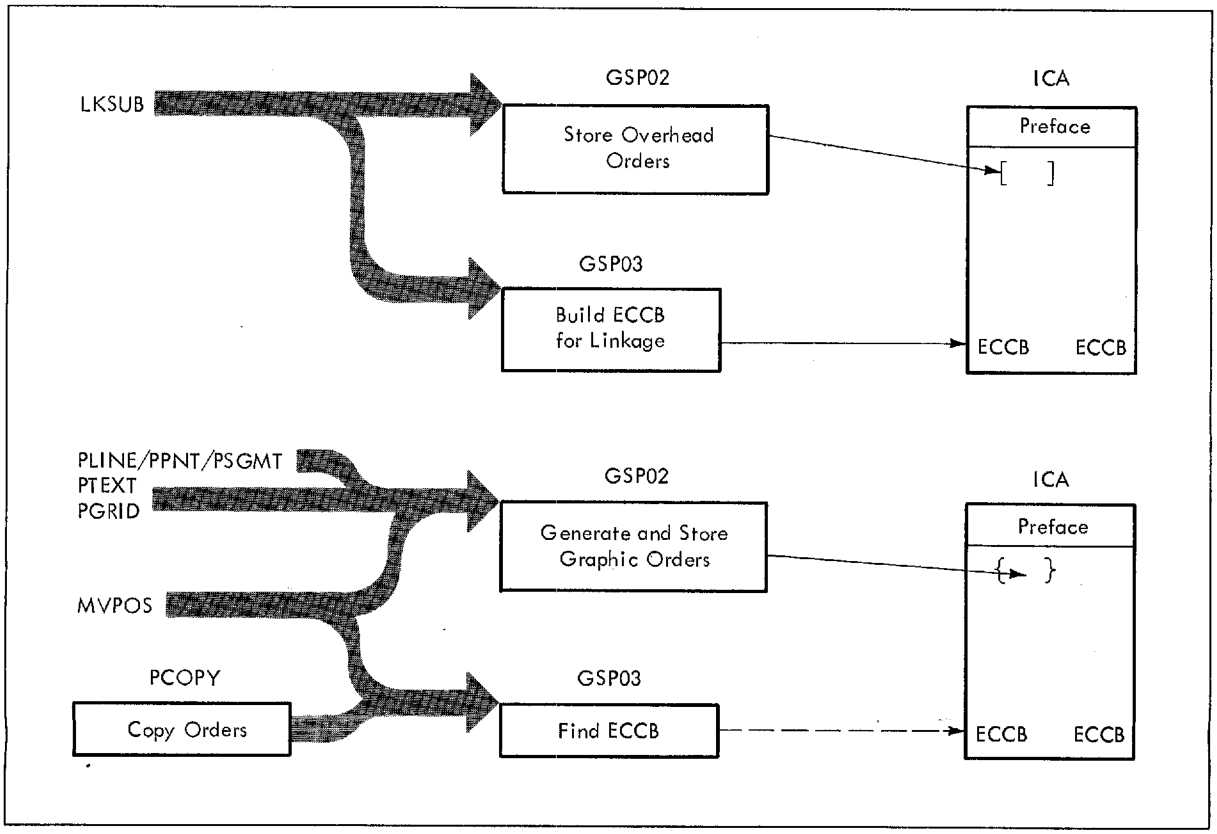


Figure 7. Image Generation

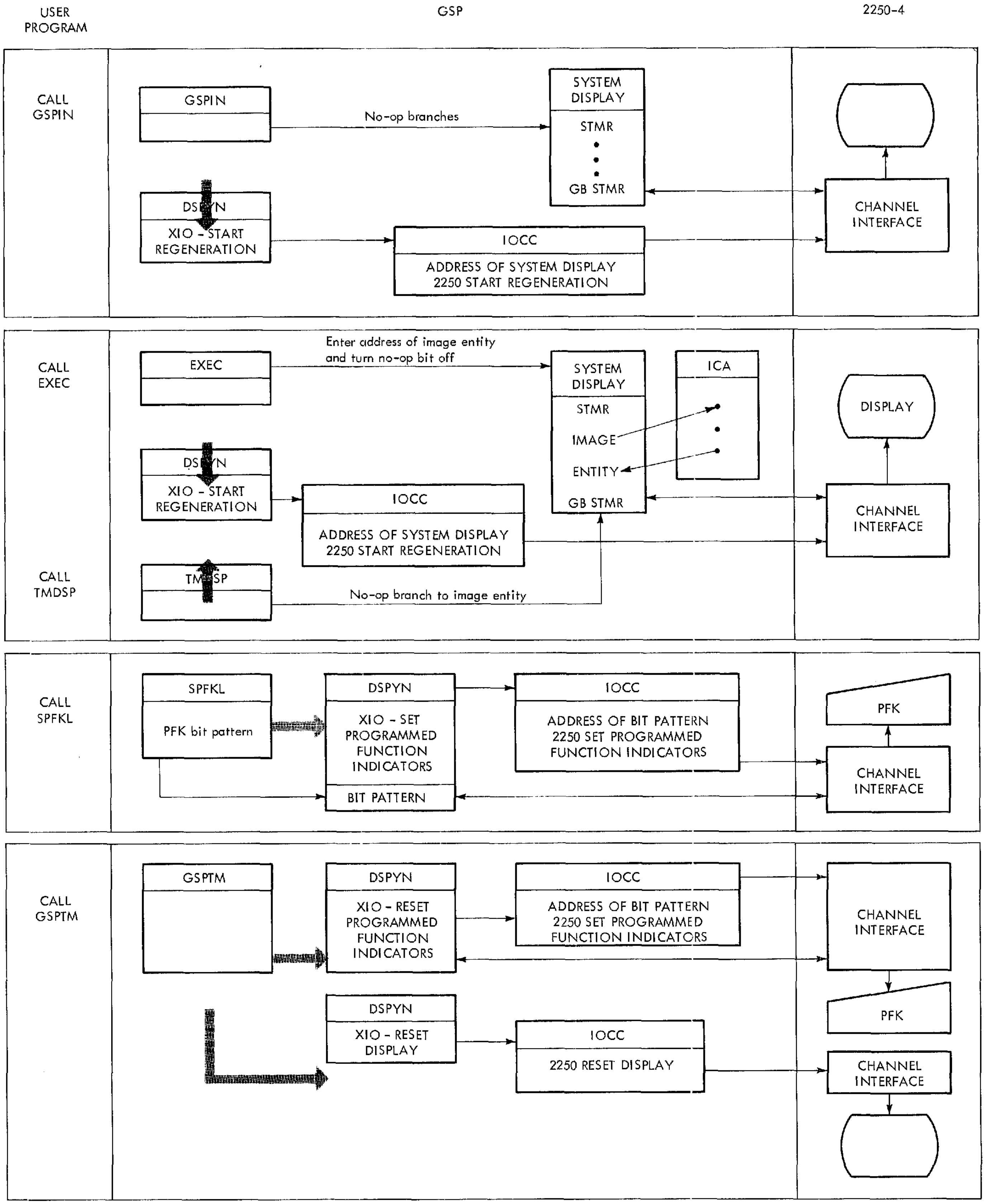


Figure 8. GSP I/O Operations

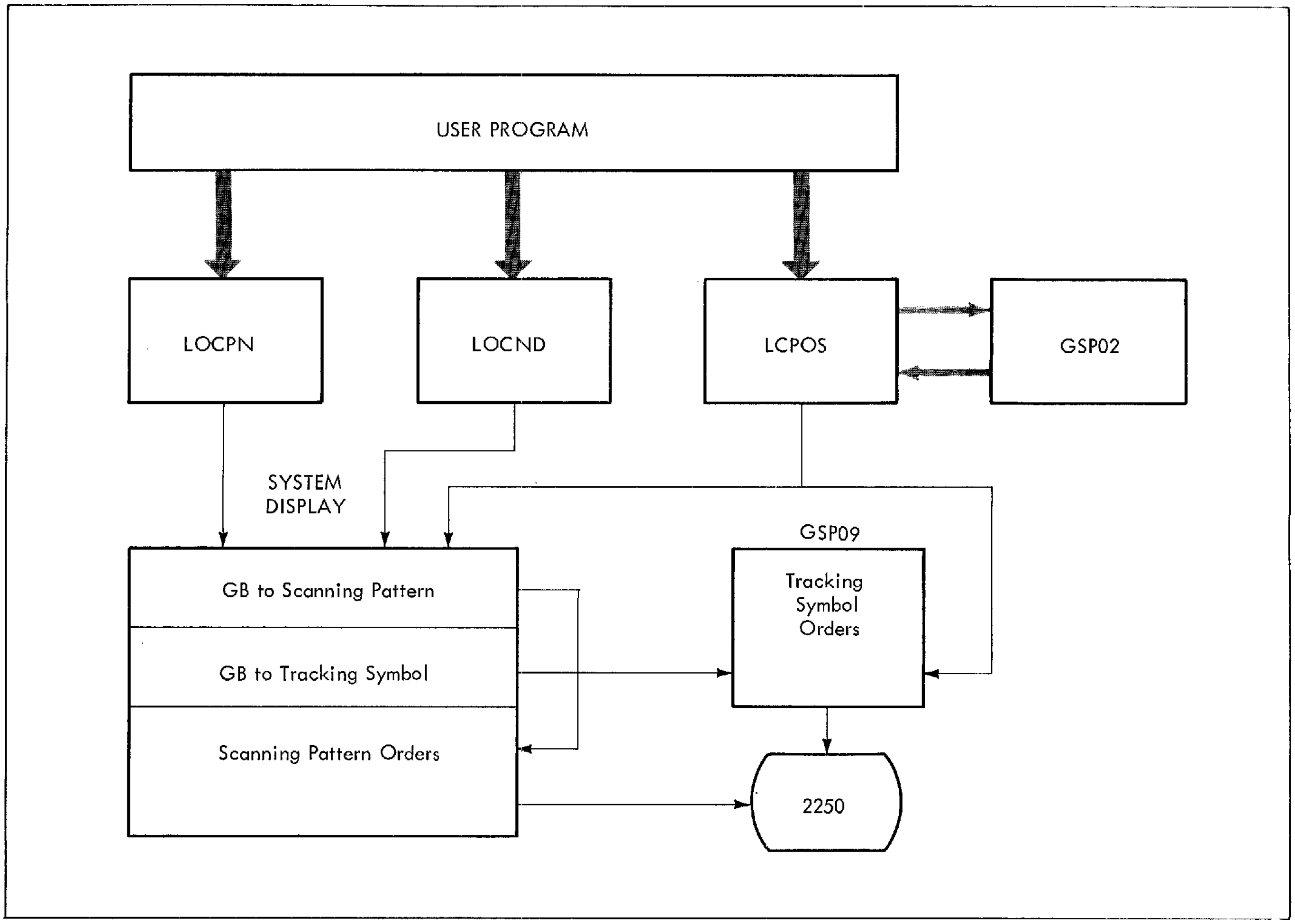


Figure 9. Locating Positions with the Light Pen

which draw the tracking symbol are in GSP09. TRACK activates the branch in the system display to the tracking symbol orders causing the symbol to appear on the screen.

To establish a tracking entity in which graphic orders may be stored, TRACK calls GSP04 to provide room in the ICA for the tracking entity. GSP03 is then called to build the ECCB for the tracking entity. TRACK also establishes several tracking default options (points, lines, axis restrictions, etc.) which are stored in GCOM for use by GSP09. For the user to designate his own drawing techniques and other options, a call is made to CTLTK. This subroutine defines the tracking characteristics the user wishes to apply to his program.

Once the tracking environment has been established and the characteristics defined, the user may begin to generate graphic orders through the tracking symbol. Each attention on the tracking symbol causes control to pass through ILSX3, DSPYN, and GSP08 to GSP09. Here the tracking symbol is centered on the light pen. An algorithm is used to determine where the light pen is positioned on the tracking symbol and to what coordinates the symbol must be moved to reposition its center on the light pen. These coordinates are also used in building graphic orders which are stored in the tracking entity. As the tracking symbol is being moved about the screen, GSP09 is simultaneously building a tracking entity.

Since light pen tracking is produced by a series of rapid attentions on the tracking symbol, and each attention causes the regeneration of the system display to be halted, DSPYN issues an XIO to start regeneration of the system display each time control is passed back to the interrupted program.

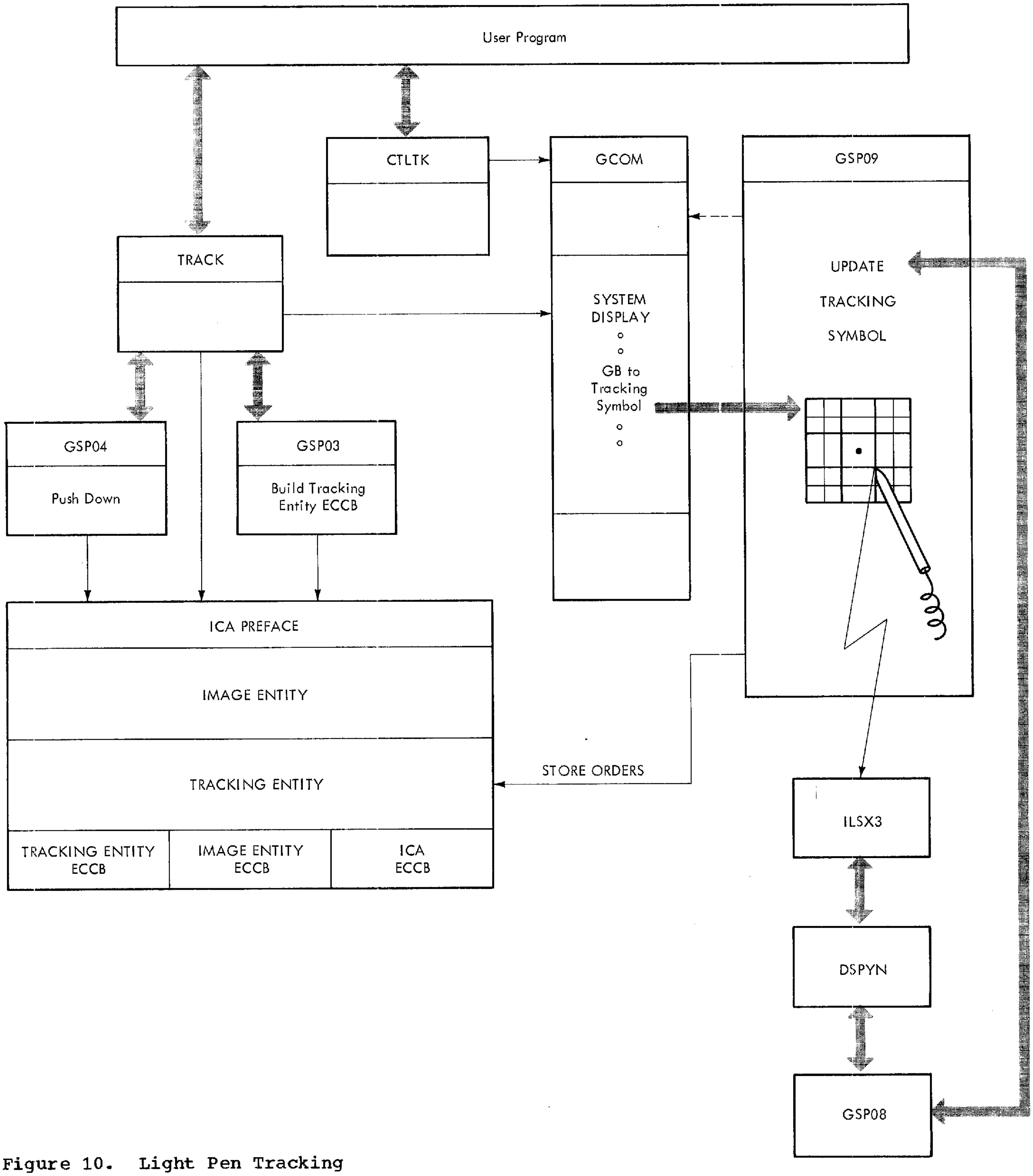


Figure 10. Light Pen Tracking

ALPHAMERIC KEYBOARD AND TEXT GENERATION SUBROUTINES

Alphameric data may be entered into graphic elements from the alphameric keyboard (see "Attention Processing") or from data in the problem program. To enter data from the alphameric keyboard, a message entity must be defined by DFMSG (see Figure 11) and message collection mode must be established by a call to ICURS, which stores in the message entity a Graphic Short Branch (GSB) order to the cursor subroutine (in GCHAR), and stores the address of the Cursor Direct Entry subroutine, GSP07, in the ANDE field in GCOM. Message collection mode is terminated by RCURS, which replaces the GSB to the cursor with a GSB to the cursor null, and removes the address of GSP07 from GCOM.

To enter data from the problem program, the subroutine DFMSG defines a message entity and the subroutine MSGIN causes the translation of data into graphic short branch orders which are' stored in the specified message entity. Message collection mode need not be established, as data is not being entered from the alphameric keyboard.

Conversely, the subroutine TLMSG translates Graphic Short Branch orders in a message entity into EBCDIC and stores the resultant code in a user defined array.

The subroutine BCNV provides the conversion of data between EBCDIC and real, and EBCDIC and integer format.

ATTENTION HANDLING

An attention is an interrupt from the 2250 and is meaningful to the program or to the GSP. Some attentions are processed internally by the GSP. These are (1) CANCEL key, (2) light pen detects on the tracking symbol, and (3) alphameric attentions while in message collection mode.

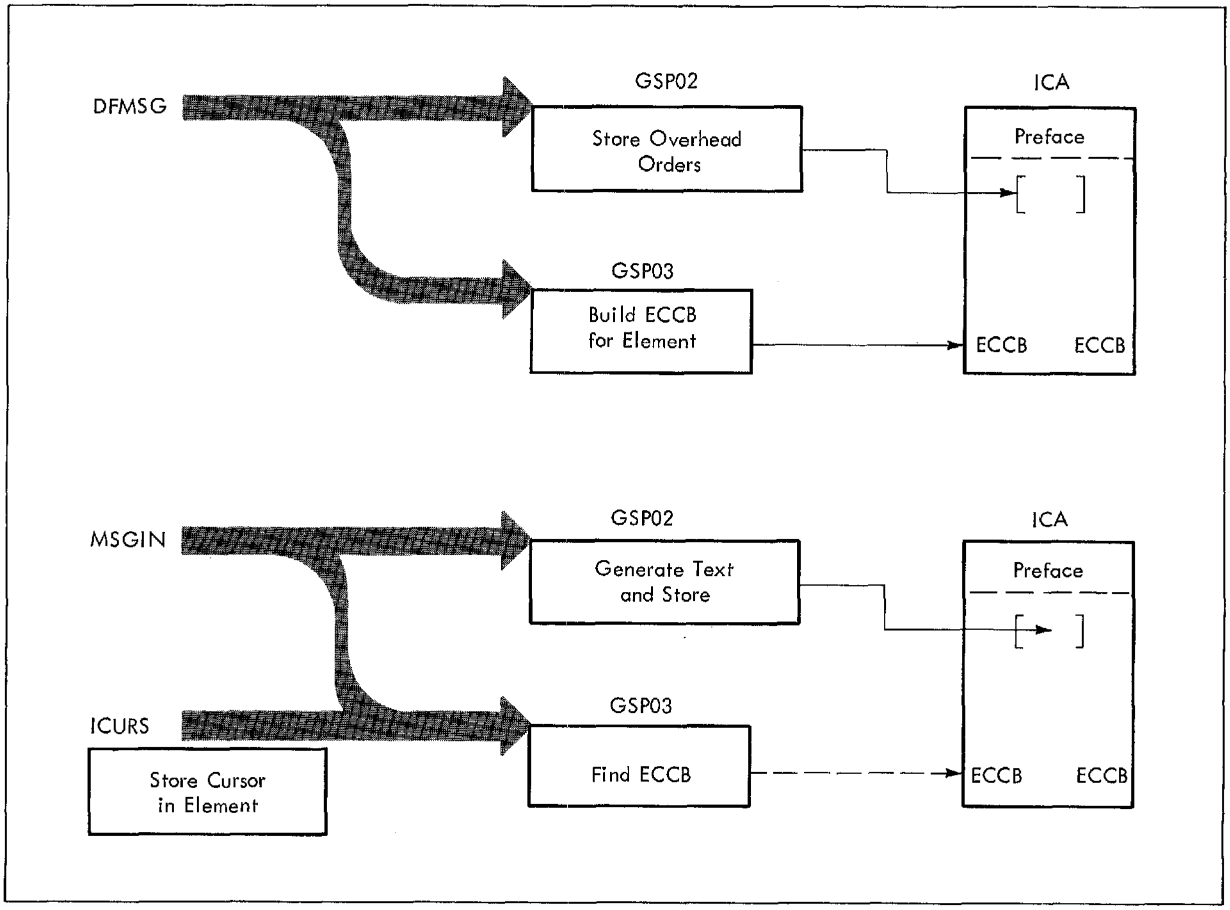


Figure 11. Alphameric and Text Generation Subroutines

For other attentions, the GSP saves information about the attention for further processing by the program, but only if (1) the source of the attention has been designated by the program as enabled, and (2) information for a previous interrupt is not waiting for program processing (attentions are not queued). Sources that can be meaningful to the program are the light pen, the alphameric keyboard, the programmed function keyboard, order controlled (for Assembler-language only) and the END key.

The following paragraphs describe the logic of the GSP subroutines that allow the program to recognize and interpret attentions in the manner prescribed by the program and the GSP.

ATTENTION ENABLING AND INFORMATION RETRIEVAL

If the source of an attention is to be meaningful to the program, the subroutine SATNS (see Figure 12) must enable the source by setting an indicator word in GCOM. This word is examined by the interrupt service subroutine DSPYN when an attention occurs. If the source is enabled, attention data is saved in a storage area in DSPYN and can thus be made available to the program.

The subroutine RQATN is used to transfer attention data from the storage area in DSPYN to a user defined array. This data is then available to the program for further processing. Attention data is not

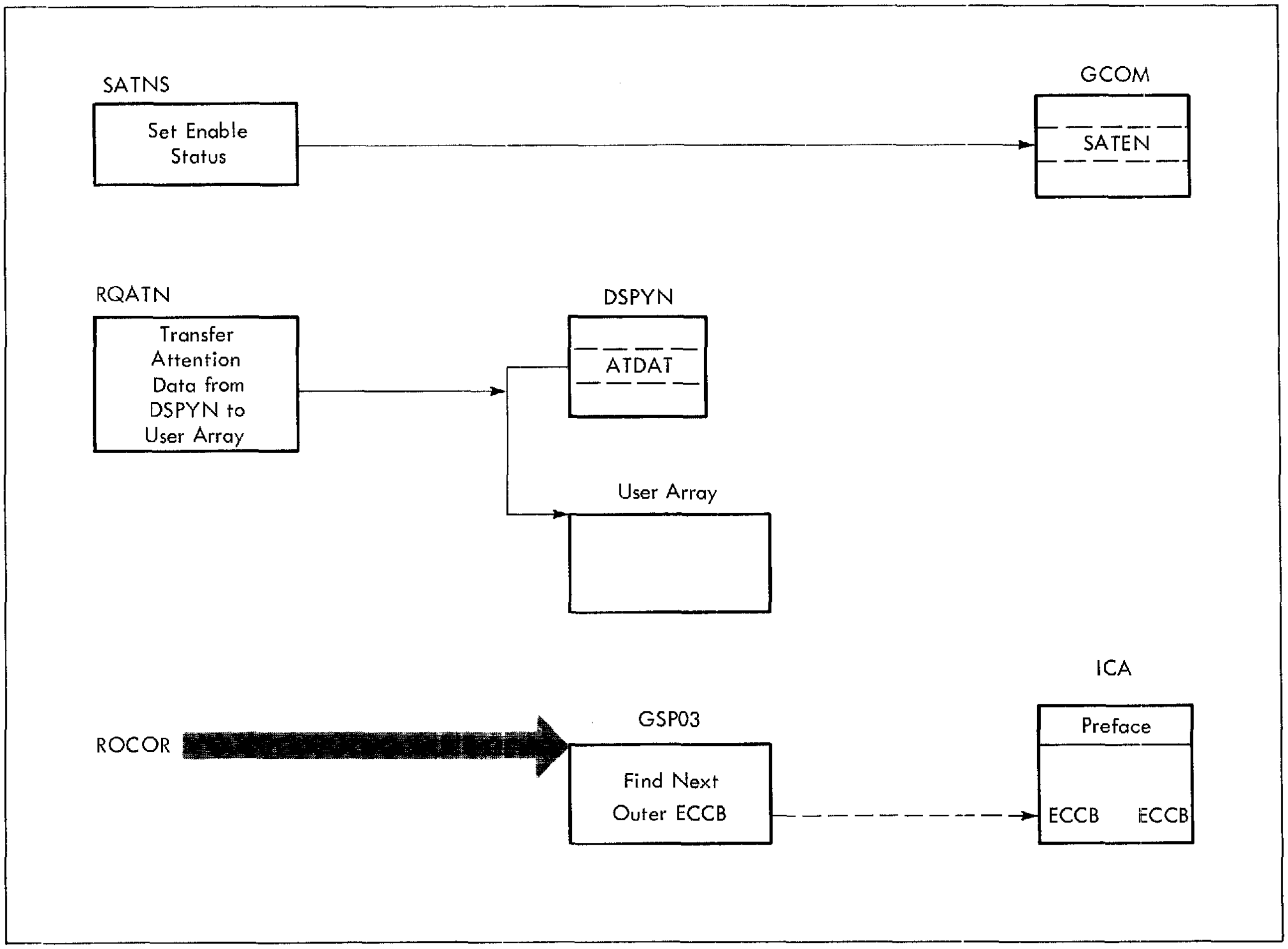


Figure 12. Attention Enabling and Information Retrieval

saved (queued) by the GSP; thus, if attention data exists in the storage area in DSPYN as a result of an enabled interrupt, further interrupts are ignored until RQATN transfers the attention data to the user's array. (The contents and format of the user array are shown with the description of RQATN.)

The subroutine ROCOR can be used to retrieve information not included in the attention array (the correlation value of the element in which an element is nested, or the correlation value of a subroutine linkage).

ATTENTION PROCESSING

The GSP processes attentions originating from the light pen, the programmed function keyboard, the alphameric keyboard, and the order program itself. An attention from any of these sources (see Figure 13) causes control to pass to DSPYN, the GSP Interrupt Service Subroutine via a hardware-generated branch to ILSX3, the 2250 Interrupt Level Subroutine. At the time of an attention, regeneration of the system display automatically ceases.

Each time an attention is generated, DSPYN tests GCOM to determine if a direct entry routine has been specified for the particular source. The direct entry subroutine may be defined by the GSP or by the user through facilities provided in DSPYN.

Note: The GSP provides six Interrupt Level Subroutines (ILSX0, ILSX1, ILSX2, ... ILSX5) which are similar to the Interrupt Level Subroutines ILS00 through ILS05 with the exception that index register 3 is saved and reset with the address of the Transfer Vector Table before ILS processing, and restored to the entry value before returning to the interrupted program.

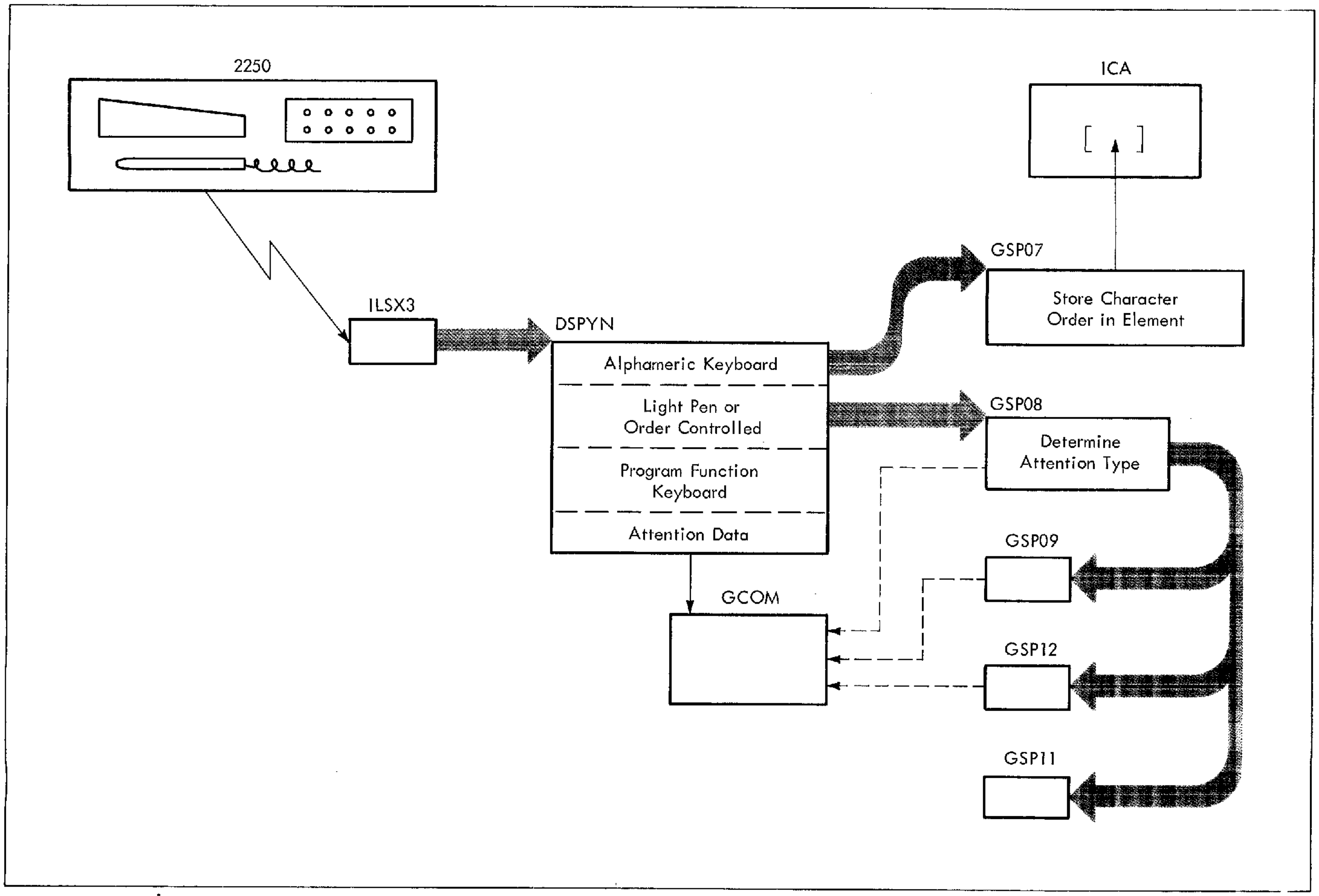


Figure 13. Attention Processing

Light Pen and Order Controlled Attentions

The light pen is only able to cause attentions when the order program has set the light pen mode to enable either detects or detect interrupts. When a detect interrupt occurs, DSPYN passes control to GSP08, the light pen direct entry subroutine. (In the case of an assembler-language program, control may pass to a user-defined direct entry subroutine.) Control returns to the interrupted program through DSPYN if GSP08 determines that the switch is not closed. If the switch is closed, GSP08 then determines if the light pen has been enabled. If so and if the attention is on an index-controlled entity, control is passed to GSP11 to determine the relative position of the detected element within the entity. For either case where the light pen is enabled, the branch in the system display is set to no-operation. A communication word in GCOM is set to inform DSPYN that a valid light pen attention has occurred and control returns to DSPYN. If the light pen is not enabled, control immediately returns to DSPYN. In either case, DSPYN tests to determine if the light pen is enabled in the event a user light pen direct entry subroutine was used in place of GSP08. If light pen attentions were enabled and the attention data area in DSPYN does not contain data which has not yet been accessed by the main processing program, the new attention data is placed in this area. In either case, the system display is restarted and control returns to the interrupted program.

A special case exists when the 2250 operator causes an attention by placing the light pen on the scanning pattern, the tracking symbol, or a level-controlled entity. Such an attention is the result of the pen mode being set to defer interrupts so that a Graphic Interrupt Conditional (GIC) order is executed when the light pen switch is closed. Although an order controlled attention has occurred, in the case of the scanning pattern it is presented to the mainline program as a light pen attention. An attention on the scanning pattern or the tracking symbol causes GSP08 to pass control to GSP09, which is used to discontinue the display of the scanning pattern or update the position of the tracking symbol. When control returns to DSPYN after an attention caused by the scanning pattern the attention data area is filled in without inspection; i.e., previously saved attention data will be overlaid. The system display is restarted and control returns to the interrupted program. An attention caused by the tracking symbol causes the display to be restarted at the next sequential order before returning to the interrupted program. An attention on a level-controlled entity causes GSP08 to pass control to GSP12 which determines the current and next order addresses at which the display will be restarted.

If it is determined that the attention was not on the scanning pattern, tracking symbol, or a level-controlled entity, it is assumed that it was from a user-generated order string. In this case control does not pass to GSP09 but returns to DSPYN where the attention data area is filled. Control returns to the interrupted program without restarting the system display.

Programmed Function Keyboard Attentions

Attentions from the programmed function keyboard (PFKB) are routed through ILSX3 to DSPYN. After any direct entry processing, GCOM is tested to verify that the PFKB has been enabled as a legitimate attention source. If the attention data area is free, the attention data associated with the PFKB attention is placed in the area. The system display is restarted and control returns to the interrupted program through ILSX3.

Alphameric Keyboard Attentions

When the 1130 CPU is interrupted from the alphameric keyboard, DSPYN determines if message-collection mode has been established by a previous call to ICURS. If so, the direct entry subroutine GSP07 performs the necessary processing and stores graphic orders in the message entity which results in the letter corresponding to the depressed key being displayed on the 2250 screen. Attention data is not made available to the program when in message-collection mode.

If message-collection mode is not established, or has been terminated by a call to RCURS, and the alphameric keyboard is enabled, attentions received from the alphameric keyboard are treated similarly to attentions received from the programmed function keyboard, and attention data is stored in the attention area of DSPYN.

ERROR HANDLING

Non-I/O error conditions encountered within the GSP subroutines may be handled in three ways, depending upon the user specifications. As each error condition is encountered, the internal subroutine GSP01 (see Figure 14) stores the code associated with the error in two fields in GCOM, so that at any point in the program the last error and all errors encountered are available to the program. If a user-provided error processing subroutine or the GSP error processing subroutine IERRS has been specified by the subroutine ERRIN, the 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 principal print device.

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.

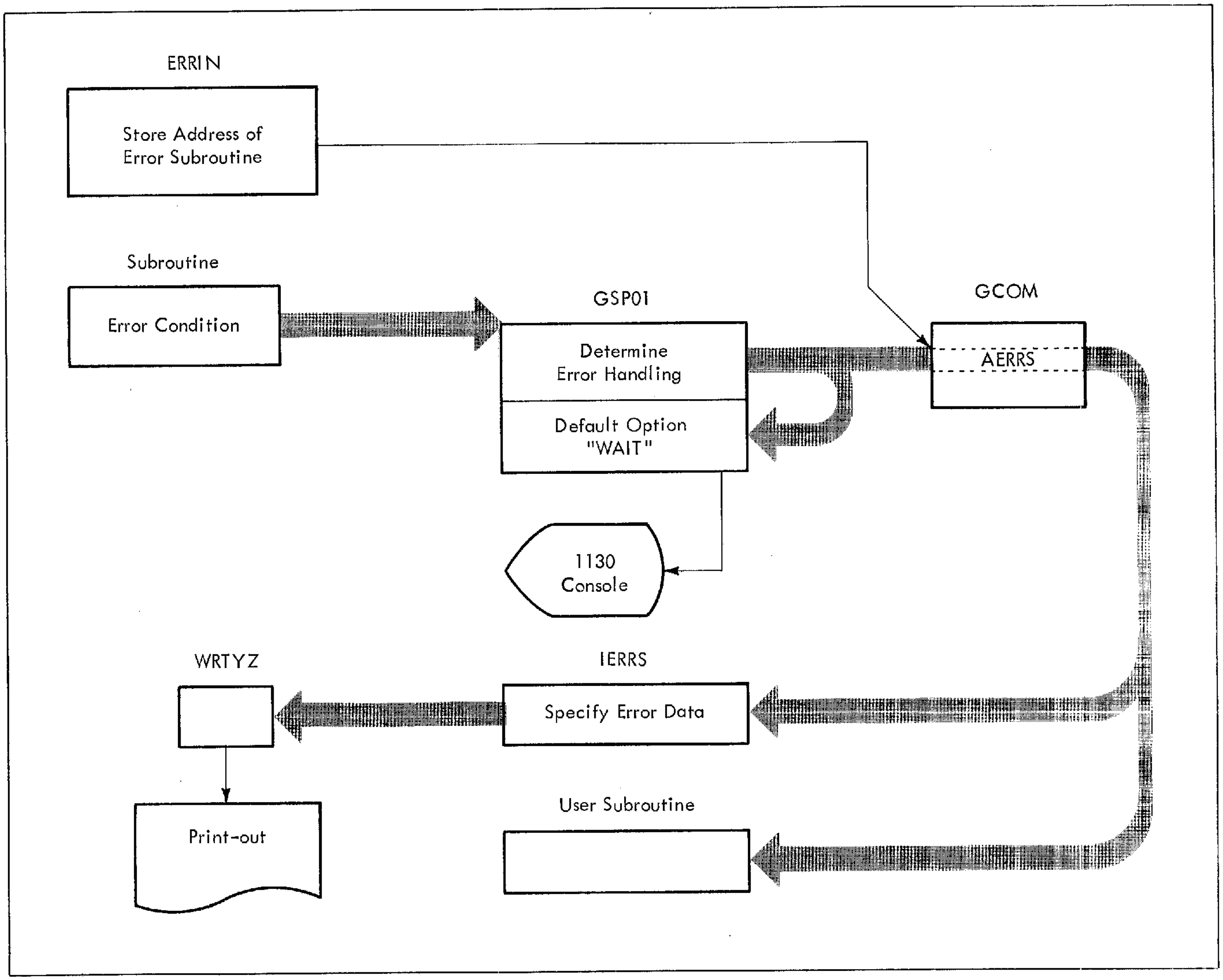


Figure 14. Error Handling

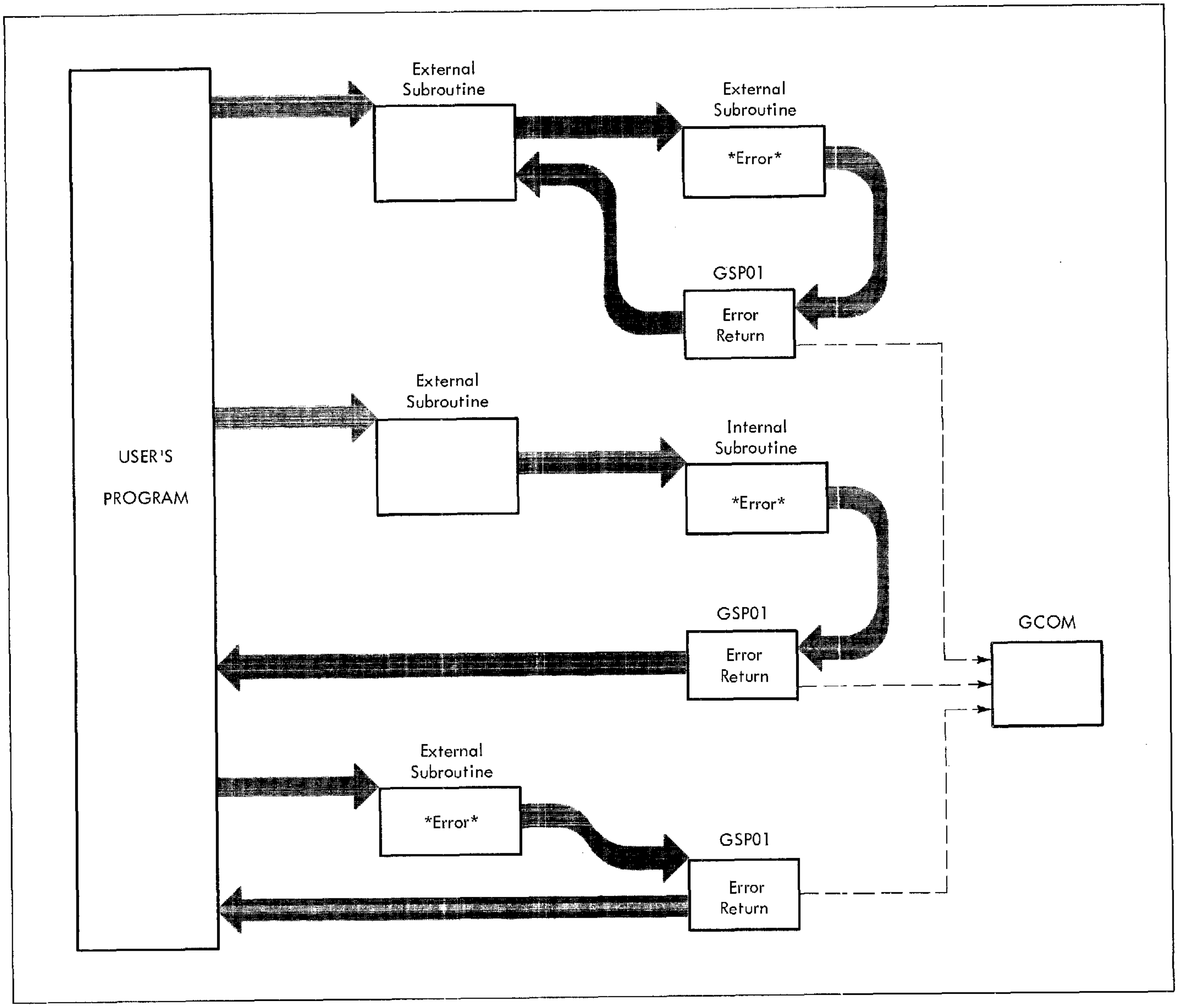


Figure 15. Error Return Linkage

The GSP subroutines are divided into several groups according to the specific functions that each performs. These groups are as follows:

- Initialization and Termination
- Image Management
- Image Generation
- 2250 I/O Subroutines
- Attention Enabling and Information Retrieval
- Light Pen Subroutines
- Alphameric and Text Generation Subroutines
- Attention Processing
- Error Handling
- Internal Service Subroutines

INITIALIZATION AND TERMINATION SUBROUTINES

This group of subroutines performs a number of services that are required prior to generating and displaying graphic images. Three GSP control areas (see Appendix A) are involved in the initialization process. Briefly, they are as follows:

GCOM -- the principal communications area for the GSP.
ICA preface -- pointers and generation control data for generating and storing image data in the ICA.
GCA -- scaling and scissoring information; increment and indexing values; input data specifications and output data requirements.

The subroutines in this group are:

- GSPIN, which initializes the GSP by setting initial values in GCOM.
- ICAIN, which initializes an ICA preface with initial values for image generation.
- GCAIN, which initializes a GCA with standard values.

The following six subroutines initialize one portion of a GCA:

- SSCAL, which sets scaling information.
- SSCIS, which sets the scissoring option (grid or screen).
- SINDX, which sets indexing values for accessing input arrays.
- SINCR, which sets increment values for coordinate generation.
- SDATM, which sets the input data mode.
- SGRAM, which sets the output graphic mode.

The subroutine GSPTM terminates use of the GSP.

GSPIN -- Graphic Subroutine Package Initialization

Chart: AA

Function: To store the address of GCOM in COMMA, initialize GCOM, and start regeneration of the system display.

Entry: GSPIN

Exit: Returns to the next sequential instruction in the calling program; or for certain error conditions (see below) returns to the supervisor.

Input:
Argument 1 - address of the integer precision option.
Argument 2 - address of the real precision option.
Argument 3 - address of the error return variable.
Argument 4 - address of the cumulative errors variable.

Output: See "Function."

Operation: GSPIN obtains the address of GCOM from the Transfer Vector table and stores it in the $GCOM field in COMMA. If the table contains addresses of subroutines whose loading was requested with the *G2250 card, these addresses are stored in fields provided in GCOM.

Generation control fields in GCOM are initialized using the initialization function of GSP03. Other fields are initialized with values provided in the calling sequence. All branch orders in the system display portion of GCOM are set to perform no-operation. DSPYN is used to start regeneration of the system display.

Errors:
 Code 1 - A precision option was not specified as 0 or 1.

The following errors are also detected by GSPIN. When one of these errors is encountered, a code identifying the error is displayed in the accumulator and extension lights, and the GSP waits. Depression of the start key causes an exit to the supervisor.

ACC.	EXT.	
/FFFF	200C	- A *G2250 control card has not been processed.
/FFFF	200D	- The error variable addresses are equal.

Comments: The subroutines whose loading is requested with the *G2250 card are GSP05 (Push-up), GSP06 (Scissor), GCHAR or GUPER (Character Stroke Subroutines), GSP11 (Index-Controlled Entity Scan), and GSP12 (Level-controlled Direct Entry).

ICAIN -- Image Construction Area Initialization

Chart: AB

Function: To define and initialize, or reactivate, an image construction area (ICA); to place the address of GSP08 (Light Pen and Order Controlled Direct Entry) in GCOM.

Entry: ICAIN

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 start of the ICA.
 Argument 3 - address of the end of the ICA.
 Argument 4 - address of the option code.

Output: See "Operation."

Operation: If an ICA is active, GSP03 (transfer function) is used to transfer data from fields LGXI through OPLST in GCOM to fields GENX through LSTMD in the preface of the active ICA. The option code (argument 4) is tested for the following values:

 0 = define and initialize an ICA.
 1 = reactivate an ICA.

Define and Initialize: ICAIN constructs an ECCB for the ICA and stores its address in the ACICA field in GCOM. The addresses of the first and last words available within the ICA are stored in fields SAVA and EAVA in the ICA preface. Generation control fields in the ICA preface are set to initial status (see Appendix A, Table 9). Where real values are required, FLOAT is used to convert integer values to real values, and GSP10 is used to store them. GSP03 is used to transfer all generation control data from the ICA preface to GCOM. DSPYN is used to store the address of GSP08 (direct entry subroutine for light pen and order controlled attentions) in fields LDPE and OCDE in GCOM. ICAIN chains ICAs as they are defined using the ICAPT field in the ICA preface.

Reactivate: ICAIN searches the chain of ICA's using the correlation value specified. When the specified ICA is found, the address of the ICA ECCB is stored in the ACICA field of GCOM, thereby making it the active ICA and the ELSTA field of the ECCB is stored in AICAP. The generation control data in the active ICA preface is transferred by GSP03 to GCOM.

Errors:
 Code 1 - The option code has a value other than 0 or 1.
 Code 2 - The correlation value is not in the range 1 to 32767.
 Code 3 - The option code is 1 and the specified ICA is not found.
 Code 8 - The ICA is less than 28 words long.

GCAIN -- Generation Control Area Initialization

Chart: AC

Function: To initialize a GCA with standard values for scaling, scissoring, indexing, incrementation, input data mode, and output graphic mode.

Entry: GCAIN

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

Input: Address of the GCA.

Output: See "Operation."

Operation: GCAIN first sets the entire GCA to zero and then sets only those fields requiring non-zero values. The GCA, after initialization, contains the following values:

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

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

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

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

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

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

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

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

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.

Output: See "Operation."

Operation: SDATM tests the mode arguments and, if valid, uses the values to set fields XIPMD and YIPMD in the GCA. The valid mode arguments and their meanings are:

1 = real absolute

2 = real incremental

3 = integer absolute

4 = integer incremental

5 = integer absolute in 2250 raster units

Errors:
Code 1 - One of the mode arguments is not a positive integer in the range 1 through 5.

SGRAM -- Set Output Graphic Mode

Chart: AE

Function: To set the output graphic mode in a GCA for generating absolute, incremental, or optimized data.

Entry: SGRAM

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 used to set the output graphic mode.

Output: See "Operation."

Operation: SGRAM tests the validity of the value for setting the output graphic mode. Any of the following is valid:

1 = optimized
2 = absolute
3 = incremental

If valid, the value is stored in the OPMD field of the GCA.

Errors:
Code 1 - The value for setting the output graphic mode is not in the range 1 through 3.

GSPTM -- Graphic Subroutine Package Termination

Chart: AF

Function: To terminate use of the GSP.

Entry: GSPTM

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

Input: None.

Output: None.

Operation: GSPTM uses DSPYN to turn off programmed function keyboard lights and to reset the 2250. The ICA chain is disconnected by storing zeros in all ICA pointers. GSPTM then sets to zero the LPDE and OCDE fields in GCOM to ensure that all GSP activity is stopped.

Errors: None.

IMAGE MANAGEMENT SUBROUTINES

The primary service of image management subroutines, to establish or modify the structure of graphic elements in a GSP program, is accomplished by BELMT, EELMT, DELMT, XELMT, and UELMT. They provide a means of working with an image construction area and changing its format to suit particular needs, while maintaining an economy of core storage. The secondary services provided by the image management subroutines BXGEN, EXGEN, and IELMT allow generation outside of an ICA, while IDPOS and RDPOS allow updating of elements with incremental input, and STPVS and SATRB allow the program to change the attributes of elements. The image management subroutines and their functions are:

- BELMT, which defines the beginning of a controlled, uncontrolled, level controlled, index controlled, subroutine, or image entity.

- EELMT, which defines the end of an element whose beginning was specified by BELMT or UELMT, or which was extended by XELMT.

- DELMT, which identifies an element that is to be deleted.

- UELMT, which identifies an element whose content is to be modified by subsequent image generation subroutines.

- XELMT, which identifies an element whose content is to be extended by subsequent image generation subroutines.

- SATRB, which sets one or both of the attributes of a controlled entity.

- STPVS, which changes the GSP verification setting.

- IDPOS, which indicates the coordinates of the next element for scissoring after use of a linkage entity or when modifying an existing element.

- RDPOS, which saves the last generated x- and y-coordinates.

- GSP04, which moves all non-deleted elements to lower core locations overlaying core previously occupied by elements indicated as deleted.

- GSP05, which moves elements following a specified element to higher core locations to allow for the extension of the specified element.

The following subroutines are used by the Assembler-language programmer:

- IELMT, which builds an ECCB for an element that is external to the image construction area.

- BXGEN, which causes the GSP to enter external generation mode, thus allowing the generation of orders outside of an ICA.

- EXGEN, which causes the GSP to end external generation mode.

BELMT -- Begin Element

Chart: BA

Function: To specify the type, name, and beginning for an image, controlled, level-controlled, index-controlled, uncontrolled, or subroutine entity.

Entry: BELMT

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 a code identifying the type of element.

Output: See "Operation."

Operation: For an uncontrolled, controlled, level-controlled, index-controlled, or image entity, BELMT requests the build function of GSP03, passing the correlation value and a code identifying the type of element. For a subroutine entity, it first sets the SUBM field in GCOM, requests the transfer function of GSP03, and requests the generation of overhead orders by GSP02 before requesting the build function. After the ECCB for the element has been built, BELMT requests the storing of overhead orders by GSP02, as follows:

Element	Orders	Reason
Uncontrolled entity	None	
Controlled entity and Index-controlled entity	SPM GB	Enable detects Branch around the element when its visibility attribute is non-display.
Level-controlled entity	GB	Branch around element when its visibility attribute is non-display.
	GSB	Branch to left bracket of verification subroutine in GSP12.
	SVRT	Return linkage to right bracket of verification subroutine in GSP12.
Subroutine entity	GB	Protective branch around the subroutine.
	SRVT	Return linkage to the main order program.
Image entity	SRVT	Return linkage to the system display.

Errors:
Code 1 - The code identifying the element type is not in the range 1 to 6.
Code 2 - The correlation value is not in the range 1 to 32767; or The correlation value has been previously defined.
Code 9 - The generation mode is external.

EELMT -- End Element

Chart: BB

Function: To define the end of one or more elements.

Entry: EELMT

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

Input: Address of the correlation value of the element to be ended.

Output: None.

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 16 - The element has already been ended; or
the element is not in the limits specified by ELSTA and ELEND of the current update ECCB.

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.

If there is an image being displayed, it is replaced by a processing message. GSP05 pushes up the words in the ICA to provide the area for extending the element.

Errors:
Code 1 - Argument 2 (LPC count) is invalid.
Code 2 - The correlation value is not in the range 1-32767.
Code 3 - The ECCB is not found by GSP03.
Code 4 - The element type is invalid.
Code 6 - The element has not been ended.
Code 7 - There is no active ICA.
Code 9 - The GSP is already in update or external generation mode.

IELMT -- Include Element

Chart: BG

Function: To build an ECCB for an element generated in external generation mode.

Entry: IELMT

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 first order in the included element.
Argument 3 - address of the last order in the included element.

Output: See "Function."

Operation: If the arguments are valid and there is an active ICA, GSP03 builds an ECCB for a subroutine entity in the active ICA.

Error:
Code 1 - The start or end address is invalid.
Code 2 - The correlation value is not in the range 1 to 32767.

BXGEN -- Begin External Generation

Chart: BH

Function: To enable the Assembler-language programmer to use GSP image generation subroutines to generate graphic data outside of an ICA.

Entry: BXGEN

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

Input:
Argument 1 - address of the generation area.
Argument 2 - address of the length of the generation area.
Argument 3 - address of the one-word field to hold the count of the number of words generated.
Argument 4 - address of the one-word field to hold the count of the number of words not generated.

Output: See "Operation."

Operation: GENIN is tested to ensure normal mode and the length argument is tested to ensure a value in the range 1-32767. ACICA (in GCOM) is checked to see if there is an active ICA. If so, GSP03 transfers generation control data from GCOM to the ICA preface, and NAGAP and LAGAP are saved in the ICA preface. For either condition of ACICA (active or inactive ICA), the necessary indicators in GCOM are initialized to prepare for external generation (NAGAP=Arg 1, LAGAP=ARG 1 + Arg 2 - 1, external generation count=0, external non-generation count=0, EGCA=Arg 3, ENGCA=Arg 4, GENIN=2).

Errors:
Code 1 - The length of the generation area is not greater than 0.
Code 9 - BXGEN is entered with the generation mode already set for external generation.

EXGEN -- End External Generation

Chart: BJ

Function: To enable the Assembler-language programmer to end the generation of graphic data outside of an ICA.

Entry: EXGEN

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

Input: None.

Output: None.

Operation: If ACICA is non-zero, indicating an active ICA, NAGAP and LAGAP are restored from the ICA preface, and GSP03 transfers data from the ICA preface to GCOM. For either condition of ACICA, the external generation count and the external non-generation count are reinitialized to zero, and GENIN is set to 0 (normal generation mode).

Errors:
Code 9 - EXGEN is entered when the generation mode is not external.

SATRB -- Set Controlled Entity Attributes

Chart: BK

Function: To change the visibility and detectability attributes of a controlled, index-controlled, or level-controlled entity.

Entry: SATRB

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 visibility attribute.
Argument 3 - address of the detectability attribute.

Output: Orders are stored in the controlled entity as described in "Operation."

Operation: GSP03 is used to locate the ECCB for the indexed, level-controlled, or controlled entity. The element's attributes are then set in accordance with the specified attributes. A visibility argument of -1 results in a graphic branch indirect order being stored in the second word of the element. This causes a branch to the end element address when the order program is being executed, resulting in the controlled entity not being displayed. A visibility argument of +1 results in a graphic branch indirect order with the no-operation bit set to be inserted in the second word of the element. This permits display of the entity. For a controlled or indexed entity, a detectability of -1, an SPM /4 order is stored in the first word of the element resulting in the disabling of light pen detects. A detectability argument of +1 causes an SPM /A order to be stored in the first word of the element resulting in the enabling of light pen detects. For a level-controlled entity, a detectability argument of -1 results in the storing of a GSB to the left bracket non-detect routine. A detectability of +1 results in the storing of a GSB to the left bracket immediate detect routine. A detectability argument of +2 results in the storing of a GSB to the left bracket verification routine. A visibility or detectability argument of 0 results in no change to the detectability or visibility attribute.

Errors:
Code 1 - An attribute argument (2 or 3) is invalid.
Code 2 - The correlation value is not in the range 1 to 32767.
Code 3 - The ECCB was not found by GSP03.
Code 4 - The element is not a controlled, level-controlled, or index-controlled entity.
Code 6 - The element is not ended.

STPVS -- Step Verification Setting

Chart: BL

Function: To change the level at which verification occurs on nested level-controlled entities.

Entry: STPVS

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

Input: Address of the value indicating the type of change as follows: negative number=step down one level (5 is lowest or least inclusive) positive number=step up one level (1 is highest or most inclusive) zero=reset to level one.

Output: See "Operation."

Operation: STPVS ensures that the Verification Direct Entry subroutine (GSP12) is loaded and saves the contents of the verification level indicator (VERLN in GCOM). If VERLN is 1 and a step up is indicated, or if VERLN is 5 and a step down is indicated, no further processing occurs. If a step up is indicated, VERLN is incremented by 1. If a step down is indicated, VERLN is decremented by 1. If a reset is indicated, VERLN is set to 1.

The saved VERLN is used as an index to set the present verification, detect, and no detect bracket subroutines in GSP12 to perform no operation. The adjusted VERLN is used as an index to insert the graphic control orders in the new level bracket subroutines which will make the subroutines operational.

Errors: Code 8 - GSP12 is not loaded.

IDPOS -- Indicate Element Position

Chart: None.

Function: To indicate to the GSP the starting point from which subsequent x- and y-coordinates are to be computed.

Entry: IDPOS

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 x-coordinate.
  Argument 3 - address of the y-coordinate.

Output: None.

Operation: IDPOS requests the GSP02 generation function to replace the values in the last-generated and last-computed fields of GCOM (LGXI through LCYI) with the x- and y-coordinates that were passed to IDPOS by the program.

Errors: None.

RDPOS -- Read Position

Chart: None.

Function: To place the last-generated x- and y-coordinates in a program-defined array.

Entry: RDPOS

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

Input: The address of the first element of a 2-element integer array.

Output: See "Operation."

Operation: The contents of fields LGXI and LGYI in GCOM are stored in a 2-element integer array. The address of the second element of the array is computed using the integer precision (INPRE) field in GCOM.

Errors: None.

GSP04 -- Push Down

Chart: BM

Function: To make area available in the active ICA when the ICA is completely full, but contains elements indicated as deleted.

Entry: GSP04

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

Input: None.

Output: None.

Operation: If the generation mode is update, a processing message will already have been displayed. If the generation mode is normal and the ICA which is the subject of the push-down contains an image entity currently being displayed, a switch is set, the address of the processing message is stored in the system display, and DSPYN starts regeneration of the system display. This prevents movement of data in the ICA while the image entity is being displayed.

If DLCNT (delete count) in the ICA preface is zero, the push-down function is completed. If the switch (above) is set, the linkage to the processing message is terminated, and the previously displayed image entity is restored.

If DLCNT is non-zero, the active ECCB chain is searched for an ECCB for a deleted element. The element start and end (ELSTA and ELEND) addresses in the ECCB are saved as pointers. If the ECCB was for a subroutine, its ELSTA pointed to the third word rather than the first, and the saved ELSTA is decremented by two. All graphic orders in higher core positions are moved to lower core positions, overlaying the area of the deleted element.

The ECCB for the deleted element is removed from the active chain, and the forward pointer index of the prior ECCB is updated to hold the contents of the removed ECCB's forward pointer index. The ECCB for the deleted element is then inserted into the available chain.

Since the deleted element may have been nested within other elements, the entire active ECCB chain is searched for ECCBs affected by the push-down. Those elements which contained the deleted element have their ELEND updated. Those which did not contain the deleted element but were affected by the push-down have ELEND and ELSTA updated by the same factor. Any image, level controlled, or subroutine entities in the active ECCB chain have their graphic branch orders adjusted to reflect the current location of the second word of their SRVT order.

DLCNT in the ICA is decremented by one, and the process is repeated beginning with the test of DLCNT.

Errors: None.

Note: If the ECCB for the deleted element is in the lowest core position of all ECCBs, its four word area is added to the available generation area, not to the available ECCB chain. The higher core orders as pointed to by ELEND and limited by the last available generation area pointer (LAGAP) are pushed down to the address indicated by ELSTA. Neither ELSTA nor ELEND is altered. The next available generation area pointer (NAGAP) is then updated to reflect the new location for storing orders.

GSP05 -- Push Up

Chart: BN

Function: To make area available in the active ICA for extending an element.

Entry: GSP05

Exit:
  Normal - Returns to the calling subroutine through the address stored in the entry word.
  Error - Returns to the mainline program through GSP01.

Input: None.

Output: None.

Operation: A normal push-up moves the words immediately following the element to be extended (pointed to by CUCB) to higher core positions, the distance of which is specified by INCTR (in GCOM). Thus, a "gap" equal to INCTR in length remains immediately following the element being extended. This "gap" is filled with GNOP orders. The element start and end addresses (ELSTA and ELEND) in the ECCBs of the moved words are updated to reflect their new locations. The branch through the SRVT order in level controlled, image, and subroutine entities is updated. If INCTR is greater than the available area, the words are moved to the limit of the available area.

When an ECCB points to a deleted element following the element to be extended and it will provide sufficient area for the extension, the area limited by the ECCB is filled with GNOP orders. If the ECCB for the deleted element is at the end of the active ECCB chain (lowest core position), its area is added to the available generation area. Otherwise, it is added to the chain of available ECCBs.

Errors:
  Code 5 - There are no ECCBs for deleted elements, and no available area in the ICA.

IMAGE GENERATION SUBROUTINES

The image generation subroutines PLINE/PPNT/PSGMT, PTEXT, PGRID, and MVPOS convert input data to 2250 format using specifications in the generation control area. The converted data is stored in the image construction area. The generation function of GSP02 generates the orders for MVPOS and PLINE/PPNT/PSGMT. Other orders are generated by the subroutines themselves and stored by GSP02. The following is a list of the image generation subroutines and their functions:

- MVPOS, which establishes the starting coordinates for the next element.
- PLINE/PPNT/PSGMT, which converts input data to 2250 format for plotting lines, points, or line segments.
- PTEXT, which converts input data to 2250 format for plotting characters.
- LKSUB, which establishes linkage to a subroutine or tracking entity.
- PGRID, which generates data in 2250 format to plot the rectangular outline of a grid.
- PCOPY, which generates a copy of an element at another place in an image or subroutine entity.
- GSP02, which is the primary subroutine for generating orders. It consists of (1) text function; (2) store function; (3) generation function; and (4) increment function.
- GSP06, which computes the coordinates at which a vector crosses a screen or grid boundary so that that part of the vector which is external to the boundary will not be displayed.

MVPOS -- Move Element to a Position

Chart: CA

Function: To generate data, in either absolute or incremental mode, to position the beam at some location on the 2250 screen without creating an image.

Entry: MVPOS

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 x-coordinate.
  Argument 3 - address of the y-coordinate.
  Argument 4 - address of the correlation value.

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.

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.

Input:
  Argument 1 - address of the GCA.
  Argument 2 - address of the first character.
  Argument 3 - address of the count of the characters.
  Argument 4 - address of the character size code.
  Argument 5 - address of the text code.

Output: See "Operation."

Operation: PTEXT tests the count and, if valid, stores the count in ICNTR and builds a character mode order which is stored in the ICA by GSP02. GSP02 text function generates the required GSBs and stores them in the ICA. A GNOP order is then generated to terminate character mode, and GSP02 stores it in the ICA.

Errors:
  Code 1 - The count is not positive; or the size is greater than two; or the text code is not in the range 1-4.
  Code 8 - GCHAR is not loaded.

LKSUB -- Linkage to a Subroutine

Charts: CD,CE

Function: To provide linkage to one of the following:

- subroutine entity
- tracking entity
- subroutine generated outside an ICA

The linkage may be named or unnamed, active or inactive; a previously named linkage entity may be updated.

Entry: LKSUB

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

Input:
  Argument 1 - address of the correlation value for the element being linked to.
  Argument 2 - address of the correlation value for the linkage entity.
  Argument 3 - address of the switch argument (active/inactive).

Output: A graphic branch or graphic branch no-operation order is stored in ICA.

Operation: LKSUB determines which of the following services is requested:

- Create an unnamed linkage entity -- indicated by a zero correlation value for argument 2.
- Create a named linkage entity -- indicated by an unassigned correlation value for argument 2.
- Update a linkage entity -- indicated by a correlation value previously assigned to a linkage entity for argument 2.

Unnamed linkage: The GSP03 search function locates the ECCB for the element being linked to. The address in the ELSTA field of the ECCB is stored in the address portion of a Graphic Branch order, and GSP02 is used to store the order in the ICA.

Named linkage: The GSP03 search function locates the ECCB for the element being linked to if a correlation value is given in the calling sequence for argument 1. The address in the ELSTA field of the ECCB is stored in the address portion of a Graphic Branch order. GSP03 is then used to build an ECCB for the linkage entity. If the switch argument specifies an inactive linkage (graphic branch no-op) or if the calling sequence does not provide a correlation value for the element to be linked to, the no-operation bit in the Graphic Branch order is set. GSP02 is used to store the Graphic Branch or graphic branch no-operation order in the ICA. The ELEND and INDIC fields of the ECCB for the linkage entity are then updated.

Update: GSP03 locates the ECCB for the element to be updated and the ECCB for the element to be linked to. If the calling sequence does not provide a correlation value for the element to be linked to, the address portion of the Graphic Branch in the old linkage is used to find the ECCB. The address of the element to be linked to is placed in the address portion of a Graphic Branch order. If the switch argument specifies an inactive linkage, the no-operation bit in the order is set. The order is then stored in the ICA, overlaying the old Graphic Branch.

Errors:
  Code 1 - The switch argument is not 1 or 2.
  Code 2 - One of the correlation values is not in the range 1 to 32767.
  Code 3 - The correlation value for the element being linked to is not defined.
  Code 4 - The element being linked to is not one of the types for which a linkage can be generated. (See types under "Function.")

PGRID -- Plot Grid Outline

Chart: CF

Function: To provide for the generation of orders for displaying the four sides of a grid on the 2250 screen.

Entry: PGRID

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

Input: Address of the GCA.

Output: See "Operation."

Operation: The start address of the generation control area (GCA) is determined and stored. The x-input mode is saved and XIPMD is set to integer absolute in 2250 raster units. The grid limits are determined from words 0-4 in the GCA and stored for the generation of orders. GSP02 stores a Set Graphic Mode Vector (SGMV) overhead order, if necessary, and the increment function of GSP02 generates a Move Beam Incremental order to the lower left limits of the grid. The generation function of GSP02 then generates the orders that will display the specified grid. XIPMD is restored.

Errors: None.

PCOPY -- Plot Copy of an Element

Chart: CG

Function: To copy the graphic data generated for a previously defined element and include it as all or part of the graphic data for the current element.

Entry: PCOPY

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

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

Output: See "Operation."

Operation: PCOPY ensures the validity of the correlation value, and GSP03 finds the ECCB for the element to be copied. Pointers reflecting the limits of the data are set using ELSTA and ELEND as bases and adjusting ELEND to eliminate overhead orders dependent on element type (e.g.; image entity -- ELEND minus 8; subroutine entity -- ELEND minus 2). A "to" pointer is set to NAGAP and the "from" pointer to ELSTA. Each order is examined. Overhead orders are ignored and the "from" pointer is incremented to the next order. Non-overhead orders are copied into the current element, and the "to" and "from" pointers are incremented for the next order. When the specified element has been copied into the current element, NAGAP in GCOM is updated.

Note: Copied elements do not affect the attributes of the current element. Nested subroutine entities are not copied, but nested linkage entities are copied.

Errors:
Code 2 - The correlation value is not in the range 1-32767.
Code 3 - The ECCB is not found by GSP03.
Code 5 - The ICA is full.
Code 6 - The ECCB found by GSP03 is for an element that is not ended.

GSP02 -- Generation Support

Charts: CH,CJ,CK,CL,CM,CN

Function: GSP02 provides one of the following functions for a calling subroutine.

- Generation Function -- to determine coordinates for a point(s) on a grid, and generate appropriate orders if necessary.
- Store Function -- to store generated orders in the ICA.
- Increment Function -- to generate incremental orders.
- Text Function -- to generate Graphic Short Branch orders to character stroke subroutines.

Entry: GSP02

Exit:
Normal - returns through the address stored in the entry word to the next sequential instruction in the calling subroutine.
Error - returns to the mainline program through GSP01 error function.

Input: For Generation Function:
Argument 1 - address of the GCA
Argument 2 - address of the x-start array
Argument 3 - address of the y-start array
Argument 4 - address of the x-end array
Argument 5 - address of the y-end array

For Text Function:

Argument 2 - address of the text array
Argument 5 - address of the Text code

For Store Function: the words to be stored are in the accumulator and the extension.

Output: See "Function."

Operation: GSP02 stores the contents of the accumulator and extension in a constants area and examines the one-word function indicator in the calling sequence. The calling sequence and an explanation of the indicator word are as follows:

```
LIBF    GSP02
DC /n1 n2 n3 n4
```

n1 indicates the function as follows:

0=Generation Function
1=Text Function
2=Store Function
3=Increment Function

n2 and n3 are unused.

n4 indicates the calling subroutine to the generation function as follows:

0=PLINE or PPNT
1=PSGMT
2=IDPOS
3=MVPOS
4=LCPOS or TRACK

n4 indicates the calling subroutine to the text function as follows:

0=PTEXT
1=MSGIN

The generation function stores the address of the GCA (argument 1) and the ID of the calling subroutine in a constants area and sets a "first time" switch. The x-start phase is entered and the x-start indicator and increment indicator are initialized. The address of the x-array (argument 2), XSIND, XSINC, XIPMD, and OPMD (in the GCA) are stored in the constants area. The x-value is determined from the x-array and scaled using the x-scaling factors in the GCA. The result is the current computed x-coordinate.

GSP02 then enters the y-start phase to calculate and scale the y-value. This phase is similar to x-start except that the y-indicator fields are used with argument 3. The result is the current computed y-coordinate.

If the current computed and/or last computed x- or y-coordinates are outside the screen or grid limits and the Scissor subroutine GSP06 is loaded, GSP06 performs the scissoring function. Otherwise, the new generation indicators are updated and the new position indicators are masked with hexadecimal FFFF.

The generation of orders is determined by a combination of the calling subroutine and the output mode. The type of order generated (move or draw) is determined by the necessity of positioning.

If the calling subroutine is MVPOS, and NGXI indicates that the coordinate point is outside the screen or grid limits, a two-word GNOP order is generated and stored, and the last computed indicators (LCXI and LCYI) are updated. If NGXI indicates that the coordinate point is inside the screen and grid limits, and the output mode is absolute or optimized, a Move Beam Absolute order is generated and stored, and the last computed indicators (LCXI and LCYI) are updated. If NGXI indicates that the coordinate point is inside the screen and grid limits, and the output mode is incremental or for subroutine generation, Move Beam Incremental orders are generated and stored, and the last generated and last computed indicators (LGXI and LGYI; LCXI and LCYI) are updated.

If the calling subroutine is not MVPOS, and NGXI indicates that the coordinates are inside the screen or grid, the output mode and generation mode determine the generation and type of order. If the output mode is incremental or the orders are for a subroutine, incremental orders are generated. The increment function splits the vector into blanked and/or unblanked increments depending upon the calling subroutine and the necessity of positioning.

If the output mode is absolute, MBAX and MBAY and/or draw beam absolute (DBAX and DBAY) orders are generated depending upon the calling subroutine and the necessity of positioning.

If the output mode is optimized, MBA, MBI, DBA, and/or DBI orders are generated depending upon the calling subroutine, the necessity of positioning, and the "first time" switch.

All generated orders are stored by GSP02 store function. For all conditions where MVPOS is not the calling subroutine, and the new coordinates are inside the screen or grid boundary, LGXI, LGYI, LCXI, and LCYI are updated. If the new coordinates are not within the screen or grid, only LCXI and LCYI are updated by GSP02 (LGXI and LGYI are updated in GSP06).

If PSGMT is the calling subroutine, the x-end and y-end phases must be entered to generate and store orders, if necessary, for both points. These phases are similar to the start phases except that x- and y-end indicators are used with arguments 4 and 5. If the calling subroutine is not PSGMT, or if all four phases have been completed, the external counter (ICNTR in GCOM) is decremented by one. When the external counter equals zero, return is made to the calling subroutine. If the external counter is non-zero, the x-start phase is re-entered.

The store function determines if there is a minimum of one word available in the ICA (LAGAP is greater than NAGAP). If area is available, the accumulator and extension are loaded from the save area, and the contents of the accumulator are stored in the ICA as referenced by NAGAP. NAGAP is then incremented by one and if the order is 2 words long, the contents of the extension are moved into the accumulator, and the above process is repeated.

If there is not a one-word area available, and the generation mode is normal or update, GSP04 (Push-down) or GSP05 (Push-up) will attempt to provide space for the storing of the order. If storing of the order is not possible, the number of words not stored (ICNTR) is stored in the external non-generation counter (ENGCA) in GCOM.

The increment function determines the number of moves and the type of order required. The coordinates for each incremental vector are calculated and stored in the model order in the constants area. Each order is then stored in the ICA by the GSP02 store function. The number of moves is decremented by one and the process repeated until the number of moves equals zero.

The text function stores the address of the text array (argument 2) and determines the character size. Using the text array as an index, the address of the GSB for the character is determined and stored in the constants area. The pointers to the ICA are updated and GSP02 store function inserts the order in the ICA. ICNTR is decremented by one and the process repeated until the counter equals zero.

Errors:

Code 8 - GSP06 is not loaded and scissoring is required.

Code 16 - Internal Error, the function indicator word does not specify one of the available GSP02 functions; or
Generation - The scissor subroutine is not loaded when the current computed x- and y-coordinates are outside the screen/grid; or
Store - There is no available area in the ICA and the delete counter is zero.

GSP06 -- Scissor

Chart: CO

Function: To compute the coordinates of the points at which a vector intersects a grid or screen boundary and to define to the calling subroutine the positioning coordinates (if any) and the new generation coordinates (if any) so that the generation will be limited to the interior of the boundary.

Entry: GSP06

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

Input: None.

Output: The following updated fields in GCOM:

NPXI - new positioning point X integer (if any)

NPYI - new positioning point Y integer (if any)

NGXI - new generation point X integer

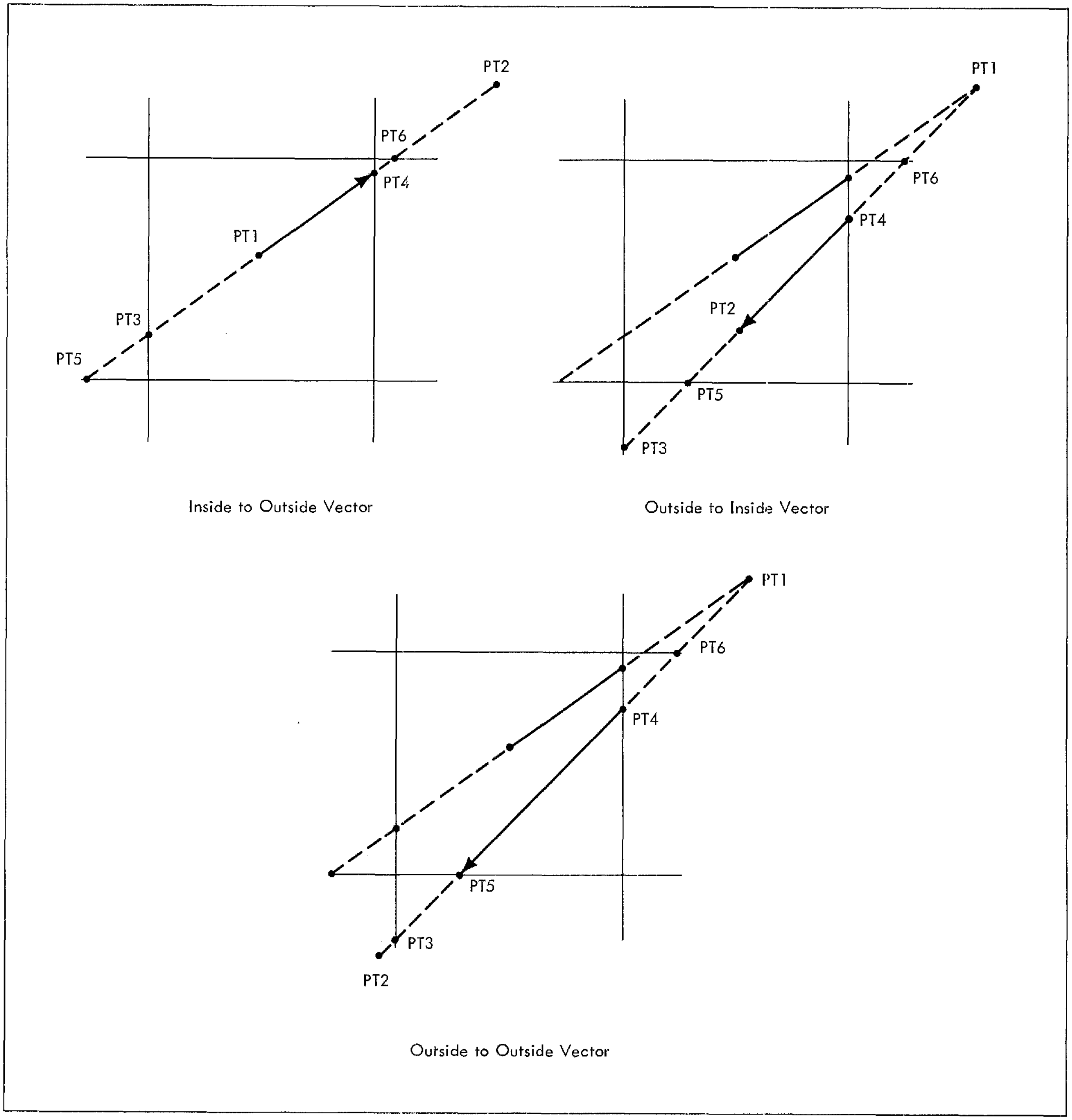
NGYI - new generation point Y integer

Operation: Three types of vectors requiring scissoring are possible in relation to a grid or screen boundary. A vector may originate inside and proceed to a point outside the boundary; it may originate outside and proceed to a point inside the boundary; or it may originate outside and proceed through the interior to a point outside the boundary.

The last computed x- and y-coordinates (PT1; see Figure 16) and the current computed x- and y-coordinates (PT2) in GCOM are used by GSP06 to compute up to four points at which the vector and its extension intersect the boundary and its extensions. All coordinates for points outside the boundary and all coordinates for points outside the limits of the original vector are indicated as deleted. If the number of remaining boundary points is zero, the vector does not pass through the boundary, and the value hexadecimal FFFF is stored in NPXI, NPYI, NGXI, and NGYI.

If the number of remaining boundary points is greater than two, duplicate sets of coordinates exist, and one set is indicated as deleted.

For the first type of vector (inside to outside), no positioning is necessary. Therefore, the value hexadecimal FFFF is stored in NPXI and NPYI. The coordinates



Inside to Outside Vector

Outside to Inside Vector

Outside to Outside Vector

Figure 16. Scissoring to Screen or Grid Boundaries

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 (SD5) 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 ILS return address. The ILS 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.

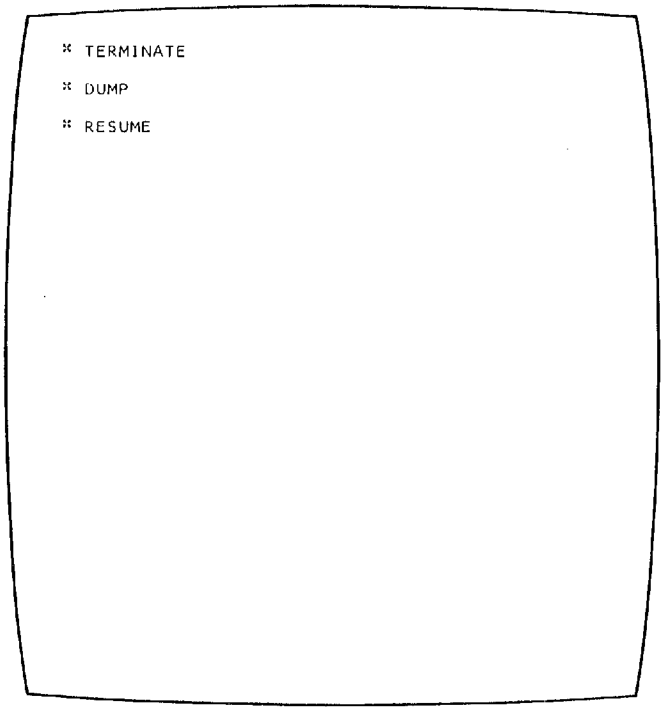


Figure 17. Cancel Frame

<u>END key attention</u>: An attention produced by the END key causes GCOM to be tested to determine if a direct entry routine is specified for the END key. Control passes to the direct entry routine. When control returns, or if no direct entry routine is specified, GCOM is tested to determine if GSP is being used. When GSP is being used by the program, the first word of ATDAT, the area containing the attention information for the RQATN subroutine, is tested for zero. A zero in this area, indicating no previous attention information is being awaited, causes the SATEN field of GCOM to be tested to determine if the END key had been enabled. If the END key is enabled, the END key code is placed in ATDAT. The remainder of ATDAT is filled with zeros, the display is restarted, and control returns through ILS03 to the interrupted program.

<u>Other ANKB attentions</u>: If the ANKB attention is neither CANCEL nor END, control passes to the direct entry routine. When control returns, or if a direct entry routine is not specified, GCOM is tested to determine whether GSP is being used. The ISSDE field of GCOM is inspected to determine if a direct entry routine has processed the attention. If ISSDE indicates that processing has been completed by the direct entry routine and that the attention is not to be saved for inspection by the problem program, the display is restarted and control returns to the interrupted program. When GSP is not being used, control returns through ILSX3 to the interrupted program without restarting the display.

If processing has not been completed by the direct entry routine, the first word of ATDAT is tested for zero. A non-zero, indicating a previous attention is waiting to be processed, causes the display to be restarted and control to be returned through ILSX3 to the interrupted program. A zero in ATDAT causes an indicator in GCOM (SATEN) to be tested to determine whether any ANKB attentions have been enabled. If not enabled, the display is restarted and control returns through ILSX3 to the interrupted program. If ANKB attentions are enabled, the ANKB code is placed in the first word of ATDAT, and the EBCDIC code for the key causing the attention in the third word. The display is restarted and control returns through ILSX3 to the interrupted program.

<u>PFKB attentions</u>: An attention from the PFKB causes control to pass to the direct entry routine specified for PFKB attentions. When control returns, or if no direct entry routine has been specified, a test is made to determine whether GSP is being used. In the event GSP is not being used, control returns through ILSX3 to the interrupted program without restarting the 2250. If the program is using GSP, the ISSDE field of GCOM is inspected to determine (1) whether a direct entry routine had been used and (2) if so, whether it had it completed processing the attention. If ISSDE indicates completion, the 2250 is restarted and control returns through ILS03 to the interrupted program. An indication that processing has not been completed results in the first word of ATDAT being tested for zero. A non-zero in ATDAT, indicating that the information from the previous attention has not yet been accessed, causes the 2250 to be restarted and control to be returned through ILSX3 to the interrupted program. A zero in ATDAT results in a test of GCOM to determine whether PFKB attentions have been enabled. If PFKB attentions have not been enabled, the display is restarted and control returns through ILSX3 to the interrupted program. If PFKB attentions have been enabled, the indicator bit in PFKB1 and PFKB2 is tested to determine if the depressed key is enabled. If it is, the first word of ATDAT is set to indicate the PFKB code, the PFKB number in the third word, and the overlay number in the fourth word. The remainder of ATDAT is filled with zeros, the display is restarted, and control returns through ILSX3 to the interrupted program. If the depressed key is not enabled, ATDAT is set to zero, the system display is restarted, and control is returned through ILSX3 to the interrupted program.

Order controlled attentions: An order controlled attention is the result of executing a Graphic Interrupt or a Graphic Interrupt Conditional order. This type attention causes GCOM to be tested to determine if a direct entry routine has been specified for the attention. Upon return from the direct entry routine, or if none was specified, GCOM is tested to determine whether the program is using GSP. If GSP is not being used, control returns through ILSX3 to the interrupted program without restarting the display. If GSP is being used, the ISSDE field in GCOM is accessed to determine whether the attention has already been processed by the direct entry routine; in which case the display is restarted and control returns through ILSX3 to the interrupted program. If the attention has not been processed, a branch is made to the correct processing routine. ATDAT is then tested to determine whether previous attention data is waiting to be accessed. Any information in ATDAT causes a branch back to the interrupted program and results in the display being restarted. If ATDAT does not already contain attention data, the SATEN field of GCOM is inspected to see if order controlled attentions have been enabled. If they have, the attention data related to the order controlled attention is placed in ATDAT. The system display is then restarted and control returns through ILSX3 to the interrupted program.

Light pen attentions: An attention from a light pen detect results in the address of the order on which the attention occurred and the address of the next order to be calculated and stored in GCOM. A test is made to determine whether a direct entry routine has been specified for light pen attentions. When control returns from the direct entry routine, or if none was specified, a test is made to determine whether the program is using GSP. Control returns to the calling program if GSP is not being used. If GSP is being used, the ISSDE field in GCOM is inspected to determine whether the attention has been processed by the direct entry routine. If ISSDE is not zero, the value of the word is used as a displacement into a branch table which directs control to pass to the appropriate attention-handling routine. If ISSDE is found to be zero, the first word of ATDAT is tested for zero. A non-zero in this area indicates data from a previous attention is waiting to be processed. The system display is then restarted and control returns through ILSX3 to the interrupted program. If ATDAT is zero, a test is made to determine whether light pen attentions have been enabled by the program. If light pen attentions have not been enabled, the system display is started and control returns through ILSX3 to the interrupted program. If light pen attentions have been enabled, the ATDAT area is filled as shown in Table 2.

Table 2. Attention Data Array

ATDAT	Light Pen Code
+1	A decimal number indicating the position, within an index-controlled entity, of the item detected by the light pen.
+2	The decimal equivalent of the character detected, if applicable, or minus one if the detect was not on a character.
+3	x-beam position.
+4	y-beam position.
+5	corrval of the ICA.
+6	corrval of the image entity.
+7	corrval of the controlled, level controlled, or index controlled entity.
+8	corrval of the innermost named element within the controlled entity or zero if not applicable.
+9	corrval of the lowest level subroutine entity or innermost element within the lowest level subroutine entity; or zero if not applicable.
+10 through +19	Reserved (filled with zeros).

Under certain circumstances, the GSP direct entry routine will present an order controlled attention to DSPYN as a priority light pen attention. In this case, the ATDAT area is filled in without inspection of SATEN or ATDAT. If RQATN was processing at the time of the interrupt, the return address is modified to an entry point in RQATN which permits the new data to be transferred in its entirety.

DSPYN/ISS ERROR ROUTINE: Upon entering the error routine, the information in GCOM resulting from the XIO-Read Status is transferred to an area within the error routine. An error code, the current status of the DSW, and the address from which the error routine was entered are also placed in this area. The error code and the address of the area where the status information is being saved are placed in the accumulator and extension and in index registers 1 and 2. The address following the NSI is stored at $PST3, and a branch is made to $PST3 + 1. This is the address of the post-operative error trap which contains a WAIT instruction. The program then branches to the WAIT. If the program is restarted by the operator pushing the PROGRAM START (or by any other interrupt) control returns to the error routine. The error routine retries the particular I/O command which failed. If the retry is successful, it then resets the 2250 and restarts the display at the last order address used by DSPYN to start regeneration. Control is then returned to the interrupted program. If the retry is unsuccessful, the entire error routine is re-executed.

ATTENTION ENABLING AND INFORMATION RETRIEVAL

Communication between the program and the 2250 operator is provided by the processing of attentions to the 1130 CPU from the light pen and keyboards. The following subroutines are concerned with establishing attention conditions and the retrieval of attention data:

- SATNS, which indicates to the GSP which sources of attentions are to be meaningful to the program.

- RQATN, which transfers attention data from a constants area in DSPYN to a user defined array.

The subroutine ROCOR returns the correlation value and the identity of an element within which a specified element is nested.

SATNS -- Set Attention Status

Chart: EA

Function: To enable attention sources for handling by the problem program and to remove any previously saved attention information.

Entry: SATNS

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

Input:
  Argument 1 - address of the enable code.

Output: None.

Operation: SATNS stores the enable code (argument 1) in the SATEN field in GCOM. If the RQNDA field in GCOM is non-zero, SATNS also sets to zero the first word of the attention data area (ATDAT) of DSPYN to prevent processing of previously saved attention information.

Errors:
  Code 1 - The attention source is invalid.

RQATN -- Request Attention Information

Chart: EB

Function: To transfer attention information to a program-defined array.

Entry: RQATN

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

Input:
  Argument 1 - address of the user's attention array.

Output: Attention information stored in appropriate elements of the attention array; or, if no attention has occurred, the first word of the attention array set to one (see Table 3).

Operation: The contents of RQNDA, the word in GCOM used to access the attention data array (ATDAT) in DSPYN, is placed in the accumulator. If the contents of the word are zero, the first word of the user's array is set to one, and control returns to the calling program.

If RQNDA is non-zero, the address of an entry point in RQATN is stored in RQNDA to prevent priority light pen information from being lost if a priority light pen attention should occur during execution of RQATN (see description of LOCPN/LOCND). This action ensures that control returns to RQATN at a point which will enable RQATN to transfer all the data associated with the priority light pen attention. (See "Light Pen Attentions" in the description of DSPYN.)

The first word of ATDAT is then tested for zero to determine if an attention has occurred since the last call to RQATN. A zero, indicating that no attention data is available, results in a one (1) being placed in the first word of the user's array. The address of ATDAT is then restored in GCOM, an indicator in DSPYN is set to permit DSPYN to accept new Read Status data, and control is returned to the calling program. If there is data in the first word of ATDAT, INPRE in GCOM is examined to determine if the user's array is in one-, two-, or three-word elements. The data is then transferred to the user's array. When all the data has been trans-

Table 3. User's Attention Array for RQATN

Element	Contents
Array(1)	One (1) if no attention occurred; or one of the following attention source codes: Code Source 2 light pen 3 END key 4 alphameric key 5 programmed function key 6 order controlled (assembler language only)
Array(2)	For light pen, a decimal number indicating the position within an index-controlled entity of the item detected by the light pen.
Array(3)	For light pen, the decimal equivalent of the character detected, if applicable; or minus one if the detect was not on a character. For programmed function key, a decimal number in the range 0 to 31 corresponding to the depressed key. For alphameric key, the decimal equivalent of the character entered from the keyboard; or one of the following values: 256 for the JUMP key 512 for the BACKSPACE key 1024 for the ADVANCE key
Array(4)	For light pen, the x-coordinate beam position in 2250 raster units. For programmed function key, a decimal number in the range 0 to 255 corresponding to the overlay.
Array(5)	For light pen, the y-coordinate beam position, in 2250 raster units.
Array(6)	For light pen, the correlation value of the ICA.
Array(7)	For light pen, the correlation value of the image entity.
Array(8)	For light pen, the correlation value of the controlled entity.
Array(9)	For light pen, the correlation value of the innermost named element within the controlled entity in which the light pen detect occurred; or zero if not applicable.
Array(10)	For light pen, the correlation value of the lowest level subroutine entity or the innermost element within the lowest level subroutine entity in which the light pen detect occurred; or zero if not applicable.
Array(11)	For order controlled, the address of the order causing the attention (Assembler-language only).
Array(12)	For order controlled, the address of the next sequential order (Assembler-language only).
Array(13)	Reserved.
. . .	
Array(20)	Reserved.

ferred, an indicator in DSPYN is set to permit DSPYN to accept new Read Status data, the first word in ATDAT is reset to zero, the address of ATDAT is restored in GCOM, and control returns to the calling program.

Errors: None.

ROCOR -- Return Outer Correlation Value

Chart: EC

Function: To obtain the correlation value and type of the element within which the specified element is nested, or the correlation value of a named linkage entity.

Entry: ROCOR

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

Input:
Argument 1 - address of the correlation value of the specified nested element.
Argument 2 - address of an integer variable where the correlation value for the next outer element is to be stored.
Argument 3 - address of an integer variable where the element type is to be stored.

Output: See "Function."

Operation: If the correlation value of the specified nested element is valid, GSP03 finds the ECCB for the nested element. If the ECCB found is for an image entity, or for a subroutine to which linkage does not exist, the variables specified by arguments 2 and 3 are set to zero. If the ECCB found is for a tracking entity, the correlation value and element type code are stored in the specified variables. Otherwise, GSP03 finds the ECCB for the next outer element and stores the correlation value and element type code for the next outer element in the specified variables. The element type codes are as follows:

1=uncontrolled entity
2=controlled entity
3=subroutine entity
4=image entity
5=index-controlled entity
6=level-controlled entity
7=linkage entity
8=tracking entity

Errors:
Code 2 - The correlation value is not in the range 1-32767.
Code 3 - The ECCB for the correlation value is not found by GSP03; or there is no outer element.

LIGHT PEN SUBROUTINES

The following subroutines allow communication between the 2250 operator and the user program for creation and modification of graphic displays by use of the light pen:

- LOCPN, which causes the immediate display of the scanning pattern and allows light pen attentions.
- LOCND, which causes the display of the scanning pattern when the light pen switch is closed, and allows light pen attentions.
- LCPOS, which displays the tracking symbol at a specified position.
- TRACK, which displays the tracking and allows tracking on the 2250 screen.
- CTLTK, which specifies non-default tracking characteristics for light pen tracking.
- CVTTD, which converts x- and y-coordinates to program units.

LOCPN/LOCND -- Locate Position of the Light Pen and Locate Position of the Light Pen on No Detect

Chart: FA

Function: To display the scanning pattern on the screen of the 2250.

Entry:
1. LOCPN
2. LOCND

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

Input: None

Output: Displays the scanning pattern on the screen of the 2250.

Operation:

Entry through LOCPN activates a branch (SD3) in the system display so that the scanning pattern is displayed immediately. Entry through LOCND activates a conditional branch (SD6) in the system display so that the scanning pattern is displayed only when the light pen switch is closed.

Within the system display, the branch to the active image entity is located before the conditional branch to the scanning pattern. The conditional branch to the scanning pattern is ignored as long as the light pen switch remains open. If the light pen switch is closed and a detect is made on the image entity, a normal attention occurs and the conditional branch to the scanning pattern is not executed. If the light pen switch is closed and no detect has occured, the conditional branch to the scanning pattern is executed causing the scanning pattern to be displayed.

Errors:
Code 9 - The subroutine is called while TRACK, LOCPN, LOCND, or LCPOS is executing.
Code 10 - There is no image entity currently being displayed.

LCPOS -- Locate a Position with the Tracking Symbol

Chart: FB

Function: To display the tracking symbol on the screen of the 2250 for use in conjunction with CTLTK.

Entry: LCPOS

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 x-coordinate
Argument 3 - address of the y-coordinate

Output: See "Operation."

Operation: GSP02 is passed the coordinates given in the calling sequence for conversion to raster units. Control returns to LCPOS, and the new absolute coordinates are used to position the tracking symbol. The branch in the system display to the tracking symbol is then activated to display the tracking symbol on the screen.

Errors:
Code 9 - The subroutine was called while TRACK, LOCPN, LOCND, or LCPOS was executing.
Code 10 - There is no active image entity.

TRACK -- Track Position of Light Pen

Chart: FC

Function: To display the tracking symbol and create a tracking entity.

Entry: TRACK

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 x-coordinate.
Argument 3 - address of the y-coordinate.
Argument 4 - address of the correlation value of the tracking entity.

Output: See "Function."

Operation: TRACK ensures that an image entity is being displayed, and calls GSP04 (Push-down) to provide a minimum of 10 words in the ICA. GSP03 then builds an ECCB for a tracking entity. The tracking output mode is set to the output graphic mode defined in the GCA, the line option is specified, and the distance used for curve tracking is set to 30 raster units. GSP02 scales the x- and y-coordinates specified in arguments 2 and 3. Overhead orders to move the beam to specified coordinates and to link to the tracking entity are stored in the image entity. Overhead orders (GB, SRVT, and GBI) are stored in the tracking entity and DSPIN in GCOM is set to indicate that TRACK was entered. The address of the tracking symbol (GSP09) is stored in the Graphic Branch (SD4) in system display in GCOM. The tracking direct entry subroutine (TRDE in GSP09) is initialized to allow movement of the tracking symbol in all axes, to assume move and vector modes, and to indicate that the last fixed position is specified by the x- and y-coordinates in arguments 2 and 3.

Errors:
Code 2 - The correlation value is not in the range 1-32767.
Code 5 - There are not 10 words available in the ICA.
Code 9 - DSPIN indicates that TRACK, LOCPN, LOCND, or LCPOS has been called and the function is not yet completed.
Code 10 - There is no image entity being displayed.

CTLTK -- Control Light Pen Tracking

Charts: FD,FE

Function: To set the tracking attributes and to permit the user to specify and to retrieve the position of the tracking symbol. CTLTK also permits a subroutine entity to be connected to or disconnected from the tracking entity.

Entry: CTLTK

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

Input:

Argument 1 - Address of the track mode option.
Argument 2 - Address of the axis option.
Argument 3 - Address of the distance option.
Argument 4 - Address of the point/line option.
Argument 5 - Address of the array containing the coordinates of the tracking symbol or containing the increments to be added to the current position of the tracking symbol.
Argument 6 - Address of the correlation value of a subroutine entity.

Output: See "Operation."

Operation: CTLTK tests the arguments which are passed and takes the following action:

1. A correlation value of zero results in no action being taken. A negative correlation value causes the subroutine entity to be disconnected from the tracking symbol. A positive value causes the subroutine entity specified by the correlation value to be attached to the tracking symbol.

2. A distance argument of zero results in no action being taken. An argument within the limits of 1 through 63 is stored in GCOM to be used by GSP09 in setting points as the 2250 operator is tracking.

3. A point/line option of zero results in no action being taken. An argument of 1 or 2 is stored in GCOM to indicate that lines or points are being generated.

4. A track mode argument of zero results in no change taking place. Arguments from 1 through 3 indicate positioning, sketching, or rubber-banding, respectively, and the argument is stored in GCOM. Arguments of 4 through 6 indicate programmed drawing, erasing, and the end function resulting in values of 0, 1, and 2 to be placed in the WORK1 field in CTLTK.

5. An axis argument of zero results in no change from the previous axial characteristic. Arguments from 1 through 5 are stored in GSP09 for use during movement of the tracking symbol.

If programmed drawing has been indicated, the increments from the user-defined array are added to the current position of the tracking symbol. If programmed drawing was not indicated the current position of the tracking symbol is stored in the same user-defined array.

In cases where rubber-band or sketch mode is in effect, or was in effect during the previous call to CTLTK, orders which were generated in the temporary area of GSP09 are stored in the tracking entity. Two exceptions to this rule are noted in the following paragraph. Depending upon the mode the program is operating in, either absolute or incremental orders are stored in the tracking entity.

To allow the program to call CTLTK while in rubber-band or sketch mode and obtain the coordinates of the tracking symbol without fixing any points in the tracking entity, CTLTK inspects certain arguments before storing orders in the ICA. If the program is in rubber-band mode, CTLTK tests the track mode and point/line arguments and, if zero, does not store orders in the ICA. If the program is sketching, in addition to testing the track mode and point/line arguments, CTLTK also tests the axis and distance arguments. Values of zero in all arguments cause no orders to be stored in the ICA. If positioning is specified in two successive calls to CTLTK, no orders are stored in the tracking entity.

All calls to CTLTK cause a test to be made to determine whether the end function has been requested. If the end function has not been requested, the appropriate line/point mode order is set in the tracking entity and the skeleton move or draw orders are reset in the temporary area of GSP09. A test is then made to determine if the erase function has been specified and if so, a graphic branch within GSP09 is set to inhibit the display of the tracking symbol.

If the end function has been requested, the linkage to the tracking symbol in the system display is removed, the branch in the image entity to the temporary order area in GSP09 is nullified, the tracking entity ECCB is indicated as ended, and the element end address is set in the ECCB.

Errors:

Code 1 - An invalid argument (argument 1 - 5) has been passed.
Code 3 - The correlation value is undefined.
Code 4 - The correlation value is not for a subroutine entity.
Code 5 - The ICA is full.
Code 9 - The TRACK or LOCPN subroutine

had not been called before the call to CTLTK.

Code 11 - GSPIN has not yet been called.

CVTTD -- Convert Tracking Data

Chart: FF

Function: To convert the x- and y-coordinates in a tracking entity, origin entity, attention data array, or array used for RDPOS or CTLTK to program coordinates.

Entry: CVTTD

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

Input:

Argument 1 - address of the GCA.

Argument 2 - address of the correlation value or the input array.

Argument 3 - address of the variable or array for the x-coordinate output data.

Argument 4 - address of the variable or array for the y-coordinate output data.

Argument 5 - address of the number of elements in each output array.

Argument 6 - address of the variable in which the number of converted coordinates is stored.

Argument 7 - address of the variable in which the address of the first coordinate not converted is stored if the data type changes.

Argument 8 - address of the variable indicating the type of argument given in argument 2, and the type of data returned in arguments 3 and 4.

Output: The x- and y-coordinate data raster units is converted to program units and stored in the arrays specified by arguments 3 and 4. The number of converted coordinates is stored in argument 6. If applicable, the address of the first coordinate not converted is stored in argument 7. A return code is stored in argument 8 indicating the type of data stored.

Operation: If the data which is being converted is associated with a tracking or origin entity, GSP03 is entered to search for the associated ECCB. The x- and y-coordinates are then converted to program units using the scaling factors in the GCA. The results are stored in the arrays specified by the user. If the end of the element has not been reached, the next order is tested to determine whether the line/point mode or move/draw mode has changed. If no change is indicated, the x- and y-coordinates are converted as described above. A change in mode results in the number of converted coordinates, the address of the first coordinate not converted, and a code indicating the type of data returned, being stored in the addresses specified in arguments 6, 7, and 8 respectively. Control then returns to the calling program.

When the attention data array or an x and y array is specified, the x- and y-coordinates are converted to program units and stored in the user-specified array. A 1 is stored in the address specified by argument 6, a code indicating the type of data returned is stored in argument 8, and control returns to the calling program.

Errors:

Code 1 - An argument, not the correlation value, is invalid.

Code 2 - The correlation value is invalid.

Code 3 - The specified correlation value is not currently defined.

Code 4 - The specified correlation value is not for an origin or tracking entity.

Code 6 - The ICA is full.

ALPHAMERIC AND TEXT GENERATION SUBROUTINES

The following subroutines provide for the creation of message entities from the alphameric keyboard or program data, for the conversion of graphic orders in a message entity to EBCDIC code, and for conversion of real or integer values to EBCDIC and vice versa:

- GCHAR, which is a series of Graphic Short Branch orders and character stroke subroutines which will cause the display of upper and lower case letters, digits, and certain special characters.

- GUPER, which provides the same function as GCHAR except that lower case letters are not available.

- DFMSG, which defines a message entity.

- MSGIN, which converts alphameric input data to Graphic Short Branch orders and stores these orders in the message entity.

- TLMSG, which converts Graphic Short Branch orders in a message entity into EBCDIC code, and stores them in a user defined array.

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

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

Operation: DFMSG tests the validity of the four arguments, and GSP03 searches for the ECCB specified by argument 1 (correlation 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 of 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 found by GSP03 is not for a message entity.
- Code 8 - GCHAR or GUPER is not loaded.

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.

Argument 4 - address of the number of characters in the data array.
Argument 5 - address of the text code.

Output: See "Operation."

Operation: MSGIN ensures the validity of the arguments, and GSP03 finds the ECCB for the specified message entity. NAGAP and LAGAP in GCOM are set to reflect the limits in the message entity within which the GSB orders will be stored. GSP02 generates and stores in the message entity the GSB orders for the data in the array.

Errors:
Code 1 - Argument 2, 3, or 4 is not in the range 1-32767; or argument 5 is not in the range 1-4.
Code 2 - The correlation value is not in the range 1-32767.
Code 3 - The ECCB is not found by GSP03.
Code 4 - The ECCB found is not for a message entity.

ICURS -- Insert Cursor

Chart: GC

Function: To place a cursor in a message entity and display the cursor on the 2250 screen.

Entry: ICURS

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 position in the message entity where the cursor is to be placed.

Output: None.

Operation: ICURS calls DSPYN to remove the direct entry address of GSP07 from GCOM. The correlation value is checked for validity, and GSP03 finds the specified ECCB. If the ECCB is for an image entity being displayed, or for a subroutine entity, the character position is checked to ensure that it is greater than zero.

If there is an active cursor, it is removed and the graphic short branch (GSB) for the null cursor (used to indicate the last cursor position) is inserted in the cursor address in GCOM. A GSB for the null cursor is interchanged with GSBs for the character stroke subroutines in succeeding positions until the position identified by argument 3 is reached. The GSB for the cursor is inserted here, and the cursor address in GCOM is updated to reflect the new position. The active cursor indicator in GCOM is set on, and DSPYN allows the use of GSP07.

Errors:
Code 1 - The address of the position where the cursor is to be placed is not within the limits of the message entity.
Code 2 - The correlation value is not in the range 1-32767.
Code 3 - The ECCB for the specified element is not found by GSP03.
Code 4 - The ECCB found by GSP03 is not for a message entity.
Code 9 - The message entity is not within the image entity being displayed.
Code 10 - The message entity is not within a previously defined element.

RCURS -- Remove Cursor

Chart: GD

Function: To remove the cursor from a message entity.

Entry: RCURS

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

Input: None.

Output: None.

Operation: RCURS calls DSPYN to remove the address of GSP07 from GCOM. If there is no cursor displayed on the screen, no further processing occurs. Otherwise, the Graphic Short Branch (GSB) of the null cursor (used to indicate the last cursor position) is stored in CURAD in GCOM, and CMECB in GCOM is loaded with zeros.

Errors: None.

TLMSG -- Translate Message Entity

Charts: GE,GF

Function: To convert alphameric data associated with a message entity from graphic representation to EBCDIC format and place the translated data in a specified area.

Entry: TLMSG

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 variable or array where the translated data will be stored.
- Argument 3 - address of the element count.
- Argument 4 - address of a code (text code) designating the format of the alphameric data.

Output: See "Function."

Operation: TLMSG ensures the validity of the arguments, and the search function of GSP03 finds the specified message entity. The data in the message entity is then translated character by character into EBCDIC values and stored in the specified array. A counter, which indicates the number of words per element, is set to 1, 2, or 3, depending on the text code and precision (see Table 4). As each word is stored, the word counter is decremented by one. When the counter reaches zero, one element has been filled and the element count (argument 3) is decremented by one. The array address is decremented by an index factor (determined by text code and precision) and the word counter is reset.

If the element count is zero and the message entity has not been fully translated, an error condition exists. If the message entity has been fully translated and the element count is not zero, the remaining area in the storage array is filled with EBCDIC blanks.

Errors:
- Code 1 - The text code is not in the range 1-4.
- Code 2 - The correlation value is not in the range 1-32767.
- Code 3 - The ECCB is not found by GSP03.
- Code 4 - The ECCB found by GSP03 is not for a message entity.
- Code 12 - The element count has been decremented to zero but the message entity has not been fully translated.

Note: All characters shown are EBCDIC values except those underlined, which are hexadecimal values. One element, that in high order position within the message entity, is shown for each valid combination. Each word is separated by brackets. "xx" denotes information that is not meaningful.

BCNV -- Basic Conversion

Chart: GG

Function: To provide the following types of conversion:

1. EBCDIC to real precision.
2. Real precision to EBCDIC.
3. EBCDIC to integer precision.
4. Integer precision to EBCDIC.

Entry: BCNV

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

Input:
- Argument 1 - address of the input array (data to be converted).
- Argument 2 - address of the output array

Table 4. Alphameric Data Format

Precision	Text Code			
	1	2	3	4
Real Standard	[AB] [CD]			
Real Extended	[AB] [CD] [EF]			
Integer One-Word		[AB]	[00A]	[0A]
Integer Standard		[AB] [xx]	[00A] [xx]	[0A] [xx]
Integer Extended		[AB] [xx] [xx]	[00A] [xx] [xx]	[0A] [xx] [xx]

(where converted data is to be stored).
Argument 3 - address of the code specifying the type of conversion.
Argument 4 - address of the text code specifying the data format.

Output: Converted data stored in the output array.

Operation: BCNV ensures the validity of the arguments and examines the code indicating the type of conversion to be performed. If the conversion is from EBCDIC to real precision, the input data is first examined for validity and then converted to a format acceptable to the 1130 subroutine FDTB (one EBCDIC character per word). FDTB then converts the data to real precision which is stored in the output array specified by argument 2.

If the conversion is from real precision to EBCDIC, the 1130 subroutine FBTD converts the data which is then stored in the output array.

If the conversion is from EBCDIC to integer, the contents of the input array are examined for validity and then converted to integer values which are stored in the output array.

If the conversion is from integer to EBCDIC, the integer value is converted to the decimal equivalent and stored in the output array.

Errors:
Code 1 - The conversion code or text code is invalid.
Code 12 - The contents of the input array are invalid.

ATTENTION PROCESSING

The following subroutines accomplish the processing necessary for communication between the program and the 2250 operator, by interpreting interrupts from the 2250:

- DSPYN, which is described under "I/O Subroutines."
- GSP07, which processes attentions from the alphameric keyboard if in message collection mode, and prevents the saving of attention data.
- GSP08, which processes light pen and order controlled interrupts.
- GSP09, which processes order controlled interrupts originating in the scanning pattern and the tracking symbol.
- GSP11, which processes light pen detects upon an index-controlled entity.
- GSP12, which processes light pen detects and attentions upon a level-controlled entity.

GSP07 -- Cursor Direct Entry

Chart: HA

Function: To accept a character from DSPYN and insert a Graphic Short Branch (GSB) to its stroke subroutine into the message entity at the location specified by the cursor; or, to control the positioning of the cursor within the message entities.

Entry: GSP07 from DSPYN

Exit: Returns through the entry word to the next sequential instruction in the 2250 ISS.

Input: The results of an alphameric keyboard attention.

Output: A GSB to a character stroke subroutine placed in the message entity.

Operation: ATNAN in GCOM is tested to determine if the attention is caused by the JUMP, ADVANCE, or BACKSPACE key. If not, the GSB for the character whose EBCDIC value is also stored in ATNAN is placed in the message entity at the location specified by the cursor address (CURAD) in GCOM. If the GSB for the cursor is not at the end of the element, it is stored in the next position in the element, and CURAD is updated. If the GSB for the cursor is at the end of the element, no further processing occurs.

If the attention is caused by the ADVANCE key, the position of the GSB for the cursor within the element is determined. If it is at the end of the element, no further processing occurs. If it is not at the end of the element, the GSB for the cursor is interchanged with the following GSB, and CURAD is updated.

If the attention is caused by the BACKSPACE key, the position of the GSB for the cursor within the element is determined. If it is in the first position after the overhead order, no further processing occurs. If it is not in the first position after the overhead order, the GSB for the cursor is interchanged with the preceding GSB, and CURAD is updated.

If the JUMP key is the cause of the attention, the next message entity within the image entity is found, the GSB for the cursor is stored in the first position after the overhead order, and the GSB to the cursor in the old message entity is replaced with a GSB to the special null cursor. If there is no other message entity within the image entity, GSP07 treats the old message entity as the new, and places the GSB for the cursor in the first position after the overhead order. GSP07 stores a code in ISSDE in GCOM to indicate that GSP07 has completed all processing of the alphameric keyboard attention.

Errors: None.

GSP08 -- Light Pen and Order Controlled Direct Entry Subroutine

Chart: HB

Function: To process light pen and to route order controlled interrupts to GSP09 or GSP12.

Entry: GSP08

Exit: Returns to DSPYN through the address stored in the entry word.

Input: None.

Output: See "Operation."

Operation: When GSP08 is entered, it determines whether the interrupt was caused by a 2250 order or the light pen.

If an order interrupt has occurred, GSP08 does the following:

1. If the order interrupt occurred in the tracking symbol subroutine, control passes to the Track Direct Entry subroutine (TRKDE in GSP09). This subroutine updates the position of the tracking symbol, generates orders in the ICA if necessary, and returns control to GSP08. GSP08 passes control back to DSPYN where the system display is restarted.

2. If the order interrupt occurred in the scanning pattern, control passes to the Scan Direct Entry subroutine (SCNDE in SGP09). This subroutine sets the Graphic Branch in the system display to the scanning pattern subroutine to perform no-operation. Control returns to GSP08, which in turn passes control back to DSPYN. DSPYN then restarts the system display.

3. If the order interrupt occurred within a level-controlled entity, control passes to GSP12 to determine the element in which it occurred. GSP12 resets OADDR and NOADR in GCOM. Control returns to DSPYN where the attention data array is filled.

4. If the order interrupt did not occur in the tracking symbol or the scanning pattern subroutine, or within a level controlled entity (see description of GSP12), an indicator in GCOM is set to indicate that the display is to be terminated. Control returns to DSPYN. The display is not restarted. Attention data is placed in an array which may be accessed by the user through a call to RQATN.

If a light pen interrupt has occurred a test is made to determine if the light pen switch has been closed. If the switch has not been closed, the interrupt is ignored and control returns to DSPYN to restart the display at the next sequential order. If the light pen switch is closed, a test is made to determine whether or not an image entity is being displayed. If an image entity is being displayed, and the attention was from the tracking entity, the beam bit is turned off, or on a character a GSB to blank is used. If not on a tracking entity, a correlation value search is made to provide the attention information for DSPYN. If the detect occurred on an index-controlled entity, control passes to GSP11 to determine the relative position of the detected element within the entity. Attention information is provided for DSPYN and control is passed to DSPYN. If an image entity was not being displayed, it indicates that the light pen detect occurred in a user generated order string. Correlation data in GCOM is set to zero. The system display is reset and an indicator in GCOM is set to indicate that the display is terminated. Control returns to DSPYN where attention data is placed in an array that is available to the user through a call to RQATN.

Errors: None.

GSP09 -- Scanning Pattern and Tracking Symbol Direct Entry Subroutine

Chart: HC

Function:

1. To replace the scanning pattern with the tracking symbol when a detect occurs on the scanning pattern.

2. To process interrupts on the tracking symbol and update the position of the tracking symbol after calls to LOCPN, LOCND, LCPOS and TRACK.

Entry:
1. SCNDE
2. TRKDE

Exit: Returns to DSPYN through GSP08.

Input: None.

Output: See "Operation."

Operation:

1. An interrupt on the scanning pattern causes control to pass to GSP09 through the SCNDE entry point. The Graphic Branch to the scanning pattern subroutine is then set to perform no-operation, and the Graphic Branch to the tracking symbol is made operational.

2. Following the first interrupt on the tracking symbol, GSP09 is entered at the TRKDE entry point. Each section of the tracking symbol is assigned a specific value, and TRKDE uses this value to analyze the distance the light pen is from the center of the tracking symbol. An algorithm is then used to determine the number of raster units in both the x- and y-direction the tracking symbol must be moved. The results of this calculation are examined to determine if the tracking symbol is to be moved at all. (A result of x = 0, y = 0 would indicate no move in either direction.) If the tracking symbol is to be moved, the STMR order in the system display is set to perform no-operation. This permits faster updating of the tracking symbol. The Move Beam Absolute order which positions the tracking symbol is then updated to reflect the new position. If it has been determined that the tracking symbol does not have to be moved, the STMR order in the system display is immediately restored. If LCPOS has been called, control passes back to DSPYN. If not, the distance the light pen is from the center of the tracking symbol is computed in the normal manner. Before the tracking symbol is moved, however, a test is made to insure that the tracking symbol is free to move in any direction. If movement in any specific direction has been restricted by CTLTK, the new calculations for adjusting the tracking symbol are made accordingly. A test is made to determine whether the tracking symbol is to be moved at all. If it is, the STMR order is set to perform no-operation, and the Move Beam Absolute order is updated to reposition the tracking symbol. A move of three or more raster units from the last position causes the 2250 ISS to restart the display at the tracking symbol. Tem porarily bypassing the orders which display the image entity permits faster tracking and minimizes the chance of the light pen's losing the tracking symbol. The ICA is tested to insure that at least four unused words are in the ICA for newly generated orders. Less than four words in the ICA causes the letter F to appear on top of the tracking symbol to notify the 2250 operator that no more orders will be generated. Control then returns to DSPYN. If the ICA is not full and sketching has been specified, each time the "dist" argument is reached the orders are stored in the ICA. If the symbol has not moved the user-specified distance, the orders are stored in a temporary area. In any other mode but sketching, the orders are stored in the temporary area. Control then returns to DSPYN.

Errors: None.

GSP11 -- Scan Index-Controlled Entity

Chart: HD

Function: To determine the number of non-overhead graphic orders in an index-controlled entity from the first order through the order upon which a detect has occurred.

Entry: GSP11 from GSP08

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

Input: None.

Output: See "Operation."

Operation: GSP11 examines each non-overhead order in the index-controlled entity. A counter is incremented by one for each Graphic Short Branch to a character stroke subroutine (except branches to the cursor or cursor null), indirect graphic branch (linkage to a subroutine, but not the orders within), no-op filler (SPM /F stored by GSP02 for scissored data), and incremental and absolute orders with the beam bit on. After each increment of the counter, the address of the examined order is compared with the address of the detected order. If the addresses are the same, processing is completed and the contents of the counter are stored in the field IDXCT in GCOM. If the addresses are not the same, the next graphic order is examined as above. Graphic branches around nested subroutine entities are not counted.

Errors: None.

GSP12 -- Verification Direct Entry

Chart: None.

Function: To determine the current and next order addresses upon an order-controlled interrupt in a level-controlled entity. GSP12 also contains the verification, detect, and non-detect order programs attached to level-controlled entities. These order programs (1) brighten a level-controlled entity upon which a light pen detect occurs, (2) enable light pen attentions, or (3) disable light pen attentions.

Entry: GSP12 from GSP08

Exit: Returns through the address stored in the entry word to the next sequential instruction in GSP08.

Input: None.

Output: None.

Operation: GSP12 obtains from the operand field of the last order executed (a Graphic Interrupt Conditional in a verification subroutine in GSP12) an index value to determine the address of the verification subroutine. Using the operand field of the SVRT order in the verification subroutine, the address of the next two orders in the level-controlled entity are determined and stored in the OADDR and NOADR field in GCOM.

Comments: Within GSP12 are eighteen order subroutines (six levels for each of three detectability options). Figure 18 shows the interaction of a level-controlled entity with a verification subroutine. The first section of the subroutine (left bracket) is branched to by the second overhead order in the level-controlled entity. This left bracket subroutine enables light pen detects on the level-controlled entity. Upon return from the left bracket subroutine, the graphic orders are executed beginning with the SRVT order. The end overhead order in the level-controlled entity branches to the second section (right bracket) of the verification subroutine. This right bracket subroutine causes the execution of the graphic orders in the level-controlled entity three more times if a light pen detect occurs on the entity. An order controlled interrupt occurs if the light pen switch is closed after the three additional executions of the graphic orders. The total of four executions of the graphic orders in the level-controlled entity before the re-execution of the start timer instruction in the system display cause the image created by the level-controlled entity to appear brighter on the 2250 screen.

ERROR HANDLING SUBROUTINES

The subroutine ERRIN permits the use of a user-provided error processing subroutine or the error processing subroutine IERRS, which causes a print-out on the principle print device of error information upon encountering an error condition within a GSP subroutine. Also see "GSP01 - Error Function."

ERRIN -- Error Initialization

Chart: None.

```
    GB N  ELEND+1

    GSB   LBV1                LBV1   SRVT             Save return address
                                     SPM /9           Enable light pen detects
 A  SRVT                             GB I LBV1+1      Branch to level-controlled
     .                                                entity
     .
     .
    Graphic Orders
     .
     .
     .
 B  GB I  A+1                        GBC I LBV1+1,D   Branch to level-controlled
                                     GBC I LBV1+1,D   entity on light pen detect
                                     GBC I LBV1+1,D
                                     GIC   1,DS       Light pen attention
                                     GBC   *,D        Reset detect latch
                                     SPM /4           Disable detects
                                     RVT              Branch to B+2
```

Figure 18. Verification Subroutine Usage

Function: To store in GCOM the address of 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: None.

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:
Argument 1 - an indicator word in the calling sequence.
Argument 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
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 LGXI 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. LAGAP, 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:

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

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.

This section is presented as an aid to those involved in examining core dumps of areas of the GSP. Figure 19 shows the core layout during execution of a GSP program. A sample GSP program is presented with dumps of the program taken before initialization and after each significant block of processing. These dumps reflect the sequential changes in the control blocks, communications areas, and data areas resulting from the processing. Comparison of the significant areas of the sample dumps with a core dump of a user's program should provide a starting point for determining the cause of the problem.

The sample program shown in Figure 20 displays a rectangle which, when the object of a light pen attention, will be redisplayed with a diamond appearing within it. The following areas contain data which is used in creating the graphic elements:

GCA, the generation control area.

ICA, the image construction area.

IATN, the array which will contain attention information.

XS, YS, XT, YT, the arrays which contain the absolute x- and y-coordinate positions used in creating the graphic orders.

GSPIN, ICAIN, and GCAIN are called to initialize GCOM, the ICA, and the GCA with standard values. The subroutines BELMT, IDPOS, PLINE, EELMT create the subroutine entity which contains graphic orders to draw a diamond. The subroutine BELMT creates an image entity within which the controlled entity created by the subroutines BELMT, MVPOS, PLINE, and EELMT is nested. The controlled entity contains graphic orders to draw a rectangle. The subroutine SATNS enables the GSP attention processing subroutine DSPYN to return attention information when a light pen attention occurs on the controlled entity (rectangle). EXEC causes the display of the rectangle. A light pen attention upon the rectangle causes the processing of the subroutines XELMT, MVPOS, LKSUB and EELMT which extend the image entity to include linkage to the subroutine entity. The second call to EXEC causes the display of the rectangle with the diamond appearing within it.

The core dumps shown are taken at the following points in the program:

PAUSE 0001 - after the FORTRAN program has been compiled but before the call to initialize the GSP.

PAUSE 0002 - after the processing of the three initialization subroutines.

PAUSE 0003 - after the graphic elements have been built, light pen attentions enabled, and the rectangle displayed.

PAUSE 0003 (second execution) - after the subroutine entity has been linked to the extended image entity, and the rectangle and diamond have been displayed.

This section of the PLM is formatted as a running commentary of the core storage dumps. It should be noted that reference to core locations and specific values apply to this sample program only.

PAUSE 0001

Figure 21 shows the contents of the areas of storage which have been defined in the FORTRAN program by the DIMENSION statement. At this time the array containing the input data is the only area which contains valid information. This is the result of the FORTRAN compilation; the remaining areas are initialized by the GSP initialization subroutines.

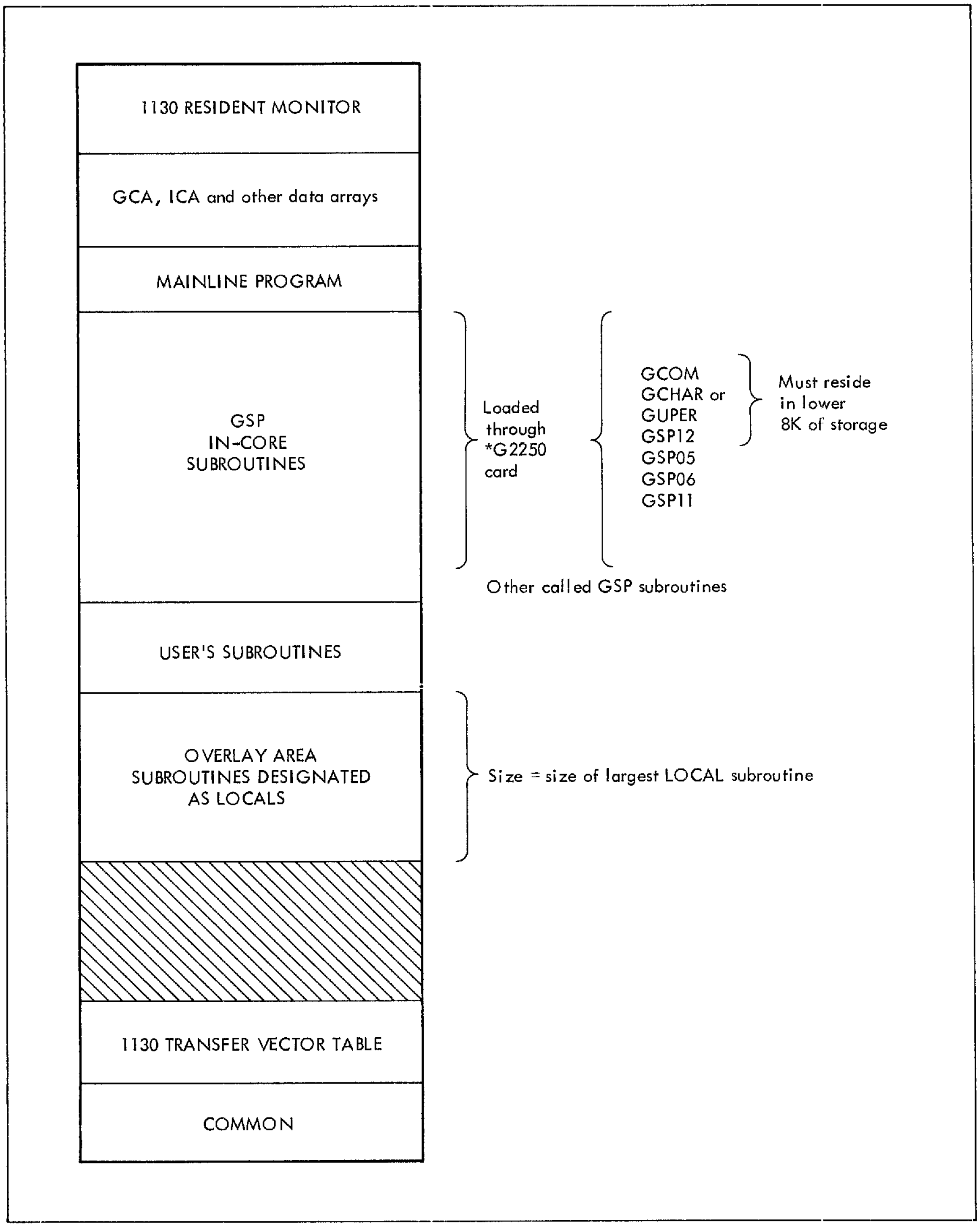
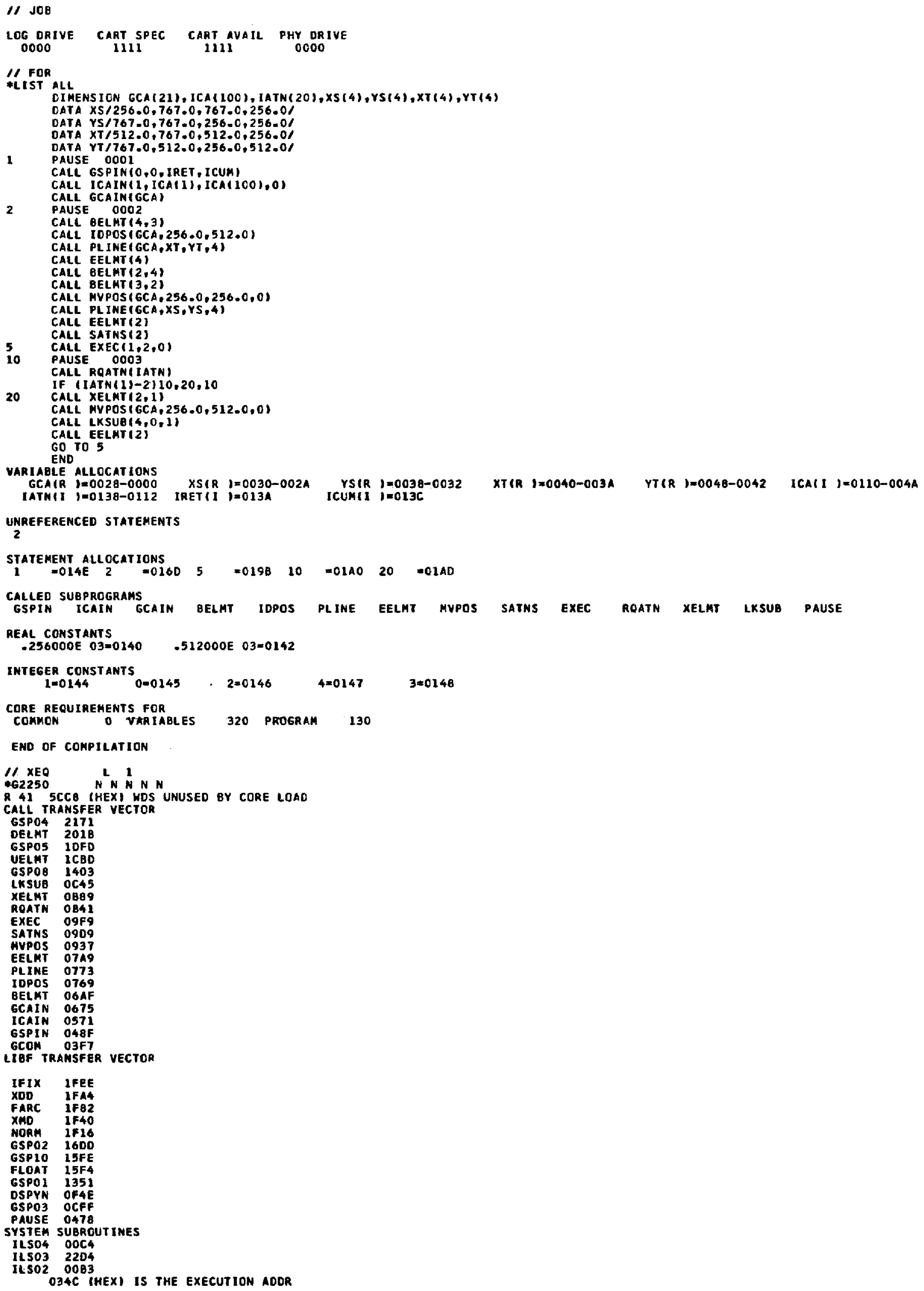
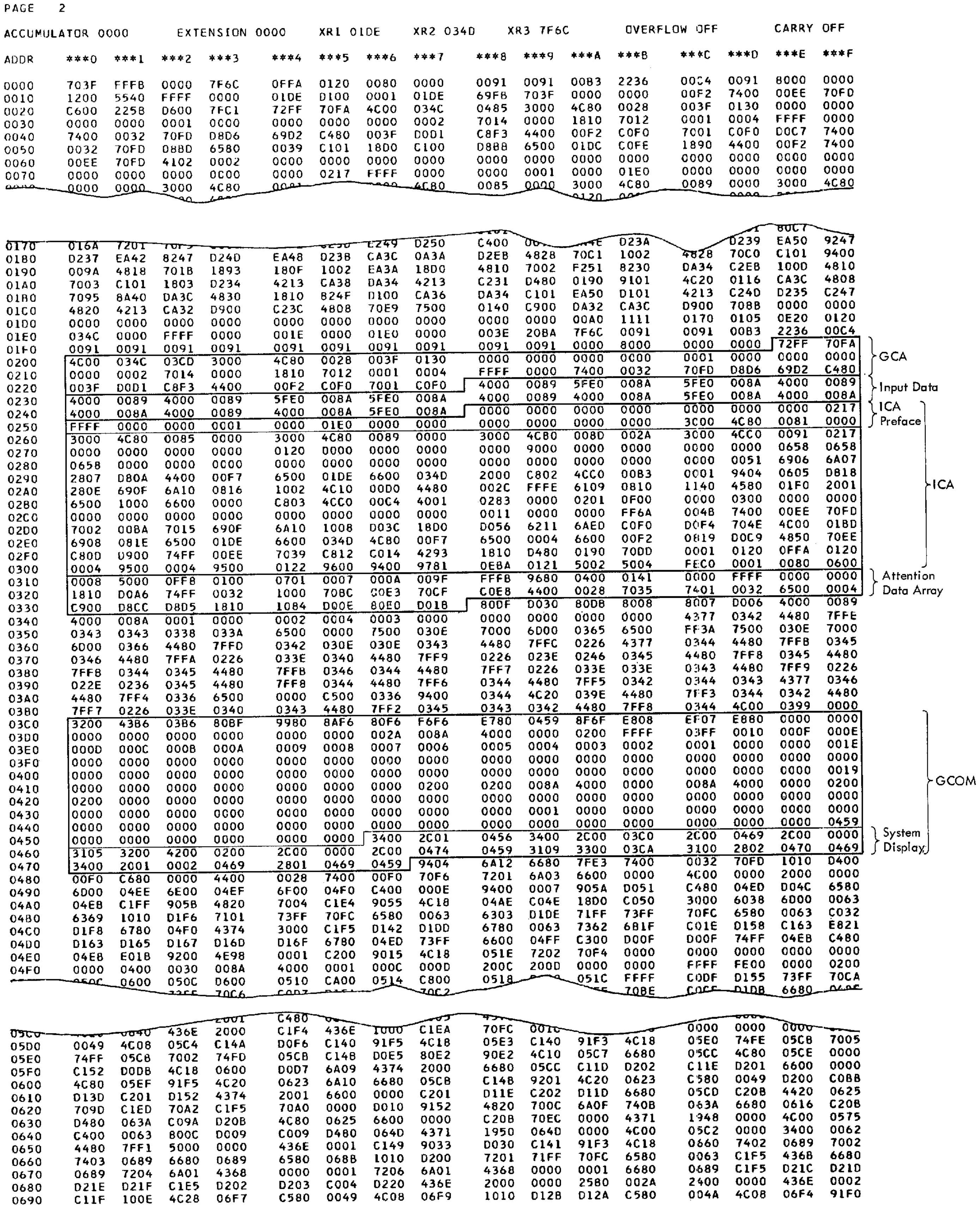

1130 RESIDENT MONITOR
GCA, ICA and other data arrays
MAINLINE PROGRAM
GSP IN-CORE SUBROUTINES
USER'S SUBROUTINES
OVERLAY AREA SUBROUTINES DESIGNATED AS LOCALS
1130 TRANSFER VECTOR TABLE
COMMON
Loaded through *G2250 card
GCOM
GCHAR or
GUPER
GSP12
GSP05
GSP06
GSP11
Must reside in lower 8K of storage
Other called GSP subroutines
Size = size of largest LOCAL subroutine

Figure 19. 1130/2250 Main Storage Map



```
// JOB

LOG DRIVE   CART SPEC   CART AVAIL  PHY DRIVE
  0000        1111        1111        0000

// FOR
*LIST ALL
      DIMENSION GCA(21),ICA(100),IATN(20),XS(4),YS(4),XT(4),YT(4)
      DATA XS/256.0,767.0,767.0,256.0/
      DATA YS/767.0,767.0,256.0,256.0/
      DATA XT/512.0,767.0,512.0,256.0/
      DATA YT/767.0,512.0,256.0,512.0/
1     PAUSE  0001
      CALL GSPIN(0,0,IRET,ICUM)
      CALL ICAIN(1,ICA(1),ICA(100),0)
      CALL GCAIN(GCA)
2     PAUSE   0002
      CALL BELMT(4,3)
      CALL IDPOS(GCA,256.0,512.0)
      CALL PLINE(GCA,XT,YT,4)
      CALL EELMT(4)
      CALL BELMT(2,4)
      CALL BELMT(3,2)
      CALL MVPOS(GCA,256.0,256.0,0)
      CALL PLINE(GCA,XS,YS,4)
      CALL EELMT(2)
      CALL SATNS(2)
5     CALL EXEC(1,2,0)
10    PAUSE   0003
      CALL RQATN(IATN)
      IF (IATN(1)-2)10,20,10
20    CALL XELMT(2,1)
      CALL MVPOS(GCA,256.0,512.0,0)
      CALL LKSUB(4,0,1)
      CALL EELMT(2)
      GO TO 5
      END
VARIABLE ALLOCATIONS
   GCA(R )=0028-0000    XS(R )=0030-002A    YS(R )=0038-0032    XT(R )=0040-003A    YT(R )=0048-0042   ICA(I )=0110-004A
  IATN(I )=0138-0112  IRET(I )=013A       ICUM(I )=013C

UNREFERENCED STATEMENTS
 2

STATEMENT ALLOCATIONS
 1    =014E  2    =016D  5    =019B  10   =01A0  20   =01AD

CALLED SUBPROGRAMS
 GSPIN   ICAIN   GCAIN   BELMT   IDPOS   PLINE   EELMT   MVPOS   SATNS   EXEC    RQATN   XELMT   LKSUB   PAUSE

REAL CONSTANTS
  .256000E 03=0140    .512000E 03=0142

INTEGER CONSTANTS
     1=0144       0=0145    . 2=0146       4=0147       3=0148

CORE REQUIREMENTS FOR
 COMMON      0  VARIABLES    320  PROGRAM    130

 END OF COMPILATION

// XEQ        L  1
*G2250       N N N N N
R 41  5CC8 (HEX) WDS UNUSED BY CORE LOAD
CALL TRANSFER VECTOR
 GSP04  2171
 DELMT  201B
 GSP05  1DFD
 UELMT  1CBD
 GSP08  1403
 LKSUB  0C45
 XELMT  0B89
 RQATN  0B41
 EXEC   09F9
 SATNS  09D9
 MVPOS  0937
 EELMT  07A9
 PLINE  0773
 IDPOS  0769
 BELMT  06AF
 GCAIN  0675
 ICAIN  0571
 GSPIN  048F
 GCOM   03F7
LIBF TRANSFER VECTOR

 IFIX   1FEE
 XDD    1FA4
 FARC   1F82
 XMD    1F40
 NORM   1F16
 GSP02  16DD
 GSP10  15FE
 FLOAT  15F4
 GSP01  1351
 DSPYN  0F4E
 GSP03  0CFF
 PAUSE  0478
SYSTEM SUBROUTINES
 ILS04  00C4
 ILS03  22D4
 ILS02  00B3
      034C (HEX) IS THE EXECUTION ADDR
```

Figure 20. Sample Program



```
PAGE    2

ACCUMULATOR 0000       EXTENSION 0000     XR1 01DE     XR2 034D     XR3 7F6C       OVERFLOW OFF        CARRY OFF

ADDR     ***0  ***1  ***2  ***3    ***4  ***5  ***6  ***7    ***8  ***9  ***A  ***B    ***C  ***D  ***E  ***F

0000     703F  FFFB  0000  7F6C    0FFA  0120  0080  0000    0091  0091  0083  2236    00C4  0091  8000  0000
0010     1200  5540  FFFF  0000    01DE  D100  0001  01DE    69FB  703F  0000  0000    00F2  7400  00EE  70FD
0020     C600  225B  D600  7FC1    72FF  70FA  4C00  034C    0485  3000  4C80  0028    003F  0130  0000  0000
0030     0000  0000  0001  0000    0000  0000  0000  0002    7014  0000  1810  7012    0001  0004  FFFF  0000
0040     7400  0032  70FD  D8D6    69D2  C480  003F  D001    C8F3  4400  00F2  C0F0    7001  C0F0  D0C7  7400
0050     0032  70FD  08BD  6580    0039  C101  18D0  C100    D8B8  6500  01DC  C0FE    1890  4400  00F2  7400
0060     00EE  70FD  4102  0002    0000  0000  0000  0000    0000  0000  0000  0000    0000  0000  0000  0000
0070     0000  0000  0000  0C00    0000  0217  FFFF  0000    0000  0001  0000  01E0    0000  0000  0000  0000
         0000  0000  3000  4C80                      4C80    0085  0000  3000  4C80    0089  0000  3000  4C80

0170     016A  7201                            E249  D250    C400              D23A          D239  EA50  9247
0180     D237  EA42  8247  D24D    EA48  D23B  CA3C  0A3A    D2EB  4828  70C1  1002    4828  70C0  C101  9400
0190     009A  4818  701B  1893    180F  1002  EA3A  18D0    4810  7002  F251  8230    DA34  C2EB  100D  4810
01A0     7003  C101  1803  D234    4213  CA38  DA34  4213    C231  D480  0190  9101    4C20  0116  CA3C  4808
01B0     7095  8A40  DA3C  4830    1810  824F  D100  CA36    DA34  C101  EA50  D101    4213  C24D  D235  C247
01C0     4820  4213  CA32  D900    C23C  4808  70E9  7500    0140  C900  DA32  CA3C    D900  708B  0000  0000
01D0     0000  0000  0000  0000    0000  0000  0000  0000    0000  0000  00A0  1111    0170  0105  0E20  0120
01E0     034C  0000  FFFF  0000    001E  0000  01E0  0000    003E  20BA  7F6C  0091    0091  00B3  2236  00C4
01F0     0091  0091  0091  0091    0091  0091  0091  0091    0091  0091  0000  8000    0000  0000  72FF  70FA  } GCA
0200     4C00  034C  03CD  3000    4C80  0028  003F  0130    0000  0000  0000  0000    0001  0000  0000  0000
0210     0000  0002  7014  0000    1810  7012  0001  0004    FFFF  0000  7400  0032    70FD  D8D6  69D2  C480
0220     003F  D0D1  C8F3  4400    00F2  C0F0  7001  C0F0    4000  0089  5FE0  008A    5FE0  008A  4000  0089  } Input Data
0230     4000  0089  4000  0089    5FE0  008A  5FE0  008A    4000  0089  4000  008A    5FE0  008A  4000  008A
0240     4000  008A  4000  0089    4000  008A  5FE0  008A    0000  0000  0000  0000    0000  0000  0000  0217  } ICA
0250     FFFF  0000  0000  0001    0000  01E0  0000  0000    0000  0000  0000  0000    3C00  4C80  0081  0000  } Preface
0260     3000  4C80  0085  0000    3000  4C80  0089  0000    3000  4C80  008D  002A    3000  4CC0  0091  0217
0270     0000  0000  0000  0000    0120  0000  0000  0000    0000  9000  0000  0000    0000  0000  0658  0658
0280     0658  0000  0000  0000    0000  0000  0000  0000    0000  0000  0000  0000    0000  0051  6906  6A07
0290     2807  D80A  4400  00F7    6500  01DE  6600  034D    2000  C802  4CC0  00B3    0001  9404  0605  D818
02A0     280E  690F  6A10  0816    1002  4C10  00D0  4480    002C  FFFE  6109  0810    1140  4580  01F0  2001  } ICA
02B0     6500  1000  6600  0000    C803  4CC0  00C4  4001    0283  0000  0201  0F00    0000  0300  0000  0000
02C0     0000  0000  0000  0000    0000  0000  0000  0000    0011  0000  0000  FF6A    004B  7400  00EE  70FD
02D0     7002  00BA  7015  690F    6A10  1008  D03C  18D0    D056  6211  6AED  C0F0    D0F4  704E  4C00  01BD
02E0     6908  081E  6500  01DE    6600  034D  4C80  00F7    6500  0004  6600  00F2    0819  D0C9  4850  70EE
02F0     C80D  U900  74FF  00EE    7039  C812  C014  4293    1810  D480  0190  700D    0001  0120  0FFA  0120
0300     0004  9500  0004  9500    0122  9600  9400  9781    0E8A  0121  5002  5004    FEC0  0001  0080  0600
0310     0008  5000  0FF8  0100    0701  0007  000A  009F    FFFB  9680  0400  0141    0000  FFFF  0000  0000  } Attention
0320     1810  D0A6  74FF  0032    1000  708C  C0E3  70CF    C0E8  4400  0028  7035    7401  0032  6500  0004  } Data Array
0330     C900  D8CC  D8D5  1810    1084  D00E  80E0  D01B    800F  D030  80DB  8008    8007  D006  4000  0089
0340     4000  008A  0001  0000    0002  0004  0003  0000    0000  0000  0000  0000    4377  0342  4480  7FFE
0350     0343  0343  0338  033A    6500  0000  7500  030E    7000  6D00  0365  6500    FF3A  7500  030E  7000
0360     6D00  0366  4480  7FFD    0342  030E  030E  0343    4480  7FFC  0226  4377    0344  4480  7FFB  0345
0370     0346  4480  7FFA  0226    033E  0340  4480  7FF9    0226  023E  0246  0345    4480  7FF8  0345  4480
0380     7FF8  0344  0345  4480    7FFB  0346  0344  4480    7FF7  0226  033E  033E    0343  4480  7FF9  0226
0390     022E  0236  0345  4480    7FF8  0344  4480  7FF6    0344  4480  7FF5  0342    0344  0343  4377  0346
03A0     4480  7FF4  0336  6500    0000  C500  0336  9400    0344  4C20  039E  4480    7FF3  0344  0342  4480
03B0     7FF7  0226  033E  0340    0343  4480  7FF2  0345    0343  0342  4480  7FF8    0344  4C00  0399  0000
03C0     3200  43B6  03B6  80BF    9980  8AF6  80F6  F6F6    E780  0459  8F6F  E808    EF07  E880  0000  0000
03D0     0000  0000  0000  0000    0000  0000  002A  008A    4000  0000  0200  FFFF    03FF  0010  000F  000E
03E0     000D  000C  000B  000A    0009  0008  0007  0006    0005  0004  0003  0002    0001  0000  0000  001E
03F0     0000  0000  0000  0000    0000  0000  0000  0000    0000  0000  0000  0000    0000  0000  0000  0000
0400     0000  0000  0000  0000    0000  0000  0000  0000    0000  0000  0000  0000    0000  0000  0000  0019
0410     0000  0000  0000  0000    0000  0000  0000  0200    0200  008A  4000  0000    008A  4000  0000  0200  } GCOM
0420     0200  0000  0000  0000    0000  0000  0000  0000    0000  0000  0000  0000    0000  0000  0000  0000
0430     0000  0000  0000  0000    0000  0000  0000  0000    0000  0001  0000  0000    0000  0000  0000  0000
0440     0000  0000  0000  0000    0000  0000  0000  0000    0000  0000  0000  0000    0000  0000  0000  0459
0450     0000  0000  0000  0000    0000  0000  3400  2C01    0456  3400  2C00  03C0    2C00  0469  2C00  0000  } System
0460     3105  3200  4200  0200    2C00  0000  2C00  0474    0459  3109  3300  03CA    3100  2802  0470  0469  } Display
0470     3400  2001  0002  0469    2801  0469  0459  9404    6A12  6680  7FE3  7400    0032  70FD  1010  D400
0480     00F0  C680  0000  4400    0028  7400  00F0  70F6    7201  6A03  6600  0000    4C00  0000  2000  0000
0490     6D00  04EE  6E00  04EF    6F00  04F0  C400  000E    9400  0007  905A  D051    C480  04ED  D04C  6580
04A0     04EB  C1FF  905B  4820    7004  C1E4  9055  4C18    04AE  C04E  18D0  C050    3000  6038  6000  0063
04B0     6369  1010  D1F6  7101    73FF  70FC  6580  0063    6303  D1DE  71FF  73FF    70FC  6580  0063  C032
04C0     D1F8  6780  04F0  4374    3000  C1F5  D142  D1DD    6780  0063  7362  6B1F    C01E  D158  C163  E821
04D0     D163  D165  D167  D16D    D16F  6780  04ED  73FF    6600  04FF  C300  D00F    D00F  74FF  04EB  C480
04E0     04E8  E01B  9200  4E98    0001  C200  9015  4C18    051E  7202  70F4  0000    0000  0000  0000  0000
04F0     0000  0400  0030  008A    4000  0001  000C  000D    200C  200D  0000  0000    FFFF  FE00  0000  0200
               0600  050C  D600    0510  CA00  0514  C800    0518        051C  FFFF    C0DF  D155  73FF  70CA

05C0                 436E  2000    C1F4  436E  1000  C1EA    70FC                      0000  0000  0000
05D0     0049  4C08  05C4  C14A    D0F6  C140  91F5  4C18    05E3  C140  91F3  4C18    05E0  74FE  05C8  7005
05E0     74FF  05C8  7002  74FD    05CB  C14B  D0E5  80E2    90E2  4C10  05C7  6680    05CC  4C80  05CE  0000
05F0     C152  D0DB  4C18  0600    D0D7  6A09  4374  2000    6680  05CC  C11D  D202    C11E  D201  6600  0000
0600     4C80  05EF  91F5  4C20    0623  6A10  6680  05CB    C14B  9201  4C20  0623    C580  0049  D200  C0BB
0610     D13D  C201  D152  4374    2001  6600  0000  C201    D11E  C202  D11D  6680    05CD  C20B  4420  0625
0620     709D  C1ED  70A2  C1F5    70A0  0000  D010  9152    4820  700C  6A0F  740B    063A  6680  0616  C20B
0630     D480  063A  C09A  D20B    4C80  0625  6600  0000    C20B  70EC  0000  4371    1948  0000  4C00  0575
0640     C400  0063  800C  D009    C009  D480  064D  4371    1950  064D  0000  4C00    05C2  0000  3400  0062
0650     4480  7FF1  5000  0000    436E  0001  C149  9033    D030  C141  91F3  4C18    0660  7402  0689  7002
0660     7403  0689  6680  0689    6580  068B  1010  D200    7201  71FF  70FC  6580    0063  C1F5  436B  6680
0670     0689  7204  6A01  4368    0000  0001  7206  6A01    4368  0000  0001  6680    0689  C1F5  D21C  D21D
0680     D21E  D21F  C1E5  D202    D203  C004  D220  436E    2000  0000  2580  002A    2400  0000  436E  0002
0690     C11F  100E  4C28  06F7    C580  0049  4C08  06F9    1010  D12B  D12A  C580    004A  4C08  06F4  91F0
```

Figure 21. PAUSE 0001

PAUSE 0002

After the initialization subroutines have been called as shown in Figure 22, the following areas contain valid data to be used by the GSP.

The GCA: The GCA contains the standard values as specified in the call to GCAIN. Table 8 in Appendix A describes the standard values found in the GCA.

The ICA: The ICA contains three sections; the ICA preface, the area where the graphic orders are to be stored, and the area containing the ECCBs for the various graphic elements. The ICA preface is initialized with the standard values found in Table 9 in Appendix A with the following exceptions:

Core		ICA	
Address	Mnemonic	Location	Comments
/0248	SAVA	1	Pointer to /0260, the address where the first graphic order will be positioned.
/0249	EAVA	2	Pointer to /030B, the address contained in the LAGAP field of GCOM. EAVA is used as a save area for LAGAP during update functions.
/024A	NAVWD	3	Pointer to /0260, the address contained in the NAGAP field of GCOM. NAVWD is used as a save area for NAGAP during update functions.
/024B	SCB	4	Pointer to /030C, the first ECCB in the chain.
/024C	ECB	5	Pointer to /030C, the last ECCB in the chain.

The ICA proper, the area where graphic orders are to be stored, contains zeros after the initialization process.

The ECCB area contains the ICA ECCB after initialization as a result of the call to the ICAIN subroutine. The ICA ECCB has the following format:

Core		ECCB	
Address	Mnemonic	Location	Comments
/030C	CORVL	0	Correlation value of 1.
/030D	ELSTA	1	Pointer to /0248, the start address of the ICA.
/030E	ELEND	2	Pointer to /030F, the end address of the ICA.
/030F			
(bit 0-10)	FORPT	3	Pointer to the next ECCB; at this time it contains zeros since the next ECCB has not yet been built.
/030F	INDIC		
(bit 11)		3	A value of 1 indicating the element is ended.
/030F	INDIC		
(bit 12-15)		3	All zeros; the code for an image entity.

```
PAGE   2

ACCUMULATOR 0000      EXTENSION 0000     XR1 0248     XR2 036C     XR3 7F6C        OVERFLOW OFF         CARRY OFF

ADDR       ***0  ***1  ***2  ***3     ***4  ***5  ***6  ***7     ***8  ***9  ***A  ***B     ***C  ***D  ***E  ***F

0000       703F  FFFB  0000  7F6C     0FFA  0120  0080  0000     0091  0091  00B3  2236     00C4  0091  8000  0000
0010       1200  5540  FFFF  0000     01DE  0100  0001  0248     69F8  703F  0000  0000     00F2  7400  00EE  70FD
0020       C600  2258  D600  7FC1     72FF  70FA  4C00  034C     0485  3000  4C80  0028     003F  0130  0000  0000
0030       0000  0000  0001  0000     0000  0000  0000  0002     7014  0000  1810  7012     0001  0004  FFFF  0000
0040       7400  0032  70FD  D8D6     69D2  C480  003F  D0D1     C8F3  4400  00F2  C0F0     7001  C0F0  D0C7  7400
0050       0032  70FD  088D  6580     0039  C101  18D0  C100     D888  6500  01DC  C0FE     1890  4400  00F2  7400
0060       00EE  70FD  4102  03F7     0000  0000  0000  0000     0000  0000  0000  0000     0000  0000  0000  0000
0070       0000  0000  0000  0000     0000  0217  FFFF  0000     0000  0001  0000  01E0     0000  0000  0000  0000
0080       0000  0000  3000  4C80     0081  0000  3000  4C80     0085  0000  3000  4C80     0089  0000  3000  4C80
0090       008D  002A  3000  4CC0     0091  0217  0000  0000     0000  0000  0120  0000     0000  0000  0000  9000
                               0000     0658                     0000  0000  0000                  0000  0000


0150       0028  7035  7401  0032     6500  0004  C900           D8D5  1810                             801B  80DF
0160       80DB  8008  8007  D006     62FD  69C2  C101  E0D0     D101  9400  00A4  4828     7006  C101  80C7  7401
0170       016A  7201  70F5  6600     00F2  C23D  E249  D250     C400  009F  EA4E  D23A     EA43  D239  EA50  9247
0180       D237  EA42  8247  D24D     EA48  D23B  CA3C  0A3A     D2EB  4828  70C1  1002     4828  70C0  C101  9400
0190       009A  4818  7018  1893     180F  1002  EA3A  18D0     4810  7002  F251  8230     DA34  C2E8  1000  4810
01A0       7003  C101  1803  D234     4213  CA38  DA34  4213     C231  D480  0190  9101     4C20  0116  CA3C  4808
0180       7095  8A40  DA3C  4830     1810  824F  D100  CA36     DA34  C101  EA50  D101     4213  C24D  D235  C247
01C0       4820  4213  CA32  D900     C23C  4808  70E9  7500     0140  C900  DA32  CA3C     D900  7088  0000  0000
01D0       0000  0000  0000  0000     0000  0000  0000  0000     0000  0000  00A0  1111     0170  0105  0E20  0120
01E0       034C  0000  FFFF  0000     001E  0000  01E0  0000     003E  208A  7F6C  0091     0091  0083  2236  00C4
01F0       0091  0091  0091  0091     0091  0091  0091  0091     0091  0091  0000  8000     0000  0000  0000  0000
0200       03FF  03FF  0081  4000     0000  0000  0000  0000     0081  4000  0000  0000     0000  0000  0000  0000  GCA
0210       0000  0000  0000  0000     0000  0000  0000  0000     0000  0000  0001  0001     0001  0001  2580  0000
0220       0000  0000  0000  0000     0000  0000  0000  0000     4000  0089  5FE0  008A     5FE0  008A  4000  0089
0230       4000  0089  4000  0089     5FE0  008A  5FE0  008A     4000  0089  4000  008A     5FE0  008A  4000  008A  Input Data
0240       4000  008A  4000  0089     4000  008A  5FE0  008A     0260  030B  0260  030C     030C  0000  0000  0000
0250       0000  0000  0000  0000     0200  0200  008A  4000     0000  008A  4000  0000     0200  0200  0000  0000
0260       0000  0000  0000  0000     0000  0000  0000  0000     0000  0000  0000  0000     0000  0000  0000  0000  ICA
                                                                                                                  ICA
0300       0000  0000  0000  0000     0000  0000  0000  0000     0000  0000  0000  0000     0001  0248  030F  0010  ECCB
0310       0008  5000  0FF8  0100     0701  0007  000A  009F     FFFB  9680  0400  0141     0000  FFFF  0000  0000
0320       1B10  D0A6  74FF  0032     1000  70BC  C0E3  70CF     C0E8  4400  0028  7035     7401  0032  6500  0004
0330       C900  D8CC  D8D5  1810     1084  D00E  80E0  D01B     0000  D030  0000  8008     8007  D006  4000  0089
0340       4000  008A  0001  0000     0002  0004  0003  0000     0000  0000  0000  0000     4377  0342  4480  7FFE
0350       0343  0343  0338  033A     6500  0000  7500  030E     7000  6D00  0365  6500     FF3A  7500  030E  7000
0360       6D00  0366  4480  7FFD     0342  030E  0248  0343     4480  7FFC  0226  4377     0344  4480  7FFB  0345
0370       0346  4480  7FFA  0226     033E  0340  4480  7FF9     0226  023E  0246  0345     4480  7FF8  0345  4480
0380       7FFB  0344  0345  4480     7FFB  0346  0344  4480     7FF7  0226  033E  033E     0343  4480  7FF9  0226
0390       022E  0236  0345  4480     7FF8  0344  4480  7FF6     0344  4480  7FF5  0342     0344  0343  4377  0346
03A0       4480  7FF4  0336  6500     0000  C500  0336  9400     0344  4C20  039E  4480     7FF3  0344  0342  4480
0380       7FF7  0226  033E  0340     0343  4480  7FF2  0345     0343  0342  4480  7FF8     0344  4C00  0399  00B3
03C0       3200  43B6  03B6  80BF     9980  8AF6  80F6  F6F6     E780  0459  8F6F  E808     EF07  E880  0000  0000
0300       0000  0000  0000  0000     0003  0000  002A  008A     4000  0000  0200  FFFF     03FF  0010  000F  000E
03E0       000D  000C  000B  000A     0009  0008  0007  0006     0005  0004  0003  0002     0001  0000  0000  0030
03F0       0000  0CE8  0000  0000     C000  5000  0000  136D     136D  0000  0000  0000     0000  0000  0000  0000
0400       0000  0000  0000  0000     0000  0000  0000  0000     0000  0000  0000  0000     0000  0000  0000  0000
0410       0000  0000  0000  0000     0260  030B  0000  0200     0200  008A  4000  0000     008A  4000  0000  0200  GCOM
0420       0200  0000  0000  0000     0000  0000  0000  0000     0000  0000  0000  0000     0000  0000  0000  0000
0430       0000  0000  0000  0000     030C  0338  033A  0002     0002  0001  0000  0000     036B  0248  0000  7F6C
0440       0226  030E  0248  0343     0000  0000  0000  0000     0000  0248  0000  0000     0000  0000  0000  0459
0450       0000  0000  0000  0000     0000  0000  3400  2C01     0456  3400  2C00  03C0     2C00  0469  2C00  0000  System
0460       3105  3200  4200  0200     2C00  0000  2C00  0474     0459  3109  3300  03CA     3100  2802  0470  0469  Display
0470       3400  2001  0002  0469     2801  0469  0459  1800     6A12  6680  7FE3  7400     0032  70FD  1010  D400
0480       00F0  C680  0000  4400     0028  7400  00F0  70F6     7201  6A03  6600  0000     4C00  034E  2000  0350
0490       6D00  04EE  6E00  04EF     6F00  04F0  C400  000E     9400  0007  905A  D051     C480  04ED  D04C  6580
04A0       04EB  C1FF  9058  4820     7004  C1E4  9055  4C18     04AE  C04E  18D0  C050     3000  6038  6D00  0063
0480       6369  1010  D1F6  7101     73FF  70FC  6580  0063     6303  D1DE  71FF  73FF     70FC  6580  0063  C032
04C0       D1F8  6780  04F0  4374     3000  C1F5  D142  D1DD     6780  0063  7362  6B1F     C01E  D158  C163  E821
0400       D163  D165  D167  D16D     D16F  6780  04ED  73FF     6600  04FF  C300  D00F     D00F  74FF  04EB  C480
04E0       04EB  E01B  9200  4E98     0001  C200  9015  4C18     051E  7202  70F4  048E     048F  7FFF  01DE  0000
04F0       7F6C  0400  0030  008A     4000  0001  000C  000D     200C  200D  0000  0000     FFFF  FE00  0000  0200
0500       050C  0600  050C  D600     0510  CA00  0514  C800     0518  D400  051C  FFFF     C0DF  D155  73FF  70CA
0510       C0DB  D157  73FF  70C6     C0D7  D154  73FF  70C2     C0D3  D159  73FF  708E     C0CF  D1DB  6680  048F
                 C203  9204  4C18     0539  C203  D13E  C204     D13F  1010  D580  003E     D580  003F  C680  0001
                                      053E                       7028                     C1F4        C1F5  D140


                       70A2  C1F5     70A0                                         740B
0630       D480  063A  C09A  D20B     4C80  0625  6600           C208  70EC  0000  4371           0000
0640       C400  0063  800C  D009     C009  D480  064D  4371     1950  064D  0000  4C00     05C2  0000  3400  0062
0650       4480  7FF1  5000  036A     436E  0001  C149  9033     D030  C141  91F3  4C18     0660  7402  0689  7002
0660       7403  0689  6680  0689     6580  068B  1010  D200     7201  71FF  70FC  6580     0063  C1F5  436B  6680
0670       0689  7204  6A01  4368     0202  0001  7206  6A01     4368  0208  0001  6680     0689  C1F5  D21C  D21D
0680       D21E  D21F  C1E5  D202     D203  C004  D220  436E     2000  01FE  2580  002A     2400  0000  436E  0002
0690       C11F  100E  4C28  06F7     C580  0049  4C08  06F9     1010  D12B  D12A  C580     004A  4C08  06F4  91F0
06A0       4C30  06F4  4C10  0733     81F5  4C20  06AF  C1DB     4C18  0740  C1ED  D02B     C1E4  D1DA  7025  C580
0680       004A  91F5  4C20  06B7     C1F4  D021  701D  C580     004A  91F4  4C20  068F     C1F5  D019  7015  C580
06C0       004A  91F3  4C20  06D0     C1F5  D1F9  4374  2000     4374  3000  C831  4365     2000  C1F3  D008  7004
06D0       1010  D005  4374  3000     C580  0049  4374  0000     C580  004A  91F5  4C18     06F2  C580  004A  91F4
06E0       4C18  06E7  91F3  4C28     06EF  4C20  0703  1090     C019  4365  2000  C812     4365  2000  7003  C810
06F0       4365  2000  436E  2000     C1F5  436E  1000  C1ED     70FC  C1F4  70FA  060C     2800  06FE  2C80  0700
0700       3500  FFFF  310A  C83E     4365  2000  6680  0712     C203  100C  180C  91EE     4818  7003  C1F5  D034
0710       7010  6600  0000  C201     D001  6600  0000  C203     1008  1808  81F5  D028     91F1  4808  7002  C1F0
```

Figure 22. PAUSE 0002

GCOM: GCOM contains the following values after the initialization process has been completed. The fields of GCOM which have not been described have the standard values found in Table 7 of the Appendix A.

Core Location	Mnemonic	GCOM Location	Comments
/03C9	-------	-47	A Graphic Short Branch order to /0459, the second start timer in the system display.
/03D4	GSPEC	-35	The number 3, representing the number of times GSP01 has been called by external subroutines.
/03F1	SSA	-6	Pointer to OCE8, the GSP03 search function save area.
/03F4	IDINT	-3	The ID word for GSP01 (/C000), indicating the last internal subroutine that executed. This word is useful in determining at what point program execution ceased.
/03F5	IDGSP	-2	The ID word for GCAIN (/5000), indicating the last external subroutine that executed. This word is useful in determining at which point program execution ceased.
/03F7	LPDE	0	Pointer to /136D, the address of the light pen direct entry subroutine.
/03F8	OCDE	1	Pointer to the address of the order controlled direct entry subroutine.
/0414	NAGAP	29	Pointer to /0260, the next available word in the ICA for storing a graphic order.
/0415	LAGAP	30	Pointer to /030B, the address of the last available word in the ICA for storing a graphic order.
/0434	ACICA	61	Pointer to /030C, the address of the ECCB for the active ICA.
/0435	RETEV	62	Pointer to /0338, the address of the return error variable.
/0436	CUMEV	63	Pointer to /033A, the address of the cumulative error variable.
/043C	MLRET	69	Pointer to /036B, the return address of the mainline program.
/043D	REG1	70	The save area for index register 1; currently pointing to /0248, the first available position for storing an order.
/043F	REG3	72	The save area for index register 3; currently pointing to /7F6C, the TV table.
/0440	ARG1	73	Pointer to /0226 (GCA), the argument 1 save area.

/0441	ARG2	74	Pointer to /030E (ICA), the argument 2 save area.
/0442	ARG3	75	Pointer to /0248 (ICA end address), the argument 3 save area.
/0443	ARG4	76	Pointer to /0343 (ICA redefine option), the argument 4 save area.
/0449	AICAP	82	Pointer to /0248, the active ICA preface.
/044F	ASD	88	Pointer to /0459, the second start timer order in the system display.
/0458	SDSW	96	Pointer to /0456, the first start timer in the system display.
/045B	SD2AD	100	Pointer to /03C0, the address of the processing message.
/045D	SD3AD	102	Pointer to /0469, the address of the scanning pattern.
/0467	xxxxx	112	Pointer to /0474, that part of the system display used to locate position of the light pen on no detect.
/0468	xxxxx	113	Pointer to /0459, the second start timer in the system display.
/046B	xxxxx	116	Graphic Short Branch order to the scanning pattern.
/046E	xxxxx	119	Address word of GBC order pointing to /0470, used in scanning pattern function.
/046F	xxxxx	120	Graphic Short Branch to /0469, used in scanning pattern functions.
/0473	xxxxx	124	Graphic Short Branch to /0469, used in scanning pattern functions.
/0475	SCNND	126	Pointer to /0469, address word of GBC to SCAN1.
/0476	xxxxx	127	Graphic Short Branch to /0459, second start timer order in system display.

PAUSE 0003 (First Execution)

Figure 23 shows the contents of the selected core locations (ICA and GCOM) after the first call to EXEC. The three graphic elements and their ECCBs have been built and light pen attentions enabled resulting in the following changes within the core locations:

The ICA

Core Address	Mnemonic	Word Location	Comments
/024C	ECB	5	Points to /0300, the last ECCB on the active chain. Note that words 2 and 3 have not changed as they are used as save areas for update functions.

/0260	-----	25	The three graphic elements have been generated in the lower core area of the ICA.
/0300	ECCBs	--	ECCBs for the three elements have been built and chained to the ECCB for the ICA in descending core positions.

GCOM

Core		Word	
Address	Mnemonic	Location	Comments
/03D4	GSPEC	-35	Indicates that external GSP subroutines have been entered 14 times.
/0405	IMECB	14	Points to /0304, the address of the image entity ECCB.
/0410	SATEN	26	Indicates that light pen attentions are enabled.
/0414	NAGAP	30	Points to /0293, the address where the next graphic order will be stored.
/0415	LAGAP	31	Points to /02FF, the address where the last graphic order can be stored.
/0465	SD5AD	110	In system display, points to /027B, the first order of the image entity.

PAUSE 0003 (Second Execution)

Figure 24 shows the contents of the selected core locations (ICA, Attention Array, and GCOM) after the second call to EXEC. The light pen attention upon the rectangle (controlled entity nested within the image entity) has caused the program to link the subroutine to the image entity resulting in the following changes:

The ICA

Core		Word	
Address	Mnemonic	Location	Comments
/0249	EAVA	2	Points to /02FF, the last position for storing orders in the ICA; has been updated by XELMT.
/024A	NAVWD	3	Points to /0298, the position for the storing of the next graphic order; has been updated by GSP05.
/028E	-----	-	Points through /0309, the subroutine ECCB, the linkage to the subroutine entity.

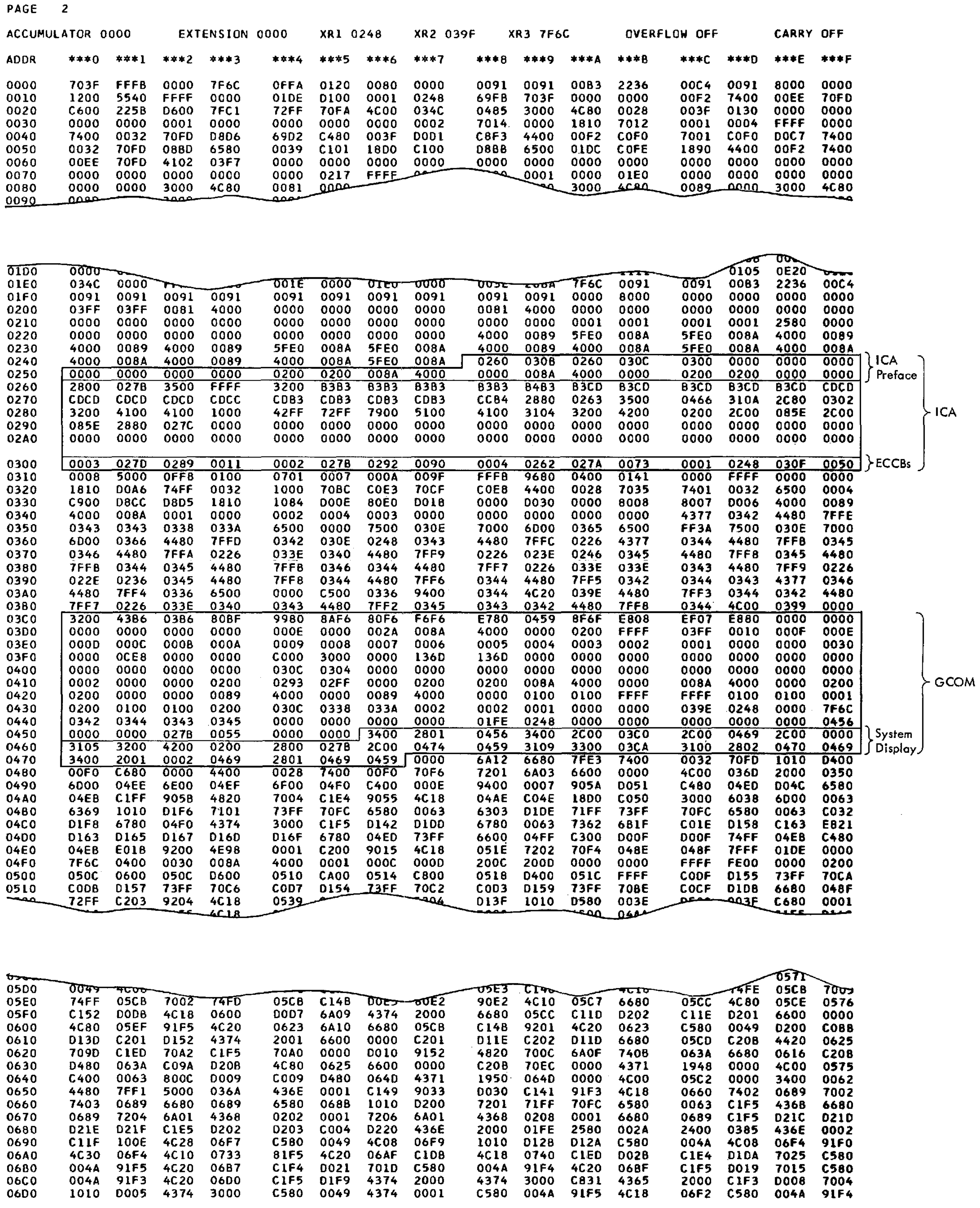
Attention Array

Core		Word	
Address	Mnemonic	Location	Comments
/0336		1	Indicates a light pen attention.
/0332		3	Indicates that detect was not on a character.

/0330		4	x-coordinate of the line on which the detect cccurred.
/032E		5	y-coordinate of the line on which the detect occurred.
/032C		6	Correlation value of the ICA.
/032A		7	Correlation value of the image entity.
/0328		8	Correlation value of the controlled entity.
/0322		11	Address of the order causing the attention.
/0320		12	Address of the next order after that causing the attention.

GCOM

Core Address	Mnemonic	Word Location	Comments
/03D4	GSPEC	-35	Indicates that external GSP subroutines have been entered 20 times.
/03FD	ATNDT	7-12	Contains the results of the XIO Read Status command issued by DSPYN.
/0406	OADDR	16	Points to /0286, address at which the interrupt occurred.
/0407	NOADR	17	Points to /0287, next address after the interrupt.
/0408	ICACV	18	Contains the correlation value for the ICA.
/0409	IECVV	19	Contains the correlation value for the image entity.
/040A	CECV	20	Contains the correlation value for the controlled entity.
/0411	RQNDA	27	Points to /11D6, the address of the attention array in DSPYN.
/0414	NAGAP	30	Points to /0298, the address where the next graphic order will be stored in ICA.



```
PAGE    2

ACCUMULATOR 0000       EXTENSION 0000     XR1 0248     XR2 039F     XR3 7F6C        OVERFLOW OFF         CARRY OFF

ADDR       ***0   ***1   ***2   ***3    ***4   ***5   ***6   ***7    ***8   ***9   ***A   ***B    ***C   ***D   ***E   ***F

0000       703F   FFFB   0000   7F6C    0FFA   0120   0080   0000    0091   0091   00B3   2236    00C4   0091   8000   0000
0010       1200   5540   FFFF   0000    01DE   D100   0001   0248    69FB   703F   0000   0000    00F2   7400   00EE   70FD
0020       C600   225B   D600   7FC1    72FF   70FA   4C00   034C    0485   3000   4C80   0028    003F   0130   0000   0000
0030       0000   0000   0001   0000    0000   0000   0000   0002    7014   0000   1810   7012    0001   0004   FFFF   0000
0040       7400   0032   70FD   D8D6    69D2   C480   003F   D0D1    C8F3   4400   00F2   C0F0    7001   C0F0   D0C7   7400
0050       0032   70FD   08BD   6580    0039   C101   18D0   C100    D8BB   6500   01DC   C0FE    1890   4400   00F2   7400
0060       00EE   70FD   4102   03F7    0000   0000   0000   0000    0000   0000   0000   0000    0000   0000   0000   0000
0070       0000   0000   0000   0000    0000   0217   FFFF                  0001   0000   01E0    0000   0000   0000   0000
0080       0000   0000   3000   4C80    0081                                       3000           0089          3000   4C80
0090

0100       0000                                                                                         0105   0E20
01E0       034C   0000                  001E   0000                                 7F6C   0091    0091   0083   2236   00C4
01F0       0091   0091   0091   0091    0091   0091   0091   0091    0091   0091   0000   8000    0000   0000   0000   0000
0200       03FF   03FF   0081   4000    0000   0000   0000   0000    0081   4000   0000   0000    0000   0000   0000   0000
0210       0000   0000   0000   0000    0000   0000   0000   0000    0000   0000   0001   0001    0001   0001   2580   0000
0220       0000   0000   0000   0000    0000   0000   0000   0000    4000   0089   5FE0   008A    5FE0   008A   4000   0089
0230       4000   0089   4000   0089    5FE0   008A   5FE0   008A    4000   0089   4000   008A    5FE0   008A   4000   008A
0240       4000   008A   4000   0089    4000   008A   5FE0   008A    0260   030B   0260   030C    0300   0000   0000   0000
0250       0000   0000   0000   0000    0200   0200   008A   4000    0000   008A   4000   0000    0200   0200   0000   0000
0260       2800   027B   3500   FFFF    3200   B3B3   B3B3   B3B3    B3B3   B4B3   B3CD   B3CD    B3CD   B3CD   B3CD   CDCD
0270       CDCD   CDCD   CDCD   CDCC    CDB3   CDB3   CDB3   CDB3    CCB4   2880   0263   3500    0466   310A   2C80   0302
0280       3200   4100   4100   1000    42FF   72FF   7900   5100    4100   3104   3200   4200    0200   2C00   085E   2C00
0290       085E   2880   027C   0000    0000   0000   0000   0000    0000   0000   0000   0000    0000   0000   0000   0000
02A0       0000   0000   0000   0000    0000   0000   0000   0000    0000   0000   0000   0000    0000   0000   0000   0000

0300       0003   027D   0289   0011    0002   027B   0292   0090    0004   0262   027A   0073    0001   0248   030F   0050
0310       0008   5000   0FF8   0100    0701   0007   000A   009F    FFFB   9680   0400   0141    0000   FFFF   0000   0000
0320       1810   D0A6   74FF   0032    1000   70BC   C0E3   70CF    C0E8   4400   0028   7035    7401   0032   6500   0004
0330       C900   D8CC   D8D5   1810    1084   D00E   80E0   D018    0000   D030   0000   8008    8007   D006   4000   0089
0340       4000   008A   0001   0000    0002   0004   0003   0000    0000   0000   0000   0000    4377   0342   4480   7FFE
0350       0343   0343   0338   033A    6500   0000   7500   030E    7000   6D00   0365   6500    FF3A   7500   030E   7000
0360       6D00   0366   4480   7FFD    0342   030E   0248   0343    4480   7FFC   0226   4377    0344   4480   7FFB   0345
0370       0346   4480   7FFA   0226    033E   0340   4480   7FF9    0226   023E   0246   0345    4480   7FF8   0345   4480
0380       7FFB   0344   0345   4480    7FFB   0346   0344   4480    7FF7   0226   033E   033E    0343   4480   7FF9   0226
0390       022E   0236   0345   4480    7FF8   0344   4480   7FF6    0344   4480   7FF5   0342    0344   0343   4377   0346
03A0       4480   7FF4   0336   6500    0000   C500   0336   9400    0344   4C20   039E   4480    7FF3   0344   0342   4480
03B0       7FF7   0226   033E   0340    0343   4480   7FF2   0345    0343   0342   4480   7FF8    0344   4C00   0399   0000
03C0       3200   4386   0386   80BF    9980   8AF6   80F6   F6F6    E780   0459   8F6F   E808    EF07   E880   0000   0000
03D0       0000   0000   0000   0000    000E   0000   002A   008A    4000   0000   0200   FFFF    03FF   0010   000F   000E
03E0       000D   000C   000B   000A    0009   0008   0007   0006    0005   0004   0003   0002    0001   0000   0000   0030
03F0       0000   0CE8   0000   0000    C000   3000   0000   136D    136D   0000   0000   0000    0000   0000   0000   0000
0400       0000   0000   0000   0000    030C   0304   0000   0000    0000   0000   0000   0000    0000   0000   0000   0000
0410       0002   0000   0000   0200    0293   02FF   0000   0200    0200   008A   4000   0000    008A   4000   0000   0200
0420       0200   0000   0000   0089    4000   0000   0089   4000    0000   0100   0100   FFFF    FFFF   0100   0100   0001
0430       0200   0100   0100   0200    030C   0338   033A   0002    0002   0001   0000   0000    039E   0248   0000   7F6C
0440       0342   0344   0343   0345    0000   0000   0000   0000    01FE   0248   0000   0000    0000   0000   0000   0456
0450       0000   0000   027B   0055    0000   0000   3400   2801    0456   3400   2C00   03C0    2C00   0469   2C00   0000
0460       3105   3200   4200   0200    2800   027B   2C00   0474    0459   3109   3300   03CA    3100   2802   0470   0469
0470       3400   2001   0002   0469    2801   0469   0459   0000    6A12   6680   7FE3   7400    0032   70FD   1010   D400
0480       00F0   C680   0000   4400    0028   7400   00F0   70F6    7201   6A03   6600   0000    4C00   036D   2000   0350
0490       6D00   04EE   6E00   04EF    6F00   04F0   C400   000E    9400   0007   905A   D051    C480   04ED   D04C   6580
04A0       04EB   C1FF   905B   4820    7004   C1E4   9055   4C18    04AE   C04E   18D0   C050    3000   6038   6D00   0063
04B0       6369   1010   D1F6   7101    73FF   70FC   6580   0063    6303   D1DE   71FF   73FF    70FC   6580   0063   C032
04C0       D1F8   6780   04F0   4374    3000   C1F5   D142   D1DD    6780   0063   7362   681F    C01E   D158   C163   E821
04D0       D163   D165   D167   D16D    D16F   6780   04ED   73FF    6600   04FF   C300   D00F    D00F   74FF   04EB   C480
04E0       04EB   E01B   9200   4E98    0001   C200   9015   4C18    051E   7202   70F4   048E    048F   7FFF   01DE   0000
04F0       7F6C   0400   0030   008A    4000   0001   000C   000D    200C   200D   0000   0000    FFFF   FE00   0000   0200
0500       050C   0600   050C   D600    0510   CA00   0514   C800    0518   D400   051C   FFFF    C0DF   D155   73FF   70CA
0510       C0DB   D157   73FF   70C6    C0D7   D154   73FF   70C2    C0D3   D159   73FF   708E    C0CF   D1DB   6680   048F
           72FF   C203   9204   4C18    0539                         D13F   1010   D580   003E                  C680   0001

                                                                                                       0571
05D0       0049                                                    05E3                                       05CB
05E0       74FF   05CB   7002   74FD    05CB   C14B          80E2    90E2   4C10   05C7   6680    05CC   4C80   05CE   0576
05F0       C152   D0DB   4C18   0600    D0D7   6A09   4374   2000    6680   05CC   C11D   D202    C11E   D201   6600   0000
0600       4C80   05EF   91F5   4C20    0623   6A10   6680   05CB    C14B   9201   4C20   0623    C580   0049   D200   C0BB
0610       D13D   C201   D152   4374    2001   6600   0000   C201    D11E   C202   D11D   6680    05CD   C20B   4420   0625
0620       709D   C1ED   70A2   C1F5    70A0   0000   D010   9152    4820   700C   6A0F   740B    063A   6680   0616   C20B
0630       D480   063A   C09A   D20B    4C80   0625   6600   0000    C20B   70EC   0000   4371    1948   0000   4C00   0575
0640       C400   0063   800C   D009    C009   D480   064D   4371    1950   064D   0000   4C00    05C2   0000   3400   0062
0650       4480   7FF1   5000   036A    436E   0001   C149   9033    D030   C141   91F3   4C18    0660   7402   0689   7002
0660       7403   0689   6680   0689    6580   068B   1010   D200    7201   71FF   70FC   6580    0063   C1F5   436B   6680
0670       0689   7204   6A01   4368    0202   0001   7206   6A01    4368   0208   0001   6680    0689   C1F5   D21C   D21D
0680       D21E   D21F   C1E5   D202    D203   C004   D220   436E    2000   01FE   2580   002A    2400   0385   436E   0002
0690       C11F   100E   4C28   06F7    C580   0049   4C08   06F9    1010   D12B   D12A   C580    004A   4C08   06F4   91F0
06A0       4C30   06F4   4C10   0733    81F5   4C20   06AF   C1DB    4C18   0740   C1ED   D02B    C1E4   D1DA   7025   C580
06B0       004A   91F5   4C20   06B7    C1F4   D021   701D   C580    004A   91F4   4C20   06BF    C1F5   D019   7015   C580
06C0       004A   91F3   4C20   06D0    C1F5   D1F9   4374   2000    4374   3000   C831   4365    2000   C1F3   D008   7004
06D0       1010   D005   4374   3000    C580   0049   4374   0001    C580   004A   91F5   4C18    06F2   C580   004A   91F4
```

Figure 23. PAUSE 0003

GRAPHIC ORDER REFERENCE TABLE

Table 5 lists the graphic orders commonly used in image generation. Variables are indicated by XXXX or XXX. IBM 2250 Display Unit Model 4 (A27-2723) gives information concerning graphic orders in greater detail.

Table 5. Graphic Order Reference Table

Hexadecimal Value	Mnemonic	Comments
2000 XXXX	GI	Graphic Interrupt (unconditional)
2001 XXXX	GIC	Graphic Interrupt Conditional (switch closed)
2002 XXXX	GIC	Graphic Interrupt Conditional (light pen detect)
2800 XXXX	GB	Graphic Branch (unconditional)
2801 XXXX	GBC	Graphic Branch Conditional (switch closed)
2802 XXXX	GBC	Graphic Branch Conditional (light pen detect)
2880 XXXX	GBI	Graphic Branch Indirect
3100 3103 310C 310F	GNOP	No operation performed
3101	SPM	Set Pen Mode (defer detect interrupts)
3102	SPM	Set Pen Mode (enable detect interrupts)
3104	SPM	Set Pen Mode (disable light pen detects)
3108	SPM	Set Pen Mode (enable light pen detects)
310A	SPM	Set Pen Mode (enable light pen detects and interrupts)
3200	SGMV	Set Graphic Mode Vector (for generation of lines)
3201	SGMP	Set Graphic Mode Point (for generation of points)
3300	SCMB	Set Character Mode Basic
3301	SCML	Set Character Mode Large
3400	STMR	Start Timer
3500 XXXX	SRVT	Store Revert Register (in second word)
3600	RVT	Revert (branch to address in revert register)
4XXX XXXX	MBA	Move Beam Absolute (invisible positioning)
5XXX XXXX	DBA	Draw Beam Absolute (vector or point visible)
6XXX	MBAX MBAY	Short Move Beam Absolute (either x- or y-direction)
7XXX	DBAX DBAY	Short Draw Beam Absolute (either x- or y-direction)

Note: If the second digit of the first word in any of the Graphic Branch or Graphic Interrupt orders is a 4 or a C, the order is a GNOP.

Graphic Short Branch orders are one word in length with the first digit usually a zero (0).

This section contains autocharts for most of the subroutines described in the preceding sections. Charts are not included for subroutines whose logic can be understood by the preceding subroutine description. The charts are to be used in conjunction with the subroutine descriptions and listings. The charts are ordered alphabetically by chart identification and match the sequence in which the subroutines are described. Related subroutines have the same first character in their chart identification (A for initialization subroutines, B for image management subroutines, etc.) Standard flowchart conventions are used.

Chart AA. GSPIN - GSP Initialization

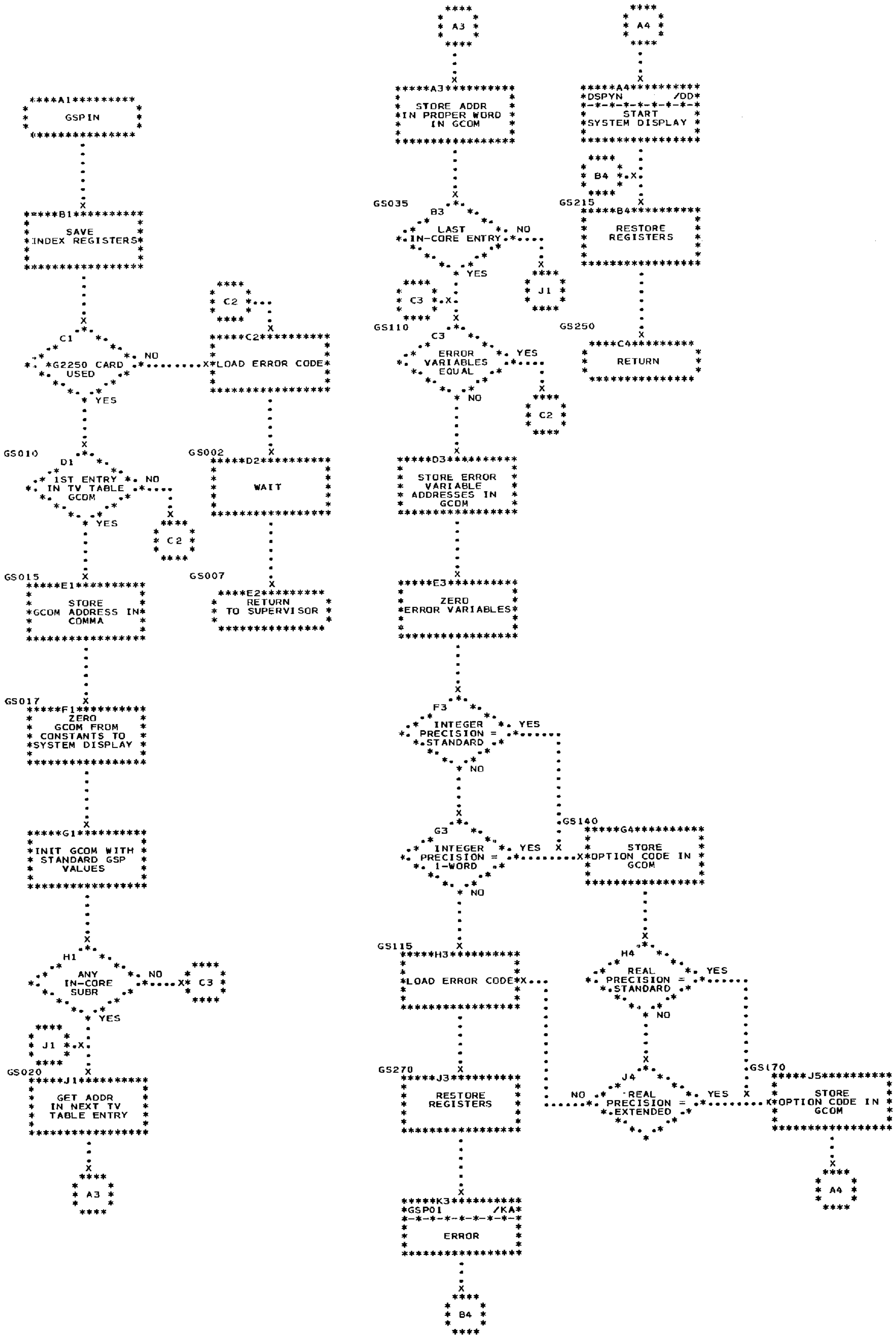


Chart AB. ICAIN - ICA Initialization

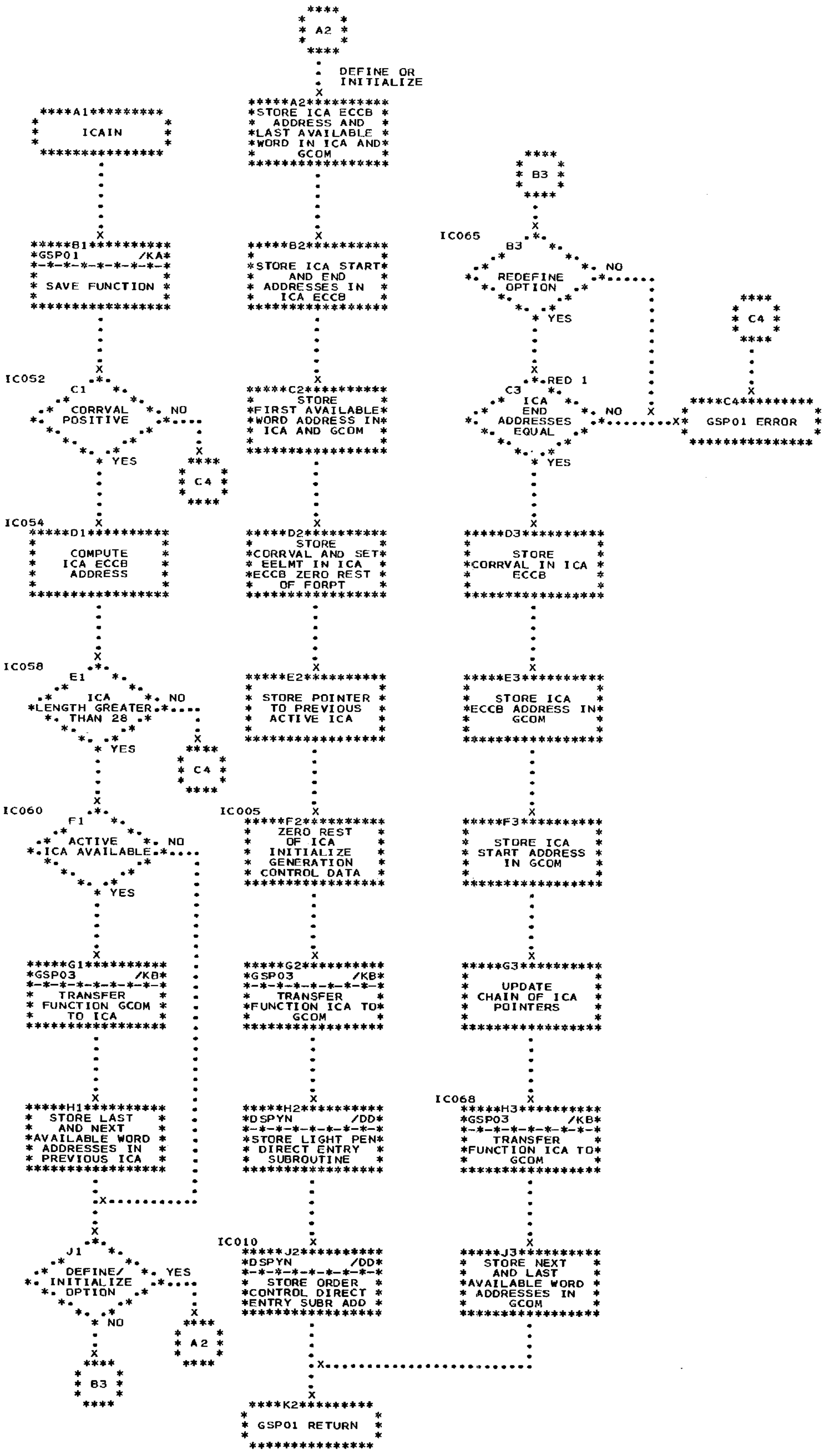

A2
DEFINE OR INITIALIZE
A1 ICAIN
A2 STORE ICA ECCB ADDRESS AND LAST AVAILABLE WORD IN ICA AND GCOM
B3
B1 GSP01 /KA
SAVE FUNCTION
B2 STORE ICA START AND END ADDRESSES IN ICA ECCB
IC065
B3 REDEFINE OPTION
NO
YES
C4
IC052
C1 CORRVAL POSITIVE
NO
YES
C4
C2 STORE FIRST AVAILABLE WORD ADDRESS IN ICA AND GCOM
RED 1
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 STORE ICA START ADDRESS IN GCOM
G1 GSP03 /KB
TRANSFER FUNCTION GCOM TO ICA
G2 GSP03 /KB
TRANSFER FUNCTION ICA TO GCOM
G3 UPDATE CHAIN OF ICA POINTERS
H1 STORE LAST AND NEXT AVAILABLE WORD ADDRESSES IN PREVIOUS ICA
H2 DSPYN /DD
STORE LIGHT PEN DIRECT ENTRY SUBROUTINE
IC068
H3 GSP03 /KB
TRANSFER FUNCTION ICA TO GCOM
J1 DEFINE/ INITIALIZE OPTION
YES
NO
A2
B3
IC010
J2 DSPYN /DD
STORE ORDER CONTROL DIRECT ENTRY SUBR ADD
J3 STORE NEXT AND LAST AVAILABLE WORD ADDRESSES IN GCOM
K2 GSP01 RETURN

Chart AC. GCAIN - GCA Initialization

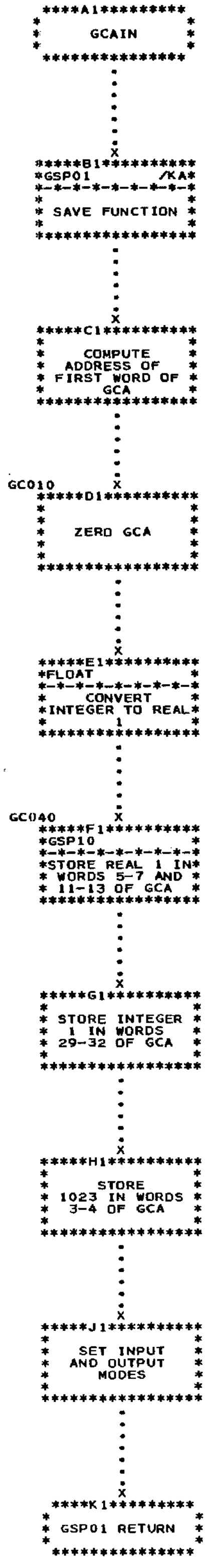

A1
GCAIN
B1
GSP01
/KA
SAVE FUNCTION
C1
COMPUTE ADDRESS OF FIRST WORD OF GCA
GC010
D1
ZERO GCA
E1
FLOAT
CONVERT INTEGER TO REAL 1
GC040
F1
GSP10
STORE REAL 1 IN WORDS 5-7 AND 11-13 OF GCA
G1
STORE INTEGER 1 IN WORDS 29-32 OF GCA
H1
STORE 1023 IN WORDS 3-4 OF GCA
J1
SET INPUT AND OUTPUT MODES
K1
GSP01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

Chart AD. SSCAL - Set Scaling Values

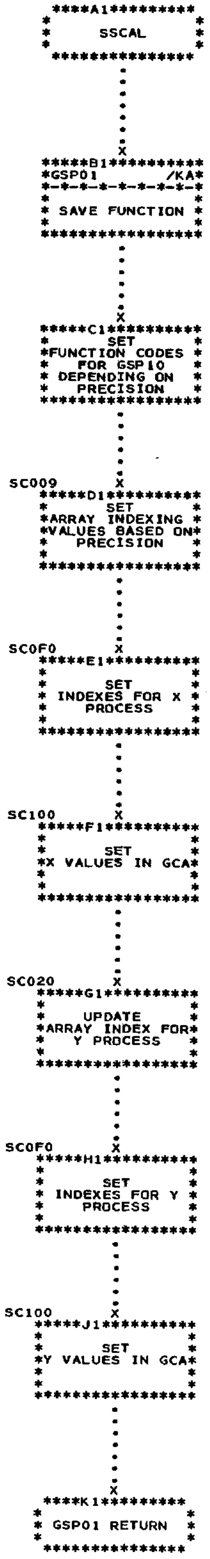

A1
SSCAL
B1
GSP01 /KA
SAVE FUNCTION
C1
SET FUNCTION CODES FOR GSP10 DEPENDING ON PRECISION
SC009
D1
SET ARRAY INDEXING VALUES BASED ON PRECISION
SC0F0
E1
SET INDEXES FOR X PROCESS
SC100
F1
SET X VALUES IN GCA
SC020
G1
UPDATE ARRAY INDEX FOR Y PROCESS
SC0F0
H1
SET INDEXES FOR Y PROCESS
SC100
J1
SET Y VALUES IN GCA
K1
GSP01 RETURN

Chart AE. SSCIS - Set Scissoring Option
SINDX - Set Index Values
SINCR - Set Increment Values
SGRAM - Set Output Graphic Mode
SDATM - Set Input Data Mode

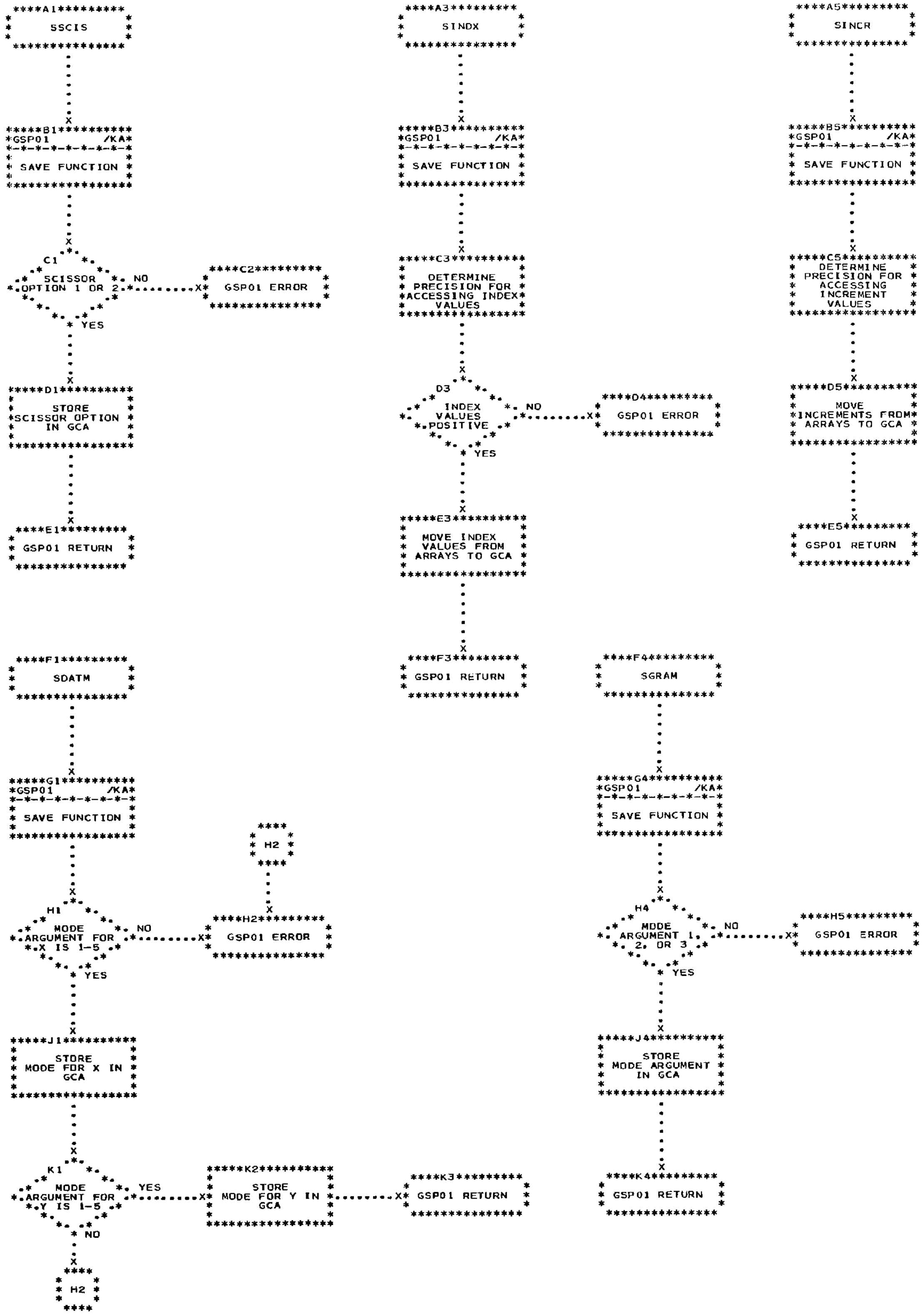


Chart AF. GSPTM - GSP Termination

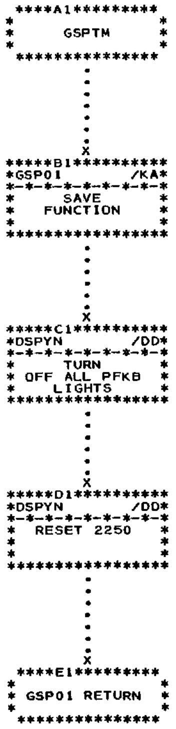


Chart BA. BELMT - Begin Element

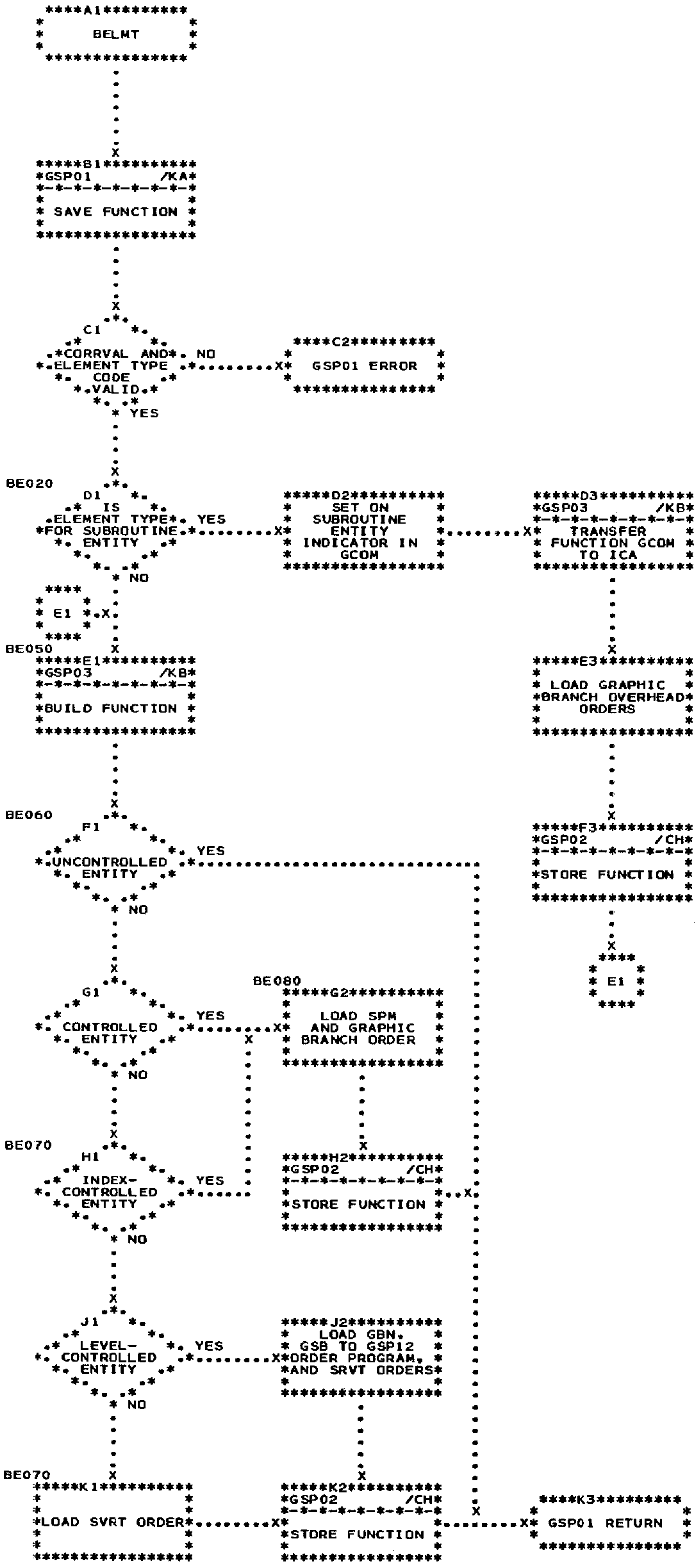


Chart BB. EELMT - End Element

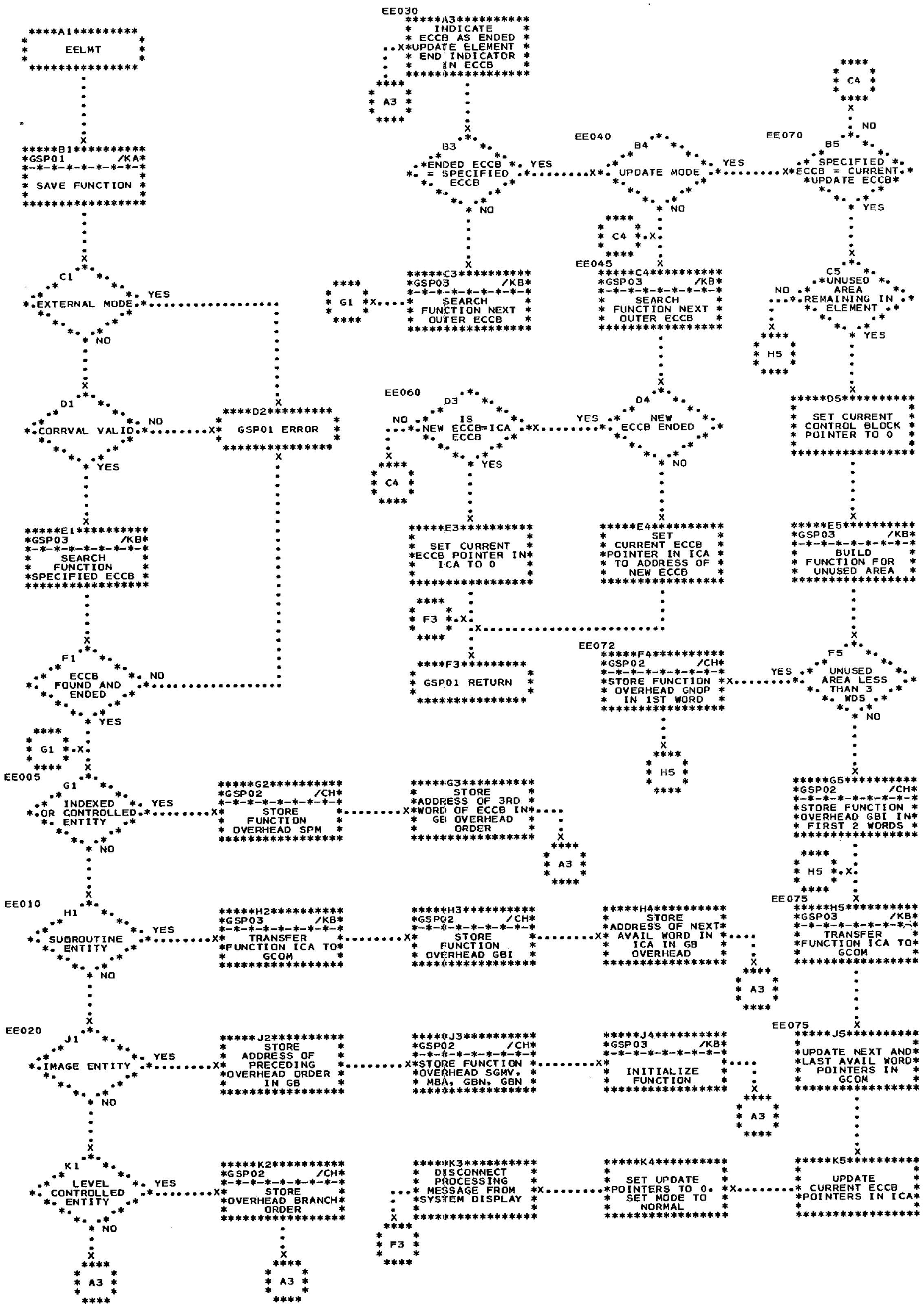

A1 EELMT
B1 GSP01 /KA SAVE FUNCTION
C1 EXTERNAL MODE
YES
NO
D1 CORRVAL VALID
NO
YES
D2 GSP01 ERROR
E1 GSP03 /KB SEARCH FUNCTION SPECIFIED ECCB
F1 ECCB FOUND AND ENDED
NO
YES
G1
EE005
G1 INDEXED OR CONTROLLED ENTITY
YES
NO
G2 GSP02 /CH STORE FUNCTION OVERHEAD SPM
G3 STORE ADDRESS OF 3RD WORD OF ECCB IN GB OVERHEAD ORDER
A3
EE010
H1 SUBROUTINE ENTITY
YES
NO
H2 GSP03 /KB TRANSFER FUNCTION ICA TO GCOM
H3 GSP02 /CH STORE FUNCTION OVERHEAD GBI
H4 STORE ADDRESS OF NEXT AVAIL WORD IN ICA IN GB OVERHEAD
A3
EE020
J1 IMAGE ENTITY
YES
NO
J2 STORE ADDRESS OF PRECEDING OVERHEAD ORDER IN GB
J3 GSP02 /CH STORE FUNCTION OVERHEAD SGMV, MBA, GBN, GBN
J4 GSP03 /KB INITIALIZE FUNCTION
A3
K1 LEVEL CONTROLLED ENTITY
YES
NO
K2 GSP02 /CH STORE OVERHEAD BRANCH ORDER
K3 DISCONNECT PROCESSING MESSAGE FROM SYSTEM DISPLAY
F3
K4 SET UPDATE POINTERS TO 0. SET MODE TO NORMAL
K5 UPDATE CURRENT ECCB POINTERS IN ICA
A3
A3
EE030
A3 INDICATE ECCB AS ENDED UPDATE ELEMENT END INDICATOR IN ECCB
A3
B3 ENDED ECCB = SPECIFIED ECCB
YES
NO
G1
C3 GSP03 /KB SEARCH FUNCTION NEXT OUTER ECCB
EE060
D3 IS NEW ECCB=ICA ECCB
NO
YES
C4
E3 SET CURRENT ECCB POINTER IN ICA TO 0
F3
F3 GSP01 RETURN
EE040
B4 UPDATE MODE
YES
NO
C4
EE045
C4 GSP03 /KB SEARCH FUNCTION NEXT OUTER ECCB
D4 NEW ECCB ENDED
YES
NO
E4 SET CURRENT ECCB POINTER IN ICA TO ADDRESS OF NEW ECCB
EE072
F4 GSP02 /CH STORE FUNCTION IN 1ST WORD OVERHEAD GNOP
H5
C4
EE070
B5 SPECIFIED ECCB = CURRENT UPDATE ECCB
NO
YES
C5 UNUSED AREA REMAINING IN ELEMENT
NO
YES
H5
D5 SET CURRENT CONTROL BLOCK POINTER TO 0
E5 GSP03 /KB BUILD FUNCTION FOR UNUSED AREA
F5 UNUSED AREA LESS THAN 3 WDS
YES
NO
G5 GSP02 /CH STORE FUNCTION OVERHEAD GBI IN FIRST 2 WORDS
H5
EE075
H5 GSP03 /KB TRANSFER FUNCTION ICA TO GCOM
EE075
J5 UPDATE NEXT AND LAST AVAIL WORD POINTERS IN GCOM

Chart BC. DELMT - Delete Element

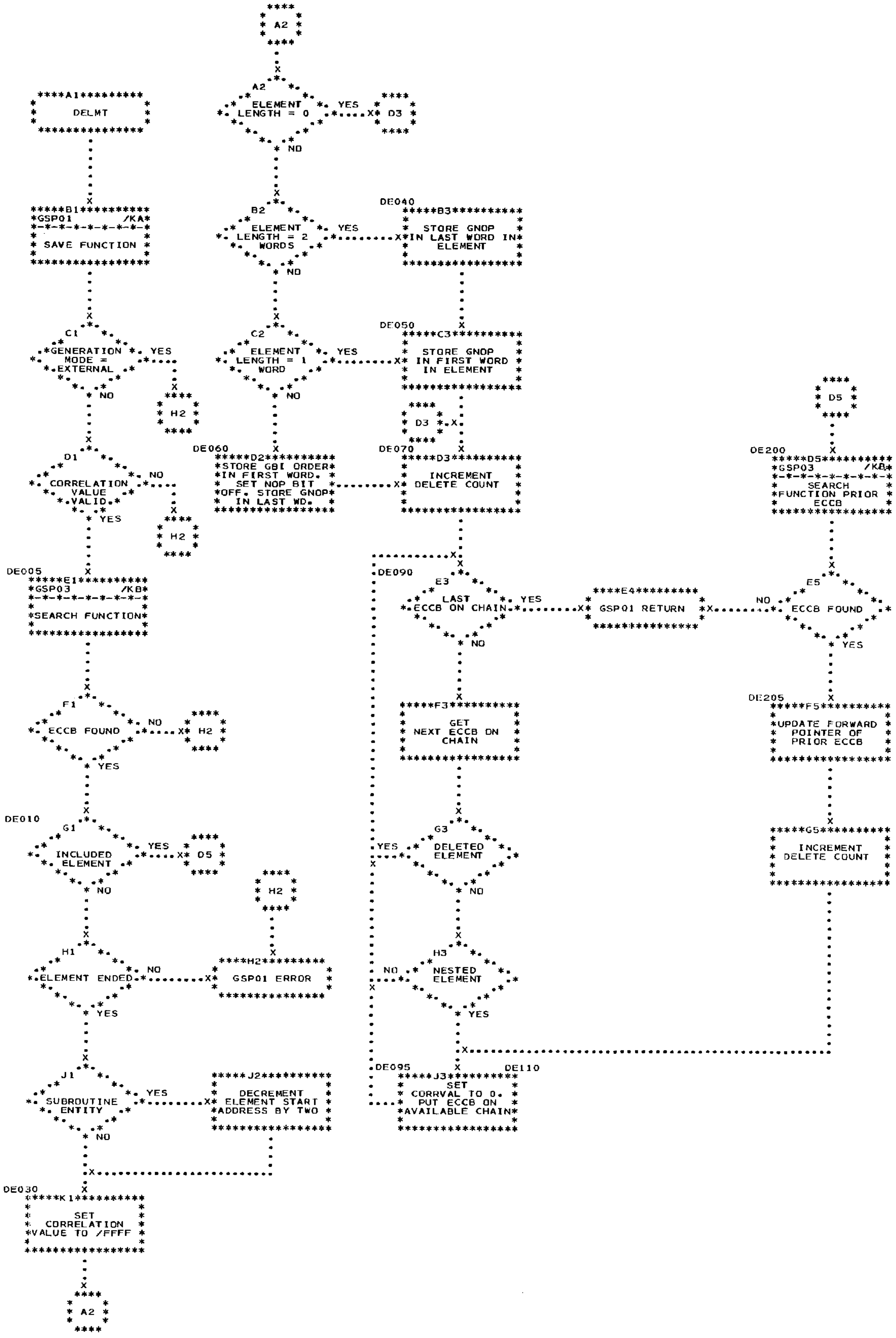


Chart BD. UELMT - Update Element

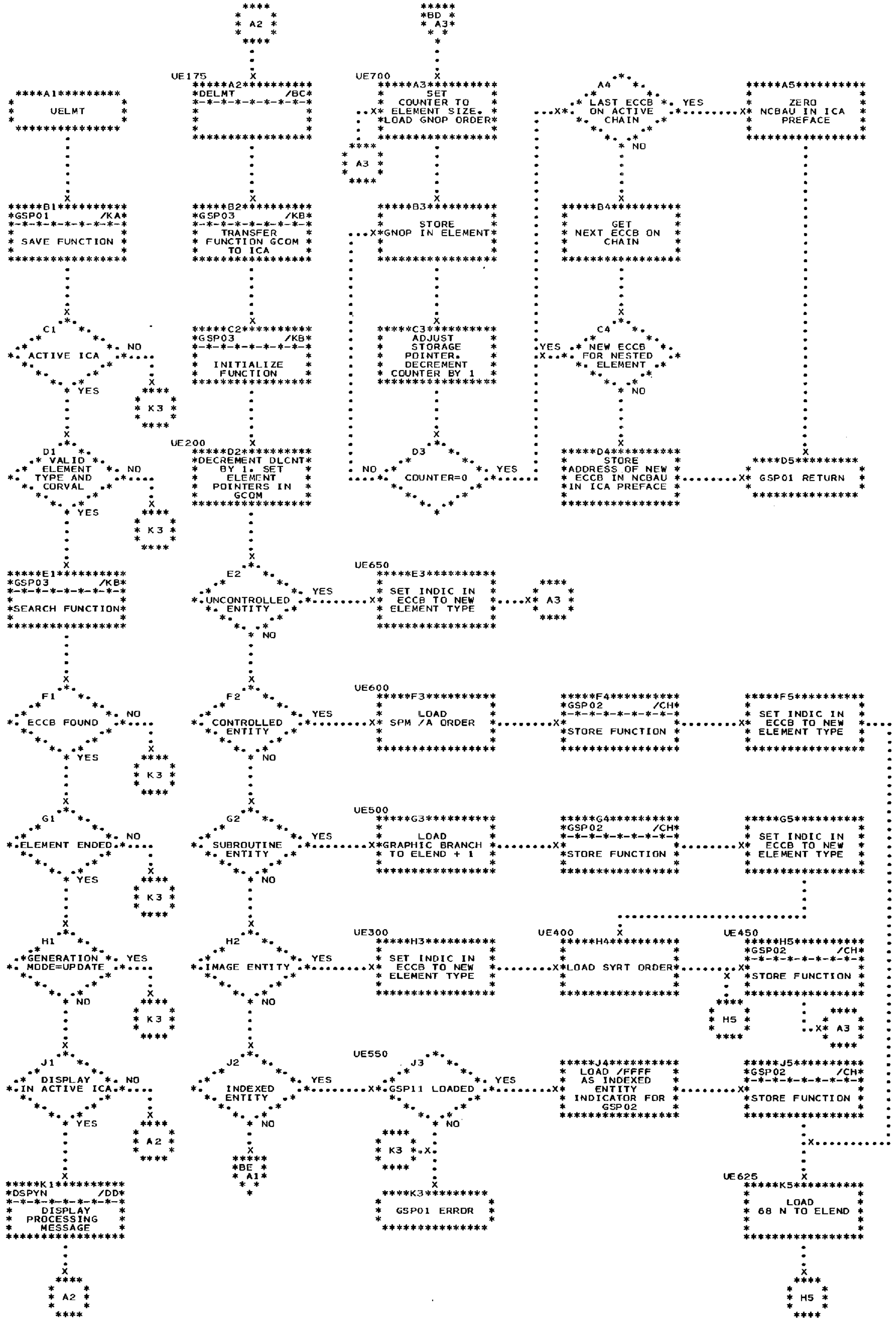


Chart BE. UELMT - Update Element (continued)

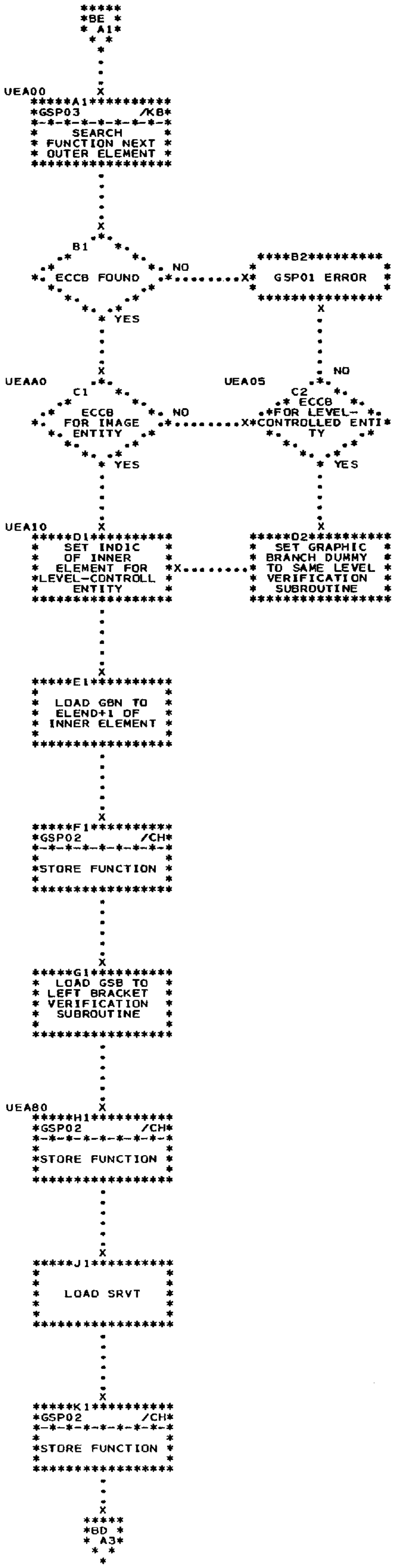

BE
A1
UEA00
A1
GSP03 /KB
SEARCH FUNCTION NEXT OUTER ELEMENT
B1
ECCB FOUND
NO
B2
GSP01 ERROR
YES
NO
UEAA0
C1
ECCB FOR IMAGE ENTITY
NO
UEA05
C2
ECCB FOR LEVEL-CONTROLLED ENTITY
YES
YES
UEA10
D1
SET INDIC OF INNER ELEMENT FOR LEVEL-CONTROLL ENTITY
D2
SET GRAPHIC BRANCH DUMMY TO SAME LEVEL VERIFICATION SUBROUTINE
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

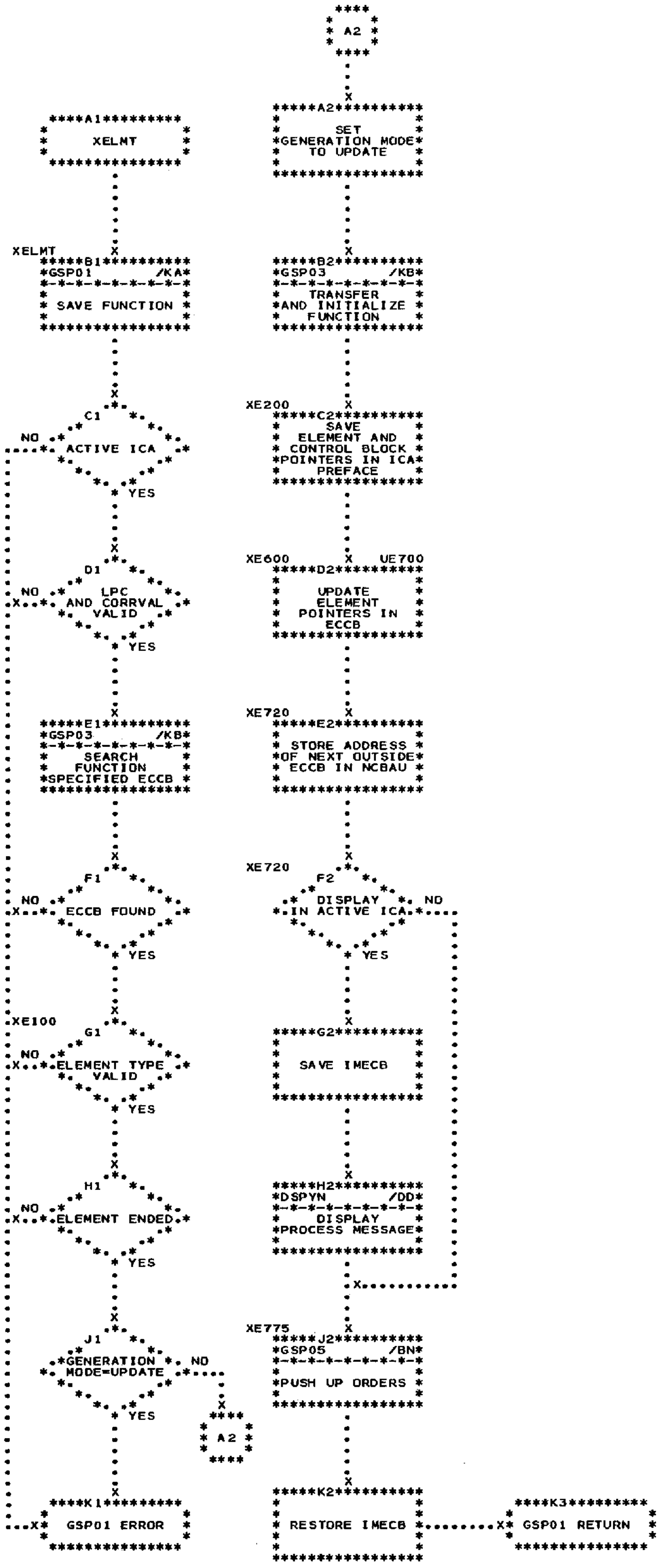


Chart BG. IELMT - Include Element

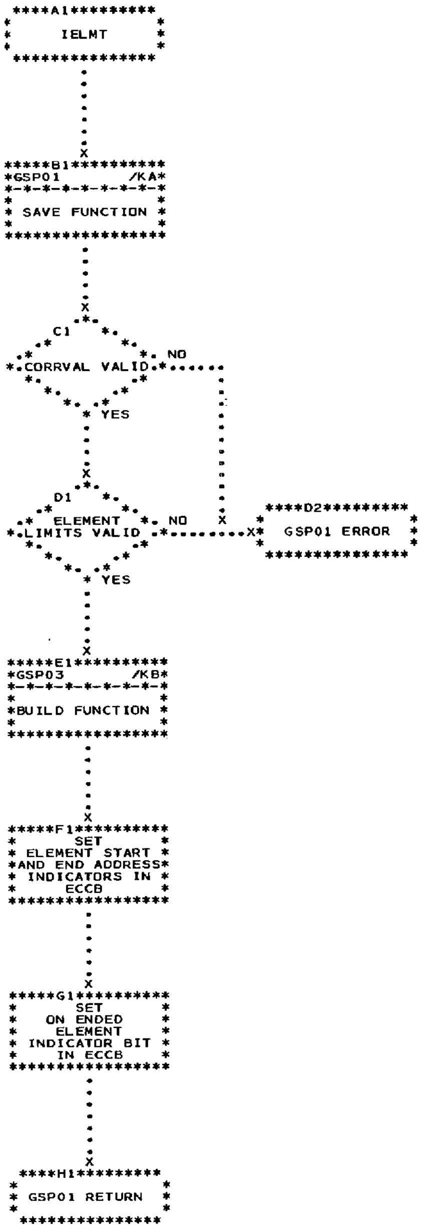


Chart BH. BXGEN - Begin External Generation

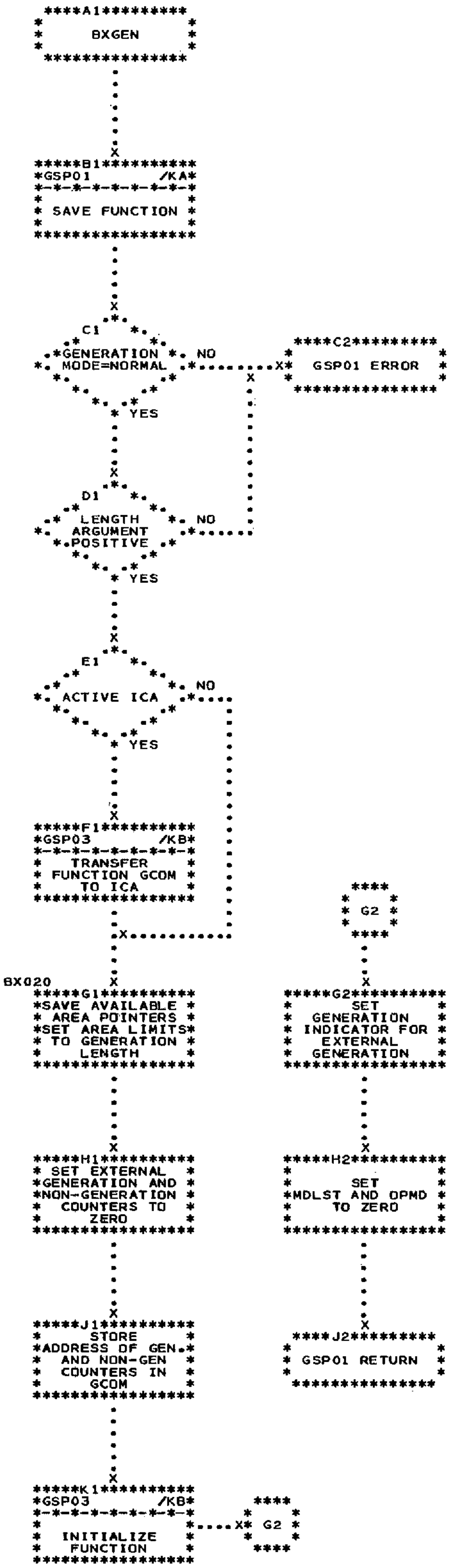

A1
BXGEN
B1
GSP01 /KA
SAVE FUNCTION
C1
GENERATION MODE=NORMAL
NO
C2
GSP01 ERROR
YES
D1
LENGTH ARGUMENT POSITIVE
NO
YES
E1
ACTIVE ICA
NO
YES
F1
GSP03 /KB
TRANSFER FUNCTION GCOM TO ICA
G2
BX020
G1
SAVE AVAILABLE AREA POINTERS SET AREA LIMITS TO GENERATION LENGTH
G2
SET GENERATION INDICATOR FOR EXTERNAL GENERATION
H1
SET EXTERNAL GENERATION AND NON-GENERATION COUNTERS TO ZERO
H2
SET MDLST AND OPMD TO ZERO
J1
STORE ADDRESS OF GEN. AND NON-GEN COUNTERS IN GCOM
J2
GSP01 RETURN
K1
GSP03 /KB
INITIALIZE FUNCTION
G2

Chart BJ. EXGEN - End External Generation

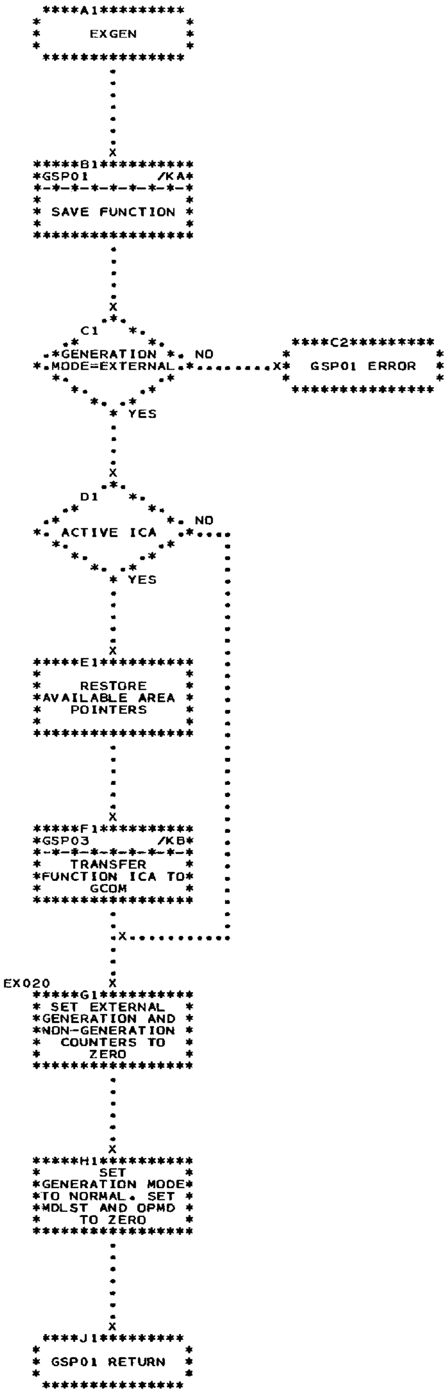

A1
EXGEN
B1
GSP01 /KA
SAVE FUNCTION
C1
GENERATION MODE=EXTERNAL
NO
C2
GSP01 ERROR
YES
D1
ACTIVE ICA
NO
YES
E1
RESTORE AVAILABLE AREA POINTERS
F1
GSP03 /KB
TRANSFER FUNCTION ICA TO GCOM
EX020
G1
SET EXTERNAL GENERATION AND NON-GENERATION COUNTERS TO ZERO
H1
SET GENERATION MODE TO NORMAL. SET MDLST AND OPMD TO ZERO
J1
GSP01 RETURN

Chart BK. SATRB - Set Controlled Entity Attributes

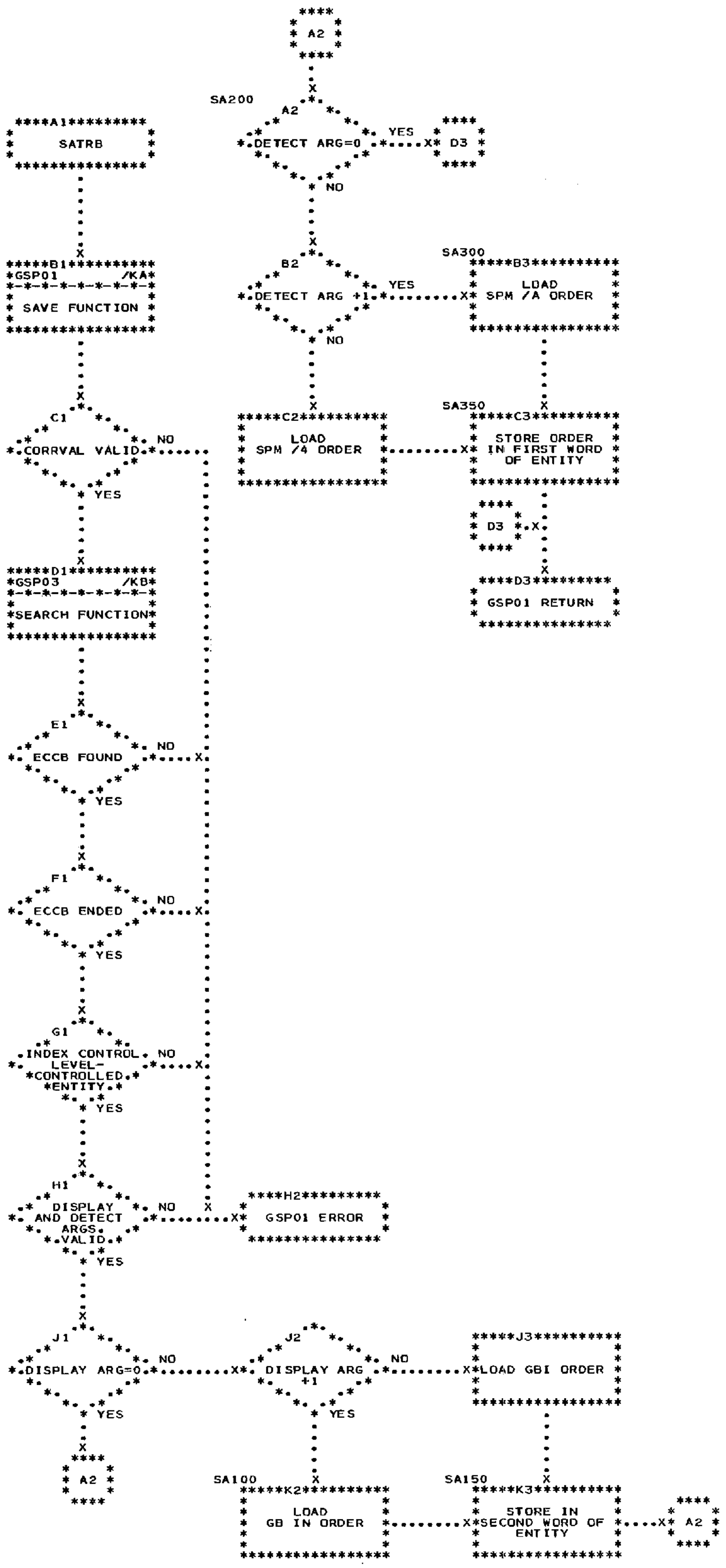


Chart BL. STPVS - Step Verification Setting

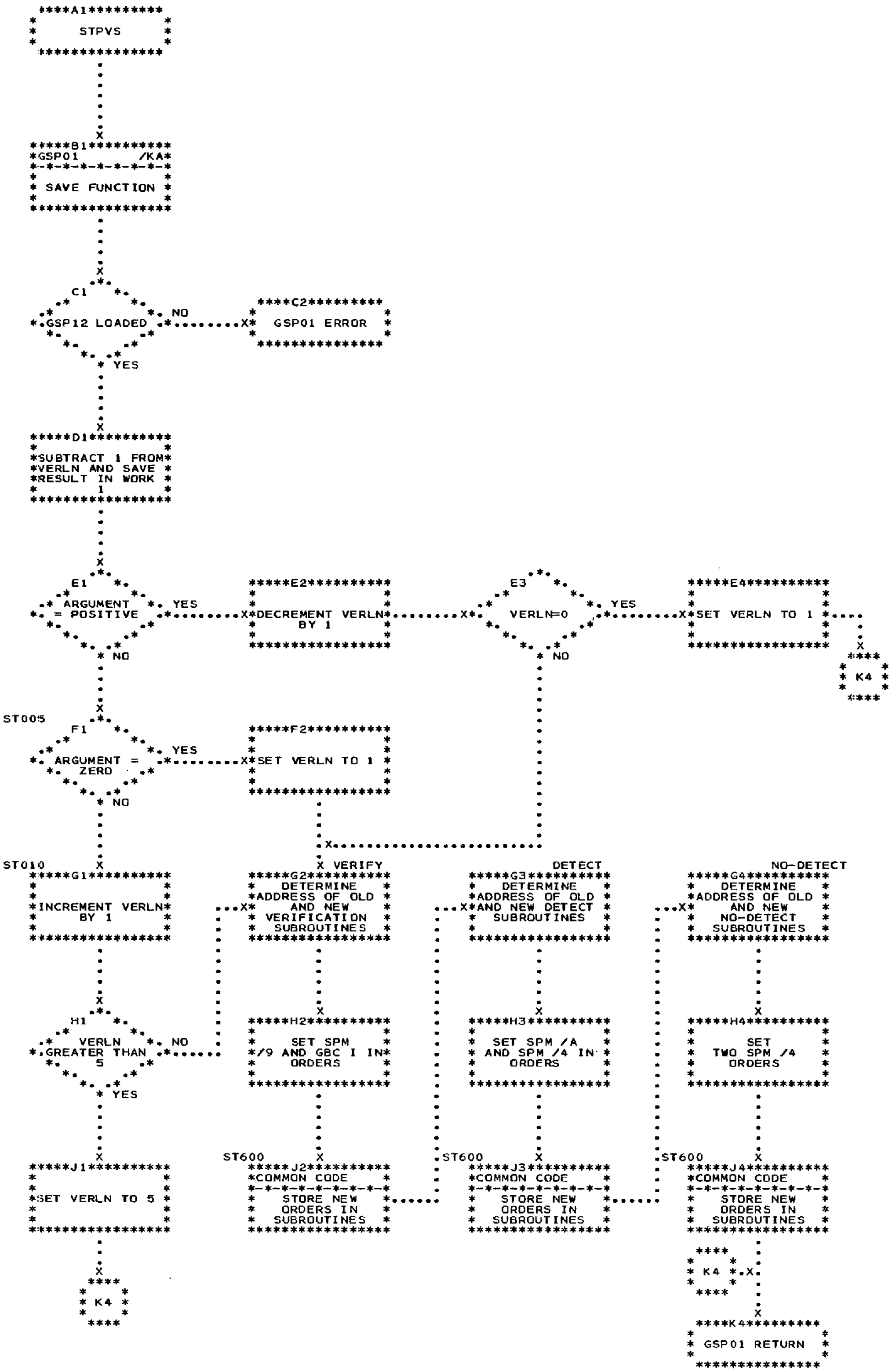


Chart BM. GSP04 - Push Down

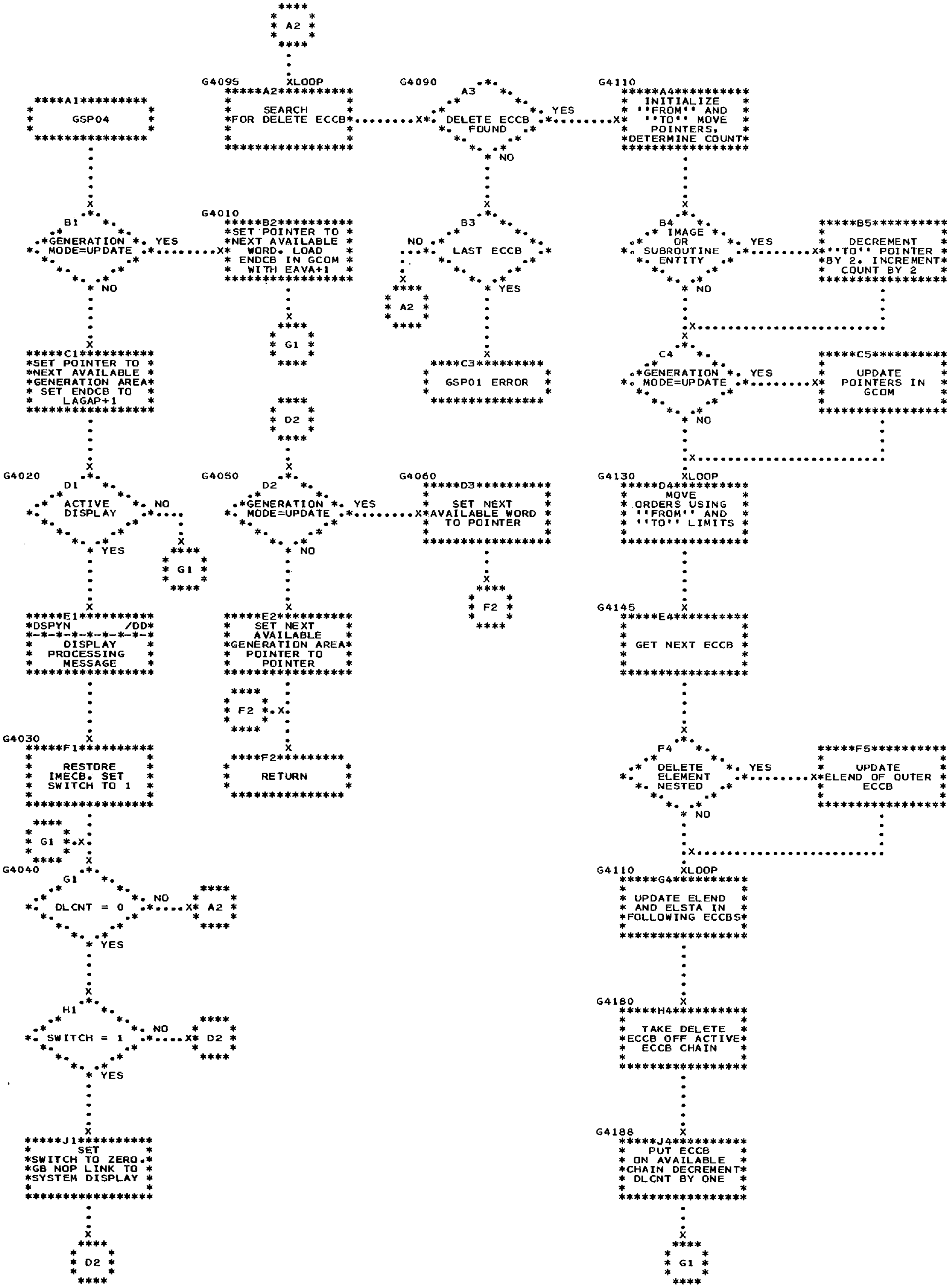

A2
G4095
XLOOP
A2
SEARCH FOR DELETE ECCB
G4090
A3
DELETE ECCB FOUND
YES
NO
G4110
A4
INITIALIZE ''FROM'' AND ''TO'' MOVE POINTERS. DETERMINE COUNT
A1
GSP04
B1
GENERATION MODE=UPDATE
YES
NO
G4010
B2
SET POINTER TO NEXT AVAILABLE WORD. LOAD ENDCB IN GCOM WITH EAVA+1
G1
B3
LAST ECCB
NO
A2
YES
B4
IMAGE OR 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
C3
GSP01 ERROR
C4
GENERATION MODE=UPDATE
YES
NO
C5
UPDATE POINTERS IN GCOM
D2
G4020
D1
ACTIVE DISPLAY
NO
G1
YES
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

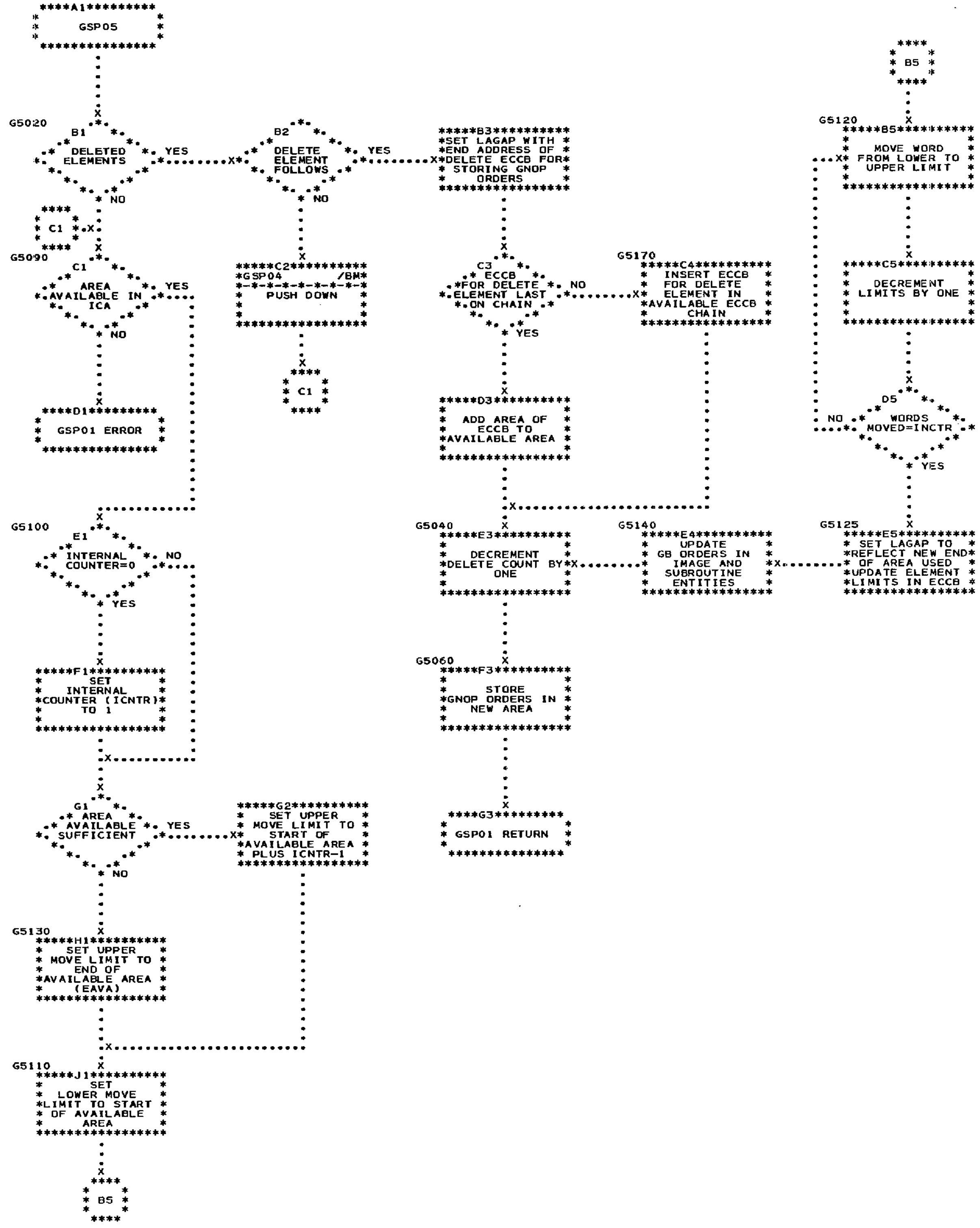

A1
GSP05
G5020
B1
DELETED ELEMENTS
YES
NO
B2
DELETE ELEMENT FOLLOWS
YES
NO
B3
SET LAGAP WITH END ADDRESS OF DELETE ECCB FOR STORING GNOP ORDERS
C1
G5090
C1
AREA AVAILABLE IN ICA
YES
NO
C2
GSP04 /BM
PUSH DOWN
C1
C3
ECCB FOR DELETE ELEMENT LAST ON CHAIN
NO
YES
G5170
C4
INSERT ECCB FOR DELETE ELEMENT IN AVAILABLE ECCB CHAIN
D1
GSP01 ERROR
D3
ADD AREA OF ECCB TO AVAILABLE AREA
B5
G5120
B5
MOVE WORD FROM LOWER TO UPPER LIMIT
C5
DECREMENT LIMITS BY ONE
D5
WORDS MOVED=INCTR
NO
YES
G5100
E1
INTERNAL COUNTER=0
NO
YES
G5040
E3
DECREMENT DELETE COUNT BY ONE
G5140
E4
UPDATE GB ORDERS IN IMAGE AND SUBROUTINE ENTITIES
G5125
E5
SET LAGAP TO REFLECT NEW END OF AREA USED UPDATE ELEMENT LIMITS IN ECCB
F1
SET INTERNAL COUNTER (ICNTR) TO 1
G5060
F3
STORE GNOP ORDERS IN NEW AREA
G1
AREA AVAILABLE SUFFICIENT
YES
NO
G2
SET UPPER MOVE LIMIT TO START OF AVAILABLE AREA PLUS ICNTR-1
G3
GSP01 RETURN
G5130
H1
SET UPPER MOVE LIMIT TO END OF AVAILABLE AREA (EAVA)
G5110
J1
SET LOWER MOVE LIMIT TO START OF AVAILABLE AREA
B5

Chart CA. MVPOS - Move Element to a Position

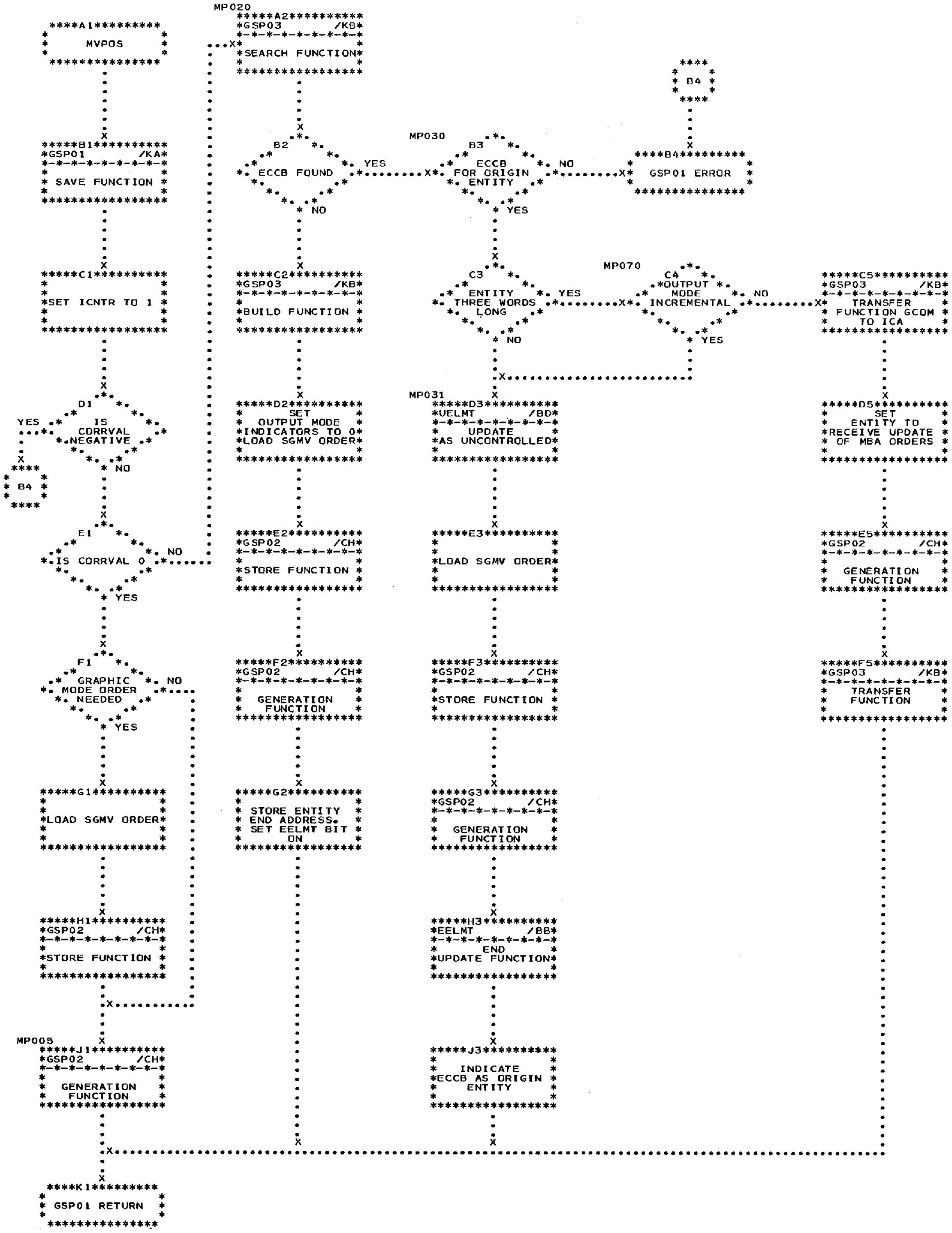


Chart CB. PLINE/PPNT/PSGMT - Plot Lines, Points, Segments

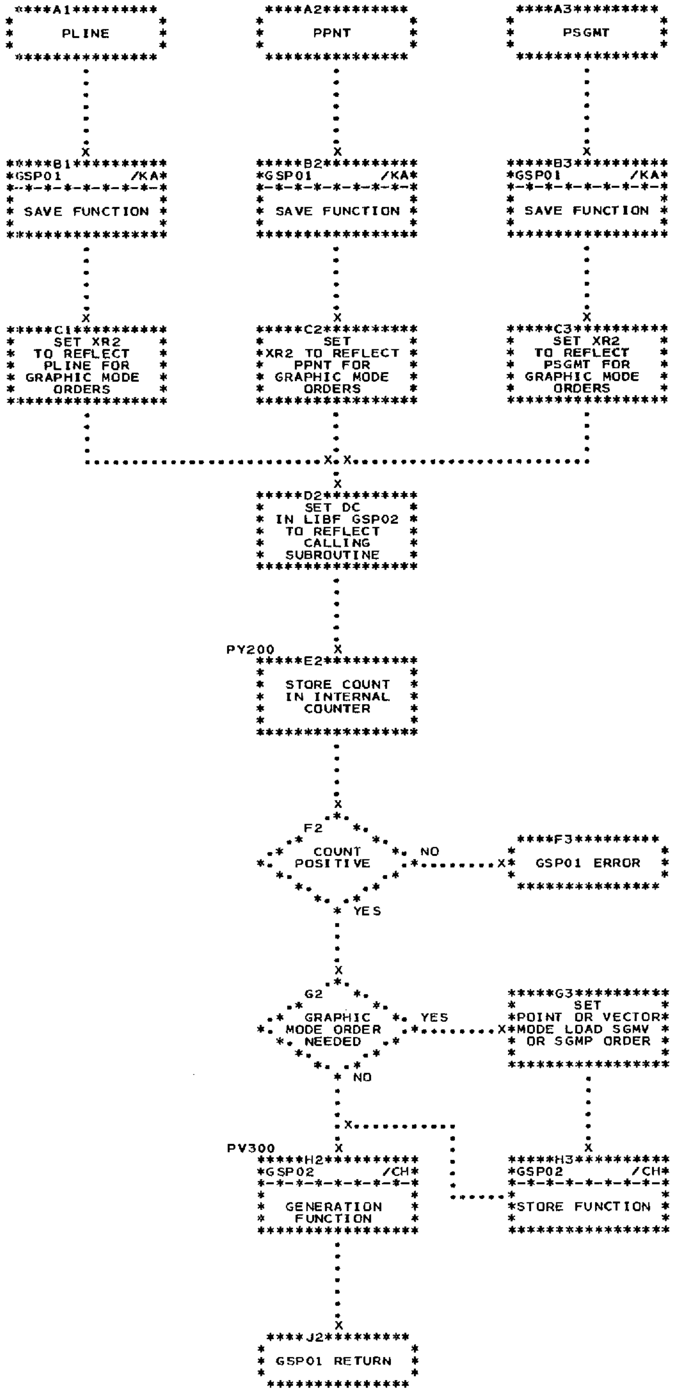

A1
PLINE
A2
PPNT
A3
PSGMT
B1
GSP01 /KA
SAVE FUNCTION
B2
GSP01 /KA
SAVE FUNCTION
B3
GSP01 /KA
SAVE FUNCTION
C1
SET XR2 TO REFLECT PLINE FOR GRAPHIC MODE ORDERS
C2
SET XR2 TO REFLECT PPNT FOR GRAPHIC MODE ORDERS
C3
SET XR2 TO REFLECT PSGMT FOR GRAPHIC MODE ORDERS
D2
SET DC IN LIBF GSP02 TO REFLECT CALLING SUBROUTINE
PY200
E2
STORE COUNT IN INTERNAL COUNTER
F2
COUNT POSITIVE
NO
YES
F3
GSP01 ERROR
G2
GRAPHIC MODE ORDER NEEDED
YES
NO
G3
SET POINT OR VECTOR MODE LOAD SGMV OR SGMP ORDER
PV300
H2
GSP02 /CH
GENERATION FUNCTION
H3
GSP02 /CH
STORE FUNCTION
J2
GSP01 RETURN

Chart CC. PTEXT - Plot Text

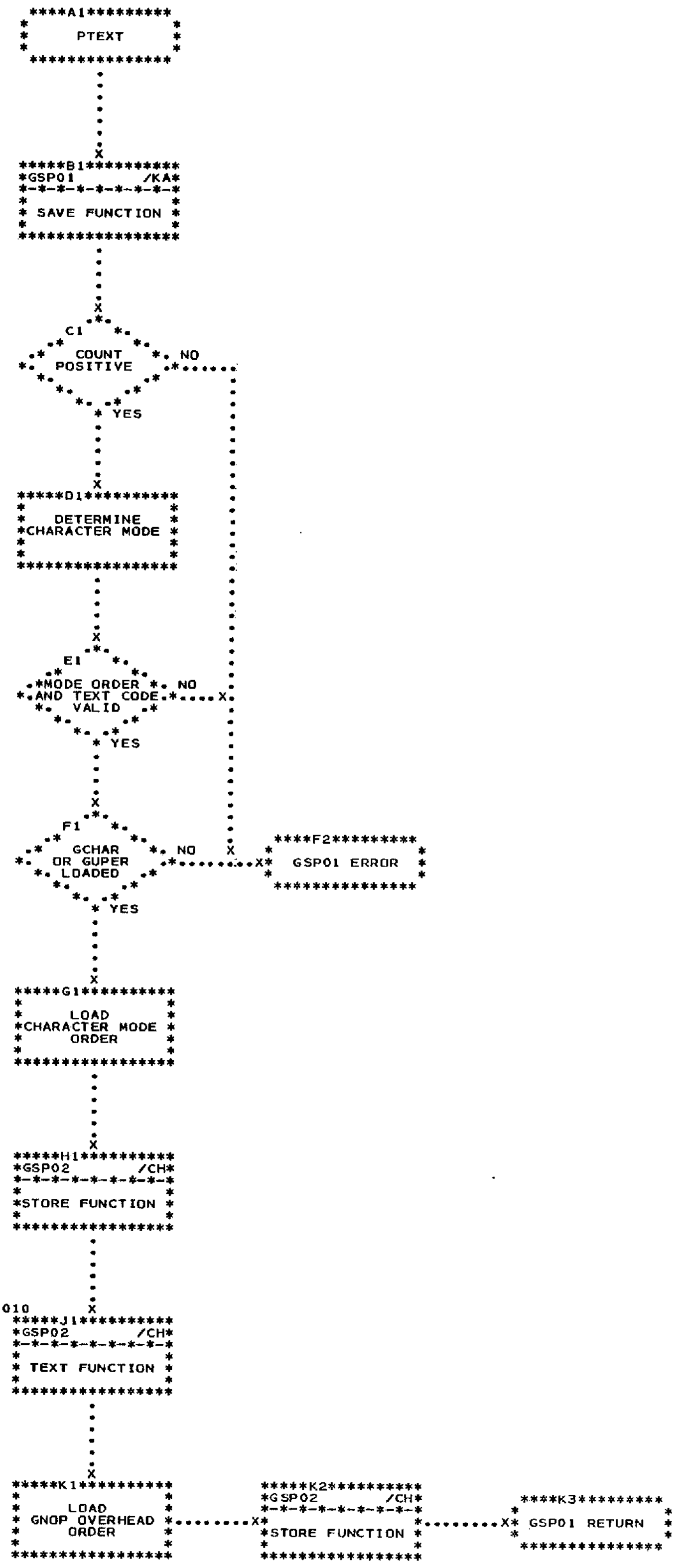


Chart CD. LKSUB - Linkage to a Subroutine

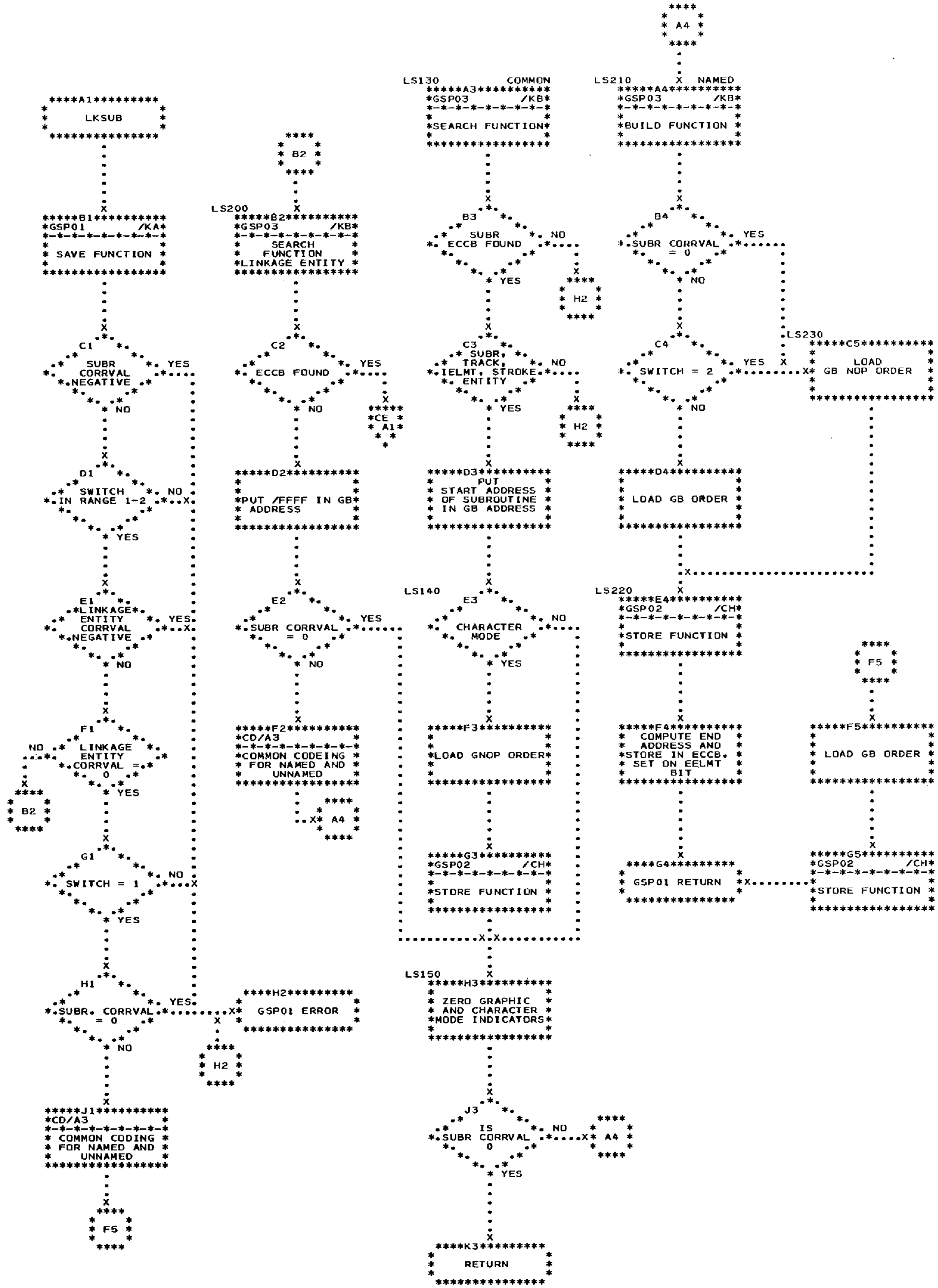


Chart CE. LKSUB - Linkage to a Subroutine (continued)

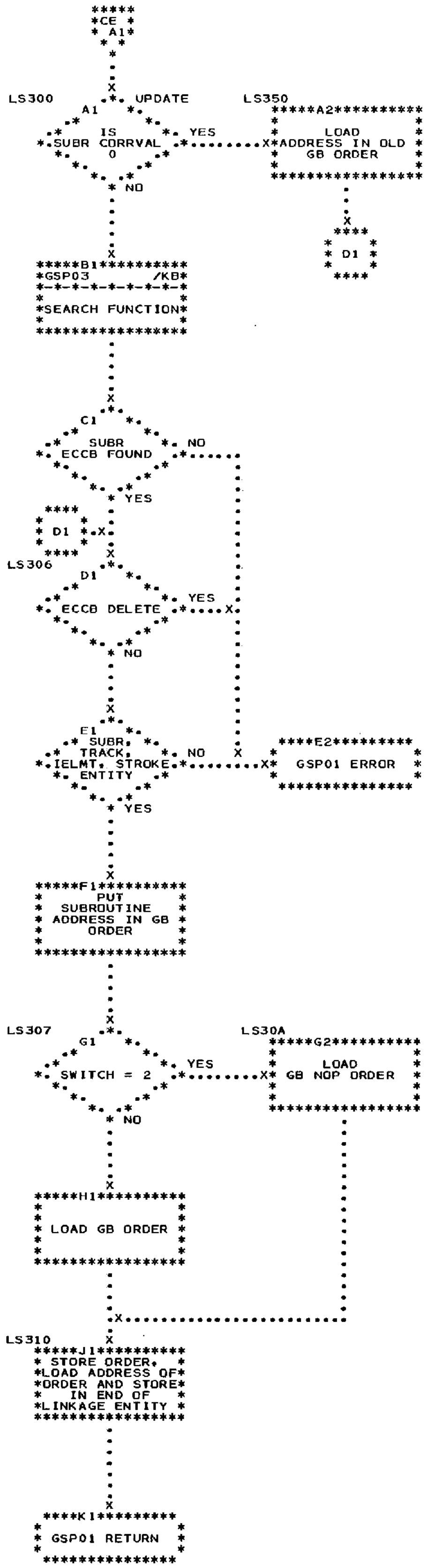


Chart CF. PGRID - Plot Grid Outline

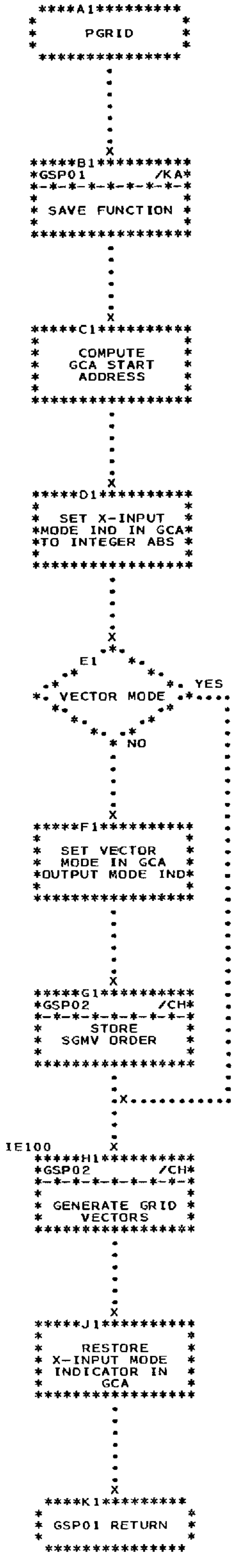

A1
PGRID
B1
GSP01 /KA
SAVE FUNCTION
C1
COMPUTE GCA START ADDRESS
D1
SET X-INPUT MODE IND IN GCA TO INTEGER ABS
E1
VECTOR MODE
YES
NO
F1
SET VECTOR MODE IN GCA OUTPUT MODE IND
G1
GSP02 /CH
STORE SGMV ORDER
IE100
H1
GSP02 /CH
GENERATE GRID VECTORS
J1
RESTORE X-INPUT MODE INDICATOR IN GCA
K1
GSP01 RETURN

Chart CG. PCOPY - Plot Copy of an Element

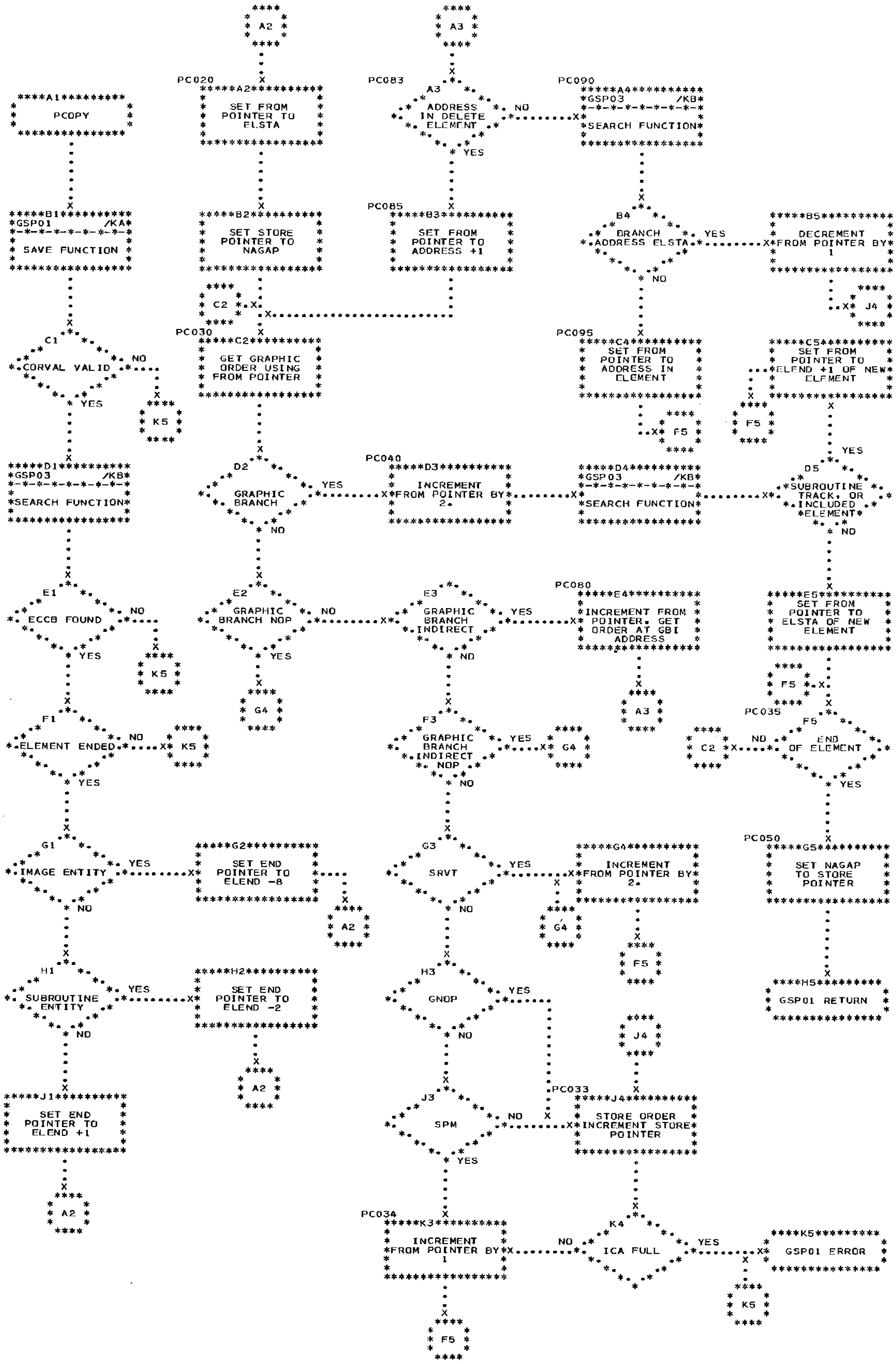

A2
A3
PC020
A2
SET FROM POINTER TO ELSTA
PC083
A3
ADDRESS IN DELETE ELEMENT
NO
YES
PC090
A4
GSP03 /KB
SEARCH FUNCTION
A1
PCOPY
B1
GSP01 /KA
SAVE FUNCTION
B2
SET STORE POINTER TO NAGAP
PC085
B3
SET FROM POINTER TO ADDRESS +1
B4
BRANCH ADDRESS ELSTA
YES
NO
B5
DECREMENT FROM POINTER BY 1
J4
C2
C1
CORVAL VALID
NO
YES
K5
PC030
C2
GET GRAPHIC ORDER USING FROM POINTER
PC095
C4
SET FROM POINTER TO ADDRESS IN ELEMENT
F5
C5
SET FROM POINTER TO ELEND +1 OF NEW ELEMENT
F5
D1
GSP03 /KB
SEARCH FUNCTION
D2
GRAPHIC BRANCH
YES
NO
PC040
D3
INCREMENT FROM POINTER BY 2.
D4
GSP03 /KB
SEARCH FUNCTION
D5
SUBROUTINE TRACK, OR INCLUDED ELEMENT
YES
NO
E1
ECCB FOUND
NO
YES
K5
E2
GRAPHIC BRANCH NOP
NO
YES
G4
E3
GRAPHIC BRANCH INDIRECT
YES
NO
PC080
E4
INCREMENT FROM POINTER, GET ORDER AT GBI ADDRESS
A3
E5
SET FROM POINTER TO ELSTA OF NEW ELEMENT
F5
F1
ELEMENT ENDED
NO
K5
YES
F3
GRAPHIC BRANCH INDIRECT NOP
YES
G4
NO
PC035
F5
END OF ELEMENT
NO
C2
YES
G1
IMAGE ENTITY
YES
NO
G2
SET END POINTER TO ELEND -8
A2
G3
SRVT
YES
NO
G4
INCREMENT FROM POINTER BY 2.
G4
F5
PC050
G5
SET NAGAP TO STORE POINTER
H1
SUBROUTINE ENTITY
YES
NO
H2
SET END POINTER TO ELEND -2
A2
H3
GNOP
YES
NO
H5
GSP01 RETURN
J4
J1
SET END POINTER TO ELEND +1
A2
J3
SPM
NO
YES
PC033
J4
STORE ORDER INCREMENT STORE POINTER
PC034
K3
INCREMENT FROM POINTER BY 1
F5
K4
ICA FULL
NO
YES
K5
GSP01 ERROR
K5

Chart CH. GSP02 - Generation Support

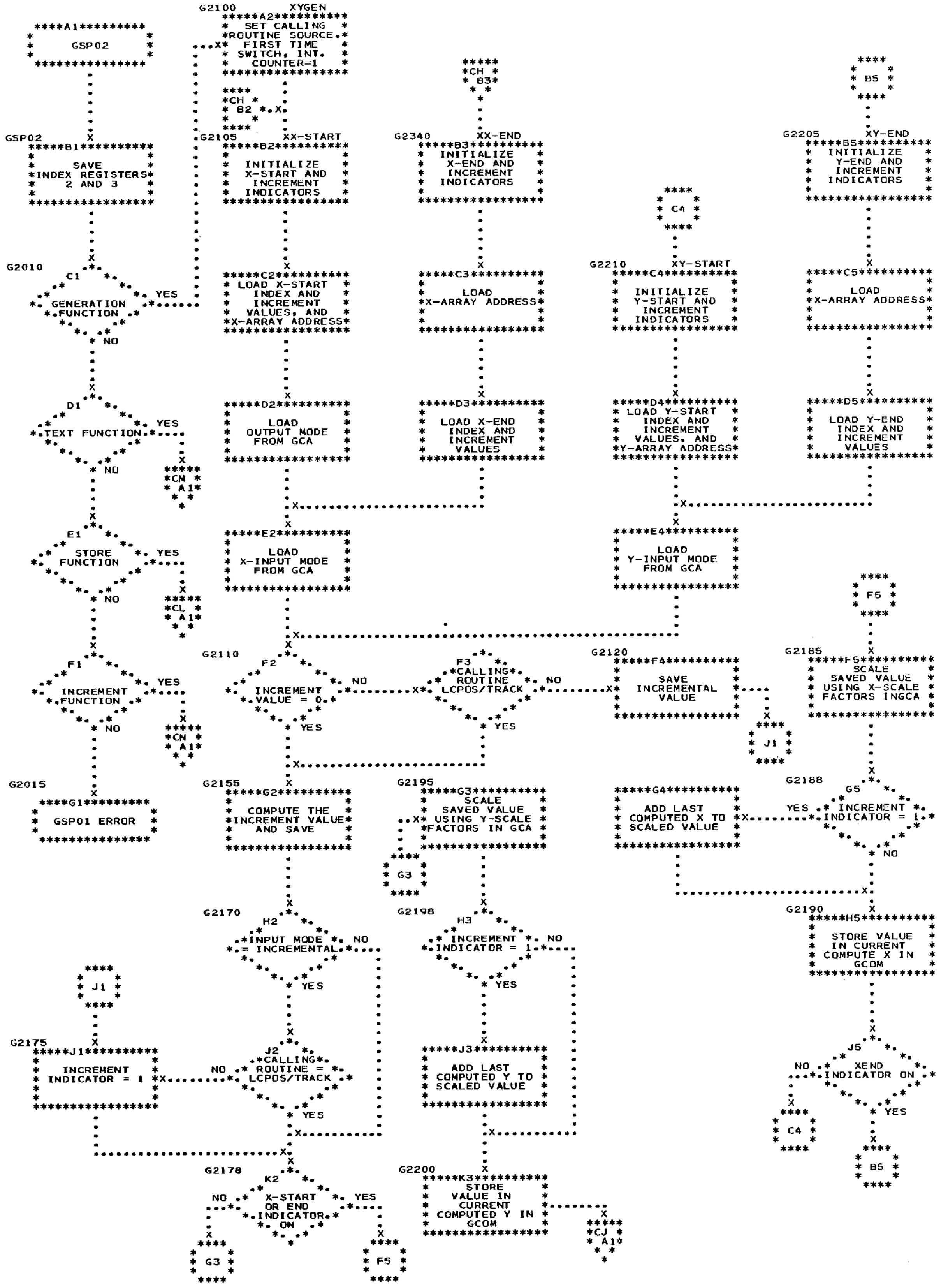

A1
GSP02
G2100 XYGEN
A2
SET CALLING ROUTINE SOURCE. FIRST TIME SWITCH. INT. COUNTER=1
CH B3
B5
CH B2
GSP02
B1
SAVE INDEX REGISTERS 2 AND 3
G2105 XX-START
B2
INITIALIZE X-START AND INCREMENT INDICATORS
G2340 XX-END
B3
INITIALIZE X-END AND INCREMENT INDICATORS
G2205 XY-END
B5
INITIALIZE Y-END AND INCREMENT INDICATORS
C4
G2010
C1
GENERATION FUNCTION
YES
NO
C2
LOAD X-START INDEX AND INCREMENT VALUES, AND X-ARRAY ADDRESS
C3
LOAD X-ARRAY ADDRESS
G2210 XY-START
C4
INITIALIZE Y-START AND INCREMENT INDICATORS
C5
LOAD X-ARRAY ADDRESS
D1
TEXT FUNCTION
YES
NO
CM A1
D2
LOAD OUTPUT MODE FROM GCA
D3
LOAD X-END INDEX AND INCREMENT VALUES
D4
LOAD Y-START INDEX AND INCREMENT VALUES, AND Y-ARRAY ADDRESS
D5
LOAD Y-END INDEX AND INCREMENT VALUES
E1
STORE FUNCTION
YES
NO
CL A1
E2
LOAD X-INPUT MODE FROM GCA
E4
LOAD Y-INPUT MODE FROM GCA
F5
F1
INCREMENT FUNCTION
YES
NO
CN A1
G2110
F2
INCREMENT VALUE = 0.
NO
YES
F3
CALLING ROUTINE LCPOS/TRACK
NO
YES
G2120
F4
SAVE INCREMENTAL VALUE
J1
G2185
F5
SCALE SAVED VALUE USING X-SCALE FACTORS INGCA
G2015
G1
GSP01 ERROR
G2155
G2
COMPUTE THE INCREMENT VALUE AND SAVE
G2195
G3
SCALE SAVED VALUE USING Y-SCALE FACTORS IN GCA
G3
G4
ADD LAST COMPUTED X TO SCALED VALUE
G2188
G5
INCREMENT INDICATOR = 1.
YES
NO
G2170
H2
INPUT MODE = INCREMENTAL.
NO
YES
G2198
H3
INCREMENT INDICATOR = 1.
NO
YES
G2190
H5
STORE VALUE IN CURRENT COMPUTE X IN GCOM
J1
G2175
J1
INCREMENT INDICATOR = 1
J2
CALLING ROUTINE = LCPOS/TRACK
NO
YES
J3
ADD LAST COMPUTED Y TO SCALED VALUE
J5
XEND INDICATOR ON
NO
YES
C4
B5
G2178
K2
X-START OR END INDICATOR. ON
NO
YES
G3
F5
G2200
K3
STORE VALUE IN CURRENT COMPUTED Y IN GCOM
CJ A1

Chart CJ. GSP02 - Generation Support (continued)

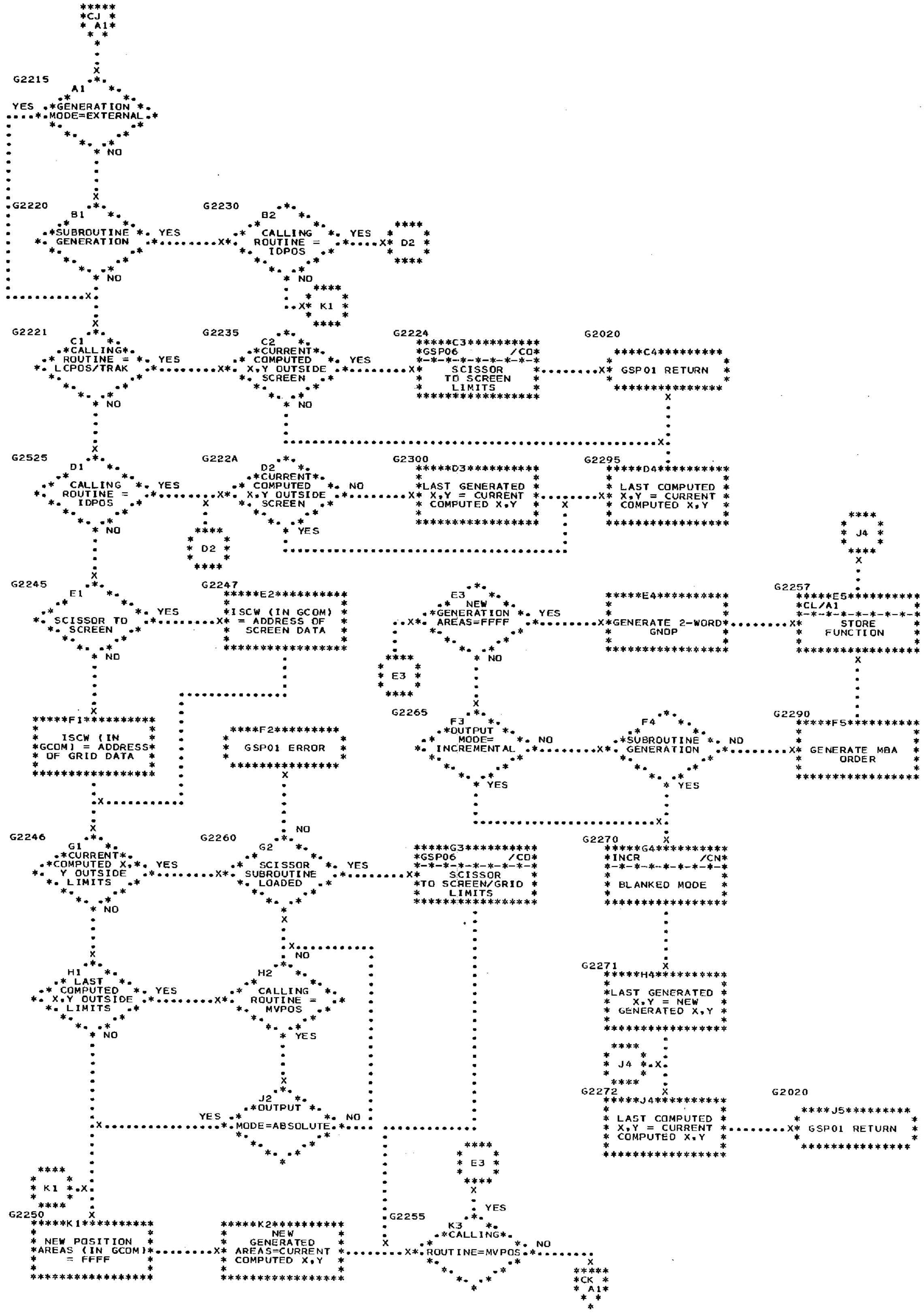


Chart CK. GSP02 - Generation Support (continued)

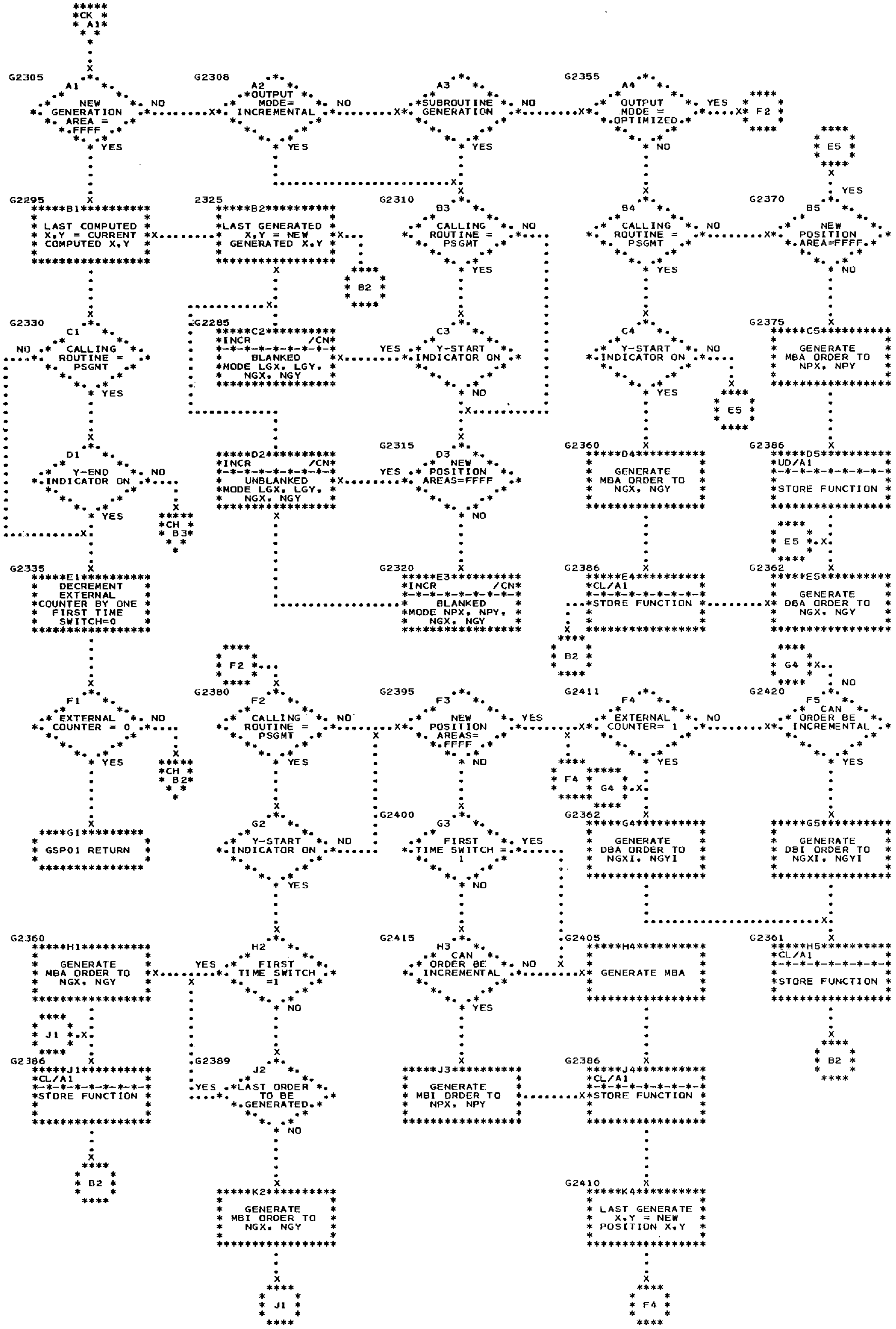

CK A1
G2305 A1 NEW GENERATION AREA = FFFF
NO
YES
G2308 A2 OUTPUT MODE= INCREMENTAL
NO
YES
A3 SUBROUTINE GENERATION
NO
YES
G2355 A4 OUTPUT MODE = OPTIMIZED
YES
F2
NO
E5
YES
G2295 B1 LAST COMPUTED X,Y = CURRENT COMPUTED X,Y
2325 B2 LAST GENERATED X,Y = NEW GENERATED X,Y
B2
G2310 B3 CALLING ROUTINE = PSGMT
NO
YES
B4 CALLING ROUTINE = PSGMT
NO
YES
G2370 B5 NEW POSITION AREA=FFFF
NO
G2330 C1 CALLING ROUTINE = PSGNT
NO
YES
G2285 C2 INCR /CN BLANKED MODE LGX, LGY, NGX, NGY
C3 Y-START INDICATOR ON
YES
NO
C4 Y-START INDICATOR ON
NO
YES
E5
G2375 C5 GENERATE MBA ORDER TO NPX, NPY
D1 Y-END INDICATOR ON
NO
YES
CH B3
D2 INCR /CN UNBLANKED MODE LGX, LGY, NGX, NGY
G2315 D3 NEW POSITION AREAS=FFFF
YES
NO
G2360 D4 GENERATE MBA ORDER TO NGX, NGY
G2386 D5 UD/A1 STORE FUNCTION
E5
G2335 E1 DECREMENT EXTERNAL COUNTER BY ONE FIRST TIME SWITCH=0
G2320 E3 INCR /CN BLANKED MODE NPX, NPY, NGX, NGY
G2386 E4 CL/A1 STORE FUNCTION
B2
G2362 E5 GENERATE DBA ORDER TO NGX, NGY
F2
G4
F1 EXTERNAL COUNTER = 0
NO
YES
CH B2
G2380 F2 CALLING ROUTINE = PSGMT
NO
YES
G2395 F3 NEW POSITION AREAS= FFFF
YES
NO
G2411 F4 EXTERNAL COUNTER= 1
NO
YES
F4
G4
G2420 F5 CAN ORDER BE INCREMENTAL
NO
YES
G1 GSP01 RETURN
G2 Y-START INDICATOR ON
NO
YES
G2400 G3 FIRST TIME SWITCH = 1
YES
NO
G2362 G4 DBA ORDER TO NGXI, NGYI
G5 GENERATE DBI ORDER TO NGXI, NGYI
G2360 H1 GENERATE MBA ORDER TO NGX, NGY
H2 FIRST TIME SWITCH =1
YES
NO
G2415 H3 CAN ORDER BE INCREMENTAL
NO
YES
G2405 H4 GENERATE MBA
G2361 H5 CL/A1 STORE FUNCTION
J1
B2
G2386 J1 CL/A1 STORE FUNCTION
G2389 J2 LAST ORDER TO BE GENERATED
YES
NO
J3 GENERATE MBI ORDER TO NPX, NPY
G2386 J4 CL/A1 STORE FUNCTION
B2
K2 GENERATE MBI ORDER TO NGX, NGY
G2410 K4 LAST GENERATE X,Y = NEW POSITION X,Y
J1
F4

Chart CL. GSP02 - Generation Support (continued)

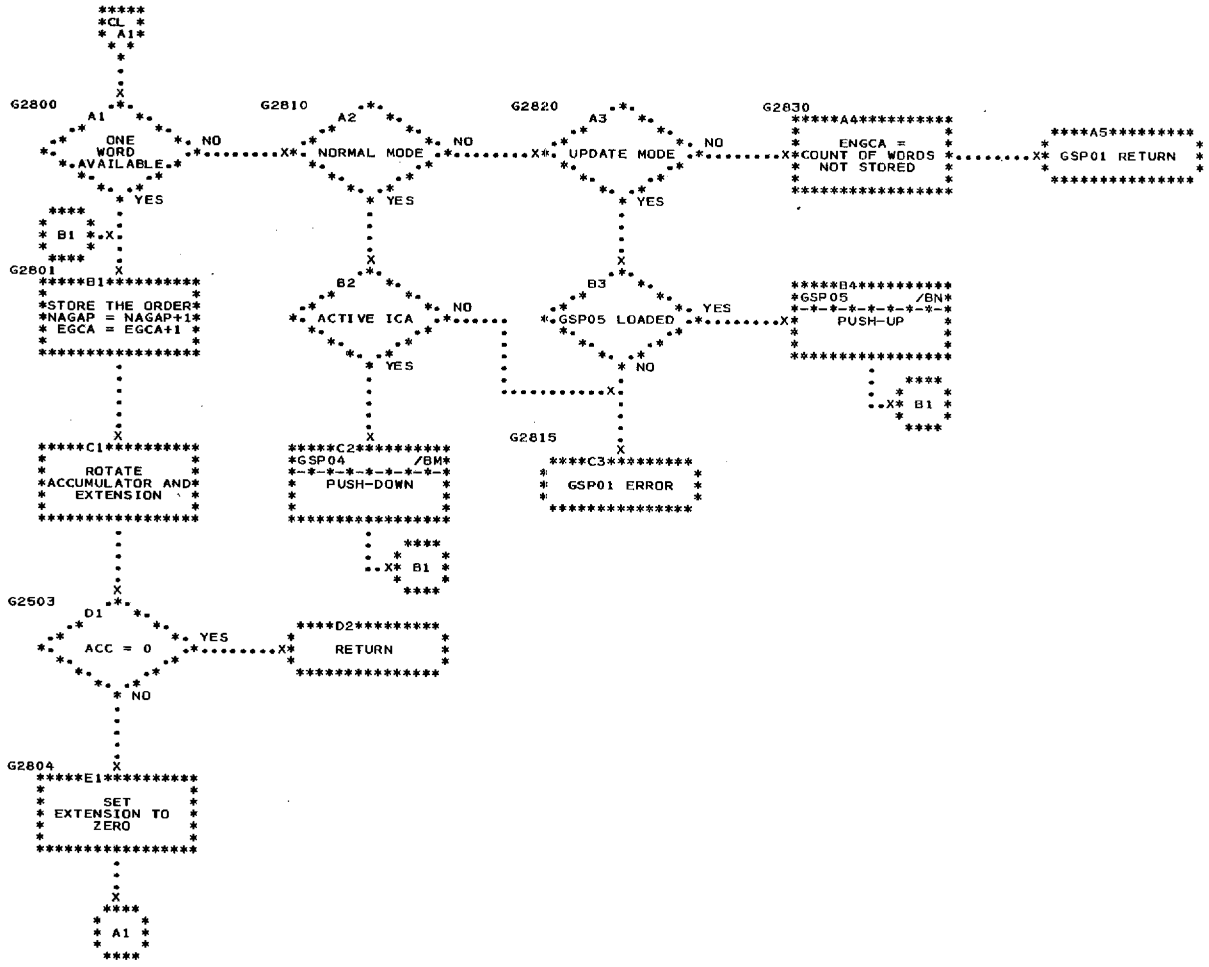


Chart CM. GSP02 - Generation Support (continued)

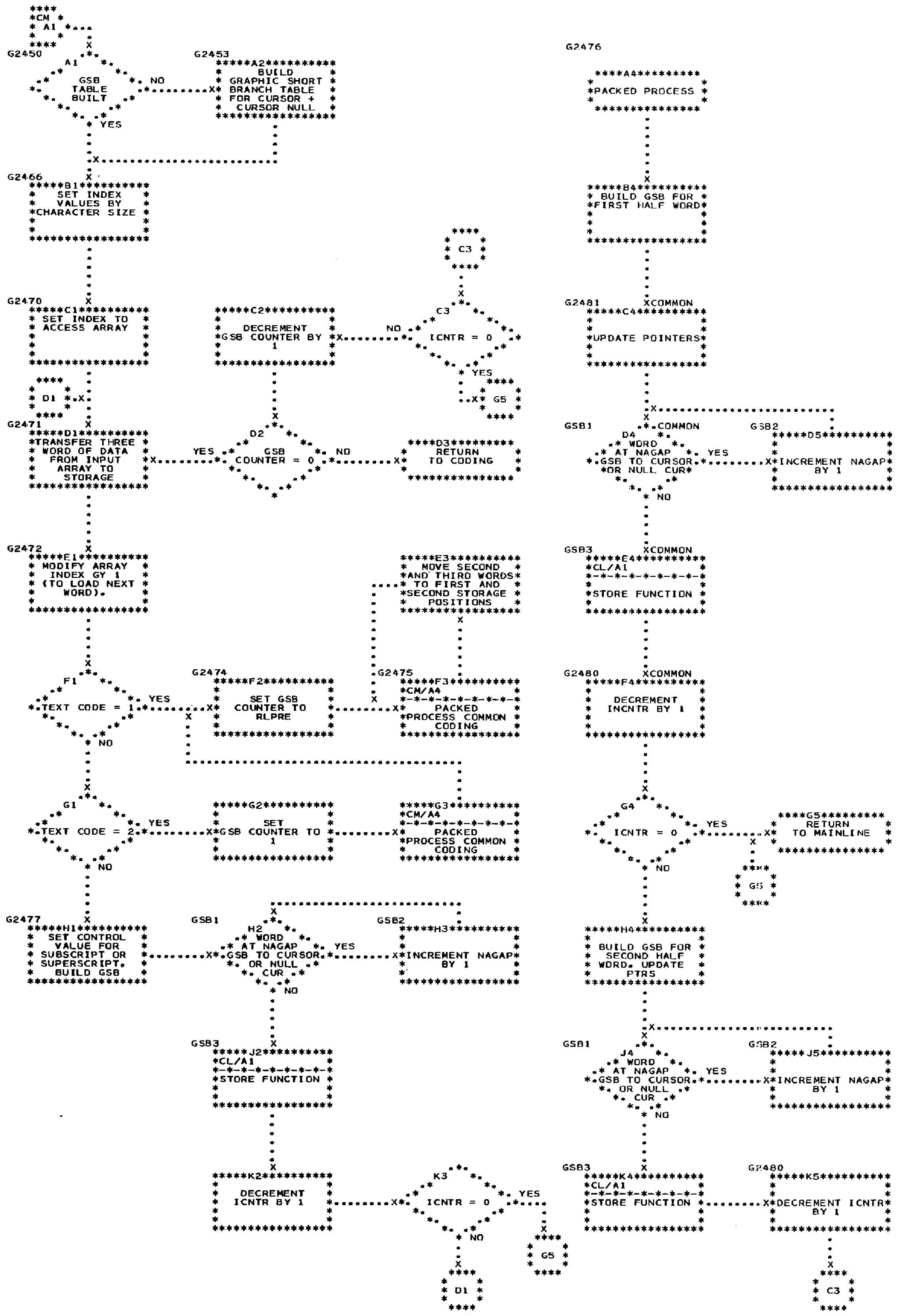


Chart CN. GSP02 - Generation Support (continued)

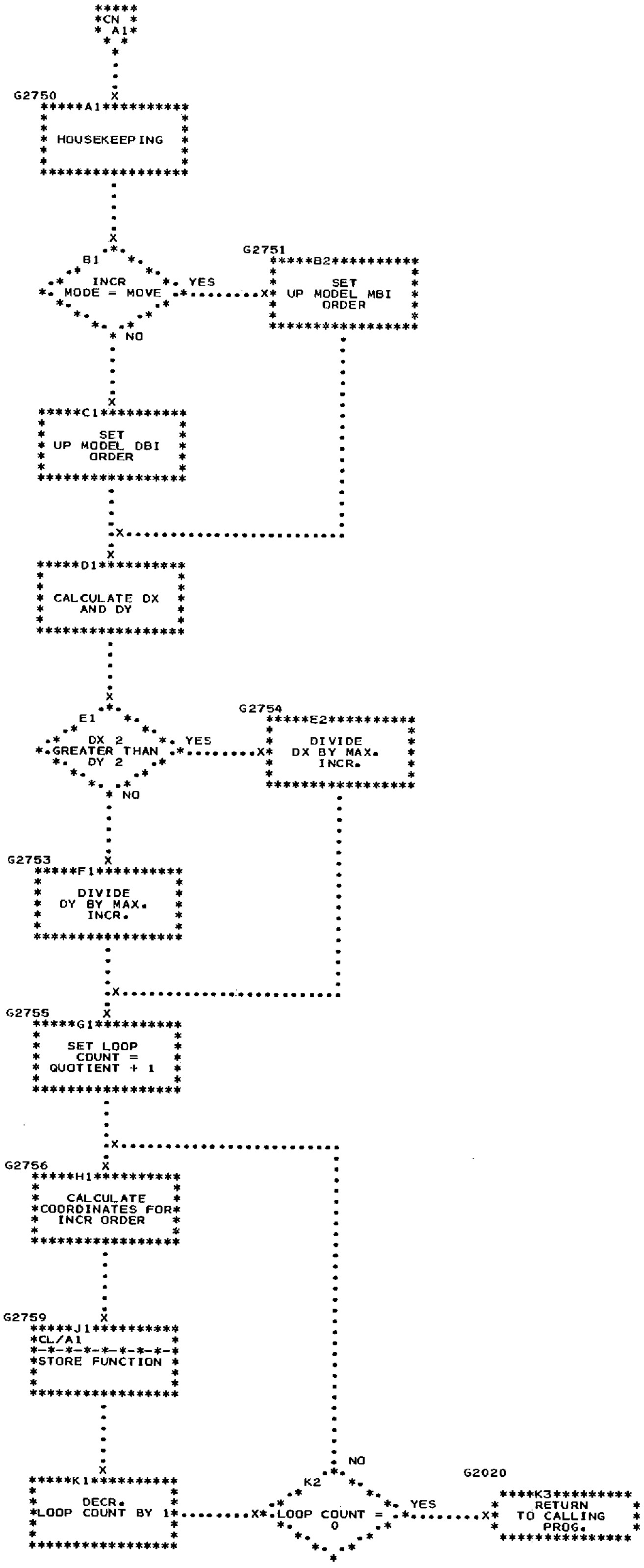


Chart CO. GSP06 - Scissor Subroutine

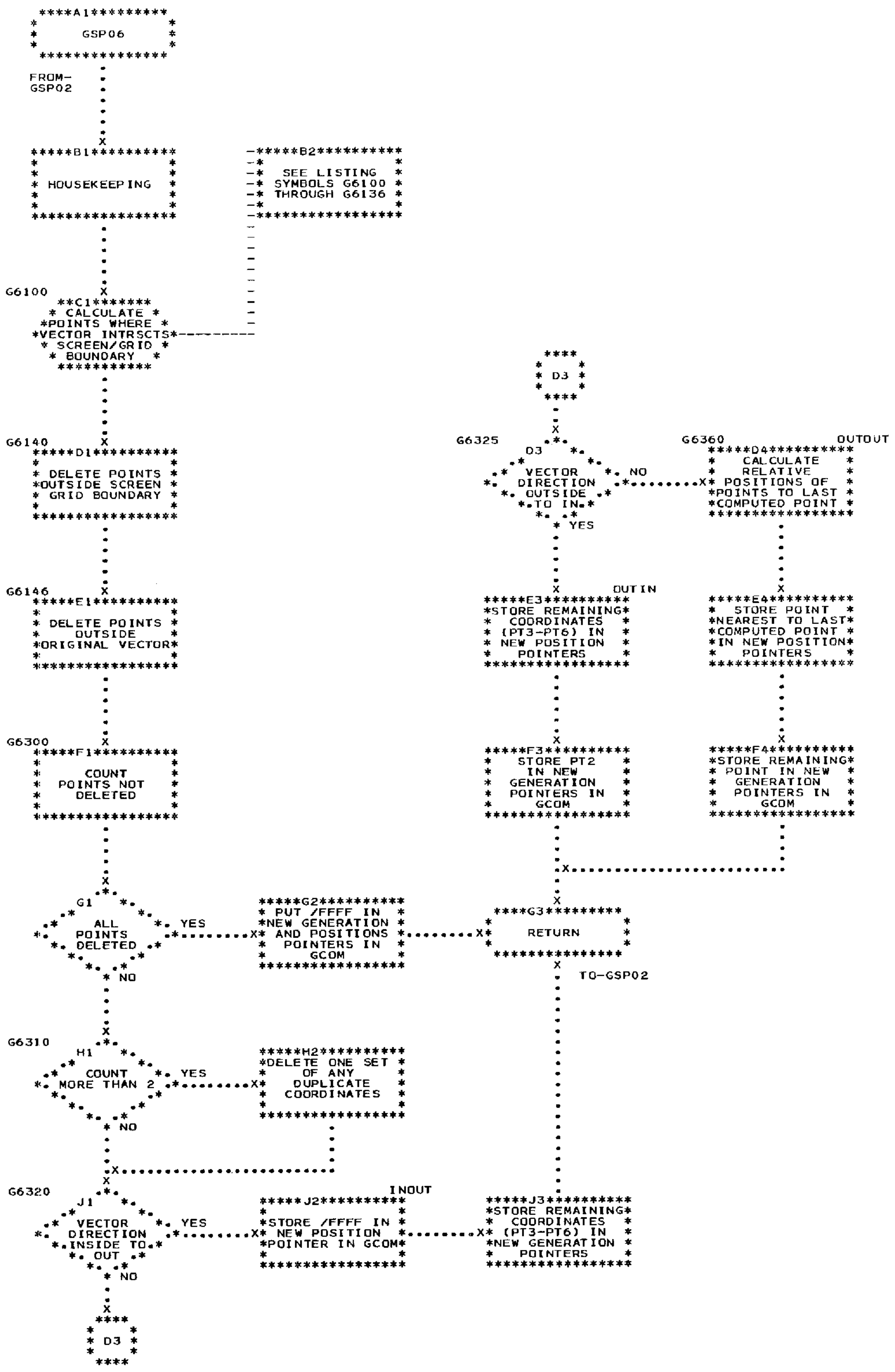


Chart DA. EXEC - Execute Display

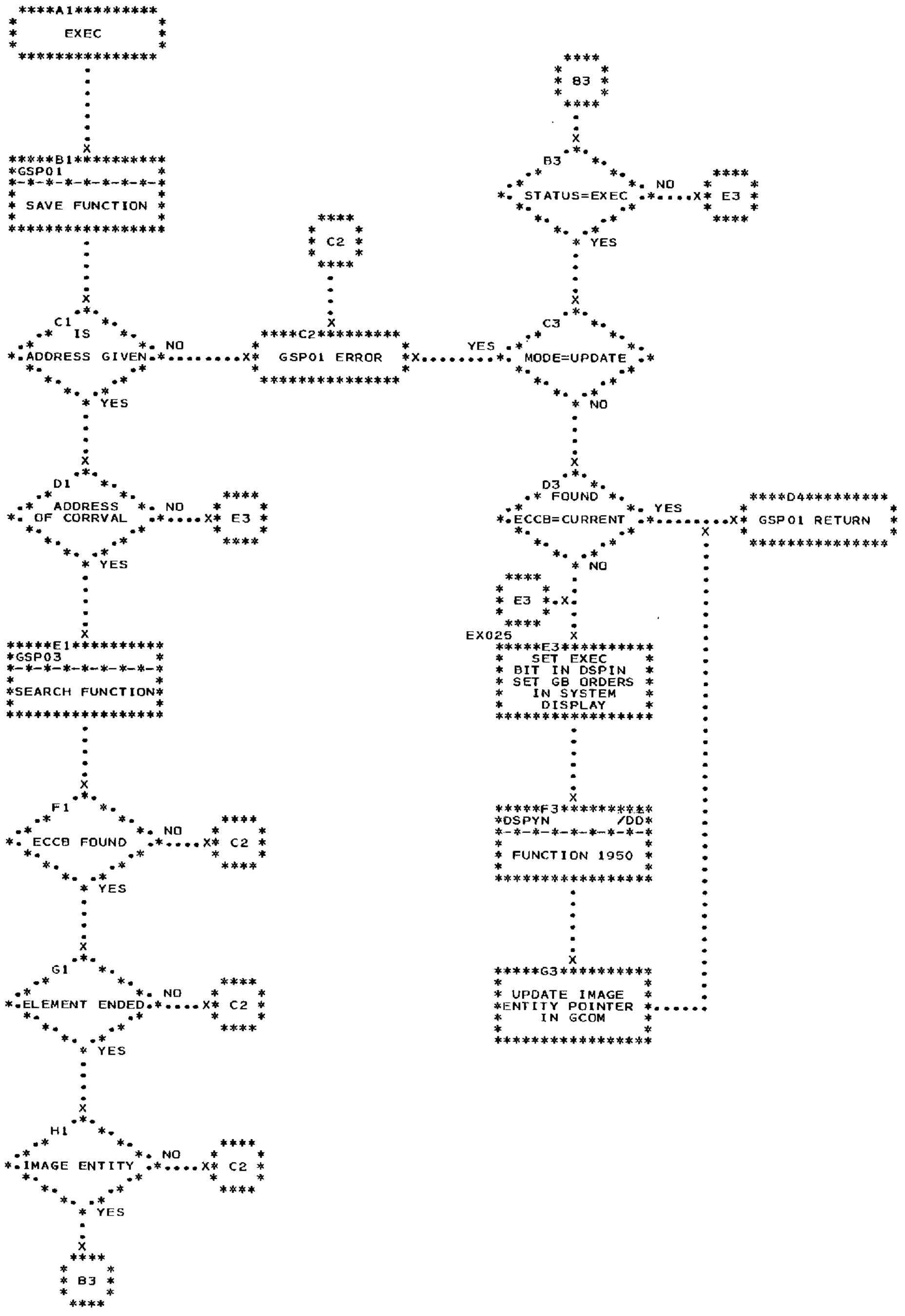


Chart DB. TMDSP - Terminate Display

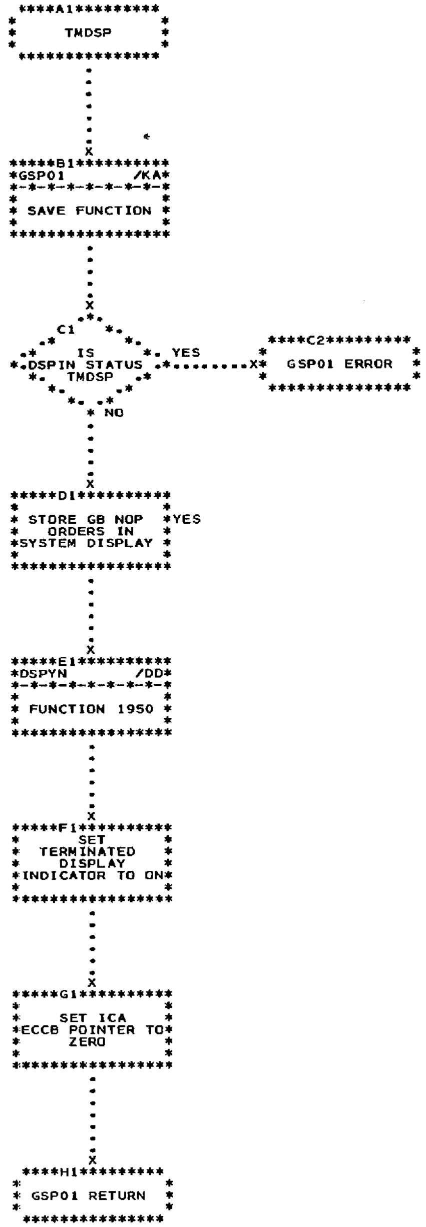


Chart DC. SPFKL - Set Programmed Function Keyboard Lights

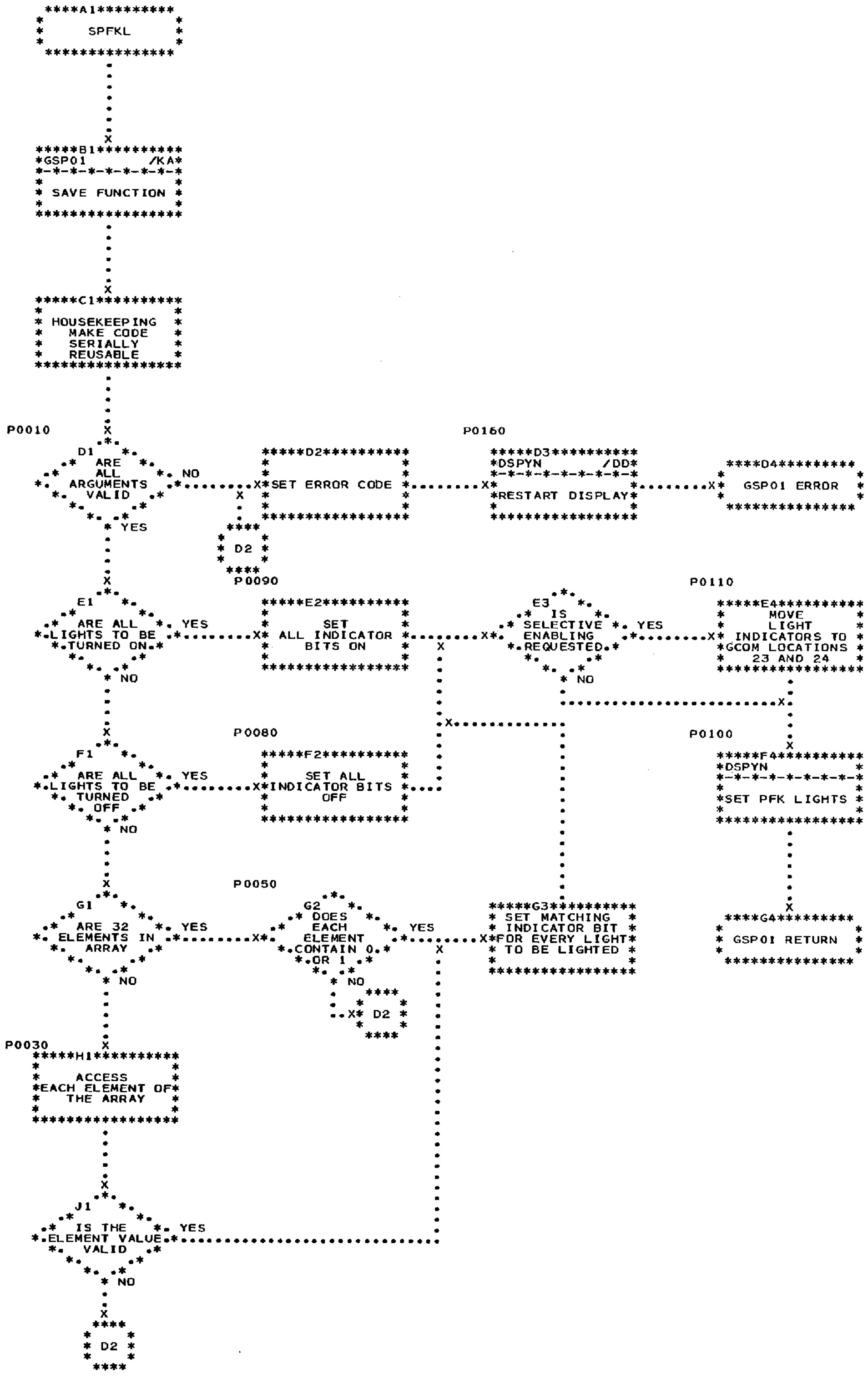

A1
SPFKL
B1
GSP01 /KA
SAVE FUNCTION
C1
HOUSEKEEPING MAKE CODE SERIALLY REUSABLE
P0010
D1
ARE ALL ARGUMENTS VALID
NO
YES
D2
SET ERROR CODE
D2
P0160
D3
DSPYN /DD
RESTART DISPLAY
D4
GSP01 ERROR
P0090
E1
ARE ALL LIGHTS TO BE TURNED ON
YES
NO
E2
SET ALL INDICATOR BITS ON
E3
IS SELECTIVE ENABLING REQUESTED
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

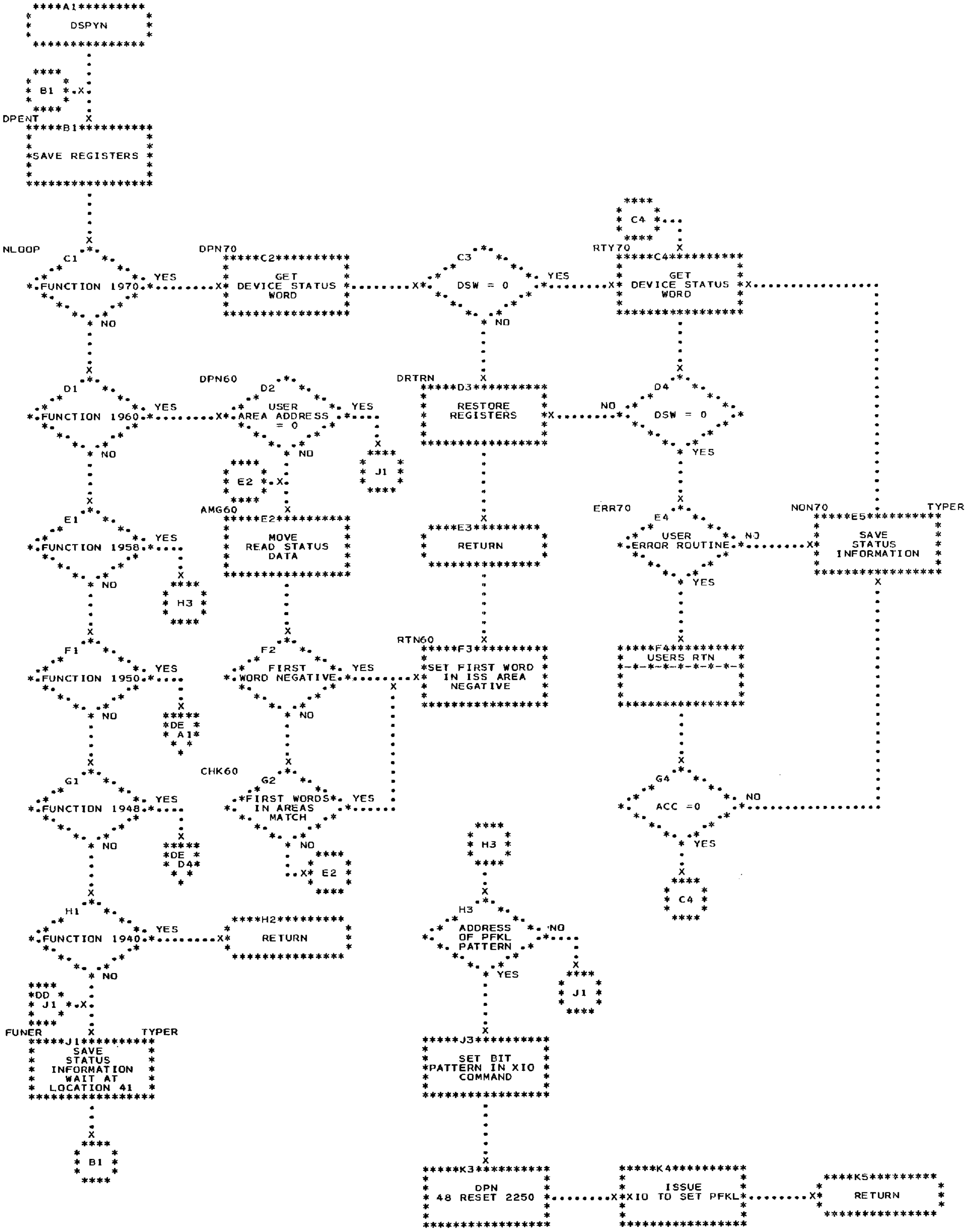


Chart DE. DSPYN - 2250 I/O Subroutine (continued)

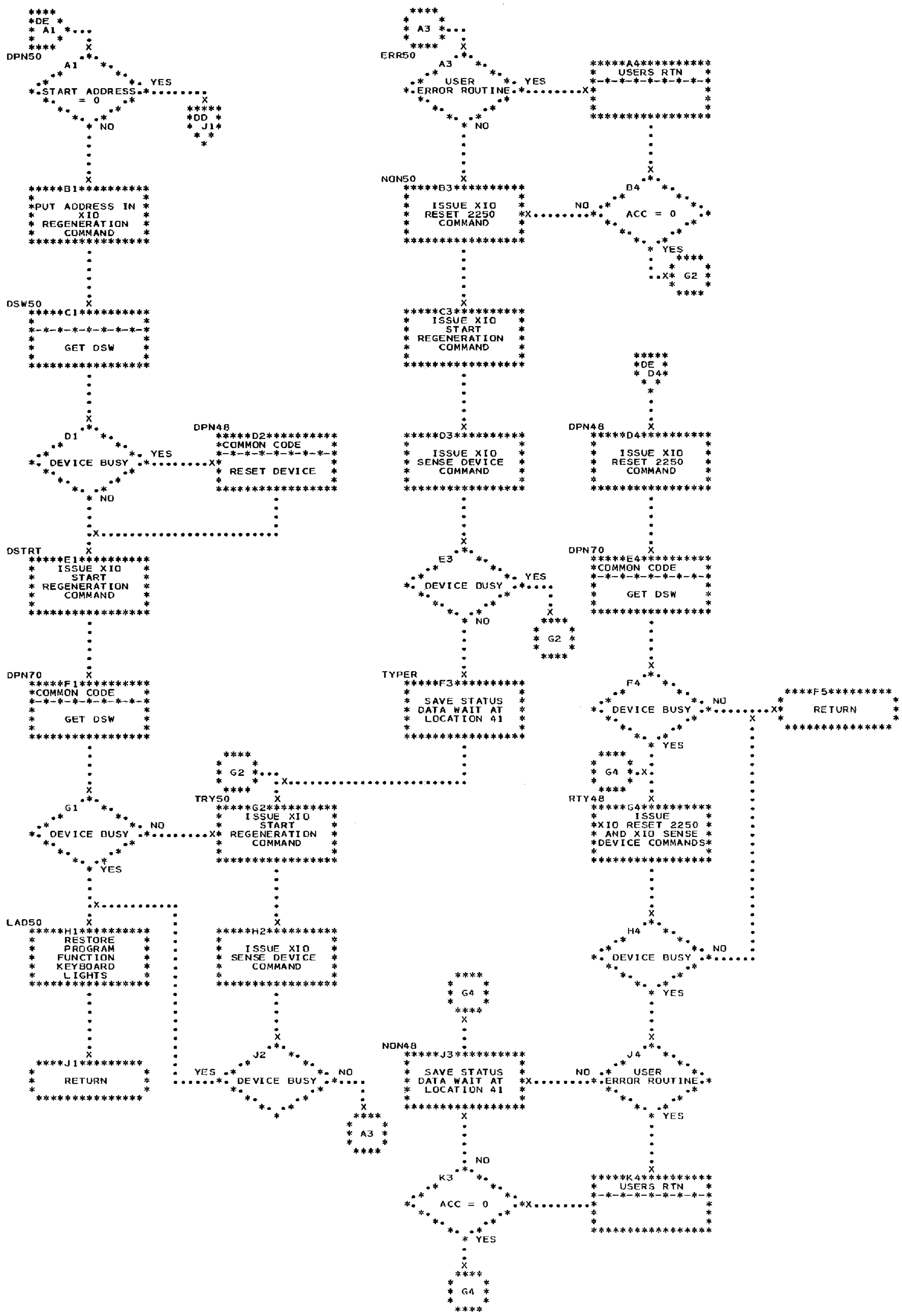


Chart DF. DSPYN - 2250 I/O Subroutine (continued)

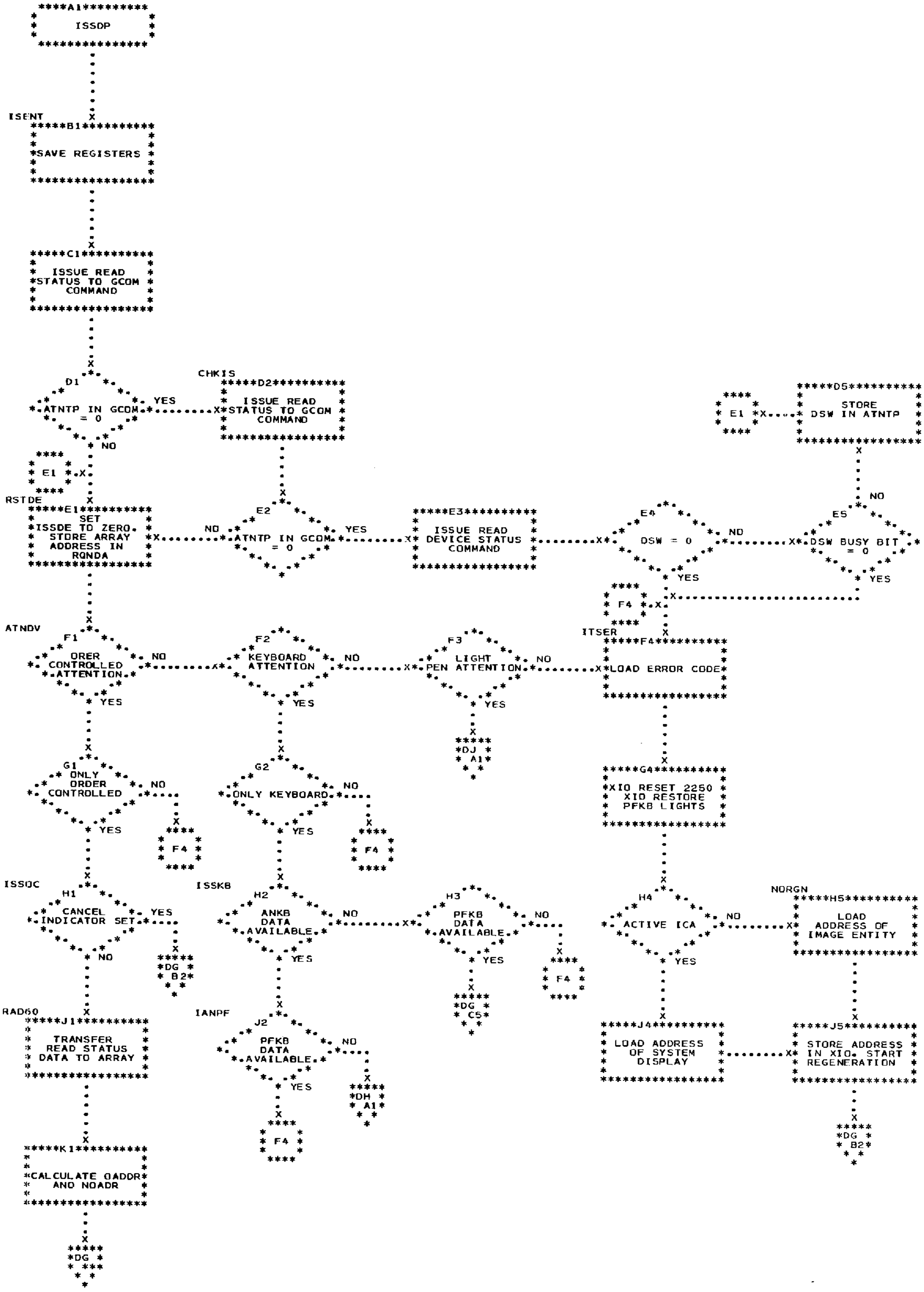


Chart DG. DSPYN - 2250 I/O Subroutine (continued)

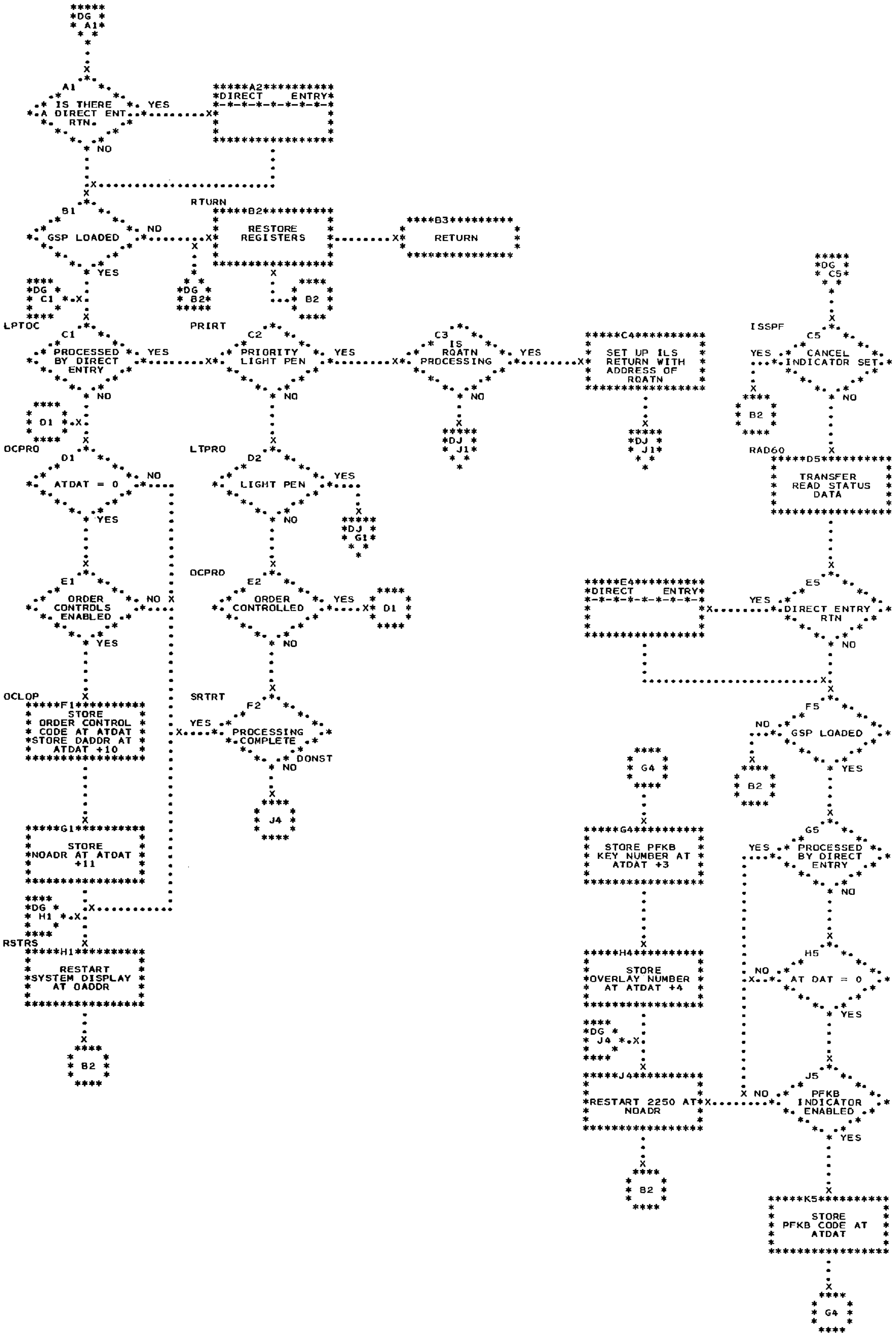


Chart DH. DSPYN - 2250 I/O Subroutine (continued)

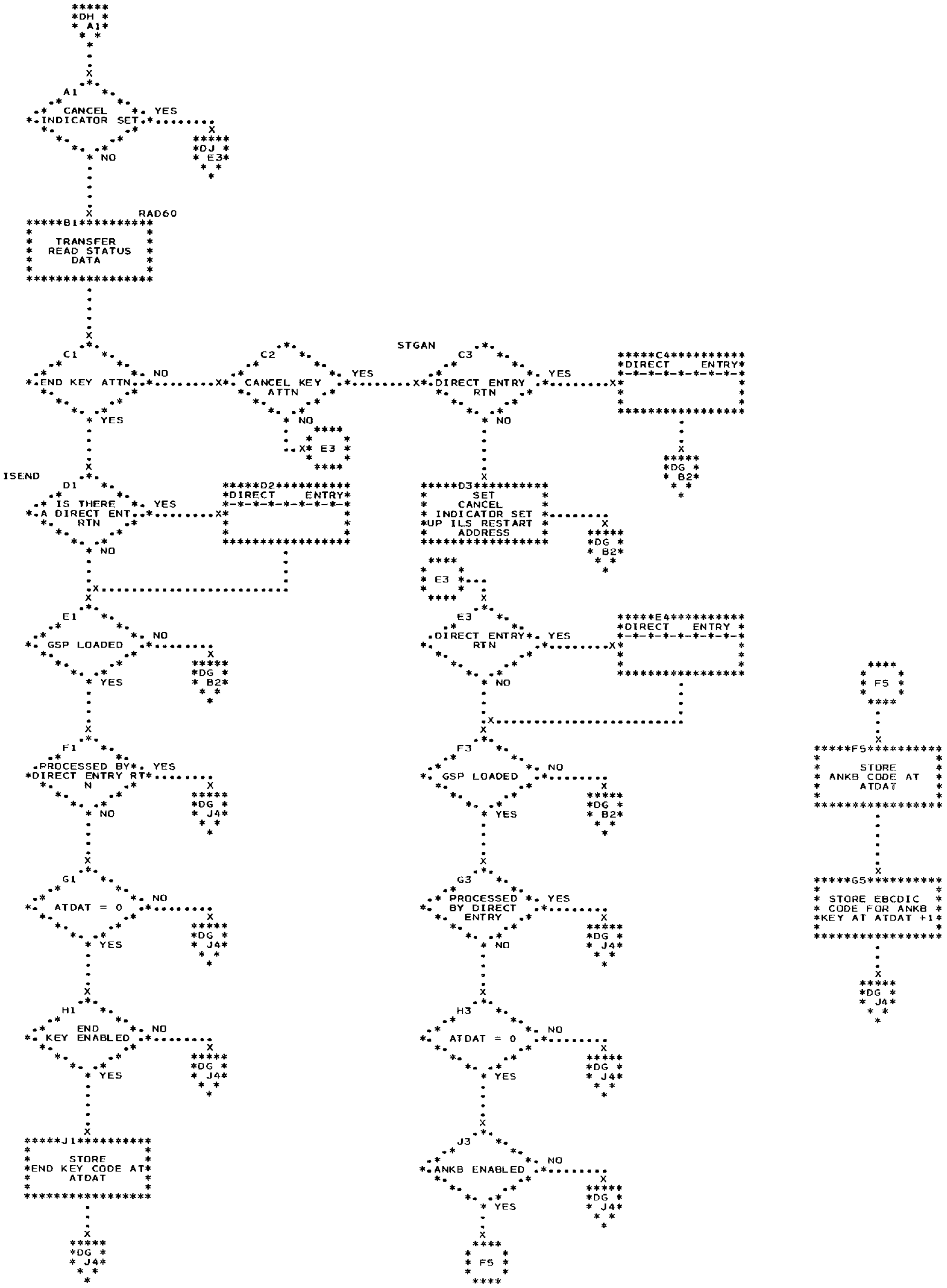


Chart DJ. DSPYN - 2250 I/O Subroutine (continued)

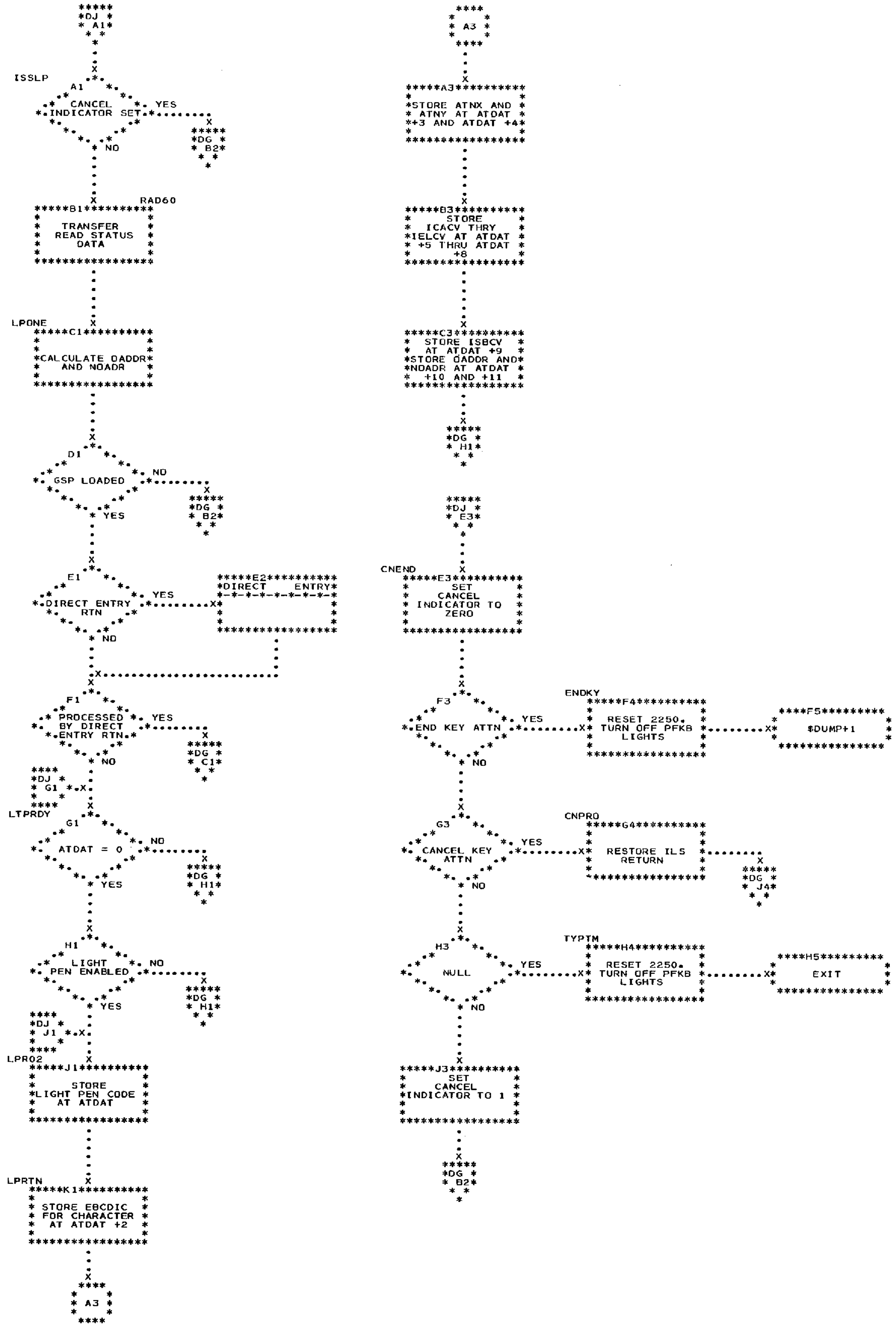

DJ A1
ISSLP
A1
CANCEL INDICATOR SET
YES
DG B2
NO
RAD60
B1
TRANSFER READ STATUS DATA
LPONE
C1
CALCULATE DADDR AND NOADR
D1
GSP LOADED
NO
DG B2
YES
E1
DIRECT ENTRY RTN
YES
E2
DIRECT ENTRY
NO
F1
PROCESSED BY DIRECT ENTRY RTN.
YES
DG C1
NO
DJ G1
LTPRDY
G1
ATDAT = 0
NO
DG H1
YES
H1
LIGHT PEN ENABLED
NO
DG H1
YES
DJ J1
LPRO2
J1
STORE LIGHT PEN CODE AT ATDAT
LPRTN
K1
STORE EBCDIC FOR CHARACTER AT ATDAT +2
A3
A3
A3
STORE ATNX AND ATNY AT ATDAT +3 AND ATDAT +4
B3
STORE ICACV THRY IELCV AT ATDAT +5 THRU ATDAT +8
C3
STORE ISBCV AT ATDAT +9 STORE DADDR AND NOADR AT ATDAT +10 AND +11
DG H1
DJ E3
CNEND
E3
SET CANCEL INDICATOR TO ZERO
F3
END KEY ATTN
YES
ENDKY
F4
RESET 2250. TURN OFF PFKB LIGHTS
F5
$DUMP+1
NO
G3
CANCEL KEY ATTN
YES
CNPRO
G4
RESTORE ILS RETURN
DG J4
NO
H3
NULL
YES
TYPTM
H4
RESET 2250. TURN OFF PFKB LIGHTS
H5
EXIT
NO
J3
SET CANCEL INDICATOR TO 1
DG B2

Chart EA. SATNS - Set Attention Status

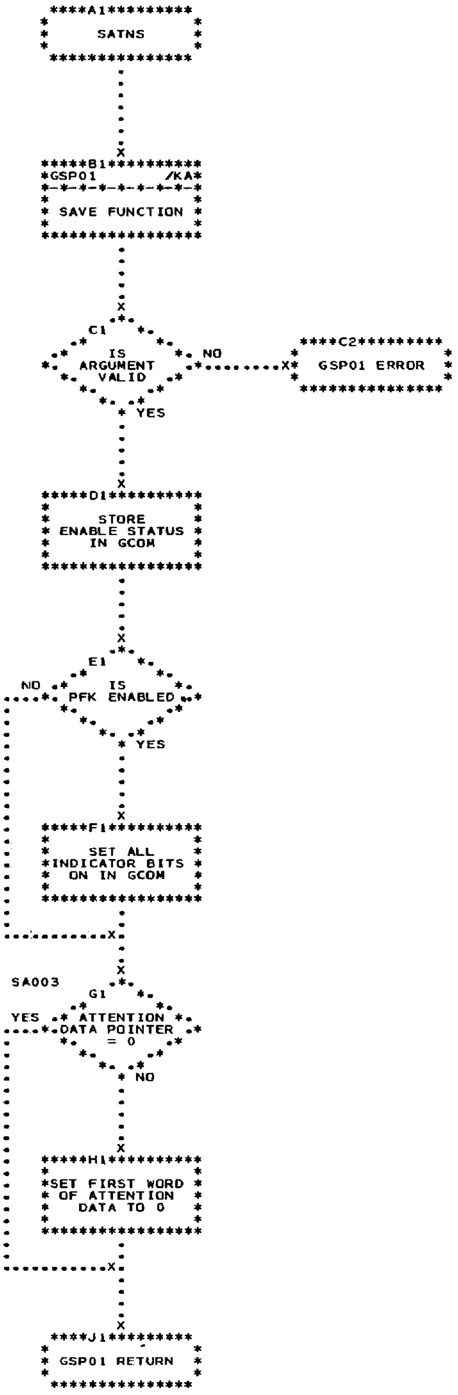

A1
SATNS
B1
GSP01
/KA
SAVE FUNCTION
C1
IS ARGUMENT VALID
NO
C2
GSP01 ERROR
YES
D1
STORE ENABLE STATUS IN GCOM
E1
NO
IS PFK ENABLED
YES
F1
SET ALL INDICATOR BITS ON IN GCOM
SA003
G1
YES
ATTENTION DATA POINTER = 0
NO
H1
SET FIRST WORD OF ATTENTION DATA TO 0
J1
GSP01 RETURN

Chart EB. RQATN - Request Attention Information

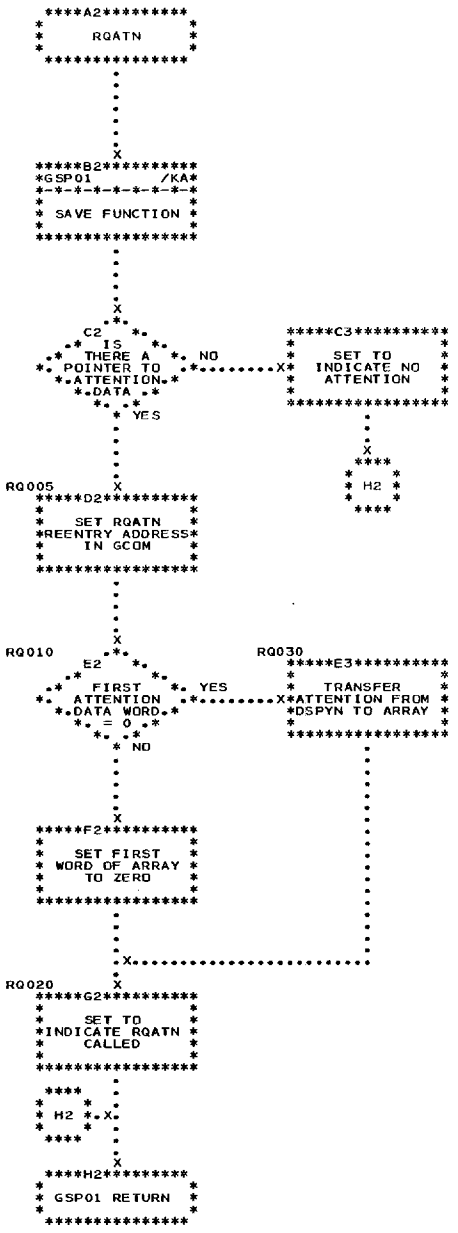

A2
RQATN
B2
GSP01 /KA
SAVE FUNCTION
C2
IS THERE A POINTER TO ATTENTION DATA
NO
YES
C3
SET TO INDICATE NO ATTENTION
H2
RQ005
D2
SET RQATN REENTRY ADDRESS IN GCOM
RQ010
E2
FIRST ATTENTION DATA WORD = 0
YES
NO
RQ030
E3
TRANSFER ATTENTION FROM DSPYN TO ARRAY
F2
SET FIRST WORD OF ARRAY TO ZERO
RQ020
G2
SET TO INDICATE RQATN CALLED
H2
H2
GSP01 RETURN

Chart EC. ROCOR - Return Outer Correlation Value

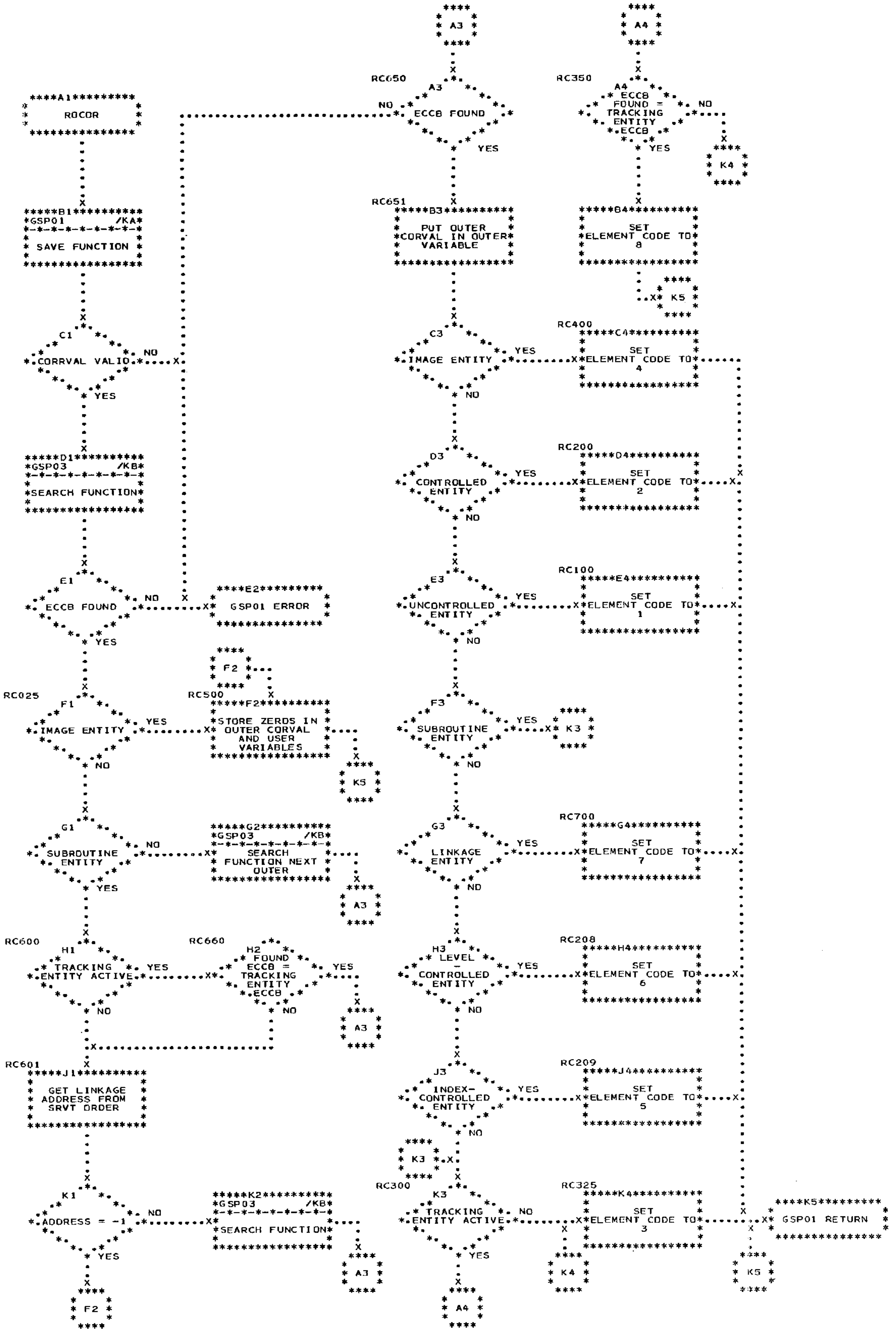


Chart FA. LOCPN - Locate Position of the Light Pen
LOCND - Locate Position of the Light Pen with No Detect

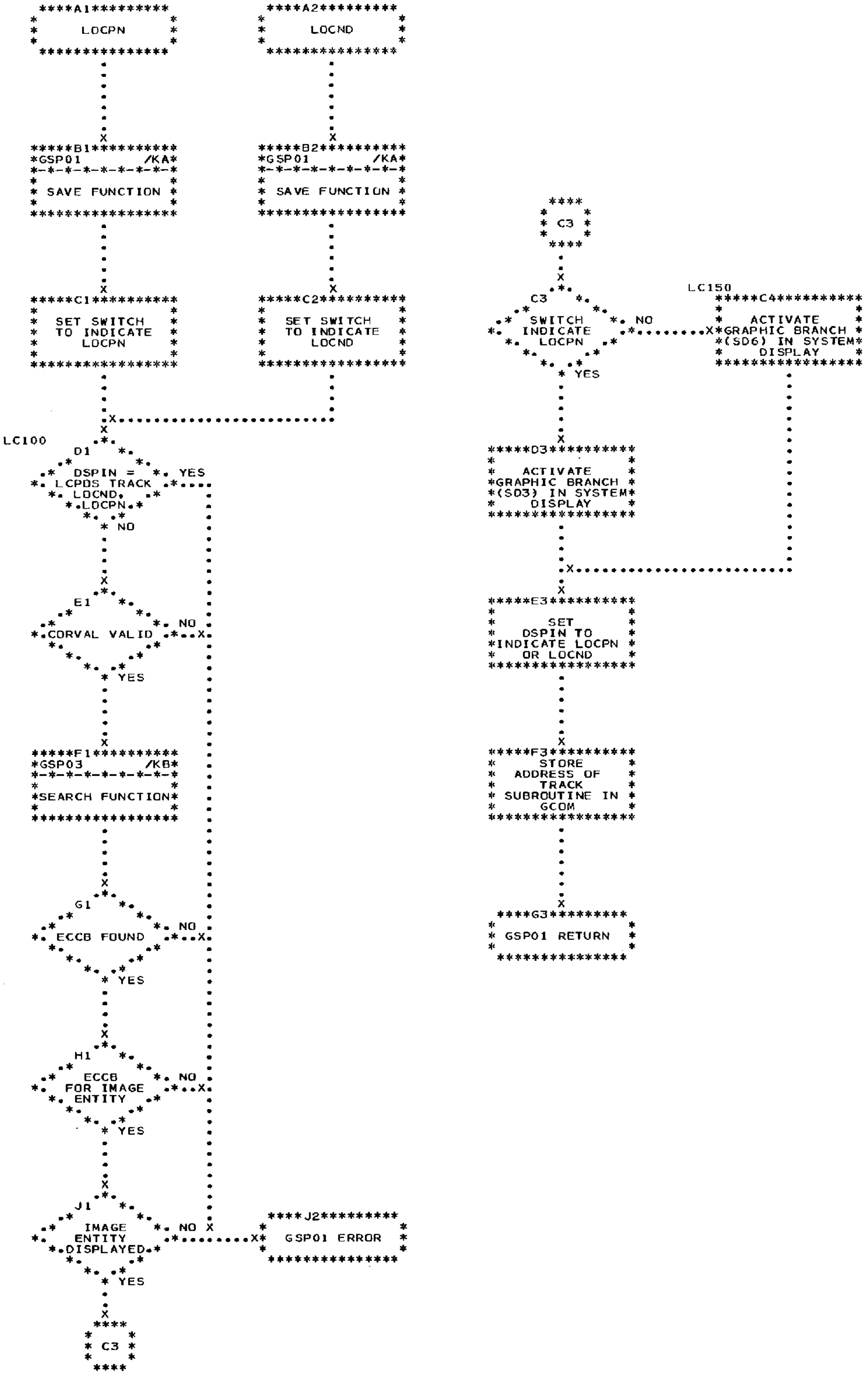


Chart FB. LCPOS - Locate a Position With the Tracking Symbol

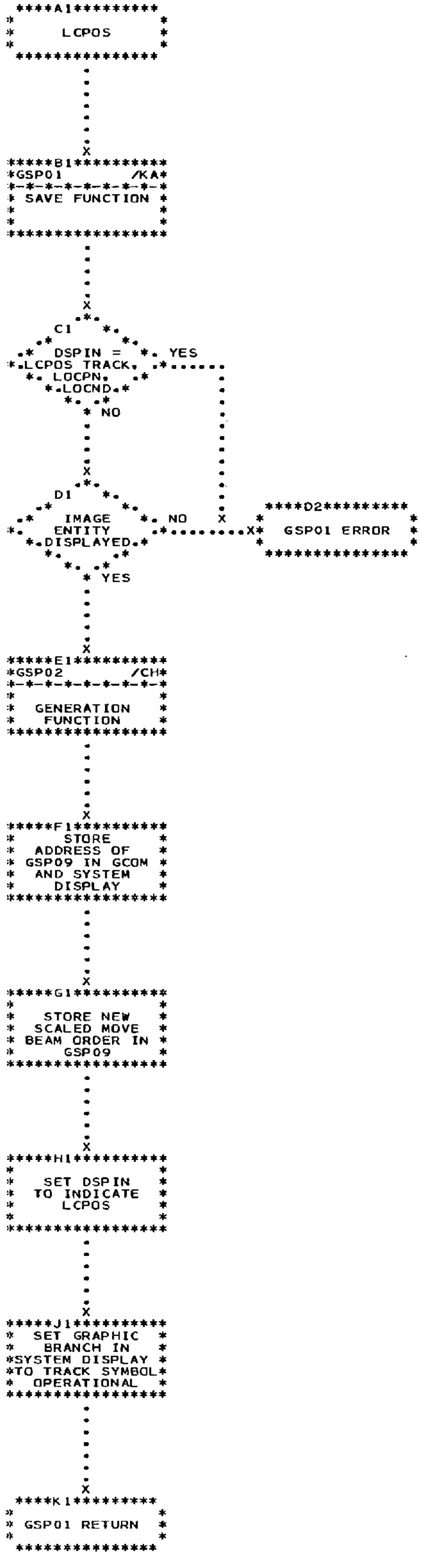

A1
LCPOS
B1
GSP01 /KA
SAVE FUNCTION
C1
DSPIN = LCPOS TRACK, LOCPN, LOCND
YES
NO
D1
IMAGE ENTITY DISPLAYED
NO
YES
D2
GSP01 ERROR
E1
GSP02 /CH
GENERATION FUNCTION
F1
STORE ADDRESS OF GSP09 IN GCOM AND SYSTEM DISPLAY
G1
STORE NEW SCALED MOVE BEAM ORDER IN GSP09
H1
SET DSPIN TO INDICATE LCPOS
J1
SET GRAPHIC BRANCH IN SYSTEM DISPLAY TO TRACK SYMBOL OPERATIONAL
K1
GSP01 RETURN

Chart FC. TRACK - Track Position of Light Pen

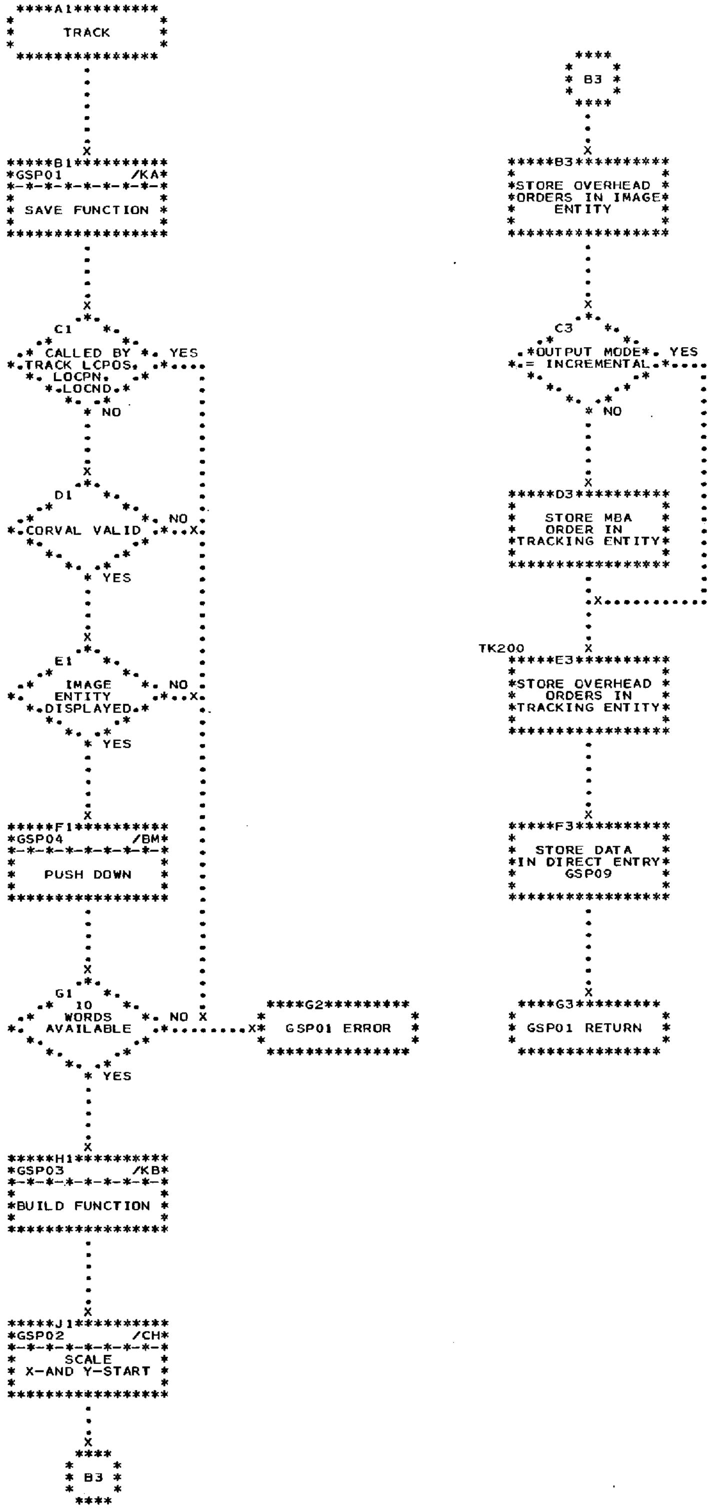


Chart FD. CTLTK - Control Light Pen Tracking

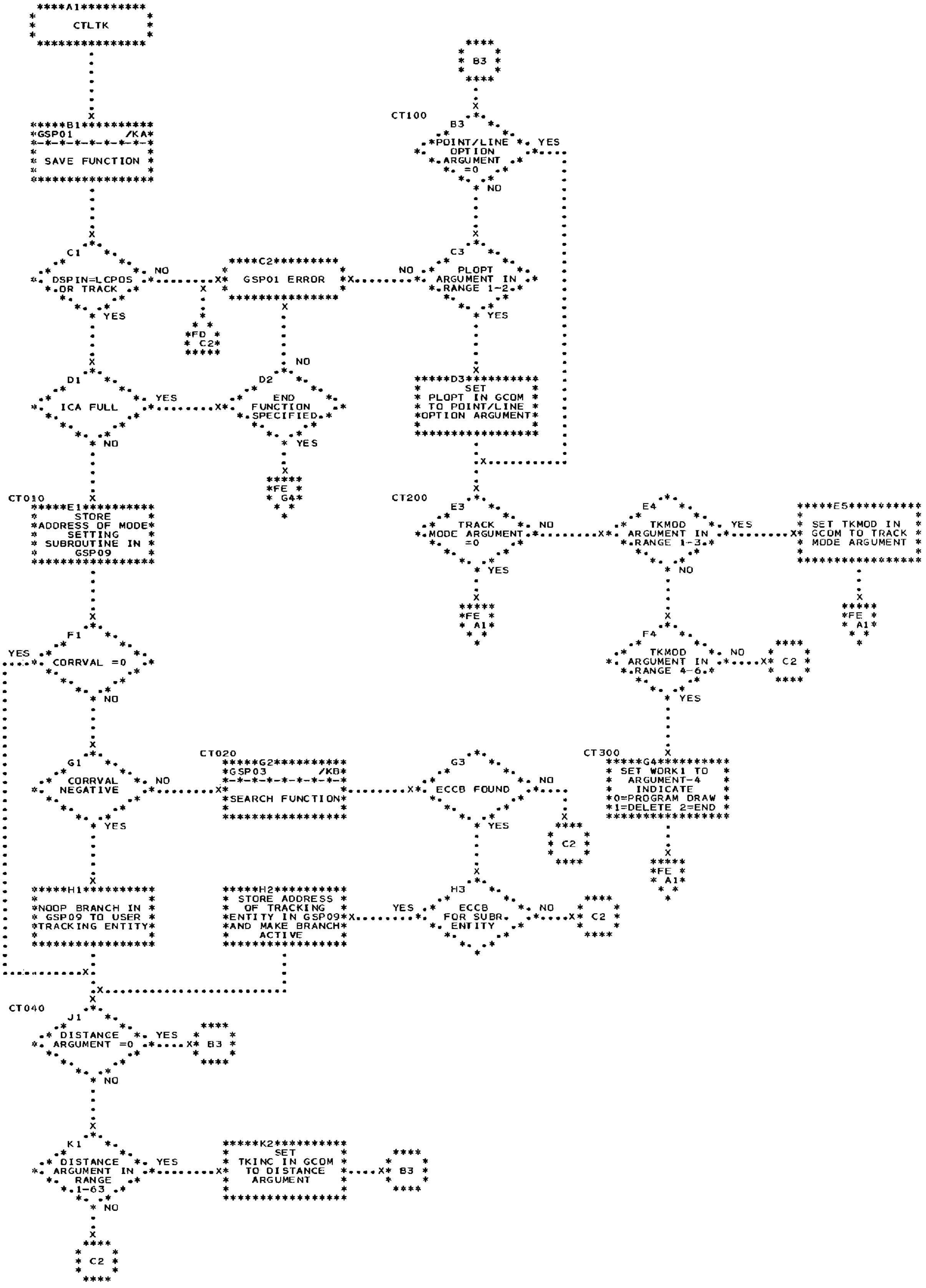

A1
CTLTK
B1
GSP01 /KA
SAVE FUNCTION
C1
DSPIN=LCPOS OR TRACK
NO
YES
C2
GSP01 ERROR
FD C2
D1
ICA FULL
YES
NO
D2
END FUNCTION SPECIFIED
NO
YES
FE G4
CT010
E1
STORE ADDRESS OF MODE SETTING SUBROUTINE IN GSP09
F1
CORRVAL =0
YES
NO
G1
CORRVAL NEGATIVE
NO
YES
CT020
G2
GSP03 /KB
SEARCH FUNCTION
G3
ECCB FOUND
NO
YES
C2
H1
NOOP BRANCH IN GSP09 TO USER TRACKING ENTITY
H2
STORE ADDRESS OF TRACKING ENTITY IN GSP09 AND MAKE BRANCH ACTIVE
H3
ECCB FOR SUBR. ENTITY
YES
NO
C2
CT040
J1
DISTANCE ARGUMENT =0
YES
B3
NO
K1
DISTANCE ARGUMENT IN RANGE 1-63
YES
K2
SET TKINC IN GCOM TO DISTANCE ARGUMENT
B3
NO
C2
B3
CT100
B3
POINT/LINE OPTION ARGUMENT =0
YES
NO
C3
PLOPT ARGUMENT IN RANGE 1-2
NO
YES
D3
SET PLOPT IN GCOM TO POINT/LINE OPTION ARGUMENT
CT200
E3
TRACK MODE ARGUMENT =0
NO
YES
FE A1
E4
TKMOD ARGUMENT IN RANGE 1-3
YES
NO
E5
SET TKMOD IN GCOM TO TRACK MODE ARGUMENT
FE A1
F4
TKMOD ARGUMENT IN RANGE 4-6
NO
C2
YES
CT300
G4
SET WORK1 TO ARGUMENT-4 INDICATE 0=PROGRAM DRAW 1=DELETE 2=END
FE A1

Chart FE. CTLTK - Control Light Pen Tracking (continued)

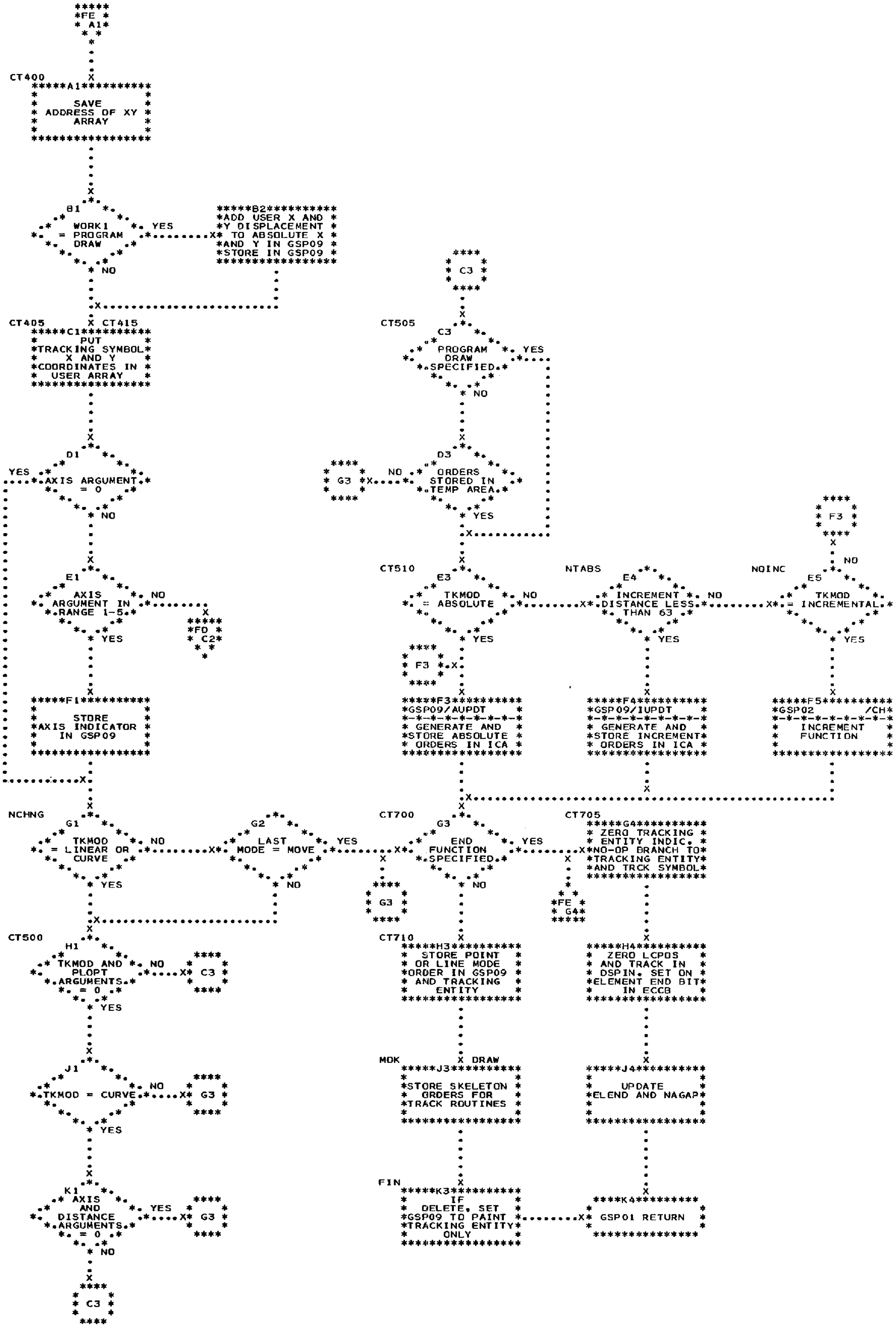

FE A1
CT400 A1
SAVE ADDRESS OF XY ARRAY
B1
WORK1 = PROGRAM DRAW
YES
NO
B2
ADD USER X AND Y DISPLACEMENT TO ABSOLUTE X AND Y IN GSP09 STORE IN GSP09
CT405 CT415 C1
PUT TRACKING SYMBOL X AND Y COORDINATES IN USER ARRAY
D1
AXIS ARGUMENT = 0
YES
NO
E1
AXIS ARGUMENT IN RANGE 1-5
NO
YES
FD C2
F1
STORE AXIS INDICATOR IN GSP09
NCHNG G1
TKMOD = LINEAR OR CURVE
NO
YES
G2
LAST MODE = MOVE
YES
NO
G3
CT500 H1
TKMOD AND PLOPT ARGUMENTS = 0
NO
YES
C3
J1
TKMOD = CURVE
NO
YES
G3
K1
AXIS AND DISTANCE ARGUMENTS = 0
YES
NO
G3
C3
C3
CT505 C3
PROGRAM DRAW SPECIFIED
YES
NO
D3
ORDERS STORED IN TEMP AREA
NO
YES
G3
CT510 E3
TKMOD = ABSOLUTE
NO
YES
F3
NTABS E4
INCREMENT DISTANCE LESS THAN 63
NO
YES
NOINC E5
TKMOD = INCREMENTAL
NO
YES
F3
F3
GSP09/AUPDT
GENERATE AND STORE ABSOLUTE ORDERS IN ICA
F4
GSP09/IUPDT
GENERATE AND STORE INCREMENT ORDERS IN ICA
F5
GSP02 /CH
INCREMENT FUNCTION
CT700 G3
END FUNCTION SPECIFIED
YES
NO
G3
CT705 G4
ZERO TRACKING ENTITY INDIC. NO-OP BRANCH TO TRACKING ENTITY AND TRCK SYMBOL
FE G4
CT710 H3
STORE POINT OR LINE MODE ORDER IN GSP09 AND TRACKING ENTITY
H4
ZERO LCPOS AND TRACK IN DSPIN. SET ON ELEMENT END BIT IN ECCB
MOK DRAW J3
STORE SKELETON ORDERS FOR TRACK ROUTINES
J4
UPDATE ELEND AND NAGAP
FIN K3
IF DELETE, SET GSP09 TO PAINT TRACKING ENTITY ONLY
K4
GSP01 RETURN

Chart FF. CVTTD - Convert Tracking Data

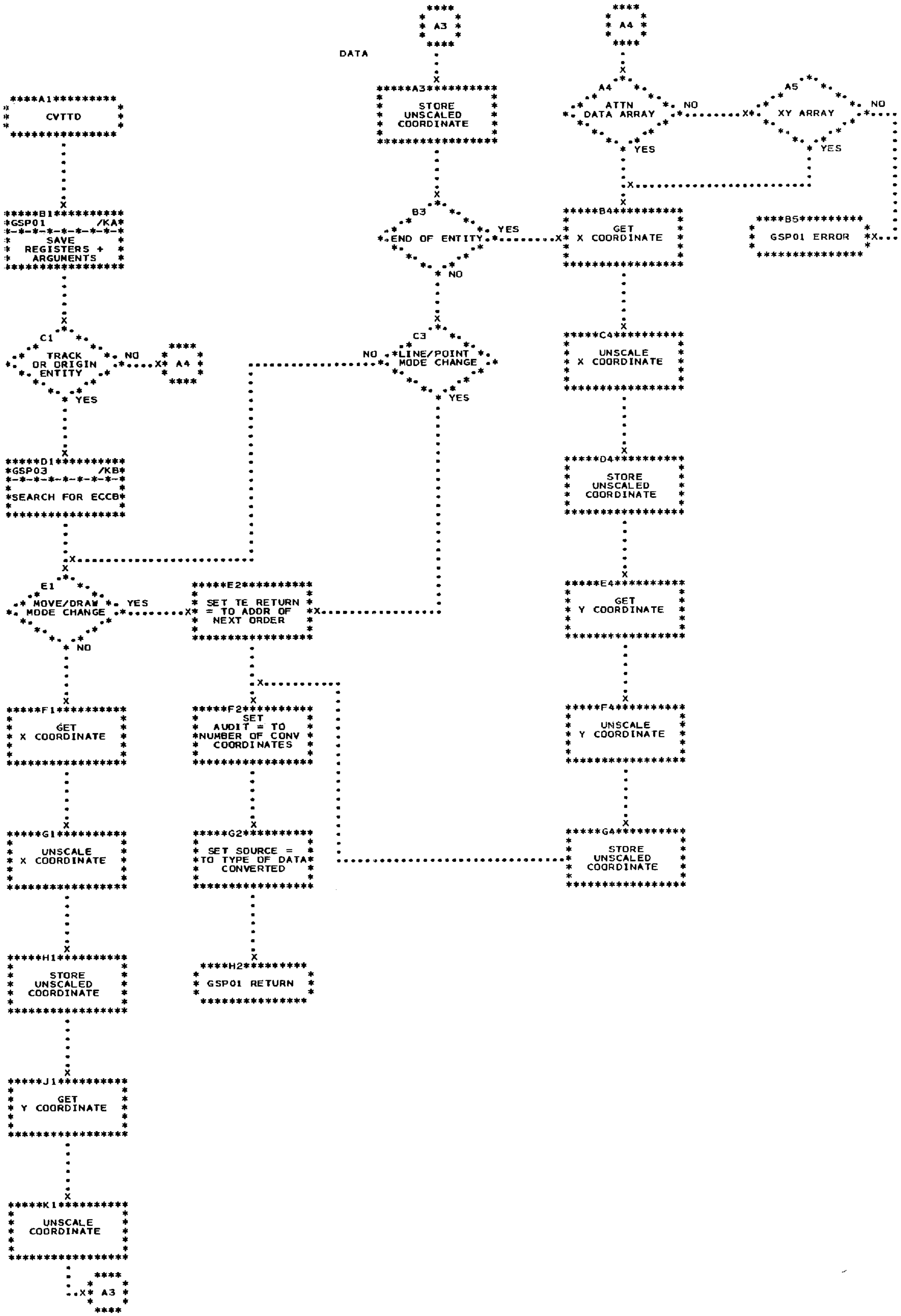


Chart GA. DFMSG - Define Message Entity

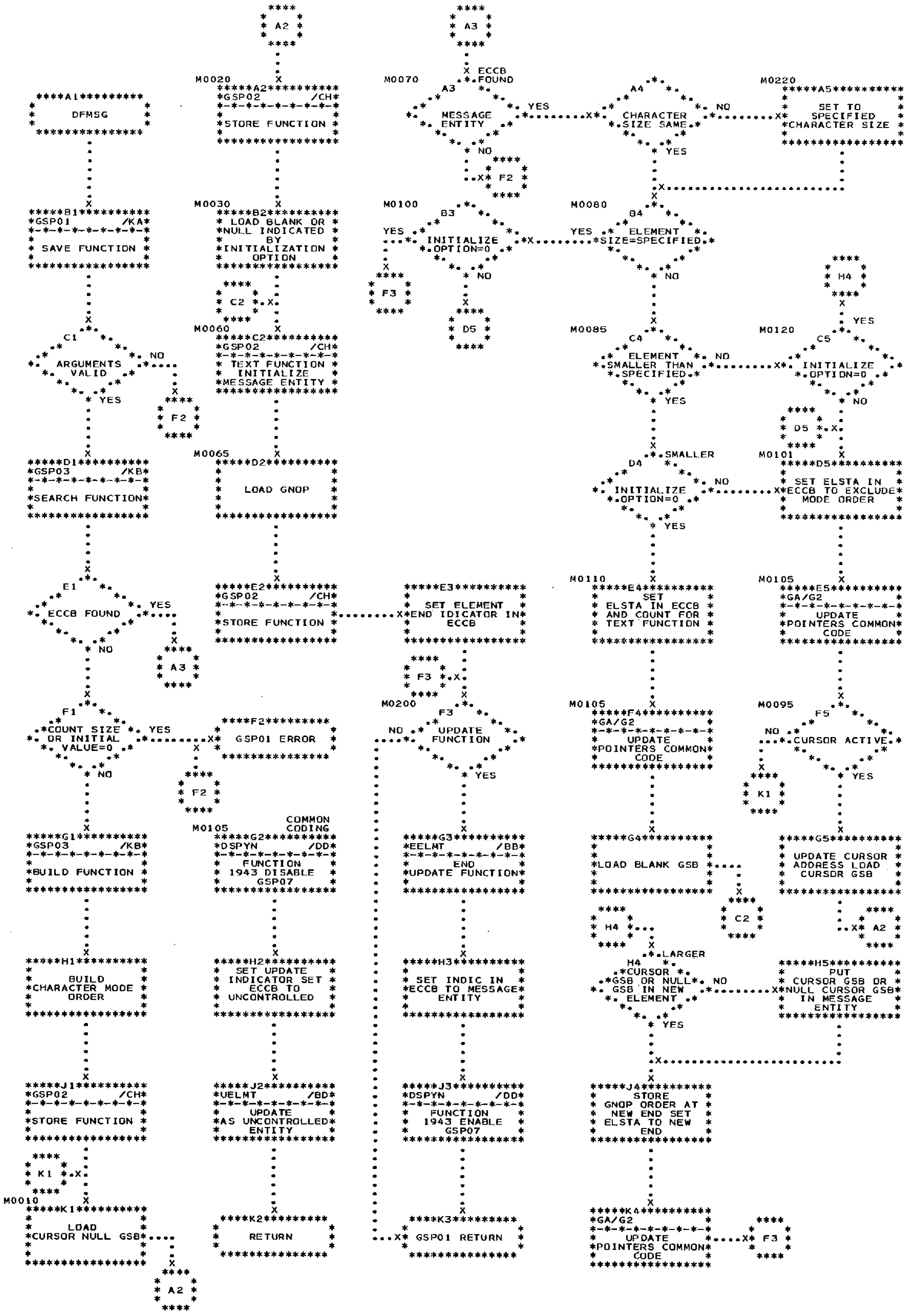
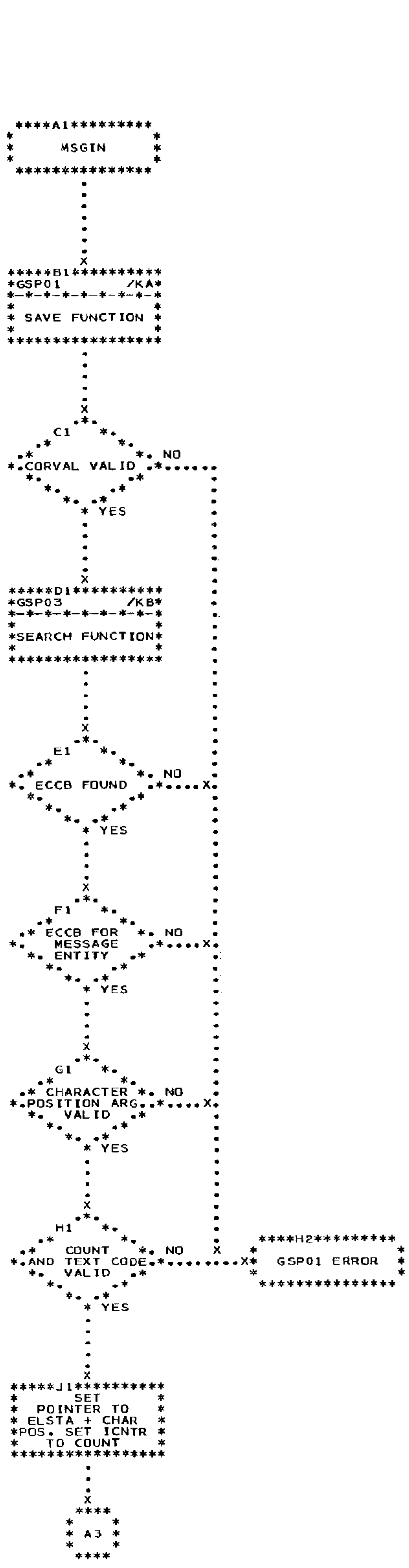

A1 DFMSG
B1 GSP01 /KA SAVE FUNCTION
C1 ARGUMENTS VALID
NO
F2
YES
D1 GSP03 /KB SEARCH FUNCTION
E1 ECCB FOUND
YES
A3
NO
F1 COUNT SIZE OR INITIAL VALUE=0
YES
F2 GSP01 ERROR
F2
NO
G1 GSP03 /KB BUILD FUNCTION
H1 BUILD CHARACTER MODE ORDER
J1 GSP02 /CH STORE FUNCTION
K1
M0010
K1 LOAD CURSOR NULL GSB
A2
A2
M0020
A2 GSP02 /CH STORE FUNCTION
M0030
B2 LOAD BLANK OR NULL INDICATED BY INITIALIZATION OPTION
C2
M0060
C2 GSP02 /CH TEXT FUNCTION INITIALIZE MESSAGE ENTITY
M0065
D2 LOAD GNOP
E2 GSP02 /CH STORE FUNCTION
COMMON CODING
M0105
G2 DSPYN /DD FUNCTION 1943 DISABLE GSP07
H2 SET UPDATE INDICATOR SET ECCB TO UNCONTROLLED
J2 UELMT /BD UPDATE AS UNCONTROLLED ENTITY
K2 RETURN
A3
ECCB FOUND
M0070
A3 MESSAGE ENTITY
YES
NO
F2
M0100
B3 INITIALIZE OPTION=0
YES
F3
NO
D5
E3 SET ELEMENT END IDICATOR IN ECCB
F3
M0200
F3 UPDATE FUNCTION
NO
YES
G3 EELMT /BB END UPDATE FUNCTION
H3 SET INDIC IN ECCB TO MESSAGE ENTITY
J3 DSPYN /DD FUNCTION 1943 ENABLE GSP07
K3 GSP01 RETURN
A4 CHARACTER SIZE SAME
NO
YES
M0080
B4 ELEMENT SIZE=SPECIFIED
YES
NO
M0085
C4 ELEMENT SMALLER THAN SPECIFIED
NO
YES
SMALLER
D4 INITIALIZE OPTION=0
NO
YES
M0110
E4 SET ELSTA IN ECCB AND COUNT FOR TEXT FUNCTION
M0105
F4 GA/G2 UPDATE POINTERS COMMON CODE
G4 LOAD BLANK GSB
C2
H4
LARGER
H4 CURSOR GSB OR NULL GSB IN NEW ELEMENT
NO
YES
J4 STORE GNOP ORDER AT NEW END SET ELSTA TO NEW END
K4 GA/G2 UPDATE POINTERS COMMON CODE
F3
M0220
A5 SET TO SPECIFIED CHARACTER SIZE
H4
YES
M0120
C5 INITIALIZE OPTION=0
NO
D5
M0101
D5 SET ELSTA IN ECCB TO EXCLUDE MODE ORDER
M0105
E5 GA/G2 UPDATE POINTERS COMMON CODE
M0095
F5 CURSOR ACTIVE
NO
K1
YES
G5 UPDATE CURSOR ADDRESS LOAD CURSOR GSB
A2
H5 PUT CURSOR GSB OR NULL CURSOR GSB IN MESSAGE ENTITY

Chart GB. MSGIN - Message Entity Initialization



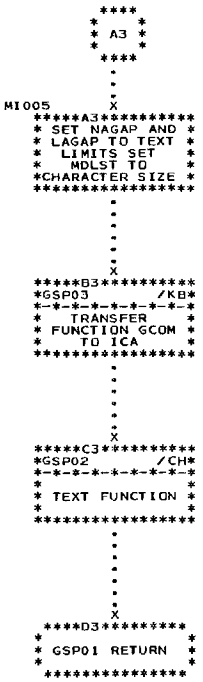

A3
MI005
A3
SET NAGAP AND LAGAP TO TEXT LIMITS SET MDLST TO CHARACTER SIZE
B3
GSP03 /KB
TRANSFER FUNCTION GCOM TO ICA
C3
GSP02 /CH
TEXT FUNCTION
D3
GSP01 RETURN

Chart GC. ICURS - Insert Cursor

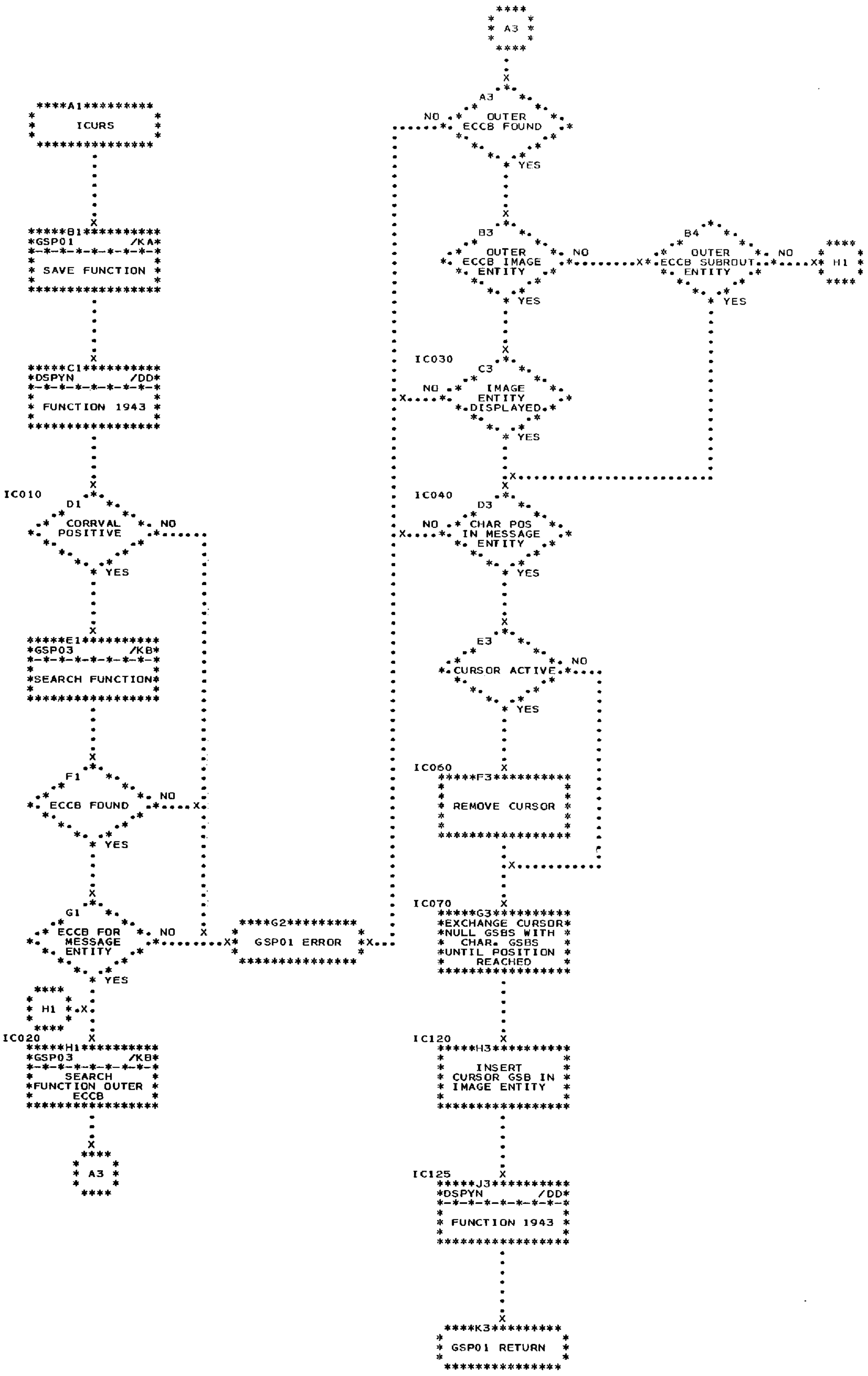


Chart GD. RCURS - Remove Cursor

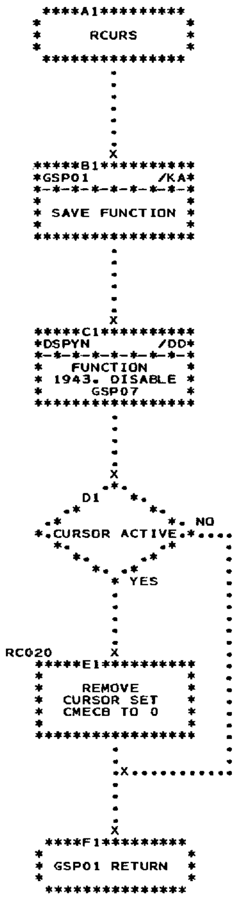


Chart GE. TLMSG - Translate Message Entity

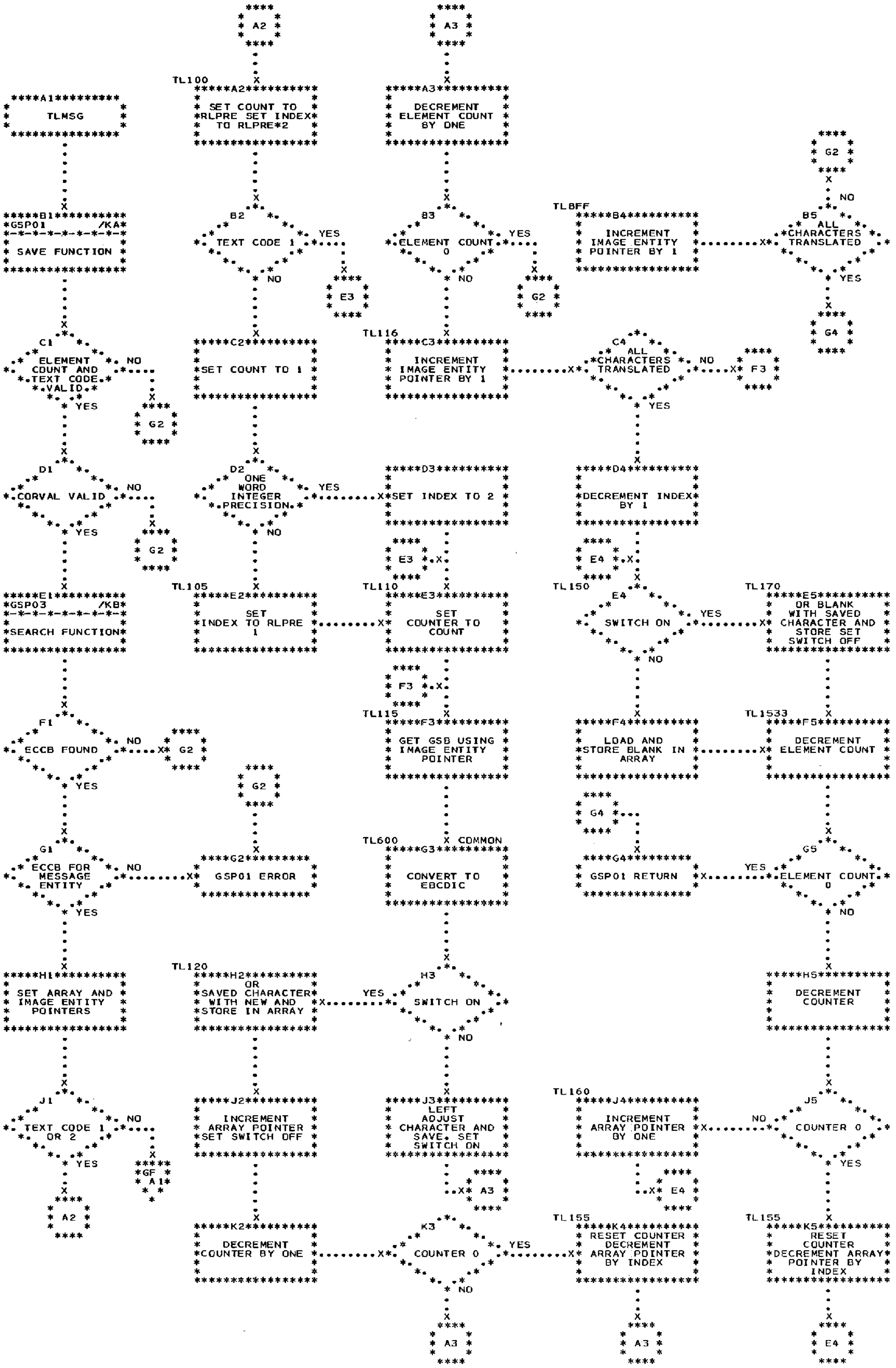


Chart GF. TLMSG - Translate Message Entity (continued)

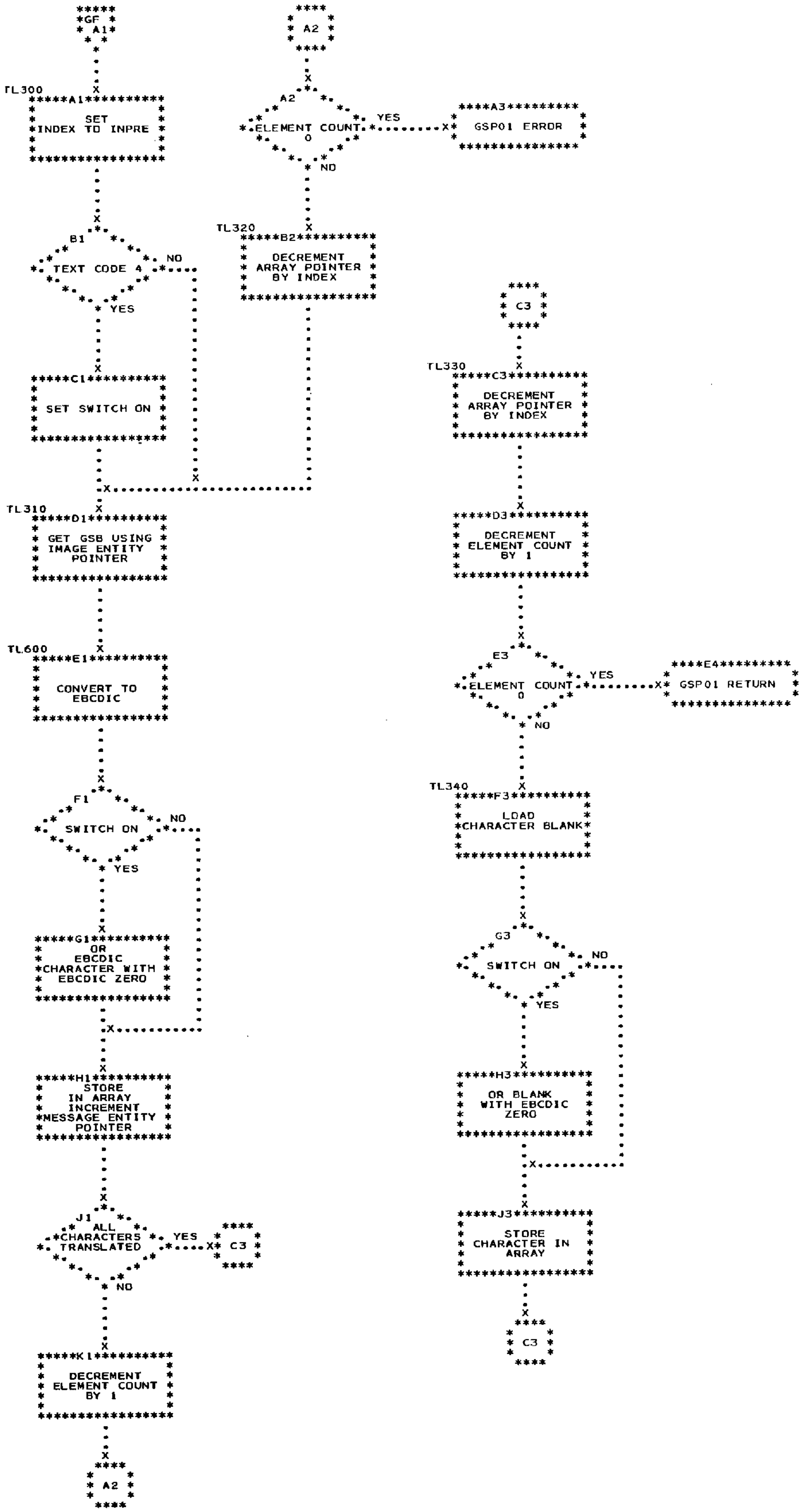


Chart GG. BCNV - Basic Conversion Subroutine

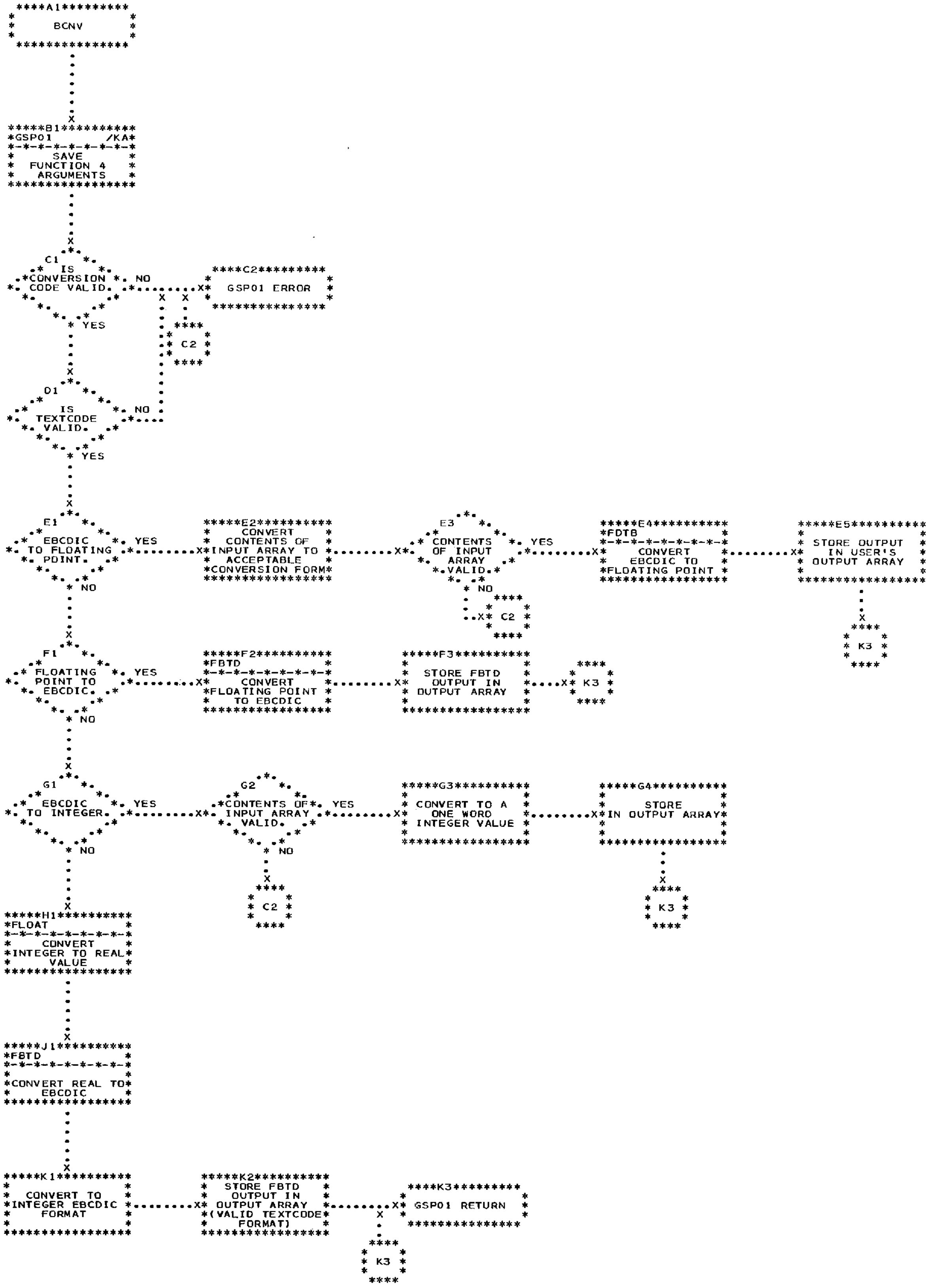

A1
BCNV
B1
GSP01 /KA
SAVE FUNCTION 4 ARGUMENTS
C1
IS CONVERSION CODE VALID.
NO
YES
C2
GSP01 ERROR
C2
D1
IS TEXTCODE VALID.
NO
YES
E1
EBCDIC TO FLOATING POINT.
YES
NO
E2
CONVERT CONTENTS OF INPUT ARRAY TO ACCEPTABLE CONVERSION FORM
E3
CONTENTS OF INPUT ARRAY VALID.
YES
NO
C2
E4
FDTB
CONVERT EBCDIC TO FLOATING POINT
E5
STORE OUTPUT IN USER'S OUTPUT ARRAY
K3
F1
FLOATING POINT TO EBCDIC.
YES
NO
F2
FBTD
CONVERT FLOATING POINT TO EBCDIC
F3
STORE FBTD OUTPUT IN OUTPUT ARRAY
K3
G1
EBCDIC TO INTEGER.
YES
NO
G2
CONTENTS OF INPUT ARRAY VALID.
YES
NO
C2
G3
CONVERT TO A ONE WORD INTEGER VALUE
G4
STORE IN OUTPUT ARRAY
K3
H1
FLOAT
CONVERT INTEGER TO REAL VALUE
J1
FBTD
CONVERT REAL TO EBCDIC
K1
CONVERT TO INTEGER EBCDIC FORMAT
K2
STORE FBTD OUTPUT IN OUTPUT ARRAY (VALID TEXTCODE FORMAT)
K3
GSP01 RETURN
K3

Chart HA. GSP07 - Cursor Direct Entry Subroutine

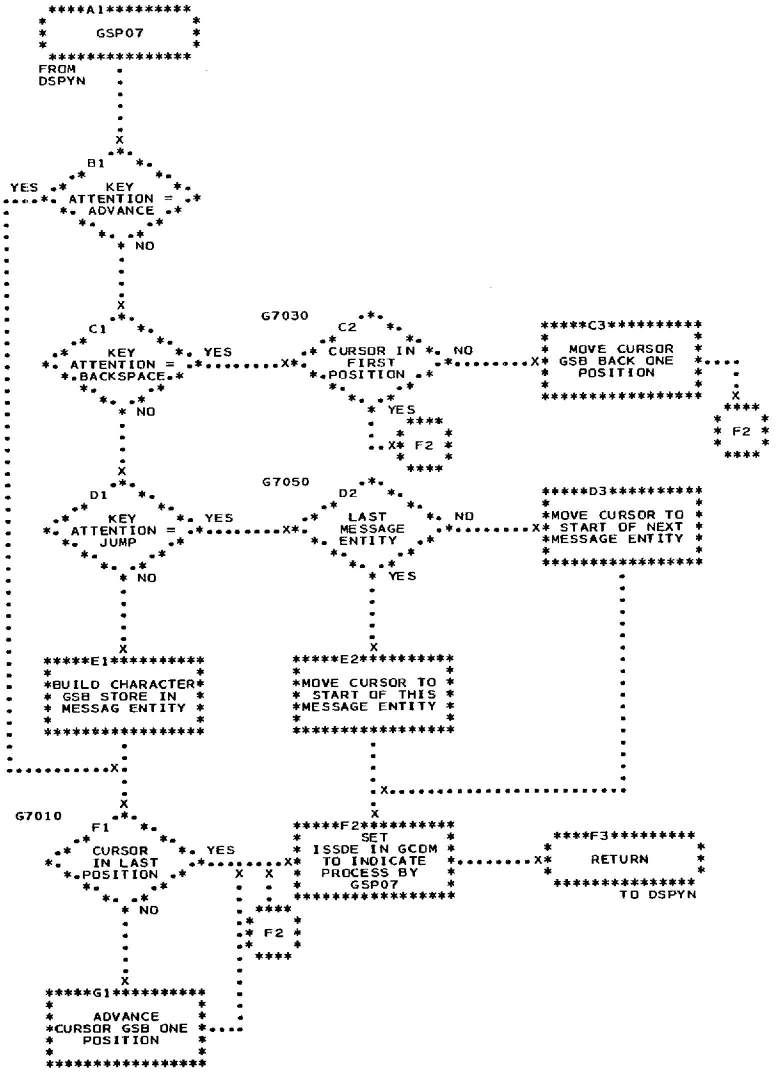

A1
GSP07
FROM DSPYN
B1
KEY ATTENTION = ADVANCE
YES
NO
C1
KEY ATTENTION = BACKSPACE
YES
NO
G7030
C2
CURSOR IN FIRST POSITION
NO
YES
F2
C3
MOVE CURSOR GSB BACK ONE POSITION
F2
D1
KEY ATTENTION = JUMP
YES
NO
G7050
D2
LAST MESSAGE ENTITY
NO
YES
D3
MOVE CURSOR TO START OF NEXT MESSAGE ENTITY
E1
BUILD CHARACTER GSB STORE IN MESSAG ENTITY
E2
MOVE CURSOR TO START OF THIS MESSAGE ENTITY
G7010
F1
CURSOR IN LAST POSITION
YES
NO
F2
SET ISSDE IN GCOM TO INDICATE PROCESS BY GSP07
F2
F3
RETURN
TO DSPYN
G1
ADVANCE CURSOR GSB ONE POSITION

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

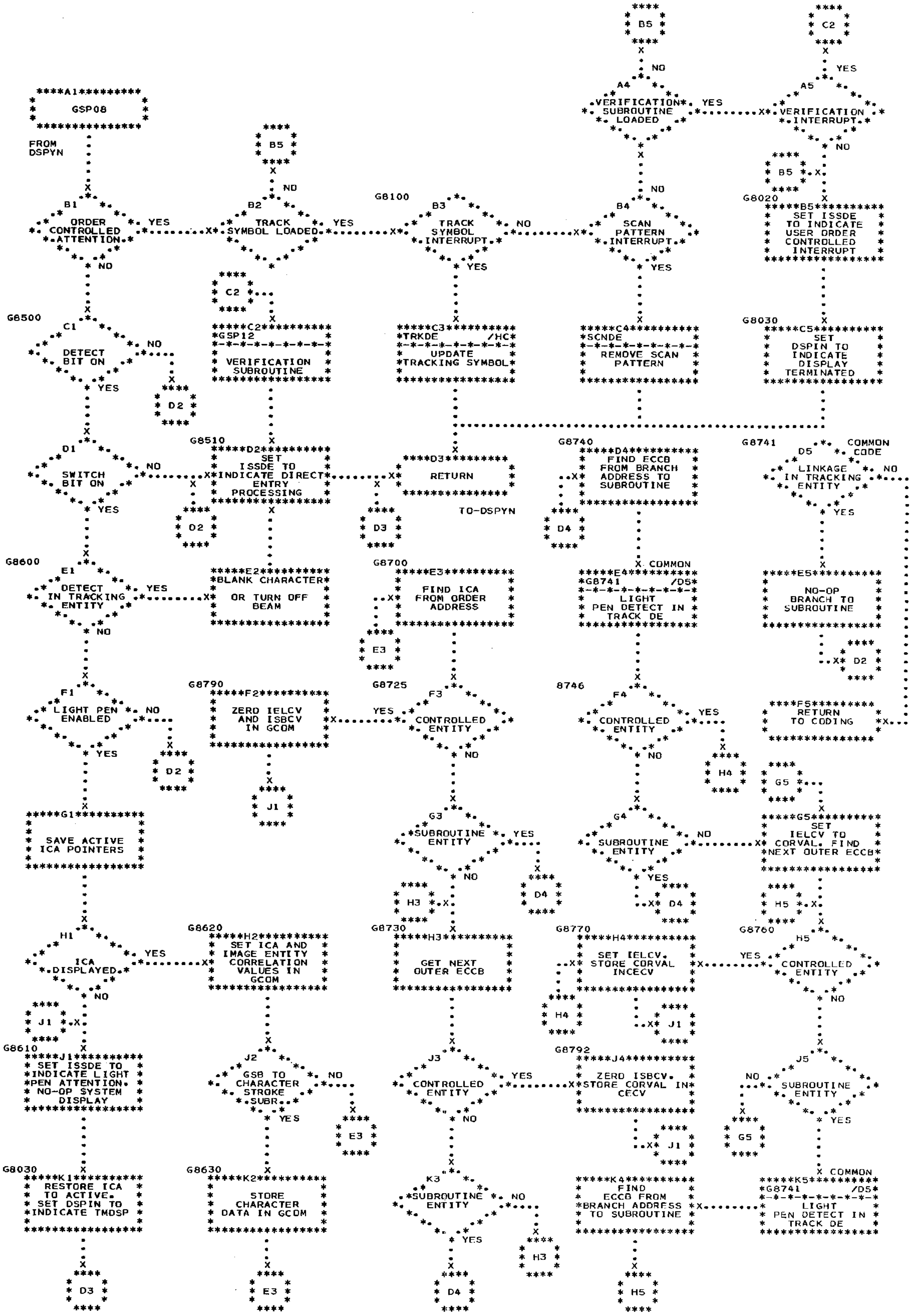

B5
C2
A4 VERIFICATION SUBROUTINE LOADED
YES
NO
A5 VERIFICATION INTERRUPT
YES
NO
A1 GSP08
FROM DSPYN
B5
B1 ORDER CONTROLLED ATTENTION
YES
NO
B2 TRACK SYMBOL LOADED
YES
NO
G8100
B3 TRACK SYMBOL INTERRUPT
NO
YES
B4 SCAN PATTERN INTERRUPT
YES
B5
G8020
B5 SET ISSDE TO INDICATE USER ORDER CONTROLLED INTERRUPT
C2
G8500
C1 DETECT BIT ON
NO
YES
D2
C2 GSP12 VERIFICATION SUBROUTINE
C3 TRKDE /HC UPDATE TRACKING SYMBOL
C4 SCNDE REMOVE SCAN PATTERN
G8030
C5 SET DSPIN TO INDICATE DISPLAY TERMINATED
D1 SWITCH BIT ON
NO
YES
G8510
D2 SET ISSDE TO INDICATE DIRECT ENTRY PROCESSING
D2
D3 RETURN
TO-DSPYN
D3
G8740
D4 FIND ECCB FROM BRANCH ADDRESS TO SUBROUTINE
D4
G8741
D5 COMMON CODE LINKAGE IN TRACKING ENTITY
NO
YES
G8600
E1 DETECT IN TRACKING ENTITY
YES
NO
E2 BLANK CHARACTER OR TURN OFF BEAM
G8700
E3 FIND ICA FROM ORDER ADDRESS
E3
COMMON
E4 G8741 /D5 LIGHT PEN DETECT IN TRACK DE
E5 NO-OP BRANCH TO SUBROUTINE
D2
F1 LIGHT PEN ENABLED
NO
YES
D2
G8790
F2 ZERO IELCV AND ISBCV IN GCOM
J1
G8725
F3 CONTROLLED ENTITY
YES
NO
8746
F4 CONTROLLED ENTITY
YES
NO
H4
F5 RETURN TO CODING
G5
G1 SAVE ACTIVE ICA POINTERS
G3 SUBROUTINE ENTITY
YES
NO
D4
H3
G4 SUBROUTINE ENTITY
NO
YES
D4
G5 SET IELCV TO CORVAL. FIND NEXT OUTER ECCB
H5
H1 ICA DISPLAYED
YES
NO
G8620
H2 SET ICA AND IMAGE ENTITY CORRELATION VALUES IN GCOM
G8730
H3 GET NEXT OUTER ECCB
G8770
H4 SET IELCV. STORE CORVAL INCECV
H4
J1
G8760
H5 CONTROLLED ENTITY
YES
NO
J1
G8610
J1 SET ISSDE TO INDICATE LIGHT PEN ATTENTION. NO-OP SYSTEM DISPLAY
J2 GSB TO CHARACTER STROKE SUBR.
NO
YES
E3
J3 CONTROLLED ENTITY
YES
NO
G8792
J4 ZERO ISBCV. STORE CORVAL IN CECV
J1
J5 SUBROUTINE ENTITY
NO
YES
G5
G8030
K1 RESTORE ICA TO ACTIVE. SET DSPIN TO INDICATE TMDSP
G8630
K2 STORE CHARACTER DATA IN GCOM
K3 SUBROUTINE ENTITY
NO
YES
H3
K4 FIND ECCB FROM BRANCH ADDRESS TO SUBROUTINE
COMMON
K5 G8741 /D5 LIGHT PEN DETECT IN TRACK DE
D3
E3
D4
H5

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

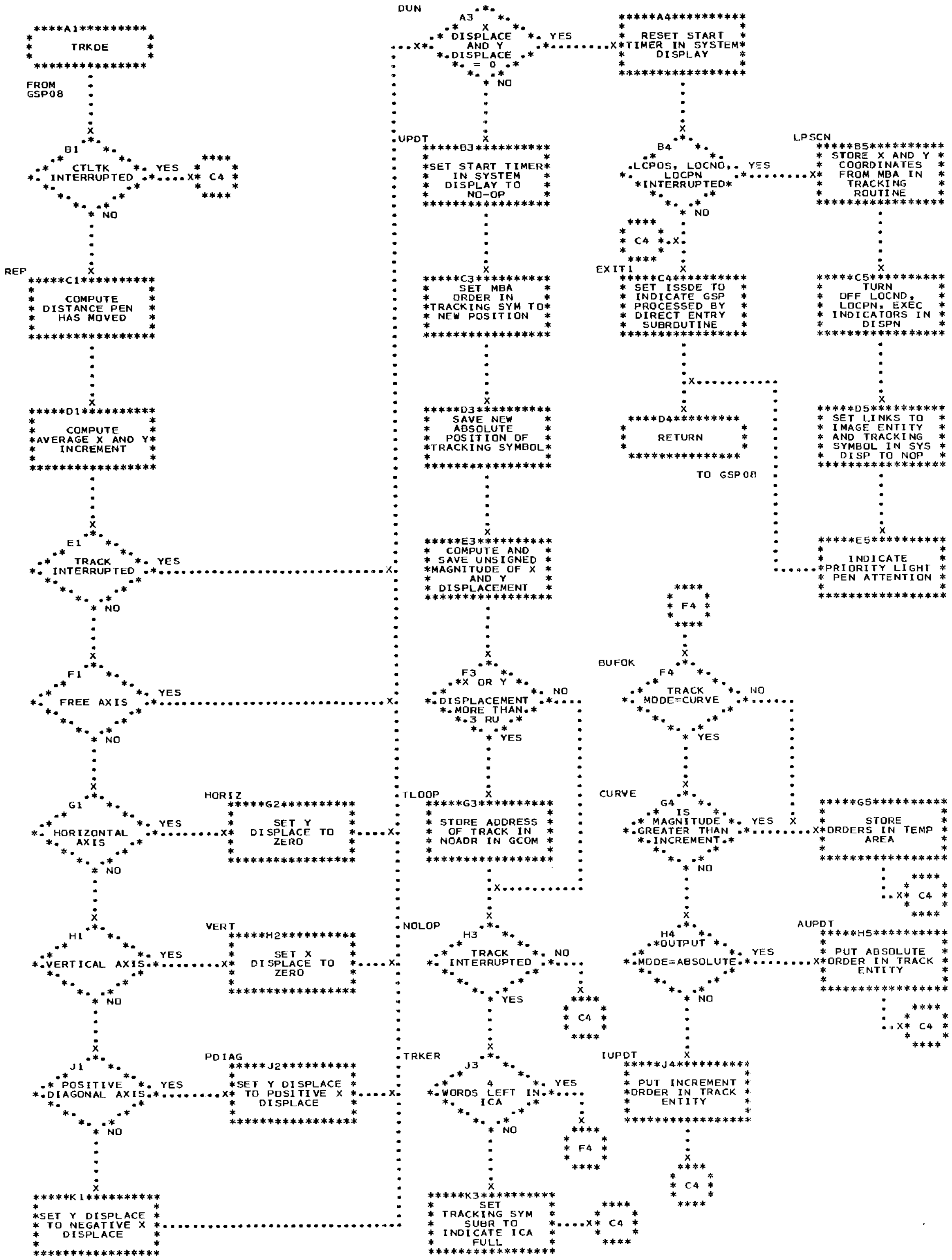

A1 TRKDE
FROM GSP08
B1 CTLTK INTERRUPTED
YES
C4
NO
REP
C1 COMPUTE DISTANCE PEN HAS MOVED
D1 COMPUTE AVERAGE X AND Y INCREMENT
E1 TRACK INTERRUPTED
YES
NO
F1 FREE AXIS
YES
NO
G1 HORIZONTAL AXIS
YES
NO
HORIZ
G2 SET Y DISPLACE TO ZERO
H1 VERTICAL AXIS
YES
NO
VERT
H2 SET X DISPLACE TO ZERO
J1 POSITIVE DIAGONAL 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
C4
YES
TRKER
J3 4 WORDS LEFT IN ICA
YES
F4
NO
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

Chart HD. GSP11 - Scan Index-Controlled Entity

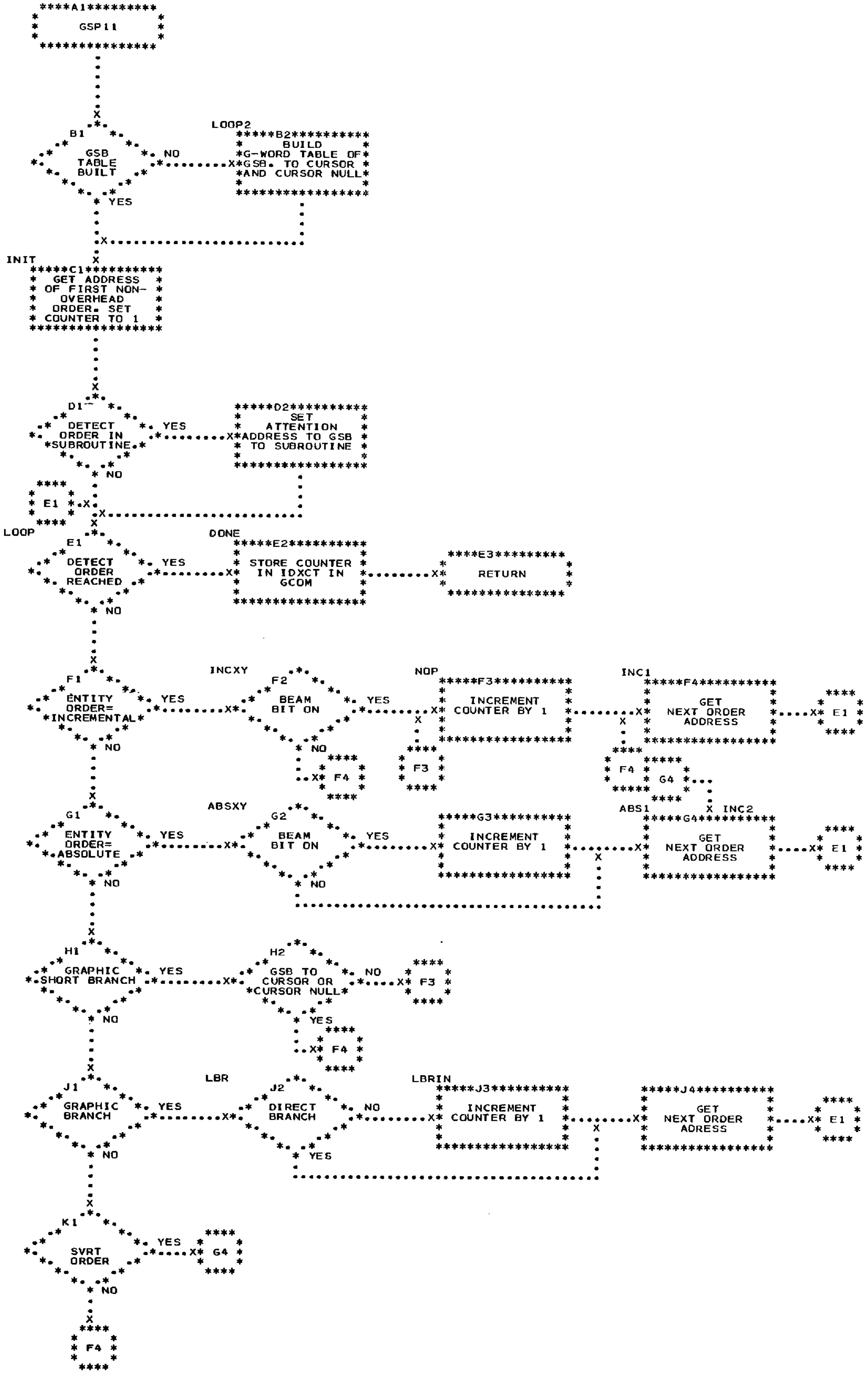


Chart JA. IERRS - Identify Errors

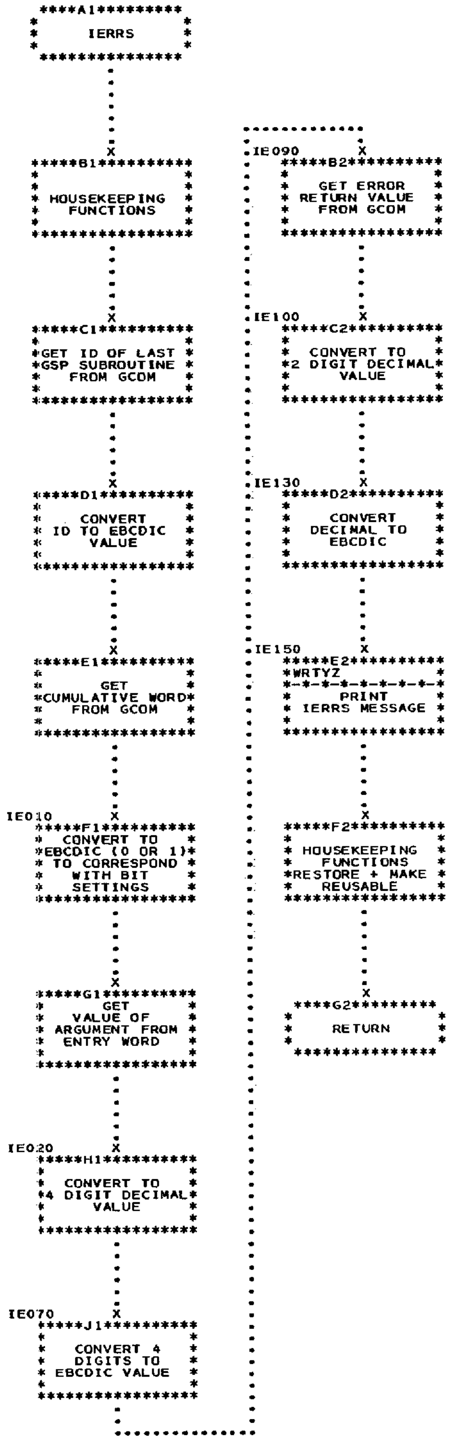

A1 IERRS
B1 HOUSEKEEPING FUNCTIONS
C1 GET ID OF LAST GSP SUBROUTINE FROM GCOM
D1 CONVERT ID TO EBCDIC VALUE
E1 GET CUMULATIVE WORD FROM GCOM
IE010
F1 CONVERT TO EBCDIC (0 OR 1) TO CORRESPOND WITH BIT SETTINGS
G1 GET VALUE OF ARGUMENT FROM ENTRY WORD
IE020
H1 CONVERT TO 4 DIGIT DECIMAL VALUE
IE070
J1 CONVERT 4 DIGITS TO EBCDIC VALUE
IE090
B2 GET ERROR RETURN VALUE FROM GCOM
IE100
C2 CONVERT TO 2 DIGIT DECIMAL VALUE
IE130
D2 CONVERT DECIMAL TO EBCDIC
IE150
E2 WRTYZ
PRINT IERRS MESSAGE
F2 HOUSEKEEPING FUNCTIONS RESTORE + MAKE REUSABLE
G2 RETURN

Chart KA. GSP01 - GSP Inner Subroutine 1

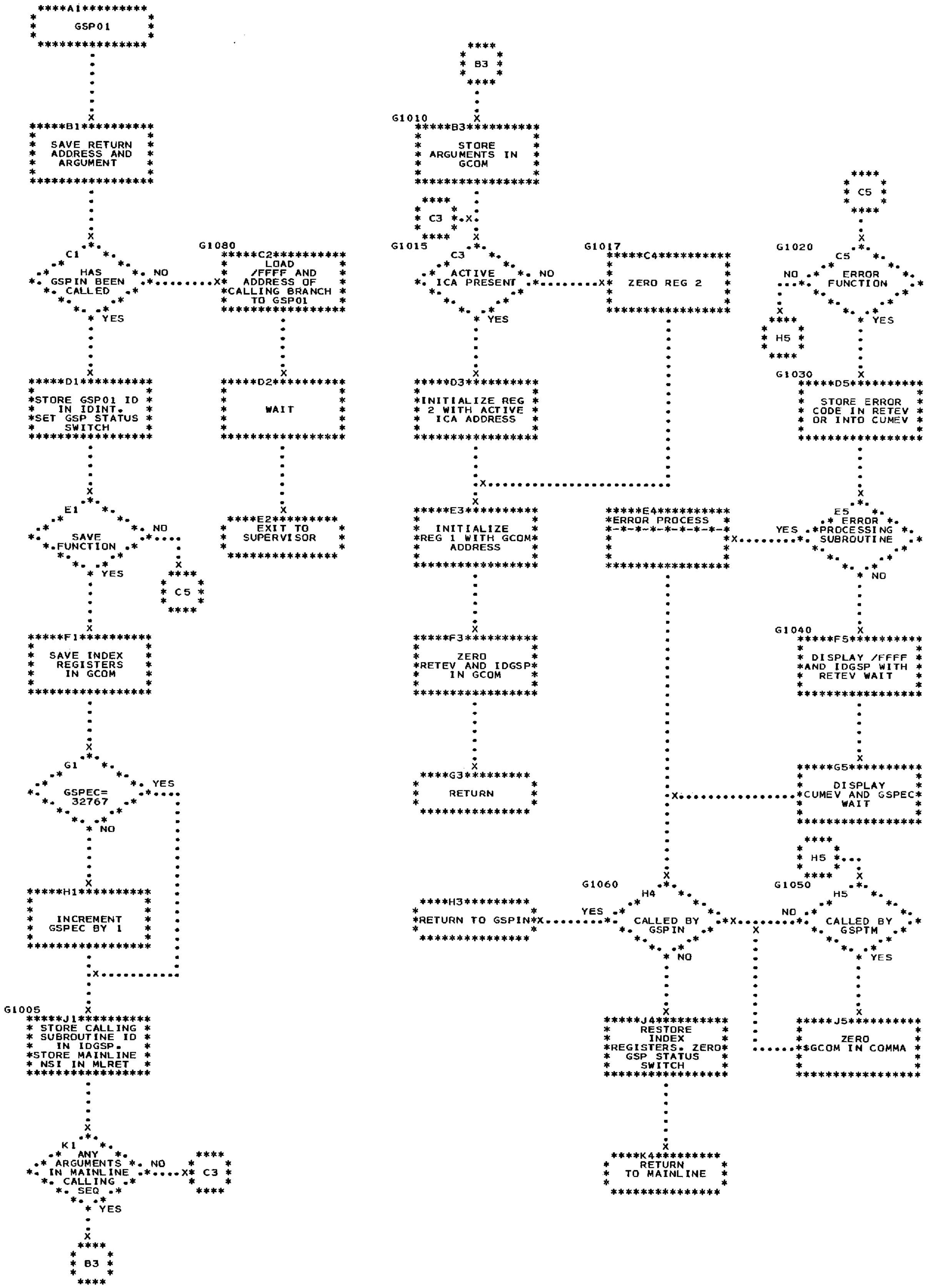

A1
GSP01
B1
SAVE RETURN ADDRESS AND ARGUMENT
C1
HAS GSPIN BEEN CALLED
NO
YES
G1080
C2
LOAD /FFFF AND ADDRESS OF CALLING BRANCH TO GSP01
D1
STORE GSP01 ID IN IDINT. SET GSP STATUS SWITCH
D2
WAIT
E1
SAVE FUNCTION
NO
YES
C5
E2
EXIT TO SUPERVISOR
F1
SAVE INDEX REGISTERS IN GCOM
G1
GSPEC= 32767
YES
NO
H1
INCREMENT GSPEC BY 1
G1005
J1
STORE CALLING SUBROUTINE ID IN IDGSP. STORE MAINLINE NSI IN MLRET
K1
ANY ARGUMENTS IN MAINLINE CALLING SEQ
NO
C3
YES
B3
B3
G1010
B3
STORE ARGUMENTS IN GCOM
C3
G1015
C3
ACTIVE ICA PRESENT
NO
YES
G1017
C4
ZERO REG 2
D3
INITIALIZE REG 2 WITH ACTIVE ICA ADDRESS
E3
INITIALIZE REG 1 WITH GCOM ADDRESS
F3
ZERO RETEV AND IDGSP IN GCOM
G3
RETURN
E4
ERROR PROCESS
C5
G1020
C5
ERROR FUNCTION
NO
YES
H5
G1030
D5
STORE ERROR CODE IN RETEV OR INTO CUMEV
E5
ERROR PROCESSING SUBROUTINE
YES
NO
G1040
F5
DISPLAY /FFFF AND IDGSP WITH RETEV WAIT
G5
DISPLAY CUMEV AND GSPEC WAIT
H5
G1060
H4
CALLED BY GSPIN
YES
NO
H3
RETURN TO GSPIN
G1050
H5
CALLED BY GSPTM
NO
YES
J4
RESTORE INDEX REGISTERS. ZERO GSP STATUS SWITCH
J5
ZERO $GCOM IN COMMA
K4
RETURN TO MAINLINE

Chart KB. GSP03 - GSP Inner Subroutine 3

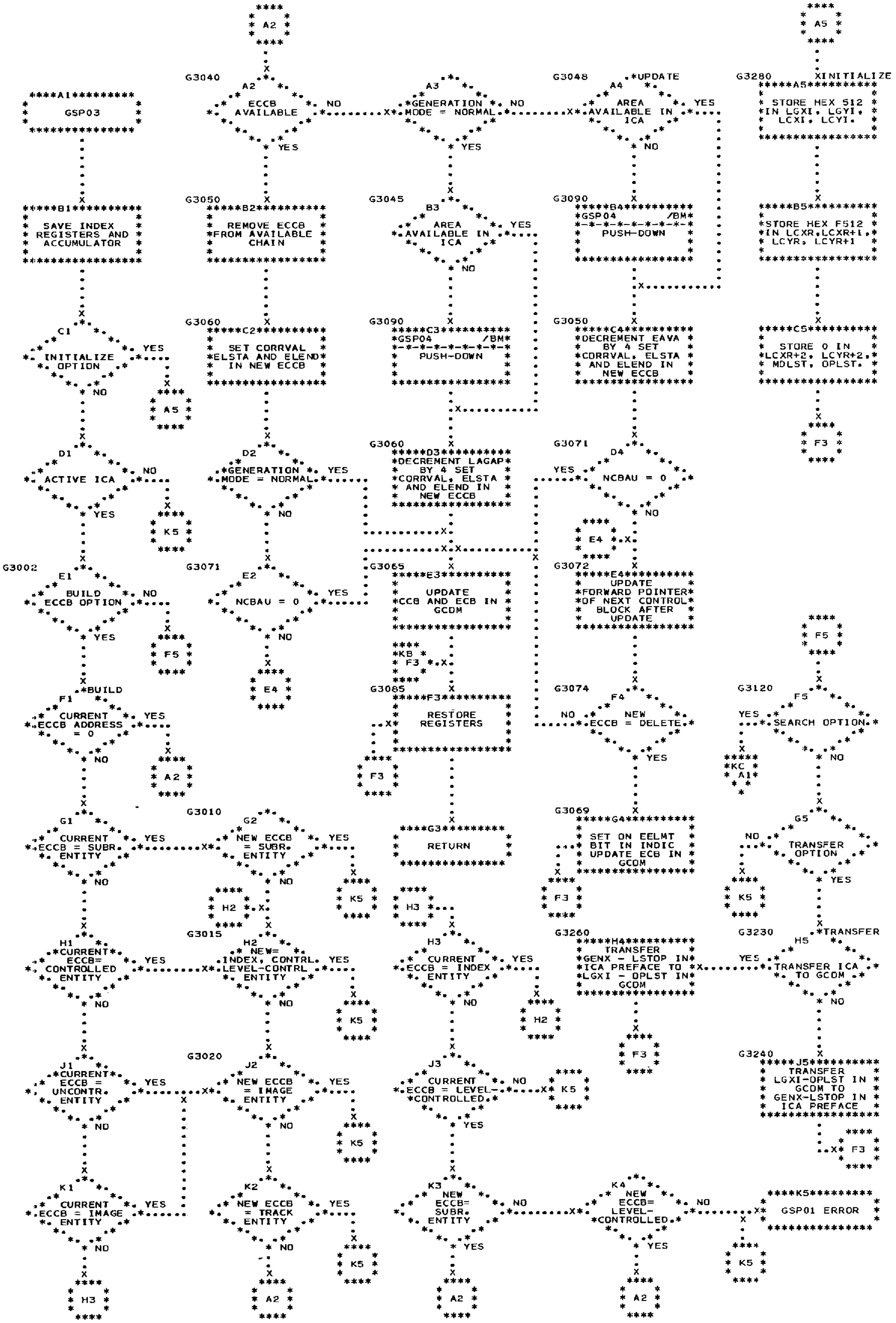
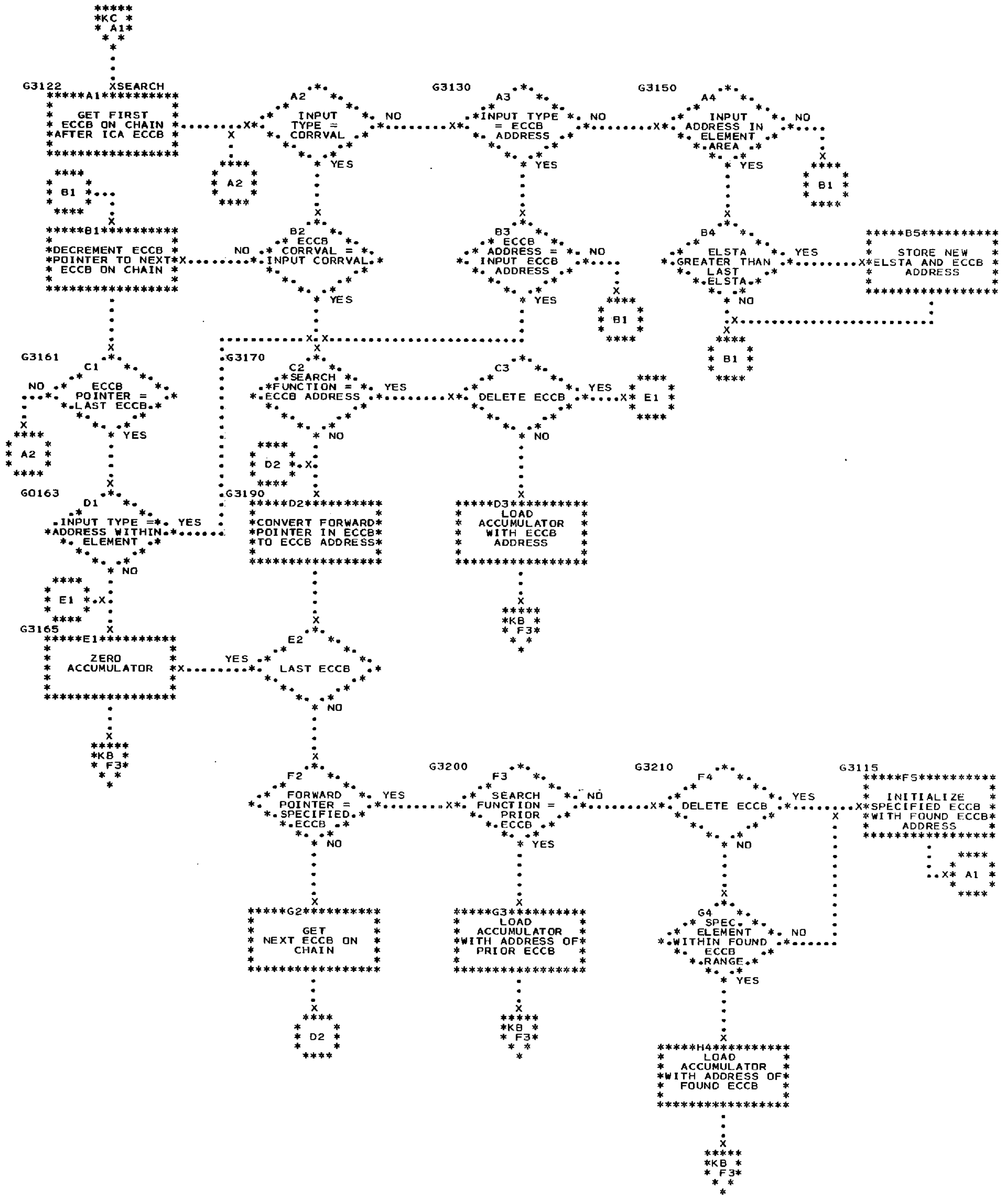


Chart KC. GSP03 - GSP Inner Subroutine 3 (continued)


KC A1
G3122 XSEARCH
A1 GET FIRST ECCB ON CHAIN AFTER ICA ECCB
A2 INPUT TYPE = CORRVAL
NO
YES
G3130
A3 INPUT TYPE = ECCB ADDRESS
NO
YES
G3150
A4 INPUT ADDRESS IN ELEMENT AREA
NO
YES
B1
A2
B1
B1 DECREMENT ECCB POINTER TO NEXT ECCB ON CHAIN
B2 ECCB CORRVAL = INPUT CORRVAL
NO
YES
B3 ECCB ADDRESS = INPUT ECCB ADDRESS
NO
YES
B1
B4 ELSTA GREATER THAN LAST ELSTA
YES
NO
B5 STORE NEW ELSTA AND ECCB ADDRESS
B1
G3161
C1 ECCB POINTER = LAST ECCB
NO
YES
A2
G3170
C2 SEARCH FUNCTION = ECCB ADDRESS
YES
NO
C3 DELETE ECCB
YES
NO
E1
D2
G0163
D1 INPUT TYPE = ADDRESS WITHIN ELEMENT
YES
NO
G3190
D2 CONVERT FORWARD POINTER IN ECCB TO ECCB ADDRESS
D3 LOAD ACCUMULATOR WITH ECCB ADDRESS
KB F3
E1
G3165
E1 ZERO ACCUMULATOR
E2 LAST ECCB
YES
NO
KB F3
F2 FORWARD POINTER = SPECIFIED ECCB
YES
NO
G3200
F3 SEARCH FUNCTION = PRIOR ECCB
NO
YES
G3210
F4 DELETE ECCB
YES
NO
G3115
F5 INITIALIZE SPECIFIED ECCB WITH FOUND ECCB ADDRESS
A1
G2 GET NEXT ECCB ON CHAIN
D2
G3 LOAD ACCUMULATOR WITH ADDRESS OF PRIOR ECCB
KB F3
G4 SPEC. ELEMENT WITHIN FOUND ECCB RANGE
NO
YES
H4 LOAD ACCUMULATOR WITH ADDRESS OF FOUND ECCB
KB F3

APPENDIX A: CONTROL BLOCK FORMATS

This appendix shows the formats of the control blocks and communications areas used by the GSP subroutines. Initial hexadecimal values are a result of a call to an initialization subroutine (GSPIN, GCAIN, ICAIN). Where xxxx is the hexadecimal value, the address is to be determined at load time or during execution.

GCOM CROSS REFERENCE

Table 6 presents in alphabetical order the names of the fields in GCOM with the respective word number. The names used in system display are not included in this table.

Table 6. Cross Reference Table for GCOM Mnemonics

Mnemonic	Word	Mnemonic	Word	Mnemonic	Word
ACHAR	85	EGCA	67	LCYR	37
ACICA	61	ENDDE	3	LGXI	32
AERRS	-36	ENGCA	68	LGYI	33
AICAP	82	GCID	-1	LPDE	0
AINDX	-37	GENIN	31	MDLST	42
ANDE	2	GSPEC	-35	MLRET	69
APD	90	GSPSW	-5	NAGAP	29
APU	89	HFFFF	-28	NGXI	54
ARG1-ARG8	73-80	ICACB	13	NGYI	55
ASCIS	84	ICACV	17	NOADR	16
ASD	88	ICNTR	83	NPXI	52
ATNAD	6	IDDE	-4	NPYI	53
ATNAN	11	IDINT	-3	OADDR	15
ATNDS	7	IDGSP	-2	OCDE	1
ATNDT	6-12	IDXCT	-39	OPLST	43
ATNOV	10	IDXIN	-38	PATRN	-45
ATNPF	10	IECV	18	PFDE	5
ATNTP	7	IELCV	20	PFKI1	23
ATNX	8	IMECB	14	PFKI2	24
ATNY	9	INCEX	59	PLOPT	-10
ATRAC	86	INCEY	60	PMSG	-55
AVER	87	INCMD	56	REG1-REG3	70-72
CANDE	4	INCSX	57	RETEV	62
CCXR-CCYI	44-51	INCSY	58	RLPRE	65
CECV	19	INPRE	64	RQNDA	26
CMECB	27	ISBCB	22	SATEN	25
CUMEV	63	ISCW	81	SSA	-6
CURAD	-34	ISSDE	12	SUBCV	21
C1-C16	-11 through -26	LAGAP	30	SUBM	-7
C42	-33	LASTX	-40	TKINC	-8
C512	-29	LASTY	-41	TKMOD	-9
C1023	-27	LCXI	40	VERLN	66
DCHAR	11	LCYI	41	WORK1-WORK4	91-94
DSPIN	28	LCXR	34	XF512	-32

GCOM FORMAT

Table 7 shows the format of GCOM giving the mnemonic, word number, a brief description of the function, and the initial value of each word. The initial value is set by the initialization subroutine GSPIN.

Table 7. GSP Communication Area Format (GCOM)

			Initial
			Hexadecimal
Mnemonic	Word	Name & Function	Value
*******************PROCESSING MESSAGE**********************************			
PMSG	-55	Set Graphic Mode Vector (SGMV) order	3200
	-54	Move Beam Absolute (MBA) order	4200
	-53	Coordinate for MBA (512)	0200
	-52	Draw Beam Incremental (DBI) 0,63	80BF
	-51	Draw Beam Incremental (DBI) 25,0	9980
	-50	Draw Beam Incremental (DBI) 10,-10	8AF6
	-49	Draw Beam Incremental (DBI) 0,-10	80F6
	-48	Draw Beam Incremental (DBI) -10,-10	F6F6
	-47	Draw Beam Incremental (DBI) -25,0	E780
	-46	Graphic Short Branch (GSB) to SD1	0xxx
*******************SCAN PATTERN**			
PATRN	-45	Draw Beam Stroke (DBS) 0,7	
		Draw Beam Stroke (DBS) 6,7	8F6F
	-44	Draw Beam Stroke (DBS) 6,0	
		Draw Beam Stroke (DBS) 0,0	E808
	-43	Draw Beam Stroke (DBS) 6,7	
		Move Beam Stroke (MBS) 0,7	EF07
	-42	Draw Beam Stroke (DBS) revert 6,0	E880
**			
LASTY	-41	Last y coordinate for CVTTD	0000
LASTX	-40	Last x coordinate for CVTTD	0000
IDXCT	-39	Index count and linkage address	0000
IDXIN	-38	Index entity indicator	0000
AINDX	-37	Address of GSP11	xxxx
AERRS	-36	Address of the error processing subroutine	xxxx
GSPEC	-35	Number of times GSP01 is called (by external subroutines)	0000
CURAD	-34	Cursor address	0000

*******************************CONSTANTS*****************************

C42	-33	Constant of 42	002A
XF512	-32	Extended floating point constant 512 of 512	008A 4000 0000
C512	-29	Constant of 512	0200
HFFFF	-28	Constant of hexadecimal FFFF	FFFF
C1023	-27	Constant of 1023	03FF
C16	-26	Constant of 16	0010
C15	-25	Constant of 15	000F
C14	-24	Constant of 14	000E
C13	-23	Constant of 13	000D
C12	-22	Constant of 12	000C
C11	-21	Constant of 11	000B
C10	-20	Constant of 10	000A
C9	-19	Constant of 9	0009
C8	-18	Constant of 8	0008
C7	-17	Constant of 7	0007
C6	-16	Constant of 6	0006
C5	-15	Constant of 5	0005
C4	-14	Constant of 4	0004
C3	-13	Constant of 3	0003
C2	-12	Constant of 2	0002
C1	-11	Constant of 1	0001
PLOPT	-10	Point or line option (0001=line; 0002=point)	0000
TKMOD	-9	Track mode (0001=position)	0000
TKINC	-8	Track distance (increment)	0030
SUBM	-7	Subroutine mode indicator	0000
SSA	-6	Address of the GSP03 search function save area	xxxx
GSPSW	-5	GSP status indicator (0=program; 1=GSP)	0000
IDDE	-4	ID word of direct entry subroutine	0000
IDINT	-3	ID word of inner GSP subroutine	0000
IDGSP	-2	ID word of outer GSP subroutine	0000
GCID	-1	GCOM identification word	0000

```
************************
* 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 for 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 coordinate (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

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

                      bit pattern for 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

LGYI	33	Last generated y integer	0200
LCXR	34	Last computed x real extended	008A 4000 0000
LCYR	37	Last computed y real extended	008A 4000 0000
LCXI	40	Last computed x integer	0200
LCYI	41	Last computed y integer	0200
MDLST	42	Last character or graphic mode (bits 12-15) bit pattern for bits 12-15 0001 vector 0010 point 0100 basic character size 1000 large character size	0000
OPLST	43	Last output mode bit pattern for bits 14-15 01 absolute 10 incremental 11 optimized	0000
******************** OUTPUT OF SCALING FORMULAS ***********************			
CCXR	44	Current computed x real extended	0000 0000 0000
CCYR	47	Current computed y real extended	0000 0000 0000
CCXI	50	Current computed x integer	0000
CCYI	51	Current computed y integer	0000
******************** OUTPUT OF SCISSOR SUBROUTINE **********************			
NPXI	52	New position point for x-coordinate	0000
NPYI	53	New position point for y-coordinate	0000
NGXI	54	New generation point for x-coordinate	0000
NGYI	55	New generation point for y-coordinate	0000
******************** INPUT TO GSP02 INCREMENT FUNCTION *****************			
INCMD	56	Increment mode (bit 15) bit pattern for bit 15 0 blanked mode 1 unblanked mode	0000
INCSX	57	Increment start x coordinate	0000
INCSY	58	Increment start y coordinate	0000
INCEX	59	Increment end x coordinate	0000
INCEY	60	Increment end y coordinate	0000
**			
ACICA	61	Address of the ECCB for the active ICA	0000
RETEV	62	Address of the return error variable	xxxx
CUMEV	63	Address of the cumulative error variable	xxxx

INPRE	64	Integer precision (1, 2 or 3 words per element)	0002
RLPRE	65	Real precision (2 or 3 words per element)	0002
VERLN	66	Verification level number	0001
EGCA	67	External generation count	0000
ENGCA	68	External non-generation count	0000

**************** FIELDS SET BY GSP01 SAVE FUNCTION ********************

MLRET	69	Mainline return address	0000
REG1	70	Save area for index register 1	0000
REG2	71	Save area for index register 2	0000
REG3	72	Save area for index register 3	0000
ARG1	73	Save area for argument 1	0000
ARG2	74	Save area for argument 2	0000
ARG3	75	Save area for argument 3	0000
ARG4	76	Save area for argument 4	0000
ARG5	77	Save area for argument 5	0000
ARG6	78	Save area for argument 6	0000
ARG7	79	Save area for argument 7	0000
ARG8	80	Save area for argument 8	0000

****

ISCW	81	Inner subroutine communication word	0000
AICAP	82	Address of the active ICA preface	0000
ICNTR	83	Counter for internal use	0000
ASCIS	84	Address of scissor subroutine (set by GSPIN if requested)	xxxx
ACHAR	85	Address of GCHAR or GUPER (set by GSPIN if requested)	xxxx
ATRAC	86	Address of TRACK subroutine	0000
AVER	87	Address of verification subroutine	0000
ASD	88	Address of system display	xxxx
APU	89	Address of GSP05 (Push-up) (set by GSPIN if requested)	xxxx
APD	90	Address of GSP04 (Push-down)	0000
WORK1-WORK4	91-94	Work areas for general use	0000

****
* Entry Point for System Display *
****

STMR1	95	Start Timer	3400

SDSW	96	Graphic Branch Conditional on light pen switch	2801
	97	Address word of SDSW	xxxx
SD1	98	Start Timer	3400
SD2	99	Linkage to processing message	2C00
SD2AD	100	Address word of SD2	xxxx
SD3	101	Linkage to scan order subroutine	2C00
SD3AD	102	Address word of SD3 (SCAN1)	xxxx
SD4	103	Linkage to tracking symbol order subroutine	2C00
SD4AD	104	Address word of SD4 (a location in GSP09)	xxxx
	105	Set Pen Mode order (SPM /5)	3105
	106	Set Graphic Mode Vector order (SGMV)	3200
	107	Move Beam Absolute order (MBA) to move	4200
	108	beam positioning to middle of screen	0200
SD5	109	Linkage to image entity of user	2C00
SD5AD	110	Address word of SD5	xxxx
SD6	111	Linkage to scan order subroutine on no detect	2C00
	112	Address word of SD6 (SCNND)	xxxx
	113	Graphic Short Branch order to SD1	0xxx

****

SCAN1	114	Set Pen Mode order (SPM /9)	3109
SCAN1A	115	Set Character Mode Basic order (SCMB)	3300
	116	Graphic Short Branch order to Scan Pattern	0xxx
	117	GNOP order	3100
SCAN2	118	Graphic Branch Conditional on light pen detect	2802
	119	Address word of GBC (DET)	xxxx
	120	Graphic Short Branch to SCAN1	0xxx
DET	121	Start Timer	3400
SCINT	122	Graphic Interrupt Conditional on light pen switch	2001
	123	A value in the GIC	0002
	124	Graphic Short Branch to SCAN1	0xxx
SCNND	125	Graphic Branch Conditional	2801
	126	Address word of GBC (SCAN1)	xxxx
	127	Graphic Short Branch to SD1	0xxx

GCA FORMAT

Table 8 shows the format of the GCA giving the mnemonic, word number, a brief description, and the initial value of each word. The initial values are set by the initialization subroutine GCAIN, or selectively set by SDATM, SGRAM, SINCR, SINDX, SSCAL, or SSCIS.

Table 8. Generation Control Area Format (GCA)

Mnemonic	Word	Name & Function	Initial Hexadecimal Value
GLLX	0	Grid lower left x coordinate	0000
GLLY	1	Grid lower left y coordinate	0000
GURX	2	Grid upper right x coordinate	03FF
GURY	3	Grid upper right y coordinate	03FF
XMSCL	4	Multiply scale factor x value	0000 0000 0001
XASCL	7	Addition scale factor for x value	0000 0000 0000
YMSCL	10	Multiply scale factor for y value	0000 0000 0001
YASCL	13	Addition scale factor for y value	0000 0000 0000
XSINC	16	Start increment for x value	0000 0000 0000
YSINC	19	Start increment for y value	0000 0000 0000
XEINC	22	End increment for x value	0000 0000 0000
YEINC	25	End increment for y value	0000 0000 0000
XSIND	28	Start index for x value	0001
YSIND	29	Start index for y value	0001
XEIND	30	End index for x value	0001
YEIND	31	End index for y value	0001
XIPMD	32	Input mode for x value (bits 0-2)	2580
		bit pattern for bits 0-2 *001 real absolute 010 real incremental 011 integer absolute 100 integer incremental 101 integer absolute in 2250 raster units	
YIPMD	32	Input mode for y value (bits 3-5)	see XIPMD
		bit pattern for bits 3-5 *001 real absolute 010 real incremental 011 integer absolute 100 integer incremental 101 integer absolute in 2250 raster units	
OPMD	32	Output mode (bits 6-7)	see XIPMD

		bit patterns for bits 6-7 *01 optimized output graphic mode 10 absolute output graphic mode 11 incremental output graphic mode	
SCIS	32	Scissor Option (bits 8-9)	see XIPMD
		bit patterns for bits 8-9 01 scissoring at screen boundaries 10 scissoring at grid boundaries	

Words 33-41 Reserved

ICA PREFACE FORMAT

Table 9 shows the format of an ICA preface giving the mnemonic, word number, a brief description and the initial value of each word. The initial values are set by the initialization subroutine ICAIN.

Table 9. ICA Preface Format

Mnemonic	Word	Name & Function	Initial Hexadecimal Value
SAVA	0	Start available area address (position for first graphic order in ICA)	xxxx
EAVA	1	Save area for LAGAP (in GCOM) during update function.	xxxx
NAVWD	2	Save area for NAGAP (in GCOM) during update function.	xxxx
SCB	3	Start of active ECCB chain (address of ICA ECCB)	xxxx
ECB	4	End of active ECCB chain (address of last ECCB)	xxxx
FAVCB	5	Address of first available ECCB on inactive chain	0000
CCB	6	Address of ECCB for current element	0000
SCCB	7	Save area for current ECCB (CCB address)	0000
CUCB	8	Current update control block (address)	0000
NCBAU	9	Address of the next control block after update ECCB	0000
DLCNT	10	Number of ECCBs for deleted elements	0000
ICAPT	11	Address of the previous ICA	
GENX	12	Last x coordinate generated	0200
GENY	13	Last y coordinate generated	0200
CALX	14	Last x coordinate computed (real extended precision)	008A 4000 0000
CALY	17	Last y coordinate computed (real extended precision)	008A 4000 0000
CALXI	20	Last x coordinate computed (integer precision)	0200
CALYI	21	Last y coordinate computed (integer precision)	0200
LSTMD	22	Last character or graphic mode (bits 12-15) bit pattern for bits 12-15 0001 vector 0100 basic character size 1000 large character size	0000

LSTOP	23	Last output mode (bits 12-15) bit pattern for bits 12-15 0001 absolute 0010 incremental 0011 optimized	0000

ECCB FORMAT

Table 10 shows the format of an ECCB giving the mnemonic, word number, and a brief description of each word. The values are set by the image management subroutines.

Table 10. Element Correlation Control Block Format

Mnemonic	Word	Name & Function
CORVL	0	Correlation value
ELSTA	1	Element start address
ELEND	2	Element end address
FORPT	3	Forward pointer for chaining ECCBs (bits 0-10)
INDIC	3	Element indicators (bits 11-15) bit 11 - 0=element begun; 1=element ended bit pattern for bits 12-15 0000 image entity 0001 controlled entity 0010 uncontrolled entity 0011 subroutine entity 0100 origin entity 0101 message entity 0110 linkage entity 0111 reserved 1000 level-controlled entity 1001 indexed entity 1010 reserved 1011 included element subroutine 1100 reserved 1101 stroke subroutine 1110-1111 reserved

This appendix is a cross reference of subroutine information and is presented in the following format:

MNEMONIC: the symbol of the subroutine.

ID: a hexadecimal value assigned to the subroutine and stored in the external or internal identification field in GCOM while the subroutine is processing.

WORDS: the decimal number of words in core occupied by the subroutine.

CALLED SUBROUTINE: the subroutine(s) that gains control to perform specific functions. Appendix B in *IBM 1130/2250 Graphic Subroutine Package for Basic FORTRAN IV* (Form C27-6934) gives the subroutines whose loading is caused by CALL statements within the subroutine being described.

TYPE/SUBTYPE: the code identifying the type of subroutine as follows:

Type 3 - a subprogram referenced by a LIBF statement.

Type 4 - a subprogram referenced by a CALL statement.

Type 5 - an interrupt service subprogram referenced by a LIBF statement.

Subtype 0 - an incore subprogram.

Subtype 8 - a function subprogram.

CHART: the two-character symbol assigned to the flowchart(s) of the subroutine.

Table 11. Subroutine Cross Reference Table

MNEMONIC	ID	WORDS	CALLED SUBROUTINES	TYPE/ SUBTYPE	CHART
BCNV	9A00	630	GSP01,GSP10	4/8	GG
BELMT	2400	186	GSP01,GSP02 GSP03	4/8	BA
BXGEN	B200	54	GSP01,GSP03	4/8	BH
CTLTK	A800	524	GSP01,GSP02 GSP03,DSPYN	4/8	FD,FE
CVTTD	AC00	688	GSP01,GSP03 GSP10	4/8	FF
DELMT	2C00	342	GSP01,GSP03 DSPYN	4/8	BC
DFMSG	9000	382	GSP01,GSP02 GSP03,DSPYN EELMT,UELMT	4/8	GA
DSPYN	B000	1026	GSP07,GSP08	5/0	DD,DE, DF,DG, DH,DJ
EELMT	2600	342	GSP01,GSP02 GSP03	4/8	BB
ERRIN	BA00	10	GSP01	4/8	none
EXEC	3000	306	GSP01,GSP03 DSPYN	4/8	DA
EXGEN	B400	32	GSP01,GSP03	4/8	BJ
GCAIN	5000	58	GSP01,GSP10	4/8	AC
GCHAR	0200	849		4/0	none
GSPIN	2000	226	GSP01,GSP03 DSPYN	4/8	AA
GSPTM	3400	39	GSP01,DSPYN	4/8	AF
GUPER	6000	726		4/0	none
GSP01	C000	179		3/0	KA
GSP02	C200	1504	GSP01,GSP04 GSP05,GSP06 GSP10	3/0	CH,CJ, CK,CL, CM,CN
GSP03	C400	590	GSP01,GSP04 DSPYN	3/0	KB,KC
GSP04	C600	354	GSP01,DSPYN	4/0	BM
GSP05	C800	272	GSP01,GSP03 GSP04	4/0	BN
GSP06	CA00	612	GSP10	4/0	CO
GSP07	CC00	198		3/0	HA

Table 11. Subroutine Cross Reference Table (continued)

MNEMONIC	ID	WORDS	CALLED SUBROUTINES	TYPE/ SUBTYPE	CHART
GSP08	CE00	498	GSP03,GSP09 GSP11,GSP12	3/0	HB
GSP09	D000	582		3/0	HC
GSP10	D200	222		3/0	none
GSP11	D400	156		3/0	HD
GSP12	D600	212		3/0	none
ICAIN	2200	241	GSP01,GSP03 DSPYN	4/8	AB
ICURS	9400	164	GSP01,GSP03 DSPYN	4/8	GC
IDPOS	6000	10	GSP01,GSP02	4/8	none
IELMT	B600	34	GSP01,GSP03	4/8	BG
IERRS	B800	254		4/0	JA
LCPOS	A400	44	GSP01,GSP02	4/8	FB
LKSUB	6A00	186	GSP01,GSP02 GSP03	4/8	CD,CE
LOCPN/ LOCND	A000 A200	48	GSP01	4/8	FA
MSGIN	9200	100	GSP01,GSP02 GSP03	4/8	GB
MVPOS	5E00	162	GSP01,GSP02 GSP03,EELMT UELMT	4/8	CA
PCOPY	6E00	228	GSP01,GSP03	4/8	CG
PGRID	6C00	70	GSP01,GSP02	4/8	CF
PLINE/ PPNT/ PSGMT	6200 6400 6600	54	GSP01,GSP02	4/8	CB
PTEXT	6800	50	GSP01,GSP02	4/8	CC
RCURS	9600	26	GSP01,DSPYN	4/8	GD
RDPOS	7000	16	GSP01	4/8	none
ROCOR	8400	120	GSP01,GSP03	4/8	EC
RQATN	8200	68	GSP01	4/8	EB
SATNS	8000	32	GSP01	4/8	EA
SATRB	2E00	156	GSP01,GSP03	4/8	BK
SDATM	5A00	40	GSP01	4/8	AE

Table 11. Subroutine Cross Reference Table (continued)

MNEMONIC	ID	WORDS	CALLED SUBROUTINES	TYPE/ SUBTYPE	CHART
SGRAM	5C00	30	GSP01	4/8	AE
SINCR	5800	40	GSP01,GSP10	4/8	AE
SINDX	5600	42	GSP01	4/8	AE
SPFKL	8600	156	GSP01,DSPYN	4/8	DC
SSCAL	5200	224	GSP01,GSP10	4/8	AD
SSCIS	5400	32	GSP01	4/8	AE
STPVS	3600	120	GSP01	4/8	BL
TLMSG	9800	324	GSP01,GSP03	4/8	GE,GF
TMDSP	3200	38	GSP01,DSPYN	4/8	DB
TRACK	A600	204	GSP01,GSP02 GSP03,GSP04	4/8	FC
UELMT	2800	314	GSP01,GSP02 GSP03,DELMT DSPYN	4/8	BD,BE
XELMT	2A00	188	GSP01,GSP03 GSP05,DSPYN	4/8	BF

The Graphics Initialization Program (GRIN) displays a frame which permits the 2250 operator to select various options from a list displayed on the 2250 screen. This selection is made by causing a light pen attention on a specific item in the list.

GRIN is initiated by entering a JOB (// JOB) and an XEQ (// XEQ GRIN) control record in the system IPL device.

GRIN's initial task is to enable the CANCEL key and wait for an attention from that source (ready status). Upon receiving a CANCEL key attention, GRIN displays the SELECT OPTION frame. The format of the SELECT OPTION frame is shown in Figure 25.

* SGJP

An attention of * SGJP causes a -1 to be stored in $GRIN (hexadecimal location 64). GRIN then executes a LINK to SGJP.

* STAND ALONE

An attention on * STAND ALONE causes GRIN to store a 2 in $GRIN. The EXECUTE 1130 PROGRAM frame is then displayed for the operator to specify his program. The format of the EXECUTE 1130 PROGRAM frame is shown in Figure 26.

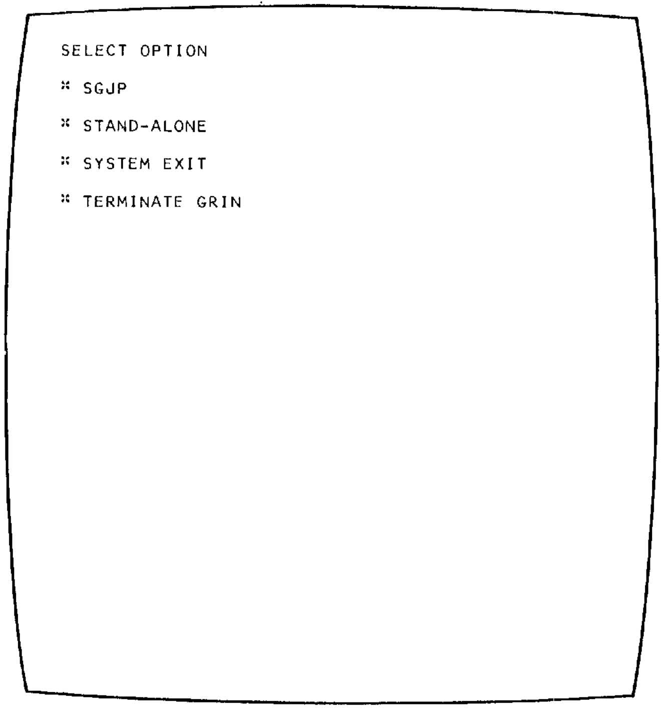
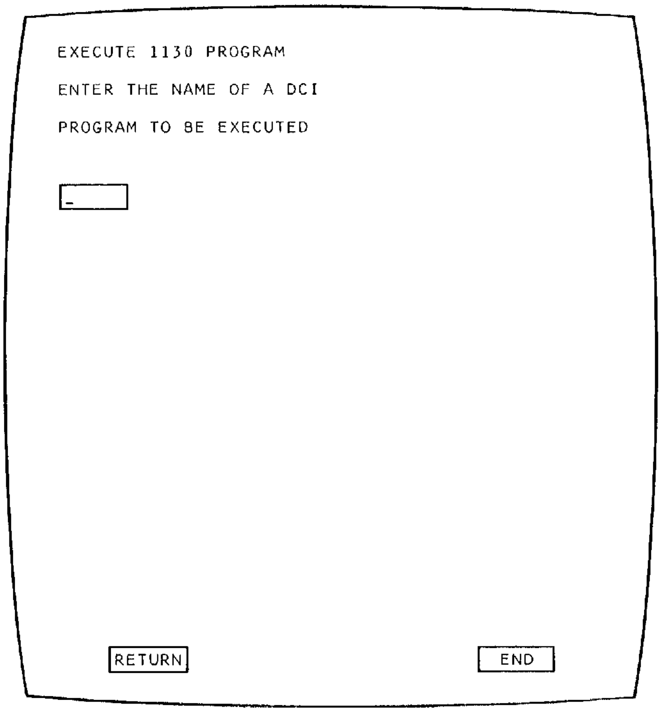


Figure 25. SELECT OPTION Frame


EXECUTE 1130 PROGRAM

ENTER THE NAME OF A DCI

PROGRAM TO BE EXECUTED

Figure 26. EXECUTE 1130 PROGRAM Frame

When the program name has been entered, an END key attention or a light pen attention on the END option in the frame causes GRIN to convert the program name to 1130 name code. 1130 name code is a modification of EBCDIC code. A search is then made of the Location Equivalence Table (LET) to determine if the program name is in the table. The third word of the program name LET entry is tested to ensure that the program is in Disk Core Image (DCI) format. (Bit 0 = 1 for DCI programs.) GRIN then builds a LINK statement to the program in order to load and pass control to it.

If the name is not in LET, the message "NAME NOT IN LET" appears to the right of the program name grid on the screen. The cursor is repositioned and the operator may then enter a new name. A program name which is contained in LET but which is not in DCI format causes the message "NOT A DCI PROGRAM" to be displayed to the right of the program name grid. An attention on the RETURN option in the frame causes GRIN to store a 1 in $GRIN and return to the SELECT OPTION frame.

* SYSTEM EXIT

This option causes GRIN to store a 0 in $GRIN and branch to $EXIT.

* TERMINATE GRIN

An attention on the * TERMINATE GRIN option results in GRIN storing a 0 in $GRIN and returning to the ready status.

Where more than one page reference is given, the first page number indicates the major reference.

alphameric
 keyboard........ 21-23,7,11,25,47-49,61-62
Assembler-language.. 7,11,15-16,22-23,35,45
attention
   enabling............. 22-23,16,51-53,64
   handling................ 47-50,15,21,152
   processing.................... 61,23-25
attentions
   on alphameric keyboard... 49,21,23,25,61
   on light pen....... 50,17,21,23-25,51,64
   on program function
    keyboard....................... 49,22-23
   on tracking symbol.................... 24
attention data area (ATDAT in
 DSPYN)................................ 24

BCNV......................... 60-61,21,56
BELMT............................... 32,31
BXGEN............................... 35,32

CALL statements................ 10-11,13,66
CANCEL key........................ 48,21
COMMA.............................. 7,14,27
communications area............... 11-12,14
   ECCB................. 12,15,18,28,68,161
   building an ECCB..................... 67
   GCA.................... 11,14-15,27,158
   GCOM............ 11,7,14,21-22,27-28,151
   ICA preface................ 11-12,15,160
   $GCOM................................ 27
control blocks.......................... 7
Core Image Buffer..................... 8-10
Core Image format...................... 7-8
Core Load Builder..................... 7-10
CTLTK...................... 54-55,19,53,63
CVTTD............................ 55-56,53

DELMT............................ 33-34,31
detectability........................... 36
DFMSG.......................... 57-58,21,56
Disk Monitor System, Version 2........... 7
Disk System format..................... 7-8
direct entry subroutine.............. 49-50
   cursor (GSP07).............. 61-62,21,25
   light pen......................... 62,23
   storing the address of.............. 48
display program......................... 7
DSPYN.......... 47-51,13-16,19,22-25,45,61

ECCB (see communications area)
EELMT............................ 32-33,31
elements (see also entity).......... 7,15,17
END key............................ 49,22
entity
   controlled.................... 32-33,36
   image..... 32,13,15,37-38,45-46,54,59,63
   index-controlled........... 63,24,32,36
   level-controlled......... 64,24,32,36,62
   linkage........................ 40-41,53
   message..................... 21,25,57-60
   origin.............................. 39,56
   subroutine............ 32-34,38,40-41,59
   tracking...................... 17-19,40
   uncontrolled...................... 32-33
ERRIN.......................... 64-65,25
error handling................... 64-65,25
error processing subroutine
 (see IERRS)
EXEC........................... 45-46,12,15
EXGEN............................ 35-36,32

FBTD..................................... 61
FDTB..................................... 61
FLOAT.................................... 28

GCA (see communications area)
GCAIN........................... 28,14,27
GCHAR........................... 57,21,56
GCOM (see communications area)
graphic orders (see 2250
 Orders)
GRIN.................................... 166
GSP01..................... 65-66,11,15,25-26
GSP02............................ 41-43,15,38
GSP03..................... 66-68,14-15,19,27
GSP04..................... 37-38,15,19,32,54
GSP05............................ 38,15,32
GSP06........................... 43-45,15,38
GSP07..................... 61-62,21,25,58-59
GSP08......................... 62,23-24,61
GSP09.................. 62-63,18-19,24,28,61
GSP10............................... 68,15
GSP11............................ 63,24,61
GSP12....................... 63-64,8,24,61
GSPIN............................ 27,13-15
GSPTM......................... 31,15-16,27
GUPER............................ 57,8,56
*G2250 Supervisor Control
 Record (see Supervisor
 Control Record)

ICA........................... 14,17,27-28
ICA preface (see
 communications area)
ICAIN............................ 28,14,27
ICURS......................... 59,21,25,56
IDPOS............................ 36-37,32
IELMT............................... 35,32
IERRS............................ 65,25-26
ILSX3............................ 23-24,19
image entity (see entity)
image generation............... 7,15,38,154
image management................... 7,15,31
initialization...................... 7,13,27
Input/Output
   commands.................. 13,15,19,48
   errors............................... 50
   operations......................... 47,15
   subroutines......................... 45
interrupts (see also

attentions)...................... 7,12,15
Interrupt Level subroutine
(see ILSX3)
Interrupt Service subroutine
(see DSPYN)

keyboard (see alphameric
keyboard or programmed
function keyboard)

LCPOS........................ 54,17-18,53
LIBF instruction.................... 10-11
light pen
attention........... 7,15,21-24,50,62,64
mode............................ 22-24
subroutines...................... 53-56
tracking...................... 17-19,64
tracking options............... 54-55,19
linkage mechanisms..................... 11
LKSUB.............................. 40,38
load table.............................. 7
locating positions..................... 17
Location Equivalence Table
(LET).................................. 8
LOCPN/LOCND................... 53-54,17-18

message collection mode.............. 21,25
message entity (see entity)
MSGIN......................... 58-59,21,56
MVPOS.............................. 38-39

overhead orders (see 2250
Orders)

PCOPY.............................. 41,38
PGRID.............................. 41,38
PLINE/PPNT/PSGMT.................... 39,38
precision.............................. 60
processing message... 11,13,33-35,37,46,151
programmed function
keyboard............... 22-24,16,31,46-48
PTEXT.............................. 39,38

RCURS......................... 59,21,25,56
RDPOS.............................. 37,32
regeneration........ 47,7,12-13,15,19,23,45
reset display.......................... 48
ROCOR.............................. 53,51
RQATN...................... 51-53,18,22,62

SATNS.............................. 51,22
SATRB.............................. 36,31
scaling...................... 28-29,15,155
scanning pattern.. 17,13,24,53-54,62-63,151
scissoring.......... 43-45,15,28,30,42,155
SDATM.......................... 30,15,27

sense device........................... 48
SGRAM.......................... 31,15,27
SINCR.......................... 30,15,27
SINDX.......................... 30,15,27
SPFKL....................... 46-47,16,45
SSCAL.......................... 29,15,27
SSCIS.......................... 30,15,27
STPVS.............................. 36,32
subroutine entity (see entity)
subroutines
cross reference table.............. 150
direct entry................. 23-25,21
external............................ 7
internal...................... 7,11,15
verification.................... 63-64
Supervisor Control Record.......... 7,27,57
analyzer............................ 7
area................................ 7
system display..........12-15,7,18-19,23-24
45-46,48,53,155

termination.................... 13,27,31
text generation.................... 56,21
TLMSG....................... 59-60,21,56
TMDSP.......................... 46,15,45
TRACK........................ 54,18-19,53
tracking entity (see entity)
tracking options (see light pen)
tracking
symbol......... 17-19,13,21,24,54-55,62-63
Transfer Vector Table............. 14,23,27

UELMT.............................. 34,31

visibility............................. 36

WAIT instruction................... 26,66

XELMT........................... 34-35,31
XEQ Monitor Control Record.............. 7
XIO commands (see Input/Output
commands)

2250 Channel Interface.................. 15
2250 Orders
Branch (GB)...................... 12-13
Draw Beam (DBA, DBI)................. 42
Interrupt (GI, GIC).................. 24
Move Beam (MBA, MBI)................. 42
Overhead............. 32-33,15,41,54,58
reference table...................... 81
Short Branch (GSB).. 8,13,21,57-59,61-63
Start Timer (STMR)............. 12-13,63
Store Revert (SRVT)......... 32,37-38,64

READER'S COMMENT FORM

IBM 1130/2250 Graphic Subroutine Package
For Basic FORTRAN IV
Program number 1130-LM-008

Y27-7174-0

Please check or fill in the items below, adding explanations and other comments in the space provided.

Which of the following terms best describes your job?

□ Programmer
□ Manager
□ Operator
□ Instructor
□ Systems Analyst
□ Engineer
□ Mathematician
□ Student/Trainee
□ Customer Engineer
□ Systems Engineer
□ Sales Representative
□ Other (explain) ______________

Does your installation subscribe to the SRL Revision Service? □ Yes □ No

How did you use this publication?

□ As an introduction
□ As a reference manual
□ As a text (student)
□ As a text (instructor)
□ For another purpose (explain) ____

Did you find the material easy to read and understand? □ Yes □ No (explain below)

Did you find the material organized for convenient use? □ Yes □ No (explain below)

Specific criticisms (explain below)

Clarifications on pages ____

Additions on pages ____

Deletions on pages ____

Errors on pages ____

Explanations and other comments:

Thank you for your cooperation. No postage necessary if mailed in the U.S.A.

Y27-7174-0

YOUR COMMENTS PLEASE . . .

This manual is one of a series which serves as reference sources for systems analysts, programmers and operators of IBM systems. Your answers to the questions on the back of this form, together with your comments, will help us produce better publications for your use. Each reply will be carefully reviewed by the persons responsible for writing and publishing this material. All comments and suggestions become the property of IBM.

Please note: Requests for copies of publications and for assistance in utilizing your IBM system should be directed to your IBM representative or to the IBM sales office serving your locality.

FOLD FOLD

FIRST CLASS
PERMIT NO. 116
KINGSTON, N. Y.

BUSINESS REPLY MAIL
NO POSTAGE STAMP NECESSARY IF MAILED IN U. S. A.

POSTAGE WILL BE PAID BY

IBM CORPORATION
NEIGHBORHOOD ROAD
KINGSTON, N. Y. 12401

ATTN: PROGRAMMING PUBLICATIONS
DEPARTMENT 637

FOLD FOLD

IBM®

International Business Machines Corporation
Data Processing Division
112 East Post Road, White Plains, N.Y. 10601
[USA Only]

IBM World Trade Corporation
821 United Nations Plaza, New York, New York 10017
[International]

Y27-7174-0

Printed in U.S.A. Y27-7174-0

International Business Machines Corporation
Data Processing Division
112 East Post Road, White Plains, N.Y. 10601
[USA Only]

IBM World Trade Corporation
821 United Nations Plaza, New York, New York 10017
[International]