GA33-3120-0
File No. S370/4300-06, S370/4300-14

IBM 3250 Graphics Display System: Attachments for Cursor Control Tablet and for Plotter RPQs 7J0070, 7J0071, 7J0072, and 7J0073

Systems

Custom Feature Description

IBM

GA33-3120-0
File No. S370/4300-06, S370/4300-14

IBM 3250 Graphics Display System: Attachments for Cursor Control Tablet and for Plotter RPQs 7J0070, 7J0071, 7J0072, and 7J0073

Systems

Custom Feature Description

First Edition, December 1982

Changes are periodically made to the information herein; before using this publication in connection with the operation of IBM systems or equipment, refer to the latest *IBM System/370 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.

This manual contains sample programs. Permission is hereby granted to copy and store the sample programs into a data processing machine and to use the stored programs for study and instruction only. No permission is granted to use the sample programs for any other purpose.

Preface

This publication describes the Cursor Control Tablet Attachment custom feature, and the Plotter Attachment custom feature, which are provided for the IBM 3250 Graphics Display System by Request for Price Quotations (RPQs).

The publication is intended for planners, system analysts and application programmers.

Organization of the Book

The book has three parts:

- **Part 1. Introduction:** This describes the four RPQ's available, and the system configurations that can be installed with them. Information that is useful for planning and ordering is also provided.
- **Part 2. Cursor Control Tablet Attachment Custom Feature:** This describes the facilities provided by the custom feature. Appendixes A and B (see below) provide example graphic macros and programs for the tablet attachment.
  - **Chapter 1.** Custom Feature Overview gives a summary of the main facilities provided by the custom feature.
  - **Chapter 2.** Interface Characteristics provides a description of the interface facilities, device attachment, interface signals and characteristics, and communications protocol.
  - **Chapter 3.** Control of Tablets describes the control facilities available to the application programmer for creating the necessary control dialog between the host application program and a cursor control tablet.
- **Part 3. Plotter Attachment Custom Feature** This describes the facilities provided by the custom feature. Appendixes C and D (see below) give examples of plotter macros and programs.
  - **Chapter 4.** The custom feature overview gives a summary of the main facilities provided by the custom feature.
  - **Chapter 5.** The interface characteristics provide a description of the interface facilities, device attachment, interface signals and characteristics, and communications protocols.
  - **Chapter 6.** The control of the plotter describes the control facilities available to the applications programmer to create the necessary control dialog between the host application program and the attached device.

The appendixes supplement the information given in Parts 2 and 3:

- **Appendix A. Example Tablet Macros:** This describes example macros that will generate the additional buffer orders introduced to support the tablet attachment custom feature.
- **Appendix B. Example Tablet Programs:** This describes example buffer programs that have been written, using the additional buffer orders, to support the tablet attachment custom feature.
- **Appendix C. Example Plotter Macros:** This describes six example macros that have been written to support the plotter attachment custom feature.

Three macros generate buffer orders of the same mnemonic, and three macros generate data blocks used during input/output operations.

- **Appendix D. Example Plotter Program:** This describes an example program that has been written, using the buffer orders, and the macros, introduced to support the plotter attachment custom feature.
- **Appendix E. Summary of Buffer Orders:** This contains a summary of the buffer orders giving their mnemonics, names and hexadecimal codes.

Associated Publications

The following publication contains reference information on the 3250 system:

IBM 3250 Graphics Display System: Component Description, GA33-3037. (Shortened title: *3250 Component Description.)*

The following publications describe other custom features:

IBM 3250 Graphics Display System: Custom Features Summary, GA33-3086. (Shortened title: *3250 Custom Features Summary.)*

IBM 3250 Graphics Display System: Continuous Refresh RPQs 7J0024 and 7J0025; Custom Features User's Guide, GA33-3085. (Shortened title: *3250 Continuous Refresh User's Guide.).*

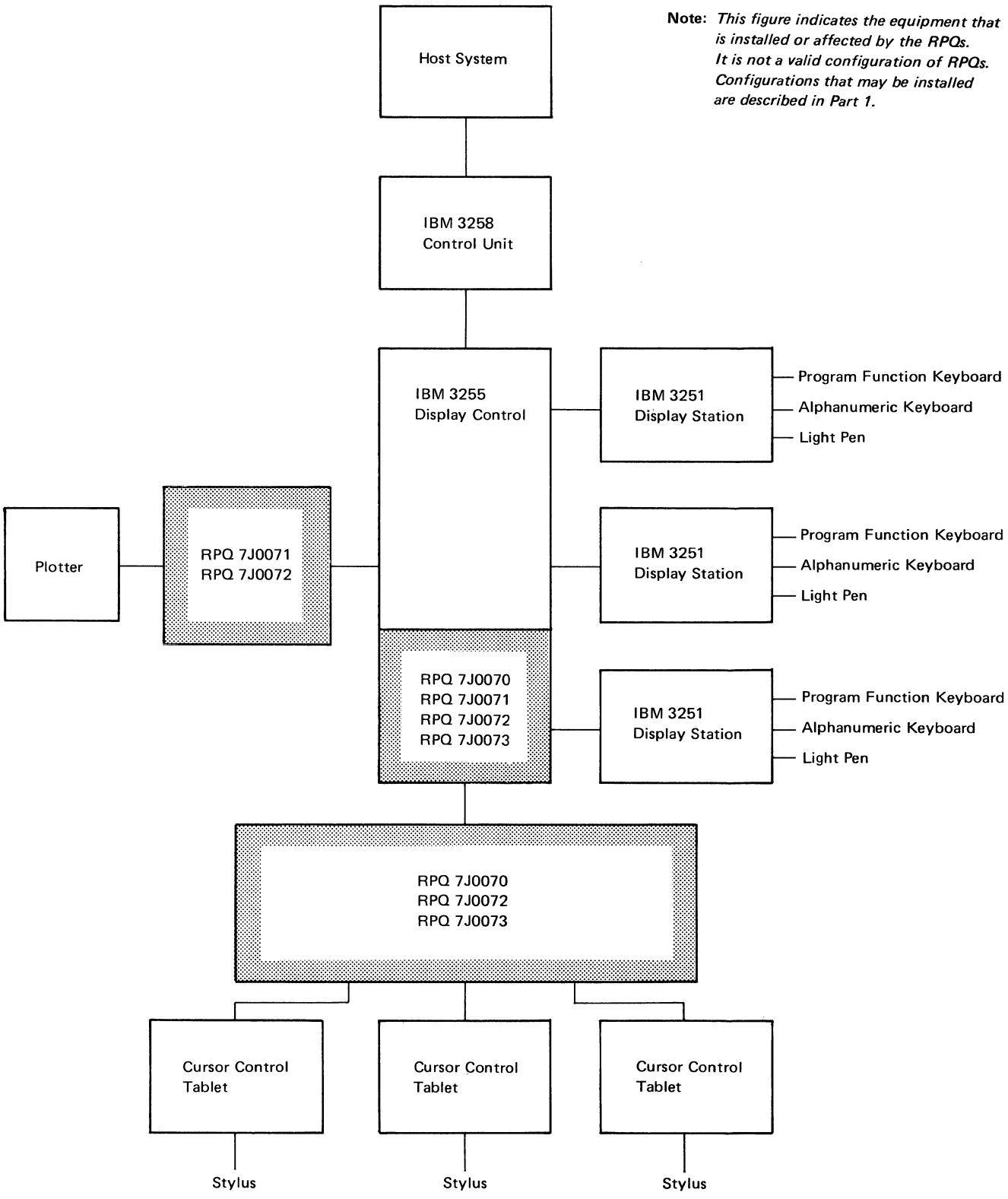
The following publications contain reference information on Binary Synchronous Communications:

IBM Systems Reference Library: General Information - Binary Synchronous Communications, GA24-3004

Component Information for the IBM 3780 Data Communication Terminal, GA27-3063.

Contents

Figures



Frontispiece. Configuration of a 3250 System

Part 1. Introduction

This publication describes the Cursor Control Tablet Attachment custom feature, and the Plotter Attachment custom feature, which are provided for the IBM 3250 Graphics Display System.

Part 2 of this publication describes *the cursor control tablet attachment.* This custom feature allows up to three non-IBM tablets to be attached to an IBM 3255 Display Control (see Frontispiece). Each tablet is used to control the position of a square cursor on the screen of an associated IBM 3251 Display Station. The feature may be used to designate displayed graphic elements for further processing, and to enter new coordinates to the 3250 system.

The facilities for attaching a non-IBM cursor control tablet can be installed in a 3250 system in two ways:

1. Plant installation of attachment facilities for an initial tablet.
2. Field installation, or plant installation, of attachment facilities for one or two additional tablets when prerequisite Requests for Price Quotations (RPQs) have been installed.

Part 3 of this publication describes the *plotter attachment.* This custom feature allows one non-IBM plotter to be attached to a 3255 Display Control. This feature enables a hard copy of displayed graphic elements to be available at the work area of the 3251 Display Station. The facilities for attaching a non-IBM plotter can only be installed at the plant.

Requests for Price Quotations

Non-IBM cursor control tablets and non-IBM plotters can be attached to a 3250 system using the attachment facilities provided by the following RPQs. The RPQs provide a selection of features (with prerequisite features) that enable the system to be configured in various ways, according to user requirements.

- RPQ 7J0070 Cursor Control Tablet Attachment
- RPQ 7J0071 Plotter Attachment
- RPQ 7J0072 Expansion Feature
- RPQ 7J0073 Additional Cursor Control Tablet

Attachment Configurations

Tablet Attachment

Up to three non-IBM tablets may be attached to one 3255, using RPQs 7J0070, 7J0072, and 7J0073 in the following combinations:

Number of Tablets		*RPQs*
one		RPQ 7J0070
	or	RPQ 7J0072
two		RPQ 7J0070 and 7J0073
	or	RPQ 7J0072 and 7J0073
three		RPQ 7J0072 and two 7J0073

Note: *RPQ 7J0072 also provides attachment facilities for one non-IBM plotter.*

Plotter Attachment

One non-IBM plotter (only) may be attached to a 3255, using RPQ 7J0071 (or RPQ 7J0072, which also provides attachment facilities for one non-IBM tablet).

Plotter-and-Tablet Attachment

One non-IBM plotter and one, two or three non-IBM tablets, may be attached to a 3255 using the RPQs listed above.

Note: *RPQ 7J0072 provides attachment facilities for one non-IBM plotter and one non-IBM tablet.*

RPQ Descriptions

RPQ 7J0070 Cursor Control Tablet Attachment (Plant Installation Only)

RPQ 7J0070, Cursor Control Tablet Attachment, provides:

- Attachment facilities for one non-IBM cursor control tablet, including an attachment cable 15.24 meters (50 feet) long.
- Continuous-refresh function, described in *IBM 3250 Graphics Display System: Continuous Refresh RPQs 7J0024 and 7J0025; Custom Feature User's Guide,* GA33-3085.

RPQ 7J0070 can only be installed at the manufacturing plant. Plant installation, or later field installation, of one additional cursor control tablet attachment is available through the Additional Cursor Control Tablet RPQ 7J0073 (described later).

RPQ 7J0070 is compatible with the following RPQs:

- RPQ 7J0001, Four-Level Intensity
- RPQs SU0091, 7J0005, 7J0014, 7J0015, and 7J0017, Data Communications Facility
- RPQs 7J0011, 7J0012, MN0077, and MN0078, Engineering Symbols.

RPQ 7J0071 Plotter Attachment (Plant Installation Only)

RPQ 7J0071, Plotter Attachment, provides:

- Attachment facilities for one non-IBM plotter, including an attachment cable, 9.14 meters (30 feet) long.
- Expansion of the buffer storage in the 3255 Display Control to provide an additional 32K bytes (K equals 1024), together with increased drawing speeds for vector plots and point plots. See "Chapter 7. Increased Capacity" in the *IBM 3250 Graphics Display System: Custom Feature Summary,* GA33-3086.
- Continuous-refresh function described in the *3250 Continuous Refresh User's Guide.*

  Note: *IBM supports the continuous-refresh function for 3251 Display Stations only.*

RPQ 7J0071 is compatible with the following RPQs:

- RPQ 7J0001, Four-Level Intensity
- RPQs SU0091, 7J0005, 7J0014, 7J0015, and 7J0017, Data Communications Facility
- RPQs 7J0011, 7J0012, MN0077, and MN0078, Engineering Symbols.

RPQ 7J0072 Expansion Feature (Plant Installation Only)

RPQ 7J0072, Expansion Feature, provides:

- Attachment facilities for one non-IBM plotter, including an attachment cable 9.14 meters (30 feet) long.
- Attachment facilities for one non-IBM cursor control tablet, including an attachment cable 15.24 meters (50 feet) long.
- Attachment of one additional 3251 Display Station. See "Chapter 2. Additional Workstation" in the *3250 Custom Feature Summary.*
- Expansion of the buffer storage in the 3255 Display Control to provide an additional 32K bytes together with increased drawing speeds for vector plots and point plots. See "Chapter 7. Increased Capacity" in the *3250 Custom Feature Summary.*
- Continuous-refresh function, described in the *3250 Continuous Refresh User's Guide.*

  Note: *IBM supports the continuous-refresh function for 3251 Display Stations only.*
- RPQ 7J0072 is available for plant installation only.
- Plant installation, or subsequent field installation, of attachment facilities for one or two additional cursor control tablets is available through RPQ 7J0073, Additional Cursor Control Tablet (described below).

RPQ 7J0072 is compatible with the following RPQs:

- RPQ 7J0001, Four-Level Intensity
- RPQs SU0091, 7J0005, 7J0014, 7J0015, and 7J0017, Data Communications Facility
- RPQs 7J0011, 7J0012, MN0077, and MN0078, Engineering Symbols.

RPQ 7J0073 Additional Cursor Control Tablet (Field or Plant Installation)

This RPQ provides attachment facilities for an additional non-IBM cursor control tablet.

Notes:

1. *One additional non-IBM tablet may be attached to a 3255 if the prerequisite RPQ 7J0070, Cursor Control Tablet Attachment, is installed.*
2. *Two additional tablets may be attached to a 3255 if the prerequisite RPQ 7J0072, Expansion Feature, is installed. In this case two of RPQ 7J0073 are installed.*

Part 2. Tablet Attachment Custom Feature

Chapter 1. Tablet Attachment Feature: Overview

RPQ 7J0070, RPQ 7J0072, and RPQ 7J0073 (as described in Part 1) provide the facilities to attach up to three non-IBM cursor control tablets to an IBM 3255 Display Control. Each tablet is permanently associated with an IBM 3251 Display Station and controls the position of a cursor (in the form of a small square) on the 3251 screen. The custom feature gives:

- Similar facilities for detecting display elements, and interacting with the system, as those provided by a light pen
- The added flexibility of an independent data source:
  - When adding a new element to the displayed image, it is not necessary to flood the display with characters in order to define the position of the element.
  - Identification of displayed elements is independent of the adjusted brightness level of the display.
  - The operator need not sit close to the display screen.

The operator may use a cursor control tablet and its associated stylus in a similar manner to a light pen to identify graphic elements on the 3251 screen to the host application program; only elements with an intensity level of 5 or greater will be selected (that is, 'detected').

Vector coordinates and status information that indicates the condition of the stylus tip-switch (operated/not-operated) are transmitted from the tablet to the 3255 as an asynchronous serial data stream. The buffer program for the associated 3251 Display Station may interrogate the serial data stream and store the data in the 3255 display buffer. The data may then be transmitted to the host program as part of the normal data traffic. Additional graphic orders to support the feature are embedded in the buffer program for the associated 3251, and are executed by the feature interface adapter.

Note: *Non-IBM tablets used with the feature must always be set to run mode, that is coordinates transmitted continuously when the tablet stylus is close to the tablet surface.*

Each tablet is connected to the 3255 via a port, a cable 15.24 meters (50 feet) long, and a serial data interface, the details of which are given in Chapter 2.

Configuration

It is possible to install a cursor control tablet attachment for every 3251 Display Station used in an IBM 3250 Graphics Display System, regardless of how the system is configured. The feature has no effect on the possible configurations of the basic system. A description of the RPQs by which this feature may be purchased is provided in Part 1.

Note: *Example graphics programs and macros provided in Appendixes A and B have been tested with a Talos Wedge*[1] *tablet. For information about the use of other tablets with a serial data interface, see your IBM Representative.*

---

[1]Trademark of Talos Systems Inc.

Control Facilities

Buffer Orders

The buffer program that drives the associated 3251 Display Station controls the following:

- Transfer, to the 3255 buffer, of coordinate data from the serial data stream originating in the tablet
- Display of the cursor symbol.

For this control, the program uses additional buffer orders to support the tablet attachment feature:

GSPOS	Store X,Y Position	This order may also be used to enable tip-switch interrupts.
GSYMB	Draw Symbol	This order may also be used to create a detection window defined by the symbol.
GTOC	Transfer On Condition	

Channel Commands

The 3250 channel commands used to control a 3251 buffer program are not changed when a tablet attachment feature is installed for use with that display station.

Diagnostic Routines

The 3255's internal diagnostic routines perform checks on the tablet-attachment ports. The routines run when selected and, automatically, during initialization. They do not check the attached tablets.

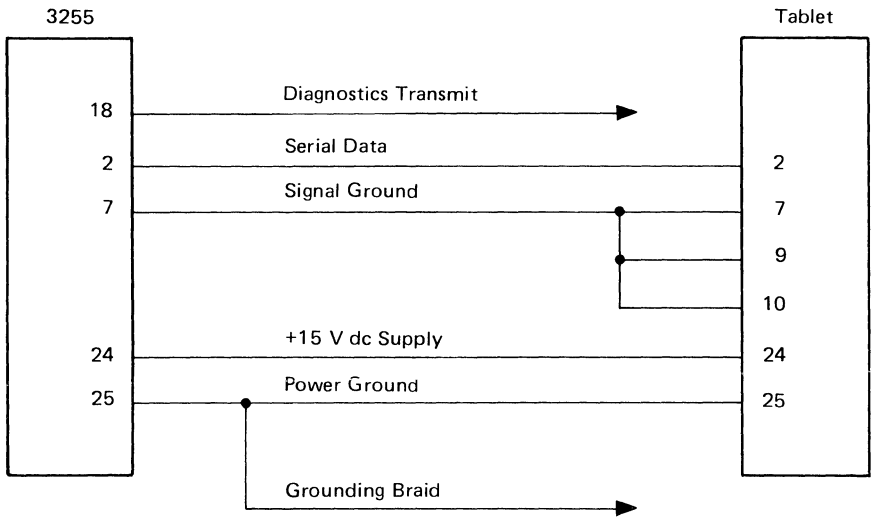
Chapter 2. Interface Characteristics

Mechanical Interface Characteristics

Each cursor control tablet is connected to a port in the 3255 Display Control by a multiway cable, which is supplied with the tablet-attachment RPQ. The cable is 15.24 meters (50 feet) long. Cannon[2] 'D' series connectors terminate the cable: a socket-type connection to the tablet, and a plug-type connection to the 3255.

Electrical Interface Characteristics

Figure 1 shows the interface cable pin connections and Figure 2 shows the interface signal conventions.



Note: *Pin numbers not shown are not connected at the 3255.*

Figure 1. Interface Cable Pin Connections

Interface Signals		
Function	Off	On
Voltage	Less than -3 V	Greater than +3 V
Binary State	1	0
Signal Condition	Marking	Spacing

Figure 2. Interface Signal Conventions

[2]Trademark of International Telephone and Telegraph Corp.

Interface Signal Lines

The signal lines shown in Figure 1 have the following purposes:

Diagnostics Transmit

The 'diagnostics transmit' line is used for diagnostic purposes only.

Serial Data

The 'serial data' line carries data from the tablet to the 3255. When the line is inactive (that is, when the stylus is not close to the tablet surface), the tablet must hold the line in a mark condition. The signal levels are TTL; the transmission is asynchronous.

Signal Ground

The 'signal ground' line provides a return path for the 'serial data' line. It is also jumpered to pins 9 and 10 at the tablet to set the correct data transmission rate (4800 bps).

+15 V DC Supply

The 3255 supplies +15 volts dc power to the tablet on this line.

Note: *The total current drawn by tablets connected to one 3255 must not exceed 800 milliamperes.*

Power Ground

The 'power ground' line is the ground return for the +15 volts dc supply.

Transmission Rate

A transmission rate of 4800 bps from the tablet is selected by hard-wired connections in the socket at the tablet end of the multiway cable. Pins 9 and 10 at the tablet are wired to 'signal ground'; pins 8 and 11 are not connected (space condition).

Chapter 3. Control of Attached Tablets

Introduction

This chapter describes the control features available to the applications programmer to create the necessary control dialogue between the host application program and the attached device.

The cursor control tablet feature is controlled by an order set embedded in the buffer program for the associated 3251 Display Station (that is, the tablet shares the same device address). Buffer orders, additional to those of the 3250 basic set (described in *IBM 3250 Graphics Display System: Component Description,* GA33-3037), are used to support the feature.

The additional buffer orders are:

Mnemonic	Name	Hex Code
GSPOS	Store X,Y Position	2AE8
GSYMB	Draw Symbol	2A88
GTOC	Transfer On Condition	2A70-F

Using these additional orders, the application program can:

- Put the stylus coordinates into the display buffer.
- Draw a square outline (cursor) on the display screen at a position defined by the stylus coordinates. The program can define the size of the square, from a minimum side measurement of 0.73 millimeter (0.03 inch, five addressable points) up to a maximum of 18.75 millimeters (0.738 inch, 64 addressable points).
- Use the square cursor as a detection window and simulate light-pen detection of graphic objects within the window.
- Take branches in the buffer program, or interrupt the host application program, according to the state of the stylus tip-switch.

With these facilities, the tablet and stylus can interact with the system in a similar manner to using a light pen on a display screen. Details of these orders are given later in this chapter.

Channel Commands

The tablet buffer program is loaded, started, and stopped, by the same channel commands; these are the same commands that operate on a normal buffer program for a 3251. Attention, unit check status, sense data (bit 3 of byte 1), and the Light Pen Interrupt (LPI) bit are returned in response to a Sense command.

Buffer Operation

When the stylus is close to the tablet surface, a continuous stream of serial data is transmitted to the 3255 Display Control. The usual transmission rate is 4800 bps, which gives a data rate of 48 coordinates per second.

The feature interface adapter in the 3255 interrogates the asynchronous data stream from a tablet at regular intervals (normally at every GSRT occurrence for the associated display) to extract the X,Y coordinate data and buffer it within the adapter. These buffered coordinates are then transferred by the GSPOS buffer order into the 3255 main storage. The interface adapter also examines the status byte (part of the asynchronous adapter) to determine the state of the stylus tip-switch, and this information is used to set the tip-switch indicator (TSI) for that device.

The cursor control tablet may be used to interact with a 3251 Display Station and perform similar functions to those available with a light-pen. The new buffer orders GSPOS and GSYMB control actions taken by the buffer program. *GSPOS* may be used to store the tablet coordinates and also generate an interrupt to the channel. *GSYMB* may be used to draw a square cursor on the screen; this cursor is then available as a detection window, and can be used as follows:

- To designate graphic elements on the 3251 screen in a similar manner to a light pen
- To inject new coordinates into the system.

The 3255 responds to a tablet detection using the same control modes that it uses for a light-pen detection. A detection made with a tablet cursor is accepted only on vectors having intensity levels of 5, 6, or 7.

Notes:

1. *Blanked portions of structured lines are also detectable.*
2. *GSYMB may be used to draw a blanked square cursor for detection and may be followed by other buffer orders to draw a cursor symbol of a different shape. However, the detection point will still be the edge of the square. (If, for example, a crosshair cursor is used as a symbol, the SIZE parameter of the GYSMB macro should be set so that the square is minimum size.)*

Data Format

The asynchronous data stream transmitted by the tablet usually consists of 10 bytes (Figure 3), each formatted in the following way:

(Start Bit) 1 2 3 4 5 6 7 8 (Stop Bit)

Byte	Bits	Contents	Meaning
1	1-4	BCD Status Code	Hex'0' = Switch operated, point mode Hex'1' = Switch operated, run mode Hex'2'- Hex'7' = (Reserved) Hex'8' = Switch not operated, point mode Hex'9' = Switch not operated, run mode
2	1-4	X Data	X13 - X16
3	1-4	X Data	X9 - X12
4	1-4	X Data	X5 - X8
5	1-4	X Data	X1 - X4
6	1-4	Y Data	Y13 - Y16
7	1-4	Y Data	Y9 - Y12
8	1-4	Y Data	Y5 - Y8
9	1-4	Y Data	Y1 - Y4
10	1-7	End of Data	1011000 = carriage return

Note: *Bit 8 is set low for each byte.*

Figure 3. Data Stream from Tablet

Additional Buffer Orders for Tablet Attachment

GSPOS - Store X,Y Position

The 3-word buffer order GSPOS is described below.

0 7	8 15	
0010 1010	1110 1000	(Hex 2AE8)
0000 UUUU	ITBO 0000	(Flags) - Extended Machine Code
AAAA AAAA	AAAA AAAA	(Address) - Address Field Code

UUUU: Four-bit number in device position register
= 0000, current beam position
= 0001, tablet register
= 0010 to 1111 (reserved)

I: Enables an interrupt from the position entry device, unit number UUUU (tablet or light pen). An interrupt occurs immediately if the tip-switch is present and was closed on the unit designated at the time of the last GSRT order.

Note: *If Continuous refresh function is used for a device address associated with a tablet, the I bit should not be set.*

T: If T = 1, the tip-switch indicator (TSI) is set at GSRT time following the execution of GSPOS for the unit for which UUUU holds the device address. The TSI remains set until another GSPOS occurs (with a different device address, and T = 1) or until the following GSRT.

B: Blanking flag. If B = 1, the data is stored with the appropriate blanking bit set.

GSPOS causes 2 words to be stored, starting at the address given in the address field. These stored words contain the X,Y data fetched from the specified unit when the most recent Start Regeneration Timer order was executed (that is, at the last GSRT(E) time). They are stored in the format of 3250 X,Y graphic data (the blanking bit set according to the B bit of GSPOS). The low order bit of the address is ignored.

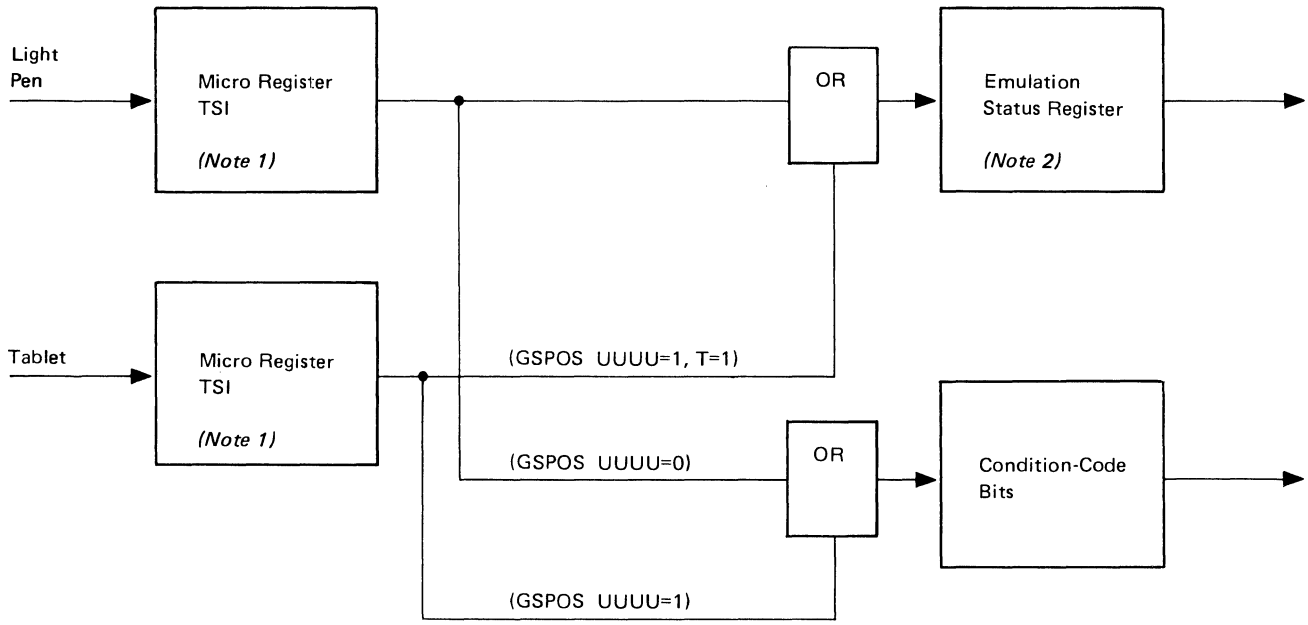
Notes:

1. *The interrupt generated by GSPOS (with the I bit set) gives attention and unit-check status, and bit 3 set in sense byte 1 to indicate a GSPOS interrupt; bit 0 of byte 1 (LPI bit) is also set. A subsequent Read X,Y Position Register command gives the X,Y data defining the position of the stylus on the tablet surface. This data is retained as the position coordinates until the display is restarted.*
2. *The GSPOS order does not reset the X,Y data and switch position, and a subsequent GSPOS order to the same tablet (in the same refresh cycle) will give identical results.*

The GSPOS order sets the condition code according to the switch condition of the unit (sampled at the previous GSRT) until the next sequential non-GTOC order. If the switch is open, the condition code is 00; if the switch is closed, the condition code is 01.

GSPOS Implementation

Figure 4 shows the relationship between GSPOS orders and the setting of the TSI and condition-code bits (when the order is used with a tablet or light-pen).



Notes:

1. *The micro register TSI bits indicate the state of the tip-switches of attached devices when the GSRT order is executed.*
2. *For the light-pen, TSI in the emulation status register is also updated at the associated GSRT order to indicate the current state of the light-pen.*

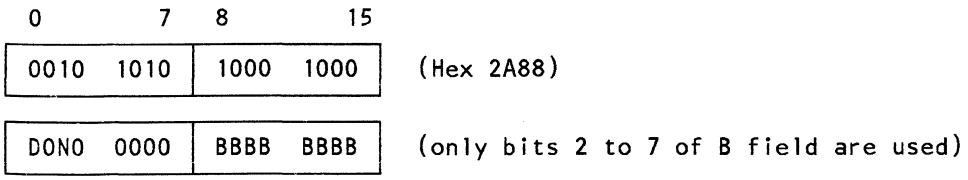
Figure 4. TSI and Condition-Code Correlation

If the address field in a GSPOS order contains the address of one of the attached devices, the state of the micro register TSI for that device is transferred to the condition code. This method of buffering enables the application programmer to include multiple GSPOS orders within the same refresh cycle in order to read the state of all attached devices. When the device address is that of a device other than a light pen, the state of the associated micro register TSI can be transferred to the emulation status register if the T bit in the order is also set to 1. Transferring this allows the programmer to use other devices (such as the tablet) to emulate the operation of a light-pen.

Although the design of the custom feature allows many types of device to coexist on the same system, the application programmer must take care, when creating the necessary orders, that the operator will not be confused about which device is doing what in the application.

GSYMB - Draw Symbol

The 1-word GSYMB buffer order is followed by a 1-word control field of the following format:



D: If D = 1, the symbol is used as a detection window

N: If N = 1, the symbol is blanked on the screen; it may still be used to control detection.

B: Size of symbol, from a minimum of 1.19 millimeters (0.047 inch, five addressable points) up to a maximum of 18.75 millimeters (0.738 inch, 64 addressable points).

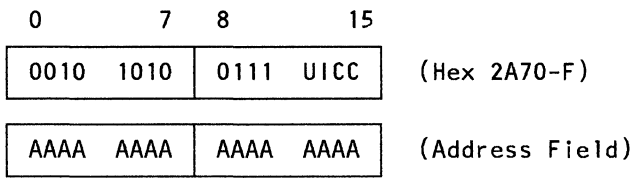
The symbol drawn is a square outline centered on the current beam position. The length of each side of the square is specified by the B bits as the number of addressable points, from a minimum of five to a maximum of 64.

Notes:

1. *If D equals 1, a subsequent drawing operation that causes a vector, point, or character to intercept the defined symbol area also causes a simulated light-pen detection and is interpreted under the same control modes as a light-pen detection.*
2. *The detection symbol remains active until it is reset by a GSRT(E) order, or a new symbol is provided by a subsequent GSYMB order.*
3. *The tip-switch indicator may have been set by a previous GSPOS order, depending on the referenced unit (that is, a light pen or tablet switch).*
4. *GSYMB may be used to draw a blanked square cursor for detection and may be followed by other buffer orders to draw a cursor symbol of a different shape. However, the detection point will still be the edge of the square. If, for example, a crosshair cursor is used as a symbol, the SIZE parameter of the GSYMB macro should be set so that the square is minimum size.*

GTOC - Transfer On Condition

The 2-word GTOC buffer order is as follows:



U: If U = 0, the two low-order bits of the order (CC) are compared to the condition code bits.
If U = 1, an unconditional branch is taken.

I: If I = 0, a transfer occurs if the condition is not satisfied.
If I = 1, a transfer occurs if the condition is satisfied.

The GTOC buffer order transfers control to the address given in the second word of the order if the conditions specified by the three low-order bits of the order code (ICC) are satisfied (that is, if CC is arithmetically equal to the condition code).

Figure 5 shows how the condition-code bits are interpreted, according to the preceding order.

	Condition if the Previous Order Is:		
Condition-Code Bits	GSPOS	GTM	Any Other Order (Note 1)
00	Switch is open (Note 2)	Selected bits are all '0's	Unconditional branch
01	Switch is closed	Selected bits are mixed '1' and '0'	Switch is closed
10	(Reserved)	(Reserved)	LPI is set (Note 2)
11	(Reserved)	Selected bits all ones	LPI is set and switch is closed (Note 2)

Notes:

1. *This is one of the orders that can modify the TSI and LPI bits for the device (that is, GPDI, GTDD, GTND, or GTSO). A GTOC following one of these orders must cause a valid transfer according to the setting of the condition-code bits at the time the GTOC occurs.*
2. *The light-pen interrupt (LPI) is reset in these cases.*
3. *The switch tested can be either the light pen switch or the tablet switch, depending on any previous GSPOS as defined by the unit number in the immediately preceding GSPOS.*
4. *The execution of GTOC does not reset the conditions (except the LPI); thus, an immediately following GTOC can retest the condition.*
5. *Any buffer order that results in the setting of the condition-code bits must be followed by a GTOC order to ensure that the transfer is made before subsequent orders reset the condition-code bits to different values.*

Figure 5. Condition Code Interpretation

Part 3. Plotter Attachment Custom Feature

Chapter 4. Plotter Attachment Feature: Overview

RPQs 7J0071 and 7J0072 (as described in Part 1) provide the facilities for attaching one non-IBM plotter to an IBM 3255 Display Control, using a general-purpose interface. The attached plotter is controlled by the host system through a buffer program in the 3255. A wide choice of plotter types (such as electrostatic, drum, or pen plotters) may be used. The Plotter Attachment custom feature enables the user to obtain hard copy of displayed images at the display station work area. The host program determines which IBM 3251 Display Station(s) may be used to request the hard copy.

The host system transmits data to the plotter via the 3255; the function of the 3255 is to buffer the data before transmitting it to the plotter. All formatting of the plot data (such as scaling and rotation) is done by the host system, and all plot data is transferred in vector form. Specific commands from the host application program to control the plotter are embedded in the data. If vector-to-raster conversion is required, for performance reasons it is recommended that the plotter or plotter controller does the conversion, and not the host application program. Transmission from the 3255 to the plotter is made using an asynchronous line control or a subset of the IBM binary synchronous line control. The host program inserts all transmission framing characters before sending the data to the 3255. This allows flexibility in the 3255 microcode, enabling device types with different protocols to be attached to the RS-232-C communications port of the 3255.

A plotter is connected to the 3255 by a 9.14-meter (30-foot) multiway cable; the user may connect a plotter directly to the cable, or via a similar extension cable that does not exceed 6 meters (20 feet) in length. A 25-way 'D' series connector on the 3255 provides exit/entry for signals to and from the attached plotter. Signal levels at the interface conform to electrical specifications EIA RS-232-C and CCITT V.24.[3] The data transmission rate is 150 to 9600 bps for asynchronous transmission, and 600 to 19200 bps for binary synchronous transmission.

Configuration

The following configuration rules apply when the Plotter Attachment custom feature is installed in an IBM 3250 Graphics Display System:

- A system may contain a mixture of basic 3255s and 3255s with plotter attachments.
- A system based on a single IBM 3258 Control Unit may contain up to sixteen addressed devices (plotters and 3251s) in any combination.
- In a system based on a single 3258, each 3255 with a plotter attachment may have a maximum of three 3251s and one plotter attached to it.

A description of the RPQs by which this feature may be purchased is provided in Part 1.

Note: *Example macros and an example program provided in Appendixes C and D have been tested with an electrostatic plotter.*

---

[3] EIA: Electrical Industries Association
CCITT: International Telegraph and Telephone Consultative Committee

Control Facilities

Buffer Orders

The host system controls the attached plotter through a dedicated buffer program in the 3255. The subset of 3250 buffer orders used is:

GEOS	End Order Sequence
GNOP2	No-Operation (2-Byte)
GNOP4	No-Operation (4-Byte)
GMVA	Move Immediate Address
GMVD	Move Immediate Data

IBM 3250 Graphics Display System: Component Description, GA33-3037, gives details of the above orders.

The following additional buffer orders are provided for the feature:

GIO	Input/Output Control
GTOC	Transfer on Condition
GTM	Transfer under Mask
GTRCT	Transfer on Count

See Chapter 6 for a description of these orders.

Channel Commands

The IBM System/370 channel commands used with the Plotter Attachment custom feature are:

Write Buffer
Read Buffer
Set Buffer Address Register and Start
Set Buffer Address Register and Stop
No-Operation

Diagnostic Routines

The 3255's internal diagnostic routines perform checks on the RS-232-C attachment port and interface adapter. The routines run when selected and, automatically, during initialization. The diagnostic routines do not check the attached plotter.

Chapter 5. Plotter Attachment Feature: Interface Characteristics

This chapter discusses the following aspects of the interface between a plotter and the plotter attachment port in the 3255 Display Control:

- Mechanical characteristics of the cabling
- Functions of the interface hardware
- Electrical characteristics and the signals that flow across the interface
- Protocols for binary synchronous and asynchronous communications, and communication speeds.

Mechanical Interface Characteristics

A plotter is connected to the attachment port of the 3255 Display Control by a multiway cable which is supplied with the plotter-attachment RPQ. The cable is plugged into a 25-way 'D' series socket mounted in the rear of the 3255.

The multiway cable has a grounded electrical screen, and is terminated at the plotter end with a 25-way 'D' series socket connector, and at the 3255 end with a 25-way 'D' series plug connector. The cable is 9.14 meters (30 feet) long, and the effective shunt capacitance (of the cable plus the terminators) does not exceed 1500 picofarads.

Note: *The user may connect a plotter directly to this cable, or via a similar extension cable (which must be supplied with the non-IBM plotter) which does not exceed 4 meters (20 feet) in length and has an effective shunt capacitance (of the cable plus the terminators) not exceeding 1000 picofarads. Any deviation from the feature signal/pin allocation that may exist in some plotters or interface equipment should be corrected within this extension cable.*

Functions of Interface Hardware

The plotter attachment port is connected to interface circuits within the 3255. The interface:

- Inserts and removes initial and fill synchronization characters, and leading and trailing pad characters
- Generates and inserts cyclic redundancy check (CRC) characters
- Inserts data link escape characters when communication is made in transparent mode.

For asynchronous communication, the port hardware:

- Generates and inserts parity check bits for all transmitted data
- May be programmed to check the parity of received data.

Electrical Interface Characteristics

Figure 6 shows the interface signal conventions.

Interface Signals		
Function	Off	On
Voltage	Less than -3 volts	Greater than +3 volts
Binary State	1	0
Signal Condition	Marking	Spacing

Figure 6. Interface Signal Conventions for Plotter Attachment

The interface transmitter and receiver circuits conform to RS-232-C and CCITT V.24 specifications. Figure 7 shows the subset of RS-232-C signals that are available at the 25-way 'D' series connector:

Pin Number	RS-232-C Circuit	CCITT V.24 Circuit	Signal Name
1	AA	-	Protective Ground
2	BA	103	Transmitted Data
3	BB	104	Received Data
4	CA	105	Request to Send
5	CB	106	Clear to Send
6	CC	107	For diagnostic use only
7	AB	102	Signal Ground
20	CD	108.2	Data Terminal Ready
15	DB	114	Transmit Signal Element Timing
17	DD	115	Receive Signal Element Timing

Notes:

1. *Pin numbers not listed are not connected.*
2. *No power supplies are provided on the cable; the attached plotter must be powered separately.*

Figure 7. Interface Signals

Interface Signal Lines

The subset of RS-232-C signals provided at the interface (see Figure 7) enable attachment of a plotter device. The purpose of each signal line is as follows.

Protective Ground (Pin 1)

The 'protective ground' line is connected to the 3255 frame ground.

Signal Ground (Pin 7)

The 'signal ground' line is the common reference ground for all interface signals except 'protective ground'.

Transmitted Data (Pin 2)

The 'transmitted data' line transfers data from the attached device to the 3255. When inactive, this line must be held in a mark condition by the attached device.

Received Data (Pin 3)

The 'received data' line transfers data from the 3255 to the attached device. When inactive, this line is held in a mark condition by the 3255.

Transmit Signal Element Timing (Pin 15)

The 'transmit signal element timing' line provides synchronous clock pulses from the 3255 to clock the 'received data' signal (pin 3) to the attached device.

Receive Signal Element Timing (Pin 17)

The 'receive signal element timing' line provides synchronous clock pulses from the 3255 to clock the 'transmitted data' signal (pin 2) from the attached device into the 3255.

Data Terminal Ready (Pin 20)

The 'data terminal ready' line is set by the attached device to indicate that it is ready to take part in data transfer. The line may (or may not) be used by either the attached device or the 3255.

Request to Send (Pin 4)

The 'request to send' line is set by an attached device to indicate that it has data to send to the 3255.

Clear to Send (Pin 5)

The 'clear to send' line is set by the 3255 (after it has received 'request to send') to indicate that it is ready to accept data. The 3255 can also use this line to inhibit data transmission from the attached device.

Interface Communication Protocols

Data communication across the RS-232-C interface conforms to IBM binary synchronous and asynchronous communication protocols:

- 3255-to-Plotter: The feature interface adapter can transmit data and responses in asynchronous or binary synchronous mode.
- Plotter-to-3255: The feature interface adapter can receive data and responses in asynchronous mode. It can receive responses (ACK, NACK, and WACK) only in binary synchronous mode.

IBM Binary Synchronous Communications

Full details of the binary synchronous protocols used for the RS-232-C interface are provided in the following publications:

- *General Information - Binary Synchronous Communications,* GA27-3004
- *Component Information for the IBM 3780 Data Communication Terminal,* GA27-3063.

All text and control characters used in a transmission to the device are created and formatted by the application program in the host system. Similarly, all data and framing control characters received in a binary synchronous transmission from the attached device are sent to the host system for interpretation.

Note: *Only a subset of IBM binary synchronous protocol is supported by the interface adapter. In particular, SOH and ITB sequences are not supported.* [4]

---

[4] SOH Start-of-Heading character
ITB Intermediate-Text-Blank character

The following binary synchronous functions are supported for communication between the RS-232-C port and an attached device. Details of which facilities are provided by the interface adapter and which must be provided by the buffer program are given in the description of each function.

- Leading sync characters and trailing mark level
- Sync fill characters
- Cyclic redundancy check (CRC) character generation
- Error recovery procedures
- Transparent mode.

Leading Sync Characters and Trailing Mark Level

Before any transmission of data or control characters to the plotter, the interface adapter sends four leading sync characters to allow the receiving device to establish bit synchronism. At the end of the transmission, the 'received data' line is held at mark level until the beginning of the next transmission.

Sync Fill Characters

If the 3255 cannot maintain the transfer of data during the transmission of a data block, the interface adapter fills the gap with sync characters (in multiples of two) to maintain continuity of character-frame transmission. All sync characters inserted in this way must be removed by the receiving device.

Cyclic Redundancy Check Generation

CRC generation is done by the interface adapter. For a transmission to the plotter, the CRC accumulation is started by (but does not include) the STX character, and ends with (and includes) the ETX/ETB[5] character. The resultant CRC characters are appended by the hardware to the end-of-text characters. No sync or pad character is included in the CRC accumulation. During a sync fill situation, the CRC accumulation is stopped.

Response characters (that is, ACK, NACK, or WACK)[6] from the attached device to the 3255 do not have accompanying CRC characters; these response characters must be checked by the buffer program.

Note: *The reception of data from the plotter is not supported by the plotter attachment feature if the communication is in IBM Binary Synchronous mode.*

Error Recovery Procedures

Positive acknowledgments (ACK1 or ACK0) from the attached device must be recognized by the buffer program, which then responds by sending the next data block.

Negative acknowledgments (NACK) must also be recognized by the buffer program, which then starts a retransmission procedure.

---

[5] ETX End-of-Text character
ETB End-of-Transmission character

[6] ACK Acknowledge character
NACK Negative-acknowledge character
WACK Wait before transmit positive acknowledgment

Wait acknowledgments (WACK) must cause the buffer program to send an enquiry (ENQ) wait sequence until the attached device is again clear to receive data.

Lack of any response from the attached device causes the interface adapter to time out and to signal a time-out-error condition to the 3255. The error condition must be set and stored by the buffer program, using the sense operation, so that this sense data is available to the host program when the 3255 sends an interrupt.

At the end of transmission, the buffer program must send an end-of-transmission (EOT) character to indicate to the attached device that the data transfer is complete. No response will be received from the attached device, and the buffer program must omit the time out indication in sense data sent to the host program.

Transparent Mode

The transparent mode of operation allows the transmitting device to send data of any bit combination, thus allowing binary synchronous control characters to be transmitted as data within a text stream. To differentiate between data and control characters, the interface adapter prefixes each true control character (and sync fill character) with a DLE (including STX).[7] A DLE within the text is also prefixed by a DLE. The attached receiving device must always remove the first DLE of a double DLE encountered within the text of a received block. No inserted DLEs (including DLE STX and DLE SYN) is included in the CRC accumulation.

The following example shows how DLE may be used.

```
Application Data
  S                   D         E             E
  T <---- Text----> L <-----> O <---------> T
  X                   E         T             X
```

In transparent mode, this is converted by the 3255 to:

```
Data on RS-232-C line
  D S                     D D         E             D E
  L T <---- Text ----> L L <-----> O <---------> L T
  E X                     E E         T             E X

      E
Where O is not a control character but a transparent data character
      T
with the same code.
```

When in transparent mode, only complete blocks of data should be sent from the application program to the 3255; that is, every block should have an ETX or ETB at the end.

To support the use of transparent mode, the interface adapter recognizes the following binary synchronous control characters:

```
STX ETX ETB EOT DLE SYN ENQ
```

---

[7] DLE Data-Link-Escape character
STX Start-of-Text character

Asynchronous Communications

To support asynchronous communications, the feature interface hardware:

- Generates and inserts parity bits for all transmitted data
- Can be programmed to check the parity of received data.

The dedicated buffer program for the feature may be designed to respond to the asynchronous conventions of the attached plotter.

Note: *When 7-bit asynchronous data is being transferred from the attached device to the 3255 buffer, the parity bit 8 (most significant bit) is left on and must therefore be removed by the application program.*

Communication Speeds

The following communication speeds are available:

- 150, 300 bps (asynchronous only)
- 600, 1200, 2400, 4800, 9600 bps (binary synchronous or asynchronous)
- 19200 bps (binary synchronous only)

Chapter 6. Control of Attached Plotter

Summary of Control Facilities

The RS-232-C port (and hence the plotter) is allocated the lowest device address at the 3255 Display Control. A dedicated program in the 3255 display buffer uses a subset of 3250-system buffer orders, and some additional orders, to control the operation of the plotter and the interface adapter. All other orders are invalid and will be changed to no-operations.

Note: *Continuous-refresh function is not supported for a plotter feature using an RS-232-C port. Continuous-refresh commands may be addressed to 3251 Display Stations attached to the 3255, but not to the attached plotter.*

3250 Buffer Orders Used

The subset of 3250 system orders used in the buffer program are:

Mnemonic	Hex Code	Name
GNOP2	2A80	No-Operation (2-byte)
GEOS	2A81	End Order Sequence
GNOP2	2ABO-F	No-Operation (2-byte)
GNOP4	2ACO	No-Operation (4-byte)
GMVA	2AEB	Move Immediate Address
GMVD	2AEC	Move Immediate Data

These orders are described in the *3250 Component Description* manual.

Additional Buffer Orders for Plotter Attachment: Summary

The additional buffer orders provided are:

Mnemonic	Hex Code	Name
GIO	2A89	Input/Output
GTOC	2A70-F	Transfer On Condition
GTM	2AED	Transfer Under Mask
GTRCT	2AFO	Transfer On Count

Using these additional orders, the application program can:

- Start and stop communications
- Set the required communication parameters: character width, speed, binary synchronous or asynchronous transmission, transparent mode, auto-termination character, and parity convention
- Provide the communication protocol (including error recovery) appropriate to the attached device.

The orders are described in detail later in this chapter.

Channel Commands

The buffer program is loaded, started, and stopped by the same channel commands as is a 3251 Display Station. The program runs continuously until it reaches a GEOS order or receives a Stop command from the channel. The channel commands used with the plotter attachment feature are:

Write Buffer
Read Buffer
Set Buffer Address Register and Start
Set Buffer Address Register and Stop
No-Operation

These commands are described in the *3250 Component Description* manual.

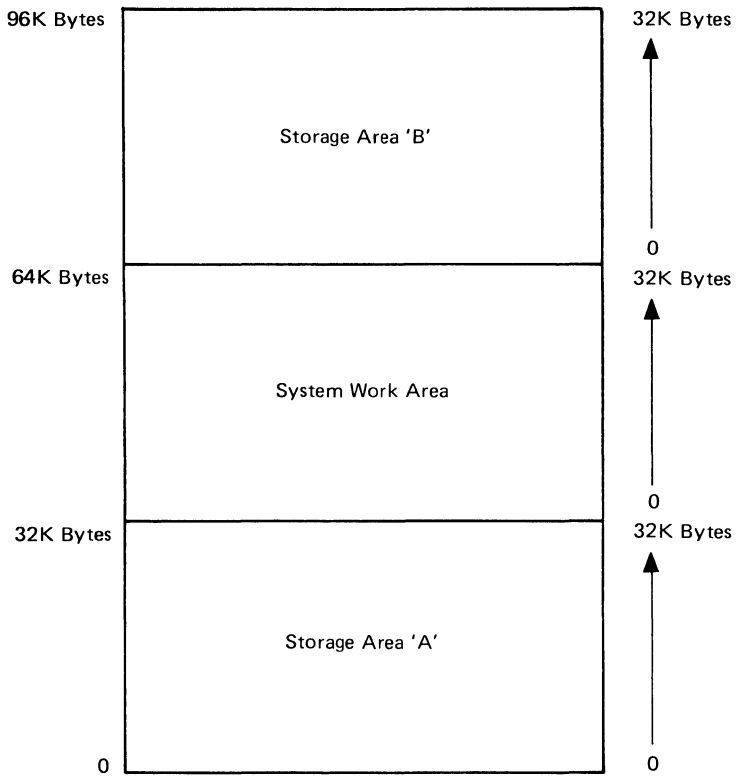
Channel commands not related to this feature cause the following responses from the 3255 if they are sent to a device address configured for an RS-232-C port and plotter attachment:

- For an Insert Cursor, Remove Cursor, or Read Cursor command: The device address has no associated keyboards, and so the response is 'command reject'.
- For a Read X,Y Position Registers command: An X,Y value of all zeros is returned to the channel.
- For a Read Manual Input command, or Set Program Function Indicators command: The response is 'command reject'.
- For a Set Audible Alarm command: This command is accepted without error but no operation is performed in the 3255.
- For a Sense command: Normal sense data is returned, but sense bits not related to the feature are meaningless and must be ignored.

Buffer Storage

The dedicated control program for the Plotter Attachment custom feature is allocated an area of buffer storage in the 3255. The feature provides additional buffer storage capacity, and buffer addresses higher than 32767 wrap around through 0 (see Figure 8). The allocation of buffer storage for each configuration is as follows:

Configuration	*Storage Allocation*
One 3251 Display Station	One 3251 in "A"
Two 3251s	One 3251 in "A", and one 3251 in "B"
Three 3251s	Two 3251s in "A", and one 3251 in "B"
Plotter and one 3251	Plotter in "A", 3251 in "B"
Plotter and two 3251s	Plotter and one 3251 in "A, one 3251 in "B"
Plotter and three 3251s	Plotter and one 3251 in "A", and two 3251s in "B"



Note: *K equals 1024*

Figure 8. Buffer Storage Areas

Buffer Operation

A GIO buffer order controls all plotter data sent by the host application program to the 3255. The GIO order points to an 8-byte input output control block (IOCB) containing command codes, flags, the data count field, and the address from where the plotter data will be read.

The Set Buffer Address Register and Start command starts the normal processing of graphic orders in the 3255 buffer; processing continues until a GIO order, pointing to an IOCB, is reached. This IOCB may then be used to cause the output configuration control (OCC) bytes to configure the RS-232-C port. Additional GIO orders and IOCBs, which point to buffer data addresses, then control the transfer of data to the plotter. IOCBs can be chained to transfer additional data if required.

Plotting begins when the plotter has interpreted the appropriate command. Plotting continues until a GIO termination sequence (such as the count field equals zero, or a Set Buffer Address Register and Stop command being given) occurs in the buffer. The buffer program runs continuously until a GEOS order occurs, or a Set Buffer Register and Stop command is received from the channel.

Additional Buffer Orders for Plotter Attachment: Details

This section describes, for the programmer, the additional buffer orders provided for the Plotter Attachment feature. Appendixes C and D give, respectively, example macros and an example program.

GIO - Input/Output Control

The 3-word GIO buffer order controls the transmission of data between the buffer storage and the RS-232-C port by causing appropriate action at the RS-232-C port interface adapter. The operation, or sequence of operations, to be done is defined by an input/output control block (IOCB) at the address indicated by the address field of the order. (The least significant bit of the address is ignored and assumed to be 0.)

0 7	8 15	
0010 1010	1000 1001	(Hex 2A89)
0000 0000	0000 0000	(Reserved)
AAAA AAAA	AAAA AAAA	(Address field of IOCB

Input/Output Control Block

The IOCB has a fixed length of 8 bytes and has the format shown in Figure 9.

Byte 0 0	Byte 1 15
Command Code	Flags
Count Field	
Reserved (or data bytes 0 and 1)	
Data Address (or data bytes 2 and 3)	

Figure 9. IOCB Format

Command Code: Command-code assignments are as follows:

0 7	
0000 0001	Write
0000 0010	Read
MM00 0011	Control
0000 0100	Sense
0000 1000	Transfer Control

Note: *All other values of the command code are invalid and cause termination of the GIO order. In the Control command code (above), bits MM provide a subcommand:*

Bits MM	Subcommand
00	No-Operation.
01	Set output configuration control (OCC) bytes from the data field.
10	Set input configuration control (ICC) bytes from the data field.
11	Set the termination character (TC) byte from the data field.

Note: The data field can be immediate (I=1) or located at a data address (I=0); see "Flag Byte" below. The count should be 2 for a Set ICC or Set OCC command, or 1 for Set TC command; otherwise, an incorrect length record is indicated. The count is ignored for no-operation.

Flag Byte: The flag byte has the format 0000 0ICS. When set, bits I, C, and S have the following meanings:

BIT I:	Immediate data; that is, the count can be up to 4, and data is contained in words 2 and 3 of the IOCB (control and write operations only).
BIT C:	Chaining flag. The next IOCB will be fetched and executed after the successful completion of this IOCB. The chained IOCB will be fetched from the next sequential IOCB location in buffer storage.
BIT S:	Suppress length indication. This inhibits the indication of incorrect length in the termination status.

Note: *Chaining is terminated if any condition causes the status (sense byte 0) to be non-zero.*

Port-Control Operations

The port-control commands cause the RS-232-C port adapter to:

- Set up the conditions for a read operation by setting the ICC register and the termination character.
- Set up the conditions for a write operation by setting the OCC register.
- Perform a no-operation.

The TC byte is used to terminate read operations.

OCC Bytes: Figure 10 shows the format of the OCC byte 1, Figure 11 shows the the format of the OCC byte 2, and Figure 12 gives the transmitting speeds specified by OCC byte 1.

Bit	Value	Function
0 (MSB)	0	(Reserved)
1	0 1	'Data terminal ready' RS-232-C signal is used 'Data terminal ready' is ignored
2 3 4	X X X	Transmitting speed (see Figure 12)
5-6	0	(Reserved)
7 (LSB)	0 1	Odd parity is generated Even parity is generated

Notes:

1. *MSB: Most significant bit.*
2. *LSB: Least significant bit.*

Figure 10. Format of OCC Byte 1

Bit	Value	Function
0 (MSB)	1	Transparent mode
1	0 1	Asynchronous transmission Binary synchronous transmission (see note 2)
2	0 1	A 7-bit data byte (asynchronous) An 8-bit data byte (asynchronous)
3	0 1	Two stop bits are used (asynchronous) One stop bit is used (asynchronous)
4-7	0	(Reserved)

Notes:

1. *MSB: Most significant bit.*
2. *Bits 2 and 3 are ignored if bit 1 is 1.*

Figure 11. Format of OCC Byte 2

Bit			Transmission Speed (bps)
2	3	4	
0	0	0	150
0	0	1	300
0	1	0	600
0	1	1	1200
1	0	0	2400
1	0	1	4800
1	1	0	9600
1	1	1	19200

Note: *Speed selection is limited to 150 through 9600 bps for asynchronous transmission, and to 600 through 19200 bps for synchronous transmission.*

Figure 12. Transmitting Speeds Specified in OCC Byte 1

ICC Bytes: Figure 13 shows the format of the ICC byte 1, Figure 14 shows the format of the ICC byte 2, and Figure 15 shows the received speeds specified by ICC byte 1.

Bit	Value	Function
0 (MSB)	0	(Reserved)
1	0 1	Only activate 'clear to send' when 'request to send' is on 'Request to send' is ignored ('clear to send' is permanently on)
2 3 4	X X X	Received speed (see Figure 15)
5	0 1	Auto-terminate mode Nonauto-terminate mode
6	0 1	No parity check on data Parity check on data
7 (LSB)	0 1	Receive odd parity Receive even parity

Notes:

1. *MSB: Most significant bit.*
2. *LSB: Least significant bit.*
3. *If a line speed of 4800 bps or greater is used, set bit 1 of byte 1 to 0. This action prevents the possibility of data loss if the attached device responds before the GIO Read command is ready.*

Figure 13. ICC Byte 1

Bit	Value	Function
0 (MSB)	1	(Reserved)
1	0 1	Asynchronous transmission Binary synchronous transmission (see Note 2)
2	0 1	A 7-bit data byte (asynchronous) is used An 8-bit data byte (asynchronous) is used
3	0 1	Two stop bits (asynchronous) are used One stop bit (asynchronous) is used
4-7	0	(Reserved)

Notes:

1. *MSB: Most significant bit.*
2. *Bits 2 and 3 are ignored if bit 1 is 1.*

Figure 14. ICC Byte 2

Bit			Receive Speed (bps)
2	3	4	
0	0	0	150
0	0	1	300
0	1	0	600
0	1	1	1200
1	0	0	2400
1	0	1	4800
1	1	0	9600
1	1	1	19200

Note: *Speed selection is limited to 150 through 9600 bps for asynchronous transmission, and to 600 through 19200 bps for synchronous transmission.*

Figure 15. Receive Speeds Specified in ICC Byte 1

Read Command

The Read command starts the transfer of data from the RS-232-C port to the buffer address indicated in the IOCB. Data transfer continues in the manner indicated by the ICC and TC bytes until it is terminated. The type of termination depends on whether the 3255 is in auto-terminate mode or nonauto-terminate mode.

3255 in Auto-Terminate Mode: The read data transfer is terminated by one of the following:

- The detection of the specified Termination Character in the received data. The Termination Character is not stored.
- A delay of 1.2 seconds from receipt of the last character if the TC byte is zero.
- The count in the IOCB reaching 0.
- A parity error detected on the RS-232-C interface (if bit 6 equals 1 in the ICC byte 1).
- The receipt of a Start or Stop command from the channel.

3255 in Nonauto-Terminate Mode: The read data transfer is terminated by one of the following:

- A delay of 1.2 seconds from receipt of the last character.
- The count in the IOCB reaching 0.
- A parity error detected on the RS-232-C interface (if bit 6 equals 1 in the ICC byte 1).
- The receipt of a Start or Stop command from the channel.

Write Command

The Write command starts the transfer of data from the buffer storage, at the address given in the IOCB, to the RS-232-C port. Data transfer continues in the manner indicated in the OCC bytes until it is terminated by one of the following:

- The count in the IOCB reaching 0.
- 'Data terminal ready' not signalled for 5 seconds (if bit 1 equals 0 in OCC byte 1).
- A Start or Stop command from the channel.

Transfer Control Command

The Transfer Control command causes no operation to be performed, except that a new IOCB is fetched from the data address.

Note: *The flag byte is ignored.*

Sense Command

The Sense command causes the status of the RS-232-C adapter to be stored in the buffer at the data address given in the IOCB. A maximum of 8 bytes is stored, depending on the count field. The format of the sense data is as follows:

- Byte 0 is the status byte.
- Byte 1 is reserved.
- Bytes 2 and 3 contain the residual count from the preceding operation.
- Bytes 4 and 5 are the current setting of the configuration register.
- Bytes 6 and 7 contain the address of the last IOCB used.

The status byte has the format 000I RCTE. When set, bits I, R, C, T, and E have the following meanings:

Bit I: This is set if the command code (byte 0) or the subcommand code (MM) contains an invalid bit combination. It is reset otherwise.

Bit R: Overrun error bit. This is set in synchronous mode if a character is lost from the received data coming in. It is set in asynchronous mode if there is a framing error in the asynchronous received data.

Bit C: This is set by the following conditions:
If the count $\neq$ 2 for a GIO Set ICC/OCC command.
If the count $\neq$ 1 for a GIO Set TC command.
If the count $>$ 8 for a GIO Sense command.
If the count $>$ 4 for a GIO write command with 'Immediate Flag Bit' set.

Bit T: Timeout bit. This is set under the following conditions:

In asynchronous mode, if the time between received characters exceeds 1.2 seconds. The timeout does not start until the first character arrives.
In binary synchronous mode, setting occurs (1) if no character is received within 5 to 6 seconds after the line has been turned round; or (2) if no character is received within 1.2 seconds after the first and subsequent characters have been received (until the IOCB count reaches 0).

Note: *"GIO Timeouts" (below) gives further information.*

Bit E: This is set if an asynchronous receive parity error is detected. It is reset otherwise.

Notes:

1. *The conditions indicated by a sense operation are reset by that sense operation, by a new GIO order, or by a new IOCB (except one having a Transfer Control command). The conditions are not reset by a Start or Stop command from the channel.*
2. *An incorrect length is indicated after a read operation if there is a difference between the IOCB count and the amount of actual data read when the TC character was encountered (unless the Suppress Length flag bit was set - refer to IOCB Flag Byte description).*

GIO Termination

The GIO order is not terminated until the operations indicated in the associated IOCB(s) have been completed.

Note: *A Start or Stop command from the channel to the RS-232-C port address terminates the GIO, even in the middle of an operation. The appropriate status is stored for a later sense operation.*

After a GIO order has been terminated, the condition code is set to indicate the result of the operation(s):

Condition Code	Condition
00	Normal termination
01	Abnormal termination (non-zero status byte in sense data)
10	(Not used)
11	(Not used)

GIO Timeouts

Two timeouts are used:

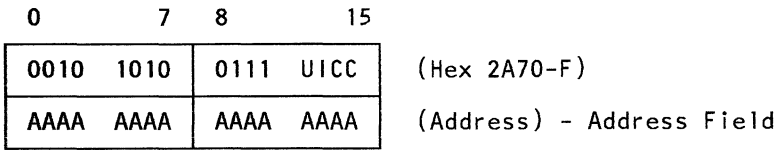
- *5-Second Timeout:* This is used only when the system is configured in binary synchronous receive mode.

  After a receive IOCB has been decoded by the interface adapter, the GIO order is terminated if a delay of greater than 5 seconds occurs before the IOCB count is decremented. When the IOCB count has been decremented once, the 5-second timeout is disabled and the 1.2-second timeout is enabled.

- *1.2-Second Timeout:* The GIO order is terminated if a delay of greater than 1.2 seconds occurs between the data characters received from the plotter after a receive operation has started, that is, after the IOCB count has been decremented by at least 1.

The occurrence of either timeout sets the timeout bit (bit T of byte 0 in the sense data).

GTOC - Transfer On Condition



Legend

U: If U=0, the two low-order bits of the instruction code are compared with the condition code.
If U=1, an unconditional branch is taken.

I: If I=0, a transfer occurs if the condition is not satisfied.
If I=1, a transfer occurs if the condition is satisfied.

The GTOC order transfers control to the address given in the second word of the order if the conditions specified by the three low-order bits of the order code (ICC) are satisfied (that is, if CC is arithmetically equal to the condition-code bit settings).

Figure 16 shows how the condition-code bits are interpreted, depending on the feature installed and the preceding order.

Condition-Code Bits	Condition if the Preceding Order is:			
	GSPOS	GIO	GTM	Any Other Order (Note 1)
00	Switch is open	Normal termination	Selected bits all 0's	Unconditional branch
01	Switch is closed	Abnormal termination	Selected bits mixed (1 and 0)	Switch is closed
10	(Reserved)	(Reserved)	(Reserved)	LPI is set (Note 2)
11	(Reserved)	(Reserved)	Selected bits all 1's	LPI is set and tip switch is closed (Note 2)

Figure 16. Condition Code Interpretation

Notes:

1. *This is one of the orders that can modify the TSI and LPI bits for the device (that is, GPDI, GTDD, GTND, GTSO).*[8] *A GTOC following one of these orders must cause a valid transfer according to the setting of the condition-code bits at the time the GTOC occurs.*
2. *The execution of GTOC does not reset the condition code bits (including the LPI). Thus an immediately following GTOC can retest the condition bits.*
3. *A Set Buffer Address Register and Start command resets the condition-code bits to 0's.*
4. *Any buffer order that results in the setting of the condition-code bits must be followed by a GTOC order to ensure that the transfer is made before later orders reset the condition-code bits to different values.*

[8] GPDI Permit Detect Interrupt
GTDD Transfer On Deferred Detect
GTND Transfer On No Detect
GTSO Transfer On Switch Open

0 7	8 15	
0010 1010	1110 1101	(Hex 2AED)
AAAA AAAA	AAAA AAAA	(Address of data to be tested)
BBBB BBBB	BBBB BBBB	(Mask bits)

This order enables two bytes of buffer storage to be tested under the control of a mask formed by the 16 bits of the second operand. These bits correspond one-for-one with the word in storage specified by the first operand address, and the condition code is set by the state of the bits selected by the mask.

A mask bit of 1 indicates that the storage bit is to be tested, and when the mask bit is 0 the storage bit is ignored. When all storage bits selected in this way are 0, the condition code is set to 00. (The condition code is also set to 00 when the mask is all 0's.) When the selected bits are all 1's, the condition code is set to 11; otherwise, the condition code is 01. The word in storage is not changed.

The condition-code bits are set at the end of the operation as follows:

Condition Code	**Condition**
00	Result bits are all 0's or the mask is all 0's.
01	Result bits are a mixture of 1's and 0's.
10	(Reserved)
11	Result bits are all 1's.

The condition-code setting remains until the next order that can set the condition code is executed.

For example:

0 7	8 15		CC
0101 0111	0101 0111	Data to be tested	01
1000 0011	0000 0000	Mask bits	
0000 0011	0000 0000	Result (Bits with mask = 0 are never tested)	

0 7	8 15	
0010 1010	1111 0000	(Hex 2AF0)
AAAA AAAA	AAAA AAAA	Buffer address (address field)
CCCC CCCC	CCCC CCCC	16-bit count (data field)

Legend:

A: Denotes a buffer address. Control passes to this address unless the count field, after being decremented, is 0. The low-order bit of this address is ignored.

C: Denotes an immediate value. This is an unsigned 16-bit integer; the count field is overwritten by the decremented value when the instruction is executed.

The GTRCT order may be used in buffer program loops, for example, to transfer data to the buffer storage under the control of a count. Each time the order is executed, the count field is decremented. When the count value reaches 0, control passes to the next sequential instruction; otherwise control passes to the buffer address specified.

The count field is set to an initial value in the range 0 - 32767. If the count field is set to 0 before being decremented, the new value will be hex FFFF and the order will continue to execute.

Appendix A. Example Tablet Macros

This appendix contains descriptions of three macro instructions that have been written to support the Cursor Control Tablet Attachment custom feature. The macros are GSPOS, GSYMB, and GTOC, and they generate 3250 buffer orders of the same mnemonics. Their use is shown in Appendix B.

Format Coding Conventions

The format of the macros employ the following coding conventions:

- Uppercase (capital) letters, numbers and punctuation marks; code these exactly as shown. Exceptions to this convention are brackets [], braces {}, and the vertical stroke |. These three exceptions are never coded.
- Lowercase (small) letters and words represent variables for which specific information or specific values must be substituted when coding.
- Items or groups of items within brackets, [], are optional.
- Braces, {}, indicate grouping. Chose one operand from the group unless a default option is indicated.
- Underscoring indicates a default option. If no operand is written, the underscored operand is assumed.
- The vertical stroke, |, means exclusive "or". For example, A | B means that either A or B can be written.
- Operands are separated by commas.
- The last operand is always followed by a blank.

Positional operands (that is, operands shown in lower case letters) must be coded in the order shown in the appropriate format. Keyword operands (that is, operands shown in upper case letters) can be coded in any order. Commas must appear in place of omitted operands preceding the last-specified positional operand.

The permitted formats for operands are given in the sections dealing with each set of macro instructions.

Unless noted otherwise, where numeric values are indicated as macro operands, these are decimal integers and need not be preceded by leading zeros.

GSPOS Macro

The GSPOS macro causes the current screen or tablet coordinates to be stored at the given buffer location.

The format of the macro call is:

```
[symbol]   GSPOS   address[,label]
                   [,SOURCE={BEAM|TABLET}]
                   [,BLANK={YES|NO}]
                   [,ATTN={YES|NO}]
                   [,SET={YES|NO}]
```

GSPOS Parameters

address[,label] (mandatory, positional parameter)

The address parameter represents the address of the buffer location at which the screen or tablet coordinates are to be stored. The addresses are as specified by 3250 macros that use symbolic buffer addresses.

SOURCE= (optional. keyword parameter)

The SOURCE= parameter specifies whether the current beam position or the coordinates of the tablet pen are to be stored. 'BEAM' and 'TABLET' are the two valid settings; 'BEAM' is the default.

BLANK= (optional, keyword parameter)

The BLANK= parameter specifies whether the coordinates are to be stored with the blanking bit on or off. (If they are subsequently used in a vector list, a move or a draw will result depending on the setting of this bit.) 'YES' and 'NO' are the valid settings; 'NO' is the default.

ATTN= (optional, keyword parameter)

The ATTN= parameter is specifies whether an attention interrupt is to be raised if the tip-switch on the pen (light-pen or tablet pen) is depressed. The interrupt will cause control to be given to the host program. 'YES' and 'NO' are the valid settings; 'NO' is the default.

SET= (optional, keyword parameter)

The SET= parameter specifies whether the tip-switch indicator (TSI) is to be set by this order. The indicator can then be tested later in the buffer program. 'YES' and 'NO' are the valid settings; 'NO' is the default.

GSPOS Code List

Note: * *in column 72 indicates a continuation character.*

```
         MACRO
&NAME    GSPOS &ADDR,&GROUP,&SOURCE=BEAM,       STORE X,Y POSITION     *
               &ATTN=NO,&BLANK=NO,&SET=NO
         GBLA  &IHBBLC                          BUFFER LOCATION
         LCLA  &EMC                             EXTENDED MC BUILDER
&EMC     SETA  0                                EXT-MC INITIALIZATION
         AIF   ('&SOURCE' EQ 'TABLET').TAB      TABLET COORDINATES ?
         AIF   ('&SOURCE' EQ 'BEAM').ATT        SCREEN COORDINATES ?
         MNOTE 4,'SOURCE PARAMETER INVALID - SET TO BEAM'
         AGO   .ATT
.TAB     ANOP
&EMC     SETA  &EMC+256                         'TABLET' FLAGGED
.ATT     ANOP
         AIF   ('&ATTN' EQ 'YES').ATTNY         ENABLE INTERRUPT ?
         AIF   ('&ATTN' EQ 'NO').SET            DISABLE INTERRUPT ?
         MNOTE 4,'ATTN PARAMETER INVALID - SET TO NO'
         AGO   .SET
.ATTNY   ANOP
&EMC     SETA  &EMC+128                         'ENABLE' FLAGGED
.SET     ANOP
         AIF   ('&SET' EQ 'YES').SETY           SET  TIP SWITCH IND. ?
         AIF   ('&SET' EQ 'NO').BLNK            DONT SET SWITCH IND. ?
         MNOTE 4,'SET PARAMETER INVALID - SET TO NO'
         AGO   .BLNK
.SETY    ANOP
&EMC     SETA  &EMC+64                          'SET SWITCH' FLAGGED
.BLNK    ANOP
         AIF   ('&BLANK' EQ 'YES').BLNKY        SET BLANKING BIT ?
         AIF   ('&BLANK' EQ 'NO').EMCOUT        DONT SET BLANKING BIT ?
         MNOTE 4,'BLANK PARAMETER INVALID - SET TO NO'
         AGO   .EMCOUT
.BLNKY   ANOP
&EMC     SETA  &EMC+32                          'BLANK' FLAGGED
.EMCOUT  ANOP
         IHBGAM3                                ENSURE EVEN BOUNDARY
&IHBBLC  SETA  &IHBBLC+4                        ADD 4 FOR CODE AND EXTN
&NAME    DC    AL2(X'2AE8')                     GSPOS ORDER CODE
         DC    AL2(&EMC)                        EXTENDED MC CODE
         IHBGAM1 ,&ADDR,&GROUP                  ADDRESS
         MEND
```

GSYMB Macro

The GSYMB macro causes a square symbol to be drawn at the current beam position. This symbol may be used as a detection window.

The format of the macro call is:

```
[symbol]     GSYMB    [SIZE=n]
                      [,DETECT={YES|NO}]
                      [,BLANK={YES|NO}]
```

Note: *When using GSYMB in a buffer program, it must occur* before *the vector list in which GSYMB-detectable vectors occur.*

GSYMB Parameters

SIZE=n (optional, keyword parameter)

The SIZE=n parameter is specifies the size of graphic symbol required. The symbol is in the form of a square and 'n' represents the length of the sides of the square in raster units, (in 4096 space). In the byte allocated to the size parameter, the two least significant bits are ignored. This means that 'n' can be any valid expression in the range $0 < n < 256$. Ignoring the two least significant bits has the following results:

- When the value of n is between 252 and 255, only the six most significant bits are used, giving a vector length of 63, which is 64 addressable points on the screen.
- The default value n = 128 gives 33 addressable points on the screen.

DETECT= (optional, keyword parameter)

The DETECT= parameter specifies whether the symbol is to be used as a detection window. If yes, a subsequent draw that causes a vector, point, or character to intersect the defined symbol area will cause a simulated light-pen detection and will be interpreted under the same control modes as a normal light-pen detection. 'YES' and 'NO' are the valid settings; 'YES' is the default.

BLANK= (optional, keyword parameter)

The BLANK= parameter specifies whether the symbol is to be blanked on the screen. (It may still be used as a detection window.) 'YES' and 'NO' are the valid settings; 'NO' is the default.

GSYMB Code List

```
         MACRO
&NAME    GSYMB &SIZE=128,&DETECT=YES,&BLANK=NO  GRAPHIC SYMBOL
         GBLA  &IHBBLC,&IHBBLIM                 BUFFER LOC. AND LIMIT
         LCLC  &SIZECD                          CHARACTER SIZE
         LCLA  &CONTROL                         CONTROL FLAGS
         AIF   ('&DETECT' EQ 'YES').DET         DETECTION WINDOW ?
         AIF   ('&DETECT' EQ 'NO').CHKBLNK      NOT DETECTION WINDOW ?
         MNOTE 4,'DETECT PARAMETER INVALID - SET TO YES'
.DET     ANOP
&CONTROL SETA  &CONTROL+32768                   'DETECT' FLAGGED
.CHKBLNK ANOP
         AIF   ('&BLANK' EQ 'YES').BLNK         BLANK SYMBOL ?
         AIF   ('&BLANK' EQ 'NO').SIZE          NOT BLANKED SYMBOL ?
         MNOTE 4,'BLANK PARAMETER INVALID - SET TO NO'
         AGO   .SIZE
.BLNK    ANOP
&CONTROL SETA  &CONTROL+8192                    'BLANKED' FLAGGED
.SIZE    ANOP
         AIF   (T'&SIZE EQ 'O').DEFSZE          SIZE MISSING ?
         DC    0AL4(X'80000000'+(&SIZE)-1)      ERROR IF SIZE LT 1
         DC    0AL4(X'7FFFFFFF'-255+(&SIZE))    ERROR IF SIZE GT 255
&SIZECD  SETC  '&SIZE'                          VALID SIZE
         AGO   .VALOUT
.DEFSZE  ANOP
&SIZECD  SETC  '128'                            DEFAULT SIZE
.VALOUT  ANOP
         IHBGAM3                                ENSURE EVEN BOUNDARY
&IHBBLC  SETA  &IHBBLC+4                        UPDATE BUF-LOC COUNTER
         AIF   (&IHBBLIM GE &IHBBLC).OK         BUFFER OVERFLOW ??
         IHBERMAC 182                           YES - ERROR MSG
.OK      ANOP
&NAME    DC    AL2(X'2A88')                     GSYMB ORDER CODE
         DC    AL2(&CONTROL+&SIZECD)            CONTROL FIELD
         MEND
```

GTOC Macro

The GTOC macro transfers data to a specified buffer location under the control of the current setting of the condition code.

The format of the macro call is:

```
[symbol]  GTOC  address[,label]
                [,COND=condition]
                [,BRANCH={YES|NO}]
```

address[,label] (mandatory, positional parameter)

The address parameter represents the address of the buffer location to which a conditional transfer is to be made. Addresses are as specified by 3250 macros that use symbolic buffer addresses.

COND=condition (optional, keyword parameter)

The COND= parameter is specifies the condition to be satisfied. The value given is compared with the setting of the condition code (set by a previous graphic order). If they are equal, the condition is satisfied. 'Condition' may be any valid absolute expression in the range....
0 LE condition LE 3

If this parameter is not coded, then a transfer will occur unconditionally.

BRANCH= (optional, keyword parameter)

The BRANCH= parameter specifies whether a transfer is to take place if the condition is satisfied. 'YES' and 'NO' are the valid settings; 'YES' is the default.

If 'YES' is specified, then a branch will be taken to the given buffer location if the condition is satisfied. Otherwise the next sequential order will be executed.

If 'NO' is specified, then a branch will be taken to the given buffer location if the condition is not satisfied. Otherwise the next sequential order will be executed.

Note: *If the COND parameter is omitted, then a branch will always occur regardless of the setting of the BRANCH operand.*

GTOC Code List

```
         MACRO
&NAME    GTOC  &ADDR,&GROUP,&COND=,&BRANCH=YES  TRANSFER ON CONDITION
         GBLA  &IHBBLC                          BUFFER LOCATION
         LCLA  &BR,&UNC                         FLAGS
&BR      SETA  1                                INITIALIZE BRANCH FLAG
&UNC     SETA  0                                INITIALIZE UNCOND FLAG
         AIF   (T'&COND EQ 'O').UNCOND          UNCONDITIONAL ?
         DC    0AL4(X'80000000'+(&COND)-0)      ERROR IF COND LT 0
         DC    0AL4(X'7FFFFFFF'-3+(&COND))      ERROR IF COND GT 3
         AGO   .CHKBRCH                         GO CHECK BRANCH PARM
.UNCOND  ANOP
&UNC     SETA  1                                FLAG UNCONDITIONAL
.CHKBRCH ANOP
         AIF   ('&BRANCH' EQ 'YES').OUT         BRANCH=YES ?
         AIF   ('&BRANCH' EQ 'NO').NOBRCH       BRANCH=NO ?
         MNOTE 4,'BRANCH OPERAND INVALID - SET TO YES'
         AGO   .OUT
.NOBRCH  ANOP
&BR      SETA  0                                FLAG NO BRANCH
.OUT     ANOP
         IHBGAM3                                ENSURE EVEN BOUNDARY
&IHBBLC  SETA  &IHBBLC+2                        ADD 2 FOR ORDER CODE
&NAME    DC    AL2(X'2A70'+(&COND)+8*&UNC+4*&BR)  ORDER CODE
         IHBGAM1 ,&ADDR,&GROUP                  ADDRESS
         MEND
```

Appendix B. Example Tablet Programs

This appendix contains the following four example programs that show the use of the macros described in Appendix A. They are not typical application programs.

- *Program 1* shows the tablet used as a digitizer. The program allows up to 500 pointings to be taken without an interrupt being generated.
- *Program 2* also shows the tablet used as a digitizer, but in this case an interrupt is generated after each pointing.
- *Program 3* shows the tablet used for cursor control. The program allows the response to be deferred until after it is tested by the GTDD buffer order.
- *Program 4* also shows the tablet used for cursor control, but in this case an interrupt is generated after each simulated light-pen detection.

Tablet Program 1: Tablet as a Digitizer (Interrupt after 500 Pointings)

This program uses the GSPOS and GTOC orders. The application demonstrates how a tablet can be used as a digitizer. In this example, the buffer program is self-modifying, enabling 500 pointings to be taken and stored before host intervention is required.

The method used requires a large amount of buffer storage (12014 bytes for 500 pointings), but it demonstrates the feasibility of transferring free-hand drawings to a 3250 buffer.

After the first pointing has been made, a vector is drawn from the most recent pointing to the position on the screen that corresponds to the position of the tablet pen. Only when the tip-switch is depressed does the vector become a permanent feature of the drawing.

Note: *Although 500 pointings are taken by this program, they will have a many-to-one relationship with the tip-switch depressions, because the state of the tip-switch is examined on each regeneration cycle.*

Program List

```
START    GSRT
TRU      GTRU  POS001
POS001   GSPOS LOC001,SOURCE=TABLET,BLANK=YES
         GTOC  DRAW,COND=0          transfer if pen switch not closed
         GMVA  TRU+2,BADDR=POS002   change target of GTRU
         GTRU  DRAW
POS002   GSPOS LOC002,SOURCE=TABLET
         GTOC  DRAW,COND=0
         GMVA  TRU+2,BADDR=POS003
         GTRU  DRAW
         :     :
         :     :
POSn     GSPOS LOCn,SOURCE=TABLET
         GTOC  DRAW,COND=0
         GMVA  TRU+2,BADDR=POSn+1
         GTRU  DRAW
         :     :
         :     :
POS500   GSPOS LOC500,SOURCE=TABLET
         GTOC  DRAW,COND=0
         GMVA  TRU+2,BADDR=EOS
         GTRU  DRAW
EOS      GEOS
DRAW     GEVM
LOC001   GNOP2
         GNOP2
         :     :
         :     :
LOC500   GNOP2
         GNOP2
         GTRU  START
```

Note: *On receipt of an EOS or END attention, the host program terminates the application.*

Tablet Program 2: Tablet as a Digitizer (Interrupt after Each Pointing)

This program uses the GSPOS order. The application demonstrates how a tablet can be used as a digitizer. In this example, the host program is interrupted for each pointing.

The method used requires a much smaller amount of buffer storage than does tablet program 1 (2014 bytes for 500 pointings), but would not be suitable for free-hand drawing because of the time taken by raising and answering the interrupt.

After the first pointing has been made, a vector is drawn from the most recent pointing to the position on the screen that corresponds to the position of the tablet stylus; the vector moves to follow the movement of the stylus. When the tip-switch is depressed, the vector becomes a permanent feature of the drawing.

Note: *When GSPOS is used to generate interrupts to the host program, only one such interrupt is generated per tip-switch closure. Thus this example program is capable of taking 500 distinct pointings.*

PFK 01 can be used to blank out the next vector.

Program List

```
START    GSRT
         GSPOS LOC001,SOURCE=TABLET,ATTN=YES,BLANK=YES
         GEVM
LOC001   GNOP2
         GNOP2
         :     :
         :     :
LOC500   GNOP2
         GNOP2
         GTRU  START
```

When attentions occur, the host program performs the following:

- For GSPOS attentions:
  - If all 500 pointings are taken, the buffer program is terminated.
  - GSPOS with BLANK=NO, and the target address increased by 4, replaces the original GSPOS order, and the buffer program is restarted.
  - For PF Key 01 attentions: GSPOS with BLANK=YES replaces the original GSPOS order, and the buffer program is restarted.
  - For PF Key 31 attentions: The buffer program is reset to its initial state, and then restarted.
  - For END attentions: The application is terminated.

Tablet Program 3: Tablet for Cursor Control (Deferred Response)

This program uses the GSPOS and GSYMB orders. The application demonstrates the use of a tablet for cursor control.

The GSYMB order is used to generate a detection window (cursor), and the position of the cursor is determined by the position of the tablet stylus. The host program is ***not*** interrupted when a simulated light-pen detection is caused (by a combination of a vector intersecting the window and the stylus tip-switch being closed). Instead, the response is deferred and tested by GTDD orders within the buffer program.

First of all, the program draws the symbol with the default size. If the associated 3251 has a program function keyboard, however, the symbol size can be varied by means of the PF keys 00 and 01.

Nine detectable points are drawn. A vector is drawn from the center point to the point that is detected by the window.

Program List

```
START     GSRT
          GDRD
          GSPOS COORDS,SOURCE=TABLET,SET=YES,BLANK=YES
          GEVM
COORDS    GNOP2
          GNOP2
          GSYMB DETECT=YES,SIZE=128
          GEPM
          GDV   xxx1,yyy1,U
          GTDD  DETECT
          GEPM
          GDV   xxx2,yyy2,U
          GTDD  DETECT
          :     :
          :     :
          GEPM
          GDV   xxx9,yyy9,U          center point
          GTDD  DETECT
          GDPD
          GEVM
POINT     GNOP2
          GNOP2
          GTRU  START
DETECT    GSXY  POINT
          GTRU  START
```

When attentions occur, the host program performs the following:

- For PF key 00 attentions: The size of the cursor symbol is reduced, and the buffer program is restarted.
- For PF key 01 attentions: The size of the cursor symbol is increased, and the buffer program is restarted.
- For END attentions: The application is terminated.

Tablet Program 4: Tablet for Cursor Control (Interrupt after Each Detection)

This program uses the GSPOS and GSYMB orders. The application demonstrates the use of the tablet for cursor control. The GSYMB order generates a detection window (cursor), and the position of the cursor is determined by the position of the tablet stylus. The host is interrupted each time a simulated light-pen detection occurs (by the combination of a vector intersecting the cursor window and the stylus tip-switch being closed).

First of all, the program draws the symbol with the default size. If the associated 3251 has a program function keyboard, however, the symbol size can be varied by means of the PF keys 00 and 01.

Four detectable vectors are drawn, together with an indication of their direction and endpoint coordinates. An area on the screen is allocated for displaying the coordinates of the endpoint of the detected vector.

Program List

```
START    GSRT
         GSPOS COORDS,SOURCE=TABLET,SET=YES,BLANK=YES
         GEVM
COORDS   GNOP2
         GNOP2
         GSYMB DETECT=YES,SIZE=128
         GEVM
         GDV   xxxx,yyyy,b     (detectable vectors)
         :     :               :
         GDPD
         GEVM
         GDV   xxxx,yyyy,b     (nondetectable vectors)
         :     :               :
         GECP
         GTXT  'cccccccc'      (endpoint coordinates (text))
         :     :               :
         GTRU  START
```

When attentions occur, the host program performs the following:

- For simulated light-pen attentions: A READ X,Y is performed to determine the end-point of the detected vector.
- The coordinates are converted to character form, the area set aside for displaying them is rewritten, and the buffer program is restarted.
- For PF key 00 attentions: The size of the cursor symbol is reduced, and the buffer program is restarted.
- For PF key 01 attentions: The size of the cursor symbol is increased, and the buffer program is restarted.
- For END attentions: The application is terminated.

Appendix C. Example Plotter Macros

This appendix contains descriptions of six macro instructions that have been written to support the Plotter Attachment custom feature. Macros GIO, GTM, and GTRCT generate buffer orders of the same mnemonic, and macros GIOCB, GICC, and GOCC generate data blocks used by the GIO macro.

Note: *The macro GTOC, which is also used with the plotter feature, is described in Appendix A.*

Format Coding Conventions

The format of the macros employ the following coding conventions:

- Uppercase (capital) letters, numbers and punctuation marks; code these exactly as shown. Exceptions to this convention are brackets [], braces {}, and the vertical stroke |. These three exceptions are never coded.
- Lowercase (small) letters and words represent variables for which specific information or specific values must be substituted when coding.
- Items or groups of items within brackets, [], are optional.
- Braces, {}, indicate grouping. Chose one operand from the group unless a default option is indicated.
- Underscoring indicates a default option. If no operand is written, the underscored operand is assumed.
- The vertical stroke | means exclusive "or." For example, A | B means that either A or B can be written.
- Operands are separated by commas.
- The last operand is always followed by a blank.

Positional operands (that is, operands shown in lower case letters) must be coded in the order shown in the appropriate format. Keyword operands (that is, operands shown in upper case letters) can be coded in any order. Commas must appear in place of omitted operands preceding the last-specified positional operand.

The permitted formats for operands are given in the sections dealing with each set of macro instructions.

Unless noted otherwise, where numeric values are indicated as macro operands, these are decimal integers and need not be preceded by leading zeros.

GIO Macro

The GIO macro transmits data between the buffer and the plotter (output or input).

The format of the macro call is:

```
[symbol]  GIO   address[,label]
```

Address Parameter

address[,label] (mandatory, positional parameter)

The address parameter represents the address of the buffer location of an input/output control block (IOCB), which defines the operation or sequence of operations to be performed. (The IOCB can be generated by the GIOCB macro.)

Addresses are as specified by 3250 macros that use symbolic buffer addresses.

Condition Code

The condition code at the end of the operation is set as follows:

00: Normal termination
01: Abnormal termination
10: (Not set by this order)
11: (Not set by this order)

GIO Code List

```
         MACRO
&NAME    GIO    &ADDR,&GROUP                GRAPHIC INPUT / OUTPUT
         GBLA   &IHBBLC                     BUFFER LOCATION
&IHBBLC  SETA   &IHBBLC+4                   ADD 4 FOR CODE AND ZERO
         IHBGAM3                            ENSURE EVEN BOUNDARY
&NAME    DC     AL2(X'2A89')                GIO ORDER CODE
         DC     AL2(0)                      RESERVED - SET TO ZERO
         IHBGAM1 ,&ADDR,&GROUP              ADDRESS
         MEND
```

GIOCB Macro

The GIOCB macro may be used to define the IOCB addressed by a GIO macro instruction.

The format of the macro call is:

```
[symbol]  GIOCB  OPER=operation
                 [,LENGTH=n]
                 [,CHAIN={YES|NO}]
                 [,SUPPR={YES|NO}]
                 [,DATADDR=(address[,label])]
                 [,DATA=expression]
```

GIOCB Parameters

OPER=operation (mandatory, keyword parameter)

The OPER= parameter defines the operation to be performed. It must be one of the following:

WRITE, READ, NOOP, SETOCC, SETICC, SETTC, SENSE, or TRANSFER

LENGTH=n (optional, keyword parameter)

The LENGTH= parameter indicates the number of bytes involved in the Input/Output operation, where 'n' is any absolute expression. The parameter is mandatory for WRITE, READ, NOOP, and SENSE operations; it is ignored for SETICC, SETOCC, SETTC, and TRANSFER operations. If the operation is WRITE or NOOP and immediate data is specified (by using the DATA= operand), then 'n' must be in the range $0 < n < 5$.

CHAIN= (optional, keyword parameter)

'YES' and 'NO' are the valid settings; 'NO' is the default.

The CHAIN= parameter indicates whether or not this IOCB is to be chained to the next IOCB (starting at the next sequential location after this IOCB).

SUPPR= (optional, keyword parameter)

'YES' and 'NO' are the valid settings; 'NO' is the default.

If 'YES', then the indication of incorrect length (on termination of the operation) is inhibited.

DATADDR= (optional, keyword parameter)

The DATADDR= parameter represents the address of the buffer location at which the data is to be found or stored. The address is specified in the format shown, which is as specified by 3250 macros that use symbolic buffer addresses.

This parameter must not be specified for the operations WRITE, NOOP, SETOCC, SETICC, and SETTC if the DATA= parameter is used. In all other cases it is mandatory.

DATA= (optional, keyword parameter)

The DATA= parameter is only valid for the operations WRITE, NOOP, SETOCC, SETICC and SETTC; if DATA= is specified for these operations then DATADDR= must be omitted. The expression can be any absolute expression, allowing symbols and self-defining terms to be used if desired. The length of this data for each operation is as follows:

WRITE: 1-4 bytes
NOOP : 1-4 bytes
SETOCC: 2 bytes
SETICC: 2 bytes
SETTC: 1 byte

GIOCB Code List

Note: * *in column 72 indicates a continuation character.*

```
         MACRO
&NAME    GIOCB &OPER=,&LENGTH=,&CHAIN=NO,&SUPPR=NO,&DATADDR=,&DATA=
         GBLA  &IHBBLC,&IHBBLIM              BUFFER LOCATION AND LIMIT
         LCLA  &CDEFLG                       CODE BUILDER
         LCLB  &IMMED                        IMMEDIATE DATA FLAG
         LCLC  &CHARLTH                      INTERNAL CHARACTER LENGTH
         AIF   (T'&OPER EQ 'O').OPERR        OPERATION SPECIFIED ?
         AIF   ('&OPER' EQ 'WRITE').WRITE    WRITE ?
         AIF   ('&OPER' EQ 'READ').READ      READ ?
         AIF   ('&OPER' EQ 'NOOP').NOOP      NOOP ?
         AIF   ('&OPER' EQ 'SETOCC').SOCC    SET OUTPUT CONFIGURATION ?
         AIF   ('&OPER' EQ 'SETICC').SICC    SET INPUT CONFIGURATION ?
         AIF   ('&OPER' EQ 'SETTC').STC      SET TERMINATE CHARACTER ?
         AIF   ('&OPER' EQ 'SENSE').SENSE    SENSE ?
         AIF   ('&OPER' EQ 'TRANSFER').TR    TRANSFER CONTROL ?
         AGO   .OPERR                        ILLEGAL VALUE
.WRITE   ANOP
&CDEFLG  SETA  &CDEFLG+256                   CODE FOR WRITE OPERATION
         AIF   (T'&DATA NE 'O').CHKDATA      DATA SPECIFIED ?
         AGO   .CHKDADR                      GO AND CHECK DATADDR
.READ    ANOP
&CDEFLG  SETA  &CDEFLG+512                   CODE FOR READ OPERATION
         AGO   .CHKDADR                      GO AND CHECK DATADDR
.NOOP    ANOP
&CDEFLG  SETA  &CDEFLG+768                   CODE FOR NOOP OPERATION
         AIF   (T'&DATA NE 'O').CHKDATA      DATA SPECIFIED ?
         AGO   .CHKDADR                      GO AND CHECK DATADDR
.SOCC    ANOP
&CDEFLG  SETA  &CDEFLG+17152                 CODE FOR SETOCC OPERATION
         AIF   (T'&LENGTH EQ 'O').SOCC2      WAS LENGTH SPECIFIED ?
         MNOTE 4,'LENGTH PARAMETER IGNORED'
.SOCC2   ANOP
&CHARLTH SETC  '2'                           SET INTERNAL LENGTH
         AIF   (T'&DATA NE 'O').CHKDATB      DATA SPECIFIED ?
         AGO   .CHKDADB                      GO AND CHECK DATADDR
.SICC    ANOP
&CDEFLG  SETA  &CDEFLG+33536                 CODE FOR SETICC OPERATION
         AIF   (T'&LENGTH EQ 'O').SICC2      WAS LENGTH SPECIFIED ?
         MNOTE 4,'LENGTH PARAMETER IGNORED'
.SICC2   ANOP
&CHARLTH SETC  '2'                           SET INTERNAL LENGTH
         AIF   (T'&DATA NE 'O').CHKDATB      DATA SPECIFIED ?
         AGO   .CHKDADB                      GO AND CHECK DATADDR
.STC     ANOP
&CDEFLG  SETA  &CDEFLG+49920                 CODE FOR SETTC OPERATION
         AIF   (T'&LENGTH EQ 'O').STC2       WAS LENGTH SPECIFIED ?
         MNOTE 4,'LENGTH PARAMETER IGNORED'
```

```
.STC2    ANOP
&CHARLTH SETC  '1'                              SET INTERNAL LENGTH
         AIF   (T'&DATA NE 'O').CHKDATB         DATA SPECIFIED ?
         AGO   .CHKDADB                         GO AND CHECK DATADDR
.SENSE   ANOP
&CDEFLG  SETA  &CDEFLG+1024                     CODE FOR SENSE OPERATION
         AGO   .CHKDADR                         GO AND CHECK DATADDR
.TR      ANOP
&CDEFLG  SETA  &CDEFLG+2048                     CODE FOR TRANSFER OPERATION
&CHARLTH SETC  '1'
         AGO   .CHKDADB                         GO AND CHECK DATADDR
.CHKDATA ANOP
         AIF   (T'&LENGTH EQ 'O').LTHERR
&CHARLTH SETC  '&LENGTH'                        SET INTERNAL LENGTH
.CHKDATB ANOP
&IMMED   SETB  (1)                              FLAG IMMEDIATE DATA
         AIF   (T'&DATADDR EQ 'O').LTHLE4       DATADDR OMITTED ?
         MNOTE 4,'DATADDR PARAMETER IGNORED'
.LTHLE4  ANOP
         DC    0AL4(X'7FFFFFFF'-4+(&CHARLTH))  ERROR IF LENGTH GT 4
         AGO   .CHKCHN                          GO CHECK CHAIN OPERAND
.CHKDADR ANOP
         AIF   (T'&LENGTH EQ 'O').LTHERR
&CHARLTH SETC  '&LENGTH'                        SET INTERNAL LENGTH
.CHKDADB ANOP
&IMMED   SETB  (0)                              FLAG INDIRECT DATA
         AIF   (T'&DATA EQ 'O').ADR             DATA NOT SPECIFIED ?
         MNOTE 4,'DATA PARAMETER IGNORED'
.ADR     ANOP
         AIF   (T'&DATADDR NE 'O').CHKCHN       DATADDR SPECIFIED ?
         MNOTE 8,'DATADDR OR DATA PARAMETER NOT SPECIFIED - MACRO EXPAN*
               SION TERMINATED'
         MEXIT
.CHKCHN  ANOP
         DC    0AL4(X'80000000'+(&CHARLTH)-1)   ERROR IF LENGTH LT 1
         AIF   ('&CHAIN' EQ 'YES').CHAIN        CHAIN = YES ?
         AIF   ('&CHAIN' EQ 'NO').CHKSUP        CHAIN = NO ?
         MNOTE 4,'CHAIN PARAMETER INVALID - SET TO NO'
         AGO   .CHKSUP
.CHAIN   ANOP
&CDEFLG  SETA  &CDEFLG+2                        CODE FOR CHAINING
.CHKSUP  ANOP
         AIF   ('&SUPPR' EQ 'YES').SUPPR        SUPPR = YES ?
         AIF   ('&SUPPR' EQ 'NO').OUTPUT        SUPPR = NO ?
         MNOTE 4,'SUPPR PARAMETER INVALID - SET TO NO'
         AGO   .OUTPUT
.SUPPR   ANOP
&CDEFLG  SETA  &CDEFLG+1                        CODE FOR SUPPRESS
.OUTPUT  ANOP
         IHBGAM3                                ENSURE EVEN BOUNDARY
&IHBBLC  SETA  &IHBBLC+6                        ADD 6 FOR CODE,LENGTH,RESVD
         AIF   (&IMMED).IMMED                   IMMEDIATE DATA ?
&NAME    DC    AL2(&CDEFLG)                     COMMAND CODE AND FLAGS
         DC    AL2(&CHARLTH)                    LENGTH FIELD
         DC    AL2(0)                           RESERVED - NEXT COMES ADDR.
```

```
         IHBGAM1  ,&DATADDR(1),&DATADDR(2)    ADDRESS
         MEXIT
.IMMED   ANOP
&IHBBLC  SETA  &IHBBLC+2                      ADD 2 MORE FOR OTHER DATA
&CDEFLG  SETA  &CDEFLG+4                      FLAG IMMEDIATE DATA
&NAME    DC    AL2(&CDEFLG)                   COMMAND CODE AND FLAGS
         DC    AL2(&CHARLTH)                  LENGTH FIELD
         DC    AL4(0)                         DEFINE DATA FIELD AREA
         ORG   *-4
         DC    AL(&CHARLTH)(&DATA)            DATA
         ORG
         AIF   (&IHBBLIM GE &IHBBLC).EXIT     BUFFER OVERFLOW ?
         IHBERMAC 182                           YES - ERROR MSG
         MEXIT
.OPERR   ANOP
         MNOTE 12,'OPER PARAMETER INVALID - MACRO EXPANSION TERMINATED'
         MEXIT
.LTHERR  ANOP
         MNOTE 12,'LENGTH PARM MISSING - MACRO EXPANSION TERMINATED'
.EXIT    ANOP
         MEND
```

GICC Macro

The GICC macro may be used to define the two input configuration control (ICC) bytes. The macro is addressed by the DATADDR= operand of a GIOCB macro instruction with an operation code of SETICC.

The format of the macro call is:

```
[symbol]  GICC  TRANSM=type
                ,SPEED=n
               [,PARITY={NONE|ODD|EVEN}]
               [,IGNORE={YES|NO}]
               [,TRANSP={YES|NO}]
               [,AUTO={YES|NO}]
               [,DBITS=d]
               [,SBITS=s]
```

GICC Parameters

TRANSM=type (mandatory, keyword parameter)

The TRANSM= parameter defines whether binary synchronous or asynchronous transmission is to be used; 'type' must be either SYNC or ASYNC.

SPEED=n (mandatory, keyword parameter)

The SPEED= parameter defines the transmission speed. 'n' is the speed in bps, and must be one of the following:

150, 300, 600, 1200, 2400, 4800, 9600, or 19200

Note: *Speed selection is limited as follows:*

ASYNC: 150 - 9600 bps
SYNC: 600 - 19200 bps

PARITY= (optional, keyword parameter)

The PARITY= parameter specifies whether odd or even parity is checked on received data, or if no check is required. No parity checks are performed if this parameter is omitted.

IGNORE= (optional, keyword parameter)

The IGNORE= parameter specifies whether the 'request to send' (RTS) signal is to be ignored. If omitted, then 'clear to send' (CTS) will be activated only when RTS is on.

TRANSP= (optional, keyword parameter)

The TRANSP= parameter specifies whether received data is to be transparent. If the parameter is omitted, data will be received as nontransparent.

AUTO= (optional, keyword parameter)

The AUTO= parameter specifies whether read operations are to be terminated by the detection of the termination character. The default is 'YES'.

DBITS=d (optional, keyword parameter)

The DBITS= parameter is valid only if TRANSM=ASYNC. DBITS specifies whether the data byte is to be 7-bits or 8-bits. Any absolute expression is valid if the value is 7 or 8.

SBITS=s (optional, keyword parameter)

The SBITS= parameter is valid only if TRANSM=ASYNC. One or two stop bits may be specified. Any absolute expression is valid if the value is 1 or 2.

GICC Code List

Note: * *in column 72 indicates a continuation character.*

```
         MACRO
&NAME    GICC  &TRANSM=,&PARITY=NONE,&IGNORE=NO,&TRANSP=NO,&AUTO=YES,   *
               &SPEED=,&DBITS=,&SBITS=
         GBLA  &IHBBLIM,&IHBBLC              BUFFER LOCATION AND LIMIT
         LCLA  &ICC
         LCLC  &DBITSC,&SBITSC               CHARACTER REP. OF D/S BITS
         IHBGAM3                             ENSURE EVEN BOUNDARY
&IHBBLC  SETA  &IHBBLC+2                     UPDATE BUFFER-LOC COUNTER
         AIF   (&IHBBLIM GE &IHBBLC).OK      BUFFER OVERFLOW ?
         IHBERMAC 182                          YES - ERROR MSG
.OK      ANOP
&ICC     SETA  0                             INITIALISE CODE
         AIF   ('&IGNORE' EQ 'NO').SPEED     USE RTS SIGNAL ?
         AIF   ('&IGNORE' EQ 'YES').IGNORE   IGNORE RTS SIGNAL ?
         MNOTE 4,'IGNORE PARAMETER INVALID - SET TO NO'
         AGO   .SPEED
.IGNORE  ANOP
&ICC     SETA  &ICC+16384                    FLAG IGNORE RTS
.SPEED   ANOP
         AIF   (T'&SPEED EQ 'O').SERR        TRANS. SPEED SPECIFIED ?
         AIF   (&SPEED EQ 150).AUTO            150 BPS ?
         AIF   (&SPEED EQ 300).S1              300 BPS ?
         AIF   (&SPEED EQ 600).S2              600 BPS ?
         AIF   (&SPEED EQ 1200).S3            1200 BPS ?
         AIF   (&SPEED EQ 2400).S4            2400 BPS ?
         AIF   (&SPEED EQ 4800).S5            4800 BPS ?
         AIF   (&SPEED EQ 9600).S6            9600 BPS ?
         AIF   (&SPEED EQ 19200).S7          19200 BPS ?
         AGO   .SERR                         INVALID IF NONE OF THESE
.S1      ANOP
&ICC     SETA  &ICC+2048                     FLAG   300 BPS
         AGO   .AUTO                           THEN GO CHECK AUTO PARM
.S2      ANOP
&ICC     SETA  &ICC+4096                     FLAG   600 BPS
         AGO   .AUTO                           THEN GO CHECK AUTO PARM
.S3      ANOP
&ICC     SETA  &ICC+6144                     FLAG  1200 BPS
         AGO   .AUTO                           THEN GO CHECK AUTO PARM
.S4      ANOP
&ICC     SETA  &ICC+8192                     FLAG  2400 BPS
         AGO   .AUTO                           THEN GO CHECK AUTO PARM
.S5      ANOP
&ICC     SETA  &ICC+10240                    FLAG  4800 BPS
         AGO   .AUTO                           THEN GO CHECK AUTO PARM
.S6      ANOP
&ICC     SETA  &ICC+12288                    FLAG  9600 BPS
         AGO   .AUTO                           THEN GO CHECK AUTO PARM
.S7      ANOP
&ICC     SETA  &ICC+14336                    FLAG 19200 BPS
```

```
.AUTO    ANOP
         AIF   ('&AUTO' EQ 'YES').PRTY      AUTO - TERMINATE ?
         AIF   ('&AUTO' EQ 'NO').NOAUTO     NON-AUTO - TERMINATE ?
         MNOTE 4,'AUTO PARAMETER INVALID - SET TO YES'
         AGO   .PRTY
.NOAUTO  ANOP
&ICC     SETA  &ICC+1024                    FLAG NON-AUTO TERMINATE
.PRTY    ANOP
         AIF   ('&PARITY' EQ 'NONE').TRANS  NO PARITY CHECKING ?
         AIF   ('&PARITY' EQ 'ODD').ODD     ODD PARITY CHECKING ?
         AIF   ('&PARITY' EQ 'EVEN').EVEN   EVEN PARITY CHECKING ?
         MNOTE 4,'PARITY PARAMETER INVALID - SET TO NONE'
         AGO   .TRANS
.EVEN    ANOP
&ICC     SETA  &ICC+256                     FLAG EVEN PARITY
.ODD     ANOP
&ICC     SETA  &ICC+512                     FLAG PARITY CHECKING
.TRANS   ANOP
         AIF   ('&TRANSP' EQ 'NO').SYN      NON-TRANSPARENT ?
         AIF   ('&TRANSP' EQ 'YES').TRANSP  TRANSPARENT ?
         MNOTE 4,'TRANSP PARAMETER INVALID - SET TO NO'
         AGO   .SYN
.TRANSP  ANOP
&ICC     SETA  &ICC+128                     FLAG TRANSPARENCY
.SYN     ANOP
         AIF   (T'&TRANSM  EQ 'O').SYNERR   TRANSMISSION SPECIFIED ?
         AIF   ('&TRANSM' EQ 'ASYNC').ASYNC ASYNCHRONOUS ?
         AIF   ('&TRANSM' EQ 'SYNC').SYNC   SYNCHRONOUS ?
         AGO   .SYNERR                      ERROR IF NEITHER
.SYNC    ANOP
&ICC     SETA  &ICC+64                      FLAG SYNCHRONOUS
         AIF   (&SPEED LT 600).SERR         SPEED OK FOR SYNC ?
         AIF   (T'&DBITS  NE 'O').BITSIG    DBITS SPECIFIED ?
         AIF   (T'&SBITS  NE 'O').BITSIG    SBITS SPECIFIED ?
         AGO   .OUTPUT                      GO OUTPUT DATA
.BITSIG  ANOP
         MNOTE 4,'''DBITS'' AND/OR ''SBITS'' CODED - IGNORED FOR SYNC'
         AGO   .OUTPUT
.ASYNC   ANOP
         AIF   (&SPEED GT 9600).SERR        SPEED OK FOR ASYNC ?
         AIF   (T'&DBITS  EQ 'O').DBITERR   DBITS SPECIFIED ?
         DC    0AL4(X'80000000'+(&DBITS)-7) ERROR IF DBITS LT 7
         DC    0AL4(X'7FFFFFFF'-8+(&DBITS)) ERROR IF DBITS GT 8
&DBITSC  SETC  '&DBITS'
         AGO   .STOPB
.DBITERR ANOP
         MNOTE 4,'''DBITS'' INVALID OR OMITTED - 7 BITS USED'
&DBITSC  SETC  '7'
.STOPB   ANOP
         AIF   (T'&SBITS  EQ 'O').SBITERR   SBITS SPECIFIED ?
         DC    0AL4(X'80000000'+(&SBITS)-1) ERROR IF SBITS LT 1
         DC    0AL4(X'7FFFFFFF'-2+(&SBITS)) ERROR IF SBITS GT 2
&SBITSC  SETC  '&SBITS'
         AGO   .OUTPUTS
.SBITERR ANOP
         MNOTE 4,'''SBITS'' INVALID OR OMITTED - 2 BITS USED'
```

```
&SBITSC  SETC  '2'
         AGO   .OUTPUTS
.OUTPUT  ANOP
&NAME    DC    AL2(&ICC)
         MEXIT
.OUTPUTS ANOP
&NAME    DC    AL2(&ICC+((&DBITSC)-7)*32+(2-(&SBITSC))*16)
         MEXIT
.SERR    ANOP
         MNOTE 12,'''SPEED'' INVALID OR NOT SPECIFIED - MACRO EXPANSION*
               TERMINATED'
         MEXIT
.SYNERR  ANOP
         MNOTE 12,'SYNC OR ASYNC MUST BE SPECIFIED    - MACRO EXPANSION*
               TERMINATED'
         MEND
```

GOCC Macro

The GOCC macro may be used to define the two output configuration control (OCC) bytes. It is addressed by the DATADDR= operand of a GIOCB macro with an operation code of SETOCC.

The format of the macro call is as follows:

```
[symbol]  GOCC  TRANSM=type
                ,SPEED=n
               [,PARITY={ODD|EVEN}]
               [,IGNORE={YES|NO}]
               [,TRANSP={YES|NO}]
               [,DBITS=d]
               [,SBITS=s]
```

GOCC Parameters

TRANSM=type (mandatory, keyword parameter)

The TRANSM= parameter defines whether binary synchronous or asynchronous transmission is to be used; 'type' must be either SYNC or ASYNC.

SPEED=n (mandatory, keyword parameter)

The SPEED= parameter defines the transmission speed. 'n' is the speed in bps, and must be one of the following:

150, 300, 600, 1200, 2400, 4800, 9600, or 19200

Note: *Speed selection is limited as follows:*

ASYNC: 150 - 9600 bps
SYNC: 600 - 19200 bps

PARITY= (optional, keyword parameter)

The PARITY= parameter specifies whether ODD or EVEN parity is to be generated. ODD parity will be generated if this parameter is omitted.

IGNORE= (optional, keyword parameter)

The IGNORE= parameter specifies whether or not the 'data terminal ready' (DTR) signal is to be ignored. If the parameter is omitted, the DTR signal will be used.

TRANSP= (optional, keyword parameter)

The TRANSP= parameter specifies whether or not data is to be sent in transparent mode. If the parameter is omitted, data will be sent in nontransparent mode.

DBITS=d (optional, keyword parameter)

The DBITS= parameter is only valid if TRANSM=ASYNC. DBITS specifies whether the data byte is to be 7-bits or 8-bits. Any absolute expression is valid if the value is 7 or 8.

SBITS=s (optional, keyword parameter)

The SBITS= parameter is only valid if TRANSM=ASYNC. One or two stop bits may be specified. Any absolute expression is valid if the value is 1 or 2.

GOCC Code List

Note: * *in column 72 indicates a continuation character.*

```
         MACRO
&NAME    GOCC  &TRANSM=,&PARITY=ODD,&IGNORE=NO,&TRANSP=NO,&SPEED=,      *
               &DBITS=,&SBITS=
         GBLA  &IHBBLIM,&IHBBLC              BUFFER LOCATION AND LIMIT
         LCLA  &OCC
         LCLC  &DBITSC,&SBITSC               CHARACTER REP. OF D/S BITS
         IHBGAM3                             ENSURE EVEN BOUNDARY
&IHBBLC  SETA  &IHBBLC+2                     UPDATE BUFFER-LOC COUNTER
         AIF   (&IHBBLIM GE &IHBBLC).OK      BUFFER OVERFLOW ?
         IHBERMAC 182                          YES - ERROR MSG
.OK      ANOP
&OCC     SETA  0                             INITIALIZE CODE
         AIF   ('&IGNORE' EQ 'NO').SPEED     USE DTR SIGNAL ?
         AIF   ('&IGNORE' EQ 'YES').IGNORE   IGNORE DTR SIGNAL ?
         MNOTE 4,'IGNORE PARAMETER INVALID - SET TO NO'
         AGO   .SPEED
.IGNORE  ANOP
&OCC     SETA  &OCC+16384                    FLAG IGNORE DTR
.SPEED   ANOP
         AIF   (T'&SPEED EQ 'O').SERR        TRANS. SPEED SPECIFIED ?
         AIF   (&SPEED EQ 150).PRTY            150 BPS ?
         AIF   (&SPEED EQ 300).S1              300 BPS ?
         AIF   (&SPEED EQ 600).S2              600 BPS ?
         AIF   (&SPEED EQ 1200).S3            1200 BPS ?
         AIF   (&SPEED EQ 2400).S4            2400 BPS ?
         AIF   (&SPEED EQ 4800).S5            4800 BPS ?
         AIF   (&SPEED EQ 9600).S6            9600 BPS ?
         AIF   (&SPEED EQ 19200).S7          19200 BPS ?
         AGO   .SERR                         INVALID IF NONE OF THESE
.S1      ANOP
&OCC     SETA  &OCC+2048                     FLAG   300 BPS
         AGO   .PRTY                           THEN GO CHECK PARITY
.S2      ANOP
&OCC     SETA  &OCC+4096                     FLAG   600 BPS
         AGO   .PRTY                           THEN GO CHECK PARITY
.S3      ANOP
&OCC     SETA  &OCC+6144                     FLAG  1200 BPS
         AGO   .PRTY                           THEN GO CHECK PARITY
.S4      ANOP
&OCC     SETA  &OCC+8192                     FLAG  2400 BPS
         AGO   .PRTY                           THEN GO CHECK PARITY
.S5      ANOP
&OCC     SETA  &OCC+10240                    FLAG  4800 BPS
         AGO   .PRTY                           THEN GO CHECK PARITY
.S6      ANOP
&OCC     SETA  &OCC+12288                    FLAG  9600 BPS
         AGO   .PRTY                           THEN GO CHECK PARITY
.S7      ANOP
&OCC     SETA  &OCC+14336                    FLAG 19200 BPS
```

```
.PRTY    ANOP
         AIF   ('&PARITY' EQ 'ODD').TRANS   ODD PARITY ?
         AIF   ('&PARITY' EQ 'EVEN').EVEN   EVEN PARITY ?
         MNOTE 4,'PARITY PARAMETER INVALID - SET TO ODD'
         AGO   .TRANS
.EVEN    ANOP
&OCC     SETA  &OCC+256                     FLAG EVEN PARITY
.TRANS   ANOP
         AIF   ('&TRANSP' EQ 'NO').SYN      NON-TRANSPARENT ?
         AIF   ('&TRANSP' EQ 'YES').TRANSP  TRANSPARENT ?
         MNOTE 4,'TRANSP PARAMETER INVALID - SET TO NO'
         AGO   .SYN
.TRANSP  ANOP
&OCC     SETA  &OCC+128                     FLAG TRANSPARENCY
.SYN     ANOP
         AIF   (T'&TRANSM  EQ 'O').SYNERR   TRANSMISSION SPECIFIED ?
         AIF   ('&TRANSM' EQ 'ASYNC').ASYNC ASYNCHRONOUS ?
         AIF   ('&TRANSM' EQ 'SYNC').SYNC   SYNCHRONOUS ?
         AGO   .SYNERR                      ERROR IF NEITHER
.SYNC    ANOP
&OCC     SETA  &OCC+64                      FLAG SYNCHRONOUS
         AIF   (&SPEED LT 600).SERR         SPEED OK FOR SYNC ?
         AIF   (T'&DBITS  NE 'O').BITSIG    DBITS SPECIFIED ?
         AIF   (T'&SBITS  NE 'O').BITSIG    SBITS SPECIFIED ?
         AGO   .OUTPUT                      GO OUTPUT DATA
.BITSIG  ANOP
         MNOTE 4,'''DBITS'' AND/OR ''SBITS'' CODED - IGNORED FOR SYNC'
         AGO   .OUTPUT
.ASYNC   ANOP
         AIF   (&SPEED GT 9600).SERR        SPEED OK FOR ASYNC ?
         AIF   (T'&DBITS  EQ 'O').DBITERR   DBITS SPECIFIED ?
         DC    0AL4(X'80000000'+(&DBITS)-7) ERROR IF DBITS LT 7
         DC    0AL4(X'7FFFFFFF'-8+(&DBITS)) ERROR IF DBITS GT 8
&DBITSC  SETC  '&DBITS'
         AGO   .STOPB
.DBITERR ANOP
         MNOTE 4,'''DBITS'' INVALID OR OMITTED - 7 BITS USED'
&DBITSC  SETC  '7'
.STOPB   ANOP
         AIF   (T'&SBITS  EQ 'O').SBITERR   SBITS SPECIFIED ?
         DC    0AL4(X'80000000'+(&SBITS)-1) ERROR IF SBITS LT 1
         DC    0AL4(X'7FFFFFFF'-2+(&SBITS)) ERROR IF SBITS GT 2
&SBITSC  SETC  '&SBITS'
         AGO   .OUTPUTS
.SBITERR ANOP
         MNOTE 4,'''SBITS'' INVALID OR OMITTED - 2 BITS USED'
&SBITSC  SETC  '2'
         AGO   .OUTPUTS
.OUTPUT  ANOP
&NAME    DC    AL2(&OCC)
         MEXIT
.OUTPUTS ANOP
&NAME    DC    AL2(&OCC+((&DBITSC)-7)*32+(2-(&SBITSC))*16)
         MEXIT
```

```
.SERR    ANOP
         MNOTE 12,'''SPEED'' INVALID OR NOT SPECIFIED - MACRO EXPANSION*
               TERMINATED'
         MEXIT
.SYNERR  ANOP
         MNOTE 12,'TRANSM OPERAND MUST BE SPECIFIED   - MACRO EXPANSION*
               TERMINATED'
         MEND
```

GTM Macro

The GTM macro enables two bytes of buffer storage to be tested under the control of a mask.

The format of the macro call is:

```
[symbol] GTM    address[,label] ,MASK=n
```

GTM Parameters

address[,label] (mandatory, positional parameter)

The address parameter represents the address of the buffer location at which the bits are to be tested. Addresses are as specified by 3250 macros that use symbolic buffer addresses.

MASK=n (mandatory, keyword parameter)

The MASK= parameter defines the 16-bit mask to be used in the operation. 'n' can be any valid absolute expression. The bits of the mask are made to correspond one for one with the bits of the word in buffer storage at the address given. A mask bit of one indicates that the storage bit is to be tested. When the mask bit is zero, the storage bit is ignored.

The state of the bits selected by the mask at the given buffer address is used, when the operation ends, to set the condition code as follows:

00 if selected bits were all 0's, (or mask all 0's)
01 if selected bits were mixed 0's and 1's
10 (not set by this order)
11 if selected bits were all 1's

GTM Code List

```
         MACRO
&NAME    GTM    &ADDR,&GROUP,&MASK=                TEST UNDER MASK
         GBLA   &IHBBLC                            BUFFER LOCATION
         AIF    (T'&MASK EQ 'O').ERR               MASK OMITTED ?
&IHBBLC  SETA   &IHBBLC+2                          ADD 2 FOR MASK
&NAME    IHBGAM1 10989,&ADDR,&GROUP                ORDER CODE + ADDRESS
         DC     AL2(&MASK)                         MASK
         MEXIT
.ERR     ANOP
         MNOTE 12,'MASK PARAMETER MISSING - MACRO EXPANSION TERMINATED'
         MEND
```

GTRCT Macro

The GTRCT macro enables transfers to be taken to the given buffer location under the control of a count.

The format of the macro call is:

```
[symbol]  GTRCT  address[,label] ,COUNT=n
```

GTRCT Parameters

address[,label] (mandatory, positional parameter)

The address parameter represents the address of the buffer location to which control passes unless the count field, after being decremented, is 0. Addresses are as specified by 3250 macros that use symbolic buffer addresses.

COUNT=n (mandatory, keyword parameter)

The COUNT= parameter specifies the initial value to be assigned to the count field. 'n' can be any valid absolute expression.

Each time the order is executed, the count field is decremented. If its new value is 0, control passes to the next sequential instruction, otherwise control passes to the buffer address specified.

Note: *If the count field is 0 before being decremented the new value will be hex 'FFFF' and the order will continue to execute.*

GTRCT Code List

```
         MACRO
&NAME    GTRCT &ADDR,&GROUP,&COUNT=              TRANSFER ON COUNT
         GBLA  &IHBBLC                           BUFFER LOCATION
         AIF   (T'&COUNT EQ 'O').ERR             COUNT SPECIFIED ?
&IHBBLC  SETA  &IHBBLC+2                         ADD 2 FOR COUNT FIELD
&NAME    IHBGAM1 10992,&ADDR,&GROUP              ORDER CODE + ADDRESS
         DC    AL2(&COUNT)                       COUNT FIELD
         MEXIT
.ERR     ANOP
         MNOTE 12,'COUNT OPERAND MISSING - MACRO EXPANSION TERMINATED'
         MEND
```

GTOC Macro

The GTOC macro is described in Appendix A.

Appendix D. Example Plotter Program

This appendix contains the following example program showing the use of the macros that support the Plotter Attachment custom feature described in Appendix C. It is not a typical application program.

After the program there are some comments on its operation.

Plotter Program

The simple program shown below tests the RS-232-C port interface.

The plotter program uses the GIO, GTOC, GTM, and GTRCT orders that have been introduced to support the use of the Plotter Attachment custom feature. The three data-generation macros GIOCB, GOCC, and GICC are also used to generate IOCB, ICC, and OCC data blocks.

The data is in the required format for the plotter. The GIO order is used first to set up the conditions for input/output through the port, and then to establish communication with the plotter, before sending the data in 80-byte blocks.

Program List

```
PSTRT   GTOC  SETTC              (unconditional)
*************************************************
*                  FLAGS                        *
*************************************************
FLAG1   DC    H'0'               'ACK'
FLAG2   DC    H'0'               'WACK'
FLAG3   DC    H'0'               'NACK'
FLAG4   DC    H'0'               BAD CODE
FLAG5   DC    H'0'               SET TC ERROR
FLAG6   DC    H'0'               SET OCC ERROR
FLAG7   DC    H'0'               SET ICC ERROR
FLAG8   DC    H'0'               WRITE ERROR
FLAG9   DC    H'0'               SENSE AFTER READ ERROR
FLAGA   DC    H'0'               BYTE ERROR
FLAGB   DC    H'0'               'ENQ' ERROR AFTER 'WACK'
FLAGC   DC    H'0'               3 'NACK's RECEIVED
FLAGD   DC    H'0'               SENSE ERROR
FLAGE   DC    H'0'               NON-ZERO AFTER READ
FLAGF   DC    H'0'               WRONG LENGTH INDICATED AFTER READ
FLAGG   DC    H'0'               *** RESERVED ***
*************************************************
*                FOOTPRINTS                     *
*************************************************
FPTC    DC    H'0'
FPICC   DC    H'0'
FPOCCN1 DC    H'0'
FPENQ   DC    H'0'
FPREAD1 DC    H'0'
FPOCCT  DC    H'0'
FPDC2   DC    H'0'
FPREAD2 DC    H'0'
FPDATA  DC    H'0'
```

```
FPREAD3 DC    H'0'
FPOCCN2 DC    H'0'
FPEOT   DC    H'0'
*************************************************
*         TARGET AREA FOR READ OPERATION        *
*************************************************
RETA    GNOP2
*************************************************
*         TARGET AREA FOR SENSE OPERATION       *
*************************************************
SENSA   GNOP2
        GNOP2
        GNOP2
        GNOP2
*************************************************
* SET THE TERMINATE CHARACTER TO 'NACK'         *
*  AND CHECK THE CONDITION CODE                 *
*************************************************
SETTC   GMVD  FPTC,BDATA=FFFF       FOOTPRINT FOR SET TC
        GIO   IOCBTC
        GTOC  STERR,COND=1
*************************************************
* SET THE OUTPUT CONFIGURATION REGISTER (NOTRANS) *
*  AND CHECK THE CONDITION CODE                 *
*************************************************
SETOCCN GMVD  FPOCCN1,BDATA=FFFF    FOOTPRINT FOR SET OCC(N)
        GIO   IOCBOCN
        GTOC  SOERR,COND=1
        GTOC  SETICC            (unconditional)
*************************************************
* SET THE INPUT CONFIGURATION CONTROL REGISTER   *
*  AND CHECK THE CONDITION CODE                 *
*************************************************
SETICC  GMVD  FPICC,BDATA=FFFF      FOOTPRINT FOR SET ICC
        GIO   IOCBICC
        GTOC  SIERR,COND=1
        GTOC  WRITE0            (unconditional)
*************************************************
* SET THE OUTPUT CONFIGURATION REGISTER (TRANSP)  *
*  AND CHECK THE CONDITION CODE                 *
*************************************************
SETOCCT GMVD  FPOCCT,BDATA=FFFF     FOOTPRINT FOR SET OCC(T)
        GIO   IOCBOCT
        GTOC  SOERR,COND=1
        GTOC  WRITE1            (unconditional)
*************************************************
* WRITE ENQ TO ATTACHED PLOTTER                 *
*  AND CHECK THE CONDITION CODE                 *
*************************************************
WRITE0  GMVD  FPENQ,BDATA=FFFF      FOOTPRINT FOR 'ENQ'
        GIO   IOCBENQ
        GTOC  WERR,COND=1
        GTOC  READ              (unconditional)
```

```
*************************************************
* WRITE STX DC2 ETB TO ATTACHED PLOTTER         *
*  AND CHECK THE CONDITION CODE                 *
*************************************************
WRITE1  GMVD  FPDC2,BDATA=FFFF      FOOTPRINT FOR 'DC2'
        GIO   IOCBDC2
        GTOC  WERR,COND=1
        GTOC  READ              (unconditional)
*************************************************
* WRITE BLOCK OF DATA TO ATTACHED PLOTTER       *
*  AND CHECK THE CONDITION CODE                 *
*************************************************
WRITE2  GMVD  FPDATA,BDATA=FFFF     FOOTPRINT FOR WRITE DATA
        GIO   IOCBDAT
        GTOC  WERR,COND=1
        GTOC  READ              (unconditional)
*************************************************
* WRITE EOT TO ATTACHED PLOTTER                 *
*  AND CHECK THE CONDITION CODE                 *
*************************************************
WRITE3  GMVD  FPEOT,BDATA=FFFF      FOOTPRINT FOR 'EOT'
        GIO   IOCBEOT
        GTOC  WERR,COND=1
        GTOC  ACK               (unconditional)
*************************************************
* NOW READ THE RESPONSE FROM THE ATTACHED PLOTTER *
*  AND CHECK THE CONDITION CODE                 *
*************************************************
READ    GMVD  FPREAD1,BDATA=FFFF    FOOTPRINT FOR READ OPERATION
        GIO   IOCBRED
        GTOC  NOTNACK,COND=0          CANT BE NACK IF COND. CODE ZERO
*************************************************
* NON-ZERO CONDITION CODE AFTER READ OPERATION.  *
*  THIS SHOULD BE BECAUSE RESPONSE WAS A 'NACK'  *
*  WHICH IS ALSO THE TERMINATE CHARACTER AND A   *
*  WRONG-LENGTH INDICATION RESULTED - CHECK THIS *
*************************************************
        GMVD  FLAGE,BDATA=0001        NON-ZERO CONDITION AFTER READ
        GIO   IOCBSEN
        GTOC  SSERR,COND=1
        GTM   SENSA,MASK=X'0400'      IF WRONG LENGTH - ALL ONES
        GTOC  NAC0,COND=3
        GTOC  BYTERR            (unconditional)
NAC0    GTM   SENSA,MASK=X'FB00'      IF WRONG LENGTH - ALL ZEROS
        GTOC  NAC1,COND=0
        GTOC  BYTERR            (unconditional)
*************************************************
* NOW TEST UNDER MASK TO CONFIRM THAT IT WAS NACK *
*  FIRST CHECKING THE 'ON' BITS, THEN THE 'OFF'  *
*************************************************
NAC1    GMVD  FLAGF,BDATA=0001        WRONG - LENGTH INDICATED
        GTM   RETA,MASK=X'3D00'       IF 'NACK', SHOULD BE ALL ONES
        GTOC  NAC2,COND=3
        GTOC  BADCDE                  NOT ACK0,ACK1,NACK, OR WACK
```

```
NAC2     GTM   RETA,MASK=X'C200'        IF 'NACK', SHOULD BE ALL ZEROS
         GTOC  NACK,COND=0
         GTOC  BADCDE                   NOT ACK0,ACK1,NACK, OR WACK
NOTNACK  GNOP2
*****************************************************
* 2-BYTES SUCCESSFULLY RECEIVED FROM PLOTTER. NOW *
*  DETERMINE WHETHER ACK0, ACK1 OR WACK. IN EACH  *
*  CASE FIRST CHECKING THE 'ON' BITS THEN THE     *
*  'OFF' BITS                                     *
*****************************************************
         GTM   RETA,MASK=X'1070'        IF ACK0 - ALL ONES
         GTOC  OK,COND=3
         GTOC  NOTA0                    CAN'T BE ACK0
OK       GTM   RETA,MASK=X'EF8F'        IF ACK0 - ALL ZEROS
         GTOC  ACK,COND=0
         GTOC  BADCDE                   NOT ACK0,ACK1,NACK, OR WACK
NOTA0    GTM   RETA,MASK=X'1061'        IF ACK1 - ALL ONES
         GTOC  OK2,COND=3
         GTOC  NOTA1                    CAN'T BE ACK1
OK2      GTM   RETA,MASK=X'EF9E'        IF ACK1 - ALL ZEROS
         GTOC  ACK,COND=0
         GTOC  BADCDE                   NOT ACK0,ACK1,NACK, OR WACK
NOTA1    GTM   RETA,MASK=X'106B'        IF WACK - ALL ONES
         GTOC  OK3,COND=3
         GTOC  BADCDE                   NOT ACK0,ACK1,NACK, OR WACK
OK3      GTM   RETA,MASK=X'EF94'        IF WACK - ALL ZEROS
         GTOC  WACK,COND=0
         GTOC  BADCDE                   NOT ACK0,ACK1,NACK, OR WACK
ACK      GNOP2
*****************************************************
* RESPONSE WAS AN 'ACK0' OR AN 'ACK1' SIGNAL HOST *
*  TO PLACE NEXT BLOCK OF DATA IN BUFFER, RESET   *
*  THE COUNT FIELD OF THE GTRCT FOR 'NACK', AND   *
*  RESTART THE BUFFER PROGRAM                     *
*****************************************************
         GMVD  FLAG1,BDATA=0001
         GEOS
WACK     GNOP2
*****************************************************
* RESPONSE WAS A 'WACK' - SEND ENQ SEQUENCE       *
*  AND CHECK THE CONDITION CODE                   *
*****************************************************
         GIO   IOCBENQ
         GTOC  ENQERR,COND=1
         GTOC  READ                (unconditional)
         GMVD  FLAG2,BDATA=0001
         GEOS
NACK     GNOP2
*****************************************************
* RESPONSE WAS A 'NACK' - TRY AGAIN THREE TIMES   *
*****************************************************
         GTRCT WRITE2,COUNT=4
         GTOC  NACK3               (unconditional)
         GMVD  FLAG3,BDATA=0001
         GEOS
```

```
BADCDE  GNOP2
        GMVD FLAG4,BDATA=0001
        GEOS
STERR   GMVD  FLAG5,BDATA=0001
        GTOC  ERREX              (unconditional)
SOERR   GMVD  FLAG6,BDATA=0001
        GTOC  ERREX              (unconditional)
SIERR   GMVD  FLAG7,BDATA=0001
        GTOC  ERREX              (unconditional)
WERR    GMVD  FLAG8,BDATA=0001
        GTOC  ERREX              (unconditional)
SSERR   GMVD  FLAG9,BDATA=0001
        GTOC  ERREX              (unconditional)
BYTERR  GNOP2
        GMVD  FLAGA,BDATA=0001
        GEOS
ENQERR  GNOP2
        GMVD  FLAGB,BDATA=0001
        GEOS
NACK3   GNOP2
        GMVD  FLAGC,BDATA=0001
        GEOS
ERREX   GIO   IOCBSEN
        GTOC  SENERR,COND=1
        GEOS
SENERR  GMVD  FLAGD,BDATA=0001
        GEOS
IOCBTC  GIOCB OPER=SETTC,DATA=X'3D'
IOCBOCT GIOCB OPER=SETOCC,DATADDR=OCC
IOCBOCN GIOCB OPER=SETOCC,DATADDR=OCC2
IOCBICC GIOCB OPER=SETICC,DATADDR=ICC
IOCBDC2 GIOCB OPER=WRITE,DATADDR=DC2,LENGTH=3
IOCBDAT GIOCB OPER=WRITE,DATADDR=STX,LENGTH=82
IOCBRED GIOCB OPER=READ,DATADDR=RETA,LENGTH=2
IOCBSEN GIOCB OPER=SENSE,DATADDR=SENSA,LENGTH=8
IOCBEOT GIOCB OPER=WRITE,DATADDR=EOT,LENGTH=1
IOCBENQ GIOCB OPER=WRITE,DATADDR=ENQ,LENGTH=1
OCC     GOCC TRANSM=SYNC,IGNORE=YES,TRANSP=YES,SPEED=19200
OCC2    GOCC TRANSM=SYNC,IGNORE=YES,SPEED=19200
ICC     GICC TRANSM=SYNC,SPEED=19200
STX     DC    BL1'00000010' THAT'S AN "STX"
DATA    DC    CL80' '       DATA RECORD FILLER
ETB     DC    BL1'00100110' THAT'S AN "ETB"
EOT     DC    BL1'00110111' THAT'S AN "EOT"
ENQ     DC    BL1'00101101' THAT'S AN "ENQ"
DC2     DC    BL1'00000010' THAT'S AN "STX"
        DC    BL1'00010010' THAT'S AN "DC2"
        DC    BL1'00100110' THAT'S AN "ETB"
```

The following modifications are suggested to improve performance:

- The assembler program should have error recovery routines, rather than cause an ABEND, when an 'ACK' is not received.
- The assembler program should transmit the data to the buffer in (for example) 8K- or 16K-byte blocks, rather than the 80-byte blocks used in this program.
- The buffer program should transmit the data to the plotter in (for example) 480-byte blocks, rather than the 80-byte blocks used in this program.
- The IOCBs should be chained together when appropriate. For example, a Write operation is always followed by a Read operation in order to check the response from the plotter; in this instance, the ICC and OCC bytes and the termination character should be set using one GIO order that refers to the head of a chain of IOCBs.

In the example program, the assembler program performs the following functions:

- On receipt of the first EOS attention, the assembler program:
  - Reads the buffer program into main storage and checks the flags
  - *If* an 'ACK' is received, *then* it:
    - Resets all flag fields to zero
    - Sets the GTOC address at PSTRT (start of buffer program) to SETOCCT
    - Sets the footprint target at READ to FPREAD2
    - Rewrites the buffer program to buffer storage and starts the program.
  - *Else* ABEND.
- On receipt of the second EOS attention, the assembler program:
  - Reads the buffer program into main storage and checks the flags
  - *If* an 'ACK' is received, *then* it:
    - Resets all flag fields to zero
    - Sets the GTOC address at PSTRT (start of buffer program) to WRITE2
    - Sets the footprint target at READ to FPREAD3
    - Reads an 80-byte block from the data file into the data area in the buffer program
    - Rewrites the buffer program to buffer storage and starts the program.
  - *Else* ABEND.

- On receipt of subsequent EOS attentions, the assembler program:
  - Reads the buffer program into main storage and checks the flags
  - *If* an 'ACK' was received, *then* it:
    - Resets all flag fields to zero
    - Reads an 80-byte block from the data file into the data area in the buffer program
    - Rewrites the buffer program to buffer storage and starts the program
  - *Else* ABEND.
- On receipt of an EOS attention following an end-of-file on data, the assembler program:
  - Reads the buffer program into main storage and checks the flags
  - *If* an 'ACK' was received, *then* it:
    - Resets all flag fields to zero
    - Sets the GTOC address at PSTRT (start of buffer program) to SETOCCN
    - Sets the footprint target at SETOCCN to FPOCCN2
    - Sets the GTOC address after SETOCCN to WRITE3
    - Rewrites the buffer program to buffer storage and starts the program.
  - *Else* ABEND.
- On receipt of next EOS attention, the assembler program:
  - Terminates the program.

Appendix E. Summary of Buffer Orders

Figures 17, 18, and 19 summarize the buffer orders that support the Cursor Control Tablet Attachment custom feature and the Plotter Attachment custom feature.

Mnemonic	Name of Tablet Buffer Orders	Hex Code
GSPOS	Store X,Y Position	2AE8
GSYMB	Draw Symbol	2A88
GTOC	Transfer on Condition	2A70-F

Figure 17. Cursor Control Tablet Buffer Orders

Mnemonic	Name of Plotter Buffer Orders	Hex Code
GIO	Input/Output	2A89
GTOC	Transfer on Condition	2A70-F
GTM	Transfer under Mask	2AED
GTRCT	Transfer on Count	2AF0

Figure 18. Plotter Buffer Orders

Mnemonic	Name of 3250 Buffer Orders Used to Support Plotters	Hex Code
GNOP2	No-Operation (2-Byte)	2A80
GEOS	End Order Sequence	2A81
GNOP2	No-Operation (2-Byte)	2AB0-F
GNOP4	No-Operation (4-Byte)	2AC0
GMVA	Move Immediate Address	2AEB
GMVD	Move Immediate Data	2AEC

Figure 19. Plotter Buffer Orders

Glossary

This glossary defines terms and abbreviations, as applicable to the 3250 Graphics Display System : Attachments for Cursor Control Tablet and for Plotter. 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 by the Computer and Business Equipment Manufacturers Association, copies of which may be purchased from the American National Standards Institute, 1430 Broadway, New York, NY10018. These definitions are identified by asterisks (*).

ACK. The acknowledge character.

ACK0. A transmission control character for even positive acknowledgment

ACK1. A transmission control character for odd positive acknowledgment

acknowledge character. (ISO) A transmission control character transmitted by a receiver as an affirmative response to a sender.

asynchronous transmission. (TC97) Transmission in which the time of occurrence of the start of each character or block of characters is arbitrary; once started, the time of occurrence of each signal representing a bit within the character, or block, has the same relationship to significant instants of fixed time frame.

Buffer program. A set of buffer orders in sequence

Buffer order. A number of coded bytes, contained in the buffer program, that specify an operation or mode of operation to the 3255 Display Control.

binary synchronous transmission. Data transmission in which synchronization of characters is controlled by timing signals generated at the sending and receiving stations.

CRC. Cyclic Redundancy Check

cyclic redundancy check. (TC97) A redundancy check in which the check key is generated by a cyclic algorithm

continuous refresh. A custom feature that allows the host program to communicate with the 3255 Display Control without interrupting the regeneration of the 3251 picture.

Data Link Escape Character. (ISO) A transmission control character that changes the meaning of a limited number of contiguously following characters or coded representations and that is used exclusively to provide supplementary transmission control characters.

display element. (TC97) In computer graphics, a basic graphic element that can be used to construct a display image; for example, a dot, a line, a segment, a character. Synonymous with graphic primitive.

DLE. Data Link Escape Character

ETB. (ANSI) The end-of-transmission-block character.

ITB. Intermediate Text Block

K. (ANSI) When referring to storage capacity, two to the tenth power, (1024) in decimal notation.

keyword parameter. A parameter that consists of a keyword, followed by one or more values. See also positional parameter.

light-pen interrupt. An I/O interrupt generated at the host as a result of a light-pen detect.

macro definition. (ISO) A set of statements that defines the name of, format of, and conditions for generating a sequence of assembler language statements from a single source statement.

NACK. A negative acknowledge character

positional parameter. A parameter that must appear in a specified location, relative to other parameters.

plotter. (ISO) An output unit that presents data in the form of a two-dimensional graphic representation.

tablet. (TC97) In computer graphics, a locator device with a flat surface and a mechanism that converts indicated positions on the surface into coordinate data.

TTL. Transistor Transistor Logic

vector. (ISO) In computer graphics, a directed line segment

WACK. (Wait before transmit positive acknowledgment). A character sequence sent by a receiving station to indicate that it is temporarily not ready to receive.

Index

IBM 3250 Graphics Display System:
RPQs 7J0070, 7J0071, 7J0072, 7J0073
Custom Feature Description

READER'S COMMENT FORM

Order No. GA33-3120-0

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, 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

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

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.)

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 with RPQs 7J0070, 7J0071, 7J0072, and 7J0073: Custom Feature Description Printed in U.S.A. GA33-3120-0

IBM 3250 with RPQs 7J0070, 7J0071, 7J0072, and 7J0073: Custom Feature Description Printed in U.S.A. GA33-3120-0

GA33-3120-0