GA33-3037-2
File No. S370-06

IBM 3250 Graphics Display System

Systems

Component Description

IBM

GA33-3037-2
File No. S370-06

IBM 3250 Graphics Display System

Systems

Component Description

IBM

Federal Communications Commission (FCC) Statement

Warning: This equipment generates, uses, and can radiate radio frequency energy and if not installed and used in accordance with the instruction manual, may cause interference to radio communications. It has been tested and found to comply with the limits for a Class A computing device pursuant to Subpart J of Part 15 of FCC Rules, which are designed to provide reasonable protection against such interference when operated in a commercial environment. Operation of this equipment in a residential area is likely to cause interference in which case the user at his own expense will be required to take whatever measures may be required to correct the interference.

In addition to the above FCC statement, the user should be aware that:

- The statement only applies to IBM 3251 Display Stations Model 1, IBM 3255 Display Control Units Model 1 and Model 2, and IBM 3258 Channel Control Units used in the United States.
- The phrase "instruction manual" means:
  For installation information:
  - *IBM 3250 Graphics Display System: Installation Manual - Physical Planning*, GA33-3036
  - *IBM 3251 Display Station IBM 3255 Display Control: Maintenance Information*, SY33-0055
  - *IBM 3251 Display Station/IBM 3255 Display Control Unit Model 2*, SY33-0109
  - *IBM 3258 Control Unit: Maintenance Information*, SY33-0054

  For user information:
  - *An Introduction to the IBM 3250 Graphics Display System*, GA33-3035
  - *IBM 3250 Graphics Display System: Component Description*, GA33-3037

Second Edition, June 1983

This major revision obsoletes GA33-3037-1, and Technical Newsletters GN33-3066, GN33-3070, GN33-3100, and GN33-3141. The revision incorporates changes brought about by the introduction of the IBM 3255 Display Control Unit Model 2. Changes or additions to the text are indicated by a vertical line to the left of the change.

Changes are periodically made to the information herein; before using this publication in connection with the operation of the IBM systems or equipment, refer to the latest *IBM System/370 and 4300 Processors Bibliography*, GC20-0001, for the editions that are applicable and current.

References in this publication to IBM products, programs, or services do not imply that IBM intends to make these available in all countries in which IBM operates. Any reference to an IBM program product in this publication is not intended to state or imply that only IBM's program product may be used. Any functionally equivalent program may be used instead.

Publications are not stocked at the addresses given below. Requests for IBM publications should be made to your IBM representative or to the IBM branch office serving your locality.

A form for readers' comments is provided at the back of this publication. If the form has been removed, comments may be addressed to either:

- International Business Machines Corporation, Department 812H, 1133 Westchester Avenue, White Plains, New York 10604

or

- IBM United Kingdom Laboratories Limited, Publications, Mail Point 95, Hursley Park, Winchester, Hampshire, England, SO21 2JN

IBM may use or distribute whatever information you supply in any way it believes appropriate without incurring any obligation to you.

Preface

This publication contains reference information for users of the IBM 3250 Graphics Display System. The information is primarily intended to assist the user who is writing programs for execution in the 3250 Graphics Display System, but is also of interest to data processing personnel, operators, and engineers who require an understanding of the functions provided by the 3250 system.

The publication provides an introduction to the functional components of the 3250 Graphics Display System, which comprise:

- IBM 3258 Channel Control Unit
- IBM 3255 Display Control Unit Model 1 or Model 2
- IBM 3251 Display Station Model 1.

It also describes the principles of operation of the system, and communication with the host system. The buffer orders recognized by the 3250 are defined, and examples show how these orders can be used in a buffer program. Operator controls at the 3258 Channel Control Unit, the 3255 Display Control Unit, and the 3251 Display Station are described.

This publication contains the following chapters and appendixes:

- Chapter 1 introduces the machine types and functional components.
- Chapter 2 describes the facilities available to the programmer.
- Chapter 3 defines the buffer orders that are used in the buffer program.
- Chapter 4 defines the channel commands that control the transfer of data between the 3250 system and the host system.
- Chapter 5 contains an example of a buffer program and describes how the operator is able to interact with the programming. This chapter also contains (1) descriptions of some programming techniques, and (2) the figures needed to calculate the total execution time of a buffer program.
- Chapter 6 describes the controls used by the operator.
- Appendixes A and B contain hexadecimal values for, respectively, absolute and incremental coordinates.
- Appendix C contains EBCDIC charts for the supported character sets.
- Appendixes D and E summarize, respectively, the channel commands and buffer orders recognized by the 3250 system.
- Appendix F summarizes the meaning of various combinations of status and sense bits presented to the host system.

Related Publications

The IBM publications related to the IBM 3250 Graphics Display System are listed in the bibliography.

Contents

Figures

Chapter 1. Introduction

The IBM 3250 Graphics Display System provides interactive graphics capabilities for its host system, the IBM System/370 or the IBM 303X, 308X, 4331, and 4341 Processors. Graphic and alphanumeric information is displayed on a cathode-ray tube. Two types of keyboard and a light pen may be installed to allow the operator to interact with the displayed image and the host system, and to update stored data.

Units of the 3250 Graphics Display System are:

- *IBM 3251 Display Station Model 1:* A terminal that gives a directed-beam, buffer-refreshed display and may be provided with an alphanumeric keyboard, a program function keyboard, and a light pen; the keyboards and light pen are special features. A fully-configured 3250 system, based on a single 3258 Channel Control Unit, may contain up to sixteen 3251 Display Stations Model 1.

- *IBM 3255 Display Control Unit Model 1 or Model 2:* The 3255 Display Control Unit controls one or more 3251 Model 1 Display Stations and contains the logic and storage needed to:

  - Store and execute a buffer program for each attached 3251.
  - Generate beam deflection and intensity signals to produce the required image on the screen of each attached 3251 Model 1.
  - Provide keyboard and light pen interfaces for each attached 3251 Model 1.
  - Provide an interface between the buffer programs and, through the IBM 3258 Channel Control Unit, the host system.

- *IBM 3258 Channel Control Unit*: A channel-attached control unit that provides an interface between the connected 3255 Display Control Units and the input/output (I/O) channel of the host system. A 3258 has four serial link outlets for the attachment of 3255s. Up to four 3255s may be attached to each serial link; thus a configuration of sixteen 3251 Model 1s may have a separate 3255 for each 3251.

Figure 1-1 illustrates the configuration of units in the 3250 system. Further introductory information and the details of attachment to a host system are contained in *An Introduction to the IBM 3250 Graphics Display System*, GA33-3035.

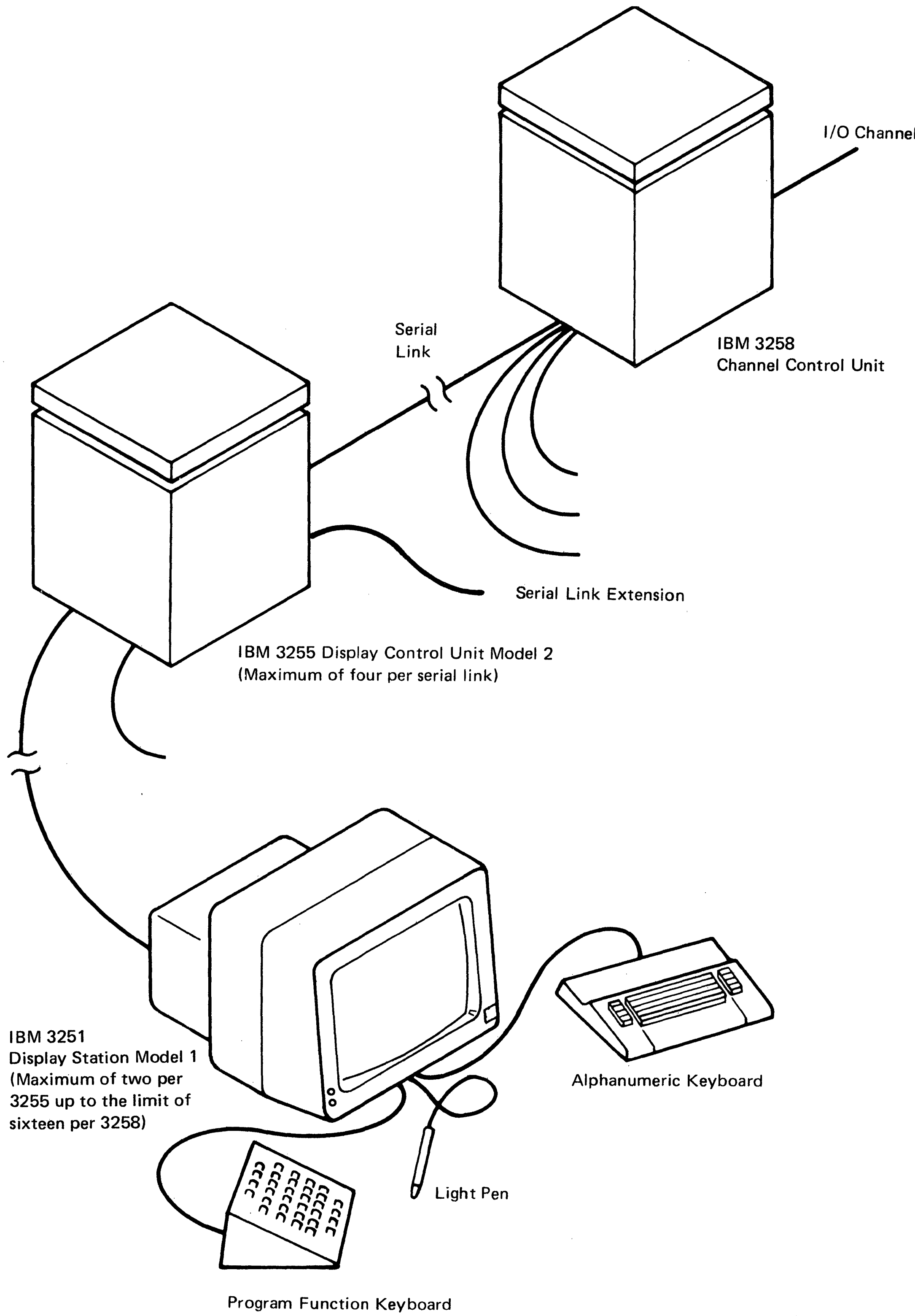

Figure 1-1. Simplified Configuration of a 3250 System

Graphic Display

The image displayed at a 3251 Display Station may contain a mixture of points, lines (vectors), and alphanumeric characters (Figure 1-2). Curves are displayed by plotting a series of points or short interconnected vectors.

The image is presented on the screen in a 305-millimeter (12-inch) square image area containing 1024 x 1024 (that is, 1 048 576) addressable points. Each of these points may be displayed as a point or addressed as a line end-point or a character-center point.

A moving electron beam in the cathode-ray tube produces the displayed image; the buffer program controls this movement and varies the intensity of the beam to produce visible points, lines, and alphanumeric characters. Each execution of the buffer program produces one transient image. By repeated execution of the buffer program, the image is regenerated to maintain it on the screen.

Each displayed point, line, or character is the result of the buffer program processing data under the control of a graphic-mode or character-mode order. The buffer program may be written to regard groups of points, lines, or characters as entities. Each entity may be assigned different attributes by the programmer. (The attributes are: blinking or steady, type of line, and intensity level relative to other components of the image.) Similarly, the programmer may set different light-pen modes for entities to enable or inhibit use of the light pen on an entity, and to determine the response of the program when the operator uses the light pen to select an entity.

The directed-beam display of the 3250 system allows components of the image to overlap. Thus, a character string may be superimposed on a graphic entity without any interaction between the corresponding data in the buffer program.

Eight programmable intensity levels are provided, from blank (intensity level 0) to maximum (intensity level 7); these levels allow the programmer to define the relative intensity level of components of the displayed image. A brightness control on the 3251 allows the operator to adjust the brightness of the whole displayed image.

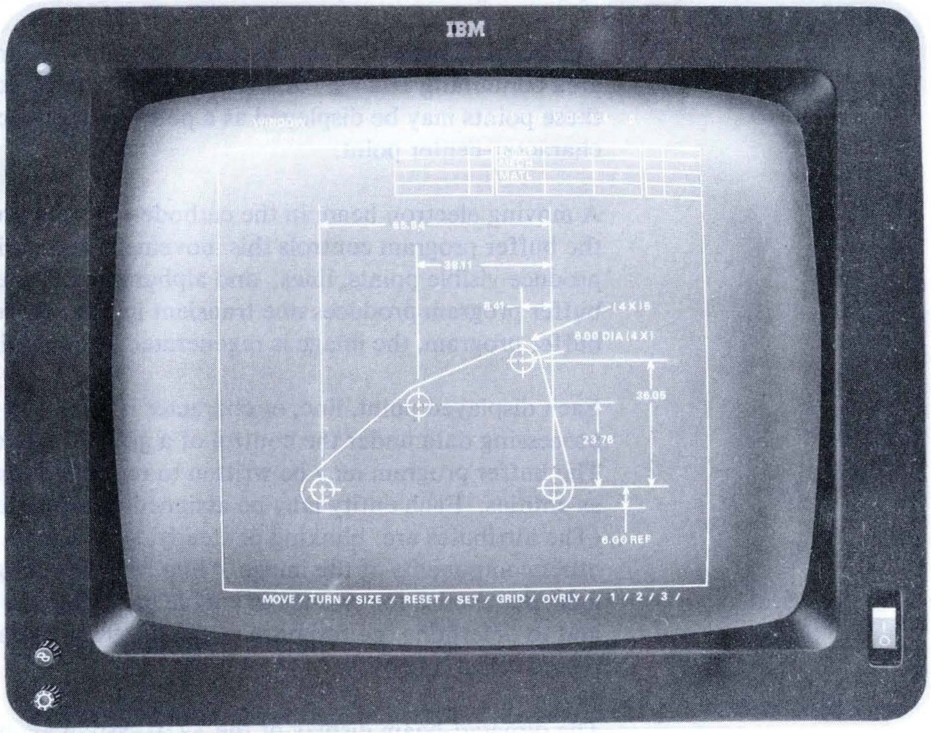

Figure 1-2. Example of a Graphic Display

Buffer Program

A buffer program is executed in the 3255 to generate the image displayed at a 3251 and to allow the operator of that 3251 to use the special keyboards and light pen. Where two or more 3251s are supported by one 3255, each 3251 has a buffer program.

The 3255 Display Control Unit Model 2 has two display buffers that operate as separate entities. A buffer program contained in one buffer cannot branch to, or access data in, the second buffer. The allocation of buffers to particular 3251s is determined by the number of attached 3251s and their device addresses:

One 3251 Model 1. Only one buffer is used.

Two 3251 Model 1s. The 3251 with the low address uses one buffer, and the 3251 with the high address uses the other buffer. Alternatively, if it is envisaged that an additional 3251 is to be added at a future date, the two original 3251s can share the same buffer leaving the other buffer unused.

Three 3251 Model 1s. The 3251s with the low and medium addresses share one buffer, and the 3251 with the high address has exclusive use of the other buffer.

Buffer programs consist of orders interleaved with data. An application program in the host system generates the buffer program. Using the I/O channel, the application program loads the buffer program into the 3255 display buffer and starts program execution.

The buffer orders supported by the 3250 system form four groups:

- Orders to display points and vectors
- Orders to display characters
- Orders to determine how the light pen interacts with the displayed image
- Orders to control the execution of the buffer program.

A detailed description of the buffer and its operation is given in Chapter 3.

Alphanumeric Keyboard

The alphanumeric keyboard of the 3251 Model 1 is a typewriter-like keyboard that can be used to compose or modify messages and annotations on the displayed image.

A cursor on the screen shows where an entered character will be displayed; when a character key is pressed, the character code replaces a data field in buffer storage and the entered character is displayed.

The alphanumeric keyboard provides keys for all characters of the supported character set. Other keys include:

- *END and CANCEL Keys*: When pressed, these raise an I/O interruption in the host system.

  Note: On a 3251 attached to a 3255 Model 2, the CANCEL key usage is changed to avoid accidental use of the key. The CANCEL key operates only when the CONTINUOUS key is first pressed and held down.

- *Cursor Control Keys*: The ADVANCE key advances the cursor one character space, the BACKSPACE key backspaces the cursor one character space, and the JUMP key jumps the cursor to the start of the next available alphanumeric field.

3255 Model 2 only. An 87-key alphanumeric keyboard with an integral numeric keypad and engineering symbols is available on the 3251 Model 1 attached to a 3255 Model 2 Display Control Unit. This keyboard is fully compatible with the standard 75-key keyboard with engineering symbols. The 3251 must be fitted with the Extended Keyboard Support specify feature.

For details of both types of alphanumeric keyboard, refer to Chapter 6.

Program Function Keyboard

The program function keyboard of the 3251 Model 1 provides direct communication between the operator and the application program in the host system. Pressing any one of the 32 keys causes an I/O interruption to be raised to the host system and supplies a code identifying the pressed key to the system.

The function allotted to particular keys depend upon the application program; for example, keys may be used to request other applications, where the choice of key indicates one of a group of applications.

Keys are not marked with a key number or function. The design of the program function keyboard allows the user to make an overlay that shows the function of individual keys. Because these functions are defined by the application program, each program may require a different overlay.

Each key is back-lit by a lamp. The application program can light or extinguish individual keys. This may be done to show which keys are valid for selection, to denote which key was the last one pressed, or to indicate the status of the program.

Light Pen

The light pen allows the operator to communicate with both the buffer program and the application program. The pen contains a switch (tip-switch) and a light-sensor; these can be used separately or together.

Pressing the tip of the pen against a hard surface, such as the display screen, closes the tip-switch; removing the light pen opens the switch. Pointing the light pen at a component of the displayed image allows the component to excite the light-sensor. To aid in the selection of a component of the display, the signal from the light-sensor may be fed back to increase the intensity of the image at the point viewed by the light pen. In the case of a 3255 Model 2, the first element seen by the light pen may be brightened if the single element intensification function is enabled.

Components with a programmed intensity level of less than 5 cannot be selected with the light pen. If the brightness control on the front of the 3251 is set too low, the electron beam may not be able to excite the light-sensor, even though the programmed intensity may be equal to or greater than 5.

If the light pen is enabled for detection when the buffer program generates a component of the image, then selecting that component with the light pen causes a light-pen detection to occur. The action initiated by a light-pen detection depends upon whether the light pen is currently enabled for an immediate interrupt or for a deferred response to a detection. Thus, selecting one component of the displayed image may cause a transfer of control in the buffer program, whereas selecting another component of the same image may raise an I/O interruption to the host system.

Using light-pen-mode orders in conjunction with control-mode orders provides considerable flexibility to the buffer program. Techniques such as light-pen dragging (that is, allowing a component of the image to follow the light pen across the screen) may be programmed in the 3255. This type of technique may be achieved without any interaction with the host system.

I/O Channel Attachment

Communications between the 3250 system and the application program in the host system is by way of an I/O channel. Channel operations are summarized in Chapter 4; for a full description of channel operations, refer to *IBM System/370 Principles of Operation*, GA22-7000.

The 3250 system can be attached to any of the following types of I/O channel:

- Byte-multiplexer channel of the IBM System/370 Models 115 and 125, operating in burst mode.
- Block-multiplexer channel (see note). Block multiplexing can occur between 3251s attached to different 3258s on the same channel, but not between 3251s attached to the same 3258.
- Selector channel.

Note: When attached to a block-multiplexer channel, shared UCWs should be used if the channel implementation allows block-multiplex operation with shared UCWs.

Chapter 2. Principles of Operation

The generation of an image and the use of the keyboards and light pen depend upon the execution of a buffer program in the 3255 Display Control Unit. This chapter describes the facilities available to the programmer, the principles of the programmable modes, and the I/O operations on the channel.

The execution of a buffer program produces a displayed image containing points, lines, and alphanumeric characters; each component of the image is an entity with, perhaps, its own attributes and light pen mode. In response to operator actions, the buffer program is able to modify itself, and thus change the image, and the application program in the host system is able to replace or modify the buffer program. The application program can also save the buffer program, or parts of it, in order to maintain a master file.

Facilities available to the programmer include:

- **Display Buffer:** A buffer storage of 32 768 bytes, in the 3255 Display Control Unit, loaded with buffer programs generated by application programs in the host system.

  Note: The 3255 Display Control Unit Model 2 has two separate buffers, each of 32 768 bytes, for program location.

- **Display:** An image area comprising 1024 x 1024 addressable points on the screen of the 3251 Display Station.

- **Graphic Mode:** A mode of operation whereby data fields in the buffer program direct the movement of the electron beam in the cathode-ray tube. *Point plotting* moves the beam to an addressed point, or series of points, with the option of displaying each point. *Vector plotting* generates a line to the addressed point, and between points in a series, with the option of displaying each line.

  A vector generator, operating under the control of graphic-mode (vector) orders, is capable of generating any one of four types of line between two addressed points in the image area, and in any one of eight programmable levels of intensity.

- **Character Mode:** A mode of operation whereby data fields in the buffer program generate alphanumeric characters in the displayed image.

  A character generator, operating under the control of character-mode orders, is capable of generating the supported character set (see Appendix C), in four sizes and two orientations.

- **Light-Pen Modes:** Modes, set by light-pen-mode orders, that control the sensitivity of the displayed data to selection by the light pen, and define the action taken in response to a light-pen detection.

- **Control Modes:** A group of orders capable of (1) transferring control in the buffer program, (2) loading and storing the attribute register (this is described under "Buffer Program" later in this chapter), (3) storing the X,Y coordinates of the current beam position, and (4) moving addresses and data from one field in the display buffer to another.

Display Buffer

Display Buffer in 3255 Model 1

The display buffer in the 3255 consists of 32 768 bytes (16 384 words) of storage. This storage contains a buffer program for each attached 3251. The buffer programs are loaded by application programs in the host system.

Each even-odd pair of display buffer bytes forms a word. Buffer orders and data fields are aligned on word boundaries and occupy an integral number of words; orders occupy 1, 2, or 3 words, and data fields occupy 1 or 2 words. Words are addressed by the even-byte address of an even-odd pair, from 0 (bytes 0 and 1) through 32 766 (bytes 32 766 and 32 767).

The 3255 forces addressing on a word boundary by ignoring the least significant bit in the buffer address register. This bit, however, is not forced to zero; if an odd address is set into the register, bit 15 (that is, bit 7 of the second byte of the address) will remain active until either the buffer program or the application program sets an even address.

Because bit 0 of the first byte of the address word is not needed for addresses from 0 through 32766, this bit is also ignored.

Display Buffers in 3255 Model 2

The same considerations addressed in the preceding paragraphs for a 3255 Model 1 with a single buffer, also apply to the 3255 Model 2 with two buffers.

Sharing a Display Buffer

If two 3251 Model 1s are attached to a 3255 and both 3251s are working, the display buffer holds two buffer programs. These programs are executed alternately by the 3255; decoding a Start Regeneration Timer (GSRT) order in the active program disconnects that program, and the 3255 executes the other program. Executing a GSRT order in the now active program returns the 3255 to the first program.

Note: If the executing buffer program does not contain a GSRT order, the 3255 is unable to switch to the buffer program for the other 3251. Thus, the other 3251 is locked out.

The 3255 keeps an address word for each attached 3251 for linking the 3251 to its own buffer program. This address word saves the address at which the 3251's buffer program stopped. When a GSRT order is decoded, the 3255 saves the current address, switches to the other 3251, and restarts execution at its saved address. The address word is not part of the display buffer.

3255 Model 2. A 3255 Display Control Unit Model 2 has the following characteristics:

- The two display buffers are separate entities, each being addressed from byte 0 through 32 767. Bit 0 of the first byte of the address word is reserved, and should be set to zero. That is, the address range in hexadecimal is '0000' through '7FFF'. A buffer program contained in one buffer cannot branch to, or access data in, the second buffer.

- The allocation of buffers to individual 3251s is determined by the number of 3251s that are attached to the 3255 Display Control Unit, and the device addresses of the 3251s:

  One 3251: Only one buffer is used.

  Two 3251s: The 3251 with the low address uses one buffer, and the 3251 with the high address uses the other buffer. Alternatively, if it is envisaged that an additional 3251 is to be added at a future date, the two 3251s can share the same buffer leaving the other buffer unused.

  Three 3251s: The 3251s with the low and medium addresses share one buffer. The 3251 with the high address has exclusive use of the other buffer.

Linking a 3251 to the Display Buffer

Initially, a Set Buffer Address Register and Start command is issued by the host system to establish a link between the selected 3251 and a buffer program in the display buffer. The 3251 is selected by the channel, the buffer address register is set to the value supplied by the channel, and execution of the buffer program starts. If any sequence of channel commands to a selected 3251 terminates execution of the buffer program, the sequence must be followed by a Set Buffer Address Register and Start command; this command restores the link between a selected 3251 and the display buffer.

Allocation of buffer storage is at the discretion of the host system application programmer; the 3255 does not place any constraints on the allocation of the display buffer. Although the two 3251s share the same display buffer, the host system addresses each 3251 as a unique device. Certain channel commands are only valid for execution if the buffer program for the addressed 3251 is stopped when the command is received. However, addressing one 3251 does not need the buffer program for the other 3251 to be stopped.

Buffer Program

The buffer programs consist of buffer orders, graphic data, and alphanumeric data. In its execution, a buffer program controls:

- Generation of the displayed image
- The sensitivity of the displayed image to a light-pen operation
- Data entry from the alphanumeric keyboard.

The buffer program also synchronizes the raising of I/O interruptions to the host system by the alphanumeric and program function keyboards.

Each execution of a buffer program produces one transient image. To be retained this image must be regenerated by repeated execution of the buffer program. The 3255 regeneration timer is set to produce an optimum rate of regeneration; the rate chosen is a compromise between the amount of data that can be displayed and the need to maintain the image. A GSRT order in the buffer program initiates automatic regeneration (see "Regeneration Timer" below for details).

After being loaded and started by the application program, a buffer program containing a GSRT order is repeatedly executed under the control of the regeneration timer.

Associated with the buffer program, and controlling its execution, are the following items:

- **Buffer Address Register:** This controls the addressing of the display buffer. During execution of the buffer program, the buffer address register is incremented to address sequentially each word in the buffer, wrapping from address 32 766 to address 0. However, the contents of the buffer address register may also be changed by:

  - A Set Buffer Address Register and Stop command issued by the host system

  - A Set Buffer Address Register and Start command issued by the host system

  - Execution of a read or write command issued by the host system

  - Buffer orders that transfer control within the buffer program.

- **Regeneration Timer:** This controls the frequency of regeneration of the displayed image to give a steady intensity level. A GSRT order initiates the time-out and starts execution of the buffer program. When the next GSRT order is encountered, the end of the previously initiated time-out allows it to be executed; this initiates the timer again, and executes the buffer program.

  If the execution time of the buffer program exceeds the set time-out (21.7 milliseconds), the rate of regeneration is controlled by the execution time. In this case, adding data to or removing data from the display may affect the quality of the image.

  Note: If two or more buffer programs are executing in the display buffer, the total execution time should be less than or equal to the regeneration time-out to maintain the optimum regeneration rate.

- **Attribute Register:** This contains the current value of the Blink, Line Type, and Intensity attributes. Attributes may be changed during execution of the buffer program to assign different attributes to components of the image.

  Default values for the attributes are: nonblinking, solid line, and normal intensity (level 5). These default values are set when a Set Buffer Address Register and Start command is received from the host system or when a GSRT order is executed.

Notes:

1. The Intensity attribute is independent of the setting of the brightness control on the 3251.

2. With a 3255 Model 2, four intensity levels can also be set by specific graphic mode orders. See chapter 3, "Four-Level Intensity" (3255 Model 2 only).

- **Cursor Location Registers:** The 3255 maintains, for each attached 3251, registers that contain the buffer address of the character-mode data field to which the cursor is assigned. Entering an alphanumeric character puts the character code into the addressed data field and, normally, moves the cursor to the next field in the data list.

  The contents of the cursor location register for a particular 3251 are also updated by the host system issuing an Insert Cursor command or Remove Cursor command.

- **X,Y Position Registers:** The 3255 maintains, for each attached 3251, registers that show the coordinates of the currently addressed point in the image area (the current beam position).

  The content of the X,Y position registers for a particular 3251 may be interrogated by the host system issuing a Read X,Y Position Registers command. This interrogation may be done, for example, to identify the coordinates of a point selected by the light pen.

Image Area

The image area is a square with 1024 addressable points on the X axis and 1024 addressable points on the Y axis (see Figure 2-1). Points are addressed by their X,Y coordinates. The distance between adjacent points on the same axis (either horizontal or vertical) is termed a *raster unit*; a raster unit is about 0.3 millimeter (0.012 inch).

X,Y coordinates of 0,0 address the bottom left-hand corner of the image area; incrementing X to 1023 moves the electron beam horizontally across the image area to the right-hand side; incrementing Y to 1023 moves the beam up the image area to the top edge.

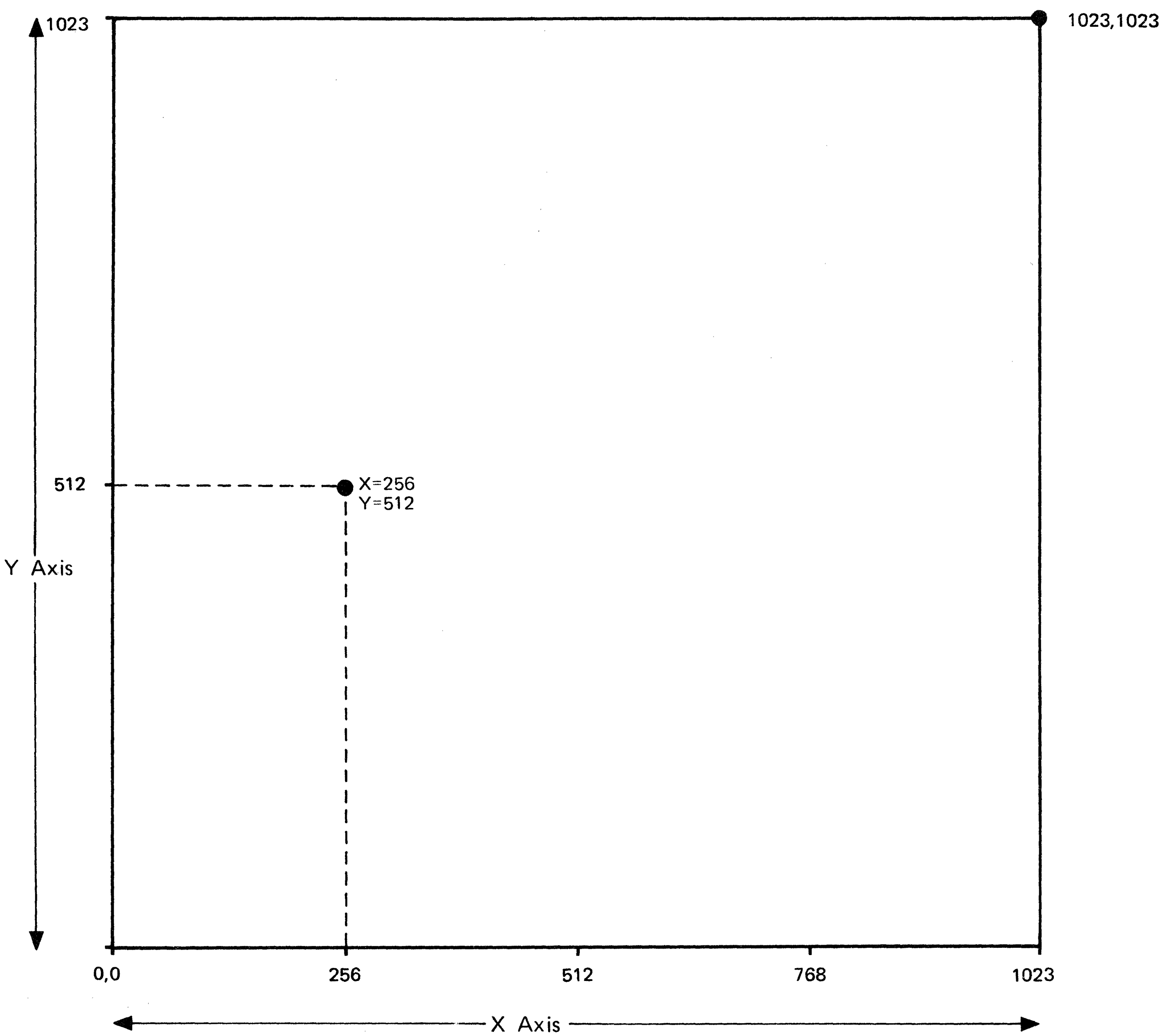

Figure 2-1. Image Area Coordinates

Addressing Point Coordinates

Any point in the image area may be addressed by graphic-mode orders in the buffer program. Graphic-mode orders are followed by data lists containing pairs of coordinates, and the beam is moved sequentially from point to point. For details, see "Graphic Modes" later in this chapter.

The coordinates contained in a graphic-mode data list may be either absolute or incremental, depending upon the buffer order that precedes the list:

- **Absolute Coordinates**: When a buffer order specifies absolute coordinates, data from the data list replaces the current value in the X,Y position registers.

Thus, if the electron beam is located at coordinates X=65, Y=137 (65, 137) and an absolute order with 2163 in its data field is executed, the beam is moved to the coordinates X=21, Y=63.

- **Incremental Coordinates**: When a buffer order specifies incremental coordinates, data from the data list is added to the current value in the X,Y position registers. Thus if the electron beam is located at coordinates 65, 137 and an incremental order with 2163 in its data field is executed, the beam is moved to the coordinates X=86, Y=200.

  Incremental coordinates are limited to increments ranging from -64 to +63 units. Negative increments are specified in twos complement form. For a definition and example of twos complement, see the glossary at the back of this manual.

Character-mode orders do not address point coordinates. Data lists following character-mode orders contain character codes. The first character is written at the current beam position, and the character generator advances the beam to the next character position.

Graphic Modes

Graphic-mode orders move the electron beam to an addressed point or series of points. The two types of graphic mode are:

- **Point Plotting**: This moves the beam from the current position to the addressed point, suppressing display of the beam between points.

- **Vector Plotting**: This uses the vector generator to generate straight lines between the current position and the addressed point.

Orders are provided to set the mode and define either absolute or incremental coordinates:

Data lists for the graphic-mode orders may contain more than one field, where each field contains the X,Y coordinates, or X,Y increments of a point. If more than one field is supplied, point orders move the beam from point to point through the list and vector orders draw lines between the points.

Graphic-mode orders take effect from the current beam position. The attributes and light pen mode that are current when the character-mode order is executed remain in force while the data list is being processed.

Offscreen Beam Tracing - Graphic Mode

Offscreen coordinates cannot be specified as absolute coordinates. Incremental coordinates could, however, produce requests to position the beam outside the image area. The display of points and lines is inhibited when either the current beam position or the addressed point or line-end point is outside the image area.

The 3255 maintains logically correct X,Y values, within the limits -1023 and +2047, even if several consecutive movements remain outside the image area. Attempts to increment either coordinate beyond these limits cause an undefined current beam position.

The X and Y coordinate values returned to the host system by the Read X,Y Position Registers command, or stored in the display buffer by the Store X,Y Position Registers (GSXY) order, are modulo 1024. It is not possible to determine, from the contents of the X,Y registers, whether the current beam position is offscreen.

Graphic-Mode Data Fields

A graphic-mode order is followed by a data list, each field in that list containing the coordinates (absolute or incremental) of a point. The size of data fields in the list is determined by the type of addressing specified in the order. For absolute coordinates, each field contains two words, the X coordinate in the first word and the Y coordinate in the second word; for incremental coordinates, each field contains one word, the X increment in the first byte and the Y increment in the second byte. Each data field also contains an indication that determines whether the point or line is to be displayed or blanked.

The data list is terminated when the hexadecimal value 2A is encountered at the even-numbered byte of an addressed word. This code tells the 3255 that the addressed word is a buffer order.

Graphic-mode data fields cannot directly accept data entered at the alphanumeric keyboard. The program (buffer or application) may, however, contain routines that modify the contents of the data field in response to the following operator actions:

- Making a selection with the light pen
- Entering coordinates in an alphanumeric field defined for that purpose and then pressing the END key
- Pressing a program function key to initiate a change to the displayed image.

Vector Generator

Operating under the control of graphic-mode orders, the vector generator produces the beam deflection needed to draw a straight line between two points. The type of line generated (solid, dotted, dashed, or dot-dashed) is determined by the current value of the Line Type attribute in the attribute register.

Character Modes

Character-mode orders control the generation of characters in the image area. These orders precede a data list that contains either a character string defined by the programmer or fields in which characters entered at the alphanumeric keyboard may be stored. Character-mode orders allow alphanumeric character strings to be defined with any combination of three characteristics:

- One of four character sizes (small, basic, medium, or large)
- Orientation (horizontal or rotated 90° counterclockwise)
- Either protected from keyboard entry of alphanumeric data, or unprotected.

Character-mode orders do not move the beam to a start point; they take effect at the current beam position. The attributes and light pen modes that are current when the character-mode order is executed remain in force while the data list is being processed. Character-mode orders do not reset the coordinate addressing defined by the most recently executed graphic-mode order; the current type of coordinate addressing, absolute or incremental, affects the automatic generation of new lines when the beam is moved offscreen, as described in "Offscreen Beam Tracing - Character Mode" below.

In processing a field in a character-mode data list, the 3255 generates the required character and advances the beam to the center point of the next character position, the X coordinate being incremented by the number of units in a character space.

Space characters in the data list advance the beam from the current character position to the center of the next character position. Null characters in the list do not affect the current beam position. Backspace (overstrike) characters move the beam back to the previous character position. New-line characters move the beam to the start of the next line; this is done by resetting the X coordinate to zero and decrementing the Y coordinate by the number of units in a line space.

A similar effect occurs with rotated character fields, and may be deduced by rotating the image area so that X=1023, Y=0 (1023, 0) is the bottom left-hand corner. 1023, 1023 is then the bottom right-hand corner, 0, 0 is the top left-hand corner, and 0, 1023 is the top right-hand corner.

Offscreen Beam Tracing - Character Mode

Under certain conditions, a new line is forced if an advance to the next character position would move the beam outside the image area. Forcing a new line does not insert a new-line character into the data list.

Data fields following character-mode orders contain alphanumeric character codes. Advancing of the beam from one character position to the next and to a new line is controlled by the character generator. Absolute coordinates cannot address points outside the image area, but incremental coordinates can; the action of the character generator depends upon the type of coordinates used in the most recent graphic-mode order. Depending upon that graphic-mode order, the following characteristics apply to the character generator when it is processing a non-rotated character field.

- **Absolute Coordinates**:
  - A new line is forced if incrementing X to the next position results in X being greater than 1023 (that is, the center of the next character is outside the image area).
  - A new-line character in the data list resets X to 0 and decrements Y by the number of units in a line space.
  - A new line at the top of the image area is forced if decrementing Y for a new line makes Y negative (that is, the center of the first character of a new line is below the image area).

- **Incremental Coordinates:**
  - A new line is never forced; if advancing to the next character position moves the beam out of the image area, the beam 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 1024, X is reset to 0 and Y is decremented by the number of units in a line space. (If decrementing Y makes Y negative, Y is set to 1023 to force a new line at the top of the image area.)
    - If Y is in the range of 1024 through 2047, and decrementing Y makes it less than 1024, then Y is set to 2047 and X is reset to 0.

Character-Mode Data Fields

A character-mode order is followed by a data list which must contain a whole number of words. Each byte of this list contains an extended binary-code decimal interchange coded (EBCDIC) character; if there is an odd number of characters, the list must be padded out by one byte (using, for example, a null character) to end on a word boundary. Appendix C shows the EBCDIC codes for the supported character sets.

The data list is terminated when the hexadecimal value 2A is encountered at the even-numbered byte of an addressed word. This hexadecimal code tells the 3255 that the addressed word is a buffer order.

Cursor Location

The cursor is displayed in the image area to show where the next character entered from the alphanumeric keyboard will be displayed. The cursor location register addresses the byte in buffer storage that will be loaded with the character code of the entered character.

Notes:

1. The cursor must be assigned to an unprotected character field but cannot be assigned to a position containing a new-line character.
2. If the addressed byte contains a null or backspace (overstrike) character, the cursor is not displayed.

Initially, the cursor location is set by the application program. The cursor advances one character position after each character entered. When an unprotected character field has been filled, the cursor remains at the last position in the field. To move the cursor from the end of a field, the operator may press either BACKSPACE or JUMP. Any character key pressed while the cursor is at the last position of a field will overwrite any previously entered character in that position.

Character-Mode Data Entry

When the operator presses a character key, the coded character is placed into the buffer location addressed by the cursor location register; the code is loaded at the next execution of a Start Regeneration Timer (GSRT) order. In the next regeneration cycle, the entered character is displayed. The size and orientation of the displayed character is determined by the character-mode order preceding the data field.

Notes:

1. The END and CANCEL keys do not enter data into the display buffer. When pressed, these keys raise an I/O interruption to the host system. The action resulting from this interruption depends upon the application program.

2. With a 3251 attached to a 3255 Model 2, the CANCEL key will operate only when the CONTINUOUS key is first pressed and held down.

Character Generator

Operating under the control of character-mode orders, the character generator produces the train of blanked and unblanked beam deflections needed to write defined characters in the image area. Each displayable character in the data list following a character-mode order is generated inside a rectangle that is centered, approximately, on the current beam position.

The rectangular character position is one character space wide and one line space high. Character and line spaces are defined in raster units for each of the four character sizes. For example, if the character-mode order specifies large characters, the rectangular character position is 21 units wide and 30 units high.

Irrespective of the character size, the character generator deflects the beam within the limits of a character grid. This grid is independent of the raster matrix, and is scaled to fit the rectangular character position. Figure 2-2 shows the character grid, illustrates the generation of characters within that grid, and shows the relationship between the four character sizes. If the center point of a character lies within the image area, that character is fully displayed, even though parts of it lie outside the image area.

When rotated characters are being generated, the rectangular character position is rotated 90° counterclockwise.

Note: Some Katakana characters and some special characters extend beyond the character limits within the grid. All characters are, however, contained within the grid.

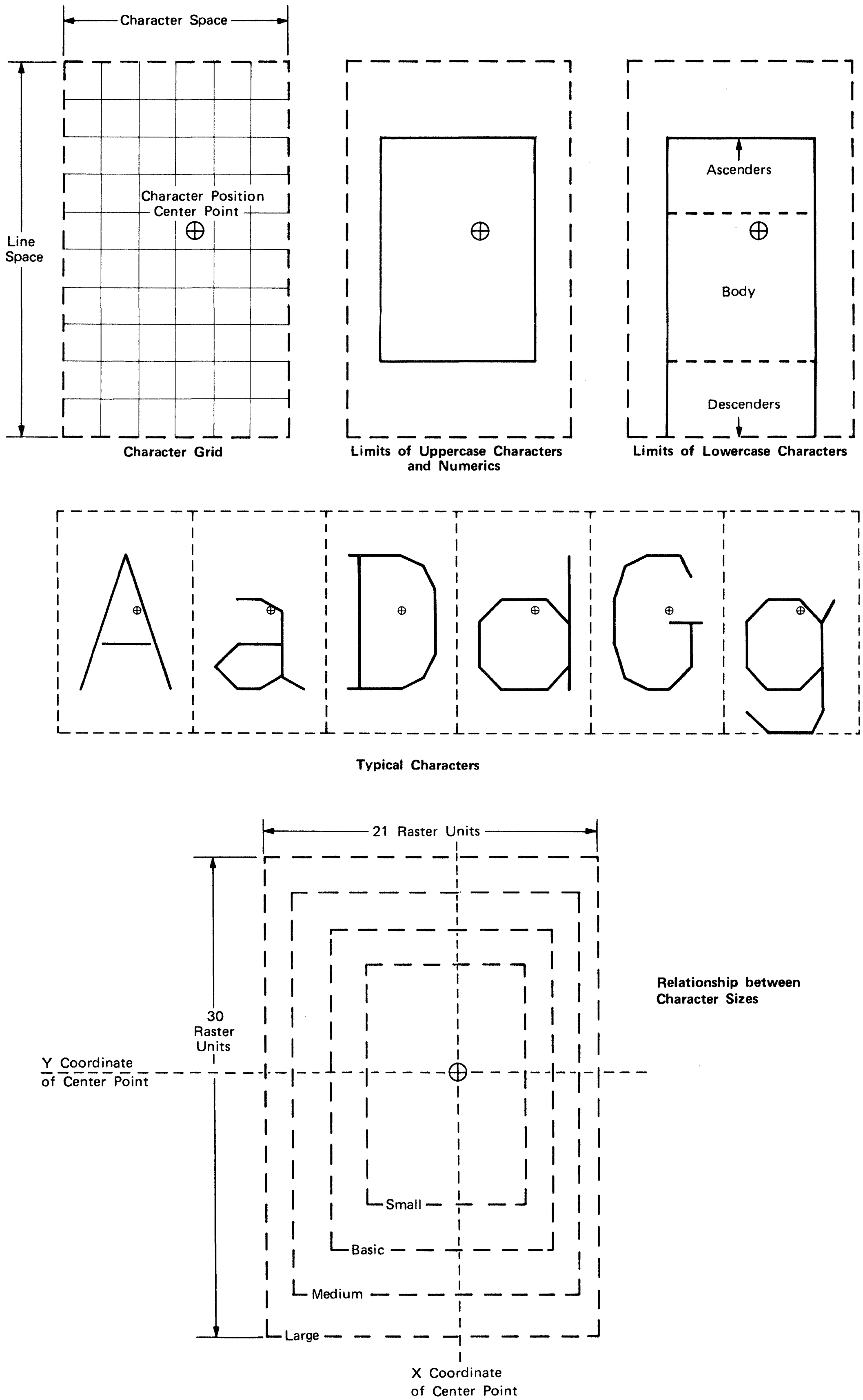

Figure 2-2. Rectangular Character Position

Character Sizes

Each of the four character sizes has a fixed spacing between the character center points on a horizontal line (character spacing) and between lines (line spacing). Figure 2-3 shows the dimensions for displayed characters and their spacings.

Chara-cter Size	Approximate Character Area - Width x Height, Millimeters (inch)	Character Spacing		Line Spacing	
		Raster Units	mm (inch)	Raster Units	mm (inch)
Small	2.0 x 3.0 (0.08 x 0.12)	10	3.0 (0.12)	15	4.5 (0.17)
Basic	2.8 x 4.1 (0.11 x 0.16)	14	4.2 (0.16)	20	6.0 (0.23)
Medium	3.3 x 5.1 (0.13 x 0.20)	18	5.4 (0.21)	25	7.5 (0.29)
Large	4.1 x 6.1 (0.16 x 0.24)	21	6.3 (0.25)	30	9.0 (0.35)

Figure 2-3. Displayed-Character Dimensions

From Figure 2-3 it can be seen that the approximate capacity of the image area is:

- Small characters: 68 lines of 103 characters per line
- Basic characters: 52 lines of 74 characters per line
- Medium characters: 41 lines of 57 characters per line
- Large characters: 35 lines of 49 characters per line.

Character Rotation

If the programmed character-mode order defines rotated characters, the generated character string is rotated 90^0 counterclockwise.

In rotated mode, the character generator advances from center point to center point by incrementing the Y coordinate, and creates a new line by incrementing the X coordinate and resetting the Y coordinate to O.

Light-Pen Modes

The sensitivity of the displayed image to selection by the light pen is determined by the setting of the light-pen modes. The programmer may set different light-pen modes for individual components of the image. Light-pen-mode orders are designed to give flexibility to the buffer program. As a result, the way in which the light pen is used can be tailored to suit the requirements of the application.

The light pen can only be used to select components of the image that are displayed at intensity levels 5, 6, or 7.

Note: To allow a light-pen detection at a point where there is no displayed data, the usual technique is to provide a buffer program that floods the whole screen with data until an interrupt occurs; this flooding may be momentary, and can be removed in response to the light-pen detection.

Light pen modes are combinations of two independent modes: (1) the *detects* mode, which determines whether or not a detection will occur if the light pen is used to select a component of the image; and (2) the *response* mode, which determines the response of the buffer program to a light-pen detection.

- *Detects Mode*: The detection of a light-pen operation can only occur on components of the image that are generated while detects are enabled, and when the intensity of the component and the brightness of the image are sufficient to excite the light- sensor. To cause a light-pen detection, the operator points the light pen at the required component; when the electron beam moves into the field of the light-sensor, a light-pen detection occurs.

  The detects mode can be *disabled* or it can be set to either *switch-enabled* or *no-switch-enabled*. If switch-enabled is set, the light pen must be pressed against the screen, closing the tip-switch, to allow detection to occur; only one detection can occur for each operation of the tip-switch. If no-switch-enabled is set, a detection occurs each time the electron beam passes the tip of the pen.

- *Response Mode*: The action taken in response to a light-pen detection can be either an *immediate* interrupt or a *deferred* response. If the state is immediate, execution of the buffer program stops when a light-pen detection occurs; the 3255 then raises an I/O interruption to the host system. If the state is deferred, the 3255 records the occurrence of a light-pen detection as a deferred detect; subsequent buffer orders can test for a deferred detection and, depending upon the buffer order used, either transfer control within the buffer program or stop execution and raise an I/O interruption to the host system.

- In addition, a 3255 Display Control Unit Model 2 can have a single element intensification function enabled. This function identifies a vector, character, or point detected by a light pen without intervention from the host system. The whole of the identified element is brightened by the 3255 at the moment of light-pen detection. However, enabling this feature will impact image content performance; refer to "Image Content" in Chapter 5.

Control Modes

Orders executed in control mode are capable of:

- Controlling the regeneration timer, synchronizing input from the keyboards, synchronizing keyboard interruptions, and setting default attributes and light-pen modes
- Branching in the buffer program
- Loading and storing the contents of the attribute register
- Storing the X,Y coordinates of the current beam position
- Moving addresses and data from one field in the display buffer to another buffer
- Performing a "no operation."

For details of the buffer orders executed in control mode, see "Control-Mode Orders" in Chapter 3.

Channel Operations

Communication between the 3250 system and the host system is across an I/O channel. Channel operations may be initiated either by the system issuing channel commands or by an I/O interruption from the 3250.

Channel Commands

Channel commands may be used for the following operations. (For more detailed information, see Chapter 4.)

- **Write Buffer:** This command transfers data from the host system to the display buffer of the 3255.
- **Read Buffer:** This command transfers data from the display buffer to the host system.
- **Read Cursor:** This command transfers data from the display buffer to the host system; data transfer stops when the cursor is encountered. The Read Cursor command can also be used to find the cursor without transferring data to the application program in the host system.
- **Read Manual Input:** This command gives the host system information concerning interrupts from the program function keyboard or the alphanumeric keyboard (END and CANCEL keys).
- **Read X,Y Position Registers:** This command sends the current contents of the X,Y position registers to the host system. Normally, the command is issued in response to an I/O interruption caused by a light-pen detection.
- **Set Audible Alarm:** This command causes a short beep at the 3251.
- **Set Buffer Address Register and Start:** This command initiates buffer program execution at the specified address.
- **Set Buffer Address Register and Stop:** This command stops buffer program execution and sets the buffer address register. Normally, it is used before a write or read operation to define where the operation is to commence in the display buffer.
- **Set Program Function Indicators:** This command causes the back-lit keys of the program function keyboard to be lit or extinguished. Each of the 32 lamps can be lit or extinguished in any combination.
- **Insert Cursor:** This command sets the cursor location register, normally causing the cursor to be displayed on the screen.
- **Remove Cursor:** This command removes the cursor from the display.

- **Sense:** This command causes information concerning the status of the 3251 to be transmitted to the host system. Normally, a Sense command is issued in response to an I/O interruption or error condition.

- **No-Operation:** This command performs no operation.

I/O Interruptions

To communicate changes in the status of the buffer program or the 3250 system to the host system, I/O interruptions are raised. The Interpretation of I/O interruptions is the responsibility of the host system. Chapter 4 defines the sense and status bytes available to the host system when an I/O interruption is raised; for a complete description of operations on an I/O channel, refer to *IBM System/370 Principles of Operation*, GA22-7000.

I/O interruptions are raised if any of the following occur:

1. A light-pen detection when the light pen is enabled for immediate response

2. Execution of a Permit Detect Interrupt (GPDI) order if there is an outstanding deferred detect

3. Execution of an End Order Sequence (GEOS) order

4. The operator pressing any of the 32 program function keys (see the note below)

5. The operator pressing either END or CANCEL on the alphanumeric keyboard (see the note below).

Note: When the buffer program is executing, keyboard interruptions are synchronized to the execution of a Start Regeneration Timer (GSRT) order. When one of the keys mentioned in 4 and 5 is pressed, the next GSRT order executed sets the relevant key code into the manual input register and raises an I/O interruption to the host system. The host system must respond with a Read Manual Input command to clear the register; the pressing of END, CANCEL, or any program function key does not cause any action if a key code is in the manual input register.

Keyboard Operation - 3251 Model 1 Attached to a 3255 Model 2

Polling

If an active picture is being displayed, the program function keyboard and the alphanumeric keyboard are polled sequentially for data/interrupts at Start Regeneration Timer (GSRT) time. If there is no GSRT order present in the buffer program, neither keyboard will be polled until the program stops.

If no active picture is being displayed, the keyboards are polled sequentially at the real time clock rate of 46 cycles per second (cps) until a picture is displayed when GSRT polling is resumed.

Program Function Keyboard Operation

When a program function keyboard (PFK) key is pressed, the key number is passed by way of a keyboard response to the 3255. The 3255 will generate an Attention interrupt and set data in the read manual input (RMI) register. If the RMI register is already set, or there is an oustanding Attention (or Attention with Unit Check), the keyboard is ignored. That is, the keyboard is logically locked out and the interrupt will not be generated. A PFK operation cannot overwrite an existing PFK condition or an existing END or CANCEL condition from the alphanumeric keyboard.

Alphanumeric Keyboard Operation

All keystrokes, with the exception of END and CANCEL, are taken and processed provided that there is an unprotected data field and a cursor available. An END or CANCEL keystroke will cause an Attention interrupt to be generated and the read manual input (RMI) register to be set, unless there is already a previous Attention interrupt with or without Unit Check outstanding. In which case, the later input is locked out.

Additional Notes on Keyboard Operation

Notes:

1. *3255 Display Control Unit Model 1.* With simultaneous interruptions from the keyboards, the first response to a keyboard poll that is received will lock out the other interruption.

2. *3255 Display Control Unit Model 2.* If an END or CANCEL interruption is received after a PFK interrupt has been raised, but before a Read Manual Input command has been received, the END or CANCEL interruption will set a new value into the RMI register and generates a new Attention interruption.

3. *Light-Pen Strike.* If a light-pen strike is received, and the Attention with Unit Check interrupt is outstanding, the RMI register cannot be set. Also, if the light-pen tip-switch is closed, the PFK interrupt is locked out.

Chapter 3. Buffer Program

Graphic display operations at a 3251 Display Station are controlled by the execution of a buffer program. This program is stored and executed in the display buffer of the 3255 Display Control Unit to which the 3251 is attached. Each 3255 can accommodate two 3251s, but the display buffer must contain a buffer program for each active 3251. Allocation of buffer storage is at the discretion of the host-system application program; the 3255 does not place any constraints on the allocation of the display buffer. Management of the display buffer storage is a user function for which host-system software conventions or controls are required. Sections of buffer programs, or a complete buffer program, can be shared among 3251s using the same 3255.

Note: A 3255 Display Control Unit Model 2 has two 32 768 byte display buffers, and can support three 3251 Display Stations Model 1. Implementation of the additional buffer is described in Chapter 2, "Sharing a Display Buffer."

The buffer program consists of orders interleaved with data. The orders are interpreted as requests to perform operations, such as the transfer of control within the buffer program, or as requests to set a particular mode. The display buffer can contain orders and data that are not in the flow of control of currently executing buffer programs.

The orders and data that form a buffer program are created by an application program in the host system. The application program is also responsible for transmitting the buffer program to the display buffer and for initiating execution of the buffer program, using the channel command outlined in Chapter 2 under "Channel Operations" and described more fully in Chapter 4.

Execution of a buffer program is started by the Set Buffer Address Register and Start command issued from the host system. The execution continues until it is ended by either an I/O interruption being raised by the 3250 system or by the receipt of a command from the host system.

As mentioned in Chapter 2, orders are grouped into four modes:

- **Graphic-mode orders**, which reposition the beam, display points, or draw lines between points

- **Character-mode orders**, which write characters centered on an addressed point

- **Light-pen mode orders**, which determine the way the 3251 responds when the light pen is used

- **Control-mode orders**, which, for example, allow transfer of control within a program.

Orders are of fixed length. Graphic-mode and character-mode orders each occupy 1 word (2 bytes) while control-mode and light pen mode orders are 1, 2, or 3 words in length, depending on the specific order.

The first byte of an order code always contains hexadecimal 2A. The second byte contains the code defining the required mode and the operation to be performed in that mode. Only a subset of the 256 available codes is used; the unused codes, if encountered, are interpreted as No-Op 2-Byte (GNOP2) orders.

Graphic-mode and character-mode orders can be followed by a data list of variable length. The length of each field in a data list depends on the specific order being executed. A data list describes the lines, points, or characters to be displayed on the screen and is delimited by the occurrence of any word of the form hexadecimal 2Axx.

In this chapter, each buffer order is shown as a series of one, two, or three 2-byte words. In turn, each word is shown as four fields of 4 bits each, followed by the hexadecimal equivalent of the operation code or a brief description of the operands. Mnemonics are used as convenient shorthand for the names of the buffer orders. Thus, each order appears as follows:

Name of Order - Mnemonic

```
     Byte 0            Byte 1
 ┌──────────────┐  ┌──────────────┐
 │ ....    .... │  │ ....    .... │  Hexadecimal Value
 └──────────────┘  └──────────────┘  or Description
```

Order Sequencing

Orders are normally executed sequentially; the buffer address register is incremented to address each word in turn.

If there is no transfer of control upon completion of the current order, the buffer address register addresses the word following the order; if this word does not contain a buffer order (that is, does not start with hexadecimal 2A), words are addressed sequentially until an order is found. Thus control-mode and light-pen-mode orders may be followed by undefined words.

When execution of a buffer order transfers control in the buffer program, a new address (the destination) is loaded into the buffer address register. The buffer address register then sequentially addresses words, starting at the destination, looking for an order code.

The buffer address register contains 16 bit positions. Because the high order bit, bit 0, is not needed for addresses from 0 through 32766, bit 0 is ignored. To force addressing on a word boundary, bit 15 of the buffer address register is also ignored.

Buffer Program Termination

Execution of the buffer program is terminated, thereby stopping generation of the displayed image, in the following ways and for the following reasons:

- It is terminated asynchronously (that is, termination is not synchronized to execution of the buffer program) by:
  - A Set Buffer Address Register and Stop channel command from the host system (perhaps to set a buffer address for the start of a read or write operation)

- A Set Buffer Address Register and Start channel command from the host system. (This command sets the buffer address and restarts execution at the addressed location.)

- It is terminated, and an I/O interruption is raised to the host system, by:

  - A light-pen detection during the processing of graphic-mode or character-mode data lists, if detects are currently enabled and immediate.

  - Execution of a Permit Detect Interrupt (GPDI) order if a deferred detect is outstanding.

  - Execution of an End Order Sequence (GEOS) order.

For a description of the above channel commands, and of the status and sense bytes associated with an interrupt, see Chapter 4.

Note: I/O interruptions raised to signal the selection of the END or CANCEL key on the alphanumeric keyboard or any program function key do not terminate the execution of the buffer program.

Graphic-Mode Orders

Graphic-mode orders are used to display points or lines or to reposition the beam on the display screen. Using coordinates contained in a data list following the order, graphic-mode orders move the electron beam from the current beam position to each point in turn; at each point, the current beam position is updated to the coordinates of the addressed point. Plotting, from point to point may be either point plotting or vector plotting:

- **Point plotting** suppresses the beam while it moves. Each addressed point is displayed if the blanking bit in the corresponding data field is off.

- **Vector plotting** draws a line from the current beam position to the addressed point. Each drawn line is displayed if the blanking bit in the corresponding data field is off. (The type of line generated is determined by the current Line Type attribute.)

Each graphic-mode order is followed by a list of data fields that determine the movement of the beam on the screen. Each data field in the list contains X,Y coordinates defining either an addressable point (absolute coordinates) or a displacement relative to the current beam position (incremental coordinates). This coordinate pair determines the point to which the beam must travel, that is, the new current beam position.

To maintain compatibility with the IBM 2250 Display Unit Model 3, a 12-bit field is used for each absolute coordinate, the two low-order bits being ignored.

The blanking bit is a flag within each data field that determines whether or not the point moved to, or the line drawn by moving the beam to the new beam position, should be displayed; when the bit is off, display occurs. For beam movements that do display a line or point, the current attributes (that is, Blink, Line Type, and Intensity) are used.

Point Plotting Orders

Two orders, Enter Graphic Mode Absolute Point (GEPM) and Enter Graphic Mode Incremental Point (GEP12), allow the addressing of point coordinates, with the option of displaying the addressed point. Moving the electron beam to an addressed point updates the current beam position to the coordinates of the addressed point.

These orders are followed by a data list, the format of which is determined by the order (absolute coordinates or incremental coordinates). The data list may contain many data fields. Each field contains the coordinates of a point and a blanking bit. The electron beam is moved from point to point until all points in the data list have been addressed.

Enter Graphic Mode Absolute Point - GEPM

This 1-word order causes the 3255 to interpret data as absolute coordinates and to move the electron beam to the points addressed by the data list. Each field in the data list following the order occupies 2 words: the first word contains the blanking bit and an X coordinate, and the second word contains a Y coordinate. Appendix A lists the hexadecimal values of absolute coordinates.

	Byte 0	Byte 1	
Order	0010 1010	0000 0000	Hex 2A00
First Field Data in Data List	0b00 xxxx	xxxx xxii	X coordinate
	0000 yyyy	yyyy yyii	Y coordinate

Legend:

b - Blanking bit: if b=1, the addressed point is not displayed.

x - 10-bit X coordinate.

y - 10-bit Y coordinate.

i - These bits are ignored.

0 - These bits must be set to 0.

Enter Graphic Mode Incremental Point - GEPI2

This 1-word order causes the 3255 to interpret data as incremental coordinates and to move the electron beam to the points addressed by the data list. Each field in the data list occupies 1 word: the first byte of that word contains an X increment, and the second byte contains a Y increment and the blanking bit. Appendix B lists the hexadecimal values of incremental point coordinates.

A maximum displacement of -64 or +63 units (approximately 19 millimeters (0.75 inch) in either direction along an axis) can be specified for each increment. The point addressed by a pair of incremental coordinates is determined by adding the increments to the current beam position.

Negative increments are specified in twos complement form. See the glossary at the back of this manual for a definition and example of twos complement.

	Byte 0	Byte 1	
Order	0010 1010	0000 0100	Hex 2A04
First Field in Data List	xxxx xxx1	yyyy yyyb	X, Y increments

Legend:

x - 7-bit X increment.

y - 7-bit Y increment.

b - Blanking bit: if b=1, the addressed point is not displayed.

Note: Bit 7 of byte 0 must be set to 1 to avoid interpretation of the data field as an order. (Setting bit 7 to 1 prevents the first byte ever being 2A hexadecimal.)

Vector Plotting Orders

Two orders, Enter Graphic Mode Absolute Vector (GEVM) and Enter Graphic Mode Incremental Vector (GEV12), allow the addressing of point coordinates, with the option of displaying the line from the current beam position to the addressed point. After a line is drawn, the line-end point becomes the current beam position. These orders are followed by a data list, the format of which is determined by the order (absolute coordinates or incremental coordinates).

The data list may contain many data fields. Each field contains the coordinates of a line-end point and a blanking bit. The electron beam is moved from line-end point to line-end point until all points in the data list have been been addressed.

Enter Graphic Mode Absolute Vector - GEVM

This 1-word order causes the 3255 to interpret data as absolute coordinates and to move the beam from the current position to the line-end points addressed by the data list. Each field in the data list following the order occupies 2 words: the first word contains the blanking bit and an X coordinate, and the second word contains a Y coordinate. Appendix A lists the hexadecimal values of absolute point coordinates.

	Byte 0	Byte 1	
Order	0010 1010	0000 0010	Hex 2A02
First Field in Data List	0b00 xxxx	xxxx xxii	X Coordinate
	0000 yyyy	yyyy yyii	Y Coordinate

Legend:

b - blanking bit: if b=1, the beam is moved without displaying a line.

x - 10-bit X coordinate.

y - 10-bit Y coordinate.

i - These bits are ignored.

0 - These bits must be set to 0.

Enter Graphic Mode Incremental Vector - GEVI2

This 1-word order causes the 3255 to interpret data as incremental coordinates and to draw lines from the current beam position to the line-end points addressed by the data list. Each field in the data list occupies 1 word: the first byte of that word contains an X increment, and the second byte contains a Y increment and the blanking bit. Appendix B lists the hexadecimal values of incremental point coordinates.

A maximum displacement of -64 or +63 units (approximately 19 millimeters (0.75 inch) in either direction along an axis) can be specified for each increment. The point addressed by a pair or incremental coordinates is determined by adding the increments to the current beam position.

Negative increments are specified in twos complement form. See the glossary at the back of this manual for a definition and example of twos complement.

	Byte 0	Byte 1	
Order	0010 1010	0000 0101	Hex 2A05
First Field in Data List	xxxx xxx1	yyyy yyyb	X, Y, increments

Legend:

x - 7-bit X increment.

y - 7-bit Y increment.

b - Blanking bit: if b=1, the beam is moved without displaying a line.

Note: Bit 7 of byte 0 must be set to 1 to avoid interpretation of the data field as an order. (Setting bit 7 to 1 prevents the first byte ever being 2A hexadecimal.)

Character-Mode Orders

Character-mode orders cause the 3255 to interpret data as EBCDIC characters and to generate and display the coded characters. Each character-mode order is 1 word long and is followed by a list of 1-word data fields that determine the alphanumeric characters to be displayed on the screen. The Intensity and Blink attributes held in the attribute register are applied to the display of these characters.

Character-mode orders determine:

- The size of character generated by fields in the data list
- The orientation of the generated characters and lines of characters displayed from the data list (normal or 90^0 counterclockwise)
- Whether or not the data list is protected from modification from the alphanumeric keyboard.

Appendix C lists the supported character codes. If unsupported codes are used in the data list, the character displayed is not defined.

Each displayable character occupies one character position on the screen. The character is generated and written with the current beam position as the center point, and the current beam position is updated to move the beam to the center point of the next character position.

The 16 character-mode orders are particular forms of:

Byte 0	Byte 1	
0010 1010	0100 rpss	Hex 2A4x

Legend:

r - Rotation: If r=0, characters are displayed upright, the displayed field is horizontal, and characters are written left to right. If r=1, characters are rotated 90 degrees counterclockwise, the displayed field is vertical, and the characters are written from the bottom to the top of the screen.

p - Protection: If p=0, the data field following the order can be updated from the alphanumeric keyboard. If p=1, the data field following the order is protected from data entered at the alphanumeric keyboard.

s - Size of characters:

Small	ss = 10
Basic	ss = 00
Medium	ss = 11
Large	ss = 01

Mnemonics and codes for the 16 character-mode orders are listed in Figure 3-1. In addition, the order codes hex 2A50, 2A51, and 2A52 are supported and have the same meanings as hex 2A40, 2A41, and 2A42 respectively.

Enter Character Mode Small – GECF (S)

0010 1010	0100 0010	Hex 2A42

Enter Character Mode Basic – GECF (B)

0010 1010	0100 0000	Hex 2A40

Enter Character Mode Medium – GECF (M)

0010 1010	0100 0011	Hex 2A43

Enter Character Mode Large – GECF (L)

0010 1010	0100 0001	Hex 2A41

Enter Character Mode Small Rotated – GECF (S,R)

0010 1010	0100 1010	Hex 2A4A

Enter Character Mode Basic Rotated – GECF (B,R)

0010 1010	0100 1000	Hex 2A48

Enter Character Mode Medium Rotated – GECF (M,R)

0010 1010	0100 1011	Hex 2A4B

Enter Character Mode Large Rotated – GECP (L,R)

0010 1010	0100 1001	Hex 2A49

Enter Character Mode Small Protected – GECP (S)

0010 1010	0100 0110	Hex 2A46

Enter Character Mode Basic Protected – GECP (B)

0010 1010	0100 0100	Hex 2A44

Enter Character Mode Medium Protected – GECP (M)

0010 1010	0100 0111	Hex 2A47

Enter Character Mode Large Protected – GECP (L)

0010 1010	0100 0101	Hex 2A45

Enter Character Mode Small Protected Rotated – GECP (S,R)

0010 1010	0100 1110	Hex 2A4E

Enter Character Mode Basic Protected Rotated – GECP (B,R)

0010 1010	0100 1100	Hex 2A4C

Enter Character Mode Medium Protected Rotated – GECP (M,R)

0010 1010	0100 1111	Hex 2A4F

Enter Character Mode Large Protected Rotated – GECP (L,R)

0010 1010	0100 1101	Hex 2A4D

Figure 3-1. Character-Mode Orders

Character-Mode Data Lists

Character-mode orders may be followed by a list of 1-word data fields. Each byte is examined for a character code:

Byte 0	Byte 1
aaaa aaaa	bbbb bbbb

Legend:

a - First EBCDIC character bits

b - Second EBCDIC character bits

The null character (hex 00) can be used to pad lists to an even number of bytes; null is not displayed, and it does not advance the beam to the next character position. If the cursor location addresses a null character, display of the cursor is inhibited; however, the operator may enter an alphanumeric character into a position containing a null character.

New-line characters (hex 15) are not displayed; they advance the current beam position to the first character position on the next line.

Backspace (overstrike) characters (hex 16) are not displayed; they return the beam to the previous character position. If the cursor location addresses a backspace character, display of the cursor is inhibited; however, the operator may enter an alphanumeric character into a position containing a backspace character.

Light-Pen-Mode Orders

The light-pen modes that are current when a component of the image is generated and displayed determine (1) whether or not a light pen detection occurs if the operator uses the light pen to select that component, and (2) the response of the 3255 to a light-pen detection. Two basic modes can be used; **detects** and **response**; buffer orders that change the state of one of these modes do not affect the state of the other mode.

- **Detects Mode**: This has three possible states: **disabled, switch-enabled**, and **no-switch-enabled.** If the detects mode is enabled, light-pen selection of a component of the displayed image will be accepted by the buffer program. Selection may require operation of the tip-switch (switch enabled), or be independent of the tip-switch (no-switch enabled).

  Note: Light-pen detections are only possible if the brightness control on the front panel of the 3251 is set to a level that allows the electron beam to excite the light-sensor on the tip-switch. Detections are inhibited on programmed intensity levels below level 5.

- **Response Mode**: This has two possible states: **immediate** and **deferred.** If the response mode is immediate when a light-pen detection occurs, the detection terminates the buffer program and causes an I/O interruption to be raised to the host system. If the response mode is deferred when a light-pen detection occurs, the detection is recorded as a deferred detect; subsequent buffer orders are able to transfer control, or to raise an interrupt, by testing for a deferred detect.

These states of light-pen-modes are described more fully below.

Figure 3-3 summarizes the way in which the light-pen-modes are modified by the buffer orders. All possible combinations of detects-mode and response-mode are shown, the orders that can change the light-pen-modes are listed, and the resulting mode for each starting combination is shown.

Mode Before Execution	Si	Sd	Ni	Nd	Di	Dd
Mode after GESD	Si	Sd	Si	Sd	Si	Sd
Mode after GENSD	Ni	Nd	Ni	Nd	Ni	Nd
Mode after GDPD	Di	Dd	Di	Dd	Di	Dd
Mode after GPDI	Si	Si	Ni	Ni	Di	Di
Mode after GDRD	Sd	Sd	Nd	Nd	Dd	Dd
Mode after GSRT	Si	Si	Si	Si	Si	Si

Legend:

Detects: S=switch-enabled, N=no-switch-enabled, D=disabled
Responses: i=immediate, d=deferred

Figure 3-2. Modification of Light Pen Modes

Default light-pen modes (switch-enabled and immediate) are set at the start of each regeneration cycle by the Start Regeneration Timer (GSRT) order or by a Set Buffer Address Register and Start command. The only other orders that can control the setting of light-pen modes are Disable Pen Detects (GDPD), Enable Switch Detect (GESD), Enable No-Switch Detect (GENSD), Permit Detect Interrupt (GPDI), and Defer Response to Detects (GDRD); these are described later in this section.

Figure 3-3 defines the orders that set the detects and response modes. Note that modes are individually set; changing the detects mode does not affect the response mode, and changing the response mode does not affect the detects mode.

Disable Pen Detects - GDPD. Sets detects mode to disabled.

0010 1010	1000 0101	Hex 2A85

Enable Switch Detect - GESD. Sets detects mode to switch-enabled.

0010 1010	1000 0100	Hex 2A84

Enable No-Switch Detect - GENSD. Sets detects mode to no-switch enabled

0010 1010	1000 0110	Hex 2A86

Permit Detect Interrupt - GPDI. Sets response mode to immediate (see Note).

0010 1010	1000 0111	Hex 2A87

Defer Response to Detects - GDRD. Sets response mode to deferred.

0010 1010	1000 0011	Hex 2A83

Note: If a deferred detect is outstanding when GPDI is executed, the buffer program is terminated and an I/O interruption is raised to the host system. When an I/O interruption is raised during the execution of a GPDI order, the buffer address returned in sense bytes 2 and 3 is the address of the word following the GPDI order.

Figure 3-3. Light-Pen-Mode Orders

Detects Mode

Disabled State

A light-pen detection cannot occur on data that is generated while the detects-mode is disabled. Detects mode would normally be disabled when the buffer program displays data on which a detection would be irrelevant, such as headings, footings, or operator instructions.

Disabling the detects mode does not affect the response mode, or the existence of a deferred detect. Although no new light-pen detections can occur, a previously established deferred detect can either (1) affect execution of a Transfer on No Detect (GTND) order, or (2) cause an I/O interruption if a Permit Detect Interrupt (GPDI) order is executed.

Switch-Enabled State

A light-pen detection occurs on data that is generated while the detects mode is in the switch-enabled state only when all of the following conditions are satisfied:

- The data is generated at intensity level 5 or higher.
- The light pen is positioned with the displayed data in the field of view of the light-sensor.
- The brightness control is set high enough to allow the beam to excite the light-sensor.
- The light-pen tip-switch is closed (that is, pressed in). The status of the tip-switch is interrogated during the execution of a GSRT order; for details, see "Start Regeneration Timer - GSRT" under "Control-Mode Orders" later in this chapter.
- No previous detection has occurred in switch-enabled mode since the light pen tip-switch was closed. Any number of detections in no-switch-enabled mode may occur between closing the tip-switch and accepting a detection in switch-enabled mode.

The switch-enabled state, sometimes called "normal" mode, is probably the most commonly used detects mode. In this state, the light pen may be carefully positioned on a displayed component before the switch is operated and a detection occurs; this care eliminates any risk of detections on unwanted components.

Notes:

1. The status of the tip-switch, open or closed, is interrogated at the start of a regeneration cycle; therefore a GSRT order must be included in the buffer program if switch-enabled detections are used.
2. Only one switch-enabled detection can occur in each regeneration cycle.
3. Only one switch-enabled detection can occur for each operation of the tip-switch. (No-switch-enabled detections may occur between the closing of the tip-switch and the acceptance of a switch-enabled detection.)

No-Switch-Enabled State

A light-pen detection occurs on data that is generated while the detects mode is in the no-switch-enabled state only when all of the following conditions are satisfied:

- The data is generated at intensity level 5 or higher.
- The light pen is positioned with the displayed data in the field of view of the light-sensor.
- The brightness control is set high enough to allow the beam to excite the light-sensor.

The no-switch-enabled state is sometimes called "continuous detect" mode because detections occur whenever the pen views data of adequate intensity. This mode is particularly suited for use in association with appropriate buffer programs to perform such functions as moving a tracking symbol across the screen.

Notes:

1. No-switch-enabled detections can be raised in a buffer program that does not contain a GSRT order. However, a GSRT order should always be included in the buffer program. The reasons are discussed later in this chapter in "Start Regeneration Timer - GSRT" (see under "Control Mode Orders").

2. Multiple no-switch-enabled detections can occur during a regeneration cycle.

3. The occurrence of a light pen detection is independent of tip-switch operation.

Response Mode

Immediate State

If the light-pen response mode is immediate, a light-pen detection stops the execution of the buffer program and causes an immediate I/O interruption to be raised to the host system. This interruption allows the host system to identify the coordinates of the character, point, or line that has been detected and to find the buffer location of the data that generated the element.

If a deferred defect is outstanding when a Permit Detect Interrupt (GPDI) order is executed, execution of the buffer program terminates and an I/O interruption is raised as though the light-pen detection had occurred at that point in the buffer program.

Deferred State

If the light-pen response mode is deferred, the occurrence of a light-pen detection is recorded as a deferred detect, and execution of the buffer program continues. Only one deferred detect can be recorded at a time.

Deferred response greatly extends the flexibility of the 3250 system. For example, the Transfer on No Detect (GTND) and Transfer on Deferred Detect (GTDD) control-mode orders make a test for a deferred detect, and transfer control in the buffer program if the required conditions are satisfied. This modification to the flow of control in the buffer program can be temporary or can be made permanent by use of Move Immediate Address (GMVA) or Move Immediate Data (GMVD) orders to modify the buffer program. Another common usage is to set deferred response before displaying each group of elements that constitute a single logical entity, then to change back to immediate response; with suitable additional logic, this action permits the host-system program to identify the entity detected, regardless of which constituent element was viewed by the pen.

Notes:

1. The flow of control of the buffer program can be made sensitive to the occurrence of deferred detects by the use of GTND and GTDD orders.

2. The GTND, GTDD, and GSRT orders and the Set Buffer Address Register and Start command cancel a deferred detect.

3. If a deferred detect is present when the response mode is changed to immediate, an I/O interruption is raised to the host system.

Control-Mode Orders

The control-mode orders are divided into three categories: basic control orders, transfer orders, and attribute register orders.

Basic Control Orders

Start Regeneration Timer - GSRT

The GSRT order synchronizes regeneration of the displayed image. GSRT also synchronizes input to the host system from the alphanumeric keyboard (END and CANCEL keys only) and the program function keyboard.

This 1-word order, initiating the regeneration timer and controlling switching between 3251s on a shared 3255, starts the regeneration sequence for each displayed image.

Byte 0	Byte 1
0100 1010	1000 0010

When decoded in an active buffer program, GSRT allows the 3255 to "disconnect" the one 3251 and to interrogate the other attached 3251(s) if present, to see if they are waiting to be "connected". A 3251 is said to be waiting when its program has been started by the host system issuing a Set Buffer Address Register and Start command and its regeneration timer has timed out.

When executed by the 3255, a GSRT order:

- Enables any outstanding keyboard interrupts to be raised, and initiates the setting of the manual input registers
- Sets default values in the attribute register (no-blink, solid line-type, and intensity level 5)
- Sets default values for the light-pen modes (switch-enabled and immediate)
- Interrogates and records the state of the light-pen tip-switch
- Initiates its regeneration timer
- Causes the 3255, if an alphanumeric character key was pressed, to enter the character code into the buffer location addressed by the cursor location register.

The regeneration timer is set to give an optimum regeneration rate. When a GSRT order is executed, its timer is initiated; when the buffer program encounters a GSRT order, the order is not executed if its timer is active, or if another buffer program is waiting to be executed. Thus, execution of the GSRT order is delayed if the previous regeneration took less than the minimum cycle time. Execution is restarted when both the minimum cycle time has elapsed and any other buffer program running in that 3255 has been serviced; the timer is then initiated.

A program that does not include a GSRT order executes continuously until interrupted. This continuous execution inhibits the operation of another display station attached to the same 3255, and prevents any interrupts being raised from the alphanumeric keyboard (END or CANCEL keys) or the program function keyboard.

Note: Great care should be taken if any buffer program is written without including a GSRT order. Normally, such a program does not cause any interrupt to be raised. If two or more 3251s share the 3255, and if the executing buffer program does not contain a GSRT order, the 3255 is unable to switch to the buffer program for the other 3251s.

End Order Sequence - GEOS

This 1-word order terminates execution of the buffer program and, hence, display regeneration for the associated 3251. An I/O interruption is raised to the host system.

Byte 0	Byte 1
0010 1010	1000 0001

Execution of GEOS sets the Attention and Unit Check bits in the status byte (see "Status Byte" in Chapter 4). In addition, the End Order Sequence bit is set in sense byte 1, and the display buffer address of the GEOS order incremented by 2 or 3 is saved in sense bytes 2 and 3 (the increment is 2 if the buffer address register contains an even address, 3 if odd). The change in status results in an I/O interruption being raised to the host system.

Store X,Y Position Registers - GSXY

This 2-word order stores the current contents of the X,Y position registers into two contiguous words of the display buffer. The second word of this order addresses the first byte of the two display-buffer words.

	Byte 0	Byte 1	
First Word	0010 1010	1110 1010	Hex 2AEA
Second Word	aaaa aaaa	aaaa aaaa	Storage Address

The format of the X and Y position data stored into the display buffer is:

	Byte 0		Byte 1		
First Word	0000	xxxx	xxxx	xx00	X Coordinate
Second Word	0000	yyyy	yyyy	yy00	Y Coordinate

If the cursor is associated with any byte of the addressed display-buffer words, it is removed.

No-Operation 2-Byte - GNOP2

This 1-word order performs no operation. GNOP2 would typically be used to reserve a word within the buffer program.

Byte 0		Byte 1		
0010	1010	1000	0000	Hex 2A80

No-Operation 4-Byte - GNOP4

This 2-word order performs no operation. The second word of the pair may contain any value whatsoever, including valid buffer orders, without affecting execution of the buffer program.

	Byte 0		Byte 1		
First Word	0010	1010	1100	0000	Hex 2AC0
Second Word	dddd	dddd	dddd	dddd	Data

GNOP4 might be used to shield a buffer order in its second word; for example, if the second word contained a graphic-mode order, the graphic-mode order would be ignored until the order code in the first word was replaced by a code such as GNOP2. If the word following the second word does not contain an order code, sequential words are addressed until an order is found; thus, the 2-word GNOP4 could be followed by a data list, which would be ignored if the GNOP4 order code remained unchanged.

Transfer Orders

Transfer orders provide a means of transferring control to a buffer order that is not the next sequential order. The transfer of control may be unconditional or conditional upon (1) the occurrence or nonoccurrence of a light-pen detection in the current regeneration cycle or (2) the state of the light-pen tip-switch at the start of the current regeneration cycle.

Transfer Unconditional - GTRU

The second word of this 2-word order contains a destination address. Execution of the order transfers control to this address.

	Byte 0		Byte 1		
First Word	0010	1010	1111	1111	Hex 2AFF
Second Word	aaaa	aaaa	aaaa	aaaa	Destination Address

To maintain regeneration of a display, a buffer program is normally written to start with a GSRT order and to end with a GTRU order back to the initial GSRT order.

GTRU can be used to link sections of a buffer program held in non-contiguous locations in the display buffer. Use of the Move Immediate Address (GMVA) order to modify the address field of a GTRU order can dynamically alter the logic of a program, or can permit a section of a program to be used as a subroutine.

Transfer on No Detect - GTND

The second word of this 2-word order code contains a destination address. Depending upon the current light pen detects mode and whether or not a light pen detection has occurred in the current regeneration cycle, this order transfers control to either the order at the destination address or the next sequential order.

	Byte 0		Byte 1		
First Word	0010	1010	1111	1101	Hex 2AFD
Second Word	aaaa	aaaa	aaaa	aaaa	Destination Address

The conditions tested by the GTND order and the actions taken if the conditions are satisfied are as follows:

- **Condition 1**: The following must all be satisfied for action 1 to be taken:

  - The detects mode is in the no-switch enabled state.
  - There is no outstanding deferred detect.

  Action 1: Control is transferred to the destination address.

- **Condition 2**: The following must all be satisfied for action 2 to be taken:

  - The detects mode is in the switch-enabled state.
  - At the start of the current regeneration cycle, the tip-switch was closed (that is, pressed).
  - During this regeneration cycle, no previous GTND order was executed while the detects mode was switch-enabled.
  - During this regeneration cycle, no light-pen detection occurred while the detects mode was switch-enabled.

  Action 2: Control is transferred to the destination address.

- **Condition 3**: There is an outstanding deferred detect.

  Action 3:

  - The deferred detect is canceled.
  - Control is not transferred to the destination address, but the next sequential order is accessed.

- **Condition 4**: None of the conditions 1, 2, and 3 are met.

  Action 4: Control is not transferred to the destination address, but the next sequential order is accessed.

Notes:

1. In no-switch-enabled state, more than one transfer of control in each regeneration cycle can result from the execution of GTND orders.

2. In switch-enabled state, only one transfer in each regeneration cycle can result from GTND orders. This is analogous to the single light-pen detection per regeneration cycle in switch-enabled state.

3. Provided that the tip-switch remains closed during several GSRT orders, one activation of the switch can satisfy condition 2 many times.

Transfer on Deferred Detect - GTDD

The second word of this 2-word order contains a destination address. If there is an outstanding deferred detect, executing this order (1) cancels the deferred detect, and (2) transfers control to the order at the destination address.

	Byte 0		Byte 1		
First Word	0010	1010	1111	1100	Hex 2AFC
Second Word	aaaa	aaaa	aaaa	aaaa	Destination Address

If there is no outstanding deferred detect, the order is executed as a No-Op 4-Byte (GNOP4) order.

Notes:

1. There can be an outstanding deferred detect only when the response mode is deferred.

2. Because GTDD cancels the deferred detect, a single detection cannot cause more than one transfer of control. If the light-pen detect mode is set to no-switch-enabled and the response mode is set to deferred, however, canceling of one deferred detect allows another to occur, and this may cause a subsequent GTDD order in the same regeneration cycle to transfer control.

Transfer on Switch Open - GTSO

This 2-word order contains a destination address in its second word. If the tip-switch was open (not pressed in) at the start of the current regeneration cycle, this order transfers control to the destination address. If the tip-switch was closed, this order is executed as a No-Op 4-Byte (GNOP4) order.

	Byte 0	Byte 1	
First Word	0010 1010	1111 0101	Hex 2AF5
Second Word	aaaa aaaa	aaaa aaaa	Destination Address

Note: Each GTSO order in the buffer program causes a transfer of control if the tip-switch was open at the start of the current regeneration cycle.

Move Immediate Orders

The Move Immediate Address (GMVA) and Move Immediate Data (GMVD) orders are of similar format Figure 3-4 and are executed in the same way. Move Immediate orders are 3-word orders; the second word contains a storage address and the contents of the third word are copied into the location addressed by the second word. The third word may contain any value; it may be an address, data, or a valid buffer order.

These orders can be used to make the buffer program modify itself by replacing order codes or by moving addresses or data within the program. For example, a section of the program can be used as a subroutine, if before transferring control to the subroutine, the final GTRU order is set to transfer back to the address following the point of invocation; for an example of a subroutine, see "Buffer Subroutines" in Chapter 5.

Although the GMVD and GMVA orders are identical in execution, their different order codes permit host-system programs to be written that are sensitive to their different intended uses. For example, a program that relocated a buffer order program from one buffer 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 GMVD order.

Move Immediate Address - GMVA

	Byte 0		Byte 1		
First Word	0010	1010	1110	1011	Hex 2AEB
Second Word	aaaa	aaaa	aaaa	aaaa	Storage Address
Third Word	dddd	dddd	dddd	dddd	Data (usually an address)

Move Immediate Data - GMVD

	Byte 0		Byte 1		
First Word	0010	1010	1110	1100	Hex 2AEC
Second Word	aaaa	aaaa	aaaa	aaaa	Storage Address
Third Word	dddd	dddd	dddd	dddd	Data

Figure 3-4. Move Immediate Orders

Attribute Register Orders

An attribute register is available to each 3251. During the processing of graphic-mode and character-mode data lists, the current contents of the attribute register affect the generation and display characteristics of the data. Fields in the attribute register control the following attributes:

- **Blink**: Components of the display that are generated while the Blink attribute is set to 1 blink on and off approximately twice a second.

- **Line Type**: This attribute governs the type of line generated by a graphic-mode vector order. Four line types are available: solid, dotted, dashed, and dot-dashed.

- **Intensity**: This attribute governs the relative intensity of displayed points, lines, and characters. Eight intensity levels (0 through 7) are available, from blank to brightest. In graphic-mode data lists, setting the blanking bit forces the intensity for that element to level 0.

- **Four-Level Intensity (3255 Model 2 only)**: Specific buffer orders can be executed to change the intensity level. These orders are described later in this chapter, and are summarized in Appendix E.

Execution of a Start Regeneration Timer (GSRT) order sets the attribute register to the default values of no-blink, solid line-type, and intensity level 5 (normal). However, the attribute register can be set and accessed, by the buffer program, with the Load Immediate Attribute Register (GLAR) order and the Store Attribute Register (GSAR) order.

Load Immediate Attribute Register - GLAR

This 2-word order sets the subfields of the attribute register for the 3251 as defined by the data in the second word of the order. If a validity bit is 1, then the corresponding subfield in the attribute register is set to the value defined by the 3 bits following the validity bit. If the validity bit is 0, the corresponding subfield in the attribute register is not changed.

	Byte 0	Byte 1	
First Word	0010 1010	1101 0001	Hex 2AD1
Second Word	0000 vaaa	vbbb vccc	Attribute Data

Legend:

v - Validity bit: if v=1, the three bits following the active validity bit are loaded into the attribute register.

a - Blink attribute:
aaa = 000, the display is steady (default value)
aaa = 001, the display blinks

b - Line-type attribute:
bbb = 000, solid lines (default value)
bbb = 001, dotted lines
bbb = 010, dashed lines
bbb = 011, dot-dashed lines

c - Intensity attribute:
ccc = 000, zero intensity (blank)
ccc = 101, normal intensity (default value)
ccc = 111, brightest

Intermediate values between 000 and 111 give intermediate levels of intensity.

When assigning the intensity attribute in the GLAR order, programmers should note the following:

- The brightness of a displayed image depends upon three variables:
  - The intensity attribute assigned by the buffer program to components of the image determines their intensity relative to each other.
  - The setting of the brightness control (at the front of the 3251) by the operator determines the intensity of the display and, thus, the range covered by the series 111 (brightest) through 000 (blank). In normal use, intensity level 011 (3) is the lowest level producing a visible image.
  - The number of times the image (or portion of it) is drawn in each refresh cycle.
- An operator may not be able to distinguish different intensities that are adjacent in the series 111 through 000 (for example, levels 101 and 110 may appear identical) or the lower intensity levels.
- Light-pen detections are inhibited for components displayed at intensity levels less than 101 (5).

Store Attribute Register - GSAR

This 2-word order stores the value of the attribute register into the buffer location addressed by the contents of the second word.

	Byte 0		Byte 1		
First Word	0010	1010	1101	0010	Hex 2AD2
Second Word	aaaa	aaaa	aaaa	aaaa	Storage Address

The attributes are stored in the form required if a subsequent GLAR order needs to restore the contents of the attribute register. The storage format is the same as that shown for the GLAR order, with the validity bits set to 1:

0000	1aaa	1bbb	1ccc

The GSAR order can be used in conjunction with a GLAR order to save and restore the current attribute register values around a section of the program that temporarily modifies the attribute register. The address field of GSAR would, typically, address the second word of the GLAR order that subsequently restores the attributes.

Four-Level Intensity (3255 Model 2 Only) - GSBL Orders

Four-Level Intensity is a subset of the eight intensity levels that are controlled by setting the attribute register. Refer to "Load Immediate Attribute Register - GLAR" earlier in this chapter. In this case, a specific buffer order can be executed to change the intensity level. Each of the following buffer orders sets a specific intensity level:

Buffer Order	3250 Intensity Attribute
GSBL(B) 2A90	Blank (Intensity level 0)
GSBL(D) 2A91	Dim (Intensity level 3)
GSBL(N) 2A92	Normal (Intensity level 5)
GSBL(B) 2A93	Bright (High Intensity level 7)

These buffer orders can be used in addition to, or in conjunction with, the intensity level controlled by the attribute register.

Notes:

1. When a Start Regeneration Timer (GSRT) order is executed, the intensity attribute is set to Normal as a default.
2. A displayed entity that is valid for light-pen selection, must be displayed at Normal or Bright intensity.
3. The Load Immediate Attribute Register (GLAR) order is not removed or changed by the four-level-intensity function.
4. Four-Level-Intensity is not supported either by the Graphics Access Method (GAM) component of the IBM OS/VS Programming Services (GPS) or by the IBM program product Graphics Access Method/System Product (GAM/SP).
5. The 3255 Display Control Unit Model 2 can be set to enhance vectors, points, or characters drawn at intensity level 7 (Bright). This improves operator discrimination between intensity level 5 (Normal), and intensity level 7 (Bright).

Chapter 4. Channel Operations

The 3250 system attaches to an I/O channel of its host system, and the channel interface performs as shown in *IBM/360 and System 370 I/O Interface Channel to Control Unit: Original Equipment Manufacturers' Information*, GA22-6974. This chapter summarizes: the operations on such a channel; the command codes accepted from the channel interface; data transfers between the host system and the 3250 system over the channel interface; and status and sense information provided by the 3250. For a full description of operations on a channel, refer to *IBM System/370 Principles of Operation*, GA22-7000.

Operations on the channel interface are governed by the channel program. A typical channel operation might be as follows:

1. The application program stores a channel program that consists of a sequence of channel command words (CCWs). The application program then causes a Start I/O instruction to be issued, containing the address of the required 3251 Display Station, to the channel.

2. The channel sends an address byte to all control units attached to the channel.

3. The 3258 Control Unit for the addressed 3251, if available, logically connects to the channel and returns its address to the channel.

4. The channel fetches the first CCW and sends the command code to the control unit. If data transfer is required, the channel also sets up the addressing of storage in the host system.

5. The control unit replies with a status byte to show whether the command is accepted.

6. If data transfer is required, the 3258 initiates the transfer of data between storage in the host system and the 3250 system.

7. At the completion of data transfer the 3250 sets the channel-end status bit and presents a status byte to the channel. This indicates that the device has finished with the channel facilities, following a control or data transfer operation. Depending on the command, device-end may be included with channel-end in the "ending" status byte. (See Appendix D for summary of channel commands.) Alternatively, device-end status may be provided later as an asynchronous I/O interrupt. Device-end indicates that the device has finished the operation and is ready to accept a new command.

8. If the "ending" status is satisfactory, and if CCW chaining is active, the channel fetches the next CCW at device-end time and sends another command code to the control unit; otherwise, the channel sets a channel status word (CSW) and raises an I/O interruption in the host system.

Initial Selection

Channel operations can be initiated by either the host system executing an I/O instruction or the 3250 system raising an I/O interruption to the host system.

Host-Initiated Operations

The host system initiates 3250 operations with a Start I/O (or Start I/O Fast Release) instruction that addresses the control unit (3258) and device (3251). In the host system, using a channel address word, the instruction causes a CCW to be fetched; the CCW contains the command code to be executed, the starting location in storage for any data transfer, and the number of bytes to be transferred.

The channel then attempts to select the addressed device by sending an address byte to all control units on the channel. The addressed 3258 logically connects to the channel and responds by returning its address byte to the channel program.

Each combination of control unit and device on the channel has a unique address. The address byte takes the following form:

0 — 4	5 — 7
Control Unit Address	Device Address

However, at installation time the 3258 is set to accept either a 5-bit control unit address as shown, or, if more than eight device addresses are required, a 4-bit control unit address (bits 0 through 3). Two or more 3258s may have the same control unit address providing that each device has a unique address. The addresses of devices attached to a control unit must be contiguous.

3250-Initiated Operations

When the 3251 requires the channel program to handle keyboard data, light-pen data, or a change in status of the buffer program, the 3258 raises an interrupt to the channel and prepares a status byte. Attention interrupts raised for keyboard input, do not affect program execution, whereas all other interrupts terminate the buffer program.

The interrupts are:

- **Attention** interrupts: These are raised by the execution of a Start Regeneration Timer (GSRT) order if keyboard data is waiting to be transferred to the host system. This keyboard input may be a code from the END or CANCEL key of the alphanumeric keyboard or a key code from the program function keyboard. If the buffer program has not been started or its execution has been terminated, keyboard data for the host system raises an Attention interrupt immediately the data is entered.

- **Attention and Unit Check** interrupts: These are raised when execution of the buffer program is terminated by:

  - Detection of a light-pen operation when detects are enabled and immediate

  - Execution of a Permit Detect Interrupt (GPDI) order if there is an outstanding deferred detect

- Execution of an End Order Sequence (GEOS) order
- A detected error condition.

In response to the interrupt, the channel transfers the status byte to the channel status word and raises an I/O interruption to the host system. The status byte is reset by the 3258 after the status information has been accepted by the channel.

Status Byte

In response to a channel command, the 3258 presents a status byte to the channel. The status byte contains indicators (Figure 4-1) that show whether or not the command is accepted for execution, and give a reason if the 3258 cannot accept the command.

All accepted commands receive an initial status of all-zeros, except for the following Immediate Control commands:

- Insert Cursor, Remove Cursor, Set Audible Alarm, and return "channel-end"
- Control No-operation returns both channel-end and device-end (bits 4 and 5).

If the 3258 has status pending for the addressed 3251, the response is a status byte with the pending status. For any command, except Test I/O, the busy bit is set in the status byte. If the command is not accepted because of an error condition, (that is, invalid command-codes) the unit-check bit alone is set. If the 3258 is busy, the response to all commands is a status byte with both busy and status pending set.

The 3258 is busy under the following conditions:

- Status pending for another 3251
- Between initial selection and device-end status for any 3251, whether the command is chained or not, or when Halt I/O is received
- When unit-check status has been accepted by the channel for another 3251 and no command, other than Test I/O or No-Op, has been subsequently addressed to that 3251
- During system reset and selective reset; the control unit "end" bit is included in these cases.

After the last data byte connected with a command is transferred over the channel interface, a control signal from the channel requests the 3258 to present "ending" status. After the last data byte, channel-end status is presented for Write Buffer, Read Buffer, Set Buffer Address Register and Stop, Set Buffer Address Register and Start, and Set Program Function Indicators. Device-end status is presented when the device has finished the operation.

Channel-end is presented with device-end at the completion of Read Manual Input, Read X,Y Position Registers, and Sense.

Channel-end is presented with device-end at the completion of Read Cursor if the cursor is found before the CCW byte count is exhausted. If the CCW count is exhausted before the cursor is found, channel-end and device-end are presented separately. Satisfactory "ending" status, if CCW chaining is active, allows the channel program to fetch the next CCW at "device-end" time; otherwise, "ending" status is used by the channel to set a channel status word and to raise an I/O interruption to the application program.

The meaning of each bit in the status byte, when that bit is active (that is, set to 1), is shown in Figure 4-1.

Sense Bytes

Sense data provides more detailed information about the condition of an addressed 3251 than does the status byte. The 3250 system holds four bytes of sense data: bytes 0 and 1 contain error and control information, and bytes 2 and 3 contain the current value of the buffer address register in the 3255 Display Control Unit. If the buffer program is executing when a Sense command is received, the buffer-running bit is set in sense byte 0 and the contents of bytes 2 and 3 are unpredictable.

The information contained in the sense bytes is reset by:

- A new status condition, if a Sense command was issued previously for data in the sense register
- The next command to the 3251 to which the Sense information applies except when the next command is a Sense, Test I/O, or Control No-Operation command
- Any command to another 3251 on the same 3258 with the exception of Test I/O or No-op, if a Sense command has been previously issued for the data present in the same registers
- An invalid command, or a command with bad parity, if there is no pending status or stacked status
- System reset
- Power on.

Further information on sense bytes is given later in this chapter under "Sense Command." The meaning of various combinations of status and sense bits is given in Appendix F.

Bit	Meaning When Active
Bit 0, Attention	Used in conjunction with bit 6 (Unit Check) as follows: • If bit 6=0, bit 0 is set to 1 to show that the interrupt was raised by action at the alphanumeric keyboard (END and CANCEL keys only) or at the program function keyboard. A Read Manual Input command must be issued, and the returned data examined, to identify the pressed key. • If bit 6=1, bit 0 is set to 1 to show that the interrupt was raised by the buffer program in response to a light pen detection or some other change in program status. A sense command must be issued, and the sense bytes examined, to identify the change in status.
Bit 1, Status Modifier	Used in conjunction with bit 3 (Busy). Bit 1 is set to 1 to show that another 3251 (that is, not the addressed one) requires use of the channel to service a change in status or that another 3251 has not yet completed execution of a command.
Bit 2, Control Unit End	Set to 1 to show that the current status clears a pending or stacked status which had been signaled by bits 3 and 1 (Busy and Status Modifier) being set in a previous status byte for another 3251 on the same 3258.
Bit 3, Busy	Set to 1 to show that a command cannot be accepted because an interrupt condition is waiting for service. Bit 3 is used in conjunction with bit 1 (Status Modifier) as follows: • If bit 1=0, bit 3 is set to 1 to show that the addressed 3251 requires the channel to service a change in status. • If bit 1=1, bit 3 is set to 1 to show that another 3251, addressed through the same 3258, is waiting to present a status byte or has not yet completed execution of a command.
Bit 4, Channel End	Set to 1 to show that the 3258 no longer requires use of the channel.
Bit 5, Device End	Set to 1 to show that the command has been executed and the 3251 is ready to accept another command.

Figure 4-1 (Part 1 of 2). Status Byte

Bit	Meaning When Active
Bit 6, Unit Check	Used in conjunction with bit 0 (Attention) to show a change in status of the 3250 system. A Sense command must be issued, and the sense bytes examined, to determine the change. • If bit 0=0, bit 6 is set to 1 to show an unusual condition in the 3250 system. (If a channel command was being executed when the condition arose, bit 5 is set. Bit 4 is also set if it was not previously presented to, and accepted by, the channel. • If bit 0=1, bit 6 is set to show a change in status of the buffer program (for example, a light pen detection causing an immediate interrupt).
Bit 7, Unit Exception	(Not used)

Figure 4-1 (Part 2 of 2). Status Byte

Channel Commands

Four types of channel commands are available: write, read, control, and sense.

- **Write Command**: The Write Buffer command initiates the transfer of data from storage in the host system to the display buffer.
- **Read Commands**: Four commands are available to transfer data from the display buffer or from defined registers to storage in the host system.
- **Control Commands**: Seven commands are available to control the operation of the selected 3251.
- **Sense Command**: This command transfers four sense bytes from the 3250 to the host system. The sense bytes show the current status of the addressed 3251 and its buffer program.

For further details, refer to "Write Buffer Command", "Read Commands", "Control Commands", and "Sense Command" later in this chapter.

Display Regeneration

Channel commands that access the display buffer or control registers can only be executed if execution of the buffer program for the addressed 3251 is stopped before the command is received. To stop the buffer program, a Set Buffer Address Register and Stop command would normally be issued. Command chaining might be used to issue the required commands after satisfactory "ending" status has been received for the Set Buffer Address Register and Stop command.

If execution of the buffer program is stopped, the application program must issue a Set Buffer Address Register and Start command in order to restart regeneration of the display.

The following channel commands require prior termination of the buffer program: Insert Cursor, Read Buffer, Read Cursor, Read X, Y Position Registers, Remove Cursor, and Write Buffer. If any of these, or other commands are issued to a 3251 while its buffer program is executing, the command is rejected; in this case, the unit-check bit (bit 6) of the status byte is set to 1 and both command-reject and buffer-running (bits 0 and 6 of sense byte 0) are set to 1.

The following channel commands do not require prior termination of the buffer program: Read Manual Input, Set Program Function Indicators, Set Audible Alarm, Set Buffer Address Register and Start, Set Buffer Address Register and Stop, Sense, and Control No-Operation.

Transmission Errors in the 3250 System

All transmissions of information between the 3258 and 3255, over the serial link, are tested for errors. If an error occurs, the 3250 system attempts to recover from the error. In the event of an unrecoverable transmission error, the current channel command is terminated with the device-end and unit-check bits (bits 5, and 6) set to 1 in the returned status byte, together with channel-end (bit 4) if it was not previously presented to and accepted by the channel. In addition, the data-check bit (bit 4) is set to 1 in the returned sense byte 0.

Write Buffer Command

Write Buffer is the only write command used with the 3250 system. Write Buffer transfers data, from storage in the host system, to the display buffer in the 3255 that controls the selected 3251.

To execute a Write Buffer command, the 3258 requests data bytes from storage in the host system and passes the data to the 3255, where it is written into sequential bytes of the display buffer. Writing starts at the byte addressed by the buffer address register and stops when the host system signals that there is no more data to be written. If the number of data bytes written exceeds the storage space above the initial address, buffer addressing wraps around from byte 32767 to continue from byte 0.

At the 3258, received data bytes are parity checked. Detection of a parity error does not terminate the write operation, but sets the unit-check bit in the status byte, when device-end is presented, and the bus-out-check bit in the sense bytes.

Execution of a Write Buffer command removes the cursor if data is written into the location addressed by the cursor location register.

At the end of data transfer channel-end status is presented to the channel, and device-end status is presented when the device is ready to accept another command.

Note: No parity correction is performed on data bytes with parity errors. If "ending" status shows unit-check status and the sense bytes show a bus-out check, the application program should rewrite the data in the buffer.

Read Commands

Read Buffer

The Read Buffer command transfers data, from the display buffer, to storage in the host system. To execute a Read Buffer command, the 3258 requests data from the display buffer in the 3255 and presents the data to the channel interface; the host system accepts this data from the interface and stores it. The number of bytes read is determined by the byte count in the current CCW (see Note). If the number of bytes read exceeds the space above the preset location, buffer addressing wraps around from byte 32767 to byte 0.

Note: For a read buffer command issued to a 3250 system, the CCW byte count must be less than 65280. If the byte count is equal to, or greater than, 65280, a data check occurs; the command terminates with channel-end, device-end, and unit-check status, and with the data-check bit set in sense byte 0.

If, during the execution of a Read Buffer command, the buffer address register addresses the same location as the cursor location register, the data contained in that location is transferred to the host system. The contents of the cursor location register are not affected when data is read from the addressed location.

At the end of data transfer, channel-end status is presented to the channel, and device-end status is presented when the device is ready to accept another command.

Read Cursor

The Read Cursor command, like the Read Buffer command, transfers data from the display buffer to storage in the host system. Execution of the Read Cursor command starts at the current buffer address; the buffer address register is incremented to address sequential locations until either (1) the buffer address register addresses the same location as addressed by the cursor location register or (2) the CCW byte count is satisfied. If the cursor location is found before the CCW byte count is satisfied, the "ending" status will be channel-end and device-end together. If the byte count is satisfied before the cursor is found, only channel-end will be presented; device-end will be presented when the device is ready to accept another command.

If, during the execution of a Read Cursor command, the buffer address register addresses the same location as the cursor location register, hexadecimal 1A is transferred to the host system irrespective of the data contained in the addressed location. The contents of the addressed location and the cursor location register are not affected by the execution of this command.

A Read Cursor command with the skip flag (bit 35) set in the CCW, followed by a Sense command, can be used to find the buffer address of the cursor. Read Cursor reads data, starting at the current buffer address and stopping at the cursor location; the data read is not transferred to the application program in the host system. A subsequent Sense command transfers the buffer address at which the Read Cursor command stopped; the returned address is for the byte that follows the byte addressed by the cursor location register.

Read Manual Input

The Read Manual Input command transfers 3 bytes of data from the manual input register to storage in the host system and sets the contents of the manual input register to hex 0000FF. This command is only valid if a keyboard is attached to the addressed 3251. Bytes transferred by the read manual input (RMI) command are shown in Figure 4-2.

Read Manual Input would usually be issued in response to an Attention interrupt raised by the 3250. This interrupt is raised to signal to the host system that the manual input register contains data for the application program. The data stored into this register is a keyboard identifier code and a key code; pressing either the END or CANCEL key on the alphanumeric keyboard or any of the 32 keys on the program function keyboard sets data into the manual input register and raises an I/O interruption to the host system. After the register has been set and the interruption raised, data sent to the manual input register is inhibited pending receipt of a Read Manual Input command.

A Read Manual Input command can be issued while the buffer program is executing, without affecting execution of the buffer program.

The "ending" status for this command contains both channel-end and device-end.

The 3 bytes that are read contain codes to identify the keyboard and the key.

Alphanumeric Keyboard

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 END key operation.
Bit 3 of byte 0 is set to 1 for a CANCEL key opetation.
All other bits of the 3 bytes are reset to 0.

Program Function Keyboard

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

Bit 1 of byte 0 is set to 1, and
the remaining bits of byte 0 are reset to 0.
Byte 1 contains the binary number of the pressed key.
(Bits 0, 1, and 2 of byte 1 are not used, and are reset to 0.)

Example:

Pressed Key	Key Code Byte 1
0	0000 0000 (hexadecimal 00)
1	0000 0001 (hexadecimal 01)
30	0001 1110 (hexadecimal 1E)
31	0001 1111 (hexadecimal 1F)

Byte 2 is set to the hexadecimal value FF.

Figure 4-2. Bytes Transferred by the Read Manual Input Command

Read X,Y Position Registers

The Read X,Y Position Registers command transfers the contents of the X,Y position registers for the selected 3251, to the host system. In response to the command, the 3258 returns four bytes, a 2-byte X coordinate, and a 2-byte Y coordinate, from the addressed 3251.

The format of the returned bytes is as follows:

	Byte 0	Byte 1	
First Word	0000 xxxx	xxxx xx00	X Coordinate
	Byte 0	**Byte 1**	
Second Word	0000 yyyy	yyyy yy00	Y Coordinate

The command may be issued in response to an interrupt raised for a light pen detection. In this case, the contents of the X, Y position registers depend upon the mode in which the detection occurred:

- For graphic mode, the returned coordinates are the absolute coordinates of the current beam position.
- For character mode, the returned coordinates are the absolute coordinates of the character center point following the character upon which the detection occurred.

If the command is not sent in response to a light pen-generated interrupt, the value of the X, Y position registers is returned, but the significance depends on the buffer program and the point at which it was stopped.

The "ending" status for this command contains both channel-end and device-end.

Control Commands

Control No-Operation

The Control No-Operation command performs no operation. Typically, the command would be used to obtain status information for the addressed 3251. In response to a Control No-Operation command, the 3258 supplies both channel-end and device-end.

Set Audible Alarm

The Set Audible Alarm command activates the audible alarm at the selected 3251, producing a short beep to attract the operator's attention. This command does not affect the status of the buffer program and may be issued while the buffer program is executing.

Channel-end is presented with initial status, and device-end is presented when the device is ready to accept another command.

Set Buffer Address Register and Start

The Set Buffer Address Register and Start command initiates execution of the buffer program for the selected 3251. Before the buffer program starts, the 3258 requests 2 bytes from the host system and transfers these bytes to the 3251's buffer address register. Execution of the buffer program, generating an image on the selected 3251, starts at the addressed location.

The supplied buffer address should be an even address in the range from 0 through 32766, and the addressed word should contain a buffer order. The low-order address bit is ignored, however, to force addressing storage on a word boundary. If a buffer order is not found at the start address, the 3255 searches sequentially through the display buffer for a location containing a buffer order.

Note: For a 3255 Model 2, bit 0 of the first byte of the address word is reserved, and should be set to zero.

A Set Buffer Address Register and Start command is valid when the buffer program is executing. In this case, execution stops, the new address is set, and execution restarts from the addressed location.

As a result of executing a Set Buffer Address Register and Start command, the buffer program is set with attributes of no-blink, solid line-type, and intensity level 5, and the light pen modes are set to switch-enabled and immediate.

Channel-end is presented after the transfer of the data, and device-end is presented when the device is ready to accept another command.

Set Buffer Address Register and Stop

The Set Buffer Address Register and Stop command terminates the buffer program for the selected 3251; the 3258 requests 2 bytes from the channel and transfers these bytes to the buffer address register of the selected 3251.

The supplied buffer address should be an address in the range from 0 through 32767. The low-order address bit is maintained in Set Buffer Address and Stop command.

3255 Model 2. Bit 0 of the first byte of the address word is reserved, and should be set to zero.

Channel-end is presented after the transfer of the data, and device-end is presented when the device is ready to accept another command.

Set Program Function Indicators

The Set Program Function Indicators command initiates the transfer of 4 bytes from the host system to the selected 3251. The back-lit keys of the program function keyboard are lit or extinguished according to the contents of the 4 bytes. Each bit of these bytes addresses a different key; if a bit is off (0), its associated lamp is switched off; if a bit is on (1), the lamp is switched on.

The program function keyboard has 32 keys, numbered left to right within each row, starting with 0 at the left of the top row. The bit positions for individual lamps are shown in Figure 4-3.

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

Figure 4-3. Bit Positions for Lamps of Program Function Keyboard

This command does not affect the status of the buffer program and may be issued while the buffer program is executing.

Channel-end is presented after the transfer of the data, and device-end is presented when the device is ready to accept another command.

Insert Cursor

The Insert Cursor command sets the cursor for the selected 3251 into the location currently addressed by the buffer address register. Upon receipt of this command, the 3255 copies the address from the selected 3251's buffer address register into that 3251s cursor location register. The contents of the addressed location are not affected by the Insert Cursor command. If a cursor is already set for the selected 3251, this command moves it to the addressed location. Once inserted, the cursor can be repositioned by alphanumeric keyboard action, or can be removed by either a Remove Cursor command or a Write Buffer command; the Write Buffer command writes data at the location addressed by the cursor.

Channel-end is presented with the "initial" status, and device-end is presented when the device is ready to accept another command.

Notes:

1. The cursor should be inserted into an unprotected character-mode data field. The cursor is only displayed in character mode, and is usually only significant to the operator if it identifies a position where an alphanumeric character can be entered. Display of the cursor is inhibited if the cursor location contains a null or overstrike character, but the cursor location is valid and alphanumeric data can replace that character.

2. The suppress-incorrect-length flag (bit 34) of the Insert Cursor CCW must be set to 1.

Remove Cursor

The Remove Cursor command clears the cursor for the selected 3251 if the location currently addressed by the buffer address register is the same location as that addressed by the cursor location register. The content of the addressed location is not affected by the Remove Cursor command.

Channel-end is presented with the "initial" status; and device-end is presented when the device is ready to accept another command.

Note: The suppress-incorrect-length flag (bit 34) of the Remove Cursor CCW must be set to 1.

Sense Command

The Sense command initiates the transfer of 4 sense bytes from the selected 3251 to the host system. Data contained in the sense bytes (Figure 4-4) shows the status of the buffer program, any error condition present, and the current value of the buffer address register. The Sense command does not affect execution of the buffer program, although this command would usually be issued in response to a unit-check status condition, and the unit check would have terminated execution of the buffer program.

If in response to a unit-check status from one 3251, a Sense command is issued to another 3251 attached to the same 3258 (by way of a 3255), the initial status response has both Busy (bit 3) and Status-Modifier (bit 1) set on. No bytes of sense data may be transferred until a Sense command is accepted for the 3251 causing the unit check, or the sense data associated with the unit check is reset - see conditions for reset under "Sense Bytes" earlier in this chapter.

At any other time, when no sense data is outstanding, 4 sense bytes are returned.

The "ending" status for this command contains both channel-end and device-end.

Byte	Meaning
Byte 0	
Bit 0, Command Reject	Set when an invalid sequence of commands is detected, when a command is received for a feature that is not fitted, or when a command contains an invalid modifier bit.
Bit 1 (Not used)	
Bit 2, Bus-Out Check	Set by a parity error in data received by the 3258 from the channel.
Bit 3 (Not used)	
Bit 4 Data Check	Set to show that an unrecoverable error on the link between the 3258 and the 3255 prevented a successful completion of a previously started operation.
Bit 5 (Not used)	
Bit 6 Buffer Running	Set when a 3251 is selected and its buffer program is executing.
Bit 7 (Not used)	
Byte 1	
Bit 0 Light Pen Detect	Set when a light-pen detection is raised and the response mode is immediate.
Bit 1 End Order Sequence	Set when the buffer program is terminated by executing an End Order Sequence (GEOS) order.
Bit 2 Character Mode	Set by a character-mode order, reset by any other order. Used in conjunction with bit 0 (Light-Pen Detect).
Bit 3 (Reserved)	
Bit 4 (Not used)	
Bit 5 (Not used)	
Bit 6 (Not used)	
Bit 7 (Not used)	

Figure 4-4 (Part 1 of 2). Sense Bytes

Byte	Meaning
Bytes 2 and 3	Contain the value of the buffer address register at which the buffer program terminated. (The buffer address occupies 15 bit positions; bit 0 of byte 2 is not used.) The value of bytes 2 and 3 is only defined for the following conditions: • When bit 0 of byte 1 (Light-Pen Detect) is set: If the buffer program is terminated by a light-pen detection, the application program must determine the conditions under which the detection raised an I/O interruption. (This may be deduced from the returned address and the known structure of the program, or it may be determined by reading the contents of the addressed location.) The possible conditions are: - Execution of a Permit Detect Interrupt (GPDI) order when there is an outstanding deferred detect. In this case, the returned address is the address of the order. - Occurrence of a light-pen detection on a character field and the response mode is immediate. In this case, the returned address identifies the character-mode data field corresponding to the selected character. - Occurrence of a light-pen detection on a graphic-mode point or vector and the response mode is immediate. In this case, the returned address identifies the graphic-mode data field containing the coordinates of the selected point or the end point of the selected line. The addressed word may contain either an X coordinate (absolute coordinates) or both X and Y coordinates (incremental coordinates), depending upon the preceding graphic-mode order. • When bit 1 of byte 1 (End Order Sequence) is set: Bytes 2 and 3 contain the address of the location immediately following the End Order Sequence (GEOS) order. • At the end of a Read Cursor command, bytes 2 and 3 contain the buffer address of the location immediately following (1) the location to which the cursor is assigned, or (2) the last location read before the command was terminated by the CCW byte count becoming zero.

Figure 4-4 (Part 2 of 2). Sense Bytes

I/O Instructions

The normal interaction between a 3250 system and its host system is controlled by the previously described channel commands. In addition to those commands, certain I/O instructions in the host system can issue a Test I/O command (hex 00) to the 3250 system.

I/O instructions contain an instruction code and channel/control-unit/device addresses. When an I/O instruction is decoded in the host system, the addressed device is selected and the function specified by the instruction code is performed. Details of I/O instructions and their function are contained in *IBM System/370 Principles of Operation*, GA22-7000. The functions performed by the I/O instructions are summarized below.

Clear I/O

The function performed by a Clear I/O instruction depends on the status of the processor channel. The 3258 will execute either a Halt I/O or a Test I/O instruction depending on the sequence received by the 3258 from the channel. Refer to publications *IBM System/370 Principles of Operation*, GA22-7000, and *IBM System/360 and System/370 I/O Interface Channel to Control Unit, Original Equipment Manufacturers' Information*, GA22-6974, for further details.

Halt Device

A Halt Device instruction causes the current operation with the addressed device to be terminated. Halt Device is similar to Halt I/O (see "Halt I/O" below), the differences being described in *IBM System/370 Principles of Operation*, GA22-7000.

Halt I/O

A Halt I/O instruction causes the channel to issue an interface disconnection sequence to the 3250, resulting in termination of the current I/O operation. This does not affect display regeneration that is currently in progress.

The interface disconnection sequence can be issued at various phases of interface activity including a data transfer. If the interface disconnection sequence is issued after initial status, and before channel-end is accepted by the channel, the 3250 disconnects. The 3250 will subsequently present one or more asynchronous interrupts with status bytes containing channel-end, device-end, and any error condition bits set, depending on the command in progress. If an interface disconnection is initiated at any other time, the interface disconnection sequence is executed with no other effect on the operation of the 3258. Any pending status is preserved and presented to the channel after the sequence has been completed.

Start I/O

A Start I/O instruction in the host system initiates a channel selection sequence (see "Channel Selection" earlier in this chapter); if the addressed device is available, Start I/O then initiates the operation defined in a channel command word (CCW). The CCW is addressed by the channel address word (CAW), the addressed CCW being usually the first of a chain of CCWs.

Start I/O Fast Release

Start I/O Fast Release instruction differs from Start I/O in that, depending upon conditions, the I/O operation may be initiated independently of the addressed device.

Test I/O

When the host system executes a Test I/O instruction and sends it to the 3258, the 3258 responds with a status byte. If no status information is outstanding for the addressed 3251 and none is pending for another 3251, an all-zeros status byte is returned. Any status for the addressed 3251 is transmitted to the channel and is then reset at the end of the sequence, but status for any other 3251 is held until that display station is addressed.

Chapter 5. 3250 Operations

This chapter contains an example of a buffer program that displays a simple image on the 3251 Display Station Model 1, and explains how the operator can modify the image using the light pen and keyboards. It also provides examples of more complex buffer programming techniques.

The example programs show the hexadecimal content of the display buffer and explain the content with mnemonics, labels, and operands styled after the macro instructions used with IBM OS/VS Graphic Programming Services (GPS). Vector coordinates and text characters are shown in a simple form.

More information on host programming support for the 3250 may be found in *OS/VS Graphic Programming Services (GPS) for IBM 2250 Display Unit and IBM 3250 Graphics Display System*, GC27-6971.

This chapter also gives the execution time for each of the buffer orders and for character-mode and vector-mode data fields. The figures given allow the programmer to calculate the total execution time of a buffer program, and estimate the effect of that program on the regeneration rate of the displayed image.

Buffer Program Examples

The following examples show:

- A buffer program generating a simple image on the 3251
- The use of the light pen, in conjunction with the program function keyboard, to delete a component of the displayed image
- The use of the alphanumeric keyboard to enter data into the displayed image.

Associated with each example is a summary of the channel commands and status sequences. System/370 programming is discussed only in general terms to describe the use of channel commands with specific examples.

Displaying an Image on the 3251

Figure 5-1 shows a simple image generated on the screen of a 3251 Display Station Model 1. This image contains a box of dotted lines, and the characters BOX NAME followed by one space and a field of six unprotected character positions where the operator may enter a name. The buffer program that generates the image is shown in Figure 5-2.

The first order in the buffer program is a Start Regeneration Timer (GSRT) order, labeled T in Figure 5-2. This order enables the regeneration timer, setting the optimum regeneration rate, and sets default values to the attributes and the light-pen-mode controls. A GSRT order should be included in each buffer program. Without a GSRT order, the operator normally cannot use the keyboards or light pen to interrupt the application program in the host system, and the 3255 Display Control Unit cannot interrupt the buffer program to service the other 3251s (if more than one 3251 shares the display buffer).

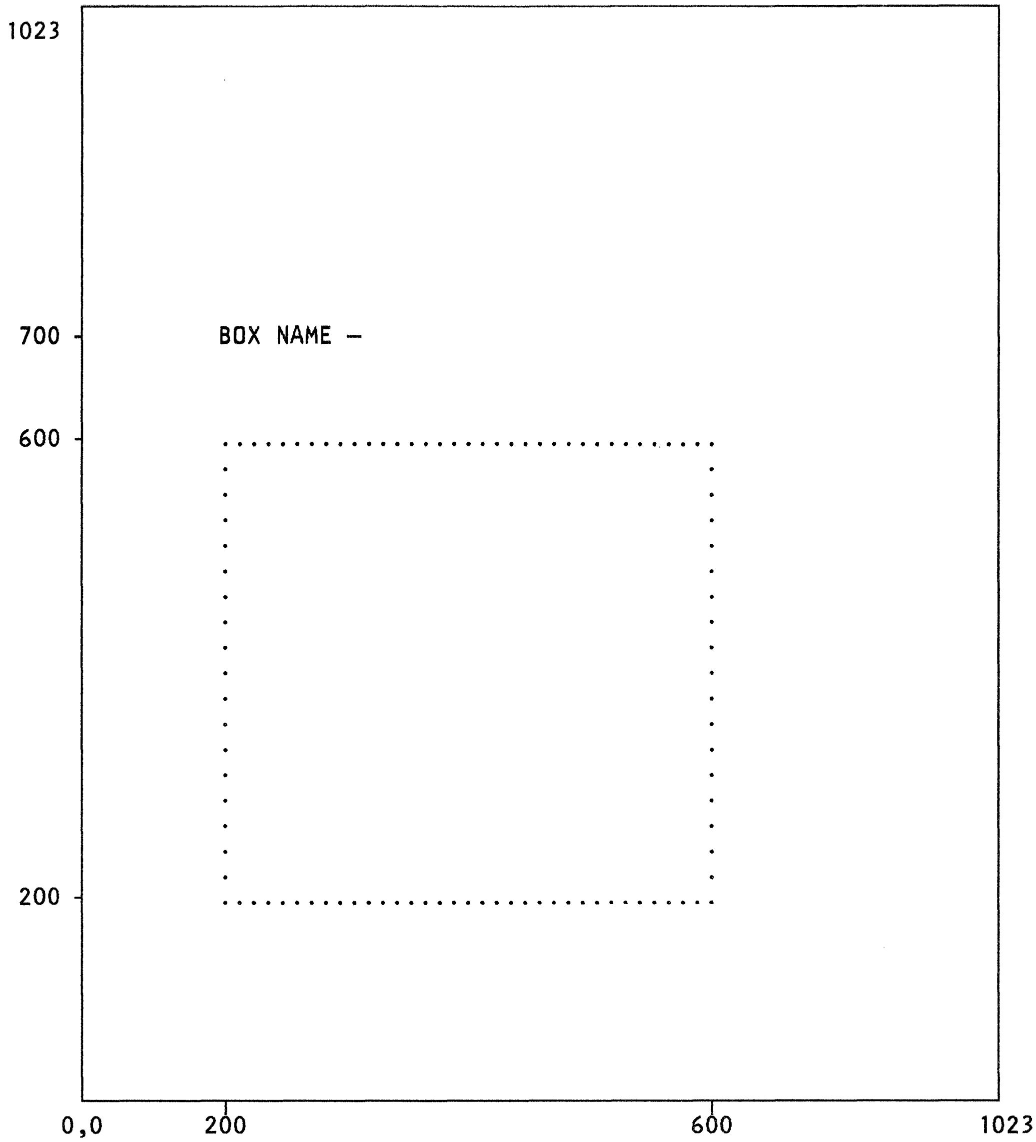

Figure 5-1. A Simple Image Generated by the Buffer Program

Buffer Location	Contents	Label	Buffer Program Mnemonic	Operands	Comment
0000	2A82	T	GSRT		Start regeneration timer.
0002	2AD10090	U	GLAR	DOTTED	Load attribute register, specifying dotted vectors; (intensity level 5 and no-blink, set by default during GSRT)
0006	2A02		GEVM		Enter graphic mode, absolute vectors.
0008	43200320		VECTOR	200,200,BLANK	Blanked deflection vector to position (200,200) establishing starting point for drawing box.
000C	09600320		VECTOR	600,200,UNBL	Unblanked vector forming bottom horizontal line.
0010	09600960		VECTOR	600,600,UNBL	Right vertical line.
0014	03200960		VECTOR	200,600,UNBL	Top horizontal line.
0018	03200320		VECTOR	200,200,UNBL	Left vertical line, completing box.
001C	43200AF0		VECTOR	200,700,BLANK	Blanked beam move to position (200,700) establishing starting point for plotting characters.
0020	2A45	V	GECP	LARGE	Enter character mode protected large.
0022	C2D6E740		TEXT	'BOXƀ'	The word BOX is followed by a space.
0026	D5C1D4C540		TEXT	'NAMEƀ'	The word NAME is followed by a space.
002B	00		NULL	1	A null character to align the next order on an even byte, since the preceding text has an odd number of characters.
002C	2A40	W	GECF	BASIC	Enter character mode fixed (unprotected) basic.
002E	404040404040		TEXT	'ƀƀƀƀƀƀ'	Six blank spaces for operator to key in text.
0034	2AFF0000	X	GTRU	T	Transfer unconditional to T (location 0), regenerating the display.

Legend: ƀ means space

Figure 5-2. Buffer Program to Generate a Simple Image

Starting at label U in Figure 5-2 is the group of orders that generates the box of dotted lines. The line-type attribute is set to dotted lines, and an Enter Graphic Mode Absolute Vector (GEVM) order sets the mode to allow the following data fields to draw a box. A blanked positioning vector moves the beam to the X,Y coordinates of a corner of the box, and four unblanked vectors draw the box. After drawing the box, a blanked vector moves the electron beam to the starting point of the alphanumeric character field.

Label V identifies the Enter Character Mode Large Protected (GECP) order that sets character mode. The data field following the GECP order generates BOXbNAMEb (where b represents a space on the screen); the center of the B is the end point of the previous blanked vector. The space character following NAME causes the electron beam to move to the second character position after the generated E. Because the character field contains an odd number of bytes, a single null character is placed at the end of the field; this aligns the next order on a word boundary.

An unprotected character field is provided in the displayed image by the Enter Character Mode Basic (GECF) order at label W. The data field for this order consists of six spaces; six basic-size character positions are reserved in the image area by these spaces. Because this field is unprotected and all six characters are displayable, the operator can position the cursor at any of the reserved character positions (null characters are not displayable, space characters are).

The following sequence of channel commands is issued by the application program:

1. Set Buffer Address Register and Stop - address hex 0000: This command stops the buffer program, if one is running, for the selected 3251. The command then sets the associated buffer address register to 0000, thus defining the first location of the buffer area into which data is to be written.

2. Write Buffer: This command causes the transfer of the buffer program (see Figure 5-2) from the channel to consecutive locations in the display buffer, starting at location 0000.

3. Set Buffer Addresss Register and Stop - address hex 002E: This command sets the buffer address register to 002E.

4. Insert Cursor: This command copies the contents of the buffer address register (hex 002E) into the cursor location register. When the buffer program is started, the processing of alphanumeric data in the location addressed by the cursor location register causes a cursor to be displayed beneath the screen position occupied by the generated character; in this example, the generated character is a space.

5. Set Buffer Address Register and Start - address hex 0000: This command sets the buffer address register of the selected 3251 to 0000 and starts execution of the buffer program. The image is generated according to the buffer orders shown in Figure 5-2. Because this command is the last in the chain, the 3258 Channel Control Unit disconnects from the channel, and the channel is free for use by other devices.

Light Pen Detect/Program Function Keyboard Operation

This example shows how the operator may use the program function keyboard and the light pen to delete a line from the image generated by the buffer program shown in Figure 5-2. The operator uses the program function keyboard to signal the delete operation to the host application program. The light-pen modes used in this example are the default modes set by the execution of a Start Regeneration Timer (GSRT) order; the default modes are switch-enabled and immediate.

The sequence of events is as follows:

1. The operator presses the program function key that requests the delete function in the application program, thereby raising an I/O interruption with the attention status bit set to 1.

2. The application program issues a Read Manual Input command.

3. Responding to the command, the 3255 Display Control Unit returns 3 bytes of data: the first byte contains a code identifying the program function keyboard, the second byte contains the binary code identifying the key that was pressed, and the third byte contains hex FF. After transferring the 3 bytes, the 3258 returns ending status with the channel-end and device-end bits set to 1.

4. The application program examines the returned data and interprets it as a request for a delete operation to be performed on the component next selected by the light pen. The program then waits for a light-pen detection.

5. The operator places the tip of the light pen on the line to be deleted and closes the light-pen tip-switch. This action raises an I/O interruption with the attention and unit-check status bits set to 1.

6. The application program issues a Sense command to determine the cause of the returned status.

7. Responding to the Sense command, the 3255 returns 4 sense bytes, which define the cause of the I/O interruption. In this example, the light-pen-detect bit is set to 1 and the third and fourth sense bytes contain the address of the first byte of the ending coordinates for the selected line. Thus, if the operator selects the bottom horizontal line of the box shown in Figure 5-1, then the buffer address hex 000C is returned to the channel. After 4 bytes have been transferred, the 3258 returns ending status with the channel-end and device-end bits set to 1.

8. The application program determines, from the sense data, the line to be deleted. One method the program can use to delete the line is to retransmit the byte in location hex 000C with the blanking bit set to 1. By changing the line to a blanked vector, the programmer suppresses the line generated while the beam moves to the start point of the right-hand vertical line of the box. The following sequence of commands and data could be sent to the 3255 to blank the selected line:

   - Set Buffer Address Register and Stop - address hex 000C

   - Write buffer - 1 byte of data containing hex 49 (the first byte of the absolute X coordinate with the blanking bit set to 1)

   - Set Buffer Address Register and Start - address hex 0000.

Alphanumeric Keyboard Operation

This example shows the sequence of events occurring when the operator uses the alphanumeric keyboard to enter data. The buffer program is that shown in Figure 5-2, and the operator is going to insert a name into the six-digit unprotected character field.

The sequence of events is as follows:

1. The operator observes the cursor to determine where the first character will be inserted. The cursor is displayed beneath the second space to the right of E in the word NAME, see Figure 5-1 and Figure 5-2. (The cursor location register was set to location hex 002E by the Insert Cursor command.)

2. The operator presses an alphanumeric character; this may be a character key or the space bar. The 1-byte EBCDIC code for the selected character is stored in the location addressed by the cursor, and that character is displayed. The cursor location register is then incremented to address the next byte in the display buffer, and the displayed cursor moves to the next character position on the display.

3. The operator inserts a box name containing up to six characters. The codes of the selected characters are stored in buffer locations hex 002E through 0033 and the characters are displayed on the screen. Inserting a character into the last location of the field does not advance the cursor; the cursor remains in that location until it is moved either by the application program or by the operator pressing BACKSPACE, ADVANCE, or JUMP on the alphanumeric keyboard. Pressing BACKSPACE decrements the character location register to move the displayed cursor one character position backward. Pressing ADVANCE increments the cursor one character position forward. Pressing JUMP advances the cursor to the first character position in the next unprotected alphanumeric field; in this example there is only one such field, and the effect of pressing JUMP is to force the cursor back to the start of the field (location hex 002E).

4. The operator checks the name and makes changes, as necessary, by positioning the cursor (with the BACKSPACE, ADVANCE, and/or JUMP key) and selecting another character.

5. When the desired name is displayed, the operator presses the END key. This action raises an I/O interruption with the attention status bit set to 1.

6. The application program issues a Read Manual Input command.

7. Responding to the command, the 3255 Display Control returns 3 bytes of data; the first byte identifies the alphanumeric keyboard and the END key, and the remaining bytes contain hex 00. After 3 bytes have been transferred, the 3258 Channel Control Unit returns ending status with the channel-end and device-end bits set to 1.

8. The application program examines the returned data and interprets it as a signal that the operator has updated an alphanumeric character field. The following sequence of commands might then be issued to retrieve the updated field from the display buffer:

   - Set Buffer Address Register and Stop - address hex 002E

   - Read Buffer - specifying a byte count of 6 to the channel.

The application program has now received the text entered at the alphanumeric keyboard by the operator.

Buffer Dump Facility - OS/VS Graphic Programming Services

When the application program in the host system is using IBM OS/VS Graphic Programming Services (GPS) with basic attention handling, the user at the 3251 may cause a dump to be taken of that part of the 3255 buffer which is assigned to him. In order to obtain a buffer dump, a device or dataset with ddname SYSBFDMP must be available. To obtain a buffer dump:

1. Press the CANCEL key (with CONTINUOUS, if a 3255 Model 2) on the alphanumeric keyboard, causing a "termination option" menu to replace the current display.

2. Select BUFFER DUMP with the light pen.

When the buffer dump has been completed, selecting RESUME causes the original display to reappear on the screen.

Note: For buffer dumping from a 3255 Model 2, refer also to "Cancel Key Attentions" in *OS/VS Graphic Programming Services (GPS) for IBM 2250 Display Unit and IBM 3250 Graphics Display System*, GC27-6971.

Operating System Considerations

It is necessary to define each buffer seperately to the operating system.

VS1 and MVS

When running under VS1 or MVS, graphics applications are supported by the *OS/VS Graphic Programming Services (GPS) for IBM 2250 Display Unit and IBM 3250 Graphics Display System* component of the operating system. In this instance, each buffer is seen as a separate physical control unit (PCU). In order to use the buffers, the IODEVICE macro instructions that describe the devices for system generation must must specify different PCU operands for the devices that use the different buffers.

VM/CMS

When running under VM/CMS, graphics applications are supported by the *IBM Graphic Access Method/System Product* (GAM/SP). To obtain support for the buffers, the GABDEV macro instructions that define the devices should be associated with different GABBUF macro instructions.

Programming Techniques

This section describes some programming techniques that can be used with the 3250 system. Sample buffer programs are included to illustrate these techniques.

Entities

In general, data consists of vectors, points, and characters; these may be grouped into **entities**. The 3250 buffer orders allow considerable flexibility in the way these entities may be linked to form a complete buffer program, and in the response of the program to the operator selecting an entity with the light pen.

Figure 5-3 shows a sample entity (a horizontal line and a small rotated label) and its associated control mode orders, embedded in a larger buffer program. The characteristics of the program allow the complete entity to be displayed at increased brightness if any part of that entity is selected with the light pen while the tip-switch is open. When the operator presses the light pen against the screen to close the tip-switch, the light-pen detection raises an I/O interruption, with the name of the selected entity saved in a location known to the host application program. The increased brightness of the selected entity is achieved by the entity being regenerated twice in each regeneration cycle; this regenerating is not the same as the automatic brightening of the portion of the image viewed by the light-sensor, nor does it use the intensity setting in the attribute register.

The starting point for the buffer program is the Start Regeneration Timer (GSRT) order labeled S. Each regeneration cycle starts at this location. The next two orders are Defer Response to Detects (GDRD) and Enable No-Switch Detect (GENSD); these set the light pen modes to allow a detection to occur when a component of the image is viewed by the light-sensor without the tip-switch being closed, and any detect that occurs is saved for future interrogation. This example tests for deferred detections after each entity; thus selection of any component of an entity selects the complete entity. The testing for detections occurring in the no-switch-enabled mode does not reset the tip-switch condition recorded by the GSRT order; any number of no-switch-enabled detections may occur in a single regeneration cycle.

After the program sets the initial modes, the first entity is generated on the display screen. This entity, labeled V, comprises a blanked vector to position the beam, an unblanked vector to generate a line, and a rotated character field to name the generated line. If the operator holds the light pen near this entity, a deferred detection is recorded. After generating the entity, the buffer program issues a Transfer on Deferred Detect (GTDD) order before passing control to the next entity.

If a deferred detection is outstanding, execution of the GTDD order transfers control to the Enable Switch Detect (GESD) order at label X. The group of orders starting at label X sets the light pen modes to switch-enabled and immediate, and moves the entity name into a fixed location (U) before transferring control to V to regenerate the selected entity.

During this second generation of the entity, light-pen detections occur only if the tip-switch is closed. If no detection occurs, the next execution of the GTDD order will not transfer control, and the subsequent Transfer Unconditional (GTRU) order will cause a transfer to the next entity (label W). However, if the tip-switch is closed, execution of the buffer program is stopped when the electron beam passes the tip of the light pen, and an I/O interruption is raised to the host system.

Buffer Location	Contents	Label	Buffer Program Mnemonic	Operands	Comment
0000	0000	U	NULL	2	Location for saving entity name.
0002	2A82	S	GSRT		Start regeneration timer.
0004	2A83		GDRD		Defer response to light-pen detects.
0006	2A86		GENSD		Enable no-switch detects.
0008	2A02	V	GEVM		
000A	43E803E8		VECTOR	250,250,BLANK	Sequence of orders, vectors,
000E	05DC05DC		VECTOR	375,375,UNBL	and characters that together
0012	2A4E		GECP	SMALL,ROTATED	constitute an entity.
0014	D3C9D5C57BF1		TEXT	'LINE#1'	
001A	2AFC0022		GTDD	X	Transfer to enable switch routine (X) if detect.
001E	2AFF0030		GTRU	W	Otherwise transfer to next entity (W).
0022	2A84	X	GESD		Enable switch detects.
0024	2A87		GPDI		Permit detect interrupt.
0026	2AEC0000D3F1		GMVD	U,DATA=C'L1'	Move characters 'L1' into program-specified location (U) for saving entity name.
002C	2AFF0008		GTRU	V	Transfer unconditional.
0030	2A83	W	GDRD		Orders for generating next entity.
	2AFF0002		GTRU	S	Final transfer back to start regeneration again at S.

Figure 5-3. Example of a Programmed Entity

Buffer Subroutines

A sequence of buffer orders that may be repeated several times, either in the same or different buffer programs, may be provided as a subroutine. Examples of such sequences are the generation and display of a tracking symbol and a routine to flood the display with characters.

The example in Figure 5-4 shows an entity that can be generated many times in the same regeneration cycle, with each generation starting from different absolute coordinates. The subroutine, labeled T, generates a small square with its bottom left-hand corner located at the beam position that is current when the subroutine is invoked. In this example, the subroutine is stored in buffer locations hex 0200 through 021D, and it is invoked by the program executing the orders Move Immediate Address (GMVA) and Transfer Unconditional (GTRU) at locations hex 0400 and 0406.

Before invoking the subroutine, the buffer program must establish a return address and may need to establish the beam position with an absolute blanked vector. To establish a return address, the GMVA order at location hex 0400 moves the address of label R into the destination field of the GTRU order at the start of the subroutine; the return address could be inserted into the transfer order at the end of the subroutine, but the caller would then need to know the length of the subroutine as well as its starting address. (Any changes made to the buffer subroutine by the systems programmer would then require reprogramming of the buffer program by the application programmer.) The GTRU order, following GMVA, transfers control to the second order in the subroutine (at offset 4).

The subroutine contains a Load Immediate Attribute Register (GLAR) order to set the same attributes each time it is executed. However, the Store Attribute Register (GSAR) order and the Load Immediate Attribute Register (GLAR) order at label S are used to save and subsequently restore the attributes current when the subroutine was invoked.

Buffer Location	Contents	Label	Buffer Program Mnemonic	Operands	Comment
0200	2AFF0000	T	GTRU	0	Transfer unconditional - Target address will be completed before invoking this subroutine.
0204	2AD20218		GSAR	S+2	Save attribute register below, ready for reloading later.
0208	2AD1088F		GLAR	7,NOBLINK,SOLID	Load attribute register, intensity 7, no blinking, and solid lines.
020C	2A05		GEVI2		Enter vector mode, incremental.
020E	7F00		VECTOR	63,0,UNBL	Bottom of square.
0210	017E		VECTOR	0,63,UNBL	Right side of square.
0212	8300		VECTOR	-63,0,UNBL	Top of square.
0214	0182		VECTOR	0,-63,UNBL	Left side of square.
0216	2AD10008	S	GLAR	0	Load attribute register with the value stored by the GSAR order above.
021A	2AFF0200		GTRU	T	Transfer unconditional to the return address that is loaded at the start of the subroutine.
. .	. .	. .			Orders to position the beam in the required place.
0400	2AEB0202040A		GMVA	T+2,ADDR=R	Move address of R into location 202 (hex).
0406	2AFF0204		GTRU	T+4	Transfer unconditional to buffer subroutine.
040A	2A02	R	GEVM		Other buffer orders.

Figure 5-4. Example of Buffer Subroutine

Light-Pen Search

A simple form of graphic input is to touch the light pen on the screen, indicating the selection of a coordinate pair. The operator presses the tip of the light pen on the screen, closing the tip-switch, to request that the 3255 reads the position of the light pen.

To enable selection of any point in the image area, the buffer program must scan the entire image area with the electron beam intensity set to level 5, 6, or 7. An incremental graphic-mode order or a character-mode order can be used to generate a raster scan of the image area. For this operation, the light-pen modes would normally be set to switch-enabled and immediate; this allows a light-pen detection to stop the buffer program and to raise an I/O interruption to the host system. In response to the interruption, the application program can issue a Read X, Y Position Registers command to determine the coordinates of the point at which the light-pen detection occurred.

The example shown in Figure 5-5 is a buffer program that floods the screen with "W" characters. Each regeneration cycle writes 1,792 large characters, sufficient to fill the image area. To ensure that new lines are forced correctly, an initial Enter Graphic Mode Absolute Vector (GEVM) order establishes absolute coordinates; the desired wrapping from line to line does not occur if the most recent graphic-mode order used incremental coordinates. The Move Immediate Address (GMVA) and Transfer Unconditional (GTRU) orders are used to modify the flow of control in the buffer program so that a form of controlled looping takes place. This method is more economical of buffer space than one using a very long character string in the buffer.

```
Label  Mnemonic   Operands
  A    GSRT
       GEVM
       GESD
       GPDI
       GTRU       K
  B    GMVA       P+2,BADDR=C
       GTRU       K
  C    GMVA       P+2,BADDR=D
       GTRU       K
  D    GMVA       P+2,BADDR=Q
       GTRU       K
  E    GMVA       N+2,BADDR=F
       GTRU       K
  F    GMVA       N+2,BADDR=G
       GTRU       K
  G    GMVA       N+2,BADDR=O
       GTRU       K
  H    GMVA       L+2,BADDR=I
       GTRU       K
  I    GMVA       L+2,BADDR=J
       GTRU       K
  J    GMVA       L+2,BADDR=M
  K    GECP       LARGE
       TEXT       'WWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWW'
  L    GTRU       H
  M    GMVA       L+2,BADDR=H
  N    GTRU       E
  O    GMVA       N+2,BADDR=E
  P    GTRU       B
  Q    GMVA       P+2,BADDR=B
       GTRU       A
```

Figure 5-5. Example of a Flooding Routine

Light-Pen Tracking

Many graphic applications require feedback to the user. This can be accomplished by light-pen tracking with a buffer program that displays a pattern at the current light pen position. Important elements of light-pen tracking are:

- A displayed tracking symbol, such as a small box, centered on the last detected light pen position
- Determining the position of the light pen relative to the tracking symbol
- Updating the reference point of the tracking symbol to reflect any change in the position of the light pen
- Storing the coordinates of the new reference point for future use (perhaps as input to the application program).

If the initial light pen position is not known, a light-pen search could be used to establish the start point before a tracking subroutine is invoked.

A sample tracking subroutine is shown in Figure 5-6. Linkage to the subroutine is as described under "Buffer Subroutines" earlier in this section. The first order of the sample routine holds a return address, and E is the entry point to the subroutine. On exit from the subroutine, offset hex 12 contains the current coordinates of the symbol's reference point.

In the example shown, the subroutine establishes no-switch-enabled and deferred detects as the light-pen modes. Each component of the tracking symbol is followed by a Transfer on Deferred Detect (GTDD) order. If the light pen is detected on a component of the tracking symbol, the GTDD order transfers control to the Store X, Y Position Registers (GSXY) order. Execution of the GSXY order stores the coordinates of the current beam position at an addressed location; in this example, the current beam position is the end point of the vector selected by the light pen, the absolute coordinates of the end point replace the absolute coordinates of the reference point for the tracking symbol.

When the tracking subroutine is next invoked (for instance, on the next regeneration cycle), the symbol is generated around the new reference point.

Buffer Location	Contents	Label	Buffer Program Mnemonic	Operands	Comment
1000	2AFF0000	T	GTRU	0	Return address.
1004	2AD21406	E	GSAR	R+2	Save attribute register.
1008	2AD1088D		GLAR	5,NOBLINK,SOLID	
100C	2A86		GENSD		
100E	2A83		GDRD		
1010	2A00		GEPM		
1012	08000800	P	VECTOR	512,512,UNBL	Holds current center point of tracking symbol.
1016	2AFC1400		GTDD	S	
					Orders to draw the tracking symbol would follow. The whole symbol might consist of three concentric squares composed of many incremental vectors. Here, just one such vector is shown.
1234	2A05		GEV12		
1236	0110		VECTOR	0,+8,UNBL	
1238	2AFC1400		GTDD	S	
1400	2AEA1012	S	GSXY	P	Store X, Y registers.
1404	2AD10008	R	GLAR	0	Restore attribute register.
1408	2AFF1000		GTRU	T	To exit point.

Figure 5-6. Example of a Light Pen Tracking Subroutine

Use of No-Operation Orders

The most common use of the No-Operation orders is in a switch. The switch may contain one of the transfer orders, and replacement of the transfer order by a No-Operation order can be used to change the flow of the buffer program.

The buffer program shown in Figure 5-7 uses such a switch, the Transfer Unconditional (GTRU) order at label Y being changed into a No-Operation order if a light-pen detection occurs on entity M (the text "DISPLAY ENTITY D"). Without a light-pen detection, the buffer program transfers control from entity M to entity N. If a light pen-detection occurs, the order at label Y is changed to No-Op 4-Byte (GNOP4), and control passes to the next sequential order at entity D, thereby displaying "ENTITY D". This example does not restore the GTRU order; once the switch has been changed, entity D remains in the flow of the buffer program, even when the light pen is removed.

Buffer Location	Contents	Label	Buffer Program Mnemonic	Operands	Comment
0000	2A82	S	GSRT		
0002	2A84		GESD		
0004	2A83		GDRD		
0006	2A02	M	GEVM		
0008	41F407D0		VECTOR	125,500,BLANK	Positioning vector
000C	2A44		GECP	BASIC	
000E	C4C9. . . etc		TEXT	'DISPLAY ENTITY D'	
001E	2AFC0026		GTDD	X	
0022	2AFF002C		GTRU	Y	
0026	2AEC002C2AC0	X	GMVD	Y,DATA=X'2AC0'	Move GNOP4 code
002C	2AFF0040	Y	GTRU	N	
0030	2A02	D	GEVM		
0032	4BB80BB8		VECTOR	750,750,BLANK	
0036	2A44		GECP	BASIC	
0038	C5. . . etc		TEXT	'ENTITY D'	
0040		N	next entity		
	2AFF0000		GTRU	S	

Figure 5-7. Example of a Program Switch

Buffer Orders Execution Time - 3255 Display Control Unit Model 1

The image displayed at the 3251 Display Station Model 1 is generated by repeated execution of a buffer program. Each execution generates one transient image. By repeatedly executing the buffer program, the image is constantly regenerated to make it visible.

To maintain an apparently constant intensity level, regeneration must occur at a fixed rate. The frequency of regeneration is controlled by the regeneration timer. If the total execution time of the buffer program is less than 21.7 milliseconds, the regeneration timer synchronizes regeneration to a rate of 46 cycles per second (cps). If total execution time exceeds 21.7 milliseconds, the rate of regeneration depends upon the execution time; in this case, adding data to (or removing data from) the display changes the regeneration rate. Where two 3251s are connected to a common 3255 Display Control Unit, they share the display buffer, and both buffer programs should be executed within 21.7 milliseconds if the optimum regeneration rate is to be maintained.

Figure 5-8 lists the execution times for all the buffer orders and for graphic-mode and character-mode data. When using Figure 5-8 to calculate the total execution time of a buffer program, the following points must be considered:

- **Overhead:** To allow for contention for access to buffer storage and polling in the 3255, an overhead must be added to the calculated execution times for complete images. Typically, this overhead is from 1 to 3 percent and rarely exceeds 4 percent.

- **Graphic-Mode Data:** The expression given for graphic-mode data is for one field of data, that is, one movement of the beam to an addressable point. This expression contains two components: (1) a base depending upon whether the graphic mode uses point-plotting or vector-plotting, and (2) a variable depending upon whether the beam is blanked or unblanked and upon the distance between the current beam position and the addressed point or line-end point. The variable component relates to distances in millimeters. (The distance between adjacent points on a common axis is about 0.3 millimeter (0.012 inch).)

- **Character-Mode Data:** The figure given for character-mode data is for one character. This figure is the sum of two components: (1) a base execution time and (2) a variable depending upon the size of the generated character. To simplify the variable, a weighted average of 14 strokes per character is used; this weighted average is the average number of strokes per character in English-language text. The time taken to move the beam to a new-line position depends upon the current beam position relative to the start of the new line; this time is the same as that taken to move a blanked vector through the same distance.

	Time (microseconds)
Character-Mode Orders	
GECF	6.3μs
GECP	6.0μs
Character-Mode Data	
Small Characters	3.6μs per character (space=2.1μs, null=1.5μs)
Basic Characters	4.2μs per character (space=2.1μs, null=1.5μs)
Medium Characters	4.8μs per character (space=2.1μs, null=1.5μs)
Large Characters	5.4μs per character (space=2.1μs, null=1.5μs)
Control-Mode Orders	
GEOS	1.8μs
GLAR	6.6μs
GMVA	3.9μs
GMVD	3.9μs
GNOP2	1.2μs
GNOP4	1.5μs
GSAR	3.6μs
GSRT	750.0μs
GSXY	4.8μs
GTRU	2.4μs
Graphic-Mode Orders	
GEPI2	4.2μs (light pen enabled) or 5.7μs (light pen disabled)
GEPM	3.9μs (light pen enabled) or 5.4μs (light pen disabled)
GEVI2	4.2μs (light pen enabled) or 5.1μs (light pen disabled)
GEVM	3.9μs (light pen enabled) or 4.8μs (light pen disabled)

Figure 5-8 (Part 1 of 2). Execution Time of Buffer Orders (3255 Model 1)

	Time (microseconds)	
Graphic-Mode Data, Point Plotting (see Note)		
(1) Light Pen Enabled		
Draw (Unblanked)	3.1 + (0.116 x distance [millimeters]) µs	(minimum=4.5µs)
Move (Blanked)	3.1 + (0.075 x distance [millimeters]) µs	(minimum=4.5µs)
(2) Light Pen Disabled		
Draw (Unblanked)	1.6 + (0.116 x distance [millimeters]) µs	(minimum=3.0µs)
Move (Blanked)	1.6 + (0.075 x distance [millimeters]) µs	(minimum=3.0µs)
Graphic-Mode Data, Vector Plotting (see Note)		
(1) Light Pen Enabled		
Draw (Unblanked)	2.0 + (0.116 x distance [millimeters]) µs	(minimum=3.3µs)
Move (Blanked)	2.0 + (0.075 x distance [millimeters]) µs	(minimum=3.3µs)
(2) Light Pen Disabled		
Draw (Unblanked)	1.1 + (0.116 x distance [millimeters]) µs	(minimum=2.4µs)
Move (Blanked)	1.1 + (0.075 x distance [millimeters]) µs	(minimum=2.4µs)
Light-Pen-Mode Orders		
GDPD	3.0µs	
GDRD	3.0µs	
GENSD	3.3µs	
GESD	3.0µs	
GPDI	3.3µs	
GTDD	4.5µs	
GTND	7.2µs	
GTSO	3.0µs	

Note: *Execution time for graphic-mode data must be rounded up to the next higher increment of 0.3 microsecond.*

Figure 5-8 (Part 2 of 2). Execution Time of Buffer Orders (3255 Model 1)

Buffer Orders Execution Time - 3255 Display Control Unit Model 2

The image displayed at the 3251 Display Station Model 1 is generated by repeated execution of a buffer program. Each execution generates one transient image. By repeatedly executing the buffer program, the image is constantly regenerated to make it visible.

To maintain an apparently constant intensity level, regeneration must occur at a fixed rate. The frequency of regeneration is controlled by the regeneration timer. If the total execution time of the buffer program is less than 21.7 milliseconds, the regeneration timer synchronizes regeneration to a rate of 46 cycles per second (cps). If total execution time exceeds 21.7 milliseconds, the rate of regeneration depends upon the execution time; in this case, adding data to (or removing data from) the display changes the regeneration rate.

Where two or more 3251 Display Stations Model 1 are connected to a common 3255 Display Control Unit Model 2, two 3251s share one buffer, and the third 3251 has exclusive use of the other buffer. All buffer programs should be executed within 21.7 milliseconds if the optimum regeneration rate is to be maintained.

Figure 5-9 lists the execution times for all the buffer orders and for graphic-mode and character-mode data. When using Figure 5-9 to calculate the total execution time of a buffer program, the following points must be considered:

- **Overhead:** To allow for contention for access to buffer storage and polling in the 3255, an overhead must be added to the calculated execution times for complete images. Typically, this overhead is from 1 to 3 percent and rarely exceeds 4 percent.

- **Graphic-Mode Data:** The expression given for graphic-mode data is for one field of data, that is, one movement of the beam to an addressable point. This expression contains two components: (1) a base depending upon whether the graphic mode uses point-plotting or vector-plotting, and (2) a variable depending upon whether the beam is blanked or unblanked and upon the distance between the current beam position and the addressed point or line-end point. The variable component relates to distances in millimeters. (The distance between adjacent points on a common axis is about 0.3 millimeter (0.012 inch).)

- **Character-Mode Data:** The figure given for character-mode data is for one character. This figure is the sum of two components: (1) a base execution time and (2) a variable depending upon the size of the generated character. To simplify the variable, a weighted average of 14 strokes per character is used; this weighted average is the average number of strokes per character in English-language text. The time taken to move the beam to a new-line position depends upon the current beam position relative to the start of the new line; this time is the same as that taken to move a blanked vector through the same distance.

- **Graphic-Mode Data:** The expression given for graphic-mode data is for one field of data, that is, one movement of the beam to an addressable point. This expression contains two components: (1) a base depending upon whether the graphic mode uses point-plotting or vector-plotting, and (2) a variable depending upon whether the beam is blanked or unblanked and upon the distance between the current beam position and the addressed point or line-end point. The variable component relates to distances in millimeters. (The distance between adjacent points on a common axis is about 0.3 millimeter (0.012 inch).)

- **Character-Mode Data:** The figure given for character-mode data is for one character. This figure is the sum of two components: (1) a base execution time and (2) a variable depending upon the size of the generated character. To simplify the variable, a weighted average of 14 strokes per character is used; this weighted average is the average number of strokes per character in English-language text. The time taken to move the beam to a new-line position depends upon the current beam position relative to the start of the new line; this time is the same as that taken to move a blanked vector through the same distance.

	Time (microseconds)
Character-Mode Orders	
GECF	3.4
GECP	3.4
Character-Mode Data	
Small Characters	3.4 per character(space= 2.0 , null= 1.0)
Basic Characters	4.0 per character(space= 2.0 , null= 1.0)
Medium Characters	4.6 per character(space= 2.0 , null= 1.0)
Large Characters	5.2 per character(space= 2.0 , null= 1.0)
Control-Mode Orders	
GEOS	16.0 (worst-case estimate)
GLAR	4.4
GMVA	3.6
GMVD	3.6
GNOP2	1.0
GNOP4	1.2
GSAR	3.0
GSBL(B)	2.4
GSBL(D)	2.4
GSBL(N)	2.4
GSBL(B)	2.4
GSRT	340.0 (two or more displays)
GSXY	4.0
GTRU	2.0
Graphic-Mode Orders	
(1) Light Pen Enabled (with SEI set on)	
GEPI2	3.4
GEPM	3.5
GEVI2	3.4
GEVM	3.5
(2) No-Switch Enabled Immediate-Mode (with SEI set on or off)	
GEPI2	3.4
GEPM	3.5
GEVI2	3.4
GEVM	3.5
(3) Any other Mode	
GEPI2	5.1
GEPM	5.1
GEVI2	4.5
GEVM	4.5

Figure 5-9 (Part 1 of 2). Execution Time of Buffer Orders (3255 Model 2)

	Time (microseconds)
Graphic-Mode Data Point Plotting (see note) (1) Light Pen Enabled: (with SEI set on) Draw (Unblanked) Move (Blanked) (2) No-Switch Enabled Immediate-Mode: (with SEI set on or off) Draw (Unblanked) Move (Blanked) (3) Light Pen Disabled Draw (Unblanked) Move (Blanked)	 2.6 + (0.075 x L) (minimum= 3.9) 1.1 + (0.075 x L) (minimum= 1.8) 2.6 + (0.075 x L) (minimum= 3.9) 1.1 + (0.075 x L) (minimum= 1.8) 1.5 + (0.075 x L) (minimum= 2.4) 1.1 + (0.075 x L) (minimum= 1.8)
Graphic-Mode Data Vector Plotting (see note) (1) Light Pen Enabled: (with SEI set on) Draw (Unblanked) Move (Blanked) (2) No-Switch Enabled Immediate-Mode: (with SEI set on or off) Draw (Unblanked) Move (Blanked) (3) Light Pen Disabled Draw (Unblanked) Move (Blanked)	 2.0 + (0.075 x L) (minimum= 2.7) 1.1 + (0.075 x L) (minimum= 1.8) 2.0 + (0.075 x L) (minimum= 2.7) 1.1 + (0.075 x L) (minimum= 1.8) 1.1 + (0.075 x L) (minimum= 1.8) 1.1 + (0.075 x L) (minimum= 1.8)
Light-Pen-Mode Orders GDPD GDRD GENSD GESD GPDI GTDD GTND GTSO	 2.8 2.8 3.0 2.8 3.0 1.6 1.8 2.2

Note: Execution time for a graphic mode order must be rounded-up to the next higher increment of 0.2 microsecond.

Figure 5-9 (Part 2 of 2). Execution Time of Buffer Orders (3255 Model 2)

Image Content Performance - (3255 Model 1)

The 3250 system has a timed maximum regeneration rate of 46 cycles per second (cps). This rate allows 21.7 milliseconds in which to execute the buffer program (or both buffer programs if two 3251 Display Stations Model 1 share a common 3255 Display Control Unit Model 1. Figure 5-10 shows the total length of displayed vectors, and Figure 5-11 shows the number of characters that can be generated, for each attached 3251, in 21.7 milliseconds. The length and numbers quoted are theoretical; time is allowed to execute the minimum number of control-mode orders, and in some cases the quoted number of characters exceeds the number that can be displayed without overwriting in the image area.

In calculating the number of displayable characters per 3251, an English-language message was used; the time included the execution times of one character-mode order, one GSRT order, and one transfer order, and the time taken to reposition the beam at the start of each text line.

		Number of 3251s Connected to One 3255			
		One		Two	
Coordinate Mode	Vector Length	Number of Vectors	Total Length	Number of Vectors	Total Length
Incremental	10mm	8440	84m	4090	40m
	15mm	6960	104m	3380	50m
	19mm	6330	120m	3075	58m
Absolute	15mm	6960	104m	3380	50m
	25mm	4975	124m	2415	60m
	75mm	2110	158m	1025	76m
	150mm	1120	168m	545	81m
	300mm	580	174m	280	84m

Figure 5-10. Maximum Number of Vectors per 3251 Attached to a 3255 Model 1. (Regeneration Rate of 46 cps)

	Number of 3251s Connected to One 3255	
	One	Two
Small Characters	5050	2460
Basic Characters	4615	2240
Medium Characters	4000	1940
Large Characters	3545	1720

Note: The theoretical number of small, basic, medium, and large characters for one 3251 using the display buffer exceeds the capacity of the image area. The maximum number of characters that can be displayed without overwriting character positions is:

Small characters	7004
Basic characters	3848
Medium characters	2337
Large characters	1715

Figure 5-11. Maximum Number of Characters per 3251 Attached to a 3255 Model 1. (Regeneration Rate of 46 cps)

Image Content Performance - 3255 Model 2

The 3250 System has a timed maximum regeneration rate of 46 cycles per second (cps). This rate allows 21.7 milliseconds in which to execute the buffer program (or programs if more than one 3251 Display Station Model 1 shares the same 3255 Display Control Unit Model 2.

Figures 5-12 and 5-13 show the total length of displayed vectors, and Figure 5-14 shows the number of characters that can be generated for each attached 3251 in 21.7 milliseconds. The length and numbers quoted are theoretical; time is allowed to execute the minimum number of control-mode orders, and in some cases, the quoted number of characters exceeds the number that can be displayed without overwriting in the image area.

In calculating the number of displayable characters per 3251 Model 1, an English language message was used; the time included the execution times of one character-mode order, one GSRT order, and one transfer order, and the time taken to reposition the beam at the start of a text line.

Note: The improvement in image content performance provided by the 3255 Display Control Unit Model 2, compared to the 3255 Model 1, is approximately halved if the single element intensification function (SEI) is enabled.

		Number of 3251s attached to one 3255 Model 2					
		One		Two		Three	
Coordinate Mode	Vector Length	Number of Vectors	Total Length	Number of Vectors	Total Length	Number of Vectors	Total Length
Incremental	10mm	7230	72.3m	3595	35.9m	2350	23.5m
	15mm	6350	95.2m	3160	47.4m	2065	31.0m
	19mm	5990	113.8m	2980	56.6m	1945	37.0m
Absolute	15mm	6350	95.2m	3160	47.4m	2065	31.0m
	25mm	5375	134.4m	2675	66.9m	1745	43.6m
	75mm	2720	204.0m	1350	101.2m	885	66.4m
	150mm	1575	236.2m	780	117.0m	510	76.5m
	200mm	1225	245.0m	610	122.0m	395	79.0m

Note:
Quoted performance with (1) light pen enabled and SEI enabled, or (2) light pen in immediate mode with no-switch enabled.

Figure 5-12. Maximum Number of Vectors per 3251 Attached to a 3255 Model 2. (Regeneration rate of 46 cps)

		Number of 3251s attached to one 3255 Model 2					
		One		Two		Three	
Coordinate Mode	Vector Length	Number of Vectors	Total Length	Number of Vectors	Total Length	Number of Vectors	Total Length
Incremental	10mm	10480	104.8m	5215	52.1m	3410	34.1m
	15mm	8735	131.0m	4345	65.2m	2840	42.6m
	19mm	8060	153.1m	4010	76.2m	2620	49.8m
Absolute	15mm	8735	131.0m	4345	65.2m	2840	42.6m
	25mm	6985	174.6m	3475	86.9m	2270	56.7m
	75mm	3080	231.0m	1530	114.7m	1000	75.0m
	150mm	1690	253.5m	840	126.0m	550	82.5m
	200mm	1290	258.0m	640	128.0m	420	84.0m

Note:
Quoted performance with (1) light pen disabled, or (2) light pen enabled, and SEI disabled.

Figure 5-13. Maximum Number of Vectors per 3251 Attached to a 3255 Model 2. (Regeneration rate of 46 cps)

	Number of 3251s Attached to one 3255 Model 2		
	One	Two	Three
Small Characters	5580	2776	1815
Basic Characters	4683	2330	1523
Medium Characters	4030	2005	1311
Large Characters	3555	1769	1156

Note: The theoretical number of small, basic, medium, and large characters for one 3251 Model 1 using the display buffer exceeds the capacity of the image area. The maximum number of characters that can be displayed without overwriting character positions is:

Small characters	7004
Basic characters	3848
Medium characters	2337
Large characters	1715

Figure 5-14. Maximum Number of Characters per 3251 Attached to a 3255 Model 2. (Regeneration Rate of 46 cps)

Chapter 6. Operator Controls

IBM 3251 Display Station Model 1

The operator controls are on the front and rear panels of the display unit. The front panel has the operator controls, and the back panel has the test controls and the status indicators.

Display Unit Controls

Front Panel

The front panel of the display unit has three controls: the power on/off switch, the brightness control, and the focus control Figure 6-1). The power switch is marked 0 and | ; when set to | , power is applied to the 3251. The brightness control is rotated fully clockwise for maximum brightness and full counterclockwise to blank the display. The focus control can be rotated clockwise or counterclockwise to adjust the sharpness of the displayed image.

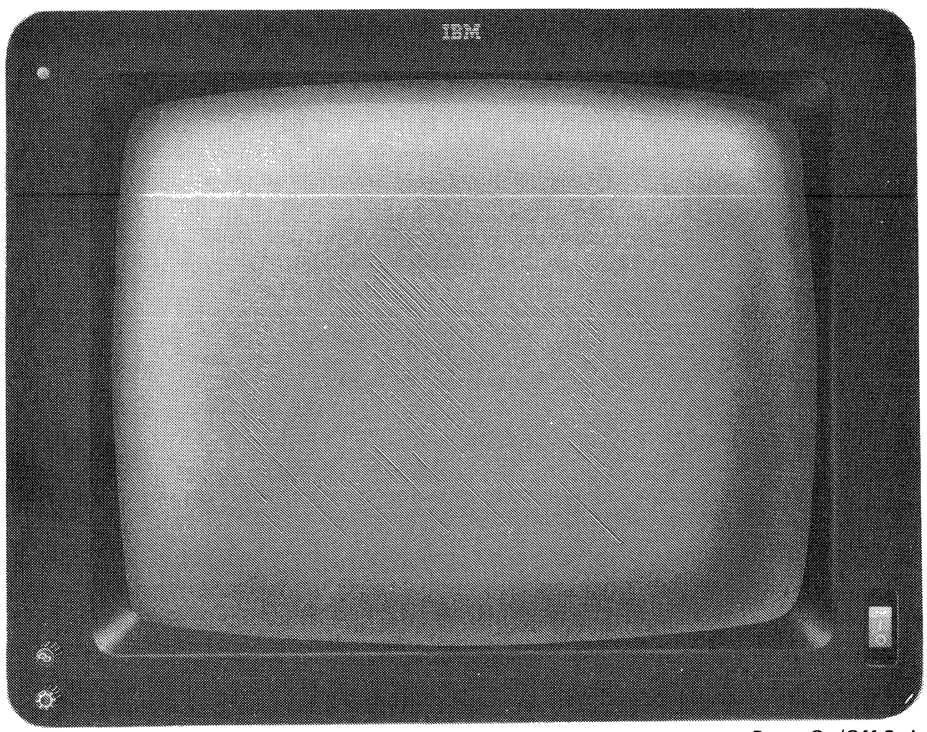

Figure 6-1. Display Unit Front Panel Controls

Rear Panel

The rear panel of the display unit (Figure 6-2) has three status lights and a test switch. If the 3251 has power on and if all three lights are lit, pressing the switch causes a circle to be displayed on the screen. The test switch is a locking pushbutton; to remove the test circle and reconnect to the 3255 Display Control Unit, the switch must be pressed again.

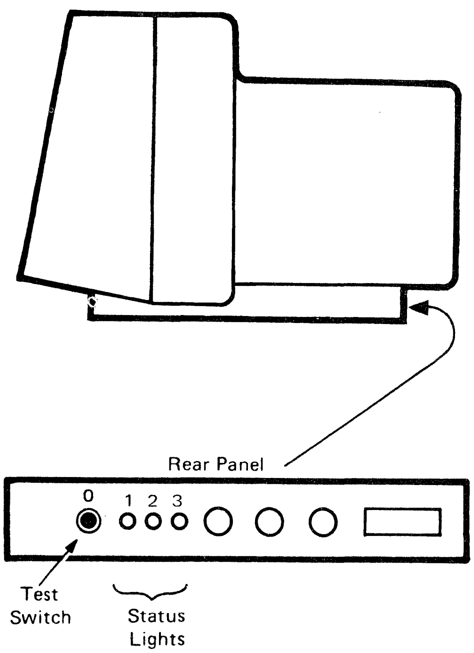

Figure 6-2. Display Unit Back Panel Controls

Alphanumeric Keyboard - Special Keys

The alphanumeric keyboard is a typewriter-like keyboard from which the operator can compose and modify messages and annotations on the displayed image. A cursor is displayed beneath a character or character position to show where the next alphanumeric character entered will be displayed.

The keyboard has, in addition to the keys for lowercase and uppercase alphanumeric characters, 13 keys whose operation give a special effect:

End: This raises an interruption to the host system. The interruption may, for example, signal the end of data entry at the alphanumeric keyboard. Interpretation of an END key operation is governed by the application program, and it could be interpreted in other ways.

Cancel: Like the END key, CANCEL raises an interruption to the host system, with the subsequent response being determined by the application program.

Note: Systems with the 3255 Display Control Unit Model 2 require the CONTINUOUS key to be held pressed while the CANCEL key is used.

Jump: This moves the cursor forward, from its current position, to the first character position of the next field in the buffer program that can accept keyboard data. If the display does not have such a field, pressing JUMP inhibits further keyboard action. (Pressing RESET restores the keyboard.)

Advance: This moves the cursor forward one character position each time the key is pressed, without affecting the displayed character. If the cursor is at the last character position of a keyboard data field, it does not advance.

Backspace: This moves the cursor backward one character position each time the key is pressed, without affecting the displayed character. If the cursor is at the first character position of a keyboard data field, it does not backspace.

Continuous: When this key is held down, the following keys operate as repeat-action (typematic) keys: ADVANCE, BACKSPACE, SPACE, NULL, and all alphanumeric character keys. Repeat-action mode ends when the CONTINUOUS key is released.

Shift: The keyboard has two SHIFT keys. When either key is pressed, uppercase alphanumeric characters are selected. Pressing and releasing either key also cancels SHIFT LOCK.

Shift Lock: This locks the keyboard in uppercase shift. The key can be reset by pressing and releasing either SHIFT key.

Overstrike: This enters a character code into the data field and backspaces the cursor. The result of a character-OVERSTRIKE-character sequence is to display the second character superimposed upon the first. The cursor is not displayed if it is located at an overstruck character.

Reset: This restores the keyboard after a keying error that has blocked (input inhibited) further keying operations from being accepted. RESET does not clear an END key or CANCEL key operation; these must be cleared by a Read Manual Input command from the application program. When input is inhibited, the keyboard clicker does not sound when a key is pressed. (Note, however, that the clicker can also be stopped by pressing the CLICK key.)

Click: This two-action key controls the noise (click) made when any key is pressed. Pressing CLICK once stops the clicker; pressing it again restores the clicker.

Null: This enters a null character code into the alphanumeric data field, replacing any character at that location. Because a character space is not generated for a null character, using NULL to replace a space or a displayable character causes a left-shift of the character after the null. Similarly, if a displayable character replaces a null, there is a right-shift of characters after the replaced null. The cursor is not displayed if it is located at a null character.

Numeric Keypad Alphanumeric Keyboard

An 87-key alphanumeric keyboard with an extended engineering symbols character set and a numeric keypad is available for 3251's attached to a 3255 Display Control Unit Model 2 as a special feature. The 3251 must be equipped with the Extended Keyboard Support specify feature. The keyboard is functionally identical with the 75-key alphanumeric keyboard, described above , except that the keys of the numeric keypad and the special engineering symbol keys produce the same character codes in both upper case and lower case shift.

Program Function Keyboard Usage

The program function keyboard contains 32 back-lit keys mounted on a sloping fascia panel. The function given to individual keys is determined by the application program in the host system with which the display station is working. When a key is pressed, an I/O interruption is raised to the host system, which should then issue a Read Manual Input command to determine which key was pressed. Illumination of a key is independent of its operation; for example, the application may be programmed to light keys that are valid for selection, light the last selected key, or light special keys to show the status of the program.

Light Pen Controls

The light pen has two controls: the tip-switch and the light-sensor. Pressing the tip of the light pen against a hard surface, such as the display screen, closes the switch and generates a signal to the 3251. Pointing the tip of the light pen at a component of the displayed image that is sufficiently bright allows the electron beam to excite the light-sensor; the signal generated by the light-sensor is synchronized to the movement of the electron beam passing the tip of the pen.

Light-pen modes set by the buffer program determine whether any action is taken in response to the signals generated.

IBM 3255 Display Control Unit (Models 1 and 2)

Power is switched on and off by means of a switch located on the front panel of the 3255. The power switch is marked 0 and | ; when set to | , power is applied to the 3255.

IBM 3258 Channel Control Unit

Power is switched on and off by means of a switch located on the front panel of the 3258. The power switch is marked 0 and 1; when set to 1, power is applied to the 3258.

In addition to the power switch, the 3258 is not operational until it is online to the host system. An ONLINE/OFFLINE switch is located at the right-hand side of the control panel. To select online operation, open the hinged front door and set the ONLINE/OFFLINE switch to the ONLINE position. When online operation is selected, the 3258 executes some self-test programs before connecting to the host system. If the tests are satisfactory, the ONLINE indicator lights to show the online status.

Appendix A. Absolute Coordinates - Hexadecimal

The tables in Figure A-1 give the three-character hexadecimal values for absolute coordinates in the range from 0 through 1023. The hexadecimal values have been adjusted to the format of the data fields following graphic-mode absolute point and absolute vector orders. This format ignores the two low-order bits and gives, in effect, a 4096 x 4096 precision grid compressed into the 1024 x 1024 addressable points of the image area.

Pairs of absolute coordinates occupy two words. The first word contains the blanking bit and the X coordinate, and the second word contains the Y coordinate. To translate a pair of absolute coordinates into two words of four hexadecimal digits:

1. Set the first digit of the first word to 0 for an unblanked point or vector, or to 4 for a blanked point or vector.

2. Set the remaining three digits of the first word to the hexadecimal value obtained for the X coordinate.

3. Set the first digit of the second word to 0.

4. Set the remaining three digits of the second word to the hexadecimal value obtained for the Y coordinate.

The resulting two words form a valid data field for a list following a graphic-mode absolute point or absolute vector order.

Example: The current order is an absolute vector order. To move the beam from the current beam position to the point addressed by the coordinates X=512 and Y=216, without displaying the line, set as shown below.

```
First digit of first word = 4
X coordinate (512) = hexadecimal 800
First digit of second word = 0
Y coordinate (216) = hexadecimal 360
```

	Byte 0	Byte 1	
First Word	0 1 0 0 1 0 0 0	0 0 0 0 0 0 0 0	Hex 4800
Second Word	0 0 0 0 0 0 1 1	0 1 1 0 0 0 0 0	Hex 0360

The values given in Figure A-1 show the two low-order bits set to 0. Because these bits are ignored, they can be set to any value. Thus, a decimal coordinate of 100 is shown as hexadecimal 190, but hexadecimal 191, 192, and 193 address the same coordinate in the image area.

Decimal Coordinates 0 through 299

	Units	0	1	2	3	4	5	6	7	8	9
Hundreds	0	000	004	008	00C	010	014	018	01C	020	024
and Tens	1	028	02C	030	034	038	03C	040	044	048	04C
	2	050	054	058	05C	060	064	068	06C	070	074
	3	078	07C	080	084	088	08C	090	094	098	09C
	4	0A0	0A4	0A8	0AC	0B0	0B4	0B8	0BC	0C0	0C4
	5	0C8	0CC	0D0	0D4	0D8	0DC	0E0	0E4	0E8	0EC
	6	0F0	0F4	0F8	0FC	100	104	108	10C	110	114
	7	118	11C	120	124	128	12C	130	134	138	13C
	8	140	144	148	14C	150	154	158	15C	160	164
	9	168	16C	170	174	178	17C	180	184	188	18C
	10	190	194	198	19C	1A0	1A4	1A8	1AC	1B0	1B4
	11	1B8	1BC	1C0	1C4	1C8	1CC	1D0	1D4	1D8	1DC
	12	1E0	1E4	1E8	1EC	1F0	1F4	1F8	1FC	200	204
	13	208	20C	210	214	218	21C	220	224	228	22C
	14	230	234	238	23C	240	244	248	24C	250	254
	15	258	25C	260	264	268	26C	270	274	278	27C
	16	280	284	288	28C	290	294	298	29C	2A0	2A4
	17	2A8	2AC	2B0	2B4	2B8	2BC	2C0	2C4	2C8	2CC
	18	2D0	2D4	2D8	2DC	2E0	2E4	2E8	2EC	2F0	2F4
	19	2F8	2FC	300	304	308	30C	310	314	318	31C
	20	320	324	328	32C	330	334	338	33C	340	344
	21	348	34C	350	354	358	35C	360	364	368	36C
	22	370	374	378	37C	380	384	388	38C	390	394
	23	398	39C	3A0	3A4	3A8	3AC	3B0	3B4	3B8	3BC
	24	3C0	3C4	3C8	3CC	3D0	3D4	3D8	3DC	3E0	3E4
	25	3E8	3EC	3F0	3F4	3F8	3FC	400	404	408	40C
	26	410	414	418	41C	420	424	428	42C	430	434
	27	438	43C	440	444	448	44C	450	454	458	45C
	28	460	464	468	46C	470	474	478	47C	480	484
	29	488	48C	490	494	498	49C	4A0	4A4	4A8	4AC

Figure A-1 (Part 1 of 3). Hexadecimal Values for Absolute Coordinates

Decimal Coordinates 300 through 599

	Units	0	1	2	3	4	5	6	7	8	9
Hundreds	30	4B0	4B4	4B8	4BC	4C0	4C4	4C8	4CC	4D0	4D4
and Tens	31	4D8	4DC	4E0	4E4	4E8	4EC	4F0	4F4	4F8	4FC
	32	500	504	508	50C	510	514	518	51C	520	524
	33	528	52C	530	534	538	53C	540	544	548	54C
	34	550	554	558	55C	560	564	568	56C	570	574
	35	578	57C	580	584	588	58C	590	594	598	59C
	36	5A0	5A4	5A8	5AC	5B0	5B4	5B8	5BC	5C0	5C4
	37	5C8	5CC	5D0	5D4	5D8	5DC	5E0	5E4	5E8	5EC
	38	5F0	5F4	5F8	5FC	600	604	608	60C	620	614
	39	618	61C	620	624	628	62C	630	634	638	63C
	40	640	644	648	64C	650	654	658	65C	660	664
	41	668	66C	670	674	678	67C	680	684	688	68C
	42	690	694	698	69C	6A0	6A4	6A8	6AC	6B0	6B4
	43	6B8	6BC	6C0	6C4	6C8	6CC	6D0	6D4	6D8	6DC
	44	6E0	6E4	6E8	6EC	6F0	6F4	6F8	6FC	700	704
	45	708	70C	710	714	718	71C	720	724	728	72C
	46	730	734	738	73C	740	744	748	74C	750	754
	47	758	75C	760	764	768	76C	770	774	778	77C
	48	780	784	788	78C	790	794	798	79C	7A0	7A4
	49	7A8	7AC	7B0	7B4	7B8	7BC	7C0	7C4	7C8	7CC
	50	7D0	7D4	7D8	7DC	7E0	7E4	7E8	7EC	7F0	7F4
	51	7F8	7FC	800	804	808	80C	810	814	818	81C
	52	820	824	828	82C	830	834	838	83C	840	844
	53	848	84C	850	854	858	85C	860	864	868	86C
	54	870	874	878	87C	880	884	888	88C	890	894
	55	898	89C	8A0	8A4	8A8	8AC	8B0	8B4	8B8	8BC
	56	8C0	8C4	8C8	8CC	8D0	8D4	8D8	8DC	8E0	8E4
	57	8E8	8EC	8F0	8F4	8F8	8FC	900	904	908	90C
	58	910	914	918	91C	920	924	928	92C	930	934
	59	938	93C	940	944	948	94C	950	954	958	95C

Figure A-1 (Part 2 of 3). Hexadecimal Values for Absolute Coordinates

Decimal Coordinates 600 through 1023

	Units	0	1	2	3	4	5	6	7	8	9
Hundreds	60	960	964	968	96C	970	974	978	97C	980	984
and Tens	61	988	98C	990	994	998	99C	9A0	9A4	9A8	9AC
	62	9B0	9B4	9B8	9BC	9C0	9C4	9C8	9CC	9D0	9D4
	63	9D8	9DC	9E0	9E4	9E8	9EC	9F0	9F4	9F8	9FC
	64	A00	A04	A08	A0C	A10	A14	A18	A1C	A20	A24
	65	A28	A2C	A30	A34	A38	A3C	A40	A44	A48	A4C
	66	A50	A54	A58	A5C	A60	A64	A68	A6C	A70	A74
	67	A78	A7C	A80	A84	A88	A8C	A90	A94	A98	A9C
	68	AA0	AA4	AA8	AAC	AB0	AB4	AB8	ABC	AC0	AC4
	69	AC8	ACC	AD0	AD4	AD8	ADC	AE0	AE4	AE8	AEC
	70	AF0	AF4	AF8	AFC	B00	B04	B08	B0C	B10	B14
	71	B18	B1C	B20	B24	B28	B2C	B30	B34	B38	B3C
	72	B40	B44	B48	B4C	B50	B54	B58	B5C	B60	B64
	73	B68	B6C	B70	B74	B78	B7C	B80	B84	B88	B8C
	74	B90	B94	B98	B9C	BA0	BA4	BA8	BAC	BB0	BB4
	75	BB8	BBC	BC0	BC4	BC8	BCC	BD0	BD4	BD8	BDC
	76	BE0	BE4	BE8	BEC	BF0	BF4	BF8	BFC	C00	C04
	77	C08	C0C	C10	C14	C18	C1C	C20	C24	C38	C2C
	78	C30	C34	C38	C3C	C40	C44	C48	C4C	C50	C54
	78	C58	C5C	C60	C64	C68	C6C	C70	C74	C78	C7C
	80	C80	C84	C88	C8C	C90	C94	C98	C9C	CA0	CA4
	81	CA8	CAC	CB0	CB4	CB8	CBC	CC0	CC4	CC8	CCC
	82	CD0	CD4	CD8	CDC	CE0	CE4	CE8	CEC	CF0	CF4
	83	CF8	CFC	D00	D04	D08	D0C	D10	D14	D18	D1C
	84	D20	D24	D28	D2C	D30	D34	D38	D3C	D40	D44
	85	D48	D4C	D50	D54	D58	D5C	D60	D64	D68	D6C
	86	D70	D74	D78	D7C	D80	D84	D88	D8C	D90	D94
	87	D98	D9C	DA0	DA4	DA8	DAC	DB0	DB4	DB8	DBC
	88	DC0	DC4	DC8	DCC	DD0	DD4	DD8	DDC	DE0	DE4
	89	DE8	DEC	DF0	DF4	DF8	DFC	E00	E04	E08	E0C
	90	E10	E14	E18	E1C	E20	E24	E28	E2C	E30	E34
	91	E38	E3C	E40	E44	E48	E4C	E50	E54	E58	E5C
	92	E60	E64	E68	E6C	E70	E74	E78	E7C	E80	E84
	93	E88	E8C	E90	E94	E98	E9C	EA0	EA4	EA8	EAC
	94	EB0	EB4	EB8	EBC	EC0	EC4	EC8	ECC	ED0	ED4
	95	ED8	EDC	EE0	EE4	EE8	EEC	EF0	EF4	EF8	EFC
	96	F00	F04	F08	F0C	F10	F14	F18	F1C	F20	F24
	97	F28	F2C	F30	F34	F38	F3C	F40	F44	F48	F4C
	98	F50	F54	F58	F5C	F60	F64	F68	F6C	F70	F74
	99	F78	F7C	F80	F84	F88	F8C	F90	F94	F98	F9C
	100	FA0	FA4	FA8	FAC	FB0	FB4	FB8	FBC	FC0	FC4
	101	FC8	FCC	FD0	FD4	FD8	FDC	FE0	FE4	FE8	FEC
	102	FF0	FF4	FF8	FFC	–	–	–	–	–	–

Figure A-1 (Part 3 of 3). Hexadecimal Values for Absolute Coordinates

Appendix B. Incremental Coordinates - Hexadecimal

The tables in Figure B-1 give the two-character hexadecimal value for incremental coordinates in the range from -64 to +63. The hexadecimal values have been adjusted to the format of the data fields following graphic-mode incremental point and incremental vector orders.

Pairs of incremental coordinates are defined in a 1-word data field. The first byte contains the X increment, and the second contains the Y increment and the blanking bit. To translate a pair of incremental coordinates to a four-digit hexadecimal value, set the first two digits to the hexadecimal value obtained for the X increment, and the other two digits to the hexadecimal value obtained for the Y increment and the blanking required.

Example: A data field, in the list following a graphic-mode incremental vector order, is required to draw an unblanked vector from the current beam position to a point 27 raster units to the left of (X increment = -27) and 18 raster units above (unblanked Y increment = 18) the current beam position. Set as shown below:

X increment (-27) = hexadecimal CB
Unblanked Y increment (18) = hexadecimal 24

Required hexadecimal data field:

Byte 0	Byte 1	
1 1 0 0 1 0 1 1	0 0 1 0 0 1 0 0	Hex CB24

Decimal Increments −64 through −32

	X Increment	Y Increment	
		Unblanked	Blanked
−64	81	80	81
−63	83	82	83
−62	85	84	85
−61	87	86	87
−60	89	88	89
−59	8B	8A	8B
−58	8D	8C	8D
−57	8F	8E	8F
−56	91	90	91
−55	93	92	93
−54	95	94	95
−53	97	96	97
−52	99	98	99
−51	9B	9A	9B
−50	9D	9C	9D
−49	9F	9E	9F
−48	A1	A0	A1
−47	A3	A2	A3
−46	A5	A4	A5
−45	A7	A6	A7
−44	A9	A8	A9
−43	AB	A0	AB
−42	AD	AC	AD
−41	AF	AE	AF
−40	B1	B0	B1
−39	B3	B2	B3
−38	B5	B4	B5
−37	B7	B6	B7
−36	B9	B8	B9
−35	BB	BA	BB
−34	BD	BC	BD
−33	BF	BE	BF
−32	C1	C0	C1

Decimal Increments −31 through 0

	X Increment	Y Increment	
		Unblanked	Blanked
−31	C3	C2	C3
−30	C5	C4	C5
−29	C7	C6	C7
−28	C9	C8	C9
−27	CB	CA	CB
−26	CD	CC	CD
−25	CF	CE	CF
−24	D1	D0	D1
−23	D3	D2	D3
−22	D5	D4	D5
−21	D7	D6	D7
−20	D9	D8	D9
−19	DB	DA	DB
−18	DD	DC	DD
−17	DF	DE	DF
−16	E1	E0	E1
−15	E3	E2	E3
−14	E5	E4	E5
−13	E7	E6	E7
−12	E9	E8	E9
−11	EB	EA	EB
−10	ED	EC	ED
−9	EF	EE	EF
−8	F1	F0	F1
−7	F3	F2	F3
−6	F5	F4	F5
−5	F7	F6	F7
−4	F9	F8	F9
−3	FB	FA	FB
−2	FD	FC	FD
−1	FF	FE	FF
0	01	00	01

Figure B-1 (Part 1 of 2). Hexadecimal Values for Incremental Coordinates

Decimal Increments 0 through 32

	X Increment	Y Increment	
		Unblanked	Blanked
0	01	00	01
1	03	02	03
2	05	04	05
3	07	06	07
4	09	08	09
5	0B	0A	0B
6	0D	0C	0D
7	0F	0E	0F
8	11	10	11
9	13	12	13
10	15	14	15
11	17	16	17
12	19	18	19
13	1B	1A	1B
14	1D	1C	1D
15	1F	1E	1F
16	21	20	21
17	23	22	23
18	25	24	25
19	27	26	27
20	29	28	29
21	2B	2A	2B
22	2D	2C	2D
23	2F	2E	2F
24	31	40	31
25	33	32	33
26	35	34	35
27	37	36	37
28	39	38	39
29	3B	3A	3B
30	3D	3C	3D
31	3F	3E	3F
32	41	40	41

Decimal Increments 33 through 63

	X Increment	Y Increment	
		Unblanked	Blanked
33	43	42	43
34	45	44	45
35	47	46	47
36	49	48	49
37	4B	4A	4B
38	4D	4C	4D
39	4F	4E	4F
40	51	50	51
41	53	52	53
42	55	54	55
43	57	56	57
44	59	58	59
45	5B	5A	5B
46	5D	5C	5D
47	5F	5E	5F
48	61	60	61
49	63	62	63
50	65	64	65
51	67	66	67
52	69	68	69
53	6B	6A	6B
54	6D	6C	6D
55	6F	6E	6F
56	71	70	71
57	73	72	73
58	75	74	75
59	77	76	77
60	79	78	79
61	7B	7A	7B
62	7D	7C	7D
63	7F	7E	7F

Figure B-1 (Part 2 of 2). Hexadecimal Values for Incremental Coordinates

Appendix C. Character Codes

This appendix defines the assigned hexadecimal codes for each of the character sets supported by the 3250 Graphics Display System.

All Character Codes (Except for the Numeric Keypad Keyboard)

Figures C-1 through C-6 give the hexadecimal code assignments of all character sets *except* those for the Numeric Keypad Alphanumeric Keyboard.

Bits 4,5,6,7	Hex 1	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
	Bits 0,1	00				01				10				11			
	Bits 2,3	00	01	10	11	00	01	10	11	00	01	10	11	00	01	10	11
0000	0	NUL				SP	&	-						{	}	\	0
0001	1							/		a	j	—		A	J		1
0010	2									b	k	s		B	K	S	2
0011	3									c	l	t		C	L	T	3
0100	4									d	m	u		D	M	U	4
0101	5		NL*							e	n	v		E	N	V	5
0110	6		BS							f	o	w		F	O	W	6
0111	7									g	p	x		G	P	X	7
1000	8									h	q	y		H	Q	Y	8
1001	9								`	i	r	z		I	R	Z	9
1010	A		CC*	SM*		$	!	¦	:								
1011	B					.	£	,	#								
1100	C					<	*	%	@								
1101	D					(	)	_	'								
1110	E					+	;	>	=								
1111	F					|	¬	?	"								

*Code not assigned by the alphanumeric keyboard

SP — Space
NUL — Null
NL — New Line
CC — Cursor Control
SM — Set Mode
BS — Backspace (Overstrike)

Note: *Character codes outlined above are displayable characters; when they are encountered in a character-mode data list, the relevant character is displayed and the beam is advanced to the next character position. If an undefined character code is encountered, undefined codes in the range hex 40 through hex FF are displayed as blanks (space), and codes in the range hex 00 through hex 3F are treated as Null characters. IBM reserves the right to change at any time the character displayed at the 3251 for an undefined character code.*

Examples:

Character	*Code*	*Character*	*Code*
A	Hex C1	%	Hex 6C
9	Hex F9	NUL	Hex 00

Figure C-1. English (U.K.) Character Code Assignments (except Numeric Keypad Keyboard)

Bits 4,5,6,7	Hex 1	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
Bits 0,1		00				01				10				11			
Bits 2,3		00	01	10	11	00	01	10	11	00	01	10	11	00	01	10	11
0000	0	NUL				SP	&	-						{	}	\	0
0001	1							/		a	j			A	J		1
0010	2									b	k	s		B	K	S	2
0011	3									c	l	t		C	L	T	3
0100	4									d	m	u		D	M	U	4
0101	5		NL*							e	n	v		E	N	V	5
0110	6		BS							f	o	w		F	O	W	6
0111	7									g	p	x		G	P	X	7
1000	8									h	q	y		H	Q	Y	8
1001	9									i	r	z		I	R	Z	9
1010	A		CC*	SM*		¢	!	¦	:								
1011	B					.	$	,	#								
1100	C					<	*	%	@								
1101	D					(	)	_	'								
1110	E					+	;	>	=								
1111	F					|	¬	?	"								

*Code not assigned by the alphanumeric keyboard

SP — Space
NUL — Null
NL — New Line
CC — Cursor Control
SM — Set Mode
BS — Backspace (Overstrike)

Note: *Character codes outlined above are displayable characters; when they are encountered in a character-mode data list, the relevant character is displayed and the beam is advanced to the next character position. If an undefined character code is encountered, undefined codes in the range hex 40 through hex FF are displayed as blanks (space), and codes in the range hex 00 through hex 3F are treated as Null characters. IBM reserves the right to change at any time the character displayed at the 3251 for an undefined character code.*

Examples:

Character	*Code*	*Character*	*Code*
A	Hex C1	%	Hex 6C
9	Hex F9	NUL	Hex 00

Figure C-2. English (U.S.) Character Code Assignments (except Numeric Keypad Keyboard)

		00				01				10				11				← Bits 0,1
		00	01	10	11	00	01	10	11	00	01	10	11	00	01	10	11	← 2,3
Bits 4,5,6,7	Hex 1	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	← Hex 0
0000	0	NUL				SP	&	-						é	è	ç	Ø	
0001	1							/		a	j	¨		A	J		1	
0010	2									b	k	s		B	K	S	2	
0011	3									c	l	t		C	L	T	3	
0100	4									d	m	u		D	M	U	4	
0101	5		NL*							e	n	v		E	N	V	5	
0110	6		BS							f	o	w		F	O	W	6	
0111	7									g	p	x		G	P	X	7	
1000	8									h	q	y		H	Q	Y	8	
1001	9								`	i	r	z		I	R	Z	9	
1010	A		CC*	SM*		°	§	ù	:									
1011	B					.	$	,	£									
1100	C					<	*	%	à									
1101	D					(	)	_	'									
1110	E					+	;	>	=									
1111	F					!	^	?	"									

*Code not assigned by the alphanumeric keyboard

SP – Space
NUL – Null
NL – New Line
CC – Cursor Control
SM – Set Mode
BS – Backspace (Overstrike)

Note: *Character codes outlined above are displayable characters; when they are encountered in a character-mode data list, the relevant character is displayed and the beam is advanced to the next character position. If an undefined character code is encountered, undefined codes in the range hex 40 through hex FF are displayed as blanks (space), and codes in the range hex 00 through hex 3F are treated as Null characters. IBM reserves the right to change at any time the character displayed at the 3251 for an undefined character code.*

Examples:

Character	*Code*	*Character*	*Code*
A	Hex C1	%	Hex 6C
9	Hex F9	NUL	Hex 00

Figure C-3. French Character Code Assignments (except Numeric Keypad Keyboard)

Bits 4,5,6,7	Hex 1 ↓ / Hex 0 →	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
Bits 0,1		00				01				10				11			
Bits 2,3		00	01	10	11	00	01	10	11	00	01	10	11	00	01	10	11
0000	0	NUL				SP	&	-						ä	ü	Ö	0
0001	1							/		a	j	ß		A	J		1
0010	2									b	k	s		B	K	S	2
0011	3									c	l	t		C	L	T	3
0100	4									d	m	u		D	M	U	4
0101	5		NL*							e	n	v		E	N	V	5
0110	6		BS							f	o	w		F	O	W	6
0111	7									g	p	x		G	P	X	7
1000	8									h	q	y		H	Q	Y	8
1001	9								`	i	r	z		I	R	Z	9
1010	A		CC*	SM*		Ä	Ü	ö	:								
1011	B					.	$	,	#								
1100	C					<	*	%	§								
1101	D					(	)	_	'								
1110	E					+	;	>	=								
1111	F					!	^	?	"								

*Code not assigned by the alphanumeric keyboard

SP — Space
NUL — Null
NL — New Line
CC — Cursor Control
SM — Set Mode
BS — Backspace (Overstrike)

Note: *Character codes outlined above are displayable characters; when they are encountered in a character-mode data list, the relevant character is displayed and the beam is advanced to the next character position. If an undefined character code is encountered, undefined codes in the range hex 40 through hex FF are displayed as blanks (space), and codes in the range hex 00 through hex 3F are treated as Null characters. IBM reserves the right to change at any time the character displayed at the 3251 for an undefined character code.*

Examples:

Character	*Code*	*Character*	*Code*
A	Hex C1	%	Hex 6C
9	Hex F9	NUL	Hex 00

Figure C-4. German Character Code Assignments (except Numeric Keypad Keyboard)

Bits 4,5,6,7	Hex 1	00				01				10				11			
	← Bits 2,3	00	01	10	11	00	01	10	11	00	01	10	11	00	01	10	11
	← Hex 0	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0000	0	NUL				SP	&	-						à	è	ç	0
0001	1							/		a	j	ì		A	J		1
0010	2									b	k	s		B	K	S	2
0011	3									c	l	t		C	L	T	3
0100	4									d	m	u		D	M	U	4
0101	5		NL*							e	n	v		E	N	V	5
0110	6		BS							f	o	w		F	O	W	6
0111	7									g	p	x		G	P	X	7
1000	8									h	q	y		H	Q	Y	8
1001	9								ù	i	r	z		I	R	Z	9
1010	A		CC*	SM*		°	é	ò	:								
1011	B					.	$	,	£								
1100	C					<	*	%	§								
1101	D					(	)	_	'								
1110	E					+	;	>	=								
1111	F					!	^	?	"								

*Code not assigned by the alphanumeric keyboard

SP – Space
NUL – Null
NL – New Line
CC – Cursor Control
SM – Set Mode
BS – Backspace (Overstrike)

Note: *Character codes outlined above are displayable characters; when they are encountered in a character-mode data list, the relevant character is displayed and the beam is advanced to the next character position. If an undefined character code is encountered, undefined codes in the range hex 40 through hex FF are displayed as blanks (space), and codes in the range hex 00 through hex 3F are treated as Null characters. IBM reserves the right to change at any time the character displayed at the 3251 for an undefined character code.*

Examples:

Character	*Code*	*Character*	*Code*
A	Hex C1	%	Hex 6C
9	Hex F9	NUL	Hex 00

Figure C-5. Italian Character Code Assignments (except Numeric Keypad Keyboard)

Bits 0,1		00				01				10				11			
Bits 2,3		00	01	10	11	00	01	10	11	00	01	10	11	00	01	10	11
Bits 4,5,6,7	Hex 1 / Hex 0	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0000	0	NUL				SP	&	-			ソ					$	0
0001	1						ェ	/		ア	タ	—		A	J		1
0010	2					「	ォ			イ	チ	ヘ		B	K	S	2
0011	3					」	ャ			ウ	ツ	ホ		C	L	T	3
0100	4					、	ュ			エ	テ	マ		D	M	U	4
0101	5		NL*			・	ョ			オ	ト	ミ		E	N	V	5
0110	6		BS			ヲ	ッ			カ	ナ	ム		F	O	W	6
0111	7					ァ				キ	ニ	メ		G	P	X	7
1000	8					ィ	ー			ク	ヌ	モ		H	Q	Y	8
1001	9					ゥ				ケ	ネ	ヤ		I	R	Z	9
1010	A		CC*	SM*		£	!		:	コ	ノ	ユ	レ				
1011	B					.	¥	,	#				ロ				
1100	C					<	*	%	@	サ		ヨ	ワ				
1101	D					(	)	_	'	シ	ハ	ラ	ン				
1110	E					+	;	>	=	ス	ヒ	リ	゛				
1111	F							¬	?	"	セ	フ	ル	゜				

*Code not assigned by the alphanumeric keyboard

SP – Space
NUL – Null
NL – New Line
CC – Cursor Control
SM – Set Mode
BS – Backspace (Overstrike)

Note: *Character codes outlined above are displayable characters; when they are encountered in a character-mode data list, the relevant character is displayed and the beam is advanced to the next character position. If an undefined character code is encountered, undefined codes in the range hex 40 through hex FF are displayed as blanks (space), and codes in the range hex 00 through hex 3F are treated as Null characters. IBM reserves the right to change at any time the character displayed at the 3251 for an undefined character code.*

Examples:

Character	*Code*	*Character*	*Code*
A	Hex C1	%	Hex 6C
9	Hex F9	NUL	Hex 00

Figure C-6. Katakana Character Code Assignments (except Numeric Keypad Keyboard)

Numeric Keypad Keyboard Character Codes

Bits 4,5,6,7	Hex 1	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
Bits 0,1		00				01				10				11			
Bits 2,3		00	01	10	11	00	01	10	11	00	01	10	11	00	01	10	11
0000	0	NUL				SP	&	-						{	}	\	0
0001	1							/		a	j	¯		A	J		1
0010	2									b	k	s		B	K	S	2
0011	3									c	l	t		C	L	T	3
0100	4									d	m	u		D	M	U	4
0101	5		NL							e	n	v		E	N	V	5
0110	6		BS							f	o	w		F	O	W	6
0111	7									g	p	x		G	P	X	7
1000	8									h	q	y		H	Q	Y	8
1001	9								`	i	r	z		I	R	Z	9
1010	A		CC	SM		$	!	¦	:								
1011	B					.	£	,	#						°	Ø	
1100	C					<	*	%	@						±		∨
1101	D					(	)	_	'					¢		≤	µ
1110	E					+	;	>	=						⎵	≥	•
1111	F					|	¬	?	"						↧	Ω	

Figure C-7. U.K. English (87- and 88-key)

Bits 4,5,6,7	Hex 1	00				01				10				11			
		00	01	10	11	00	01	10	11	00	01	10	11	00	01	10	11
		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0000	0	NUL				SP	&	-						{	}	\	0
0001	1							/		a	j			A	J		1
0010	2									b	k	s		B	K	S	2
0011	3									c	l	t		C	L	T	3
0100	4									d	m	u		D	M	U	4
0101	5		NL							e	n	v		E	N	V	5
0110	6		BS							f	o	w		F	O	W	6
0111	7									g	p	x		G	P	X	7
1000	8									h	q	y		H	Q	Y	8
1001	9									i	r	z		I	R	Z	9
1010	A		CC	SM		¢	!	¦	:								
1011	B					.	$	,	#						°	Ø	
1100	C					<	*	%	@						±		∨
1101	D					(	)	_	'					₵		≤	μ
1110	E					+	;	>	=						␣	≥	•
1111	F					|	¬	?	"						↧	Ω	

Figure C-8. U.S. English (86- and 88-key)

Bits 4,5,6,7	Hex 1	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
Bits 0,1		00				01				10				11			
Bits 2,3		00	01	10	11	00	01	10	11	00	01	10	11	00	01	10	11
0000	0	NUL				SP	&	-						é	è	ç	0
0001	1							/		a	j	¨		A	J		1
0010	2									b	k	s		B	K	S	2
0011	3									c	l	t		C	L	T	3
0100	4									d	m	u		D	M	U	4
0101	5		NL							e	n	v		E	N	V	5
0110	6		BS							f	o	w		F	O	W	6
0111	7									g	p	x		G	P	X	7
1000	8									h	q	y		H	Q	Y	8
1001	9								`	i	r	z		I	R	Z	9
1010	A		CC	SM		○	§	ù	:								
1011	B					.	$	,	£						°	Ø	
1100	C					<	*	%	à						±		∨
1101	D					(	)	_	'					¢		≤	µ
1110	E					+	;	>	=						␣	≥	•
1111	F					!	∧	?	"						↧	Ω	

Figure C-9. French (87- and 88-key)

		00				01				10				11				← Bits 0,1
		00	01	10	11	00	01	10	11	00	01	10	11	00	01	10	11	← 2,3
	Hex 1 ↓	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	← Hex 0
0000	0	NUL				SP	&	-						ä	ü	Ö	0	
0001	1							/		a	j	ß		A	J		1	
0010	2									b	k	s̄		B	K	S	2	
0011	3									c	l	t		C	L	T	3	
0100	4									d	m	u		D	M	U	4	
0101	5		NL							e	n	v		E	N	V	5	
0110	6		BS							f	o	w		F	O	W	6	
0111	7									g	p	x		G	P	X	7	
1000	8									h	q	y		H	Q	Y	8	
1001	9								`	i	r	z		I	R	Z	9	
1010	A		CC	SM		Ä	Ü	ö	:									
1011	B					.	$	,	#						°	Ø		
1100	C					<	*	%	§						±		∨	
1101	D					(	)	_	'					¢		≤	µ	
1110	E					+	;	>	=						␣	≥	•	
1111	F					!	^	?	"						↧	Ω		

Bits 4,5,6,7

Figure C-10. German (87- and 88-key)

Bits 4,5,6,7	Hex 1	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
Bits 0,1		00				01				10				11			
Bits 2,3		00	01	10	11	00	01	10	11	00	01	10	11	00	01	10	11
0000	0	NUL				SP	&	-			ソ					$	0
0001	1					。	ェ	/		ア	タ	ー		A	J		1
0010	2					「	ォ			イ	チ	ヘ		B	K	S	2
0011	3					」	ャ			ウ	ツ	ホ		C	L	T	3
0100	4					、	ュ			エ	テ	マ		D	M	U	4
0101	5		NL			・	ョ			オ	ト	ミ		E	N	V	5
0110	6		BS			ヲ	ッ			カ	ナ	ム		F	O	W	6
0111	7					ァ				キ	ニ	メ		G	P	X	7
1000	8					ィ	–			ク	ヌ	モ		H	Q	Y	8
1001	9					ゥ				ケ	ネ	ヤ		I	R	Z	9
1010	A		CC	SM		£	!		:	コ	ノ	ユ	レ				
1011	B					.	¥	,	#				ロ		°	Ø	
1100	C					<	*	%	@	サ		ヨ	ワ		±		∨
1101	D					(	)	_	'	シ	ハ	ラ	ン	¢		≤	µ
1110	E					+	;	>	=	ス	ヒ	リ	゛		␣	≥	•
1111	F					|	¬	?	"	セ	フ	ル	゜			Ω	

Figure C-11. Japanese (Katakana) (87- and 88-key)

Bits 4,5,6,7	Hex 1	0 (00 00)	1 (00 01)	2 (00 10)	3 (00 11)	4 (01 00)	5 (01 01)	6 (01 10)	7 (01 11)	8 (10 00)	9 (10 01)	A (10 10)	B (10 11)	C (11 00)	D (11 01)	E (11 10)	F (11 11)
0000	0	NUL				SP	&	-						ä	å	É	0
0001	1							/		a	j	ü		A	J		1
0010	2									b	k	s		B	K	S	2
0011	3									c	l	t		C	L	T	3
0100	4									d	m	u		D	M	U	4
0101	5		NL							e	n	v		E	N	V	5
0110	6		BS							f	o	w		F	O	W	6
0111	7									g	p	x		G	P	X	7
1000	8									h	q	y		H	Q	Y	8
1001	9								ù	i	r	z		I	R	Z	9
1010	A		CC	SM		§	¤	ö	:								
1011	B					.	Å	,	Ä						°	Ø	
1100	C					<	*	%	Ö						±		∨
1101	D					(	)	_	'					¢		≤	µ
1110	E					+	;	>	=						␣	≥	•
1111	F					!	^	?	"						↧	Ω	

Figure C-12. Swedish (87- and 88-key)

Appendix D. Summary of Channel Commands

The 3250 Graphics Display System uses four types of channel command: write, read, control, and sense. The write command initiates the transfer of data from the host system into the refresh buffer. Read commands cause a transfer of data either from the refresh buffer or from a register (for example, X and Y position registers) in a selected 3251 Display Station. Control commands initiate the setting of program function indicators or activate the audible alarm for operator attention at a selected 3251, and control display regeneration and cursor insertion. The Sense command transfers data to the channel, the transferred data indicating various control and/or check conditions in the graphics system.

The channel commands and command codes are shown in Figure D-1.

Type of Command	Channel Command	Command Code	Notes
Write	Write Buffer	Hex 01	5
Read	Read Buffer	Hex 02	5
	Read Manual Input	Hex 0E	1 and 2, 6
	Read Cursor	Hex 06	1, 7
	Read XY Position Registers	Hex 12	6
Control	Control No-Operation	Hex 03	4
	Set Buffer Address Register and Start	Hex 27	5
	Set Buffer Address Register and Stop	Hex 07	5
	Insert Cursor	Hex 0F	1, 3
	Remove Cursor	Hex 1F	1, 3
	Set Program Function Indicators	Hex 1B	2, 5
	Set Audible Alarm	Hex 0B	3
Sense	Sense	Hex 04	6

Notes:

1. *Needs the attachment of an alphanumeric keyboard for operation of the command.*
2. *Needs the attachment of a program function keyboard for operation of the command.*
3. *For these commands, channel-end is presented as 'initial' status, and device-end is presented when the 3251 is ready to accept another command.*
4. *This command causes the return of channel-end and device-end as 'initial' status.*
5. *These commands present channel-end after the last data byte is transferred, and device-end when the 3251 is ready to accept another command.*
6. *These commands present channel-end and device-end together at the completion of the operation.*
7. *This command presents channel-end and device-end together if the cursor is found before the CCW count is exhausted. Otherwise, channel-end is presented before device-end.*

Figure D-1. Channel Commands and Command Codes

Appendix E. Summary of Buffer Orders

Figure E-1 summarizes the buffer orders supported by the 3250 Graphics Display System.

Graphic-Mode Orders

Mnemonic	Name	Code
GEPM	Enter Graphic Mode Absolute Point	Hex 2A00
GEVM	Enter Graphic Mode Absolute Vector	Hex 2A02
GEPI2	Enter Graphic Mode Incremental Point	Hex 2A04
GEVI2	Enter Graphic Mode Incremental Vector	Hex 2A05

Character-Mode Orders

Mnemonic	Name	Code
GECF (B)	Enter Character Mode Basic	Hex 2A40 or Hex 2A50
GECF (L)	Enter Character Mode Large	Hex 2A41 or Hex 2A51
GECF (S)	Enter Character Mode Small	Hex 2A42 or Hex 2A52
GECF (M)	Enter Character Mode Medium	Hex 2A43
GECP (B)	Enter Character Mode Basic Protected	Hex 2A44
GECP (L)	Enter Character Mode Large Protected	Hex 2A45
GECP (S)	Enter Character Mode Small Protected	Hex 2A46
GECP (M)	Enter Character Mode Medium Protected	Hex 2A47
GECF (B,R)	Enter Character Mode Basic Rotated	Hex 2A48
GECF (L,R)	Enter Character Mode Large Rotated	Hex 2A49
GECF (S,R)	Enter Character Mode Small Rotated	Hex 2A4A
GECF (M,R)	Enter Character Mode Medium Rotated	Hex 2A4B
GECP (B,R)	Enter Character Mode Basic Protected Rotated	Hex 2A4C
GECP (L,R)	Enter Character Mode Large Protected Rotated	Hex 2A4D
GECP (S,R)	Enter Character Mode Small Protected Rotated	Hex 2A4E
GECP (M,R)	Enter Character Mode Medium Protected Rotated	Hex 2A4F

Figure E-1 (Part 1 of 2). Buffer Orders

Light-Pen-Mode Orders

Mnemonic	Name		Code
GDRD	Defer Response to Detects	2-byte class	Hex 2A83
GESD	Enable Switch Detect		Hex 2A84
GDPD	Disable Pen Detects		Hex 2A85
GENSD	Enable No-Switch Detect		Hex 2A86
GPDI	Permit Detect Interrupt		Hex 2A87

Control-Mode Orders

Mnemonic	Name		Code
GNOP2	No-Op 2-Byte	2-byte class	Hex 2A80
GEOS	End Order Sequence		Hex 2A81
GSRT	Start Regeneration Timer		Hex 2A82
GNOP4 (ADDR)	No-Op 4-Byte	4-byte class	Hex 2AC0
GTRU (ADDR)	Transfer Unconditional		Hex 2AFF
GSXY (ADDR)	Store X, Y Position Registers		Hex 2AEA
GLAR (VALUE)	Load Immediate Attribute Register		Hex 2AD1
GSAR (ADDR)	Store Attribute Register		Hex 2AD2
GSBL (B)	Blank (Intensity level 0)	2-byte class	Hex 2A90
GSBL (D)	Dim (Intensity level 3)		Hex 2A91
GSBL (N)	Normal (Intensity level 5)		Hex 2A92
GSBL (B)	Bright (High Intensity level 7)		Hex 2A93
GTDD (ADDR)	Transfer on Deferred Detect	4-byte class	Hex 2AFC
GTND (ADDR)	Transfer on No Detect		Hex 2AFD
GTSO (ADDR)	Transfer on Switch Open		Hex 2AF5
GMVA (ADDR,DATA)	Move Immediate Address	6-byte class	Hex 2AEB
GMVD (ADDR,DATA)	Move Immediate Data		Hex 2AEC

Figure E-1 (Part 2 of 2). Buffer Orders

Appendix F. Summary of Status-Sense Combinations

Figure F-1 lists the status bits and sense bits that are set for various conditions.

Conditions	*Status Bits Set*	*Sense Bits Set*
A. General Conditions		
1. Initial Status response to any command except Control No-Operation, Insert Cursor, Remove Cursor and Set Audible Alarm, or response to a Test I/O Instruction if there is no stacked status.	None (indicating that the command is accepted or that there is no stacked status)	None
2. Initial status response to: a. Control No-Operation command, or b. Insert Cursor, Remove Cursor and Set Audible Alarm.	 a. Channel End, Device End, or b. Channel End	None
3. Initial status response to any command except Test I/O when the selected 3251 has stacked status.	Busy plus outstanding status	None
4. Initial status response to a Test I/O instruction when the selected 3251 has stacked status.	Outstanding status	None
5. Initial status response to any command including a Test I/O instruction to a selected 3251 while the control unit is busy (see Note 1).	Busy, Status Modifier (Note 2)	None
6. a. After completion of data or control information transfer (except Read X-Y position register, Read Manual Input, or Sense commands). b. After completion of the operation for the commands in 6a. and 2b.	a. Channel End b. Device End, Channel End, (if not previously accepted by channel), Control Unit End, (Note 2)	None
7. 'Ending' status for Read X-Y position registers, Read Manual Input, or Sense commands.	Channel End, Device End, Control Unit End (Note 2)	None
8. Asynchronous response after a Halt I/O instruction, if the Halt I/O is issued after initial status and before ending status.	Channel End and/or Device End (presented asynchronously after Halt I/O to terminate command in progress, Control Unit End (Note 2).	None
B. Manual Input Conditions		
1. Asynchronous status when the alphanumeric keyboard END or CANCEL key is pressed.	Attention	None
2. Asynchronous status when any key is pressed on the program function keyboard.	Attention	None
3. Asynchronous status when an I/O interruption is raised for a light-pen detection.	Attention, Unit Check	Buffer Address (Note 3)
4. Asynchronous status when an End Order Sequence (GEOS) order is executed.	Attention, Unit Check	Buffer Address (Note 3)
C. Error Conditions — I/O Channel		
1. Initial status response to a command with invalid modifier bits. (Note 4)	Unit Check	Command Reject
2. Initial status response to a command with bad parity. (Notes 5 and 6)	Unit Check	Bus-Out Check

Figure F-1 (Part 1 of 2). Status-Sense Combinations

Conditions	*Status Bits Set*	*Sense Bits Set*
3. Initial status response to a Write Buffer, Read Buffer, Insert Cursor, Read Cursor, Remove Cursor, or Read XY Position Register command when the buffer is running. (Note 4)	Unit Check	Command Reject, Buffer Running
4. Ending status when data from the channel contains a byte with bad parity. (Notes 5 and 6)	Channel End (if not previously presented to, and accepted by, the channel), Device End, Unit Check	Bus-Out Check
D. Error Condition — 3250 System: *(Notes 5 and 7)* Ending status when an unrecoverable error is detected on the serial link.	Channel End, (if not previously presented to, and accepted by, the channel) Device End, Unit Check	3258 Check

Notes:

Note 3 explains an apparent anomaly in the buffer address value returned with the sense bytes. Notes 4 through 6 list host system program procedures suggested for handling errors.

1. *Busy. The 3258 is busy under the following conditions:*
   - *Status pending for another 3251.*
   - *Between initial selection and device-end for any 3251 (whether the command is chained or not, or Halt I/O is received). The 3258 is never busy to reselect within a command chain.*
   - *When unit-check status has been accepted by the channel for another 3251, and no command other than Test I/O or Control No-Operation has been subsequently addressed to that 3251.*
   - *During system reset and selective reset. The control unit end bit is included in these cases.*
2. *A control-unit-end pending condition is established whenever a control unit busy and status modifier (condition 5) response is given. This will result in a control-unit-end status bit being presented, either asynchronously, or combined with other status, as follows:*
   - *At the end of a CCW chain with device-end if the last CCW command results in channel-end and device-end being*
   - *presented separately.*
   - *As an asynchronous control-unit-end if the last CCW command results in channel-end and device-end being presented together.*
   - *Together with the pending status if the 'busy' condition was due to a pending asynchronous status (Attention, Unit Check etc).*
   - *With channel-end and device-end if the chain is broken by unit-check status.*
3. *The sense bytes always contain a buffer address, the value returned is only significant for conditions B3 and B4. The returned value of the buffer address may appear to have been incremented by an extra 1 if the address has previously been set on an odd-byte boundary.*
4. *Terminate this task or job.*
5. *Record the error and retry the operation once. Retrying the operation should involve the re-execution of the complete channel program which contained the failing command. If, for example, a Write Buffer command failed, the buffer address register may well have been changed, necessitating re-issuing the Set Buffer Address Register and Stop command to reset the address before re-issuing the Write command.*
6. *If the error is present on the retry, provide the operator with an error message and consider the selected 3250 system inoperative.*
7. *If the error is present on the retry, provide the operator with an error message and consider the selected 3258 and/or 3255 inoperative.*

Figure F-1 (Part 2 of 2). Status-Sense Combinations

Glossary

This glossary defines terms and abbreviations, as applicable to the 3250 Graphics Display System, that are used in the publication. If you do not find the term you are looking for, refer to the index or to *IBM Data Processing Glossary,* GC20-1699.

The glossary includes definitions developed by the American National Standards Institute (ANSI) and the International Organization for Standardization (ISO). This material is reproduced from the *American National Dictionary for Information Processing,* copyright 1977 by the Computer and Business Equipment Manufacturers Association, copies of which may be purchased from the American National Standards Institute, 1430 Broadway, New York, NY 10018.

These definitions are identified by asterisks (*).

absolute coordinate. (1) A grid location having reference to the origin of the grid. (2) Contrast with *incremental coordinate.*

absolute point. An addressing method whereby a display point is defined by the coordinates of its reference grid location.

absolute vector. (1) An addressing method whereby a vector is defined by the coordinates of the absolute point at each end of the vector. (2) Contrast with *incremental vector.*

alphanumeric. *Pertaining to a character set that contains letters, digits, and usually other characters, such as punctuation marks.

alphanumeric data. (1) Information presented in the form of letters, digits, and punctuation marks. (2) Contrast with *graphic data.*

application. The way a customer chooses to use the 3250 Graphics Display System.

application program. A program written for or by a user that applies to a particular application.

audible alarm. An alarm that is activated when predetermined events occur that require operator attention or intervention for system operation.

blink attribute. An attribute that may be assigned to a field to alternately suppress and enable the display of the field.

block-multiplexer channel. (1) A multiplexer channel that interleaves blocks of data. (2) See also *byte-multiplexer channel.* (3) Contrast with *selector channel.*

block multiplexing. A method of interleaving blocks of data. In a block-multiplexer channel, this method permits concurrent processing of multiple channel programs, thereby improving the efficiency of the channel during periods of device activity that do not need use of the channel.

buffer order. See *order.*

buffer program. A set of orders in sequence that, when executed, produces a display on the cathode-ray tube, and permits entry of keyboard data, or enables the light pen, or both.

buffer-refreshed. Pertaining to a displayed image that is maintained by repeated executions of a buffer program.

byte-multiplexer channel. (1) A multiplexer channel that interleaves bytes of data. (2) See also *block-multiplexer channel.* (3) Contrast with *selector channel.*

cathode-ray tube (CRT). An electronic vacuum tube, such as a television picture tube, that can be used to display graphic images.

CCW. Channel command word.

channel. (1) *A path along which signals can be sent, for example, data channel, output channel. (2) In System/370, a hardware device that connects the processing unit and main storage with the I/O control units.

channel command word (CCW). A doubleword at the location in main storage specified by the channel address word. One or more CCWs make up the channel program that directs channel operation.

channel status word (CSW). An area in storage that provides information about the termination of input/output operations.

character-mode. (1) A mode of display operation whereby characters are displayed on the screen of a 3251 Display Station Model 1 by way of a character generator in the 3255 Display Control. (2) Contrast with *graphic mode.*

character-mode order. An order that sets the 3255 Display Control Unit to character mode, and allows buffer program data to be displayed as alphanumeric characters.

command. A coded byte from main storage of the host system that specifies, to the channel and the 3258 Channel Control Unit, the operation to be performed.

computer graphics. (1) The technique whereby information may be converted to or from graphic display by use of a computer. (2) See *interactive computer graphics.*

control-mode order. An order that is executed by the 3255 Display Control Unit to control operations within the 3255.

cps. Regeneration cycles per second of a displayed image.

CRT. Cathode-ray tube.

CSW. Channel status word

cursor. An underscore character, displayed on the screen, that shows where the next entered character will be displayed.

diagnostic. *Pertaining to the detection and isolation of a malfunction or mistake.

directed-beam. A technique whereby the constituents of a display image are generated or recorded by programming the deflection of the electron beam in the cathode-ray tube.

display. (1) *(ISO) A visual presentation of data. (2) To present a display image on a display surface.

display buffer. Buffer storage in the 3255 Display Control Unit that contains the buffer program.

display image. Those elements, such as graphic or alphanumeric data, that are presented at any one time on the screen of a display device.

EBCDIC. *Extended binary-coded decimal interchange code. A coded character set consisting of 8-bit coded characters.

graphic data. (1) Information presented as electronically drawn symbols or lines. (2) Contrast with *alphanumeric data.*

graphic mode. (1) A mode of display operation whereby points, vectors, and shapes can be presented on the screen of the 3251 Display Station Model 1. (2) Contrast with *character mode.*

graphic-mode order. An order that sets the 3255 Display Control Unit to graphic mode, and allows buffer program data to generate lines (vectors) on the screen.

graphics display. The presentation of information in graphic or alphanumeric form on the face of a TV-like display tube.

hertz (Hz). A unit of frequency equal to one cycle per second.

host system. A data processing system to which a 3250 Graphics Display System is connected. The host system, which comprises a computer, its programs, and its peripheral equipment, provides overall control of the 3250 system.

Hz. Hertz.

image. See *display image.*

in. Inch.

incremental coordinate. (1) An addressing method whereby the coordinates of a point are defined as a displacement from another point that is not the origin. (2) Contrast with *absolute coordinate.*

incremental vector. (1) An addressing method whereby the end points of a vector are defined as a displacement from the current beam position. (2) Contrast with *absolute vector.*

instruction. (1) A statement that specifies an operation and the values or locations of its operands. (2) A program step that is decoded and executed by the processing unit of the host system.

intensity attribute. An attribute that may be assigned to a field to govern the amount of light emitted by the electron beam in that field.

interaction. An operator action (for example, a light-pen action) that invokes a response from the host system.

interactive computer graphics. In the 3250 Graphics Display System, the technique of using a keyboard or a light pen of the 3251 Display Station to work with information displayed.

I/O. Input/output.

I/O interruption. An interruption caused by the termination of an I/O operation or by operator intervention at the I/O device.

K. *When referring to storage capacity, two to the tenth power; 1024 in decimal notation.

keyboard clicker. An audible feedback device that emits a sound ("click") when a keyboard key is pressed. (The 3250 Graphics Display System causes the click to confirm that a key code has been accepted from the alphanumeric keyboard.)

light pen. A hand-held pointer that allows the operator to select a point or a field on the display surface of a cathode-ray tube.

light-pen detection. A condition set in response to the signal generated when the electron beam passes through the field of view of the light pen.

light-pen-mode order. An order that sets the various modes of operation of the light pen.

light-pen tracking. The process of tracking the movement of a light pen across the screen of a CRT device.

line type attribute. An attribute that governs the form (solid, dashed, dotted, or dot-dashed) of a displayed vector.

mm. Millimeter.

order. (1) A number of coded bytes, contained in the buffer program, that specify an operation or mode of operation to the 3255 Display Control Unit. Orders are set up, by the programmer, in main storage of the processing unit and are treated as data by both the channel and the 3258 Channel Control Unit. (2) See *character-mode order, control-mode order, graphic-mode order,* and *light-pen-mode order.*

regeneration. The continual refreshing of the display on a cathode-ray tube to provide a visible image.

repeat-action key. A key that, when held pressed, causes an action (such as typing of a character) to be repeated until the key is released, for example, a typematic key.

selector channel. (1) An I/O channel designed to operate with only one I/O device at a time. Once the I/O device is selected, a complete record is transferred one byte at a time. (2) Contrast with *block-multiplexer channel* and *byte-multiplexer channel.*

sense. Device-dependent coded data that defines the internal state of the device, normally during unusual conditions.

special feature. A feature that may involve extra cost but is not required to make the system functional; special features provide additional operational capability.

SEI. Single-element intensification. The ability to locally brighten a vector, character, or point by use of a light pen without intervention from the host system.

twos complement. (1) *(ISO) The radix complement in the pure binary numeration system. (2) In the 3250 Graphics Display System, the 7-bit twos complement of a negative increment requires the most significant bit set to 1 (representing -64) and the remaining bits set to the difference between the negative base (-64) and the required displacement. For example; a displacement of -53 is equal to displacement of -64 (1000000) and -11 (0001011); therefore the twos complement of -53 is 1001011.

typematic key. A repeat-action key.

vector. A displayed line connecting any two addressable points.

Bibliography

The IBM publications listed here contain further information that may be relevant to the user of a 3250 system.

- *An Introduction to the IBM 3250 Graphics Display System*, GA33-3035. This publication provides introductory information about the system. The information addresses (1) the graphics functions of the 3250 system; (2) the units of the 3250 system; (3) attachment to an input/ output channel of an IBM System/370; (4) performance considerations; (5) programming support; (6) conversion of IBM 2250 Display Unit Model 3 applications to use the 3250 system; and (7) introductory installation-planning information.
- *IBM 3250 Graphics Display System: Installation Manual - Physical Planning*, GA33-3036. This manual provides physical and environmental data, cabling requirements, and information needed to prepare for the installation of a 3250 system.
- *IBM 3250 Graphics Display System: Custom Feature Summary*, GA33-3086. This publication summarizes the Custom Features available for the IBM 3250 Graphics Display System. It presents an overview of each custom feature, describes their implementation and installation, and discusses any software considerations inherent with their implementation.
- *IBM System/370 Principles of Operation*, GA22-7000. This publication provides, for reference purposes, a detailed definition of the machine functions performed by the System/370. Its contents are addressed primarily to assembler language programmers, although it is of interest to anyone concerned with the functional details of System/370. It is not, however, an introduction or a textbook for System/370.
- *IBM System/360 and System/370 I/O Interface Channel to Control Unit: Original Equipment Manufacturers' Information*, GA22-6974. This publication provides the definitions and functional description of the interface lines for the I/O interface-channel to control unit. The publication also contains the electrical, mechanical, and cabling considerations and specifications of this interface.
- *OS/VS Graphic Programming Services (GPS) for IBM 2250 Display Unit and IBM 3250 Graphics Display System*, GC27-6971. This publication describes macro instructions and routines provided for use with OS/VS to aid in writing assembler language programs that use the 2250 Model 1 or 3 or 3251. It also provides general programming information for the 2250 or 3250.

Requests for the above publications should be made to your IBM Representative, or to the IBM branch office serving your locality.

Index

G

I

K

L

O

P

R

S

T

U

V

W

X

3

IBM 3250 Graphics Display System
Component Description

Order No. GA33-3037-2

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.*

Possible topics for comments are:

Clarity Accuracy Completeness Organization Index Figures Examples Legibility

What is your occupation? . . . .

Number of latest Newsletter associated with this publication: . . . .

If you wish a reply, give your name, company, mailing address, and date: . . . .

. . . .

. . . .

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 either address in the Edition Notice on the back of the title page.)

Note: *Staples can cause problems with automated mail sorting equipment. Please use pressure-sensitive or other gummed tape to seal this form.*

Cut Along Dotted Line

GA33-3037-2

Reader's Comment Form

Fold and Tape Fold and Tape Fold and Tape

No postage necessary if mailed in the United States

BUSINESS REPLY MAIL

FIRST CLASS PERMIT 40 ARMONK, NEW YORK

Postage will be paid by addressee:

International Business Machines Corporation
Department 812H
1133 Westchester Avenue
White Plains, New York 10604

Fold and Tape Fold and Tape Fold and Tape

IBM®

IBM 3250 Graphics Display System: Component Description Printed in U.S.A. GA33-3037-2

IBM 3250 Graphics Display System: Component Description Printed in U.S.A. GA33-3037-2

GA33-3037-2