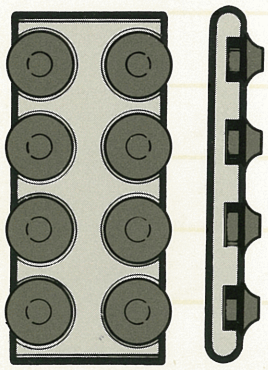
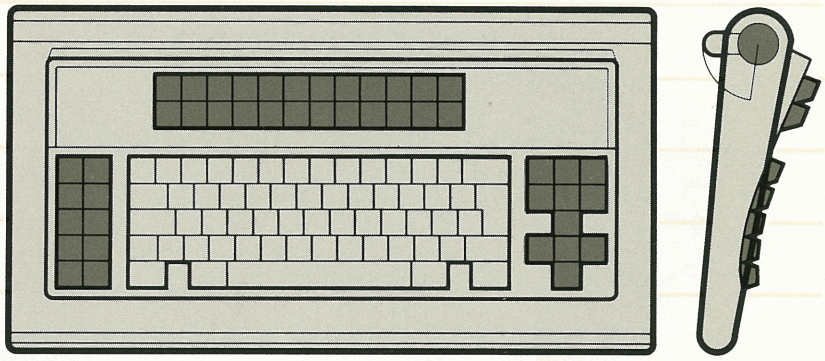
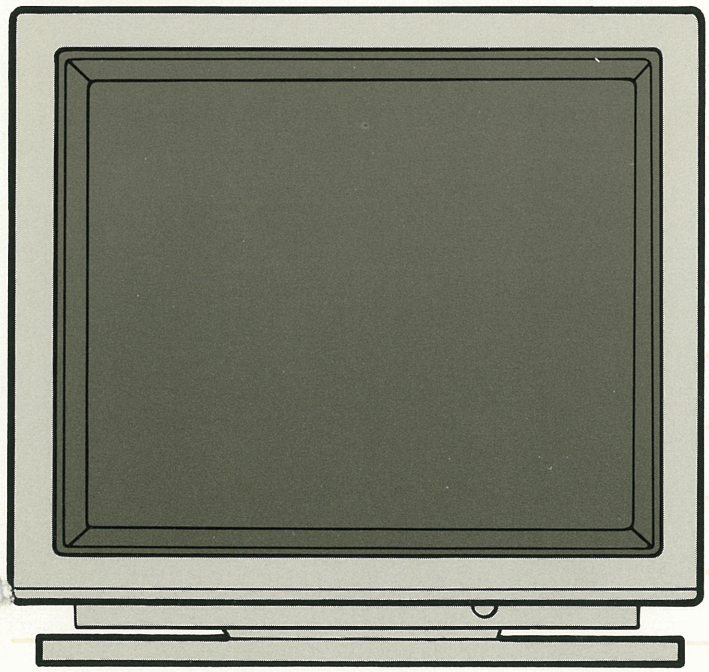
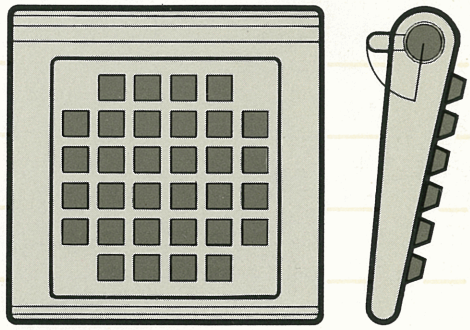
IBM 5080 Principles of Operation

Graphics System









GA23-0134-0

IBM 5080 Principles of Operation

Graphics System

GA23-0134-0

First Edition (March 1984)

This book describes the principles of operation for the IBM 5080 Graphics System.

Information in this book is subject to change. Subsequent revisions will include such changes.

This book may contain references to, or information about, IBM products (machines and programs), programming, or services that are not announced in your country. Such references must not be construed to mean that IBM intends to announce such IBM products, programming, or services in your country.

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

A form for readers' comments has been provided at the back of this book. Please send any additional comments to IBM Corporation, Department 52Q, Neighborhood Road, Kingston, New York 12401. IBM may distribute or use any of the information you supply in any way it believes appropriate without incurring any obligation whatever. You may, of course, continue to use the information you supply.

© Copyright International Business Machines Corporation 1984

Preface

About This Book

This book describes the principles of operation of the IBM 5080 Graphics System. It is also a comprehensive reference source of information about 5080 system channel commands, structured fields, instructions, status/sense data, and graphic orders.

Who Should Read This Book?

This book is for system programmers who are developing programming support for the IBM 5080 Graphics System.

What Is in This Book?

This book contains six chapters and five appendixes. They are:

- Chapter 1. General Description
- Chapter 2. Host Interface to the IBM 5080 Graphics System
- Chapter 3. Graphics Processor Architecture
- Chapter 4. Channel Commands, Structured Fields, Instructions, and Status/Sense Information
- Chapter 5. Graphic Orders
- Chapter 6. RS232C Attachment
- Appendix A. 3250-Compatible Channel Commands and Graphic Orders
- Appendix B. TCF Number Representation and Mathematics of the Transformation Process
- Appendix C. Summary of Status/Sense Byte Bit Assignments
- Appendix D. 5080 Graphics System 3270 Feature
- Appendix E. 5080 Graphics System I/O Interface Codes

Related Publications

Important related publications with which you should be familiar include:

- *IBM 5080 Graphics System: Operation and Problem Determination,* GA23-0133
- *IBM 5080 Graphics System: Programmer's Reference Card,* SX23-0257
- *IBM 5080 Graphics System: Setup Instructions,* GA23-0130
- *IBM 5080 Graphics System: Site Planning and Preparation Guide,* GA23-0129

- *IBM 5080 Graphics System: System Planning and Installation,* GA23-0135
- *IBM 5080 Graphics System: System Problem Determination,* GA23-0132
- *IBM 3250 Graphics Display System Component Description,* GA33-3037
- *IBM System/370 Principles of Operation,* GA22-2700

Other publications you may find helpful in using this book are:

- *OEMI Channel-to-Control-Unit Interface,* GA22-6974
- *OEMI Power Sequencing,* GA22-6906
- *IBM 3270 Information Display System Data Stream Programmer's Reference,* GA23-0059
- *Systems Network Architecture Concepts and Products,* GC30-3072
- *Systems Network Architecture Format and Protocol Reference Manual: Architectural Logic,* SC30-3112
- *IBM Synchronous Data Link Control General Information,* GA27-3093

Contents

Figures

Chapter 1. General Description

Introduction

The IBM 5080 Graphics System provides powerful graphics functions featuring high user interactivity capabilities, high performance, and an ergonomically advanced design. System features include a choice of color or monochrome high-resolution raster displays, a range of performance capabilities, including advanced graphics functions, a selection of graphics input/output devices, and IBM 3250 Display System compatibility.

An optional feature allows the 5080 workstation to operate also as a member of the 3270 family of displays. Thus, a workstation user is able to work alternately with graphics applications and 3270 data bases and management information systems. Both the graphics applications and 3270 applications may be concurrently active and are interchangeably displayable with the pressing of a single key. Logon/logoff actions are not required after the sessions have been activated.

Steady, clear displays of sharp, clearly delineated images in color or monochrome enable prolonged operator interaction by minimizing fatigue.

The 5080 Graphics System is designed for placement within the normal office environment. The workstation units are compact, consistent in appearance, and unobtrusively cabled.

The primary components of the 5080 Graphics System are:

- The 5081 Display
- The 5085 Graphics Processor
- The 5088 Graphics Channel Controller

A Graphics System Workstation

A workstation in the 5080 Graphics System (see Figure 1-1 on page 1-2) consists of two major components and a number of peripheral devices and features.

The two major components of a workstation are:

- The 5081 Display, available in a color model and a monochrome model.
- The 5085 Graphics Processor, containing an attachment processor, a serial interface processor, system memory, a display storage contained within system memory, and a display processor.

For graphics applications, the workstation communicates to a host CPU (IBM System/370/43XX/30XX) through a 5088 Graphics Channel Controller or a 3258 Control Unit. The 5088 Graphics Channel Controller provides a channel interface to the host system and a serial interface to the attached 5085 Graphics Processor (or 3255 in a 3250 system).

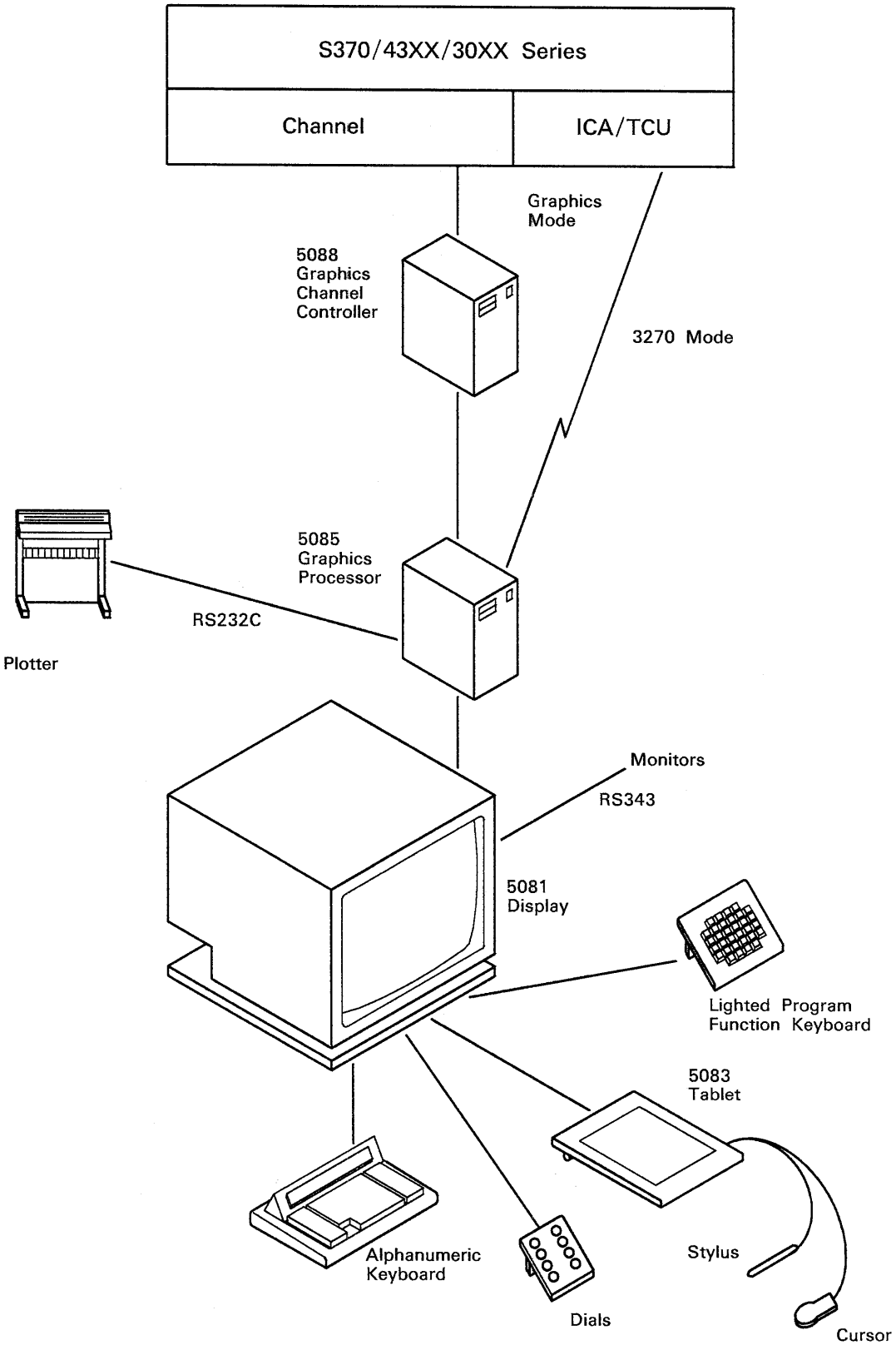


Figure 1-1. IBM 5080 Graphics System Configuration

5081 Display

The 5081 Display is physically separate from the processor. It is offered with a choice of monochrome and color models, each with tilt-swivel bases. The design of the display incorporates advanced human factors considerations to encourage fatigue-free, high user interactivity. It is compact and designed for desktop or tabletop placement in normal office environments.

Each model of the 5081 Display offers high resolution (1024 x 1024 pixels), 50Hz noninterlaced refresh rates, and a bright, sharply defined image. Each model has a 19-inch (diagonal) screen on which the image is displayed in an area approximately 11.2 inches (284 mm) square.

Either a color or monochrome display may be attached to a graphics processor. If desired, additional monitors of either type can be attached to a graphics processor as external output-only units.

The display with the peripheral connection assembly (shipped with the processor) inserted (by the user) in its base is the main, or interactive, monitor. Additional displays cannot accommodate peripheral attachments because the necessary connection facility (one peripheral connection assembly per 5085) will have been inserted by the user in the base of the main, or interactive, monitor. The *peripheral connection assembly (PCA)* enables the connection of an alphanumeric keyboard, tablet, lighted program function keyboard, and the dials feature to the workstation.

Refer to *IBM 5080 Graphics System: Planning and Installation,* GA23-0135, for details on attaching additional displays to a processor.

5085 Graphics Processor

The 5085 Graphics Processor attaches to the 5088 Graphics Channel Controller (or a 3258 Control Unit) and provides the buffering and control necessary to support all workstation functions. The input/output peripherals connect to the processor through the PCA inserted, as described above, in the tilt-swivel base of the 5081 Display and are controlled by the processor. The processor is contained in a slim, compact housing designed for floor placement alongside a desk or table. The standard features included in this unit are described in the following paragraphs. See Figure 1-2 on page 1-4 for an overview of the 5085 architecture.

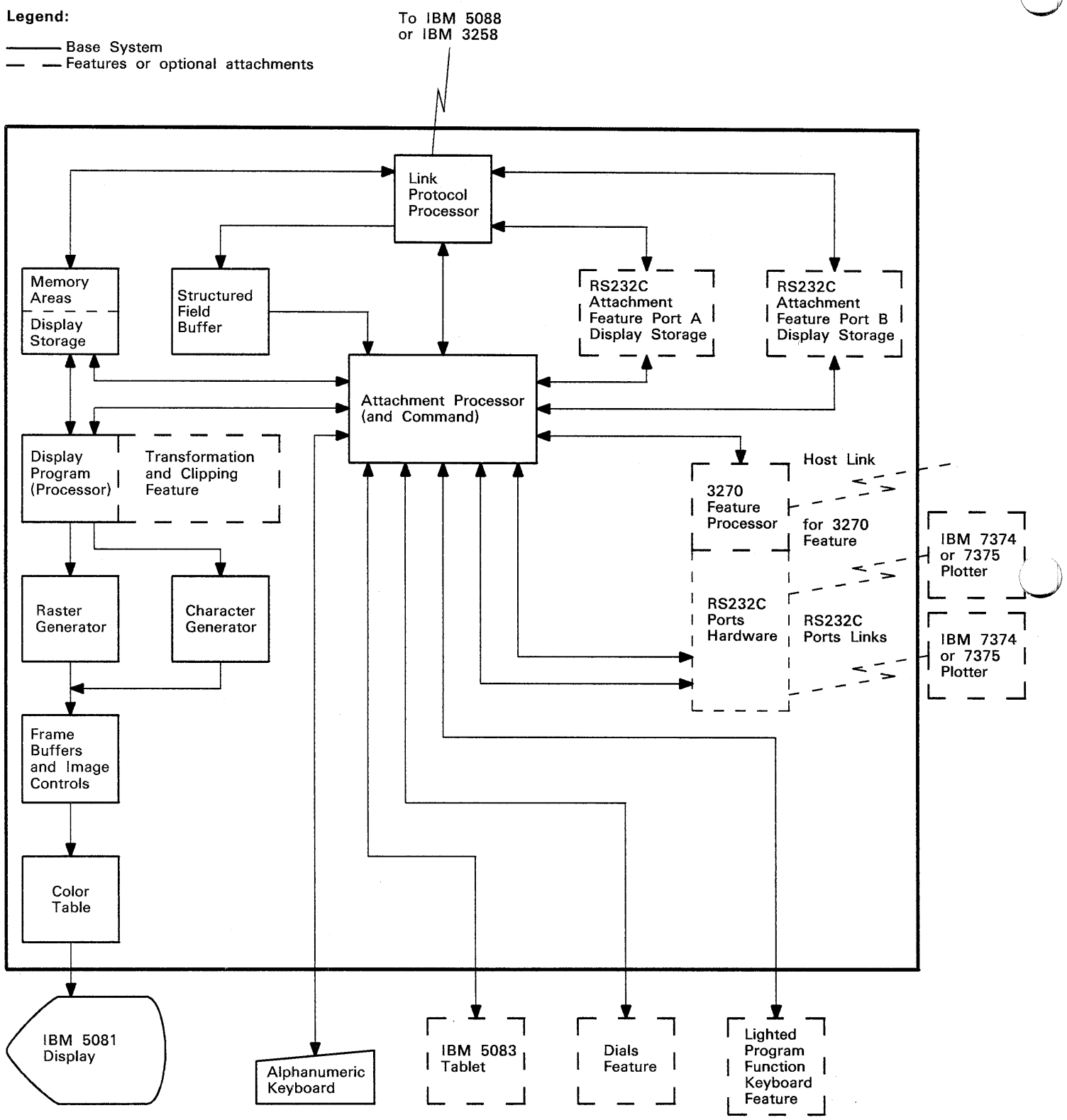


Figure 1-2. IBM 5085/5081 System Architecture and Component Overview

System Memory

The base graphics processor provides 128K bytes of system memory, including 64K bytes for the processor control program. Of the remaining 64K bytes, an additional 8K bytes is required for such functions as area fill workspace, temporary storage of color palettes, and future expansion. This leaves 56K bytes for display storage and user-defined memory areas containing programmable character sets, larger area fill work areas, and so on. This memory is contiguous and can be shared by all of the 5085's processors, as can all expansion memory. System memory is not associated with any specific function.

Expansion to 1.1 megabytes is provided in increments of 512K bytes.

Serial Interface Processor

The serial interface processor is a separate microprocessor that handles data transfers between the 5085 and the 5088 on a high-speed serial link.

High-Speed Serial Link: One-million- or 2-million-bits-per-second rate for processor-to-channel-control-unit (3258 or 5088, respectively) communications via SDLC protocols. Attachment can be up to 5000 meters from the channel control unit, provided special low-loss coaxial cable is used.

Attachment Processor

The attachment processor is a microprocessor that interfaces with the serial interface processor and, jointly with the serial interface processor, controls communications with the host (CPU and application). The attachment processor also controls the keyboards and other attached input/output hardware, manages the display storage, and interfaces to the display processor and 3270 feature.

Display Processor

The display processor consists of a 16-bit-slice microprocessor, working storage, control storage, and double-frame buffers (pixel memory). The double-frame buffers permit instantaneous display image replacement that produces smooth picture transition without blanking or flashing. The display processor executes the display program, which drives a selected list of converters, such as:

- Order decoder
- Vector-to-raster converter
- General area fill process
- Character generator
- Programmable vector character generator
- Transformations that use a high-speed multiplier
- Circle generator

Diskette

A floppy diskette drive is provided for Initial Program Load (IPL), error recording, and problem determination dumps. Configuration setup input is retained on the system diskette for use in subsequent IPLs. Because the system diskette is used for online error recording, it must remain mounted during normal operation of the graphics system.

Audible Alarm

An audible alarm with volume control is provided so the host system can alert the workstation operator in both graphics and 3270 modes of operation.

5088 Graphics Channel Controller

The 5088 Graphics Channel Controller is plug-compatible with the 3258 and provides functional enhancements beyond those of the 3258. The controller attaches to a standard IBM System/370 block multiplexer, selector or byte multiplexer (burst mode) channel interface, and IBM System/43XX/30XX channel interfaces. It presents the appearance of a 3258 to the host system. It sustains a burst mode data rate of at least 1 million bytes per second when the rate is not limited by the channel transfer rate. The controller can also operate with a channel at data rates up to 2.5 megabytes per second in data streaming mode.

The controller supports the attachment of four high-speed serial links. A maximum of 16 processors may be spread across these links and attached to the controller in any combination (for example, all 16 processors on one link or 4 processors on each link). This is possible due to daisy chaining techniques through passive connections employed in the processor. The total length of these links can be up to 5000 meters.

The controller is offered in two models. Model 001 can attach up to 16 processors or 3251/3255 stations, or a mix of each not to exceed 16. Model 002 can attach up to 32 processors or 3251/3255 stations, or a mix of each not to exceed 32.

The presence of a 3255 Display Control Unit on a 5088 Model 001 reduces the data rate between the controller and all attached 3255s or 5085s to 1 megabit per second. Because the 5088 Model 002 is, conceptually, two 16-port Model 001s, the presence of a 3255 on one 16-port side reduces the data rate between the controller and all attached 3255s or 5085s, on that 16-port side only, to 1 megabit per second. If the other 16-port side has only 5085s attached, it will operate at 2 megabits per second.

The presence of a 3270 and RS232C Attachment Feature on any processor attached to a controller or 3258 decreases the number of 5085s or 3251/3255s attachable to that 5088 Graphics Channel Controller or 3258 Control Unit by one for each of the two possible RS232C ports present and active. (See "Addressing Devices Attached to the IBM 5085" on page 2-1.)

Input/Output Devices

Alphanumeric Keyboard (ANK)

This required feature is a low-profile, cable-attached EBCDIC typewriterlike keyboard that can be adjusted easily to any of three surface inclinations matching the preferred hand position of the user. It provides keys for all characters of the supported character set, keys for raising I/O interruptions to the host system, cursor control keys, and other special function keys.

The keyboard has 104 or 106 (for Katakana) keys that may be operated with or without audible feedback. The key arrangement is in four clusters: Three are side by side (left to right); the fourth is located at the rear center. The clusters are:

- 10 keys (2 x 5) for special functions.
- 59 (61 for Katakana) keys for graphics, engineering graphics, APL graphics, shifts, and certain controls.
- 11 keys for cursor control and special functions such as shifting between graphics mode and 3270 mode.
- 24 program function (PF) keys arranged in two horizontal rows of 12 each. In graphics mode, these PF keys can be used in place of the first 24 lighted program function keyboard (LPFK) keys if lighted keytops are not required. They can also be used as a unique set of 24 program function keys by any graphics application. Normal PF key functions are supported in 3270 mode.

APL Keyboard

The APL keyboard has modified keytops to allow the entry of 81 specific APL characters. Exit from and entry to APL mode is controlled through a special APL ON/OFF key. APL mode is recognized only when the graphics system is in 3270 mode. In other respects the APL keyboard resembles the alphanumeric keyboard. Audible indication of character input to the 5080 system is provided.

Note: Only 12 PF keys operate in APL mode.

Lighted Program Function Keyboard (LPFK)

The lighted program function keyboard (LPFK) is a separate assembly with 32 keytops. An amber-colored light-emitting diode (LED) is embedded in each keytop. The profile of the housing and the tilt adjustment are designed to match those of the alphanumeric keyboard.

The amber LEDs are under program control. They may be turned on or off to indicate which keys may be selected at a given moment. A unique signal is returned to the application program when a key is pressed. The keyboard functions operate compatibly with 3250 programs.

The LPFK is designed to accept overlays. This feature permits users to prepare and retain application-specific overlays.

5083 Tablet

The IBM 5083 Tablet may also be attached. The tablet is a thin, flat-surfaced compact unit that offers a comfortable, easy way for users to interact with the 5081 Display. It has a palm rest and a tilt height adjustment for the user's convenience. User interaction with the graphics processor from the tablet is through either the stylus or the cursor special feature. The X-Y coordinate position of the stylus or the cursor is transmitted to the 5085 Graphics Processor as either of these units is moved within the boundaries of the tablet's active surface area.

The cursor is a convenient handheld "mouse-shaped" unit. Four buttons are provided for application use. Precise alignment and accurate digitizing of source data is facilitated through the use of a fine crosshair.

The stylus is a penlike device for user interaction with the display. When the user presses the stylus against the surface of the tablet, a switch in the tip of the stylus causes an impulse to be sent to the application program. Tactile feedback from the stylus indicates to the user that the switch has been closed.

Either the stylus or the cursor may be used as a system pick device.

Only one of the two features may be connected to the tablet at one time.

Dials Feature

The dials feature consists of eight cone-shaped dials placed on a flat, compact, low-profile, desktop unit. The dials can be turned easily by finger pressure along the edge, or by two or more fingers, as the user prefers. The dials are continuous-turn types. The direction and the extent of dial rotation is transmitted to the processor, where it is converted to a value relative to the value that existed when the dial was last reset for interpretation by the application. Translation, scaling, and rotation of two- and three-dimensional images are typical applications of the dials feature.

3270 and RS232C Attachment Feature

The RS232C attachment portion of this feature provides a general-purpose RS232C interface capable of supporting asynchronous protocols. It has two attachment ports for plotters or other devices. The host is responsible for the RS232C protocol processing and formatting. The 5085 provides the attachment buffering and a set of orders that allows the user to perform the protocol processing and formatting. The IBM 7374 or 7375 Color Plotter or equivalent pen plotter is recommended for the range of medium to high pen-plotting requirements. (See "3270 Mode Feature," which follows, for an explanation of the 3270 portion of this "combined" feature.)

3270 Mode Feature

The 5085 Graphics Processor provides a 3270 capability. The 3270 portion of the 3270 and RS232C Attachment Feature permits a 5080 system user to access 3270 applications in host systems that support an SNA/SDLC attachment. A separate microprocessor is also provided with this feature to permit 3270 processing in parallel with all other processor operations; thus, separate and simultaneous interactions with 3270 and graphics applications can be maintained. Although 3270 and graphics images cannot be displayed concurrently, the integrity of the display images for both applications is preserved by the 5085. The operator may alternate between graphics and 3270 images on the 5081 Display by pressing the Jump Screen (Jmp Scr) key on the alphanumeric keyboard.

With this feature, a single workstation can support the data and the graphics computing requirements of engineering and other technical disciplines.

The following 3270 functions are supported:

- Base 3270 functions.
- 14 colors (on 5081 Model 002-color display) using regular and high intensity for each of seven specifiable colors plus black.
- Screen sizes (80 characters per line):
  - 12 lines (960 characters)
  - 24 lines (1920 characters) – the default
  - 32 lines (2560 characters)
  - 43 lines (3440 characters)
- Multiple partitions
- Extended highlighting:
  - Blinking
  - Reverse video
  - Underscore
- Field validation
- Audible alarm
- Alphanumeric keyboard with 24 program function keys
- APL using the APL keyboard and 12 of the 24 program function keys.
- Teleprocessing communications using:
  - SNA and SDLC protocols
  - External, self-clocking modems only, at speeds up to 96 baud

To achieve color support (seven colors each with two levels of intensity), at least one expansion pixel memory feature must be added to the 5085 when a color 5081 Display is the interactive display.

Graphics Display

The image displayed at the 5081 Display may contain a mixture of markers, lines (vectors), fixed alphanumeric and programmed characters, circles, pixel arrays, and area fill primitives.

Each displayable primitive results from the display processor in the 5085 processing orders and data in a user-defined display program.

The display program may be written to regard groups of primitives as segments. Each primitive or segment may be assigned different attributes (for example, blink pattern, color or grayshade, line type, character set identifier, and so on). The programmer may also set different pick modes for primitives or segments to enable or inhibit use of the pick device on the primitives or segments, and to determine the response of the program when the operator uses the pick device to select a primitive or segment.

The 5080 system correctly maintains the coordinates of image primitives in a 64K (± 32K) virtual world coordinate space. In the base system, only a 4K x 4K virtual image space (for example, 4K x 4K virtual pixels) can be displayed. This is called the visible area. The image primitives created by the user with the base system may be specified in greater than 4K coordinates; however, only that portion within the 4K x 4K visible area can be displayed. The 5080 system maps the visible area to the 1K x 1K display screen.

If the *Transformation and Clipping Feature (TCF)* is installed and active, the programmer can create two-dimensional (2D) and three-dimensional (3D) images in the ± 32K (X, Y, or Z direction) virtual world coordinate space. The programmer is provided with orders to automatically position that portion of the image to be displayed into the 4K x 4K visible area.

The image is rastered into one of two frame buffers of the same size as the display screen before it is actually displayed. The current image visible on the screen is generated out of one frame buffer while the next image is being rastered into the other. When the next image is ready to be displayed, the frame buffers are "swapped" and their roles are exchanged.

Each frame buffer can contain from two to eight bit planes, allowing 4 to 256 colors or grayshades to be displayed simultaneously in the same image. The bit planes contain the number (actually, the color or grayshade assigned to the image primitive or segment) of the Color Table entry to be used when displaying the image on the screen.

Display Program

A display program is executed by the display processor of the 5085 to generate the image on the display and to allow the operator at the 5080 Graphics System workstation to interact with the host system and the display program itself using the various devices at the workstation (tablet, keyboards, dials feature).

Display programs consist of orders interleaved with data. An application in the host system generates the display program. The application program, using I/O channel commands and structured fields, loads the display program into the processor display storage and begins execution of the display program.

Base Graphics Processor Display Program Orders

The base 5080 Graphics System provides display program orders to allow the user programmer to perform the following functions:

- Frame buffer/plane control
- Output primitive generation
- Circle generation
- Segment definition
- Attribute control
- Stack definition and control
- Positional device control
- Pick control
- Branch control
- Arithmetic operations
- Data movement
- Polygon area fill
- Definition and use of programmable character sets
- Configuration data storing
- RS232C port input/output control
- 3250-compatible orders

Transformation and Clipping Feature (TCF)

Additional advanced programming functions are available with the 5080 Graphics System and require the *Transformation and Clipping Feature (TCF)* in the 5085 Graphics Processor. This feature includes the following advanced graphic order capabilities in addition to those provided as standard (area fill and circle generation) in the graphics processor to perform 2D and 3D transformation and clipping (that is, translation, scaling, and rotation within a ± 32K world coordinate space).

- Transformation and viewing
- Advanced arithmetic
- Additional stack control

Host Software Support

5080 Graphics System Software Support

Programming Support

All functions of the IBM 5080 Graphics System, including a subset that provides compatibility with the IBM 3250 Graphics Display System, are supported by Release 2 of the IBM Graphics Access Method/System Product (GAM/SP). Under GAM/SP Release 1 with VM/SP, or the Graphics Access Method (GAM) subcomponent of the Graphics Programming Services component in MVS, MVS/XA, or VS/1, support of the 5080 Graphics System is limited to the 3250 subset.

The following table shows the support elements and the system control programs under which they operate.

Element	VM/SP	MVS	MVS/XA	VS/1
GAM/GPS		X	X	X
GAM/SP Release 1	X			
GAM/SP Release 2	X	X	X	

Application Support

IBM CAD/CAM application support of the 5080 Graphics System is available in the CADAM, CATIA, CAEDS, and CBDS2 application products.

- CADAM (a registered trademark of CADAM, Inc.) stands for Computer-Graphics-Augmented Design and Manufacturing.
- CATIA (a registered trademark of Dassault Systems) stands for Computer-Graphics-Aided Three-dimensional Interactive Application.
- CAEDS (a registered trademark of Structural Dynamics Research Corporation) stands for Computer-Aided Engineering Design System.
- CBDS2 stands for Circuit Board Design System 2.

3270 Feature Software Support

The 3270 feature can operate with host systems that support the SNA/SDLC 3270 attachment and applications that use the supported features. IBM-provided programs (for example, TSO, CICS, and IMS) will also operate with the 5080 Graphics System 3270 feature.

Migration/Conversion

3250 Compatibility

Because the 5080 Graphics System uses a raster display to emulate a directed beam display (in the case of the 3250), and due to certain other characteristics of the 5080, some differences with the 3250 Graphics Display System do exist. These are discussed extensively in Appendix A.

3258 Compatibility

The 5088 Graphics Channel Controller provides full 3258 compatibility as a subset. Refer to Appendix A for details.

3270 Feature

5080 Graphics System operation in 3270 mode requires the 3270 and RS232C Attachment Feature on the 5085 Graphics Processor. To achieve color support (seven colors each with two levels of intensity), at least one expansion pixel memory increment should be added to a processor driving a 5081 color display.

Existing 3270 programs can operate with the 5080 system using host systems that support the SNA/SDLC 3270 attachment, and applications that use the supported features. IBM-provided programs (TSO, CICS, IMS, and so on) will also operate with the 5080 system when it is equipped with the 3270 and RS232C Attachment Feature.

The 3270 family of displays supports a broad range of functions and attachments. The 5080 Graphics System 3270 feature, like other members of the 3270 family, does not support the full range. A few 3270 functions and attachments are not supported (see Appendix D for details on the 3270 feature).

Reliability, Availability, and Serviceability (RAS) Highlights

The maintenance philosophy for the 5080 Graphics System provides maximum system reliability and availability to the customer. Online diagnostics (operating with the host system) are provided in the 5085 Graphics Processor. Individual processors may be tested without affecting other units attached to the same channel control unit. In addition, the 5088 Graphics Channel Controller may be tested offline without affecting other channel control units. The 5080 Graphics System is designed for customer setup (CSU) of the processor and the display. (The controller requires installation by a field engineer.)

For details, refer to *IBM 5080 Graphics System: System Planning and Installation,* GA23-0135; *IBM 5080 Graphics System: Setup Instructions,* GA23-0130; *IBM 5080 Graphics System: Operation and Problem Determination,* GA23-0133; and *IBM 5080 Graphics Systems: System Problem Determination,* GA23-0132.

Customer Setup

Certain system functions and parameters that have an effect on the display program and the operator's interactive capability may be specified at system installation time, or subsequently at IPL time, or even online, depending on the function or parameter. The areas of the processor that are dependent on setup are mentioned in this document where discussion warrants it. Most of these functions and parameters have system-provided defaults, whereas others must be specifed. Refer to *IBM 5080 Graphics System: System Planning and Installation,* GA23-0135; *IBM 5080 Graphics System: Operation and Problem Determination,* GA23-0133; and *IBM 5080 Graphics System: Setup Instructions,* GA23-0130 for details on the setup function.

Mode Switching

The preceding sections in this chapter indicate that there are three modes among which the operator may switch the processor. Switching from one mode to another allows the operator to perceive the graphics system workstation as being a 5080 system graphics device, a 3270 type device, or a setup input device that allows the specification and/or modification of various 5080 system parameters.

When the processor is operating in a particular mode, that mode is said to be "active." The "inactive" modes continue to function in the background (with the exception of setup) such that when any mode again becomes "active," the operator correctly perceives the latest state of that mode.

To change from one mode to another involves making the current mode inactive and the desired (or target) mode active. A mode change may be initiated in either of two ways: by the operator as a result of pressing the Jump Screen or SetUp key, or by the setup function initiating a restore mode sequence as a result of the operator terminating setup mode.

Pressing the Jump Screen key in either graphics mode or 3270 mode invokes the other mode (3270 or graphics). Pressing the SetUp key when in either graphics mode or 3270 mode invokes the setup function. When the setup function is terminated, the mode that was active when the SetUp key was pressed is restored.

Note: If a static image is being displayed in graphics mode (that is, the frame buffer nonswitch option is selected or if a portion of the image is protected at buffer regeneration time), the image will not reappear when the graphics mode is restored. (See "Graphics/3270/Setup Mode-Switching Considerations" on page 3-45.)

Chapter 2. Host Interface to the IBM 5080 Graphics System

The 5085 Graphics Processor is a special-purpose processing unit that, in addition to providing interactive graphics capabilities for its host systems, provides functions and advanced graphics capabilities to allow significant offloading of a host processor.

The IBM 5080 Graphics System provides the required capabilities through the use of a command set that is composed of both IBM System/370–43XX–30XX *channel command words (CCWs)* and *structured fields* within a structured field buffer that is the object of a Write Structured command. (See "Write Structured" on page 4-12 and "Structured Fields" on page 4-13.) The command set is rich in the sense that there are many alternatives open to the host application, and different commands and formats may be selected, depending on the specific requirements.

There are circumstances in which display programs running in the 5085 can cause conditions that require host program intervention. The graphic orders, how they are used, and what they can do for your application are described in "Display Programs" on page 3-3. Refer to it for a more complete understanding of display programs and how they operate.

Hardware Considerations

The 5085 Graphics Processor does not execute IBM System/370–43XX–30XX CCWs directly; it accepts, over a serial link, command frames sent by the *channel control unit* (that is, the 5088 Graphics Channel Controller or 3258 Control Unit) to which the processor is attached. In general, there is a one-to-one relationship between the IBM System/370–43XX–30XX CCWs accepted by the control unit and the command frames sent to the graphics processor. Since the 3258 accepts only a subset of the commands processed by the 5088 Graphics Channel Controller, and the 3255 accepts only a subset of the command frames processed by the 5085 Graphics Processor, compatibility with all possible hardware configurations may be a concern that the host programmer must address. (See Appendix A for details on 3250 compatibility and "5085 Attachment to a 3258" on page 3-46.)

Addressing Devices Attached to the IBM 5085

The IBM 5088 Model 1 supports up to 16 IBM System/370–43XX–30XX I/O device addresses. These are translated into link addresses and used, via the serial link, to select 5085s and/or 3255s. An IBM 5088 Model 2 is logically two 5088 Model 1 controllers. Each logical controller has its own set of IBM System/370–43XX–30XX device addresses and its own link to select its independent group of 5085s and/or 3255s.

Each 5085, depending on the installed features, may respond to one, two, or three link addresses generated from the IBM System/370–43XX–30XX I/O device addresses. The link addresses, and therefore IBM System/370–43XX–30XX I/O device addresses, assigned to a single 5085 need not be sequential; they are

independently established each time the 5085 is IPLed. The following can have device addresses assigned to them:

- The graphics device—that is, a 5081 Display.
- RS232C Port A, if the 3270 and RS232C Attachment Feature is installed.
- RS232C Port B, if the 3270 and RS232C Attachment Feature is installed.

Each 3255 attached via the serial link also responds to one link (and therefore device) address. It responds to more than one if certain features are attached.

The 5085 supports concurrent operations for the devices attached to it. Devices physically attached to the same 5085 are logically separate entities and may be thought of as having a unique and dedicated graphics processor at their disposal.

Useful Background Reading

The information contained in the following documents is pertinent to the discussions in this chapter. Take the time to become familiar with them in order to gain a knowledge of the graphics system environment.

- *IBM System/370 Principles of Operation,* GA22-2700
- *IBM 3250 Graphics Display System Component Description,* GA33-3037
- *OEMI Channel-to-Control-Unit Interface,* GA22-6974
- *OEMI Power Sequencing,* GA22-6906

Appendix A of this document also contains useful 3250 reference material that describes the 3250 commands and orders and how they function in the 5080 environment.

Graphics Channel Controller Commands

Listed below in alphabetical order are the IBM System/370–43XX–30XX channel commands and structured fields processed by the 5085 Graphics Processor and accepted by the 5088 Graphics Channel Controller. Along with each is a description of a routine task performed using the command or structured field.

- *No-Operation*—This command is used to show that the path to the 5088 or 3258 controller is operational.
- *Read Manual Input*—This command is used to retrieve the contents of the manual input register, which may have been set by an end user interacting with the 3251 or 5081 Display keyboard or program function keys.
- *Read Memory Area*—This command is used in conjunction with the Select Read Memory Area command to specify the starting address of an input area, and its length, in IBM System/370–43XX–30XX memory into which data returned from the graphics processor is to be placed.
- *Select Read Memory Area*—This command can be used to specify the starting address and length of data in graphics processor memory to be returned to the host.

- *Select Write Memory Area*—This command can be used to specify the starting address and length of data or display programs to be written to the graphics processor.

- *Sense*—This command is used to retrieve device-specific sense data, for example, the current X-Y-Z position. It is normally issued as a response to an interrupt from the device.

- *Sense ID*—This command is used to retrieve the control unit and device model numbers of the addressed device.

- *Set Mode*—This command is used to transmit modal information to the addressed graphics processor; for example, to set the number of sense bytes to be retrieved from the addressed processor when a subsequent Sense command is issued to a device attached to the graphics processor.

- *Write Memory Area*—This command is used in conjunction with the Select Write Memory Area command to specify the starting address and length in IBM System/370–43XX–30XX memory of the data or display programs to be written to the graphics processor.

- *Write Structured*—This command is used to specify the starting address and length of a buffer that contains one or more structured fields to be processed by the graphics processor. A structured field is a contiguous block of data defined by a length, command code, and any parameters associated with the command. The structured fields processed by the 5085 Graphics Processor are:

  - *Set Display Storage Address Register*—This structured field can be used to set the display storage address register for later use when starting a display program in graphics processor memory.

  - *Start Display Program*—This structured field starts a display program at the address contained in the display storage address register.

  - *Stop Display Program*—This structured field stops a display program if one is running in the addressed device.

  - *Sound Alarm*—This structured field is used as an auditory feedback mechanism to the 5085 user. It activates, for a short period, a single-stroke audible alarm at the selected 5080 Graphics System workstation.

  - *Set All Indicators*—This structured field is used as a visual feedback mechanism to the user of a *lighted program function keyboard (LPFK).* Its function is to light and extinguish the lamps under the LPFKs according to a host application's specification. This form is convenient for some applications.

  - *Set Selected Indicators On*—This structured field is used as a visual feedback mechanism to an LPFK user. It allows a host application to set on only the selected indicators without affecting the settings of other indicators.

- *Set Selected Indicators Off*—This structured field is used as a visual feedback mechanism to an LPFK user. It resets the selected indicators only.

- *Set Cursor Address Register*—This structured field is used to allow end-user input from the alphanumeric keyboard and to indicate from which field it is desired.

- *Reset Cursor Address Register*—This structured field is used to disable end-user input from a processor keyboard.

- *Load Blinking Patterns*—This structured field is used to to allow a host application programmer to add his own blinking patterns to a table of display program selectable blinking patterns.

- *Load Line Patterns*—This structured field is used to allow a host application programmer to add his own line patterns to a table of display program selectable line patterns.

- *Define Memory Area*—This structured field can be used to prepare for the loading of programmable character sets. It assigns graphics processor memory partitions to special-purpose functions. (See "Memory Areas" on page 2-6 for more information on the concept of memory areas.)

- *Delete Memory Area*—This structured field is used to free graphics processor memory partitions for later use in display storage or for other special-purpose functions. (See "Memory Areas" on page 2-6 for more information on the concept of memory areas.)

- *Rename Memory Area*—This structured field can be used to reassign graphics processor memory partitions to functions other than the one originally designated. (See "Memory Areas" on page 2-6 for more information on the concept of memory areas.)

Details on the parameters associated with these channel commands and structured fields are given in Chapter 4.

In addition to the commands in the preceding list, the standard Transfer in Channel command (TIC), which is processed by the IBM System/370–43XX–30XX channel, may appear in channel programs directed at devices attached to the IBM 5085 Graphics Processor. Details on the parameters associated with these channel commands are given in "Graphics Channel Controller Channel Commands" on page 4-3.

Using Controller Commands to Perform Operations with RS232C Devices

Since some of the commands in the foregoing list are not appropriate for an RS232C-attached device, they may perform no operation (NO-OP) when addressed to a device attached to an RS232C Attachment Feature. See Chapter 6 for details associated with the limitations of the RS232C attached devices and the differences in display programs written for them.

Returned Status Conditions

The following may cause an input/output interrupt to occur and a *channel status word (CSW)* to be stored:

- Termination of a channel program.
- A change in device state—for example, from not ready to ready.
- Display program execution.
- End-user manual interaction.
- A hardware problem.

The unit and channel status bytes (bits 32–47) of the CSW should be interrogated to see if additional action is needed. Specific actions may be required, depending on the type of interrupt.

Interrupt conditions may be placed in either of two general classes:

1. *Ending status:* received at the termination of the channel program.
2. *Asynchronous status:* an unsolicited interrupt that signals a need for attention to the device.

The 5085 Graphics Processor provides both types of status in a normal application environment.

Ending Status

After an IBM System/370–43XX–30XX channel program is directed at a 5085-attached device and the Start I/O instruction is successful, at least one interrupt may be expected. Depending on the commands in the channel program, different ending status conditions may result. If the ending status presented is Channel End with Device End, and no other status bits are set, the channel program completed successfully. If the ending status is anything other than Channel End with Device End, it is necessary to initiate a Sense operation and, from the received sense data, determine the recovery action required.

Asynchronous Status

The following conditions arising in the graphics processor cause an Attention interrupt to be presented to the host processor:

- The graphics processor is ready—that is, it has come "online."
- An ENTER, Cancel (Cncl), or program function key (LPFK or ANK) is pressed by an end user.
- A pick device detection occurred or a Store Device Input (GSDEVI) order requesting an interrupt is interpreted in a running display program.
- An End Order Processing (GEOP) order is interpreted in a running display program.

- A display program error is detected.
- A display program loop causing a 30-second timeout is detected.
- An uncorrectable hardware error is detected.

Depending on the status and sense data received, any of a number of actions could be initiated. For example:

- In response to a 5085 Graphics Processor device that has become ready, use the Set Mode command to allow the full complement of sense to be retrieved (24 bytes).
- In response to an ENTER, Cancel, or program function key being pressed (status is Attention with no Unit Check), use the Read Manual Input command to retrieve the identity of the pressed key.
- In response to a pick device detection (status is Attention with Unit Check; Pick Detect sense bit is set) or GSDEVI interpretation (status is Attention with Unit Check; Pick Detect and GSDEVI sense bits are set), use the Sense command to retrieve the X-Y-Z coordinates of the pick.
- In response to a GEOP order interpretation (status is Attention with Unit Check; End Order Sequence sense bit is set), the action is application-dependent.
- In response to a display program error (status is Attention with Unit Check; Display Program Error sense bit is set), the action is application-dependent.
- In response to a display program loop (status is Attention with Unit Check; Buffer Program Loop sense bit is set), the action is application-dependent.
- In response to an uncorrectable hardware error (status is Attention with Unit Check; Hardware Error sense bit is set), log the error and terminate processing.

Memory Areas

Graphics processor storage may be dynamically partitioned into memory areas that can be assigned to special-purpose functions. For the graphics display, as memory areas are defined by the user, the storage associated with them is taken from the device's display storage. The storage assigned to a memory area may be returned to display storage when the function that required the storage is no longer in use. Commands defined in "Graphics Channel Controller Commands" on page 2-2 allow application programs to manipulate the memory areas conveniently.

Functions That Require Memory Areas

A memory area *must* be defined in order to use:

- Area fill graphic orders
- Programmable character sets

The area fill processor requires a work area for its intermediate results. Only one work area is required and may be defined for each graphics processor. Depending

on the complexity of the figure being filled, more or less storage should be assigned to the work area. See "Area Fill Orders" on page 5-80 for details on how to use the area fill orders.

When a display program requires programmable character sets, the sets must be preloaded into a defined memory area. Data written into memory areas tagged with a programmable character set type require special processing that allows the data to be used as programmable character sets during the execution of the display program.

The Define Memory Area, Rename Memory Area, and Delete Memory Area structured fields are used to assign an area, reassign an old area to a new use, and remove an area, respectively. The formats of these structured fields and their parameters are detailed in "Memory Area Control Structured Fields" on page 4-14.

System-Defined Memory Areas

For the purpose of convenience, some memory areas are defined at IPL time by the graphics processor initialization process. For each logical device present (that is, graphics device, RS232C port A, and RS232C port B), a memory area is identified and associated with the display storage for that device. A memory area is also defined, again for each logical device, that contains graphics processor control blocks describing the memory areas allocated to the device. The aggregate of these control blocks is referred to as the *Memory Area Control Table* for that device.

As a convenience to the user, the initial process defines a memory area, associated with the graphics display, which is an area fill work area. The initial area fill work area may be deleted and redefined as required. The keys to accessing a memory area are the device address and the memory area identifier, which is a 16-bit number. The identifier of a memory area containing display storage for each device is X'FFFE'. The identifier of a memory area containing a Memory Area Control Table is X'FFFF'. The identifier of the memory area defined by the initialization process as an area fill work area is X'FFFD'.

User-Defined Memory Areas

If your application requires a large area fill work area or the use of programmable character sets, you will have to define a memory area for that use. The following should be determined before any code is written to define the memory area:

- The size of the area required, expressed in units that are multiples of 2048 bytes, because memory areas are allocated in blocks that are 2048 bytes long.
- The type of the area required, whether it is an area fill work area or a programmable character set area.
- The identifier to be assigned to the area. This identifier must be unique in the environment in which it will be used. It cannot be X'FFFF', which is assigned to the Memory Area Control Table, or X'FFFE', which is assigned to the device's display storage, or X'FFFD', which is assigned to initial area fill work area if it exists, or X'0000', which is reserved, or any other identifier that may be currently defined. To determine the identifiers currently in use, read and analyze the Memory Area Control Table.

Use the detailed information on the Define Memory Area structured field in "Memory Area Control Structured Fields" on page 4-14 to generate and issue the structured field. If the parameters are appropriate and enough memory exists in the display storage area to form the requested memory area, the memory area will be defined.

Algorithm Used in the Assignment of Memory Areas

When the system defines memory area X'FFFD' (area fill work area) or a user defines a memory area, the storage allocated to them is taken from what was the display storage area. Since the define/delete memory area process dynamically changes the amount of display storage available to the user, it is important to understand the algorithm used in allocating storage to and reclaiming storage from defined memory areas.

The graphics processor memory area manager uses display storage as if it were a free space. In the simple case, memory partitions are taken from the top of the free space and returned to the top of the free space when the memory area with which they are associated is deleted.

Because a number of memory areas may be defined, the last one defined, which is adjacent to the free space, is not always the first one to be deleted. What occurs in this case is the phenomenon known as *memory fragmentation* and is a problem common to most memory allocation schemes. The graphics processor memory area manager maintains entries in the Memory Area Control Table describing memory fragments as well as defined memory areas. The storage assigned to all the defined memory areas and fragmentary memory areas is referred to as the *memory area storage pool.* The memory area storage pool can be thought of as the storage, which, at initialization time, belonged to the display storage area for the device but which does not currently.

When a request to define a memory area is received, a *best-fit search* is performed to determine if there is a memory fragment that can satisfy the request. If a best fit is found, the memory area is assigned to the memory fragment. If no fit is found, storage for the memory area is taken from the top of the remaining free space.

When a request to delete a memory area is received, the Memory Area Control Table is scanned to determine if the now fragmentary memory area can be combined with other adjacent memory area fragments and/or combined with the free space, returning it to display storage. The memory area manager attempts to provide optimal storage management, but there may be times when, in a very dynamic environment, storage fragmentation will exhaust its resources. The access method or application may want to provide a method for "garbage" collection. It is possible to do this by:

- Reading the Memory Area Control Table to find the defined memory areas.
- Retrieving the data from defined memory areas.
- Deleting all existing memory areas.

- Defining the required memory areas.
- Writing the required data back into the memory areas.

There are limits enforced in the memory area definition process; see "Memory Area Control Structured Fields" on page 4-14 for the details.

Analysis of the Memory Area Control Table

Associated with each memory area are the following attributes:

- A 16-bit identifier assigned by the application.
- A size, in 2048-byte blocks, assigned by the application.
- An area starting address assigned by the graphics processor.
- A current transfer address assigned by the graphics processor.
- A type, assigned by the application. Types include:
  - Area fill work area
  - Programmable character set area
  - Display storage (this type may not be defined by an application)
- A state code, assigned by the graphics processor, which is used for housekeeping.
- Special processing flags assigned by the graphics processor. Bit 0 indicates read/write or read-only memory status.

For each memory area there is a descriptor in the Memory Area Control Table that contains all of the attributes listed above. The addresses in the memory area descriptor, area starting address, and current transfer address are 5085 24-bit absolute system memory addresses. The address is righthand-adjusted in a 4-byte field and padded to the left with a zero byte. They are used by the graphics processor whenever a memory area is selected for a data transfer by a Select Read Memory Area command or Select Write Memory Area command. They have no effect on any other graphics processor control registers; for example, there is no effect on the display storage address register.

The Memory Area Control Table is itself contained within memory area X'FFFF'. This memory area cannot be defined, renamed, or deleted by an application and is a read-only memory area.

The Memory Area Control Table has a header that describes:

- The total number of 2048-byte blocks of memory assigned to the device's display storage at initialization time.
- The total number of 2048-byte blocks of memory currently assigned to the device's memory area storage pool. With the exception of areas X'FFFF' (the Memory Area Control Table) and X'FFFE' (display storage), all memory areas are in the pool.

- The maximum number of memory area descriptors in the Memory Area Control Table.

For a more detailed layout of the Memory Area Control Table, see "Memory Area Control Table" on page 2-11.

The device's display storage is the first entry to be placed in the Memory Area Control Table. This is done at initialization time by the 5085 Graphics Processor when the extent of available memory is determined. The device's display storage has the identifier X'FFFE'. A display storage area cannot be defined, renamed, or deleted by an application.

Using the Information in the Memory Area Control Table

Most of the information in the Memory Area Control Table is useful for the application/access method when dealing with complicated memory area allocations. Here are some uses that are foreseen for the information:

- In the Memory Area Control Table header:
  - The first field (initial device display storage size) can be used to determine the limits of the largest possible memory area storage pool.
  - The second field (current memory area storage pool size) can be used to determine the current display storage size or the room for expansion in the memory area storage pool.
  - The third field (maximum number of memory area descriptors) is very important for the application/access method to know, because this is a limiting factor on the number of memory areas that can be defined. It can be used to tell how many, if any, additional memory areas can be defined.
- In the memory area descriptors:
  - The first field (memory area identifier) uniquely identifies the memory area for subsequent references. All references to the descriptor are qualified by this value.
  - The second field (memory area size) defines the extent of the memory area or storage fragment.
  - The third field (memory area starting address) is important to note when debugging programmable character sets, because they cannot span a 64K boundary.
  - The fourth field (current transfer address) is useful only during error recovery procedures during the execution of some Select Read Memory Area or Select Write Memory Area channel commands.
  - The fifth field (memory area type) is used when a memory area checkpoint is performed, so that the proper area type may be redefined later.

Type codes are assigned as follows:

X'00'	Reserved
X'01'	Reserved
X'02'	Display storage
X'03'	Programmable character set
X'04'	Area fill work space
X'05'	Reserved for IBM use
X'06' to X'FF'	Reserved

- The sixth field (memory area descriptor state code) is used to indicate if the descriptor:

  1. Is in use or is simply a placeholder in the table. A state code of X'00' indicates a placeholder. The descriptor is not in use and has no memory assigned.

  2. Defines a memory fragment. The state code for this is X'01'.

  3. Defines an active memory area. The state code for this is X'02'.

  This is useful in identifying which descriptors to interrogate when determining how much fragmented memory exists.

- The seventh field (memory area special processing flags) is useful for determining the cause of some program error conditions. The high-order bit (bit 0) is used to indicate read/write status of the area. If the bit is set, it indicates that the area is a read-only memory area.

Memory Area Control Table

Field Definition	Offset	Length
Initial device display storage size in 2048-byte blocks	Memory area X'FFFF'+0	2 bytes
Current device memory pool size in 2048-byte blocks	Memory area X'FFFF'+2	2 bytes
Maximum number (*n*) of memory area descriptors in this Memory Area Control Table (see Note)	Memory Area X'FFFF'+4	2 bytes

Field Definition	Offset	Length
First memory area descriptor in this Memory Area Control Table	Memory area X'FFFF'+6	16 bytes
.		
.		
.		
*n*th memory area descriptor in this Memory Area Control Table	Memory area X'FFFF'+6+16x(n-1)	16 bytes

Note: *n* indicates the number of memory area descriptors in the Memory Area Control Table for this device. For the graphics devices, *n* is 9; for the RS232C port devices, *n* is 1.

Memory Area Descriptors

Field Name	Offset	Length
Memory area ID	Memory area descriptor+0	2 bytes
Memory area size in 2048-byte blocks	Memory area descriptor+2	2 bytes
Memory area starting address in 5085 system memory	Memory area descriptor+4	4 bytes
Memory area current transfer address in 5085 system memory	Memory area descriptor+8	4 bytes
Memory area type	Memory area descriptor+12	1 byte
Memory area descriptor state code	Memory area descriptor+13	1 byte
Memory area special processing flags	Memory area descriptor+14	1 byte
Reserved	Memory area descriptor+15	1 byte

Chapter 3. Graphics Processor Architecture

Display Storage

The IBM 5085 Graphics Processor contains either one or three physical display storage areas, depending on whether the RS232C Attachment Feature is included. Channel I/O commands direct data to the appropriate display storage according to the device address.

- The first, or graphics, display storage is associated with the 5081 Display and consists of a minimum of 32K bytes of random access memory (RAM). This storage is used to hold the display program and data that control activity at the 5080 Graphics System workstation. Although the 5085 comes with a 56K-byte display storage area, the user may reduce the size of this area to 32K bytes through the use of memory area definitions.

- Two additional display storage areas are associated with the RS232C Attachment Feature and consist of 16K bytes of RAM each. These are used to hold programs and data that control the RS232C ports. Refer to Chapter 6 for details.

The display storage area associated with the 5081 Display contains pages of 64K (65,536) bytes. The last page may range from 2K bytes to 64K bytes, but, if only one page exists, it must never be less than 32K bytes. A Branch Page (GBPAGE) order and a Branch after Push Link (GBAPL) order are provided to transfer control between pages. The result of any attempt to cross the display storage page boundary without using these orders is unpredictable. The pages are numbered from zero upward. Programs can reference pages other than zero only by using the Write Structured and memory area channel commands and the GBPAGE and GBAPL orders.

Each pair of display storage bytes, beginning with bytes 0 and 1, forms a display storage word. Graphic orders are aligned on word boundaries, and each order occupies an integral number of words. Graphic and alphanumeric data fields are word-aligned, each field occupying one or more words.

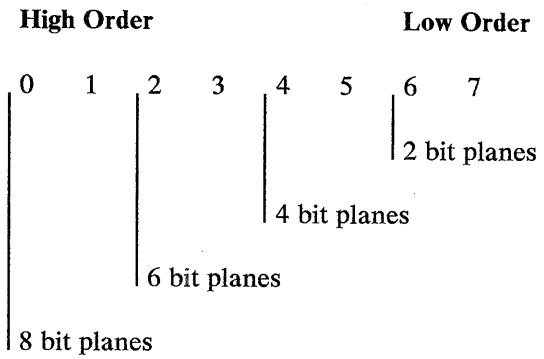
The display storage must contain a display program in order for the graphics system workstation to become active. This program controls the display of graphic and alphanumeric data on the display, as well as the action taken upon detection by the pick device of a line, marker, character, and so on, within the displayed image. The display program also coordinates the generation of interrupts due to keyboard action at the workstation and at other input devices, such as the dials feature.

Frame Buffers

The processor is a raster device that employs frame buffers to hold the image that is ultimately displayed on the display. As the processor interprets the display program in display storage, the image data produced by such functions as line-to-raster conversion, character generation, and so on, is stored into the frame buffer. The stored data is scanned and converted to video signals and passed to the display.

Each frame buffer is 1024 x 1024 pixels (bits) and may be up to eight planes in depth. The processor provides options of 2-, 4-, 6-, and 8-bit planes. The number of bit planes controls the number of colors or grayshades that you may simultaneously display.

The bit planes are numbered as follows:



If a processor has less than eight bit planes, the high-order bits are discarded and the remaining bits are loaded according to the diagram above.

To avoid flashing on the display screen (due to dynamic image updating and graphic monitor refresh memory contention), the processor is equipped with a double buffer. This provides two sets of frame buffers and enables preparation of a new image in one frame buffer while the other frame buffer is used for refreshing the image being displayed. After preparing the new image, the buffers are switched. The buffer having the new image is used to refresh the display image, and the one having the old image is used for preparing the next new image. The frame buffers are switched at vertical retrace time in order to keep the image from breaking up.

The frame buffers are switchable and erasable under display program control. (See "GBGOP—Begin Order Processing" on page 5-8.)

Display Storage Page Numbering

The graphics processor supports a maximum of 17 pages; accordingly, valid page number values are in the range X'0000' to X'0010'. However, the number of pages supported by your processor may be less, depending on the amount of storage installed, and may vary from time to time, depending on whether a memory area storage pool exists and, if it does, how large it is.

You may calculate the number of pages available at any time in your processor, and, by doing so, the valid page number values, by issuing a Store Configuration Data (GSCONF) order and reading back the configuration data stored in display storage. Then, from parameter 2 (graphics system storage size) extract the number of 2K blocks of storage assigned to display storage. Divide this number by 32; the quotient gives you the number of full 64K pages available for display storage use and the remainder, if not zero, indicates that you have an additional (last) page of 2K to 64K. (See "Memory Areas" on page 2-6 for details.)

An attempt to branch to an invalid page results in a display program error.

Contact your system programmer for details on page numbers available to your application.

Display Programs

A display program, which consists of graphic orders interleaved with data, is contained in one or more areas within display storage.

The graphic orders and data that form a display program are assembled by software in the host CPU. The host CPU software also transmits the display program to the display storage and initiates execution of the display program using the commands described in "Graphics Channel Controller Commands" on page 2-2.

Each display program must contain either a Begin Order Processing (GBGOP) or Start Regeneration Timer (GSRT) order. If a display program executes for 30 seconds without encountering either a GBGOP or GSRT order, program execution is terminated and an Attention/Unit Check interrupt is presented to the host system.

Management of display storage is a user function for which host software conventions or controls are required. The display storage can contain orders and data not in the flow of control of currently executing display programs.

After the host has initiated the execution of a display program (with a Set Buffer Address Register and Start command or a Start Display Program structured field), execution proceeds continuously until terminated either by the host, a pick detect interrupt, an End Order Processing (GEOP) order or other similar condition (see "Display Program Termination" on page 3-10).

If the switch buffer mode is invoked by a GBGOP order (with d=0), or a GSRT order (by default), the frame buffers are switched during execution of the GBGOP or GSRT order. The frame just executed (and rastered) is displayed, and the next frame is rastered into the alternate frame buffer. Frame buffer switching is synchronized with video refreshing; the display program processor does not complete execution of the GBGOP or GSRT order until frame buffer switching and clearing are complete. During the clearing operation, only those frame buffer bit planes with the corresponding write protect mask bits set to zero are cleared.

If the nonswitch frame buffer mode is on (GBGOP order, with d=1), the frame buffers are not switched or cleared.

Notes:

1. A GBGOP or GSRT order should begin each display program to ensure the appropriate states are set and that the frame buffer is cleared. However, display programs that do not begin with a GBGOP or GSRT order will execute.

2. A GBGOP or GSRT order must be issued to cause an image to be displayed on the display screen. Execution of a GEOP order does not result in switching the frame buffers. Therefore, a simple program starting with a GBGOP or GSRT order and ending with a GEOP order will not result in the displaying of an image.

3. Due to the speed of the host link, it may be possible to load, start, and stop a display program before image generation and frame buffer switching can take place. Accordingly, the user should use a timer in any host application that would behave in such an unsolicited mode, to ensure that the image is displayed.

Display Program Processor

The following registers and table are maintained by the display program processor to control execution of the display program:

- Display storage address register
- Regeneration address register
- Attribute registers
- Stack registers
- Cursor address register
- X-Y-Z position registers
- Condition code register
- Color Table

Display Storage Address Register (DSAR)

The display storage address register, which contains a page number and an address within that page, controls display storage access during certain channel commands.

The contents of the display storage address register may be changed by:

- Execution of a Write Structured command containing a Stop Display Program or Set Display Storage Address Register structured field issued by the host system.

In the context of 3250 compatibility, the DSAR and buffer address register (BAR) of the 3250 are equivalent. Accordingly, the contents of the DSAR are also changed by the following 3250-compatible commands:

- A Set Buffer Address Register and Stop command issued by the host system. The page number is set to zero.
- A Set Buffer Address Register and Start command issued by the host system (also sets the regeneration address register). The page number is set to zero.
- Execution of a Write Buffer command issued by the host system.
- Execution of a Read Buffer or Read Cursor command issued by the host system.

The display storage address register is set to zero at power-on or system reset, that is, address zero of page zero.

Notes:

1. During Read Buffer, Read Cursor or Write Buffer commands, the display storage address register is incremented to address each word sequentially in the buffer.

2. During Select Write Memory Area, Select Read Memory Area, Write Memory Area, and Read Memory Area channel commands, the display storage address register is not changed. In addition, the memory area control structured fields have no effect on this register.

Regeneration Address Register (RAR)

The regeneration address register, which contains a page number and an address within that page, controls the processing of instructions in the display program.

The contents of the regeneration address register may be changed by:

- Normal display program processing.
- A Write Structured command containing a Start Display Program structured field (at the next GBGOP or GSRT order if the display program is running).
- A Set Buffer Address Register and Start command issued by the host system (the DSAR is copied into the RAR). The page number is set to zero.

The regeneration address register is set to zero at power-on or system reset, that is, address zero on page zero.

Attribute Registers

Attribute registers contain the current values of blink, line type, marker, image plane control, highlight control, color/grayshade attributes, pick controls, and so on. Attribute registers may be changed during execution of a display program to assign different attributes to components of the image.

For further details on these registers, their control and default settings, see "Attribute Control Orders" on page 5-31.

Stack Registers

Stack registers contain the stack size, location, and stack current pointer. Stack registers may be changed during execution of a display program.

For further details on stack registers, their control and default settings, see "Stack Control Orders" on page 5-57.

Cursor Address Register (CAR)

The cursor address register contains the display storage address and page number of the character mode data field to which the cursor is assigned. Entering an alphanumeric character places the character code into the addressed data field and, normally, moves the cursor to the next character location in the data list. For further details, see "GDCHAR—Draw Character" on page 5-24 and "ANK Input and the Cursor" on page 3-12.

The contents of the cursor address register are also updated by the host system with an Insert Cursor or Remove Cursor command or a Set Cursor Address or Reset Cursor Address structured field.

The cursor address register is set to zero at power-on or system reset.

X-Y-Z Position Registers

X-Y-Z position registers show the coordinates of the currently addressed point, or *current draw position (CDP),* in world coordinate space. They are maintained as signed 16-bit pretransformation world coordinate values in the range of -32K to +32K. Negative values are maintained in twos complement form. (See "Coordinate System Specification" on page 3-27.)

The contents of the X-Y-Z position registers may be interrogated by the host system by issuing a Sense command. Bytes 14–19 of the sense data contain the X-Y-Z values of the X-Y-Z position registers. If the Sense command is issued in response to a pick detect (see "Pick Control" on page 3-19), the X-Y-Z values should be interpreted as follows:

- In graphics mode the X-Y-Z values represent the coordinates of the current draw position.
- In character mode, if the character set ID (CSID) in attribute register 18 is X'00', the X-Y-Z values represent the center point of the character following the character upon which the detect occurred.
- In character mode, if the CSID is *not* X'00', the X-Y-Z values represent the lower left corner point of the character following the character upon which the detect occurred.
- If the Sense command is issued and the pick interrupt was the result of a Store Device Input (GSDEVI) order causing byte 1, bit 6 of the sense data to be set, the value of the X-Y-Z coordinates returned represents the current tablet (stylus or cursor) position. The Z value is set to zero.

If the Sense command is not sent in response to a pick detection interrupt, the value of the X-Y-Z (current draw) position registers is returned, but the significance of the value depends on the display program and the point at which it was last stopped.

The Store Device Input (GSDEVI) order may also be used to access the X-Y-Z position registers from a display program.

The contents of the X-Y-Z position registers may also be interrogated (in 3250-compatibility mode) by issuing a Read X-Y Position Register command or a Store X-Y Position Registers (GSXY) order. In this case, the Z value is ignored and the four high-order bits of the X and Y values are set to zero.

The X-Y-Z position registers are set to zero at power-on or system reset.

The Z value is maintained even if the Transformation and Clipping Feature (TCF) is not installed or is installed and 3D mode is not active. If output primitive (graphic) orders are *not* used that specify a Z coordinate, the Z register remains zero. If orders specifying a Z coordinate *are* used, the Z register is updated accordingly. In those cases where the TCF is not installed or 3D mode is not active, the Z coordinate and/or the Z register is ignored during vector-to-raster (line-to-raster) conversion.

Condition Code Register

A 2-bit condition code register is provided to retain certain conditions that may occur while processing display programs. The contents of this register can be tested by the Branch on Condition (GBC) order (see "Branch Control Orders" on page 5-53 for details).

The condition code register is set under the following conditions:

Setting Condition	00	01	10	11
GSRT GBDD GBND GBGSEG SBA Start WS Start	PI off and TSI off	PI off and TSI on	Not set	Not set
GEAF GESEG GLATR GDPXL GDCIR GBGOP	PI off and TSI off	PI off and TSI on	PI on and TSI off	PI on and TSI on
GADD	Sum = 0	Sum < 0	Sum > 0	Overflow
GSUB	Diff. = 0	Diff. < 0	Diff. > 0	Overflow
GCOMP	Equal	1st oper low	1st oper high	Not set
GMUL	16-bit product	32-bit product	Not set	Not set
GDIV GSHIFT	No overflow	Overflow	Not set	Not set
GSDEVI	All switches off for referenced device	All switches on for referenced device	Out of presence	Device not installed
GTM	All 0s	Mixed 0s and 1s	Not set	All 1s

Notes:

1. See "Pick Detection Modes and Indicators" on page 3-20.
2. For a more complete description of the condition code settings, see the individual order descriptions.

3. The RS232C Attachment Feature has separate condition code registers (see "RS232C Port Condition Codes" on page 6-5 for details).

None of the remaining orders affect the condition code setting; it is preserved until changed by one of the above conditions.

Color Table (CT)

The Color Table (CT) controls the grayshade level on 5081 monochromatic models and the color on 5081 color models.

The Color Table is loaded by the Load Color Table (GLCT) order executed by the display processor, and allows flexibility to choose the colors or grayshade levels that are used concurrently from a larger number of potential colors or grayshades.

The Color Table contains 256 locations for color or for monochrome. Each location is 12 bits wide.

In a monochromatic model, all 12 bits control the grayshade level. In a color model, the 4 high-order bits control the intensity of the blue, the next 4 bits control the intensity of the green, and the low-order 4 bits control the intensity of the red.

On a color system there is a choice from a range of 4096 possible colors, up to 256 of which can be active at one time, depending on the number of planes in the frame buffer. A monochrome system supports the simultaneous display of up to 256 grayshades, depending on the number of planes in the frame buffer.

In summary, the Color Table provides a means of translating the output of the bit planes in the frame buffer to actual colors/grayshades on the display.

Figure 3-1 on page 3-9 illustrates the format and use of the Color Table and its relationship to the frame buffer bit planes.

Default Color Table

At system initialization time the Color Table is loaded with default values.

The first eight entries (0–7) simulate 3250-compatible intensity levels, providing varying shades of gray increasing in intensity from entry 1 to entry 6. Entry 0 specifies black and entry 7 specifies white. The entries are:

Entry Number	Color Value	
0	X'000'	
1	X'333'	For 2-bit planes, value is X'BBB'
2	X'666'	For 2-bit planes, value is X'CCC'
3	X'999'	For 2-bit planes, value is X'FFF'
4	X'AAA'	
5	X'BBB'	
6	X'DDD'	
7	X'FFF'	

The remaining entries are loaded with various color/grayshade values to ensure a display if the entry is referenced.

Notes:

1. The same values are loaded whether the 5081 is a color or monochrome model.

2. Color Table entries 3, 15, 63, and 255 are loaded with a value of X'FFF' to generate a white color/grayshade. These entries constitute the logical end of the Color Table for systems with 2-, 4-, 6-, or 8-bit planes.

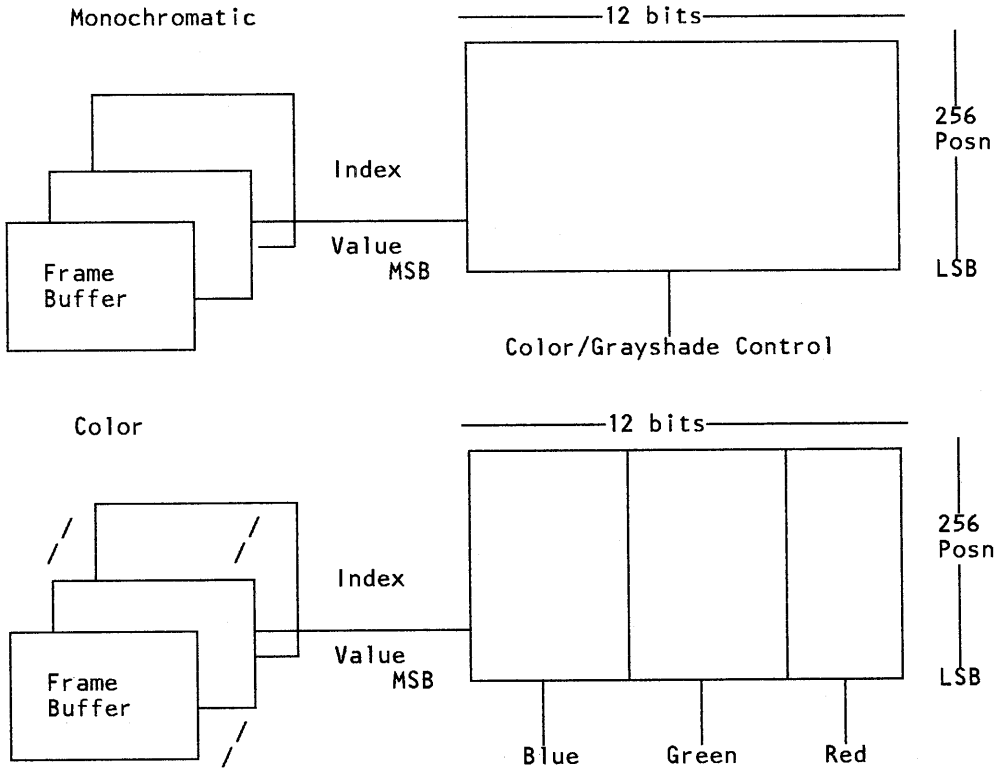


Figure 3-1. Color Table Use and Structure

Display Program Initiation

A display program is started upon:

- Receipt from the host of a Write Structured command containing a Start Display Program structured field (code X'27'). The program is started at the storage address currently set in the display storage address register. If the display program is already running, it continues until it encounters a GBGOP or GSRT order; then it is immediately restarted.

- Receipt of a Set Buffer Address Register and Start command from the host CPU. The program is started at the new display storage address contained in the command. If the display program is already running, it is immediately terminated and restarted.

Display Program Termination

A display program is terminated by any of the following events:

1. The program is asynchronously terminated upon:

   - Receipt from the host CPU of a Write Structured command containing a Stop Display Program structured field (code X'07').

   - Receipt of a Set Buffer Address Register and Stop command from the host CPU.

   - Receipt of a Set Buffer Address Register and Start command from the host CPU. The program is terminated and restarted from the new storage address contained within the command.

     Note: A Write Structured command containing a Start Display Program structured field (code X'27') from the host does not terminate the display program; the program continues to run to the next GBGOP or GSRT order, at which point the program is restarted.

     See "Graphics Channel Controller Commands" on page 2-2 for the associated channel actions.

2. The program is terminated upon execution of:

   - End Order Processing (GEOP) order.

   - Permit Detect Interrupt (GPDI) order if a previous pick detect is currently remembered (pick indicator on).

   - Store Device Input (GSDEVI) order, if the switch on the position entry device is closed and the order has enabled a switch interrupt (GSDEVI order with i=1).

   - A pick control order that sets pick immediate interrupt mode and a previous pick detect is currently remembered (pick indicator on), and not reset by the order, or the pick indicator (PI) is set on by the order.

   - A Branch Page (GBPAGE), Branch after Push Link (GBAPL) or Move Data Block (GMVBLK) order specifying an invalid page number.

- A stack error.
- An uncorrectable memory error.
- A Transformation and Clipping Feature (TCF) error (see "TCF Error Conditions" on page 3-45).
- A GBGOP or GSRT order not encountered for 30 seconds.

3. The program is terminated during interpretation of a data list due to:
   - Following an output primitive order [for example, a draw line or marker order, or a character mode, Draw Pixel Array (GDPXL) order, or Draw Circle (GDCIR) order], if a pick detect occurs on the data to be displayed and the program is in the immediate interrupt state (see "Pick Control" on page 3-19 and "Pick Control Orders" on page 5-50).
   - A TCF error (see "TCF Error Conditions" on page 3-45).
   - An uncorrectable memory error.
   - A GBGOP or GSRT order not encountered for 30 seconds.

As a result of either 2 or 3 above (but not 1), a control-unit-initiated status sequence will be attempted. See Appendix C for definitions of the resulting status and sense.

In all the preceding cases, the status of the display program (attribute and stack register settings, current draw position, pick modes, graphics mode, and so on) is preserved until default values are reset by a subsequent Set Buffer Address Register and Start command or a GBGOP or GSRT order.

Following termination of the display program, character input to the keyboard is ineffective. The *lighted program function keyboard (LPFK)* keys, the *alphanumeric keyboard (ANK)* program function (PF), ENTER, and Cancel (Cncl) keys remain effective; the tablet and the dials feature are rendered ineffective.

Interruption from the Keyboards

In addition to the situations described in the preceding section, the display program allows an interrupt at the host CPU with the pressing of certain keys at the ANK or LPFK (see Appendix C for resulting status and sense). However, in these circumstances the display program is not terminated and execution continues in normal order sequence.

Interruption occurs during execution of a display program if an LPFK key, an ANK PF key, ANK ENTER key, or ANK Cancel key has been pressed since the last GBGOP or GSRT order. Execution of the GBGOP order can also simulate the pressing of an LPFK key, ANK ENTER key, or Cancel key. (See "Frame Control Orders" on page 5-6 for details.)

Pressing any other ANK key does not result in an interrupt at the host CPU. (See "GDCHAR—Draw Character" on page 5-24 for details.)

Alphanumeric Keyboard (ANK)

Character Mode Data Fields

A character mode order (GDCHAR or GECM) is followed by a data list. Each byte of the data list contains an extended binary coded decimal interchange code (EBCDIC) character.

The data list is terminated when a subsequent set mode (SM) control byte (X'28', X'2A', X'2C', or X'2E') is encountered. Because a data list may be terminated on an odd boundary, the set mode control of a subsequent order may have to be preceded by a single set mode control.

The data list is seen as a character mode data field on the screen when a display program is executing.

ANK Input and the Cursor

Using the ANK, the operator can enter characters into that part of display storage occupied by a data list defined by an unprotected character mode order (GDCHAR or GECM) that contains a cursor. The data list is seen by the viewer as a character mode data field on the screen. When received, the character code is inserted into the display storage location to which the cursor is assigned, replacing the code currently in the location. The cursor is automatically assigned to the next sequential location in the data list unless it is currently in the last position of an unprotected field; then, it is not moved. Further character codes entered from the ANK will not replace the last position, in this case, unless the Reset key has been pressed.

The insertion of a cursor does not disturb the data in that location. The cursor is displayed only if it is associated with a character (unprotected), which is itself displayed on the screen, or with a space. A cursor associated with a null, backspace, new-line, or a set mode character is not displayed. The cursor is affected by the blink and color/grayshade attributes that apply to the associated character. If the cursor does not currently satisfy the requirements for data entry, it can be moved to the next unprotected character field using the Tab key on the ANK. Pressing the Cursor Right, Cursor Left, or Tab key at the ANK moves the cursor correspondingly. After the required characters have been entered, the user can cause an interrupt by pressing ENTER. The host CPU program can then retrieve all or part of the entered character string from display storage (via the channel).

Notes:

1. The characters can be entered or displayed only if a display program is executed, subsequent to keying into an unprotected field containing the cursor, thus creating an image in the frame buffer.

2. New-line and backspace (reqd.) are supported only when the CSID is X'00'.

Cursor Left and Cursor Right Keys: The Cursor Left and Cursor Right keys move the cursor on the screen within an unprotected data list (or field). If the data list is defined by a GDCHAR order [that is, the character set identifier (CSID) in attribute register 18 is X'04' or greater], the cursor can move only to the left or right within a single line on the screen because the new-line functions are not supported by the GDCHAR order. If data entry on the next line is required by the display program, it will have defined a new data list beginning at the next line. In this case, the Tab key must be pressed to move to the next line. When the CSID is X'04' or greater and new-line control is encountered in the data list, it performs no function and is displayed as a hyphen.

If the data list is defined by the 3250-compatible GECM order (that is, the CSID = X'00'), movement to new-line codes is controlled by new-line codes contained within the data list.

Backspace (Reqd.) Key: If the Backspace key is pressed when the cursor is positioned in an unprotected data list (field) defined by a GDCHAR order (that is, CSID in attribute register 18 is X'04' or greater), the control code generated by the Backspace key is treated as an undefined control code and is displayed as a hyphen.

If the unprotected data list (or field) is defined by a 3250-compatible GECM order (that is, the CSID in attribute register 18 is X'00'), this key causes a backspace control code to be entered into the data list at the present cursor location.

The result of a character–backspace (reqd.)–character sequence is to display the second character superimposed on the first.

The cursor is not displayed when it is located at a backspace code. Note that the cursor is displayed when it is located at either of the two displayable characters that make up the superimposed character.

Rate of ANK Cursor Movement and Character Display

The rate of movement of the ANK cursor and the display of input characters depends on the image generation cycle of the user's display program. Accordingly, the user is ultimately responsible for satisfactory cursor movement and character display rates by organizing applications with the following understanding of cursor movement and character display in mind.

- When the Tab key is pressed, the display processor of the 5085 effectively scans the display storage to locate the next unprotected data field. This scan does not begin until the next GBGOP or GSRT order is executed, thus imposing a delay that, on the average, is equal to the image generation cycle. Note that this is not a scan in the sense that the display processor is devoted to locating the next unprotected field; it looks for the next field as it is normally interpreting orders. The cursor is removed from the screen during this process and is not displayed again until the next unprotected field is entered (that is, a GDCHAR or GECM order specifying an unprotected field is interpreted).

- Although characters are stored into display storage as the appropriate keys are pressed (as long as the cursor is in an unprotected field), they are not displayed until the next GBGOP or GSRT order is processed. If the regeneration cycle is prolonged for too much time, the display of characters and cursor movement will lag behind the keystroking process and the display may come in groups of 2 or 3 or more as the length of the regeneration cycle increases.

ANK-LPFK and Manual Input Register

Pressing ENTER, Cancel, one of the 24 program function keys (PF)keys on the ANK, or any of the 32 keys on the LPFK sets data into the manual input register and raises an I/O interrupt on the host system. After the register has been set and the interrupt raised, data sent to the manual input register is inhibited, pending receipt of a Read Manual Input command or the pressing of the Reset key. This command can be issued while the display program is executing, without affecting execution of the display program. The 3 bytes that are read contain the following codes identifying the keyboard:

ANK

Byte 0	Byte 1	Byte 2
10ec 0000	0000 0000	0000 0000

Bit 0 of byte 0 is set to 1.
Bit 2 of byte 0 is set to 1 for an ENTER key operation (e).
Bit 3 of byte 0 is set to 1 for a Cancel key operation (c).
All other bits of the 3 bytes are reset to zero.

LPFK

Byte 0	Byte 1	Byte 2
0100 0000	000k kkkk	1111 1111

Bit 1 of byte 0 is set to 1; the remaining bits of byte 0 are set to zero.

Byte 1 contains the binary number of the pressed key. (Bits 0, 1, and 2 of byte 1 are not used and are set to zero.)

Example:

Pressed Key	**Key Code Byte 1**
0	0000 0000 (X'00')
1	0000 0001 (X'01')
.	.
.	.
.	.
30	0001 1110 (X'1E')
31	0001 1111 (X'1F')

Byte 2 is set to X'FF'.

ANK Program Function (PF) Keys

Byte 0	Byte 1	Byte 2
0100 1000	000k kkkk	1111 1111

Bit 1 and bit 4 of byte 0 are set to 1; the remaining bits of byte 0 are set to zero. Byte 1 contains the binary number of the pressed key (bits 0, 1, and 2 of byte 1 are not used and are set to zero).

Example:

Pressed Key	Key Code Byte 1
1	0000 0000 (X'00')
2	0000 0001(X'01')
.	.
.	.
.	.
24	0001 0111 (X'17')

The user may choose to treat a response from the ANK program function (PF) keys as coming from a unique source of 24 keys. Optionally, by ignoring the setting of bit 4 of byte 0, the user may treat a response from the ANK PF keys as an alternative LPFK input source of 24 keys, regardless of whether an LPFK is configured at the processor.

A setup option is also provided at the processor to allow the user to specify the setting of bit 4 of byte 0. The system default is to set bit 4 of byte 0 to 1, indicating the ANK as the source.

Tablet

The processor supports the IBM 5083 Tablet, which can be used as a picking device and for drawing by specifying a series of coordinate positions to a display program.

The tablet can be featured with either:

- A cursor feature that provides the user with a convenient handheld "mouse-shaped" unit having four buttons (function keys) and a fine-crosshair for precise alignment.
- A stylus feature that provides a penlike device with a tip switch.

In general, both the cursor and the stylus features are supported by the processor in the same ways. When either the cursor or stylus is in presence with the tablet surface, coordinates are received by the processor and processed according to the active tablet mode, as defined below. The cursor buttons and the stylus tip switch, when pressed, provide an indication to the processor and/or the user's display program that a selection (or pick) has been made. In addition, if the cursor is attached, the user is also given a hexadecimal code indicating which button was pressed.

The 5083 Tablet has an actual active area of 11.5 x 11.5 inches and a resolution of 200 lines (or points) per inch. Therefore, it is possible to receive pointings

ranging from 0 to 2299. Because the display virtual image space is 4096 x 4096 pixels, it is necessary to map (or convert) to 4096 space, the pointings received. This is done by defining a 2048 x 2048 line (or point) virtual active area on the tablet, leaving a dead zone of 126 lines (or points) around the perimeter. This results in a tablet surface consisting of an active area of 10.24 x 10.24 inches, with an inactive perimeter of approximately 0.63 inch. Thus, when an X-Y coordinate pair is received from the tablet, each coordinate is selected and mapped to the display virtual image space. If the value is equal to or greater than 126, or less than or equal to 2173, the coordinates are considered to fall within the active area. In this case, 126 is subtracted from the received pointing and multiplied by 2 to convert it to 4096 space.

Tablet Support Modes

The processor provides two modes of tablet operation: user control mode and system pick device mode. These modes may be selected by invoking the system setup function from the alphanumeric keyboard (ANK) and following the prompts. (Refer to *IBM 5080 Graphics System: Operation and Problem Determination,* GA23-0133, for details.) The system default is the system pick device mode.

User Control Mode: In the user control mode, the tablet and tablet stylus or cursor may be used to interact with the display program and perform functions similar to those associated with a light pen. The level of function used is at the discretion of the user and is under control of the user's display program.

In this mode, coordinate data can be stored in display storage by using the Store Device Input (GSDEVI) order, addressing device number 1. Stylus tip switch or cursor button indications may also be stored in display storage when the GSDEVI order is issued. Subsequent action depends on the user's display program and the use of the GSDEVI order and/or the 3250-compatible Draw Symbol (GSYMB) order.

In addition, the GSDEVI order may be used to generate an interrupt to the channel. The GSYMB order may be used to define a pick detect window on the display and, optionally, to draw a tracking symbol. The pick detect window can produce a simulated light-pen detect, after which the processor will react under existing mode control as for the system pick device mode.

System Pick Device Mode: When the system pick device mode is selected, the tablet and tablet stylus or cursor are generally under system control. If the stylus or cursor is in presence with the tablet surface, moving the stylus or cursor across the surface and/or pressing the stylus tip switch or a cursor button yields the same results as if a picking device (for example, a light pen) were moved across the surface of the display. A pick window is moved across the virtual image space in correspondence with the location of the stylus or cursor on the tablet. (The system default pick window is 24 virtual pixels, but it can be changed through the setup function.) All graphic orders, switches, registers, and indicators associated with a picking device (for example, a light pen) are valid for tablet operation when in this mode.

At the option (via setup) of the user, in system device pick mode a system-generated tracking symbol appears on the display centered on the pick window when the stylus or cursor is brought into presence with the tablet. (The tracking symbol is a 25 x 25 real pixel cross.) The tracking symbol provides visual tracking of the stylus or cursor (crosshair) pointing on the tablet surface,

indicating its relationship to the display surface as the stylus or cursor moves. When the stylus or cursor is removed from presence with the tablet surface, the tracking symbol disappears.

Tracking Symbol Update

The 5080 Graphics System dynamically updates the location of the system tracking symbol on the display screen while the next image is being rastered. This is done approximately 25 times per second while the stylus or cursor is in presence with the tablet surface.

The processor, at each update, erases the previous tracking symbol, reads the current coordinates of the stylus or cursor, and rewrites the tracking symbol at the comparable coordinates on the display screen. In addition, the tracking symbol is always updated during the execution of a GBGOP or GSRT order.

The pick window is also cleared and rewritten each time the tracking symbol is updated such that the center of the symbol and the pick window coincide.

Note: If there is heavy use of the following orders in the display program during the current image generation cycle, the cursor speed may degrade:

- Begin Area Fill (GBGAF)/End Area Fill (GEAF)
- Draw Pixel Array (GDPXL)
- Load Color Table (GLCT)
- Move Data Block (GMVBLK)
- Draw Circle (GDCIR)
- Draw Symbol (GSYMB)

Destructive versus Nondestructive Tracking Symbol

Writing and erasing the tracking symbol repeatedly during an extended image generation time can be destructive to the image on the display. The degree of nuisance that this creates for the operator depends on the image generation time, complexity of the image, density of the image, and the setting of the background color. In a complex dense image, a trail of symbols in background color trails behind the current tracking symbol, leaving holes in the image. These holes disappear when the next image is displayed, at which time a new trail of holes begins.

In recognition of the human factors characteristics of a destructive tracking symbol, the processor provides a nondestructive tracking symbol capability as well as the default destructive cursor. For those images in which the density is not high, the image generation time is not long, and the background color is set correctly, the destructive symbol is adequate. In cases where the destructive cursor causes a human factors problem for the operator, the nondestructive symbol option can be selected. This requires the dedication of bit plane 7 to the tracking symbol.

Destructive Tracking Symbol

When the default destructive tracking symbol is used, the processor writes the tracking symbol regardless of the frame buffer bit plane erase protect mask set by the previous GBGOP order. In this case, the third entry (counting from zero) in the Color Table is used to generate the color or grayshade of the symbol and the first (zero) entry is used to erase the symbol. The symbol is erased by rewriting the symbol in the color or grayshade specified by Color Table entry zero.

Accordingly, the user can specify the color or grayshade of the symbol by loading the third Color Table entry with the appropriate value. In addition, the first (or zero) entry of the table should be loaded with the desired background color or grayshade value so the erased trail of symbols will blend into the background.

Nondestructive Tracking Symbol

If the user wants a nondestructive tracking symbol, bit plane 7 of the frame buffer must be dedicated to the tracking symbol (that is, avoid specifying color or grayshade entries that result in bit plane 7 being set). In addition, the Color Table must be loaded appropriately, as described below, and the nondestructive tracking symbol load bit should be set on in the GLCT order.

When the nondestructive tracking symbol option is selected, the processor writes or erases the cursor by protecting all existing bit planes except bit plane 7 and, as a result, writes or erases only in bit plane 7. (Erasing is performed by writing zeros.) The frame buffer mask in attribute register 5 is ignored when the tracking symbol is being updated.

Because bit plane 7 is always the low-order plane regardless of the number of bit planes installed, the effect is that when the symbol is written, odd-numbered Color Table entries are used to display the part of the image where the symbol occurs. Conversely, where the symbol does not exist (or where it was erased), even-numbered Color Table entries are used to display the image. Therefore, the user may specify the color or grayshade of the symbol by loading every other Color Table entry (odd entries) with the color or grayshade value desired.

Notes:

1. This option reduces the number of concurrent colors or grayshades that may be displayed in an image as follows:

   - 8 bit planes from 256 to 128
   - 6 bit planes from 64 to 32
   - 4 bit planes from 16 to 8
   - 2 bit planes from 4 to 2

2. The default value provided by the system for attribute register 3 (color value used when drawing) is 5, an odd value. Be sure to specify an even-numbered value for attribute register 3 if you use the nondestructive tracking symbol option. This is done because the odd-numbered entries in the Color Table are reserved for the tracking symbol.

Tracking Symbol Contrast

In an image in which many colors or grayshades are used, it may be difficult to see the tracking symbol when it moves through a portion of the image that is approximately the same color or grayshade as the symbol itself. The nondestructive tracking symbol option just discussed is useful in solving this problem. A different color or grayshade value can be loaded into the odd-numbered entries of the Color Table such that the symbol displays in a different color or grayshade, depending on the portion of the image through which it passes.

Tracking Symbol and Frame Buffer Nonswitch Mode

If the frame buffer nonswitch mode option is specified on a GBGOP order and there is a reason (or cause) to track the stylus or cursor on the tablet, the destructive tracking symbol could ultimately destroy the image on the screen. However, the nondestructive tracking symbol option allows symbol tracking, even in this case, without harming the image.

Dials Feature

The processor dials feature provides eight cone-shaped dials, each of which may be turned continuously in either direction. Upon rotating the dials, a range of scalar values indicating direction and extent of rotation (since last reset) are input to the processor. A display program can read and interpret these scalar values for any suitable purpose, such as translating, scaling, and rotating two- or three-dimensional images when the Transformation and Clipping Feature (TCF) is installed. The dial values are read by issuing a GSDEVI order addressing device number 2. (See "GSDEVI—Store Device Input" on page 5-45 for details.)

Each dial has a possible 256 increments (interrupts/revolution) in the 360 degrees the dial can be turned. The dials feature supports seven interrupt settings that can be specified by using the setup option (see *IBM 5080 Graphics System: Operation and Problem Determination,* GA23-0133, for details). The specifiable interrupts are 4, 8, 16, 32, 64, 128, and 256 possible points.

Because each dial is a high-resolution potentiometer, an electrically inactive area (dead zone) of approximately 1/16th (20 degrees) of the total potentiometer area exists. Thus, only 242 (00–240 and 248) interrupt values out of a total granularity of 256 are ever transmitted. The value 248 is transmitted only on interrupt settings greater than 16. Accordingly, only 62, 122, and 242 points are transmitted for interrupt settings of 64, 128, and 256, respectively.

Pick Control

Tablet Interactions

A pick module that compares the coordinates of the pickable primitives (for example, lines, markers, and characters) performs a correlation with a pick window established around an X-Y point in virtual image space.

The pick window can be set directly from the tablet in the following manner:

- From the tablet X-Y value. This value may be updated periodically during the processing of a display program, provided there has been no GSYMB order since the last GBGOP or GSRT order.
- From a GSYMB order. If the d bit of the GSYMB order is set to 1, the pick window defined in the GSYMB order overrides any previous settings and prevents further dynamic updating from the tablet. If the d bit is set to zero, GSYMB does not have any effect on the setting of the pick window.

Note: The GSYMB order is a 3250-compatible order. When a GSYMB order is encountered in display storage, the tablet system pick device mode (if active) is negated. Picking is now done based on the current draw position when the

GSYMB order was issued. Tablet system pick device mode will be reinstated upon execution of the next Set Buffer Address Register and Start Command or Start Display Program structured field.

Pick Detection Modes and Indicators

Two modes and four indicators per processor interact to define the pick detect environment at any instant during display program execution. These modes and indicators are defined below along with brief descriptions of their use. Precise definitions of their interactions and relation to the use of the stylus, stylus switch, and execution of the graphic orders are detailed in a set of decision tables. In addition, a less formal description of each graphic order is given, and the action of each order is defined with reference to the modes and indicators.

The two pick modes determine whether a pick detect is allowed, and, if it is allowed, what the reaction to the detect will be. A pick detect occurs when the modes are set appropriately, and displayed data of adequate intensity (color or grayshade) is seen by the pick module. [If the Load Attribute Register (GLATR) order is used, adequate intensity is assumed; if the 3250-compatible Load Attribute Register (GLAR) order is used, intensity must be set appropriately.] See "Interaction with Pick Indicators" on page 3-24 for a definition of the conditions required for a pick detect. See Appendix C for the status and sense data for an interrupt resulting from a pick detect.

The pick modes are pick detect mode and pick interrupt mode.

Pick Detect Mode: Possible states are switch enabled (S), nonswitch enabled (N), and disabled (D).

- No pick detects can occur during the disabled (D) state.
- One pick detect interrupt can occur per stylus switch closure during the switch enabled (S) state.
- Multiple pick detects can occur in a display program execution cycle during the nonswitch enabled (N) state; these are independent of the use of the stylus switch.

(Both nonswitch enabled and switch enabled detects can occur in the same execution cycle.)

Pick Interrupt Mode: Possible states are deferred (d) and immediate (i).

- In the deferred state, if a pick detect occurs, an indicator (the pick indicator, or PI) is set in the processor and display program execution continues normally (the detect is "remembered").
- In the immediate state, if a pick detect occurs, display program execution ceases immediately and an I/O interrupt is sent to the host CPU.

The two modes, pick detect and pick interrupt, are independent of each other, thus giving six possible combined states: Sd, Nd, Dd, Si, Ni, and Di.

The four indicators are:

Abbreviation	Name
PI	Pick indicator
PBI	Permit branch indicator
SDI	Single detect indicator
TSI	Tip switch indicator

Each of the four indicators has two possible states: on (1) and off (0). The descriptions below list the means of setting each indicator on or off as well as the functions whose action is modified by the state of the indicator.

Note: The setting and resetting of the PI and TSI may affect the condition code (see "Condition Code Register" on page 3-7 for details).

Pick Indicator (PI)

The *pick indicator (PI)* is in attribute register 13.

The PI is set on by:

- The pick module detecting the intersection of a line, character, pixel array, area fill, circle, or marker independent of the color/grayshade with the pick window (previously set by the pick device, that is, the tablet or a GSYMB order). When the pick detect mode is in either the switch enabled or nonswitch enabled state, the pick interrupt mode is in the deferred state.

  Note: This comparison is independent of the frame buffer mask, frame buffer function, pixel array, and area fill pattern values.

- A Load Attribute Register (GLATR) order (to attribute register 13).

The PI is set off by:

- Execution of a GBGOP or GSRT order.
- Execution of a Branch on Deferred Detect (GBDD) or Branch on No-Detect (GBND) order.
- Receipt by the processor of a Set Buffer Address Register and Start command or Write Structured command containing a Start Display Program structured field.
- A Begin Segment (GBGSEG) order.
- A GLATR order (addressing attribute register 13).

The PI is used to determine the action of the following orders:

- Branch on No Detect (GBND)
- Branch on Deferred Detect (GBDD)
- Permit Detect Interrupt (GPDI)
- Branch on Condition (GBC)

Permit Branch Indicator (PBI)

The *permit branch indicator (PBI)* is in attribute register 13 and indicates a closed switch state with no pick.

The PBI is set on by:

- Execution of a GBGOP or GSRT order when the stylus switch is closed.
- Execution of a GSDEVI order when the unit's switch is closed or the function keys of the unit are pressed.
- Receipt of a Set Buffer Address Register and Start command or Write Structured command containing a Start Display Program structured field when the stylus switch is closed.
- A GLATR order (addressing attribute register 13).

The PBI is set off by:

- A pick detect when the pick detect mode is in the switch enabled state.
- Execution of a GBGOP or GSRT order when all stylus switches are open.
- Receipt of a Set Buffer Address Register and Start command or Write Structured command containing a Start Display Program structured field when all stylus switches are open.
- Execution of a GBND order when the pick detect mode is in the switch enabled state.
- A GLATR order (addressing attribute register 13).

The PBI is used to determine the action of:

- Branch on No Detect (GBND) order.

Single Detect Indicator (SDI)

The *single detect indicator (SDI)* is in attribute register 13 and is used to prevent multiple picking during the time the switch is closed.

The SDI is set on by:

- A pick detect when the pick detect mode is in the switch enabled state.
- A GSDEVI order interrupt from a closed unit switch or a pressed program function key.
- A GLATR order (addressing attribute register 13).

The SDI is set off by:

- Execution of a GBGOP or GSRT order when all stylus switches are open.
- Receipt of a Set Buffer Address Register and Start command or Write Structured command containing a Start Display Program structured field when the pick stylus switch is open.
- Receipt of a Set Buffer Address Register and Start command or Write Structured command containing a Start Display Program structured field when the processor has an outstanding pick interrupt or a GSDEVI order interrupt.
- Any processor reset.
- A GLATR order (addressing attribute register 13).

The SDI is used to determine the action of:

- A pick detect when the pick detect mode is in the switch enabled state.
- A GSDEVI order interrupt.

Tip Switch Indicator (TSI)

The *tip switch indicator (TSI)* is in attribute register 13 and indicates the state of the tablet stylus switch.

The TSI is set on by:

- Execution of a GBGOP or GSRT order when one or more stylus switches is closed or one or more of the tablet function keys is pressed.
- Receipt of a Set Buffer Address Register and Start command or Write Structured command containing a Start Display Program structured field when one or more stylus switches is closed or one or more of the tablet function keys is pressed.
- A GSDEVI order with t=1 if the referenced unit switch is closed or one or more function keys of the referenced unit is pressed (at the previous GBGOP or GSRT order).
- A GLATR order (addressing attribute register 13).

The TSI is set off by:

- Execution of a GBGOP or GSRT order when the pick stylus switch is open.
- Receipt of a Set Buffer Address Register and Start command or Write Structured command containing a Start Display Program structured field when the stylus switch is open.
- A GSDEVI order with t=1 if the referenced unit switch was open and no function key was pressed (at the previous GBGOP or GSRT order).
- A GLATR order (addressing attribute register 13).

The TSI is used to determine the action of:

- A Branch on Switch Open (GBSO) order.
- A Branch on Condition (GBC) order.
- A pick detect when the pick detect mode is in the switch enabled state.

Interaction with Pick Indicators

This section defines those functions that interact with the settings of the pick indicators. The definitions are presented in the form of decision tables. The actions indicated by an X in the tables are taken only if all tests in the same column are satisfied. A test without a YES or NO indicated is always satisfied. Within a table the columns are mutually exclusive (that is, one set of tests cannot satisfy two columns). However, the set of columns does not always include all possible combinations of the tests. Any combinations not shown result in no action (that is, no-operation).

1. The Begin Order Processing (GBGOP) and Start Regeneration Timer (GSRT) orders test:

Is stylus switch closed? Actions:	YES	NO*
Set PI off	X	X
Set PBI on	X	
Set PBI off		X
Set SDI off		X
Set state to Si	X	X
Set TSI on	X	
Set TSI off		X

* SDI and PBI are set off only if all stylus switches are open and no function key is pressed.

2. The Set Buffer Address Register and Start command and Start Display Program structured field test:

Is tip switch closed?	YES	YES	NO *	NO *
Is an enable switch detect interrupt pending?	YES	NO	YES	NO
Actions:				
Set PI off	X	X	X	X
Set PBI on	X	X		
Set PBI off			X	X
Set SDI off	X		X	X
Set state to Si	X	X	X	X
Set TSI on	X	X		
Set TSI off			X	X

* SDI and PBI are set off only if all stylus switches are open and no function key is pressed.

3. The Branch on Deferred Detect (GBDD) order tests:

Is PI on?	YES	NO
Actions:		
Branch to target address	X	
Set PI off	X	

4. The Branch on No-Detect (GBND) order tests:

Is state = Si or Sd?	YES	YES	NO	NO	NO
Is state = Ni or Nd?	NO	NO	YES	YES	NO
Is PBI on?	YES				
Is PI on?	NO	YES	NO	YES	
Actions:					
Branch to target address	X		X		
Set PBI off	X	X			
Set PI off		X		X	X

5. The Permit Detect Interrupt (GPDI) order tests:

Is PI on?	YES	NO
Actions:		
Interrupt	X	
Set pick detect mode from deferred to immediate (i) state		X

6. The Branch on Switch Open (GBSO) order tests:

Is TSI on?	YES	NO
Actions:		
Branch to target address	NO	YES

7. The Store Device Input (GSDEVI) order tests:

Is I = 1?	NO	YES
Is tip switch closed? or		
Is any PF key pressed?		YES
Is SDI off?		YES
Actions:		
Interrupt		X
Set SDI on		X

8. System action when a pick detect occurs:

Adequate color/ grayshade for pick device? (see Note)		YES	YES	YES	YES
Is state = Si?	NO				YES
Is state = Sd?	NO			YES	
Is state = Ni?	NO	YES			
Is state = Nd?	NO		YES		
Is SDI off?				YES	
Is TSI on ?				YES	
Actions:					
Interrupt		X			X
Set PI on			X	X	
Set PBI off				X	X
Set SDI on				X	X

Note: Test applies only if 3250-compatible GLAR order used.

Coordinate System Specification

The processor supports a world coordinate system extending from ±32K (-32768 to +32767) for X,Y, and Z coordinates. The visible region of the coordinate space is called the virtual image space. (See Figure 3-2.)

Coordinates existing completely outside the virtual image space should be specified only if the Transformation and Clipping Feature (TCF) is installed and active; otherwise, the results are unpredictable. Lines beginning in the virtual image space, or passing through it, may wrap. (For an explanation of the uses of world coordinate space, refer to the description of the TCF on page 3-37.)

Virtual Image Space

There are 4096 addressable points in each of the X (horizontal) and Y (vertical) dimensions of the virtual image space; the bottom left point is 0,0 and the top right point is 4095,4095. The viewable area is the 4096 x 4096 space, which is mapped by the raster generator into "screen space" (1024 x 1024) by truncating the 2 low-order bits.

Coordinate Addressing for Lines and Markers

Any point in the world coordinate space may be addressed by graphic orders in the display program.

The coordinates contained in a graphic data list may be absolute, incremental, or relative, depending on the graphic order that precedes the list.

- *Absolute Coordinates*: When a graphic order specifies absolute coordinates, X-Y (and Z) values from the data list replace the current value in the X-Y-Z position registers. Thus, a move or draw occurs from the previous current draw position to the new position, or a marker is plotted at the new position in the same way.

- *Incremental or Relative Coordinates*: When a graphic order specifies incremental or relative coordinates, X-Y (and Z) values from the data list are added to the current value in the X-Y-Z position registers, creating a new current draw position. Thus, a move or draw occurs from the previous current draw position incremented by the value in the data list, or a marker is plotted at the new position in the same way.

Coordinates outside the virtual image space are usually not specified as absolute coordinates unless coordinate values are specified using a 16-bit signed binary number. However, 12-bit absolute, incremental, and relative coordinates could produce requests to place the current draw position outside the image area. The display of markers and lines is inhibited outside this area. In particular, a line having some part outside the virtual image space has that part inside the area displayed correctly up to the edge of the area.

Note: This "clipping" of lines at the virtual image space boundary is not compatible with the 3250, which would blank the entire vector.

The processor maintains logically correct X-Y-Z values within the limits -32768 and +32767 even if several consecutive movements remain outside the virtual image space. Attempts to increment these coordinates outside these limits causes an undefined current draw position.

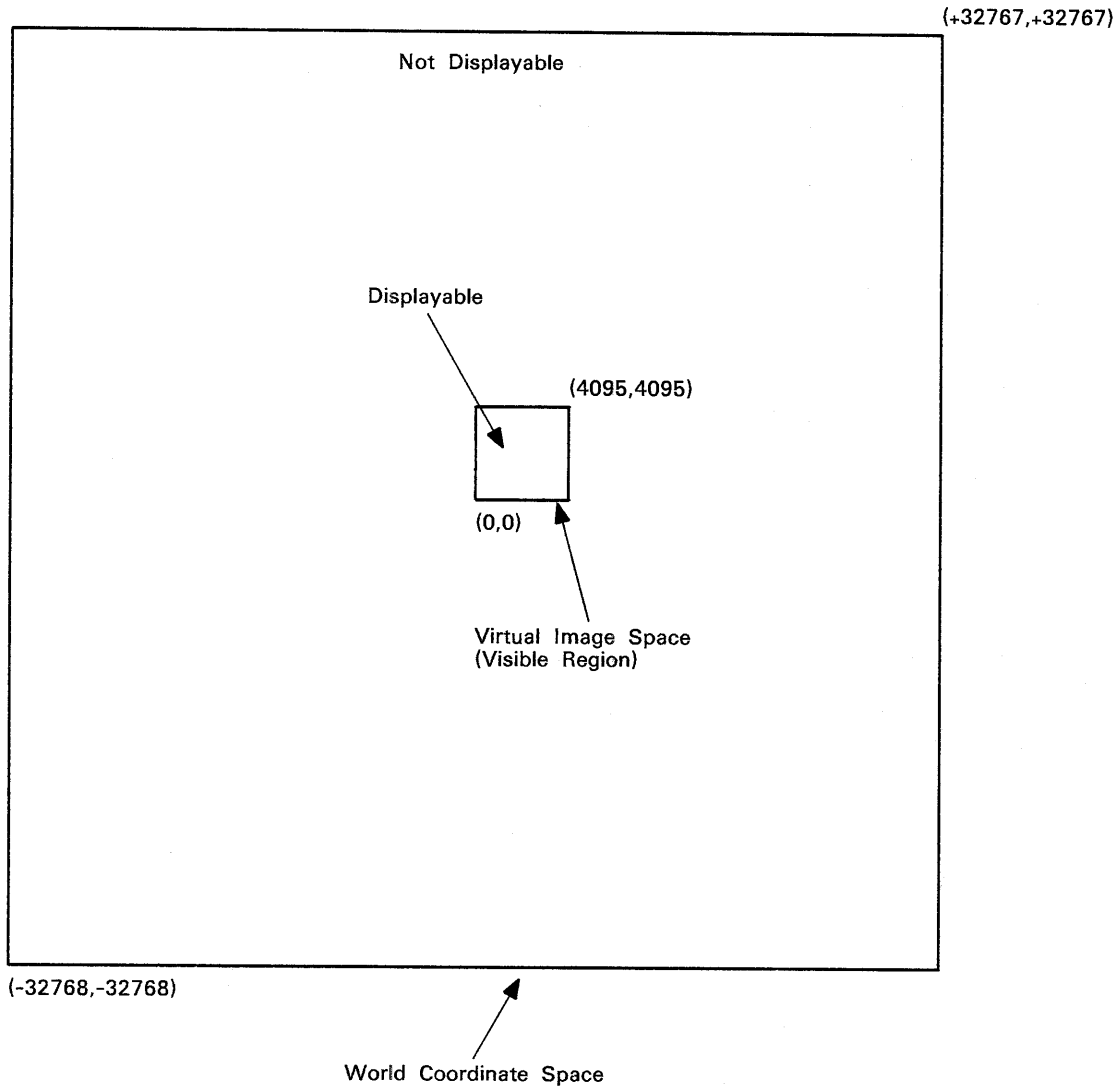


Figure 3-2. Base 5080 Coordinate Space

The host system may determine the contents of the X-Y-Z position registers by issuing a Sense channel command in response to a pick detect or at any other time. (See "X-Y-Z Position Registers" on page 3-6.) The GSDEVI order optionally returns coordinate values in the range of -32768 to +32767 to the display program.

The X and Y coordinate values returned to the host system by the 3250-compatible Read X-Y Position Register command, or stored in the display storage by the Store X-Y Position Registers (GSXY) order, are modulo 4096. The processor drops the Z coordinate value and sets the 4 high-order bits of the X and Y coordinates to zero in processing these functions. In this case, it is not possible to determine from the contents of the X-Y position values whether the current draw position is outside the virtual image space and, hence, outside the displayable area.

Specification of Coordinate Values

With the exception of character mode orders, output primitive (graphic) orders contain one or more groups of words specifying values of a coordinate point. There are three types of coordinate value specifications: absolute-relative, incremental, and long. The description of each order indicates which specification type is valid for that order.

Incremental Specification (6-Bit)

Programmable character set definitions (see "Type-1 Character Definition" on page 3-36) consist of a list of incremental-type point coordinate values in format 12. The incremental values are from -64 to +63 for X and Y coordinates.

Format 12. Group contains incremental X and Y coordinate values.

sxxx	xxxw	syyy	yyyb

s Sign bit (0 = positive, 1 = negative); negative numbers are in twos complement notation.

b Blanking bit; if b=1, the primitive is blanked, causing movement without display.

w Must be set to 1 to avoid possible interpretation of data field as an order.

Absolute-Relative Specification (12-Bit)

Absolute-relative graphic orders are followed by groups of one, two, or three data words. Each group defines the position of the next endpoint, and there may be zero or more groups comprising the data list. In the following definitions, b is the blanking bit, s is the sign bit, and X, Y, and Z bits represent X, Y, and Z coordinates that are integers. Note that the sign bit is included with all coordinates.

Format 1. Group contains X coordinate value.

1b0s	xxxx	xxxx	xxxx	X coordinate

Format 2. Group contains Y coordinate value.

1b1s	yyyy	yyyy	yyyy	Y coordinate

Format 4. Group contains X and Y coordinate values.

0b0s	xxxx	xxxx	xxxx	X coordinate
000s	yyyy	yyyy	yyyy	Y coordinate

Format 7. Group contains X, Y, and Z coordinate values.

0b0s	xxxx	xxxx	xxxx	X coordinate
011s	yyyy	yyyy	yyyy	Y coordinate
000s	zzzz	zzzz	zzzz	Z coordinate

b Blanking bit. If b=1, the primitive is blanked, causing movement without display.

s Sign bit (0 = positive, 1 = negative). Negative numbers are in twos complement notation.

Absolute Specification (16-Bit)

The Draw Line Absolute 3D 16 Bits (GDLA3L) order is the only order containing the long type specification of coordinate values of a line (format 14).

Format 14. Group contains absolute 16-bit X, Y, and Z values.

sxxx	xxxx	xxxx	xxxx
syyy	yyyy	yyyy	yyyy
szzz	zzzz	zzzz	zzzz

s Sign bit (0 = positive, 1 = negative); negative numbers are in twos complement notation.

Note: Blanking is specified via the mode control byte of the order (see "Order Format" on page 5-1).

Table of Valid Coordinate Formats by Order

The table below indicates the valid formats for the line drawing and marker graphic orders.

Graphic Order	Format Number(s)
GDLA2	4
GDMA2	4
GDLR1	1,2
GDMR2	4
GDLR2	4
GDMR3	7
GDLR3	7
GDLA3L	14

Note: Format 12 is used for programmable character set (PCS) definition.

Order Addressing and Data Modes

The first word of each order is a *set mode (SM) byte* that can have a hexadecimal value of X'28', X'2A', X'2C', or X'2E'. Not only does the SM byte allow the processor to detect the beginning of an order and, hence, the potential end of a data list, but its value indicates the addressing and data modes employed in the order specification. Absolute and relative addressing and indirect and immediate data fields may be specified in this manner.

Absolute versus Relative Addressing

If the addressing mode is specified as absolute addressing, the absolute address from the start of display storage is contained in the address word of the order.

If the addressing mode is specified as relative addressing, the processor calculates the effective address by adding the address word to the address of the first word of the order (that is, the address of the SM byte). In this case the address word contains a 15-bits plus sign (negative numbers are in twos complement) number relative to the address of word 1 of the order.

In the processor it may be advantageous for the user to write display programs using relative addressing so they may be relocated as subroutines, and so on.

Immediate Data versus Indirect Data

If the data mode is immediate, the data is contained in the order itself in a data word (and in successive words, if necessary).

If the data mode is indirect, the first and only word of the data field contains an address that points to the display storage location of the data word or words. This allows the user to isolate orders from data in writing display programs.

For more information on the SM byte order formats and addressing and data modes, see "Order Format" on page 5-1.

Summary of Graphic Order Functions

The 5080 Graphics System graphic order set consists of those orders that, together with 3250-compatible orders, are implemented on the 5080 system and required in accessing and using graphics system functions. Graphic orders are divided into the functional sets described in the following sections of this chapter. Detailed descriptions of individual graphic orders are given in Chapter 5; 3250-compatible graphic orders are described in Appendix A.

Frame Control

The frame control orders start display program execution, clear and swap frame buffers, set system initial conditions, set the proper system modes that apply to the overall display program, and terminate the program.

Image Generation and Mode Setting

Output primitive orders are used to draw images on the graphics system display screen using the image coordinate system. These orders generate pixels, markers, lines, and characters. The marker, line, and character orders are mode-setting orders; each order is followed by a data list. Data lists following marker and line

orders contain one or more groups of point coordinate values, as defined in "Specification of Coordinate Values" on page 3-29. Data lists following character orders contain EBCDIC characters and selected control codes.

Segment Definition

A segment is a logically related collection of output primitives that can be manipulated as a whole. A bracketing pair of segment orders is provided that delimits one or more orders that comprise a segment. Each segment is assigned a 2-byte name.

Segment nesting is permitted. Stack control orders (defined later in this chapter) can be used to save and restore the state of the prior segment(s).

Attribute Register Control

The processor allows the user to specify a number of attributes relative to the display image. These attributes, as well as default attributes, are maintained by the graphics system in attribute registers. Although many of these registers are reset to a default value by the GBGOP or GSRT order, they can be set and accessed under display program control using the GLATR and GSATR orders. The displayed color/grayshade is determined by the contents of an attribute register and a color/grayshade lookup table (the Color Table) that is loaded through a GLCT order.

Orders are provided to load the Color Table and load and store attribute registers.

Positional Device Control

A positional device control order stores X-Y position coordinates of the current draw position of the tablet or a signed value representing the amount of change in the dials feature since last accessed.

Pick Control

Pick control orders control the system pick function of the processor. Orders are provided to test conditions and transfer control, set pick windows, and set pick attributes.

Branch Control

The four branch control orders cause a transfer of control to a new address in display storage.

Stack Control

Stack control orders specify the size and location of the graphics system stack and push or pop information into it or from it. The stack is used to store orders requested by stack control orders. Each push order writes an appropriate display program order into the stack; each pop operation reads and processes the order on top of the stack.

Arithmetic Functions

Arithmetic orders add data, subtract data, divide data, compare data, shift data, and test data under masks. TCF provides additional arithmetic functions (see "TCF Advanced Arithmetic Orders" on page 5-97).

Data Move

Data move orders move addresses, move data, and move blocks of data.

No-Operation

A 2-byte no-operation order and a 4-byte no-operation order are provided that perform no operation and can be used to reserve words within display storage.

Configuration Data Storing

The GSCONF order stores the system configuration in display storage.

Graphics System Character Set Generation

The character set generation function of the processor consists of:

- Two orders: the Draw Character (GDCHAR) order and the Enter Character Mode (GECM) order for 3250 compatibility.
- Character set ID (CSID) attribute register (see "GLATR—Load Attribute Register" on page 5-34).
- Programmable character set (PCS) description.

Character Set Operation

Three methods of drawing characters are provided by the processor according to the value of the CSID attribute:

1. CSID X'00' to X'07'

   These CSIDs specify hardware-generated characters.

   - CSID=X'00': 3250-compatible character mode

     Orders operate as described in the 3250-compatible Enter Character Mode (GECM) order. (See "GECM—Enter Character Mode" on page A-13.)

   - CSID = X'01' to X'03': Reserved
   - CSID = X'04' to X'07': These are the graphics system base character sets and are processed as described in "GDCHAR—Draw Character" on page 5-24.

2. CSID = X'08' to X'BF'

   These CSIDs specify programmable character sets using 1-byte code points. Each 8-bit byte within the data list defines one character.

   Characters are drawn using descriptions of a preloaded programmable character set (PCS). The format of the PCS is described in "Programmable Character Set Descriptor Record," which follows.

These CSID values are used to select a PCS that has been loaded into the system storage by Select Write Memory Area and Write Memory Area commands addressing a user-defined PCS-type memory area.

The EBCDIC value of the character (from the data list following the GDCHAR order) is used as an offset into the PCS index, in the PCS descriptor record, to select the appropriate programmable character description.

For the effect of attribute registers 17, 29, and 30 on drawing of the PCS, see "GDCHAR—Draw Character" on page 5-24.

The p (protected) bit, when 1, prevents manual changes from the alphanumeric keyboard.

Cursor handling (and display) is identical to that for basic GDCHAR order fields.

Each character is a "primitive" for pick purposes, and acts in the same manner as base characters.

3. CSID = X'C0' to X'FE'

   These CSIDs specify programmable character sets using 2-byte code points.

   Each 16-bit word within the data list defines one character, thus allowing for character sets containing more than 256 characters (for example, Kanji). Otherwise, all other functions are the same except that keyboard cursor tracking is not provided, except for Kanji.

Note: CSIDs X'00' to X'1F', X'C0' to X'CF', and X'FF' are reserved for IBM use.

Programmable Character Set Descriptor Record

A PCS descriptor record contains data describing a programmable character set. It is loaded into a user-defined memory area using a Write Memory Area command chained from a Select Write Memory Area command. The descriptor record consists of control data followed by individual character definitions.

P and Q together define the character box within which a normal character will fit. The values of P and Q are defined in virtual image space (4K x 4K) and control spacing between characters and "new line" spacing. The bottom left corner of the box is 0,0 and the top right corner is P,Q.

Notes:

1. Characters may extend outside this box; P and Q control only the intercharacter spacing.

2. The value of P defined in the PCS descriptor record may be overridden if desired, by specifying a character inline spacing value in attribute registers 29 and 30.

Undefined character codes (codes outside the range CP0–CPn, or those with an index value of zero) are displayed as hyphens.

Byte	Field Length	Content	Meaning
0	2	Length	Total length of structure (PCS descriptor record), including length field.
2	4	Reserved	X'00000000'
6	1		Bits 0–2—Reserved Bits 3–7 (type) define the data format for the definitions of the programmable characters. One is defined: '00001'B = type 1
7	1		Reserved (must be zero)
8	1	Character Set ID	This field is used to identify the programmable character set.
9	1	Segment ID	If the CSID is larger than X'BF', this byte contains the first byte of the 2-byte character code of the programmable characters defined in this PCS descriptor record. If the CSID is from X'01' to X'BF', this, byte is ignored.
10	2	P	Range of X (between 0 and P)
12	2	Q	Range of Y (between 0 and Q)
14	1	CP0	Starting character code within this PCS (in the range X'41' through X'FE')
15	1	CPn	Last character code within this PCS. If this operand is zero, X'FE' is assumed. CPn must not be less than CP0.
16 to n	V	PCS Index	This field contains 2 bytes of data per programmable character described in this PCS descriptor record. (See definition of format type 1 for function and description of these bytes.)
n+1 to m	v	CDEF (CP0-CPn)	PCS definitions, starting at character code point CP0, in ascending order. (See below for format type 1.)

Note: Up to 48 PCS descriptor records may be loaded into processor storage using the memory area channel commands and the structured fields described in Chapter 4. This is dependent on the amount of storage installed in your processor.

The definitions start with an index of (CPn + 1 - CP0) 2-byte values. Each index value is the offset from the start of the PCS descriptor record that points to the actual character definition. This index must always be represented in its entirety, even if not all of the characters in the code range are defined. The maximum length of the index, if CP0 is specified as X'41', and CPn as X'FF', is, therefore, 191 x 2 bytes. Undefined values should be represented by a zero in the index.

Each character is defined as a series of moves or draws that define the shape of the character. The moves and draws are specified as X-Y pairs of signed relative values (relative to the previous ending point, or to the bottom left of the character box for the first X-Y pair). Each X-Y pair is specified using format 12 (see "Specification of Coordinate Values" on page 3-29). If the first X-Y pair is a draw rather than a move, the line is drawn from the bottom left corner of the character box. A move is specified by the low-order bit of the Y-value byte being on; a draw is specified by the low-order bit being off. The last X-Y pair in the series of moves and draws for the character is followed by 2 bytes containing the value X'2AFE'. This format of the data for an individual code point symbol is illustrated below:

sxxxxxx	1	syyyyyy	b	2 bytes
sxxxxxx	1	syyyyyy	b	2 bytes
.	.	.	.	.
.	.	.	.	.
.	.	.	.	.
sxxxxxx	1	syyyyyy	b	2 bytes
X'2AFE'				2 bytes

Loading Programmable Character Set Descriptor Records

A single PCS descriptor record must be loaded into a user-defined memory area as a single entity in one of two ways:

1. As a result of a single Select Write Memory Area command chained to a Write Memory Area command—that is, loaded in a single write.

2. As a result of multiple Select Write Memory Area commands chained to Write Memory Area commands specifying strictly ascending memory area offsets.

This is required because the graphics processor must perform certain postprocessing steps in order for the PCS descriptor record to be usable. If the user does not load the PCS descriptor record in one of the preceding ways, the postprocessing algorithm may fail. At that time a "Program Error" will be declared. The user should then reload the entire PCS descriptor record.

Multiple PCS descriptor records may be loaded into a user-defined memory area using either of the above techniques by:

- Chaining together the Select Write Memory Area/Write Memory Area command sequence(s) for one PCS descriptor record to the next, and so on.

- Issuing unique Select Write Memory Area/Write Memory Area command sequence(s) for each PCS descriptor record and recording the data block count of the previous record (Select Write Memory Area) so the data block offset can be calculated for the next Select Write Memory Area command, and so on.

- Reading the Memory Area Control Table to acquire the current transfer address to be used as the data block offset on a subsequent Select Write Memory Area command.

If more than one PCS descriptor record is written into a single memory area, the descriptor records must be contiguous—that is, if a PCS descriptor record ends at byte *n*, the next PCS descriptor record must begin at byte $n + 1$. Note that, in this case, byte *n* is determined by the length field of the descriptor record, which may actually be specified to indicate that the PCS descriptor record ends a number of bytes beyond the last PCS definition. This allows the user to define a PCS type memory area that crosses a 64K system memory boundary by defining an appropriate gap between the actual end of the PCS definition data and the 64K boundary at which the next PCS descriptor record may begin. In this case, the PCS memory area postprocessing is not impacted by this gap.

Area Fill

The area fill orders define the perimeter of an area to be filled and specify the related area fill controls or the pattern to be used during the fill operation.

These orders provide a polygon fill function with the following capabilities:

- Use of solid fill or user-defined patterns.

- Filling of two-dimensional (2D) objects, including those being dynamically transformed and clipped.

- Filling of circles as well as concave and convex objects.

- Filling of multiple overlapped and disjointed polygons that contain shapes ("islands") and/or crossing sides ("bow ties").

Circle Generation

A circle order specifying a radius is provided to control the drawing of circles outboard of the host in the processor.

Transformation and Clipping Feature

A two-dimensional (2D) and three-dimensional (3D) Transformation and Clipping Feature (TCF) is available on the graphics system to perform such graphics functions as scaling, clipping, rotation, translation, perspective, and mapping of coordinates outboard of the host system.

The TCF integrates two-dimensional and three-dimensional functions into a single definition, with the 2D and 3D orders common but for a few exceptions. In general, the differences between 2D and 3D functions are the presence of the Z coordinate in the graphic order data lists, the dimensions of the transformation matrix when in 3D mode, and optional perspective operation.

Functions supported by TCF include:

- Matrix loading/storing facility
- Matrix concatenation
- Transformations
  - Translation
  - Scaling
  - Rotation
- Clipping
  - Nonperspective clipping (box)
  - Perspective clipping (truncated pyramid)
- Perspective
- Window-to-viewport mapping
- Additional stacking orders
- Additional arithmetic orders

The basic function provided by TCF is to transform and clip coordinate values of an image in world coordinate space and map them from a window in world coordinates into a viewport in virtual image (coordinate) space to be displayed. (See Figure 3-3.)

In a 3D application, the image is projected into a 2D window in world coordinates before being mapped into the viewport. World coordinate space extends ±32K in X, Y, and Z directions. (See Figure 3-4.) Virtual image space extends from 0 to 4095 in X and Y directions (see "Virtual Image Space" on page 3-27).

The TCF also provides additional orders to set clipping boundaries, transform matrixes, and perspective viewpoint and viewport boundaries.

Hardware-generated character strings can be transformed (except for scaling of the individual characters) and clipped by TCF (see "TCF Character Mode Orders" on page 5-88). Programmable characters or markers, however, can be transformed, clipped, and mapped (see "Programmable Character Set Descriptor Record" on page 3-34).

Additional stack control orders and arithmetic orders are also provided by TCF.

Only the high-order 15 bits of the coordinate values are retained for TCF calculations.

TCF Three-Dimensional Assumptions

The TCF 3D coordinate system is left-handed; that is, the value of an X coordinate increases by moving from left to right, the value of a Y coordinate increases by moving up, and the value of a Z coordinate increases by moving away from the viewer into the device. Positive rotations are clockwise. The projection or viewplane is the X-Y plane located at Z = 0. The perspective viewpoint is on a negative Z axis and, for parallel projection, the direction of projection is perpendicular to the view of the (X-Y) plane.

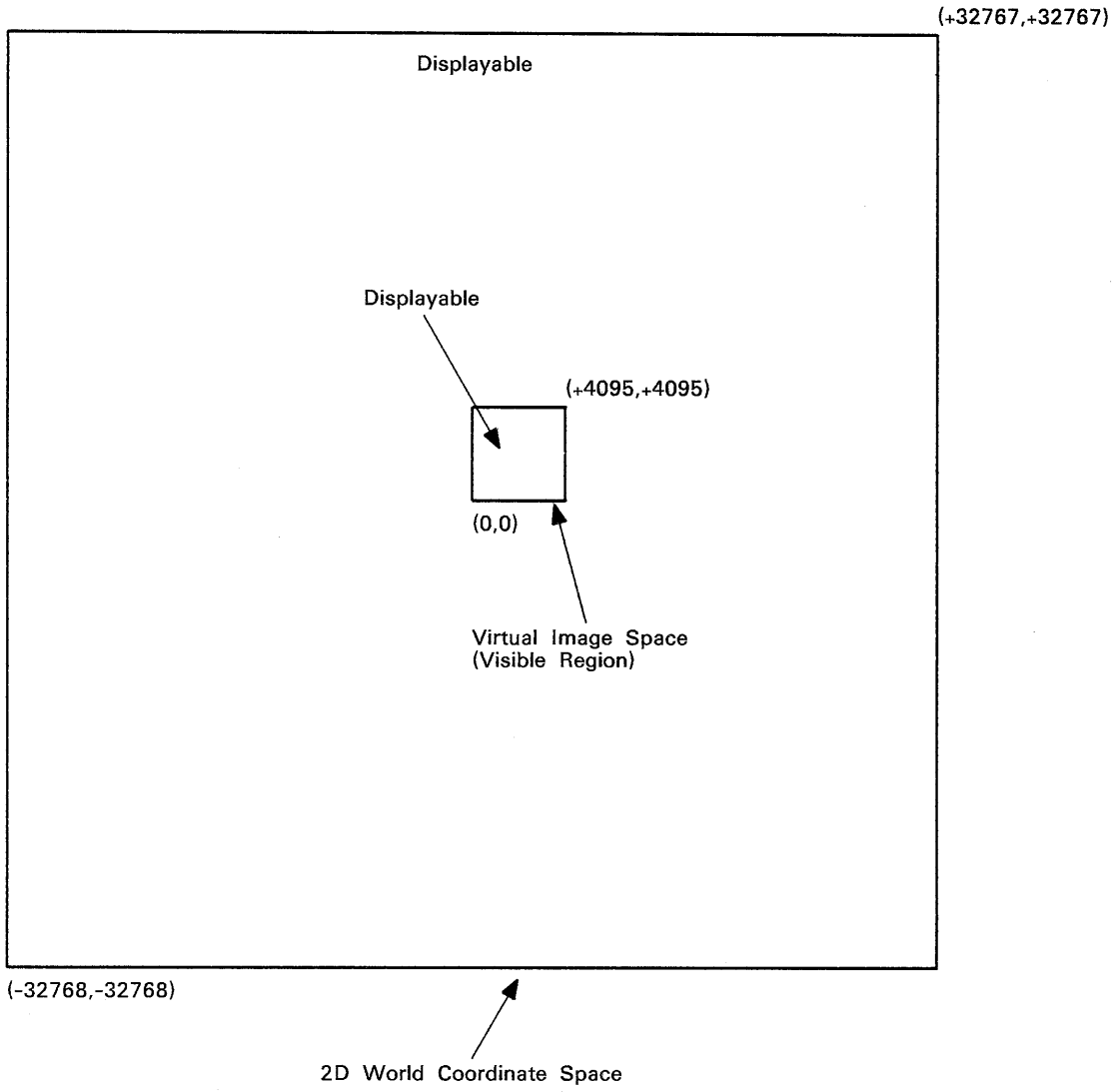


Figure 3-3. 5080 Transformation and Clipping Feature Two-Dimensional (2D) Coordinate Space

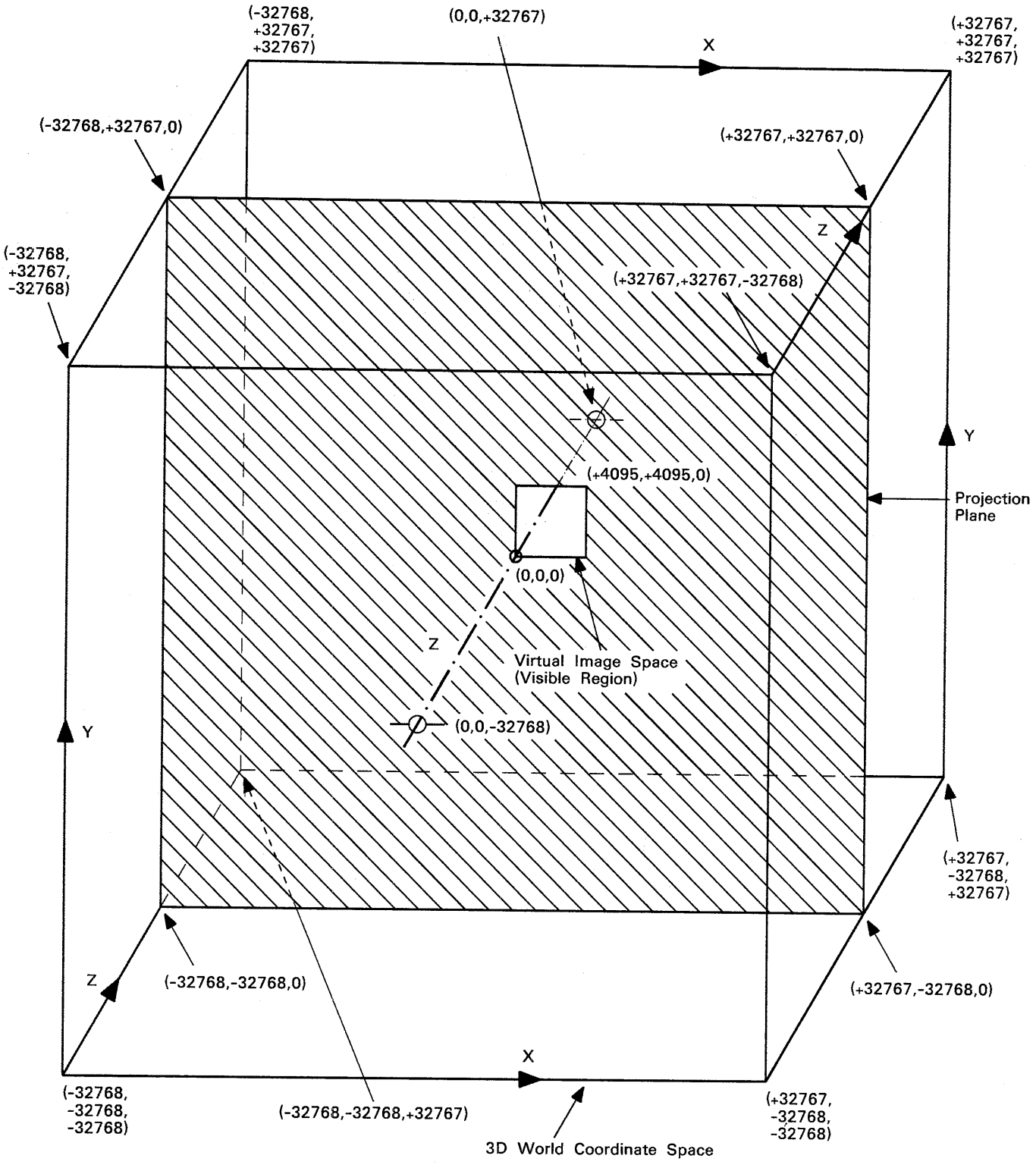


Figure 3-4. 5080 Transformation and Clipping Feature Three-Dimensional (3D) Coordinate Space

TCF Modes of Operation

The TCF operates in five TCF-unique modes in addition to the base system modes, which continue to apply when TCF is active. The five TCF modes are shown in the following table:

Mode	ON	OFF
M	Mapping	No mapping
P	Perspective	No perspective
D	Three-dimensional	Two-dimensional
T	Transformation	No transformation
C	Clipping	No clipping

TCF modes are controlled by bits 2 through 6 of attribute register 19 (see "Attribute Register Control" on page 3-32 and "Attribute Control Orders" on page 5-31), and may be set, changed, stored, or pushed into the stack by using the GLATR, GSATR, or GPATR orders.

The default modes at power-on, system reset, selective (device) reset, and GBGOP and GSRT order time are:

- No transformation
- Two-dimensional
- No clipping
- No perspective
- No mapping

Transformation Mode: In the transformation mode each point is transformed under control of the transformation matrix. A 4-bit power-of-two exponent called the scale factor is associated with each transformation matrix. The scale factor is used only when scale-up is required (see "Number Representation" on page B-1 for details of the transformation process).

If the transformation mode is off, points are not transformed. (See "Coordinate Addressing for Lines and Markers" on page 3-27 for a description of the formats of coordinate values for various transformation modes.)

Note: When the transformation mode is set, the current X-Y-Z position is not changed.

- Two-Dimensional Transformation Mode

  In the 2D transformation mode, X and Y coordinate values in the range from -32768 through +32767 are recognized and processed by TCF. If relative coordinates or conversion cause an X or Y coordinate value to exceed this range, no overflow indication is given. Exceeding this range is handled by wrapping from +32767 to -32768 and vice versa.

- Three-Dimensional Transformation Mode

  In the 3D transformation mode, TCF recognizes and processes the Z (or depth) coordinate in addition to the X and Y coordinates. The range of Z coordinates handled is the same as X and Y above.

Clipping Mode: When the clipping mode is set, the functions performed depend on the other TCF modes active at the time the coordinate data is processed.

- No-Transformation Mode

  In this case, the X,Y, and Z coordinates in virtual image space are compared with the clipping rectangle set by the Load Clipping Boundaries (GLCB) order.

  Lines that lie entirely outside this rectangle are discarded (that is, not passed to the raster generator). Lines that cross the edges of the rectangle are clipped; in other words, only that part of the line that lies on or within the rectangle is represented in the image. Lines entirely inside the rectangle are handled normally.

- Two-Dimensional Transformation Mode

  In this case, the posttransformation X-Y coordinates are compared with the clipping rectangle set by the GLCB order.

  Lines that lie entirely outside this rectangle are discarded (that is, not passed to the raster generator). Lines that cross the edges of the rectangle are clipped; in other words, only that part of the line that lies on or within the rectangle is represented in the image. Lines entirely inside the rectangle are handled normally.

- Three-Dimensional Transformation, No Perspective Mode

  In this case, the transformed X-Y-Z coordinates are compared against the viewbox defined by the X, Y, and Z boundaries set by the GLCB order.

  Lines that lie entirely outside this rectangle are discarded (that is, not passed to the raster generator). Lines that cross the edges of the rectangle are clipped; in other words, only that part of the line that lies on or within the rectangle is represented in the image. Lines entirely inside the rectangle are handled normally.

- Three-Dimensional Transformation, Perspective Mode

  In this case, the transformed X-Y-Z coordinates are compared against the view volume, a truncated pyramid. The apex of the pyramid is the perspective viewpoint, the intersection of the pyramid and the X-Y viewplane is bounded by X and Y clipping boundaries, and the pyramid is truncated by Z plane clipping boundaries.

  Lines that lie entirely outside this view volume are discarded. Lines that cross the edges of the view volume are clipped; in other words, only that part of the line that lies on or within the view volume is represented in the image. Lines entirely inside the box are handled normally.

Perspective Mode: When perspective mode is specified, the viewpoint must also be specified. The viewpoint constitutes the apex of the viewing pyramid, which is expressed as a negative Z coordinate value. This value ranges from -2, the X-Y viewplane (display screen), outward toward the viewer to a maximum negative value of -32K. The viewline is assumed to lie through the center of the view

volume. The image is projected on a window (bounded by X and Y clipping values) in the viewplane by dividing the X and Y coordinates by a perspective parameter derived from the viewpoint value and the Z coordinate values.

If no-perspective mode is specified, the viewpoint is at minus infinity and the view volume is a rectangular box bounded by X, Y, and Z values of the clipping boundaries. The image is projected on a window by simply dropping the Z coordinate values.

Mapping Mode: In this mode, the X and Y values of the clipping boundary defined by a Load Clipping Boundaries (GLCB) order are used to define a rectangular window in world coordinate space. That portion of the image in world coordinate space that falls within the window is mapped into a rectangular viewport in virtual image space specified by a Load Viewport Boundaries (GLVB) order. The mapping operation is a scale-translation operation. After mapping, the X and Y coordinates are passed to the raster generator for drawing of the image.

If the mapping mode is off, the 16-bit world coordinate values are truncated to 12 bits by dropping the 4 high-order bits; then they are passed to the raster generator.

Hardware-generated characters (CSID = X'00' and X'04' to X'07') are discarded if the character cell is not entirely within the viewport when in the mapping mode.

Note: The use of the new-line character code in conjunction with mapping to a viewport is not supported and may give undesirable results.

Programmable character sets (CSID = X'08' to X'FE') in a data list following a Draw Character (GDCHAR) order are transformed and clipped at the stroke (incremental coordinate) level.

In no-clipping mode, all lines with endpoints in the range handled by TCF are processed as in base mode (clipped at the virtual image space boundary). All character data list entries, including "new-line" characters, are also treated as if in base mode.

TCF Transformation, Clipping, and Mapping Scenario with Perspective

Figure 3-5 illustrates the transformation process provided by the TCF. The user's data is provided in a pretransformation world coordinate space of ±32K in X, Y, and Z. The data may be scaled, rotated and translated into a coordinate system that is, again, ±32K. This data can then be clipped to a user's specified set of clipping boundaries; then, mapped to a two-dimensional window in X and Y. The window is also within a ±32K frame of reference.

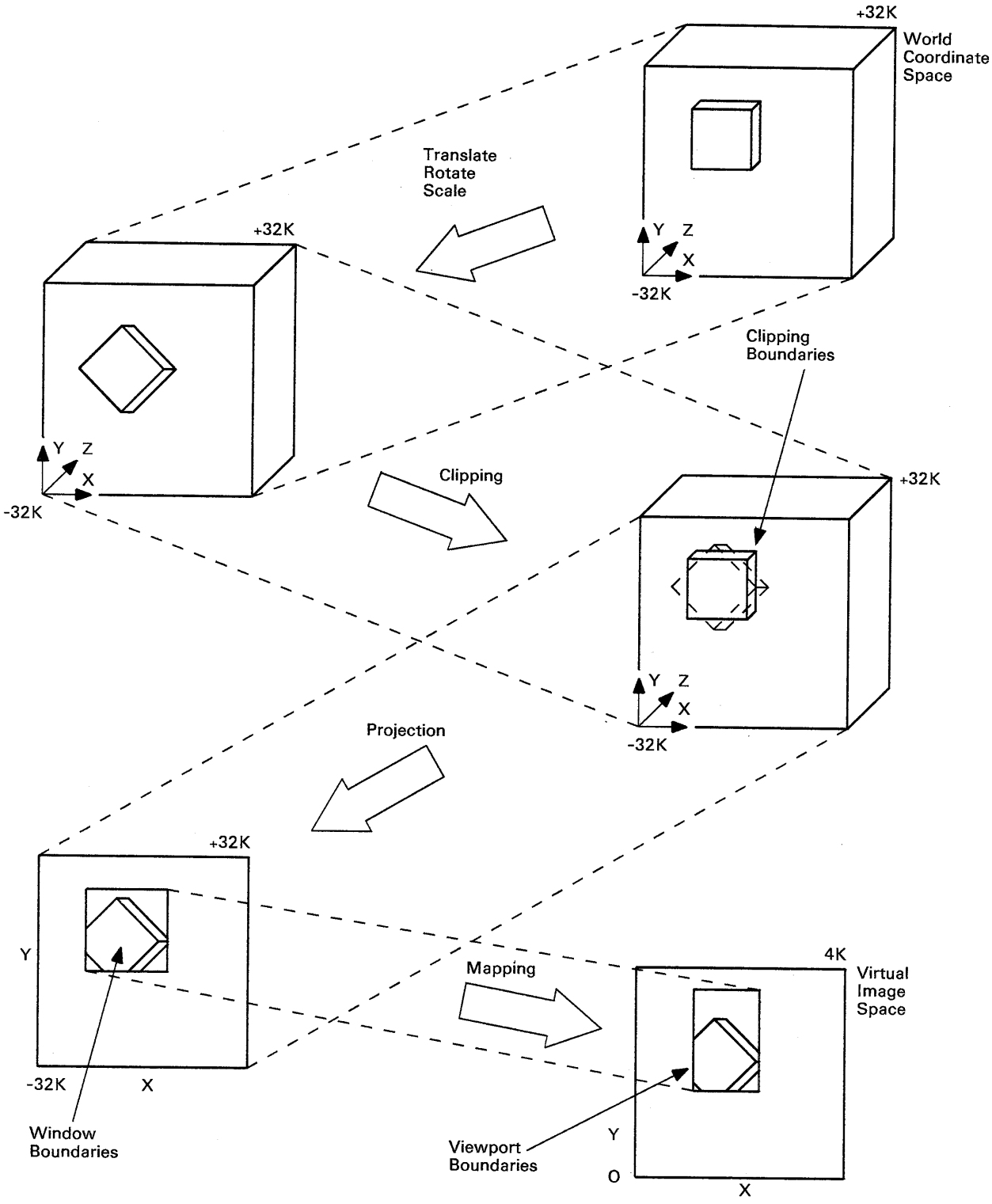


Figure 3-5. Transformation, Clipping, and Mapping Scenario with Perspective

Finally, the data in the window may be mapped to a viewport in virtual image space. The viewport is in the range of zero to 4K. After mapping, the X-Y coordinates are passed to the raster generator for drawing of the images on the display screen.

The example in Figure 3-5 shows a single viewport. However, multiple boundary conditions and multiple viewports in various sections of the display could be specified on the 5080 Graphics System.

TCF Pixel Arrays

The TCF does not operate on pixel arrays.

TCF Area Fill

The TCF provides a two-dimensional area fill capability only. If three-dimensional mode is specified, area fill defaults to two-dimensional mode and the Z coordinates are ignored.

When the TCF is installed and active, area fill operates on the posttransformation coordinate data; that is, the fill is performed in "screen space." Only the screen X-Y coordinates of the perimeter are accumulated. If pattern fill is specified, the patterns are not transformed.

If the clipping mode is on, the area fill is clipped to the X-Y clipping boundaries on the screen or the default. If the clipping mode is off, the area fill is not clipped and the result of area fill boundaries going off the screen is indeterminate.

TCF Error Conditions

TCF-unique error conditions that can occur during TCF operation are:

- TCF Matrix Element Overflow

  This error condition occurs during a Load Transformation Matrix (GLTM) order if concatenation causes the resultant scale factor to exceed the allowable value.

  This error condition causes the display program to be terminated and a Unit Check sent to the host system. Bit 7 of byte 1 is set in the sense data together with bit 4 in byte 4 (see "Sense Information" on page 4-29).

- Transformed Coordinates Overflow and Underflow

  These error conditions occur when, while performing a transformation, projection or mapping operation, a resultant coordinate value exceeds the range of -32K to +32K. No overflow indication is given. Exceeding this range is handled by wrapping from +32767 to -32768 and vice versa.

Graphics/3270/Setup Mode-Switching Considerations

There are three modes between which an operator may switch the graphics processor:

- Graphics mode
- 3270 mode (see Appendix D)

- Setup mode (see *IBM 5080 Graphics System: Operation and Problem Determination,* GA23-0133)

Switching from one mode to another allows the operator to perceive the graphics system workstation as a 5080 graphics device, a 3270 type device, or a setup input device that allows specification and/or modification of various 5080 system parameters.

When the graphics processor is operating in a particular mode, that mode is said to be "active." The "inactive" modes continue to function in background (with the exception of setup) such that when any mode becomes "active," the operator correctly perceives that latest state of that mode.

The Jump Screen key must be pressed to switch the graphics processor from graphics mode to 3270 mode or vice versa. The Setup key is pressed to change from graphics mode or 3270 mode to setup mode. When changing from graphics mode to either 3270 mode or setup mode, if no display program is running, the mode change is executed immediately. If a display program *is* running, the graphics processor waits until the display program stops normally (for example, as a result of a GBGOP or GSRT order) unless a Set Buffer Address Register and Start (or Stop) command is received before the mode switch is performed.

When a mode switch back to graphics mode is made, the display program is restarted from the order following the GBGOP (or GSRT) order or the address specified in the optional address word of the GBGOP order, when appropriate (see "GBGOP—Begin Order Processing" on page 5-7). If a Set Buffer Address Register and Start command or Set Display Storage Address Register structured field followed by a Start Display Program structured field was the last command received by the graphics processor while 3270 mode or setup mode was active, the display program is restarted at the new address. If a Set Buffer Address Register and Stop command or a Stop Display Program structured field was the last command received in 3270 mode or setup mode, the display program is not restarted.

Each mode must use the frame buffers to generate its own display. If a static image was previously generated and is being maintained on the screen as a result of specifying the frame buffer nonswitch mode on the GBGOP order or by protecting a portion of an image by using a frame buffer mask in attribute register 5 (see "GLATR—Load Attribute Register" on page 5-34), the static portion of the image will not reappear when graphic mode is restored. To minimize this problem, the graphics processor sets a mode switch indicator in attribute register 10 when the display program is restarted after a GBGOP order if a mode switch occurred. Accordingly, if mode switching is to be allowed by your application, be certain to test the mode switch indicator when your display program is restarted, and, if it is on, regenerate the static portion of the image.

5085 Attachment to a 3258

Channel Command Set

When a 5085 Graphics Processor is attached to a 3258 Channel Control Unit, the 5085 responds compatibly with the 3250 command set described in *IBM 3250 Graphics Display System Component Description,* GA33-3037, except as otherwise indicated in this document. See Appendix A for 3250-compatible channel commands and graphic order descriptions.

Although most processor functions can be used through attachment to a 3258 using the 3250 channel command set, doing so results in a significant loss in capability. These limitations are described below.

5085/3258 Attachment Function Limitations

Certain 5085 Graphics Processor functions are precluded or limited in scope when the processor is attached to a 3258 because the functions are dependent on the new channel commands and structured fields supported by the processor and the 5088 Graphics Channel Controller. These functions include the following:

- Multiple display storage pages

  The user is limited to the first page of display storage. Even if expansion memory is installed, it cannot be accessed. Required page number specifications in orders must be set to page number zero or a program error or inconsistent results may occur. Orders of concern are:

  - GPAGE
  - GLSR
  - GBAPL
  - GMVBLK

- Programmable character sets

  The use of programmable character sets is prevented. When using the GLATR order, the user is limited to the use of the system-defined fixed CSID settings X'00' or X'04' to X'07' in attribute registers 18 and 20.

- Area fill

  Area fill work areas cannot be defined. The area fill process is limited to shapes whose edge control information can be contained in the 2K system default area fill work area.

- Loadable line patterns

  The loading of user-defined line patterns is prevented. This limits the user to four system-defined line types (0 to 3) that provide a 3250- compatible set of line types. Accordingly, the user should limit line type specification in attribute register 2 to values in the range 0 to 3. If a value representing a loadable line pattern is set into attribute register 2 when using the GLATR order, the line type displayed is defaulted to a solid line.

- Loadable blink patterns

  The loading of user-defined blink patterns is prevented. This limits the user to the use of the two base blink patterns.

- Sense data

  Only 4 bytes of sense data can be returned to the host, restricting X-Y-Z position retrieval to the GSDEVI order and limiting the level of error recovery and problem determination that can be performed by the host and/or host application.

Chapter 4. Channel Commands, Structured Fields, Instructions, and Status/Sense Information

Channel Command Set

The 5080 Graphics System channel command set is described in this chapter and consists of the following commands:

- No-Operation
- Read Manual Input
- Read Memory Area
- Select Read Memory Area
- Select Write Memory Area
- Sense
- Sense ID
- Set Mode
- Write Memory Area
- Write Structured

The 5088 Graphics Channel Controller, with the 5085 Graphics Processor, also accepts and operates on the 3258 channel commands in the manner described in *IBM 3250 Graphics Display System Component Description,* GA33-3037, except as otherwise indicated in this document. (Also see Appendix A for 3250-compatible channel commands and graphic orders.)

Graphics Processor Memory Management Considerations

Processor storage (that is, the storage associated with the 5081 Display) is divided into two or more memory areas, where memory areas X'FFFE' and X'FFFF' are permanently assigned to display storage and the Memory Area Control Table, respectively. (See "System-Defined Memory Areas" on page 2-7.)

Display storage is divided into pages. In an environment in which user-defined memory areas exist, the last page may be from 2K to 64K bytes in length. All other pages other then the last page are 64K bytes in length. The graphics processor memory manager ensures that a minimum of 32K bytes is always assigned to display storage (memory area X'FFFE').

The page number is defined in those Write Structured command structured fields that specify display storage addresses. The display storage address for the device is set by the last unpacked field of the most recent Write Structured command structured field containing a page number and address.

Select Write Memory Area and Select Read Memory Area commands are used to set the current transfer address for the subsequent write or read memory area operation unless the addressed memory area is X'FFFF'. These commands do not have any effect on the display storage address register set by the preceding structured fields or the channel commands.

The default at processor power-on is page 0. The Branch Page (GBPAGE) order does not change or set the current page in the display storage address register or the current transfer address for channel commands.

The commands Write Structured, Select Write Memory Area, Set Mode, Select Read Memory Area, Read Memory Area, and Write Memory Area are accepted by the 5088 controller only if addressed to a 5085 Graphics Processor. If addressed to devices attached to a 3255, they are rejected by the controller with the Command Reject bit set in the sense byte.

RS232C Attachment Feature Memory Management Considerations

As in the case of the processor storage associated with display storage, the storage assigned to each RS232C Attachment Feature port is divided into memory areas. Each port has two permanent memory areas, X'FFFE' and X'FFFF', assigned to its 16K buffer and Memory Area Control Table, respectively. This allows the memory area commands Select Read Memory Area, Read Memory Area, Select Write Memory Area, and Write Memory Area to function properly with the RS232C Attachment Feature. Additional memory areas cannot be defined by the user, however; if any of the memory area control structured fields are addressed to an RS232C port, they are rejected with a structured field error.

Graphics Channel Controller Parity Checks

For all commands involving the transfer of data from the channel to the controller, the controller parity checks the data bytes as they are received from the channel interface. Detection of a channel bus-out parity error terminates the operation to prevent the possible display of erroneous data. This is signaled to the device as a Selective Reset request. Channel End, Device End, and Unit Check status bits are generated and the Bus-Out Check bit is set in sense byte 0.

The commands (and their codes) supported by the controller and the processor are:

5080 Graphics System Commands	Hex Code	Type	Notes
No-Operation	03	Control	
Read Manual Input	0E	Read	
Read Memory Area	0A	Read	1
Select Read Memory Area	2B	Control	1
Select Write Memory Area	FB	Control	1
Sense	04	Sense	
Sense ID	E4	Sense	
Set Mode	F7	Control	1
Write Memory Area	09	Write	1
Write Structured	05	Write	1

Notes:

1. These commands result in a Command Reject sense indication if issued to a 3255/Release 1 device.

2. All other command codes except those defined in Appendix A may result in a Command Reject sense indication.

The following I/O instructions, requests, and conditions are also supported:

- Start I/O (SIO)
- Start I/O Fast Release (SIOF)
- Test I/O (TIO)
- Halt I/O
- Halt Device
- Clear I/O
- Selective Reset
- System Reset
- Stack Status
- Command Chaining

Graphics Channel Controller Channel Commands

This section describes 5088 Graphics Channel Controller/5085 Graphics Processor channel commands as implemented by the controller and processor. (See Appendix A for descriptions of 3250-compatible commands.)

No-Operation

The No-Operation command performs no operation. It is an immediate command and no data bytes are transferred. The status response is Channel End and Device End status indications.

Read Manual Input

The Read Manual Input command transfers alphanumeric or lighted program function keyboard information to the channel. Pressing the alphanumeric keyboard (ANK) ENTER or Cancel (Cncl) key, any program function key or simulating these keys via a Begin Order Processing (GBGOP) order causes the Attention bit to be set in the status byte if the status and sense registers are clear.

The Attention status is passed from the processor to the controller at the first opportunity. The controller passes the Attention status to the channel at the earliest appropriate time, such as at the end of current channel operations, causing a separate asynchronous I/O interrupt.

When an ENTER key, Cancel key, or a program function key is pressed, the keyboards remain logically locked until the channel program responds with a Read Manual Input command to determine which key was pressed. This command does not affect graphic order processing. The Read Manual Input command or the pressing of the ANK Reset key resets the data and unlocks the keyboards if they are locked. If there is no manual input data, the data returned is zeros in the first 2 bytes and X'FF' in the third byte.

The Read Manual Input command provides Channel End and Device End together.

When the graphics system is powered on, the manual input data is reset.

Alphanumeric Keyboard (ANK)

If an alphanumeric keyboard ENTER or Cancel key caused the Attention status, the processor sends 3 bytes to the channel in response to a Read Manual Input command. Bits 0 and 1 of byte 0 are set to 10 to indicate that an alphanumeric key was pressed. Bytes 1 and 2 always contain all zeros.

Byte 0	Byte 1	Byte 2
10ec 0000	0000 0000	0000 0000
0 7	0 7	0 7

Notes:

1. Bits 0 and 1 of byte 0 are 10, indicating the response is from the alphanumeric keyboard.
2. Bit 2 of byte 0 is 1 if the ENTER key caused the Attention status.
3. Bit 3 of byte 0 is 1 if the Cancel key caused the Attention status.
4. All other bits of the 3 bytes are reset to zero.

Lighted Program Function Keyboard (LPFK)

If the lighted program function keyboard caused the Attention status, the processor sends 3 bytes to the channel in response to the Read Manual Input command. Bits 0 and 1 of byte 0 are set to 01 to indicate that one of the 32 LPFK keys has been pressed. Byte 1 contains a 5-bit binary code that corresponds to the number of the key.

Byte 0	Byte 1	Byte 2
0100 0000	000 key code	1111 1111
0 7	0 2 3 7	0 7

Notes:

1. Bits 0 and 1 of byte 0 are 01, indicating the response is from the LPFK. The remaining bits of byte 0 are reset to zeros.
2. The relationship of the key code in byte 1 to the pressed key is:

Pressed Key	Key Code (Bits 3–7)
0	00000
1	00001
.	.
.	.
.	.
30	11110
31	11111

3. Byte 2 is set to X'FF'.

ANK Program Function (PF) Keys

If an alphanumeric keyboard (ANK) program function (PF) key caused the Attention status, the processor sends 3 bytes to the channel in response to the Read Manual Input command. Bits 0 and 1 of byte 0 are set to 01 and bit 4 of byte 0 is set to 1 to indicate that one of the 24 program function keys has been pressed. Byte 1 contains a 5-bit binary code that corresponds to the number of the key.

Byte 0	Byte 1	Byte 2
0100 0000	000 key code	1111 1111
0 7	0 2 3 7	0 7

Notes:

1. Bits 0 and 1 of byte 0 are 01 and bit 4 of byte 0 is 1, indicating the response is from an ANK PF key. The remaining bits of byte 0 are reset to zero.

2. The relationship of the key code in byte 1 to the pressed key is:

Pressed Key	Key Code (Bits 3–7)
1	00000
2	00001
.	.
.	.
.	.
24	10111

3. The user may choose to treat a response from the ANK PF keys as coming from a unique input source of 24 keys. Optionally, by ignoring the setting of bit 4 of byte 0, the user may treat a response from the ANK PF keys as an alternative LPFK input source of 24 keys, regardless of whether an LPFK is configured at the graphics system.

Read Memory Area

The Read Memory Area command defines the data input area address in the host system's memory and then transfers data from the controller to the System/370 channel. A Select Read Memory Area command must precede and be executed in the graphics processor to read the data block from the memory area to the controller before the Read Memory Area command is processed. If the previous command is not a Select Read Memory Area command, the Read Memory Area command is not processed and the command is terminated with Channel End, Device End, and Unit Check bits set in the status byte and a Command Reject bit (byte 0, bit 0) set in the sense byte. This command does not update the current transfer address of the memory area referred to.

The length of the data block transmitted by executing the Read Memory Area command is equal to the *lesser* of the counts in this command and the preceding Select Read Memory Area command. If the Select Read Memory Area command count is *greater* than the Read Memory Area command count, the excess data is not held in the controller after the end of the Read Memory Area command. If the Select Read Memory Area command count is *not equal* to the Read Memory Area command count value, an incorrect length is indicated if the SLI bit is off.

If a 5085 hardware error occurs during the read operation, the command is terminated with Device End and Unit Check bits set in the status byte, and Hardware Error (byte 1, bit 4) and Memory Error (byte 5, bit 0) set in the sense byte.

A Read Memory Area command chained from a Select Read Memory Area command returns a Channel End status indication at the end of data transfer, followed by a separate Device End status indication.

Select Read Memory Area

The Select Read Memory Area command transfers a block of data from a processor memory area to the controller. The memory area is identified by a name (identifier). The identifier of the memory area, 2 flag bytes, address of the data block to be transmitted (that is, data block offset), and the size of the data block are defined in a 10-byte field transmitted from the System/370 channel. The format of these 10 bytes is:

Bytes 0–1	Memory area identifier	
Byte 2	Common flags	
	Bit 0	Data block offset validity
	Bits 1–7	Reserved
Byte 3	Memory area type-dependent flags	
	Bit 0	Read to the cursor indicator for type X'02'
	Bit 1	Read cursor address indicator for type X'02'
	Bits 2–7	Reserved
Bytes 4–7	Data block offset	
Bytes 8–9	Data block count	

Memory Area Identifier: The memory area identifier (bytes 0 and 1) specifies a predefined memory area (see "Memory Areas" on page 2-6).

Common Flags: The functions defined for the common flags (byte 2) apply in the same manner to all memory area types.

Bit 0 of the common flag byte is the validity bit. If this bit is zero, bytes 4 through 7 contain a valid data block offset. If this bit is 1, the contents of bytes 4 through 7 are ignored and the current transfer address is used to define the start of the data block to be transmitted.

If the memory area identifier is X'FFFF', these flags are ignored.

Memory Area Type-Dependent Flags: The functions defined for the memory area type-dependent flag (byte 3) depend on the type of the addressed memory area. The meaning of bits 0 and 1 when the memory area type is X'02' (display storage) is:

- Bit 0 = 1: The transfer of data from display storage to the controller is terminated when either the number of the data bytes transferred is equal to the value of the count field or a display storage location to which the cursor is assigned is encountered.

- Bit 0 = 0: The transfer of data from display storage to the controller is terminated when the number of the data bytes transferred is equal to the value of the count field.

- Bit 1 = 1: A 4-byte offset of the cursor is transferred to the controller. The values of bit 0 of byte 2, bit 0 of byte 3, and bytes 4 through 7 are ignored.

- Bit 1 = 0: Normal Select Read Memory Area command operation.

Data Block Offset: Data block offset (bytes 4 through 7) is a 4-byte field that specifies the address of the first byte of the data block to be transmitted. This is not an absolute processor memory address; it is the relative address of the first byte of the data block with respect to the beginning of the memory area. Data block offset values range from 0 through *n*, where *n* is the number of 2K blocks assigned to the specified memory area times (x) 2048 bytes minus 1. If the validity bit is zero, the data block offset is added to the starting address of the memory area by the processor to generate the current transfer address (see "Memory Area Control Table" on page 2-11). The value of the current transfer address in the Memory Area Control Table is not replaced by the contents of the data block offset field if the validity bit is 1.

Data Block Count: The data block count field (bytes 8 and 9) contains the length, in bytes, of the data block to be transmitted. The maximum allowable value is 32,768.

When the processor receives a Select Read Memory Area command, the data is sent to the controller. The value of the current transfer address in the Memory Area Control Table is incremented as data is read from the memory area. The controller stores the data. It does not send it across the channel to the host system until the data is fetched from the controller to the CPU by a chained Read Memory Area command. If no command is chained or the chained command is not a Read Memory Area command, the data stored in the controller is not retained.

The length of the data transmitted to the controller in response to a Select Read Memory Area command is determined by the data count passed by that command. The data transmitted across the interface in response to a subsequent chained Read Memory Area command is the smaller of the *channel control word (CCW)* count fields in that Read Memory Area command and the data count passed by the preceding Select Read Memory Area command. Incorrect length (generated by the channel if the SLI bit is off) is returned with the chained Read Memory Area command if the preceding Select Read Memory Area command passed a different count value. A Read Memory Area command chained to a Select Read Memory Area command returns a Channel End status indication at the end of data transfer, followed by a separate Device End.

If the Select Read Memory Area command count is greater than the channel Read Memory Area command count, the excess data is not held in the controller after the end of the Read Memory Area command. In other words, it is not possible to command-chain multiple Read Memory Area commands to pick up data from a single Select Read Memory Area command.

Note: Data chaining may be used to transfer data between the System/370 CPU and the channel, as data chaining is a channel function and not a control unit (controller) function.

Channel End and Device End status indications are returned separately for the Select Read Memory Area command. Channel End is presented when the command and the count value are accepted by the controller. Device End is presented when the requested data is stored in the controller and the dialogue between the controller and the processor is complete.

The following exception conditions cause Device End and Unit Check to be presented. Sense byte 1, bit 7 (Program Error) is set to 1. The exception conditions are:

- The addressed memory area does not exist; that is, the memory area has not been defined previously by a Define Memory Area or Rename Memory Area structured field and is neither memory area X'FFFF' nor X'FFFE'.

- The data block offset is larger than the size of the addressed memory area or the data block offset plus the data block count exceeds the size of the memory area (that is, wrap would have occurred). If the latter is true, sense byte 4, bit 6 (Invalid Memory Area Address) is also set on. The current location is not changed.

The following exception condition causes Device End and Unit Check to be presented and sense byte 0, bit 0 (Command Reject) to be set to 1.

- Length of the data associated with the command is less than 10.

If the processor is busy processing the data for more than the device working timeout period (about 3 seconds), Device End and Unit Check are presented and Device Working Timeout (sense byte 4, bit 1) is set.

Any Unit Check status error causes chaining to be broken. The integrity of the data in the controller is not guaranteed under these circumstances.

If a hardware error occurs during a read operation, the Select Read Memory Area command is terminated with Device End and Unit Check bits set in the status byte and Hardware Error (byte 1, bit 4) and Memory Error (byte 5, bit 0) set in the sense bytes.

Notes:

1. A Read Memory Area command chained to a Select Read Memory Area command addressing memory area X'FFFF' can be used by the application program to obtain the hexadecimal image of a segment of, or the entire, Memory Area Control Table.

2. If the value of the data block count field exceeds 32,768 bytes, the Select Read Memory Area command is rejected with Device End and Unit Check status indications; and Command Reject (sense byte 0, bit 0) is set.

Select Write Memory Area

The Select Write Memory Area command selects a processor memory area for subsequent write operations. The name of the memory area (memory area identifier), a flag byte, the address of a location within the named memory area from which the writing is to start, and the size of the data to be written are transmitted from the channel in a 10-byte field in the following format:

Bytes 0–1	Memory area identifier
Byte 2	Common flags
	Bit 0 Data block offset validity
	Bits 1–7 Reserved
Byte 3	Reserved
Bytes 4–7	Data block offset
Bytes 8–9	Data block count

Memory Area Identifier: The memory area identifier (bytes 0 and 1) specifies a predefined memory area (see "Memory Areas" on page 2-6).

Common Flags: The functions defined for the common flags (byte 2) apply in the same manner to all memory area types.

Bit 0 of the common flag byte is the validity bit. If this bit is zero, bytes 4 through 7 contain a valid data block offset. If this bit is 1, the contents of bytes 4 through 7 are ignored and the current transfer address is used to define the address where the first byte of the data block to be transmitted is to be stored.

Data Block Offset: Data block offset (bytes 4 through 7) is a 4-byte field that specifies the address of where the first byte of the data block to be transmitted is to be stored. This is not an absolute memory address; it is the relative address of the first byte of the data block with respect to the beginning of the memory area. Data block offset values range from 0 through *n*, where *n* is the number of 2K blocks assigned to the specified memory area times (x) 2048 bytes minus 1. If the validity bit is zero, the data block offset is added to the memory area starting address contained in the Memory Area Control Table to generate the current transfer address (see "Memory Area Control Table" on page 2-11). If the validity bit is 1, the value of the current transfer address in the Memory Area Control Table is not changed.

Data Block Count: The data block count field (bytes 8 and 9) contains the length, in bytes, of the data block to be transmitted. The maximum allowable value is 65,536.

The Select Write Memory Area command initializes the current transfer address of the memory area.

Channel End and Device End status indications are returned separately for the Select Write Memory Area command.

An exception condition causes Device End and Unit Check to be presented; the current transfer address is not changed. Sense byte 1, bit 7 (Program Error) is set to 1. The exception conditions are:

- The addressed memory area does not exist; that is, the memory area has not been defined previously by a Define Memory Area or Rename Memory Area structured field and is neither memory area X'FFFF' nor X'FFFE'.
- The addressed memory area is a read-only memory area.
- The data block offset is larger than the size of the addressed memory area or the data block offset plus the data block count exceeds the size of the memory area (that is, wrap would have occurred). If the latter is true, sense byte 4, bit 6 (Invalid Memory Area Address) is also set on. The current location is not changed.

The following exception condition causes Device End and Unit Check status bits to be presented and sense byte 0, bit 0 (Command Reject) to be set to 1.

- Length of the data associated with the command is less than 10.

If the processor is busy processing the data for more than the device working timeout period (about 3 seconds), Device End and Unit Check status bits are presented and sense byte 4, bit 1 (Device Working Timeout) is set.

Any Unit Check status error causes chaining to be broken.

Sense

The Sense command obtains data relative to the status of the addressed device. It can be issued by the host system at any time, but is usually the response to a Unit Check status. When a Sense command is issued at any other time and status is pending for another device on the controller, the controller responds with Busy and Status Modifier at initial selection.

The amount of sense data returned depends on the type of device, controller, and mode setting (see "Sense Information" on page 4-29). The error indicator portions of the sense data are reset following successful completion of this command. Additional conditions for the resetting of sense data are defined in "Sense Information."

Selections to other devices are allowed while one or more devices are in a Sense command pending state, because the controller maintains the sense data individually by device.

The Sense command provides Channel End and Device End together.

Determining X-Y-Z Position

The host system, when communicating with a 5085 Graphics Processor attached to a 5088 Graphics Channel Controller, may retrieve the X-Y-Z position registers associated with the display by issuing a Sense channel command in response to a pick detect or at any other time. See "X-Y-Z Position Registers" on page 3-6 and Appendix C for details.

This technique for retrieving position registers should be used in lieu of the 3250-compatible Read X-Y Position Registers channel command when writing 5080 system applications when attached to a 5088 Graphics Channel Controller.

Sense ID

The Sense ID command initiates the transfer of 7 bytes of data to the channel. These bytes contain the control unit type and model number, and device type and model number, of the addressed device. The controller accepts this command regardless of the type and state of any attached devices. Channel End and Device End status bits are returned together.

The device type and model number reflect the last known powered-up device attached to that controller address. In the case of a device address for which no device had ever powered up, the device type and model number are not valid.

The valid Sense ID values in hexadecimal representation are:

Bytes	0	1	2	3	4	5	6
5085 Device	FF	50	88	00	50	85	00
3255 Device	FF	50	88	00	32	55	00

Set Mode

The Set Mode command transfers, to the processor, 2 bytes of data that are retained in the processor on a device basis. Channel End and Device End status indications are returned separately. If either zero or 1 byte is passed from the channel to the controller, the command is not executed and is terminated with separate Channel End and Device End/Unit Check status, and Command Reject (byte 0, bit 0) is set in the sense data. If the count field in the CCW specifies more than 2 bytes, an incorrect length is indicated, unless the SLI bit is on in the CCW.

The following two values are defined for processor use:

- X'00F0'—Indicates that only sense bytes 0–3 are to be retrieved by the controller. This is the default mode.
- X'00FE'—Indicates that the full complement of sense bytes is to be retrieved by the controller.

All other values are reserved for future use.

Write Memory Area

The Write Memory Area command causes the data received from the channel to be placed into the consecutive locations of a memory area selected by the preceding Select Write Memory Area command. The Write Memory Area command should be chained to the Select Write Memory Area command. The write operation starts at the memory area current transfer address, which is either taken directly from the Memory Area Control Table (validity bit = 1) or is calculated by adding the starting address of the memory area to the data block offset parameter of the Select Write Memory Area (validity bit = 0). The length of the data stored in the memory area by executing the Write Memory Area command is equal to the lesser of the counts in this command and the preceding

Select Write Memory Area command. The command is terminated normally, even when the counts do not match, if the SLI bit is on. The memory area current transfer address is updated appropriately at the end of the write operation.

Note: When the target memory area is the display storage (memory area X'FFFE'), the write operation does not affect either the display storage address register (DSAR) or the regeneration address register (RAR).

The exception conditions that may be encountered during Write Memory Area command processing are given below.

- If the previous command is not a Select Write Memory Area command, the Write Memory Area command is not processed and the command is terminated with Device End and Unit Check bits set in the status bytes and the Command Reject bit (byte 0, bit 0) set in the sense bytes.

- If a 5085 hardware error occurs during the write operation, the Write Memory Area command is terminated with Device End and Unit Check bits set in the status byte and byte 1, bit 4 (Hardware Error) and byte 5, bit 0 (Memory Error) set in the sense bytes.

- If the maximum number (48) of character sets allowed is exceeded while writing in a programmable character set type memory area, the command is terminated with Device End and Unit Check bits set in the status byte, and sense byte 1, bit 7 (Program Error) set to 1.

- If a memory area type-dependent data error is encountered during the processing of a programmable character set type memory area, the command is terminated with Device End and Unit Check bits set in the status byte, and sense byte 1, bit 7 (Program Error) set to 1.

If the processor is busy processing the data for more than the device working timeout period (about 3 seconds), Device End and Channel End status bits are presented and sense byte 4, bit 1 (Device Working Timeout) is set.

A Write Memory Area command chained to a Select Write Memory Area command returns Channel End and Device End status bits separately at the end of data transfer.

Note: Incorrect Length (IL) is always returned if the SLI bit is off.

Write Structured

The Write Structured command transmits a block of data from the host system to the processor. A block of data can be a maximum of 512 bytes in length. Each block can contain one or more structured fields describing changes to be made to display storage and/or commands to the graphics system.

This command is executed whether or not the addressed device is executing a display program (display program running).

A Write Structured command buffer, separate from display storage, stores the structured data transmitted by a Write Structured command until processing is completed. The display storage address register is not changed by transmission to the Write Structured command buffer.

If a Bus-Out Check bit is detected, the controller prevents the processor from starting the unpack process.

Note: The controller is not aware of the contents of the data block.

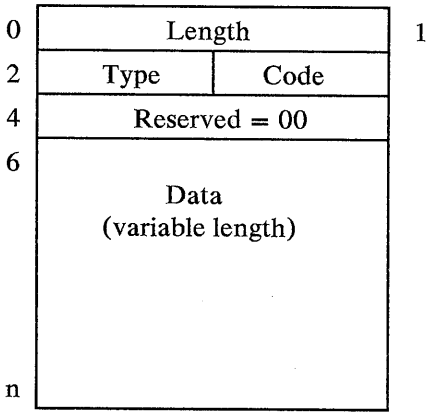
Channel End and Device End status bits are returned separately for the Write Structured command. Channel End is presented as soon as the data transmission is completed between the channel and the controller; Device End is presented when the structured fields (that is, the last structured field) transmitted by the command have been processed. The processor will be busy until the structured fields have been processed. If an error occurs while a field is being processed, further fields are not processed and Device End/Unit Check status is returned. A following Sense command returns a structured field error in sense byte 1, bit 7 (Program Error) and byte 4, bit 0 (Structured Field Error); and sense bytes 2 and 3 contain the address of the first byte of the structured field in error, where the address origin is the beginning of the data block transmitted with the Write Structured command.

If insufficient data to complete unpacking is sent, the processor returns a structured field error. If more than 512 bytes are transmitted, the data wraps in the structured field buffer and indicates a structured field error at the end of link transfer. In this event, the structured field error in the sense data indicates a count of X'0000'. The controller continues to transfer data across the link until the channel count is exhausted.

If the processor is busy unpacking for more than the device working timeout period (about 3 seconds), Device End and Channel End status bits are presented, and sense byte 4, bit 1 (Device Working Timeout) is set.

Structured Fields

Message blocks sent using the Write Structured command are sets of one or more structured fields. Each structured field is at least 6 bytes long in the following format:



Length: The 2-byte length subfield is a 16-bit positive integer that defines the total length of the structured field in bytes. This count includes the 2-byte length subfield itself. It can have a minimum value of 6 and a maximum value of 512. If the controller does not transfer at least 6 bytes or transfers more than 512 bytes, a structured field error is indicated and sense bytes 2 and 3 are set to zero, indicating that no unpacking has been attempted.

Type/Code: The type/code values defined in "Summary of Structured Field Codes and Types" on page 4-24 are the only valid structured field type/code values. All other values are reserved and, if used, result in structured field error indications.

- Type

  The 1-byte type subfield is always set to X'71'.

- Code

  The 1-byte code subfield defines the nature of the accompanying data and is, in effect, a command for the processor. The permissible values of the code subfield are defined in "Summary of Structured Field Codes and Types" on page 4-24. Other values result in a structured field error.

Reserved: This 2-byte field is reserved and should always be X'0000'. Any other value results in a structured field error.

Data: The data subfield may be between zero and 506 bytes in length. Proper interpretation of this subfield is based on the value in the accompanying code subfield. The maximum length of a Write Structured command data transmission is 512 bytes. Any number of structured fields may be contained in this data block, providing the sum of the lengths of the individual structured fields is equal to the sum of the lengths specified in the headers of the structured fields and to the transmission count. Violation of these rules results in a structured field error.

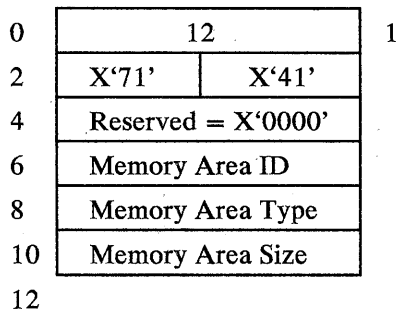
The following sections describe the structured field codes and formats supported in the base processor. For all fields, if the length is less than 6 bytes (or less than the minimum specified for each code), a structured field error is indicated; sense bytes 2 and 3 indicate the address of the first byte of the structured field in error. The unpacking process always works from the length defined in the field for all codes and does not rely on fixed lengths. If the length exceeds the required value, the extra bytes are discarded by the graphics system. A structured field error is indicated if the count exceeds 512.

Memory Area Control Structured Fields

This section describes the structured fields used to control processor memory area management.

Define Memory Area

The Define Memory Area structured field defines a memory area and its associated type and allocates one or more 2K blocks of storage to it. The format of the Define Memory Area structured field is:



- Memory area ID

  Memory area ID is a 2-byte number. (For definitions of this parameter, see "User-Defined Memory Areas" on page 2-7.)

- Memory area type

  This 2-byte field contains the memory area type code to be assigned to the defined memory area. The graphics processor records this code in the Memory Area Control Table (see "Memory Area Control Table" on page 2-11) for use in processing subsequent define requests and memory management commands.

  Valid memory area type codes are:

  X'0003' = Programmable character set

  X'0004' = Area fill work area

  Note that the graphics processor discards the high-order byte, which must always be zero, storing only the low-order byte in the Memory Area Control Table.

- Memory area size

  Memory area size is a 2-byte binary number that specifies the size of the storage partition to be allocated to this memory area and assigned to the memory area storage pool. It is specified in the number of contiguous 2K blocks of display storage.

The conditions that cause a structured field error are:

- The memory area ID is not valid; where an invalid memory area ID is a memory area ID that matches that of an existing memory area or the ID equals X'FFFF', X'FFFE', or X'0000' (which is a reserved memory area ID).

- The memory area type is not valid (type code=X'02' or type code=X'04' if a memory area of type area fill work space already exists) or if type code is not in the range X'03' to X'05'.

- The requested number of contiguous 2K blocks of storage is not available in the memory area pool.

- The Memory Area Control Table is full; all entries are assigned; no more areas may be defined.

- Satisfying the defined request would reduce the display storage assigned to the graphic device to less than 32K bytes. This is not allowed.

- The next available storage partition that will satisfy a request for type = area fill work area crosses a 64K boundary in system memory. This would cause inconsistent results in the area fill process. Accordingly, the user should attempt to define another area. This may be done by first examining the Memory Area Control Table and reserving the appropriate number of 2K blocks or performing another required Define Memory Area of another type

that will force the next allocation to begin beyond the 64K boundary. If this cannot be accomplished, the Define Memory Area structured field request may have to be resequenced.

Note: If a Define Memory Area structured field with type=programmable character set is processed and the assigned partition of storage crosses a 64K system memory boundary in the processor, no error is reported. However, if an individual programmable character set definition in the 1-byte case or segment definition in the 2-byte case crosses a 64K system memory boundary, the results of the character generation are undefined for that definition. Accordingly, the user must determine whether or not the resultant programmable character set load crosses a 64K boundary, and, if it does, must strategically arrange the definitions on either side of the boundary. In addition, the length field of the last character set or segment definition must be adjusted to include any unused space between the end of the last definition on the lower side of the 64K boundary and the first definition on the upper side of the 64K boundary or a PCS integrity check (that is, a program error) will result.

Rename Memory Area

The Rename Memory Area structured field changes the memory area identifier and memory area type of an existing memory area. The new memory area is allocated the same number of 2K blocks of storage at the same processor storage location as the old memory area. The format of the Rename Memory Area structured field is:

0	12		1
2	X'71'	X'42'	
4	Reserved = X'0000'		
6	Old Memory Area ID		
8	New Memory Area ID		
10	New Memory Area Type		
12			

- Old memory area ID and new memory area ID

  Old memory area ID and new memory area ID are 2-byte numbers. (For definitions of these parameters, see "User-Defined Memory Areas" on page 2-7.)

- New memory area type

  This 2-byte field contains the memory area type code to be assigned to the renamed memory area. The graphics processor records this code in the Memory Area Control Table (see "Memory Area Control Table" on page 2-11) for use in processing subsequent define requests and memory management commands.

  Valid memory area type codes are:

  X'0003' = Programmable character set

  X'0004' = Area fill work area

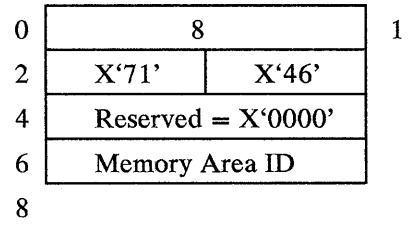
Note that the graphics processor discards the high-order byte, which must always be zero, storing only the low-order byte in the Memory Area Control Table.

The conditions that cause a structured field error are:

- The old memory area ID is not valid; where an invalid old memory area ID is a memory area that does not exist or the ID equals X'FFFF', X'FFFE' or X'0000'. (Memory area X'0000' is reserved.)
- The new memory area ID is not valid; where an invalid new memory area ID is a memory area ID that matches the ID of an existing memory area or the ID equals X'FFFF', X'FFFE', or X'0000'.
- The new memory area type is not valid (type code=X'02' display storage) or is not in the range X'03' to X'05'.
- The new memory area type = area fill work area and the assigned storage partition crosses a 64K boundary in system memory. This will cause inconsistent results in the area fill process. An attempt should be made to "define" a new memory area for the area fill work area.
- The new memory area type = area fill work area and an area fill work area already exists.

Delete Memory Area

The Delete Memory Area structured field deletes an existing memory area from the Memory Area Control Table and the memory area storage pool, and returns the storage partition assigned to it to display storage. After this structured field is processed, any reference to the deleted memory area is invalid. The format of the Delete Memory Area structured field is:



- Memory area ID

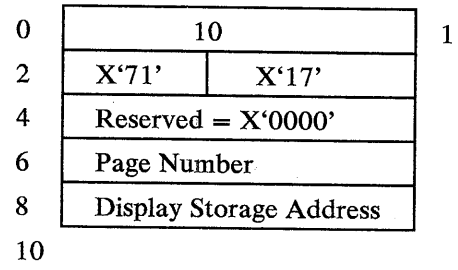
  The memory area ID is a 2-byte number identifying a defined memory area. (For a definition of this parameter, see "User-Defined Memory Area" on page 2-7.)

The condition that causes a structured field error is:

- The memory area ID is not valid; where an invalid memory area ID is a memory area that does not exist or the ID equals X'FFFE', X'FFFF', or X'0000'.

Set Display Storage Address Register

The Set Display Storage Address Register structured field sets the display storage address register for the addressed device at the processor. Display program processing is *not* affected. The format of this structured field is:



The Set Display Storage Address Register structured field does not affect the current transfer address of any memory area, including memory area X'FFFE' (display storage). The display storage address register is used to hold the starting address for a subsequent Start Display Program structured field for a subsequent Read Buffer, Write Buffer, or Insert Cursor channel command.

Minimum length is 10 (X'0A'); if the length is greater, additional bytes are discarded by the processor, although the command is executed.

- Page number

  Valid processor page numbers are X'0000' to X'0010'. (See "Display Storage Page Numbering" on page 3-2 for information on determining valid page numbers on your processor.) A structured field error is indicated and the command is not executed if the page number is not valid for your processor.

  Note: Only page number X'0000' is valid for the RS232C Attachment Feature.

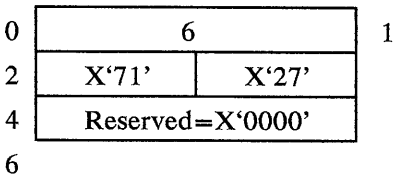
- Display storage address

  This display storage address field contains the address within the specified page and can range from 0 to 65,535. If the address is not a valid address within the page (the page is less than 64K bytes in length), a structured field error is indicated.

Start Display Program

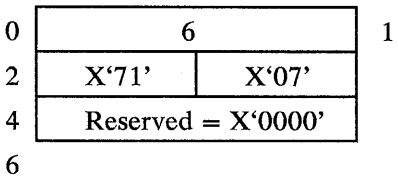
The Start Display Program structured field starts the display program at the current display storage address register setting. If the display program is stopped, this structured field starts it without resetting or modifying the attribute registers, stack registers, or any other information relating to the processor state, except for the condition code registers. If the display program is already running, this structured field does not become effective until a Begin Order Processing (GBGOP) or a Start Regeneration Timer (GSRT) order is executed. If an interrupt stops the display program before the GBGOP or GSRT order becomes

effective (including execution of a GEOP order), the pending Start Display Program structured field is discarded.



Stop Display Program

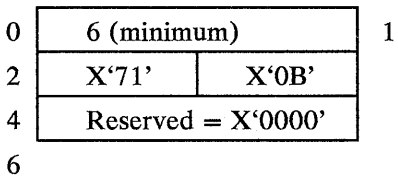
The Stop Display Program structured field causes display program processing to stop immediately. The display storage address register is set to the next sequential address following the last executed order.



Note: If the display program is already stopped, the Stop Display Program structured field performs no operation at the processor.

Sound Alarm

The Sound Alarm structured field activates, for a short period, a single-stroke audible alarm at the selected graphics system workstation. The format of the outbound field is:

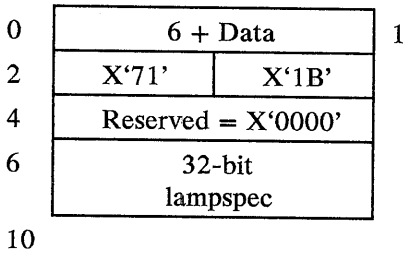


Lighted Program Function Keyboard (LPFK) Indicator Structured Fields

The lighted program function keyboard (LPFK) has 32 keys, each backlit by a lamp. These indicators can be set on or off individually using the following structured fields.

Set All Indicators

The Set All Indicators structured field sets all 32 indicators either on or off according to the user-defined 32-bit lamp specification (lampspec) in the data field.

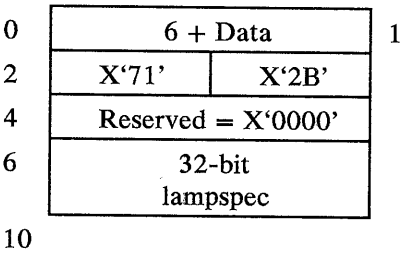


- 32-bit lamp specification

  If a bit is 1 in the lampspec, an indicator is turned on. If a bit is zero in the lampspec, an indicator is turned off.

Set Selected Indicators On

The Set Selected Indicators On structured field allows users to set only the selected indicators on without affecting the settings of the other indicators.

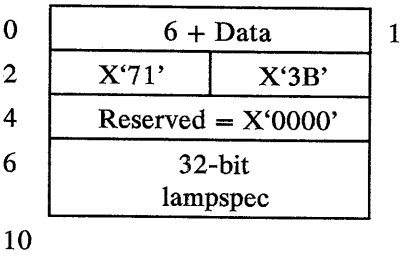


- 32-bit lamp specification

  If a bit is 1 in the lampspec, an indicator is turned on. If a bit is zero in the lampspec, there is no change to the associated indicator.

Set Selected Indicators Off

The Set Selected Indicators Off structured field resets the selected indicators only.



- 32-bit lamp specification

  If a bit is 1 in the lampspec, an indicator is turned off. If a bit is zero in the lampspec, there is no change to the associated indicator.

32-Bit Lamp Specification

The 32-bit lamp specification (lampspec) is associated with the 4 data bytes as follows:

	Bit Position							
Byte	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
0	0	1	2	3	4	5	6	7
1	8	9	10	11	12	13	14	15
2	16	17	18	19	20	21	22	23
3	24	25	26	27	28	29	30	31

Note: If the length of one of the LPFK indicator structured fields specifies more than 10 bytes, additional data is ignored. If the length specifies less than 10 bytes, only the corresponding lamps are affected.

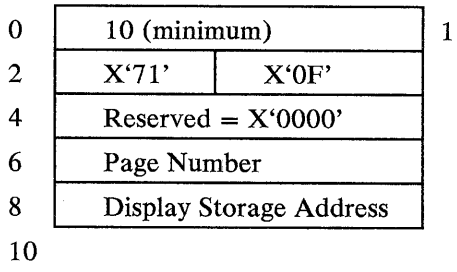
The operation of the LPFK indicator structured fields does not affect graphic order processing.

Cursor Operation Structured Fields

Each graphics system has a single *cursor address register (CAR).* This register, when nonnull, positions an alphanumeric cursor at the location in the display program that is to receive the next character entered from the keyboard.

Set Cursor Address Register

The Set Cursor Address Register structured field sets the cursor address register and, in so doing, enables alphanumeric input from the addressed location if it represents an unprotected field in display storage.



- Page number

  Valid processor page numbers are X'0000 to X'0010'. (See "Display Storage Page Numbering" on page 3-2 for information on determining valid page numbers for your processor.) A structured field error is indicated and the command is not executed if the page number is not valid for your processor.

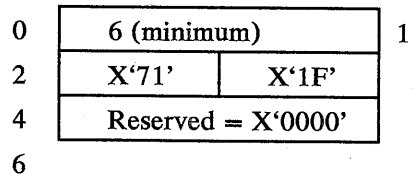
- Display storage address

  This display storage address field contains the address within the specified page and can range from 0 to 65,535. If the address is not a valid address within the page (the page is less than 64K bytes in length), a structured field error is indicated.

A structured field error is indicated if the page number is invalid.

Reset Cursor Address Register

The Reset Cursor Address Register structured field resets the cursor address register.



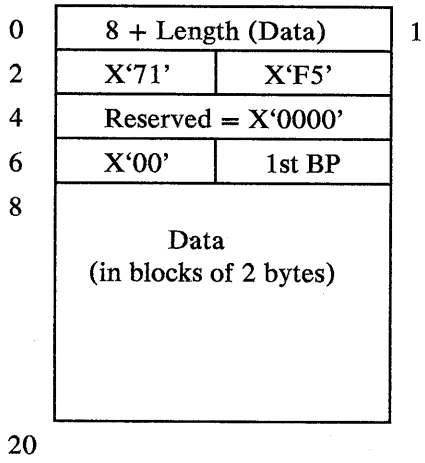
This structured field sets the cursor address register to null (no cursor).

Load Blinking Patterns

Blinking patterns (BPs) control the rate at which lines, circles, and so on, blink on the display screen.

Eight BPs (0 to 7) are supported by the processor, allowing the user simultaneously to blink different portions of the same image at different rates. Each BP is 2 bytes long and takes the form of a 16-bit binary number. The first two BPs (0 and 1) are system-defined and cannot be changed by the user. The remaining six BPs (2 to 7) can be changed by the user through the use of the Load Blinking Patterns structured field.

The format of the Load Blinking Patterns structured field is:



Data is loaded into the graphics system such that the first 2 bytes are assigned to the BP indicated by the first BP byte. Loading continues with each successive group of 2 bytes assigned to the next sequential BP until the data is exhausted. If the last BP is reached before the count is exhausted, the loading continues at BP 2; that is, the operation wraps. BPs 0 and 1 cannot be changed from their default values. Valid values for the last first BP byte are from 2 to 7. (The formats and meanings of BPs are defined in "Attribute Control Orders" on page 5-31.)

A structured field error is indicated if:

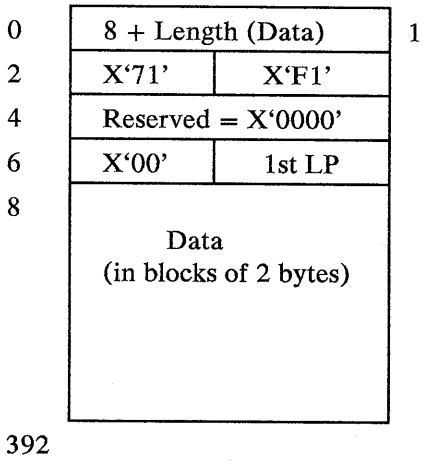
- The first BP is 0 or 1 or greater than 7.
- The count is not 8 plus a multiple of 2.

Load Line Patterns

Line patterns (LPs) allow the user to specify the format of the lines to be used when drawing lines, markers, circles, and so on, on the display screen.

Sixteen LPs (0 to 15) are supported by the processor, allowing the user to employ up to 16 different line types in the same image. The first four LPs (0–3) are system-defined and cannot be changed by the user. The remaining 12 LPs (4 to 15) can be changed by the user through the use of the Load Line Patterns structured field.

The format of the Load Line Patterns structured field is:



Each LP is 32 bytes in length. Data is loaded into the graphics system such that the first 32 bytes are assigned to the LP indicated by the first LP byte. Loading continues with each successive group of 32 bytes assigned to the next sequential LP, until the data is exhausted. If the last LP is reached before the count is exhausted, the loading continues at LP 4; that is, the operation wraps. LPs 0 to 3 cannot be changed from their default values. Valid values for the first LP byte are from 4 to 15. (The formats and meanings of LPs are defined in "Attribute Control Orders" on page 5-31.)

A structured field error is indicated if:

- The first LP is 0, 1, 2 or 3 or greater than 15.
- The count is not 8 plus a multiple of 32.

Summary of Structured Field Codes and Types

The structured field codes defined in the preceding sections are summarized below.

Function	Type	Code
Define Memory Area (Note 1)	X'71'	X'41'
Rename Memory Area (Note 1)	X'71'	X'42'
Delete Memory Area (Note 1)	X'71'	X'46'
Set Display Storage Address Register (Note 2)	X'71'	X'17'
Start Display Program (Note 2)	X'71'	X'27'
Stop Display Program (Note 2)	X'71'	X'07'
Sound Alarm (Note 3)	X'71'	X'0B'
Set All Indicators (Note 3)	X'71'	X'1B'
Set Selected Indicators On (Note 3)	X'71'	X'2B'
Set Selected Indicators Off (Note 3)	X'71'	X'3B'
Set Cursor Address Register (Note 3)	X'71'	X'0F'
Reset Cursor Address Register (Note 3)	X'71'	X'1F'
Load Blinking Patterns (Note 3)	X'71'	X'F5'
Load Line Patterns (Note 3)	X'71'	X'F1'

Notes:

1. Structured fields with type/code hexadecimal values X'7141', X'7142', and X'7146' are rejected with a structured field error if addressed to an RS232C Attachment Feature port.

2. Structured fields with type/code hexadecimal values X'7117', X'7127', and X'7107' are processed normally if sent to an RS232C Attachment Feature port.

3. Structured fields with type/code hexadecimal values X'710B', X'711B', X'712B', X'713B', X'710F', X'711F', X'71F5', and X'71F1' are processed but have no effect if the Write Structured command is addressed to an RS232C Attachment Feature port.

4. All nonallocated codes cause structured field errors.

Instructions and Resets

The CPU input/output instructions and resets operate in a manner identical to those specified in *IBM 3250 Graphics Display System Component Description,* GA33-3037; *IBM System/370 Principles of Operation,* GA22-2700; and *OEMI Channel-to-Control-Unit Interface,* GA22-6974.

The normal interaction between the controller and its host CPU is controlled by channel commands; these commands are conveyed to the controller by the Start I/O (SIO) or Start I/O Fast Release (SIOF) instructions from the CPU. Several additional CPU instructions, including Test I/O, Halt I/O, Halt Device, and Clear I/O, also give the host program a means of access to the processor.

Test I/O

The controller responds to the Test I/O instruction, executed by the CPU, with a status byte. If there is no outstanding status information for the addressed device, an all-zeros status byte is returned. Status for any other device is not reset. Any status for the addressed device is reset at the end of the sequence.

Following an initial selection sequence issued to the controller, the current poll in progress is subjected to a controlled termination, except when the command is a Test I/O or a No-Operation, in which case the poll in progress is allowed to continue normally. In addition, if a poll is not in progress, a Test I/O or No-Operation does not prevent initiation of the next poll cycle or prevent the presentation of data from a completed poll cycle. This allows pending interrupt conditions to be cleared by the Test I/O loop, although this is not recommended. The poll in progress or the next poll issued is not necessarily to the device holding the interrupt condition, but, if sequential polling is allowed to continue, a poll is eventually issued to the required device and the interrupt condition is cleared from the device.

Halt I/O and Halt Device

A Halt I/O instruction may cause the channel to issue an interface disconnect sequence to the controller, resulting in the termination of the current I/O operation. This does not affect display program processing currently in progress.

The interface disconnect sequence can be issued at various phases of interface activity, including during a data transfer. Any pending status is preserved and provided to the channel after the sequence has been completed.

Halt Device instruction processing is similar to Halt I/O processing. Refer to *IBM System/370 Principles of Operation,* GA22-2700, for a description of the differences.

System Reset

A System Reset causes the controller and all attached devices to be reset. A 5085, as a result of seeing the System Reset, effects a Selective Reset to each 5085 device. Any outstanding status or interrupts are lost. The diagnostics are not run following a System Reset. Also, the ready/not ready state of attached devices is not changed by a System Reset, but may be changed if the reset clears inhibiting conditions. If selected during the reset procedure, the controller responds with a status of Busy/Modifier during its reset procedure and automatically begins polling after the reset is complete. An asynchronous Control Unit End is generated at this time.

Selective Reset

The Selective Reset resets the device that is in operation at the time Selective Reset is detected. The reset clears attention pending status and sense, if any. In addition, the reset clears the link between the controller and processor and ensures that the alphanumeric keyboard is enabled (if not the RS232C Attachment Feature). Other devices on the same controller are not affected. If selected during the reset procedure, the controller responds with a status of Busy/Status Modifier and generates an asynchronous Control Unit End, if required, at the completion of reset processing.

Status Information

The status indication is a byte of information sent from the controller to the channel:

- During the initial selection sequence
- At the ending phase of an operation
- When an asynchronous condition (for example, pick detect) occurs

The status byte is reset after it has been accepted by the channel.

Initial Selection Sequence

During the initial selection sequence a status byte is sent in response to a command. An all-zeros status is sent if the command is accepted, except for the following immediate control commands: Insert Cursor, Remove Cursor, and Set Audible Alarm, which return Channel End; and No-Operation, which returns Channel End and Device End status.

If a command is not accepted because of error conditions, the Unit Check bit alone is set in the status byte; that is, for invalid command codes, Bus-Out Check on the command byte, and Intervention Required conditions.

If the controller has status pending for the addressed processor, the response is a status byte containing the pending status; for any command except Test I/O, a Busy status bit is included and the command is not accepted.

If the controller has status pending for another workstation, the response is a Busy bit plus a Status Modifier bit (Control Unit Busy) and the command is not accepted.

If a selection is issued to a device for which the controller has returned a Channel End, but Device End has not been generated, the response is either Busy, Status Modifier (Control Unit Busy), or Busy alone, depending on timing and the command in progress. The command is not accepted.

If the controller is selected during System Reset, Selective Reset, and interface disconnect processing (for example, Halt I/O), the response is Busy/Status Modifier set in the status byte (Control Unit Busy), in which case Control Unit End status is returned at the completion of reset processing.

Ending Conditions

When a data transfer is involved in a command, the channel and the controller work together to perform the transfer. At the end of the operation the status for that operation is presented by the controller.

Two status conditions, Channel End and Device End, indicate the end of a channel operation. The Channel End status condition means that the device has finished with the channel facilities following a control or data transfer operation, if any. The Device End status condition indicates that the device has finished the operation and is ready to accept a new command. Device End can occur at the same time as Channel End or later. If command chaining is taking place, a new operation may be initiated following presentation of the Device End bit. Disconnection from the channel will be signaled by the channel after Channel End, if block multiplexing is taking place.

Channel End is presented with initial status and Device End is presented at the completion of the following commands:

- Insert Cursor
- Remove Cursor
- Set Audible Alarm

Separate Channel End and Device End status indications are generated and presented to the channel at the end of data transfer for the following commands:

- Read Buffer
- Read Cursor (if no cursor, or cursor not found)
- Read Memory Area
- Select Read Memory Area
- Select Write Memory Area
- Set Buffer Address Register and Start
- Set Buffer Address Register and Stop
- Set LPFK Indicators
- Set Mode
- Write Buffer
- Write Memory Area
- Write Structured

Channel End is presented with Device End during initial status for:

- No-Operation

Channel End and Device End are presented together at completion for:

- Read Cursor (if cursor is found)
- Read Manual Input
- Read X-Y Position Register
- Sense
- Sense ID

Unit Check may be returned either alone, with Device End, or with both Channel End and Device End, and resets the chaining condition if the condition existed.

Interrupt Conditions

An asynchronous interrupt condition exists when the controller is not selected by the channel (for example, the host does not have a read outstanding), but the controller and/or processor have generated or detected a condition that requires assistance from the host system. Interrupts may occur when any of the following occurs:

- Pick detect
- Lighted program function keyboard (LPFK) action
- Simulated LPFK action, ENTER key, or Cancel key from the Begin Order Processing (GBGOP) order
- Store Device Input (GSDEVI) order interrupt
- ENTER, Cancel, or program function key from the alphanumeric keyboard (ANK)
- End Order Processing (GEOP) order
- Uncorrectable memory error (see Note below)
- Asynchronous Device End
- Unsolicited Attention, Device End, Unit Exception status indications (the processor becomes ready; for example, by being switched on)
- Asynchronous status after a Halt I/O instruction
- Asynchronous Control Unit End status indication
- Errors encountered during display program processing

The controller initiates a sequence to gain selection on the channel. When selection is achieved, the controller sends a status byte containing the appropriate bits, depending on the condition causing the interrupt. The status may also be retrieved by a Test I/O instruction or Start I/O instruction.

When a GBGOP order simulates (via Set Manual Input) an LPFK, ENTER or Cancel keystroke, an Attention interrupt is generated.

Unsolicited Attention/Device End/Unit Exception status indications are generated by the controller when a not-ready device becomes ready.

Note: If an uncorrectable memory error occurs in a processor display storage at a time when no channel command to that processor is in progress, all graphic order processing within the processor stops and an Attention/ Unit Check interrupt is generated, with Memory Error set in the sense byte (the Unit Specify bit is set to zero to indicate an error in the processor). Restart/retry of any operations that were in progress vary with the application.

Control Unit End

A Control Unit End status indication is generated at the termination of a previously interrogated Control Unit Busy condition. It may be included with asynchronous interrupts, that is, Attention for the same address. If Control Unit End alone is presented, it may be identified by any address assigned to the control unit.

Sense Information

The sense data consists of either 4 or 24 bytes:

- A 5085 device returns 4 or 24 bytes, depending on the last Set Mode command received.
- A 3255-base device or a 5085 when attached to a 3258 returns 4 bytes.
- A powered-off device defaults to 3255 base when powered on and the controller performs initial configuration, returning 4 bytes. When a Set Mode command specifying 24 sense bytes is received for a processor, 24 sense bytes are returned until a subsequent power-on, at which time the default to 4 sense bytes is reinstated. The number of sense bytes may also be set back to 4 by issuing the appropriate Set Mode command.

The contents of the sense bytes are defined in Appendix C.

Error portions of the sense data set up by a Unit Check status bit are reset by:

- The next command to that device address, with the exception of Test I/O, No-Operation, and Sense ID.
- Successful completion of a Sense command.
- A machine reset, system reset, selective reset, or power-on.

The display program running bit, character mode, and the display storage address are not reset by the first two cases if these conditions are still valid.

When a Unit Check status has been accepted by the channel, the controller contains the associated sense data in local storage and does not need to interrogate the device for sense data; the Sense command can be handled immediately by the controller while polling continues to other devices. Therefore, polling may be resumed and any status returned may be presented or queued for the channel.

Chapter 5. Graphic Orders

Introduction

Graphic orders supported by the 5085 Graphics Processor are grouped into four major categories:

1. The base order set of the 5080 Graphics System (see "Base Order Set" on page 5-6), which contains all the graphics system display storage orders required if your system has no special features.

2. Feature orders containing all the display storage orders required to use the Transformation and Clipping Feature [see "Transformation and Clipping Feature (TCF) Orders" on page 5-87].

3. The 3250 compatibility order set containing all the orders not in the base order set that are required to run an existing 3250 display program (see "3250-Compatible Graphic Orders" on page A-11).

4. The RS232C port order set that contains a subset of graphic orders and several new orders used with the RS232C Attachment Feature (see Chapter 6).

Order Format

The first word of each order is a *set mode (SM) byte,* which can have the value X'28', X'2A', X'2C', or X'2E', followed by a *mode control (MC) byte,* which has a value defining the specific operation required. These orders are defined in detail in later sections of this document. Undefined MC bytes are treated as a 2-Byte No-Operation (GNOP2) order.

Some orders have, following the first word, a second word, or pair of words that extends the MC byte. This extension (if present), called the MC extension, provides increased function control beyond that of the MC byte.

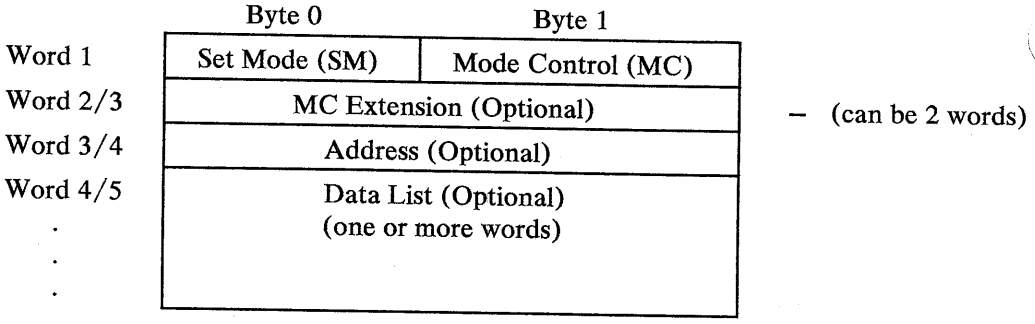
Following the order, which may be one to three words in length (SM, MC, and optional MC extension), can be an optional address word. The address word can contain an absolute or relative address.

Finally, the order may have one or more data words comprising a data list. A data list contains additional parameters required by the order. These parameters may contain immediate data, absolute pointers to data, a branch address, or a combination of any two or three of these parameter types.

The data list may be immediate to the order or indirect. An immediate data list follows immediately after the order or optional address word, if present. If the data list is indirect, the first and only data word contains an absolute address (within the current page) that points to the location in display storage of the data list.

The unique SM byte contained within every order is provided to enable the processor to detect the beginning of an order and, hence, the potential end of a data list.

The generic order format is as shown below:



All the words after word 1 are optional. Whether these words are present or not depends on the order code (MC byte), and, when present, they are always in the order shown.

The SM byte has the following format (in binary):

0 0 1 0 1 a d 0

where the a and d bits are interpreted as follows:

a = 0 *Absolute addressing.* The address word contains an absolute address within the current page. Valid values are in the range from 0 to 65,534.

a = 1 *Relative addressing.* The address word contains an offset (15 bits + sign, negative numbers are in twos complement) relative to the address of the order (word 1). The effective address is generated by adding the contents of the address word to the address of the order (word 1). A relative address may be in the range from -32,768 to +32,766.

d = 0 *Indirect data.* The first (and only) word of the data list contains an absolute address within the current page that points to the location in display storage of the data word or words.

d = 1 *Immediate data.* The data is contained in the data word (and successive words, if necessary).

Accordingly, the resultant SM byte values assign the following meanings to the address words and data words, respectively:

X'28' = Absolute addressing and indirect data

X'2A' = Absolute addressing and immediate data

X'2C' = Relative addressing and indirect data

X'2E' = Relative addressing and immediate data

If the order has no address or data words, the a and/or d bits are ignored.

The only valid hexadecimal values for the SM byte are X'28', X'2A', X'2C', and X'2E'. Any other values of the SM byte may may be treated as No-ops or data (depending on the preceding orders).

Note: No order or any part of an order, including the data list, should span a page boundary. If this occurs, the action taken by the graphics system is unpredictable.

Order Classification

The orders can be classified into eight classes according to their use of the optional words in the generic format (see "Order Format" on page 5-1):

	SM/MC	MC Ext	Address	Data
Class 1	YES	–	–	–
Class 2	YES	YES	–	–
Class 3	YES	YES	YES	–
Class 4	YES	–	YES	–
Class 5	YES	–	YES	YES
Class 6	YES	–	–	YES
Class 7	YES	YES	–	YES
Class 8	YES	YES	YES	YES

Notes:

1. One or more data, MC extension, or address words can be present, depending on the order. The data (words) list following graphics mode and character mode orders is not part of the order structure and is treated only as immediate data.
2. Classes 1 and 2 have no address or data words and are not affected by the value of the a or d bits in the SM byte.
3. Classes 6 and 7 have no address word and are not affected by the a bit in the SM byte.
4. Classes 3 and 4 have no data word and are not affected by the d bit in the SM byte.
5. Only Classes 5 and 8 are affected by both the a and d bits in the SM byte.

Valid Set Mode (SM) Bytes by Class

The following table shows the valid SM byte values for each order class.

Class	Valid SM Byte Values
1	2A
2	2A
3	2A,2E
4	2A,2E
5	28,2A,2C,2E
6	28,2A
7	28,2A
8	28,2A,2C,2E

Order Summary

The following table is an alphabetical list of all graphic orders defined for the 5080 Graphics System. For each order, the code point, class, and the reference page number are given. All unallocated and unimplemented code points are treated as 2-Byte No-Operation (GNOP2) orders during display program execution. They are reserved for possible future extensions or uninstalled features and, in order to avoid future incompatibilities, should not be relied on by users to continue to behave in this manner.

Mnemonic	Description	Code	Class	Page No.
GADD	Add Data	EE	5	5-67
GB	Branch Unconditional	FF	4	5-55
GBAPL	Branch after Push Link	27	6	5-62
GBAPLS	Branch after Push Link Short	28	4	5-63
GBC	Branch on Condition	70–7F	4	5-53
GBCNT	Branch on Count	F0	5	5-53
GBDD	Branch on Deferred Detect	FC	4	5-51
GBGAF	Begin Area Fill	9C	2	5-80
GBGIOP	Begin I/O Processing	89	3	6-6
GBGOP	Begin Order Processing	8A	2,3	5-7
GBGSEG	Begin Segment	68	7	5-30
GBND	Branch on No Detect	FD	4	5-51
GBPAGE	Branch Page	F7	6	5-56
GBSO	Branch on Switch Open	F5	4	5-50
GCOMP	Compare Data	E9	5	5-71
GCOS	Cosine	E6	5	5-100
GDCHAR	Draw Character	40–44	1	5-24
GDCIR	Draw Circle	98	7	5-27
GDIV	Divide Data	E5	5	5-69
GDLA2	Draw Line Absolute 2D 12 Bits	02	1	5-19
GDLA3L	Draw Line Absolute 3D 16 Bits	B2,B3	6	5-22
GDLR1	Draw Line Relative 1D 12 Bits	03	1	5-19
GDLR2	Draw Line Relative 2D 12 Bits	A2	1	5-20
GDLR3	Draw Line Relative 3D 12 Bits	A3	1	5-21
GDMA2	Draw Marker Absolute 2D 12 Bits	00	1	5-18
GDMR2	Draw Marker Relative 2D 12 Bits	A0	1	5-20
GDMR3	Draw Marker Relative 3D 12 Bits	A1	1	5-20
GDPXL	Draw Pixel Array	8D	6	5-14
GEAF	End Area Fill	9D	1	5-81
GEOP	End Order Processing	81	1	5-12
GESEG	End Segment	61	1	5-30
GLATR	Load Attribute Register	D3	6	5-34
GLCB	Load Clipping Boundaries	22	6	5-92
GLCT	Load Color Table	D0	6	5-31
GLPAT	Load Pattern	D7	6	5-82
GLSR	Load Stack Register	D5	6	5-59
GLTM	Load Transformation Matrix	20	2,7	5-89
GLVB	Load Viewport Boundaries	1A	6	5-96
GLVPT	Load Viewpoint	24	6	5-94
GMUL	Multiply Data	E4	5	5-98
GMVA	Move Address	EB	5	5-75
GMVBLK	Move Data Block	DB	6	5-77

Mnemonic	Description	Code	Class	Page No.
GMVW	Move Word	EC	5	5-76
GNOP2	2-Byte No-Operation	80	1	5-79
GNOP4	4-Byte No-Operation	C0	2	5-79
GPATR	Push Attribute Register	2B	2	5-63
GPCB	Push Clipping Boundaries	2A	1	5-102
GPCDP	Push Current Draw Position	2D	1	5-64
GPOPNOP	Pop Stack and No-Operation	A4	1	5-65
GPOP	Pop Stack	2F	1	5-66
GPSMC	Push Stack Marker Code	B1	1	5-65
GPTM	Push Transformation Matrix	29	1	5-101
GPVB	Push Viewport Boundaries	1C	1	5-103
GPVPT	Push Viewpoint	26	1	5-102
GSATR	Store Attribute Register	D4	3	5-43
GSCB	Store Clipping Boundaries	23	4	5-94
GSCONF	Store Configuration Data	DF	4	5-83
GSDEVI	Store Device Input	E8	3	5-45
GSHIFT	Shift Data	E3	3	5-72
GSIN	Sine	E7	5	5-99
GSSR	Store Stack Register	D6	3	5-61
GSTM	Store Transformation Matrix	21	4	5-91
GSUB	Subtract Data	EF	5	5-68
GSVB	Store Viewport Boundaries	1B	4	5-97
GSVPT	Store Viewpoint	25	4	5-95
GTM	Test under Mask	ED	5	5-74
GWAIT	Wait on Timer	8F	2	6-14

Notes:

1. Depending on size, orientation, and protection modes of the GDCHAR order, different mnemonics are used.

2. For the GLTM order, if the concatenation control =00 (unit matrix), CLASS=2; if cc=01, 10, or 11, CLASS=7.

3. For the GBGOP order, if the manual input indicator (s) bit = 0, CLASS=2; if s=1, CLASS=3.

Order Sequencing

For character mode and graphics mode orders that are followed by zero or more data fields forming a variable-length data list, the next order to be executed is precisely that order terminating the data list. The next order to be executed following a control mode order is determined by first fixing a search start location and then scanning each word from that location for an order (SM code in the first byte). The search start location is either the byte following the last byte of the current order (including its data field), or, in the case of a branch order for which the conditions necessary for the transfer are satisfied, the target location specified in the branch order.

When beginning order execution as the result of a Write Structured command containing a Start Display Program structured field, the first order to be executed

is found by applying the above rules as if a Branch Unconditional (GB) order had been executed, specifying the address in the display storage address register as the target address.

The first order to be executed upon receipt of a Set Buffer Address Register and Start command is found by applying the above rules as if a control mode Branch Unconditional (GB) order had been executed, specifying the address received with the command as the target address.

Base Order Set

The base order set for the 5080 Graphics System is divided into the following subsets:

1. Frame Control Orders
2. Output Primitive Orders
3. Segment Orders
4. Attribute Control Orders
5. Positional Device Control Order
6. Pick Control Orders
7. Branch Control Order
8. Stack Control Orders
9. Arithmetic Orders
10. Data Move Orders
11. No-Operation Orders
12. Area Fill Orders
13. Configuration Data Order

Frame Control Orders

The frame control orders control the rate of display program execution, set system initial conditions, set the proper system modes that apply to the overall display program, and terminate the program.

The graphics system supports three frame control orders. Two of the orders, Begin Order Processing (GBGOP) and End Order Processing (GEOP), are described in this section; the third, Start Regeneration Timer (GSRT), is a 3250-compatible order and is described on page A-12.

The following table briefly states the functions of the GBGOP and GEOP orders:

Mnemonic	Name	Function
GBGOP	Begin Order Processing	Begins execution of a display program
GEOP	End Order Processing	Terminates execution of a display program

GBGOP—Begin Order Processing

The GBGOP order begins execution of a display program. Its format is:

SM MC (Op Code)	MC Extension Word 1: Flags	MC Extension Word 2: Erase Protect Mask	Address Word (optional)
1	2	3	4

The GBGOP order contains the following values:

Name	Bit(s)	Value
Set Mode Mode Control	0–7 8–15	X'2A' or X'2E' X'8A'
Mode Control Extension Word 1: Flag Word	0–3 4 5–9 10–11 12 13 14–15	B'0000' B'd' B'00000' B'00' B'k' B's' B'tt'
Mode Control Extension Word 2: Erase Protect Mask	0–15	X'ee00'
Address Word (optional)	0–15	X'aaaa'

Note: The address word is present only if B's' = 1.

Mode Control Extension Word 1 (Flag Word) Variables

B'd' Frame buffer switching mode indicator.

d = 0 Frame buffer switching mode; switch frame buffer at GBGOP order time.

d = 1 Frame buffer nonswitch mode; do not switch frame buffer at GBGOP order time. Process the display program into the same frame buffer as the previous cycle and refresh the screen from the same frame buffer as before.

B'k' Denotes the enabled/disabled state of the keyboards: alphanumeric keyboard (ANK) and lighted program function keyboard (LPFK).

k = 1 The ANK and LPFK are disabled.

k = 0 The ANK and LPFK are enabled.

Note: Pressing the Reset key enables the keyboards.

B's' Manual input indicator: Set only if the 2 t bits and the address word are to be used. Note that, if this bit is set on, the fourth word is mandatory; the GBGOP order is 8 bytes long. If this bit is set off, the GBGOP order is only 6 bytes long and execution continues at the byte following the third word.

s = 0 The manual input function is not selected, the optional address word should not be present, and only the SM value X'2A' is valid.

s = 1 The manual input function is selected, the optional address word should be present, and the SM value can be either X'2A' or X'2E'.

B'tt' Denotes the type of interrupt key to be simulated.

tt = 00 No key; this is a test manual input
tt = 01 ENTER key
tt = 10 Cancel(Cncl) key
tt = 11 PF (program function) key

Mode Control Extension Word 2 (Erase Protect Mask) Variables

X'ee00' Denotes frame buffer bit plane erase protect. The e bits specify which frame buffer bit planes are protected from being cleared. Each e bit corresponds to a bit plane. The e bit 0 (high-order bit) corresponds to frame buffer bit plane 0; e bit 1 corresponds to frame buffer bit plane 1, etc.; e bit 7 corresponds to frame buffer bit plane 7. If an e bit is 1, the corresponding frame buffer bit plane is not cleared. If e bits are set to X'00', all frame buffer planes are cleared before the following graphic order is executed. If e bits are set to X'FF', no frame buffer bit planes are cleared.

Address Word

X'aaaa' Denotes a storage address (if s = 1):

- Of a word holding the value to be set into the last bytes of the *manual input (MI) register* (if tt is nonzero).
- To which control is transferred if the MI register is 'busy', awaiting a Read Manual Input command from the host system (if tt is zero).
- The address may be relative or absolute, depending on the SM byte option selected. If relative addressing is used, valid values are in the range from -32,768 to +32,766. Negative numbers are in twos complement. Absolute address values may be in the range from 0 to 65,534. The low-order bit of the address is ignored and assumed to be zero.

Although a GBGOP order should be used to begin execution of each display program, display programs that begin *without* a GBGOP order still run. Also, any display program that does not have a subsequent GBGOP order in its execution

will run continuously only for 30 seconds until a timeout occurs terminating the program execution (see "Display Program Termination" on page 3-10 for further details).

Execution of the GBGOP order (and subsequent graphic orders) is delayed to synchronize the switching of buffers with the refresh cycle.

The actions requested by the GBGOP order do not take effect if there is an outstanding Set Buffer Address Register and Start command when the GBGOP order begins execution. See "Interaction with Pick Indicators" on page 3-24 for a decision table representation of PI, PBI, SDI, and TSI indicator-setting rules.

The GBGOP order functions are:

1. Switch/Nonswitch Buffer Mode

   For a description of this mode, see "Display Programs" on page 3-2 and the d bit definition elsewhere in this section.

2. Frame Buffer Bit Planes Erase Protect

   For a description of this operation, see "Display Programs" on page 3-2 and the e bit definition elsewhere in this section.

3. GBGOP Keyboard Enable/Disable

   The keyboards attached to each graphics system workstation are locked or unlocked, depending on the state of the manual input register; in the case of the alphanumeric data keys, the keyboards are unlocked by the presence of a cursor in an unprotected input field. The keyboard enable/disable function provides another level of control whereby a display program can inhibit the keyboards. The default state, *enabled*, pertains after any of the following events:

   - Power-on, or system or selective reset.
   - Receipt of a Set Buffer Address Register and Start command or a Write Structured command containing a Start Display Program structured field for that device.
   - Execution of a Start Regeneration Timer (GSRT) order.
   - Execution of a GBGOP order with the k bit set to zero.

   Note: The Set Buffer Address Register and Stop command or a Stop Display Program structured field does not alter the enabled/disabled state of the keyboards.

   The disabled state is set only upon execution of a GBGOP order with the k bit set on.

   A GBGOP order that disables the keyboards does not accept further input for ENTER or Cancel keys while the keyboards are disabled, and it turns off the "clicker" if one of these keys is pressed. The keyboard disabled state can be reset (keyboard enabled) by pressing the Reset key.

4. GBGOP Test Manual Input (TMI)

   This GBGOP function is invoked only if the s bit is set on and the tt field is zero. It tests whether input from the interrupt-generating keys will be accepted or will be inhibited pending a reset of the manual input register. If the test indicates input would be accepted, execution continues at the next sequential order, that is, at the byte following the fourth word. If manual input could not be accepted immediately because the GBGOP order received a real keyboard interrupt in response to an earlier poll, or the MI register has not yet been cleared following an earlier keyboard interrupt (or simulated keyboard interrupt), control is transferred to the address specified in the fourth word.

5. GBGOP Set Manual Input (SMI)

   This GBGOP order function is invoked only if the s bit is set on and the tt field is nonzero. It sets the MI register and raises an Attention interrupt at the host system. This function simulates the pressing of an ENTER, Cancel, or LPFK key.

   Pressing an interrupt-generating key or executing the Set Manual Input function inhibits further keyed input until the MI register is reset. The register is reset by the Read Manual Input command, which also causes transmission of the 3 bytes of manual input data to the host system, or by the pressing of the Reset key.

   A GBGOP order incorporating a Set Manual Input request is not complete until the Set Manual Input request is satisfied. The graphics system is effectively stopped until the Set Manual Input request can be honored. The Set Manual Input request is unable to complete if the MI register is "busy," pending receipt of a Read Manual Input command from the host system; it completes when the MI register is cleared by that host command, at which time the new value is set into the MI register.

   If the 5085 is processing input (ENTER or Cancel key or an LPFK key) from a real keyboard, the MI register is busy. In this case, a GBGOP order containing a Set Manual Input request is held off until the real keyboard interrupt has been sent to the host and the Read Manual Input command has been received and has cleared the MI register.

   The Test Manual Input function of the GBGOP order and the keyboard disable function can be used to avoid long delays in servicing Set Manual Input requests.

   For Set Manual Input requests, the fourth word of the GBGOP order specifies the address of 2 bytes containing the value to be placed in the last 2 bytes of the MI register. The tt value is encoded into the first MI byte as follows:

   3-Byte Manual Input Register

ENTER	tt = 01	1010 0000 vvvv vvvv vvvv vvvv
Cancel	tt = 10	1001 0000 vvvv vvvv vvvv vvvv
LPFK	tt = 11	0100 0000 vvvv vvvv vvvv vvvv

   where

   v denotes the value taken from the 2 bytes at the specified address

When an interrupt-generating key is being simulated, the location addressed by the fourth byte of the GBGOP order should contain valid data for the key that is being simulated.

- GBGOP SMI ENTER Key Simulation

  To simulate the ENTER key, set tt = 01. For this version of SMI the manual input register is set to X'A0' concatenated with a copy of the 2-byte value at the address specified in the fourth word of the GBGOP order.

  Pressing the ENTER key results in a manual input register value of X'A00000'; this same value would be set if the value addressed by the GBGOP order were zero.

- GBGOP SMI Cancel Key Simulation

  To simulate the Cancel key, set tt = 10. For this version of SMI the manual input register is set to X'90' concatenated with a copy of the 2-byte value at the address specified in the fourth word of the GBGOP order.

  Pressing the Cancel key results in a manual input register value of X'900000'; this same value would be set if the value addressed by the GBGOP order were zero.

- GBGOP SMI LPFK Key Simulation

  To simulate a program function (PF) key, set tt = 11. For this version of SMI the manual input register is set to X'40' concatenated with a copy of the 2-byte value at the address specified in the fourth word of the GBGOP order.

  Pressing a PF key results in a manual input register value of X'40nnFF', where nn is the number of the pressed key in the range X'00' to X'1F'. To simulate the pressing of PF key 7, for example, the 2-byte data field addressed by the GBGOP order would contain the value X'07FF'.

  The second byte of manual input for a PF key interrupt defines which key was pressed. The key number, in the range 0 to 31, is expressed in 8-bit binary form. The third byte in the register is set to X'FF', as for real LPFK interrupts.

  Note: Only one SMI operation can be originated with the execution of a GBGOP order.

This order sets the condition code (cc), which may be tested by the Branch on Condition (GBC) order, to reflect the current pick indicator (PI) and tip switch indicator (TSI) settings.

cc = 00 PI off and TSI off (open)
cc = 01 PI off and TSI on (closed)
cc = 10 Not set
cc = 11 Not set

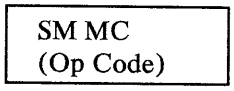
Note: The PI will be off after execution of the GBGOP order and the TSI will be set/reset to match the stylus switch.

Upon completion of the GBGOP order the attribute registers and stack registers are reset to their default values. See "GLATR—Load Attribute Register" on page 5-34 and "GLSR—Load Stack Register" on page 5-59.

GEOP—End Order Processing

The GEOP order terminates execution of a display program.

The format of the GEOP order is:



The GEOP order contains the following values:

Name	Bit(s)	Value
Set Mode	0–7	X'2A'
Mode Control	8–15	X'81'

This one-word order causes the Attention and Unit Check bits to be set in the status byte and the End Order Sequence bit to be set in sense byte 1. It results in an I/O interruption at the host CPU. The display storage address of the GEOP order plus 2 is saved as part of the sense information from the interrupting graphics system. (See Appendix C for definitions of the display storage address returned in the sense data.)

Output Primitive Orders

Output primitive orders are used to draw images on the graphics system display screen using the image coordinate system (see "Coordinate System Specification" on page 3-27). These orders generate pixel arrays, markers, lines, characters, and circles. The marker, line, and character orders (except for GDLA3L) are mode-setting orders; each order is followed by a data list. Data lists following marker and line orders contain one or more groups of coordinate values, as defined in "Specification of Coordinate Values" on page 3-29. Data lists following character orders contain EBCDIC characters and the null code.

For output primitive orders the color, grayshade, line type, blink, frame buffer mask, marker character, and pick attributes currently held in the attribute registers are applied to the image subject to the blanking bit where appropriate.

Note: The highlighting state and visibility state controls can also affect the output primitive orders (see "GLATR—Load Attribute Register" on page 5-34).

The graphic orders used for drawing images are described in this section in the sequence given in the following table. The table lists the mnemonic form and name of each order and presents a brief statement of the function of the order.

Mnemonic	Name	Function
GDPXL	Draw Pixel Array	Writes pixel data from display storage directly into the frame buffer.
GDMA2	Draw Marker Absolute 2D 12 Bits	Draws markers in two-dimensional (2D) space based on absolute X-Y coordinates.
GDLA2	Draw Line Absolute 2D 12 Bits	Draws lines in two-dimensional space based on absolute X-Y coordinates.
GDLR1	Draw Line Relative 1D 12 Bits	Draws lines in one-dimensional (1D) space based on relative X or Y coordinates.
GDLR2	Draw Line Relative 2D 12 Bits	Draws lines in two-dimensional space based on relative X-Y coordinates.
GDMR2	Draw Marker Relative 2D 12 Bits	Draws markers in two-dimensional space based on relative X-Y coordinates.
GDMR3	Draw Marker Relative 3D 12 bits	Draws markers in three-dimensional (3D) space using X, Y, and Z relative coordinates. If the Transformation and Clipping Feature (TCF) is not installed or 3D mode is not active, the Z coordinate is ignored during vector-to-raster conversion.

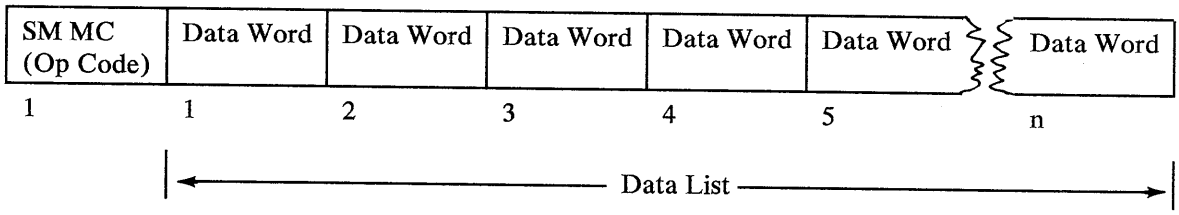
Mnemonic	Name	Function
GDLR3	Draw Line Relative 3D 12 bits	Draws lines in three-dimensional space using X, Y, and Z relative coordinates. If the Transformation and Clipping Feature (TCF) is not installed or 3D mode is not active, the Z coordinate is ignored during vector-to-raster conversion.
GDLA3L	Draw Line Absolute 3D 16 Bits	Draws lines in three-dimensional space based on absolute coordinates. If the Transformation and Clipping Feature (TCF) is not installed or 3D mode is not active, the Z coordinate is ignored during vector-to-raster conversion.
GDCHAR	Draw Character	Draws characters in any one of four character sizes if the character set is fixed or draws characters based on a programmable character set descriptor record.
GDCIR	Draw Circle	Draws a complete circle in world coordinate space.

Pixel Order

The pixel order allows the direct writing of pixel data from display storage into the frame buffer, thereby providing an image capability.

GDPXL—Draw Pixel Array

The GDPXL order writes pixel data directly from display storage into the frame buffer. This one-word order is followed by a variable-length data list. Its format is:



The GDPXL order contains the following values:

Name	Bit(s)	Value
Set Mode Mode Control	0–7 8–15	X'28' or X'2A' X'8D'
Data Word 1: Byte Count	0–15	X'0000'–X'FFFF'
Data Word 2: X Delta	0–2 3 4–15	B'000' B's' B'x...x'
Data Word 3: Y Delta	0–2 3 4–15	B'000' B's' B'y...y'
Data Word 4: Reserved	0–15	X'0000'
Data Word 5: Pixel Data	0–15	X'0000'–X'FFFF' (X'00'–X'FF' and X'00'–X'FF')
Data Word n: Pixel Data	0–15	X'0000'–X'FFFF' (X'00'–X'FF' and X'00'–X'FF')

Data Word 1 Variable

Byte Count — The total number of bytes of pixel data contained in data words 5 to n plus 8 (the count includes data words 1–4).

Data Word 2 Variables

B's' — Sign bit

s = 0 — The sign is positive.
s = 1 — The sign is negative.

B'x...x' — X delta

A 12-bit binary number representing an X delta from the *current draw position (CDP)*. If the sign is negative, twos complement notation should be used.

Data Word 3 Variables

B's' — Sign bit

s = 0 — The sign is positive.
s = 1 — The sign is negative.

B'y...y' — Y delta

A 12-bit binary number representing a Y delta from the CDP. If the sign is negative, twos complement notation must be used.

Data Word 4 Variable

X'0000' This word is reserved and must be set to X'0000'.

Data Words 5 to n Variables

X'0000'–X'FFFF' These words contain 2 bytes of pixel data each. The format of the pixel data is 8 bits (1 byte) per pixel. Valid values are in the range X'00' to X'FF'.

The CDP together with the X and Y deltas (which are arithmetically added to the CDP) define a rectangle in the virtual image space and hence, by truncation, a rectangular portion of the frame buffer. The pixel data (packed beginning in word 5) is loaded directly into the frame buffer in a horizontal direction beginning at the top left corner pixel of the area. When the right edge is reached, pixel loading continues from the left edge of the next row down.

The pixel data is loaded into the bit planes in the following manner:

Bit plane	0	1	2	3	4	5	6	7
Pixel data	x	x	x	x	x	x	x	x

If the number of bit planes installed is less than eight, the pixel data corresponding to the nonexistent bit plane will be ignored. In the case of six bit planes:

Bit plane	0	1	2	3	4	5	6	7
Pixel data	x	x	x	x	x	x	x	x

Ignored

A pixel bit for a given bit plane can be stored only if the frame buffer mask for that plane is on (attribute registers 5 and 27).

Pixel loading is terminated if the area becomes filled before the count runs out. However, the graphics system starts looking for the next order after the list terminates (as defined by the count).

Following execution of the GDPXL order, the current draw position registers (X-Y-Z) contain the old current draw position value with X and Y deltas added.

If the count runs out before the area is filled, the last pixel in the data list is replicated in the remaining locations.

The pixel array is in "primitive" form for the purposes of the pick function, that is, the *pick indicator (PI)* may be set during the GDPXL order and, thus, the condition code (cc) setting after the order reflects the PI and tip switch indicator (TSI) settings as follows:

cc = 00 PI off and TSI off (open)
cc = 01 PI off and TSI on (closed)
cc = 10 PI on and TSI off (open)
cc = 11 PI on and TSI on (closed)

Notes:

1. Although the rectangular area to be loaded is defined in virtual image space (4096 x 4096), the pixels themselves are loaded into the "real" frame buffer locations. If necessary, the rectangular perimeter is rounded "inward" to define these locations. When using the GDPXL order, the user must be aware of the "real" screen resolution.

2. If the rectangle lies across a screen boundary, it will be clipped at the edge of virtual image space.

3. If the GDPXL order spans page boundaries, the action that the graphics system will take is not predictable.

Mode-Setting Orders

Each mode-setting order is followed by a data list containing fields that determine the movement of the *current draw position (CDP)* in either virtual image space or world coordinate space, depending on the order and the TCF mode. The data list is terminated by the occurrence of any order (that is, X'28', X'2A', X'2C', or X'2E' in the high-order, or SM, byte of the first word of a coordinate group).

Each data field in a list contains X, Y, X-Y, or X-Y-Z coordinate values (formats 1, 2, 4, 7 and 14) defining either an addressable point (absolute coordinates) or a displacement relative to the CDP (relative coordinates). This coordinate, coordinate pair, or coordinate triplet determines the new point to which the line will be drawn or a move performed, that is, the new CDP. A flag within a data field (of formats 1, 2, 4, 7, or 14) determines whether the marker or line should be drawn (made visible). This flag is called the blanking bit.

The shape of the marker is specified by values contained in attribute registers 1 and 20. Attribute register 20 contains a character set ID (CSID) from which a graphic character is specified by an EBCDIC character code (which is used as the marker) in attribute register 1.

If attribute register 1 contains X'FF':

- For CSIDs X'00' and X'04', the marker character is a 3 x 3 pixel point centered at the location specified for the marker.

- For CSID X'06' the marker character is a 1-pixel point.

- For single byte programmable character sets (CSIDs X'08' to X'BF') the user may define the marker character (graphic symbol) for the X'FF' code point.

Note: The user should be aware that CSIDs X'C0' to X'FE' should not be used to specify a marker character set in attribute register 20 because they represent 2-byte programmable character sets and attribute register 1 can contain only a 1-byte value. If attribute register 20 contains X'C0' to X'FE', the system generates a hyphen for the marker character.

The default value for attribute register 1 is X'FF'; the default value for attribute register 20 is X'04'.

In marker drawing, a marker is displayed at the position defined by each coordinate group in the data list. After each marker in the list has been generated, the current draw position reflects the absolute coordinates of that marker.

In line drawing, a line is drawn from the current draw position to the position defined by the first coordinate group; then, the current draw position is updated to the new endpoint. This process is repeated for each following coordinate group in the data list, resulting in:

- A series of displayed lines that join the initial current draw position to the first point, and joining adjacent points in the list (if any).
- The current draw position updated to the last endpoint.

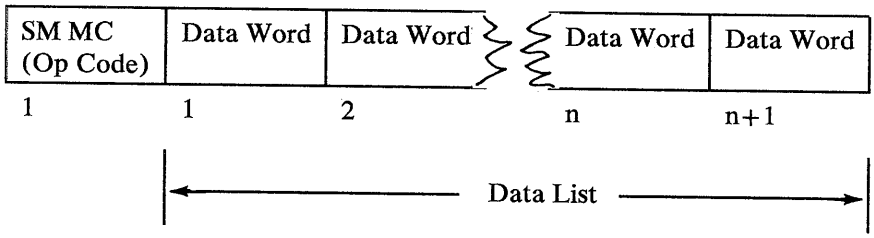
For line or marker drawing, if the blanking bit is on, the result is simply that the current draw position in the X-Y-Z position registers is updated.

Absolute Mode-Setting Orders

Two absolute drawing orders, Draw Marker Absolute 2D 12 Bits (GDMA2) and Draw Line Absolute 2D 12 Bits (GDLA2), enable the display of markers or lines, respectively, by specifying the absolute coordinates of the endpoint to move to/from the current draw position. The endpoint of moves and draws is in ±4096 addressable space (-4096 to +4095).

GDMA2—Draw Marker Absolute 2D 12 Bits

The format of the GDMA2 order is:



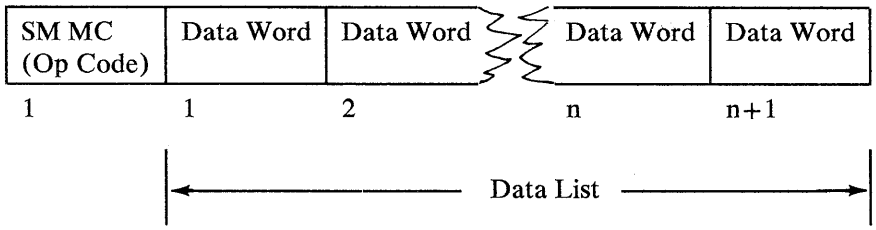
The GDMA2 order contains the following values:

Name	Bit(s)	Value
Set Mode	0–7	X'2A'
Mode Control	8–15	X'00'

This one-word order is followed by a data list containing an integral number of words. The data list comprises zero or more groups of coordinate values in format 4 (see "Specification of Coordinate Values" on page 3-29).

GDLA2—Draw Line Absolute 2D 12 Bits

The format of the GDLA2 order is:



The GDLA2 order contains the following values:

Name	Bit(s)	Value
Set Mode	0–7	X'2A'
Mode Control	8–15	X'02'

This one-word order is followed by a data list containing an integral number of words. The data list comprises zero or more groups of coordinate values in format 4 (see "Specification of Coordinate Values" on page 3-29).

Relative Mode-Setting Orders

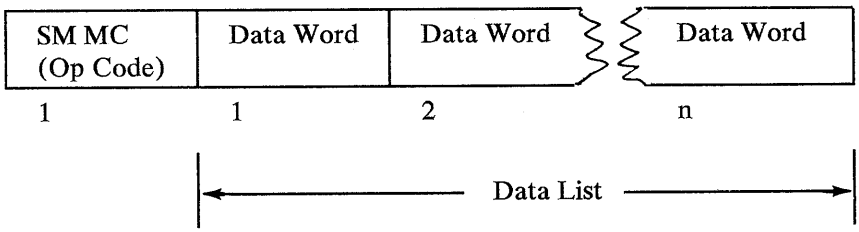
The Draw Line Relative 1D 12 bits (GDLR1), Draw Line Relative 2D 12 Bits (GDLR2), Draw Marker Relative 2D 12 Bits (GDMR2), Draw Marker Relative 3D 12 Bits (GDMR3), and Draw Line Relative 3D 12 Bits (GDLR3) orders enable lines or markers to be displayed by specifying the relative coordinates of the endpoint to which to move.

The GDMR2, GDLR1 and GDLR2 orders can be used in two-dimensional (2D) and three-dimensional (3D) modes. The values of the missing coordinates (X-Z, Y-Z, or Z) are the same as the values of the coordinates for the current draw position (that is, X-Y-Z position registers).

The GDMR3 and GDLR3 orders are provided to enable the drawing of markers and lines using relative coordinates in 3D space. If the Transformation and Clipping Feature is not installed, or, if installed and 3D mode is not active, the Z position register is updated but the Z value is ignored in the vector-to-raster conversion.

GDLR1—Draw Line Relative 1D 12 Bits

The format of the GDLR1 order is:



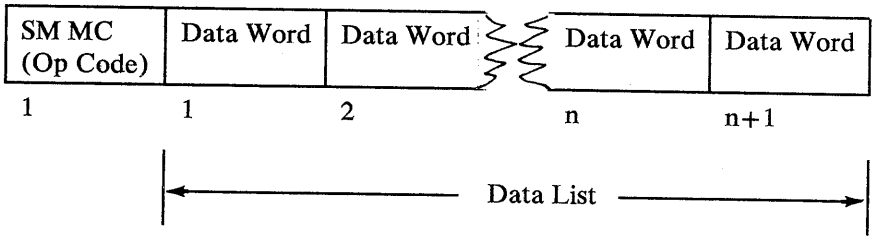
The GDLR1 order contains the following values:

Name	Bit(s)	Value
Set Mode	0–7	X'2A'
Mode Control	8–15	X'03'

This one-word order is followed by a data list containing an integral number of words. The data list comprises zero or more groups of coordinate values in format 1 or 2 (see "Specification of Coordinate Values" on page 3-29).

GDLR2—Draw Line Relative 2D 12 Bits

The format of the GDLR2 order is:



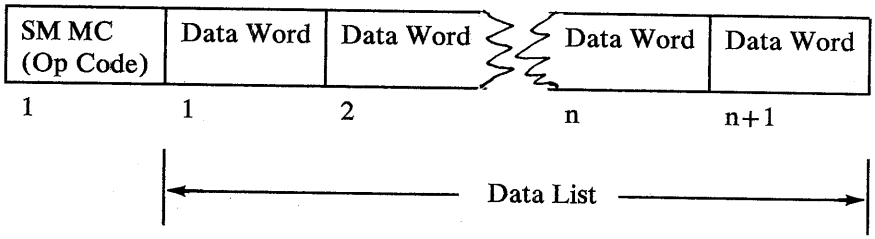
The GDLR2 order contains the following values:

Name	Bit(s)	Value
Set Mode	0–7	X'2A'
Mode Control	8–15	X'A2'

This one-word order is followed by a data list containing an integral number of words. The data list comprises zero or more groups of format 4 coordinate values (see "Specification of Coordinate Values" on page 3-29).

GDMR2—Draw Marker Relative 2D 12 Bits

The format of the GDMR2 order is:



The GDMR2 order contains the following values:

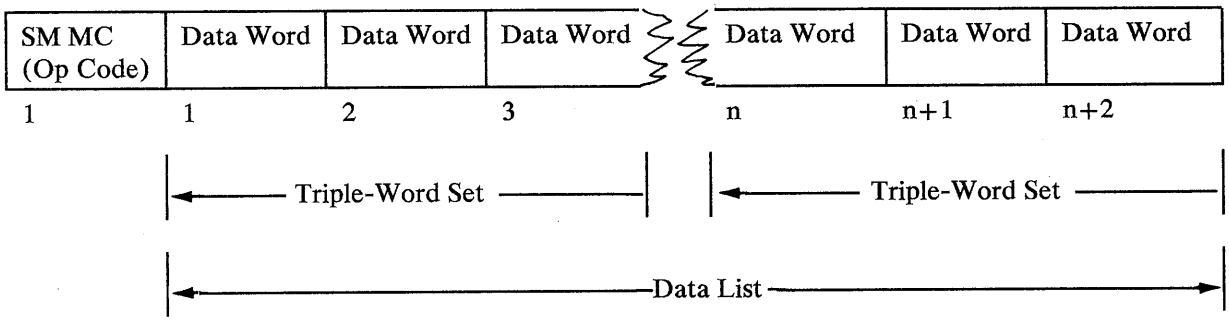
Name	Bit(s)	Value
Set Mode	0–7	X'2A'
Mode Control	8–15	X'A0'

This one-word order is followed by a data list containing an integral number of words. The data list following GDMR2 comprises zero or more groups of format 4 coordinate values (see "Specification of Coordinate Values" on page 3-29).

GDMR3—Draw Marker Relative 3D 12 Bits

The GDMR3 order enables marker drawing in three-dimensional mode, giving the ability to specify 12-bit signed-binary values of X-Y-Z coordinates. Displacements can be either positive or negative; if a negative displacement, the data is presented in twos complement form.

The format of the GDMR3 order is:



The GDMR3 order contains the following values:

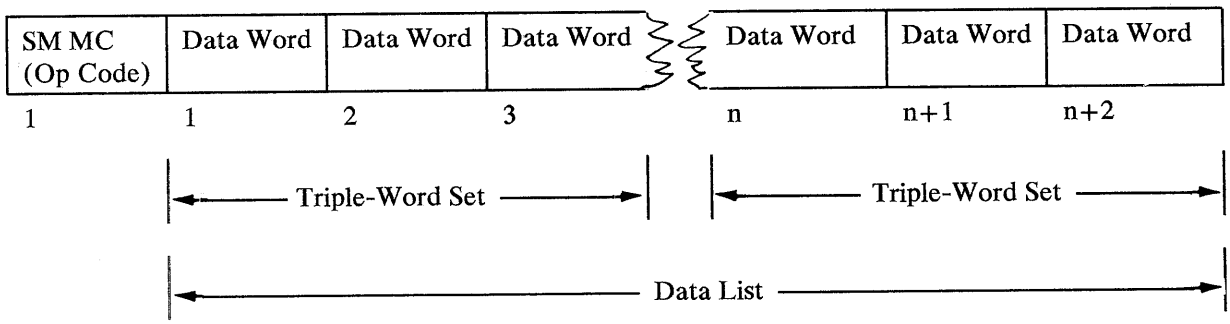
Name	Bit(s)	Value
Set Mode	0–7	X'2A'
Mode Control	8–15	X'A1'

The one-word GDMR3 order is followed by a data list containing an integral number of triple words. This data list contains zero or more graphic coordinate values in format 7 (see "Specification of Coordinate Values" on page 3-29). While the Z value is updated in the X-Y-Z position registers, the Z coordinate data is ignored when drawing the line if not in 3D mode.

GDLR3—Draw Line Relative 3D 12 Bits

The GDLR3 order enables line drawing in three-dimensional mode, giving the ability to specify 12-bit signed-binary values of X-Y-Z coordinates. Displacements can be either positive or negative; if a negative displacement, the data is presented in twos complement form.

The format of the GDLR3 order is:



The GDLR3 order contains the following values:

Name	Bit(s)	Value
Set Mode Mode Control	0–7 8–15	X'2A' X'A3'

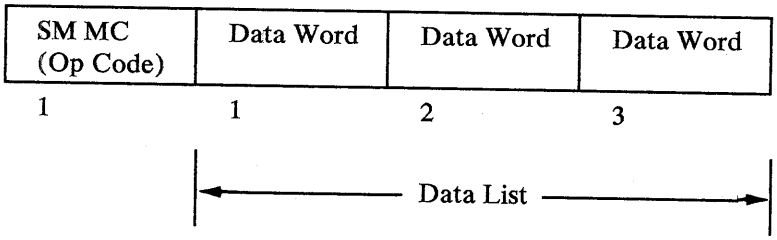
The one-word GDLR3 order is followed by a data list containing an integral number of triple words. This data list contains zero or more graphic coordinate values in format 7 (see "Specification of Coordinate Values" on page 3-29). While the Z value is updated in the X-Y-Z position registers, the Z coordinate data is ignored when drawing the line if not in 3D mode.

16-Bit Absolute Draw or Move Order

The Draw Line Absolute 3D 16 Bits (GDLA3L) order is used to draw or move from the current draw position to the position specified in the order. There is only one group of format 14 coordinate values per order (see "Specification of Coordinate Values" on page 3-29). The blanking bit is part of the order code (mode control field); when the bit is zero (X'B2'), the line is drawn, and when the bit is set to 1 (X'B3'), the line is blanked.

GDLA3L—Draw Line Absolute 3D 16 Bits

The one-word GDLA3L order is followed by a three-word data list in the following format:



The GDLA3L order contains the following values:

Name	Bit(s)	Value
Set Mode Mode Control	0–7 8–11 12–14 15	X'28' or X'2A' X'B' B'001' B'b'
Data Word 1: X Coordinate	0 1–15	B's' B'x...x'
Data Word 2: Y Coordinate	0 1–15	B's' B'y...y'
Data Word 3: Z Coordinate	0 1–15	B's' B'z...z'

Mode Control Variable

B'b' Blanking bit

b = 0 ('2AB2') The line is drawn to the position indicated by the coordinates in the data list and the current draw position is updated.

b = 1 ('2AB3') The line is blanked; the current draw position is updated, but a line is not displayed.

Data Word 1 Variables

B's' Sign bit

s = 0 The sign is positive.

s = 1 The sign is negative (negative numbers are in twos complement notation).

B'x...x' X coordinate; a 15-bit binary value.

Data Word 2 Variable

B's' Sign bit

s = 0 The sign is positive.

s = 1 The sign is negative (negative numbers are in twos complement notation).

B'y...y' Y coordinate; a 15-bit binary value.

Data Word 3 Variable

B's' Sign bit

s = 0 The sign is positive.

s = 1 The sign is negative (negative numbers are in twos complement notation).

B'z...z' Z coordinate; a 15-bit binary value. The Z coordinate is ignored if the 2D mode is operative.

The GDLA3L order specifies an absolute move (b = 1) or draw (b = 0) to the position indicated by the coordinates in the data list. The X, Y, and Z coordinates are in the range from -32768 to +32767. While the Z value is updated in the X-Y-Z position registers, the Z coordinate data is ignored when drawing the line if not in 3D mode.

Character Drawing Order

The Draw Character (GDCHAR) order is used to display characters when the character set identifier (CSID) is equal to X'04' or greater.

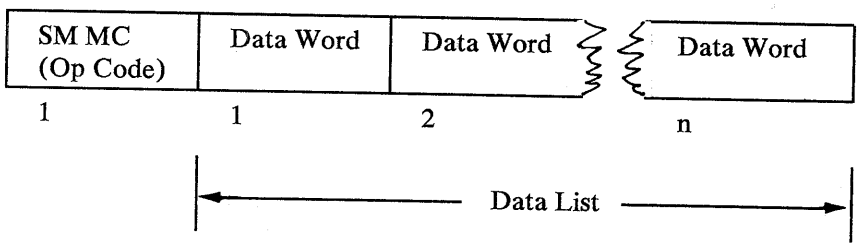
The GDCHAR order description that follows applies only if the CSID loaded into the CSID attribute register (register 18) using a Load Attribute Register

(GLATR) order is X'04' or greater. If the CSID loaded into attribute register 18 is X'00', refer to "GECM—Enter Character Mode" on page A-13 for information (CSIDs X'01', X'02' and X'03' are reserved).

Note: Attribute register 18 is defaulted to X'00' when either a Start Regeneration Timer (GSRT) order or a Set Buffer Address Register and Start command is executed by the graphics system. When a Begin Order Processing (GBGOP) order or a Start Display Program structured field is executed, the CSID in attribute register 18 is set to X'04'.

GDCHAR—Draw Character

The GDCHAR order is followed by a variable-length data list. Its format is:



The GDCHAR order contains the following values:

Name	Bit(s)	Value
Set Mode	0–7	X'2A'
Mode Control	8–11	X'4'
	12	B'0'
	13	B'p'
	14–15	B'00'

Mode Control Variable

B'p' Protection bit

p = 0 (X'2A40') The character field is unprotected.

p = 1 (X'2A44') The character field is protected against manual changes from the alphanumeric keyboard.

A GDCHAR order is followed by a data list of zero or more words of the form X'aabb', where: X'aa' are the first EBCDIC character bits and X'bb' are the second EBCDIC character bits, as shown below:

aaaa aaaa bbbb bbbb

The null character can be used to pad lists containing an odd number of characters to an integral number of words. The null character is not displayed nor does it cause movement of the current draw position.

A set mode (SM) code byte (X'28', X'2A', X'2C' and X'2E') may also be used in the right (low-order) byte of the last data list byte and terminates the character string. This SM code byte is not displayed, does not cause movement of the current draw position, and cannot be replaced from the keyboard.

The color/grayshade, blink, frame buffer mask, character path orientation, character rotation, character set identifier, highlighting color/grayshade, highlighting frame buffer mask, and pick attributes held in the attribute registers are applied to the display of these characters.

A character mode order determines whether the characters are protected against overwriting by manual changes from the alphanumeric keyboard.

Attribute register 18 must contain a valid CSID. Valid CSIDs are X'00' and X'04' to X'07' and X'08' to X'FE' if the description of the character set is already loaded into a *programmable character set (PCS)* type memory area. If attribute register 18 contains an invalid CSID, all character codes following the GDCHAR order are processed as null and the current draw position remains unchanged, unless the CSID is X'01' to X'03', in which case the result is indeterminate. If attribute register 18 does contain a valid CSID, the undefined characters are displayed as hyphens.

For CSIDs X'04' to X'07', code points X'01' to X'3F' and X'FF' are considered to be undefined and are displayed as hyphens; in addition, an undefined character code in the range X'40' to X'FE' is displayed as a space. For CSIDs X'08' to X'BF', those character codes not in the range of CP0 to CPN are not assigned a graphic character and are also considered to be undefined. For CSIDs X'C0' to X'FE' (2-byte PCSs), in addition to the conditions defined in CSIDs X'08' to X'BF', if a segment ID with a character code does not exist, that character code is also considered to be undefined. The codes X'28', X'2A', X'2C' and X'2E' may not appear within the data list, as they are recognized as terminating the data list.

Each displayable character (not null) occupies one character position on the screen and is drawn within a rectangle whose lower left corner is the current draw position; then, the current draw position is updated to locate the lower left corner of the next available character position (see Figure 5-1).

The orientation of the character box (the rectangle within which a character is drawn) and the direction of the line on which the next character is placed are specified by the contents of the character path/box orientation register (attribute register 17).

Character inline spacing, which is used to specify the number of the virtual pixels (units in -32K to +32K coordinate space) between the lower left corners of two successive character boxes, may be specified in attribute registers 29 and 30. If each register contains X'FF', the default spacing value of the CSID in effect is used.

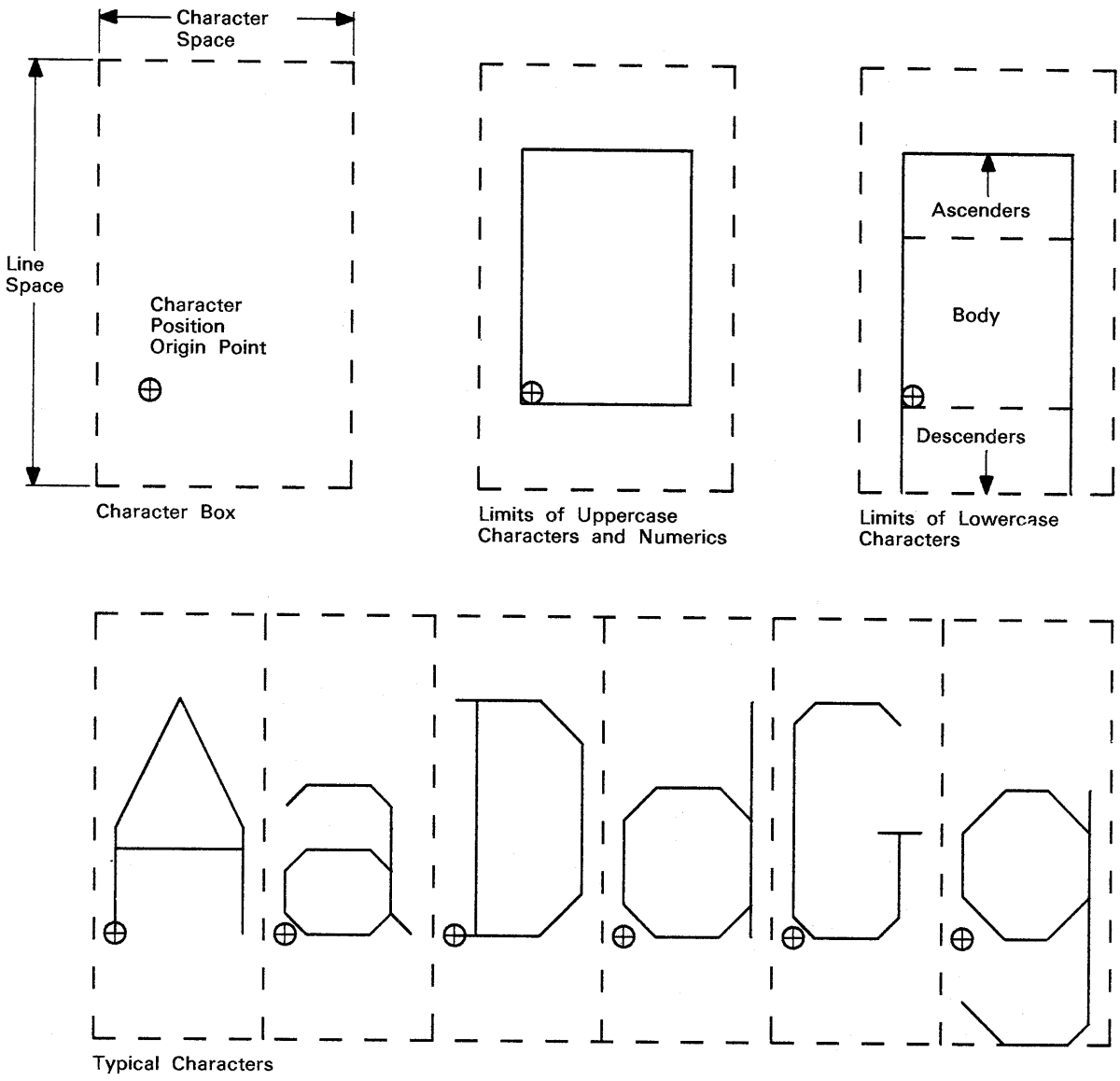


Figure 5-1. Character Box and Origin (CSID = X'04'–X'07')

The default value for character inline spacing is either the width or the height of the character box, depending on the value of the character box orientation. If the value of the character box orientation is X'0', the default value of the character inline spacing is the same as the width of the character box. If the value of the character box orientation is X'4', the default value of the character inline spacing is the same as the height of the character box. The sizes of the character boxes:

- For CSIDs X'00' and X'04' to X'07' are specified as part of the description of attribute register 18 under "GLATR—Load Attribute Register" on page 5-34.
- For CSIDs X'08' to X'FE' are specified in the P and Q fields of the PCS description under "Programmable Character Set Descriptor Record" on page 3-34.

A null character is not displayed; it can appear anywhere in the data list. Because the null character occupies a data byte, but has no effect on character positioning, it can be used for reserving, initializing, or filling out a string of character bytes to satisfy a boundary requirement.

The new-line and backspace character codes are not supported for nonzero CSIDs. If encountered, they are treated as invalid characters (displayed as hyphens).

Clipping of hardware-generated character strings (CSID = X'00' and X'04' to X'07') on a character basis can be accomplished by specifying viewport boundaries and TCF mapping mode. This function works whether or not TCF is installed in the machine. (See "TCF Character Mode Orders" on page 5-88 for details.)

When processing a GDCHAR order, the graphics system maintains logically correct X-Y-Z values for the current draw position, within the limits -32K to +32K, even if one or more characters remains outside the image area. An attempt to draw a character outside of these world coordinates results in an undefined current draw position.

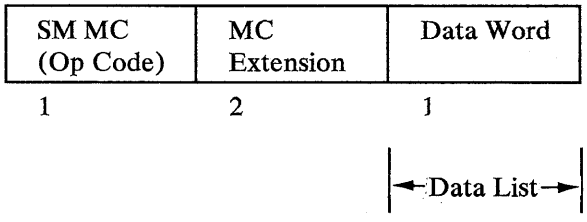
Circle Drawing Order

The graphics system provides an order to draw circles: Draw Circle (GDCIR).

GDCIR—Draw Circle

The two-word GDCIR order draws a complete circle (subject to the blanking bit) in world coordinate space. The magnitude of the radius of the circle is given by the number of pixels in the world coordinate space. The allowable range of values is from 0 to 32,767; the center of the circle is at the current draw position.

The GDCIR order is followed by a one-word data list. Its format is:



The GDCIR order contains the following values:

Name	Bit(s)	Value
Set Mode Mode Control	0–7 8–15	X'28' or X'2A' X'98'
Mode Control Extension Word	0–11 12–13 14 15	B'0...0' B'00' B'b' B'0'
Data Word: Radius	0 1–15	B'0' B'radius'

Mode Control Extension Word Variable

B'b' Blanking Bit

b = 0 Draw the circle.
b = 1 Blank the circle.

Data Word Variable

B'radius' The value of radius can be 0 to 32,767. The radius is always positive and, therefore, the sign is always zero.

The circle is a "primitive" graphic order and can be picked (if the pick modes allow), thus setting the PI. The condition code following this order is set to reflect the PI and TSI as:

cc = 00 PI off and TSI off (open)
cc = 01 PI off and TSI on (closed)
cc = 10 PI on and TSI off (open)
cc = 11 PI on and TSI on (closed)

The GDCIR order does not change the current draw position.

Segment Orders

Segments are a collection of logically related output parameters that can be manipulated as a whole; each segment is assigned a 2-byte name. Paired segment orders Begin Segment (GBGSEG) and End Segment (GESEG) delimit segments. The GBGSEG order resets the pick indicator (P=0 in register 13).

Segment nesting is permitted. For example, stack orders (GPATR, GPSMC, and GPOP) can be used to save and restore the prior segment(s) picking state attributes and any other information desired defining the state of the graphics system. To save the correct states, the stack orders are placed as follows:

```
GBGSEG      (Start base segment)
.
.
GPSMC       (Push stack marker)
GPATR       (Push attribute registers to stack)
GBGSEG    – Start nested segment
.
.
GESEG     – End nested segment
GPOP        (Retrieve base segment attributes from stack)
.
.
.
GESEG       (End base segment)
```

The information pushed into a stack is delimited by a stack marker being pushed in preceding any other data. A segment *must not* contain a Begin Order Processing (GBGOP) or Start Regeneration Timer (GSRT) order. Also, a nested segment may be called by many base segments. The number of nesting levels depends on the size of the stack (that is, the number of sets of graphics system state information that can be pushed into the stack before it is overflowed).

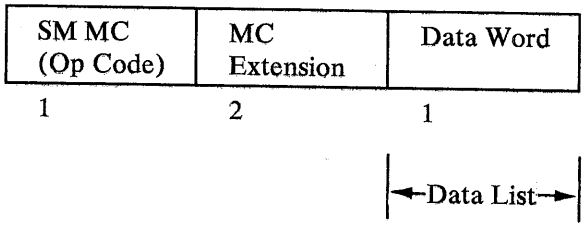
The two graphic orders used for segment control are described in this section. The following table lists the mnemonic form and name of each order along with a statement of function.

Mnemonic	Name	Function
GBGSEG	Begin Segment	Begins a segment
GESEG	End Segment	Ends a segment

Note: The GBGSEG order and the GESEG order comprise a bracketing pair that delimits the start and end of a segment. Therefore, there must be a terminating GESEG order for every opening GBGSEG order.

GBGSEG—Begin Segment

The GBGSEG order begins a segment. Its format is:



The GBGSEG order contains the following values:

Name	Bit(s)	Value
Set Mode Control	0–7 8–15	X'28' or X'2A' X'68'
Mode Control Extension Word (Reserved)	0–15	X'00'
Data Word	0–15	X'iiii'

Mode Control Extension Word

X'00' Reserved

Data Word Variable

X'iiii' This 16-bit data word contains the name of the segment

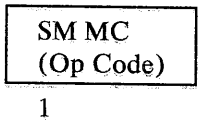
The pick indicator is reset by this order. Accordingly, the condition code after execution of this order always reflects the PI and TSI as follows:

cc = 00 PI off and TSI off (open)
cc = 01 PI off and TSI on (closed)
cc = 10 Not set
cc = 11 Not set

GESEG—End Segment

The GESEG order terminates a segment.

The format of the GESEG order is:



The GESEG order contains the following values:

Name	Bit(s)	Value
Set Mode Mode Control	0–7 8–15	X'2A' X'61'

The GESEG order updates the condition code register. After execution of this order the condition code always reflects the PI and TSI setting as:

cc = 00 PI off and TSI off (open)
cc = 01 PI off and TSI on (closed)
cc = 10 PI on and TSI off (open)
cc = 11 PI on and TSI on (closed)

Attribute Control Orders

The graphics system contains a number of attribute registers. Although many of these registers are reset to a default value by the Begin Order Processing (GBGOP) or Start Regeneration Timer (GSRT) order, they can be set and accessed under display program control using the Load Attribute Register (GLATR) and Store Attribute Register (GSATR) orders. The displayed color/grayshade is determined by the contents of an attribute register and a color/grayshade lookup table that is loaded through the Load Color Table (GLCT) order.

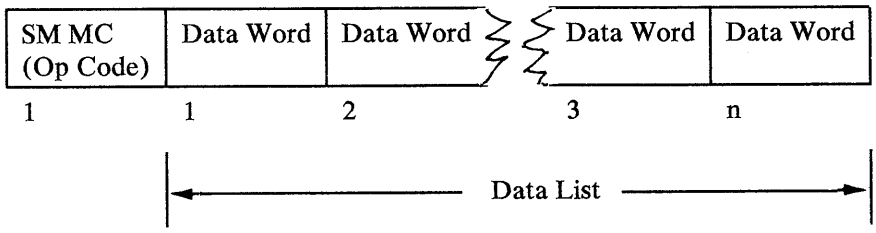
The three attribute control orders are described in this section in the sequence shown in the following table. The table lists the mnemonic form and name of each order along with a statement of function.

Mnemonic	Name	Function
GLCT	Load Color Table	Loads the Color Table according to loading instructions specified in a data list
GLATR	Load Attribute Register	Loads the attribute register(s) stipulated in a data list
GSATR	Store Attribute Register	Stores the specified attribute register(s) starting at the word specified by the address field

GLCT—Load Color Table

The GLCT order loads the Color Table (CT). This one-word order is followed by a variable-length data list.

The format of the GLCT order is:



The GLCT order contains the following values:

Name	Bit(s)	Value
Set Mode	0–7	X'28' or X'2A'
Mode Control	8–15	X'D0'

Data Word 1	t000 0000 aaaa aaaa	Start Address
Data Word 2	0000 000c cccc cccc	Count
	If Color Display	
Data Word 3	0000 bbbb gggg rrrr	Data Words
	. . .	. . .
Data Word n	0000 bbbb gggg rrrr	.
	If Monochrome Display	
Data Word 3	0000 ssss ssss ssss	Data Words
	. . .	. . .
Data Word n	0000 ssss ssss ssss	

Word 1 Variables

t — Tracking bit

t = 0 — Indicates that this is a normal Color Table load. The system resets to the destructive tracking symbol.

t = 1 — Indicates that this load of the Color Table is being performed to prepare for nondestructive tracking symbol use.

See "Tracking Symbol Update" and "Nondestructive Tracking Symbol" on pages 3-17 and 3-18, respectively.

aaaa aaaa — Specifies the location (entry) in the Color Table at which loading of the following data words is to start.

Word 2 Variables

c cccc cccc — Specifies the count (number) of Color Table entries to be loaded. Valid values are from 1 to 256.

If Color Display

Word 3 Variables

.
.
.
.
.

Word n

bbbb Specifies the amount of blue.
gggg Specifies the amount of green.
rrrr Specifies the amount of red that, in combination, will comprise the color that will be generated on the display when the Color Table entry into which the word is loaded is referenced via the color attribute register.

If Monochrome Display:

Word 3 Variables

.
.
.
.
.

Word n

ssss ssss ssss Specifies the grayshade that will be generated on the display when the Color Table entry into which the word is loaded is referenced via the color attribute register.

Although the physical size of the Color Table is 256 entries, the logical size available to the user depends on the number of bit planes featured on your system:

Number of Bit Planes	Logical Size of Table (Note 1)	Addressable Locations (Note 2)
2	4	0–3
4	16	0–15
6	64	0–63
8	256	0–255

Notes:

1. Represents the maximum count (cccccccc) value for a system featured with the specified number of bit planes.

2. Represents the valid addressable logical Color Table locations (that is, aaaa aaaa) for a system featured with the specified number of bit planes.

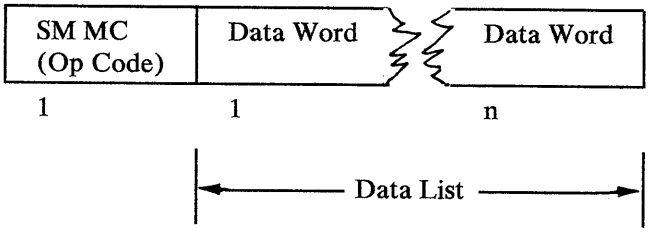
The data words (words 3 and onward) are loaded sequentially into the Color Table starting at the entry referred to by word 1. Loading continues until the count expires. All other locations of the Color Table remain unchanged. If the count has not expired at the time the logical end of the Color Table is reached, the remaining data words are ignored.

Execution of this order is synchronized with video refresh, so the Color Table loading is performed between refresh cycles.

GLATR—Load Attribute Register

The GLATR order loads the attribute register(s) specified in the data list; the order is followed by a variable-length data list containing one or more words.

The format of the GLATR order is:



The GLATR order contains the following values:

Name	Bit(s)	Value
Set Mode	0-7	X'28' or X'2A'
Mode Control	8-15	X'D3'
Data Word	0	B'e'
	1-2	B'00'
	3-7	B'ttttt'
	8-15	B'c...c'

Data Word Variables

B'e' Last-word indicator

e = 0 Not the last word; more words to follow.
e = 1 Indicates last word in data list.

B't...t' Attribute register number

B'c...c' Data bits to be set into the appropriate attribute register.

GLATR causes the attribute register(s) addressed in the data list to be loaded with the associated values contained in the c bits. Registers not referred to are not changed.

The user is cautioned to set all reserved c bits and registers to zero to avoid undefined results.

Each attribute register is 1 byte (8 bits) in length.

Formats, contents, lengths, and associated default values of the attribute registers are given in the following table.

Reg No.	Bit No.	Contents	No. of Valid/ Defined Bits	Format	Default Value
0	5–7	Blinking Pattern Identifier	3	0000 0bbb	B'000'
1		Marker Character Code	8	xxxx xxxx	X'FF'
2	4–7	Line Type	4	0000 1111	X'00'
3		Normal Color/Grayshade	8	xxxx xxxx	X'05'
4		Reserved	0		
5		Normal Frame Buffer Mask	8	xxxx xxxx	X'00'
6		Reserved	0		
7		Reserved	0		
8		Reserved	0		
9		Reserved	0		
10	0 1 7	Mode Switch Indicator (m) Normal Highlighting State (h) Visible/Invisible State (i)	1 1 1	mh00 000i	B'0' B'0' B'0'
11		Segment Name, High Byte	8	xxxx xxxx	X'00'
12		Segment Name, Low Byte	8	xxxx xxxx	X'00'
13	4 5 6 7	Pick Indicator (p) Single Detect Indicator (s) Permit Branch Indicator (e) Tip Switch Indicator (t)	1 1 1 1	0000 pset	B'0' B'x' B'x' B'x' (Note 1)
14	1 2–3	Pick Interrupt Mode (d) Pick Detect Mode (aa)	1 2	0daa 0000	B'1' B'11'
15		Reserved	0		
16		Reserved	0		
17	0–3 4–7	Character Path Orientation Character Box Orientation	4 4	iiii rrrr	X'00'
18		Character Set Identifier	8	xxxx xxxx	X'04' (Note 2)
19	2-6	Transformation and Clipping Feature Modes	5	00mp dtc0	X'00'
20		Marker Character Set Identifier	8	iiii iiii	X'04'
21		Reserved	0		
22		Reserved	0		
23		Reserved	0		
24		Reserved	0		
25		Highlighting Color/Grayshade	8	xxxx xxxx	X'05'
26		Reserved	0		
27		Highlighting Frame Buffer Mask	8	xxxx xxxx	X'00'
28		Reserved	0		
29		Character Inline Spacing, High Byte	8	xxxx xxxx	X'FF'
30		Character Inline Spacing, Low Byte	8	xxxx xxxx	X'FF'
31		Reserved	0		

Notes:

1. When attribute register 13 is reset, the TSI (t) bit is set to zero if the pick device (or if the device is in system pick device mode) stylus switch is open, and is set to 1 if the stylus switch is closed.

2. After a Begin Order Processing (GBGOP) order or a Start Display Program structured field is processed, the default value is set to CSID X'04'. After a Start Regeneration Timer (GSRT) order or a Set Buffer Address Register and Start command is processed, the default value is set to CSID X'00'.

The contents and function of each attribute register are described below. All bits set to zero are reserved for future use.

- Register 0—Blinking Pattern Identifier

  The 3 low-order bits of this register contain the blinking pattern identifier. Each blinking pattern identifier is assigned a 16-bit binary number default pattern:

bbbb bbbb bbbb bbbb

  where:

  b = 0 Image is blanked for approximately 1/4 second.

  b = 1 Image is displayed for approximately 1/4 second.

  Eight blinking patterns (0—7) are supported. The first two patterns (0—1) are system-defined and cannot be changed by the user. The remaining six patterns (2—7) can be changed by a Load Blinking Patterns structured field (see "Load Blinking Patterns" on page 4-22).

Pattern	Default Pattern	Comment
000	1111111111111111	No blinking pattern
001	1010101010101010	3250-compatible
010	1010101010101010	Loadable pattern
011	1010101010101010	Loadable pattern
100	1010101010101010	Loadable pattern
101	1010101010101010	Loadable pattern
110	1010101010101010	Loadable pattern
111	1100110011001100	Loadable pattern

  Note: If the regeneration cycle of the display program is long, the blink patterns may not take effect or may become distorted, because the system blink control mask is updated only at GBGOP or GSRT order execution time.

- Register 1—Marker Character Code

  This register contains the EBCDIC value of the character (or graphic symbol) in the marker character set, identified in attribute register 20, that is to be used as the marker. The default value is X'FF'; assignment of the character/graphic to this value is described in "Mode-Setting Orders" on page 5-17.

- Register 2—Line Type

  The 4 low-order bits of this register contain the identifier of a line pattern to be used when drawing lines and circles. Valid values are in the range 0 to 15. The pattern starts at the first unblanked line of a line drawing order. It continues to be repeated as the order and associated data list are processed; however, the pattern is restarted after a blank move is encountered.

  Each line pattern consists of a 256-bit (32 bytes) binary number. Each bit of the number represents:

  If = 0 Blank 16 virtual pixels.
  If = 1 Draw 16 virtual pixels.

  In this way each pattern can stipulate the drawing or blanking of a distance of between 0 and 4096 virtual pixels on the screen. The actual distance on the screen is the number of raster units nearest the specified distance (rounded off).

  Sixteen line patterns (0—15) are supported. The first four patterns (0—3) are system-defined and provide 3250-compatibility.

  Pattern 000 – Solid
  Pattern 001 – Dotted
  Pattern 010 – Dashed
  Pattern 011 – Dot-Dashed

  The remaining 12 patterns (4—15) can be changed by a Load Line Patterns structured field (see "Load Line Patterns" on page 4-23).

  Default values for line patterns 4 to 15 are all 1's (solid line).

- Register 3—Normal Color/Grayshade

  This register contains the color/grayshade index. It contains as many bits as there are bit planes in the frame buffer, starting from the low-order bit. The value in this register is set into the bit planes at each pixel location during any draw function (subject to the frame buffer mask). When the bit planes are scanned out to the video during image refresh, this value indexes the Color Table to determine the actual color/grayshade value placed on the screen at that pixel position.

- Register 4—Reserved

  This register is reserved for future use.

- Register 5—Normal Frame Buffer Mask

  This register controls writing into the frame buffer. Each bit in the register corresponds to a bit plane in the frame buffer in the same order as the color/grayshade attribute. If a bit is set to zero, writing in the corresponding bit plane is permitted; if the bit is set to 1, writing in that bit plane is inhibited (the existing contents remain unchanged).

- Registers 6, 7, 8, and 9—Reserved

These registers are reserved for future use.

- Register 10—Mode Switch Indicator and Normal Highlighting and Visibility States

  - Mode Switch Indicator (m)

    Bit 0 = 0 No mode switch has occurred.

    Bit 0 = 1 This bit is set to 1 by the system to indicate that a mode switch to 3270 mode or Setup mode occurred between the time the last GBGOP or GSRT order was executed and the display program was restarted. If your display program has generated a static image on the screen as a result of specifying the frame buffer nonswitch mode on the GBGOP order or by protecting a portion of the image using a frame buffer mask (see register 5), the static portion of the image will not reappear once the graphic mode is restored. Accordingly, if mode change is to be allowed by your application, make certain that you test this indicator when your display program is restarted, and, if it is on, regenerate the entire image. The mode switch indicator may be tested by storing register 10 and testing the display storage location using a Test under Mask (GTM) order.

  - Normal Highlighting State (h)

    Bit 1 = 0 Causes the normal color/grayshade and frame buffer mask (attribute registers 3 and 5) to be used in generating the display image.

    Bit 1 = 1 Causes the highlighting color/grayshade value in attribute register 25 and the highlighting frame buffer mask in attribute register 27 to be used in generating the display image.

  - Visible/Invisible State (i)

    Bit 7 = 0 Indicates that all lines, markers, circles, and so on, should be drawn and displayed.

    Bit 7 = 1 Indicates that all draws are to be invisible. When this bit is set, the entry of pixels into the frame buffer is prevented, thus blanking all draws. However, the current draw position in world coordinate space is still correctly updated when the i bit is on (i = 1).

- Registers 11 and 12—Segment Name

  These registers contain a 16-bit segment name that identifies the latest segment being processed. Register 11 contains the high byte of the segment name; register 12 contains the low byte.

These registers are updated only by the GBGSEG order. Placing this information in a register permits the user to save and restore segment states when segment nesting is required.

- Register 13—Pick Indicator (PI), Single Detect Indicator (SDI), Permit Branch (PBI) Indicator, and Tip Switch Indicator (TSI)

  This register contains the PI (in bit 4), the SDI (in bit 5), the PBI (in bit 6), and the TSI (in bit 7).

  The graphics system records changes of state of the PI, SDI, PBI, and TSI in this attribute register. Subsequently, the graphics system tests these indicators to control its processing flow. The user may also change the state of these indicators by setting appropriate bits in this register.

  x = 0 The specified indicator is off.
  x = 1 The specified indicator is on.

  Care should be taken when loading this attribute register because you may inadvertently override the state of one of the indicators.

  For a description of the setting and resetting of these indicators, see "Pick Detection Modes and Indicators" on page 3-20.

  Note: If the PI and the immediate interrupt mode are set at the exit from the GLATR order, a pick detect interrupt is generated immediately.

- Register 14—Pick Modes

  – Pick Interrupt Mode (d)

    When set to 1, bit 1 indicates immediate detect mode; when set to zero, bit 1 indicates deferred detect mode.

  – Pick Detect Mode (aa)

    Bits 2 and 3 indicate the pick detect mode.

    aa = 00 Reserved (not used)
    aa = 01 Disable detect
    aa = 10 Nonswitch enable detect
    aa = 11 Switch enable detect

    For an explanation of these modes, see "Pick Detection Modes and Indicators" on page 3-20.

- Register 15 and 16—Reserved

  These registers are reserved for future use.

- Register 17—Character Path and Character Box Orientation

  This register specifies the character path inline orientation and the character box orientation. These specifications are in terms of angle of direction. The value of the angle increases in a counterclockwise direction.

- Character Path Inline Orientation

  Character path inline orientation defines the direction for placing the next character on the line with respect to the current character. The 4 high-order bits (iiii) of attribute register 17 specify character path inline direction angle. Valid values for iiii are:

  X'0' Zero degrees, left to right horizontal; default value.
  X'4' 90 degrees, vertical up.
  X'8' 180 degrees, right to left horizontal.
  X'C' 270 degrees, vertical down.

  All unused values of iiii are reserved for future use.

- Character Box Orientation

  Character box orientation defines the direction of the rectangle within which the character is drawn with respect to the X axis of the graphics system coordinate system. The low-order 4 bits (rrrr) of attribute register 17 specify character box direction angle. Valid values for rrrr are:

  X'0' Zero degrees; character box width is parallel to the X-axis and character box height is parallel to the Y-axis; default value.

  X'4' 90 degrees; character box width is parallel to the Y-axis and character box height is parallel to the X-axis (character box is rotated 90 degrees counterclockwise).

  All unused values of rrrr are reserved for future use.

  Register 17 applies if CSID = X'04' or greater.

- Register 18—Character Set Identifier

  This register specifies the local character set identifier that defines the character set from which graphic characters are generated when the Draw Character (GDCHAR) order is used.

  A character set is designated by issuing a GLATR order, with register 18 specifying a character set ID. A character set is invoked if it is designated and a GDCHAR order is issued. If a programmable character set is invoked, before its description is loaded into a memory area all of the characters are treated as null.

CSIDs X'04' to X'07' are nonprogrammable and are the graphics system character sets. CSID X'04' is the system default character set after a Begin Order Processing (GBGOP) order has been processed. The assignment of local CSIDs is given below.

Character Set Local ID	Character Set Name	Character Box Size Width—Height
X'00'	3250-compatible base character set	(Note 1)
X'01' to X'03'	Reserved	
X'04'	System base character set—basic size, default	12–20 (48–80) (Note 2)
X'05'	System base character set—large	21–30 (84–120) (Note 2)
X'06'	System base character set—small	10–15 (40–60) (Note 2)
X'07'	System base character set—medium	18–25 (72–100) (Note 2)
X'08' to X'BF'	Programmable character sets (PCSs) with 1-byte character code	(Note 3)
X'C0' to X'FE'	Programmable character sets (PCSs) with 2-byte character code	(Note 3)
X'FF'	Reserved	

Notes:

1. See "GECM—Enter Character Mode" on page A-13 for the size of the character box.
2. The width and height are given in number of raster units for a 1024 x 1024 monitor. The numbers in parentheses are values of the width and height in virtual pixels.
3. Character box size is specified in "Programmable Character Set Descriptor Record" on page 3-34.

- Register 19—Transformation and Clipping Feature Modes

  – Mapping Mode (m)

    Bit 2 = 1 causes an image to be mapped from a window in world coordinates to a viewport in virtual image space coordinates; bit 2 = zero indicates mapping is turned off.

  – Perspective Mode (p)

    Bit 3 = 1 activates three-dimensional (3D) perspective; bit 3 = zero indicates perspective is not active.

Notes:

1. When this mode is on, the clipping volume is a truncated pyramid; when this mode is off, the clipping volume is a rectangular box.
2. This bit is ignored unless the 3D mode is set.

– 3D Mode (d)

Bit 4 = 1 specifies 3D transformation; bit 4 = zero specifies 2D transformation is active.

– Transformation Mode (t)

Bit 5 = 1 activates transformation; bit 5 = zero indicates transformation is inactive.

– Clipping Mode (c)

If 3D mode (d = 1): bit 6 = 1 indicates that clipping to the clipping rectangle box or the truncated pyramid (depending on perspective mode) is active; bit 6 = zero indicates that clipping is to the default boundary.

If 2D mode (d = 0): bit 6 = 1 indicates that clipping rectangle is active; bit 6 = zero indicates that clipping is to the default boundary.

See "TCF Modes of Operation" on page 3-41 for additional details.

- Register 20–Marker Character Set Identifier

This register specifies an 8-bit marker character set identifier. The character code defined in register 1 specifies a graphic character from the character set identified by this register.

Note: If the CSID is zero, the character set/size is basic.

- Registers 21, 22, 23 and 24–Reserved

These registers are reserved for future use.

- Register 25–Highlighting Color/Grayshade

This register contains a color/grayshade index into the Color Table that is used only when the system is in the highlighting state (see attribute register 10, h bit). See the description of attribute register 3 for the function of the color/grayshade index.

- Register 26–Reserved

This register is reserved for future use.

- Register 27–Highlighting Frame Buffer Mask

This register contains a frame buffer mask that is used only when the system is in the highlighting state (see register 10, h bit). (See the description of attribute register 5 for details on the function of the frame buffer mask.)

- Register 28—Reserved

  This register is reserved for future use.

- Registers 29 and 30—Character Inline Spacing

  These registers specify a 16-bit positive character inline spacing value, which is the number of virtual pixels between the lower left corners of two successive character boxes. If these registers contain X'FFFF', the default value of the character set in use is used. The default values for those character sets provided by the graphics system (CSIDs X'00' and X'04' to X'07') are defined in the description of attribute register 18; for PCSs, the contents of the P and Q fields are used. See the P and Q field descriptions under "Programmable Character Set Descriptor Record" on page 3-34.

- Register 31—Reserved

  This register is reserved for future use.

Attribute Register General Information

The conditions under which the attribute registers are reset to their default values are:

- At power-on, and system and device reset.
- By a Begin Order Processing (GBGOP) order.
- By a Start Regeneration Timer (GSRT) order.
- By a Set Buffer Address Register and Start command from the host system.

Because the PI may be changed during a GLATR order, the condition code setting following the order reflects the PI and TSI settings as:

cc = 00 PI off and TSI off (open)
cc = 01 PI off and TSI on (closed)
cc = 10 PI on and TSI off (open)
cc = 11 PI on and TSI on (closed)

GSATR—Store Attribute Register

The GSATR order stores the current value of the attribute register(s) specified beginning at the word identified by the address field of the order.

The format of the GSATR order is:

SM MC (Op Code)	MC Extension Word 1	MC Extension Word 2	Address Word
1	2	3	4

The GSATR order contains the following values:

Name	Bit(s)	Value
Set Mode Mode Control	0–7 8–15	X'2A' and X'2E' X'D4'
Mode Control Extension Word 1	0–15	X'mmmm'
Mode Control Extension Word 2	0–15	X'mmmm'
Address Word	0–15	X'aaaa'

The second and third words of this four-word order contain a two-word MC extension. The fourth word contains an address field.

Mode Control Extension Words 1 and 2

The m bits in the MC extension words form a mask that controls which attributes are stored by the order. Bit zero (high order) of the first MC extension word corresponds to attribute register 0, bit 1 corresponds to attribute register 1, and so on; bit 15 of the second MC extension word corresponds to attribute register 31. If the corresponding m bit is 1, the attribute is stored; if the m bit is zero, that attribute is not stored.

Address Word Variable

X'aaaa' Denotes a display storage address within the current page (that is, a word boundary). The address may be relative or absolute, depending on the SM byte option selected. If relative addressing is used, valid values are in the range from -32,768 to +32,766. Negative numbers are in twos complement. Absolute address values may be in the range from 0 to 65,534. The low-order bit of the address is ignored and assumed to be zero.

Executing the GSATR order causes the data to be stored starting at the word identified by the address field of the order. The data is stored on a word boundary and comprises the contents of the attribute registers defined by the m bits in the GSATR order. Each attribute is stored as one word, each containing the 5 t bits and the 8 low-order bits described under the GLATR order. The attribute registers are stored in ascending order in contiguous display storage locations. The last attribute word stored has the e bit set to 1 to denote the end of the data field.

Note: The stored data list varies in length and is compatible with the GLATR order.

Positional Device Control Order

The graphic order used for controlling positional devices is the Store Device Input (GSDEVI) order. It is described in this section.

Mnemonic	Name	Function
GSDEVI	Store Device Input	Stores the output of an addressed device at a user-specified location

GSDEVI—Store Device Input

The GSDEVI order stores two or more words containing the output of an addressed device, beginning at the address given in the order.

SM MC (Op Code)	MC Extension: Flag Word	Address Word
1	2	3

The GSDEVI order contains the following values:

Name	Bit(s)	Value
Set Mode Mode Control	0–7 8–15	X'2A' or X'2E' X'E8'
Mode Control Extension: Flags Word	0–3 4–7 8 9 10 11 12 13 14–15	B'0000' B'uuuu' B'i' B't' B'b' B'1' B'f' B'r' B'00'
Address Word	0–15	X'aaaa'

Mode Control Extension Word Variables

B'uuuu' 4-bit/device position register number

uuuu	= X'0'	Current draw position registers (X,Y,Z)
uuuu	= X'1'	Tablet registers
uuuu	= X'2'	Dials feature registers
uuuu	= X'3' to X'F'	Reserved

B'i' = 0 Do not interrupt

B'i' = 1 Enable interrupt from position entry device, unit number B'0001'. An interrupt occurs immediately if a switch or one or more function keys exists and was closed or pressed on the addressed unit at the time of the last Begin Order Processing (GBGOP) or Start Regeneration Timer (GSRT) order (for example, tablet stylus switch/function keys for unit B'0001').

Note: An interrupt occurs only if the SDI is off. The SDI is set on if an interrupt is taken from the GSDEVI order and is reset only when all stylus switches are open (at GBGOP or GSRT order time). For a full description of the set/reset, and function of the SDI, see "Pick Detection Modes and Indicators" on page 3-20.

An interrupt generates Attention, Unit Check status. The sense data contains the unit number and the display storage address of the GSDEVI order that caused the interrupt.

B't' Tip switch indicator (valid only for tablet)

The t bit is shown only to maintain format compatibility with the GSPOS order described in the 3250 RPQ, *Attachment for Cursor Control Tablet and Plotter,* GA33-3120. The t bit is ignored by the 5085 processor.

B'b' Blanking flag

b = 0 The data is stored without the blanking bit set.

b = 1 The data is stored with the appropriate blanking bit set if l = 0.

B'l' Long format. This bit is significant only for unit B'0000'.

l = 0 The current draw position is stored in base X-Y form in two words in the format:

0b00 xxxx xxxx xxxx	X Coordinate
0000 yyyy yyyy yyyy	Y Coordinate

l = 1 The current draw position is stored in long X-Y-Z form in three words in the format:

sxxx xxxx xxxx xxxx	X Coordinate
syyy yyyy yyyy yyyy	Y Coordinate
szzz zzzz zzzz zzzz	Z Coordinate

Note: The long form (l=1) is compatible with the data list following a Draw Line Absolute 3D 16 Bits (GDLA3L) order.

B'f' Tablet function key identifier.

f = 0 The 16-bit mask word is not stored.

f = 1 The first word stored is a 16-bit function key mask word; that is, for each bit in this word set to 1, the corresponding function key was pressed at the time of the last GBGOP or GSRT order. Function key 1 is represented by the rightmost bit of the mask, function key 2 by the bit to the left of the rightmost bit and so on.

B'r' Dials reset

r = 0 All dial values and the corresponding mask bits are reset to positive zero after the execution of this order.

r = 1 The dial values are not reset after execution of this order and subsequent issuance of the order will result in the retrieval of the cumulative relative value of each dial since the last time they were reset (that is, signed addition or subtraction of each dial's input is performed on the dials' cumulative total as the values are received by the system). This order retrieves the net result of these calculations at the time it is issued.

Address Word Variable

X'aaaa' Denotes the starting display storage address within the current page (that is, a word boundary) of an area into which the specified device's input is to be stored. The address may be relative or absolute, depending on the SM byte option selected. If relative addressing is used, valid values are in the range from -32,768 to +32,766. Negative numbers are in twos complement. Absolute address values may be in the range from 0 to 65,534. The low-order bit of the address is ignored and assumed to be zero.

This three-word order causes two or more words to be stored beginning at the address given in the order.

The device data is stored according to device type as follows:

Device Type	Device No.	Mode Control	Data Length	Word No.	Content	Format
Current Draw Position	0000	l=0	2 words	0	X	0b00 xxxx xxxx xxxx
				1	Y	0000 yyyy yyyy yyyy
		l=1	3 words	0	X	sxxx xxxx xxxx xxxx
				1	Y	syyy yyyy yyyy yyyy
				2	Z	szzz yyyy yyyy yyyy
Tablet	0001	f=0 l=0	2 words	0	X	0b00 xxxx xxxx xxxx
				1	Y	0000 yyyy yyyy yyyy
		f=1 l=0	3 words	0	Mask	mmmm mmmm mmmm mmmm
				1	X	0b00 xxxx xxxx xxxx
				2	Y	0000 yyyy yyyy yyyy

Note: If the tablet is "out of presence" (that is, the stylus is above the minimum height for sensing or outside the active area), no data is stored, the switch is considered to be open, and the condition code is set to 10. Function key 1 (or the stylus) is represented by the rightmost bit of the mask, function key 2 by the bit to the left of the rightmost bit, and so on.

Dials Feature	0010		9 words	0	Mask	mmmm mmmm 0000 0000
				1	Dial Scalar	sddd dddd dddd dddd
				2	Dial Scalar	sddd dddd dddd dddd
				3	Dial Scalar	sddd dddd dddd dddd
				4	Dial Scalar	sddd dddd dddd dddd
				5	Dial Scalar	sddd dddd dddd dddd
				6	Dial Scalar	sddd dddd dddd dddd
				7	Dial Scalar	sddd dddd dddd dddd
				8	Dial Scalar	sddd dddd dddd dddd
						(d = Dial Scalar Value)

Note: A mask bit is set to 1 to indicate the value of the corresponding dial has changed since the previous GSDEVI order with the unit address of B'0010'. Dial 1 is represented by the leftmost bit of the mask, Dial 2 by the second bit, and so on. The last GSDEVI order may not be in the same GBGOP cycle. The mask bits are cleared to X'0000' after the mask word and the eight-word dials output data is stored at the location specified by the address field of the GSDEVI order.

The dials feature output value is a signed binary number representing the relative amount of change in the dial positions since the last time the dial values were reset (refer to r bit definition). Negative values are presented in twos complement notation. The dial output value is relative to the dial setting at the time the dial values were last reset, not necessarily to dial position zero. If more than 127 cumulative full turns (360 degrees) of a dial are made in either the positive or negative direction, that dial's output value is reset to a positive zero.

Notes:

1. A subsequent GSDEVI order to the same unit (in the same display program execution cycle) may give different data. (The switch indicator remains the same for the duration of the GBGOP order cycle.)

2. If the addressed device is not installed, no data is stored and the condition code is set = 11.

The GSDEVI order sets the condition code (which may be tested by the Branch on Condition order) to reflect the status of the referenced unit:

cc	=	00	Switch open or unit has no switch (that is, dials)
cc	=	01	Switch closed
cc	=	10	Out of presence (tablet)
cc	=	11	Device not installed

Pick Control Orders

The pick control orders that control the graphics system pick function are described in this section in the sequence given in the following table. The table lists the mnemonic form and name of each order and briefly states the function of each.

Mnemonic	Name	Function
GBSO	Branch on Switch Open	Branches control to specified address on switch-open conditions
GBDD	Branch on Deferred Detect	Branches control to specified address on deferred detects
GBND	Branch on No-Detect	Branches control to specified address on no-detect conditions

In addition to the three orders listed above, the Load Attribute Register (GLATR) order is also used to set pick attributes.

GBSO—Branch on Switch Open

The GBSO order branches control to a specified address on a switch-open condition.

The format of the GBSO order is:

SM MC (Op Code)	Address Word
1	2

The GBSO order contains the following values:

Name	Bit(s)	Value
Set Mode	0–7	X'2A' or X'2E'
Mode Control	8–15	X'F5'
Address Word	0–15	X'aaaa'

Address Word Variable

X'aaaa' Denotes a display storage address within the current page (that is, a word boundary). The address may be relative or absolute, depending on the SM byte option selected. If relative addressing is used, valid values are in the range from -32,768 to +32,766. Negative numbers are in twos complement. Absolute address values may be in the range from 0 to 65,534. The low-order bit of the address is ignored and assumed to be zero.

This two-word order:

- Branches control to the specified address if the TSI is open (t = 0 in attribute register 13).

- Continues execution at the next order if the TSI is closed.
- Is not affected by the state of the pick detect or interrupt modes.

The condition code settings are not affected by execution of this order.

GBDD—Branch on Deferred Detect

The GBDD order branches control to a specified address on a deferred detect.

The format of the GBDD order is:

SM MC (Op Code)	Address Word
1	2

The GBDD order contains the following values:

Name	Bit(s)	Value
Set Mode Mode Control	0–7 8–15	X'2A' or X'2E' X'FC'
Address Word	0–15	X'aaaa'

Address Word Variable

X'aaaa' Denotes a display storage address within the current page (that is, a word boundary). The address may be relative or absolute, depending on the SM byte option selected. If relative addressing is used, valid values are in the range from -32,768 to +32,766. Negative numbers are in twos complement. Absolute address values may be in the range from 0 to 65,534. The low-order bit of the address is ignored and assumed to be zero.

This two-word order:

- Sets the PI off.
- Branches control to the specified address if the PI was on prior to execution of this order.

The GBDD order branches control to the address specified in the last word of the order if the PI is on; that is, if a deferred detect is outstanding.

The pick detect and interrupt modes are not altered, nor are the SDI, TSI, or PBI changed.

The condition code setting following execution of the GBDD order reflects the PI and TSI settings as follows:

cc = 00 PI off and TSI off (open)
cc = 01 PI off and TSI on (closed)
cc = 10 Not set
cc = 11 Not set

GBND—Branch on No-Detect

The GBND order branches control to a specified address on a no-detect condition.

The format of the GBND order is:

SM MC (Op Code)	Address Word
1	2

The GBND order contains the following values:

Name	Bit(s)	Value
Set Mode Mode Control	0–7 8–15	X'2A' or X'2E' X'FD'
Address Word	0–15	X'aaaa'

Address Word Variable

X'aaaa' Denotes a display storage address within the current page (that is, a word boundary). The address may be relative or absolute, depending on the SM byte option selected. If relative addressing is used, valid values are in the range from -32,768 to +32,766. Negative numbers are in twos complement. Absolute address values may be in the range from 0 to 65,534. The low-order bit of the address is ignored and assumed to be zero.

This two-word order branches control to the specified address in either of the following two cases:

- Pick detect mode is in the switch enabled state, the PI is off and the PBI is on.
- Pick detect mode is in the nonswitch enabled state and the PI is off.

In all other cases, execution continues at the next order and the PI is set off.

Whether or not the branch occurs, this order sets the PBI off if the pick detect mode is in the switch enabled state.

The condition code setting after execution of the GBND order reflects the PI and TSI settings as:

cc = 00 PI off and TSI off (open)
cc = 01 PI off and TSI on (closed)
cc = 10 Not set
cc = 11 Not set

Branch Control Orders

A branch control order causes a transfer of control to a new address in display storage. (See "Order Sequencing" on page 5-5 for the technique used to find the next executable order.)

The four branch control orders are described in this section in the sequence shown in the following table. The table lists the mnemonic form and name of each order and gives a short statement of the function of each.

Mnemonic	Name	Function
GBC	Branch on Condition	Branches control to specified address based on condition code settings
GBCNT	Branch on Count	Branches control to specified address if count is nonzero after being decremented
GB	Branch Unconditional	Branches control unconditionally to specified address
GBPAGE	Branch Page	Branches to address specified on another page

GBC—Branch on Condition

The GBC order branches control to a specified address based on the condition code setting.

The format of the GBC order is:

SM MC (Op Code)	Address Word
1	2

The GBC order contains the following values:

Name	Bit(s)	Value
Set Mode Mode Control	0–7 8–11 12 13 14–15	X'2A' or X'2E' X'7' B'u' B'i' B'cc'
Address Word	0–15	X'aaaa'

Mode Control Variables

B'u'

u=0 The 2 low-order bits (cc) of the MC byte are compared with the setting of the condition code bits, and subsequent action is taken according to the setting of the i bit.

u=1 The branch is taken unconditionally to the location indicated by the address word and the setting of the i bit is ignored.

B'i'

i=0 If the condition is not satisfied (that is, cc is not equal to the condition code), a branch is taken to the location indicated by the address word; otherwise, the next sequential order is executed (branch if condition is not true).

i=1 If the condition is satisfied (that is, cc equals the condition code), a branch is taken to the location indicated by the address word; otherwise, the next sequential order is executed (branch if condition true).

X'aaaa' Denotes the display storage address within the current page (that is, a word boundary) to which a branch will be made if the specified conditions are satisfied. The address may be relative or absolute, depending on the SM byte option selected. If relative addressing is used, valid values are in the range from -32,768 to +32,766. Negative numbers are in twos complement. Absolute address values may be in the range from 0 to 65,534. The low-order bit of the address is ignored and assumed to be zero.

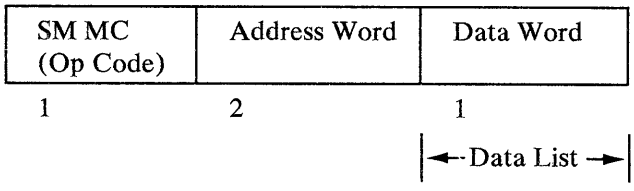
For a description of the conditions that set the condition code, see "Condition Code Register" on page 3-7.

The GBC order does *not* change the condition code setting.

GBCNT—Branch on Count

The GBCNT order branches control to a specified address if the count is nonzero after being decremented.

The format of the GBCNT order is:



The GBCNT order contains the following values:

Name	Bit(s)	Value
Set Mode	0–7	X'28' or X'2A' or X'2C' or X'2E'
Mode Control	8–15	X'F0'
Address Word	0–15	X'aaaa'
Data Word	0–15	X'cccc'

Address Word Variable

X'aaaa' Denotes a display storage address within the current page (that is, a word boundary). Control passes to this address unless the count field is zero after being decremented. The address may be relative or absolute, depending on the SM byte option selected. If relative addressing is used, valid values are in the range from -32,768 to +32,766. Negative numbers are in twos complement. Absolute address values may be in the range from 0 to 65,534. The low-order bit of the address is ignored and assumed to be zero.

Data Word Variable

X'cccc' Denotes a count value, an unsigned 16-bit integer. The count field is overwritten by the decremented value at instruction execution time.

Note: If the count field is zero before being decremented, the count is decremented to X'FFFF' and this order continues to execute as a transfer and count instruction.

GB—Branch Unconditional

The GB order branches control unconditionally to a specified address.

The format of the GB order is:

SM MC (Op Code)	Address Word
1	2

The GB order contains the following values:

Name	Bit(s)	Value
Set Mode Mode Control	0–7 8–15	X'2A' or X'2E' X'FF'
Address Word	0–15	X'aaaa'

This two-word order causes a branch of control to the location identified by the address in the last word of the order.

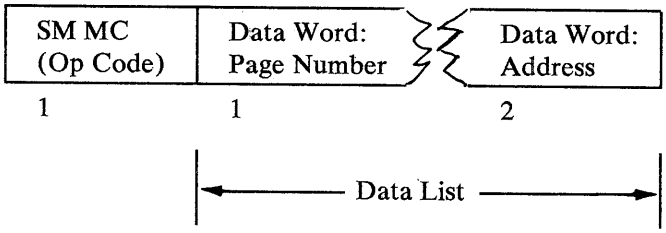
Address Word Variable

X'aaaa' Denotes a display storage address within the current page (that is, a word boundary). The address may be relative or absolute, depending on the SM byte option selected. If relative addressing is used, valid values are in the range from -32,768 to +32,766. Negative numbers are in twos complement. Absolute address values may be in the range from 0 to 65,534. The low-order bit of the address is ignored and assumed to be zero.

GBPAGE—Branch Page

The GBPAGE order branches to an address on another page.

The format of the GBPAGE order is:



The GBPAGE order contains the following values:

Name	Bit(s)	Value
Set Mode Mode Control	0–7 8–15	X'28' or X'2A' X'F7'
Data Word 1: Page Number	0–15	X'pppp'
Data Word 2: Address	0–15	X'aaaa'

Data Word 1 Variable: Page Number

X'pppp' This word contains the number of the page to which you want to jump.

The 5080 Graphics System supports a maximum of 17 pages; accordingly, valid page number values are in the range X'0000' to X'0010'. However, the number of pages supported by your graphics system may be less, depending on the amount of storage installed, and may vary from time to time, depending on whether a memory area storage pool is currently defined and, if it is, how large it is.

You may calculate the number of pages available at any time on your graphics system and, by doing so, the valid page number values, by issuing a Store Configuration Data (GSCONF) order and reading back the configuration data stored in the display storage. Then, from parameter 2 (graphics system storage size) extract the number of 2K blocks of storage assigned to the display storage. Divide this number by 32; the quotient gives you the number of full 64K pages available for display storage use and the remainder, if not zero, indicates that you have an additional (last) page of 2K to 64K. (See "Memory Areas" on page 2-6 for details.)

An attempt to branch to an unavailable page results in a display program error.

Contact your system programmer for details on page numbers available to your application.

Data Word 2 Variable: Address

X'aaaa' This 16-bit word represents an absolute address within the page (specified in X'pppp' above) to which you want to branch.

Valid values are in the range from 0 to 65,534. The low-order bit of the address is ignored.

The GBPAGE order allows the display program to branch to an address on another page.

Stack Control Orders

Nine stack orders are defined that specify the size and location of the stack and control the pushing and popping of data to and from the stack. This section describes these stack orders and the stack format. The stack is used to store orders requested by the stack control orders.

The stack control orders are described in the order shown in the following table. For each order the table indicates mnemonic form and name plus a brief statement of the function of the order.

Note: The internal operation of the stack is described in this section to assist the user in debugging only. The internal operation is subject to change in the future and should not be relied on to remain unchanged by using this knowledge to perform programming, etc.

Mnemonic	Name	Function
GLSR	Load Stack Register	Specifies stack characteristics and initializes the stack
GSSR	Store Stack Register	Stores stack register data at a specified address
GBAPL	Branch after Push Link	Writes a Branch Page (GBPAGE) order into the stack and branches to a page/byte address
GBAPLS	Branch after Push Link Short	Writes a Branch Page (GBPAGE) order into the stack and branches to a byte address
GPATR	Push Attribute Register	Writes a Load Attribute Register (GLATR) order into the stack
GPCDP	Push Current Draw Position	Writes a Draw Line Absolute 3D 16 Bits (GDLA3L) order into the stack
GPSMC	Push Stack Marker Code	Writes a stack marker (X'2AB0') into the stack
GPOPNOP	Pop Stack and No-Operation	Removes orders from the stack without processing them
GPOP	Pop Stack	Reads orders from the top of the stack

Before information can be pushed into a stack, a Load Stack Register (GLSR) order is issued to specify the stack characteristics and initialize the stack. The stack is initialized by writing X'2AFE0000' at the display storage address defined by the stack initial pointer, and setting the stack current pointer to the stack initial pointer plus 2 if registers 4 and 5 are not specified; otherwise, the values of registers 4 and 5 are used as the stack current pointer.

An empty stack is signified by a stack current pointer addressing a length field of X'0000'.

Each push order writes an appropriate order into the stack by:

- Incrementing the stack current pointer by 2.
- Writing the order at the location specified by the stack current pointer.
- Writing X'2AFE' after the order.
- Writing the length of the order in the next 2 bytes.

Assuming prior stack initialization and push processes, a nonempty stack has the format:

2AFE0000 (Order, 2AFE, Length of Order)...(Order, 2AFE, Length of Order)

The bracketed information is repeated for each order pushed into the stack. The stack current pointer always points to the last length field in the stack.

A pop operation reads and processes the order on top of the stack. The first byte of this order is located using the stack current pointer location and the contents of the length field to which it is pointing. Following each pop operation the current pointer points to the length field following the order that is on top of the stack. The pop operation is then repeated for the new order at the top of the stack unless it is a stack marker or a GBPAGE order placed there by a Push Stack Marker order or a Branch after Push Link or a Branch after Push Link Short order, respectively. If the current pointer is at a length field containing X'0000', and a pop order is used, a stack error is issued, since the stack is assumed to be empty.

The exception conditions that cause a stack error are:

- A pop operation is requested while the stack is empty (the stack current pointer points to a length field of zero).

- During the processing of a push order, the stack current pointer becomes equal to or greater than the stack limit pointer.

If a stack error occurs, the graphic order program is terminated, the host is interrupted, and Attention/Unit Check bits are presented. Sense byte 1, bit 7 and byte 4, bit 5 are set to 1; sense bytes 6 and 7, 8 and 9, 2 and 3, and 12 and 13 contain the stack page number, stack initial pointer, stack current pointer, and stack limit pointer, respectively.

Note: Processing orders other than those created and pushed into the stack by push orders (defined both elsewhere in this section and in "TCF Stack Control Orders" on page 5-101) yields unpredictable results. Processing of this sort

would occur only as a result of a user error, such as setting the stack current pointer incorrectly or moving data or orders into the stack area through the use of a Move Data Block (GMVBLK) order, and so on, with an incorrect address.

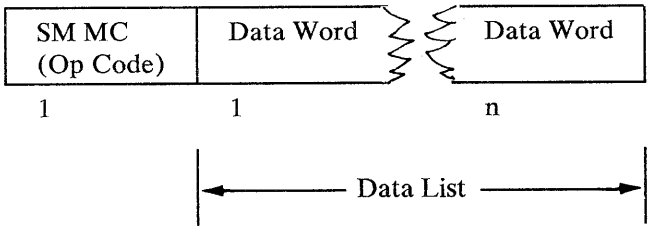
GLSR—Load Stack Register

The GLSR order specifies the size, location and current pointer of the stack, and initializes the stack. The size and location are defined by the stack initial pointer and stack limit pointer. These pointers, along with the stack current pointer, define the display storage addresses within the page specified by the stack location page number. Because the stack cannot span pages, all pointers must have the same page number. The stack is initialized by writing X'2AFE0000' at the location defined by the stack initial pointer.

If neither register 4 or 5 is specified, the stack current pointer (registers 4 and 5) is set equal to the stack initial pointer (registers 2 and 3) plus 2. Otherwise, if either register 4 or 5 is specified, the stack current pointer is set to the value of registers 4 and 5.

The GLSR order is followed by a variable-length data list containing one or more words.

The format of the GLSR order is:



The GLSR order contains the following values:

Name	Bit(s)	Value
Set Mode Mode Control	0–7 8–15	X'28' or X'2A' X'D5'
Data Word 1	0 1–3 4 5–7 8–15	B'e' B'000' B'0' B'ttt' B'c...c'

Data Word Variables

B'e' Last-word indicator

e = 0 Not the last word; more words to follow.
e = 1 Last word in data list.

B'ttt' Stack register number

B'c...c' Data bits to be set into the appropriate stack register.

This order causes the stack registers indicated in the data list to be loaded with the values specified by the associated c bits. Registers not referred to are not changed.

Data List

The formats, contents, and associated default values of the stack registers are:

Stack Reg No.	Contents	No. of Bits	Format	Value at GBGOP and Start
0	Stack Location Page Number, High Byte	8	xxxx xxxx	X'00'
1	Stack Location Page Number, Low Byte	8	xxxx xxxx	X'00'
2	Stack Initial Pointer, High Byte	8	xxxx xxxx	X'00'
3	Stack Initial Pointer, Low Byte	8	xxxx xxxx	X'00'
4	Stack Current Pointer, High Byte	8	xxxx xxxx	X'00'
5	Stack Current Pointer, Low Byte	8	xxxx xxxx	X'00'
6	Stack Limit Pointer, High Byte	8	xxxx xxxx	X'00'
7	Stack Limit Pointer, Low Byte	8	xxxx xxxx	X'00'

The contents and function of each stack register are described below:

- Registers 0 and 1—Stack Location Page Number

  These registers contain a 16-bit page number (the high-order 8 bits are in register 0 and the low-order 8 bits are in register 1) of the page in which the stack is located. The stack cannot span pages. See "Display Storage Page Numbering" on page 3-2 for information on how to determine valid page numbers for your processor.

- Registers 2 and 3—Stack Initial Pointer

  These registers contain the 16-bit stack initial pointer (the high-order 8 bits are in register 2 and the low-order 8 bits are in register 3) specifying the lower limit of the stack address space within the page specified in stack registers 0 and 1.

- Registers 4 and 5—Stack Current Pointer

  These registers contain the 16-bit stack current pointer (the high-order 8 bits are in register 4 and the low-order 8 bits are in register 5) that is used to locate the order on top of the stack. This pointer is always the address of a length field containing the length of the order on top of the stack. An empty stack is signified by a length field containing X'0000'. If a pop operation is requested when the stack is empty, the operation is not executed; the display program is terminated and a stack error is indicated.

- Registers 6 and 7—Stack Limit Pointer

  These registers contain the 16-bit stack limit pointer (the high-order 8 bits are in register 6 and the low-order 8 bits are in register 7) that specifies the upper limit of the stack address space. It is the user's responsibility to define enough

display storage space for stack manipulation, including the initial X'2AFE0000'. If the stack current pointer becomes equal to or greater than the stack limit pointer as a result of a push operation, the push operation is not executed; the display program is terminated and a stack error is indicated.

Note: Valid pointer values represent a 16-bit address within the page specified in stack registers 0 and 1, which may be in the range from 0 to 65,534. The low-order bit of the address is ignored and assumed to be zero.

The conditions under which the stack registers are reset to their default values are:

- At power-on, or system and device reset.
- By a Begin Order Processing (GBGOP) order.
- By a Start Regeneration Timer (GSRT) order.
- By a Set Buffer Address Register and Start command from the host system.

Note: Resetting the stack registers at GBGOP or GSRT order time effectively disables the stack (that is, no stack is present).

GSSR—Store Stack Register

The GSSR order stores stack register data starting at the word specified in the address field. The second and third words of this four-word order are MC extension words. The fourth word contains an address field; bit 15 in the address word is the least significant bit and is ignored and assumed to be zero.

The format of the GSSR order is:

SM MC (Op Code)	MC Extension Word 1	MC Extension Word 2	Address Word
1	2	3	4

The GSSR order contains the following values:

Name	Bit(s)	Value
Set Mode Mode Control	0–7 8–15	X'2A' or X'2E' X'D6'
Mode Control Extension Word 1	0–7 8–15	X'mm' X'00'
Mode Control Extension Word 2	0–15	X'0000'
Address Word	0–15	X'aaaa'

Mode Control Extension Word 1 Variable

X'mm' The m bits in the first MC extension word form a mask that controls which stack registers are stored by the order. The m bit 0 (high order) of the first MC extension word corresponds to stack register 0, m bit 1 corresponds to stack register 1, and so on. If the corresponding m bit is 1, the register is stored; if the corresponding m bit is zero, that register is not stored.

Mode Control Extension Word 2 Variable

X'0000' This word is reserved and should always be set to X'0000'.

Address Word Variable

X'aaaa' Executing GSSR causes the data to be stored beginning at the word identified by the address word of the order. The data is stored on a word boundary and comprises the contents of the stack registers defined by the m bits in the GLSR order. Each register is stored as one word, each containing the 3 t bits and 8 low-order bits as described in "GLSR—Load Stack Register" on page 5-59. The last word stored has the e bit set to 1 to denote the end of the data field.

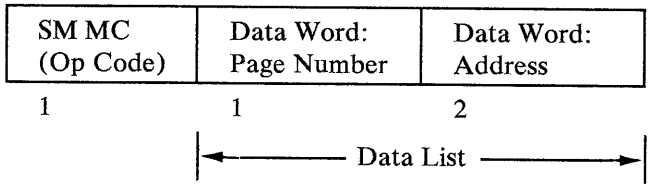
The address is a display storage address within the current page (that is, a word boundary). The address may be relative or absolute, depending on the SM byte option selected. If relative addressing is used, valid values are in the range from -32,768 to +32,766. Negative numbers are in twos complement. Absolute address values may be in the range from 0 to 65,534. The low-order bit of the address is ignored and assumed to be zero.

Note: The stored data list is variable in length and is compatible with the GLSR order.

GBAPL—Branch after Push Link

The GBAPL order pushes a Branch Page (GBPAGE) order into the stack.

The format of the GBAPL order is:



The GBAPL order contains the following values:

Name	Bit(s)	Value
Set Mode Mode Control	0–7 8–15	X'28' or X'2A' X'27'
Data Word 1: Page Number	0–15	X'pppp'
Data Word 2: Address	0–15	X'aaaa'

This order pushes (writes) into the stack a Branch Page (GBPAGE) order followed by X'2AFE0006'. The stack current pointer is set to point to the length field. The page number and address field in the GBPAGE order are set to point to the order following the GBAPL order. Control is transferred to the display storage page and byte address specified by the page number and address words of the GBAPL order. Branching to an unavailable page results in a graphic order program error (see Appendix C for definitions of the sense data).

Data Word 1 Variable: Page Number

This is a 16-bit number representing the display storage page to which control is transferred upon execution of this order. Refer to "Display Storage Page Numbering" on page 3-2 for information on how to determine valid page numbers for your processor.

Data Word 2 Variable: Address Word

This 16-bit word contains an absolute address within the page (specified in page number) to which control is transferred upon execution of the GBAPL order.

Valid values are in the range from 0 to 65,534. The low-order bit of the address is ignored and assumed to be zero.

During the push operation, if the address in the stack pointer register is equal to or greater than that in the stack limit register, the GBAPL order is not executed; the display program is terminated and a stack error is indicated.

GBAPLS—Branch and Push Link Short

The function of the GBAPLS order is the same as the GBAPL order. It also pushes a GBPAGE order followed by X'2AFE0006' into the stack. However, the GBPAGE order branch address is limited to 2 bytes in length, may be relative or absolute and does not include a page number. GBAPLS is used for branching within a page without crossing the page boundary. In addition, the GBAPLS order supports only immediate data, whereas the GBAPL order supports both immediate and indirect data.

The format of the GBAPLS order is:

SM MC (Op Code)	Address Word
1	2

The GBAPLS order contains the following values:

Name	Bit(s)	Value
Set Mode Mode Control	0–7 8–15	X'2A' or X'2E' X'28'
Address Word	0–15	X'aaaa'

Address Word Variable

X'aaaa' Denotes a display storage address within the current page (that is, a word boundary). The address may be relative or absolute, depending on the SM byte option selected. If relative addressing is used, valid values are in the range from -32,768 to +32,766. Negative numbers are in twos complement. Absolute address values may be in the range from 0 to 65,534. The low-order bit of the address is ignored and assumed to be zero.

GPATR—Push Attribute Register

The three-word GPATR order pushes (writes) into the stack a GLATR order containing the attribute registers specified in the mask word. The GLATR order is followed by X'2AFEnnnn', where nnnn is a word containing the length of the GLATR order.

The format of the GPATR order is:

SM MC (Op Code)	MC Extension Word 1	MC Extension Word 2
1	2	3

The GPATR order contains the following values:

Name	Bit(s)	Value
Set Mode Mode Control	0–7 8–15	X'2A' X'2B'
Mode Control Extension Word 1	0–15	X'mmmm'
Mode Control Extension Word 2	0–15	X'mmmm'

The m bits form a mask word wherein each m bit corresponds to an attribute register in the same way as in the Store Attribute Register (GSATR) order.

During the push operation, if the address in the stack current pointer register is equal to or greater than that in the stack limit register, the GPATR order is not executed; the display program is terminated and a stack error is indicated.

GPCDP—Push Current Draw Position

The one-word GPCDP order pushes (writes) into the stack a Draw Line Absolute 3D 16 Bits (GDLA3L) order (MC = X'B3') followed by X'2AFE0008'. The stack current pointer is set to point to the length field.

The X-Y-Z values specified in the Draw Line Absolute 3D 16 Bits (GDLA3L) order are the coordinates of the current draw position in coordinate value group format 14 (see "Specification of Coordinate Values" on page 3-29) with the blanking bit on. The blanking bit of the GDLA3L order is set to 1.

The format of the GPCDP order is:

SM MC (Op Code)
1

The GPCDP order contains the following values:

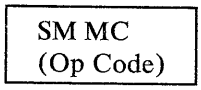
Name	Bit(s)	Value
Set Mode	0–7	X'2A'
Mode Control	8–15	X'2D'

During the push operation, if the address in the stack current pointer register is equal to or greater than that in the stack limit register, the GPCDP order is not executed; the display program is terminated and a stack error is indicated.

GPSMC—Push Stack Marker Code

The one-word GPSMC order pushes (writes) into the stack a stack marker (X'2AB0') followed by X'2AFE0002'. The stack current pointer is set to point to the length field.

The format of the GPSMC order is:



The GPSMC order contains the following values:

Name	Bit(s)	Value
Set Mode	0–7	X'2A'
Mode Control	8–15	X'B1'

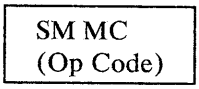
The stack marker delimits a set of orders in the stack. It should be pushed into the stack when the stack is being used to save the graphics system state before the processing of a nested segment is started. During the processing of orders in a stack, the stack marker (X'2AB0') is treated as a 2-Byte No-Operation (GNOP2) order, but terminates a pop operation, causing normal processing to resume at the next address after the GPOP order.

During the push operation, if the address in the stack current pointer register is equal to or greater than that in the stack limit register, the GPSMC order is not executed; the display program is terminated and a stack error is indicated.

GPOPNOP—Pop Stack and No-Operation

The one-word GPOPNOP order pops orders from the top of the stack without processing them until a stack marker or a Branch Page (GBPAGE) order is found.

The format of the GPOPNOP order is:



The GPOPNOP order contains the following values:

Name	Bit(s)	Value
Set Mode	0–7	X'2A'
Mode Control	8–15	X'A4'

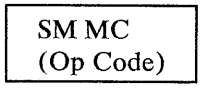
The GPOPNOP order pops (reads) the orders from the top of the stack and discards them. The orders on the stack read by this order are not processed until a stack marker (X'2AB0') or a GBPAGE order is encountered and discarded. Normal order execution is then resumed at the next order (address) following the GPOPNOP order.

If, at any time before encountering the stack marker or GBPAGE order, the stack current pointer points to a length field containing X'0000', the display program is terminated and a stack error is indicated.

GPOP—Pop Stack

The one-word GPOP order pops (reads) the orders from the top of the stack and processes them until a Branch Page (GBPAGE) order or stack marker (X'2AB0') is encountered. After the GBPAGE order or stack marker is processed, normal processing is resumed starting at the address contained within the GBPAGE order. Following the processing of the stack marker, normal processing resumes at the next address following the GPOP order.

The format of the GPOP order is:



The GPOP order contains the following values:

Name	Bit(s)	Value
Set Mode	0–7	X'2A'
Mode Control	8–15	X'2F'

If at any time before encountering the GBPAGE order or stack marker the stack current pointer points to a length field containing X'0000', the display program is terminated and a stack error is indicated.

If a GLSR order is encountered during the processing of a popped order, the GLSR order is not processed, the display program is terminated and a stack error is indicated.

Note: Processing orders other than those created and pushed into the stack by the push orders (defined elsewhere in this section and in "TCF Stack Control Orders" on page 5-101) yields unpredictable results.

Arithmetic Orders

The six arithmetic orders described in this section are listed in the following table in the order in which they are described. For each order the mnemonic form and name are given along with a brief statement of function.

Mnemonic	Name	Function
GADD	Add Data	Adds the contents of a data word to the contents of the word pointed to by the address word.
GSUB	Subtract Data	Subtracts the contents of a data word from the contents of the word pointed to by the address word.
GDIV	Divide Data	Divides the contents of the location pointed to by the address word (the dividend) by the contents of the data word (the divisor).
GCOMP	Compare Data	Compares the contents of a data word with the contents of the word pointed to by the address word.
GSHIFT	Shift Data	Shifts a data field of either 16 or 32 bits.
GTM	Test under Mask	Checks one-for-one correspondence between mask bits and bits of the word pointed to by the address word.

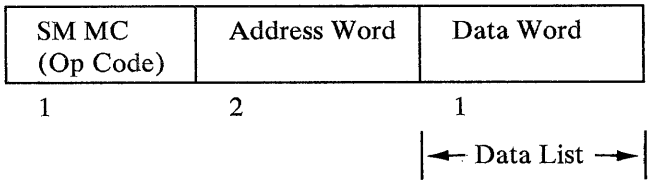
GADD—Add Data

The GADD order adds two 16-bit signed integers.

In the GADD order the second operand (data word) is added to the contents of the first operand location (address), and the sum is placed in the first operand location.

Both operands are considered to be 16-bit signed integers (negative numbers are expressed in twos complement, signed 15-bit binary format). The second operand is unchanged by the operation.

The format of the GADD order is:



The GADD order contains the following values:

Name	Bit(s)	Value
Set Mode	0–7	X'28' or X'2A' or X'2C' or X'2E'
Mode Control	8–15	X'EE'
Address Word	0–15	X'aaaa'
Data Word	0–15	X'bbbb'

Address Word Variable

X'aaaa' Denotes a display storage address within the current page (that is, a word boundary) that contains a 16-bit signed integer. The address may be relative or absolute, depending on the SM byte option selected. If relative addressing is used, valid values are in the range from -32,768 to +32,766. Negative numbers are in twos complement. Absolute address values may be in the range from 0 to 65,534. The low-order bit of the address is ignored and assumed to be zero.

Data Word Variable

X'bbbb' The data word variable contains the second operand of the addition, which is also a 16-bit signed integer. Depending on the SM byte option selected, this word may optionally contain an absolute address that points to a word in display storage containing a 16-bit signed integer.

Addition is performed in this three-word order by adding all 16 bits of both operands. If the carry out of the sign-bit position and the carry out of the high-order numeric position agree, the sum is satisfactory; if they disagree, an overflow occurs. A positive overflow yields a negative final sum, and a negative overflow yields a positive final sum.

The condition code bits are set at the end of the operation as:

cc = 00 Sum is zero
cc = 01 Sum is less than zero
cc = 10 Sum is greater than zero
cc = 11 Overflow

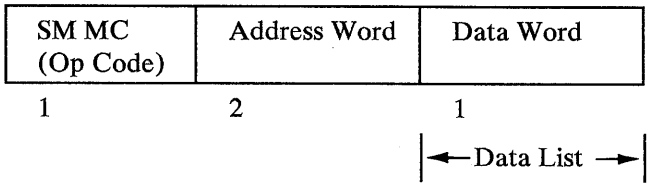
The condition code is preserved until the next order that sets the condition code is executed.

GSUB—Subtract Data

The GSUB order subtracts two 16-bit signed integers.

In the GSUB order the second operand (data word) is subtracted from the contents of the first operand location (pointed to by address), and the difference is placed in the first operand location. Both operands are considered to be 16-bit signed integers (negative numbers are expressed in twos complement, signed 15-bit binary format). The second operand is unchanged by the operation.

The format of the GSUB order is:



The GSUB order contains the following values:

Name	Bit(s)	Value
Set Mode	0–7	X'28' or X'2A' or X'2C' or X'2E'
Mode Control	8–15	X'EF'
Address Word	0–15	X'aaaa'
Data Word	0–15	X'bbbb'

Address Word Variable

X'aaaa' Denotes a display storage address within the current page (that is, a word boundary) that contains a 16-bit signed integer. The address may be relative or absolute, depending on the SM byte option selected. If relative addressing is used, valid values are in the range from -32,768 to +32,766. Negative numbers are in twos complement. Absolute address values may be in the range from 0 to 65,534. The low-order bit of the address is ignored and assumed to be zero.

Data Word Variable

X'bbbb' The data word variable contains the second operand of the subtraction, which is also a 16-bit signed integer. Depending on the SM byte option selected, this word may optionally contain an absolute address that points to a word in display storage containing a 16-bit signed integer.

Subtraction is performed in this three-word order by adding the ones complement of the second operand and a low-order 1 to the first operand. All 16 bits of both operands participate, as in the GADD order. If the carry out of the sign-bit position and the carry out of the high-order numeric position agree, the difference is satisfactory; if they disagree, an overflow occurs. A positive overflow yields a negative final difference, and a negative overflow yields a positive final difference.

The condition code bits are set at the end of the operation as follows:

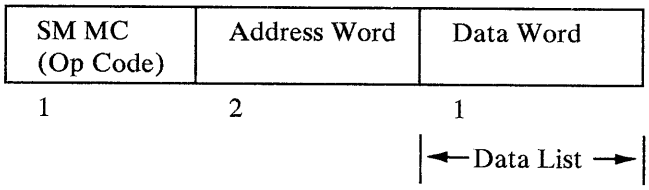
cc = 00 Difference is zero
cc = 01 Difference is less than zero
cc = 10 Difference is greater than zero
cc = 11 Overflow

The condition code is preserved until the next order that sets the condition code is executed.

GDIV—Divide Data

In the GDIV order the dividend operand (contents of the location specified by the address word) is divided by the divisor operand (data word) and replaced by the remainder and the quotient. The dividend operand is assumed to be a 32-bit signed binary number. The user may expand a 16-bit signed binary number to a 32-bit signed binary number by propagating the sign bit into the high-order 16 bits. The remainder and the quotient require 4 bytes of storage. They are both 16-bit values with the quotient on the right and the remainder on the left. The sign of the quotient is determined by rules of algebra; the remainder has the same sign as the dividend, except that a zero quotient and a zero remainder are always positive.

The format of the GDIV order is:



The GDIV order contains the following values:

Name	Bit(s)	Value
Set Mode	0–7	X'28' or X'2A' or X'2C' or X'2E'
Mode Control	8–15	X'E5'
Address Word	0–15	X'aaaa'
Data Word	0–15	X'dddd'

Address Word Variable

X'aaaa' Denotes a display storage address within the current page (that is, a word boundary) of a 4-byte area containing the 32-bit dividend. At the end of the division this area contains the remainder and the quotient. The address may be relative or absolute, depending on the SM byte option selected. If relative addressing is used, valid values are in the range from -32,768 to +32,766. Negative numbers are in twos complement. Absolute address values may be in the range from 0 to 65,534. The low-order bit of the address is ignored and assumed to be zero.

Data Word Variable

X'dddd' The data word variable contains the divisor, which is a 16-bit signed binary number. Depending on the SM byte option specified, this word may optionally contain an absolute address that points to a word in display storage containing the 16-bit divisor.

When the relative magnitudes of the dividend and divisor are such that the quotient cannot be expressed by a 16-bit number, the condition code is set to indicate quotient overflow and no division takes place.

The condition code is set at the end of the operation as:

cc = 00 No overflow; quotient was not overflowed
cc = 01 Overflow; quotient overflowed 16-bit binary number
cc = 10 Not set
cc = 11 Not set

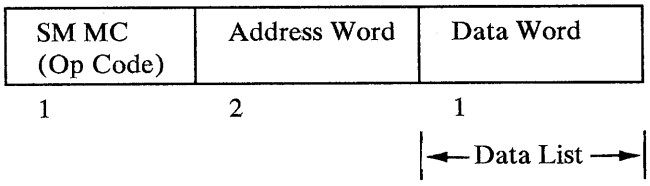
GCOMP—Compare Data

The GCOMP order compares two 16-bit signed integers.

The second operand (data word) of the GCOMP order is compared with the first operand (word addressed by the address word) and the result is indicated in the condition code.

Both operands are considered to be 16-bit signed integers (negative numbers are expressed in twos complement, signed 15-bit binary format). Neither operand is changed by the operation.

The format of the GCOMP order is:



The GCOMP order contains the following values:

Name	Bit(s)	Value
Set Mode	0–7	X'28' or X'2A' or X'2C' or X'2E'
Mode Control	8–15	X'E9'
Address Word	0–15	X'aaaa'
Data Word	0–15	X'bbbb'

Address Word Variable

X'aaaa' Denotes a display storage address within the current page (that is, a word boundary) of the first operand. The address may be relative or absolute, depending on the SM byte option selected. If relative addressing is used, valid values are in the range from -32,768 to +32,766. Negative numbers are in twos complement. Absolute address values may be in the range from 0 to 65,534. The low-order bit of the address is ignored and assumed to be zero.

Data Word Variable

X'bbbb' The data word contains the second operand (a 16-bit signed integer) of the compare operation. Depending on the SM byte option specified, this word may optionally contain an absolute address that points to a word in display storage containing the 16-bit signed integer.

The condition code bits are set at the end of the operation as:

cc = 00 Operands are equal
cc = 01 First operand less than second operand
cc = 10 First operand greater than second operand
cc = 11 Not set (reserved)

The condition code is preserved until the next order that sets the condition code is executed.

GSHIFT—Shift Data

The GSHIFT order shifts a data field that is either 16 or 32 bits long. The data field is located at the address specified in the address word. The shifted data is written over the original data.

The format of the GSHIFT order is:

SM MC (Op Code)	MC Extension	Address Word
1	2	3

The GSHIFT order contains the following values:

Name	Bit(s)	Value
Set Mode Mode Control	0–7 8–15	X'2A' or X'2E' X'E3'
Mode Control Extension Word	0–3 4 5 6 7–10 11–15	B'0000' B't' B'd' B'1' B'0000' B'sssss'
Address Word	0–15	X'aaaa'

Mode Control Extension Word Variables

B't' Type

t=0 Logical shift; all data bits participate in the shift operation.

t=1 Arithmetic shift; the data sign bit does not participate in the shift operation, that is, the data bits are shifted as a signed binary number.

B'd' Direction

d=0 Shift left

d=1 Shift right

B'l' Length

l=0	Data to be shifted is 16 bits in length.
l=1	Data to be shifted is 32 bits in length.

B's...s' Number of bit positions to be shifted; for example, a count (b'00000' to b'11111' or decimal 0 to 31)

Address Word Variable

X'aaaa' Denotes a display storage address within the current page (that is, a word boundary) of the data field to be shifted. The address may be relative or absolute, depending on the SM byte option selected. If relative addressing is used, valid values are in the range from -32,768 to +32,766. Negative numbers are in twos complement. Absolute address values may be in the range from 0 to 65,534. The low-order bit of the address is ignored and assumed to be zero.

When an arithmetic shift (t = 1) is indicated, the data is considered to be a 16-bit or 32-bit signed binary number.

Positive numbers are represented in true binary form, with a zero in the sign bit. Negative numbers are represented in twos complement notation, with a 1 bit in the sign position. The high-order bit of the data is assumed to be the sign bit. Only the low-order 15 or 31 bits participate in the shift operation.

During an arithmetic left shift (t = 1, d = 0), zeros are supplied to vacated low-order positions. The high-order bits are shifted out and lost. If a bit unlike the sign bit is shifted out of the second high-order bit, an overflow occurs and the condition code is set accordingly.

During an arithmetic right shift (t = 1, d = 1), bits equal to the sign bits are supplied to the vacated high-order positions. The low-order bits are shifted out without inspection and are lost.

When a logical shift (t = 0) is indicated, the data is considered to be a 16-bit or 32-bit unsigned binary number and all data bits participate in the shift operation.

During a logical left shift (t = 0, d = 0), zeros are supplied to vacated low-order positions. The high-order bits are shifted out without inspection and are lost.

During a logical right shift (t = 0, d = 1), zeros are supplied to the vacated high-order positions. The low-order bits are shifted out without inspection and are lost.

The s...s field contains the number of bit positions that the data field is shifted to the right (d = 1) or to the left (d = 0). The condition code is set at the end of the operation as:

cc =	00	No overflow
cc =	01	Overflow caused by an arithmetic left shift
cc =	10	Not set
cc =	11	Not set

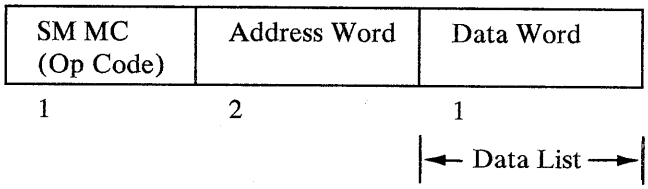
GTM—Test under Mask

The GTM order checks one-for-one correspondence between bits of a mask and bits of a word in storage.

The second operand (data word) of the GTM order contains a 16-bit mask. The state of first-operand bits selected by the mask is used to set the condition code.

The bits of the mask are made to correspond one-for-one with the bits of the word in storage specified by the first-operand address (address word). A mask bit of 1 indicates that the storage bit is to be tested. When the mask bit is zero, the storage bit is ignored. Neither operand is changed by execution of this order.

The format of the GTM order is:



The GTM order contains the following values:

Name	Bit(s)	Value
Set Mode	0–7	X'28' or X'2A' or X'2C' or X'2E'
Mode Control	8–15	X'ED'
Address Word	0–15	X'aaaa'
Data Word	0–15	X'bbbb'

Address Word Variable

X'aaaa' Denotes a display storage address within the current page (that is, a word boundary) of the word in storage to be tested. The address may be relative or absolute, depending on the SM byte option selected. If relative addressing is used, valid values are in the range from -32,768 to +32,766. Negative numbers are in twos complement. Absolute address values may be in the range from 0 to 65,534. The low-order bit of the address is ignored and assumed to be zero.

Data Word Variable

X'bbbb' The data word contains the 16-bit mask word described above. Depending on the SM byte option selected, this word may optionally contain an absolute address that points to a word in display storage containing the 16-bit mask.

The condition code bits are set at the end of the operation as:

cc = 00 Selected bits all zeros or the mask is all zeros
cc = 01 Selected bits mixed zeros and 1s
cc = 10 Not set (reserved)
cc = 11 Selected bits all 1s

The condition code is preserved until the next order that sets the condition code is executed.

Data Move Orders

Each of the three data move orders described in this section is summarized in the following table by mnemonic form and name, plus a short description of function.

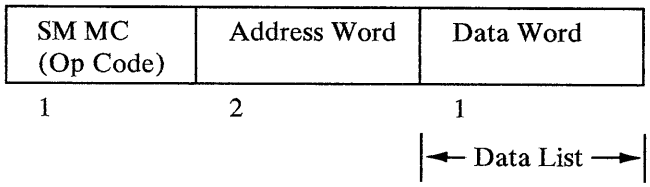
Mnemonic	Name	Function
GMVA	Move Address	Copies data held in the third (last) word of the order to the address defined by the second word
GMVW	Move Word	
GMVBLK	Move Data Block	Copies a block of data from a display storage location specified by an origin page number/address into a location specified by a destination page number/address

Note: The GMVA and GMVW orders are identical in execution. Their different order codes permit the writing of host system programs that are sensitive to their different intended uses. For example, a program that relocated a display program order from one display storage start location to another would modify the third word of the GMVA order, assuming it to hold an address, but would not modify the third word of a GMVW order.

GMVA—Move Address

The GMVA order copies the data (an address) held in the last word of the order to the address defined by the second word of the order.

The format of the GMVA order is:



The GMVA order contains the following values:

Name	Bit(s)	Value
Set Mode	0–7	X'28' or X'2A' or X'2C' or X'2E'
Mode Control	8–15	X'EB'
Address Word: Destination	0–15	X'aaaa'
Data Word: Data to Be Stored	0–15	X'bbbb'

Address Word Variable

X'aaaa' Denotes a display storage address within the current page (that is, a word boundary) of the destination to which the data word is to be moved. The address may be relative or absolute, depending on the SM byte option selected. If relative addressing is used, valid values are in the range from -32,768 to +32,766. Negative numbers are in twos complement. Absolute address values may be in the range from 0 to 65,534. The low-order bit of the address is ignored and assumed to be zero.

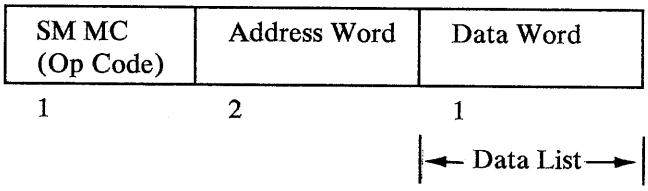
Data Word Variable

X'bbbb' The data word contains the data (an address) that is to be moved to the location specified by the address word (X'aaaa' above). Depending on the SM byte option specified, this word may optionally contain an absolute address that points to a word in display storage that holds the address.

GMVW—Move Word

The GMVW order copies the data held in the last word of the order to the address defined by the second word of the order.

The format of the GMVW order is:



The GMVW order contains the following values:

Name	Bit(s)	Value
Set Mode	0–7	X'28' or X'2A' or X'2C' or X'2E'
Mode Control	8–15	X'EC'
Address Word: Destination	0–15	X'aaaa'
Data Word: Data to Be Stored	0–15	X'bbbb'

Address Word Variable

X'aaaa' Denotes a display storage address within the current page (that is, a word boundary) of the destination to which the data word is to be moved. The address may be relative or absolute, depending on the SM byte option selected. If relative addressing is used, valid values are in the range from -32,768 to +32,766. Negative numbers are in twos complement. Absolute address values may be in the range from 0 to 65,534. The low-order bit of the address is ignored and assumed to be zero.

Data Word Variable

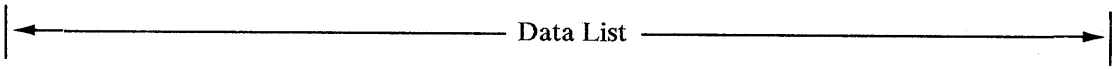
X'bbbb' The data word contains the data word (1 word) that is to be moved to the location specified by the address word (X'aaaa' above). Depending on the SM byte option specified, this word may optionally contain an absolute address that points to a word in display storage that holds that data.

GMVBLK—Move Data Block

The GMVBLK order copies a block of data from a display storage location specified by the origin page number and origin address into a location specified by the destination page number and destination address. Overlapping of the origin and destination areas is allowed, provided the final content of the destination area is not affected.

The format of the GMVBLK order is:

SM MC (Op Code)	Data Word: Count	Data Word: Origin Page	Data Word: Origin Address	Data Word: Destination Page	Data Word: Destination Address
1	1	2	3	4	5



The GMVBLK order contains the following values:

Name	Bit(s)	Value
Set Mode Mode Control	0–7 8–15	X'28' or X'2A' X'DB'
Data Word 1: Count	0–15	X'cccc'
Data Word 2: Origin Page Number	0–15	X'pppp'
Data Word 3: Origin Address	0–15	X'aaaa'
Data Word 4: Destination Page Number	0–15	X'pppp'
Data Word 5: Destination Address	0–15	X'aaaa'

The number of bytes in the data block is specified in the count field. Only even numbers of the bytes are moved. Bit 15 of the count field is ignored and assumed to be zero. The result of any attempt to have a data block span pages in either origin or destination locations is unpredictable.

See "GBPAGE—Branch Page" on page 5-56 for a definition of valid page numbers and address word values.

An attempt to copy from or into an unavailable destination page results in a display program error (see Appendix C for a definition of the sense data).

Bit 15 of the origin address and bit 15 of the destination address are ignored and assumed to be zero.

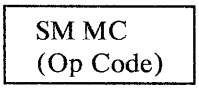
No-Operation Orders

The two no-operation orders are the 2-Byte No-Operation (GNOP2) and 4-Byte No-Operation (GNOP4) orders.

GNOP2—2-Byte No-Operation

The one-word GNOP2 order performs no operation.

The format of the GNOP2 order is:



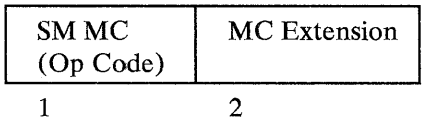
The GNOP2 order contains the following values:

Name	Bit(s)	Value
Set Mode Mode Control	0–7 8–15	X'2A' X'80'

GNOP4—4-Byte No-Operation

The two-word GNOP4 order performs no operation. The second word (X'xxxx') of the order is ignored and may contain any value; it is treated as an MC extension word for purposes of order classification.

The format of the GNOP4 order is:



The GNOP4 order contains the following values:

Name	Bit(s)	Value
Set Mode Mode Control	0–7 8–15	X'2A' X'C0'
Mode Control Extension Word	0–15	X'xxxx'

Area Fill Orders

Area fill orders define the perimeter of an area to be filled and specify the related area fill controls or the pattern to be used during the fill operation.

Mnemonic forms, names, and functions of the three area fill orders described in this section are summarized in the table below.

Mnemonic	Name	Function
GBGAF	Begin Area Fill	Indicates the start of the definition of an area perimeter
GEAF	End Area Fill	Terminates the definition of an area perimeter
GLPAT	Load Pattern	Loads a data list containing an area fill pattern into area fill pattern storage

Notes:

1. The area fill function requires a working space in system memory to record and sort information related to each edge of the area to be filled (perimeter edge list). The graphics system provides a 2K default area fill work area capable of supporting shapes containing up to 64 edges in their perimeters.

   If the area you want to fill contains more edges, you must define a memory area of type=area fill of sufficient size to hold the appropriate information for the largest shape you want to fill. The size of the memory area required can be determined from the following formula:

   Number of Edges x 32 = Size of Work Area in Bytes

   See "Memory Areas" on page 2-6 for details on how to define a memory area.

2. If three-dimensional (3D) mode is specified (d bit = 1 in attribute register 19), area fill defaults to two-dimensional (2D) mode and the Z coordinates are ignored.

3. Only output primitive orders (with the exception of the GDPXL order), no-operation orders, and branch control orders may be encompassed within an area fill definition (that is, between a GBGAF order and a GEAF order).

GBGAF—Begin Area Fill

The two-word GBGAF order indicates the start of an area perimeter definition. The perimeter definition can be specified by using any line draw order or the Draw Circle (GDCIR) order. The accumulation of perimeter information continues until an End Area Fill (GEAF) order is encountered.

If the area is not "closed" correctly, subsequent action is unpredictable. The perimeter is covered by the area fill. The area fill is clipped at the screen boundary. The current draw position is not affected by the GBGAF order.

The format of the GBGAF order is:

SM MC (Op Code)	MC Extension
1	2

The GBGAF order contains the following values:

Name	Bit(s)	Value
Set Mode	0–7	X'2A'
Mode Control	8–15	X'9C'
Mode Control	0–11	B'0...0'
Extension Word	12–13	B'00'
	14–15	B'pp'

Mode Control Extension Word Variable

B'pp' Type of area fill
- pp = 00 No fill
- pp = 01 Solid fill
- pp = 10 Pattern fill; use pattern loaded by GLPAT order
- pp = 11 Reserved

GEAF—End Area Fill

The one-word GEAF order terminates the definition of an area perimeter and initiates the area fill process. At this time the area fill control information (as described under the GBGAF order) is passed to the area fill process. Order processing is complete when the area is filled. If the area was not closed correctly, the results of the fill process are unpredictable.

If the area fill working space (system default or memory area defined) does not exist or have enough space for the perimeter edge list, the area fill terminates and processing resumes with the next order following the GEAF order. The perimeter may not be filled, or may be partially filled, depending on when the lack of work space was detected. For definitions of the memory area and area fill working space memory area, see "Memory Areas" on page 2-6.

The format of the GEAF order is:

SM MC (Op Code)
1

The GEAF order contains the following values:

Name	Bit(s)	Value
Set Mode	0–7	X'2A'
Mode Control	8–15	X'9D'

The GEAF order updates the condition code. The condition code following the GEAF order indicates the status of the PI and TSI as:

cc = 00 PI off and TSI off (open)
cc = 01 PI off and TSI on (closed)
cc = 10 PI on and TSI off (open)
cc = 11 PI on and TSI on (closed)

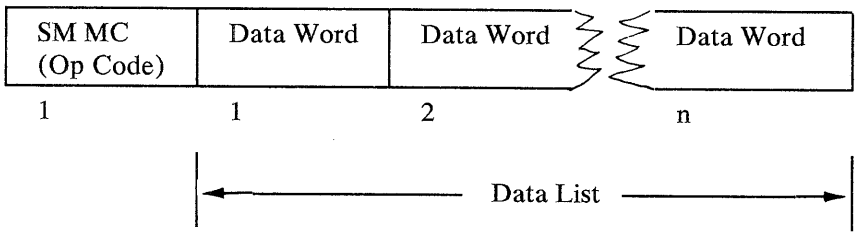
The current draw position is not affected by this order.

GLPAT—Load Pattern

The GLPAT order loads the following data list into area fill pattern storage.

This one-word order is followed by a string of 256 bytes of data specifying a 16 x 16 pixel pattern. Each successive 16 bytes of the string constitutes one row of pixels along the X coordinate of the pattern.

The format of the GLPAT order is:



The GLPAT order contains the following values:

Name	Bit(s)	Value
Set Mode	0–7	X'28 or X'2A'
Mode Control	8–15	X'D7'

The data bytes in the data list represent the color/grayshade value to be assigned to the pixels to create the desired pattern effect.

The default pattern is all zeros in each pixel. The condition code is not changed by executing this order.

Configuration Data Order

The system configuration is stored by a single graphic order: Store Configuration Data (GSCONF).

Mnemonic	Name	Function
GSCONF	Store Configuration Data	Stores the system configuration into display storage

GSCONF—Store Configuration Data

The GSCONF order stores the system configuration into display storage at the location specified.

The format of the GSCONF order is:

SM MC (Op Code)	Address Word
1	2

The GSCONF order contains the following values:

Name	Bit(s)	Value
Set Mode Mode Control	0–7 8–15	X'2A' or X'2E' X'DF'
Address Word	0–15	X'aaaa'

Address Word Variable

X'aaaa' Denotes the starting display storage address within the current page (that is, a word boundary) of a 100-byte area in which the configuration data is to be stored. The address may be relative or absolute, depending on the SM byte option selected. If relative addressing is used, valid values are in the range from -32,768 to +32,766. Negative numbers are in twos complement. Absolute address values may be in the range from 0 to 65,534. The low-order bit of the address is ignored and assumed to be zero.

The two-word GSCONF order causes a parameter list of 42 bytes containing configuration data to be stored beginning at the address in the address word.

Note: The user must reserve 100 bytes at the location pointed to by the address to allow for future growth. The number of parameters may be expanded at any time without notice.

The data is stored as self-identifying parameters in the format:

Length	Code
Data (variable length)	

where:

Length is a 1-byte field that specifies the length of the parameter, including the length field itself, the 1-byte code, and the data.

Code is a 1-byte field containing a hexadecimal code that uniquely identifies the parameter.

Data is a variable-length field that contains the data of the parameter.

Parameters are:

1. Graphics System Configuration

   The first parameter stored has the format:

   - Length = X'06'
   - Code = X'01'
   - 4 data bytes specifying the configuration:

   Byte 0 Device identifier

   Bits

0	Reserved
1	IBM 5085 Graphics Processor
2	IBM 5085–RS232C Port A
3	IBM 5085–RS232C Port B
4-7	Reserved

   Byte 1 I/O devices attached

   Bits

0	Pick device attached
1	ANK
2	LPFK
3	Tablet
4	Dials Feature
5-7	Reserved

   Bytes 2–3 Reserved

2. Graphics System Storage Size

   This parameter specifies the size of the processor total storage and the segment assigned to the display storage.

   - Length = X'06'
   - Code = X'02'
   - 2 data bytes specifying the total number of available 2K-byte blocks of storage.
   - 2 data bytes specifying the number of 2K-byte blocks of storage assigned to the display storage.

3. Graphics System Display Characteristics

   This parameter identifies the technology, color capability and resolution of the display.

   - Length = X'04'
   - Code = X'03'
   - 2 data bytes specifying the display characteristics.

   Byte 0 Display technology

   Bits 0–3 Technology Identifier

   = X'0' — Reserved
   = X'1' — Raster device
   = X'2' to X'F' Reserved

   Bit 4 Mono/color indicator

   = 0 — Monochrome
   = 1 — Color

   Bit 5–7 Reserved

   Byte 1 Display area resolution

   = X'00' — 1024 x 1024
   = X'01' to X'FF' Reserved

4. Frame Buffer Characteristics

This parameter identifies the frame buffer characteristics.

- Length = X'04'
- Code = X'04'
- 2 data bytes specifying the frame buffer characteristics

Byte 0 Frame buffer characteristic

Bit 0 Reserved

Bits 1–3 Number of bit planes per frame buffer

= 000 Reserved
= 001 2 bit planes/frame buffer
= 010 Reserved
= 011 4 bit planes/frame buffer
= 100 Reserved
= 101 6 bit planes/frame buffer
= 110 Reserved
= 111 8 bit planes/frame buffer

Bits 4–7 Reserved

Byte 1 Reserved

5. Graphics System Features

This parameter contains the list of features installed.

- Length = X'04'
- Code = X'05'
- 2-bit encoded bytes specifying the features installed.

Bytes 0–1 Feature list

Bit 0 TCF 2D and 3D
Bits 1–15 Reserved

6. Reserved

This parameter is reserved for future use.

- Length = X'06'
- Code = X'06'
- Set to X'00000000'

7. EC Level

   This parameter contains the EC level of the system.

   - Length = X'0A'
   - Code = X'07'
   - 8-byte EC level identifier.

8. Terminator Parameter

   This parameter indicates the end of the parameter list.

   - Length = X'02'
   - Code = X'FF'
   - No data bytes follow this code.

Transformation and Clipping Feature (TCF) Orders

This section describes the following Transformation and Clipping Feature (TCF) orders:

1. TCF Character Mode Orders

2. TCF Transformation and Viewing Orders

- GLTM Load Transformation Matrix
- GSTM Store Transformation Matrix
- GLCB Load Clipping Boundaries
- GSCB Store Clipping Boundaries
- GLVPT Load Viewpoint
- GSVPT Store Viewpoint
- GLVB Load Viewport Boundaries
- GSVB Store Viewport Boundaries

3. TCF Advanced Arithmetic Orders

- GMUL Multiply Data
- GSIN Sine
- GCOS Cosine

4. TCF Stack Control Orders

- GPTM Push Transformation Matrix
- GPCB Push Clipping Boundaries
- GPVPT Push Viewpoint
- GPVB Push Viewport Boundaries

Note: The result of executing any TCF order when the feature is not installed is unpredictable.

TCF Character Mode Orders

The Transformation and Clipping Feature can be used to cause transformation and clipping of hardware-generated character strings (CSID = X'00' and X'04' to X'07'), with the restriction that scaling of the actual characters is not performed. Clipping is performed at the viewport boundaries when mapping mode is specified.

Note: Clipping of hardware-generated character strings is *not* specified by means of the clipping mode bit in attribute register 19.

Clipping of hardware-generated character strings is performed on a character basis (fractional characters do not result).

The *programmable character sets (PCSs)* are fully transformable and clippable using TCF. The move and draw data defined by the programmable character set (see "Graphic System Character Set Generation" on page 3-33 for details) is treated in the same manner as all other graphic orders.

Note: Type-1 PCS definition specifies only X-Y coordinates; Z values are retrieved from the current draw position at the start of the symbol string, if necessary.

TCF Transformation and Viewing Orders

The eight TCF transformation and viewing orders are described in this section in the order indicated in the following table. This table gives mnemonic forms and names and functions of the orders.

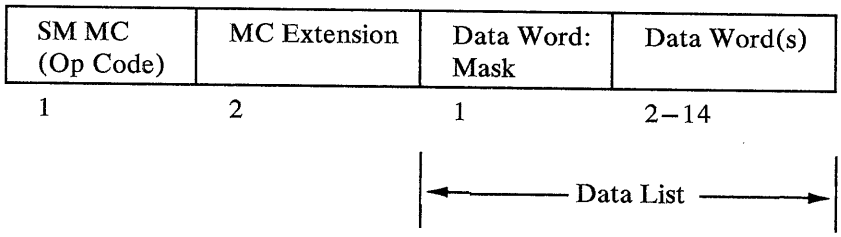
Mnemonic	Name	Function
GLTM	Load Transformation Matrix	Loads a transformation matrix in TCF
GSTM	Store Transformation Matrix	Stores the one-word transformation mask and the 14 elements of the current transformation values at the address defined by the order
GLCB	Load Clipping Boundaries	Loads clipping boundary values and defines X-Y-Z limits for a window
GSCB	Store Clipping Boundaries	Stores the one-word clipping boundary mask and the six one-word clipping boundary values at the address defined by the order
GLVPT	Load Viewpoint	Specifies the position of the apex of the viewing pyramid relative to the display screen
GSVPT	Store Viewpoint	Stores the viewpoint position at the address specified
GLVB	Load Viewport Boundaries	Loads X-Y limits of the viewpoint rectangle
GSVB	Store Viewport Boundaries	Stores the one-word viewport boundary values at the address defined by the order

GLTM—Load Transformation Matrix

The GLTM order loads the transformation matrix.

The two-word GLTM order is followed by a variable-length data list if the concatenation control (cc) bits (bits 14 and 15 in the MC extension word) are *not* zero. The first word of the data list is a mask word. If the concatenation control bits *are* zero (00) there is no data list.

The format of the GLTM order is:



The GLTM order contains the following values:

Name	Bit(s)	Value
Set Mode Mode Control	0–7 8–15	X'28' or X'2A' X'20'
Mode Control Extension Word	0–7 8–10 11 12–13 14–15	B'0...0' B'000' B'd' B'00' B'cc' (see "Concatenation Control" below)
Mask Word	0–15	(see "Mask Word Variables" below)
Matrix Element Data List Words (up to 13, depending on mask setting)	0–15	(see "Matrix Element Variables" below)

Mode Control Extension Word Variable

d=0 The transformation matrix is to be used for 2D transformation.

d=1 The transformation matrix is to be used for 3D transformation.

Concatenation Control (cc): The *concatenation control,* or *cc,* bits of the MC extension word define how the supplied elements change the transformation matrix stored in the TCF.

If cc = 00, the transformation matrix is set to the identity or unit matrix (there is no data list).

If d = 0, the unit matrix is the following 3 x 3 matrix:

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

If d = 1, the unit matrix is the following 4 x 4 matrix:

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

If cc = 01, the matrix defined by the supplied elements is concatenated with the transformation matrix and the result replaces the transformation matrix as follows:

$$[U] = [T]\,[E]$$

where U is the new transformation matrix, T is the old transformation matrix, and E is the matrix defined by the supplied elements.

If cc = 10, the transformation matrix is concatenated with the matrix defined by the supplied elements and the result replaces the transformation matrix, as follows:

$$[U] = [E]\,[T]$$

where U is the new transformation matrix, T is the old transformation matrix, and E is the matrix defined by the supplied elements.

A third column (if d = 0) or a fourth column (if d = 1) is added to each matrix before concatenation. The values of the elements in the third column (for d = 0) are:

$$\begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

The values of the elements in the fourth column (for d = 1) are:

$$\begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}$$

In the two cases of concatenation, any elements of the matrix e that are not supplied are defaulted to zero.

If concatenation causes the resultant scale factor to exceed the allowable value (see "Number Representation" on page B-1), the display program terminates and a matrix element overflow error is signaled to the host via a Unit Check (see Appendix C for the definition of the sense data).

If cc = 11, the matrix defined by the supplied elements becomes the transformation matrix; any element not supplied is defaulted to zero.

The default value of the transformation matrix is the unit matrix.

Mask Word Variables

If d = 0, the transformation matrix is used for 2D transformation. The mask word has the following format:

m11 m12 x x	m21 m22 x x	x x x x	m41 m42 x s

If d = 1, the transformation matrix is a 3 x 4 matrix and is used for 3D transformation. The mask word has the following format:

m11 m12 m13 x	m21 m22 m23 x	m31 m32 m33 x	m41 m42 m43 s

Matrix Element Variables

The mask word is followed by the remainder of the data list with a word length equal to the number of 1-bits in the mask word; this part of the data list contains matrix elements and an optional scale factor(s) in the same order as specified by the mask word, where m11, m12,...m43, s refer to the individual matrix elements and the scale factor, which are labeled as follows:

$$\begin{bmatrix} m11 & m12 & m13 \\ m21 & m22 & m23 \\ m31 & m32 & m33 \\ m41 & m42 & m43 \end{bmatrix}$$

The s bit indicates that the GLTM order contains a scale factor. Scale factor is a 4-bit power-of-2 exponent with a valid range from 0 to 12. In 2D mode, matrix elements m11, m12, m21, and m22 are each multiplied by the scale factor. In 3D mode, the matrix elements m11, m12, m13, m21, m22, m23, m31, m32, and m33 are each multiplied by the scale factor.

s=0 No scale factor is included in this order. Use the default value (X'0') of the scale factor.

s=1 The low-order 4 bits of the last word in the data list contain a scale factor.

The value of a mask bit is ignored if 'x' is assigned to it and no word should be assigned to it in the data list.

GSTM—Store Transformation Matrix

The two-word GSTM order causes the TCF to store 14 words at the address defined in the order. The stored words consist of the mask word followed by the elements of the current transformation matrix and the scale factor; they are stored in the order in which they appear in the mask.

The format of the GSTM order is:

SM MC (Op Code)	Address Word
1	2

The GSTM order contains the following values:

Name	Bit(s)	Value
Set Mode	0–7	X'2A' or X'2E'
Mode Control	8–15	X'21'
Address Word	0–15	X'aaaa'

Address Word Variable

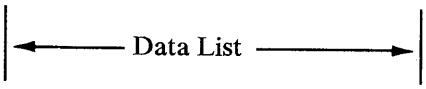
X'aaaa' Denotes the starting display storage address within the current page (that is, a word boundary) of a 14-word area in which transformation matrix data is to be stored. The address may be relative or absolute, depending on the SM byte option selected. If relative addressing is used, valid values are in the range from -32,768 to +32,766. Negative numbers are in twos complement. Absolute address values may be in the range from 0 to 65,534. The low-order bit of the address is ignored and assumed to be zero.

GLCB—Load Clipping Boundaries

The one-word GLCB order is followed by a data list consisting of a mask word and one to six words that contain the X, Y, Z limits of the clipping rectangle or view volume.

The format of the GLCB order is:

SM MC (Op Code)	Mask Word	Data Word(s)
1	1	2–7



The GLCB order contains the following values:

Name	Bit(s)	Value
Set Mode Mode Control	0–7 8–15	X'28' or X'2A' X'22'
Mask Word	0 1 2 3 4 5 6–15	B'm1' B'm2' B'm3' B'm4' B'm5' B'm6' B'x...x'
Data Words (up to six)	0–15	(see "Data Word Variables" below)

Data Word Variables

The mask word is followed by the remainder of the data list with a word length equal to the number of 1-bits in the mask word; this part of the data list contains the coordinate values of X, Y, and Z clipping boundaries in the same order as specified by the mask word, where m1, m2, m3, m4, m5, and m6 refer to individual limits in the following order:

ti 05

m1	X left limit	(minimum displayable X coordinate)
m2	X right limit	(maximum displayable X coordinate)
m3	Y bottom limit	(minimum displayable Y coordinate)
m4	Y top limit	(maximum displayable Y coordinate)
m5	Z near limit	(minimum displayable Z coordinate)
m6	Z far limit	(maximum displayable Z coordinate)

The value of a mask bit is ignored if x is assigned to it. The allowable ranges of values are from -32K to +32K for X and Y coordinates and from zero to +32K for Z coordinates.

The boundary value is not changed if the corresponding mask bit is zero. Default clipping boundary values are zero for X,Y, and Z minimum limits and 4095 for X, Y, and Z maximum limits.

The least significant bit of each of the X,Y, and Z minimum limits are assumed to be zero, and the least significant bit of each of X,Y, and Z maximum limits are assumed to be 1.

The X and Y limits are also used to define a window. In 3D mode, the image is projected on this window. If the mapping mode is on, the image on the window is mapped into an image on a viewport.

Note: When default values are used for clipping boundaries and the viewport, the window and viewport coincide, and mapping from the window to the viewport is on a one-to-one basis.

GSCB—Store Clipping Boundaries

The two-word GSCB order causes seven words to be stored at the address defined in the order. The stored words consist of the mask word, as defined by the GLCB order, followed by the clipping boundary values. Mask bits m1 to m6 are set to 1; the remaining bits are set to zero.

The format of the GSCB order is:

SM MC (Op Code)	Address Word
1	2

The GSCB order contains the following values:

Name	Bit(s)	Value
Set Mode Mode Control	0–7 8–15	X'2A' or X'2E' X'23'
Address Word	0–15	X'aaaa'

Address Word Variable

X'aaaa' Denotes the starting display storage address within the current page (that is, a word boundary) of a seven-word area in which clipping boundaries are to be stored. The address may be relative or absolute, depending on the SM byte option selected. If relative addressing is used, valid values are in the range from -32,768 to +32,766. Negative numbers are in twos complement. Absolute address values may be in the range from 0 to 65,534. The low-order bit of the address is ignored and assumed to be zero.

GLVPT—Load Viewpoint

The GLVPT order specifies the position of the apex of the viewing pyramid (the perspective viewpoint) relative to the viewplane (display screen).

The format of the GLVPT order is:

SM MC (Op Code)	Data Word
1	1

←Data List→

The GLVPT order contains the following values:

Name	Bit(s)	Value
Set Mode Mode Control	0–7 8–15	X'28' or X'2A' X'24'
Data Word	0 1–15	B's' B'z...z'

Data Word Variables

B's' Sign; always minus (set to 1)

B'z...z' Viewpoint in a range from -2 to -32K increasing in value as the viewpoint is moved outward from the viewplane. The default value of the viewpoint is -32K.

The viewpoint is valid only when 3D mode and perspective mode are selected. If 2D is active, the order is executed but the viewpoint is ignored in the transformation process.

GSVPT—Store Viewpoint

The two-word GSVPT order causes the viewpoint position to be stored at the address specified. The format of the stored viewpoint is the same as the second word of the GLVPT order.

The format of the GSVPT order is:

SM MC (Op Code)	Address Word
1	2

The GSVPT order contains the following values:

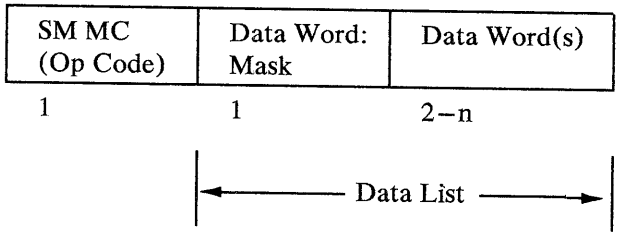
Name	Bit(s)	Value
Set Mode Mode Control	0–7 8–15	X'2A' or X'2E' X'25'
Address Word	0–15	X'aaaa'

Address Word Variable

X'aaaa' Denotes a display storage address within the current page (that is, a word boundary) of the word into which the viewport is to be stored. The address may be relative or absolute, depending on the SM byte option selected. If relative addressing is used, valid values are in the range from -32,768 to +32,766. Negative numbers are in twos complement. Absolute address values may be in the range from 0 to 65,534. The low-order bit of the address is ignored and assumed to be zero.

The one-word GLVB order is followed by a data list consisting of a mask word and one or more words that contain the X and Y limits in virtual image coordinates of the viewport rectangle.

The format of the GLVB order is:



The GLVB order contains the following values:

Name	Bit(s)	Value
Set Mode	0–7	X'28' or X'2A'
Mode Control	8–15	X'1A'
Data Word 1: Mask	0	B'm1'
	1	B'm2'
	2	B'm3'
	3	B'm4'
	4–15	B'x...x'
Data Words 2–n		

Data Word 1 Variables: Mask

The mask word is followed by the remainder of the data list with a word length equal to the number of 1-bits in the mask word; this part of the data list contains the coordinate values of X and Y of the viewport boundaries in the same order specified by the mask word, where m1, m2, m3, and m4 refer to individual limits in the following order:

m1	X left limit	(minimum displayable X coordinate)
m2	X right limit	(maximum displayable X coordinate)
m3	Y bottom limit	(minimum displayable Y coordinate)
m4	Y top limit	(maximum displayable Y coordinate)

The value of a mask bit is ignored if x is assigned to it.

Data Word 2–n Variables

The viewport is used if the mapping mode is on. If the mapping mode is off, the GLVB order is executed, but the viewport is ignored in the transformation process. Any number of words up to four can appear in the data list. The range of values of the limits is zero through +4095.

The default values of the viewport boundaries are the screen boundaries—that is, the left and bottom limits are zero and the right and top limits are +4095.

A boundary value is not changed if the corresponding mask bit is zero.

GSVB—Store Viewport Boundaries

The two-word GSVB order causes five words to be stored at the address specified in the order. The stored words consist of the mask word, as defined by the GLVB order, followed by the viewport boundary values. Mask bits m1 to m4 of the mask word are set to 1; the remaining bits are set to zero.

The format of the GSVB order is:

SM MC (Op Code)	Address Word
1	2

The GSVB order contains the following values:

Name	Bit(s)	Value
Set Mode Mode Control	0–7 8–15	X'2A' or X'2E' X'1B'
Address Word	0–15	X'aaaa'

Address Word Variable

X'aaaa' Denotes the starting display storage address within the current page (that is, a word boundary) of a five-word area into which the viewport boundaries are to be stored. The address may be relative or absolute, depending on the SM byte option selected. If relative addressing is used, valid values are in the range from -32,768 to +32,766. Negative numbers are in twos complement. Absolute address values may be in the range from 0 to 65,534. The low-order bit of the address is ignored and assumed to be zero.

TCF Advanced Arithmetic Orders

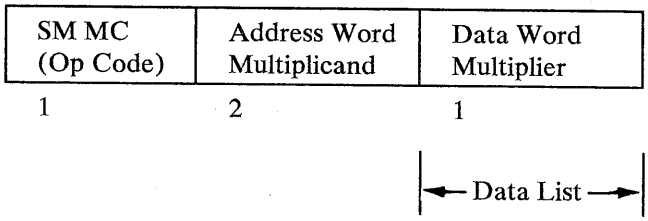
The three TCF advanced arithmetic orders described in this section are summarized in the following table, which indicates the mnemonic form, name, and function of each.

Mnemonic	Name	Function
GMUL	Multiply Data	Multiplies the data word by the address word, and uses the product to replace the address word and the next 16-bit word
GSIN	Sine	Stores the sine of the angle defined by the data word in the location specified by the address word
GCOS	Cosine	Stores the cosine of the angle defined by the data word in the location specified by the address word

GMUL—Multiply Data

In the GMUL order the product of the multiplier operand (data word) and multiplicand operand (contents of location specified by the address word) replace the multiplicand operand and the next 16-bit word (that is, ascending in address). Both operands are considered 16-bit signed binary numbers. The sign of the product is determined by rules of algebra from the signs of the multiplier and the multiplicand.

The format of the GMUL order is:



The GMUL order contains the following values:

Name	Bit(s)	Value
Set Mode	0–7	X'28' or X'2A' or X'2C' or X'2E'
Mode Control	8–15	X'E4'
Address Word	0–15	X'aaaa'
Data Word	0–15	X'dddd'

Address Word Variable

X'aaaa' Denotes the display storage address within the current page (that is, a word boundary) of a two-word field, the first word of which contains the multiplicand. At the completion of the multiply operation the product is stored in the two-word field, replacing the multiplicand. The address may be relative or absolute, depending on the SM byte option selected. If relative addressing is used, valid values are in the range from -32,768 to +32,766. Negative numbers are in twos complement. Absolute address values may be in the range from 0 to 65,534. The low-order bit of the address is ignored and assumed to be zero.

Data Word Variable

X'dddd' The data word contains the multiplier in the form of a 16-bit signed binary number. Depending on the SM byte option selected, this word may optionally contain an absolute address that points to a word in display storage that contains the multiplier.

The product is a 32-bit signed binary number. The most significant part (MSP) of the product is in the high-order 16 bits (the multiplicand) and the least significant part (LSP) of the product is in the low-order 16 bits (the next word).

Positive numbers are represented in true binary form, with a zero in the sign bit. Negative numbers are represented in twos complement notation, with a 1 bit in the sign position. If the high-order 17 bits are not the same (all zeros or all 1s), the condition code is set to 01; otherwise, it is set to 00.

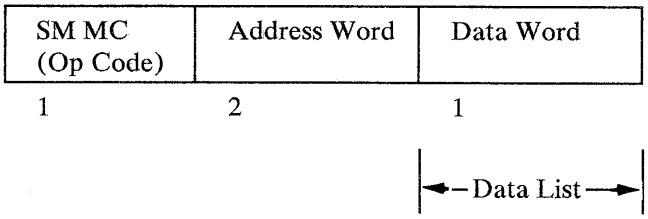
The condition code is set at the end of the operation as:

cc = 00 Product is in low-order 16 bits
cc = 01 Product is 32 bits
cc = 10 Not set
cc = 11 Not set

GSIN—Sine

In the GSIN order the sine of the angle defined by the second operand (data word) is stored in the location specified by the first operand (address word). The format of the sine value is a 16-bit fractional number from -1.0 to +1.0. The accuracy of the sine value is 2^{-14}.

The format of the GSIN order is:



The GSIN order contains the following values:

Name	Bit(s)	Value
Set Mode	0–7	X'28' or X'2A' or X'2C' or X'2E'
Mode Control	8–15	X'E7'
Address Word	0–15	X'aaaa'
Data Word	0–15	X'0ddd'

Address Word Variable

X'aaaa' Denotes the display storage address within the current page (that is, a word boundary) of a word into which the sine is to be stored. The address may be relative or absolute, depending on the SM byte option selected. If relative addressing is used, valid values are in the range from -32,768 to +32,766. Negative numbers are in twos complement. Absolute address values may be in the range from 0 to 65,534. The low-order bit of the address is ignored and assumed to be zero.

Data Word Variable

X'0ddd' The data word contains a 12-bit binary number from 0 to 4095, representing the number of 2048th parts of 'pi'; that is, the value of the angle in radians is given by:

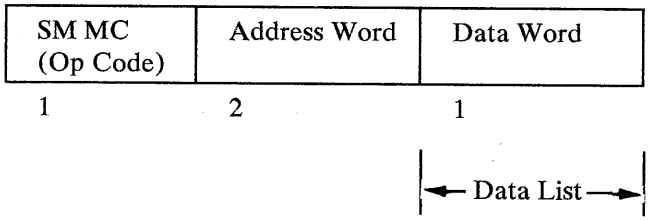
$$\frac{\text{X'0XXX'}}{\text{X'0800'}} \times \text{Pi}$$

The condition code remains unchanged following execution of the GSIN order.

GCOS—Cosine

In the GCOS order the cosine of the angle defined by the second operand (data word) is stored in the location specified by the first operand (address word). The format of the cosine value is a 16-bit fractional number from -1.0 to +1.0. The accuracy of the cosine value is 2^{-14}.

The format of the GCOS order is:



The GCOS order contains the following values:

Name	Bit(s)	Value
Set Mode	0–7	X'28' or X'2A' or X'2C' or X'2E'
Mode Control	8–15	X'E6'
Address Word	0–15	X'aaaa'
Data Word	0–15	X'0ddd'

Address Word Variable

X'aaaa' Denotes the display storage address within the current page (that is, a word boundary) of a word into which the cosine is to be stored. The address may be relative or absolute, depending on the SM byte option selected. If relative addressing is used, valid values are in the range from -32,768 to +32,766. Negative numbers are in twos complement. Absolute address values may be in the range from 0 to 65,534. The low-order bit of the address is ignored and assumed to be zero.

Data Word Variable

X'0ddd' The data word contains a 12-bit binary number from 0 to 4095, representing the number of 2048th parts of 'pi'; that is, the value of the angle in radians is given by:

$$\frac{X'0XXX'}{X'0800'} \times Pi$$

The condition code remains unchanged following execution of the GCOS order.

TCF Stack Control Orders

This section describes additional stack control orders required to push/pop TCF-related information into/from the graphics system stack. These orders, along with the orders described in "Stack Control Orders" on page 5-57, define the complete set of stack control orders.

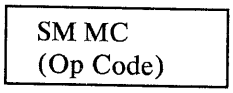
The four TCF stack control orders described in this section are summarized in the following table.

Mnemonic	Name	Function
GPTM	Push Transformation Matrix	Writes into the stack a GLTM order containing all the current matrix elements and the scale factor
GPCB	Push Clipping Boundaries	Writes into the stack a GLCB order containing the current clipping boundaries
GPVPT	Push Viewpoint	Writes into the stack a GLVPT order containing the current viewpoint
GPVB	Push Viewport Boundaries	Writes into the stack a GLVB order containing the current viewport boundaries

GPTM—Push Transformation Matrix

The one-word GPTM order pushes (writes) into the stack a Load Transformation Matrix (GLTM) order containing a mask word, the current matrix elements and the scale factor known to the graphics system. The bits of the mask word are set according to the d bit setting in the GLTM order MC extension word. (Refer to "Mask Word Variables" in the GLTM order.) The cc bits in the MC extension word of the GLTM order are set to 11, indicating that any element of the transformation matrix not supplied is defaulted to zero. The GLTM order is followed by X'2AFE0020', where 0020 is the length (in hexadecimal) of the GLTM order in bytes.

The format of the GPTM order is:



The GPTM order contains the following values:

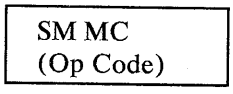
Name	Bit(s)	Value
Set Mode	0–7	X'2A'
Mode Control	8–15	X'29'

During the push operation, if the address in the stack current pointer register is equal to or greater than that in the stack limit register, the GPTM order is not executed; the display program is terminated and a stack error is indicated.

GPCB—Push Clipping Boundaries

The one-word GPCB order pushes (writes) into the stack a Load Clipping Boundaries (GLCB) order containing a mask word and the current clipping boundaries known to the graphics system. Mask bits m1 to m6 are set to 1; the remaining bits are set to zero. The GLCB order is followed by X'2AFE0010', where 0010 is the length (in hexadecimal) of the GLCB order in bytes.

The format of the GPCB order is:



The GPCB order contains the following values:

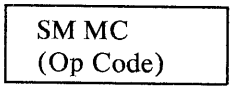
Name	Bit(s)	Value
Set Mode	0–7	X'2A'
Mode Control	8–15	X'2A'

During the push operation, if the address in the stack current pointer register is equal to or greater than that in the stack limit register, the GPCB order is not executed; the display program is terminated and a stack error is indicated.

GPVPT—Push Viewpoint

The one-word GPVPT order pushes (writes) into the stack a Load Viewpoint (GLVPT) order containing the current value of the apex of the viewing pyramid followed by X'2AFE0004', where 0004 is the length, in bytes, of the GPVPT order. The stack pointer is set to point to the length field.

The format of the GPVPT order is:



The GPVPT order contains the following values:

Name	Bit(s)	Value
Set Mode	0–7	X'2A'
Mode Control	8–15	X'26'

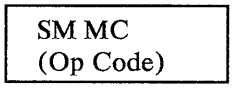
The Z value specified in the GLVPT order represents the value of the current viewpoint.

During the push operation, if the address in the stack current pointer register is equal to or greater than that in the stack limit register, the GPVPT order is not executed; the display program is terminated and a stack error is indicated.

GPVB—Push Viewport Boundaries

The one-word GPVB order pushes (writes) into the stack a Load Viewport Boundaries (GLVB) order containing a mask word and the current viewport boundaries known to the graphics system. Mask bits m1 to m4 of the mask word are set to 1; the remaining bits are set to zero. The GLVB order is followed by X'2AFE000C', where 000C is the length (in hexadecimal) of the GLVB order in bytes.

The format of the GPVB order is:



The GPVB order contains the following values:

Name	Bit(s)	Value
Set Mode	0–7	X'2A'
Mode Control	8–15	X'1C'

During the push operation, if the address in the stack current pointer register is equal to or greater than that in the stack limit register, the GPVB order is not executed; the display program is terminated and a stack error is indicated.

Chapter 6. RS232C Attachment

The 3270 and RS232C Attachment Feature (subsequently referred to in this chapter as simply the RS232C Attachment Feature) provides the ability to output hardcopy of host data by remote attachment to the 5085 Graphics Processor. A plotter can be attached by a general-purpose interface (RS232C/CCITTV24), allowing flexibility of device types.

The RS232C Attachment Feature provides two RS232C interfaces (ports) for the attachment of up to two devices. (See Figure 6-1.)

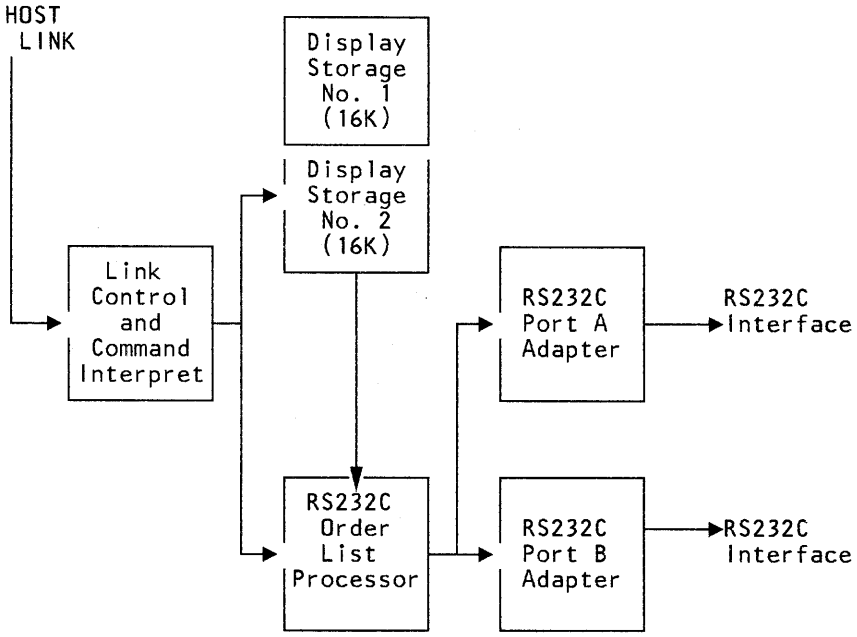


Figure 6-1. Graphics System RS232C Attachment Feature Virtual Machine

The RS232C ports support data rates up to 9600 baud for each asynchronous port and are suitable for the attachment of devices such as the IBM 7374 or IBM 7375 Color Plotters. Each RS232C port requires a separate host address and reduces by one the number of addresses that can be used to attach to a 5088 Graphics Channel Controller or a 3258 Control Unit.

Each RS232C port has its own 16K display storage space. A user-defined display program operating in this display storage space controls the port in a manner similar to the 5081 Display (see "Display Programs" on page 3-3). A subset of 5080 Graphics System graphic orders plus two additional orders, Begin I/O Processing (GBGIOP) and Wait on Timer (GWAIT), unique to the RS232C Attachment Feature, control the transmission of data to and from the RS232C ports.

The display programs are loaded, started, and stopped using the same channel commands used for the 5081 Display. If both ports are installed, a program may be loaded for each, and started and stopped individually by means of Start Display Program and Stop Display Program structured fields, respectively, sent by

means of a Write Structured command; or by means of Set Buffer Address Register and Start (or Stop) commands. In this way, there may be a running program for each port.

All formatting of plot data (scaling, rotation, and so on) must be done in the host, because the responsibility of the RS232C Attachment Feature is mainly to buffer the data before transmission to the plotter. All plot data should be transferred in vector form. Specific commands from the host to control the plotter are embedded in the data. If line-to-raster conversion is required for performance reasons, it is recommended that the plotter or plotter controller, and not the host application program, do the conversion. Transmission from the processor to the plotter is performed using an asynchronous line control. The host program is also responsible for inserting all transmission framing characters before sending the data to the processor.

All plotter data sent by the host to the processor is controlled by a GBGIOP order that points to an 8-byte I/O control block (IOCB) containing port command codes, flags, data count field, and address from which the plotter data will be read. Display program execution is started using the Write Structured command containing Set Display Storage Address Register and Start Display Program structured fields, which initiate normal processing of the graphic orders in display storage until a GBGIOP order is encountered that points to an IOCB. This IOCB may then be used to address output configuration control (OCC) bytes to configure the I/O port. All data to be plotted is now transferred to the plotter under control of additional IOCBs that point to the display storage address location of this data. Further chained IOCBs may be used to transfer additional data, if required. Plotting continues until a GBGIOP order termination sequence occurs (count field decremented to zero, Stop Display Program structured field, and so on).

Note: The sequence described above could also have been started by a Set Buffer Address Register and Start command.

Plotting begins when the appropriate plotter command has been interpreted by the plotter from the data.

Notes:

1. It is also necessary to issue IOCBs containing reads to receive acknowledgments and status from the plotter. In these cases, the RS232C port must be configured for input by addressing input configuration control (ICC) bytes in an IOCB.

2. When using GAM/SP, the RS232C Attachment Feature must be SYSGENed as a 2250-3. In this case, only 3250-compatible channel commands are addressed to the RS232C ports (see Appendix A).

RS232C Channel Operations

The following channel commands are executed identically on the RS232C Attachment Feature as they are on the base 5080 Graphics System except as noted (see Chapter 4 for descriptions of 5080 Graphics System channel commands and Appendix A for descriptions of 3258 channel commands).

- No-Operation (Note 1)
- Read Buffer

- Read Cursor (Note 2)
- Read Manual Input (Notes 1 and 3)
- Read Memory Area (Note 1)
- Read X-Y Position Register (Note 4)
- Select Read Memory Area (Note 1)
- Select Write Memory Area (Note 1)
- Sense (Notes 1 and 5)
- Sense ID (Note 1)
- Set Buffer Address Register and Start
- Set Buffer Address Register and Stop
- Set Mode (Note 1)
- Write Memory Area (Note 1)
- Write Buffer
- Write Structured (Note 1)

Notes:

1. 5080 Graphics System channel command.

2. The Read Cursor command returns zero data. Channel End and Device End are returned separately.

3. The manual input register can be set only by means of a simulated program function key, ENTER key or Cancel (Cncl) key from the Begin Order Processing (GBGOP) order and returns zeros at all other times in response to the Read Manual Input command. A manual input register is maintained for each port.

4. The Read X-Y Position Register command returns zero coordinate data.

5. Normal sense data is returned (according to the latest Set Mode command), but sense bits not related to the feature are meaningless and should be ignored.

6. The following commands are treated as No-Operation commands when sent to an RS232C port:

    - Insert Cursor
    - Remove Cursor
    - Set Audible Alarm
    - Set LPFK Indicators

Write Structured Command—Structured Field Processing

The following structured fields are processed normally when addressed to an RS232C port:

- Set Display Storage Address Register
- Start Display Program
- Stop Display Program

The following structured fields are processed but have no effect (that is, perform no operation) if addressed to an RS232C port:

- Sound Alarm
- Set All Indicators

- Set Selected Indicators On
- Set Selected Indicators Off
- Set Cursor Address Register
- Reset Cursor Address Register
- Load Blinking Patterns
- Load Line Patterns

The following structured fields are rejected with a program error and a structured field error if addressed to an RS232C port:

- Define Memory Area
- Rename Memory Area
- Delete Memory Area

Refer to "Structured Fields" on page 4-13 for details.

RS232C Port Control Orders

The following graphics system orders execute from a display program associated with the RS232C Attachment Feature in exactly the same manner as described in "Display Programs" on page 3-3. See Chapter 5 for details on the individual orders.

Mnemonic	Name	SM/MC Op Code
GADD	Add Data	X'2AEE'
GBC	Branch on Condition	X'2A70'–X'2A7F'
GBCNT	Branch on Count	X'2AF0'
GBPAGE	Branch Page	X'2AF7'
GEOP	End Order Processing	X'2A81'
GMVA	Move Address	X'2AEB'
GMVW	Move Word	X'2AEC'
GNOP2	2-Byte No-Operation	X'2A80' X'2AB0'–X'2ABF'
GNOP4	4-Byte No-Operation	X'2AC0'
GSCONF	Store Configuration Data	X'2ADF'
GSUB	Subtract Data	X'2AEF'
GTM	Test under Mask	X'2AED'

Note: The allowable range of values for page numbers assigned to the RS232C Attachment Feature is from 0 to 255. However, because of the size of storage assigned to the feature, page numbers other than zero result in a display program error.

The following 5080 Graphics System base order is modified when used by the RS232C Attachment Feature (the differences are described below):

GBGOP	Begin Order Processing	X'2A8A'

The following orders are unique to the RS232C Attachment Feature:

GBGIOP	Begin I/O Processing	X'2A89'
GWAIT	Wait on Timer	X'2A8F'

Notes:

1. All other orders are treated as no-operations.

2. An RS232C port control display program runs continuously until the occurrence of a GEOP order in the display program, a Set Buffer Address Register and Stop command, or a Stop Display Program structured field from the channel.

If a Set Buffer Address Register and Start command is received, the display program is stopped immediately and restarted at the specified address. If a Start Display Program structured field is received, the display program is allowed to continue until the next GBGOP order is executed. At this time the display program is resumed at the new display storage address established by a previous Set Display Storage Address Register structured field.

RS232C Attachment Feature Control Registers

The RS232C Attachment Feature contains a *display storage address register (DSAR)* and a *regeneration address register (RAR)* for each port. (See "Display Programs" on page 3-3 for details on these registers.) In addition, a unique 2-bit condition code register and a manual input register are provided for each port (see "RS232C Port Condition Codes" below for details).

The only control registers uniquely associated with the RS232C Attachment Feature are *output configuration control (OCC), input configuration control (ICC),* and *termination character (TC)* bytes. One set of these registers is provided for each port installed. They are set by means of control IOCBs, as described in "Control Command Operation" on page 6-9.

RS232C Port Condition Codes

A separate 2-bit condition code register is provided with each RS232C port to retain certain conditions that may occur while RS232C port order programs are being processed.

The contents of this register can be tested by the Branch on Condition (GBC) order (see "Branch Control Orders" on page 5-53 for details).

The condition code register is set under the following conditions:

Setting Condition	Condition Code Value 00	01	10	11
GBGOP and SBA Start WS Start	Set	Not Set	Not Set	Not Set
GADD	Sum = 0	Sum < 0	Sum > 0	Overflow
GBGIOP	Normal End	Abnormal End	Not Set	Not Set
GSUB	Diff. = 0	Diff. < 0	Diff. > 0	Overflow
GTM	All 0s	Mixed 0s&1s	Not Set	All 1s

Note: For more details on the condition code settings, see the individual graphic order descriptions.

The remaining orders do not affect the condition code setting. The setting is preserved until changed by one of the conditions given above.

GBGOP Order Interpretation

A Begin Order Processing (GBGOP) order in an RS232C Attachment Feature display program operates in the following manner:

- The display program does not stop or wait for a frame synchronization timer; it restarts immediately.
- The d, e, and k bits in the MC extension (flags) are ignored and should be set to zero.
- The s and tt bits are interpreted as described in "Frame Control Orders" on page 5-6. These controls can be used to initiate a simulated program function key, ENTER key, or Cancel (Cncl) key interrupt.

Note: There is no GBGOP order timeout in the RS232C Attachment Feature.

RS232C Transmission Control

A new two-word order, Begin I/O Processing (GBGIOP), is provided to control the transmission of data between buffer storage and RS232C ports.

GBGIOP—Begin I/O Processing

The GBGIOP order causes an operation to be performed at the RS232C port interface adapter. The operation, or sequence of operations, is defined by an *input/output control block (IOCB)* or chain of IOCBs located at the address indicated by the address field of the order (the low-order bit of the address is ignored and assumed to be zero). Valid values for the address range from 0 to 16K.

The format of the GBGIOP order is:

SM MC (Op Code)	MC Extension (Reserved)	Address Word
1	2	3

The GBGIOP order contains the following values:

Name	Bit(s)	Value
Set Mode Mode Control	0–7 8–15	X'2A' or X'2E' X'89'
Mode Control Extension Word (Reserved)	0–15	X'00'
Address Word	0–15	X'aa'

The MC extension word is reserved and should be set to zero.

Input/Output Control Block (IOCB) Format

The IOCB has a fixed length of 8 bytes in the following format:

Byte 0	Byte 1
Command Code	Flags
Count Field	
Data Word 1 Page Number (or data bytes 0 and 1)	
Data Word 2 Data Address (or data bytes 2 and 3)	

0 15

Command Code Assignments:

0000	0001	Write
0000	0010	Read
mm00	0011	Control
0000	0100	Sense
0000	1000	Transfer Control

Note: All other values of the command code are invalid.

mm = 00 No-operation
mm = 01 Set OCC bytes from data field
mm = 10 Set ICC bytes from data field
mm = 11 Set TC byte from data field

Flag Byte Bit Assignments:

0000	0ics

i = 1 Immediate data; that is, the count can be up to 4, and the data is contained in words 2 and 3 of the IOCB.

i = 0 The data is addressed by the page number and data address given in the IOCB.

c = 1 Chaining flag; at the successful completion of this IOCB, the next IOCB is fetched and executed from the next sequential location in buffer storage after this IOCB.

c = 0 No chaining; this is the last IOCB

s = 1 Suppress length indication; inhibits indication of incorrect length in termination status.

s = 0 Indicates any incorrect length indication in the termination status (also terminates chaining on an incorrect length indication).

Count Field:

This 2-byte field contains a count of the number of bytes of data to be read or written by the IOCB.

Data Field:

Immediate Data (i = 1 in the flag byte)

The data is contained within the IOCB itself. In this case, the count can be up to 4, and the data is contained in words 2 and 3 of the IOCB (data words 1 and 2).

Indirect Data (i = 0 in the flag byte)

The data is addressed by a page number and data address given in words 2 and 3 of the IOCB, respectively:

Page Number (data word 1)

The allowable range of values for page number is zero to 255; however, because of the size of storage assigned to the RS232C Attachment Feature, page numbers other than zero result in a display program error.

Crossing a page boundary causes a display program error and no wrapping of addresses takes place.

Data Address (data word 2)

This is the data address within the page specified by page number (data word 1). The valid values for this field are in the range from zero to 16,383.

Note: Chaining is terminated under any condition that causes the status (sense byte 0) to be nonzero.

Control Command Operation

The control command byte initiates operations that control the RS232C port adapter to:

- Set up the conditions for a read operation by setting the input configuration control (ICC) register and the termination character (TC).
- Set up the conditions for a write operation by setting the output configuration control (OCC) register.
- Perform a no-operation.

The TC byte is used to terminate read operations (see "Read Command Operation" on page 6-11).

The formats of the OCC bytes are:

Byte 0

Bit	Value	Function
0(MSB)	0	Reserved
1	0 1	Use Data Terminal Ready (RS232C signal) Data Terminal Ready ignored
2 3 4	X X X	Transmit speed (see below)
5–6	0	Reserved
7(LSB)	0 1	Odd parity generated Even parity generated

Byte 1

Bit	Value	Function
0(MSB)	0	Reserved
1	0 1	Asynchronous transmission Reserved
2	0 1	7-bit data byte (asynch) 8-bit data byte (asynch)
3	0 1	2 stop bits (asynch) 1 stop bit (asynch)
4–7	0	Reserved

Transmit Speed

Bit			Speed (bauds)
2	3	4	
0	0	0	150
0	0	1	300
0	1	0	600
0	1	1	1200
1	0	0	2400
1	0	1	4800
1	1	0	9600
1	1	1	Reserved

The formats of the ICC bytes are:

Byte 0

Bit	Value	Function
0(MSB)	0	Reserved
1	0 1	Activate Clear to Send only when Request to Send is on Request to Send ignored (Clear to Send permanently on)
2 3 4	X X X	Receive speed (see below)
5	0 1	Auto-terminate mode Non-auto-terminate mode
6	0 1	No parity check on data Parity check on data
7(LSB)	0 1	Receive odd parity Receive even parity

Note: Regardless of whether parity checking is enabled, the RS232C Attachment Feature expects to receive parity bits on all read operations.

Byte 1

Bit	Value	Function
0(MSB)	1	Reserved
1	0	Asynchronous transmission
2	0 1	7-bit data byte (asynch) 8-bit data byte (asynch)
3	0 1	2 stop bits (asynch) 1 stop bit (asynch)
4	0 1	Do not modify high-order data bit Zero high-order data bit
5 6 7	0 0 0	Reserved Reserved Reserved

Receive Speed

Bit			Speed (bauds)
2	3	4	
0	0	0	150
0	0	1	300
0	1	0	600
0	1	1	1200
1	0	0	2400
1	0	1	4800
1	1	0	9600
1	1	1	Reserved

Read Command Operation

The read command operation initiates the input of data from the RS232C ports to the display storage address indicated in the IOCB. Data transfer continues in the manner indicated by the ICC and TC bytes until terminated in one of the following ways:

- Auto-Terminate Mode

  The read transmission is terminated by:

  - Detection of the termination character (set into the TC byte).
  - A 1.2-second delay since the receipt of the last character.
  - Count in the IOCB goes to zero.
  - Detection of a parity error on the RS232C interface (if bit 6 is 1 in the ICC byte).

- Receipt of a Start Display Program or Stop Display Program structured field or a Set Buffer Address Register and Start (or Stop) command via the channel. Refer to Note 2 under "RS232C Port Control Orders" on page 6-4 for details.

- Non-Auto-Terminate Mode

  The read transmission is terminated by:

  - A 1.2-second delay since the receipt of the last character.
  - Count in the IOCB goes to zero.
  - Detection of a parity error on the RS232C interface (if bit 6 is 1 in the ICC byte).
  - Receipt of a Start Display Program or Stop Display Program structured field or a Set Buffer Address Register and Start (or Stop) command via the channel. Refer to Note 2 under "RS232C Port Control Orders" on page 6-4 for details.

Note: The high-order bit (parity bit) in each data byte can be forced to zero by setting byte 1, bit 4 in the ICC to 1.

Write Command Operation

The write command operation initiates the transfer of data from display storage, at the address given in the IOCB, to the RS232C port. Data transmission continues in the manner indicated in the OCC byte until terminated by:

- Count in the IOCB goes to zero.
- Data Terminal Ready not signaled for 5 seconds (if bit 1 is zero in the OCC byte).
- Receipt of a Start Display Program or Stop Display Program structured field or a Set Buffer Address Register and Start (or Stop) command from the channel. Refer to Note 2 under "RS232C Port Control Orders" on page 6-4 for details.

Transfer Control Command Operation

No operation is performed except that a new IOCB is fetched from the data address, using the page number and the data address field.

Bit 15 of the address field is ignored.

Note: The flag byte is ignored.

Sense Command Operation

The sense command operation causes the status of the RS232C adapter to be stored in display storage at the data address given in the IOCB. Depending on the count field, a maximum of 8 bytes is stored.

The format of the sense data is:

- Byte 0 is the status byte (see below).
- Byte 1 contains the page number of the last IOCB used (only zero is valid).
- Bytes 2 and 3 contain the residual count from the preceding operation.
- Bytes 4 and 5 contain the setting of the configuration register (ICC or OCC) of the last operation.
- Bytes 6 and 7 contain the address of the last IOCB executed.

The status byte has the following format:

000i rcte

i = 1 Invalid IOCB

r = 1 Overrun error; a character was lost as a result of an overrun or a framing error

c = 1 Incorrect record length

t = 1 Timeout termination

e = 1 Read parity error

Notes:

1. The conditions indicated by a sense command operation are reset by any subsequent I/O operation (that is, IOCB) except the sense operation itself. A Begin Order Processing (GBGOP) order also resets the sense conditions. The sense conditions are *not* reset as the result of a Start Display Program or Stop Display Program structured field or a Set Buffer Address Register and Start (or Stop) command from the host.

2. An incorrect-length indication is set after a read command operation if there is a difference between the IOCB count and the amount of actual data read when the TC character is encountered.

GBGIOP Termination

The GBGIOP order does not terminate until the operations indicated in the associated IOCB(s) have completed their functions. At that time, graphic order processing is resumed at the next order.

Note: A Set Buffer Address Register and Start (or Stop) command, a Stop Display Program structured field or a Start Display Program structured field from

the channel to the RS232C port address terminates the GBGIOP order even though in the middle of an operation (appropriate status is stored for a subsequent sense operation). A nonzero Sense status byte also causes termination of the GBGIOP order.

After a GBGIOP order terminates, the condition code (cc) is set to indicate the result of the operation(s).

cc	=	00	Normal termination
cc	=	01	Abnormal termination (status byte in sense data is nonzero)
cc	=	10	Not set
cc	=	11	Not set

Timeouts

Reads

A 1.2-second timer is set on a receive operation for every character after the first character has been received. If this timer expires prior to the receipt of the next character of the transmission, the receive operation is terminated and Timeout Termination is indicated in the status byte.

Writes

If the Data Terminal Ready (DTR) bit is set in the OCC (byte 1, bit 1 is zero), the DTR line is monitored during transmission. The transmission is terminated if DTR is off for 5 seconds and Timeout Termination is indicated in the status byte.

External Status Change

If 500 or more external status change interrupts occur from an RS232C port within 100 milliseconds, the port is shut down and Timeout Termination is indicated in the status byte.

GWAIT Order Interpretation

GWAIT—Wait on Timer

The GWAIT order is an additional order provided in the RS232C Attachment Feature to allow the user to build timed wait loops without causing interference to other functions (for example, another port or the graphic display) running in the graphics system.

The format of the GWAIT order is:

SM MC (Op Code)	MC Extension
1	2

The GWAIT order contains the following values:

Name	Bit(s)	Value
Set Mode Mode Control	0–7 8–15	X'2A' X'8F'
Mode Control Extension Word	0–15	X'00tt'

The two-word GWAIT order causes order processing to be suspended for the period of time indicated by the t bits in the MC extension word. The t bits indicate the number of milliseconds (approximately) in the time interval, within a range from 0 to 255 milliseconds.

After the indicated time has expired, display program processing continues at the next order following the GWAIT order.

Appendix A. 3250-Compatible Channel Commands and Graphic Orders

This appendix describes the IBM 3250 Graphics Display System channel commands and graphic orders supported by the IBM 5080 Graphics System.

To use this appendix, you should be familiar with the 5085 Graphics Processor architecture defined in Chapter 3. Also, to ease the transition from the 3250 system to the 5080 system, the terminology used in this appendix is generally consistent with the *IBM 3250 Graphics Display System Component Description,* GA33-3037.

The following list of key 3250 terms, and, in the context of 3250 compatibility, their 5080 equivalents, is presented as an aid to making the appropriate correlation:

3250	5080
Buffer address register (BAR)	Display storage address register (DSAR)
Buffer running	Display program running
Display list buffer	Display storage
Display list buffer address	Display storage address
Display list buffer program	Display program
END key	ENTER key
Null key	Character delete key
Overstrike key	Backspace (reqd.) key
Point	Marker
Vector	Line

3250 Channel Command Set

The complete 3250-compatible command set and the code point of each command are listed below:

3250 Commands	Hex Code	Type
Write Buffer	01	Write
Read Buffer	02	Read
Read Manual Input	0E	Read
Read Cursor	06	Read (see Note)
Read X-Y Position Register	12	Read (see Note)
No-Operation	03	Control
Set Buffer Address Register and Start	27	Control
Set Buffer Address Register and Stop	07	Control
Set Audible Alarm	0B	Control (see Note)
Set LPFK Indicators	1B	Control (see Note)
Insert Cursor	0F	Control (see Note)
Remove Cursor	1F	Control (see Note)
Sense	04	Sense

Note: These channel commands are modified if addressed to an RS232C Attachment Feature. (See "RS232C Channel Operations" on page 6-2.)

Write Buffer

The Write Buffer command causes the data bytes received across the channel following the command to be placed into consecutive buffer locations in the 5088 Graphics Channel Controller. They are then transferred into consecutive locations in the display list buffer associated with the addressed device in the 5085 Graphics Processor.

The starting buffer address in the display list buffer will normally have been preset by a Set Buffer Address Register and Stop command (or by a Write Structured command containing a Set Buffer Address Register structured field).

Note: During the processing of this command, the display list buffer is not considered to be memory area X'FFFE'. The value of the current transfer address of memory area X'FFFE' is not changed by this command.

Data transmission is terminated under channel byte count control when the number of bytes specified by the channel program has been transferred. If the channel program attempts to write past the last display list buffer location of the current page, wraparound occurs (writing continues from the first display list buffer location of the current page); this destroys any data previously stored in these locations. The processor wraps at the page boundary regardless of where it is established (for example, 32K, 48K, 56K). The maximum page size supported by the processor is 64K. Writing into a display list buffer location that contains a cursor causes the cursor to be removed.

The Write Buffer command provides separate Channel End and Device End status indications.

Read Buffer

The Read Buffer command causes the transfer of consecutive data bytes from the display list buffer of the appropriate device to the channel via the controller and the I/O interface.

The value in the *buffer address register (BAR)*—actually, the 5085's display storage address register—for the selected device defines the starting location for the data. The read operation is terminated by the channel byte count control, which determines that the number of bytes specified by the channel program has been read. If the channel attempts to read past the last display list buffer location of the current page, wraparound occurs (reading continues from the first display list buffer location of the current page). The processor wraps at the page boundary regardless of where it is established (for example, 32K, 48K, 56K). The maximum page size supported by the processor is 64K. Note that whenever a location that contains a cursor is read, only the data and not the cursor is sent to the channel; the cursor is preserved in the same location.

Reading does not change the contents of the display list buffer.

Note: When processing this command, the display list buffer is not considered to be memory area X'FFFE'. The value of the current transfer address of memory area X'FFFE' is not changed by this command.

The Read Buffer command provides separate Channel End and Device End status indications.

Read Manual Input

The Read Manual Input command transfers alphanumeric keyboard (ANK) or lighted program function keyboard (LPFK) information to the channel. Pressing the ANK END or Cancel (Cncl) keys or any LPFK key causes the Attention bit to be set in the status byte if the status and sense registers are clear.

The Attention status is passed from the processor to the controller at the next opportunity. The controller then passes the Attention status to the channel at the earliest appropriate time, such as at the end of current channel operations, causing a separate asynchronous I/O interrupt.

When an END, Cancel, or LPFK key is pressed, the keyboards remain logically locked until the channel program responds with a Read Manual Input command to determine which key was pressed. This command does not affect buffer order processing. The Read Manual Input command or the pressing of the Reset key resets the data and unlocks the keyboards if they are locked. If there is no manual input data, the data returned is zeros in the first 2 bytes and X'FF' in the third byte.

The Read Manual Input command provides Channel End and Device End status indications together.

Alphanumeric Keyboard (ANK)

If an alphanumeric keyboard (ANK) END or Cancel key caused the Attention status, the processor sends 3 bytes to the channel in response to a Read Manual Input command. Bits 0 and 1 of byte 0 are set to 10 to indicate that an ANK key was pressed. Bytes 1 and 2 always contain all zeros.

Byte 0	Byte 1	Byte 2
10ec 0000	All zeros	All zeros
0 7	0 7	0 7

Notes:

1. Bits 0 and 1 of byte 0 are 10, indicating the response is for the ANK.
2. Bit 2 of byte 0 is 1 if the END key caused the Attention status.
3. Bit 3 of byte 0 is 1 if the Cancel key caused the Attention status.

Lighted Program Function Keyboard (LPFK)

If the lighted program function keyboard (LPFK) caused the Attention status, the processor sends 3 bytes to the channel in response to the Read Manual Input command. Bits 0 and 1 of byte 0 are set to 01 to indicate that one of the 32 LPFK keys has been pressed. Byte 1 contains a 5-bit binary code that corresponds to the number of the key.

Byte 0	Byte 1	Byte 2
0100 0000	0 0 0 Key Code	Note 3
0 7	0 1 2 3 7	0 7

Notes:

1. Bits 0 and 1 of byte 0 are 01, indicating the response is for the LPFK.

2. The relationship of the key code in byte 1 to the pressed key is:

Pressed Key	Key Code (Bits 3–7)
0	00000
1	00001
.	.
.	.
30	11110
31	11111

3. Byte 2 is set to X'FF'.

ANK Program Function (PF) Keys (A 5080 Extension)

If an ANK program function (PF) key caused the Attention status, the processor sends 3 bytes to the channel in response to the Read Manual Input command. Bits 0 and 1 of byte 0 are set to 01 and bit 4 of byte 0 is set to 1 to indicate that one of the 24 program function keys has been pressed. Byte 1 contains a 5-bit binary code that corresponds to the number of the key.

Byte 0								Byte 1								Byte 2							
0	1	0	0					0	0	0	Key Code					X'FF'							
0							7	0							7	0							7

Notes:

1. Bits 0 and 1 of byte 0 are 01 and bit 4 of byte 0 is 1, indicating the response is for an ANK PF key.

2. The relationship of the key code in byte 1 to the pressed key is:

Pressed Key	Key Code Bits (3–7)
1	00000
2	00001
.	.
.	.
.	.
24	10111

3. The user may treat a response from the ANK PF keys as coming from a unique input source of 24 keys. Optionally, by ignoring the setting of bit 4 of byte 0, the user may treat a response from the PF keys as an alternative LPFK input source of 24 keys whether or not an LPFK is configured at the processor.

4. A setup option is also provided at the processor to allow the user to specify the setting of bit 4 of byte 0. The system default is to set bit 4 of byte 0 to 1, indicating the ANK as the source.

When the processor is powered on, the manual input is reset.

Read Cursor

The Read Cursor command, like the Read Buffer command, reads consecutive data bytes from the display list buffer to the channel via the controller and the I/O interface. However, data transmission is terminated by whichever of the following occurs first:

- Channel byte count control determines that the number of bytes specified by the channel program has been read.
- A display list buffer location to which a cursor is assigned is encountered.

If no cursor is present for the graphics system workstation, reading continues until stopped by the channel. Wraparound occurs if necessary.

When a cursor is encountered during a Read Cursor command, if the display list buffer program is stopped, the code X'1A' is transmitted to the channel; the byte from the location to which the cursor is assigned is not transmitted and both this byte and the cursor are preserved. If the display list buffer program is running, the delimiting X'1A' value is not returned as the last data byte, which is terminated by reading a byte containing the cursor. (The byte at the cursor location is read.)

Reading does not change the contents of the display list buffer.

If the cursor location is found before the channel control word (CCW) byte count is satisfied, the 'ending' status is Channel End and Device End indications together. If the byte count is satisfied before the cursor is found, only Channel End is provided; Device End is provided when the device is ready to accept another command.

Sense bytes 2 and 3 contain the display list buffer address at which the Read Cursor command stops when the processor is attached to a 3258 or only 4 sense bytes are requested via the 5080 Set Mode command; otherwise, the display list buffer address is in sense bytes 2, 3, 6, and 7.

Read X-Y Position Register

The Read X-Y Position Register command transfers the contents of the X-Y position registers of the 5085's X-Y-Z position registers from the processor to the channel via the controller and the I/O interface. The Z value is ignored and the four high-order bits of the X and Y values are set to zero. Therefore, in response to this command, the controller returns 4 bytes (a 2-byte X coordinate and a 2-byte Y coordinate) from the processor. The format of the returned bytes is:

Word 1	0000	xxxx	xxxx	xxxx	X Coordinate
Word 2	0000	yyyy	yyyy	yyyy	Y Coordinate

This command may be issued in response to an interrupt for a pick detection, in which case the contents of the X-Y position register depend on the mode in which the detection occurred.

- In graphics mode, the returned coordinates are the absolute coordinates of the current position.

- In character mode when CSID = X'00' (see attribute register 18 under "Attribute Control Orders" on page 5-31), the returned coordinates are the absolute coordinates of the character center point following the character on which the detection occurred. This is the 3250 default.

- In character mode when CSID is not zero (see attribute register 18), the returned coordinates are the absolute coordinates of the character lower left corner point following the character on which the detection occurred.

If the command is not sent in response to a pick detect interrupt, the value of the X-Y position registers is returned, but the significance of the value depends on the buffer program and the point at which it was stopped. If the buffer is running, the value returned in response to this command is undefined.

The Read X-Y Position Register command provides Channel End and Device End status indications together.

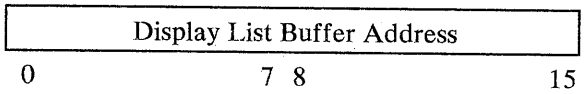
No-Operation

The No-Operation command performs no operation. It is an immediate command and no data bytes are transferred.

This command provides Channel End and Device End status indications together.

Set Buffer Address Register and Start

The Set Buffer Address Register and Start command initiates graphic order processing for the selected device. The command also resets any Attention interrupt condition for the device that has not been accepted by the controller. Two bytes of data follow this command and specify the display list buffer location at which graphic order processing is to start.



The command causes graphic order processing to stop, if it has not already stopped. The 2 bytes are passed to the *buffer address register (BAR)* and then to the *regeneration address register (RAR)* of the addressed device. The page number contained in the BAR is set to zero and also passed to the RAR. The command then initiates graphic order processing for the addressed device.

The display list buffer address provided by this command should always have an even-numbered address and point to the first byte (the set mode byte) of an order. If neither condition is met, sequential display list buffer locations are read and decoded until a *set mode (SM)* byte is decoded from an even address. At this time the byte in the next sequential location is read and decoded, and graphic order processing is started. The effect is to round down an odd starting address to the preceding even boundary. The least significant bit is maintained until the next

Set Buffer Address Register and Start (or Stop) command. Any sense data following an End Order Processing (GEOP)/End Order Sequence (GEOS) order, pick interrupt, and so on, returns the display list buffer address with the least significant bit set.

The Set Buffer Address Register and Start command sets default values of attributes, pick modes, and indicators, except for the single detect indicator (SDI), in the same way as the Start Regeneration Timer (GSRT) order. Current position is not affected. The SDI is set off if all stylus switches are open and no function key is pressed or a pick detect interrupt was pending that resulted from a switch enabled detect for the workstation; otherwise, the SDI is not changed. Any keyboard interrupt not yet sent to the controller is reset.

A pick interrupt will be pending if a pick detect that should normally result in an interrupt at the host has occurred on a previously displayed image on the workstation, but the interrupt has not yet been raised. In this circumstance, the interrupt is never raised, but the SDI is set off during the Set Buffer Address Register and Start command. This permits a second pick detect to occur for a single tip switch closure.

The Set Buffer Address Register and Start command provides separate Channel End and Device End status indications.

If either zero or 1 byte of data is passed from the channel to the controller, the command is not executed and is terminated with separate Channel End and Device End, Unit Check; Command Reject is set in the sense data (byte 0, bit 0). If the count field in the CCW specifies more than 2 bytes, an Incorrect Length is indicated unless the *suppress length indicator (SLI)* is on in the CCW.

Note: The default value of the character set identifier (CSID) is set to X'00'.

Set Buffer Address Register and Stop

The Set Buffer Address Register and Stop command stops graphic order processing for the addressed device. It causes order processing to stop and initiates a request to the channel for 2 bytes. When received from the channel, these bytes are placed into the buffer address register for the selected device (the page number is set to zero). The bytes are coded as shown for the Set Buffer Address Register and Start command. No check is made of the contents of the addressed location nor of the current BAR value prior to loading the value specified in the command. If the address set by this command is used for a Read Buffer, Read Cursor, or Write Buffer command, an address on an odd byte boundary is valid; the least significant bit is maintained.

If either zero or 1 data byte is passed from the channel to the controller, the command is not executed and is terminated with separate Channel End and Device End, Unit Check; Command Reject is set in the sense data (byte 0, bit 0). If the count field in the CCW specifies more than 2 bytes, an Incorrect Length is indicated, unless the SLI bit is on in the CCW.

The Set Buffer Address Register and Stop command provides separate Channel End and Device End status indications.

Set Audible Alarm

The Set Audible Alarm command activates, for a short period, a buzzer (single-stroke audible alarm) at the selected device.

The Set Audible Alarm command provides Channel End status at initial status and a Device End status indication at completion.

Set LPFK Indicators

The Set LPFK Indicators command lights and extinguishes the LPFK indicators as specified in 4 data bytes that follow the command byte. These indicators, numbered 0 to 31, are associated with the 4 data bytes as follows:

Byte	Bit Position							
	0	**1**	**2**	**3**	**4**	**5**	**6**	**7**
0	0	1	2	3	4	5	6	7
1	8	9	10	11	12	13	14	15
2	16	17	18	19	20	21	22	23
3	24	25	26	27	28	29	30	31

An indicator is on (lit) if its associated data bit is a 1; it is off if its associated data bit is a zero. The operation of this command does not affect graphic order processing.

The Set LPFK Indicators command provides separate Channel End and Device End status indications.

Insert Cursor

The cursor indication is set at the display list buffer location addressed by the buffer address register without disturbing the data in that location. Any previous cursor indication is removed first. After insertion, the cursor can be repositioned by ANK action, a Remove Cursor command, another Insert Cursor command, or a Write Buffer command (which replaces the byte at the cursor position). Only one cursor per device may be indicated in the display list buffer; it may be inserted at an even or odd location.

The Insert Cursor command provides a Channel End status indication at initial status and a Device End status indication at completion.

Remove Cursor

The cursor indication is cleared at the display list buffer location addressed by the buffer address register without disturbing the data in that location. If the location does not contain a cursor, the command is executed but performs no operation. A cursor in either byte of a word is removed whether the address in the BAR specifies an even or odd byte.

Note: Any time data is written into a display list buffer location containing a cursor, the cursor indication is cleared.

The Remove Cursor command provides a Channel End status indication at initial status and a Device End status indication at completion.

Note: A "word" is 2 adjacent bytes starting on an even address.

Sense

The Sense command is used to obtain data relative to the status of the addressed device. It can be issued by the host system at any time, but is usually the response to a Unit Check status indication.

The amount of sense data returned depends on the type of device and the controller. The error indicator portions of the sense data are reset following successful completion of this command. Additional conditions for resetting sense data are defined in "Sense Information" on page 4-29.

Because the controller maintains the sense data individually by device, selections to other devices are allowed while one or more devices are in a sense-pending state.

When a Sense command is issued at any other time status is pending for another device on the controller, the controller responds with Busy and Status Modifier status indications at initial selection.

The Sense command provides Channel End and Device End status indications together.

Programming Considerations

If the addressed device is a 3255/Release 1, the Write Buffer, Read Buffer, Read Cursor, Read X-Y Position Register, Insert Cursor, and Remove Cursor commands require that graphic order processing stop before the command can be executed. If the command is rejected because graphic order processing has not ceased, Unit Check status is returned with the Command Reject and Buffer Running bits set in the sense bytes.

Some commands require a fixed number of bytes to be transferred to complete the command successfully. It is possible for the channel to terminate the command before the required number of bytes has been transferred.

- For a Read Manual Input command the data is reset (hence, lost) following channel termination.
- In the case where only 1 address byte is received from the channel on a Set Buffer Address Register and Start command, the controller sets the second byte to X'00' and changes the command to the processor to a Set Buffer Address Register and Stop command.
- If a Set Buffer Address Register and Stop command receives only 1 byte of channel data, the second byte is taken as zero and the graphic order program is stopped, with the buffer address register set accordingly.
- A Set LPFK Indicators command with insufficient bytes transferred acts only on those bytes received and clears all other indicators.

Note: The memory area management function is neither accessible from the base channel commands nor has any effect on their operation. The only available processor storage to the user of the 3250 commands is the first 64K bytes of the display list buffer or to the size of the first or only page of display storage, which may range from 32K to 64K bytes.

Instructions

The normal interaction between the controller and its host CPU is controlled by channel commands; these commands are conveyed to the controller by the Start I/O (SIO) or Start I/O Fast Release (SIOF) instruction from the CPU. Several additional CPU instructions, including Test I/O, Halt I/O, and Halt Device, give the host program a means of access to the processor.

Test I/O

The controller responds to the Test I/O instruction, executed by the CPU, with a status byte. If there is no outstanding status information for the addressed device, an all-zeros status byte is returned. Status for any other device is not reset. Any status for the addressed device is reset at the end of the sequence.

Following an initial selection sequence issued to the controller, the current poll in progress is subjected to a controlled termination except when the command is a Test I/O or a No-Operation, in which case the poll in progress is allowed to continue normally. In addition, if a poll is not in progress, a Test I/O or No-Operation will not prevent initiation of the next poll cycle or prevent the presentation of data from a completed poll cycle. This allows pending interrupt conditions to be cleared by a Test I/O loop, although this is not recommended. The poll in progress or the next poll issued is not necessarily to the device holding the interrupt condition, but, if sequential polling is allowed to continue, a poll is eventually issued to the required device and the interrupt condition is cleared from the device.

Halt I/O and Halt Device

A Halt I/O instruction may cause the channel to issue an interface disconnect sequence to the controller, resulting in the termination of the current I/O operation. This does not affect graphic order processing currently in progress.

The interface disconnect sequence can be issued at various phases of interface activity, including during a data transfer. Any pending status is preserved and provided to the channel after the sequence has been completed.

Halt Device processing is similar to Halt I/O processing. Refer to *IBM System/370 Principles of Operation,* GA22-2700, for a description of the differences.

System Reset

A System Reset causes the controller and all attached devices to be reset. The controller broadcasts a System Reset request to each attached device. Any outstanding status or interrupts are lost. The diagnostics will not be run following a System Reset instruction. Also, the ready/not ready state of attached devices is not changed by a System Reset instruction but may be changed if the reset clears inhibiting conditions. If selected during the reset procedure, the controller responds with a status of Busy/Status Modifier during its reset procedure and automatically begins polling when the reset is complete. An asynchronous Control Unit End is generated at this time.

A 5085, as a result of seeing the System Reset, effects a Selective Reset to each 5085 address.

Selective Reset

The Selective Reset resets the device that is in operation at the time Selective Reset is detected. The reset clears attention, pending status, and sense, if any. In addition, Selective Reset clears the link between the controller and the processor and ensures that the ANK is enabled (if not the RS232C Attachment Feature). Other devices on the same controller are not affected. If selected during the reset procedure, the controller responds with a status of Busy/Status Modifier and generates an asynchronous Control Unit End, if required, at the completion of reset processing.

3250-Compatible Graphic Orders

The following table is an alphabetical list, in 3250 mnemonic order, of all 3250 graphic orders. The corresponding 5080 Graphics System mnemonics are given for those orders incorporated into the 5080 order set. However, only those orders that have *not* been incorporated into the 5080 order set are described in this section.

3250 Mnemonic	5080 Mnemonic	3250 Name	Hex Code	Class	Page
GDPD		Disable Pick Detects (see Note)	85	1	A-23
GDRD		Defer Response to Detects	83	1	A-23
GECM		Enter Character Mode	40–4F	1	A-13
GENSD		Enable No-Switch Detect	86	1	A-24
GEOS	GEOP	End Order Sequence	81	1	5-12
GEPI2		Enter Graphic Mode Incremental Point	04	1	A-19
GEPM	GDMA2	Enter Graphic Mode Absolute Point	00	1	5-18
GESD		Enable Switch Detect	84	1	A-23
GEVI2		Enter Graphic Mode Incremental Vector	05	1	A-19
GEVM	GDLA2	Enter Graphic Mode Absolute Vector	02	1	5-19
GIO	GBGIOP	Input/Output Control	89	3	6-6
GLAR		Load Immediate Attribute Register	D1	6	A-20
GMVA	GMVA	Move (Immediate) Address	EB	5	5-75
GMVD	GMVW	Move (Immediate) Data	EC	5	5-76
GNOP2	GNOP2	NO-OP 2-byte	80	1	5-79
GNOP4	GNOP4	NO-OP 4-byte	C0	2	5-79
GPDI		Permit Detect Interrupt	87	1	A-24
GSAR		Store Attribute Register	D2	4	A-21
GSBL		Set Intensity	91	1	A-24
GSPOS	GSDEVI	Store X-Y Position	E8	3	5-45
GSRT		Start Regeneration Timer	82	1	A-12
GSXY		Store X-Y Position Registers	EA	4	A-22
GSYMB		Draw Symbol	88	2	A-17
GTDD	GBDD	Transfer on Deferred Detect	FC	4	5-51
GTM	GTM	Test under Mask	ED	5	5-74
GTND	GBND	Transfer on No-Detect	FD	4	5-51
GTOC	GBC	Transfer on Condition	70–7F	4	5-53
GTRCT	GBCNT	Transfer on Count	F0	5	5-54
GTRU	GB	Transfer Unconditional	FF	4	5-55
GTSO	GBSO	Transfer on Switch Open	F5	4	5-50

Note: True 3250 name is Display Pen Detects.

3250-Compatible Start Regeneration Timer Order

GSRT—Start Regeneration Timer

The GSRT order emulates the 3250 GSRT order. It forces the following default 5085 processing modes:

- Frame buffer switching; that is, the frame buffers are switched during the execution of a GSRT order, causing the next execution of the graphic order program to create the image in the frame buffer that is not being refreshed.
- Clear mode; that is, the new frame buffer is cleared (erased) before execution of the graphic order program is continued. In this case all bit planes are cleared.

Name	Bit(s)	Value
Set Mode	0–7	X'2A'
Mode Control	8–15	X'82'

Either the one-word GSRT order or a Begin Order Processing (GBGOP) order should begin each graphic order program execution. (However, graphic order programs that begin without a GSRT or a GBGOP order will run.)

In addition, graphic order programs that do not have a subsequent GSRT or GBGOP order in their execution cycle run continuously only for 30 seconds until a timeout occurs, terminating the program execution (see "Display Program Termination" on page 3-10).

The GSRT order resets all attribute and stack registers to their default values (see "Attribute Control Orders" on page 5-31) and deactivates the pick detect window and the symbol generated by the Draw Symbol (GSYMB) order.

See "Interaction with Pick Indicators" on page 3-24 for a decision table representation of the PI, PBI, SDI, and TSI indicator-setting rules.

These settings of the attribute registers, PBI and SDI are compatible with the 3250 (that is, line type is solid, intensity is normal level, blink is off, and so on).

Note: Normal or relative character mode is not affected by the GSRT order; the mode remains unchanged.

The GSRT order sets the condition code (cc), which may be tested by the Transfer on Condition (GTOC) order, to reflect the current pick indicator (PI) and TSI settings:

cc = 00 PI off and TSI off (open)
cc = 01 PI off and TSI on (closed)
cc = 10 Not set
cc = 11 Not set

Notes:

1. The PI is always off after the GSRT order is executed, and the TSI is set/reset to match the stylus switch.

2. The actions requested by the GSRT order will not take effect if there is an outstanding Set Buffer Address Register and Start command when the GSRT order begins execution, or a Write Structured command containing a Start Buffer Program structured field or a Set Buffer Address Register and Start command is received by the time the graphic order program following the GSRT order prepares to begin running. If a Set Buffer Address Register and Start command or a Start Buffer Program structured field is received before execution restarts, the GSRT order is ignored. No reset is performed.

3. The default CSID for the GSRT order is X'00'.

3250-Compatible Enter Character Mode Order

If the CSID is zero or the Enter Character Mode (GECM) order is part of a graphic order program that started with a GSRT order and the CSID was not changed before encountering GECM, the GECM orders are interpreted as described below. IBM 3250 character sets are hardware-generated (fixed characters).

GECM—Enter Character Mode

Name	Bit(s)	Value
Set Mode	0–7	X'2A'
Mode Control	8–11	X'4'
	12	B'r'
	13	B'p'
	14–15	B'ss'

r Rotation

r =	0	The characters are displayed upright, the displayed field is horizontal, and characters are written left to right.
r =	1	The character string is drawn vertically, with characters rotated 90 degrees counterclockwise.

p Protection

p =	0	The data field following the order is unprotected; it can be updated from the ANK.
p =	1	The field is protected against manual changes from the ANK.

ss Character size

ss =	00	Basic
ss =	01	Large
ss =	10	Small
ss =	11	Medium

Mnemonics, names and codes for GECM orders are given in the following table.

Mnemonic	Name	Value
GECF(B)	Enter Character Mode Basic	X'2A40'
GECF(L)	Enter Character Mode Large	X'2A41'
GECF(S)	Enter Character Mode Small	X'2A42'
GECF(M)	Enter Character Mode Medium	X'2A43'
GECP(B)	Enter Character Mode Basic Protected	X'2A44'
GECP(L)	Enter Character Mode Large Protected	X'2A45'
GECP(S)	Enter Character Mode Small Protected	X'2A46'
GECP(M)	Enter Character Mode Medium Protected	X'2A47'
GECF(BR)	Enter Character Mode Basic Rotated	X'2A48'
GECF(LR)	Enter Character Mode Large Rotated	X'2A49'
GECF(SR)	Enter Character Mode Small Rotated	X'2A4A'
GECF(MR)	Enter Character Mode Medium Rotated	X'2A4B'
GECP(BR)	Enter Character Mode Basic Protected Rotated	X'2A4C'
GECP(LR)	Enter Character Mode Large Protected Rotated	X'2A4D'
GECP(SR)	Enter Character Mode Small Protected Rotated	X'2A4E'
GECP(MR)	Enter Character Mode Medium Protected Rotated	X'2A4F'

A GECM order is followed by a data list of zero or more words of the form X'aabb' where: aa are the first EBCDIC character bits and bb are the second EBCDIC character bits.

The null character (X'00') can be used to pad lists containing an odd number of characters to an integral number of words. The null character is not displayed nor does it cause movement of the current position.

The set mode (SM) byte (X'28', X'2A', X'2C', and X'2E') can also be used in the right (low-order) byte of the last data list byte and terminates the character string. This SM code byte is not displayed, does not cause movement of the current position, and cannot be replaced from the keyboard.

The color/grayshade (intensity), blink, frame buffer mask, and pick attributes held in the attribute registers are applied to the display of these characters. A character mode order determines which of four sizes applies to the data list, the rotation to be applied to the characters and lines of characters displayed from the data list (normal or 90 degrees counterclockwise), and whether these characters are to be protected against overwriting by manual changes from the alphanumeric keyboard.

Undefined codes are reserved. If they are used in the data list, an undefined code in the range from X'40' to X'FF' is displayed as a space and all other undefined codes are treated as null characters. However, the values X'28', X'2A', X'2C', and X'2E' may not appear within the data list, as they are recognized as terminating the data list.

Each displayable character (not null or new-line) occupies one character position on the screen and is drawn within a rectangle centered approximately on the current position; then, the current position is updated to locate the center point of the next available character position. (See Figure A-1.)

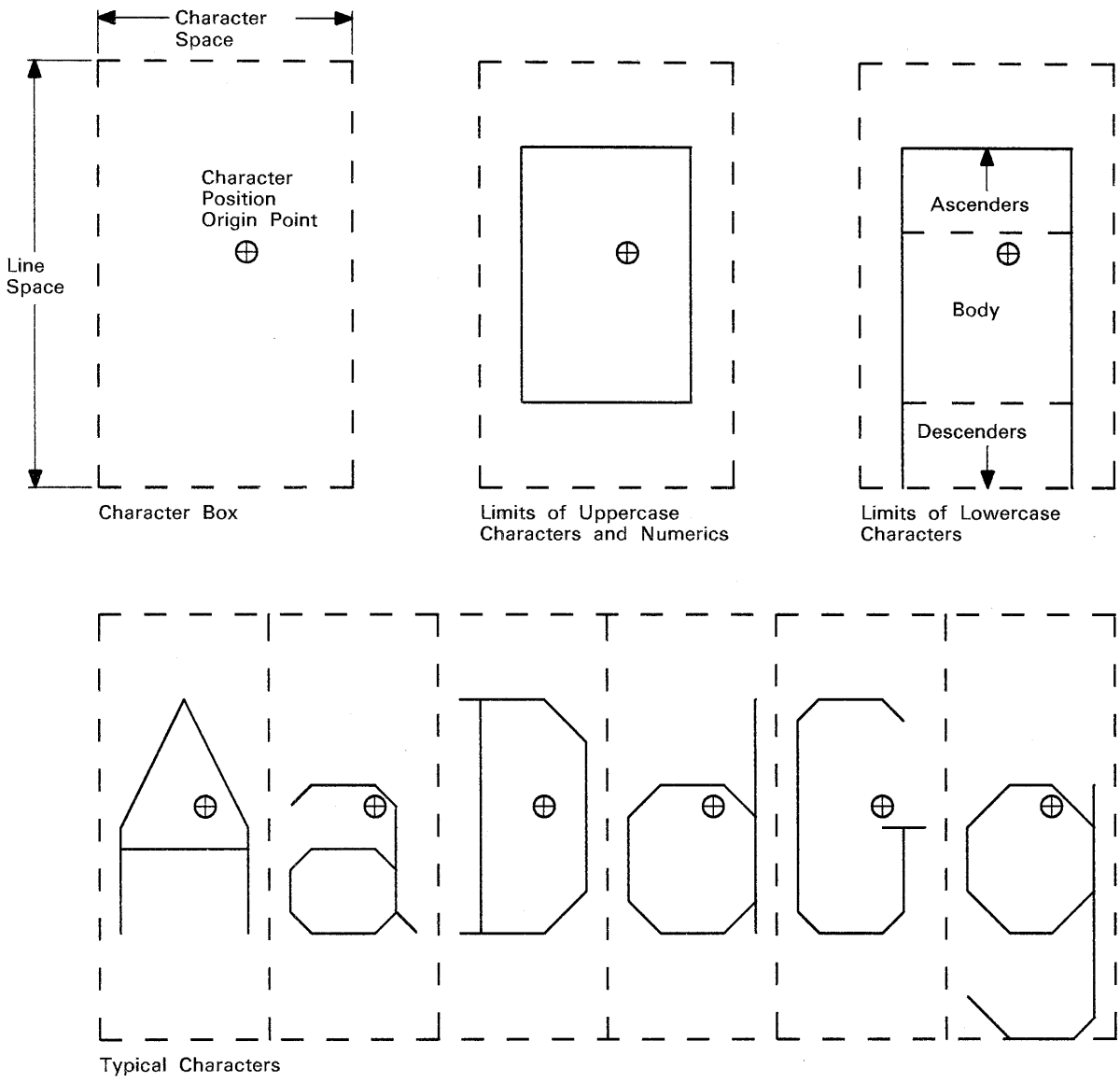


Figure A-1. Character Box and Origin (CSID = X'00')

The rotation and size of the rectangle within which a character is drawn and, hence, the center-to-center spacing between adjacent characters and lines are determined by the specific GECM order preceding the data list. The center-to-center distances in raster units (real pixels) are:

Small:	10 between characters, 15 between lines
Basic:	14 between characters, 20 between lines
Medium:	18 between characters, 25 between lines
Large:	21 between characters, 30 between lines

New-line and null characters are not displayed; they can appear anywhere in the data list. Because the null character occupies a data byte, but has no effect on character positioning, it can be used for reserving, initializing, or filling out a string of character bytes to satisfy a boundary requirement. See "ANK Input and the Cursor" on page 3-12 for information related to the handling of the overstrike key (Backspace [reqd.]) and additional information on new-line processing.

If the center point of a character lies within the viewable area, that character will optionally be fully displayed, even though parts of it extend outside the 4096 x 4096 virtual image area.

In the execution of GECM orders there are two types of character modes. Each is determined by the preceding graphic mode. If the most recently executed graphic order was Enter Graphic Mode Absolute Point (GEPM) or Enter Graphic Mode Absolute Vector (GEVM), any GECM order sets the *normal* character mode. However, if the most recently executed graphic order was Enter Graphic Mode Incremental Point (GEPI2), Enter Graphic Mode Incremental Vector (GEVI2), Draw Marker Relative 2D 12 Bits (GDMR2), Draw Marker Relative 3D 12 Bits (GDMR3), Draw Line Relative 2D 12 Bits (GDLR2) or Draw Line Relative 3D 12 Bits (GDLR3), then any GECM order sets the *relative* character mode. This action is independent of any intervening orders (except GSRT or GBGOP), and is independent of stopping or starting refresh except by device or system reset; the Set Buffer Address Register and Start command should not force normal character mode in a subsequent GECM order.

Depending upon the character mode, the following characteristics apply to the graphics system character generator during the processing of a nonrotated character field:

- Normal Character Mode
  - A new line is forced if incrementing X to the next position, and results in X being greater than 4095 (that is, the center of the next character is outside of the image area).
  - A new-line character in the data list resets X to 0 and decrements Y by a number of units in a line space.
  - A new line at the top of the image area is forced if decrementing Y for the new line makes Y negative (that is, the center of the first character of the new line is below the image area).
- Relative Character Mode
  - A new line is never forced; if advancing to the next character position moves the current draw position out of the image area, the character is blanked.

- A new-line character in the data list forces new coordinates, depending upon the current value of Y:

  — If Y is less than 4096, X is reset to 0 and Y is decremented by the number of the units in a line space. (If decrementing Y makes Y negative, Y is set to 4095 to force a new line at the top of the image area.)

  — If Y is in range 4096 to 8191, and decrementing Y makes it less than 4096, Y is set to 8191 and X is reset to zero.

In either mode, the next available character position is on the same line and to the right of (or above, if rotated) the preceding character if this point is within the viewable area (does not cause X or Y to overflow the range (0, 4095)).

The new-line character causes Y to be decremented (or X to be incremented if rotated) by one line space determined by character size, and X (Y if rotated) to be reset to zero. In either mode, if this action causes Y to become negative (or X to be greater than 4095 if rotated), a new line is forced at the top of the image area by setting Y (X if rotated) to 4095.

Note: When a graphic order program starts with a GSRT order and the '3250 Image' function is requested, characters produced by the character generator may overhang this area, that is, the center of the character may be positioned anywhere in the virtual image area. However, if a character is positioned such that its center lies close to, or on, the boundary of the virtual image area, the character itself may have some portion outside this area. The character will still be displayed correctly in these circumstances, although markers and lines will not be displayed outside the virtual image area.

3250-Compatible Draw Symbol Order

GSYMB—Draw Symbol

The GSYMB order defines a pick window of a specified size and displays a tracking symbol on the screen.

The GSYMB order contains the following values:

Name	Bit(s)	Value
Set Mode	0–7	X'2A'
Mode Control	8–15	X'88'
Mode Control	0	B'd'
Extension	1	B'0'
Word: Flags	2	B'n'
	3–7	B'00000'
	8-15	B'b...b'

d If d=0, no pick detection window is used and the b bits are ignored; the n bit is also ignored and assumed to be 1.
If d = 1, a square pick detection window of the size specified by the b field is established centered around the current position.

n If n = 0, a predefined tracking symbol (a cross) is displayed at the current position.
If n = 1, the predefined tracking symbol is not displayed at the current position.

b Size of pick detection window from 4 x 4 pixels to a maximum of 255 x 255 pixels in the virtual image space coordinates on a side.

When the GSYMB order is encountered in display storage, the system pick device mode (light-pen emulation mode) is negated, if active. The system pick device mode is reinstated upon execution of the next Set Buffer Address Register and Start command or Start Buffer Program structured field.

Notes:

1. If d = 1, a subsequent draw that causes a line, marker, point, pixel array, area fill, circle, or character to intersect the defined detect symbol area also causes a pick detect and is interpreted under the same control modes as any other pick detect.

2. The pick detect window detect symbol remains active until either reset by a Start Regeneration Timer (GSRT) or Begin Order Processing (GBGOP) order or until a new symbol is provided by a subsequent GSYMB order.

3. The tip switch indicator may have been set by a previous Store Device Input (GSDEVI) order according to the referenced unit, that is, the pick device or tablet switch.

3250-Compatible Incremental Graphic Mode Orders

Two orders, Enter Graphic Mode Incremental Point (GEPI2) and Enter Graphic Mode Incremental Vector (GEVI2), make it possible to display points or vectors, respectively, by specifying incremental displacements from the current position.

To help maintain image compatibility between the 5080 and the 3250, these two orders operate in 1024 addressable space, as there are no spare bits in the incremental graphic orders. Therefore, an effective maximum displacement of +252 or -256 (equivalent to +63 to -64 on a 3250) can be reached by increments of 4 on the displayed picture for X and Y (relative to 4096 addressable space).

Each displacement value can be positive or negative. When negative, the data is presented in twos complement form. The position defined by a pair of incremental coordinates is determined by adding the X and Y increments to the X and Y values, respectively, of the current position.

These one-word orders are each followed by a data list containing an integral number of words. The data list comprises zero or more pairs of signed coordinate increments.

GEPI2—Enter Graphic Mode Incremental Point

Name	Bit(s)	Value
Set Mode Mode Control	0–7 8–15	X'2A' X'04'
X-Delta/Y-Delta	0 1–6 7 8 9–14 15	B's' B'xxxxxx' B'w' B's' B'yyyyyy' B'b'

s Sign bit of X increment

x...x X increment (delta)

w Must be set to 1 by user to avoid possible interpretation of the data field as an order.

s Sign bit of Y increment

y...y Y increment (delta)

b Blanking bit

If b = 1, the vector or point is blanked, causing movement (of the current position) without display.

GEVI2—Enter Graphic Mode Incremental Vector

Name	Bit(s)	Value
Set Mode Mode Control	0–7 8–15	X'2A' X'05'
X-Delta/Y-Delta	0 1–6 7 8 9–14 15	B's' B'xxxxxx' B'w' B's' B'yyyyyy' B'b'

s Sign bit of X increment

x...x X increment (delta)

w Must be set to 1 by the user to avoid possible interpretation of the data field as an order.

s Sign bit of Y increment

y...y Y increment (delta)

b Blanking bit

If b = 1, the vector or point is blanked, causing movement (of the current position) without display.

3250-Compatible Attribute Control Orders

GLAR—Load Immediate Attribute Register

The two-word GLAR order contains, as its second word, data comprising validity bits and attribute values.

Name	Bit(s)	Value
Set Mode	0–7	X'28' or X'2A'
Mode Control	8–15	X'D1'
Data Word	0–3	B'0000'
	4	B'v'
	5–7	B'aaa'
	8	B'v'
	9–11	B'ddd'
	12	B'v'
	13–15	B'ccc'

v Value (blink mode change)

v = 0 No change
v = 1 New blink value is in the 3 bits immediately following.

aaa Blink mode; these bits are placed in the 3 low-order bits of the blink pattern ID register (attribute register 1).

aaa = 000 Normal
aaa = 001 Blink

v Value (line type change)

v = 0 No change
v = 1 New line type value is in the 3 bits immediately following.

ddd Line type; these bits are placed in the 3 low-order bits of the line type register (attribute register 2).

ddd = 000 Solid
ddd = 001 Dotted
ddd = 010 Dashed
ddd = 011 Dot-dashed
ddd = 100-111 Reserved

v Value (intensity change)

v = 0 No change
v = 1 New intensity value is in the 3 bits immediately following.

ccc Intensity (color/grayshade); these bits are stored in the 3 high-order bits of the color/grayshade register (attribute register 3); all values are allowed.

ccc = 000 Off
ccc = 101 Normal
ccc = 111 Brightest in a monochromatic display

Execution of the GLAR order sets the subfields of the logical attribute register for the workstation as defined by the data in the second word of the order. If a validity bit is 1, the corresponding subfield in the attribute register is set to the value in the 3 bits following the validity bit. If the validity bit is zero, the corresponding attribute register field is not changed.

Note on Color/Grayshade: The actual color/grayshade appearing on the monitor depends on the Color Table and frame buffer mask settings (see "Attribute Registers" on page 3-32). If the last load attribute order was a GLAR order and the color/grayshade value is 1–4, a pick will not occur on subsequent primitives until changed.

Note: This order is provided only for 3250 compatibility; it does not load the full values of the processor attributes. The remaining bits of the line type and color/grayshade attributes are set to zero by this order (if the validity bit = 1).

GSAR—Store Attribute Register

The two-word GSAR order stores the current settings of the 3250- compatible attributes at the address specified in the second word of the order.

Name	Bit(s)	Value
Set Mode	0–7	X'2A' or X'2E'
Mode Control	8–15	X'D2'
Address Word	0–15	X'aaaa'

Address Word

x'aaaa' The attribute values are stored in the word pointed to by the address word. The data is always stored on a word boundary; bit 15 of the address word is ignored and assumed to be zero.

The values stored at the specified addresses are determined by the current values of the attribute registers. The values are stored in the following form:

0000 1aaa	1ddd 1ccc

aaa Blink mode; these bits are stored from the 3 low-order bits of the blink pattern ID register (attribute register 1).

aaa = 000 Normal
aaa = 001 Blink

ddd Line type; these bits are stored from the 3 low-order bits of the line type register (attribute register 2).

ddd = 000 Solid
ddd = 001 Dotted
ddd = 010 Dashed
ddd = 011 Dot-dashed
ddd = 100-111 Reserved

ccc Intensity (color/grayshade); these bits are stored from the 3 high-order bits of the color/grayshade register (attribute) register 3).

ccc = 000 Off
ccc = 101 Normal
ccc = 111 Brightest in a monochromatic display

3250-Compatible Current-Position Order

GSXY—Store X-Y Position Registers

The two-word GSXY order causes 12 bits of X-position data and 12 bits of Y-position data from the current position registers of the workstation to be stored into the two-word target location specified by the address field. The low-order bit of this address is ignored.

Name	Bit(s)	Value
Set Mode Mode Control	0–7 8–15	X'2A' or X'2E' X'EA'
Address Word	0–15	X'aaaa'

Target Location

The two-word target location (field) specified by the address word is formatted as follows:

Field	Bit(s)	Value
X-Position Data	0–15	X'0xxx'
Y-Position Data	0–15	X'0yyy'

3250-Compatible Pick Control Orders

For a description and details of the setting and resetting of the pick detection indicators (PI, TSI, PBI, and SDI) referred to in the following pick control orders, see "Pick Detection Modes and Indicators" on page 3-20.

GDRD—Defer Response to Detects

The one-word GDRD order sets the pick interrupt mode to the deferred state. The pick detect mode is not altered by this order, nor are the PI, TSI, PBI, or SDI indicators. A pick detect in the deferred state sets on the pick indicator. The detect is remembered; it does not cause an interrupt at the host CPU.

Name	Bit(s)	Value
Set Mode	0–7	X'2A'
Mode Control	8–15	X'83'

GESD—Enable Switch Detect

The one-word GESD order sets the pick detect mode to the switch enabled state. The pick interrupt mode is not altered, nor are any of the four indicators (PI, TSI, PBI, and SDI). While in the switch enabled state, only one detect can occur per stylus switch closure. The change of state of the stylus switch is sensed at GSRT or GBGOP order execution time and at Set Buffer Address Register and Start command time. The TSI may also be changed at GSDEVI order execution. A pick detect occurs only if displayed data of sufficient intensity is intersected by the pick window when a tablet switch (stylus) or a cursor function key is closed and the PBI is on and the SDI is off.

Name	Bit(s)	Value
Set Mode	0–7	X'2A'
Mode Control	8–15	X'84'

Note: The intensity value specified in the GLAR order is used as an index into the Color Table. The intensity value must be 5 or greater for a pick to occur. To provide 3250 compatibility, the Color Table must be loaded appropriately. The system is initialized with a default Color Table that provides (approximate) 3250-compatible intensity (grayshade). (See "Color Table (CT)" on page 3-8.) If the Color Table is changed by subsequent Load Color Table (GLCT) orders, when running new display programs written for the 5085 Graphics Processor, the Color Table must be restored to the values shown in the default Color Table using a GLCT order if 3250-compatible programs are to be resumed.

GDPD—Disable Pick Detects

The one-word GDPD order sets the pick detect mode to the disabled state. The pick interrupt mode is not altered, nor are any of the four indicators (PI, TSI, PBI, and SDI). Pick detects cannot occur in the processor while it is in the disabled state.

Name	Bit(s)	Value
Set Mode	0–7	X'2A'
Mode Control	8–15	X'85'

GENSD—Enable No-Switch Detect

The one-word GENSD order sets the pick detect mode to the nonswitch detect state. The pick interrupt mode is not altered, nor are any of the four indicators (PI, TSI, PBI, and SDI). In the nonswitch enabled mode a pick detect can occur each time display information is passed to the pick module (that is, on one or more vectors, points or characters while in this mode). Pick detection is independent of the PBI and SDI indicators. In the deferred state, multiple detects can occur in one regeneration cycle.

Name	Bit(s)	Value
Set Mode	0–7	X'2A'
Mode Control	8–15	X'86'

GPDI—Permit Detect Interrupt

The one-word GPDI order sets the pick detect interrupt mode to the immediate state. If the PI is on, a pick detect interrupt is generated and the address of the display list buffer location containing the GPDI order is saved as part of the sense data.

The pick detect mode is not altered, nor are any of the four indicators (PI, TSI, PBI, and SDI). This order provides the means to generate pick detect interrupts on the basis of detecting an entity or subpicture rather than a specific line, point, or character, by deferring detects on any component of the subpicture until this order is executed.

Name	Bit(s)	Value
Set Mode	0–7	X'2A'
Mode Control	8–15	X'87'

3250 RPQ-Compatible Orders

The following orders are not assigned as standard processor orders. No standard support will be provided for them at either the box level or host level.

GSBL—Set Intensity

The GSBL order consists of the four suborders listed in the following table:

Mnemonic	Name	Value
GSBL(B)	Set Blank Intensity Level	X'2A90'
GSBL(L)	Set Low Intensity Level	X'2A91'
GSBL(M)	Set Medium Intensity Level	X'2A92'
GSBL(H)	Set High Intensity Level	X'2A93'

These one-word orders allow the intensity (color/grayshade) level to be selected by setting the intensity (color/grayshade) attribute register to X'00', X'03', X'05', and X'07' from GSBL(B), GSBL(L), GSBL(M), GSBL(H), respectively. The last order issued, that is, one of the four GSBL orders, GLAR, or GLATR, together with the Color Table determines the intensity (color/grayshade) to be displayed.

3250 Functional Compatibility

A basic objective of the 5080 Graphics System in 3250 mode is to be plug-compatible with the 3255/3251. This implies end-user and program compatibility with the 3255/3251. Existing applications for the 3250 should run without change. To use the new functions on the 5080 Graphics System, modifications to existing application software will be required.

However, there are a number of differences between the 5080 and the 3250, most of which are inherent in the use of a raster display to emulate a directed beam display.

- *Character Overhang:* Scaling of all positioning addresses is available if a user wants to provide for character overhang. A setup option (see *IBM 5080 Graphics System: Operation and Problem Determination,* GA23-0133) is provided that permits the image to be displayed at 31/32 of its size. When the '3250 Image' mode flag is on during GSRT order time, the coordinates of all positional orders are transformed from their Xc, Yc form to their Xc,Yc form as follows:

  $$Xc = (31/32)Xc + Xo$$

  $$Yc = (31/32)Yx + Yo$$

  where Xo and Yo are offsets of 64 counts in the 4096 space (16 in the 1024 x 1024 screen).

  Present positions are stored in the display list buffer prior to their scaling in order to respond to the requirements of orders such as GSXY and GSDEVI (see "GSXY—Store X-Y Position Registers" on page A-22 and "GSDEVI—Store Device Input" on page 5-45 for details). New processor positional orders are not scaled.

- *Entity Verification:* Certain 3255 buffer programs are written to reexecute a portion of a vector list upon occurrence of a light-pen detect. This causes the rewritten portion of the displayed image to be intensified. However, since the processor uses a raster-scan and frame buffer technology, this does not work.

  The 5080 Graphics System provides new orders (segment control) and capabilities that allow a section of the display list buffer to be reexecuted with a different display attribute (color or grayshade) when a detect is recognized for any part of that display list buffer section.

- *Never-Ending Display List Buffer:* In the graphics processor with its double frame buffer, a display list buffer terminator (for example, a GSRT order) causes the new image being rastered to be displayed on the 5081 Display. If a display list buffer terminator does not occur in the display list buffer, the new image is not displayed. Users are instructed to include a display terminator in their display list buffers. A timer is provided in the processor to protect against such possible loops.

- *End Order Sequence (GEOS) Order:* The address returned by the GEOS order is always even.

- *Character Sizes and Shapes:* The actual length and shape of individual strokes are chosen for maximum legibility. They are different in shape because they are defined as dot matrixes. The character cells are the same as for the 3250.

- *Display List Buffer Size:* The 3250 commands and orders address the display list buffer as pages of 32K bytes. That is, the top bit of the 16-bit address fields is ignored. The processor does honor this bit, and provides for continuous addressing through 64K pages. In addition, orders are provided for the management of multiple (64K) pages.

- *Buffer Wrap:* The 3250 wraps addresses around on the 32K boundary such that the next address examined after (32K-2) is 0, not 32K. The processor wraps at the page boundary regardless of where it is established (for example, 32K, 48K, 52K). The maximum page size supported by the processor is 64K.

- *Keyboard Layouts and Labels:* The location and naming of a number of controls are changed from what they are on the 3250. This is done in order to bring about convergence with the location and names being used on the 3270 family of products. This provides for consistency of location and name for similar functions in the two modes of operation, and with other 3270 products that may also be used by the end user.

- *End of Application:* When a 3250 finishes an application, the screen goes blank. On the processor the last image generated continues to be displayed.

- *Last Position of an Unprotected Field:* The 3250 replaces characters when additional characters are entered. In the 5085 processor, when the character cursor is in the last position of an unprotected field, any further character codes entered from the ANK do not replace the last position.

- *Overlapping Entities or Vectors:* In the 3250 if vectors overlap, there is an intensification of the overlap area. In the processor when vectors overlap, there is no intensification because it is a raster device.

- *Vectors with a Single Endpoint Off the Screen:* In the 3250, if a vector has an endpoint that is off the screen and has not been clipped, the visible portion of the vector is not displayed. The processor displays the visible portion.

- *Wrapping of Vectors in Coordinate Space:* In the 3250, if a vector or character wraps with the 4K space, its wrapped portion is visible. In the processor mode, this wrap occurs on the 32K boundary.

  In the 3250, current position is maintained in a 4K space. In the processor, the current draw position is maintained in ± 32K space.

- *Enter Graphic Mode Absolute Point (GEPM) Order:* In the 3250, a point is a single point on the screen. In the processor, a point is a 3 x 3 pixel array in order that points drawn on top of lines may be visible.

- *Draw Symbol (GSYMB) Order:* In the 5085, a predefined tracking symbol (a cross) is drawn rather than a square symbol if n=0.

- Pressing the Reset key on the 5085 ANK also clears the manual input register, thus enabling the ANK and LPFK if they were previously locked awaiting a response (Read Manual Input command) to an ANK ENTER or Cancel key or to an LPFK key.

3258 Compatibility

The 5088 Graphics Channel Controller is a plug-compatible replacement for the 3258 Control Unit.

Mixing 5080 Orders with 3250-Compatible Orders, and Scaling

IBM 3250-compatible orders can be intermixed in the same display program with 5080 orders, as desired. However, the 3250 graphics mode orders (GEPM/GDMA2, GEVI2, GEVM/GDLA2 and GEPI2) should not be intermixed with the unique 5080 drawing orders if '3250 Image' mode is specified at setup time and the GSRT order is being used to regenerate the image. The 5080 Graphics System executes these orders when issued, regardless of the mix, but, if '3250 Image' mode is specified, the image may be distorted.

If a GBGOP order is used, 3250 display mode is temporarily suspended even if it is selected at setup time. Accordingly, in this case the 3250 graphics mode orders can be intermixed with 5080 drawing mode orders with consistent results.

Appendix B. TCF Number Representation and Mathematics of the Transformation Process

Number Representation

The numbers processed by the Transformation and Clipping Feature (TCF) are in various formats, depending on what the numbers represent and on the stage within the transformation process.

Display Storage Coordinates

The formats of the numbers representing the absolute, relative and incremental coordinates in the display storage are described in "TCF Transformation and Viewing Orders" on page 5-88 and "Specification of Coordinate Values" on page 3-29.

Pretransformation Space Coordinates

The display storage coordinates define the absolute coordinates of points in pretransformation space world coordinates, which are represented by 16-bit integers (the high-order bit is the sign bit). The format for the X, Y and Z coordinates is:

s b b b b b b b b b b b b b b b

where s is the sign bit. Negative numbers are in twos complement notation. This is called a "16-bit integer" format.

Transformation Matrix Elements

The elements of the bottom row of the matrix (m31, m32 in two-dimensional mode, or m41, m42, m43 in three-dimensional mode) are in 16-bit integer format. All other elements have the following format:

s.b b b b b b b b b b b b b b b

where s is the sign bit, the point, or period (.), is the position of the binary point, and the b bits have fractional weights, being coefficients of negative powers of 2. This format is called a "16-bit fractional" format.

The scale factor, which is a shift factor for the first two or three rows of the transformation matrix (depending on 2D or 3D mode), is a 4-bit positive binary number of the following format:

0 0 0 0 0 0 0 0 0 0 0 0 b b b b

The scale factor value is a power-of-2 exponent with an allowable range of values from 0 to 12. The scale factor default value is B'0000'. The default value is assumed if the scale factor is not specified.

If matrix concatenation causes the value of any matrix element in the first two rows in two-dimensional (2D) mode or the first three rows in three-dimensional (3D) mode to exceed the values expressible in fractional number format (from -1 to +1), an attempt is made to bring the element value within the permissible range by adjusting the values of the scale factor and other elements. If this

attempt fails to correct the problem, the process is halted and a "matrix element overflow" error is signaled to the host. If the values of the elements in the last row exceed the range of -32K to +32K, the process is also halted and a matrix element overflow error is signaled to the host.

Transformed Coordinates

Transforming a coordinate consists of multiplying two pairs or three pairs of numbers, depending on the mode (2D or 3D). Each pair consists of a 16-bit integer format number (coordinate value) and a 16-bit fractional format number (an element from the first two or three rows of the transformation matrix) shifted left SF times, where SF is the value of the scale factor. The product is a 32-bit number consisting of a sign bit, an n bit integer and a 31-n bit fractional number. The value of n depends on how many times the product is shifted to the left. The fractional part is used to round off the number up or down to a 16-bit integer; then the fractional part is discarded. This number is added to an element in the last row of the transformation matrix to obtain the transformed coordinate value.

Mathematics of the Transformation Process

The operation of coordinate transformation can be defined using the notation of matrix algebra:

$$[V] = [T]\ [E]$$

where V and T are the resultant and pretransformation coordinates, respectively, in row-matrix form, and E is the transformation matrix. The transformation matrix always has four rows and three columns. When in two-dimensional (2D) mode, the third row and column are ignored and assumed to be zero.

The first two elements of the first two rows (2D) or all elements of the first three rows (3D) of the transformation matrix are used for rotation and scaling. Because these elements are fractional signed binary numbers between -1 and +1, they can be used only for scale-down and rotation transformation.

If scale-up transformation is required, a 4-bit positive binary number called the "scale factor" is used in addition to the transformation matrix elements.

During the transformation process, each matrix element for rotation and scaling is multiplied by 2-to-the power-of-the-scale-factor (shifted left), and then used to transform a coordinate value.

Use of the scale factor allows an image to be magnified up to 4096 times. If only reduction and/or rotation of an image is required, the scale factor is set to zero. The first two elements of row 4 (2D) or all the elements of row 4 (3D) are used to perform the translation of coordinates, that is, the addition of a constant to each coordinate.

Following are the mathematical equations of the transformation process:

In 2D mode	**In 3D mode**
A point [Xp Yp]	or a point [Xp Yp Zp]
Is transformed into a point	
[Xr Yr]	[Xr Yr Zr]
by the transformation matrix	
$\begin{bmatrix} M11 & M12 & . \\ M21 & M22 & . \\ . & . & . \\ M41 & M42 & . \end{bmatrix}$	$\begin{bmatrix} M11 & M12 & M13 \\ M21 & M22 & M23 \\ M31 & M32 & M33 \\ M41 & M42 & M43 \end{bmatrix}$
and a scale factor (SF)	and a scale factor (SF)

according to the following equations:

In 2D mode:

$$Xr = (Xp.M11 + Yp.M21 + M41/(2^{**}SF)).(2^{**}SF)$$

$$Yr = (Xp.M12 + Yp.M22 + M42/(2^{**}SF)).(2^{**}SF)$$

In 3D mode:

$$Xr = (Xp.M11 + Yp.M21 + Zp.M31 + M41/(2^{**}SF)).(2^{**}SF)$$

$$Yr = (Xp.M12 + Yp.M22 + Zp.M32 + M42/(2^{**}SF)).(2^{**}SF)$$

$$Zr = (Xp.M13 + Yp.M23 + Zp.M33 + M43/(2^{**}SF)).(2^{**}SF)$$

When matrix [E] having a scale factor of SF1 is concatenated with matrix [T] having a scale factor of SF2 to produce the matrix [U] having a scale factor of SF3, that is, [U] = [T] [E], then the elements of [U] are given by the following equations:

In 2D mode:

$$U11 = (T11.E11 + T12.E21)(2^{**}SF3)$$

$$U12 = (T11.E12 + T12.E22)(2^{**}SF3)$$

$$U21 = (T21.E11 + T22.E21)(2^{**}SF3)$$

$$U22 = (T21.E12 + T22.E22)(2^{**}SF3)$$

$$U41 = (T41.E11 + T42.E21)(2^{**}SF1) + E41$$

$$U42 = (T41.E12 + T42.E22)(2^{**}SF1) + E42$$

$$SF3 = SF1 + SF2$$

(The value of SF3 may be adjusted to prevent underflow or overflow of the matrix elements.)

In 3D mode:

U11 = (T11.E11 + T12.E21 + T13.E31)(2**SF3)

U12 = (T11.E12 + T12.E22 + T13.E32)(2**SF3)

U13 = (T11.E13 + T12.E23 + T13.E33)(2**SF3)

U21 = (T21.E11 + T22.E21 + T23.E31)(2**SF3)

U22 = (T21.E12 + T22.E22 + T23.E32)(2**SF3)

U23 = (T21.E13 + T22.E23 + T23.E33)(2**SF3)

U31 = (T31.E11 + T32.E21 + T33.E31)(2**SF3)

U32 = (T31.E12 + T32.E22 + T33.E32)(2**SF3)

U33 = (T31.E13 + T32.E23 + T33.E33)(2**SF3)

U41 = (T41.E11 + T42.E21 + T43.E31)(2**SF1) + E41

U42 = (T41.E12 + T42.E22 + T43.E32)(2**SF1) + E42

U43 = (T41.E13 + T42.E23 + T43.E33)(2**SF1) + E43

SF3 = SF1 + SF2

(The value of SF3 may be adjusted to prevent underflow or overflow of the matrix elements.)

Note: This is not the same as [E] [T].

The following matrix can be used to translate X and Y coordinates by the

Amounts Tx, Ty:

In 2D mode

$$\begin{bmatrix} 1 & 0 \\ 0 & 1 \\ Tx & Ty \end{bmatrix}$$

Amounts Tx, Ty, Tz:

In 3D mode

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ Tx & Ty & Tz \end{bmatrix}$$

The following matrix plus a scale factor (SF) can be used to scale X and Y coordinates by the

Amounts Sx, Sy:

In 2D mode

$$\begin{bmatrix} Sx & 0 \\ 0 & Sy \\ 0 & 0 \end{bmatrix}$$

Amounts Sx, Sy, Sz:

In 3D mode

$$\begin{bmatrix} Sx1 & 0 & 0 \\ 0 & Sy & 0 \\ 0 & 0 & Sz \\ 0 & 0 & 0 \end{bmatrix}$$

The following matrix can be used to rotate the position of a point about the Z axis through a counterclockwise angle whose sine is S and whose cosine is C:

$$\begin{bmatrix} +C & +S & 0 \\ -S & +C & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix}$$

Mapping from a window in world coordinate space to a viewport in virtual image space is done by, first, performing a scaling operation on the transformed, clipped and projected X and Y values, where scaling values are the ratios of the window and the viewport X and Y dimensions; and then translating the scaled coordinate values to the correct location in virtual image space based on the lower limit X and Y values of the viewport boundaries. If the mapping mode is off, the 16-bit world coordinate values are truncated to 12 bits by dropping the 4 high-order bits; coordinate values are then passed to the raster generator.

Note: To rotate about a point not at the origin, it is necessary first to translate the point of rotation to the origin, perform the rotation, and translate back to the original point. This can be achieved in a single matrix operation if the three transformation matrixes are first concatenated, using the Load Transformation Matrix (GLTM) order, before processing the points.

Appendix C. Summary of Status/Sense Byte Bit Assignments

Tables C-1 through C-3 summarize the assignment of status bits, sense bits, and combinations of status and sense bits while operating with a 5080 Graphics System.

Table C-1. Status Byte Bit Assignments

Bit	Name	Condition
0	Attention	Indicates (if not with Unit Check) a request from the alphanumeric keyboard (ENTER, Cancel or program function keys) or the lighted program function keyboard (LPFK) or the Begin Order Processing (GBGOP) order (simulated LPFK) and should be followed by a Read Manual Input command. If accompanied by a Unit Check (byte 0, bit 6), indicates either a pick detect, an End Order Processing (GEOP) order, or a Store Device Input (GSDEVI) order interrupt, when byte 1, bits 0–2 and 5–6 of the sense data contain an indication of the cause, or when a Program Error (byte 1, bit 7) occurs. The sense data contains an indication of the cause. If accompanied by Device End and Unit Exception, indicates a not-ready-to-ready state transition.
1	Status Modifier	Set with the Busy bit to indicate a Control Unit Busy condition exists, possibly with a device other than the one addressed.
2	Control Unit End	Indicates the termination of a previously indicated Control Unit Busy condition.
3	Busy	Set in response to all commands except the Test I/O instruction if outstanding status for the addressed device is pending. The outstanding status accompanies the Busy indication. Set alone in response to all commands and the Test I/O instruction if no outstanding status for the addressed device is pending, and a previously issued command is still in progress with the addressed device (that is, Device End outstanding). The Busy bit is also set with the Status Modifier bit to indicate a Control Unit Busy condition.

Bit	Name	Condition
4	Channel End	Set when the transfer of data and/or control information (if any) between the 5088 Graphics Channel Controller and the channel is complete.
5	Device End	Set when the 5085 Graphics Processor has completed operation on a command and is prepared to accept a new command. Set with Attention and Unit Exception to indicate a not-ready-to-ready transition.
6	Unit Check	Set when an error condition is detected by the controller or graphics processor in order to notify the host. (See "Sense Information" on page 4-29 and Table C-2 for further definition.) The host program should respond by issuing a Sense command for further definition of the condition. If command execution has started, Unit Check is set with Device End and, possibly, Channel End. Both Attention and Unit Check are set to indicate an unsolicited input from the graphics workstation. (For further details, see the Attention bit in this table.)
7	Unit Exception	Indicates a not-ready-to-ready state transition when accompanied by Attention and Device End.

Table C-2. Sense Byte Assignments

Byte	Bit	Name	Indication
0	0	Command Reject	Indicates: • A nonimplemented command or a required feature is not present, that is, a Write Structured, Select Write Memory Area, Write Memory Area, Select Read Memory Area, or Set Mode command issued to a 3255; or • An invalid command sequence, that is, certain commands issued to a 3255 while the buffer (display program) is running or a Read Memory Area command not preceded by a Select Read Memory Area command or a Write Memory Area command not preceded by a Select Write Memory Area command; or • Insufficient data on the following commands: Select Write Memory Area, Select Read Memory Area, Set Buffer Address Register and Start, Set Buffer Address Register and Stop, and Set Mode.
0	1	Intervention Required	Indicates the addressed device is not available.
0	2	Bus Out Check	Indicates a Bus Out parity error on a command or data byte.
0	3	Equipment Check	Not used.
0	4	Data Check	Indicates that the automatic retry has been unsuccessful on the controller-to-workstation link. Also, link timeout due to device Busy response.
0	5	Overrun	Data streaming overrun.
0	6	Display Program Running	Indicates display program execution is in process for the addressed processor. May not be valid following a Write Structured command unless there has been a subsequent Set Buffer Address Register and Start (or Stop) command, or the Sense command is either unsolicited or the result of an asynchronous Attention, Unit Check presentation.
0	7	Reserved	

Byte	Bit	Name	Indication
1	0	Pick Detect	Indicates pick interrupt or GSDEVI switch interrupt, if byte 1, bit 6 is also set. Also set if Segment Pick detected, byte 1, bit 5 (see Note).
1	1	End Order Processing	Set when an End Order Processing (GEOP) order is encountered.
1	2	Character Mode	Set when in character mode (used with pick detect) (see Note).
1	3	Reserved	
1	4	Hardware Error	Set when a hardware error is detected. Byte 5 indicates the cause of the error.
1	5	Segment Pick	Set when an interrupt is generated by a segment pick. Bit 0 (Pick Detect) is also set (see Note).
1	6	GSDEVI	Set when an interrupt is generated by a position entry device switch. Bit 0 (Pick Detect) is also set (see Note).
1	7	Program Error	Indicates a display program error or an error in the channel commands and structured fields. Byte 4, bits 2–5 and 7 indicate cause of the error in the display program. Byte 4, bits 0–1 and 6 indicate cause of the channel command and structured field errors.
2 3	0–7 0–7		When the Structured Field Error bit is set, these bytes contain the address of the structured field in error. The origin of the address is the beginning of the data transmitted with the Write Structured command. When the Stack Error bit is set (byte 4, bit 5) these bytes contain the stack current pointer. When the GSDEVI bit is set (byte 1, bit 6) these bytes contain the unit number (byte 3, bits 4–7). Bytes 8, 9, 6, and 7 contain the display storage address and the page number of the GSDEVI order. When the Pick Detect bit is set, these bytes contain the display storage address at which the pick occurred (see Note).

Byte	Bit	Name	Indication
			When the End Order Processing bit is set, bytes 2 and 3 contain the display storage address of the location immediately following the GEOP. When byte 4, bit 2, 3, 4, or 6 is set, these bytes contain the display storage address of the order or the coordinate specification at which the buffer program is stopped. At the completion of a Read Cursor command these bytes contain the display storage address following the cursor location, or the display storage address following the point at which the channel command word count becomes zero. At other times the contents of these bytes are unpredictable.
4	0	Structured Field Error	Set when an error occurs during the execution of any structured field.
4	1	Device Working Timeout	Set when the time for processing Select Write Memory Area, Select Read Memory Area, Write Memory Area, and Write Structured commands exceeds the device working timeout (about 3 seconds).
4	2	Invalid Page	Set when a Branch Page (GBPAGE), Branch after Push Link (GBAPL), or Move Data Block (GMVBLK) order or an input/output control block pointed to by a Begin I/O Processing (GBGIOP) order (RS232C Attachment Feature) indicates an invalid page. Bytes 2, 3, 6, and 7 contain the display storage address of the order.
4	3	Display Program Loop	Set when a loop is detected in a display program; that is, the program ran for more than 30 seconds without a Begin Order Processing (GBGOP) or Set Regeneration Timer (GSRT) order being executed. Bytes 2, 3, 6, and 7 contain the display storage address at which the program is stopped.

Byte	Bit	Name	Indication
4	4	TCF Matrix Element Overflow	Set when a scale factor for a matrix in the Transformation and Clipping Feature (TCF) exceeds a value of 12 (that is, during concatenation). Bytes 2, 3, 6, and 7 contain the display storage address of the Load Transformation Matrix (GLTM) order at which the error occurred and the program is stopped.
4	5	Stack Error	Set when a stack error occurs, and bytes 2, 3, 6, 7, 8, 9, 12, and 13 contain the stack pointers.
4	6	Invalid Memory Area Address	Set when a parameter of the Select Read Memory Area or Select Write Memory Area command indicates the subsequent read or write operation will cause a memory area overflow.
4	7	Reserved	
5	0	Memory Error	Set when a memory error is detected. This bit is valid if byte 1, bit 4 = 1.
5	1–6	Reserved	
5	7	Unit Specify	Set if the error is in the controller.
6 7	0–7 0–7	Page Number MSB Page Number LSB	Indicates page number corresponding to display storage address contained in sense bytes 2 and 3 or bytes 8 and 9.
8 9	0–7 0–7	Segment Name MSB Segment Name LSB	Most recent segment name associated with a Segment Pick interrupt. When the GSDEVI bit is set (byte 1, bit 6) these bytes contain the display storage address of the GSDEVI order. When the Stack Error bit is set (byte 4, bit 5), these bytes contain the stack initial pointer.
10	0–7	Reserved	
11	0–7	Reserved	
12 13	0–7 0–7	Stack Limit MSB Stack Limit LSB	When the Stack Error bit is set (byte 4, bit 5), these bytes contain the stack limit pointer.

Byte	Bit	Name	Indication
14 15 16 17 18 19	0–7 0–7 0–7 0–7 0–7 0–7	X-Y-Z – X MSB X-Y-Z – X LSB X-Y-Z – Y MSB X-Y-Z – Y LSB X-Y-Z – Z MSB X-Y-Z – Z LSB	Bytes 14–19 contain the X-Y-Z position registers (or current draw position) associated with the display and are stored each time the display program is stopped. They contain the X, Y, and Z values in signed 16-bit pretransformation world coordinate space. Negative values are presented in twos complement form. When in graphics mode, the returned coordinates represent the specific current draw position. When in character mode, the returned coordinates, if CSID = X'00' (in attribute register 18), represent the centerpoint of the character following the character upon which detection occurred. If CSID is not X'00' (in attribute register 18), the coordinate represents the lower left corner point of the character following the character upon which detection occurred. If the Sense command is issued and the pick interrupt was the result of a Store Device Input (GSDEVI) order causing byte 1, bit 6 of the sense data to be set, the value of the X-Y-Z coordinates returned represents the current tablet (stylus or cursor) position. The Z value is set to zero. If the Sense command is not sent in response to a pick detection interrupt, the value of the X-Y-Z position registers is returned, but the significance of the value depends upon the display program and the point at which it was stopped. If the Sense command is issued to an RS232C Attachment Feature port, the contents of bytes 14–19 will be zero. Refer to "X-Y-Z Position Registers" on page 3-6 for more details.
20–23		Reserved	

Note: The following bits of sense byte 1 may be set as a result of a pick interrupt:

- Pick Detect; byte 1, bit 0 (B0)
- Character Mode; byte 1, bit 2 (B1)
- GSDEVI; byte 1, bit 6 (B2)
- Segment Pick; byte 1, bit 5 (B3)

The information set in sense bytes 2, 3, 6, 7, 8, and 9 depends on the value of B0B1B2B3 as defined in the following table:

B0B1B2B3	Contents of Bytes 2 and 3	Contents of Bytes 6 and 7	Contents of Bytes 8 and 9
0XXX	N/A	N/A	N/A
1000	Display storage address of one of the following orders: Permit Detect Interrupt (GPDI), Draw Circle (GDIR).	Page number	N/A
	Display storage address of the first word of the selected coordinate group following the orders: Draw Line Absolute 2D 12 Bits (GDLA2), Enter Graphic Mode Incremental Point (GEPI2), Enter Graphic Mode Incremental Vector (GEVI2).	Page number	N/A
	Display storage address of a byte containing the selected pixel data that is part of a Draw Pixel Array (GDPXL) order.	Page number	N/A
11X0	Display storage address of the selected character following a Draw Character (GDCHAR) order.	Page number	N/A
11X1	Display storage address of the selected character following a GDCHAR order that is part of the selected segment.	Page number of the character	Name of the selected segment
1001	Display storage address of the same orders, coordinate group, and pixel data defined for B'1000'. These selected items are part of a segment.	Page number of the item	Name of the selected segment
101X	Unit address of the interrupting GSDEVI order.	Page number of the GSDEVI	Display storage address of the GSDEVI

Appendix D. 5080 Graphics System 3270 Feature

Introduction

The 3270 feature of the 5080 Graphics System consists of part of an additional card in the 5085 Graphics Processor, and additional code that operates in the Attachment Processor (AP) and the Display Processor (DP). It uses the physical keyboard, AP, and DP of the graphics processor. The 5080 Graphics System 3270 feature provides additional capability to the graphics processor to allow it to act as a 3270 type of device, thus allowing the graphics processor operator to access 3270 type applications at the same time and from the same terminal. The 3270 feature is a member of the 3270 Data Stream and the Systems Network Architecture (SNA) families of terminals.

Note: In the remainder of this appendix the designation "5080 Graphics System 3270 Feature" is abbreviated simply as "3270 feature."

Operator Interfaces

Keyboards

The same keyboard is used in 3270 mode as in graphic mode. In 3270 mode both APL and non-APL keyboard types are supported for each of the several languages. Additional, or different, controls become active in the 3270 mode than are active in the graphic mode. Details of the layout, and the meaning of keys is described in *IBM 5080 Graphics System: Operation and Problem Determination,* GA22-0133. A programmer-oriented description of the keys may be found in "Keyboard Controls" on page D-8.

Display Screen

The 5081 Display is used to display the information being presented in the 3270 mode of operation. Information is displayed in a format similar to other 3270 type implementations. The character sizes use the maximum amount of the screen in order to give good readability. The content of the screen is dictated by the host application program. Four different screen formats are supported to provide the flexibility needed by the customer application. Further information on these formats, and how they use the screen, is given in "3270 Feature Display Screen" on page D-12.

Setup Options

Certain options of the 3270 feature can be selected to meet installation requirements. These options are:

- Screen usable area

  The screen usable area (that is, the number of character positions on the screen) can be 960, 1920, 2560, or 3440 characters.

- Link connection characteristics

  The link connection characteristics are provided to allow for the use of different coding and protocol options. These are chosen by the system planner as required by the needs of his network.

The first of the characteristics is the use of NRZ or NRZI coding on the link. The second of these is the use of 2-wire or 4-wire protocol on the link, reflecting the type of modem and link in use.

- Modem wrap capability

  This option controls the operation of certain of the 3270 mode tests. These tests attempt to wrap data at various stages of the card, modem cable, and modem in an attempt to locate an error. This option tells these tests whether the attached modem supports the wrapping of data back to the 3270 feature under control of the TEST line. For more details, refer to *IBM 5080 Graphics System: Operation and Problem Determination,* GA23-0133.

- Modem link address

  This option specifies the address that the particular 3270 feature will respond to on the TP link.

These options can be selected or changed by the user using SETUP; see the manuals *IBM 5080 Graphics System: Operation and Problem Determination,* GA23-0133, and *IBM 5080 Graphics System: System Planning and Installation,* GA23-0135, for instructions on how to make the desired changes.

Data Stream Support

The 3270 feature is a member of the family of 3270 terminals. As such, it receives and transmits the data stream supported by such terminals. This data stream is described in *IBM 3270 Information Display System Data Stream Programmer's Reference,* GA23-0059.

Information about which data stream elements are supported, and values for variables, are presented in this manual. For details of structure, sequence, and values, the manual just mentioned should be consulted.

The following paragraphs give an overview of 3270 feature commands and attention identifiers (AIDs).

Data Stream Elements Supported

The commands and optional data that can be accepted by the 3270 feature are as follows:

Commands	Optional Data after Command
Write	Orders and/or user data
Erase/Write	Orders and/or user data
Erase/Write Alternate	Orders and/or user data
Erase All Unprotected	None
Read Modified	None
Read Modified All	None
Read Buffer	None
Write Structured Field	Structured fields

These commands are described in "3270 Data Stream Element Considerations" on page D-16.

Orders

The orders that can be sent with the write commands are as follows:

Set Buffer Address
Start Field
Start Field Extended
Modify Field Attribute
Set Attribute
Insert Cursor
Program Tab
Repeat to Address
Erase Unprotected to Address

These orders are described in "3270 Data Stream Element Considerations" on page D-16.

Structured Fields

The structured field functions that can be accepted by the 3270 feature are as follows:

Create Partition
Activate Partition
Destroy Partition
Set Reply Mode
Reset Partition
Erase/Reset
Write Partition:
- Write
- Erase/Write
- Erase/Write Alternate
- Erase All Unprotected

Read Partition:
- Read Modified
- Read Modified All
- Read Buffer
- Query

Specific structured field function considerations are described in "3270 Data Stream Element Considerations" on page D-16.

Attention Identifiers (AIDs)

An attention identifier, or AID, which is always the first byte of an inbound data stream, describes the nature of the data and controls that follow it. AID values supported, and their meanings, are shown in Figure D-12 on page D-23 and Figure D-13 on page D-24.

Compatibility with IBM 3270 Information Display System

The commands and orders that can be used with the 3270 feature correspond closely with those available on the IBM 3270 Information Display System, the IBM 8775 Display Terminal, and the IBM 3290 Information Panel. If a 3270 feature data stream is also required to operate with these other 3270 type devices,

the reader should consult *IBM 3270 Information Display System Data Stream Programmer's Reference,* GA23-0059, for operational differences that may affect host programming and the display operator.

Communication Link Support

The 3270 feature functions as a terminal node in a Systems Network Architecture (SNA) network. The 3270 feature is viewed by the network as a single Type-2 Logical Unit (LU) contained in a Type-2 Physical Unit (PU). It communicates with a **Boundary Function Node** (for example, an IBM 3705 Communications Controller) using Synchronous Data Link Control (SDLC) protocol. It is in the Boundary Function Node that certain information in the SNA message header is converted to a form acceptable to the 3270 feature.

Note: The term *network* has at least two meanings. A *public network* is a network established and operated by common carriers or telecommunication administrations for the specific purpose of providing circuit-switched, packet-switched, and leased-circuit services to the public. A *user application network* is a configuration of data processing products, such as processors, controllers, and terminals, established and operated by users for the purpose of data processing or information exchange, which may use communication facilities offered by common carriers or telecommunication administrators.

The term *network*, as used in this appendix, refers to a *user application network*.

Attachment to an SNA Network

The physical connection to the Boundary Function Node is over a communication data link. The data link can consist of a direct connection or a direct transmission facility. Figure D-1 on page D-5 illustrates the types of data link attachments that are possible. The 3270 feature supports data rates of between 1200 and 9600 bits per second.

SNA Relationships

The 3270 feature LU and PU communicate with associated components in a host system via SNA sessions. A full description of SNA, including sessions and layers, is given in detail in *Systems Network Architecture Concepts and Products,* GC30-3072.

Further details concerning the 3270 feature and its relationship to an SNA network may be found in "SNA and SDLC Considerations" on page D-30.

Partitions

The host program can divide the screen area into rectangular areas called *viewports*. Up to eight viewports can be defined (by the host program) and each viewport is associated with a separate character buffer called a *partition buffer*. The combination of viewport and associated buffer is called a *partition*. Each partition has a unique *Partition Identifier (PID)* whose value is in the range 0 through 7.

In order to provide compatibility with existing 3270 applications that do not recognize multiple partitions, the 3270 feature can operate in one of two states: the *base state* or the *partitioned state*.

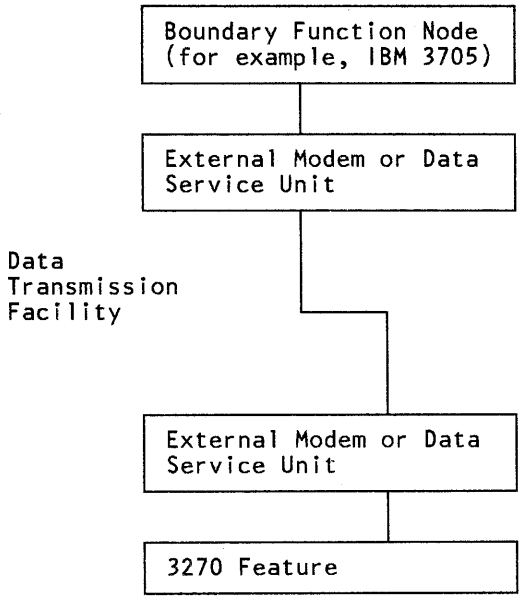


Figure D-1. Data-Link Attachment of 3270 Feature

Receipt of the Bind command (see Figure D-24 on page D-32) initially places the 3270 feature into base state, and automatically creates a single implicit partition having a PID of 0 and a screen size with *default* dimensions as specified or implied in the Bind command. (Note that the Bind command also specifies or implies *alternate* dimensions, which may or may not be the same as the default dimensions.)

The host program can replace the implicitly created base-state partition by **explicitly** creating one or more partitions of its own, thereby placing the 3270 feature into the partitioned state.

Full details of partitions, including base and partitioned states, how data may be entered and deleted from them, and unformatted and formatted partitions, are contained in *IBM 3270 Information Display System Data Stream Programmer's Reference,* GA23-0059.

Field and Character Attributes

The 3270 feature supports **Basic Field Attributes, Extended Field Attributes**, and **Character Attributes.**

The Attributes supported by the 3270 feature are:

- Basic Field Attributes
  - Protected or unprotected
  - Alphanumeric or numeric
  - Autoskip
  - Nondisplay or Display/Intensified Display
  - Detectable or Nondetectable

- Extended Field Attributes
  - Extended Highlighting
  - Color
  - Field Validation
- Character Attributes
  - Extended Highlighting Character Attribute
  - Symbol Set Character Attribute
  - Color Character Attribute

Further details on how these are supported by the 3270 feature are given in "Field/Character Attributes" on page D-16.

Character Codes

Appendix E in this manual gives the hexadecimal values of characters that may be transferred to and from the 3270 feature.

Character codes in the range X'40'–X'FF' are used to represent the graphic symbols in the (nonloadable) base and APL character sets.

Undefined Character Codes

Character codes in the range X'40'–X'FF' that are not associated with a specific graphic character are **undefined**. The action taken by the 3270 feature when an undefined character is received is dependent on the character set in use at the time.

If the base character set is in use when the undefined character code is received, the graphic that is displayed is not specified. Undefined character code values, however, are stored and retransmitted by the 3270 feature to the host upon demand without change.

If the APL character set is in use when the undefined character code is received, either as a result of the use of the Graphic Escape code or the use of a Character Attribute, the 3270 feature stores, displays, and retransmits these code values as the hyphen character X'60'.

Graphic Escape Character Code

The Graphic Escape character is a 1-byte control character with a value of X'08'.

A Graphic Escape character in the outbound data stream ensures that the following single data character is taken from the APL symbol set. If the data character is read back, it will be transmitted inbound immediately following a Graphic Escape character.

Invalid Character Codes

Character codes in the range X'00'–X'3F' are **invalid,** with the following exceptions. The exceptions are as follows: Null (X'00'), FF (X'0C'), CR (X'0D'), NL (X'15'), EM (X'19'), DUP (X'1C'), FM (X'1E'), and SUB (X'3F'); the 3270 feature rejects invalid characters with a sense code of X'1003'. The 3270 feature stores and retransmits the valid characters without change, and displays them as follows:

Character		Displayed as
Null	(X'00')	Blank
FF	(X'0C')	Blank
CR	(X'0D')	Blank
NL	(X'15')	Blank
EM	(X'19')	Blank
DUP	(X'1C')	Asterisk with overscore
FM	(X'1E')	Semicolon with overscore
SUB	(X'3F')	Solid circle

The displayed form of all the above characters is subject to Extended Highlighting and Color attributes.

Detectable Fields and Designator Characters

A field defined as **detectable** (by bits 4 and 5 of the Basic Field Attribute character) is one that can be detected by cursor select operations. However, unless a valid **designator character** is placed in the first data position of the field, no detect operation can occur.

Designator characters are used to define two types of detectable fields: *selection fields* and *attention fields*. The five designator characters supported by the 3270 feature are shown in the following table:

Designator Character	Hexadecimal Code	Purpose
? >	X'6F' X'6E'	To define a "selection" field
null space &	X'00' X'40' X'50'	To define an "attention" field

Programming Notes:

1. Both normal display and intensified display fields can be defined as detectable.

2. The application programmer should be aware that, if the field attribute byte has bit 2 = B'0' (unprotected) and bits 4 and 5 = B'01' or B'10' (detectable), the operator could modify or create a designator character, and, thus, could create either a selection field or an attention field.

3270 Mode Operator Interface

This section does not provide operator instructions; such information is given in *IBM 5080 Graphics System: Operation and Problem Determination,* GA23-0133, *IBM 5080 Graphics System: System Problem Determination,* GA23-0132, and *IBM 5080 Graphics System: Setup Instructions,* GA23-0130, supplied with the terminal. Instead, this section describes the operator interface associated with the 3270 Feature from the point of view of a host programmer. Information is given under the following headings:

- Keyboard controls
- Display screen
- Indicators in Operator Information Area

Keyboard Controls

The layouts of the various types of keyboards are illustrated in *IBM 5080 Graphics System: Operation and Problem Determination,* GA23-0133. A listing of the keys follows, in alphabetical order of the name of the key. For those keys that are common to most 3270 implementations, and for which there are no differences of operations from those implementations, no further description is given. For those keys that are not common, that operate in different ways, or that are unique to the 3270 feature, a description follows the name of the key.

Alpha Symbol Shift Key (Katakana language keyboards)

Alphanumeric Shift Key (Katakana language keyboards)

Alternate Cursor Key: Successive depressions of this key (with an Alternate Shift key also pressed) cause the displayed form of the cursor to alternate between normal underscore and reverse video.

Alternate Shift Keys: With either of the Alternate Shift keys pressed, characters or functions that appear on the **front** faces of keys can be entered.

APL On/Off Key (APL keyboards only)

Attention Key

Backspace Key

Backtab Key

Blink Key (character attribute)

Clear Key

Clear Partition Key

Clicker Key

Color Select Keys: The seven color select keys are identified on the keyboard by means of color "blobs" on the overlay. They are located on the keys labeled (on the keytops) as PF13 through PF19. When pressed (with an Alternate Shift key), these keys cause the value of the Color Character Attribute for each subsequently entered character to be set as listed below, which in turn causes the characters to be displayed on the screen in the proper color.

Color Name	Attribute Value	Key
Red	X'F2'	PF13
Pink	X'F3'	PF14
Green	X'F4'	PF15
Yellow	X'F6'	PF16
Blue	X'F1'	PF17
Turquoise	X'F5'	PF18
Neutral	X'F7'	PF19

Note that the Neutral color displays as white.

The operator-selected color remains in force until one of the following occurs:

- The operator makes another color selection.
- Field Inherit (Color) is selected by the operator.
- The operator switches to another partition (by means of the Jump Partition key) in which operator selection of color is disallowed. When this happens, the color that was most recently selected by the operator is "remembered." When the operator switches back to a partition in which operator selection of color is allowed, the "remembered" color comes back into force again.

An operator can change character color only when Color for the active partition has been specified as operator-selectable in a Set Reply Mode structured field function from the host program.

Use of these keys when the function is not allowed raises the "Input Inhibited - Minus Function" condition.

Cursor Blink Key: Successive depressions of this key (with Alternate Shift Key also pressed) cause the displayed form of the cursor to alternate between blinking and nonblinking.

Cursor Home Key

Cursor Left and Cursor Right Keys

Cursor Select Key

Cursor Up and Cursor Down Keys

Delete Character Key

Double-Speed Cursor Left and Right Keys: These keys (with an Alternate Shift key also pressed) cause the cursor to move left or right **two** character positions at a time. Wrapping action is the same as for the Cursor Left and Cursor Right keys.

Duplicate Key

Enter Key

Erase Input Key

Erase to End of Field Key

Error Override Key: With the cursor in an unprotected field, pressing this key (with an Alternate Shift key also pressed) causes the following actions:

1. A special substitute (SUB) character, value X'3F' is entered into the active partition buffer at the current cursor position.
2. The MDT bit for the field is set to 1.
3. The cursor position is advanced by one position as for normal data entry.
4. The character displays as a solid circle.
5. The character attribute is updated according to the highlighting currently specified.

The Error Override key can be used by the operator (if necessary) to fill a Mandatory Fill field with SUB characters; the SUB character is also accepted as a valid **numeric** character.

If this key is used when the cursor is at an attribute location or is in a protected field, the "Input Inhibited—Go Elsewhere" indicator tells the operator that the action is not accepted.

If the Error Override key is pressed while the 3270 feature is owned by the SSCP-LU session, the "Input Inhibited—Minus Function" condition is raised.

Field Inherit (Color) Key: This key is effective only if a color selection has previously been made by the operator. Pressing this key (with the Alternate Shift key also pressed) then causes the value of the Color Character Attribute for each subsequently entered character to be set to X'00'. This value, in turn, causes the displayed form of entered characters to be dictated by (inherited from) the Color **Field** Attribute.

The Field Inherit (Color) key has no effect if there has been no operator selection of color, either because the operator has chosen not to make a selection or because the program has disallowed operator selection of color.

Use of this key when the function is not allowed raises the "Input Inhibited—Minus Function" condition.

Field Inherit (Highlighting) Key

Field Inherit (Programmed Symbols) Key: A label for this function is present on the keyboard, but support is not provided for the function.

Field Mark Key

Insert Mode Key

Jump Partition Key

Jump Screen Key: Operation of this key (with an Alternate Shift key also pressed) causes the display screen and keyboard to become associated with the Graphic mode of operation. From the operator's point of view, the 3270 mode of operation is suspended and its status with respect to keyboard operations is stored. The graphic mode of operation is restored. Operation of the Jump Screen key while in graphic mode causes the graphic mode to be suspended, and the 3270 mode of operation is resumed, with the keyboard status being the same as when 3270 mode was left.

While the processor is in graphic mode, the 3270 feature continues in operation and may interact with the host. In particular, if host interaction was initiated before the Jump Screen operation, the operations associated with this continue while graphic mode is active. Data may be transmitted to the host, data may be received from the host and the character buffer and any status will be updated. This may result in a completely different display when 3270 mode is next resumed.

If the interaction with the host changes the status of the keyboard, the new status will be effective when the 3270 mode is next resumed.

Kana Symbol Shift Key (Katakana language keyboards)

Katakana Shift Key (Katakana language keyboards)

New Line Key

Program Access (PA) Keys

Program Function (PF) Keys

PSA-PSF (Programmed Symbols) Keys: Labels for this function are present on the keyboard, but support is not provided for the function.

Reset Key

Reverse Video Key (character attribute)

Setup Key: Operation of this key (with the Alternate Shift key also pressed) causes the display screen and keyboard to be associated with the processor SETUP function. The 3270 mode of operation is suspended, and its status with respect to keyboard operations is stored. The SETUP function (or mode) is started. At the end of the SETUP function, the 3270 mode of operation is resumed, with the keyboard status being the same as when 3270 mode was left.

While the processor is in SETUP mode, the 3270 feature continues in operation and may interact with the host. In particular, if host interaction was initiated before the SETUP operation, the operations associated with this continue while SETUP is active. Data may be transmitted to the host, data may be received from the host, and the character buffer and any status will be updated. This may result in a completely different display when 3270 mode is resumed.

If the interaction with the host changes the status of the keyboard, this new status will be effective when 3270 mode is resumed.

System Request Key

Tab Key

Test Key: Operation of this key (with an Alternate Shift key also pressed) causes the 3270 feature to go into Test mode (see *IBM 5080 Graphics System: Operation and Problem Determination,* GA23-0133, and *IBM 5080 Graphics System: System Problem Determination,* GA23-0132) unless the screen is owned by an LU-LU session, or is owned by an SSCP-LU session reached from an LU-LU session by use of the System Request key; in either of these cases, pressing the Test key causes the "Input Inhibited—Minus Function" condition to be raised. Thus, Test mode cannot be entered while the 3270 feature is logged on to the application program.

Underscore (Highlighting) Key (character attribute)

Uppercase Shift Keys (Table V language keyboards)

Uppercase Shift Lock Key (Table V language keyboards)

3270 Feature Display Screen

The 3270 feature uses a rectangular area of the screen (reflected in the Frame Buffer) as shown in Figure D-2. Since the 3270 mode uses fewer pixels in both directions than are used by graphic mode, there are margins all around the 3270 area; these margins will be black. The margins are balanced in the horizontal direction. In the vertical direction, the top margin is smaller than the bottom margin.

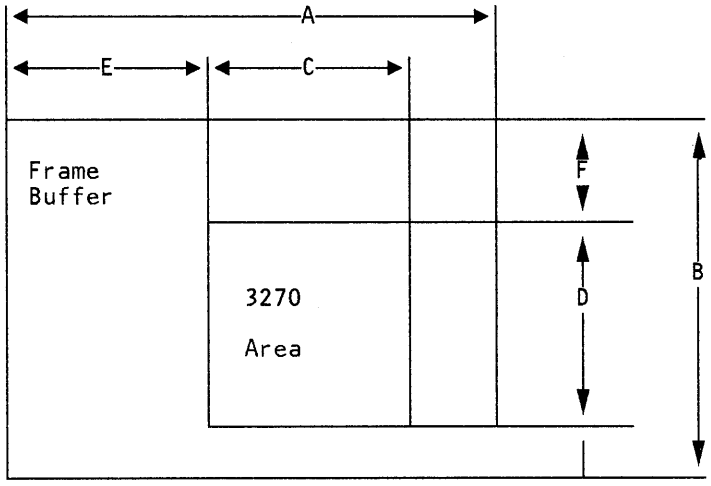
The Frame Buffer pixels described in Figure D-2 are used to support rows of characters as shown in Figure D-3.

In Figure D-3:

- A Line is one set of pixels across the screen,
- A Row is a number of lines sufficient to display a set of characters across the screen.
- Skip Lines are blank lines whose contents are **not** controlled by the contents of the character buffer, but which are added to provide more space between rows of characters.
- The Indicator Space is a set of blank lines always present that provide a blank margin at the bottom of the Data area of the screen.
- The Indicator Line is a set of solid visible lines always present that separates the Data area from the Operator Information Area.

- The Operator Information Area is a set of symbols that show the status of the 3270 feature. This area is described below in "Indicators in Operator Information Area" on page D-14.

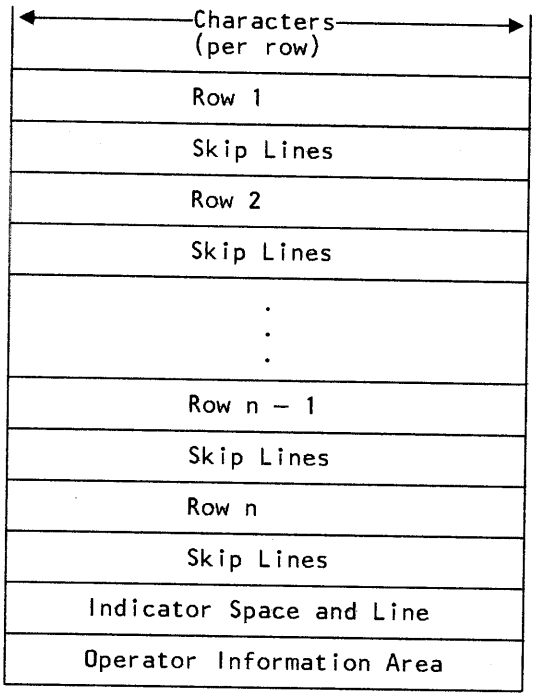
The rows of characters described in Figure D-3 are composed of character cells that are adjacent to one another. These cells contain both the character pixels and the pixels that provide the interletter spacing. The number of rows, and the characters per row, for the various different Usable Areas supported are shown in Figure D-3.



A	Frame Buffer size, horizontal	1024
B	Frame Buffer size, vertical	1024
C	3270 Pixel Area, horizontal	960
D	3270 Pixel Area, vertical	Note
E	3270 Margin area, left (and right)	32
F	3270 Margin area, top	16

Note: Value depends on Usable Area specified (see Figure D-3).

Figure D-2. 3270 Feature Use of Monitor Screen and Frame Buffer



Usable Area Number:	1	2	3	4
Cell size	12X22	12X22	12X22	12X22
Characters per row	80	80	80	80
Pels per row	960	960	960	960
Lines per row	22	22	22	22
Number of rows	12	24	32	43
Skip Lines per row	5	5	0	0
Total Lines per row	27	27	22	22
Unused lines, top	162	0	0	0
Unused lines, bottom	162	0	0	0
Subtotal lines	648	648	704	946
Indicator space, lines	18	18	18	18
Indicator line, lines	4	4	4	4
Oper info area, lines	22	22	22	22
Subtotal lines	44	44	44	44
Total Lines	692	692	748	990

Figure D-3. 3270 Feature Screen Layout, Rows and Characters

Indicators in Operator Information Area

The Operator Information Area appears as a line of characters and symbols in the bottom row of the screen. This area is independent from the screen area accessible by program.

The format of the Operator Information Area depends on whether the terminal is in run mode or in test mode. The mode is indicated by the symbol that appears in the extreme left position of the Operator Information Area.

Use of the Operator Information Area in run mode and test mode is described in *IBM 5080 Graphics System: Operation and Problem Determination,* GA23-0133, which explains the meaning of the displayed symbols. A brief description of the information that can be displayed in **run mode** follows.

Operator Information Area in Run Mode: In run mode, the Operator Information Area is divided into zones in which indicators give the operator status information in the following categories:

State of Terminal Readiness:	Indicators show when the terminal is ready and when a communication link has been established.
Screen Ownership:	Indicators show whether the terminal is owned by an SSCP-LU session or an LU-LU session, or is unowned.
Input Inhibited Conditions:	There are over 20 different Input Inhibited conditions, each with a unique indicator (consisting of symbols and characters). Each condition is assigned a priority; when multiple Input Inhibited conditions exist, the one with the highest priority is indicated. All Input Inhibited indicators start with an "X" symbol, which means that the keyboard is disabled.
Communication Status:	Indicators show whether the data link is ready, or if there is a communication problem (either inside or outside the 3270 feature terminal).
Programmed Symbols Selection:	An indicator shows the symbol set (corresponding to APL, or the base character set) that will be applied to the next character entered at the keyboard. If the keyboard does not have APL, as specified via SETUP, no Programmed Symbols Selection indicators are displayed on the screen, and the only character set available is the base character set.
Extended Highlighting Selection:	Indicators show the extended highlighting that will be applied to the next characters entered at the keyboard; a supplementary indicator shows that the operator has either (1) selected the highlighting characteristic specifically, or (2) chosen to allow the highlighting to be dictated by (inherited from) the Extended Highlighting Field Attribute supplied by the host program. (The absence of the supplementary indicator shows that the character highlighting is determined solely by the Extended Highlighting Field Attribute and cannot be changed by the operator.)

Color Selection:

Indicators show the color that will be applied to the next characters entered at the keyboard; a supplementary indicator shows that the operator has either (1) selected the color characteristic specifically, or (2) chosen to allow the color to be dictated by (inherited from) the Color Field Attribute supplied by the host program. (The absence of the supplementary indicator shows that the character color is determined solely by the Color Field Attribute and cannot be changed by the operator.)

If the 5081 Display is monochromatic, or if there are fewer than four bit planes in the graphics processor, the 3270 feature operates in a monochromatic manner, and no color indicators are displayed.

3270 Data Stream Element Considerations

This section deals with the programming considerations of the 3270 data stream elements supported by the 3270 feature. A listing of all elements is included. For those elements that have no operands, or where all operands and operand values are supported, there is no further description. For those elements where the 3270 feature expects or supplies specific values that may differ from other 3270 implementations a description of the element is included.

Field and Character Attributes

The 3270 feature supports **Basic Field Attributes**, **Extended Field Attributes**, and **Character Attributes**.

Field Attributes

Field attributes specify the appearance on the display screen of **all** characters in the field that follows the Field Attribute character (unless overridden by Character Attributes). The field attributes supported by the 3270 feature are:

- Basic Field Attributes
  - Protected or unprotected
  - Alphanumeric or numeric
  - Autoskip
  - Nondisplay or Display/Intensified Display
  - Detectable or Nondetectable
- Extended Field Attributes
  - Extended Highlighting
  - Color
  - Field Validation

Basic Field Attributes: (The 3270 feature's Basic Field Attribute corresponds to the single field **attribute** character referred to in 3270 publications.) Figure D-4 shows the bit definitions for the Basic Field Attribute byte.

Bit No.	Bit Values and Meanings
0	Value can be B'0' or B'1'; actual value is ignored; see Note 1.
1	Value can be B'0' or B'1'; actual value is ignored; see Note 1.
2 and 3	Value B'00' = Unprotected; alphanumeric. Value B'01' = Unprotected; numeric. See Note 2. Value B'10' = Protected; alphanumeric. Value B'11' = Protected; numeric; autoskip.
4 and 5	Value B'00' = Normal display; "nondetectable." Value B'01' = Normal display; "detectable." See Note 3. Value B'10' = Intensified display; "detectable." Value B'11' = Nondisplay; "nondetectable."
6	Value B'0' (Reserved)
7	Modified Data Tag (MDT) Bit identifies modified data fields during "Read Modified" operations Value B'0' = Field has not been modified Value B'1' = Field has been modified This bit can also be set by program in the outbound data stream.

Notes:

1. If this attribute is to be used in a data stream for a 3270 device, bit 1 must be B'1', and bit 0 should be determined by the contents of bits 2 through 7; see *IBM 3270 Information Display System Data Stream Programmer's Reference,* GA23-0059.
2. Bits 2 and 3 = B'11' cause the cursor to perform an "automatic skip" over this field.
3. "Detectable" means that the first character in the field is treated as a **designator** character. If the character is a space, a null, an "&," a ">," or a "?," the field can be detected by Cursor Select key; see "Detectable Fields and Designator Characters" on page D-7.

Figure D-4. Basic Field Attribute (Attribute Type X'C0')

Extended Highlighting Extended Field Attribute: Figure D-5 shows the values for the Extended Highlighting Field Attribute that are supported by the 3270 feature.

Value	Meaning
X'00'	Normal (default) (in a Character Attribute,X'00' means Field Inherit).
X'F1'	Blink
X'F2'	Reverse Video
X'F4'	Underscore

Note: These values are permissible in both Extended Highlighting **Field** and **Character** attribute bytes.

Figure D-5. Extended Highlighting Attribute (Attribute Type X'41')

Symbol Set Extended Field Attribute: The Symbol Set Field Attribute is defined by the host program to specify the symbol set to be used by the 3270 feature in displaying data characters.

The symbol sets that can be specified in a Symbol Set Field Attribute are as follows:

- The nonloadable base character set related to the character set language of the terminal.

Figure D-6 on page D-18 shows the possible values for the Symbol Set Field Attribute.

Value	Meaning
X'00'	Base character set, nonloadable (default)

Notes: This is the only permissible value in a Symbol Set **Field** Attribute; see Figure D-9 on page D-19 for the values that are permitted in the Symbol Set **Character** Attribute.

Figure D-6. Symbol Set Field Attribute (Attribute Type X'43')

Color Extended Field Attribute: The Color Field Attribute is defined by the host program to specify the color to be used by the 3270 feature in displaying data characters.

In all cases the 3270 feature stores the Color Field Attribute value and returns it, unchanged, to the application program (upon request) in subsequent inbound transmissions. The 3270 feature ignores the Color Field Attribute, and displays all characters in white whenever the display is a monochrome monitor, or when there are less than 4 bit planes in the Frame Buffer. If the display is a color monitor and there are 4 or more bit planes in the Frame Buffer, colors as specified are used.

The 3270 feature, when displaying color, also honors the Field Attribute of Intensity, so that any given color can be displayed at either of two levels of intensity.

The 3270 feature does not support "base color" mode, wherein four colors are generated based upon the intensity and protect attributes and there is no color specified via the color attributes.

Figure D-7 shows the values for the Color Field Attribute that can be accepted by the 3270 feature.

Value	Meaning
X'00'	Default color (in a Character Attribute, X'00' means Field Inherit).
X'F1'	Blue
X'F2'	Red
X'F3'	Pink
X'F4'	Green
X'F5'	Turquoise
X'F6'	Yellow
X'F7'	Neutral

Figure D-7. Color Field Attribute (Attribute Type X'42')

Field Validation Extended Field Attribute: The validation tests and actions performed by the 3270 feature for each of these types of validation classes, and the ways in which they can interact, are described more fully in "Field Validation Tests and Actions Performed" on page D-45.

Figure D-8 on page D-19 shows the values for the Field Validation Field Attribute that are supported by the 3270 feature.

Bit No.	Bit Values and Meanings
0-4	Value must be '00000'. Reserved.
5	Value '1' = Mandatory Fill
6	Value '1' = Mandatory Enter
7	Value '1' = Trigger Field

Figure D-8. Field Validation Field Attribute (Attribute Type X'C1')

Character Attributes

The following Character Attributes are supported by the 3270 feature:

- Extended Highlighting Character Attribute
- Symbol Set Character Attribute
- Color Character Attribute

Extended Highlighting Character Attribute: The permissible values for the Extended Highlighting Character Attribute are as shown for the Field Attribute in Figure D-5 on page D-17.

Symbol Set Character Attribute: The permissible values for the Symbol Set Character Attribute are as shown in Figure D-9.

Value	Meaning
X'00'	Field inherit (default)
X'F1'	LCID for APL (nonloadable) symbol set; see Note.

Note: The value X'F1' is permissible only in the Symbol Set **Character** Attribute, not in the Symbol Set Extended **Field** Attribute.

Figure D-9. Symbol Set Character Attribute (Attribute Type X'43')

Color Character Attribute: The permissible values for the Color Character Attribute are as shown for the Field Attribute in Figure D-7 on page D-18.

Outbound Data Streams

Write-Type Commands

The following write-type commands are supported by the 3270 feature.

- Write command

  If the data stream consists of the command byte only, it is treated as a no-operation (except that it serves as a negative reply to a trigger-field input transmission).

- Erase/Write command

  If the data stream consists of the command byte only, it is treated as a no-operation (except that it serves as a negative reply to a trigger-field input transmission).

- Erase/Write Alternate command

  If the data stream consists of the command byte only, it is treated as a no-operation (except that it serves as a negative reply to a trigger-field input transmission).

- Erase All Unprotected command
- Read Modified command
- Read Modified All command
- Read Buffer command
- Write Structured Field (WSF) command

The Write Control Character

The format of the outbound Write Control Character is shown in Figure D-10.

Byte(s)	Content	Meaning
0	Bits 0-3	Any value; actual value is ignored; see Note 1.
	Bit 4	Start Printer; see Note 2.
	Bit 5	Sound Alarm; when set to 1, this bit causes the alarm to be sounded.
	Bit 6	Restore Keyboard; when set to 1, this bit restores operation of the keyboard (by resetting the "Input Inhibited" System Lock or Wait indicator) and resets the AID byte to X'60'.
	Bit 7	Reset MDT Bits; when set to 1, this bit causes the MDT bits of all Basic Field attribute bytes in the buffer of partition 0 to be reset to 0 before any orders or data characters are processed.

Notes:

1. If this command is to be used in a data stream for a 3270 device, bit 1 of the WCC must be B'1', and bit 0 should be determined by the contents of bits 2 through 7; see *IBM 3270 Information Display System Data Stream Programmer's Reference*, GA23-0059.
2. If a WCC with bit 4 set to 1 is sent to a 3270 feature, a sense code of X'0801' is returned.

Figure D-10. Outbound Write Control Character

Buffer Control Orders

The following buffer control orders are supported by the 3270 feature:

- Set Buffer Address order
- Start Field order
- Start Field Extended order

The values for the Symbol Set extended field are not supported.

- Modify Field Attribute order

  The values for the Symbol Set extended field are not supported.

- Set Attribute order

  The value of Symbol Set Character Attribute is supported for the base character set and the APL character set only.

- Insert Cursor order
- Program Tab order
- Repeat to Address order
- Erase Unprotected to Address order

Structured Field Functions

The following Structured Field Functions are supported by the 3270 feature:

- Set Reply Mode structured field function
- Query structured field function
- Create Partitions structured field function

  The formats and functions supported are shown in Figure D-11 on page D-22.

- Activate Partition structured field function
- Destroy Partition structured field function
- Reset Partition structured field function
- Erase/Reset structured field function
- Write Partition structured field function
  - Write
  - Erase/Write
  - Erase/Write Alternate
  - Erase All Unprotected
- Read Partition structured field function
  - Read Modified
  - Read Modified All
  - Read Buffer
  - Query

Byte(s)	Content	Meaning
0 and 1	X'0003' min.	Length of this structure (in bytes); see Notes 1 and 2.
2	X'0C'	Create Partition function code
3	PID X'00'-X'07'	Partition Identifier of partition being created; default = X'00'.
4	Bits 0-3 = X'0'	Reserved; default = X'0'.
	Bits 4-7 = X'0' = X'1'	Address mode; 12/14-bit address mode Default = X'0'. 16-bit address mode
5	X'00'	Reserved; default = X'00'.
6 and 7	D	Depth of partition presentation space in rows; see Notes 3 and 4; default is the default screen depth specified in the Bind command.
8 and 9	W	Width of partition presentation space in columns see Notes 3 and 5; default is the default screen width specified in the Bind command.
10 and 11	Rv	Offset (in rows) of viewport's top edge from top of screen; default = X'0000'; see Note 3.
12 and 13	Cv	Offset (in columns) of viewport's left edge from left side of screen; default = X'0000'; see Note 3.

Notes:

1. Only bytes 0 through 2 of this function are mandatory; optionally, the parameters after byte 2 may be added progressively up to the maximum length (see Note 2). For parameters that are omitted, the 3270 feature assumes the default values.

2. The maximum length of this structure is 14 (X'000E').

3. "Guidelines on the Creation of Partitions" on page D-53 gives the restrictions to be observed when establishing the values of D, W, Rv, and Cv.

Figure D-11. Outbound Data Stream for the Create Partition Structured Field Function

Read Operations

The following types and formats of inbound operations are supported by the 3270 feature, in support of the corresponding outbound commands:

- Short Read
- Read Modified
- Read Modified All
- Read Buffer
- Query Replies (see "Query Replies Operation on page D-25 for details").

The formats of the data stream are standard. The operator enter actions supported, with the corresponding AID values, are shown in Figure D-12.

Operator Enter Action	AID Value	Resultant Type of Inbound Operation
Clear key Clear Partition key (see Note 1) PA1 key PA2 key PA3 key	X'6D' X'6A' X'6C' X'6E' X'6B'	Short Read operation
PF1 - PF9 keys PF10 - PF12 keys PF13 - PF21 keys PF22 - PF24 keys Enter key Detection of attention field with " & " designator character; see Note 2.	X'F1'-X'F9' X'7A'-X'7C' X'C1'-X'C9' X'4A'-X'4C' X'7D' X'7D'	Read Modified operation (addresses and data of all modified fields in the presentation space, nulls suppressed)
Detection of attention field with null or space designator character; see Note 2.	X'7E'	Read Modified operation (addresses only of all modified fields in the presentation space); see Note 4.
Trigger Field action, when operator attempts to move cursor out of a trigger field; see Note 3.	X'7F'	Read Modified operation (address and data of trigger field only, nulls suppressed)

Notes:

1. The Clear Partition key can be used only when the 3270 feature is in partitioned state.
2. Detection of an attention field can be performed by the Cursor Select key.
3. Trigger fields can exist only when the Field Validation display function is in use.
4. The host program can issue a Read Modified All command to transmit the data from the modified fields if it is required subsequently.

Figure D-12. Operator Enter Actions, Corresponding AID Values, and Types of Inbound Operation

Summary of Read States

The 3270 feature supports the standard read states for an SNA connected terminal. These states are summarized in Figure D-13 and in Figure D-14.

Initiating Command (or Structured Field Function) from Host Program with Terminal in Normal Read State	AID Value	Resultant Type of Inbound Operation
Read Modified command	X'60'	Read Modified operation
Read Modified All command	X'60'	X'60' Read Modified All operation
Read Buffer command	X'60'	Read Buffer operation
Read Partition—Read Modified structured field function *	X'61'	Read Modified operation
Read Partition—Read Modified All structured field function *	X'61'	Read Modified All operation
Read Partition—Read Buffer structured field function *	X'61'	Read Buffer operation
Read Partition—Query structured field function	X'88'	Query Replies operation

* Note that unsolicited read-type structured field functions are acceptable in base state, but only if the identified partition (PID value) is 0.

Figure D-13. Normal Read State: Inbound Operations and AID Values

Initiating Command (or Structured Field Function) from the Host Program with Terminal in Retry State	AID Inbound	Resultant Type of Operation
Read Modified command in response to an operator-initiated transmission	Same as original AID	The operator-initiated transmission is repeated
Read Modified command in response to any other (nonoperator) transmission	Same as original AID	The operation is repeated
Read Modified All command	Same as original AID	Read Modified All operation
Read Buffer command	Same as original AID	Read Buffer operation
Read Partition structured field function of any type	Function is rejected with a sense code of X'0829'	

Figure D-14. Retry State: Inbound Operations and AID Values

In response to an outbound Read Partition—Query structured field function (the Query function), the 3270 feature transmits query replies as a string of structured fields. The types of query replies that the 3270 feature can transmit are as follows:

Value in Byte 3	Type of Query Reply
X'81'	Usable Area query reply
X'84'	Partitions query reply
X'85'	Symbol Sets query reply
X'86'	Color query reply
X'87'	Highlighting query reply
X'88'	Reply Modes query reply
X'8A'	Field Validation query reply

The following paragraphs describe each of these query replies.

Usable Area Query Reply: The Usable Area query reply has the format and values shown in Figure D-15 on page D-26.

Byte(s)	Content	Meaning
0 and 1	X'0015'	Length of this structured field (in bytes)
2	X'81'	Query Reply type code
3	X'81'	Usable Area type of query reply
4	Bits 0 and 1	Reserved
	Bit 2 = B'0'	The Write Partition and the Reset Usable Area structured field functions can be accepted.
	Bit 3 = B'0'	Indicates that this is not a hardcopy device.
	Bits 4-7 = X'3'	12/14-bit and 16-bit addressing modes are supported; see Note 1.
5	X'00'	Reserved
6 and 7	X'0050'	Width of usable screen area in character columns (= 80)
8 and 9		Depth in character rows of usable screen area (the screen size chosen via SETUP)
	X'000C'	12 rows
	X'0018'	24 rows
	X'0020'	32 rows
	X'002B'	43 rows
10	X'01'	Units-of-measure code (= millimeters)
11-14	X'011C0400'	Horizontal pitch of display points = 0.278 millimeters
15-18	X'011C0400'	Vertical pitch of display points = 0.278 millimeters
19	X'0C'	Number of display points (horizontally) in the displayed dot matrix for each character = 12 for all screen sizes
20	X'16'	Number of display points (vertically) in the displayed dot matrix for each character = 22 for all screen sizes

Note:

1. Addressing modes, and the values of addresses in the various forms can be found in *IBM 3270 Information Display System Data Stream Programmer's Reference*, GA23-0059.

Figure D-15. Inbound Data Stream for a Usable Area Query Reply

Partitions Query Reply: The Partitions query reply is transmitted with the format and values shown in Figure D-16. Note that the content of bytes 5 through 7 depends on the screen size selected via SETUP.

Byte(s)	Content	Meaning
0 and 1	X'0008'	Length of this structured field (in bytes)
2	X'81'	Query Reply type code
3	X'84'	Partitions type of query reply
4	X'08'	Maximum number of partitions supported
5 and 6	X'0D70' X'0A00' X'0780' X'03C0'	Maximum partition size; see Notes 1 and 2. = 3440 characters (= screen size of 80 x 43) = 2560 characters (= screen size of 80 x 32) = 1920 characters (= screen size of 80 x 24) = 960 characters (= screen size of 80 x 12)
7	X'00'	Vertical scrolling not supported

Notes:

1. The maximum partition size is attainable only for a single partition.
2. The value returned here is the same as the screen size selected via SETUP.

Figure D-16. Inbound Data Stream for a Partitions Query Reply

Symbol Sets Query Reply: The Symbol Sets query reply is transmitted with the format and values shown in Figure D-17.

Byte(s)	Content	Meaning
0 and 1	L	Length of this structured field (in bytes) (see Note 1)
2	X'81'	Query Reply type code
3	X'85'	"Symbol Sets" type of query reply
4	Bit 0 = B'1'	Graphic Escape (APL) is supported
	Bit 1	Reserved
	Bits 2 and 3 = B'00'	The Load Programmed Symbols structured field function is not supported.
	Bits 4-7	Reserved
5	X'00'	Reserved
6	X'0C'	Width of character (symbol) cell in symbol store = 12 display points (dots)
7	X'16'	Depth of character (symbol) cell in symbol store = 22 display points (dots)
8	X'00'	No Data type formats supported.
9-11	X'000000'	Reserved
12	X'03'	Length of each following descriptor
13	X'00'	Store number for base character set
14	Bit 0 = B'0'	Symbol store is nonloadable
	Bits 1-7 = B'0000000'	Reserved
15	X'00'	LCID for base character set
16	X'01'	Store number for APL character set
17	Bit 0 = B'0'	Symbol store is nonloadable
	Bits 1-7 = B'0000000'	Reserved
18	X'F1'	LCID for APL character set

Figure D-17. Inbound Data Stream for a Symbol Sets Query Reply

Color Query Reply: The Color query reply is transmitted with the format and values shown in Figure D-18 on page D-28 or Figure D-19 on page D-29.

The Color query reply indicates to a host program that the 3270 Feature accepts and stores color attribute values. Depending upon the attached display and number of bit planes installed, one of two forms of the reply will be sent. The form sent when the display is color and there are four or more bit planes (Figure D-18) indicates that the colors displayed will be as specified by the character or extended field attribute. Otherwise, the form sent (Figure D-18) indicates that all characters are displayed in white.

Byte(s)	Content	Meaning
0 and 1	X'0016'	Length of this structured field
2	X'81'	Query Reply type code
3	X'86'	Color type of query reply
4	X'00'	Reserved
5	X'08'	The number of attribute specifications (byte pairs) that follow
6	X'00'	Default color attribute value
7	X'F7'	Actual color displayed = Neutral (White)
8	X'F1'	Color attribute value for Blue
9	X'F1'	Actual color displayed = Blue
10	X'F2'	Color attribute value for Red
11	X'F2'	Actual color displayed = Red
12	X'F3'	Color attribute value for Pink
13	X'F3'	Actual color displayed = Pink
14	X'F4'	Color attribute value for Green
15	X'F4'	Actual color displayed = Green
16	X'F5'	Color attribute value for Turquoise
17	X'F5'	Actual color displayed = Turquoise
18	X'F6'	Color attribute value for Yellow
19	X'F6'	Actual color displayed = Yellow
20	X'F7'	Color attribute value for Neutral
21	X'F7'	Actual color displayed = Neutral

Figure D-18. Inbound Data Stream for a Color Query Reply with Color Displayed. This form of the data stream is used when color is displayed: There is a color display and four or more bit planes.

Byte(s)	Content	Meaning
0 and 1	X'0016'	Length of this structured field
2	X'81'	Query Reply type code
3	X'86'	Color type of query reply
4	X'00'	Reserved
5	X'08'	The number of attribute specifications (byte pairs) that follow
6	X'00'	Default color attribute value
7	X'FA'	Actual color displayed = White
8	X'F1'	Color attribute value for Blue
9	X'00'	Actual color displayed = Default (white)
10	X'F2'	Color attribute value for Red
11	X'00'	Actual color displayed = Default (white)
12	X'F3'	Color attribute value for Pink
13	X'00'	Actual color displayed = Default (white)
14	X'F4'	Color attribute value for Green
15	X'00'	Actual color displayed = Default (white)
16	X'F5'	Color attribute value for Turquoise
17	X'00'	Actual color displayed = Default (white)
18	X'F6'	Color attribute value for Yellow
19	X'00'	Actual color displayed = Default (white)
20	X'F7'	Color attribute value for Neutral
21	X'00'	Actual color displayed = Default (white)

Figure D-19. Inbound Data Stream for a Color Query Reply with Color Not Displayed. This form of the data stream is used when color is not displayed: There is a monochrome display or there is less than four bit planes.

Highlighting Query Reply: The Highlighting query reply is transmitted with the format and values shown in Figure D-20.

Byte(s)	Content	Meaning
0 and 1	X'000D'	Length of this structure (in bytes)
2	X'81'	Query Reply type code
3	X'87'	Highlighting type of query reply
4	X'04'	Number of byte pairs that follow
5	X'00'	Attribute value for default highlighting
6	X'F0'	Default Highlight action code (= normal)
7	X'F1'	Attribute value for blink highlighting
8	X'F1'	Blink action code
9	X'F2'	Attribute value for reverse video highlighting
10	X'F2'	Reverse Video action code
11	X'F4'	Attribute value for underscore highlighting
12	X'F4'	Underscore action code

Figure D-20. Inbound Data Stream for a Highlighting Query Reply

Reply Modes Query Reply: The Reply Modes query reply is transmitted with the format and values shown in Figure D-21.

Byte(s)	Content	Meaning
0 and 1	X'0007'	Length of this structured field (in bytes)
2	X'81'	Query Reply type code
3	X'88'	Reply Modes type of query reply
4	X'00'	Field mode of reply is supported
5	X'01'	Extended field mode of reply is supported
6	X'02'	Character mode of reply is supported

Figure D-21. Inbound Data Stream for a Reply Modes Query Reply

Field Validation Query Reply: The Field Validation query reply is transmitted with the format and values shown in Figure D-22.

Byte(s)	Content	Meaning
0 and 1	X'0005'	Length of this structured field
2	X'81'	Query Reply type code
3	X'8A'	Field Validation type of query reply
4	X'07'	Indicates that Mandatory Fill, Mandatory Enter, and Trigger Field functions are supported.

Figure D-22. Inbound Data Stream for a Field Validation Query Reply

SNA and SDLC Considerations

SNA Support

Systems Network Architecture (SNA) is a set of rules to which certain IBM telecommunications products conform. As a member of the SNA family of terminals, the 3270 feature performs its functions in an identical manner to similar functions implemented in other SNA products. This section provides information about specific support provided by the 3270 feature within the variations allowed by the architecture.

See *Systems Network Architecture Format and Protocol Reference Manual: Architectural Logic,* SC30-3112 for the complete descriptions and details of SNA.

SNA Commands Supported by 3270 Feature

The SNA commands used and supported by the 3270 feature are shown in Figure D-23. Sections following this provide notes about specific commands and concepts as they relate to the 3270 feature support.

SNA Command		Command Is Received by or Sent by Terminal	Type of Session
Name	Type		
REQMS	NS	Received	SSCP–PU
RECFMS	NS	Sent	SSCP–PU
ACTPU	SC	Received	SSCP–PU
DACTPU	SC	Received	SSCP–PU
ACTLU	SC	Received	SSCP–LU
DACTLU	SC	Received	SSCP–LU
BIND	SC	Received	LU–LU
UNBIND	SC	Received	LU–LU
CLEAR	SC	Received	LU–LU
SDT	SC	Received	LU–LU
CANCEL	DFC	Received and sent	LU–LU
CHASE	DFC	Received	LU–LU
LUSTAT	DFC	Sent	LU–LU and SSCP–LU
SHUTD	DFC	Received	LU–LU
SHUTC	DFC	Sent	LU–LU
BID	DFC	Received	LU–LU
SIGNAL	DFC	Received and sent	LU–LU

Figure D-23. SNA Commands Supported by the 3270 Feature

The 3270 feature LU is designed to operate with the Bind parameters given in Figure D-24. Where a choice is indicated, the value most appropriate for the application may be chosen.

Byte	Value	Meaning
0	X'31'	BIND Request code
1	0 0 0 0 . . . . 0 0 0 1	Bind Format 0 Bind Type 1 (cold)
2	X'03'	FM Profile 3 (LU-LU)
3	X'03'	TS Profile 3 (LU-LU)
4		PRIMARY LU PROTOCOL
	0	Single Element Chains Allowed; see Note 1.
	1	Multiple Element Chains Allowed; see Note 1.
	. 0	Immediate Request Mode
	. . 0 1	Exception Response; see Note 1.
	. . 1 0	Definite Response; see Note 1.
	. . 1 1	Definite or Exception Response; see Note 1.
	. . . . 0 0 . .	Reserved
	. . . . . 0 .	No Compression
	 1	Primary May Send EB
5		SECONDARY LU PROTOCOL
	1	Multiple Element Chains Allowed
	. 0	Immediate Request Mode; see Note 1.
	. 1	Delayed Request Mode; see Note 1.
	. . 0 1	Exception Response; see Note 1.
	. . 1 0	Definite Response; see Note 1.
	. . 1 1	Definite or Exception Response; see Note 1.
	. . . . 0 0 . .	Reserved
	. . . . . 0 .	No Compression
	 0	Secondary May Not Send EB
6		COMMON LU PROTOCOL
	0	Reserved
	. 0	FMH Not Allowed
	. . 1	Brackets Are Used
	. . . 1	Brackets Termination Rule 1 (Conditional) Applies
	. . . . 0 . . .	Alternate Code Not Allowed
	. . . . . 0 0 0	Reserved
7		COMMON LU PROTOCOL
	1 0	Half-Duplex Flip-Flop
	. . 0	PLU Responsible for Error Recovery
	. . . 0	SLU Is The First Speaker
	. . . . 0 0 0 .	Reserved
	 0	SLU Wins Contention
8	0 0	Reserved
	. . x x x x x x	SLU Send Pacing Count; see Note 2.
9	0 0	Reserved
	. . x x x x x x	SLU Receive Pacing Count; see Note 2.
10		SLU TO PLU RU SIZE; see Note 3.
	x x x x	Mantissa
	. . . . x x x x	Exponent

Figure D-24 (Part 1 of 2). Bind Parameters for the 3270 Feature

Byte	Value	Meaning
11	x x x x	PLU TO SLU RU SIZE; see Note 4. Mantissa
	. . . . x x x x	Exponent
12-13	X'0000'	Reserved
14	0	Reserved
	. 0 0 0 0 0 1 0	LU Type 2
15-19	XL5'00'	Reserved
20	x x x x x x x x	Default Screen Depth (Rows)
21	x x x x x x x x	Default Screen Width (Columns)
22	x x x x x x x x	Alternate Screen Depth (Rows)
23	x x x x x x x x	Alternate Screen Width (Columns)
24		SCREEN SIZE FOR SESSION; see Note 5.
	0 0 0 0 0 0 0 0	Unspecified
	0 0 0 0 0 0 0 1	12 rows x 40 columns (Alternate dimensions = Default dimensions)
	0 0 0 0 0 0 1 0	24 rows x 80 columns (Alternate dimensions = Default dimensions)
	0 1 1 1 1 1 1 0	As specified in bytes 20-21 (Alternate dimensions = Default dimensions)
	0 1 1 1 1 1 1 1	Default size as specified in bytes 20-21 Alternate size as specified in bytes 22-23
25	0 0 0 0 0 0 0 0	Reserved
26	0 0 0 0 0 0 0 0	Cryptography Not Supported

Notes:

1. When multiple values are given, the programmer can choose the mode of operation most suitable to the application environment. Otherwise, the parameters should be specified as shown.
2. Any value can be specified for the pacing counters. Because the pacing counters are tuning parameters they can be adjusted to match the characteristics of the particular network environment. The SLU Send Pacing Count should be tuned to the capabilities of the Boundary Function node to which the 3270 feature is attached. A zero value specifies no pacing; this should generally be used for the SLU Receive Pacing Count because the display is capable of receiving outbound requests as fast as they can be sent. Specification of no pacing prevents unneeded Isolated Pacing Responses (IPRs) from flowing.
3. The minimum size that may be specified for the SLU to PLU request unit is 256. Hence the exponent must be greater than or equal to 5. ($8 \times 2^5 = 256$.)
4. In order to be valid, byte 11 must obey the following rules:
   a. The maximum size that may be specified for the PLU to SLU request unit is 1536; so the exponent must be less than or equal to 7.
   b. The mantissa must not be less than 8.
   c. If the exponent is equal to 7, the mantissa must be less than or equal to X'C'. ($12 \times 2^7 = 1536$.)
5. Figure D-25 on page D-34 gives more detail regarding the screen size for the session.

Figure D-24 (Part 2 of 2). Bind Parameters for the 3270 Feature

Value of Bind byte 24	Screen dimensions established at Bind time when the screen size specified via SETUP is:			
	960 80 cols (max) 12 rows (max)	1920 80 cols (max) 24 rows (max)	2560 80 cols (max) 32 rows (max)	3440 80 cols (max) 43 rows (max)
X'00' See Note 1	Default=480 Alternate=480	Default=1920 Alternate=1920	Default=1920 Alternate=1920	Default=1920 Alternate=1920
X'01' See Note 2	Default=480 Alternate=480	Default=480 Alternate=480	Default=480 Alternate=480	Default=480 Alternate=480
X'02' See Note 3	(Error) Alternate=1920	Default=1920 Alternate=1920	Default=1920 Alternate=1920	Default=1920
X'7E' See Note 4	Default=D1 Alternate=D1	Default=D1 Alternate=D1	Default=D1 Alternate=D1	Default=D1 Alternate=D1
X'7F' See Note 5	Default=D1 Alternate=A1	Default=D1 Alternate=A1	Default=D1 Alternate=A1	Default=D1 Alternate=A1

"D1" and "A1" are respectively the default and alternate screen dimensions specified in bytes 20-23 of the Bind command.

The application programmer can subsequently specify that the screen dimensions are to be "default" or "alternate" by means of (1) an Erase/Write command (default dimensions), (2) an Erase/Write Alternate command (alternate dimensions), or (3) an Erase/Reset structured field function (default or alternate dimensions as specified in byte 3).

When the default and alternate dimensions are the same, an Erase/Write command is equivalent to an Erase/Write Alternate command. See also Note 6.

Notes:

1. If a screen size of 960 was specified via SETUP, the screen dimensions for the session are 12 rows x 40 columns. Otherwise, the screen dimensions for the session are 24 rows x 80 columns.
2. The default and the alternate dimensions are both 12 rows x 40 columns.
3. If a screen size of 960 was specified via SETUP, a negative response with sense code X'0821' is returned. Otherwise, the default and the alternate dimensions are both 24 rows x 80 columns.
4. The screen dimensions, both default and alternate, are as specified in bytes 20 and 21—subject to the number of columns and number of rows being not greater than the values specified via SETUP.
5. The default screen dimensions are specified in bytes 20 and 21 and the alternate screen dimensions are specified in bytes 22 and 23. Both default and alternate dimensions are subject to the number of columns and rows being less than or equal to the values implied by the setup size. The Erase/Write Alternate command performs a switch to the alternate screen dimensions.
6. If the 3270 feature is put into partitioned state, the screen dimensions become those specified via SETUP, regardless of the values indicated in the Bind command.

Figure D-25. Default and Alternate Screen Sizes in Base State

SNA Notes

Pacing (LU-LU Session): Because the 3270 feature can accept information as fast as it can be transmitted, pacing need not be specified on the PLU-to-SLU flow (that is, the PLU-to-SLU pacing value in the Bind command may be set to zero).

Segmenting (LU-LU Session): Messages sent to the 3270 feature are limited by buffer constraints to a maximum length of 256 bytes, not including headers. Thus, although an outbound RU may be up to 1536 bytes long, the Boundary Function must split the RU into segments which will fit into the 3270 feature buffers. The segment length (256 bytes) is specified during system definition time. The 3270 feature will assemble the segments into a complete RU before processing the message. As there are only eight receive buffers in the 3270 feature, an RU can be split into at most eight segments. If for any reason the

buffers are used up before the RU has been assembled, the 3270 feature discards the partial RU and returns a negative response with sense code X'8010' (Segmented RU Error).

Segmenting is not performed by the 3270 feature on inbound messages.

Chaining (LU-LU Session): The 3270 feature uses chaining as a means of splitting a large request (for example, a screenful of data) into manageable pieces. The application program may also perform chaining of requests to the 3270 feature. (Responses are **not** chained.)

SSCP-LU Session: The 3270 feature uses TS profile 1 and FM profile 0 in the SSCP-LU screen. The maximum size of the complete RU for this type of session is 256 bytes.

Request Maintenance Statistics (REQMS): The 3270 feature supports and responds to REQMS for the following four types of problem determination data:

Type 01 Counts generated by the SDLC TEST function

Type 02 Physical node summary data

Type 03 Detailed link/loop error counts

Type 05 The Engineering Change (EC) level of the 3270 feature

SDLC Support

The Synchronous Data Link Control (SDLC) protocol is used to transmit messages to and from the 3270 feature over a telecommunications link. This section discusses SDLC as implemented in the 3270 feature, but does not include SDLC fundamentals; for a discussion of the SDLC architecture see *IBM Synchronous Data Link Control General Information,* GA27-3093.

The 3270 feature is attached to a host processor via a data link. The SLDC link protocol is directed by the host processor's SDLC layer, functioning as the primary station; the 3270 feature is the secondary station.

SDLC Commands and Responses Supported

The SDLC commands and responses supported by the 3270 feature are shown in Figure D-26.

SDLC Command/ Response	SDLC Command/Response received or sent by terminal
	On Data Link
DISC	Received
SNRM	Received
TEST	Received and Sent
XID	Received and Sent
UA	Sent
DM	Sent
FRMR	Sent

Figure D-26. SDLC Unnumbered Commands and Response Supported by the 3270 Feature

SDLC Frame Address Field

This field contains the SDLC address of the secondary station. The 3270 feature recognizes both its own unique SDLC address as well as the all-ones "Broadcast" address. The 3270 feature does not recognize the all zeros (null) address. When transmitting, the 3270 feature always puts its own specific address in the field.

Response to XID Request

The response sent to the host in response to a Request for XID is shown in Figure D-27.

Byte(s)	Value	Meaning
0	0 0 0 1 . . . . 0 0 1 0	Format 1 (variable length XID) PU Type 2
1	X'15'	XID Information field length
2-5	X'019.....' X'...YYYYY'	Block number Specific terminal identification; see Note.
6-7	X'0000'	Reserved
8	X'00'	Secondary station; two-way alternating transmissions
9	X'A0'	FMD RUs allowed on SSCP-PU sessions
10-11	X'0109'	Maximum information field length (265 bytes)
12	X'00'	SDLC command profile (SNA link set)
13	X'00'	SDLC function flags (SREJ, SARM, SIM, and RIM not supported)
14-15	X'0000'	Reserved
16	X'07'	Modulus and maxout count
17	X'00'	Reserved
18	X'01'	SDLC address field length
19		SDLC address
20	X'00'	Number of dial digits

Notes: No specific ID is assigned at time of manufacture. The field will contain zero for all processors.

Figure D-27. 3270 Feature Response to an XID Request

SDLC Operations

The 3270 feature operates in one of two modes, Normal Response Mode (NRM) and Normal Disconnected Mode (NDM).

When power is turned on at the graphics processor, or the 3270 feature is reset and restarted, the Normal Disconnected Mode (NDM) is assumed. No information or supervisory frames are transmitted or accepted. No Unnumbered Responses are transmitted unless solicited by the primary station (the SNRM, XID, and TEST commands are valid in NDM). Any invalid or non-implemented command (including DISC) results in transmission of a DM response to indicate offline status. When an SNRM command is received, the 3270 feature responds UA and enters Normal Response Mode (NRM).

The 3270 feature in NRM initiates transmission only at a **response opportunity**, when solicited by the primary station. The transmission may consist of a single frame or multiple frames. A single frame, or the last in a series, generally has the F-bit on in the control (C) field.

After relinquishing its transmit opportunity, the 3270 feature awaits another solicitation from the primary station before transmitting again.

Every I frame (whether sent by the primary or the secondary) contains an Ns count which is incremented by 1 (modulo 8) for each successive I frame. If a valid I frame with a good FCS is received, but its Ns count is not 1 greater (modulo 8) than the Ns count in the previous I frame received, then a sequence error occurs. That frame is ignored, as are all subsequent frames until one with the correct Ns value is received.

The Nr count, which can be carried in RR and RNR as well as in an I frame, is used to acknowledge receipt of outstanding I frames. I frames are retained at the sending station until so acknowledged. An acknowledgment carrying Nr implies that the frame whose Ns count is one less than Nr can be released, as well as previous frames.

At each response opportunity that permits the transmission of I frames, transmission begins at the point referred to by the last Nr count received. The 3270 feature does not support selective retransmission, so **all** unacknowledged I frames will be sent. I frames pending initial transmission may also be sent, within the constraints of the modulo count. All I frames are sent in contiguous sequence by Ns count.

Certain responses **require** primary station acknowledgment. They are:

- Any I frame
- FRMR
- RR, when transmitted with the poll bit off to report clearing a previous busy condition

If left unacknowledged through the next primary station solicitation, the 3270 feature will retransmit the response. Except for I frames, such responses are transmitted indefinitely until acknowledged by the primary station. I frame responses are transmitted 15 times, at which point the 3270 feature goes offline and performs a wrap test.

The beginning flag is constructed in the 3270 feature.

Idle Timeout

An **idle timeout** occurs whenever the 3270 feature does not recognize an outbound frame with a valid address for a duration of 8 seconds. The timer is started at the end of each valid frame and is reset whenever the 3270 feature detects an outbound frame with a valid address.

When the timeout occurs, the Line Ready indicator is turned off, and a wrap test is performed. If the wrap test fails, the Machine Check indicator is turned on. If the wrap test is successful, the Communication Check indicator is turned on and the 3270 feature merely waits for a valid address; if a valid address is received, the Line Ready indicator goes back on.

SNA Sense Codes

This section lists, in numerical order, the SNA sense codes that can be issued by the 3270 feature. The explanations of these codes relate specifically to the 3270 feature; for a more generalized and formal definition of the codes, see the *System Network Architecture Format and Protocol Reference Manual: Architectural Logic,* SC30-3112.

The sense codes listed here fall into four categories:

- X'08xx'—Request Reject
- X'10xx'—Request Errors
- X'20xx'—State Errors
- X'80xx'—Path Errors

Request-Reject Sense Codes

X'0805'
Session Limit Exceeded: A Bind command with DAF=X'02' was received from a PLU while the 3270 feature was in session with another PLU.

X'0806'
Resource Unknown: Request contained an unrecognized address.

X'080C'
Procedure Not Supported: The 3270 feature cannot support the procedure requested (such as Trace).

X'0812'
Insufficient Resource: The 3270 feature temporarily has insufficient buffer size to handle the request.

X'0813'
Bracket Bid Reject: A Bid command or an FM data RU with BB set was received while the 3270 feature was neither in between-bracket state nor in pending-begin-bracket state.

X'0815'
Function Active: A Bind command with DAF=X'02' was received from a PLU with which the LU was already in session.

X'081B'
Receiver in Transmit Mode: An outbound request was received while the operator was keying in data but before an operator enter action has been performed.

X'0821'
Invalid Session Parameters: An ACTPU command or an ACTLU command was received with activation type other than "cold" or ERP, or TS and FM profile other than 1 and 0, respectively.

A Bind command with DAF=X'02' was received, the SSCP-LU session was active, the 3270 feature was not in session with a PLU, and the Bind command contained one or more invalid parameters.

(1) An ACTPU or ACTLU command was received with activation type other than "cold" or ERP; (2) an ACTPU command was received with TS and FM profile other than 1; (3) an ACTLU command was received with TS and FM profile other than 0; or (4) a Bind command was received for an SLU that was not in session with a PLU, and the Bind command contained one or more invalid parameters (for example, the SLU Send Pacing Count in byte 8 was not zero).

X'0829'

Change Direction Required: A read-type command or a read-type structured field function was received without the Change Direction bit set in the Request Header.

A read-type command or a read-type structured field function was received with the End Bracket bit set in the Request Header.

A Signal or Shutdown request was sent to the 3270 feature by the PLU.

X'082B'

Presentation Space Integrity Lost: A chain of FM data was received from the PLU which was not an Erase/Write or Erase/Write Alternate command, but the screen contents had previously been changed and the condition had not yet been reported by LUSTAT.

X'082D'

LU Busy: A chain of FM data was received from the PLU but the LU-LU session did not own the screen.

An FM data request was received from the SSCP but the LU-LU session owned the screen.

X'084C'

Permanent Insufficient Resource: Invalid or nonexistent store number was specified in a Load Programmed Symbols structured field function.

Byte 12, bits 5 through 7 of a Load Programmed Symbols structured field function (extended form) were nonzero.

X'0863'

Referenced Local Character Set Identifier (LCID) Not Found: The LCID value specified as a Character or Field Attribute is not associated with any programmed symbols store.

X'0871'

Read Partition State Error: The 3270 feature received a Read Partition structured field function while an unacknowledged inbound transmission was outstanding.

X'1003'

Function Not Supported: An RU specified a function not supported by the 3270 feature, such as one of the following:

- A session control command other than ACTPU, DACTPU, ACTLU, DACTLU, Bind, Unbind, Clear, or SDT was received.
- A data flow control command other than Cancel, Bid, Chase, Signal, Shutdown, Shutdown Complete, or LUSTAT was received.
- The first two bytes of the code on a Signal request were not X'0001'.
- Unsupported command code.
- Data was received following a read-type request or an Erase All Unprotected command.
- An unsupported structured-field type code was received in a Write Structured Field command.
- An order was received that was unsupported, or had an unrecognized code value of less than X'40'.
- A Repeat to Address order specified an invalid character.
- An unsupported attribute type or value was received in a buffer control order.
- A Graphic Escape character was received followed by a character whose code value was less than X'40'.

X'1005'

Parameter Error: An invalid parameter in an order or structured field was received such as one of the following:

- The address following a Set Buffer Address, a Repeat to Address, or an Erase Unprotected to Address order was too large.
- An order with an incomplete parameter list was received.
- Length of structured field incorrect.
- Structured field parameter missing; that is, the RU was not long enough to carry all the required parameters.
- A structured field function (other than Destroy Partition) was received with a PID value for a partition that was not created.
- A Read Partition structured field function was received but was not the last in the chain.

- A Create Partition structured field function was received in which:
  - Width or depth of presentation space = 0.
  - The specified viewport would have overlapped an existing viewport or could not have been contained on the screen.
  - The PID value was not in the range 0 through 7.
- A Read Partition—Query structured field function was received, but the PID value was not X'FF'.
- A Modify Field Attribute order was received while the current buffer address was not at a buffer position containing a field attribute.
- An address was received following a Set Buffer Address, Repeat to Address, or Erase Unprotected to Address order that was incorrectly specified (that is bits 0 and 1 of the first byte specified B'10').

X'1007'
Category Not Supported: A network control request was received on a session other than the PU-PU session.

State-Error Sense Codes

X'2001'
Sequence Number Error: The sequence number on a normal flow RU on the LU-LU session was not exactly one greater than the sequence number on the previous RU.

X'2002'
Chaining Error: Elements in a chain were not received in valid order.

X'2003'
Bracket Protocol Error: A request was received with bracket bit settings invalid for the current state of the 3270 feature bracket state manager.

X'2004'
Direction Error: The PLU did not follow proper rules for changing direction under the half-duplex flip-flop send-receive protocol.

X'2005'
Data Traffic Reset: A DFC or FMD request was received after a Bind or Clear command but before the SDT command.

X'8004'

Unrecognized Destination Address Field: The DAF on an outbound request did not correspond to the local address of the LU (X'02') or the PU (X'00').

X'8005'

No Session: A message was received on a session that was not active, and the message was not an activation request.

X'8007'

Segmenting Error: Segments were received in the wrong order.

X'8008'

PU Not Active: A request on the SSCP-LU session or the LU-LU session was received but the SSCP-PU session was not active.

X'8009'

LU Not Active: A request on the LU-LU session was received but the SSCP-LU session was not active.

X'800F'

Invalid Address Combination: A request was received with DAF=X'00', but the OAF was not X'00' or X'FF'.

X'8010'

Segmented RU Length Error: The 3270 feature buffers were used up before the complete RU was received.

Program Check Codes

Program check codes are listed in the following table.

Program Check Displayed Code (X PROG nn)	SNA Sense Code	Meaning
11	—	Negative SNA response from the host
21	—	SNA Exception Request received
22	800F	Invalid OAF for 3270 feature PU
23	8005	RU, other than ACTPU, received on SSCP-PU session while 3270 feature PU is not active
23	8008	RU received on either SSCP-LU or LU-LU session while 3270 feature PU is not active
24	8004	Unrecognized DAF
25	8007	SNA segmenting error
26	8005	RU, other than ACTLU, received on SSCP-LU session while 3270 feature LU is not active
26	8009	RU received on LU-LU session while the SSCP-LU session is not active

Program Check Displayed Code (X PROG nn)	SNA Sense Code	Meaning
27	8005	RU, other than Bind, received from a PLU before an LU-LU session has been bound
28	0821	Invalid ACTPU parameter
29	0806	Request for Formatted Maintenance Statistics received with invalid device address
30	2005	Data Traffic Reset state
31	2001	SNA sequence number error
32	2002	FM data chaining error
33	2002	Normal flow DFC received while in in-chain state
34	2003	FM data does not carry BB when BETB or PEND.BB
35	2003	DFC carries EB when BETB or PEND.BB
37	2004	HDX error (HDX flip-flop protocol)
40	1003	Invalid or unsupported 3270 feature command
41	1003	Data follows a Read Modified, a Read Modified All, Read Buffer or Erase All Unprotected command
41	1005	Read Partition structured field function received but not last in chain
42	1003	Nonsupported SNA command received; including IPL commands that are not supported
42	1007	A network control SNA command was received on a non-PU-PU session
43	1003	SNA Control Function carries Null RU
44	1003	Invalid request code in Signal RU
45	1003	Create Partition structured field with field length of X'0000' received with too many parameters
46	1003	Unsupported 3270 feature order received, or code value less than X'40' not recognized
47	1005	Nonzero PID received when in Base state
48	1003	Create Partition structured field function received with an invalid address mode specified
49	1003	Create Partition structured field function received with bits 0–3 in byte 4 of the structured field not equal to X'0'
50	1005	Order received with invalid buffer address
51	1005	Incomplete order sequence received; includes graphic escape without character

Program Check Displayed Code (X PROG nn)	SNA Sense Code	Meaning
52	1003	Order received with unsupported parameters
52	1005	Order received with invalid parameters
53	1005	Received structured field function addresses a partition number that is invalid or not created
54	1003	Structured field function received with unsupported parameters
54	1005	Structured field function received with invalid parameters
55	1003	Structured field function received with an unsupported command code
56	1005	Structured field with L = X'0000' was not last in transmission
57	1005	Structured field received with a length that is different from that specified in bytes 0 and 1
58	1005	Structured field function received in which the field length in bytes 0 and 1 is inconsistent with the command code in byte 2
59	1005	A Create Partition structured field function specified viewport that overlapped an existing viewport
60	0829	A Read Modified, Read Modified All, Read Buffer command, or a Read Partition structured field function was received for which EB.CD is required in the SNA request header (RH)
61	0871	A Read Partition structured field function was received while an unacknowledged inbound transmission was outstanding
63	0863	The LCID value specified as a field or character attribute is not associated with any programmed symbols store.
64	084C	A Load Programmed Symbols structured field function was received in which either (1) the specified store number was invalid or nonexistent; or (2) byte 12 bits 5–7 (in extended form) were nonzero.
66	1003	A Set MD Control structured field function was received, but no magnetic-stripe reading device is attached to the 3270 feature
68	0821	Invalid ACTLU parameter received
69	0815	A second Bind is received from current PLU
70	0805	While the 3270 feature was in an LU-LU session an SNA Bind command was received from another PLU

Program Check Displayed Code (X PROG nn)	SNA Sense Code	Meaning
71	0821	Bind parameter error: not enough bytes in Bind RU
72	0821	Bind parameter error: invalid parameters in bytes 1–3
73	0821	Bind parameter error: invalid PLU protocol in byte 4
74	0821	Bind parameter error: invalid SLU protocol in byte 5
75	0821	Bind parameter error: invalid common protocol in bytes 6 and 7
76	0821	Bind parameter error: less than 256 RU length specified in byte 10
77	0821	Bind parameter error: either (1) more than 1536 RU length specified in byte 11, or (2) invalid value specified in byte 11
78	0821	Bind parameter error: invalid LU type in byte 14
79	0821	Bind parameter error: invalid screen size in bytes 20–24
80	0821	Bind parameter error: byte 26 does not contain X'00' (cryptography not supported)
98	0815	Unsolicited IPL command received

Field Validation Tests and Actions Performed

This section describes the effects of the Field Validation attribute. By means of bits in the Field Validation Attribute (see Figure D-8 on page D-19), a field can be classified as being one or more (or none) of the following:

- A **Mandatory Enter** field
- A **Mandatory Fill** field
- A **Trigger** field

The following paragraphs describe the tests and actions performed by the 3270 feature for each of these validation classes; notes are also included to show how the validation classes interact with one another.

The descriptions of mandatory fill and trigger fields use the terms **primed** and **modified**; these terms have similar meanings but are nevertheless quite distinct:

- A field is said to be **primed** when its **contents** have been modified by the operator. A field can be primed **only** by the operator. Priming a field sets the MDT bit to 1. The operator can prime a field as follows:

  - Entering data characters, space characters, Error Override characters, or Duplicate characters;

    and/or

  - Deleting characters with the Delete key or the Erase to End of Field key.

  A field is subsequently **unprimed**:

  - By a write-type request to the partition containing the primed field.

  - By the operator pressing the Erase Input key.

  - By a successful mandatory fill validation test or a trigger field action as described below; thus there can be only one primed field in a partition, and the cursor will be in that field.

- A field is said to be **modified** when its MDT bit is set to 1. The MDT bit can be set to 1 in any of the following ways:

  - By the operator **priming** the field, as described above.

  - By the operator detecting on an attention-type or selection-type detectable field with the Cursor Select key. (Note that this action does not prime the field: The contents of the field are not changed.)

  - By the host program transmitting the Basic Field attribute with the MDT bit (bit 7) = 1.

    The MDT bit can be reset to 0 (that is, the field becomes "unmodified") as follows:

    — By the host program transmitting a write-type request to the partition with a write control character (WCC) that specifies "reset MDT bits" (WCC bit 7 = 0).

    — By the operator detecting on a detectable field or pressing the Erase Input key.

Note: In the following discussion about keystrokes, it should be noted that the Jump Screen (JMP SCR) key and the SETUP key may be pressed at any time with no effect on the status of tests, states, and so on. If either of these two keys are pressed, their corresponding action will be initiated (see "Keyboard Controls" on page D-8) and the status of the 3270 feature will remain as it was until return is made to 3270 mode. Operation then resumes as if the mode change had not taken place.

Mandatory Enter

The 3270 feature performs a mandatory enter validation test when an operator enter action (other than a Trigger Field action) tries to initiate a "Read Modified" type of inbound operation (an inbound transmission that includes **data**). As shown in Figure D-12 on page D-23, the operator actions that can do this are: (1) operation of a program function (PF) key or the ENTER key; and (2) detection of a detectable field that has an "&" designator character (by using the Cursor Select key).

The validation test searches the partition, starting at the top, for an unprotected, mandatory field that has its MDT bit set to 0.

- If such a field is found, the cursor is moved to the first position in that field, no transmission occurs, and the "Input Inhibited—Mandatory Enter" condition is raised. The operator can remove the Input Inhibited indicator by pressing the Reset key, and then modify the field and repeat the enter action; the 3270 feature then repeats the mandatory enter validation test.

- If no such field is found, the inbound transmission is allowed to take place. The mandatory enter validation test allows transmission to take place only when all unprotected mandatory enter fields in the partition have their MDT bit set to 1.

Programming Note for Use of Mandatory Enter Fields: The mandatory enter field classification has no effect if specified for a protected field.

Mandatory Fill

The 3270 feature performs a mandatory fill validation test when the cursor is in a **primed** mandatory fill field and either of the following operator actions is performed: (1) the operator tries to move the cursor out of the field; or (2) an operator enter action tries to initiate an inbound transmission that includes **data** (see Figure D-12 on page D-23). (If the operator moves the cursor into and out of an **unprimed** mandatory fill field, the mandatory fill validation test does **not** take place.)

The mandatory fill validation test searches the field for null characters.

- If a null character is found, the keystroke that caused the validation test is not processed, the cursor is not moved out of the field, and the "Input Inhibited—Mandatory Fill" condition is raised. The operator can remove the Input Inhibited indicator (by pressing the Reset key), fill the field with nonnull characters, and then move the cursor out of the field or repeat the enter action; the 3270 feature then repeats the mandatory fill validation test.

- If no null character is found, then the keystroke that caused the validation test is processed normally, after which the field is unprimed.

Programming Notes for the Use of Mandatory Fill Fields:

1. Mandatory Fill has no effect if specified for a protected field.

2. If a field is classed as a mandatory fill field **and** as a trigger field, the mandatory fill validation test is performed before the trigger field action.

3. The host program should be aware that an unfilled mandatory fill field (that is, a field that still contains at least one null character) **can** be transmitted inbound. This can happen in the following circumstances:

   a. If the host program presets the MDT bit for the field to 1; and/or if the host program leaves the MDT bit set to 1 by not specifying "reset MDT bits" in a write request (WCC bit 7 = 1). Even though the operator may not have filled the field with nonnull characters, the field will be transmitted inbound in a "Read Modified" operation that transmits data.

   b. If the operator action causes the MDT bit to be set to 1 without the field being primed. This occurs when the operator detects (by Cursor Select key) a field that has a question mark (?) or an ampersand (&) designator character. (Detection when the designator character is an ampersand also constitutes an operator enter action that initiates the inbound transmission.)

4. Use of the Jump Partition key is independent of the mandatory fill validation test. If the operator switches to another partition while the cursor is in a primed mandatory fill field, the **screen** cursor leaves the field, but the current cursor position for the partition is not changed and the mandatory fill validation test is not performed. When the operator switches back to the original partition, the original situation is recreated as if the screen cursor had never left the partition.

5. The SUB character (X'3F') can be used as well as, or instead of, data characters in order to complete a mandatory fill field; a SUB character is entered by pressing the Error Override key (with an Alternate Shift key also pressed).

Trigger

The 3270 feature performs a trigger field action when the operator tries to move the cursor out of a **primed** trigger field. (If the operator moves the cursor into and out of an **unprimed** trigger field, the trigger field action does **not** occur.)

The trigger field action causes a "Read Modified" inbound operation to occur, with an AID of value X'7F'. This operation transmits inbound only the current cursor position (within the trigger field) and the address and contents of the trigger field; no other fields or addresses are transmitted. The operation sets INPID to the currently active partition identifier, and INOP to "Read Modified." (If the field is also classified as a mandatory fill field, then the mandatory fill validation test must be satisfied before the trigger field can be transmitted; see "Mandatory Fill" on page D-47.)

While the host program is validating the field, the cursor remains in the trigger field, but the keystroke that caused the trigger action (for example, Tab, Cursor Up, Cursor Down) determines the **next position** for the cursor.

The 3270 feature then waits for a reply from the host program during which time the operator may continue keying data. Up to 30 more keystrokes are accepted and are kept on a queue waiting to be processed when an affirmative reply is received (see text below). The following keystrokes and actions however are acted upon immediately: the Attention key, all shift keys, and the System Request key. Note that pressing the System Request key causes the keystroke queue to be purged (that is, emptied and not processed).

While keystrokes are being queued, the cursor does not move and the keystrokes are not displayed. If the operator attempts to queue more than 30 keystrokes, the "Input Inhibited—Trigger Queue Full" condition is raised.

The reply from the host program can be **affirmative** or **negative** (or a nonreply); see "Replies to Trigger Field Input" on page D-50.

- When an **affirmative** reply is received:

  1. The cursor is moved to its next position (unless overridden by an Insert Cursor order contained in the reply).

  2. The 3270 feature processes queued keystrokes (see Note below). Data characters are displayed, starting at the cursor position.

  3. If the "Input Inhibited—Trigger Queue Full" condition had been raised, it is now reset and the 3270 feature processes the queued keystrokes (see Note below). When the 3270 feature completes the processing of the queued keystrokes, the "Input Inhibited—What?" condition is raised to warn the operator that one or more keystrokes will have been lost. The operator can clear this condition by pressing the Reset key and then continue keying. The keyboard shift is set to lowercase shift (non-Katakana) or alphanumeric (Katakana).

  Note: The 3270 feature processes queued keystrokes in their original sequence as if they were being keyed in normally. If any queued keystroke raises an Input Inhibited condition, then following keystrokes in the queue are ignored until a Reset keystroke is encountered. If the transmission carrying the affirmative reply contains more than one structured field function, then processing of the second (and subsequent) structured field functions is delayed until after the queued keystrokes have been processed. If in these circumstances, a queued keystroke attempts to initiate an inbound operation, then the "Input Inhibited—What?" condition is raised and the following keystrokes in the queue are ignored (unless a Reset keystroke is encountered).

- When a **negative** reply is received, the keystroke queue is emptied (that is, the queued keystrokes are lost), and an "Input Inhibited—Invalid Field" condition is raised (and any "Trigger Queue Full" condition is cleared). Typically, the host program might send a write-type request, with a WCC that does not specify "restore keyboard" (WCC bit 6 = 0) to display an error message, to highlight the error, and to reposition the cursor; the operator could then use the Reset key to remove the Input Inhibited indicator and try entering the data again. If a "Trigger Queue Full" condition was present, the keyboard shift is set to lowercase shift (non-Katakana) or alphanumeric (Katakana).

The following paragraphs define the types of reply that the host program can send in response to a trigger field inbound operation, namely:

- An affirmative reply
- A negative reply
- A nonreply

When the host program sends several structured field functions in a "Write Structured Field" command after a trigger field inbound transmission, only the first such function is interpreted as the reply; any subsequent structured field function is not regarded as a reply.

The following descriptions of affirmative and negative replies assume that there are no data stream errors; if there are errors, then the transmission is interpreted as a nonreply, and the keystroke queue is not affected.

Affirmative Reply: An affirmative reply can be one of the following:

- A Write Partition—Write structured field function addressed to the inbound partition (INPID) with a WCC that specifies "restore keyboard" (bit 6 = 1).
- A Write command with a WCC that specifies "restore keyboard" (bit 6 = 1), if the inbound partition (INPID) is 0.
- An SNA null RU with CD or EB.

Negative Reply: A negative reply can be one of the following:

- Any of the following Write Partition structured field functions addressed to the inbound partition INPID:
  - Write with a WCC that does **not** specify "restore keyboard" (bit 6 = 0).
  - Erase/Write or Erase/Write Alternate with WCC bit 6 having the value of 0 **or** 1.
  - Erase All Unprotected.
- Any of the Write Partition structured field functions addressed to a partition other than INPID, irrespective of the command code and WCC value.
- Any of the following structured field functions, **regardless** of the partition addressed:
  - Set Reply Mode
  - Reset Partition
  - Create Partition
  - Destroy Partition
  - Activate Partition
  - Erase/Reset

- An Erase/Write, Erase/Write Alternate, or an Erase All Unprotected command (regardless of the value of INPID).
- A Write command with the WCC bit 6 ("restore keyboard") set as follows:
  - If INPID is 0, WCC bit 6 = 0.
  - If INPID is not 0, WCC bit 6 = 0 or 1.
- A Write Structured Field command with no structured fields.
- A Write, Erase/Write, or Erase/Write Alternate command or structured field function with no WCC.

Nonreplies: Any outbound transmission other than an affirmative reply or a negative reply is a nonreply. A nonreply preserves the keystroke queue, but does not cause it to be processed. A nonreply can be one of the following:

- A null RU without CD or EB.
- A Read Partition structured field function.
- A Read Modified, Read Modified All, or Read Buffer command.

Programming Notes for Use of Trigger Fields:

1. The trigger field classification has no effect if specified for a protected field.
2. To avoid operator confusion, the host should not move the cursor **out** of the Trigger field in a negative reply.

   In an affirmative reply, the host program can move the cursor from its position within the Trigger field to a new position; the 3270 feature then processes the keystroke queue from there. Thus, any data entry keystrokes in the queue will be placed at the host-specified position. This allows the host program to perform "logical cursor movement" in which the position of the cursor is made to depend on the validity or content of a previous field. If such function is programmed, care should be taken to ensure that the operator understands the operating procedure for that application.
3. To avoid annoying the operator, the host program should send a reply to a trigger field input with minimum delay; a reply time of less than one second is suggested. Whether or not this reply time can be achieved depends on several factors, such as software path lengths, hardware configuration, and the amount of processing time required by the host program to validate the data in the trigger field. No firm rules can be given, but the host validation time should be minimized as far as possible—for example, by avoiding lengthy searches of disk files or data bases; such tests should be performed separately without delaying the reply.

4. The host program should be aware of the events that occur when:

   - The partition contains an unmodified mandatory enter field; and

   - The operator performs an enter action (other than moving the cursor) when the cursor is in a primed trigger field.

     The operator enter action invokes the mandatory enter validation test which **tries** to move the cursor out of the trigger field into the mandatory enter field. However, the cursor does not move and a trigger action takes place (as described earlier). Keystrokes are queued while a trigger reply is awaited.

     – If an affirmative reply is received, the mandatory enter validation test continues by moving the cursor to the unmodified mandatory enter field and raising the "Input Inhibited—Mandatory Enter" condition. The queued keystrokes are then processed. However, because of the "Input Inhibited" condition, these keystrokes are discarded unless a Reset keystroke is encountered, after which subsequent keystrokes in the queue are accepted and processed normally. (If the keystroke queue did not contain a Reset keystroke, then all the queued keystrokes are discarded.) The operator can then modify the Mandatory Enter field (after pressing the Reset key if necessary) and repeat the enter action.

     – If a negative reply is received, then the "Input Inhibited—Invalid Field" condition is raised, and the cursor remains in the trigger field unless that cursor position is overridden by an Insert Cursor order in the reply.

   - An operator enter action performed while the cursor is in a trigger field does **not** cause a trigger action - unless an unmodified mandatory enter field also exists in the partition as described in the preceding note. (By contrast, an operator enter action when the cursor is in a primed mandatory fill field **does** cause the mandatory fill validation test to take place.)

   - At the time of a trigger field input operation, the MDT bit is normally set to 1 because the operator input will have primed the field; the trigger field action does not change the MDT bit setting. However, the host program should be aware that the MDT bit can subsequently be reset to 0 and that the field would not then be transmitted inbound in a Read Modified operation caused, for example, by the operator pressing ENTER. Ways in which the MDT bit of a trigger field (or any other field) can be reset include:

     – The host program specifying "reset MDT bits" (WCC bit 7 = 1) in the reply to the trigger field input.

     – The operator detecting the field with the Cursor Select key when the field is detectable and has the greater-than sign (>) in the designator character position.

     – Pressing the Erase Input key.

– Use of the Jump Partition key is independent of the trigger field action. If the operator switches to another partition while the cursor is in a primed trigger field, the **screen** cursor leaves the field, but the current cursor position for the partition is not changed and the trigger field input function is not performed. When the operator switches back to the original partition, the original situation is re-created as if the screen cursor had never left the partition.

Guidelines on the Creation of Partitions

This section describes some of the factors and limitations that need to be considered when creating partitions.

For each viewport that the application user sees on the screen, the application program must create a partition. A partition is created by the program issuing a Create Partition structured field function (see Figure D-11 on page D-22). Up to eight partitions can be created for use at any one time; each partition has a unique partition identifier (PID) in the range 0 through 7.

The remainder of this section deals with:

- Usable Screen Area
- Row and Column Numbering
- Layout of Viewports on Screen
- Presentation Spaces

The concepts of viewport and presentation space are presented under the heading "Partitions" on page D-4.

For the purpose of discussion, it is assumed that the application designer decides upon the viewport layout (that is, the sizes and positions of all viewports on the screen) before considering the creation of any one partition.

Usable Screen Area

The usable screen area available for the display of viewports depends on the screen size selected via SETUP, as shown below:

Screen size selected at setup time:	Screen depth (rows)	Screen width (columns)
960	12	80
1920	24	80
2560	32	80
3440	43	80

Row and Column Numbering

A particular position in the usable screen area and in the area of a presentation space is identified by its row number and column number: row numbering starts at the top of the area with row number 1; column numbering starts at the left of the area with column number 1.

Layout of Viewports on Screen

A viewport on the screen is defined by the following parameters (examples of which are shown in Figure D-28 on page D-55):

- Dv and Wv, which refer to the **size** of the viewport. Dv and Wv respectively specify the **viewport depth** (in screen rows) and the **viewport width** (in screen columns).

- Rv and Cv, which refer to the **position** on the screen of the viewport's origin (top left corner). Rv specifies the offset (in rows) between the top of the usable screen area and the top row of the viewport; similarly, Cv specifies the offset (in columns) between the left edge of the usable screen area and the leftmost column of the viewport.

  Because they represent **offsets**, the values of Rv and Cv are each one less than the absolute row and column numbers of the viewport's origin. For example, to place the viewport origin in the top left corner of the screen area, specify Rv=0 and Cv=0, or allow the default values to take effect.

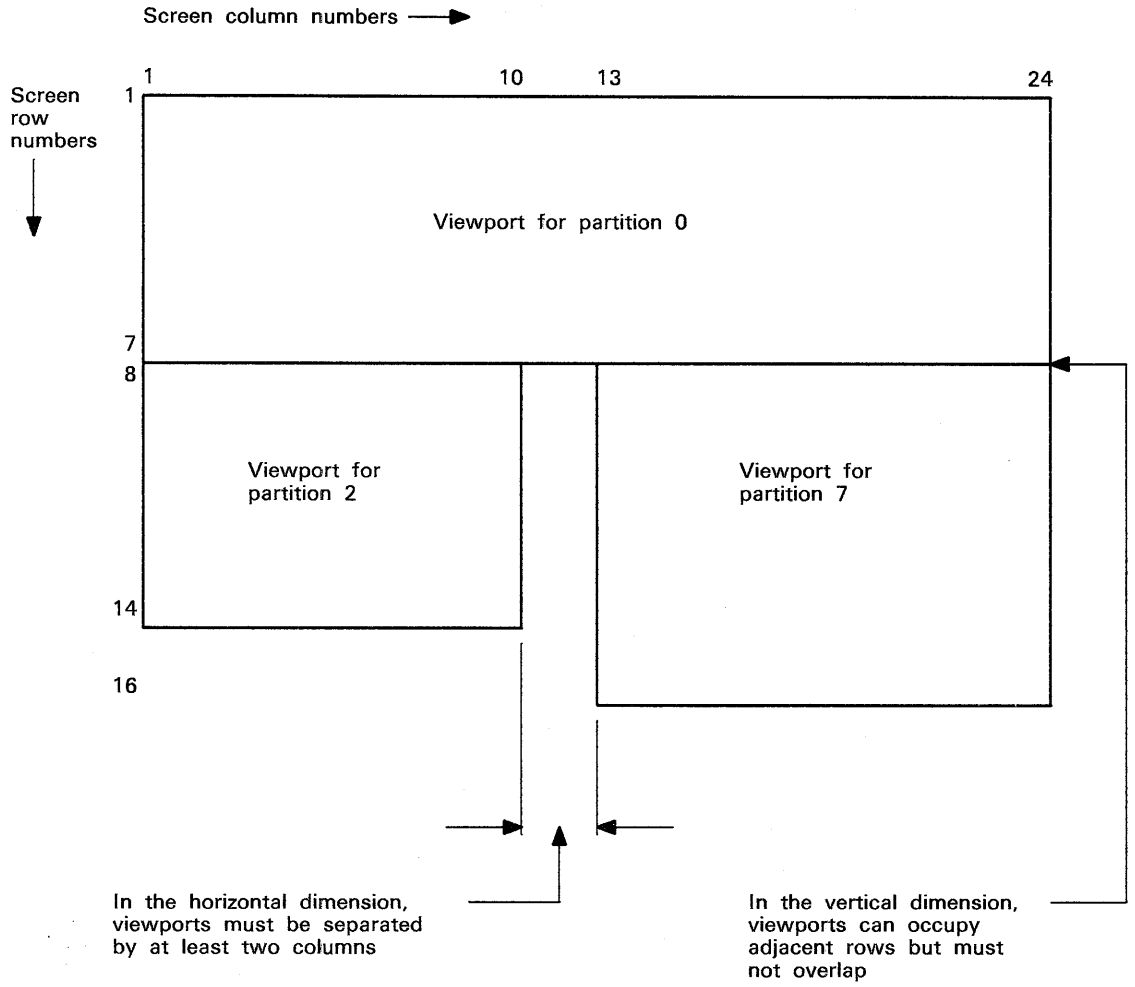
The main rule concerning the layout of viewports on the screen is that they must not overlap; additionally, in the horizontal dimension, adjacent viewports must be separated by at least two columns. Figure D-28 illustrates this rule with an example screen layout; the corresponding Create Partition parameters (Dv, Wv, Rv, and Cv) are also shown.

The only other rule is that the size (Dv and Wv) and the position (Rv and Cv) of each viewport must be chosen so that the whole of the viewport appears within the usable screen area. (The usable screen area, in partitioned state, is the size established via SETUP, namely: 12x80, 24x80, 32x80, or 43x80.) Therefore, to accommodate the example layout shown in Figure D-28, the usable screen area would need to be at least 24 rows x 80 columns.

Presentation Spaces

For each partition being created, the presentation space is defined by D and W, which respectively specify the presentation space **depth** (in rows) and **width** (in columns).

Since the 3270 feature does not support scrollable partitions, the presentation space is always the same size as the viewport. Accordingly, the depth of the presentation space (D) for a partition must have a value corresponding to the required depth of the viewport (Dv). (The Create Partition structured field function cannot specify the Dv or Wv parameters; see Figure D-11 on page D-22. The D and W parameters, which define the size of the presentation space, also define the size of the viewport.)



Note: *For this example layout of viewports, the viewport parameters required in the three 'Create Partition' structured field functions are as follows:*

For partition 0:
Dv = 7 (X'0007'), Wv = 24 (X'0018')
Rv = 0 (X'0000'), Cv = 0 (X'0000')
For partition 2:
Dv = 7 (X'0007'), Wv = 10 (X'000A')
Rv = 7 (X'0007'), Cv = 0 (X'0000')
For partition 7:
Dv = 9 (X'0009'), Wv = 12 (X'000C')
Rv = 7 (X'0007'), Cv = 12 (X'000C')

Figure D-28. Example of Adjacent Viewports on a Screen—Showing the Rules of Juxtapositioning

3270 Feature Use of RS232C Interface

The 3270 feature connection on the 5085 backpanel for attachment of the 3270 modem cable is a 25-pin male D-connector. The following RS232C lines are supported:

Pin Number	RS232C	Circuit Descriptive Name
1	AA	Protective Ground
2	BA	Transmitted Data (TxD)
3	BB	Received Data (RxD)
4	CA	Request to Send (RTS)
5	CB	Clear to Send (CTS)
6	CC	Data Set Already (DSR)
7	AB	Signal Ground
8	CF	Received Line Signal Detector (DCD)
11		— Select Standby
15		DB Transmit Signal Element Timing (DCE Source) (TxC)
17		DD Receive Signal Element Timing (DCE Source) (RxC)
18		— Test
20		CD Data Terminal Ready (DTR)
22		CE Ring Indicator (RI)
23		CH Data Signal Rate Selector

These lines are driven or terminated using 1488 and 1489 type drivers and receivers, except lines on pins 1 and 7. Protective ground (pin 1) is supplied to the modem cable by means of a separate post and nut on the backpanel, which is connected to machine ground. Signal ground (pin 7) is connected to signal ground internally.

Input line 22 (ring indicator) is terminated on the card, but the output of the receiver is not used.

Output lines 11 (select standby) and 23 (data signal rate selector) are driven by transmitters at a fixed level of OFF.

3270 Feature Modem Cable

The modem cable is supplied as a part of the 3270 feature. These cables provide a Test/Operate switch at the modem end of the cable that provides a wrap capability to aid in problem determination procedures. This is the same cable that is supplied by the TP connected versions of the IBM 3274 Display Control Unit.

Abbreviations

A	Address
ACTLU	Activate Logical Unit
ACTPU	Activate Physical Unit
AID	Attention Identifier
APL	A Program Language
ATTC	Attribute Count
ATTN	Attention
BB	Begin Bracket
BC	Begin Chain
BCN	Beacon
BETB	Between Brackets
BSM	Brackets State Machine, SNA
C	Column (screen)
C	Control
Cv	Horizontal offset of viewport origin
CBA	Current Buffer Address
CCP	Current Cursor Position
CD	Change Direction
CFGR	Configure
Ch.	Chapter
CMDR	Command Reject (see FRMR)
col(s)	Column(s)
CR	Carriage Return
D	Depth of presentation space
Dv	Depth of viewport
DACTLU	Deactivate Logical Unit
DACTPU	Deactivate Physical Unit
DAF	Destination Address Field
DAF'	Destination Address Field' (known as DAF "prime")
DFC	Data Flow Control
DISC	Disconnect
DM	Disconnected Mode (also called Request Online)
DUP	Duplicate
EB	End Bracket
EBCDIC	Extended Binary-Coded Decimal Interchange Code
EC	End Chain
EM	End of Message
EO	Eight Ones
EOI	End of Inquiry
EOR	End of Record
ERP	Error Recovery Protocol
EUA	Erase Unprotected to Address
F	Flag
FCS	Frame Check Sequence
FF	Form Feed
FIC	First-in-Chain
FM	Field Mark
FM	Function Management
FMD	Function Management Data
FRMR	Frame Reject (also called Command Reject)
GA	Go Ahead
GE	Graphic Escape

HDX	Half Duplex
I frame	Information frame
I/O	Input/Output
IC	Insert Cursor
ID	Identification
INB	In Bracket
INOP	Inbound Operation
INPID	Inbound Partition Identifier
IPR	Isolated Pacing Response
L	Length
LCID	Local Coded Identifier
LH	Link Header
LIC	Last-in-Chain
LRC	Longitudinal Redundancy Check
LU	Logical Unit
LUSTAT	Logic Unit Status
max.	Maximum
MDT	Modified Data Tag
MFA	Modify Field Attribute
MIC	Middle-in-Chain
min.	Minimum
MRP	Mandatory Response Poll
Nr	SDLC Receive Count
Ns	SDLC Send Count
NC	Network Control
NDM	Normal Disconnected Mode
NL	New Line
NRM	Normal Response Mode
NSA	Nonsequenced Acknowledgment (see UA)
NSP	Nonsequenced Poll (also called Unnumber Poll)
NU	National Use
NUL	Null
OAF	Origin Address Field
OAF'	Origin Address Field' (known as OAF "prime")
OIC	Only-in-Chain
ORP	Optional Response Poll
P/F bit	Poll/Final bit (SDLC)
PA	Program Access
PF	Program Function
PID	Partition Identifier
PLU	Primary Logical Unit
PS	Programmed Symbols
PT	Program Tab
Rv	Vertical offset of viewport origin on screen
RA	Repeat to Address
RCV	Receive
RECFMS	Record Formatted Maintenance Statistics
REQMS	Request Maintenance Statistics
RH	Request/Response Header
RNR	Receive Not Ready
ROL	Request Online (see DM)
RR	Receive Ready
RU	Request/Response Unit

SA	Set Attribute
SBA	Set Buffer Address
SC	Session Control
SDLC	Synchronous Data Link Control
SDT	Start Data Traffic
Send-NT	Send-Not-Transmit
Send-T	Send Transmit
SF	Start Field
SFE	Start Field Extended
SHUTC	Shutdown Complete
SHUTD	Shutdown
SLU	Secondary Logical Unit
SNA	Systems Network Architecture
SNRM	Set Normal Response Mode
SP	Space
SSCP	System Services Control Point
SUB	Substitute
SYS REQ	System Request
TA	Turnaround
TH	Transmission Header
UA	Unnumbered Acknowledgment (also called Nonsequenced Acknowledgment)
UF	Unnumbered Format
UP	Unnumbered Poll (also called Nonsequenced Poll)
W	Width of presentation space
Wv	Width of viewport
WCC	Write Control Character
WSF	Write Structured Field
XID	Exchange Identification

Appendix E. 5080 Graphics System I/O Interface Codes

This appendix describes and illustrates the codes accepted from and presented to the host processor for character data.

Graphics Mode

The following notes apply to the Graphics mode I/O interface codes for all languages:

1. The Graphics mode charts in this section apply when the character set ID (CSID) in use is either X'00' (3250 compatibility) or X'04', X'05', X'06', or X'07.' Other rules apply when other CSID values are used. See "Graphics System Character Set Generation" on page 3-33 for information related to these other values.

2. Codes with asterisks cannot be created from the keyboard.

3. Codes indicated by multiple-letter symbols are:

NUL	Null
NL	New Line
BS	Backspace (Reqd)
CC	Cursor Control
SM	Set Mode
SP	Space

4. Codes with an entry in the code table are valid codes. All other codes are invalid and are treated as follows:

   - Invalid codes in the range X'40' to X'FF' are displayed as character spaces (X'40').

   - When CSID = X'00' is in use, invalid codes in the range X'00' through X'3F' are treated like nulls (X'00'). New Line (X'15') and Backspace (Reqd) (X'16') are valid codes and are processed as described in "ANK Input and the Cursor" on page 3-12.

   - When CSID = X'04, X'05', X'06', or X'07' is in use, invalid codes in the range X'00' through X'3F' are displayed on the screen using a character representation of a hyphen. New Line (X'15') and Backspace (Reqd) (X'16') are treated as invalid characters.

   - In all cases, the code value received, or entered via the keyboard, is kept and is transmitted to the host on demand.

   - IBM reserves the right to make changes to the display representation of invalid characters (in the full range of X'00' through X'FF') for these character set identification values.

See the following figures for illustrations of Graphics mode I/O interface codes:

Figure E-1 on page E-4. English (US) I/O Interface Codes for Graphics Mode

Figure E-3 on page E-6. English (UK) I/O Interface Codes for Graphics Mode

Figure E-5 on page E-8. Swedish I/O Interface Codes for Graphics Mode

Figure E-7 on page E-10. German I/O Interface Codes for Graphics Mode

Figure E-9 on page E-12. French I/O Interface Codes for Graphics Mode

Figure E-11 on page E-14. Italian I/O Interface Codes for Graphics Mode

Figure E-13 on page E-16. Katakana I/O Interface Codes for Graphics Mode

3270 Mode

The following notes apply to the 3270 mode I/O interface codes for all languages:

1. All entries that are not blank are considered as defined; entries that are blank are considered undefined.

2. Codes for defined entries in the range of X'40' to X'FE' result in the display of the character shown. Codes for undefined entries may result in the display of characters. The character displayed for a given undefined code may be different from other devices. IBM reserves the right to change at any time the character displayed for an undefined character code.

3. The NL (X'15') (New Line), EM (X'19') (End of Message), FF (X'0C') (Form Feed), CR (X'0D') (Carriage Return), and NUL (X'00') (Null) are not displayed.

4. The DUP (X'1C') (Duplicate) and FM (X'1E') (Field Mark) are displayed as '*' and ';', respectively, each with an overbar; the SUB (X'3F') is displayed as a solid circle.

5. Other codes in the range of X'00' to X'3F' are rejected with a sense code of X'1003'.

See the following figures for illustrations of 3270 mode I/O interface codes:

Figure E-2 on page E-5. English (US) I/O Interface Codes for 3270 Mode

Figure E-4 on page E-7. English (UK) I/O Interface Codes for 3270 Mode

Figure E-6 on page E-9. Swedish I/O Interface Codes for 3270 Mode

Figure E-8 on page E-11. German I/O Interface Codes for 3270 Mode

Figure E-10 on page E-13. French I/O Interface Codes for 3270 Mode

Figure E-12 on page E-15. Italian I/O Interface Codes for 3270 Mode

Figure E-14 on page E-17. Katakana I/O Interface Codes for 3270 Mode

3270 APL Mode

The following notes apply to the APL mode I/O interface codes:

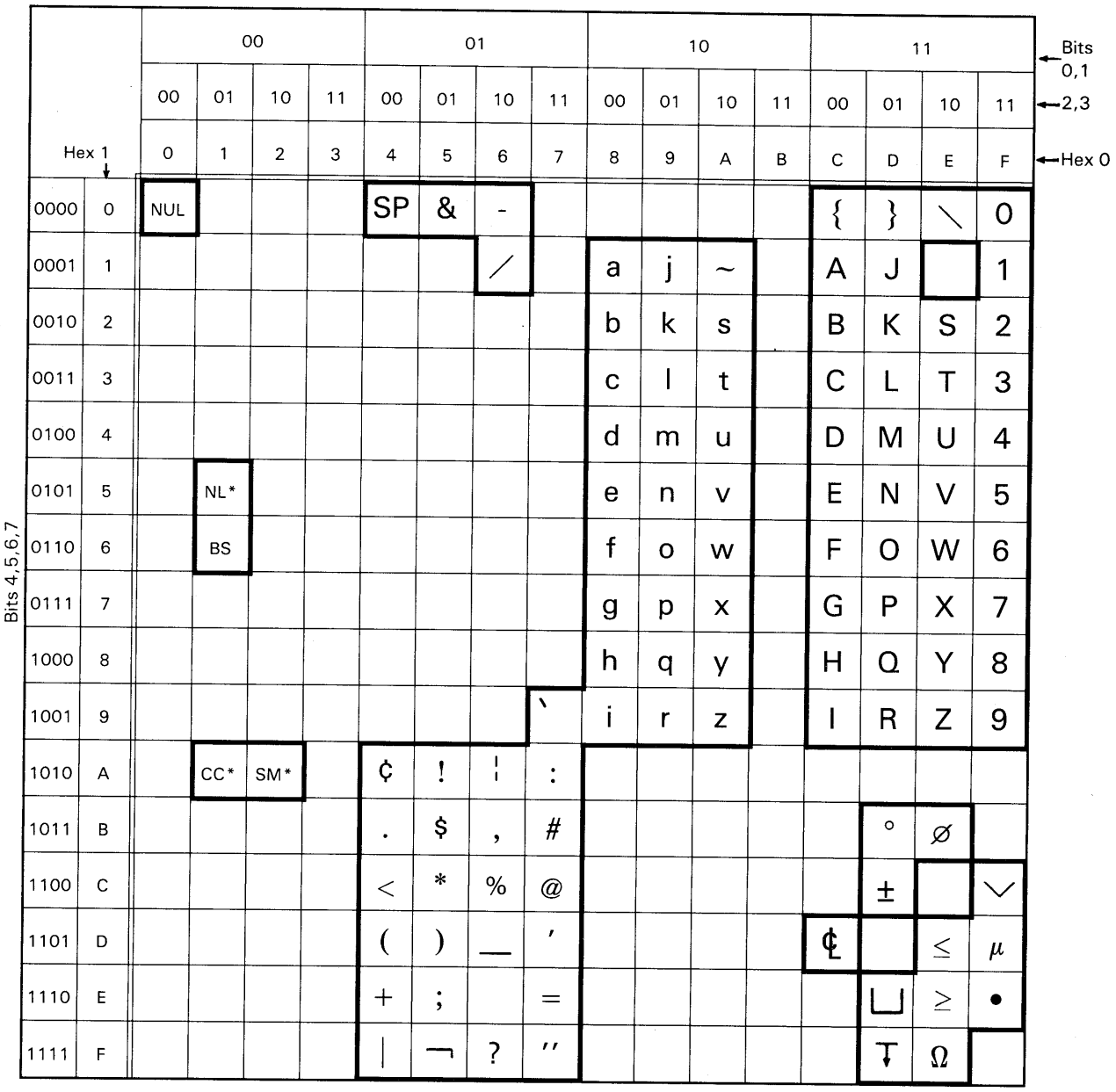
1. All entries that are not blank are considered as defined; entries that are blank are considered undefined.

2. Codes for defined entries result in the display of the character shown.

3. Codes for undefined entries result in the display of a hyphen (-) and the return of an X'60' to the host.

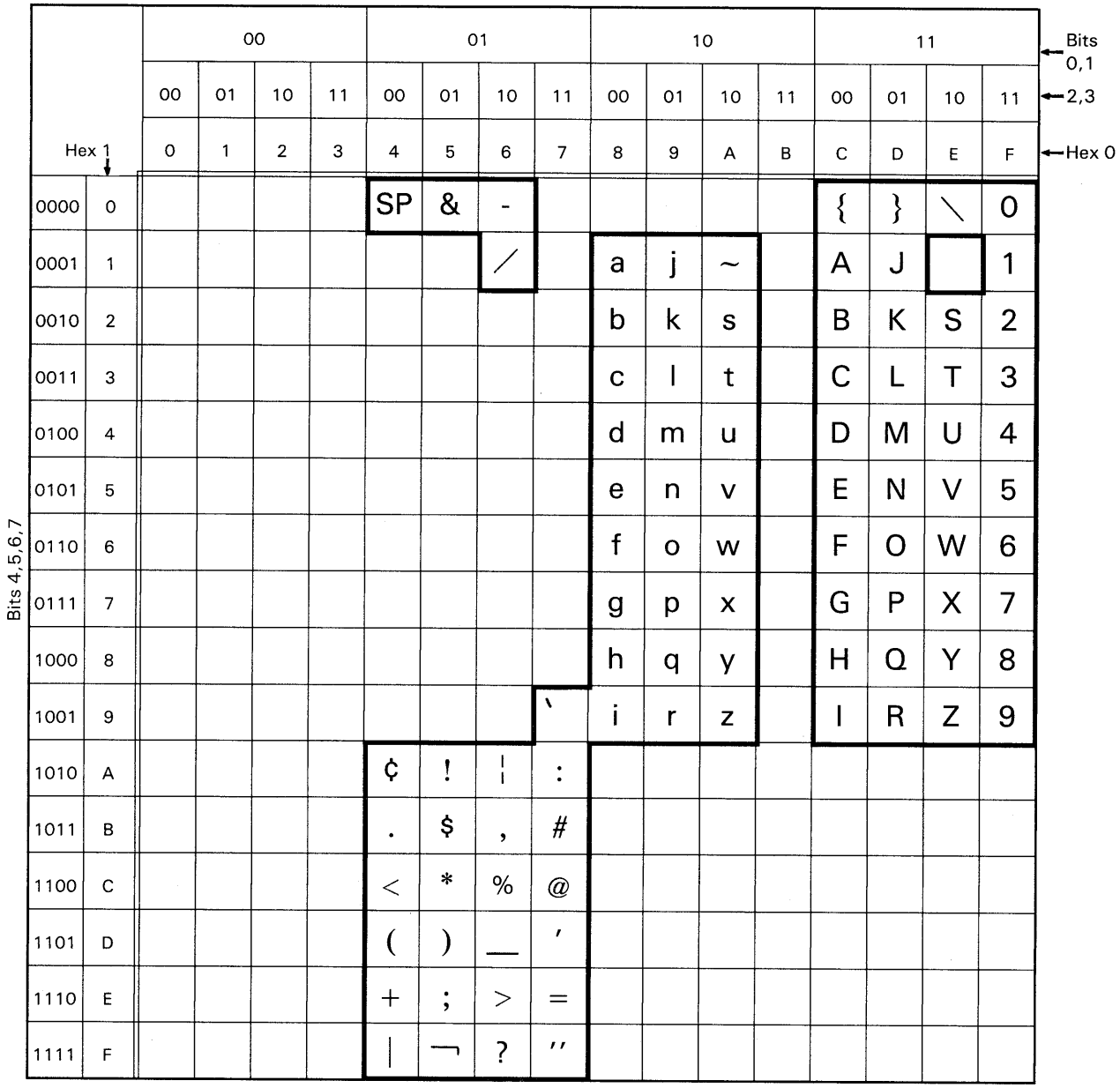
See the following figure for an illustration of 3270 APL mode I/O interface codes:

Figure E-15 on page E-18. I/O Interface Codes for 3270 APL Mode



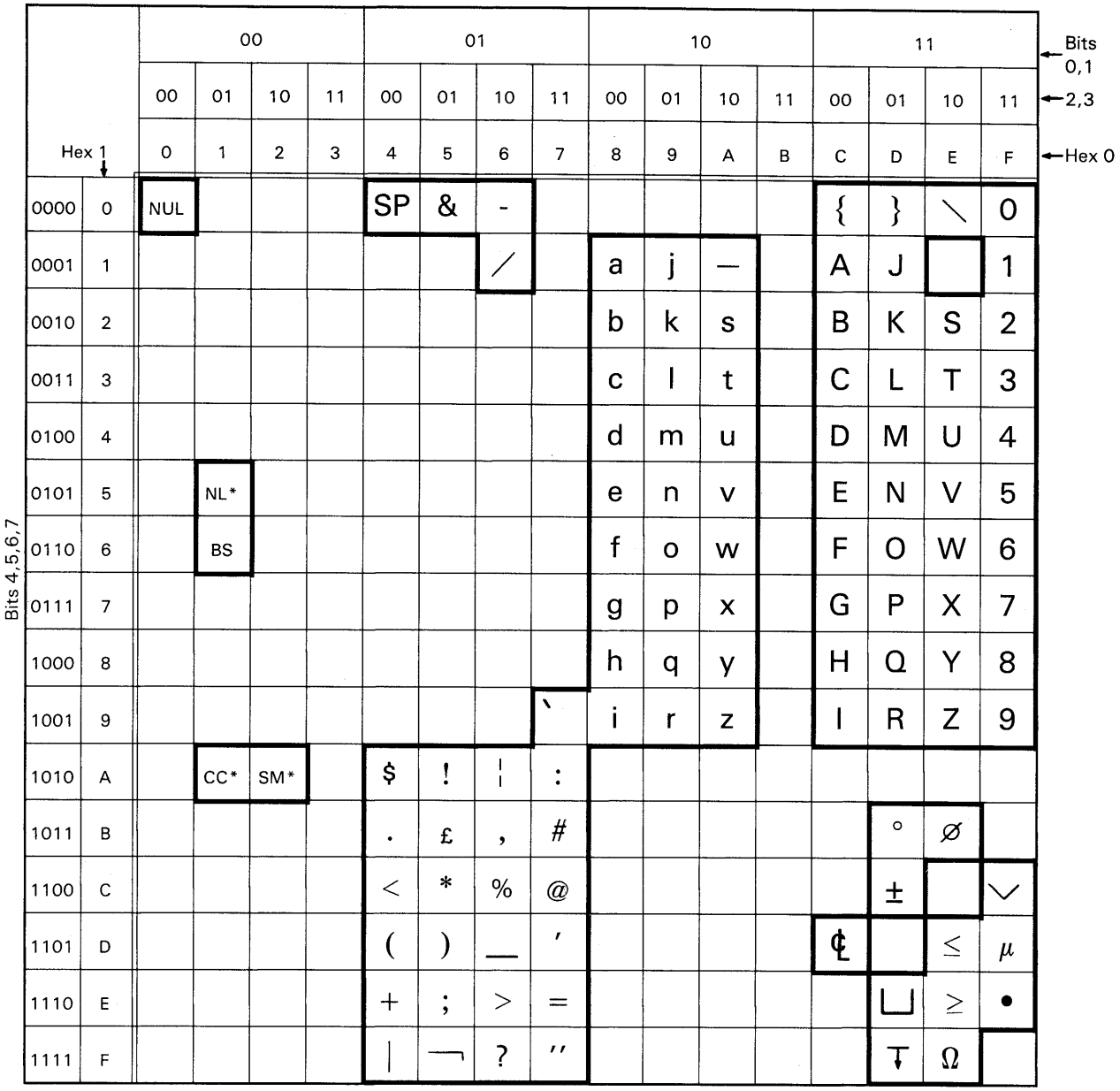
See notes under "Graphics Mode" on page E-1.

Figure E-1. English (US) I/O Interface Codes for Graphics Mode



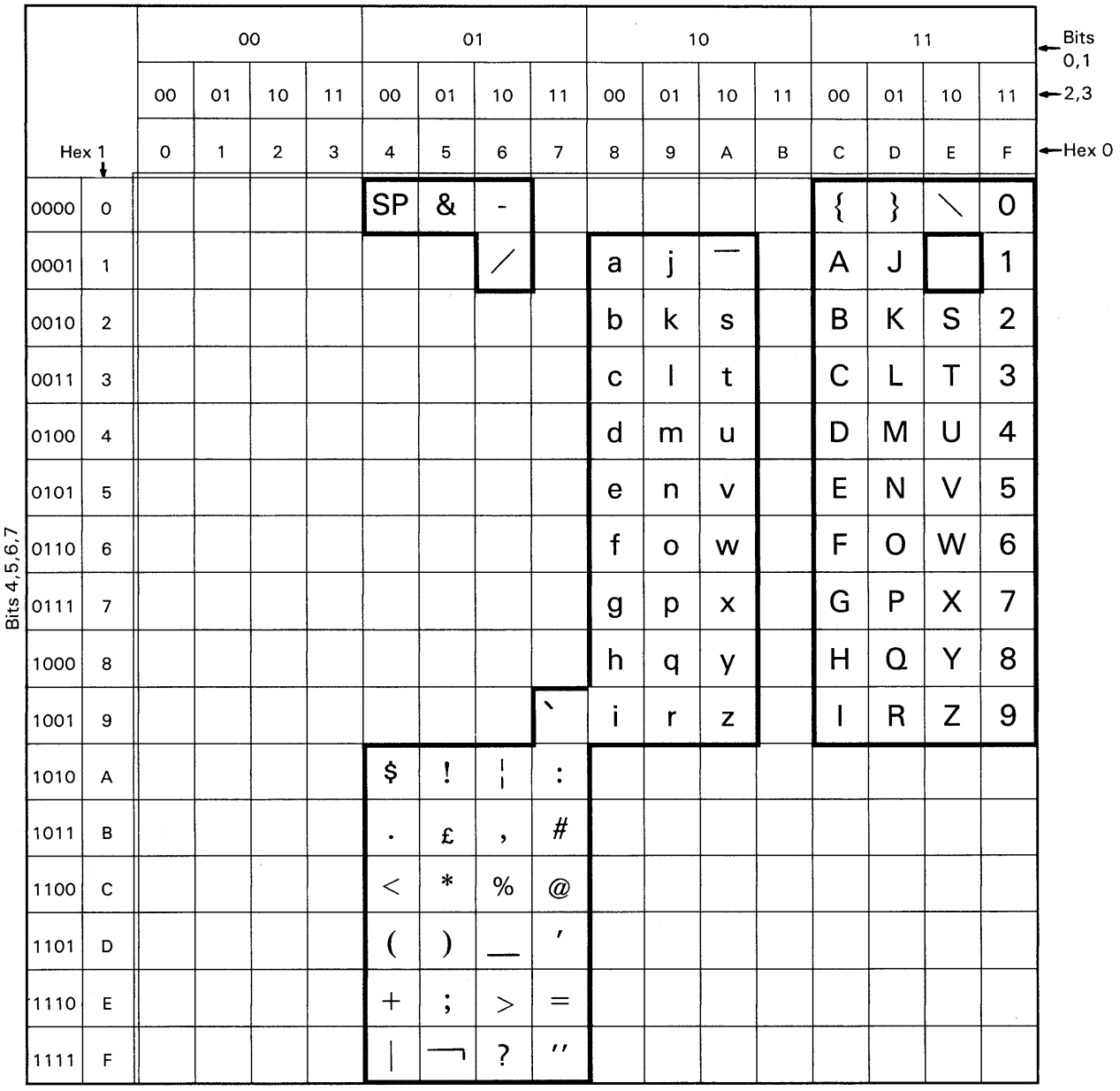
See notes under "3270 Mode" on page E-2.

Figure E-2. English (US) I/O Interface Codes for 3270 Mode



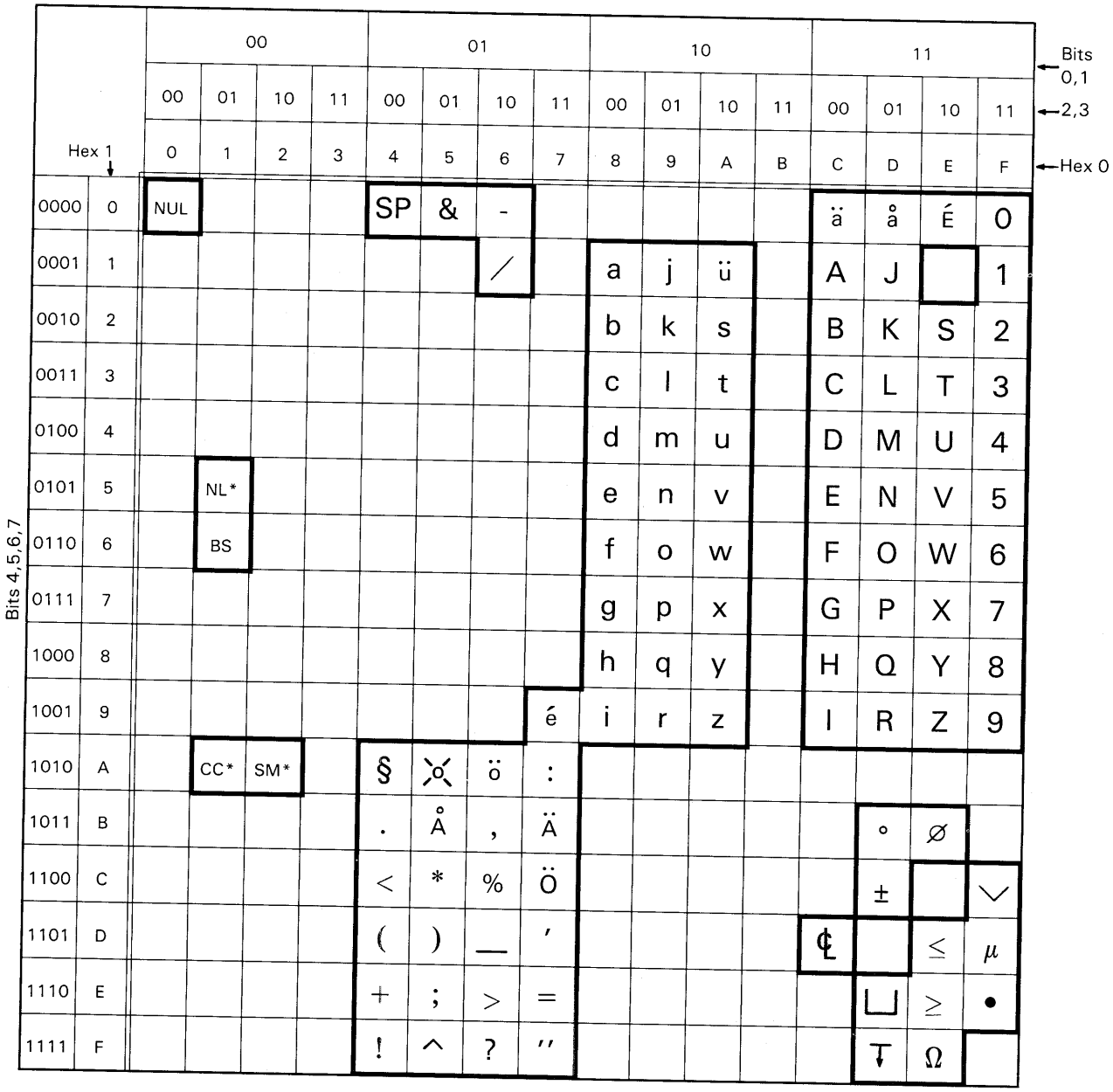
See notes under "Graphics Mode" on page E-1.

Figure E-3. English (UK) I/O Interface Codes for Graphics Mode



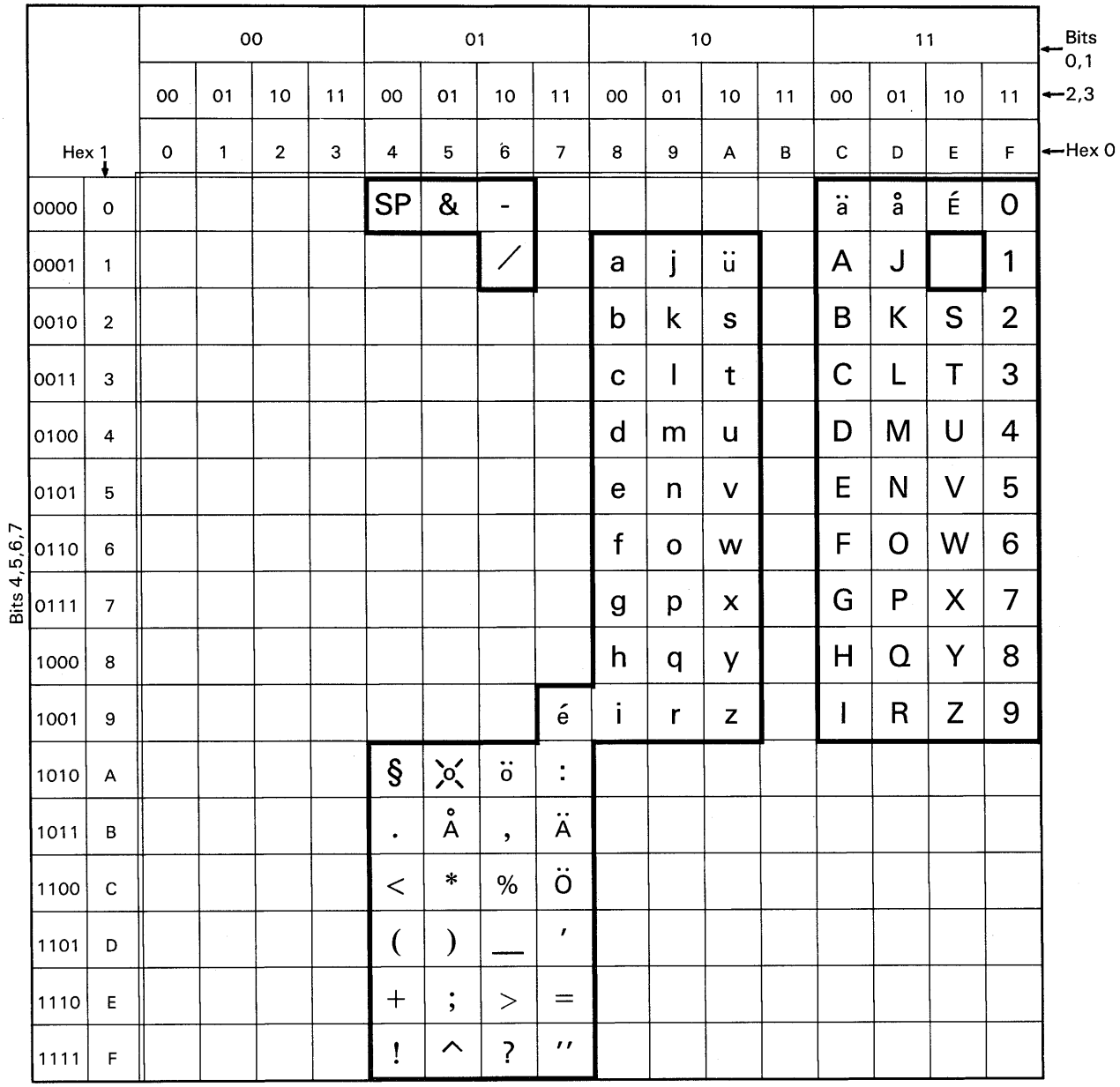
See notes under "3270 Mode" on page E-2.

Figure E-4. English (UK) I/O Interface Codes for 3270 Mode



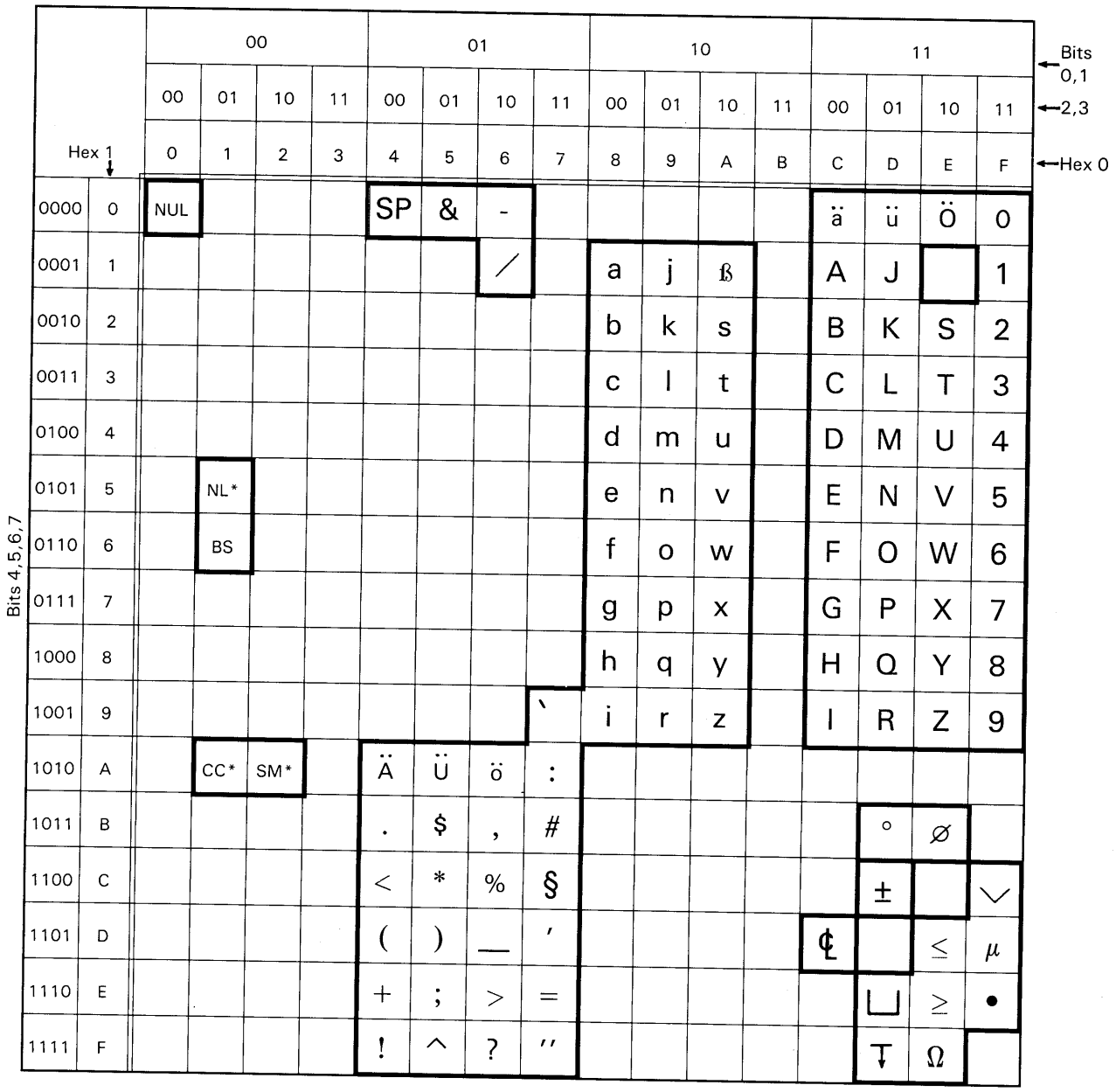
See notes under "Graphics Mode" on page E-1.

Figure E-5. Swedish I/O Interface Codes for Graphics Mode



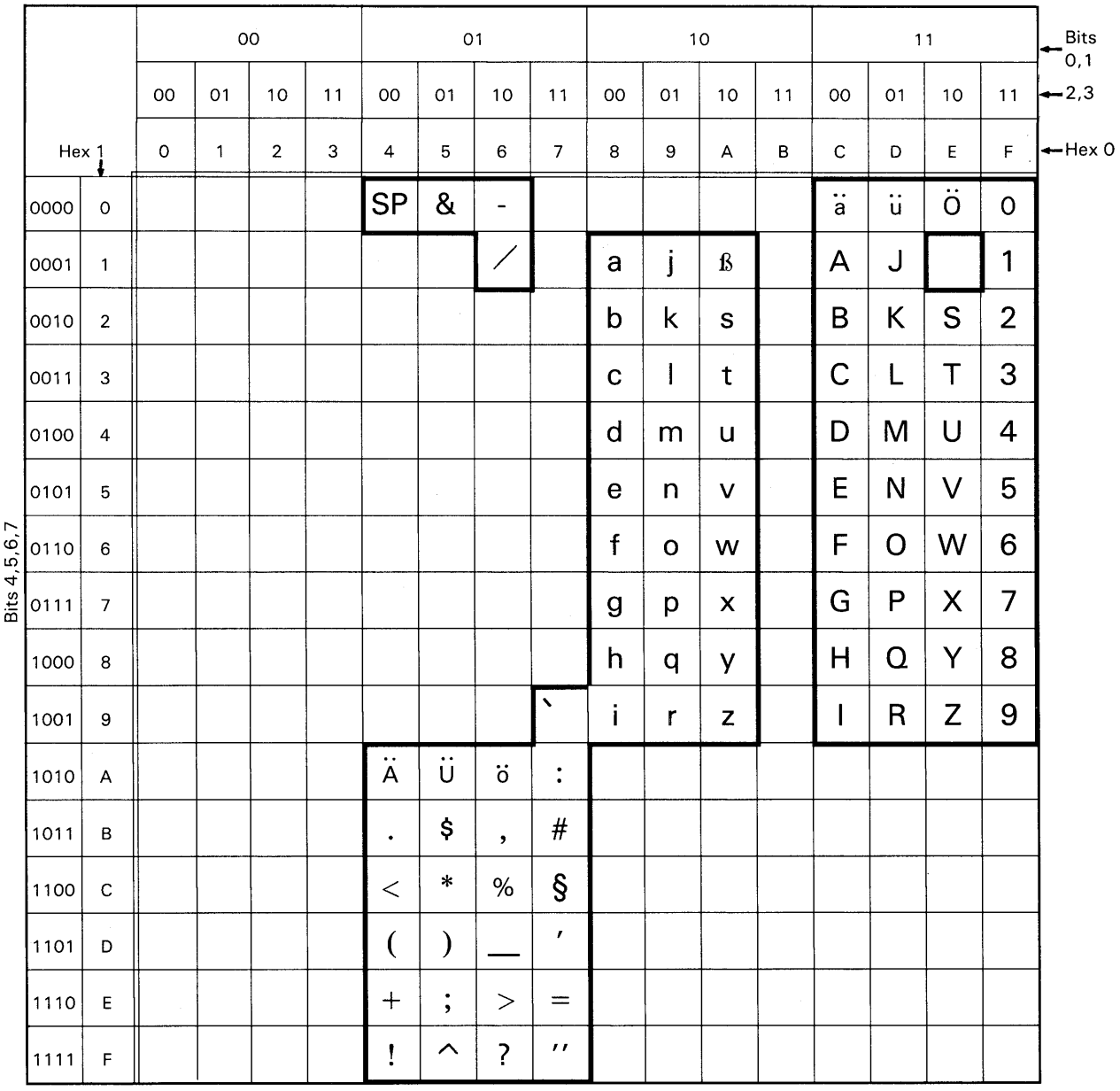
See notes under "3270 Mode" on page E-2.

Figure E-6. Swedish I/O Interface Codes for 3270 Mode



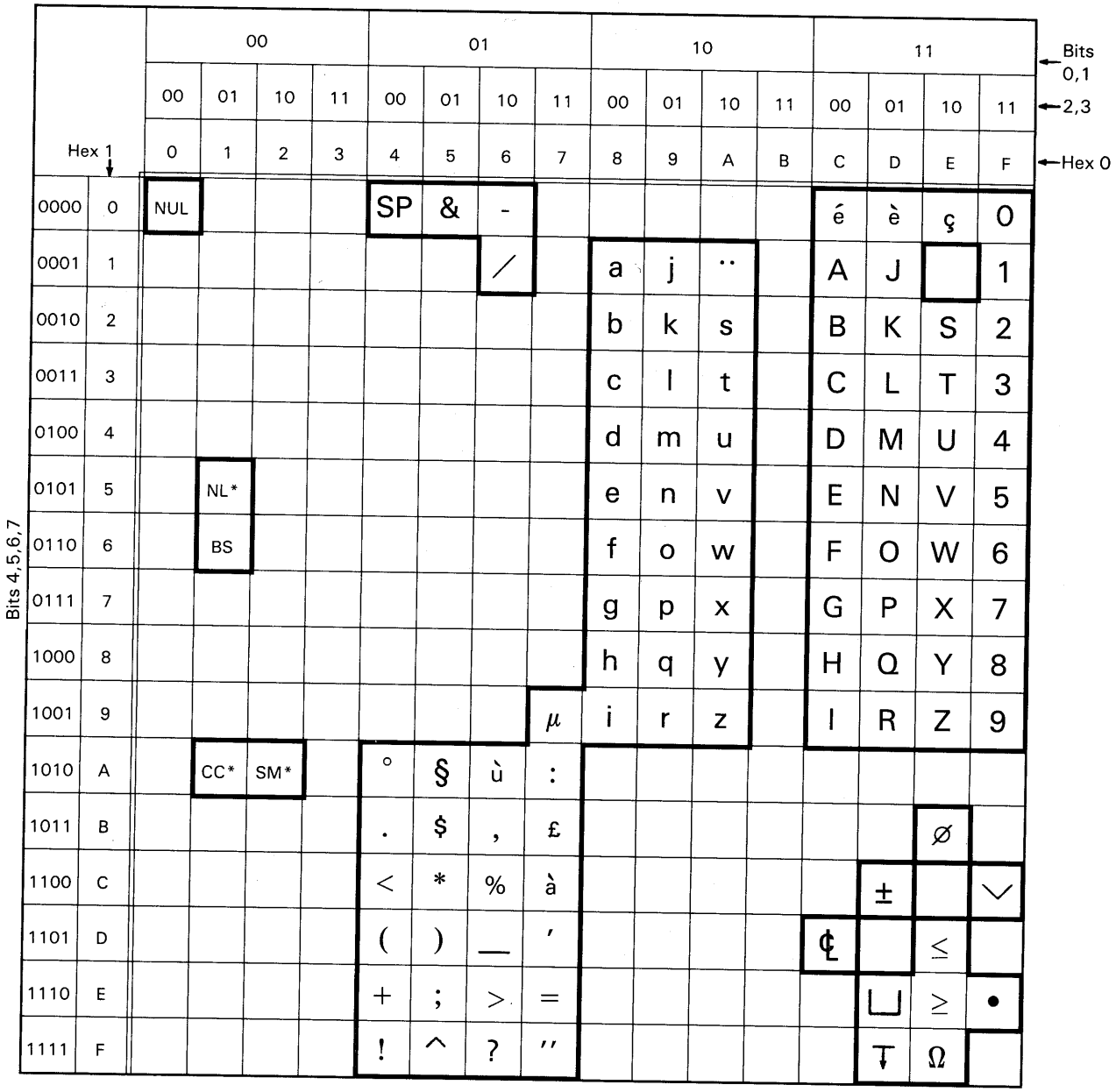
See notes under "Graphics Mode" on page E-1.

Figure E-7. German I/O Interface Codes for Graphics Mode



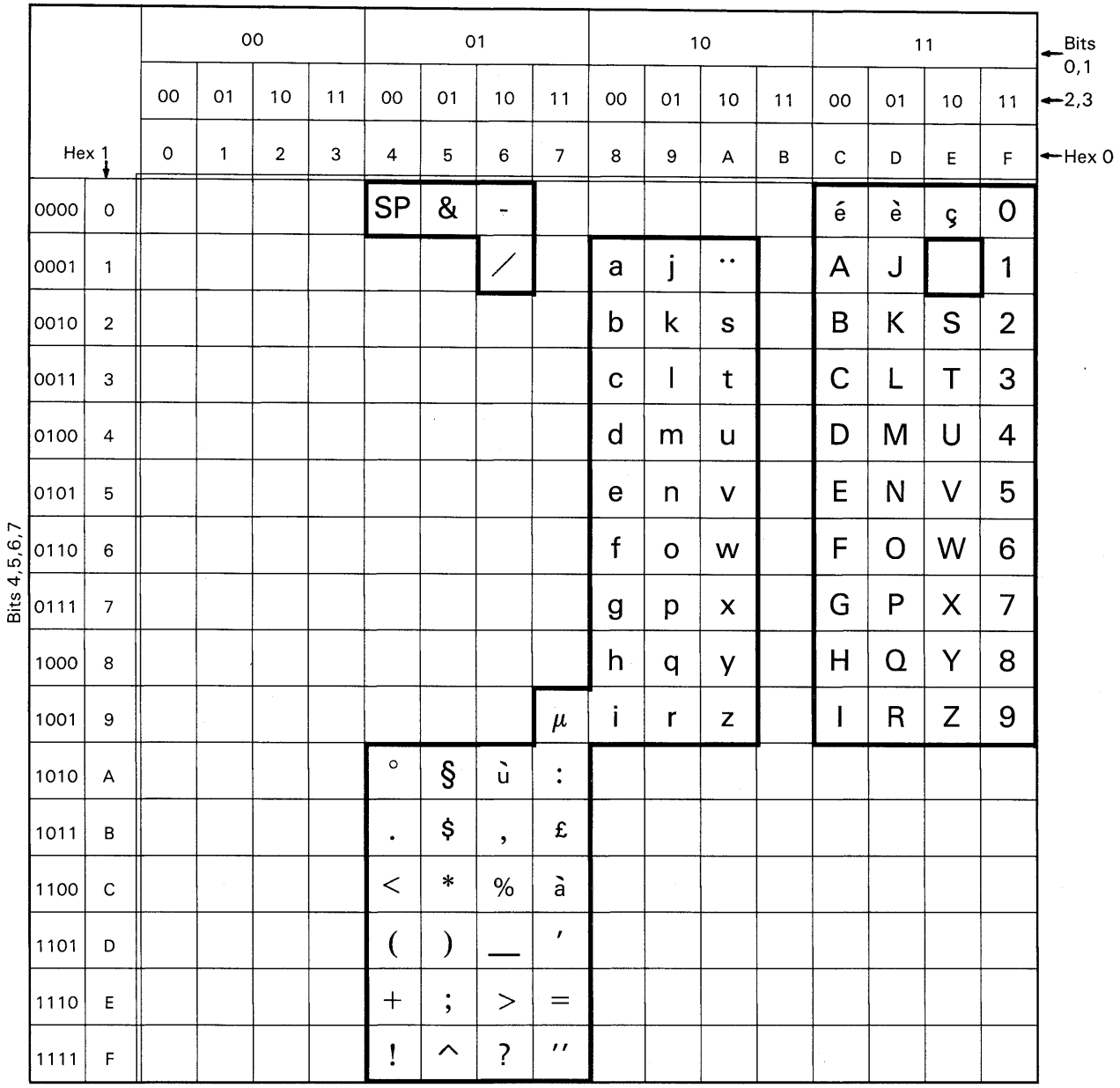
See notes under "3270 Mode" on page E-2.

Figure E-8. German I/O Interface Codes for 3270 Mode



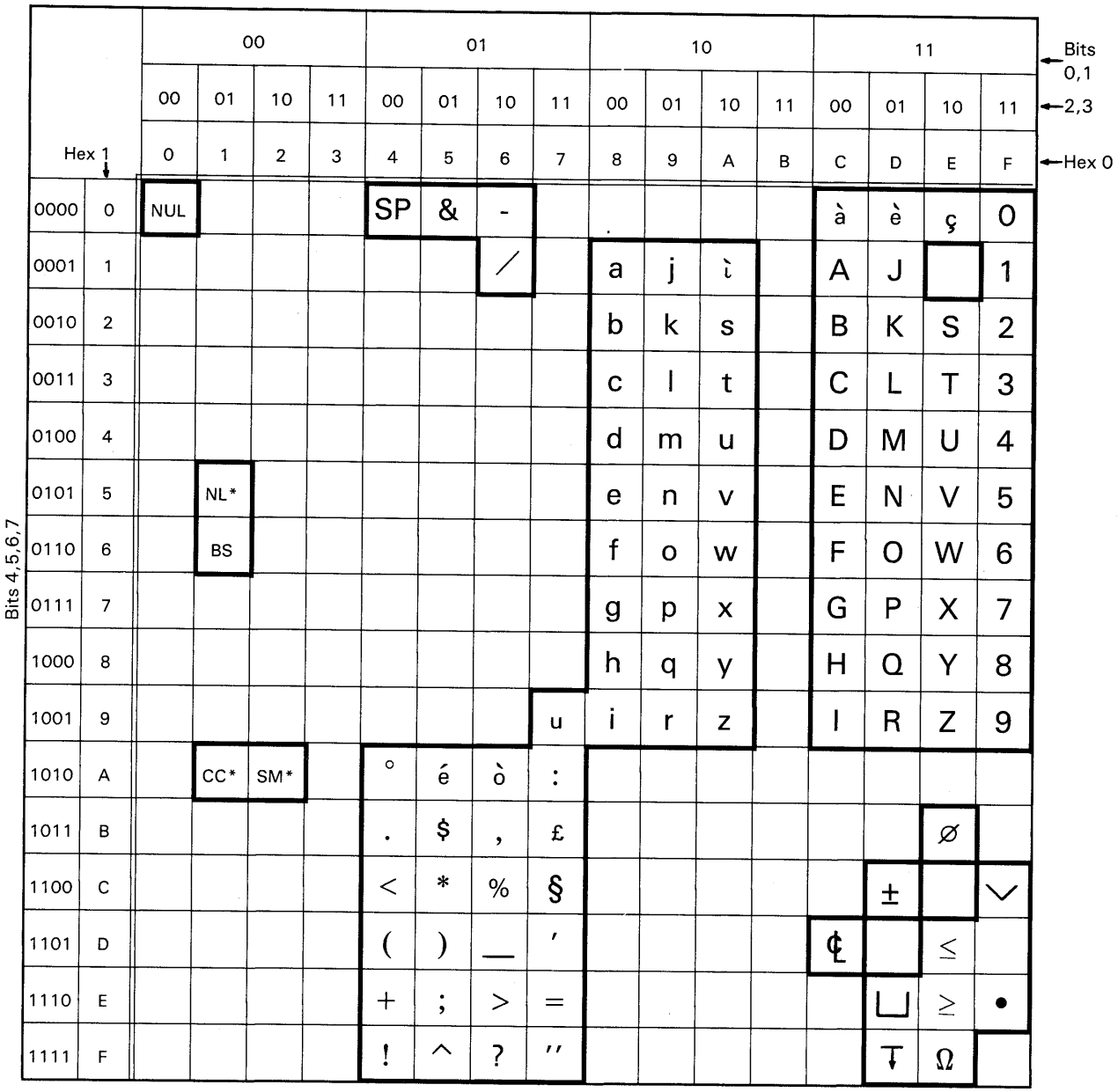
See notes under "Graphics Mode" on page E-1.

Figure E-9. French I/O Interface Codes for Graphics Mode



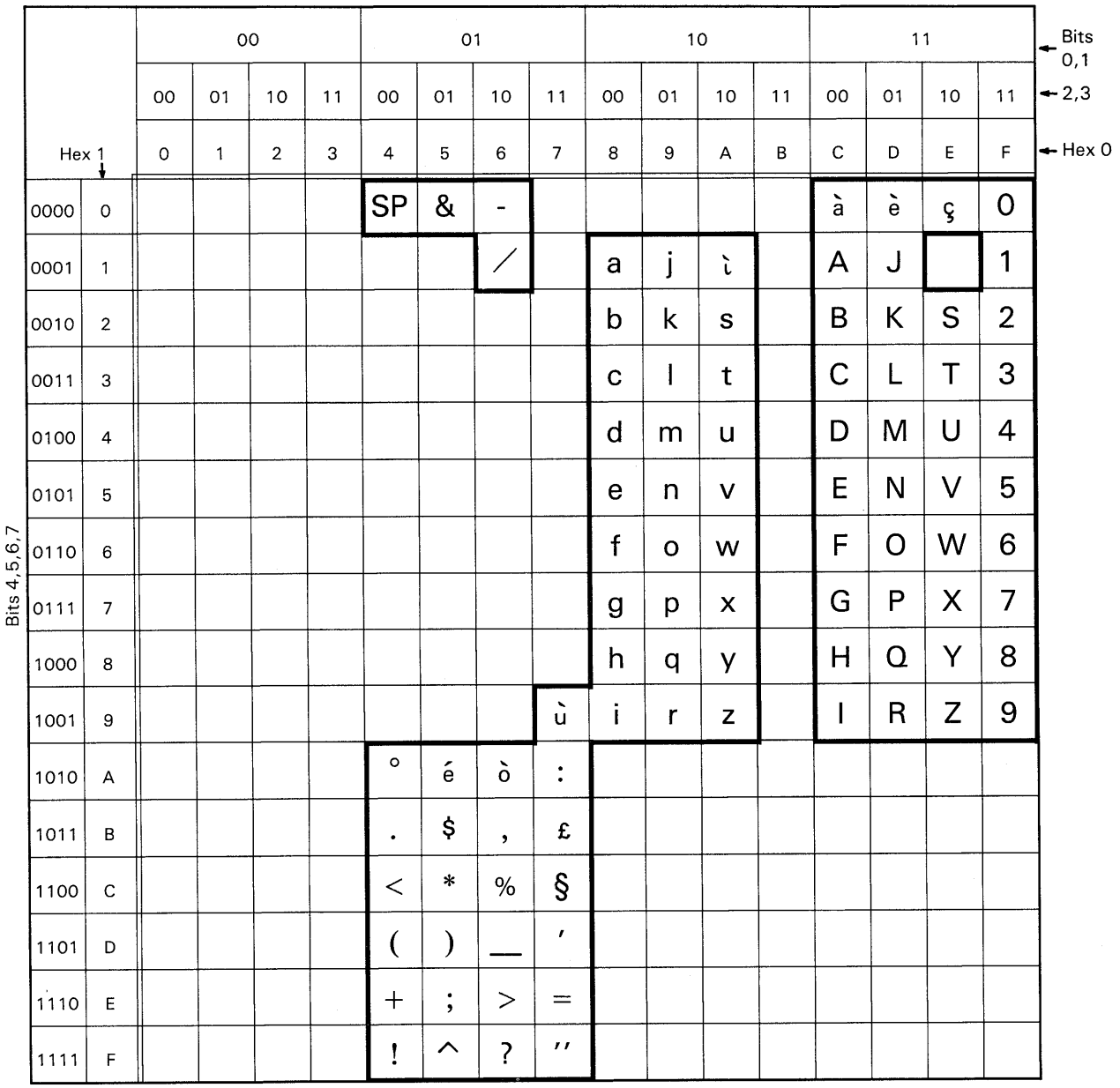
See notes under ''3270 Mode'' on page E-2.

Figure E-10. French I/O Interface Codes for 3270 Mode



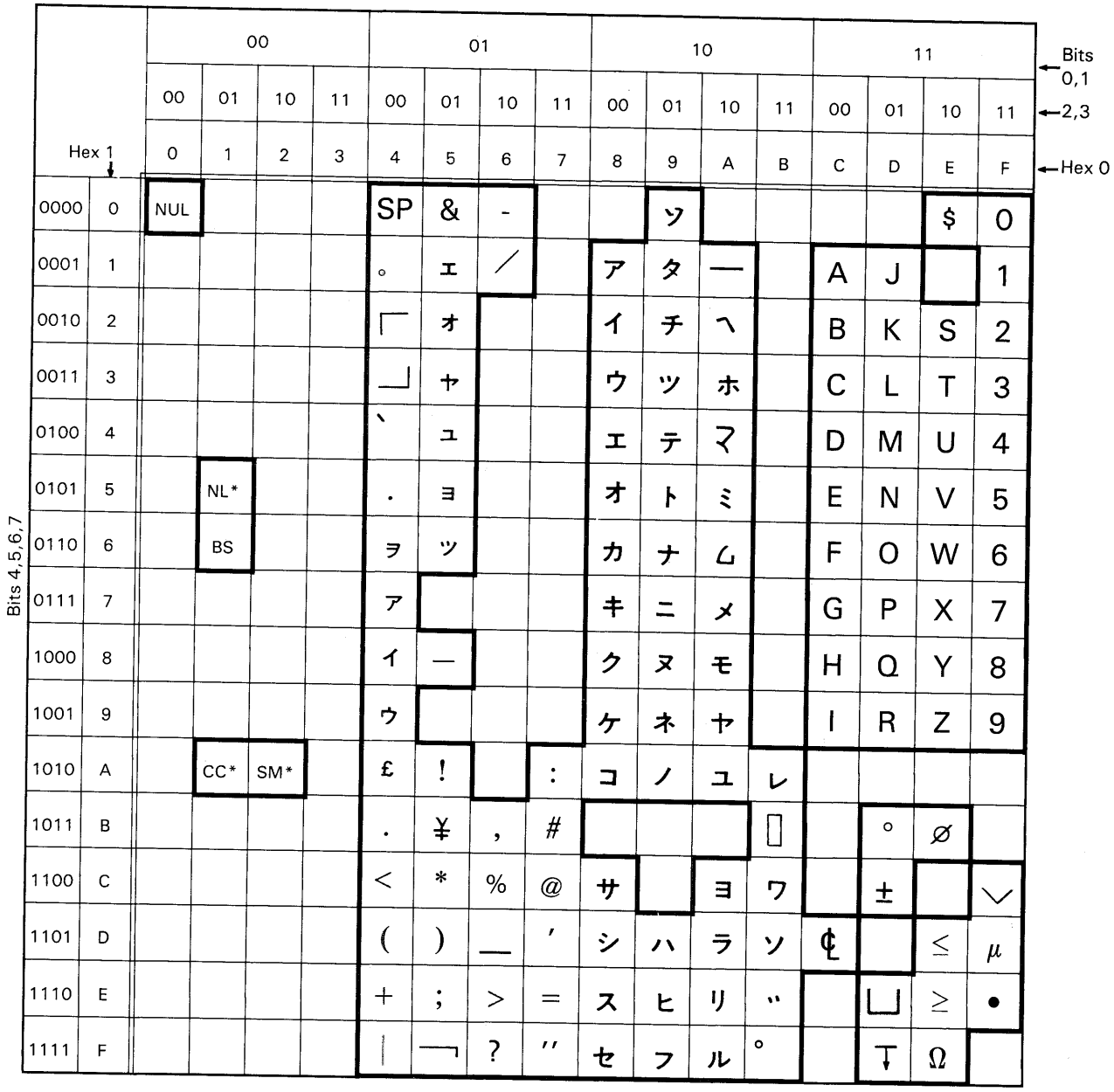
See notes under "Graphics Mode" on page E-1.

Figure E-11. Italian I/O Interface Codes for Graphics Mode



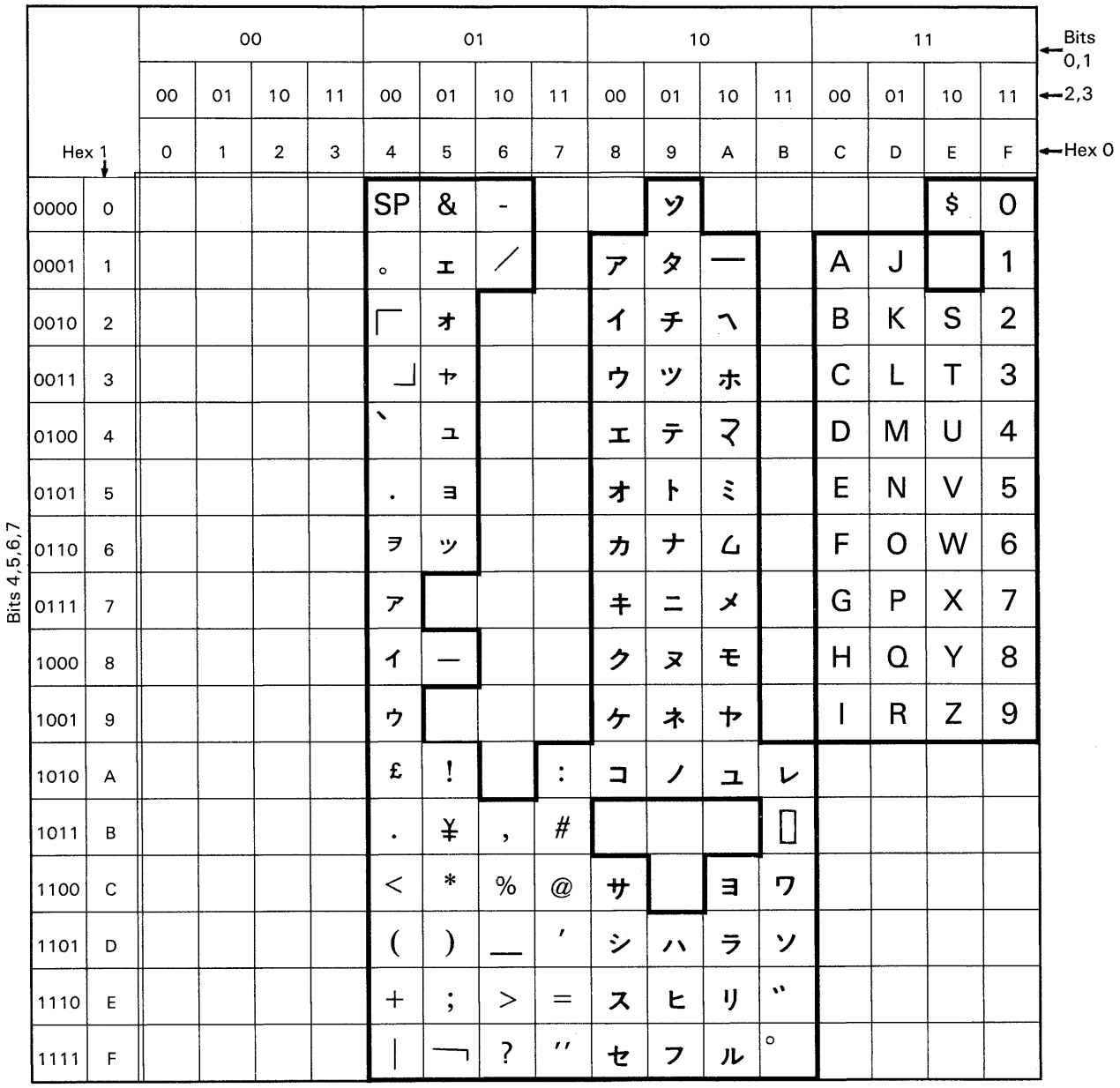
See notes under "3270 Mode" on page E-2.

Figure E-12. Italian I/O Interface Codes for 3270 Mode



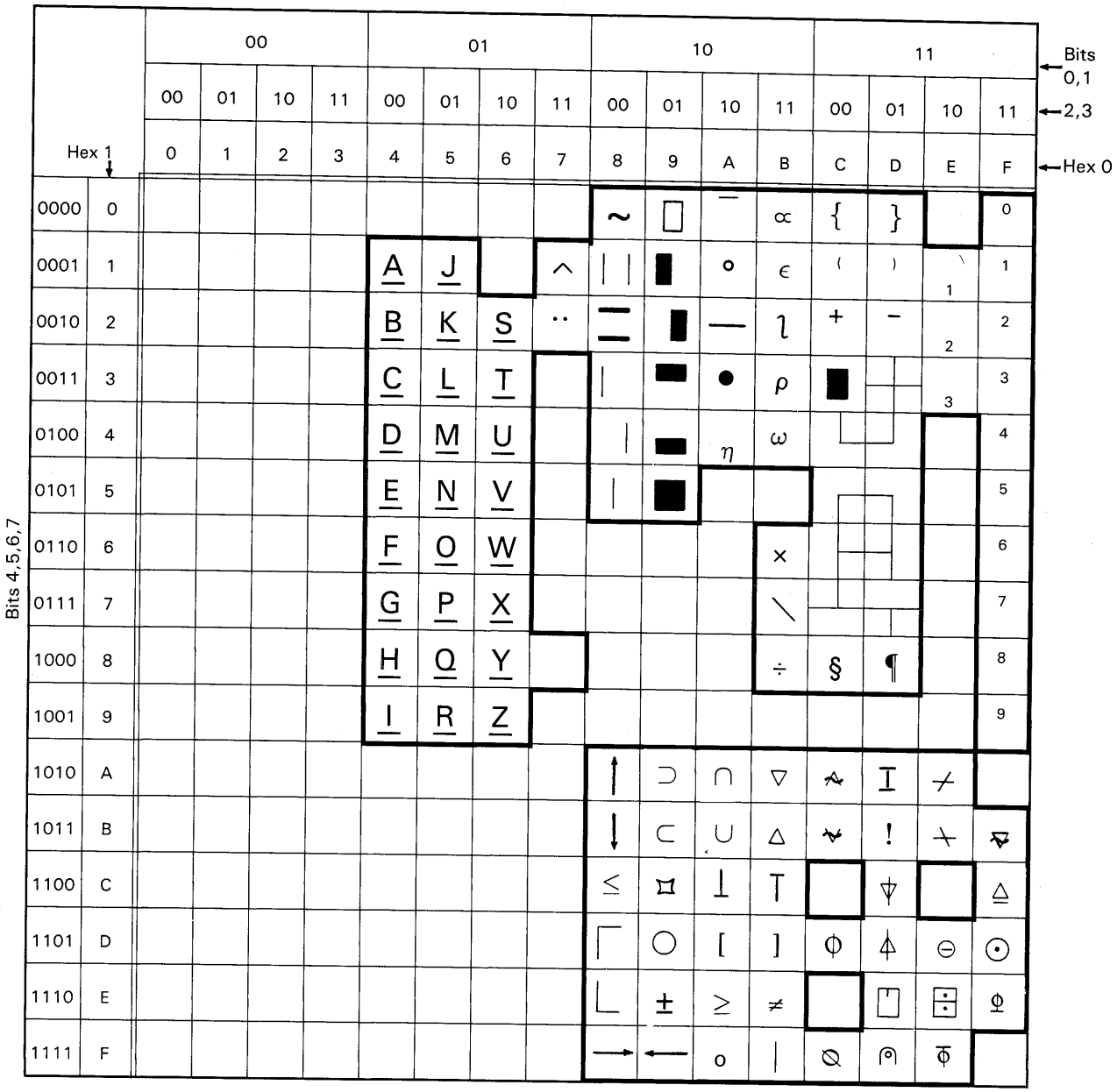
See notes under "Graphics Mode" on page E-1.

Figure E-13. Katakana I/O Interface Codes for Graphics Mode



See notes under "3270 Mode" on page E-2.

Figure E-14. Katakana I/O Interface Codes for 3270 Mode



See notes under "3270 APL Mode" on page E-2.

Figure E-15. I/O Interface Codes for 3270 APL Mode

Index

D

R

X

3

5

IBM 5080 Graphics System
Principles of Operation

Order No. GA23-0134-0

READER'S COMMENT FORM

This manual is part of a library that serves as a reference source for systems analysts, programmers, and operators of IBM systems. You may use this form to communicate your comments about this publication, its organization, or subject matter, with the understanding that IBM may use or distribute whatever information you supply in any way it believes appropriate without incurring any obligation to you. Your comments will be sent to the author's department for whatever review and action, if any, are deemed appropriate.

Note: *Copies of IBM publications are not stocked at the location to which this form is addressed. Please direct any requests for copies of publications, or for assistance in using your IBM system, to your IBM representative or to the IBM branch office serving your locality.*

How did you use this publication?

[ ] As an Introduction
[ ] As a text (student)
[ ] As a reference manual
[ ] As a text (instructor)
[ ] For another purpose (explain) _____

_____

Is there anything you especially like or dislike about the organization, presentation, or writing in this manual? Helpful comments include general usefulness of the book; possible additions, deletions, and clarifications; specific errors and omissions.

Page Number: Comment:

What is your occupation? _____

Newsletter number of latest Technical Newsletter (if any) concerning this publication: _____

If you wish a reply, give your name and address: _____

_____

_____

IBM branch office serving you _____

Thank you for your cooperation. No postage stamp necessary if mailed in the U.S.A. (Elsewhere, an IBM office or representative will be happy to forward your comments or you may mail directly to the address in the Edition Notice on the back of the title page.)

NOTE: STAPLES CAN CAUSE PROBLEMS WITH AUTOMATIC MAIL SORTING EQUIPMENT.
PLEASE USE PRESSURE SENSITIVE OR OTHER GUMMED TAPE TO SEAL THIS FORM.

Reader's Comment Form

Fold and Tape **Please Do Not Staple** Fold and Tape

NO POSTAGE NECESSARY IF MAILED IN THE UNITED STATES

BUSINESS REPLY MAIL

FIRST CLASS PERMIT NO. 40 ARMONK, N.Y.

POSTAGE WILL BE PAID BY ADDRESSEE:

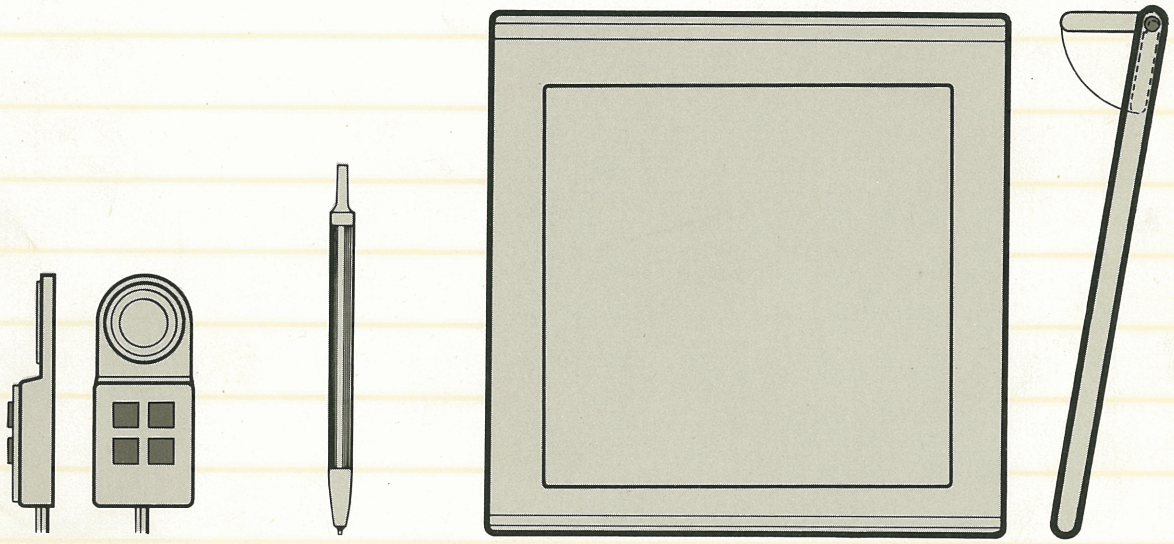
International Business Machines Corporation
Department 52Q
Neighborhood Road
Kingston, New York 12401

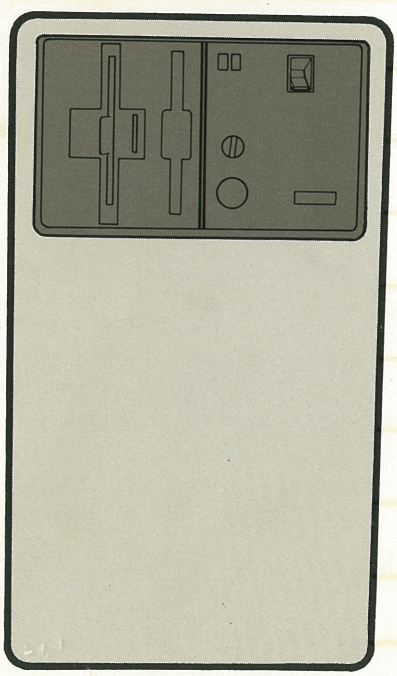
Fold and Tape **Please Do Not Staple** Fold and Tape

IBM®

PRINTED IN U.S.A. GA23-0134-0

GA23-0134-0
Printed in U.S.A.





GA23-0134-0