Personal System/2™
Display Adapter 8514/A
Technical Reference

68X2248
S68X-2248-0
April 1987

DISPLAY ADAPTER 8514/A

First Edition (April 1987)

The following paragraph does not apply to the United Kingdom or any country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This publication could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time.

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

Requests for copies of this publication and for technical information about IBM products should be made to your IBM Authorized Dealer or your IBM Marketing Representative.

© Copyright International Business Machines Corporation 1987. All rights reserved.

The following statement applies to this IBM product. The statement for other IBM products intended for use with this product will appear in their accompanying materials.

FEDERAL COMMUNICATIONS COMMISSION (FCC) STATEMENT

Warning: This equipment has been certified to comply with the limits for a Class B computing device, pursuant to Subpart J of Part 15 of FCC rules. Only peripherals (computer input/output devices, terminals, printers, etc.) certified to comply with the Class B limits may be attached to this computer when this computer is operated in a residential environment. Operation with non-certified peripherals is likely to result in interference to radio and TV reception.

Preface

This manual gives technical information about the Display Adapter 8514/A. It is intended for anyone who needs to understand the technical details of how the adapter works when connected to the IBM Personal System/2™.

- **Chapter 1** describes the Display Adapter 8514/A
- **Chapter 2** describes the programming considerations
- **Chapter 3** describes the shippable code items, the font file formats, and the monitor, physical, and electrical interfaces.
- **Appendix A** describes the advanced-function default palette.

An index is provided at the back of the manual.

Related Publications

Further information about the Display Adapter 8514/A is given in:

- *IBM Personal System/2 Display Adapter 8514/A; Installation Instructions*
- *IBM Personal System/2 Display Adapter 8514/A; Hardware Maintenance Reference Supplement*
- *IBM Personal System/2 Display Adapter 8514/A; Hardware Maintenance Service Supplement*

Suggested Related Publications

For Basic Input Output System (BIOS) information used for some operating modes (for example, Extended Graphics Adapter modes), see the *IBM Personal System/2 Display Adapter; Technical Reference* manual.

Personal System/2 is a trademark of IBM Corporation

Contents

Chapter 1. The Display Adapter 8514/A

Figures

Chapter 1. The Display Adapter 8514/A

Introduction

The Display Adapter 8514/A is an optional feature that provides an advanced function display interface for the IBM Personal Computer.

Notes:

1. Throughout the remainder of this manual, the terms "Personal Computer" and "PC" are used to mean any IBM Personal System/2 that has the Display Adapter 8514/A installed, unless specifically stated otherwise.
2. The term "Adapter Interface" is used to mean the programming interface between the Display Adapter 8514/A and the controlling system.

The IBM Color Display 8514 is supported at 1024 x 768 and 640 x 480 PELs. The adapter also drives the IBM Monochrome Display 8503, IBM Color Display 8512, and the IBM Color Display 8513 at PEL densities up to 640 x 480. A schematic drawing of the adapter is shown in Figure 1-1.

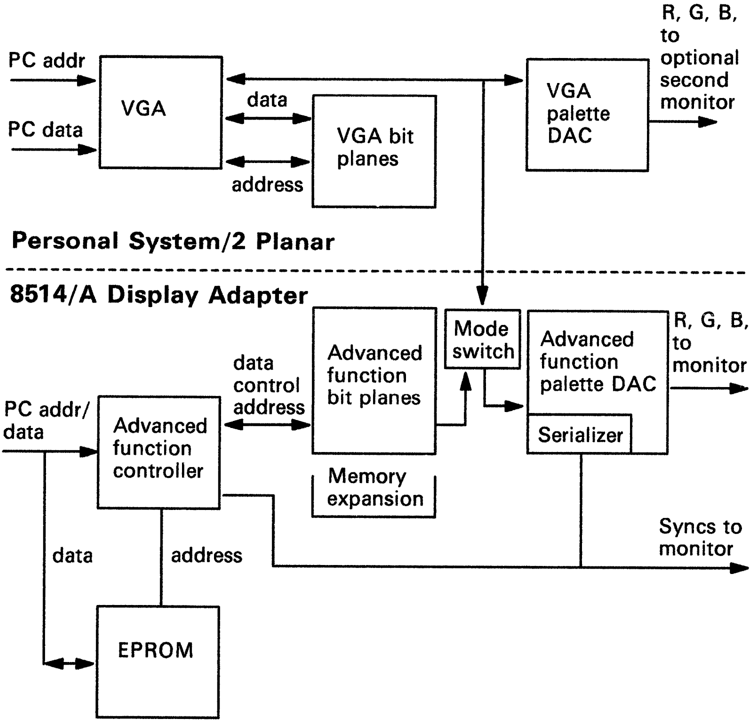


Figure 1-1. Schematic Drawing of the Display Adapter 8514/A

Note: The Video Graphics Array (VGA) resides on the system unit planar along with its own bit planes and Palette/DAC. Either the VGA or the Display Adapter 8514/A can display on the monitor attached to the Display Adapter 8514/A without corrupting the other adapter's bit planes. The Palette/DAC on the Display Adapter 8514/A is loaded with VGA palette contents if the VGA is to use the IBM Color Display 8514 monitor. Alternatively, the VGA can simultaneously display on its own monitor whilst the Display Adapter 8514/A uses the IBM Color Display 8514.

Operating Modes

The adapter has two modes of operation, selected under program control:

1. Video-graphics array (VGA) mode
2. Advanced function mode.

Video-Graphics Array Mode

VGA is the power-on mode of the adapter; all the function of the VGA adapter is available for full program compatibility.

The bit planes of the VGA and the advanced-function adapter can be updated, independently of which bit planes are being displayed. See "Palettes" on page 1-5 for information on palette control.

Advanced Function Mode

The Adapter Interface

The adapter interface is a programming interface for the adapter. Programming through the interface is much easier than programming directly to the hardware, because the adapter interface handles much of the complexity of the hardware-register interface for you, and it provides a call interface which supports commonly required display adapter functions.

The display modes listed in Figure 1-2 are supported by the adapter interface through entry point calls to the list of adapter interface functions described in "Adapter Interface Function Set" on page 2-1.

Horizontal PELs	1024	640
Vertical PELs	768	480
Interlaced	Yes	No
Video rate (MHz)	44.90	25.17
Line rate (KHz)	35.52	31.47
Frame rate (Hz)	43.48	59.94
PEL time (nsec)	22.27	39.72
H blanking (usec)	5.35	6.36
V blanking (usec)	690	1430

Figure 1-2. Display Modes

Color/Gray-Scale Capability

The *color capabilities* of the adapter can:

1. Provide 16 or 256 out of 256K colors, depending on the number of bit planes installed.
2. Provide the multicolor capability by means of a loadable palette.

(This allows applications to specify their own colors, and caters for applications that wish to use the palette to control the visibility - or highlighting - of different types of data in a picture.)

The *gray-scale capabilities* of the adapter can provide 16 or 64 levels of gray scale, depending on the number of bit planes installed. This supports “image” and “anti-aliased character sets”.

Notes:

1. Bit plane memory in the adapter is arranged in planes of 1024 x 1024. The displayed data is taken from the low 1024 x 768, or 640 x 480 bits. The non-displayed bit-plane memory is used by the adapter as auxiliary storage for functions like area-fill working storage, as a cache for programmable character sets and marker drawing. This area of memory can be addressed by PC programming, although it is not a supported feature.
2. When the adapter operates in 640 x 480 mode, there is sufficient memory for eight 1024 x 512 planes without the memory-expansion feature, but these planes are addressable only as two

separate groups of four planes, rather than as a single set of planes 8 bits deep.

3. A "Memory Expansion Kit" is available for the adapter, which increases the amount of bit-plane storage from 1024 x 1024 x 4 to 1024 x 1024 x 8. This increases the number of simultaneous colors available from 16 to 256.
4. The addition of the memory-expansion feature allows the 8 bit planes to be addressed as a full 8 bits deep for both screen modes.

Palettes

The video output from the adapter is via a palette (color look-up table). This allows applications to select the actual colors (or gray levels) that appear on the display from a greater range of colors than that provided by mapping the number of bits per PEL to fixed colors. The palette is a loadable RAM that translates the number of bits per PEL into three analog signals that drive the red, green, and blue guns of the display. The adapter palette provides a 64-level analog output to each gun, and therefore provides a choice of 256K colors or 64 gray levels. The number of colors that can be displayed on the screen at any one time depends upon the number of bit planes enabled for access.

VGA Palette

Default palettes are provided to avoid every application having to load a palette. For VGA modes, these are optionally loaded by BIOS whenever the mode is changed. The actual palette that is loaded depends upon the mode that is selected, and upon the monitor that is being driven. See the *IBM Personal System/2 Display Adapter; Technical Reference* manual for further details.

Advanced-Function Palette

In advanced-function mode, the default palette is optionally loaded by the HOPEN command of the adapter interface. Two default advanced-function palettes are provided, for the color and monochrome monitors respectively. These palettes are based on the 4-plane VGA palettes, but the values are repeated so they support two banks of 4 planes for the 4 x 2 plane advanced-function modes.

See Appendix A, "Default Palettes" for further information.

If a VGA palette is required for an advanced-function application, then:

- Switch to the VGA mode that uses the required palette.
- Execute HOPEN command with no default-palette load selected.
- Continue with adapter-interface commands.

This technique can be used to load the VGA 256-color palette of mode 13 for advanced-function applications.

Note: The default palettes for advanced-function mode are consistent with those for the VGA modes. Some of the advanced-function features (for example, update mixes) for image processing, or for interfacing with image or graphics data streams, require different palettes. Palettes based on the following are suggested:

- 4-bit (16-color) color palette

  Based on CGA 4 bit colors

- 4-plane monochrome palette

  16-level gray scale with linear steps

- 8-plane (256-color) color palette

  Bit assigned as 4 bits green, 2 bits red, 2 bits blue, with linear intensity steps

- 8-plane monochrome palette

  64-level gray scale with linear steps (bit compatible with 4-plane monochrome)

- When the adapter is not in the advanced-function mode, any VGA palette-loading operation loads the same data into the advanced-function palette.

- When the adapter is in advanced-function mode, VGA-palette loads affect only the VGA palette. The advanced-function palette hardware is loaded with the HLDPAL order.

- The advanced-function palette is used by advanced function and VGA modes to drive the monitor attached to the Display Adapter 8514/A; it should be updated to suit the bit planes being displayed.

- The VGA palette-data is not restored to the advanced-function palette when returning from advanced-function mode to VGA mode.

APA Function

The adapter has high-content bit-buffers and support hardware for applications that manipulate and display text, image, and graphics. The following functions are supported by the hardware:

- Bit-block transfer (BITBLT)
- Line drawing
- Area fill
- Patterns
- Color mixing
- Scissoring.

Bit-Block Transfer (BITBLT)

A set of BITBLT functions handle text and image; these allow transfers from PC main storage to bit planes, from bit planes to PC storage, or between the planes. Data can be read or written across (or through) the planes, and a full set of logical operations are available. Programs can initiate BITBLT functions on any rectangular area, leaving the function to complete independently.

Line Drawing and Area Fill

The adapter provides graphic facilities for drawing lines and polylines with programmable-line type and programmable thickness.

Areas can also be defined with solid color, or with a color pattern (the color pattern may be monochrome or multicolored); you can define the bit pattern used to fill the area. Complex areas may be defined in which the area boundary crosses itself. A full set of logical operations is available with these functions so that lines and areas can be overpainted, underpainted, XORed (and so on) and the full color capability of the adapters realized.

Rectangular Scissor

The rectangular scissor function provides clipping of all drawing functions, and allows multiple-screen windows to be displayed.

Text and Alphanumeric Support

Two forms of character support are available:

- **Alphanumeric support** provides fixed-size character cells. These are for basic alphanumeric operations such as listing directories and files. Character attributes are supported (fonts are provided with the adapter).
- **Text support** is provided for applications that use fonts of varying sizes (proportional spacing is supported) on the screen, with "what-you-see-is-what-you-get" (WYSIWYG) viewing of text documents. (Text support uses the APA capability of the adapter.)

Alphanumeric Support

The adapter supports 3270 alphanumeric operations through the adapter interface. Two characters sets with fixed cell sizes are provided for use when in 1024 x 768 mode, with 3270 extended attribute support. (Blinking is not supported.) Cell sizes available are:

- 12 x 20, giving 85 characters per row and 38 rows per screen (when in 1024 x 768 mode)
- 7 x 15, giving 146 characters per row and 51 rows per screen (when in 1024 x 768 mode).

A third character set with a cell size of 8 x 14 is also available for use when in 640 x 480 mode, allowing 80 x 34 characters to be displayed.

When writing characters to the screen using this alphanumeric support, the alphanumeric data replaces any existing data in the cell positions written. If your application requires that the data overwritten by alphanumeric characters should be retained, ensure that you save/restore the contents of the bit planes.

Alternatively, the bit planes can be subdivided into two groups (or layers) for alphanumerics and APA operations. (The default palette is designed to be used with two layers of 4 planes each.)

Fonts

The adapter interface installation diskette contains fonts (see "Font File Format" on page 3-2) defined as "short stroke vectors"; six code pages are provided in three sizes (see "Short Stroke Vector Characters" on page 2-85).

Code-page switching support is part of the operating system function; no specific support is included with the interface (although the default code page is customizable at installation time).

Text Support

Character set definitions are held in PC main memory. Character definitions may be bit arrays or vector strings.

The adapter interface places a maximum constraint of 255 x 255 PELs on the size of characters that may be displayed.

Cursors.

A 3270-like alphanumeric cursor is provided for use with alphanumerics. The cursor is full-width, any height, with an XOR'd foreground color. (Blinking is not supported.)

No special adapter support is provided for graphics cursors. (Graphics cursors can be implemented either as bit arrays and manipulated by BIT_BLT operations, or as vector display lists.)

Query Adapter/Display Configuration

PC programming can determine via the adapter interface:

1. Which display unit is attached (PEL density, and color or monochrome)
2. Whether the Memory Expansion Kit is fitted

3. What is the current mode of the adapter.

Dual Screen

The adapter can support dual-screen applications by attaching a second display to the VGA output on the IBM Personal System/2 planar board.

In a dual-screen system, the display attached to the Display Adapter 8514/A is the primary display.

When the adapter is in VGA mode, both displays show the VGA data. Any VGA palette-load or mode-change loads both palettes with the data for the primary display. Hence, the secondary display (that is, the one attached to the system planar) may show incorrect colors or grayscale.

When the Display Adapter 8514/A is in advanced-function mode, the VGA supports any of its modes. This allows VGA applications to be displayed on the second screen, while advanced-function applications can be displayed on the Display Adapter 8514/A screen.

Chapter 2. Programming Considerations

Introduction

This chapter describes the adapter interface.

The adapter interface provides a set of function that allows access to the hardware capabilities of the display adapter.

The adapter interface consists of a set of entry points that can be called directly by the controlling environment. The interface driver code that incorporates these entry points is a serially reusable resource.

Adapter Interface Function Set

The Adapter Interface Function Set includes the following graphic primitives:

- Markers/text
- Alphanumerics
- Images
- Lines
- Areas.

The set includes the following graphics operations and drawing attributes:

- Color and mix control:
  - Color index
  - Color lookup table
  - Logic and arithmetic mix.
- Programmable primitives:
  - User-defined line types

  - User-defined area patterns
  - User-defined markers
- Windowing assist:
  - Scissor
  - Restorable state
- Layering assist:
  - Bit plane control
  - Color lookup table
- Bit block transfer:
  - Between bit planes and memory
  - Between bit planes and bit planes.

Adapter Functional Model

Bit Plane Model

The adapter interface defines a set of between 1 and 32 bit planes. These planes are numbered from 0 to 31, with plane 0 associated with bit 0 of the color index. The number of planes available depends on the adapter mode and the amount of video RAM installed in the adapter.

Where 1- and 2-byte color indices are used in orders, these are converted to 32-bit color indices by locating the data in the least significant byte or bytes of the 32-bit word.

Each bit plane can be selectively enabled/disabled for update and/or display (video out).

Scissor Rectangle

The Scissor is a feature of the drawing process; it performs clipping by selectively preventing an area of the bit planes outside a rectangle from being updated. The clipping has the following features which differentiate it from clipping performed on drawing orders:

- The scissor does not prevent the wrapping of objects at the edge of the coordinate space.
- When part of a patterned line is redrawn under the scissor, the pattern will be correctly aligned with the original line.
- When part of a thick line is redrawn under the scissor, the part within the scissor rectangle will be drawn even if the thin line joining the two end points does not cross the scissor rectangle.
- The scissor (implemented in the hardware) is much faster than clipping by the transformation of drawing primitives.

The scissor is used to support the clipping of images, character strings, thick lines, and patterned lines.

The scissor rectangle is controlled by the Set Scissor (HSHS) order.

Graphics Functions

Image and Bit Block Transfer

The adapter-interface functions support the following classes of image format:

- Across the planes (1 bit-per-pel pattern)

  The source data is applied as a pattern. The foreground color is written using the current foreground mix wherever there is a binary 1 in the pattern.

  The background color is written using the current background mix wherever there is a binary 0 in the pattern.
- Through the planes (8 bits-per-pel color)

  The source data is applied as a color index. The color indices are written pel by pel using the current foreground mix.

The functions support the following bit block operations:

- Bit block output (write to bit planes)
- Bit block input (read from bit planes)
- Bit block copy (copy from bit plane(s) to bit plane(s))

The bit block transfers are controlled by the following drawing processor attributes:

- Planes enabled for update
- Logic or arithmetic mix function
- Color (across the planes only)
- Scissor.

Area Fill

Bounded Areas

Bounded areas are defined by a list of orders beginning with a Begin Area (HBAR) and ending with an End Area (HEAR). Between the Begin Area and End Area orders is a list of line drawing orders, which define the boundaries of one or more closed areas.

Rectangular areas can be filled using the HRECT order which typically will give a faster fill.

Programmable Line Types

The adapter interface supports programmable line types and line widths (see the HSLT and HSLW orders).

The line types are defined by a list of counts of PELs on and PELs off. The following orders can be used in a list, to form a continuous line in which the start of the line pattern generated by an order matches the end of the line pattern generated by the the previous order.

- HCLINE - line at current position
- HCRLINE - relative line at current position
- HSCP - set current position.

All other line orders reset the line pattern.

Interface Mechanisms

Using the Call Interface

A set of entry points executable in the controlling environment provides a call interface to the adapter.

Issuing Orders

Orders are issued by calling "order entry points"; there is a separate entry point for each order. Before calling an entry point, the address of the parameter block must be put on the stack. The stack is always cleared by the called entry point before control is returned to the caller.

The Call Interface Specification

The Call Interface consists of a set of procedures for processing orders.

Entry Point Calling Procedure

Function

Each order or pseudo-order has a procedure entry point. Although not all the orders have parameters, they all have parameter blocks.

Calling Sequence

```
   CALL entry-point ( INOUT a: ref_parms );
```

Note: 'Entry-point' stands for the name of an order entry point.

The input to CALL is a pointer to an order parameter block, somewhere in the controlling environment.

PC Environment Linkage Mechanism

PC-DOS

In a PC-DOS environment, the code is in EXE format. The data loaded from the file contains a table with the 32-bit addresses of each of the entry points. Figure 2-1 on page 2-9 and Figure 2-2 on page 2-10 give the order of entry points.

General: Most orders are designed to be used *after* the adapter has been initialized by an HOPEN and *before* issuing an HCLOSE. However, certain orders are meaningful outside the HOPEN/HCLOSE providing the code has been installed:

1. HQMODE - Only some of the returned data is valid before HOPEN. (That is, code level, adapter type, number of bit planes, mono/color and intensity levels.) All other fields are invalid before HOPEN.
2. HQDPS - All fields valid.
3. HQDFPAL - All fields valid.
4. HSPAL/HRPAL - Used for saving and restoring palettes.

Modes: The adapter has two memory options:

1. **Maximum memory configuration.** This is with the Memory Expansion Kit installed; it supports the following screen sizes:
   - 1024 x 768 x 8 planes (1 bank)
   - 640 x 480 x 8 planes (1 bank)
2. **Minimum memory configuration.**This is with the base memory; it supports the following screen sizes:
   - 1024 x 768 x 4 planes (1 bank)
   - 640 x 480 x 4 planes (2 banks) - see notes below

Notes:

1. To update the banks, switching is achieved with the graphics bit-plane enable mask in the HSBP command (where 0 ≤ X ≤ F):
   - To update bank 0, set graphics bit-plane enable to “0X”.
   - To update bank 1, set graphics bit-plane enable to “X0”,
2. To display the banks, use the “Plane Display Enable Mask” in the HSBP command. For example:
   - To display bank 0, set the plane display enable mask to “0F”.
   - To display bank 1, set the plane display enable mask to “F0”.
3. It is not possible to update planes in different banks at the same time. To access planes in different banks, the operation must be performed on each bank individually
4. To display more than 16 colors, set the plane display enable mask to values that enable both banks simultaneously. (For example, a value of “FF” enables all planes of both banks and can give up to 256 colors.)
5. The default palette provides identical palettes for both banks if the “Plane Display Enable Mask” is used to select between them. If all the planes are enabled, bank 1 overlays bank 0 (apart from color 0 - black - which is defined as transparent and allows bank 0 to show through).

Color indices have to be adjusted for this configuration by writing with the color index normally for planes 0-3. Then the nibble for planes 4-7 should be duplicated into the other nibble for the 4-7 write. (For example, to get color ′F3′x, ′F3′x is first written to planes 0-3 and then ′FF′x is written to planes 4-7.)

Entry Point	**Order**
HLINE	Line at given position
HCLINE	Line at current position
HRLINE	Relative line at given position
HCRLINE	Relative line at current position
HSCP	Set current position
HBAR	Begin area
HEAR	End area
HSCOL	Set color
HOPEN	Open adapter
HSMX	Set mix
HSBCOL	Set background color
HSLT	Set line type
HSLW	Set line width
HEGS	Erase graphics screen
HSGQ	Set graphics quality
HSCMP	Set Color Comparison register
HINT	Interrupt
HSPATTO	Set pattern reference point
HSPATT	Set Pattern shape
HLDPAL	Load palette
HSHS	Set scissor
HBBW	Bit block write-image data
HCBBW	Bit block write at current position
HBBR	Bit block read-image data
HBBCHN	Bit block chained data
HBBC	Bit block copy
HSCOORD	Set coordinate type
HQCOORD	Query coordinate type
HSMODE	Set mode
HQMODE	Query current mode
HQMODES	Query adapter modes
HQDPS	Query drawing process state size
HRECT	Fill rectangle
HSBP	Set bit plane controls
HCLOSE	Close adapter
HESC	Escape - terminate processing
HXLATE	Assign MP text color index table
HSCS	Set character set
HCHST	Character string at given position
HCCHST	Character string at current position

Figure 2-1. Entry Point Table (1 of 2)

Entry Point	Order
ABLOCKMFI	Write character block (MFI)
ABLOCKCGA	Write character block (CGA)
AERASE	Erase rectangle
ASCROLL	Scroll rectangle
ACURSOR	Set cursor position
ASCUR	Set cursor shape
ASFONT	Set character set
AXLATE	Assign alpha color index
HINIT	Initialize state
HSYNC	Synchronize adapter
HMRK	Marker at given position
HCMRK	Marker at current position
HSMARK	Set marker shape
HSLPC	Save line-pattern count
HRLPC	Restore line-pattern count
HQCP	Query current position
HQDFPAL	Query default palette
HSPAL	Save palette
HRPAL	Restore palette

Figure 2-2. Entry Point Table (2 of 2)

Calling Mechanism

Each entry point is called by:

1. Pushing the 32-bit address of the order parameter block onto the stack
2. Calling the entry point with a CALL FAR.

The call is returned with the RETURN FAR instruction, which removes the parameters used from the stack.

The following registers are preserved across the call:

- BP, SP, DS, SS, CS.

All other registers may be changed.

Note: Some entry points use the PC processor's direction flag, and clear it before returning.

The controlling system has first to access the address of the adapter interface call table as follows:

```
MOV    AH,35H        ;
MOV    AL,7FH        ; Get interrupt vector 7FH ...
INT    21H           ; ... by function call to DOS.
MOV    AX,ES         ; OR vector 7FH segment ...
OR     AX,BX         ; ... with vector 7FH offset.
JZ     NOCODE        ; If NULL then interface code is not installed.

MOV    AX,0105H      ; Get interface link table address ...
INT    7FH           ; ... by function call to the adapter.
JC     NOCODE        ; If CARRY then interface code is not installed.
                     ; CX:DX is the address of the link table.
```

State Data

The state data initialization is divided into two areas:

- Task-dependent state
- Task-independent state.

Note: In a single tasking environment like PC-DOS there is normally only one copy of a task-dependent state, but it is possible to define a number of different states and switch from one to another to support multiple applications.

Task-Dependent State

The task-dependent state is the part of the state that varies between tasks or applications.

When an HOPEN order is processed, the task-independent state is set to the adapter defaults. (The adapter initializes to the maximum screen size possible.) At HINIT time, the task-dependent state is initialized to the following default values:

- Current position = 0,0
- Current drawing attributes
  - General
    - Foreground color = white (with default palette loaded)
    - Background color = black (with default palette loaded)
    - Foreground mix = overpaint
    - Background mix = leave alone
    - Comparison color = NOT INITIALIZED

    - Comparison logic = FALSE
  - Lines
    - Line type = solid
    - User line definition = NOT INITIALIZED
    - Line width = 1
    - Line pattern position = NOT INITIALIZED
    - Saved line pattern position = NOT INITIALIZED
    - In area flag reset
  - Areas
    - Pattern definition = solid
    - Pattern origin = 0,0
  - Text
    - Text control block pointer = NOT INITIALIZED
    - Marker shape = NOT INITIALIZED
- Current drawing controls
  - Scissor rectangle = full screen
  - Graphics quality flags set to high precision, last PEL null
  - Plane enable mask set to all planes for update
  - Color index translate table = 8 linear values (0 to 7)
- Alphanumeric state data
  - Cursor position = top left of screen
  - Cursor definition = invisible
  - Plane enable mask set to all planes
  - Color index translate table = 16 linear values (0 to 15) for foreground, and 16 linear values (0 to 15) for background
  - Character set definition = NOT INITIALIZED.

Task-Independent State

The task-independent state is the part of the state data that is common to all tasks or applications:

- Color palette contents
- Adapter mode
- Task-dependent state buffer pointer
- Working storage.

Note that:

- Plane-display mask set to all planes
- Default color-palette data set:
  - 16 default CGA color set (color monitor attached)
  - 16 optimized gray shade mapping of default color set (monochrome monitor attached).

  The rest of the palette is loaded as follows:

  16 color 1
  16 color 2
  .
  .
  .
  16 color 15

- The adapter mode can be set at HOPEN time or default to mode 0 (for 1024 x 768 resolution display) or mode 1 (for 640 x 480 resolution display)
- Task dependent state buffer pointer is NOT INITIALIZED.

State Control

Initially, the interface driver has to be given the task-independent state buffer at link time and the buffer is initialized by an HOPEN order. The task-dependent buffer pointer is passed either by HINIT for uninitialized buffers, or by HSYNC for previously initialized buffers. The following operations can be achieved by using the adapter interface control entry points:

- Initialize the adapter - HOPEN followed by an HINIT
- Initialize a new task - HINIT
- Perform a task switch - HSYNC
- Change state data - HSYNC
- Change mode - HSMODE
- Query mode - HQMODE
- Switch to VGA mode - HCLOSE

Note: HCLOSE returns the adapter from advanced-function mode to the VGA mode that was previously in use.

Data Definitions

Byte Numbering Convention

Bytes are accessed using the IBM PC Ordering convention.

Three integer formats are defined:

- Byte (8 bits)
- Word (16 bits)
- Doubleword (32 bits)

Memory is addressed in units of a byte. The least significant byte of a word or doubleword resides in memory at the given address. The most significant byte of a word resides at an offset of + 1 from the given address. The most significant byte of a doubleword resides at an offset of + 3 from the given address:

`Doubleword:`

+3	+2	+1	+0
MSB			LSB

`Word:`

+1	+0
MSB	LSB

Figure 2-3. Byte numbering convention

Bit Numbering Convention

The bit numbering conventions for doublewords, words, and bytes is shown below:

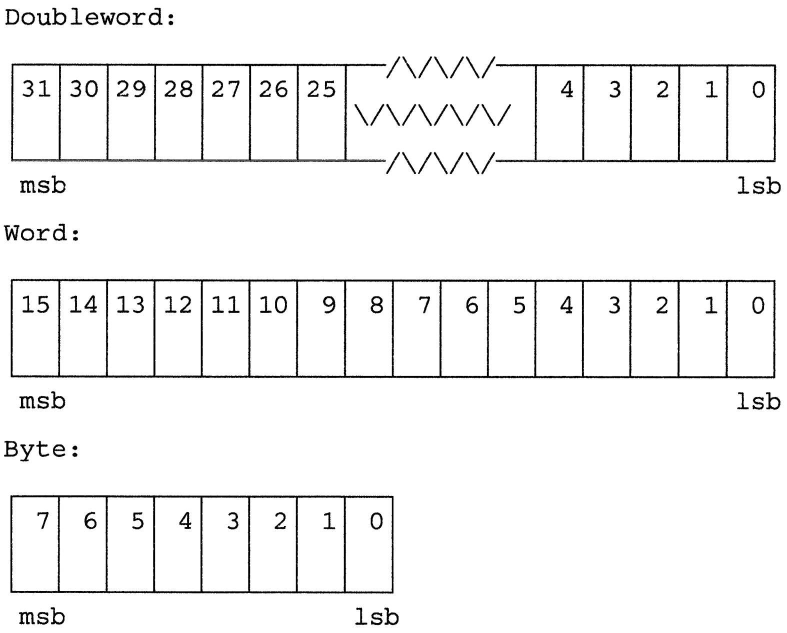


Figure 2-4. Bit numbering convention

The least significant bit has the smallest bit number.

Data Formats

The data formats defined here are the "templates" used to locate data within the fields defined in the orders. Individual orders may use only part of the data.

Coordinate Data

Coordinate data is represented in 2-byte fields; each coordinate is in 2's complement binary representation.

<table>
<tr><td>15
msb</td><td>.....</td><td>0
lsb</td></tr>
<tr><td colspan="3">Coordinate</td></tr>
</table>

Figure 2-5. An Example of Coordinate Data

Relative Coordinate Data

Relative coordinate data is represented in 1-byte fields. Each relative coordinate is an integer in 2's complement binary representation.

<table>
<tr><td>7
sign</td><td>6
msb</td><td>....</td><td>0
lsb</td></tr>
<tr><td colspan="4">Integer</td></tr>
</table>

Figure 2-6. An Example of Relative Coordinate Data

Dimensions

Points are represented in two dimensions:

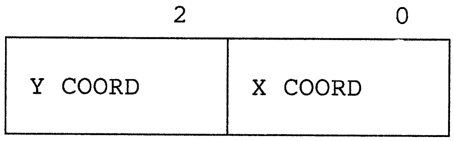


Figure 2-7. The Ordering of Coordinates

- The adapter supports two sets of screen coordinates; 1024 x 768, and 640 x 480
- The origin (0,0) on the adapter is the top left corner of the screen, with X increasing to the right and Y increasing to the bottom
- The adapter coordinate ranges are X (-512 to 1535) and Y (-512 to 1535) with Y going to (-256 to 767) for the smaller mode. The screen coordinates are X (0 to 1023) and Y (0 to 767) or X (0 to 639) and Y (0 to 479). The larger screen size is not supported by all monitors. The adapter coordinate range is greater than the screen coordinate range to allow parts of objects that fall outside the screen to be displayed without wrapping.
- The adapter supports two dimensional coordinates with no fractions and 1-byte relative offsets.

Address Data

At the call interface, entry point parameters refer to areas of memory in the controlling environment. The call interface address data format depends on the controlling environment.

Call interface address data is represented in doubleword (32 bit) fields. The word at displacement plus two is the Segment Selector of the data. The word at displacement zero is the Offset address of the data. Within each word, the least significant byte is the lower addressed byte.

+3 ms byte	+2 ls byte	+1 ms byte	+0 ls byte
Segment Part		Offset Part	

Figure 2-8. IBM PC Address Data

The hardware address is generated using the Intel ®[1] 80286 address-mapping mechanism. In Protect mode, it is the responsibility of the controlling system to:

- Set the 80286 GDT and LDT to address the correct data area
- Ensure that the adapter microcode has an appropriate level of access (read, write, or execute) to the data.

---

[1] Registered trademark of the Intel Corp.

Drawing Entry Points

HLINE - Line at Given Position

Function

The HLINE order defines zero or more connected straight lines.

Entry Point Parameter Block

Byte	Content	Meaning
0	LEN	Length of the following data (≥4)
2	P0	Coordinate data of line start
6	P1	Coordinate data of first line end
.		
.		
2+(n+1)*4		

Figure 2-9. HLINE: Line at Given Position

Description

A line is drawn from the point P0 to the point P1, from P1 to P2, and Pn-1 to Pn. Any number of points can be present, bounded only by the maximum order length. Consecutive points in the order are joined by straight lines.

Current position is set to the last point specified.

An HLINE with only one coordinate sets the Current Position.

The order resets the line pattern.

HCLINE - Line at Current Position

Function

This order defines zero or more connected straight lines.

Entry Point Parameter Block

Byte	Content	Meaning
0	LEN	Length of following data (≥0)
2	P1	Coordinate data of first line end
6	P2	Coordinate data of second line end
.		
.		
2+n*4		

Figure 2-10. HCLINE: Line at Current Position

Description

A line is drawn from the current position to the point P1, from P1 to P2, and Pn-1 to Pn. Any number of points can be present, bounded only by the maximum order length. Consecutive points in the order are joined by straight lines.

The current position is set to the last point specified.

The order does not reset the line pattern.

HRLINE - Relative Line at Given Position

Function

This order defines zero or more connected straight lines, as with HLINE, except that the end point of each line is given as an offset from the start of the line, rather than absolute coordinates.

Entry Point Parameter Block

Byte	Content	Meaning
0	LEN	Length of following data (≥4)
2	P0	Coordinate data of line start
6	OFF1	Offset data of first line end relative to the start point
7	OFF2	Offset data of second line end relative to the first line end
.		
.		
6+n		

Figure 2-11. HRLINE: Relative Line at Given Position

Description

The offsets are added cumulatively to P0 to generate a sequence of points P1, P2, ..., Pn, where n is the number of offsets specified.

$Pa = P0 + OFF1 + OFF2 + \ldots + OFFa$
for $1 \leq a \leq n$

Straight lines are drawn connecting the points in sequence, that is from P0 to P1, from P1 to P2, ... and from Pn-1 to Pn. Any number of offsets can be present, bounded only by the maximum order length. The current position is set to the last point specified.

An HRLINE with no offsets sets the current position. The effect of an HRLINE that causes coordinates to exceed the coordinate space is undefined. The order resets the line pattern.

HCRLINE - Relative Line at Current Position

Function

This order defines zero or more connected straight lines, as with HCLINE, except that the end point of each line is given as an offset from the start of the line, rather than absolute coordinates.

Entry Point Parameter Block

Byte	Content	Meaning
0	LEN	Length of following data (≥ 0)
2	OFF1	Offset data of first line end relative to its start point
3	OFF2	Offset data of second line end relative to the first line end
.		
.		
2+n		

Figure 2-12. HCRLINE: Relative Line at Current Position

Description

The offsets are added cumulatively to the current position (CP) to generate a sequence of points P1, P2, ..., Pn, where n is the number of offsets specified.

$Pa = CP + OFF1 + OFF2 + ... + OFFa$
for $1 \leq a \leq n$

Straight lines are drawn connecting the points in sequence, that is from CP to P1, from P1 to P2, ... and from Pn-1 to Pn.

Any number of offsets can be present, bounded only by the maximum order length. The current position is set to the last point specified. An HCRLINE which causes coordinates to exceed the coordinate space is undefined.

HBAR - Begin Area

Function

This order is used to indicate the beginning of a set of primitives that define the boundary of an area. The end of the set of primitives is indicated by an End Area (HEAR) order.

Entry Point Parameter Block

Dummy.

Description

The HBAR sets an area drawing mode in the drawing process.

Line-drawing orders which follow the HBAR order and precede an HEAR order are drawn into the area-fill buffer in the form required to construct a filled area.

The attributes used to draw the area are those current when the HEAR order is interpreted.

Optimization

Area fill processing may be optimized to process the smallest rectangle that contains the area. A few small widely-spaced areas should be drawn separately.

Rectangles should be filled using the special HRECT order.

Usage Notes

The area boundary definition is cleared by the HEAR order.

The area boundary consists of one or more closed figures. Each closed figure is made up of a contiguous set of line objects. The first closed figure within an area is defined as starting at the first drawing order after the HBAR order. It is delimited either by a HEAR order, or any "move" type of order valid within an area definition (that is, HSCP, HLINE and HRLINE). This means that if a closed figure is not properly closed (that is, the start and end points are not identical), it is automatically closed by a straight line connecting the start and end pointer.

The figures formed in this way jointly define the area boundary. Any connected region with an odd number of line crossings from the edge of the screen will be shaded for that part of its area which lies within the scissor rectangle. Note that connecting regions with an even number of line crossings will *not* be shaded, and in counting line crossings, coincident boundary lines are all counted.

When an area is shaded, left or top boundaries will be treated as part of the area, and shaded. Right or bottom boundaries will not be treated as part of the area, and will be unchanged by the area fill.

The value of Current Position is not changed by the HBAR order itself, but is changed by those orders used to define the area boundary, including any implicit figure-closing orders that are required (that is, HSCP, HLINE and HRLINE).

Area orders may not be nested; If they are, the effect of all nested area definitions becomes undefined.

This order has a dummy parameter block which means that the length field of the parameter block should be zero.

Areas

The following orders are *not* valid between HBAR and HEAR:

HOPEN, HCLOSE, and HSMODE Control

HSHS	Set scissor
HSCMP	Set color comparison
HSBP	Set bit plane controls
HCHST, HCCHST	Character string orders

The use of any of these orders within an area definition causes the effect of the whole area definition to become undefined.

Note: Other drawing primitives that are allowed between HBAR and HEAR (for example, Image) interfere with the "current position"; this should be managed by the controlling environment.

There are parts of the bit plane storage which cannot be displayed, called the **auxiliary bit plane storage**. This storage can be used by the controlling environment, but any conflict with the adapter's use of the storage must be avoided.

Auxiliary bit plane storage has three uses:

1. Area fill operations
2. Graphics text (image) cache
3. Marker drawing operations.

Normally, the separate demands on this resource (although mutually exclusive) do not compete. However, if an area definition is suspended, then any multi-colored image-graphics text drawn may invalidate the suspended area definition. It is the responsibility of the controlling environment to manage this possible resource exclusivity.

HEAR - End Area

Function

This order is used to indicate the end of a set of primitives that define the boundary of an area.

Entry Point Parameter Block

Byte	Content	Meaning
0	LEN	Length of following data (1)
2	BYTE	Flags Bit 7-6 00 - Fill area defined previously 01 - Suspend area definition 10 - Abort area definition Bit 5-0 Reserved
.		
.		
3		

Figure 2-13. HEAR: End Area

Description

A HEAR order identifies the end of an area.

The HEAR order causes the area between the special outline drawn in the area fill buffer to be shaded (see the HBAR order). The shaded area is then merged with the picture using the current values of pattern, color, and mix.

If the Fill Defined Area option is selected then the area-fill algorithm will be applied to the area outline.

If the Suspend Area Definition option is selected, the area outline will be left in the area-fill buffer. However, when the area definition is resumed, processing should continue on the same area that was suspended. Also, if any attribute values needed for the fill operation have been corrupted during suspension, then it is the responsibility of the controlling environment to restore them.

The abort option means that no area is filled and the boundary definition is cleared.

Optimization

See the HBAR order.

Usage Notes

See the HBAR order.

HRECT - Fill Rectangle

Function

This order performs a rectangular fill.

Entry Point Parameter Block

Byte	Content	Meaning
0	LEN	Length of following data (8)
2	COORD	Top left corner of rectangle
6	WIDTH	Width of rectangle
8	HEIGHT	Height of rectangle
.		
.		
10		

Figure 2-14. HRECT: Fill Rectangle

Description

The rectangular area specified in the parameter block is filled and then merged with the bit-plane contents, using the current values of pattern, color, and mix. The HRECT update is limited to that part of the screen space within the hardware scissor rectangle.

Usage Notes

The current position is set to the top left-hand corner of the rectangle after it has been displayed.

This is a faster fill than the HBAR-HEAR fill.

HRECT supports only mono-patterned fills (that is, multi-colored patterned fills are not supported).

HMRK - Marker at Given Position

Function

This order draws the current marker symbol at one or more positions.

Entry Point Parameter Block

Byte	Content	Meaning
0	LEN	Length of following data (≥4)
2	P0	Coordinate data of first marker
6	P1	Coordinate data of second marker
.		
.		
2+(n+1)*4		

Figure 2-15. HMRK: Marker at Given Position

Description

This order draws a marker symbol at one or more points in order space. The first marker is drawn at point P0; further marker symbols are drawn at the remaining points specified in the order.

The current position is set to the last coordinate specified.

At least one marker is always drawn.

Usage Notes

The marker will be positioned such that the top left hand corner of the marker cell is located at Xpn-(cx/2), Ypn-(cy/2); that is, the positioning coordinate specifies the approximate middle of the marker (where cx=width and cy=height).

HCMRK - Marker at Current Position

Function

This order draws the current marker symbol at one or more positions.

Entry Point Parameter Block

Byte	Content	Meaning
0	LEN	Length of following data (≥0)
2	P1	Coordinate data of second marker
6	P2	Coordinate data of third marker
.		
.		
2+n*4		

Figure 2-16. HCMRK: Marker at Current Position

Description

This order draws a marker symbol at one or more points. The first marker is drawn at the current position; further marker symbols are drawn at the remaining points specified in the order.

Current position is set to the last coordinate specified, or, if none, is unchanged.

The current marker symbol is the marker drawn.

At least one marker is always drawn.

Usage Notes

The marker will be positioned such that the top left-hand corner of the marker cell is located at Xpn-(cx/2), Ypn-(cy/2); that is, the positioning coordinate specifies the approximate middle of the marker (where cx=width and cy=height).

Image Orders

"Across the plane image" performance is faster for nibble-aligned coordinates (that is, X coordinates divisible by 4). It is also faster if the image is not a subrectangle of a bigger image in storage.

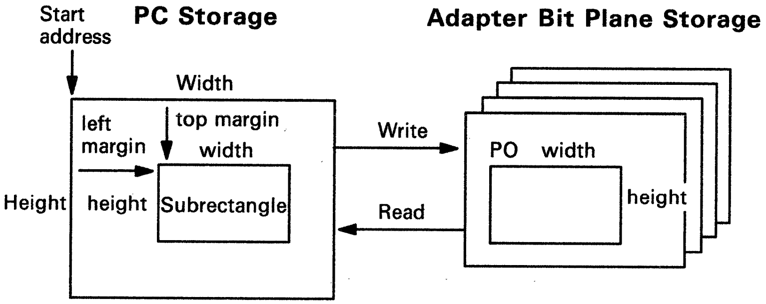


Figure 2-17. Image Processing Illustration

Notes:

In Figure 2-17:

1. Width and height are in bits for across-the-plane data, and in bytes for through-the-plane data.
2. If "subrectangle" is specified, then:
   - width/height = subrectangle width/height, else -
   - width/height = full image width/height.
3. All the information in Figure 2-17 is specified in one of the image orders HBBW, HCBBW, or HBBR, except for the "start address", which is given in the HBBCHN order.

HBBW - BITBLT Write Image Data

Function

The HBBW order identifies the start of a rectangular block of data which is to be written to the bit planes.

Entry Point Parameter Block

Byte	Content	Meaning
0	LEN	Length of following data (10 or 18)
2	FORMAT	The format of the BITBLT data X'0000' - across the planes X'0008' - through the planes byte
4	WIDTH	The width of the BITBLT data
6	HEIGHT	The height of the BITBLT data
8	P0	Coordinates of the position at which the image data is drawn
12	WORD	Left margin (PELs)
14	WORD	Top margin (PELs)
16	WIDTH	Width of rectangle
18	HEIGHT	Height of rectangle
.		
.		
20		

Figure 2-18. HBBW: BITBLT Write Image Data

Description

The HBBW order defines a rectangular image at the given position. Subsequent HBBCHN orders pass data for the image.

The size of the image block in the controlling environment is given by the WIDTH and HEIGHT parameters in order space units.

The position of the image (top/left corner) is given by P0 in order space coordinates.

If the order length field is equal to 18 bytes, a rectangle is extracted from the data in storage. The rectangle is defined (in PELs) by the left margin, top margin, width of rectangle, and height of rectangle. Only this rectangle is written to the bit planes.

Current position is set to P0.

FORMAT X'0000'

Format X'0000' defines "across the plane" data. Each bit of data corresponds to one order space element. Wherever a binary one is encountered, the current color is written with the current mix. Wherever a binary 0 is encountered, the background color is written with the background mix.

The first PEL in each row of data must begin on a byte boundary. Padding bits are ignored where the width is not exactly divisible by 8.

FORMAT X'0008'

Format X'0008' defines "through the plane" data with one byte per picture element. Each byte of data corresponds to part of the color index of an order space picture element. The byte is padded with zeros to the left to form a color index. The resulting color index is written to the bit planes under the control of the current mix, and the planes are enabled for update.

Usage Notes

The data for the image is supplied by the following HBBCHN orders. The image is terminated by any image order other than HBBCHN.

The image data is drawn in adjacent picture elements starting with the picture element at the top/left corner of the block given by P0, and working horizontally to the right along the first row of picture elements. The second row begins in the picture element immediately below the start of the first row, and continues to the right.

HCBBW - BITBLT Write Image Data at Current Position

Function

This order identifies the start of a rectangular block of data which is to be written to the bit planes.

Entry Point Parameter Block

Byte	Content	Meaning
0	LEN	Length of following data (6 or 14)
2	FORMAT	The format of the BITBLT data X′0000′ - across the planes X′0008′ - through the planes byte
4	WIDTH	The width of the BITBLT data
6	HEIGHT	The height of the BITBLT data
8	WORD	Left margin (PELs)
10	WORD	Top margin (PELs)
12	WIDTH	Width of rectangle
14	HEIGHT	Height of rectangle
.		
.		
16		

Figure 2-19. HCBBW: BITBLT Write Image Data at Current Position

Description

This order defines an image block at the current position.

The size of the image block in the controlling environment is given by the WIDTH and HEIGHT parameters in order space units.

The position of the image (top/left corner) is the current position.

If the order length field is equal to 14 bytes then a rectangle will be extracted from the data in storage. The rectangle is defined by the left margin, top margin, width of rectangle and height of rectangle in PELs. Only this rectangle is written to the bit planes.

The current position is unchanged.

FORMAT X′0000′

Format X′0000′ defines "across the plane" data. Each bit of data corresponds to one order space element. Wherever a binary 1 is encountered, the current color is drawn with the current mix. Wherever a binary 0 is encountered, the background color is written with the background mix.

The first PEL in each row of data must begin on a byte boundary. Padding bits will be ignored where the width is not exactly divisible by 8.

FORMAT X'0008'

Format X'0008' defines "through the plane" data with one byte per picture element. Each byte of data corresponds to part of the color index of an order space picture element. The byte is padded with zeros to the left to form a color index. The resulting color index is written to the bit planes under the control of the current mix and the planes enabled for update.

Usage Notes

The data for the image is supplied by the following HBBCHN orders. The image is terminated by any image order other than HBBCHN.

Each HBBCHN order supplies an arbitrary length of data. The image data may be split between more than one HBBCHN order.

The image data is drawn in adjacent order space picture elements, starting with the order space element at the top/left corner of the block given by P0, and working horizontally to the right along the first row of order space elements. The second row begins in the order space picture element immediately below the start of the first row, and continues to the right.

HBBR - BITBLT Read Image Data

Function

The HBBR order identifies the start of a rectangular block of data to be moved from the the bit planes.

Entry Point Parameter Block

Byte	Content	Meaning
0	LEN	Length of following data (12 or 20)
2	FORMAT	The format of the BITBLT data X′0000′ - across the planes X′0008′ - through the planes byte
4	WIDTH	The width of the BITBLT data
6	HEIGHT	The height of the BITBLT data
8	BYTE	Across the planes source bit plane (see Note below)
9	BYTE	Reserved
10	P0	Coordinates of the position from which the image data is read
14	WORD	Left margin (PELs)
16	WORD	Top margin (PELs)
18	WIDTH	Width of rectangle
20	HEIGHT	Height of rectangle
.		
.		
22		

Figure 2-20. HBBR: BITBLT Read Image Data

Note: This is a number, not a mask.

Description

The HBBR order defines a rectangular image at the given position. Subsequent HBBCHN orders receive data from the image.

The size of the image block given by the WIDTH and HEIGHT parameters in order space units.

The position of the image (top left corner) is given by P0 in order space coordinates.

The current position is set to P0.

If the order length field is equal to 20 bytes, then a rectangle is inserted into data in storage. The rectangle is defined (in PELs) by the left margin, top margin, width of rectangle, and height. Only this rectangle is read into storage. The rectangle in storage need not be byte aligned. Data outside the rectangle will not be corrupted by the read operation.

FORMAT X'0000'

Format X'0000' defines "across-the-plane" data. The data is extracted from the specified bit plane. Each bit of data corresponds to one order space picture element.

The first PEL in each row of data begins on a byte boundary. Padding bits (zeros) will be added where the width is not exactly divisible by eight.

FORMAT X'0008'

Format X'0008' defines "through-the-plane" data with 1 byte per picture element. Each byte of data corresponds to part of the color index of an order-space picture element. The least significant byte of the color index is extracted and stored in the HBBCHN order.

Usage Notes

The data for the image is returned in the following HBBCHN orders. The image is terminated by any image order other than HBBCHN.

Each HBBCHN order supplies an arbitrary buffer length. The image data may be split between one or more HBBCHN orders. If the chain is terminated before the image area has been completely read, the remaining part of the image block remains unread. If the chain provides an excessive area of storage, the remaining storage is not modified.

The image data is read from the top/left corner of the block given by P0, and working horizontally to the right along the first row of picture elements. The second row begins in the picture element immediately below the start of the first row and continues to the right.

HBBCHN - BITBLT Chained Data

Function

The HBBCHN order holds image data for the BITBLT orders.

Entry Point Parameter Block

Byte	Content	Meaning
0	LEN	Length of following data (6)
2	WORD	Address of data in controlling system
6	WORD	Length of data in controlling system
.		
.		
8		

Figure 2-21. HBBCHN: BITBLT Chained Data

Description

This order defines data forming (part of) an image block.

The data buffer is defined by an address and length within the order.

The content of the data buffer supplied by the order is defined by the parameters of the HBBW, HCBBW, or HBBR order that preceded the HBBCHN order.

The data addressed by an HBBCHN order must constitute a complete rectangle within the image.

The current position is not changed by the order.

Usage Notes

See the HBBW, HCBBW, and HBBR orders for a description of the BITBLT operations.

HBBC - BITBLT Copy

Function

This order copies a BITBLT block within the bit planes.

Entry Point Parameter Block

Byte	Content	Meaning
0	LEN	Length of following data (16)
2	FORMAT	The format of the BITBLT data X'0000' - across the plane copy X'0008' - through the plane copy
4	WIDTH	The width of the BITBLT data
6	HEIGHT	The height of the BITBLT data
8	BYTE	Across-the-plane source bit plane
9	BYTE	Reserved
10	P0	Coordinates of the source data
14	P1	Coordinates of the destination
.		
.		
18		

Figure 2-22. HBBC: BITBLT Copy

Description

The HBBC order copies a rectangular image at the given source position in order space to the given destination position.

If the areas overlap, the destination will overlay the source. In all other cases the source will be unchanged.

The size of the image block is given by the parameters WIDTH and HEIGHT in order-space units.

The source and destination positions refer to the top left corner of the image in order space.

Current position is set to P1.

FORMAT X'0000'

Format X'0000' defines "across the plane" data. The data is extracted from the specified bit plane. Wherever a binary one is encountered in the source bit plane, the current color is written with the current mix. Wherever a binary zero is encountered, the background color is written with the background mix.

FORMAT X'0001'

Format X'0001' defines "through the planes" data. The color index for each picture element of the source image is written to the destination image under the current mix.

Control Entry Points

HOPEN - Open Adapter

Function

The HOPEN order initializes the adapter.

Entry Point Parameter Block

Byte	Content	Meaning
0	LEN	Length of following data (3)
2	BYTE	Flags Bit 7 - don't clear bit planes Bit 6 - don't load a default palette Bit 5-0 reserved
3	BYTE	Mode
4	BYTE	Return flags Bit 7 - driver hardware mismatch Bit 6-0 reserved
.		
.		
5		

Figure 2-23. HOPEN: Open Adapter

Description

The HOPEN order initializes the task-independent state; this includes loading a default palette.

Usage Notes

- If byte 2 of the parameter block is 0, the bit planes are cleared and the default palette is loaded.
- For details of byte 3 (mode) see "HSMODE - Set Mode" on page 2-50.
- If HOPEN returns zero in byte 4, then the "open" was successful.

See "State Data" on page 2-11 for a description of states and tasks.

HCLOSE - Close Adapter

Function

The HCLOSE order returns control of the display device to the default display adapter in the system unit.

Entry Point Parameter Block

Dummy.

Description

The HCLOSE order switches from advanced-function mode back to the previous VGA mode.

Usage Notes

The length field of the dummy parameter block should be zero.

HSCP - Set Current Position

Function

The HSCP order sets the current position.

Entry Point Parameter Block

Byte	Content	Meaning
0	LEN	Length of following data (4)
2	P0	Coordinate data
.		
.		
6		

Figure 2-24. HSCP: Set Current Position

Description

The current position is set to the value specified in the order.

The order does not reset the line pattern.

HQCP - Query Current Position

Function

The HQCP order returns the current position coordinates.

Entry Point Parameter Block

Byte	Content	Meaning
0	LEN	Length of following data (4)
2	P0	Coordinate data
.		
.		
6		

Figure 2-25. HQCP: Query Current Position

Description

The coordinates of the current position are returned in the data area provided.

HQDFPAL - Query Default Palette

Function

The HQDFPAL order returns a table that can be used to identify specific colors in the default palette.

Entry Point Parameter Block

Byte	Content	Meaning
0	LEN	Length of following data (64)
2	DATA	Color index data
.		
.		
66		

Figure 2-26. HQDFPAL: Query Default Palette

Description

The adapter returns the color index values which give the following colors when the default palette is loaded:

Index	Color
0	Black
1	Blue
2	Green
3	Cyan
4	Red
5	Magenta
6	Brown
7	White
8	Gray
9	Light Blue
10	Light Green
11	Light Cyan
12	Light Red
13	Light Magenta
14	Yellow
15	High Intensity White

Usage Notes

Each color index value is 4 bytes.

HINIT - Initialize State

Function

The HINIT order sets the task dependent data to an initial state.

Entry Point Parameter Block

Byte	Content	Meaning
0	LEN	Length of following data (2)
2	ADDRESS	Task state buffer
.		
.		
4		

Figure 2-27. HINIT: Initialize State

Description

The HINIT order sets the task dependent data to an initial state. The initial state is defined by the adapter implementation.

The address parameter is a 2-byte segment (selector) only.

Usage Notes

See "State Data" on page 2-11 for a description of task-dependent state and single/multi-tasking environments.

HSYNC - Synchronize Adapter

Function

The HSYNC order synchronizes the adapter hardware with a given task state.

Entry Point Parameter Block

Byte	Content	Meaning
0	LEN	Length of following data (2)
2	ADDRESS	Task state buffer
.		
.		
4		

Figure 2-28. HSYNC: Synchronize Adapter

Description

The HSYNC order sets the adapter state to match the task dependent state supplied as a parameter.

The address parameter is a 2-byte segment (selector) only.

Usage Notes

In a multi-tasking environment, this order would be used to implement a task switch. (See “State Data” on page 2-11 for a description of the task-dependent state and single/multi-tasking environments.)

HINT - Interrupt

Function

This order synchronizes with a hardware event or interrupt.

Entry Point Parameter Block

Byte	Content	Meaning
0	LEN	Length of following data (4)
2 . . 6	DATA	Interrupt/event identifier

Figure 2-29. HINT: Interrupt

Description

This order returns control to the controlling environment when the requested interrupt/event occurs.

Usage Notes

The controlling system can use this order to synchronize with a hardware event.

Bits in the interrupt/event identifier are defined as follows:

- Bit 31 = Frame flyback
- Bits 30-0 = Reserved.

The controlling system should set bits in the interrupt/event identifier to indicate which events should cause the order to complete. Following any of the specified events, the adapter will clear all the event bits except the one bit that indicates the cause of the interrupt.

HSMODE - Set Mode

Function

The HSMODE order sets the adapter mode.

Entry Point Parameter Block

Byte	Content	Meaning
0	LEN	Length of following data (1)
2	BYTE	Screen format example

number alpha cells	alpha cell size	screen size (PELs)
00 - 85x38	12x20	1024x768
01 - 80x34	8x14	640x480
02 - 128x54	8x14	1024x768
03 - 146x51	7x15	1024x768

.
.
3

Figure 2-30. HSMODE: Set Mode

Description

The adapter is set to the mode specified by the mode byte.

HQMODE - Query Current Mode

Function

The HQMODE order returns data specifying mode and configuration.

Entry Point Parameter Block

Byte	Content	Meaning
0	LEN	Length of following data (18)
2	DATA	Mode number
3	DATA	Driver code level
5	DATA	Adapter type
6	DATA	Display type (reserved)
7	DATA	Alpha cell width (PELs)
8	DATA	Alpha cell height (PELs)
9	DATA	Number of bit planes
10	DATA	Screen width (PELs)
12	DATA	Screen height (PELs)
14	DATA	PELs/inch horizontal
16	DATA	PELs/inch vertical
18	DATA	Mono/Color (0 = mono, FF = color)
19	DATA	Intensity levels. Specified as the number of significant bits in each palette color entry (number of bits in the DAC for each color). Zero means non-loadable palette.
.		
.		
20		

Figure 2-31. HQMODE: Query Current Mode

Description

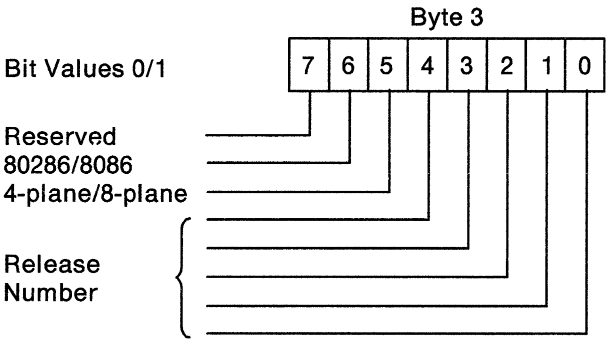
Using this order the controlling environment is able to query the current adapter mode and configuration. The data is returned in the data area supplied.

Usage Notes

- Mode number - valid replies are 00, 01, 02, and 03. These are defined in the HSMODE command.
- Driver code level - this is adapter dependent and individual adapter implementations should specify code-level numbering conventions.

For this adapter, the following code-level numbering convention is used:

HQMODE returned data



- Adapter type - each adapter has an identifier which is “3” for this adapter.
- Mono/Color and Intensity Levels enables you to load the palette correctly.
- Number of bit planes - valid replies are 4 (for minimum memory) and 8 (for maximum memory).
- PELs-per-inch (horizontal and vertical) - the valid replies are:
  - 79 - IBM Color Display 8512 or IBM Color Display 8513 in 640 x 480 mode
  - 59 - IBM Color Display 8514 in 640 x 480 mode
  - 92 - IBM Color Display 8514 in 1024 x 768 mode.

HQMODES - Query Adapter Modes

Function

The HQMODES order returns data specifying modes available at the interface.

Entry Point Parameter Block

Byte	Content	Meaning
0	LEN	Length of following data (33)
2	DATA	Adapter Type
3	DATA	Modes
.		
.		
35		

Figure 2-32. HQMODES: Query Adapter Modes

Description

Using this order, the controlling environment is able to query the modes that are available. The mode data is returned in the data area supplied.

Because the number of modes supported is variable, the returned data is terminated by an 'FF'X byte. The controlling environment should reserve sufficient buffer space for the modes data (that is, 35 bytes).

For a description of adapter types, see "HQMODE - Query Current Mode" on page 2-51; for a description of modes see "HSMODE - Set Mode" on page 2-50.

HEGS - Erase Graphics Screen

Function

This causes the screen to be cleared.

Entry Point Parameter Block

Dummy

Description

The HEGS order sets bit planes to 0.

The area set to 0 is limited by the hardware scissor.

Planes that are not enabled for update are not modified by this order. The current position is set to the top left-hand corner of the screen.

Usage Notes

This order has a dummy parameter block which means that the length field should be 0.

HSGQ - Set Graphics Quality

Function

This order sets the value of Current Graphics Quality to that specified in the order.

Entry Point Parameter Block

Byte	Content	Meaning
0	LEN	Length of following data (2)
2	DATA	Graphics Quality value. Values are:
		Bit 15 = B'0' Reserved
		Bit 14 = B'0' High precision
		Bit 14 = B'1' Low precision
		Bit 13 = B'0' Reserved
		Bit 12-11 = B'00' Last PEL null = B'01' Last PEL on = B'10' Conditional last PEL
		Bit 10-0 Reserved
.		
.		
4		

Figure 2-33. HSGQ: Set Graphics Quality

Description

Graphics Quality is set to the value specified in the order.

Each of the bits of the Graphics Quality definition affect the speed of drawing and the quality of the picture:

- High precision

  The adapter performs calculations that draw the object correctly to the nearest picture element.

- Low precision

  The adapter uses algorithms that draw an approximation of the object as rapidly as possible.

- Last PEL null

  When this bit is set, vector drawing always draws the last PEL; when it is not set, the last PEL is not drawn.

- "Conditional last PEL" means lines drawn in overpaint/AND/OR mixes will be drawn, including first and last PEL. Polylines drawn in other mixes (for example, XOR) will be drawn with an adapter-dependent approximation.

Optimization

The highest drawing speed will be achieved with low precision.

Usage Notes

The HSGQ order allows the controlling system to globally control the compromise between performance and picture quality. The setting of the Graphic Quality attribute overrides attributes and parameters of other orders.

If the foreground mix is one of the following, then each line segment within the line orders (for example, HLINE) is drawn with the last PEL null. Otherwise, the last PEL of each line segment is drawn. Low precision drawing is not supported.

- X'04' XOR
- X'08' Add
- X'09' Subtract (screen-new)
- X'0A' Subtract (new-screen)
- X'0B' Average
- X'12' not screen and new
- X'16' screen XOR new
- X'18' not screen and not new
- X'19' not screen xor new
- X'1A' not screen
- X'1B' not screen or new
- X'1E' not screen or not new.

HSHS - Set Scissor

Function

This causes the drawing process scissor rectangle to be set.

Entry Point Parameter Block

Byte	Content	Meaning
0	LEN	Length of following data (0 or 8)
2	COORD	Left limit of the rectangle
4	COORD	Right limit of the rectangle
6	COORD	Bottom limit of the rectangle
8	COORD	Top limit of the rectangle
.		
.		
10		

Figure 2-34. HSHS: Set Scissor

Description

The scissor rectangle is set to the values specified by the order.

If the data length is 0, the scissor is restored to HOPEN default.

If the left limit is to the right of the right limit, or the bottom limit is above the top limit, all graphic objects are discarded.

All drawing operations are limited to the rectangle defined by the scissor.

Usage Notes

The drawing process scissor inhibits the drawing of objects outside a rectangle defined in screen coordinate space. The boundary of the rectangle is inclusive (that is, PELs on the boundary are drawn). The scissoring is performed by the drawing process after vector-to-raster conversion.

Objects that fall outside screen space may be drawn within the scissor window. In this case, an additional clip function is required to limit the data to adapter coordinate space.

The scissor can be used in conjunction with clip to ensure that dotted, dashed and thick lines join correctly with existing data.

The scissor is done by the adapter.

The scissor should normally be set to screen coordinates only; if the scissor is set to coordinates outside the screen space, the controlling environment has to be aware that it may then be possible to invalidate area-fill operations or the caching of character definitions by writing to the area-fill buffer.

HLDPAL - Load Palette

Function

The HLDPAL order loads a palette into the color lookup tables.

Entry Point Parameter Block

Byte	Content	Meaning
0	LEN	Length of following data (1 or 10)
2	BYTE	Palette Identification Bits 0 to 7: X′00′ Load user-defined palette X′01′ Load adapter default
3		Reserved
4	INDEX	Number of first entry to be loaded
6	COUNT	Number of entries to be loaded
8 . . 12	ADDRESS	Address of the palette entries in storage

Figure 2-35. HLDPAL: Load Palette

Description

The HLDPAL order loads the color lookup table with a predefined or user-defined palette.

The address of the palette entries is a program (virtual) address in the controlling environment.

Each palette entry is a 4-byte field. The first byte defines the red intensity, the second byte defines the blue intensity, and the third byte defines the green intensity. The fourth byte is reserved. Each intensity is an integer between 0 and 255, with 255 representing the maximum intensity.

The order allows a part of the color lookup table to be loaded. All other parts of the color lookup table are left unmodified.

Usage Notes

If the adapter default is specified, bytes 3-8 are redundant and need not be supplied.

- Color index bits 8 through 31 are ignored.
- The color palette has 256 entries. The high order bits of any color index beyond this range are ignored.

  The palette used by the adapter is the same as that used by the system planar display adapter. Therefore, if you use an advanced-function mode and load a palette, the original base mode palette will be lost. If you want to return to base mode and use the original palette, then the HSPAL/HRPAL orders must be used in order to save (before HOPEN) and restore (after HCLOSE) the palette contents.

  Also, because mode changes *may* corrupt the palette, if that palette is required in the new mode then it is advisable to (1) save the palette before a mode change, and (2) restore it after the mode change.
- Only the most significant 6 bits of each of the bytes in the palette entries are used.

HSPAL - Save Palette

Function

The HSPAL order saves the contents of the color palette and the display mask.

Entry Point Parameter Block

Byte	Content	Meaning
0	LEN	Length of following data (defined by HQDPS)
2	DATA	Buffer
.		
.		
771		

Figure 2-36. HSPAL: Save Palette

Description

This order saves the contents of the palette and the display mask, which can later be restored using the HRPAL order.

Usage Notes

This order, along with the HRPAL order, can be used to preserve the palette integrity during operations that may corrupt it.

The buffer size necessary for the palette contents is returned as a parameter of the HQDPS order.

HRPAL - Restore Palette

Function

The HRPAL order restores the contents of the color palette and the display mask.

Entry Point Parameter Block

Byte	Content	Meaning
0	LEN	Length of following data (defined by HQDPS)
2	DATA	Buffer
.		
.		
771		

Figure 2-37. HRPAL: Restore Palette

Description

This order restores the contents of the palette and the display mask which were previously saved by the HSPAL order.

Usage Notes

This order, along with the HSPAL order, can be used to preserve the palette integrity during operations that may corrupt it.

The buffer size necessary for the palette contents is returned as a parameter of the HQDPS order.

HSLPC - Save Line Pattern Count

Function

The HSLPC order saves the current line-pattern count.

Entry Point Parameter Block

Dummy

Description

This order saves the current line pattern count, which can later be restored using the HRLPC order.

Usage Notes

This order, along with the HRLPC order, can be used to achieve continuity in a patterned line which straddles a scissor boundary.

This order has a dummy parameter block, which means that the length field should be 0.

HRLPC - Restore Line Pattern Count

Function

The HRLPC order restores the saved line-pattern count.

Entry Point Parameter Block

Dummy.

Description

This order restores the line-pattern count that was previously saved using the HSLPC order.

Usage Notes

This order, along with the HSLPC order, can be used to achieve continuity in a patterned line which straddles a scissor boundary.

The value that is restored will be either the adapter default (if no HSLPC orders have been issued since initialization) or the value saved by the last HSLPC order.

This order has a dummy parameter block, which means that the length field should be 0.

HSBP - Set Bit Plane Controls

Function

The HSBP order selects and deselects bit planes for update and display.

Entry Point Parameter Block

Byte	Content	Meaning
0	LEN	Length of following data (12)
2	DOUBLEWORD	Planes selected for update bit mask (Graphics/Text)
6	DOUBLEWORD	Planes selected for update bit mask (Alphanumerics)
10	DOUBLEWORD	Planes enabled for display bit mask
.		
.		
14		

Figure 2-38. HSBP: Set Bit Plane Controls

Description

The "planes selected for update bit mask" specifies which planes can be updated under the drawing mix (see "HSMX - Set Mix" on page 2-80). All other planes will remain unmodified.

The "planes enabled for display bit mask" specifies which planes will be enabled to index the color palette. All other planes will give binary 0 values to the serializer/palette at all times.

The specification of nonexistent bit planes in the masks are ignored.

Usage Notes

These controls override all other controls. For example, if the current color index of X′00000018′ is set, but the planes enabled for update of X′00000010′ is set, then only plane 4 will be altered by a draw in current color.

HQCOORD - Query Coordinate Types

Function

The HQCOORD order verifies support for a coordinate type.

Entry Point Parameter Block

Byte	Content	Meaning
0	LEN	Length of following data (4)
2	BYTE	Coordinates format Bits 4-7: No. of bytes in each coordinate Bits 0-3: No. of fractional bytes in each coordinate
3	BYTE	Relative coordinate format Bits 4-7: No. of bytes in each relative coordinate Bits 3-0: No. of fractional bytes in each relative coordinate
4	BYTE	Dimensions: (2,3,4)
5	RESULT	Flags: Bit 7 0 - Coord format supported 1 - Coord format not supported Bit 6 0 - Relative format supported 1 - Relative format not supported Bit 5 0 - Dimensions supported 1 - Dimensions not supported
. . 6		

Figure 2-39. HQCOORD: Query Coordinate Types

Description

The HQCOORD order examines the contents of the coordinate, relative coordinate, and dimensions fields, and sets the appropriate flag if the format is not supported by the adapter.

Usage Notes

The HQCOORD order can be used to verify an HSCOORD order before the HSCOORD order is passed to the adapter.

HSCOORD - Set Coordinate Types

Function

The HSCOORD order sets the coordinate format.

Entry Point Parameter Block

Byte	Content	Meaning
0	LEN	Length of following data (3)
2	BYTE	Coordinate format Bits 4-7: No. of bytes in each coordinate Bits 0-3: No. of fractional bytes in each coordinate
3	BYTE	Relative coordinate format Bits 4-7: No. of bytes in each relative coordinate Bits 0-3: No. of fractional bytes in each relative coordinate
4 . . 5	BYTE	Dimensions: (2,3,4)

Figure 2-40. HSCOORD: Set Coordinate Types

Description

The HSCOORD order sets the current coordinate type and number of dimensions representing a point in coordinate space.

Usage Notes

The HSCOORD order is a NOOP for this adapter.

HESC - Escape - Terminate Processing

Function

The HESC order is used to stop adapter processing.

Entry Point Parameter Block

Dummy.

Description

When the HESC order is received, adapter processing is stopped.

Usage Notes

This order has a dummy parameter block which means that the length field should be 0.

The HESC order is a NOOP for this adapter.

HQDPS - Query Drawing Process State Size

Function

The HQDPS order returns the size of the task dependent state data area and the driver stack usage.

Entry Point Parameter Block

Byte	Content	Meaning
0	LEN	Length of following data (6)
2	WORD	Buffer size (in bytes)
4	WORD	Stack usage (in bytes)
6	WORD	Save palette buffer size (in bytes)
.		
.		
8		

Figure 2-41. HQDPS: Query Drawing Process State Size

Description

The buffer-size field is set to the length (in bytes) of the data area required for the task-dependent state.

The stack usage field is set to the maximum stack requirement of the adapter interface driver code.

The save palette field is set to the buffer size required to hold the palette contents.

Usage Notes

See “State Data” on page 2-11 for a description of states.

Attribute Entry Points

HSMARK - Set Marker Shape

Function

The HSMARK order defines the shape of the current marker symbol.

Entry Point Parameter Block

Byte	Content	Meaning
0	LEN	Length of following data (10 or 14)
2	BYTE	Cell width in PELs (cx)
3	BYTE	Cell height in PELs (cy)
4	BYTE	Flags Bit 7 0 - monochrome 1 - multicolor Bits 6-0 reserved - must be 0
5	BYTE	Reserved
6	WORD	Length of image definition in bytes i = CEIL(cx*cy/8)
8	ADDRESS	Address of marker image definition
12	ADDRESS	Address of marker color definition
. . 16		

Figure 2-42. HSMARK: Set Marker Shape

Description

The current marker symbol is set to the image specified by the order.

The marker image is a rectangular bit pattern, starting at the top left-hand corner with bit 7 of byte 0, and scanning from left to right, and from top to bottom. There are no padding bits between rows.

The marker color definition consists of a list of 1-byte color indices. There is a color index for every PEL of the image including background (0) PELs. Each color index is applied in turn to each PEL in the image definition.

If the monochrome flag is selected, the marker will be drawn with the current foreground color and mix on the current background color and mix. If the multicolor flag is selected, the marker will be drawn with the foreground colors extracted from the marker color definition and the background color set to the current background color and mix.

HSPATT - Set Pattern Shape

Function

The HSPATT order defines the shape of the current area-fill pattern symbol.

Entry Point Parameter Block

Byte	Content	Meaning
0	LEN	Length of following data (10 or 14)
2	BYTE	Cell width in PELs (cx)
3	BYTE	Cell height in PELs (cy)
4	BYTE	Flags Bit 7 0 - monochrome 1 - multicolor Bits 6-0 reserved - must be zero
5	BYTE	Reserved
6	WORD	Length of image definition in bytes i = CEIL(cx*cy/8) cx > 0 cy > 0
8	ADDRESS	Address of pattern image definition
12	ADDRESS	Address of pattern color definition
.		
.		
16		

Figure 2-43. HSPATT: Set Pattern Shape

Description

The current pattern symbol is set to the image specified by the order.

The pattern image is a rectangular bit pattern, starting at the top left-hand corner with bit 7 of byte 0, and scanning from left to right, and from top to bottom. There are no padding bits between rows.

The pattern color definition consists of a list of 1-byte color indices. There is a color index for every PEL of the pattern including background PELs. Each color index is applied in turn to each PEL in the image definition.

If the monochrome flag is selected, the pattern will be drawn with the foreground color and mix on the background color and mix. If the multicolor flag is selected, the pattern will be drawn with the foreground colors extracted from the pattern color definition and the background color set to the background color.

Usage Notes

When used for area fill, the pattern position within the area is controlled by the HSPATTO order.

To select solid fill, specify a pattern cell size of 1 x 1.

The maximum area-fill pattern size is 32 x 32 PELs.

HSPATTO - Set Pattern Reference Point

Function

The HSPATTO sets the reference point or origin for area-fill pattern symbols.

Entry Point Parameter Block

Byte	Content	Meaning
0	LEN	Length of following data (4)
2 . . 6	P0	Reference point

Figure 2-44. HSPATTO: Set Pattern Reference Point

Description

The pattern reference point is set to that specified in the order.

Usage Notes

The pattern reference point specifies where the top left corner of an instance of the pattern cell is located.

HSLT - Set Line Type

Function

The HSLT order sets the value of current line type to that specified in the order.

Entry Point Parameter Block

Byte	Content	Meaning
0	LEN	Length of following data (1 or 6)
2	LTYPE	Line type value X′00′ - Load user line type X′01′ - Dotted line X′02′ - Short dashed line X′03′ - Dash-dot line X′04′ - Double dotted line X′05′ - Long dashed line X′06′ - Dash-double-dot line X′07′ - Solid line X′08′ - Invisible line
3		Reserved
4 . . 8	ADDRESS	Address of user line-type definition

Figure 2-45. HSLT: Set Line Type

Description

Line type is set to the value specified in the order.

The user line-type data is a word byte count followed by a list of pairs of bytes. The first byte of each pair gives the number of screen space PELs on; the second byte gives the number of screen space PELs off.

Optimization

The adapter-defined line types are drawn faster than user-defined line types.

There is an optimum line pattern length for each adapter.

A maximum length of pattern run of 48 PELs (in user line definition) is allowed. The PEL-runs for the various line types are as follows:

- Dotted line - 1 on 2 off, repeat
- Short dashed line - 5 on 3 off, repeat
- Dash-dot line - 6 on 4 off, 2 on 4 off, repeat
- Double dotted line - 2 on 4 off, 2 on 8 off, repeat
- Long dash line - 9 on 3 off, repeat
- Dash-double-dot line - 8 on 4 off, 2 on 4 off, 2 on 4 off, repeat.

Usage Notes

The following orders inherit the line-pattern count from previous orders:

- HSCP, HCLINE, HCRLINE.

All other line orders, including HSLT, reset the line-pattern count.

HSLW - Set Line Width

Function

The HSLW order sets the value of the current line width.

Entry Point Parameter Block

Byte	Content	Meaning
0	LEN	Length of following data (1)
2 . . 3	WIDTH	Line-width value

Figure 2-46. HSLW: Set Line Width

Description

Line width is set to the value specified in the order.

Usage Notes

1. Only line widths of 1 and 3 are supported. The HSLW order with values of 0 and 1 selects 1-PEL-wide lines. Values greater than 1 select 3-PEL-wide lines.
2. Lines are thickened as follows:
    a. The lines are symmetrical about the theoretical line between the end points.
    b. Lines of width 3 are thickened by drawing an extra line to the left and right (Y-major) or below and above (X-major). This means that thick lines are not retraceable.
3. The line-width value is an unsigned integer (that is, there are no negative line widths).

HSCOL - Set Color

Function

The HSCOL order sets the value of the foreground color index to that specified in the order.

Entry Point Parameter Block

Byte	Content	Meaning
0	LEN	Length of following data (4)
2	DOUBLEWORD	Color Index
.		
.		
6		

Figure 2-47. HSCOL: Set Color

Description

The current foreground color index is set to the value specified in the order.

Usage Notes

See “Bit Plane Model” on page 2-2 for a description of the bit plane model.

The reserved color X′FFFFFFFF′ is used to enable multi-plane character sets (see “Programmable Character Definitions” on page 2-83).

Only the least significant 16 bits of the color index are used.

HSBCOL - Set Background Color

Function

The HSBCOL order sets the value of background color index to that specified in the order.

Entry Point Parameter Block

Byte	Content	Meaning
0	LEN	Length of following data (4)
2	DOUBLEWORD	Color Index
.		
.		
6		

Figure 2-48. HSBCOL: Set Background Color

Description

The current background color index is set to the value specified in the order.

HSMX - Set Mix

Function

This order sets the value of current mix to that specified in the order.

Entry Point Parameter Block

Byte	Content	Meaning
0	LEN	Length of following data (2)
2	MIX	Foreground mix value X'00' - Retain previous mix X'01' - OR X'02' - Overpaint X'03' - Reserved X'04' - Exclusive OR X'05' - Leave alone X'06' - Maximum X'07' - Minimum X'08' - Add: (screen + new) X'09' - Subtract: (screen-new) X'0A' - Subtract: (new-screen) X'0B' - Average: (screen + new)/2 X'0C' - Reserved : X'0F' - Reserved X'10' - zero X'11' - screen and new X'12' - screen and new X'13' - new X'14' - screen and new X'15' - screen X'16' - screen xor new X'17' - screen or new X'18' - screen and new X'19' - screen xor new X'1A' - screen X'1B' - screen or new X'1C' - new X'1D' - screen or new X'1E' - screen or new X'1F' - one X'20' - Reserved : X'FF' - Reserved
3 . 4	MIX	Background mix value same as foreground mix

Figure 2-49. HSMX: Set Mix

Description

The foreground and background mix are set to the values specified in the order.

Usage Notes

The arithmetic mix functions (maximum, minimum, add, subtract, average) operate on a contiguous set of graphics planes. Planes that are disabled for update take no part in the mix. If non-contiguous planes are enabled then the results are undefined.

The maximum mix yields the maximum binary value of the plane color index and the new color.

The minimum mix yields the minimum binary value of the plane color index and the new color.

The add mix yields the binary addition of the plane color index and the new color. If the result overflows then it is clipped to the maximum binary value.

The subtract mix yields the binary difference between the plane color index and the new color. If the result is less than 0, it is clipped to 0.

The average mix yields half of the the binary sum of the plane color index and the new color.

HSCMP - Set Color Comparison Register

Function

The HSCMP order sets the value of the Color Comparison register.

Entry Point Parameter Block

Byte	Content	Meaning
0	LEN	Length of following data (5)
2	DOUBLEWORD	Comparison color
6	BYTE	Logic function X'00' - True X'01' - Plane data > Comparison X'02' - Plane data = Comparison X'03' - Plane data < Comparison X'04' - False X'05' - Plane data ≥ Comparison X'06' - Plane data = Comparison X'07' - Plane data ≤ Comparison
.		
.		
7		

Figure 2-50. HSCMP: Set Color Comparison Register

Description

The HSCMP sets the Color Comparison register to the Comparison Color and the Comparison Logic.

Whenever the comparison between the plane data and the comparison register is TRUE then the existing bit plane data is left unmodified. When the test is FALSE the current foreground or background mix is used. Only planes enabled for update take part in the comparison.

Usage Notes

The Color Comparison register can be used to implement underpaint by setting the Color Comparison register to the color of the erased screen.

The Color Comparison register has uses in image processing.

This function does not apply to alphanumeric orders.

Programmable Character Definitions

Character definitions are either rectangular, across-the-plane images, or short stroke vectors. Each character set has a "character set definition block", in the following format:

Byte	Content	Meaning
0	BYTE	Reserved
1	TYPE	Type of character set: 0 - Image/multiplane image 1 - Reserved 2 - Reserved 3 - Short stroke vector
2	BYTE	Reserved
3-6	DOUBLEWORD	Reserved
7	BYTE	Cell width in PELs (cx)
8	BYTE	Cell height in PELs (cy)
9	BYTE	Reserved
10-11	WORD	Cell size = CEIL(cx*cy/8)
12-13	WORD	Flags: Bit 15 reserved - must be 0 Bit 14 0 - single plane 1 - multiplane Bit 13 0 - not proportionally spaced 1 - proportionally spaced Bits 12-0 reserved - must be 0
14-17	ADDRESS	Address of index table
18-21	ADDRESS	Address of character envelope table
22	BYTE	Initial code point
23	BYTE	Final code point
24-27	ADDRESS	Address of character definition table
28-29	WORD	Reserved
30-33	ADDRESS	Address of 2nd character definition table
34-35	WORD	Reserved
36-39	ADDRESS	Address of 3rd character definition table

Image Characters

Each character code point is used as an index into the index table. The index table starts at the Initial Code Point and ends at the Final Code Point. Any code point outside the range of defined characters defaults to the initial code point character.

The index table entry defines the location as follows:

Byte	
0–1	offset of character definition
2	

Figure 2-51. The Image Index Table Entry

The address of the character definition is calculated by adding the unsigned offset field to the address of the character definition table.

Each character definition starts on a byte boundary and occupies “m” bytes.

m = CEIL(cx*cy/8)

Each image character definition is in the form of an array of bits; each bit represents one (PEL). The PEL takes the current color where the corresponding bit is a binary 1, and the background color where the corresponding bit is a binary 0.

The image character definition starts at the top left-hand PEL of the character cell and finishes at the bottom right-hand PEL of the character cell.

pattern for row 1			pattern for row n	pad

Multiplane Image Characters

Multi-plane image characters consist of three monochrome images. (The second and third plane images are stored in the second and third character-definition tables.) They are drawn as follows:

- If the current character foreground color is not X′FFFFFFFF′ then:

  The three images are mixed with the OR logic function to create a single image. The image is drawn with the current foreground color and mix on the current background color and mix.

- If the current character foreground color is the reserved value X′FFFFFFFF′ then:

  For each PEL in the image character, a color index is obtained from the multi-plane text color index table (see “HXLATE - Assign Multiplane Text Color Index Table” on page 2-91). For a color index of 0, the background color and mix (non-arithmetic only) is used.

Short Stroke Vector Characters

The index table is used to identify the location of each character definition within the character definition block. Each character code point is used as an index into the index table. The index table starts at the Initial Code Point and ends at the Final Code Point.

The short-stroke vector definition must start at the bottom left-hand PEL of the character cell, and finishes one PEL to the right of the bottom right-hand PEL of the character cell.

The index table entry defines the location as follows:

Byte	Short stroke vectors
0–1	offset of character definition
2	

Figure 2-52. The Short Stroke Vector Index Table Entry

The address of the character definition is calculated by adding the unsigned offset field to the address of the character definition table.

Short-stroke vector-character definitions are in the form of an array of short drawing orders. Each order is 1 byte long.

The end of a short stroke definition is defined by a ′00′X.

```
d d d m   l l l l
```

Figure 2-53. Short Stroke Vector Definition

- ddd is the direction of the vector
  - ddd = 000 => 0 degrees (anti-clockwise from horizontal)
  - ddd = 001 => 45 degrees
  - ddd = 010 => 90 degrees
  - ddd = 011 => 135 degrees
  - ddd = 100 => 180 degrees
  - ddd = 101 => 225 degrees
  - ddd = 110 => 270 degrees
  - ddd = 111 => 315 degrees
- m is the draw/move flag
  - m = 0 => move
  - m = 1 => draw
- llll is the projection of the length of the line on the horizontal or vertical axis (depending on the direction ddd) in screen space PELs. The last PEL is always null. The binary value 0000 performs no operation. In draw mode, the binary value N writes N PELs and moves the drawing position 1 PEL further.

Character Envelope Table

The Character Envelope Table is used to provide proportional spacing and only needs to be supplied if proportional spacing has been selected.

Each character code point is used as an index into the Character Envelope Table. The Character Envelope Table starts at the Initial Code Point and ends at the Final Code Point.

Each entry in the Character Envelope Table defines a left margin (in screen space PELs) and a right margin (in screen space PELs).

byte	
0	left margin
1	right margin
2	

Figure 2-54. Character Envelope Table Entry

The left margin is subtracted from the horizontal character start position before the character is drawn; the right margin is subtracted from the horizontal character start position after the character is drawn.

For image fonts, the left and right margins represent areas of the character definition that are not drawn on the screen.

Character Cell Size and Spacing

For text character strings, the character cell size is taken from the Character Set Definition Block. In the case of proportionally-spaced characters, the width value “cx” from the definition block is decremented by a value from the envelope table.

In a string, characters following the first character are drawn immediately to the right of the preceding character.

For alphanumeric character blocks, the character cell size is taken from the display mode, as determined by the Set Mode (HSMODE) order.

Usage Notes

- The symbol sets selected should match the cell size selected, or the results will be unpredictable
- The 7 x 15 alpha cell size (HSMODE 3) must be used with short-stroke vector characters. It cannot be used with image characters.

Character Entry Points

HSCS - Set Character Set

Function

The HSCS order sets the current text character set to the character set specified by the order.

Entry Point Parameter Table

Byte	Content	Meaning
0	LEN	Length of following data (≥4)
2 . . 6	ADDRESS	Address of the character set definition block.

Figure 2-55. HSCS: Set Character Set

Description

The HSCS order changes the current text character set.

The address of the character set definition is a program address in the controlling environment. The controlling system must preserve the contents of the character set buffer until a subsequent HSCS order has been processed by the adapter.

The character set definition format is given in "Programmable Character Definitions" on page 2-83.

If the character set definition is changed in any way then the HSCS order must be re-issued.

HCHST - Text Character String at Given Position

Function

This order draws a character string at a given position.

Entry Point Parameter Table

Byte	Content	Meaning
0	LEN	Length of following data (≥4)
2	P0	Coordinate data of point at which the bottom left corner of the character string is placed.
6	STRING	List of code points in the string
.		
.		
6+s		

Figure 2-56. HCHST: Character String at Given Position

Description

This order draws a text character string at point P0.

The current position is changed to P0.

Each code point is 1 byte in length.

HCCHST - Text Character String at Current Position

Function

This order draws a character string at the current position.

Entry Point Parameter Block

Byte	Content	Meaning
0	LEN	Length of following data (≥0)
2	STRING	List of code points in the string
.		
.		
2+s		

Figure 2-57. HCCHST: Character String at Current Position

Description

This order draws a text character string at the current position.

The current position is unchanged.

Each code point is 1 byte in length.

HXLATE - Assign Multiplane Text Color Index Table

Function

The HXLATE order provides a color index translate table for use with multiplane text orders.

Entry Point Parameter Block

Byte	Content	Meaning
0	LEN	Length of following data (32)
2 . . 34	DOUBLEWORD	An 8 entry translate table

Figure 2-58. HXLATE: Assign Multiplane Color Index Table

Description

In processing the HCHST and HCCHST orders, with a multi-plane image character set defined, the color index used for each PEL of the character will be obtained from the table provided by this order. The combination of the PEL value in each of the three-plane definitions will be used to address the table.

The table address = ((pd1)2**0) + ((pd2)2**1) + ((pd3)2**2), where:

- pd1 = plane 1 PEL value (0 or 1)
- pd2 = plane 2 PEL value (0 or 1)
- pd3 = plane 3 PEL value (0 or 1)

Alphanumeric Orders

These orders give the ability to display alphanumerics that are constrained by character cells; characters are arranged in rows and columns, and have attributes associated with them. The Set Mode order (HSMODE) defines the current character cell size.

Each character is defined by a pattern of foreground and background bits and a set of character attributes. A single-byte code point is used to select the character shape from one of four symbol sets. The attributes consist of:

- Foreground color (one of 16)
- Background color (one of 16)
- Reverse video
- Underscore
- Overstrike
- Background transparent or opaque
- Font (one of four).

There is an alphanumeric cursor which marks one character cell. The following cursor options are supported:

- Hidden
- Normal (variable size)
- Left arrow
- Right arrow.

Note: If alpha characters are written into the same bit planes as graphics, it is the responsibility of the controlling environment to:

- Manage the alpha-graphic interaction
- Save background information to enable transparency to work.

Alphanumeric Entry Points

ABLOCKMFI - Write Character Block

Function

The ABLOCKMFI order writes a block of characters to the bit planes from a character buffer in PC storage.

Entry Point Parameter Block

Byte	Content	Meaning
0	LEN	Length of following data (9)
2	BYTE	Start column (0 to n)
3	BYTE	Start row (0 to n)
4	BYTE	Number of character cells across
5	BYTE	Number of character cells down
6	ADDRESS	Start address of character block
10	BYTE	Width of character buffer
.		
.		
11		

Figure 2-59. ABLOCKMFI: Write Character Block

Description

The character string located by the Start Address of Character Block is written to the display buffer, starting with the character cell at the Starting Column and Starting Row. Subsequent characters are placed in order of increasing column addresses, until the specified number of character cells across have been written. The row count then increments to the next row, and the start column is reset to the specified start column. The next character address in the character string is then calculated by adding the width of the character buffer (multiplied by 4), to the start address of the current line in the character buffer.

Character cells are addressed from the top left (0,0) with column addresses increasing to the right, and row addresses increasing downward.

Each character in the character string is represented by a 4-byte field:

```
X'0000'  ┌────────────────────────────────────┐
         │ Character code                     │
X'0001'  ├────────────────────────────────────┤
         │ Color attribute                    │
         │  Bits 7–4 background color         │
         │  Bits 3–0 foreground color         │
X'0002'  ├────────────────────────────────────┤
         │ Highlight attribute                │
         │  Bit 7 – underscore                │
         │  Bit 6 – reverse video             │
         │  Bit 5 – overstrike                │
         │  Bit 4 – opaque/transparent        │
         │          0 = background opaque     │
         │          1 = background            │
         │                transparent         │
         │  Bit 3 – spare                     │
         │  Bit 2 – spare                     │
         │  Bit 1 – font bit 0                │
         │  Bit 0 – font bit 1                │
X'0003'  ├────────────────────────────────────┤
         │ Reserved                           │
X'0004'  └────────────────────────────────────┘
```

Figure 2-60. The Adapter Character Representation (MFI)

Note: The font bits specify the font number to be used. Character sets are assigned to font numbers by the ASFONT order (see "ASFONT - Set Character Set" on page 2-101).

Optimization

The AERASE and ASCROLL orders should be used for erasing, scrolling, and inserting data on the screen to obtain the best performance.

Usage Notes

The scissor applies to both alphanumeric updates and graphics.

ABLOCKCGA - Write Character Block (CGA)

Function

The ABLOCKCGA order is similar to the ABLOCKMFI order except that it supports a 2-byte character attribute sequence.

Entry Point Parameter Block

Byte	Content	Meaning
0	LEN	Length of following data (10)
2	BYTE	Start column (0 to n)
3	BYTE	Start row (0 to n)
4	BYTE	Number of character cells across
5	BYTE	Number of character cells down
6	ADDRESS	Start address of character block
10	BYTE	Width of character buffer
11	BYTE	Highlight attribute for block
.		
.		
12		

Figure 2-61. ABLOCKCGA: Write Character Block (CGA)

Description

The character string located by the Start Address of Character Block is written to the display buffer, starting with the character cell at the Starting Column and Starting Row. Subsequent characters are placed in order of increasing column addresses, until the specified number of character cells across have been written. The row count then increments to the next row, and the start column is reset to the specified start column. The next character address in the character string is calculated by adding the width of the character buffer (multiplied by 2), to the start address of the current line in the character buffer.

Character cells are addressed from the top left (0,0), with column addresses increasing to the right, and row addresses increasing downward.

Each character in the character string is represented by a 2-byte field:

The highlight attribute byte in the parameter block is the same format as the highlight attributes in the ABLOCKMFI.

X'0000'	Character code
X'0001'	Color attribute Bits 7—4 background color Bits 3—0 foreground color

Figure 2-62. The Adapter Character Representation (CGA)

Optimization

The AERASE and ASCROLL orders should be used for erasing, scrolling, and inserting data on the screen to obtain the best performance.

Usage Notes

The scissor applies to both alphanumeric updates and graphics.

AERASE - Erase Rectangle

Function

The AERASE sets a rectangle of character cells to a background color.

Entry Point Parameter Block

Byte	Content	Meaning
0	LEN	Length of following data (5)
2	BYTE	Starting column (0 to n)
3	BYTE	Starting row (0 to m)
4	BYTE	Number of character cells across
5	BYTE	Number of character cells down
6	BYTE	Color Bits 7-4 background color Bits 3-0 - reserved = B′0000′
.		
.		
7		

Figure 2-63. AERASE: Erase Rectangle

Description

The rectangle whose top left-hand corner is given by the Starting Row and Starting Column and whose size (in character cells) is given by the numbers of character cells across and down, is set to the background color given by the color field.

ASCROLL - Scroll Rectangle

Function

The ASCROLL order copies a rectangle of character cells on the screen.

Entry Point Parameter Block

Byte	Content	Meaning
0	LEN	Length of following data (6)
2	BYTE	Starting column (0 to n) of source
3	BYTE	Starting row (0 to m) of source
4	BYTE	Number of character cells across
5	BYTE	Number of character cells down
6	BYTE	Starting column of destination
7	BYTE	Starting row of destination
.		
.		
8		

Figure 2-64. ASCROLL: Scroll Rectangle

Description

The rectangle whose top left-hand corner is given by the Starting Row and Starting Column of Source, and whose size (in character cells) is given by the numbers of character cells across and down, is copied to the rectangle whose top left hand corner is given by the Starting Row and Starting Column of the destination.

Both the origin and destination rectangles must be entirely on the screen. The effect of copying a rectangle to an overlapping rectangle is as if the source data were stored in a separate buffer.

ACURSOR - Set Cursor Position

Function

The ACURSOR order sets the alphanumeric cursor position.

Entry Point Parameter Block

Byte	Content	Meaning
0	LEN	Length of following data (2)
2	BYTE	Column (0 to n) of cursor
3	BYTE	Row (0 to m) of cursor
.		
.		
4		

Figure 2-65. ACURSOR: Set Cursor Position

Description

The cursor is removed from the previously marked character cell and drawn in the character cell defined by the row and column fields.

ASCUR - Set Cursor Shape

Function

The ASCUR order sets the alphanumeric cursor shape.

Entry Point Parameter Block

Byte	Content	Meaning
0	LEN	Length of following data (3)
2	BYTE	Cursor start line (0 to n)
3	BYTE	Cursor stop line (0 to n)
4	BYTE	Attribute 00 - normal 01 - hidden 02 - left arrow 03 - right arrow
.		
.		
5		

Figure 2-66. ASCUR: Set Cursor Shape

Note: 0 is the top of the character cell

Description

The cursor shape and attributes are set for subsequent cursor operations.

Usage Notes

After a mode change the cursor is hidden. The cursor is drawn with an XOR mix and all interaction between alpha and the alpha cursor is handled by the adapter. However, any graphics corruption of characters and/or the cursor must be handled by the controlling environment.

If the start line is greater than the stop line, no cursor is drawn. The start and stop lines should be within the bounds of the character cell, and between the top + 2 and the bottom if it is an arrow cursor.

If the start line is 'FF'X, then the current start-line/stop-line definition is used (that is, inherit current size).

ASFONT - Set Character Set

Function

The ASFONT order sets one of the four alphanumeric character sets.

Entry Point Parameter Block

Byte	**Content**	**Meaning**
0	LEN	Length of following data (6)
2	BYTE	Font number (0..3)
3	BYTE	Reserved
4	ADDRESS	Address of the character set definition block
.		
.		
8		

Figure 2-67. ASFONT: Set Character Set

Description

The character set located at the address of the character set definition block is used for all subsequent alphanumeric characters written with the specified font.

The character set definition format is in "Programmable Character Definitions" on page 2-83. The character set must be a single- plane image character set or a short stroke vector set.

AXLATE - Assign Alpha Attribute Color Index Table

Function

The AXLATE order provides an attribute to color index translate table.

Entry Point Parameter Block

Byte	Content	Meaning
0	LEN	Length of following data (128)
2	DOUBLEWORD	Two 16-entry translate tables. The first 16 entries are foreground and the second 16 are background.

Figure 2-68. AXLATE: Assign Alpha Color Index Table

Description

Bits 3-0 of the character attribute are used to address the background color index table when processing the alphanumeric orders. The value obtained from the table is used for the character background. Bits 7-4 of the character attribute are used to address the foreground color index table, and the value obtained is used for the character.

Chapter 3. Shippable Code, Fonts, and Interface

Shippable Code Items

The Adapter Interface Code diskette contains the installation instructions, the procedures, and the adapter interface module which contains a set of entry points (described in Chapter 2, "Programming Considerations").

The Adapter Interface Code diskette contains:

8PLANE	Directory for adapters with the memory expansion feature
4PLANE	Directory for adapters without the memory expansion feature
HDIMSGS	Directory containing installation program messages
READ.ME	Explanation file
INSTALL.COM	Installation program
HDIDEMO.EXE	A demonstration program to check the Adapter Interface
HDIDEMO.C	Source for the demonstration program
IBMAFI.H	Include files, used by the demonstration program, which provide an IBM 1.0 'C' macro 'binding' to the Interface
AFI_DATA.H	Include files, used by the demonstration program, which provide an IBM 1.0 'C' macro 'binding' to the Interface
CALLAFI.ASM	Source of MASM 'glue' functions, allowing IBM 1.0 'C' to call the Interface Entry Points
CALLAFI.OBJ	OBJect code from CALLAFI.ASM
NOAFI.MSG	Message file used by the demonstration program
NOHOPEN.MSG	Message file used by the demonstration program
AFIINTRO.MSG	Message file used by the demonstration program
STAN1220.FNT	12 x 20 size standard font
STAN0814.FNT	8 x 14 size standard font
STAN0715.FNT	7 x 15 size standard font

All the files needed to load the adapter interface are contained in a directory called \HDIPCDOS. There is a READ.ME file, which describes this directory.

The AUTOEXEC.BAT can be changed (or built) to load the adapter interface automatically on system startup, by inserting the following command:

```
\HDIPCDOS\HDILOAD.EXE
```

Alternatively, the interface can be installed (when required) by the same command.

To install this interface onto another disk or diskette, run the INSTALL.COM command as follows:

```
<drive:path> INSTALL source_drive target_drive
```

For example, A:INSTALL A: C: will install the interface from the diskette in the A-drive to the fixed-disk C drive.

Font File Format

There are three fonts supplied with the IBM Adapter Interface Code diskette; each font contains five code pages:

- **437:** US/English
- **850:** Multilingual
- **860:** Portuguese
- **863:** Canadian/French
- **865:** Nordic.

The Adapter Interface Code installation procedure allows you to select your *Default Code Page ID* (for example, 437 for US/English) and *Alternate Default Code Page ID* (for example, 850 for Multilingual) for use by applications using these supplied fonts.

The code page ID (for example, 437) is an ASCII string followed by a null.

Each font file contains a header (see the following table) that is updated by the installation procedure.

Each code page has a *Character Set Definition Block* (see "Programmable Character Definitions" on page 2-83). The five address fields within the character set definition block are offsets within the file; *you must add the address at which the font is loaded to give the correct values.*

Byte	**Content**	**Meaning**
0	WORD	Number of code pages within this file
2	WORD	Default code page index into table below (range 0 to 4)
4	WORD	Alternate default code page index into table below (range 0 to 4)
6	STRING	4-byte code page ID (437) of the first Code Page in the file
10	WORD	Offset within the file to the character set definition block of the first code page in the file
12	STRING	4-byte code page ID (850) of the second code page in the file
16	WORD	Offset within the file to the character set definition block of the second code page in the file
18	STRING	4-byte code page ID (860) of the third code page in the file
22	WORD	Offset within the file to the character set definition block of the third code page in the file
24	STRING	4-byte code page ID (863) of the fourth code page in the file
28	WORD	Offset within the file to the character set definition block of the fourth code page in the file
30	STRING	4-byte code page ID (865) of the fifth code page in the file
34	WORD	Offset within the file to the character set definition block of the fifth code page in the file

Characteristics of the Adapter

The Display Adapter 8514/A (see Figure 3-1) has two input/output connectors:

- A 15 way D-type connector to the monitor (see "Physical Interface" on page 3-5 for details)
- A 140-way edge PC connector

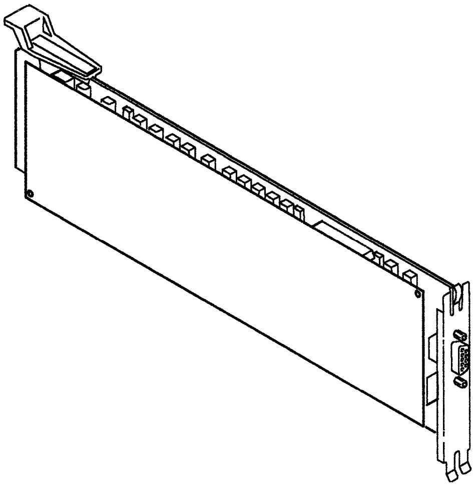


Figure 3-1. Display Adapter 8514/A

Card Power Requirements

The following are the nominal and maximum power requirements:

+ 5v	Nominal: 11 watts; Maximum: 16 watts
- 12v	Nominal: 0.15 watts; Maximum: 0.22 watts

Logic Voltage Tolerances + 5 volts dc (±5 %), -12 volts dc (±10%)

Physical Interface

The connection to the monitor cable is by a 15-way female AMP ®[1] connector. The connector pins are allocated as follows:

Pin Number	Allocation
1	Red video
2	Green video
3	Blue video
4	Monitor ID bit 2
5	Ground
6	Red return (Ground)
7	Green return (Ground)
8	Blue return (Ground)
9	Key (blanked hole)
10	Sync return (Ground)
11	Monitor ID bit 0
12	Monitor ID bit 1
13	Horizontal sync
14	Vertical sync
15	Not used (reserved)

Figure 3-2. Connector Pin Allocations

---

[1] Registered trademark of the Amphenol Corp.

Electrical Interface

Monochrome monitors use the green video signal, and make no connection to the red and blue video signals.

Video Signals

The following characteristics apply when each video signal reaches an external resistance of 75 ohms to ground. Inside the adapter these signals are terminated to ground through 150-ohm resistors. The red, green, and blue returns are connected to ground by the adapter.

The black level output is 0 volts. The nominal full intensity level output is + 0.70 volts.

The maximum rise time (10% to 90%) is 8 nanoseconds. The maximum full scale settling time (1% to 99%) is 20 nanoseconds.

The minimum time between transitions is 22 nanoseconds.

Assuming the external 75-ohm resistor has a tolerance of ±2%, the full scale accuracy of the output voltage is + 6.8%, - 6.1%. The full scale tracking between the three video signals is ±2.3%. These tolerances include temperature variations.

Sync Signals

The sync signals are normal 'F' TTL levels. These outputs are capable of sinking 20 mA at 0.5 volts and sourcing 1.0 mA at 2.7 volts. The position, duration and polarity of the sync pulses are programmable. The sync return is connected to ground by the adapter.

Monitor ID Signals

These inputs to the adapter are each pulled up to +5 volts by 4.7 Kohm resistors in the adapter. Normal TTL input thresholds apply to these signals. To achieve a valid down level the driver of these signals must be able to sink 2.8 mA.

The adapter provides the ability for software to read the state of these signals. Monitors will either leave these signals unconnected or connect them to sync return (ground) in various combinations in order to indicate the type of monitor connected.

The monitor ID bits are left unconnected to the adapter card, or are connected to 0 volts by each display as follows:

Display Type	**ID Bit:**	**2**	**1**	**0**
IBM Monochrome Display 8503	-	N/C	0V	N/C
IBM Color Display 8513	-	N/C	N/C	0V
IBM Color Display 8512	-	N/C	N/C	0V
IBM Color Display 8514	-	0V	N/C	0V

N/C = No Connection

Sync Parameters

The adapter supports four monitor modes of operation, known as monitor modes 1, 2, 3, and 4, which are selected by sync pulse polarity. (A positive sync signal polarity is one where a TTL up level signals flyback.) The characteristics of these four modes are as follows:

Monitor Mode	**1**	**2**	**3**	**4**
Horizontal sync polarity	+	−	−	+
Vertical sync polarity	−	+	−	+
Total PELs per line	900	900	800	1264
Active PELs per line	738	738	656	1024
Line blanking time (PELs)	162	162	144	240
Line sync pulse width (PELs)	108	108	96	176
Line front porch width (PELs)	9	9	8	8
Line back porch width (PELs)	45	45	40	56
Total lines per frame	449	449	525	817
Active lines per frame	362	414	496	768
Fields per frame	1	1	1	2
Field blanking time (E/O lines)	87	35	29	25/24
Field sync pulse width (lines)	2	2	2	4
Front field porch width (E/O lines)	32	6	3	0.5/0
Field back porch width (E/O lines)	53	27	24	20.5/20
PEL time (ns)	35.31	35.31	39.72	22.27
Total line time (us)	31.78	31.78	31.78	28.15
Total frame time (ms)	14.27	14.27	16.68	23.00
Frame rate (Hz)	70.08	70.08	59.94	43.48

Memory Addresses

The Display Adapter 8514/A uses the following memory addresses:

Memory Addresses

C6800-C7FFF
CA000-CA7FF

Input/Output Addresses

The Display Adapter 8514/A uses the the following I/O addresses:

I/O Addresses

XX3C6-9
XX7C6-9
XXBC6-9
XXFC6-9
XX2E8-9
XX6E8-9
XXAE8-9
XXEE8-9
XX2EA-D
XX6EA-D
XXAEA-D
XXEEA-D
XX100
XX101
XX102

Appendix A. Default Palettes

The advanced function default palette provides two banks of 16 colors; the palette is loaded with the 16 values shown in Figure A-1.

Planes 4 - 7

Planes 0 - 3

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

Figure A-1. Default Palette Values

The meaning of the values depends on whether there is a color or monochrome display attached to the adapter:

For a color display, the values 0 - F are treated as having the bit significance I, R, G, or B. Hence, the displayed colors are:

I	**R**	**G**	**B**	**Color**
0	0	0	0	Black
0	0	0	1	Blue
0	0	1	0	Green
0	0	1	1	Cyan
0	1	0	0	Red
0	1	0	1	Magenta
0	1	1	0	Brown
0	1	1	1	White
1	0	0	0	Gray
1	0	0	1	Light blue
1	0	1	0	Light green
1	0	1	1	Light cyan
1	1	0	0	Light red
1	1	0	1	Light magenta
1	1	1	0	Yellow
1	1	1	1	White (high intensity)

For a monochrome display, the palette values are defined so that color applications have good gray-level discrimination when shown on a monochrome display. The following table defines the relationship between the palette value and the display intensity (with the display intensity on a scale of 0 - 63):

Palette Value	**Display Intensity**
0	0
1	5
2	17
3	28
4	8
5	11
6	20
7	40
8	14
9	24
A	45
B	50
C	32
D	36
E	56
F	63

Index