IBM

Personal Computer Hardware Reference Library

IBM RT PC Hardware Technical Reference

Volume III

84X0873

IBM

Personal Computer
Hardware Reference
Library

IBM RT PC Hardware Technical Reference

Volume III

Second Edition (September 1986)

Changes are made periodically to the information herein; these changes will be incorporated in new editions of this publication.

Products are not stocked at the address given below. Requests for copies of this product and for technical information about the system should be made to your authorized IBM RT PC dealer.

A reader's comment form is provided at the back of this publication. If the form has been removed, address comments to IBM Corporation, Department 997, 11400 Burnet Road, Austin, Texas 78758. IBM may use or distribute whatever information you supply in any way it believes appropriate without incurring any obligation to you.

About This Book

Purpose

The options and adapters are the second part of the IBM RT PC Hardware Technical Reference manual. They are to be used in conjunction with the IBM RT PC Hardware Technical Reference, Volume I.

Audience

The information in this manual is for reference. It is intended for hardware and program designers, programmers, engineers, and anyone else who needs to understand the design and operation of the IBM RT PC Product Family.

How to Use This Book

This manual is modular in format, with each module providing information about a specific option or adapter available for the IBM RT PC family of products. Modules having a large amount of text contain indexes.

The modules are grouped by type of device. To find a specific module:

1. Locate the full length hard tab with the type of device (Displays, Communications, Storage Devices, etc.) printed on it that describes the option or adapter you need information about.
2. Open the manual to that section.
3. Leaf through that section to find the proper module.

Personal Computer
Hardware Reference
Library

IBM Advanced Color Graphics Display

IBM ADVANCED COLOR GRAPHICS DISPLAY

Contents

Description

The Advanced Color Graphics Display is a high resolution color display that supports an addressable format of 720 dots horizontally by 512 scan lines vertically. The display screen is interlaced at a refresh rate of 46/92 Hz.

The Advanced Color Graphics Display has two attached cables. One cable provides the direct-drive signal interface to the Advanced Color Graphics Display Adapter. The second cable provides ac power to the display. The display operates on 120 Vac, 50/60 Hz or 220 Vac, 50/60 Hz depending on the model.

The display uses a 14-inch (diagonal) shadow mask color cathode ray tube (CRT). The CRT and associated analog circuits of the display are packaged in an enclosure that allows the display to sit either on top of the IBM 6151, on a nearby tabletop or desk with the IBM 6150. The display enclosure includes a tilt and swivel base, which allows the display screen to be adjusted by the user for the best viewing conditions. The display an operator control that provides brightness control function and a power on and off switch.

Operating Characteristics

Operating characteristics of the display are as follows:

Screen

- Etched surface for reduced glare
- 720 dots horizontal by 512 scan lines vertical.

Video Signal

- Two-level (on and off) video for 6 video lines
- Maximum bandwidth of approx. 25.7 Mhz
- Compatible with standard TTL driver.

Horizontal Drive

- Free-running horizontal oscillator
- Normally low, positive going TTL pulse
- Nominal horizontal frequency of 24.68 Khz
- Retrace blanking time of 8.0 usec.

Vertical Drive

- Free-running vertical oscillator
- Normally low, positive going TTL pulse
- Nominal vertical frequency of 92 Hz
- Nominal frame rate of 46 Hz.
- Retrace blanking time of 527.0 usec.

Operator Control

- Brightness control adjusts brightness of displayed image. (Clockwise rotation of control increases brightness.)
- Power on and off switch and light.

Size

- Width 372 mm (14.6 in.)
- Depth 400 mm (15.7 in.)
- Height 360 mm (14.2 in.)

Weight

13.3 Kg (29.26 pounds)

Power Cable

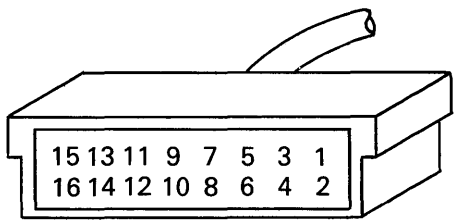
Length 2.8 m (9.19 ft.)

Signal Cable

Length 2.5 m (8.20 ft.)

Connector Specifications

Pin	Function
1	Signal ground for vertical sync
2	Vertical sync
3	R1 signal ground
4	Low order red bit (R1)
5	R2 signal ground
6	High order red bit (R2)
7	G1 signal ground
8	Low order green bit (G1)
9	G2 signal ground
10	High order green bit (G2)
11	B1 signal ground
12	Low order blue bit (B1)
13	B2 signal ground
14	High order blue bit (B2)
15	Signal ground for horizontal sync
16	Horizontal sync



IBM

Personal Computer
Hardware Reference
Library

Advanced Monochrome Graphics Display

Contents

Description

The Advanced Monochrome Graphics Display is a high resolution monochrome display which supports an addressable format of 720 dots horizontally by 512 scan lines vertically. The display screen is interlaced at a refresh rate of 46/92 Hz.

The Advanced Monochrome Graphics Display attaches to the RT PC system unit through two cables approximately 2.5 meters (8.25 feet) in length. One cable provides the direct-drive signal interface to the RT PC Advanced Monochrome Graphics Display Adapter. The second cable provides ac power to the display from the system unit. This arrangement allows the system unit power switch to also control the power to the display and reduces the number of electrical wall outlets required to power the system. The display operates on 120 vac, 60 Hz power line voltage.

The display uses a 12-inch (diagonal) high contrast monochrome CRT. The CRT and associated analog circuits of the display are packaged in an enclosure that allows the display to sit either on top of the Model 10 system unit or on a nearby tabletop or desk with other RT PC system units. The display enclosure includes a tilt and swivel base which allows the display screen to be adjusted by the user for the best viewing conditions. The display has only one operator control which provides two functions: brightness control and raster switch (used only to verify operation of the display).

Operating Characteristics

Operating characteristics of the display are as follows:

Screen

- Etched surface for reduced glare
- 720 dots horizontal by 512 scan lines vertical.

Video Signal

- Two-level (on and off) video
- Maximum bandwidth of approx. 25.7 Mhz
- Compatible with standard TTL driver.

Horizontal Drive

- Free-running horizontal oscillator
- Normally low, positive going TTL pulse
- Nominal horizontal frequency of 24.68 Khz
- Retrace blanking time of 8.0 usec.

Vertical Drive

- Free-running vertical oscillator
- Normally low, positive going TTL pulse
- Nominal vertical frequency of 92 Hz
- Nominal frame rate of 46 Hz.

Operator Control

- Brightness control adjusts brightness of displayed image. (Clockwise rotation of control increases brightness)
- Raster switch (diagnostic aid) produces visible raster independent of video input. (Raster switch function occurs at full clockwise rotation of brightness control)

Size

Width 327 mm (12.9 in.)

Depth 331 mm (13.0 in.)

Height 266 mm (10.5 in.) without pedestal
305 mm (12.0 in.) with pedestal

Weight

8.5 Kg (18.74 pounds)

Power Cable

Length 2.5 m (8.25 ft.)

Signal Cable

Length 2.5 m (8.25 ft.)

Connector Specifications

Pin	Function
1	Signal ground for vertical sync
2	Vertical sync
3	Signal ground
4	Reserved
5	Signal ground
6	Reserved
7	Signal ground for video
8	Video
9	Signal ground
10	Reserved
11	Signal ground
12	Reserved
13	Signal ground
14	Reserved
15	Signal ground for horizontal sync
16	Horizontal sync

IBM

Personal Computer Hardware Reference Library

Enhanced Color Display

Contents

Description

The IBM Enhanced Color Display is an advanced color display capable of operating in two separate modes. Mode 1 is a 16 color 640 by 200 overscan mode with a horizontal scan frequency of 15.75 kHz. Mode 2 is a 64 color 640 by 350 mode with a horizontal scan frequency of 21.8 kHz. Both modes are non-interlaced. The monitor determines which mode to operate in by decoding the vertical sync polarity.

The IBM Enhanced Color Display attaches to the system unit by a signal cable that is approximately 3.5 feet (1.07 meters) in length. This signal cable provides a direct-drive interface from the IBM Personal Computer.

A second cable provides ac power to the display from a standard wall outlet. The display has its own power control and indicator. Three models are provided. Model 001 is for northern hemisphere operation and operates on 120 volts 50/60 Hz. Model 002 is for northern hemisphere operation and operates on 220/240 volts 50/60 Hz. Model 003 is for southern hemisphere operation and operates on 220/240 volts 50/60 Hz.

The display has a 13-inch, high-contrast CRT. The CRT and analog circuits are packaged in an enclosure so the display may sit either on top of the system unit or on a nearby tabletop or desk. Front panel controls and indicators include: Power-On control, Power-On indicator, Brightness and Contrast controls. Additional controls on the rear of the display are: Vertical Size 1 and Vertical Size 2. There are two service controls on the rear of the unit, black level adjustment and contrast default value adjustment.

Operating Characteristics

Screen

- Etched anti-glare screen
- 0.31mm dot mask
- Displays 16 or 64 colors depending on the mode selected

User Controls

- Brightness control affects the contribution of all input bits by controlling the gain of the video stages. The display contains a protection circuit which may overide this control.
- Contrast control affects the contribution of the least significant bits only. When pushed in, the contrast control is rendered inoperative and contrast is determined by the setting of the contrast default value adjustment on the rear of the display. Pulling the contrast control knob out engages the front contrast control.
- V. Size 1 control controls the vertical size of the screen in mode 1.
- V. Size 2 control controls the vertical size of the screen in mode 2.

Service Controls

- Black level adjust control is adjusted to make the raster lines just disappear when black input signal is supplied.
- Contrast default value control is used to set the contrast value when the front contrast control is pushed in. Normally adjust for best brown color.

Vertical Sync

- Uses polarity of Vertical Sync signal to automatically select Mode 1 or Mode 2 operation. Mode 1 is selected by a normally low positive going TTL pulse. Mode 2 is selected by a normally high negative going TTL pulse.
- Screen may be refreshed from 50 to 60 Hz. At 60 Hz there are either 200 or 350 vertical lines of resolution depending on the mode selected.
- 700 μsec retrace time

Horizontal Sync

- Normally low, positive going TTL pulse
- In Mode 1, 15.75 kHz.
- In Mode 2, 21.8 kHz.
- 6 μsec retrace time

When operating in Mode 1, the display maps the 4 input bits into 16 of the possible 64 colors as shown in the following chart.

I	R	G	B	Color	R r	G g	B b
0	0	0	0	Black	0 0	0 0	0 0
0	0	0	1	Blue	0 0	0 0	1 0
0	0	1	0	Green	0 0	1 0	0 0
0	0	1	1	Cyan	0 0	1 0	1 0
0	1	0	0	Red	1 0	0 0	0 0
0	1	0	1	Magenta	1 0	0 0	1 0
0	1	1	0	Brown	1 0	0 1	0 0
0	1	1	1	Gray 1	1 0	1 0	1 0
1	0	0	0	Gray 2	0 1	0 1	0 1
1	0	0	1	Light Blue	0 1	0 1	1 1
1	0	1	0	Light Green	0 1	1 1	0 1
1	0	1	1	Light Cyan	0 1	1 1	1 1
1	1	0	0	Light Red	1 1	0 1	0 1
1	1	0	1	Light Magenta	1 1	0 1	1 1
1	1	1	0	Light Yellow	1 1	1 1	0 1
1	1	1	1	White	1 1	1 1	1 1

Note: The R G and B are the most significant bits. The r g and b are the least significant bits.

Specifications

Size:

Length - 15.4 in (392 mm)

Depth - 15.6 in (407 mm)

Height - 11.7 in (297 mm)

Weight:

32 lbs

Heat Output:

300 BTU/hr

Power Cable:

Length - 6 ft (1.83 m)

Size - 18 AWG

Signal Cable:

Length - 3.5 ft (1.07 m)

Connector Information

The signals that are on the pins vary with the driver card being used and the mode in which it is operating. All signals are expected to be TTL levels supplied by totem pole drivers.

Pin	Mode 1 (16 Color)	Mode 2 (64 Color)
1	Shield Gnd	Ground
2	Signal Gnd	r
3	Red	R
4	Green	G
5	Blue	B
6	Intensity	g
7	Unused	b
8	Horiz Sync	Horiz Sync
9	Vert Sync	Vert Sync

Note: The R G and B are the most significant bits. The r g and b are the least significant bits.

IBM

Personal Computer
Hardware Reference
Library

Extended Monochrome Graphics Display

Contents

Description

The Extended Monochrome Graphics Display is a high resolution monochrome display which supports an addressable format of 1024 dots horizontally by 768 scan lines vertically. The display screen is refreshed at a rate of 60 Hz.

The Extended Monochrome Graphics Display has two attached cables. One cable provides the direct-drive signal interface to the Extended Monochrome Graphics Display Adapter. The second cable provides ac power to the display. The display operates on 120 vac, 50/60 Hz or 220 vac, 50/60 Hz depending on the model.

The display uses a 15-inch (diagonal) high contrast monochrome cathode ray tube (CRT). The CRT and associated analog circuits of the display are packaged in an enclosure that allows the display to sit either on top of the IBM 6151 system unit or on a nearby tabletop or desk with other RT PC system units. The display enclosure includes a tilt and swivel base which allows the display screen to be adjusted by the user for the best viewing conditions. The display has two operator controls. One operator control is used for contrast. The other control is used for brightness and the power on and off switch.

Operating Characteristics

Operating characteristics of the display are as follows:

Screen

- Etched surface for reduced glare
- 1024 dots horizontal by 768 scan lines vertical.

Video Signal

- Two-level (on and off) video
- Maximum bandwidth of approx. 70 Mhz
- Compatible with standard TTL driver.

Horizontal Drive

- Free-running horizontal oscillator
- Normally low, positive going TTL pulse
- Nominal horizontal frequency of 47.52 Khz
- Retrace blanking time of 4.21 usec.

Vertical Drive

- Free-running vertical oscillator
- Normally high, negative going TTL pulse
- Nominal vertical frequency of 60 Hz
- Retrace blanking time of 505.0 usec.

Operator Control

- Brightness control adjusts the overall monitor screen including background raster. Adjust the control so that the background raster disappears.
- Contrast control adjusts the brightness of the displayed image without affecting the background. (Clockwise rotation of control increases brightness.)
- Power On and Off switch and light.

Size

- Width 372 mm (14.6 in.)
- Depth 400 mm (17.7 in.)
- Height 360 mm (14.2 in.)

Weight

13.75 Kg (30.25 pounds)

Power Cable

Length 2.8 m (9.19 ft.)

Signal Cable

Length 2.5 m (8.2 ft.)

Connector Specifications

Pin	Function
1	Gnd – Vertical sync
2	Vertical sync
3	Reserved
4	Reserved
5	Gnd – Video
6	Video
7	Gnd – Video
8	Video
9	Gnd – Video
10	Video
11	Gnd – Video
12	Video
13	Gnd – Video
14	Video
15	Gnd – Horizontal sync
16	Horizontal sync

IBM

Personal Computer
Hardware Reference
Library

Monochrome Display

Contents

Description

The high resolution IBM Monochrome Display connects to the system unit through two cables. One cable is a signal cable from the display adapter to the display, and the other provides power to the display from the system unit. This arrangement eliminates the need for a wall outlet and allows the system-unit Power switch to control power to the display. The display unit has a 28.3 cm (11.5 in.) diagonal, 90° deflection cathode ray tube (CRT). The display may be placed on the system unit or on a nearby table or desk. Brightness and contrast controls are on the front surface and are easily accessible to the operator.

The characteristics of the display are as follows:

- Screen
  - High-persistence, green phosphor (P39).
  - Etched surface to reduce glare.
  - Presentation of 80 characters wide by 25 rows deep.
  - Characters are defined in a 14 PEL-high by 9 PEL-wide matrix.
- Video Signal
  - Maximum bandwidth of 16.257 MHz at -3dB
- Vertical Drive
  - Screen refreshed at 50 Hz with 350 lines of vertical resolution and 720 lines of horizontal resolution
- Horizontal Drive
  - Positive level, TTL-compatibility, at a frequency of 18.432 kHz

2 Monochrome Display

Specifications

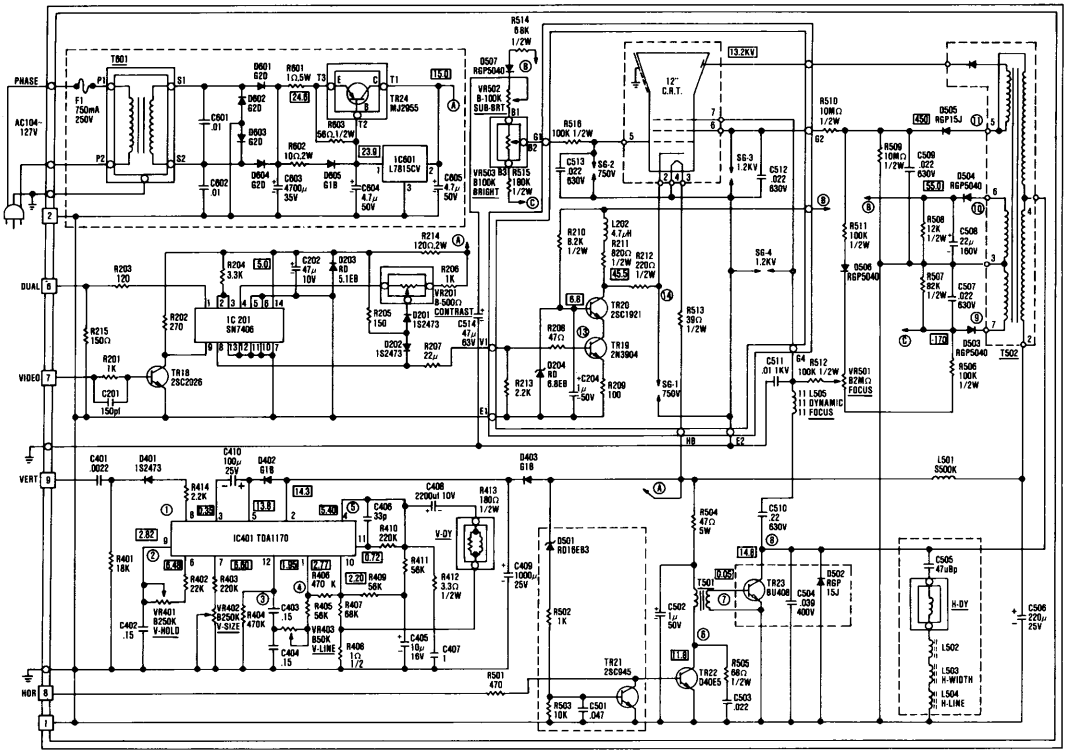
Size	
Height	280 mm (11 in.)
Length	380 mm (14.9 in.)
Depth	350 mm (13.7 in.)
Weight	7.9 kg (17.3 lb)
Heat Output	325 BTU/hr
Power Cable	
Length	0.914 m (3 ft)
Size	18 AWG
Signal Cable	
Length	1.22 m (4 ft)
Size	22 AWG

Physical Specifications

4 Monochrome Display

Logic Diagrams

The IBM Monochrome Display has two models: a 110-Vac model and a 220/240-Vac model. A logic diagram for each follows.



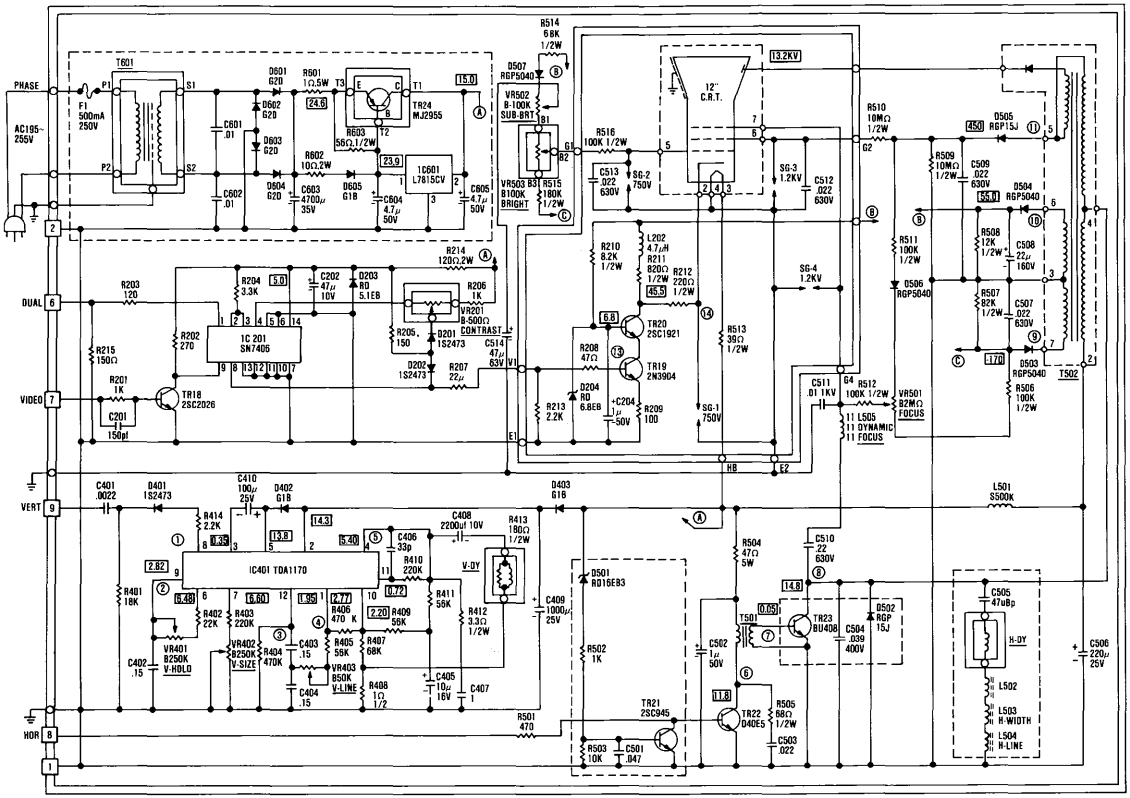
110Vac Monochrome Display (Sheet 1 of 1)

DANGER

HAZARDOUS VOLTAGES UP TO 450 VOLTS EXIST ON THE PRINTED CIRCUIT BOARDS

NOTES:
1. RESISTOR VALUES ARE IN (OHM) Ω K = 1000Ω M = 1,000,000Ω
2. ALL RESISTOR ARE 1/4W EXCEPT WHERE OTHERWISE INDICATED.
3. ALL CAPACITORS ARE 50V EXCEPT WHERE OTHERWISE INDICATED.
4. CAPACITORS VALUES ARE μF UNLESS OTHERWISE INDICATED μ = μF = 10^{-6}.
5. AC WIRING INFORMATION
   PHASE = BLACK/BROWN WIRE
   NEUTRAL = WHITE/BLUE WIRE
   GROUND = GREEN AND YELLOW WIRE

IMPORTANT: THE PHASE WIRE MUST GO TO THE FUSED SIDE OF TRANSFORMER.



220/240Vac Monochrome Display (Sheet 1 of 1)

DANGER

HAZARDOUS VOLTAGES UP TO 450 VOLTS EXIST ON THE PRINTED CIRCUIT BOARDS

NOTES:
1. RESISTOR VALUES ARE IN (OHM) Ω K = 1000Ω M = 1,000,000Ω
2. ALL RESISTOR ARE 1/4W EXCEPT WHERE OTHERWISE INDICATED.
3. ALL CAPACITORS ARE 50V EXCEPT WHERE OTHERWISE INDICATED.
4. CAPACITORS VALUES ARE μF UNLESS OTHERWISE INDICATED μ = μF = 10^{-6}.
5. AC WIRING INFORMATION
   PHASE = BLACK/BROWN WIRE
   NEUTRAL = WHITE/BLUE WIRE
   GROUND = GREEN AND YELLOW WIRE

IMPORTANT: THE PHASE WIRE MUST GO TO THE FUSED SIDE OF TRANSFORMER.

8 Monochrome Display

IBM

Personal Computer Hardware Reference Library

IBM 5080 Peripheral Adapter

Contents

Description

The IBM 5080 Peripheral Adapter provides three serial output ports on a 4.25- by 13.12-inch board that plugs into one I/O position. The adapter system control signals and voltage requirements are provided through a 2- by 31-position and a 2- by 18-position tab on the bottom of the adapter.

Up to four adapters may be used in one RT PC system. A DIP switch on the adapter is used to assign the adapter I/O address range. The port I/O address assignments are contained in the adapter's I/O address range.

The adapter is fully programmable and supports asynchronous communications only. It adds and removes start bits, stop bits, and parity bits. A programmable baud-rate generator allows operation from 50 bps to 19200 bps. Five-, 6-, 7- or 8-bit characters with 1, 1-1/2, or 2 stop bits are supported. A priority interrupt system controls transmit, receive, error, line status, and data set interrupts.

Three 10-pin male connectors on the adapter provide external access to the three ports.

The heart of the adapter is an NS16450 LSI chip or a functional equivalent. Features in addition to those listed above include:

Note: The NS16450 is functionally equivalent to all INS8250.

- Full double buffering that eliminates the need for precise synchronization
- Independent receiver clock input
- False start bit detection
- Line-break generation and detection.

All communications protocol is a function of the system microcode that must be loaded before the adapter is operational. All pacing of the interface and control signal status must be handled by the system software. Figure 1 on page 2 is a block diagram of the IBM 5080 Peripheral Adapter.

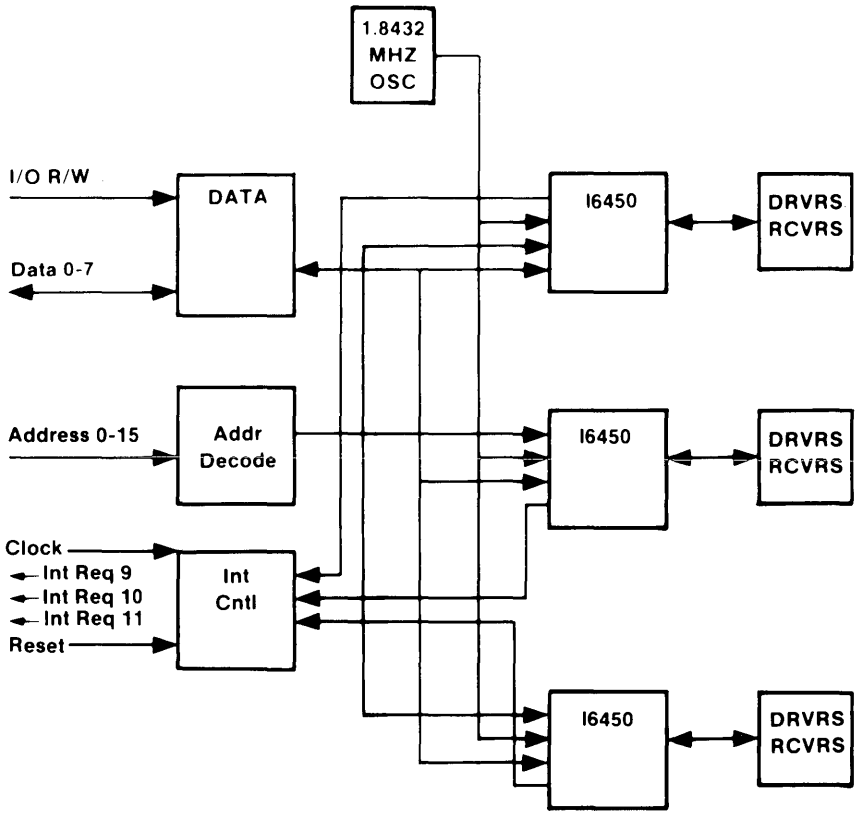


Figure 1. IBM 5080 Peripheral Adapter Block Diagram

IBM 5080 Peripheral Adapter Switch Settings

The IBM 5080 Peripheral Adapter switch settings select the interrupt level and the address range of adapters installed.

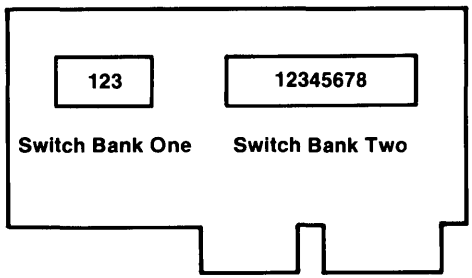


Figure 2. IBM 5080 Peripheral Adapter Switches

Interrupt Level Selected	Switch Bank One Setting		
	Switch 1	**Switch 2**	**Switch 3**
Level 9	On	Off	Off
Level 10	Off	On	Off
Level 11	Off	Off	On

Figure 3. Switch Bank One Settings

Address Range of Adapters	Switch Bank Two Setting			
	Switch 1	Switch 2	Switch 3	Switch 4
1230-1247	On	Off	Off	Off
2230-2247	Off	On	Off	Off
3230-3247	Off	Off	On	Off
4230-4247	Off	Off	Off	On

Figure 4. Switch Bank Two Settings

Note: Switches 5 through 8 are not used.

Modes of Operation

The different modes of operation are selected by programming the NS16450 asynchronous communications element. This is done by selecting the I/O address and writing data out to the I/O address. Address bits A0, A1, and A2 select the different registers that define the modes of operation. Also, the divisor latch access bit (bit 7) of the line control register is used to select certain registers.

The address range for this adapter is X'1230' through X'4247'. Figure 5 and Figure 6 on page 6 depict a value of n, which represents a variable determined by the setting of switch bank two. Switches 1, 2, 3, and 4 of switch bank two allow the card to operate and select the appropriate address range.

I/O Decode (In Hex) Port B	Port A	Register Selected	DLAB State
n238	n230	TX Buffer	DLAB=0 (Write)
n238	n230	RX Buffer	DLAB=0 (Read)
n238	n230	Divisor Latch LSB	DLAB=1
n239	n231	Divisor Latch MSB	DLAB=1
n239	n231	Interrupt Enable Register	DLAB=0
n23A	n232	Interrupt Identification Register	
n23B	n233	Line Control Register	
n23C	n234	Modem Control Register	
n23D	n235	Line Status Register	
n23E	n236	Modem Status Register	

Figure 5. I/O Decodes, Port A and Port B

Notes:

1. n is equal to the first digit of the adapter address range
2. DLAB means Divisor Latch Access Bit.

I/O Decode (In Hex) Port C	Register Selected	DLAB State
n240	TX Buffer	DLAB=0 (Write)
n240	RX Buffer	DLAB=0 (Read)
n240	Divisor Latch LSB	DLAB=1
n241	Divisor Latch MSB	DLAB=1
n241	Interrupt Enable Register	DLAB=0
n242	Interrupt Identification Register	
n243	Line Control Register	
n244	Modem Control Register	
n245	Line Status Register	
n246	Modem Status Register	

Figure 6. I/O Decodes, Port C

Notes:

1. n is equal to the first digit of the adapter address range
2. DLAB means Divisor Latch Access Bit.

A9—>A3 Decode	A2	A1	A0	DLAB	Register
See	x	x	x		
Note 1	0	0	0	0	Receive Buffer Reg. (read) Transmit Holding Reg. (write)
	0	0	1	0	Interrupt Enable
	0	1	0	x	Interrupt Identification
	0	1	1	x	Line Control
	1	0	0	x	Modem Control
	1	0	1	x	Line Status
	1	1	0	x	Modem Status
	1	1	1	x	Scratch (See note 3)
	0	0	0	1	Divisor Latch (LSB)
	0	0	1	1	Divisor Latch (MSB)

Figure 7. Address Bits

Notes:

1. Bits A9 through A3 are used to select specific adapter and serial port.
2. A2, A1, and A0 bits are *don't cares* and are used to select the different registers of the NS16450 chip.
3. The Scratch Register of the NS16450 module should be initialized to all ones and new data should not enter afterwards. This would cause indeterminate data when read address X237 (see "Interrupts" on page 8) is executed.

Interrupts

Three interrupts lines are provided to the system. The interrupt level (9, 10, or 11) is selected by placing the appropriate switch on switch bank one to the on position. Interrupt levels 9, 10, and 11 are shared interrupts. An interrupt register (read address n237, where n is the first digit of address range) is provided to store pending port interrupts. Interrupt register bit assignment as shown in Figure 8.

Hex Address n237							
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
1	1	0	1	0	Port 3	Port 2	Port 1

Figure 8. Interrupt Register Read Format

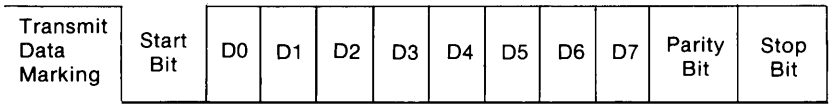
The reset or enable for interrupt level 9 is hex address 02F2.

The reset or enable for interrupt level 10 is hex address 06F2.

The reset or enable for interrupt level 11 is hex address 06F3.

Serial Data Format

The data format is as follows:



Data bit 0 is the first bit to be transmitted or received. The adapter automatically inserts the start bit, the correct parity bit (if programmed to do so), and the stop bit (1, 1-1/2, or 2 depending on the command in the line control register).

External Interface Description

The adapter provides an asynchronous-like interface.

The pin functions for the 10-pin connector are shown in Figure 9.

1	Transmit Data
2	+12 VDC
3	+5 VDC
4	Not Used
5	-12 VDC
6	Receive Data
7	Not Used
8	Not Used
9	Not Used
10	Ground

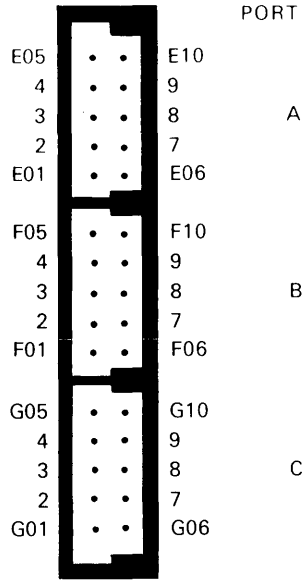


Figure 9. 10-Pin Interface Signals Connector (viewed from rear of adapter)

The adapter converts the data signals from TTL levels to EIA RS232C voltage levels, and vice versa. These signals are sampled or generated by the communications control chip and can then be sensed by the system software to determine the state of the interface or peripheral device. The drivers and receivers used on the adapter are the inverting type; therefore, a 0 EIA level on the line is received or transmitted as a 0 TTL level, and a 1 EIA level is received or transmitted as a 1 TTL level.

Voltage Interchange Information

The signal is considered in the marking condition when the voltage on the interchange circuit, measured at the interface point, is more negative than -3 Vdc with respect to signal ground. The signal is considered in the spacing condition when the voltage is more positive than +3 Vdc with respect to signal ground. The region between +3 Vdc and -3 Vdc is the transition region and is considered an invalid level. The voltage that is more negative than -25 Vdc or more positive than +25 Vdc is also considered an invalid level.

During the transmission of data, the marking condition denotes the binary state 1 and the spacing condition denotes the binary state 0.

For interface control circuits, the function is on when the voltage is more positive than +3 Vdc with respect to signal ground and is off when the voltage is more negative than -3 Vdc with respect to signal ground.

Interchange Voltage	**Binary State**	**Signal Condition**	**Interface Control Function**
+3Vdc to +25Vdc	Binary 0	Spacing	= On
-3Vdc to -25Vdc	Binary 1	Marking	= Off

Figure 10. IBM 5080 Peripheral Adapter Signal Levels

Asynchronous Communications Element Pin Description

The following describes the function of all NS16450 input/output pins. Some of these descriptions reference internal circuits. The use of each signal as implemented on the IBM 5080 Peripheral Adapter is described.

Note: In the following descriptions, a low represents a logic 0 (0 Vdc nominal) and a high represents a logic 1 (+2.4 Vdc nominal).

Input Signals

Chip Select (CS0, CS1, -CS2), Pins 12-14: When CS0 and CS1 are high and -CS2 is low, the chip is selected. Chip selection is complete when the decoded chip select signal is latched with an active (low) address strobe (-ADS) input. This enables communications between the NS16450 and the processor.

Data Input Strobe (DISTR, -DISTR), Pins 22 and 21: When DISTR is high or -DISTR is low while the chip is selected, the processor can read status information or data from a selected register of the NS16450.

Note: Only an active DISTR or -DISTR input is required to transfer data from the NS16450 during a read operation. Therefore, tie either the DISTR input permanently low or the -DISTR line permanently high, if not used.

Data Output Strobe (DOSTR, -DOSTR), Pins 19 and 18: When DOSTR is high or -DOSTR is low while the chip is selected, the processor can write data or control words into a selected register of the NS16450.

Note: Only an active DOSTR or -DOSTR input is required to transfer data to the NS16450 during a write operation. Therefore, tie either the DOSTR input permanently low or the -DOSTR input permanently high, if not used.

-Address Strobe (-ADS), Pin 25: When low, this signal provides latching for the register select (A0, A1, A2) and chip select (CS0, CS1, -CS2) signals.

Note: An active -ADS input is required when the register select (A0, A1, A2) signals are not stable for the duration of a read or write operation. If not required, tie the -ADS input permanently low.

Register Select (A0, A1, A2), Pins 26-28: These three inputs are used during a read or write operation to select an NS16450 register to read from or write into as indicated in Figure 11. Note that the state of the divisor latch access bit (DLAB), which is the most significant bit of the line control register, affects the selection of certain NS16450 registers. The DLAB must be set high by the system software to access the baud-generator divisor latches.

DLAB	A2	A1	A0	Register
0	0	0	0	Receiver Buffer (Read) Transmitter Holding Register (Write)
0	0	0	1	Interrupt Enable
x	0	1	0	Interrupt Identification (Read Only)
x	0	1	1	Line Control
x	1	0	0	Modem Control
x	1	0	1	Line Status
x	1	1	0	Modem Status
x	1	1	1	Scratch
1	0	0	0	Divisor Latch (Least Significant Byte)
1	0	0	1	Divisor Latch (Most Significant Byte)

Figure 11. NS16450 Register Selection

Master Reset (MR), Pin 35: When high, this signal clears all the registers (except the receive buffer, transmitter holding, and divisor latches), and the control logic of the NS16450. Also, the state of various output signals (SOUT, INTRPT, -OUT 1, -OUT 2, -RTS, -DTR) is affected by an active MR input. Refer to the table in Figure 12 on page 14 for reset functions.

Register/Signal	Reset Control	Reset State
Interrupt Enable Register	Master Reset	All Bits Low 0-3 Forced and 4-7 Permanent
Interrupt Identification Register	Master Reset	Bit 0 is High, Bits 1 and 2 are Low, and Bits 3-7 are Permanently Low
Line Control Register	Master Reset	All Bits Low
Modem Control Register	Master Reset	All Bits Low
Line Status Register	Master Reset	All Bits Low, except Bits 5 and 6 are High
Modem Status Register	Master Reset	Bits 0-3 are Low Bits 4-7 = Input Signal
SOUT	Master Reset	High
INTRPT (RCVR Errors)	Read LSR/MR	Low
INTRPT (RCVR Data Ready)	Read RBR/MR	Low
INTRP (THRE)	Read IIR/ Write THR/MR	Low
INTRPT (Modem Status Changes) • OUT 2 • RTS • DTR • OUT 1	Read MSR/MR Master Reset Master Reset Master Reset Master Reset	Low High High High High

Figure 12. NS16450 Reset Functions

Receiver Clock (RCLK), Pin 9: This input is the 16x baud-rate clock for the receiver section of the chip.

Serial Input (SIN), Pin 10: Serial data input from the communications link (peripheral device, modem, or data set).

-Clear to Send (-CTS), Pin 36: The -CTS signal is a modem control function input whose condition can be tested by the processor by reading bit 4 (CTS) of the modem status register. Bit 0 (DCTS) of the modem status register indicates whether the -CTS input has changed state since the previous reading of the modem status register.

Note: This pin is permanently tied low.

-Data Set Ready (-DSR), Pin 37: The -DSR signal is a modem control function input whose condition can be tested by the processor by reading bit 5 (DSR) of the modem status register. When low, this signal indicates that the modem or data set is ready to establish the communications link and transfer data with the NS16450. Bit 1 (DDSR) of the modem status register indicates whether the -DSR input has changed since the previous reading of the modem status register.

Note: This pin is permanently tied low.

-Received Line Signal Detect (-RLSD), Pin 38: The -RLSD signal is a modem control function input whose condition the processor can test by reading bit 7 (RLSD) of the modem status register. When low, this signal indicates that the data carrier had been detected by the modem or data set. Bit 3 (RLSD) of the modem status register indicates whether the -RLSD not input has changed state since the previous reading of the modem status register.

Notes:

1. Received Line Signal Detect is also called Data Carrier Detect (DCD), or Carrier Detect (CD).
2. This pin is permanently tied low.

-Ring Indicator (-RI), Pin 39: The -RI signal is a modem control function input whose condition the processor can test by reading bit 6 (RI) of the modem status register. When low, this signal indicates that a telephone ringing signal has been received by the modem or data set. Bit 2 (TERI) of the modem status register indicates whether the -RI input has changed from a low to high state since the previous reading of the modem status register.

Note: This pin is permanently tied high.

VCC, Pin 40: +5 Vdc supply.

VSS, Pin 20: Ground (0 Vdc) reference.

Output Signals

-Data Terminal Ready (-DTR), Pin 33: When low, this signal informs the modem or data set that the NS16450 is ready to communicate. The -DTR output signal can be set to an active low by programming bit 0 (DTR) of the modem control register to a high level. The -DTR signal is set high by a master reset operation. The -DTR signal is set high during loop mode operation.

Note: No connection, not used.

-Request to Send (-RTS), Pin 32: When low, this signal informs the modem or data set that the NS16450 is ready to transmit data. The -RTS output signal can be set to an active low by programming bit 1 (RTS) of the modem control register. The -RTS signal is set high by a master reset operation. The -RTS signal is set high during loop mode operation.

Note: No connection, not used.

-Output 1 (-OUT 1), Pin 34: With this signal, user-designated output can be set to an active low by programming bit 2 (-OUT 1) of the modem control register to a high level. The -OUT 1 signal is set high by a master reset operation. The -OUT 1 signal is set high during the loop mode operation.

Note: No connection, not used.

-Output 2 (-OUT 2), Pin 31: With this signal, user-designated output can be set to an active low by programming bit 3 (-OUT 2) of the modem control register to a high level. The -OUT 2 signal is set high by a master reset operation. The -OUT 2 signal is set high during the loop mode operation.

Note: No connection, not used.

Chip Select Out (CSOUT), Pin 24: When high, this signal indicates that the chip has been selected by active CS0, CS1, and -CS2 inputs. No data transfer can be initiated until the CSOUT signal is a logic 1.

Note: No connection, not used.

Driver Disable (DDIS), Pin 23: This signal goes low whenever the processor is reading data from the NS16450. A high-level DDIS output can be used to disable an external transceiver (if used between the processor and NS16450 on the D7-D0 data bus) at all times, except when the processor is reading data.

Note: No connection, not used.

-Baudout (-BAUDOUT), Pin 15: This signal is a 16x clock signal for the transmitter section of the NS16450. The clock rate is equal to the main reference oscillator frequency divided by the specified divisor in the baud-generator division latches. The -Baudout may also be used for the receiver section by tying this output to the RCLK input of the chip.

Note: This pin is tied to RCLK pin 9.

Interrupt (INTRPT), Pin 30: This signal goes high whenever any one of the following interrupt types has an active high condition and is enabled through the IER: receiver error flag, received data

available, transmitter holding register empty, or modem status. The Interrupt signal is reset low on the appropriate interrupt service or a master reset operation.

Note: Generates interrupt request.

Serial Output (SOUT), Pin 11: Composite serial data output to the communications link (peripheral, modem or data set). The SOUT signal is set to the marking (logic 1) state on a master reset operation.

Note: Provides data to attached device.

Input/Output Signals

Data Bus (D7-D0), Pins 1-8: This bus consists of eight tri-state I/O lines. The bus provides bidirectional communications between the NS16450 and the processor. Data, control words, and status information are transferred through the D7-D0 data bus.

External Clock Input/Output (XTAL1, XTAL2), Pins 16 and 17: These two pins connect the main timing reference (crystal or signal clock) to the NS16450.

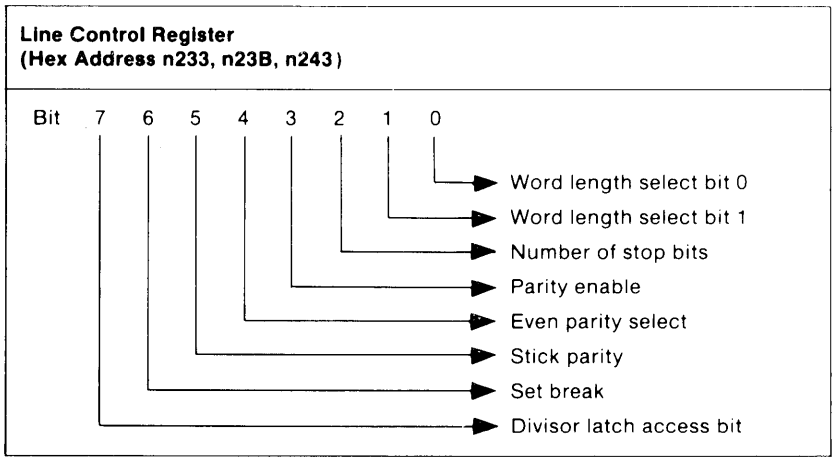
Programming Considerations

The NS16450 has several accessible registers. The system programmer may access or control any of the NS16450 registers through the processor. These registers are used to control NS16450 operations and to transmit and receive data.

Note: The **n** in address is the card number (1-4).

Line Control Register

The system programmer specifies the format of the asynchronous data communications exchange through the line control register. In addition to controlling the format, the programmer may retrieve the contents of the line control register for inspection. This feature simplifies system programming and eliminates the need for separate storage of the line characteristics in system memory. The contents of the line control register are described below:



Bits 0, 1 These two bits specify the number of bits in each transmitted or received serial character. The encoding of bits 0 and 1 is as follows:

Bit 1	Bit 0	Word Length
0	0	5 bits
0	1	6 bits
1	0	7 bits
1	1	8 bits

Bit 2 This bit specifies the number of stop bits in each transmitted or received serial character. If bit 2 is a logical 0, one stop bit is generated or checked in the transmit or receive data, respectively. If bit 2 is a logical 1 when a 5-bit word length is selected through bits 0 and 1, 1-1/2 stop bits are generated or checked. If bit 2 is a logical 1 when either a 6-, 7-, or 8-bit word length is selected, two stop bits are generated or checked.

Bit 3 This bit is the parity enable bit. When bit 3 is a logical 1, a parity bit is generated (transmit data) or checked (receive data) between the last data word bit and stop bit of the serial data. (The parity bit is used to produce an even or odd number of 1's when the data word bits and the parity bit are summed.)

Bit 4 This bit is the even parity select bit. When bit 3 is a logical 1 and bit 4 is a logical 0, an odd number of logical 1 is transmitted or checked in the data word bits and parity bit. When bit 3 is a logical 1 and bit 4 is a logical 1, an even number of bits are transmitted or checked.

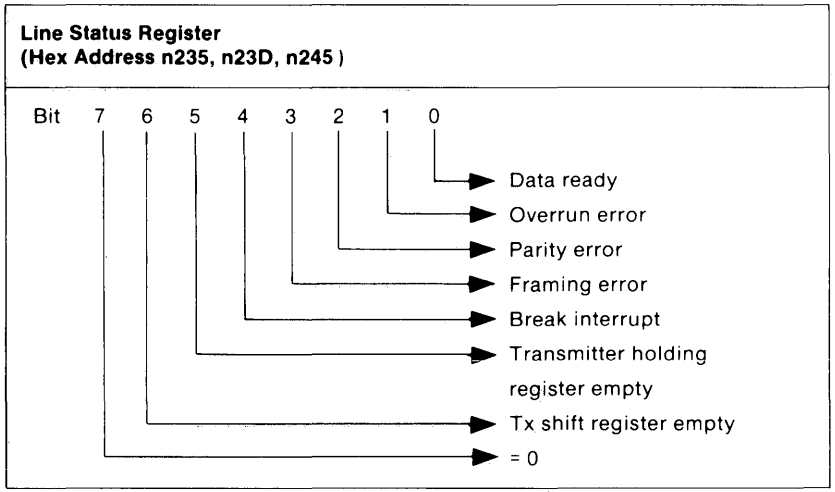
Bit 5 This bit is the stick parity bit. When bit 3 is a logical 1 and bit 5 is a logical 1, the parity bit is transmitted and then detected by the receiver as a logical 0 (space parity) if bit 4 is a logical 1, or as a logical 1 (mark parity) if bit 4 is a logical 0.

Bit 6 This bit is the set break control bit. When bit 6 is a logical 1, the serial output (SOUT) is forced to the spacing (logical 0) state and remains there regardless of other transmitter activity. The set break is disabled by setting bit 6 to a logical 0. This feature enables the processor to alert a terminal in a computer communications system.

Bit 7 This bit is the divisor latch access bit (DLAB). It must be set high (logical 1) to access the divisor latches of the baud-rate generator during a read/write operation. It must be set low (logical 0) to access the receiver buffer, the transmitter holding register, or the interrupt enable register.

Line Status Register

This 8-bit register provides status information to the processor about the data transfer. The contents of the line status register are described below:



Bit 0 This bit is the receiver data ready (DR) indicator. Bit 0 is set to a logical 1 whenever a complete incoming character has been received and transferred into the receiver buffer register. Bit 0 may be reset to a logical 0 either by the processor reading the data in the receiver buffer or by writing a logical 0 into it from the processor.

Bit 1 This bit is the overrun error (OE) indicator. Bit 1 indicates that data in the receiver buffer register was not read by the processor before the next character was transferred into the receiver buffer register, and thereby destroyed the previous character. The OE indicator is reset whenever the processor reads the contents of the line status register.

Bit 2 This bit is the parity error (PE) indicator. Bit 2 indicates that the received data character does not have the correct even or odd parity as selected by the even parity select bit. The PE bit is set to a logical 1 whenever a parity error is detected and is reset to a logical 0 whenever the processor reads the contents of the line status register.

Bit 3 This bit is the framing error (FE) indicator. Bit 3 indicates that the received character does not have a valid stop bit. Bit 3 is set to a logical 1 whenever the stop bit following the last data bit or parity is detected as a zero bit (spacing level).

Bit 4 This bit is the break interrupt (BI) indicator. Bit 4 is set to a logical 1 whenever the received data input is held in the spacing (logical 0) state for longer than a full word transmission time (that is, the total time of start bit + data bits + parity + stop bits).

Note: Bits 1 through 4 are the error conditions that produce a receiver line status interrupt whenever any of the corresponding conditions are detected.

Bit 5 This bit is the transmitter holding register empty (THRE) indicator. Bit 5 indicates that the NS16450 is ready to accept a new character for transmission. In addition, this bit causes the NS16450 to issue an interrupt to the processor when the THRE interrupt enable is set high. The THRE bit is set to a logical 1 when a character is transferred from the transmitter holding register into the transmitter shift register. The bit is reset to logical 0 concurrently with the loading of the transmitter holding register by the processor.

Bit 6 This bit is the transmitter empty (TEMT) indicator. Bit 6 is set to a logical 1 whenever the transmitter holding register (THR) and the transmitter shift register (TSR) are both empty. It is reset to a logical 0 whenever either the THR or TSR contain a data character. Bit 6 is a read-only bit.

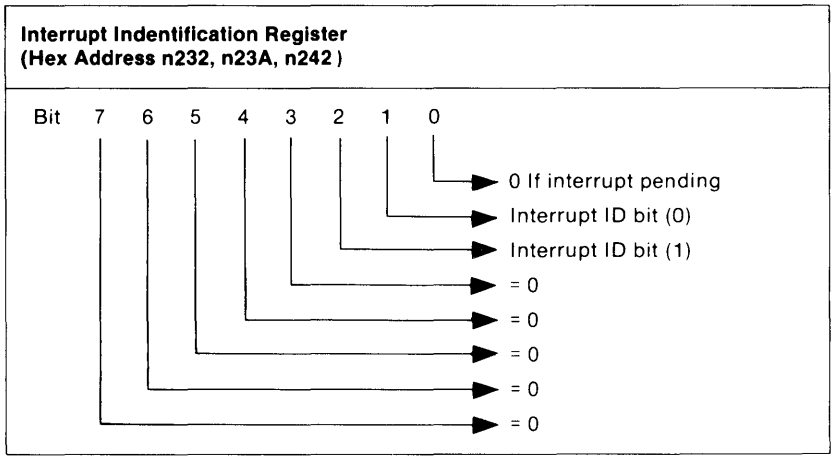
Bit 7 This bit is permanently set to logical 0.

Interrupt Identification Register

The NS16450 has an on-chip interrupt capability that allows for complete flexibility in interfacing to microprocessors. To provide minimum software overhead during data character transfers, the NS16450 ranks interrupts into four levels:

- Receiver line status (priority 1)
- Received data ready (priority 2)
- Transmitter holding register empty (priority 3)
- Modem status (priority 4).

Information indicating that a priority interrupt is pending and information on the type of interrupt is stored in the interrupt identification register. Refer to the "Interrupt Control Functions" table in Figure 13 on page 23. The interrupt identification register (IIR), when addressed during chip-select time, freezes the highest priority interrupt pending, and no other interrupts are acknowledged until that particular interrupt is serviced by the processor. The contents of the IIR are described below.



Bit 0 This bit can be used in hardwired, priority, or polled environment to indicate whether an interrupt is pending. When bit 0 is a logical 0, an interrupt is pending and the IIR contents may be used as a pointer to the appropriate interrupt service routine. When bit 0 is a logical 1, no interrupt is pending and polling (if used) is continued.

Bits 1, 2 These two bits of the IIR are used to identify the highest priority interrupt pending as indicated in Figure 13 on page 23.

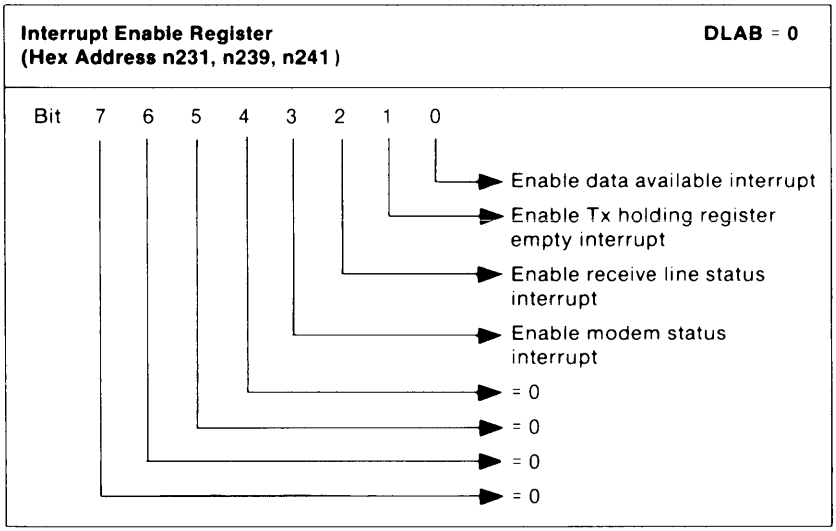
Bits 3-7 These five bits of the IIR are always logical 0.

Interrupt ID Register			Interrupt Set and Reset Functions			
Bit 2	**Bit 1**	**Bit 0**	**Priority Level**	**Interrupt Type**	**Interrupt Source**	**Interrupt Reset Control**
0	0	1	—	None	None	—
1	1	0	Highest	Receiver Line Status	Overrun Error or Parity Error or Framing Error or Break Intrpt.	Reading the Line Status Register
1	0	0	Second	Received Data Available	Receiver Data Available	Reading the Receiver Buffer Register
0	1	0	Third	Transmitter Holding Register Empty	Transmitter Holding Register Empty	Reading the IIR Register or Writing into the Transmitter Holding Register
0	0	0	Fourth	Modem Status	Clear to Send or Data Set Ready or Ring Indicator or Received Line Signal Detect	Reading the Modem Status Register

Figure 13. Interrupt Control Functions

Interrupt Enable Register

This 8-bit register enables the four types of interrupts of the NS16450 to separately activate the chip interrupt (INTRPT) output signal. The interrupt system can be totally disabled by resetting bits 0 through 3 of the interrupt enable register. Similarly, by setting the appropriate bits of this register to a logical 1, selected interrupts can be enabled. Disabling the interrupt system inhibits the interrupt identification register and the active (high) INTRPT output from the chip. All other system functions operate in their normal manner, including the setting of the line status and modem status registers. The contents of the interrupt enable register are described below:



Bit 0 This bit enables the received data available interrupt when set to logical 1.

Bit 1 This bit enables the transmitter holding register empty interrupt when set to logical 1.

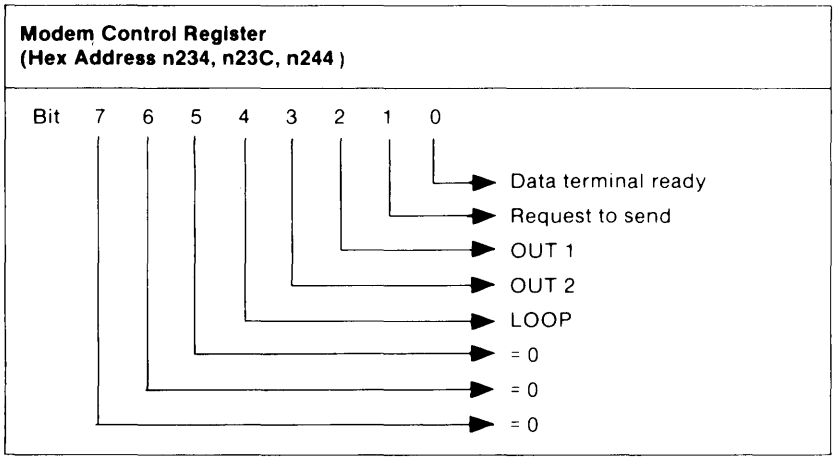
Bit 2 This bit enables the receiver line status interrupt when set to logical 1.

Bit 3 This bit enables the modem status interrupt when set to logical 1.

Bits 4-7 These four bits are always logical 0.

Modem Control Register

This 8-bit register controls the interface with the modem or data set (or other peripheral device). The contents of the modem control register are described below:



Bit 0 This bit controls the data terminal ready (-DTR) output. When bit 0 is set to a logical 1, the -DTR output is forced to a logical 0. When bit 0 is reset to a logical 0, the -DTR output is forced to a logical 1.

Note: The -DTR output of the NS16450 may be applied to an EIA inverting line driver to obtain the proper polarity input at the modem or data set.

Bit 1 This bit controls the request to send (-RTS) output. Bit 1 affects the -RTS output in a manner identical to that described above for bit 0.

Note: The -RTS output of the NS16450 may be applied to an EIA-inverting line driver to obtain the proper polarity input at the modem or data set.

Bit 2 This bit controls the output 1 (-OUT 1) signal, which is an auxiliary user-designated output. Bit 2 affects the -OUT 1 output in a manner identical to that described above for bit 0.

Note: The -OUT 1 output of the NS16450 may be applied to an EIA inverting line driver to obtain the proper polarity input at the modem or data set.

Bit 3 This bit controls the output 2 (-OUT 2) signal, which is an auxiliary user-designated output. Bit 3 affects the -OUT 2 output in a manner identical to that described above for bit 0.

Note: The -OUT 2 output of the NS16450 may be applied to an EIA inverting line driver to obtain the proper polarity input at the modem or data set.

Bit 4 This bit provides a loopback feature for diagnostic testing of the NS16450. When bit 4 is set to logical 1, the following occurs:

The transmitter serial output (SOUT) is set to the marking (logical 1) state.

The receiver serial input (SIN) is disconnected.

The output of the transmitter shift register is "looped back" into the receiver shift register input.

The four modem control inputs (-CTS, -DSR, -RLSD, and -RI) are disconnected.

The four modem control outputs (-DTR, -RTS, -OUT 1, and -OUT 2) are internally connected to the four modem control inputs, and the modem control output pins are forced high.

In the the diagnostic mode the receiver and transmitter interrupts are fully operational. The modem control interrupts are also operational, but the sources of the interrupts are now the lower 4 bits of the modem control register instead of the 4 modem control inputs. The interrupts are still controlled by the interrupt enable register.

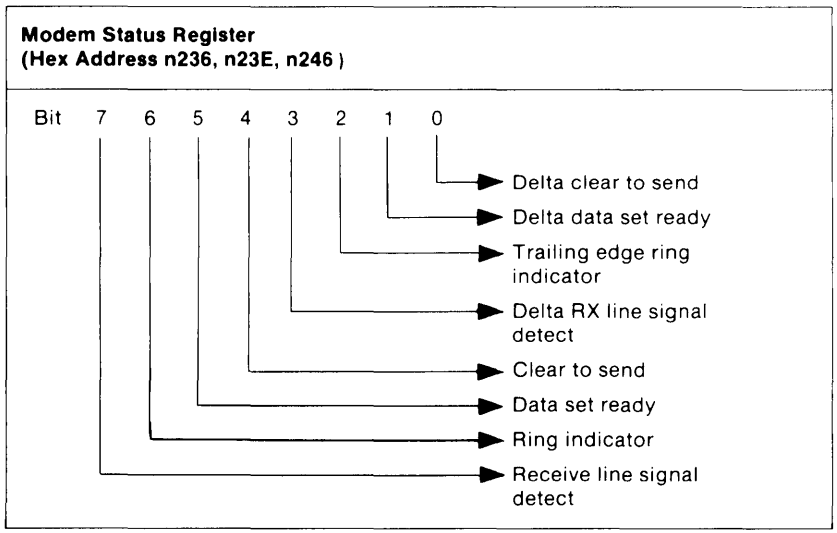
The NS16450 interrupt system can be tested by writing into the lower 6 bits of the line status register and into the lower 4 bits of the modem status register. Setting any of these bits to a logical 1 generates the appropriate interrupt (if enabled). The resetting of these interrupts is the same as in normal NS16450 operation. To return to normal operation, the registers must be reprogrammed for normal operation and then bit 4 of the modem control register must be reset to logical 0. The transmitter should be idle when this bit changes state.

Bits 5-7 These bits are permanently set to logical 0.

Modem Status Register

This 8-bit register provides the current state of the control lines from the modem (or peripheral device) to the processor. In addition to this current-state information, 4 bits of the modem status register provide change information. These bits are set to a logical 1 whenever a control input from the modem changes state. They are reset to logical 0 whenever the processor reads the modem status register.

The contents of the modem status register are described below:



Bit 0 This bit is the delta clear-to-send (DCTS) indicator. Bit 0 indicates that the -CTS input to the chip has changed state since the last time it was read by the processor.

Bit 1 This bit is the delta data set ready (DDSR) indicator. Bit 1 indicates that the -DSR input to the chip has changed state since the last time it was read by the processor.

Bit 2 This bit is the trailing edge of the ring indicator (TERI) detector. Bit 2 indicates that the -RI input to the chip has changed from an ON (logical 1) to an OFF (logical 0) condition.

Bit 3 This bit is the delta received line signal detector (DRLSD) indicator. Bit 3 indicates that the -RLSD input to the chip has changed state since the last time it was read by the processor.

Note: Whenever bit 0, 1, 2, or 3 is set to a logical 1, a modem status interrupt is generated, if the appropriate interrupt enable bit is set in the IER.

Bit 4 This bit is the complement of the clear to send (-CTS) input. Setting bit 4 (loop) of the MCR to a logical 1, is equivalent to RTS in the MCR.

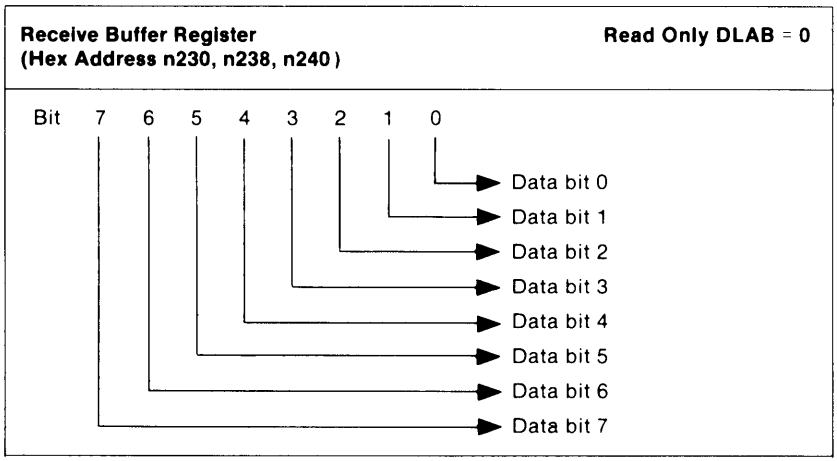
Bit 5 This bit is the complement of the data set ready (-DSR) input. If bit 4 (loop) of the MCR is set to a logical 1, this bit is equivalent to DTR in the MCR.

Bit 6 This bit is the complement of the ring indicator (-RI) input. If bit 4 (loop) of the MCR is set to a logical 1, this bit is equivalent to -OUT 1 in the MCR.

Bit 7 This bit is the complement of the received line signal detect (-RLSD) input. If bit 4 (loop) of the MCR is set to a logical 1, this bit is equivalent to -OUT 2 of the MCR.

Receiver Buffer Register

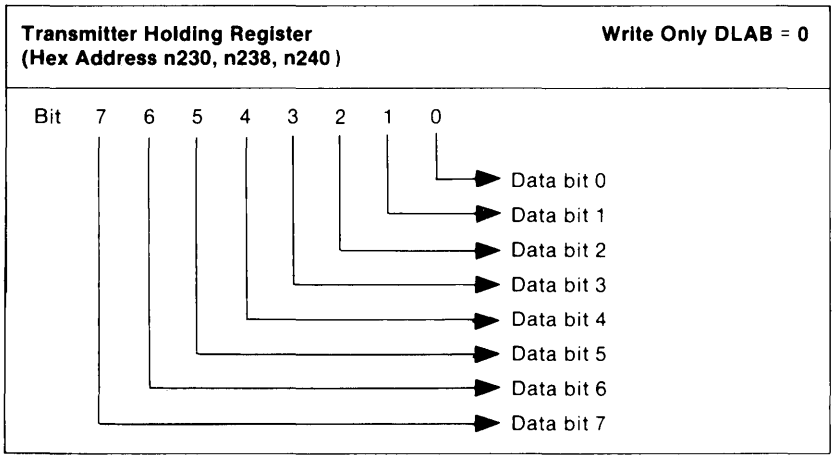
The receiver buffer register contains the received character as defined below:



Bit 0 is the least significant bit and is the first bit serially received.

Transmitter Holding Register

The transmitter holding register contains the character to be serially transmitted and is defined below:



Bit 0 is the least significant bit and is the first bit serially transmitted.

Programmable Baud-Rate Generator

The NS16450 contains a programmable baud-rate generator that can divide the clock input (1.8432 MHz) by any divisor from 1 to 655,535 or 2^{16}-1. The output frequency of the baud-rate generator is the baud rate multiplied by 16. Two 8-bit latches store the divisor in a 16-bit binary format. These divisor latches must be loaded during initialization to insure desired operation of the baud-rate generator. Upon loading either of the divisor latches, a 16-bit baud counter is immediately loaded. This prevents long counts on initial load. The contents of the divisor latches are indicated below:

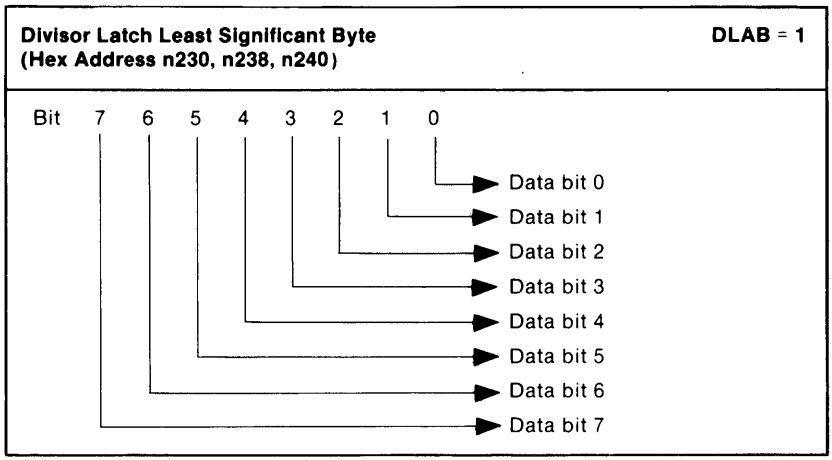


Figure 14. Divisor Latch Least Significant Byte

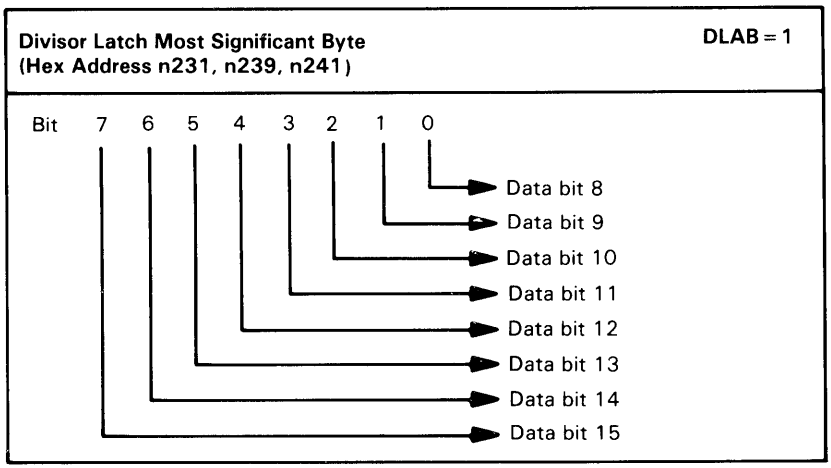


Figure 15. Divisor Latch Most Significant Byte

Figure 16 illustrates the use of the baud-rate generator with a frequency of 1.8432 MHz. For baud rates of 19,200 and below, the error obtained is minimal.

Note: The maximum operating frequency of the baud generator is 3.1 MHz. The data rate should never be greater than 19,200 baud.

Desired Baud Rate	Divisor Used to Generate 16x Clock (Decimal)	(Hex)	Percent Error Difference Between Desired and Actual
50	2304	900	—
75	1536	600	—
110	1047	417	0.026
134.5	857	359	0.058
150	786	300	—
300	384	180	—
600	192	C0	—
1200	96	60	—
1800	64	40	—
2000	58	3A	0.69
2400	48	30	—
3600	32	20	—
4800	24	18	—
7200	16	10	—
9600	12	C	—
19200	6	6	—

Figure 16. Baud Rates at 1.8432 MHz

Connector Specifications

The adapter has a 10-pin connector at the rear of the adapter. The following figure shows the signals and their pin assignments.

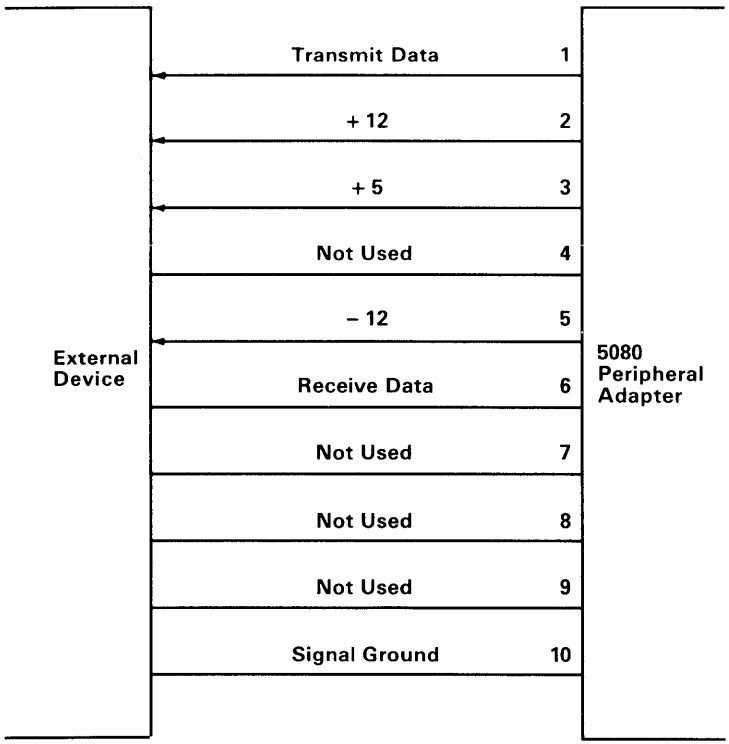


Figure 17. Connector Specifications

Personal Computer
Hardware Reference
Library

Advanced Color Graphics Display Adapter

Contents

Description

The Advanced Color Graphics Display Adapter attaches to the I/O channel and drives a 14-inch color display at a 46 Hz (frame rate) or 92 Hz (field rate) interlace refresh rate. The adapter provides a 256K-byte bit map that is translated to the screen as 720 pels horizontally by 512 pels vertically, at 4 bits per pel. A 4-plane bit map (64K-byte each plane) is provided from which 16 colors are picked from a 64 color palette.

The adapter includes several hardware performance assists, including a write mask to protect bit fields within a byte, a barrel shifter to rotate bits within a byte, and a logic unit to combine source bytes before they are written into the bit map. In addition, a plane select register and foreground or background register are provided as a means to select individual planes or all planes for update with each system access using preprogrammed foreground or background color data. A 4-bit to 6-bit video lookup Random Access Memory (RAM), writeable from the system, is also provided for the selection of 16 usable colors from a palette of 64 available colors, and is represented to the monitor using 2-bits per primary color (red, red intense, green, green intense, blue and blue intense).

Figure 1 on page 2 is a block diagram of the Advanced Color Graphics Display Adapter.

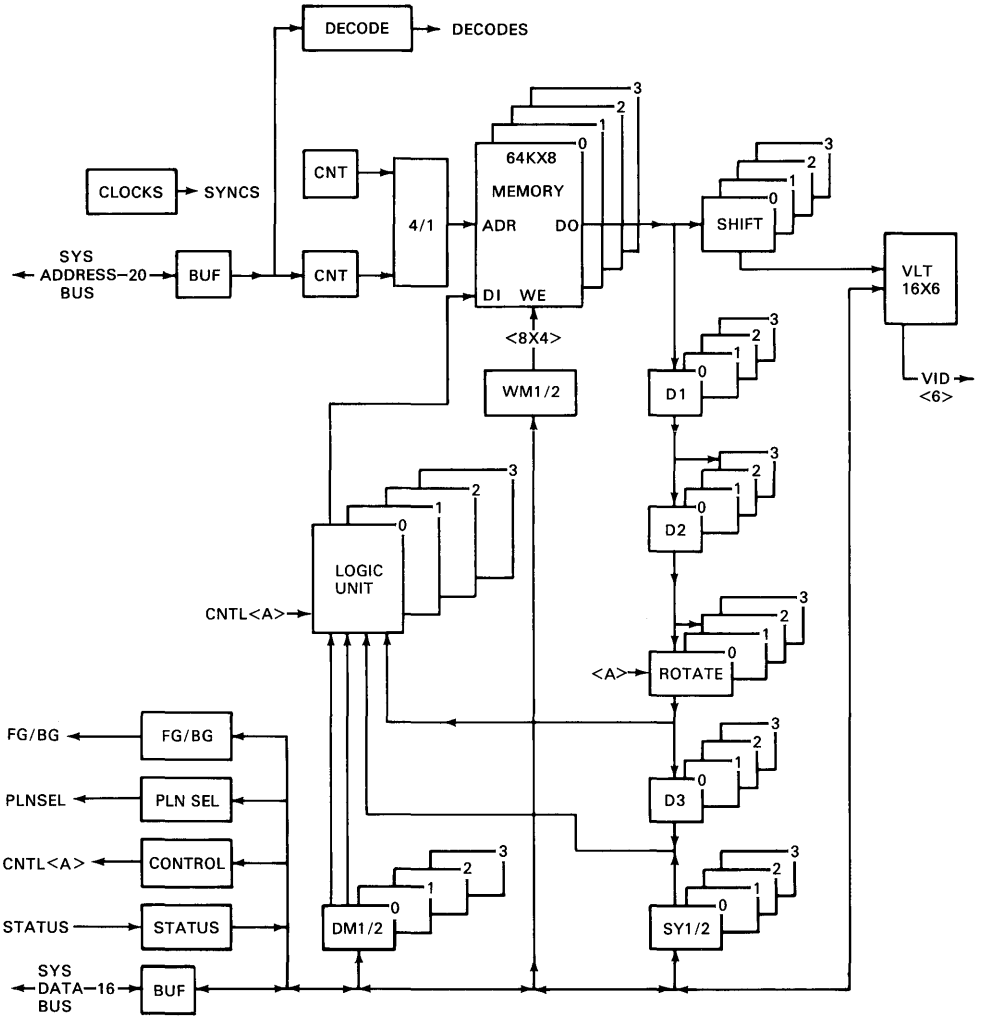


Figure 1. Advanced Color Graphics Display Adapter Functional Block Diagram

2 Advanced Color Graphics Display Adapter

Operation

Data to be displayed is written into the 64K x 8 x 4 plane bit map memory by the system. This data is then scanned out of the memory and sent as a serial video stream to the monitor.

The system processor can manipulate data residing in the bit map memory in several ways. The choice of a particular method is controlled by the type of instruction (Load or Store) and the current value of a 2-bit mode field which resides within a 16-bit control register. The contents of the control register must be initialized prior to the memory operation.

Hardware on the card allows data in the bit map to be manipulated without passing through a processor. This hardware includes:

- A 16-bit write mask that enables write operations to individual bits
- A shifter that realigns bits before they are written to memory
- Data registers to hold data being processed
- Two 8-bit data mask registers that select bits being merged from data registers
- A logic unit that does all the actual merging.

Using this hardware, source data from either the system or from the bit map may replace or be merged with data in the bit map. Images on the screen (in the bit map) can thus be moved, inverted, overlaid, or replaced.

The adapter occupies two address ranges on the I/O channel. Sixteen bytes of I/O address space at X '0150' through '015F' are used to load the data mask register, and the control registers. See the section entitled "I/O Operations" on page 6 for specific register addresses and bit assignments.

The second address range occupied by the adapter is the bit map memory that consists of 128K-bytes starting at hex address 'D20000' in memory space on the I/O channel. When addressing the bit map memory, the low order address bit must be 0. Address bits A01 through A16 are used to access memory. Address bit A00 is not used during memory accesses. This restriction arises because the adapter allows word accesses to the bit map to begin on either even or odd byte boundaries. Because the system restricts half-word accesses to even bytes, address lines to the bit map memory are offset by one bit.

The most significant data byte of the I/O channel is written into or read from the bit map memory location specified by the shifted address lines. The least significant data byte is written into or read from the memory location specified by the shifted address lines increased or decreased by one in the X or Y direction. The increase and decrease bits of the control register determine whether the address is increased or decreased. The X and Y bits of the control register determine whether the X or Y direction is changed.

As a result of the addressing scheme described above, each physical memory location in the bit map can be accessed at either of two addresses. Access directly through the most significant byte, or after the address bus has been increased or decreased, through the least significant byte.

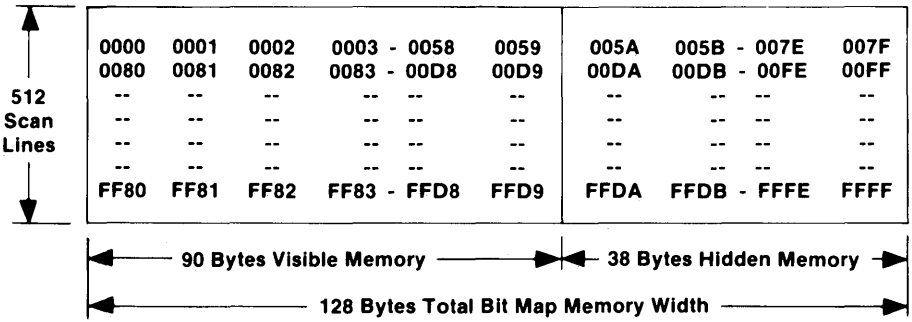


Figure 2. Bit Map Memory Addresses

The mapping between the physical addresses shown in Figure 2 and denoted as 'P' in the equations and the I/O channel address denoted as 'C' in the equations is shown, as follows, for each of the increase and decrease bits and the X and Y status bits.

		Word Storage Location			
I/O Channel Byte	**Inc/Dec**	**X/Y**	**Physical to I/O Channel**		**I/O Channel to Physical**
MS Byte	either	either	C=2P	=>	P=C/2
LS Byte	+	X	C=2P-2	=>	P=C/2 +1
LS Byte	-	X	C=2P+2	=>	P=C/2 -1
LS Byte	+	Y	C=2P-256	=>	P=C/2 +128
LS Byte	-	Y	C=2P+256	=>	P=C/2 -128

Figure 2 shows how on-card physical memory byte addresses are mapped to the display screen. The bits in each byte are shifted onto the screen with the most significant bit to the left.

Bit Map Memory Operations

The following are possible memory operations and respective mode bit definitions.

System Write Operation (Mode = 00)

This operation writes 16 bits of external data to the bit map memory at the address specified in the system processor 'Store' instruction.

Overlay Write Operation (Mode = 01)

This operation writes 16 bits of external data to the onboard write-mask registers (WM1 and WM2). It also initiates a write operation of the onboard system data latches to the bit map memory at the address specified in the system processor 'Store' instruction. The new value in the write mask controls the write to the bit map. This operation will not affect the contents of the system data latches. Preloading the system data latches with the proper constant makes it possible to 'AND' or 'OR' system data with the bit map memory in one memory operation.

Adapter Write Operation (Mode = 10)

This operation writes data from the onboard data latches (D1,D2,D3) to the bit map memory using the rotate count, write mask, data mask, data bus and the logic unit. The following operations must be completed before executing this operation:

- Load the data latches using an adapter read operation.
- Load the write mask using an overlay write operation.
- Load the operation mode, rotate count and logic function using an I/O 'Store' to the adapter control register.

System Read Operation (Mode = XX)

This operation reads 16 bits of data from the bit map memory and places it on the system data bus. Onboard data latches D1, D2, and D3 are updated by this operation. Data is loaded into latches D1, D2, and D3 regardless of the setting of the mode. However, valid data is returned to the system processor only if the mode is '00'. Data returned to the system is from the lowest numbered plane that is enabled for reading.

Automatic Read/Write Operation (Mode = 11)

In this mode of operation, data is alternately read into D1, D2 and D3 and then written to memory with successive write operations from the microprocessor. Data read from memory is not gated onto any busses external to the data path, but is used only to update D1, D2, and D3. This allows the fast 'Store' operation to be used for block transfer operations with no 'Loads' required.

To get the alternate read/write operations into phase with the source and destination addresses, the system processor should do a 'Load' operation from the first source address. Subsequent 'Stores' will automatically alternate between read and writes to the bit map.

I/O Operations

To manipulate the various registers on the adapter, a set of I/O operations are required.

Data Mask Register (X'0152') Write Only

Typically these two 8-bit registers (DM1 and DM2) are initialized with data representing the inverse of each other, and are used to mask the bits on or off for logical combination through the logic unit.

- An 8 bit for DM1 Mask (Data Bus MS Byte, bits 0-7)
- An 8 bit for DM2 Mask (Data Bus LS Byte, bits 0-7).

Data Control Register (X'0150') Write Only

Bits 0 through 11 should not be changed for at least 1.4 microseconds after a memory operation since the memory operation may still be in progress.

Bits 0-2 Rotate Count

The rotate count determines the number of bits that the data read from the bit map is shifted to the left prior to being written back into the bit map. This value has no effect during system write or overlay write modes. It has effect only during adapter write or automatic write modes.

Bits 3-5 Logic Unit Function Control

Bit
543
---
000 - Pass through B
010 - Pass through A
100 - Pass through 'Not' B
101 - A 'OR' B
110 - Pass through 'Not' A
111 - A 'NOR' B

Logic unit function control bits (bits 3-5) determine how data previously read from the bit map or written from the system are merged before being written into the bit map. When set to '101'B, the bit fields masked off by the data mask register are merged with system data before being written back into the bit map memory. This is the function normally needed during adapter write and automatic write modes. When set to '010'B, the bit field masked off by data mask register 1 (DM1) is written into the bit map memory. This is the function normally needed in system write and overlay write modes. When set to '000'B, the bit field masked off by data mask register 2 (DM2) is written into the bit map memory. This function can be used in adapter write and automatic write modes when no bit shift is used. In each case, an inverting form of the operation exists to allow data to be inverted as it is written to the bit map. See "Bit Map Memory Operations" on page 5 for further explanations.

Bits 6-7 Reserved

Bits 8-9 Memory Mode

Bit
98
--
00 = System write operation
01 = Overlay write operation
10 = Adapter write operation
11 = Automatic read/write operation

Bit 10 Address Counter Mode

0 = Increase the address counter
1 = Decrease the address counter

Bit 11 Address Counter Stepping

0 = Y stepping
1 = X stepping

Bits 10 and 11 control whether the LSB of a bit map memory operation is accessed from the left, right, above or below the MSB. See "Bit Map Memory Operations" on page 5 for further explanation.

Bit 12 Block transfer

Bit 12 controls block transfer mode. When this bit is set from a 0 to a 1, the address of the next memory location accessed is stored in a pointer register on the card. This address is adjusted to point at the next memory location at the end of each memory cycle. That is, the X or Y address is increased or decreased. After the first block transfer memory cycle has loaded the on-card pointer register, the memory address supplied on the data bus is ignored, and the pointer is used to access memory. To reload the pointer register, either clear and set the block transfer bit again or read from the block transfer reload I/O address, X'0152'.

Bit 13 Interrupt enable

If bit 13 is 1, an interrupt is generated at the start of vertical sync. If this bit is 0, the interrupt is not generated. See "Interrupts" on page 11.

Bit 14 Sync enable (always 1)

Bit 15 Video enable

0 = Disables video to monitor
1 = Enables video to monitor

Color Plane Select Register X'0154' Write Only

Color planes can only be written or read by the system if the corresponding plane select bit is active.

If more than one plane is enabled during a read operation, only the lowest numbered is enabled to the I/O channel.

Bits 0-3 Color plane select

- **Bit 0** 1 = Plane select 0
- **Bit 1** 1 = Plane select 1
- **Bit 2** 1 = Plane select 2
- **Bit 3** 1 = Plane select 3

Bit 4 1 = Foreground/background multi-plane write enable

If bit 4 is a 0, data bits written into a bit map plane are stored as written. If bit 4 is a 1, data is translated to the foreground bit (data = 1) or into the background bit (data = 0). This allows all selected planes to be updated simultaneously with the selected foreground and background colors.

Bits 5-7 Reserved

Foreground/Background Register X'156' Write Only

The foreground/background register is physically contained in the data path LSI modules, with a foreground and background bit pair implemented per plane.

Bits 0-3 Foreground color

- **Bit 0** Foreground plane 0 - LSI module 0
- **Bit 1** Foreground plane 1 - LSI module 1
- **Bit 2** Foreground plane 2 - LSI module 2
- **Bit 3** Foreground plane 3 - LSI module 3

Bits 4-7 Background Color

- **Bit 4** Background plane 0 - LSI module 0
- **Bit 5** Background plane 1 - LSI module 1
- **Bit 6** Background plane 2 - LSI module 2
- **Bit 7** Background plane 3 - LSI module 3

RAS Status Register (X'0150') Read Only

Bit 0 1 = Even 0 = Odd field

Bit 1 Horizontal sync toggle latch

Bit 2 0 = Vertical sync inactive 1= Vertical sync active

Bit 3 Reserved

Bit 4 1= X 0 = Y Stepping

Bit 5 1 = Interrupt pending

Bit 6 0 = Increase 1 = Decrease the address counter

Bit 7 1 = Enable load address

Bits 8-13 Serialized Color Video

- **Bit 8** Red (R1)
- **Bit 9** Red intense (R2)
- **Bit 10** Green (G1)
- **Bit 11** Green intense (G2)
- **Bit 12** Blue (B1)
- **Bit 13** Blue intense (B2)

Bits 14-15 Reserved

Video Look-up Table Register X'0158' Write Only

The video look-up table is selected via the I/O channel address. The data written, specified on the I/O channel most significant byte (bits 0 through 5), is written to the address specified on the I/O channel least significant byte (bits 0 through 3). To avoid scintillation of the screen, this operation should only be performed during vertical blanking.

The mapping of the most significant byte, bits 0-5 to output colors, is as follows:

		Most Significant Byte					
Pel	**Color**	**0**	**1**	**2**	**3**	**4**	**5**
Low	Intense red	1	0	0	0	0	0
Med	Intense red	0	1	0	0	0	0
High	Intense red	1	1	0	0	0	0
Low	Intense green	0	0	1	0	0	0
Med	Intense green	0	0	0	1	0	0
High	Intense green	0	0	1	1	0	0
Low	Intense Blue	0	0	0	0	1	0
Med	Intense blue	0	0	0	0	0	1
High	Intense blue	0	0	0	0	1	1

Other combinations of the three basic colors described above produce alternate hues.

Block Transfer Reload (X'0152') Read Only

Reading from this I/O location loads the address of the next memory access into the on-card address pointer register. No significant data is returned when this location is read. See "Data Control Register (X'0150') Write Only" on page 7 for more information regarding the use of this I/O command.

Interrupts

The adapter generates a level 11 interrupt at the start of vertical sync if bit 13 (interrupt enable) of the control register is a 1. This interrupt will not occur if bit 13 is a 0. When the adapter generates an interrupt, bit 5 of the RAS status register is set to 1. To clear bit 5 and reenable level 11 interrupts, an output to hex 6F3 with any data value must be issued. Interrupt 11 is a shared interrupt.

Connector Specifications

The adapter has a 16-pin connector at the rear of the adapter. The following figure shows the signals and their pin assignments.

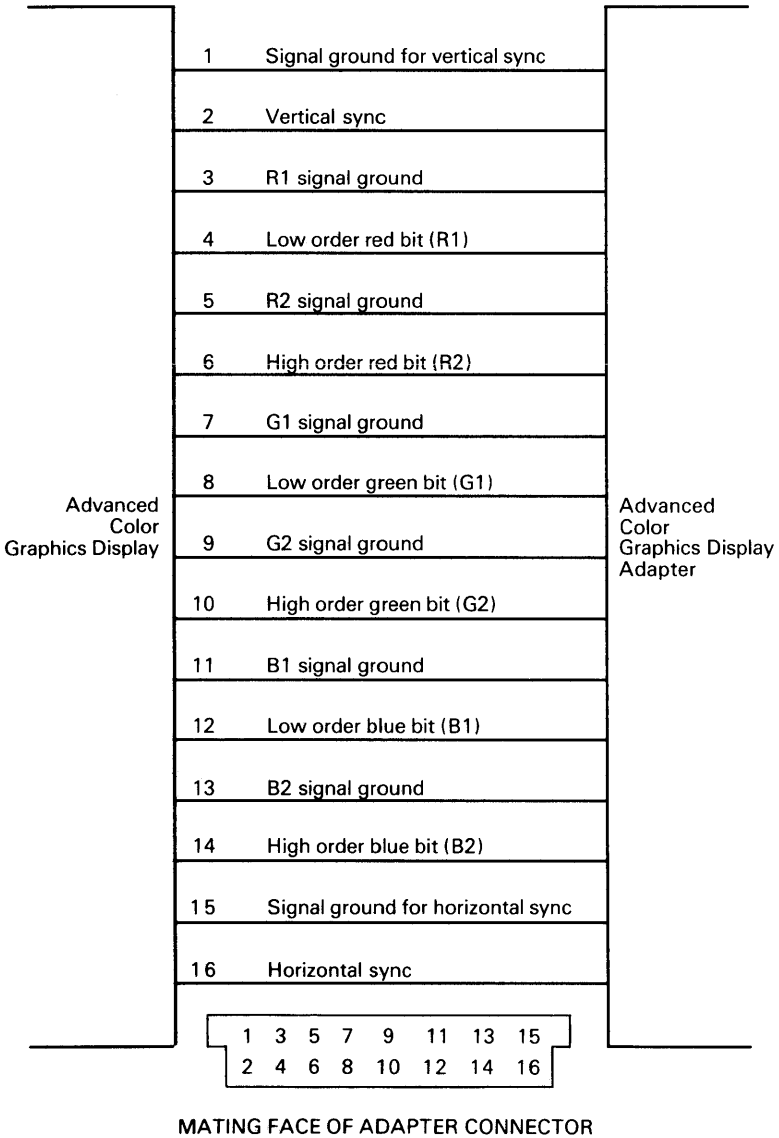


Figure 3. Advanced Color Graphics Display Adapter Connector Specifications

IBM

Personal Computer Hardware Reference Library

Advanced Monochrome Graphics Display Adapter

ADVANCED MONOCHROME GRAPHICS DISPLAY ADAPTER

Contents

Description

The Advanced Monochrome Graphics Display Adapter attaches to the I/O channel and drives a 12-inch monochrome display at a 46 Hz (frame rate) or 92 Hz (field rate) interlace refresh rate. The adapter provides a 64K-byte bit map that is translated to the screen as 720 pels horizontally by 512 pels vertically, one bit per pel.

The adapter includes a number of hardware performance assists, including a write mask to protect bit fields within a byte, a barrel shifter to rotate bits within a byte, and a logic unit to combine source bytes before they are written into the bit map.

Figure 1 on page 2 is a block diagram of the Advanced Monochrome Graphics Display Adapter.

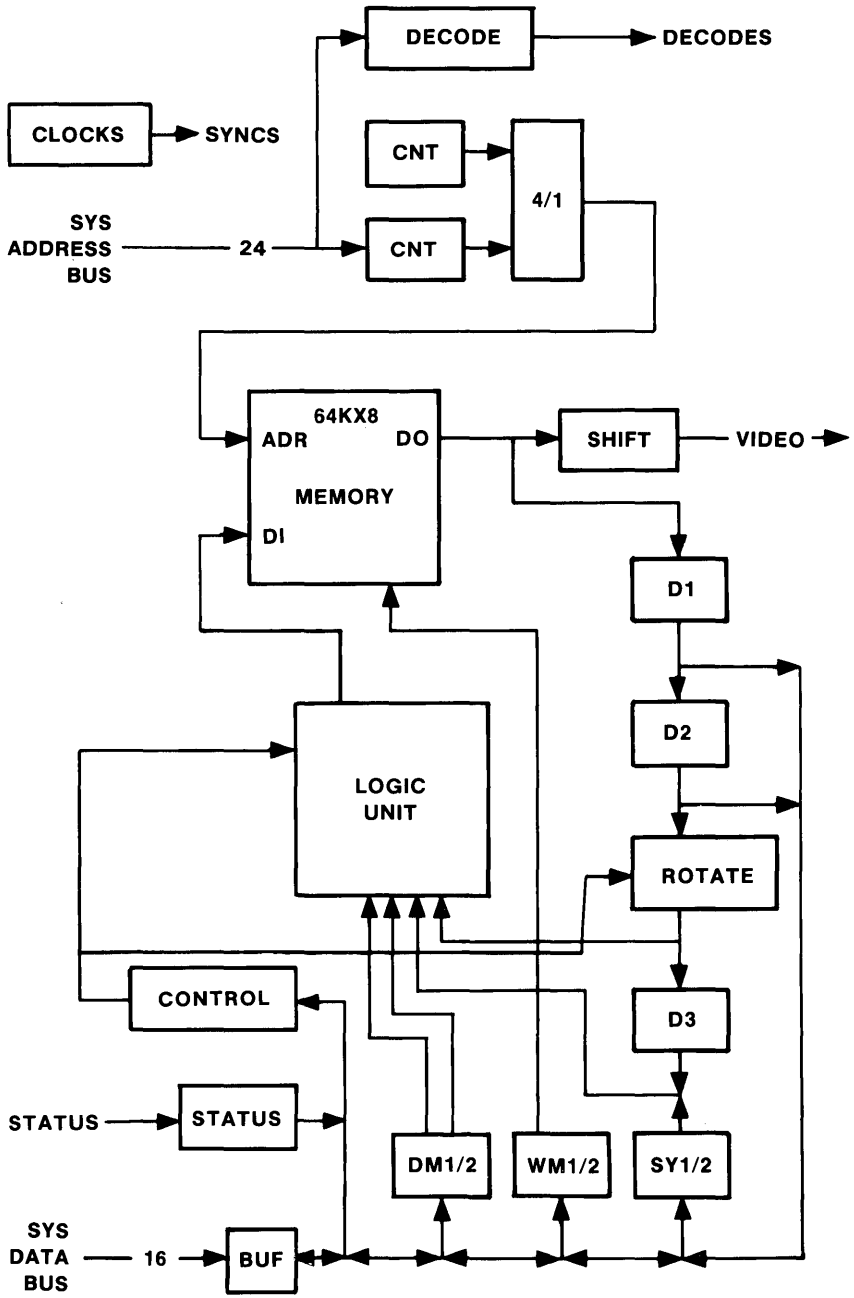


Figure 1. Advanced Monochrome Graphics Display Adapter Functional Block Diagram

Operation

Data to be displayed is written into the 64K x 8 byte bit map memory by the system. This data is then scanned out of the memory and sent as a serial video stream to the monitor.

The system processor can manipulate data residing in the bit map memory in several ways. The choice of a particular method is controlled by the type of instruction (Load or Store) and the current value of a 2-bit mode field which resides within a 16-bit control register. The contents of the control register must be initialized prior to the memory operation.

Hardware on the card allows data in the bit map to be manipulated without passing through a processor. This hardware includes:

- A 16-bit write mask that enables write operations to individual bits
- A shifter that realigns bits before they are written to memory
- Data registers to hold data being processed
- Two 8-bit data mask registers that select bits being merged from data registers
- A logic unit that does all the actual merging.

Using this hardware, source data from either the system or from the bit map may replace or be merged with data in the bit map. Images on the screen (in the bit map) can thus be moved, inverted, overlaid, or replaced.

The adapter occupies two address ranges on the I/O channel. Sixteen bytes of I/O address space at X '0160' through '016F' are used to load the data mask register, and the control registers. See the section entitled "I/O Operations" on page 6 for specific register addresses and bit assignments.

The second address range occupied by the adapter is the bit map memory that consists of 128K-bytes starting at hex address 'D00000' in memory space on the I/O channel. When addressing the bit map memory, the low order address bit must be 0. Address bits A01 through A16 are used to access memory. Address bit A00 is not used during memory accesses. This restriction arises because the adapter allows word accesses to the bit map to begin on either even or odd byte boundaries. Because the system restricts half-word accesses to even bytes, address lines to the bit map memory are offset by one bit.

The most significant data byte of the I/O channel is written into or read from the bit map memory location specified by the shifted address lines. The least significant data byte is written into or read from the memory location specified by the shifted address lines increased or decreased by one in the X or Y direction. The increase and decrease bits of the control register determine whether the address is increased or decreased. The X and Y bits of the control register determine whether the X or Y direction is changed.

As a result of the addressing scheme described above, each physical memory location in the bit map can be accessed at either of two addresses: either directly, through the most significant byte, or after the address bus has been increased or decreased, through the least significant byte.

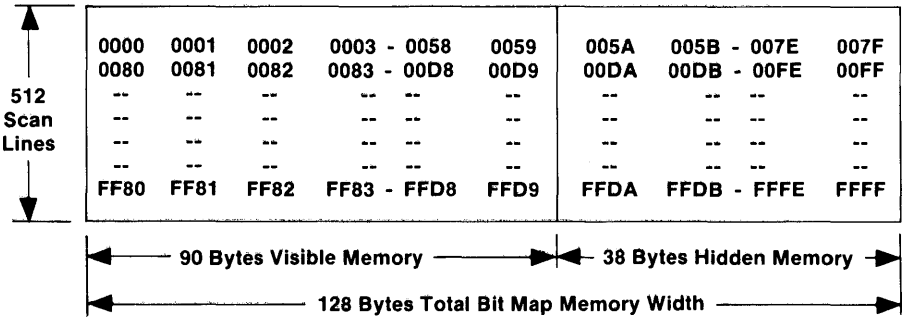


Figure 2. Bit Map Memory Addresses

The mapping between the physical addresses shown in Figure 2 and denoted as 'P' in the equations and the I/O channel address denoted as 'C' in the equations shown as follows for each of the increase and decrease bits and the X and Y status bits.

I/O Channel Byte	Inc/Dec	Word Storage Location X/Y	Physical to I/O Channel to Physical		I/O Channel to Physical
MS Byte	either	either	C=2P	=>	P=C/2
LS Byte	+	X	C=2P-2	=>	P=C/2 +1
LS Byte	-	X	C=2P+2	=>	P=C/2 -1
LS Byte	+	Y	C=2P-256	=>	P=C/2 +128
LS Byte	-	Y	C=2P+256	=>	P=C/2 -128

Figure 2 shows how on-card physical memory byte addresses are mapped to the display screen. The bits in each byte are shifted onto the screen with the most significant bit to the left.

Bit Map Memory Operations

The possible memory operations and respective mode bit definitions are as follows:

System Write Operation (Mode = 00)

This operation writes 16 bits of external data to the bit map memory at the address specified in the system processor 'Store' instruction.

Overlay Write Operation (Mode = 01)

This operation writes 16 bits of external data to the on-card write mask registers (WM1 and WM2). It also initiates a write operation of the on-card system data latches to the bit map memory at the address specified in the system processor 'Store' instruction. The new value in the write mask is used to control the write to the bit map. This operation will not affect the contents of the system data latches. Preloading the system data latches with the proper constant makes it possible to 'AND' or 'OR' system data with the bit map memory in one memory operation.

Adapter Write Operation (Mode = 10)

This operation writes data from the on-card data latches (D1,D2,D3) to the bit map memory making use of the rotate count, write mask, data mask, data bus and the logic unit. The following operations must be completed prior to executing this operation:

- Load the data latches using an adapter read operation.
- Load the write mask using an overlay write operation.
- Load the operation mode, rotate count and logic function using an I/O 'Store' to the adapter control register.

System Read Operation (Mode = XX)

This operation reads 16 bits of data from the bit map memory and places it on the system data bus. On-card data latches D1, D2 and D3 are updated by this operation. Data is loaded into latches D1, D2 and D3 regardless of the setting of the mode. However, valid data is returned to the system processor only if the mode is '00'.

Automatic Read/Write Operation (Mode = 11)

In this mode of operation, data is alternately read into D1, D2 and D3 and then written to memory with successive write operations from the microprocessor. Data read from memory is not gated onto any busses external to the data path, but is used only to update D1, D2 and D3. This allows the relatively fast 'Store' operation to be used for block transfer operations with no 'Loads' required.

To get the alternate read/write operations into phase with the source and destination addresses, the system processor should do a 'Load' operation from the first source address. Subsequent 'Stores' will automatically alternate between read and writes to the bit map.

I/O Operations

In order to manipulate the various registers on the adapter, a set of I/O operations are required.

Data Mask Register (X'0162')

Typically these two 8-bit registers (DM1 and DM2) are initialized with data representing the inverse of each other, and are used to mask the bits on or off for logical combination through the logic unit.

- An 8 bit for DM1 Mask (Data Bus MS Byte, bits 0-7)
- An 8 bit for DM2 Mask (Data Bus LS Byte, bits 0-7).

Data Control Register (X'0160') Write Only

Bits 0 through 11 should not be changed for at least 1.4 microseconds after a memory operation since the memory operation may still be in progress.

Bits 0-2 Rotate Count

The rotate count determines the number of bits that the data read from the bit map is shifted to the left prior to being written back into the bit map. This value has no effect during system write or overlay write modes. It has effect only during adapter write or automatic write modes.

Bits 3-5 Logic Unit Function Control

Bit
543
---
000 - Pass through B
010 - Pass through A
100 - Pass through 'Not' B
101 - A 'OR' B
110 - Pass through 'Not' A
111 - A 'NOR' B

Logic unit function control bits (bits 3-5) determine how data previously read from the bit map or written from the system are merged before being written into the bit map. When set to '101'B, the bit fields masked off by the data mask register are merged with system data before being written back into the bit map memory. This is the function normally needed during adapter write and automatic write modes. When set to '010'B, the bit field masked off by data mask register 1 (DM1) is written into the bit map memory. This is the function normally needed in system write and overlay write modes. When set to '000'B, the bit field masked off by data mask register 2 (DM2) is written into the bit map memory. This function can be used in adapter write and automatic write modes when no bit shift is used. In each case, an inverting form of the operation exists to allow data to be inverted as it is written to the bit map. See "Bit Map Memory Operations" on page 5 for further explanations.

Bits 6-7 Reserved

Bits 8-9 Memory Mode

Bit
98
--
00 = System write operation
01 = Overlay write operation
10 = Adapter write operation
11 = Automatic read/write operation

Bit 10 Address Counter Mode

0 = Increase the address counter
1 = Decrease the address counter

Bit 11 Address Counter Stepping

0 = Y stepping
1 = X stepping

Bits 10 and 11 control whether the LSB of a bit map memory operation is accessed from the left, right, above or below the MSB. See "Bit Map Memory Operations" on page 5 for further explanation.

Bit 12 Block transfer

Bit 12 controls block transfer mode. When this bit is set from a 0 to a 1, the address of the next memory location accessed is stored in a pointer register on the card. This address is adjusted to point at the next memory location at the end of each memory cycle. That is, the X or Y address is increased or decreased. After the first block transfer memory cycle has loaded the on-card pointer register, the memory address supplied on the data bus is ignored, and the pointer is used to access memory. To reload the pointer register, either clear and set the block transfer bit again or read from the block transfer reload I/O address, X'0162'.

Bit 13 Interrupt enable

If bit 13 is 1, an interrupt is generated at the start of vertical sync. If this bit is 0, the interrupt is not generated. See "Interrupts" on page 9.

Bit 14 Sync enable (always 1)

Bit 15 Video enable

0 = Disables video to monitor
1 = Enables video to monitor

RAS Status Register (X'0160') Read Only

Bit 0 1 = Even 0 = Odd field

Bit 1 Horizontal sync toggle latch

Bit 2 0 = Vertical sync inactive 1= Vertical sync active

Bit 3 Serialized Video

Bit 4 1= X 0 = Y Stepping

Bit 5 1 = Interrupt pending

Bit 6 0 = Increase 1 = Decrease the address counter

Bit 7 1 = Enable load address

Block Transfer Reload (X'0162') Read Only

Reading from this I/O location loads the address of the next memory access into the on-card address pointer register. No meaningful data is returned when this location is read. See "Data Control Register (X'0160') Write Only" on page 6 for more information regarding the use of this I/O command.

Interrupts

The adapter generates a level 11 interrupt at the start of vertical sync if bit 13 (interrupt enable) of the control register is a 1. This interrupt will not occur if bit 13 is a 0. When the adapter generates an interrupt, bit 5 of the RAS status register is set to 1. To clear bit 5 and reenable level 11 interrupts, an output to hex 6F3 with any data value must be issued.

Specifications

The adapter has a 16-pin connector at the rear of the adapter. The following figure shows the signals and their pin assignments.

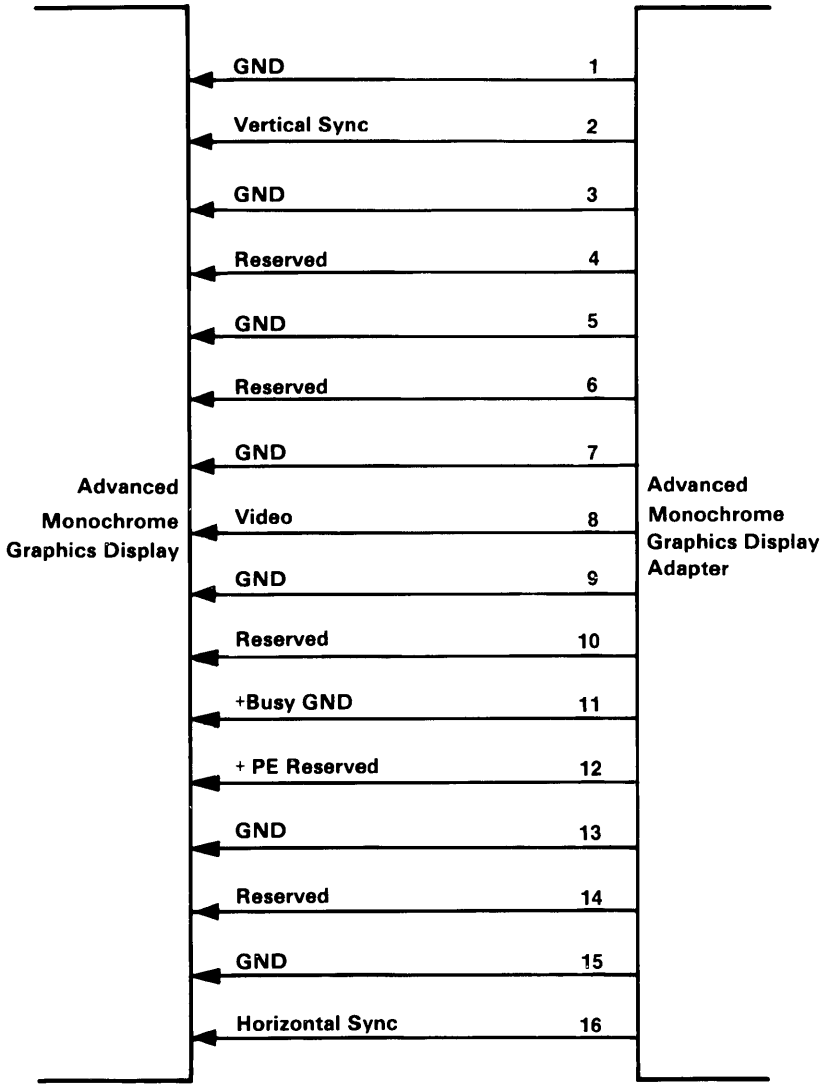
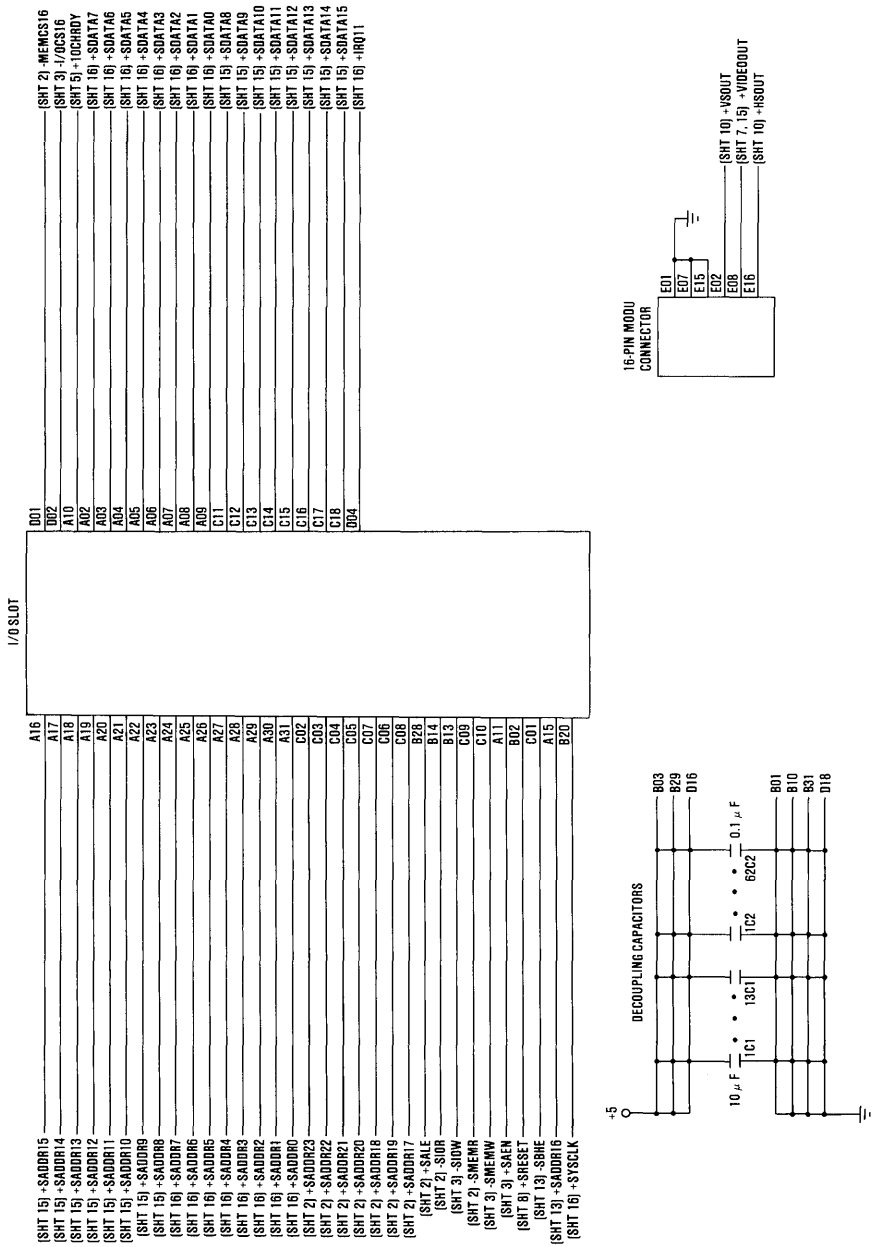


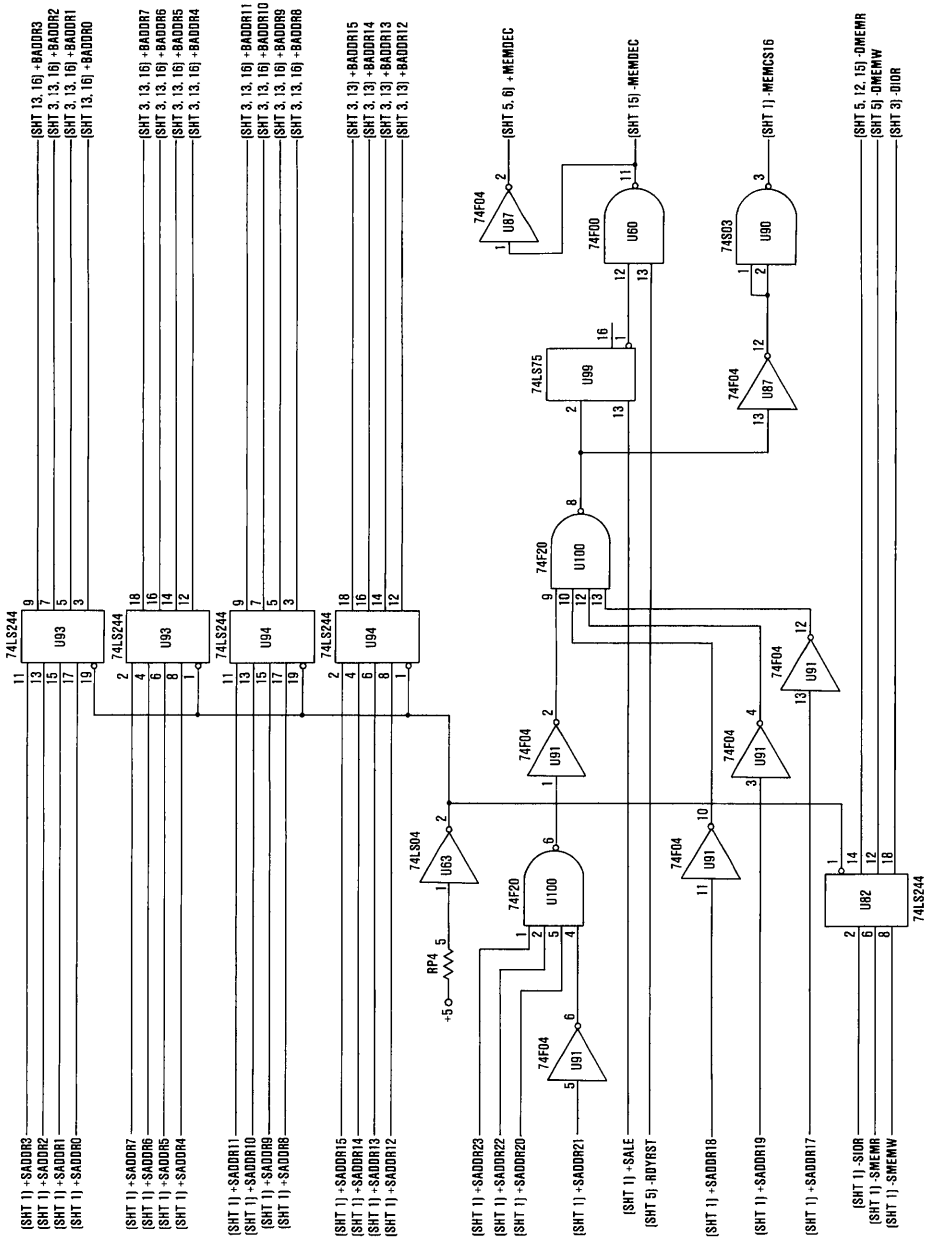
Figure 3. Advanced Monochrome Graphics Display Adapter Interface Specifications

Logic Diagrams

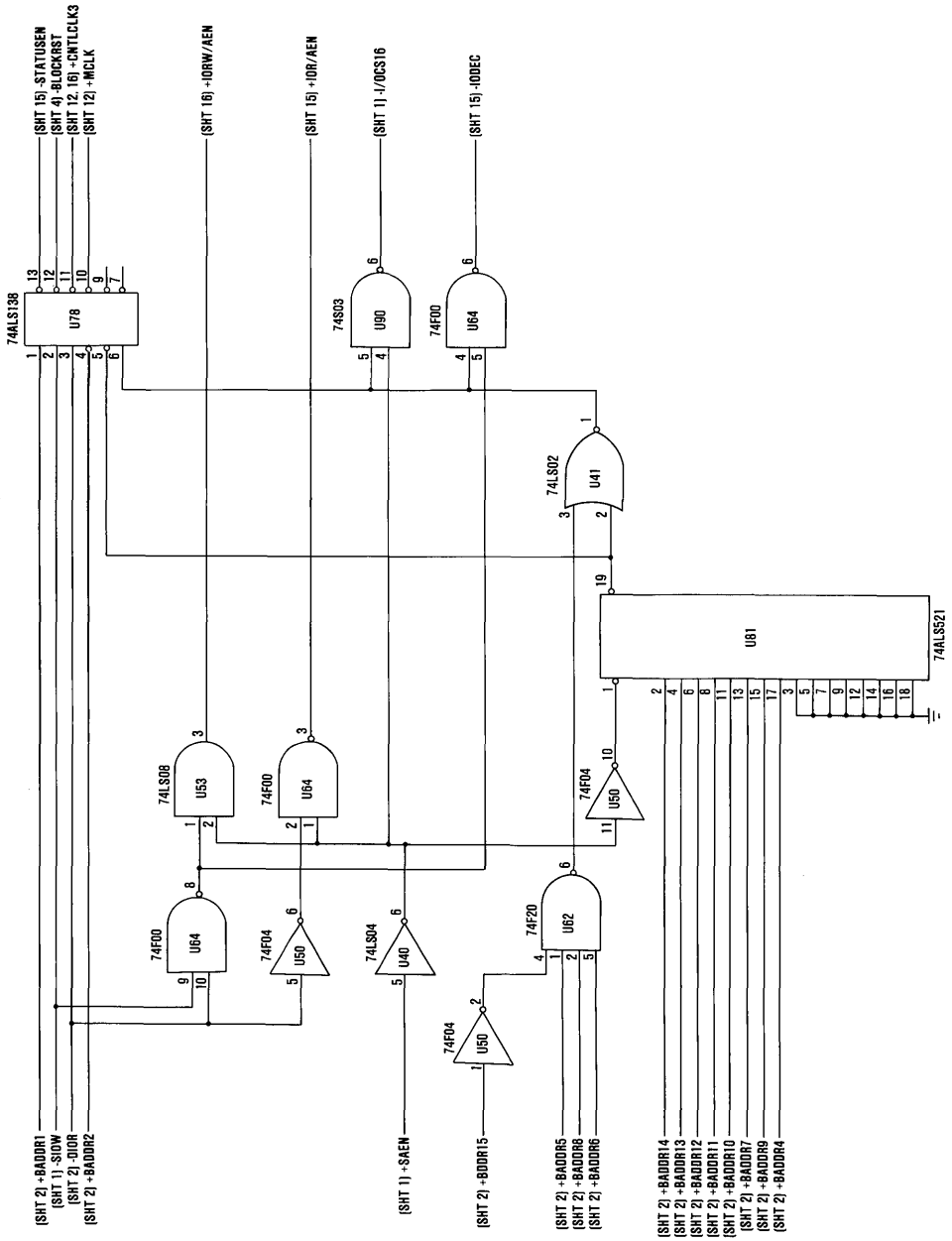
Sheet 1 of 16



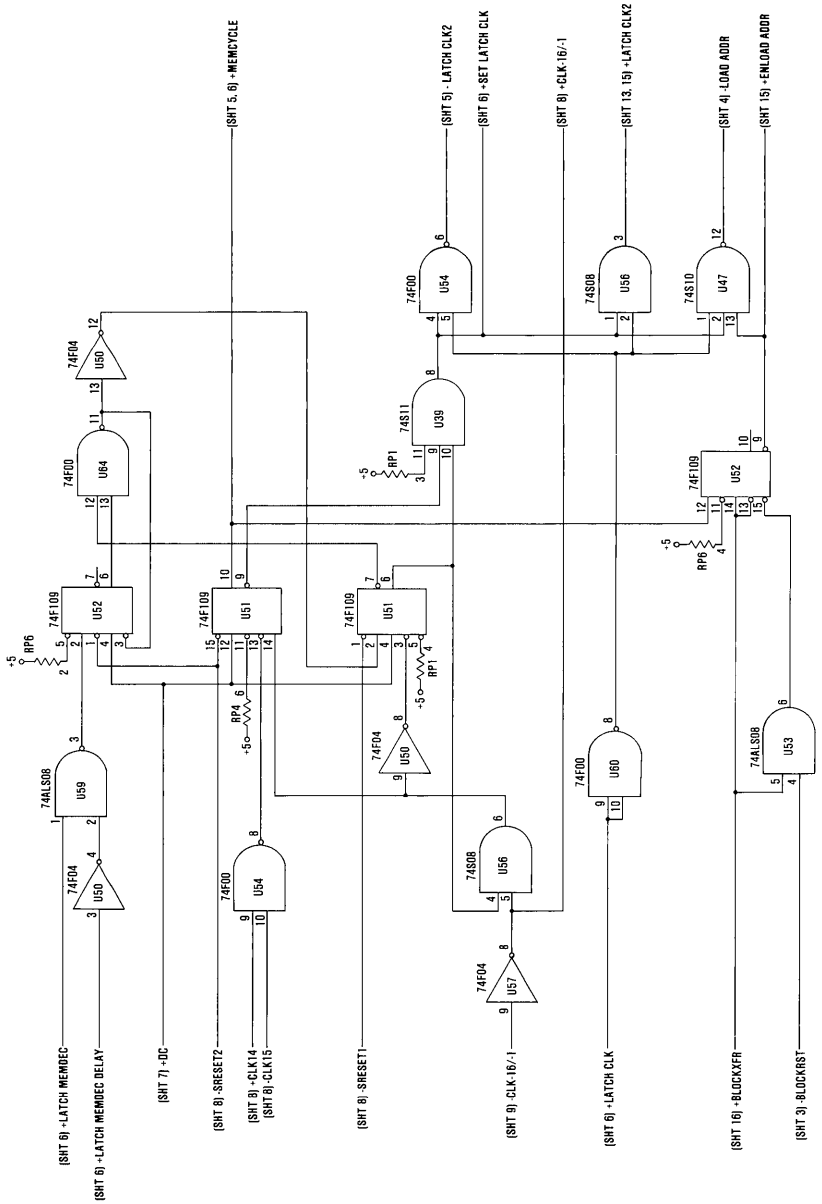
Sheet 2 of 16


74LS244
U93
U94
U82
74F20
U100
74F04
U91
74LS04
U63
74LS75
U99
74F00
U60
74S03
U90
RP4
+5
(SHT 13, 16) +BADDR3
(SHT 3, 13, 16) +BADDR2
(SHT 3, 13, 16) +BADDR1
(SHT 13, 16) +BADDR0
(SHT 3, 13, 16) +BADDR7
(SHT 3, 13, 16) +BADDR6
(SHT 3, 13, 16) +BADDR5
(SHT 3, 13, 16) +BADDR4
(SHT 3, 13, 16) +BADDR11
(SHT 3, 13, 16) +BADDR10
(SHT 3, 13, 16) +BADDR9
(SHT 3, 13, 16) +BADDR8
(SHT 3, 13) +BADDR15
(SHT 3, 13) +BADDR14
(SHT 3, 13) +BADDR13
(SHT 3, 13) +BADDR12
(SHT 5, 6) +MEMDEC
(SHT 15) -MEMDEC
(SHT 1) -MEMCS16
(SHT 5, 12, 15) -DMEMR
(SHT 5) -DMEMW
(SHT 3) -DIOR
(SHT 1) +SADDR3
(SHT 1) +SADDR2
(SHT 1) +SADDR1
(SHT 1) +SADDR0
(SHT 1) +SADDR7
(SHT 1) +SADDR6
(SHT 1) +SADDR5
(SHT 1) +SADDR4
(SHT 1) +SADDR11
(SHT 1) +SADDR10
(SHT 1) +SADDR9
(SHT 1) +SADDR8
(SHT 1) +SADDR15
(SHT 1) +SADDR14
(SHT 1) +SADDR13
(SHT 1) +SADDR12
(SHT 1) +SADDR23
(SHT 1) +SADDR22
(SHT 1) +SADDR20
(SHT 1) +SADDR21
(SHT 1) +SALE
(SHT 5) -RDYRST
(SHT 1) +SADDR18
(SHT 1) +SADDR19
(SHT 1) +SADDR17
(SHT 1) -SIOR
(SHT 1) -SMEMR
(SHT 1) -SMEMW

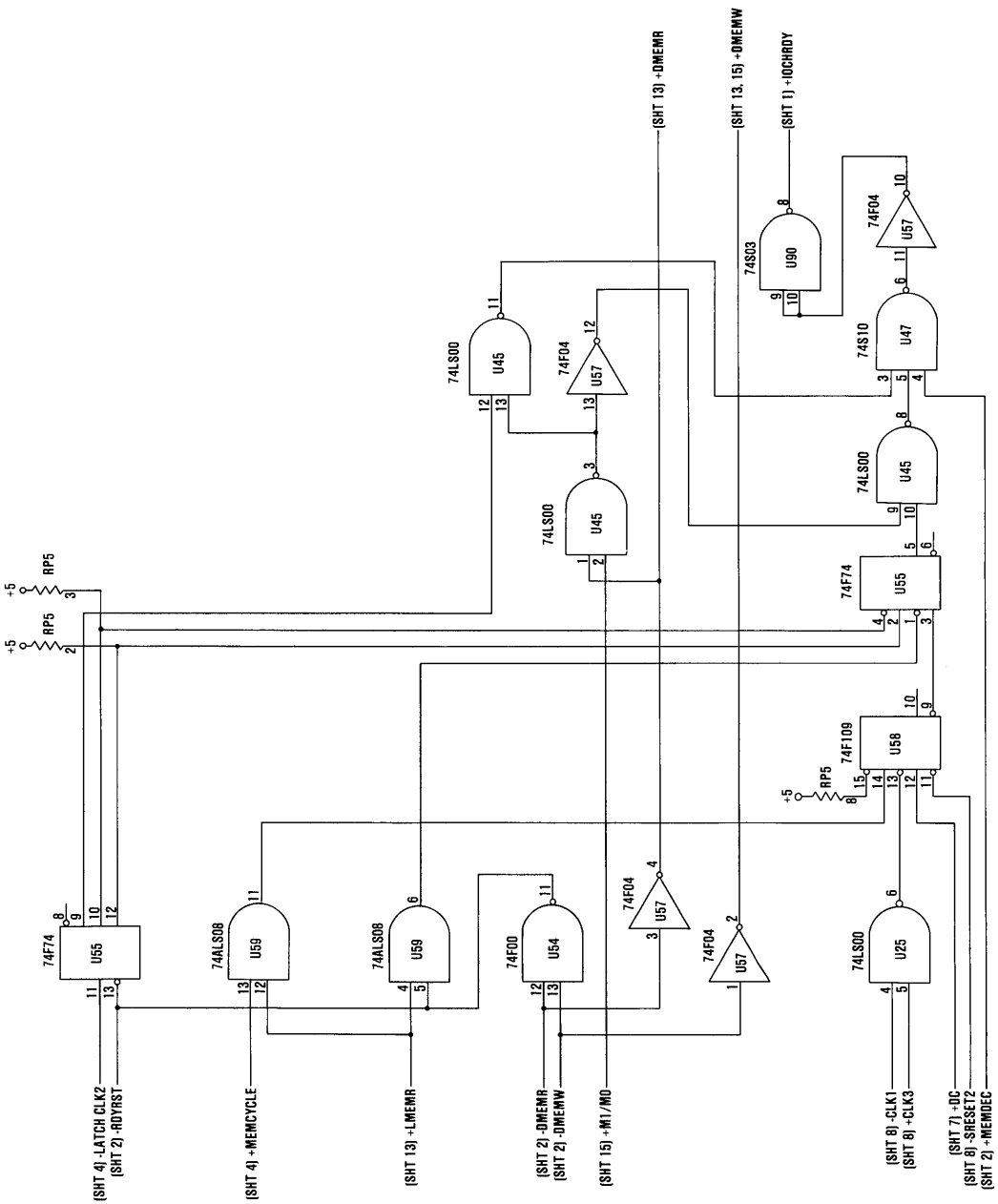
Sheet 3 of 16


74ALS138
U78
(SHT 15) -STATUSEN
(SHT 4) -BLOCKRST
(SHT 12, 16) +CNTLCLK3
(SHT 12) +MCLK
(SHT 16) +IORW/AEN
(SHT 15) +IOR/AEN
(SHT 1) -I/OCS16
(SHT 15) -IODEC
74S03
U90
74F00
U64
74LS02
U41
U81
74ALS521
74LS08
U53
74F00
U64
74F04
U50
74F00
U64
74F04
U50
74LS04
U40
74F20
U62
74F04
U50
(SHT 2) +BADDR1
(SHT 1) -SIOW
(SHT 2) -DIOR
(SHT 2) +BADDR2
(SHT 1) +SAEN
(SHT 2) +BDDR15
(SHT 2) +BADDR5
(SHT 2) +BADDR8
(SHT 2) +BADDR6
(SHT 2) +BADDR14
(SHT 2) +BADDR13
(SHT 2) +BADDR12
(SHT 2) +BADDR11
(SHT 2) +BADDR10
(SHT 2) +BADDR7
(SHT 2) +BADDR9
(SHT 2) +BADDR4

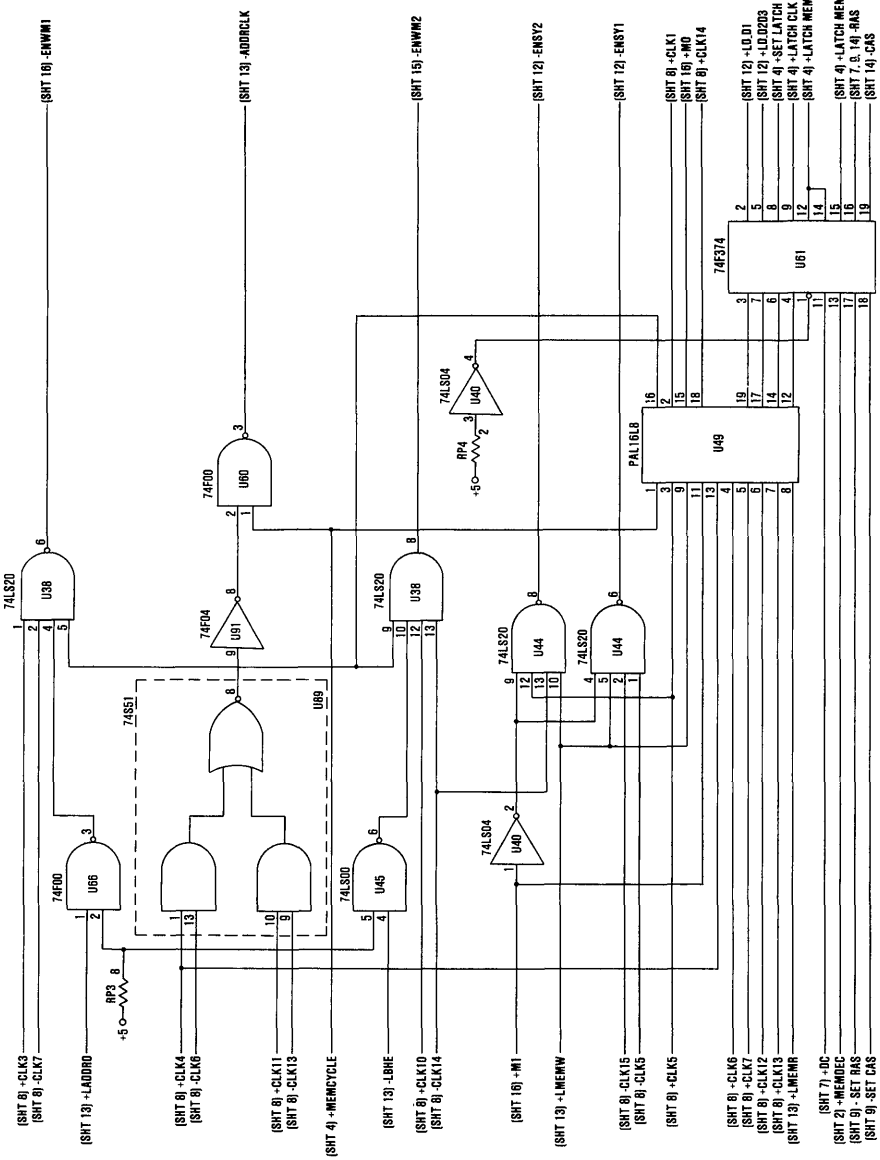
Sheet 4 of 16


[SHT 5, 6) +MEMCYCLE
[SHT 5) - LATCH CLK2
[SHT 6) +SET LATCH CLK
[SHT 8) +CLK-16/-1
[SHT 13, 15) +LATCH CLK2
[SHT 4) -LOAD ADDR
[SHT 15) +ENLOAD ADDR
74F04 U60
74F00 U64
74S11 U39
74F00 U54
74S08 U56
74S10 U47
74F109 U52
74F109 U51
74F109 U51
RP1
RP6
RP4
RP1
74ALS08 U59
74F04 U50
74F00 U54
74F04 U50
74S08 U56
74F04 U57
74F00 U60
74ALS08 U53
[SHT 6) +LATCH MEMDEC
[SHT 6) +LATCH MEMDEC DELAY
[SHT 7) +DC
[SHT 8) -SRESET2
[SHT 3) +CLK14
[SHT 8) -CLK15
[SHT 8) -SRESET1
[SHT 9) -CLK-16/-1
[SHT 6) +LATCH CLK
[SHT 16) +BLOCKXFR
[SHT 3) BLOCKRST

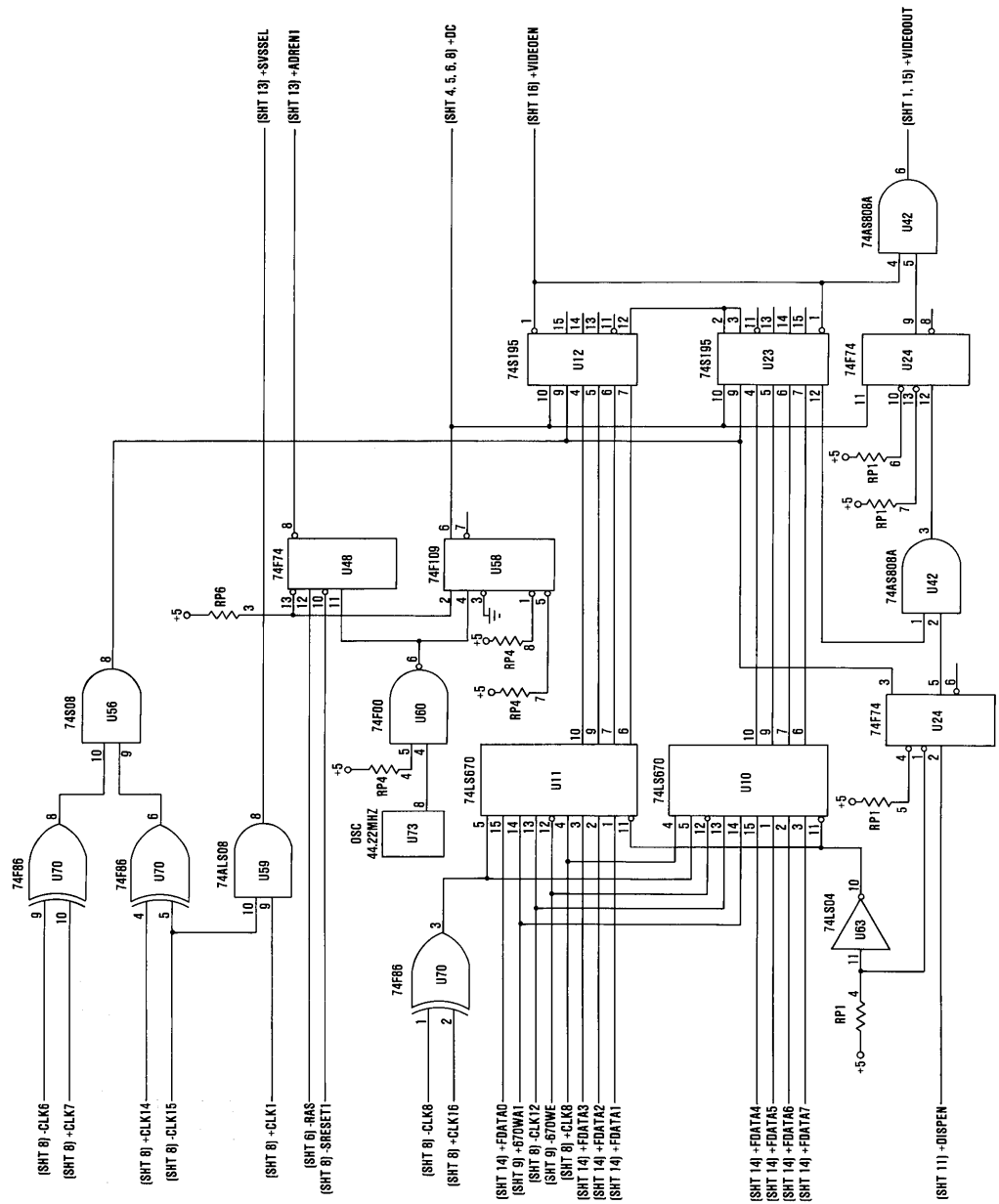
Sheet 5 of 16


(SHT 13) +DMEMR
(SHT 13, 15) +DMEMW
(SHT 1) +IOCHRDY
74S03 U90
74F04 U57
74S10 U47
74LS00 U45
74F109 U58
74F74 U55
RP5
+5
74ALS08 U59
74F00 U54
74LS00 U25
(SHT 4) -LATCH CLK2
(SHT 2) -RDYRST
(SHT 4) +MEMCYCLE
(SHT 13) +LMEMR
(SHT 2) -DMEMR
(SHT 2) -DMEMW
(SHT 15) +M1/MIO
(SHT 8) -CLK1
(SHT 8) +CLK3
(SHT 7) +DC
(SHT 8) -SRESET2
(SHT 2) +MEMDEC

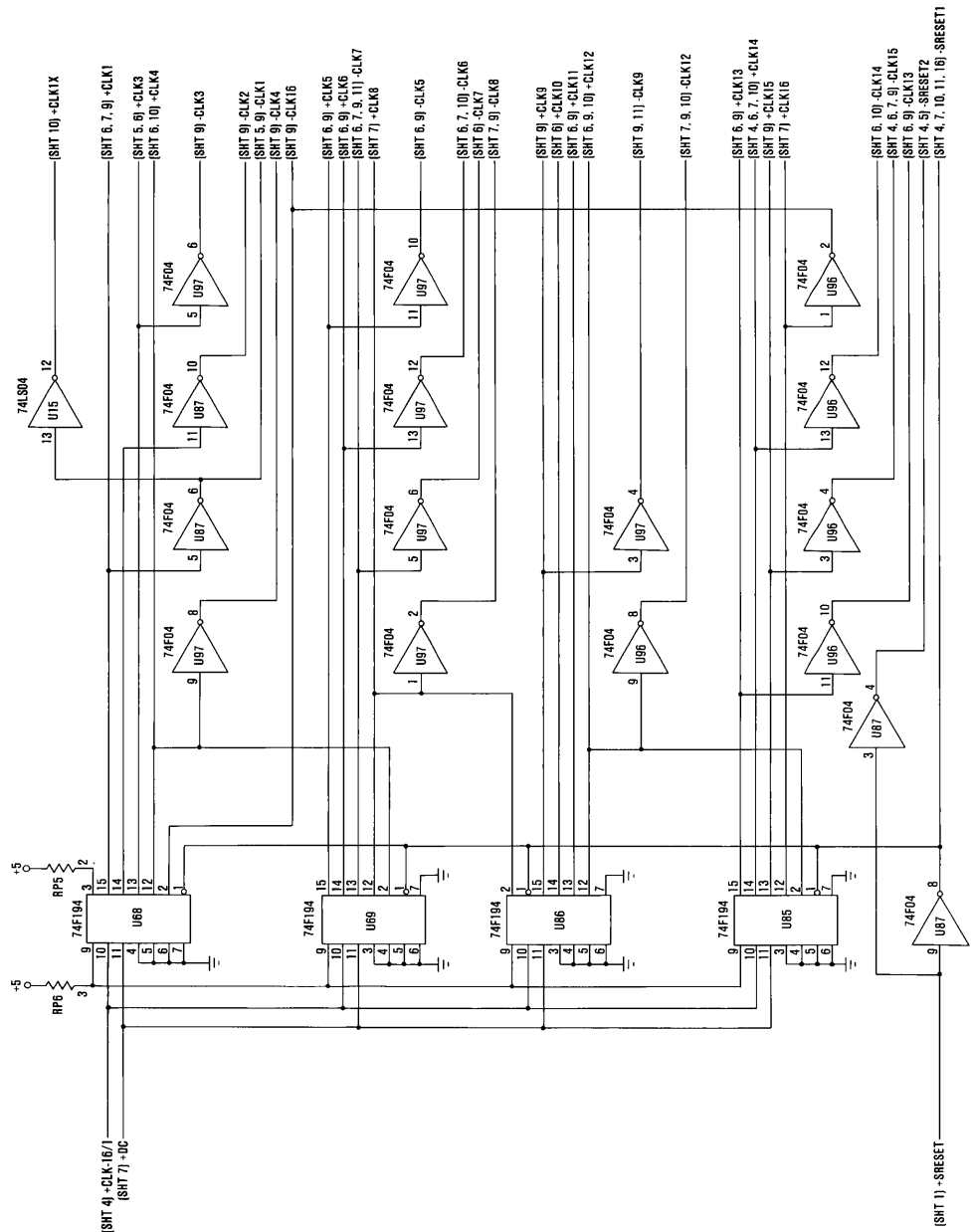
Sheet 6 of 16


(SHT 16) -ENWMI
(SHT 13) -ADDRCLK
(SHT 15) -ENWMI2
(SHT 12) -ENSY2
(SHT 12) -ENSY1
(SHT 8) +CLK1
(SHT 16) +MIO
(SHT 8) +CLK14
(SHT 12) +LD.DI
(SHT 12) +LD.D2D3
(SHT 4) +SET.LATCH.CLK
(SHT 4) +LATCH.CLK
(SHT 4) +LATCH.MEMDEC
(SHT 4) +LATCH.MEMDEC.DELAY
(SHT 7, 9, 14) -RAS
(SHT 14) -CAS
74LS20
U38
74F00
U66
74F00
U60
74F04
U91
74S51
U89
74LS00
U45
74LS04
U40
RP4
RP3
+5
74LS20
U44
74LS20
U44
74LS04
U40
PAL16L8
U49
74F374
U61
(SHT 8) +CLK3
(SHT 8) -CLK7
(SHT 13) +LADDRD
(SHT 8) -CLK4
(SHT 8) -CLK6
(SHT 8) +CLK11
(SHT 8) -CLK13
(SHT 4) +MEMCYCLE
(SHT 13) -LBHE
(SHT 8) +CLK10
(SHT 8) -CLK14
(SHT 16) +MI
(SHT 13) +LMEMW
(SHT 8) -CLK15
(SHT 8) -CLK5
(SHT 8) +CLK5
(SHT 8) -CLK6
(SHT 8) +CLK7
(SHT 8) +CLK12
(SHT 8) +CLK13
(SHT 13) +LMEMR
(SHT 7) -DC
(SHT 2) +MEMDEC
(SHT 9) -SET.RAS
(SHT 9) -SET.CAS

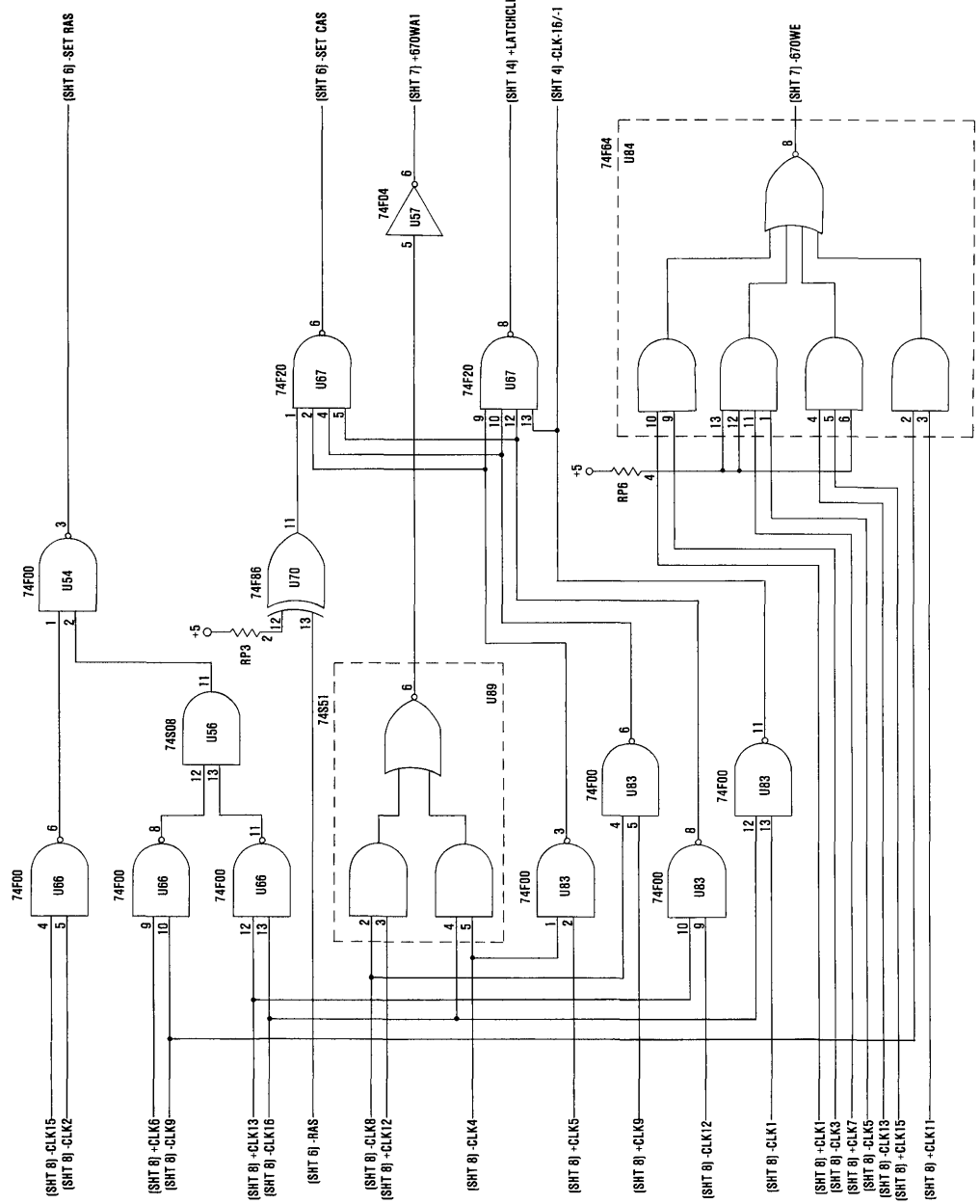
Sheet 7 of 16


[SHT 13] +SVSSEL
[SHT 13] +ADREN1
[SHT 4, 5, 6, 8] +DC
[SHT 16] +VIDEOEN
[SHT 1, 15] +VIDEOOUT
74S08 U56
74F86 U70
74F86 U70
74ALS08 U59
74F74 U48
74F109 U58
74F00 U60
OSC 44.22MHZ U73
74LS670 U11
74LS670 U10
74S195 U12
74S195 U23
74AS808A U42
74F74 U24
74LS04 U63
RP6
RP4
RP1
[SHT 8] -CLK6
[SHT 8] -CLK7
[SHT 8] +CLK14
[SHT 8] -CLK15
[SHT 8] +CLK1
[SHT 6] -RAS
[SHT 8] -SRESET1
[SHT 8] -CLK8
[SHT 8] +CLK16
[SHT 14] +FDATA0
[SHT 9] +670WA1
[SHT 8] -CLK12
[SHT 9] -670WE
[SHT 8] +CLK8
[SHT 14] +FDATA3
[SHT 14] +FDATA2
[SHT 14] +FDATA1
[SHT 14] +FDATA4
[SHT 14] +FDATA5
[SHT 14] +FDATA6
[SHT 14] +FDATA7
[SHT 11] +DISPEN

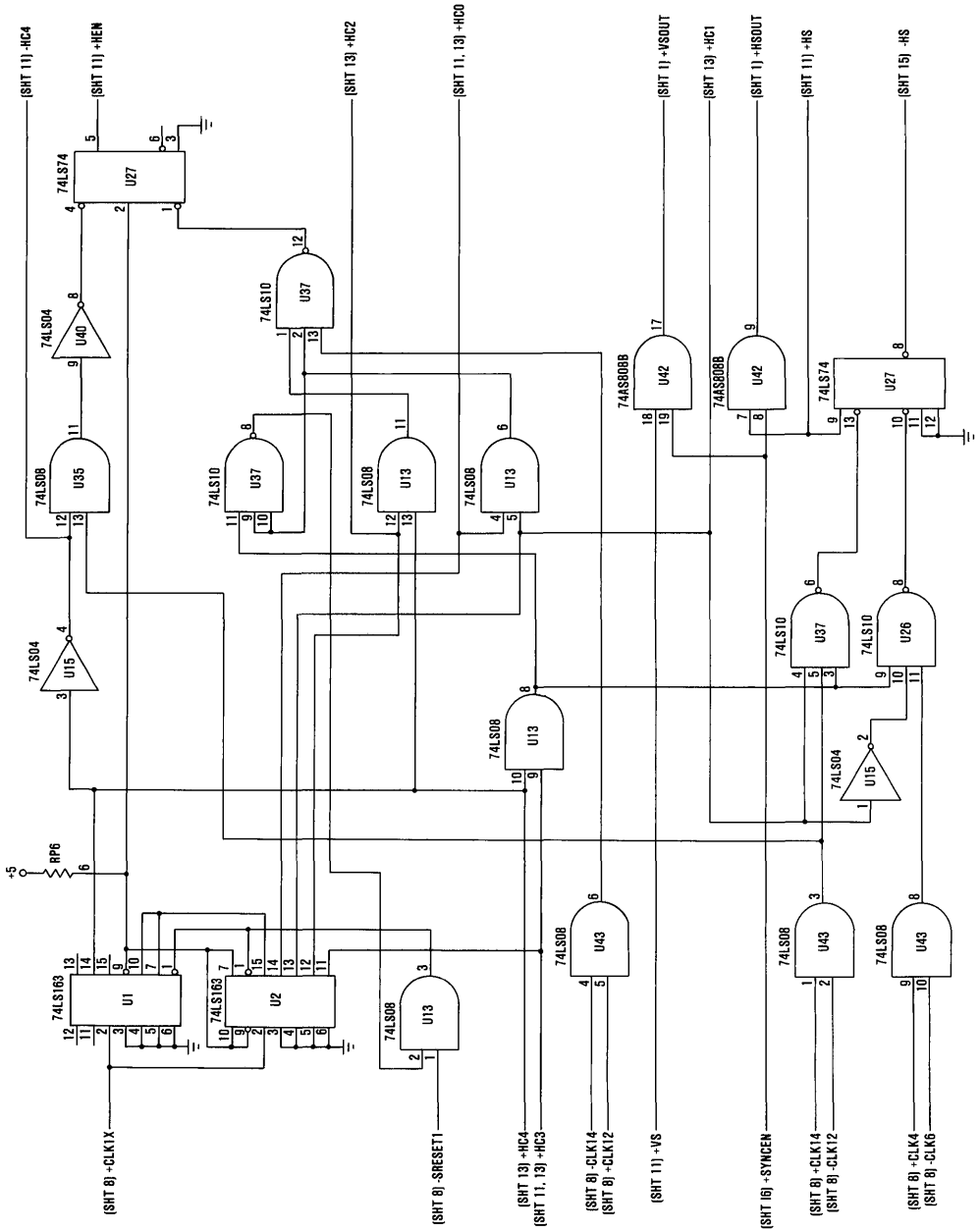
Sheet 8 of 16


74LS04
74F04
74F194
RP5
RP6
U15
U68
U69
U86
U85
U87
U96
U97
(SHT 10) +CLK1X
(SHT 6, 7, 9) +CLK1
(SHT 5, 9) +CLK3
(SHT 6, 10) +CLK4
(SHT 9) -CLK3
(SHT 9) -CLK2
(SHT 5, 9) -CLK1
(SHT 9) -CLK4
(SHT 9) -CLK16
(SHT 6, 9) +CLK5
(SHT 6, 9) +CLK6
(SHT 6, 7, 9, 11) -CLK7
(SHT 7) +CLK8
(SHT 6, 9) -CLK5
(SHT 6, 7, 10) -CLK6
(SHT 6) -CLK7
(SHT 7, 9) -CLK8
(SHT 9) +CLK9
(SHT 6) +CLK10
(SHT 6, 9) +CLK11
(SHT 6, 9, 10) +CLK12
(SHT 9, 11) -CLK9
(SHT 7, 9, 10) -CLK12
(SHT 6, 9) +CLK13
(SHT 4, 6, 7, 10) +CLK14
(SHT 9) +CLK15
(SHT 7) +CLK16
(SHT 6, 10) -CLK14
(SHT 4, 6, 7, 9) CLK15
(SHT 6, 9) -CLK13
(SHT 4, 5) -SRESET2
(SHT 4, 7, 10, 11, 16) -SRESET1
(SHT 4) +CLK-16/1
(SHT 7) +DC
(SHT 1) -SRESET
+5

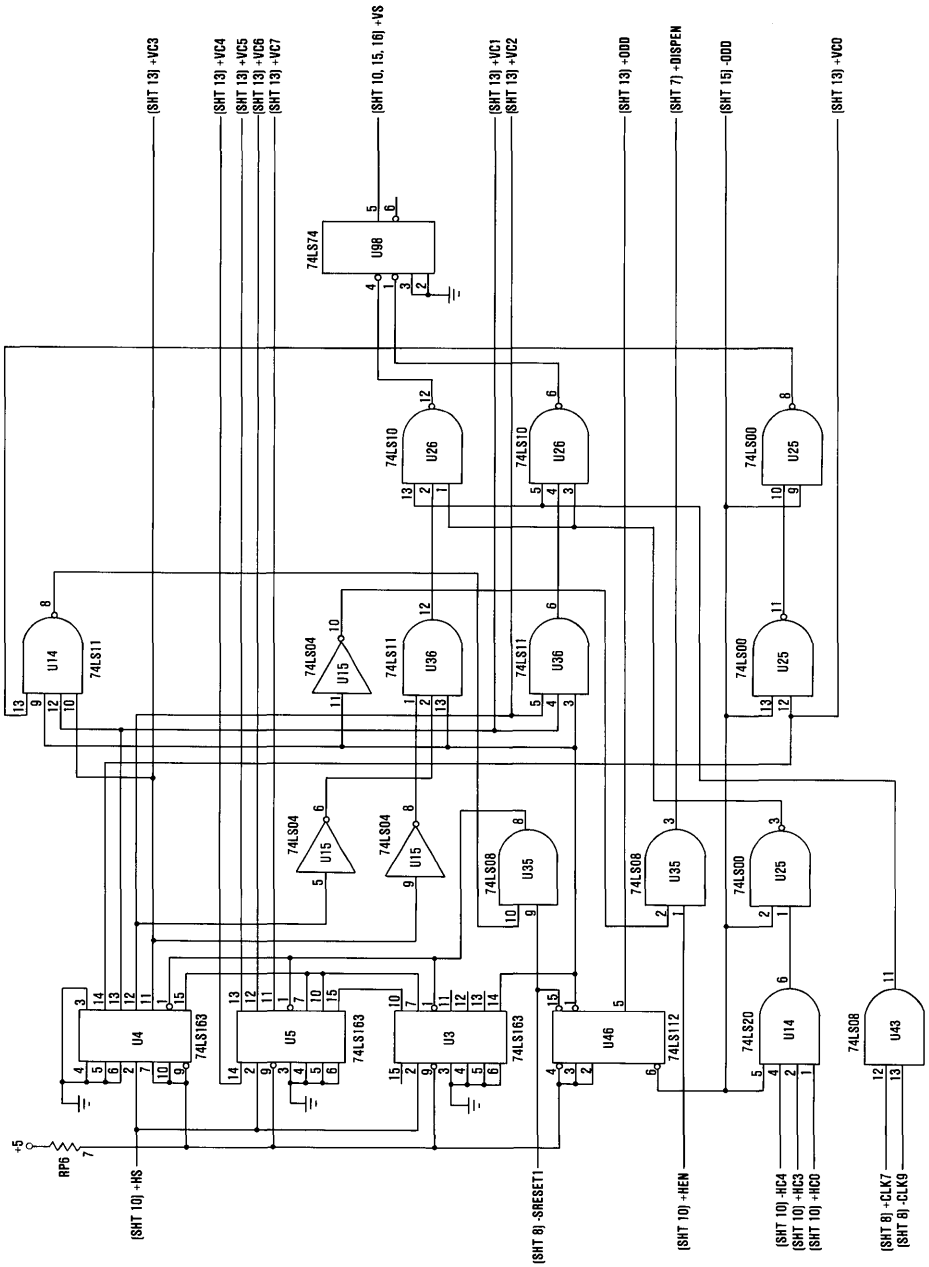
Sheet 9 of 16


(SHT 6) -SET RAS
(SHT 6) -SET CAS
(SHT 7) +670WA1
(SHT 14) +LATCHCLK1
(SHT 4) -CLK-16/-1
(SHT 7) -670WE
74F00 U54
74F86 U70
74F20 U67
74F20 U67
74F04 U57
74F64 U84
+5
RP3
RP6
74S08 U56
74S51
U89
74F00 U66
74F00 U66
74F00 U66
74F00 U83
74F00 U83
74F00 U83
(SHT 8) -CLK15
(SHT 8) -CLK2
(SHT 8) +CLK6
(SHT 8) -CLK9
(SHT 8) +CLK13
(SHT 8) -CLK16
(SHT 6) -RAS
(SHT 8) -CLK8
(SHT 8) +CLK12
(SHT 8) -CLK4
(SHT 8) +CLK5
(SHT 8) +CLK9
(SHT 8) -CLK12
(SHT 8) -CLK1
(SHT 8) +CLK1
(SHT 8) -CLK3
(SHT 8) +CLK7
(SHT 8) -CLK5
(SHT 8) -CLK13
(SHT 8) +CLK15
(SHT 8) +CLK11

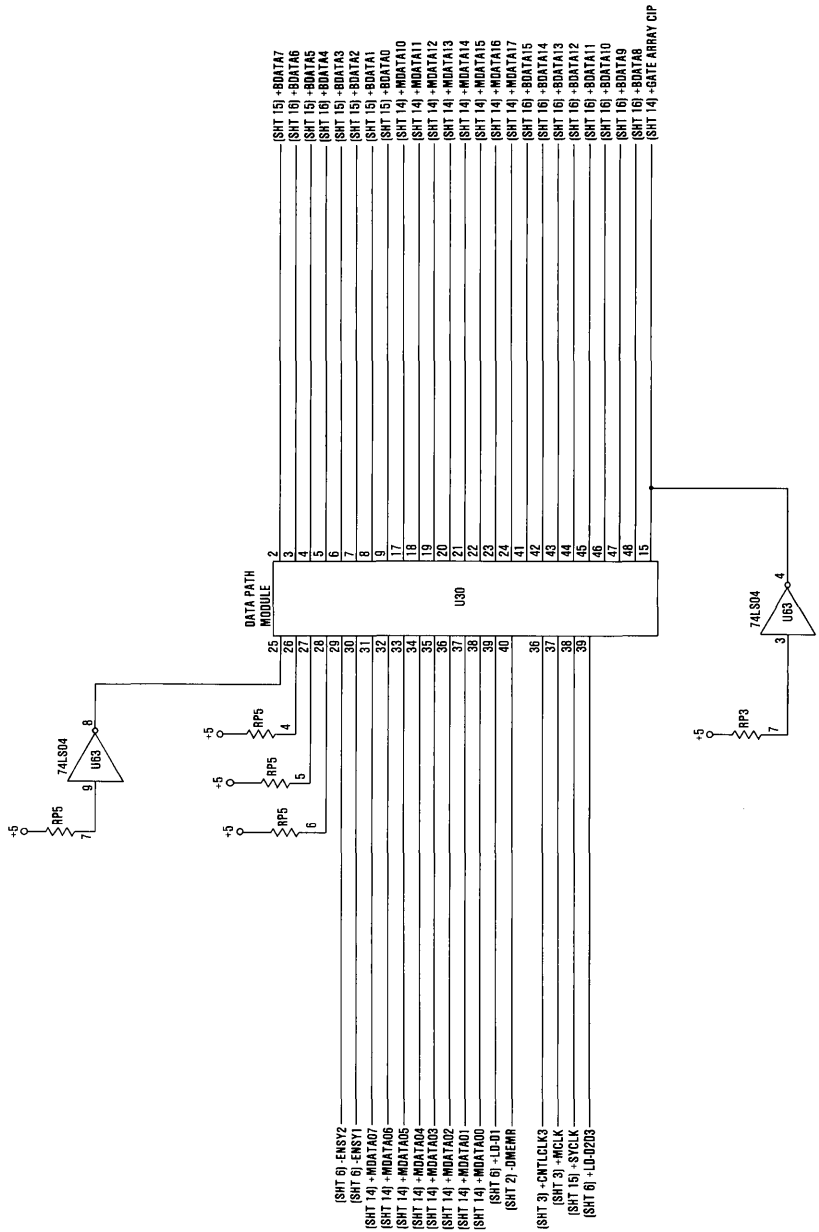
Sheet 10 of 16



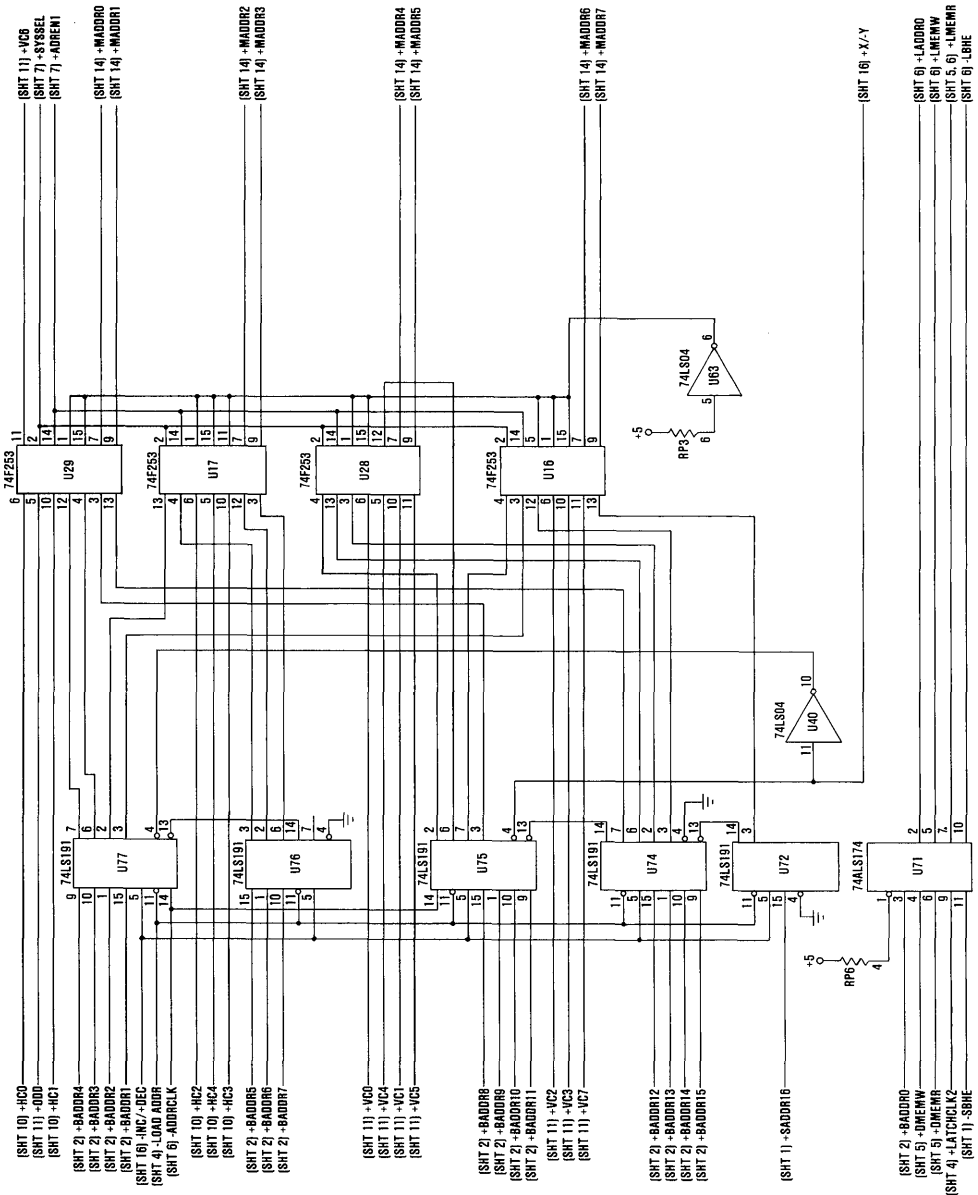
Sheet 11 of 16



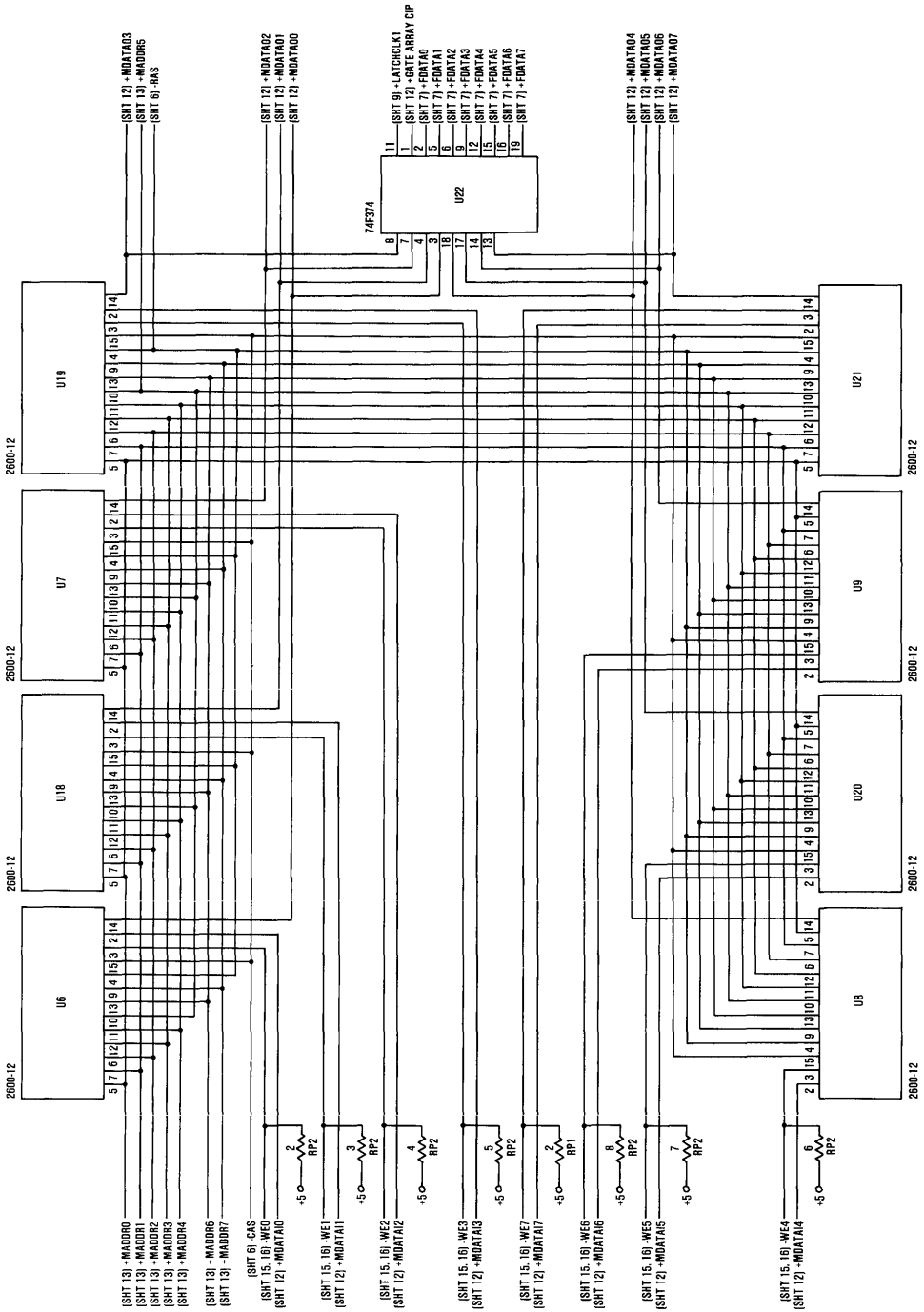
Sheet 12 of 16



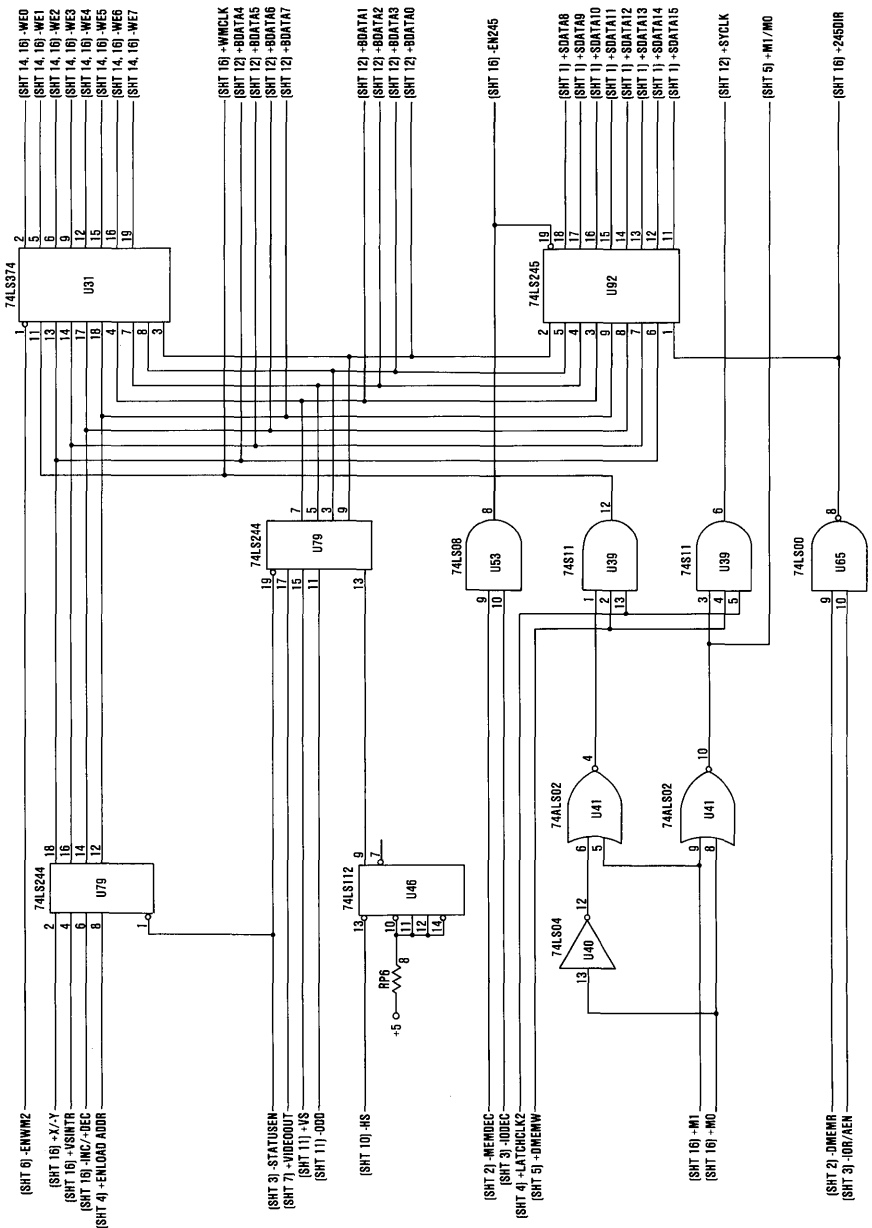
Sheet 13 of 16



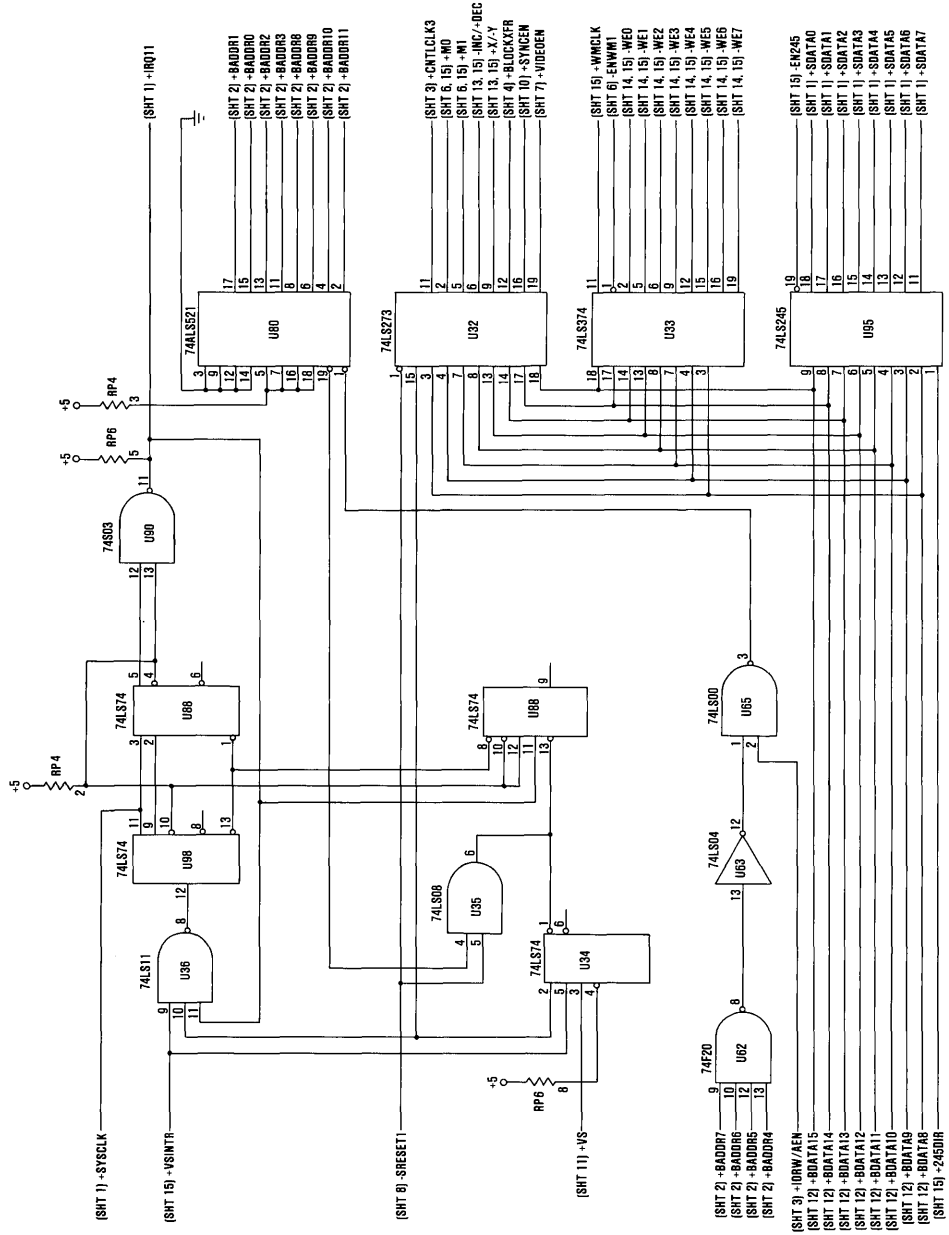
Sheet 14 of 16



Sheet 15 of 16



Sheet 16 of 16



IBM

Personal Computer Hardware Reference Library

Enhanced Graphics Adapter

Contents

Notes:

Description

The IBM Enhanced Graphics Adapter (EGA) is a graphics controller that supports both color and monochrome direct drive displays in a variety of modes. In addition to the direct drive port, a light pen interface is provided. Advanced features on the adapter include bit-mapped graphics in four planes and a RAM (Random Access Memory) loadable character generator. Design features in the hardware substantially reduce the software overhead for many graphics functions.

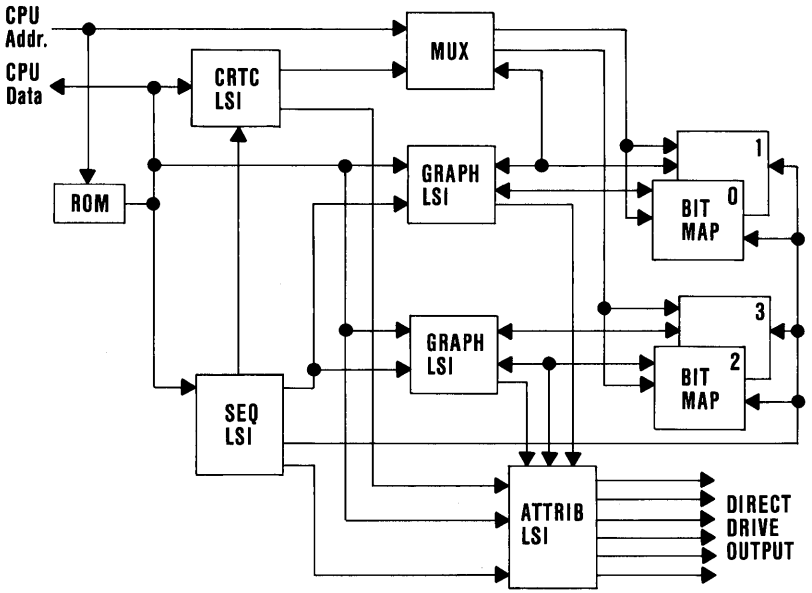
The Enhanced Graphics Adapter provides Basic Input Output System (BIOS) support for both alphanumeric (A/N) modes and all-points-addressable (APA) graphics modes, including all modes supported by the Monochrome Display Adapter and the Color/Graphics Monitor Adapter. Other modes provide APA 640x350 pel graphics support for the IBM Monochrome Display, full 16 color support in both 320x200 pel and 640x200 pel resolutions for the IBM Color Display, and both A/N and APA support with resolution of 640x350 for the IBM Enhanced Color Display. In alphanumeric modes, characters are formed from one of two ROM (Read Only Memory) character generators on the adapter. One character generator defines 7x9 characters in a 9x14 character box. For Enhanced Color Display support, the 9x14 character set is modified to provide an 8x14 character set. The second character generator defines 7x7 characters in an 8x8 character box. These generators contain dot patterns for 256 different characters. The character sets are identical to those provided by the IBM Monochrome Display Adapter and the IBM Color/Graphics Monitor Adapter.

The adapter contains 64K bytes of storage configured as four 16K byte bit planes. Memory expansion options are available to expand the adapter memory to 128K bytes or 256K bytes.

The adapter is packaged on a single 13-1/8 inch (333.50 mm) card. The direct drive port is a right-angle mounted connector at the rear of the adapter and extends through the rear panel of the system unit. Also on the card are five large scale integration (LSI) modules custom designed for this controller.

Located on the adapter is a feature connector that provides access to internal functions through a 32-pin berg connector. A separate 64-pin connector provides an interface for graphics memory expansion.

The following is a block diagram of the Enhanced Graphics Adapter:



Enhanced Graphics Adapter Block Diagram

Major Components

CRT Controller

The CRT (Cathode Ray Tube) Controller (CRTC) generates horizontal and vertical synchronous timings, addressing for the regenerative buffer, cursor and underline timings, and refresh addressing for the dynamic RAMs.

Sequencer

The Sequencer generates basic memory timings for the dynamic RAMs and the character clock for controlling regenerative memory fetches. It allows the processor to access memory during active display intervals by inserting dedicated processor memory cycles periodically between the display memory cycles. Map mask registers are available to protect entire memory maps from being changed.

Graphics Controller

The Graphics Controller directs the data from the memory to the attribute controller and the processor. In graphics modes, memory data is sent in serialized form to the attribute chip. In alpha modes the memory data is sent in parallel form, bypassing the graphics controller. The graphics controller formats the data for compatible modes and provides color comparators for use in color painting modes. Other hardware facilities allow the processor to write 32 bits in a single memory cycle, (8 bits per plane) for quick color presetting of the display areas, and additional logic allows the processor to write data to the display on non-byte boundaries.

Attribute Controller

The Attribute Controller provides a color palette of 16 colors, each of which may be specified separately. Six color outputs are

available for driving a display. Blinking and underlining are controlled by this chip. This chip takes data from the display memory and formats it for display on the CRT screen.

Display Buffer

The display buffer on the adapter consists of 64K bytes of dynamic read/write memory configured as four 16K byte video bit planes. Two options are available for expanding the graphics memory. The Graphics Memory Expansion Card plugs into the memory expansion connector on the adapter, and adds one bank of 16K to each of the four bit planes, increasing the graphics memory to 128K bytes. The expansion card also provides DIP sockets for further memory expansion. Populating the DIP sockets with the Graphics Memory Module Kit adds two additional 16K banks to each bit plane, bringing the graphics memory to its maximum of 256K bytes.

The address of the display buffer can be changed to remain compatible with other video cards and application software. Four locations are provided. The buffer can be configured at segment address hex A0000 for a length of 128K bytes, at hex A0000 for a length of 64K bytes, at hex B0000 for a length of 32K bytes, or at hex B8000 for a length of 32K bytes.

BIOS

A read-only memory (ROM) Basic Input Output System (BIOS) module on the adapter is linked to the system BIOS. This ROM BIOS contains character generators and control code and is mapped into the processor address at hex C0000 for a length of 16K bytes.

Support Logic

The logic on the card surrounding the LSI modules supports the modules and creates latch buses for the CRT controller, the

processor, and character generator. Two clock sources (14 MHz and 16 MHz) provide the dot rate. The clock is multiplexed under processor I/O control. Four I/O registers also resident on the card are not part of the LSI devices.

Modes of Operation

IBM Color Display

The following table describes the modes supported by BIOS on the IBM Color Display:

MODE #	TYPE	COLORS	ALPHA FORMAT	BUFFER START	BOX SIZE	MAX. PAGES	RESOLUTION
0	A/N	16	40x25	B8000	8x8	8	320x200
1	A/N	16	40x25	B8000	8x8	8	320x200
2	A/N	16	80x25	B8000	8x8	8	640x200
3	A/N	16	80x25	B8000	8x8	8	640x200
4	APA	4	40x25	B8000	8x8	1	320x200
5	APA	4	40x25	B8000	8x8	1	320x200
6	APA	2	80x25	B8000	8x8	1	640x200
D	APA	16	40x25	A8000	8x8	2/4/8	320x200
E	APA	16	80x25	A8000	8x8	1/2/4	640x200

Modes 0 through 6 emulate the support provided by the IBM Color/Graphics monitor Adapter.

Modes 0,2 and 5 are identical to modes 1,3 and 4 respectively at the adapter's direct drive interface.

The Maximum Pages fields for modes D and E indicate the number of pages supported when 64K, 128K or 256K bytes of graphics memory is installed, respectively.

IBM Monochrome Display

The following table describes the modes supported by BIOS on the IBM Monochrome Display.

MODE #	TYPE	COLORS	ALPHA FORMAT	BUFFER START	BOX SIZE	MAX. PAGES	RESOLUTION
7	A/N	4	80x25	B0000	9x14	8	720x350
F	APA	4	80x25	A0000	8x14	1/2	640x350

Mode 7 emulates the support provided by the IBM Monochrome Display Adapter.

IBM Enhanced Color Display

The Enhanced Graphics Adapter supports attachment of the IBM Enhanced Color Display. The IBM Enhanced Color Display is capable of running at the standard television frequency of 15.75 KHz as well as running 21.85 KHz. The table below summarizes the characteristics of the IBM Enhanced Color Display:

Parameter	TV Frequency	High Resolution
Horiz Scan Rate	15.75 KHz.	21.85 KHz.
Vertical Scan Rate	60 Hz.	60 Hz.
Video Bandwidth	14.318 MHz.	16.257 MHz.
Displayable Colors	16 Maximum	16 or 64
Character Size	7 by 7 Pels	7 by 9 Pels
Character Box Size	8 by 8 Pels	8 by 14 Pels
Maximum Resolution	640x200 Pels	640 by 350 Pels
Alphanumeric Modes	0,1,2,3	0,1,2,3
Graphics Modes	4,5,6,D,E	10

In the television frequency mode, the IBM Enhanced Color Display displays information identical in color and resolution to the IBM Color Display.

In the high resolution mode, the adapter provides enhanced alphanumeric character support. This enhanced alphanumeric support consists of transforming the 8 by 8 character box into an 8 by 14 character box, and providing 16 colors out of a palette of

64 possible display colors. Display colors are changed by altering the programming of the color palette registers in the Attribute Controller. In alphanumeric modes, any 16 of 64 colors are displayable. the screen resolution is 320x350 for modes 0 and 1, and 640x350 for modes 2 and 3.

The resolution displayed on the IBM Enhanced Color Display is selected by the switch settings on the Enhanced Graphics Adapter.

The Enhanced Color Display is compatible with all modes listed for the IBM Color Display. the following table describes additional modes supported by BIOS for the IBM Enhanced Color Display:

MODE #	TYPE	COLORS	ALPHA FORMAT	BUFFER START	BOX SIZE	MAX. PAGES	RESOLUTION
0*	A/N	16/64	40x25	B8000	8x14	8	320x350
1*	A/N	16/64	40x25	B8000	8x14	8	320x350
2*	A/N	16/64	80x25	B8000	8x14	8	640x350
3*	A/N	16/64	80x25	B8000	8x14	8	640x350
10*	APA	4/16 16/64	80x25	A8000	8x14	1/2	640x350

* Note that modes 0, 1, 2, and 3, are also listed for IBM Color Display support. BIOS provides enhanced support for these modes when an Enhanced Color Display is attached.

The values in the "COLORS" field indicate 16 colors of a 64 color palette or 4 colors of a sixteen color palette.

In mode 10, The dual values for the "COLORS" field and the "MAX. PAGES" field indicate the support provided when 64K or when greater than 64K of graphics memory is installed, respectively.

Basic Operations

Alphanumeric Modes

The data format for alphanumeric modes on the Enhanced Graphics Adapter is the same as the data format on the IBM Color/Graphics Monitor Adapter and the IBM Monochrome Display Adapter. As an added function, bit three of the attribute byte may be redefined by the Character Map Select register to act as a switch between character sets. This gives the programmer access to 512 characters at one time. This function is valid only when memory has been expanded to 128K bytes or more.

When an alphanumeric mode is selected, the BIOS transfers character patterns from the ROM to bit plane 2. The processor stores the character data in bit plane 0, and the attribute data in bit plane 1. The programmer can view bit planes 0 and 1 as a single buffer in alphanumeric modes. The CRTC generates sequential addresses, and fetches one character code byte and one attribute byte at a time. The character code and row scan count address bit plane 2, which contains the character generators. The appropriate dot patterns are then sent to the palette in the attribute chip, where color is assigned according to the attribute data.

Graphics Modes

320x200 Two and Four Color Graphics (Modes 4 and 5)

Addressing, mapping and data format are the same as the 320x200 pel mode of the Color/Graphics Monitor Adapter. The display buffer is configured at hex B8000. Bit image data is stored in bit planes 0 and 1.

640x200 Two Color Graphics (Mode 6)

Addressing, mapping and data format are the same as the 640x200 pel black and white mode of the Color/Graphics

Monitor Adapter. The display buffer is configured at hex B8000. Bit image data is stored in bit plane 0.

640x350 Monochrome Graphics (Mode F)

This mode supports graphics on the IBM Monochrome Display with the following attributes: black, video, blinking video, and intensified video. Resolution of 640x350 requires 56K bytes to support four attributes. By chaining maps 0 and 1, then maps 2 and 3 together, two 32K bit planes can be formed. This chaining is done only when necessary (less than 128K of graphics memory). The first map is the video bit plane, and the second map is the intensity bit plane. Both planes reside at hex address A0000.

Two bits, one from each bit plane, define one picture element (pel) on the screen. The bit definitions for the pels are given in the following table. The video bit plane is denoted by C0 and the Intensity Bit Plane is denoted by C2.

C2	C0	Pixel Color	Valid Attributes
0	0	Black	0
0	1	Video	3
1	0	Blinking Video	C
1	1	Intensified Video	F

The byte organization in memory is sequential. The first eight pels on the screen are defined by the contents of memory in location A000:0H, the second eight pels by location A000:1H, and so on. The first pel within any one byte is defined by bit 7 in the byte. The last pel within the byte is defined by bit 0 in the byte.

Monochrome graphics works in odd/even mode, which means that even CPU addresses go into even bit planes and odd CPU addresses go into odd bit planes. Since both bit planes reside at address A0000, the user must select which plane or planes he desires to update. This is accomplished by the map mask register of the sequencer. (See the table above for valid attributes).

16/64 Color Graphics Modes (Mode 10)

These modes support graphics in 16 colors on either a medium or high resolution monitor. The memory in these modes consists of using all four bit planes. Each bit plane represents a color as shown below. The bit planes are denoted as C0,C1,C2 and C3 respectively.

C0 = Blue Pels
C1 = Green Pels
C2 = Red Pels
C3 = Intensified Pels

Four bits (one from each plane) define one pel on the screen. The color combinations are illustrated in the following table:

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	Dark 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	Intensified White

The display buffer resides at address A0000. The map mask register of the sequencer is used to select any or all of the bit planes to be updated when a memory write to the display buffer is executed by the CPU.

Color Mapping

The Enhanced Graphics Adapter supports 640x350 Graphics for both the IBM Monochrome and the IBM Enhanced Color

Displays. Four color capability is supported on the EGA without the Graphics Memory Expansion Card (base 64 KB), and sixteen colors are supported when the Graphics Memory Expansion Card is installed on the adapter (128 KB or above). This section describes the differences in the colors displayed depending upon the graphics memory available. Note that colors 0H, 1H, 4H, and 7H map directly regardless of the graphics memory available.

Character Attribute	Monochrome	Mode 10H 64KB	Mode 10H >64KB
00H*	Black	Black	Black
01H*	Video	Blue	Blue
02H	Black	Black	Green
03H	Video	Blue	Cyan
04H*	Blinking	Red	Red
05H	Intensified	White	Magenta
06H	Blinking	Red	Brown
07H*	Intensified	White	White
08H	Black	Black	Dark Gray
09H	Video	Blue	Light Blue
0AH	Black	Black	Light Green
0BH	Video	Blue	Light Cyan
0CH	Blinking	Red	Light Red
0DH	Intensified	White	Light Magenta
0EH	Blinking	Red	Yellow
0FH	Intensified	White	Intensified White

* Graphics character attributes which map directly regardless of the graphics memory available.

Registers

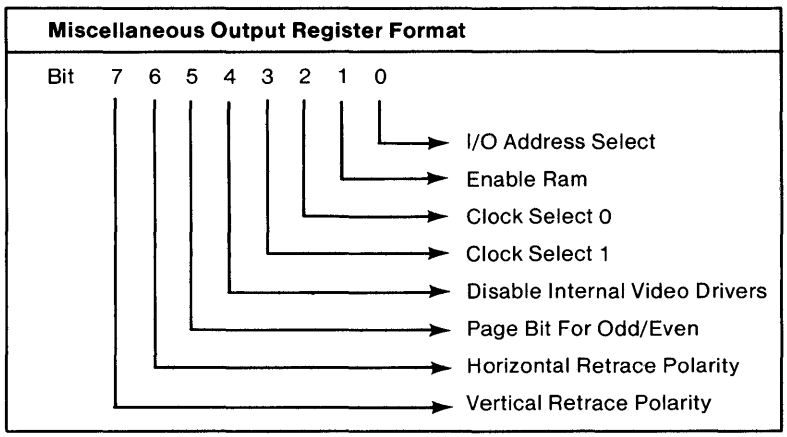
External Registers

This section contains descriptions of the registers of the Enhanced Graphics Adapter that are not contained in an LSI device.

Name	Port	Index
Miscellaneous Output Register	3C2	-
Feature Control Register	3?A	-
Input Status Register 0	3C2	-
Input Status Register 1	3?2	-
? = B in Monochrome Modes	? = D in Color Modes	

Miscellaneous Output Register

This is a write-only register. The processor output port address is hex 3C2. A hardware reset causes all bits to reset to zero.



Bit 0 3BX/3DX CRTC I/O Address—This bit maps the CRTC I/O addresses for IBM Monochrome or Color/Graphics Monitor Adapter emulation. A logical 0 sets CRTC addresses to 3BX and Input Status Register 1 's address to 3BA for Monochrome emulation. A logical 1 sets CRTC

addresses to 3DX and Input Status Register 1's address to 3DA for Color/Graphics Monitor Adapter emulation.

Bit 1 Enable RAM—A logical 0 disables RAM from the processor; a logical 1 enables RAM to respond at addresses designated by the Control Data Select value programmed into the Graphics Controllers.

Bit 2-Bit 3 Clock Select—These two bits select the clock source according to the following table:

Bits 3	2	
0	0	Selects 14 MHz clock from the processor I/O channel
0	1	Selects 16 MHz clock on-board oscillator
1	0	Selects external clock source from the feature connector.
1	1	Not used

Bit 4 Disable Internal Video Drivers—A logical 0 activates internal video drivers; a logical 1 disables internal video drivers. When the internal video drivers are disabled, the source of the direct drive color output becomes the feature connector direct drive outputs.

Bit 5 Page Bit For Odd/Even—Selects between two 64K pages of memory when in the Odd/Even modes (0,1,2,3,7). A logical 0 selects the low page of memory; a logical 1 selects the high page of memory.

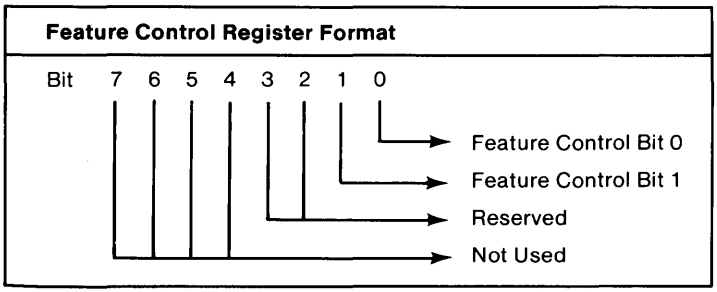
Bit 6 Horizontal Retrace Polarity—A logical 0 selects positive horizontal retrace; a logical 1 selects negative horizontal retrace.

Bit 7 Vertical Retrace Polarity—A logical 0 selects positive vertical retrace; a logical 1 selects

negative vertical retrace. The IBM Monochrome display requires a negative vertical retrace polarity.

Feature Control Register

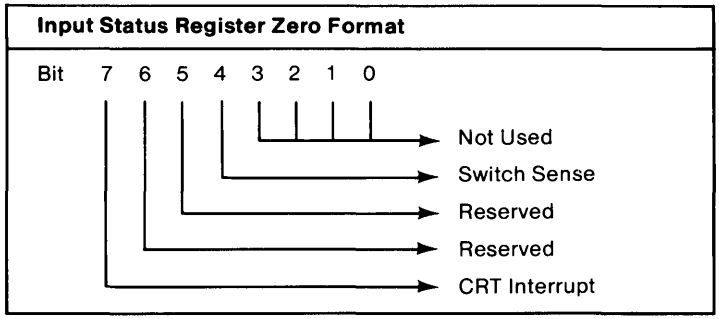
This is a write-only register. The processor output register is hex 3BA or 3DA.



Bits 0 and 1 Feature Control Bits—These bits are used to convey information to the feature connector. The output of these bits goes to the FEAT 0 (pin 19) and FEAT 1 (pin 17) of the feature connector.

Input Status Register Zero

This is a read-only register. The processor input port address is hex 3C2.



Bit 4 Switch Sense—When set to 1, this bit allows the processor to read the four configuration switches on the board. The setting of the CLKSEL field determines which switch is being read. The switch configuration can be determined by reading byte 40:88H in RAM.

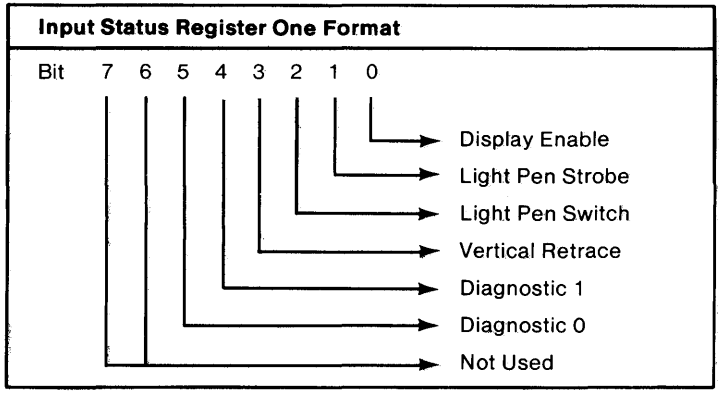
Bit 3: Switch 4 ; Logical 0 = switch closed
Bit 2: Switch 3 ; Logical 0 = switch closed
Bit 1: Switch 2 ; Logical 0 = switch closed
Bit 0: Switch 1 ; Logical 0 = switch closed

Bits 5 and 6 Feature Code—These bits are input from the Feat (0) and Feat (1) pins on the feature connector.

Bit 7 CRT Interrupt—A logical 1 indicates video is being displayed on the CRT screen; a logical 0 indicates that vertical retrace is occurring.

Input Status Register One

This is a read-only register. The processor port address is hex 3BA or hex 3DA.



Bit 0 Display Enable—Logical 0 indicates the CRT raster is in a horizontal or vertical retrace interval. This bit is the real time status of the display enable signal. Some programs use this status bit to restrict screen updates to inactive display intervals. The Enhanced Graphics Adapter does not require the CPU to update the screen buffer during inactive display intervals to avoid glitches in the display image.

Bit 1 Light Pen Strobe—A logical 0 indicates that the light pen trigger has not been set; a logical 1 indicates that the light pen trigger has been set.

Bit 2 Light Pen Switch—A logical 0 indicates that the light pen switch is closed; a logical 1 indicates that the light pen switch is open.

Bit 3 Vertical Retrace—A logical 0 indicates that video information is being displayed on the CRT screen; a logical 1 indicates the CRT is in a vertical retrace interval. This bit can be programmed to interrupt the processor on interrupt level 2 at the start of the vertical retrace. This is done through bits 4 and 5 of the Vertical Retrace End Register of the CRTC.

Bits 4 and 5 Diagnostic Usage—These bits are selectively connected to two of the six color outputs of the

Attribute Controller. The Color Plane Enable register controls the multiplexer for the video wiring. The following table illustrates the combinations available and the color output wiring.

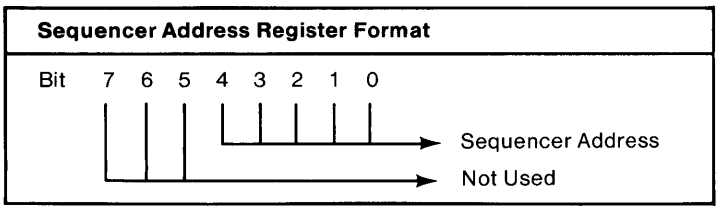
Color Plane Register		Input Status Register One	
Bit 5	**Bit 4**	**Bit 5**	**Bit 4**
0	0	Red	Blue
0	1	Secondary Blue	Green
1	0	Secondary Red	Secondary Green
1	1	Not Used	Not Used

Sequencer Registers

Name	Port	Index
Address	3C4	-
Reset	3C5	00
Clocking Mode	3C5	01
Map Mask	3C5	02
Character Map Select	3C5	03
Memory Mode	3C5	04

Sequencer Address Register

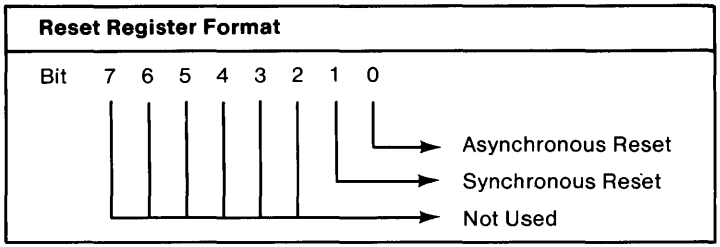
The Address Register is a pointer register located at address hex 3C4. This register is loaded with a binary value that points to the sequencer data register where data is to be written. This value is referred to as "Index" in the table above.



Bit 0-Bit 3 Sequencer Address Bits—A binary value pointing to the register where data is to be written.

Reset Register

This is a write-only register pointed to when the value in the address register is hex 00. The output port address for this register is hex 3C5.

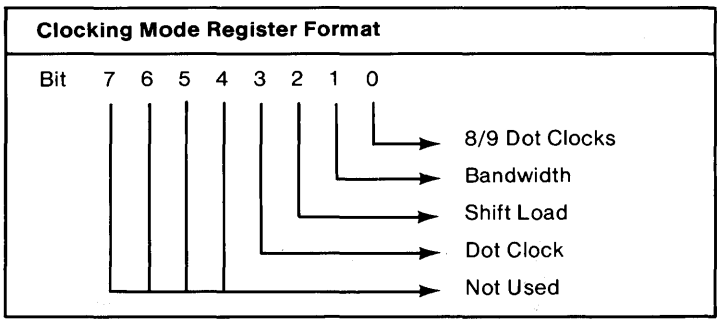


Bit 0 Asynchronous Reset—A logical 0 commands the sequencer to asynchronous clear and halt. All outputs are placed in the high impedance state when this bit is a 0. A logical 1 commands the sequencer to run unless bit 1 is set to zero. Resetting the sequencer with this bit can cause data loss in the dynamic RAMs.

Bit 1 Synchronous Reset—A logical 0 commands the sequencer to synchronous clear and halt. Bits 1 and 0 must both be ones to allow the sequencer to operate. Reset the sequencer with this bit before changing the Clocking Mode Register, if memory contents are to be preserved.

Clocking Mode Register

This is a write-only register pointed to when the value in the address register is hex 01. The output port address for this register is hex 3C5.



Bit 0 8/9 Dot Clocks—A logical 0 directs the sequencer to generate character clocks 9 dots wide; a logical 1 directs the sequencer to generate character clocks 8 dots wide. Monochrome alphanumeric mode (07H) is the only mode that uses character clocks 9 dots wide. All other modes must use 8 dots per character clock.

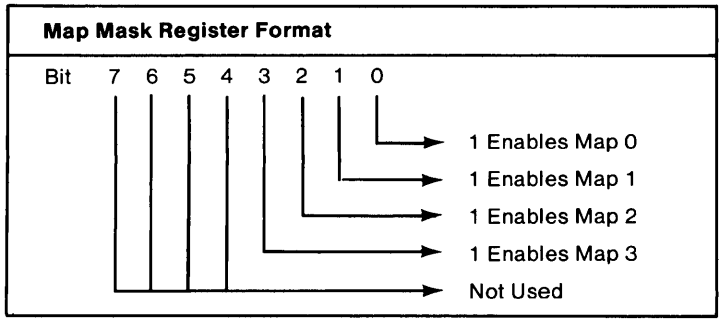
Bit 1 Bandwidth—A logical 0 makes CRT memory cycles occur on 4 out of 5 available memory cycles; a logical 1 makes CRT memory cycles occur on 2 out of 5 available memory cycles. Medium resolution modes require less data to be fetched from the display buffer during the horizontal scan time. This allows the CPU greater access time to the display buffer. All high resolution modes must provide the CRTC with 4 out of 5 memory cycles in order to refresh the display image.

Bit 2 Shift Load—When set to 0, the video serializers are reloaded every character clock; when set to 1, the video serializers are loaded every other character clock. This mode is useful when 16 bits are fetched per cycle and chained together in the shift registers.

Bit 3 Dot Clock—A logical 0 selects normal dot clocks derived from the sequencer master clock input. When this bit is set to 1, the master clock will be divided by 2 to generate the dot clock. All the other timings will be stretched since they are derived from the dot clock. Dot clock divided by two is used for 320x200 modes (0, 1, 4, 5) to provide a pixel rate of 7 MHz, (9 MHz for mode D).

Map Mask Register

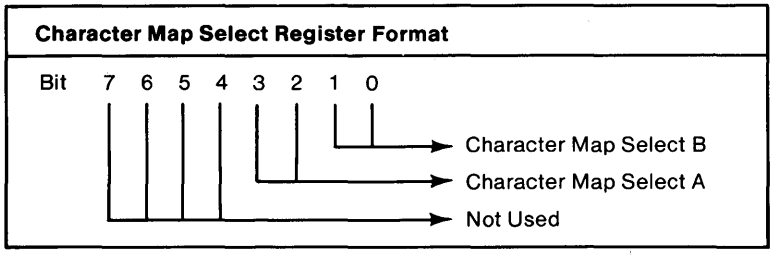
This is a write-only register pointed to when the value in the address register is hex 02. The output port address for this register is hex 3C5.



Bit 0-Bit 3 Map Mask—A logical 1 in bits 3 through 0 enables the processor to write to the corresponding maps 3 through 0. If this register is programmed with a value of 0FH, the CPU can perform a 32-bit write operation with only one memory cycle. This substantially reduces the overhead on the CPU during display update cycles in graphics modes. Data scrolling operations are also enhanced by setting this register to a value of 0FH and writing the display buffer address with the data stored in the CPU data latches. This is a read-modify-write operation. When odd/even modes are selected, maps 0 and 1 and maps 2 and 3 should have the same map mask value.

Character Map Select Register

This is a write-only register pointed to when the value in the address register is hex 03. The output port address for this register is 3C5.



Bit 0-Bit 1 Character Map Select B—Selects the map used to generate alpha characters when attribute bit 3 is a 0, according to the following table:

Bits 1	0	Map Selected	Table Location
Value			
0	0	0	1st 8K of Plane 2 Bank 0
0	1	1	2nd 8K of Plane 2 Bank 1
1	0	2	3rd 8K of Plane 2 Bank 2
1	1	3	4th 8K of Plane 2 Bank 3

Bit 2-Bit 3 Character Map Select A—Selects the map used to generate alpha characters when attribute bit 3 is a 1, according to the following table:

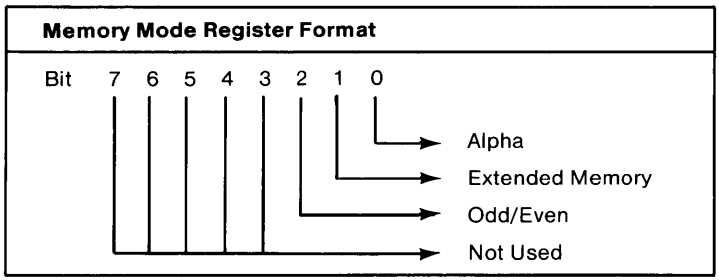
Bits 3	2	Map Selected	Table Location
Value			
0	0	0	1st 8K of Plane 2 Bank 0
0	1	1	2nd 8K of Plane 2 Bank 1
1	0	2	3rd 8K of Plane 2 Bank 2
1	1	3	4th 8K of Plane 2 Bank 3

In alphanumeric modes, bit 3 of the attribute byte normally has the function of turning the foreground intensity on or off. This bit however may be redefined as a switch between character sets. This function is enabled when there is a difference between the value in Character Map Select A and the value in Character Map Select B. Whenever these two values are the same, the character select function is disabled. The memory mode register bit 1 must be a 1 (indicates the memory extension card is installed in the unit) to enable this function; otherwise, bank 0 is always selected.

128K of graphics memory is required to support two character sets. 256K supports four character sets. Asynchronous reset clears this register to 0. This should be done only when the sequencer is reset.

Memory Mode Register

This is a write-only register pointed to when the value in the address register is hex 04. The processor output port address for this register is 3C5.



Bit 0 Alpha—A logical 0 indicates that a non-alpha mode is active. A logical 1 indicates that alpha mode is active and enables the character generator map select function.

Bit 1 Extended Memory—A logical 0 indicates that the memory expansion card is not installed. A logical 1 indicates that the memory expansion card is installed and enables access to the extended memory through address bits 14 and 15.

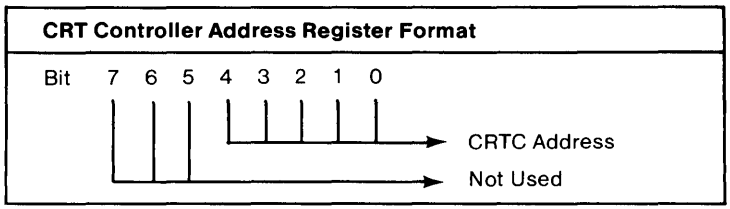
Bit 2 Odd/Even—A logical 0 directs even processor addresses to access maps 0 and 2, while odd processor addresses access maps 1 and 3. A logical 1 causes processor addresses to sequentially access data within a bit map. The maps are accessed according to the value in the map mask register.

CRT Controller Registers

Name	Port	Index
Address Register	3?4	-
Horizontal Total	3?5	00
Horizontal Display End	3?5	01
Start Horizontal Blank	3?5	02
End Horizontal Blank	3?5	03
Start Horizontal Retrace	3?5	04
End Horizontal Retrace	3?5	05
Vertical Total	3?5	06
Overflow	3?5	07
Preset Row Scan	3?5	08
Max Scan Line	3?5	09
Cursor Start	3?5	0A
Cursor End	3?5	0B
Start Address High	3?5	0C
Start Address Low	3?5	0D
Cursor Location High	3?5	0E
Cursor Location Low	3?5	0F
Vertical Retrace Start	3?5	10
Light Pen High	3?5	10
Vertical Retrace End	3?5	11
Light Pen Low	3?5	11
Vertical Display End	3?5	12
Offset	3?5	13
Underline Location	3?5	14
Start Vertical Blank	3?5	15
End Vertical Blank	3?5	16
Mode Control	3?5	17
Line Compare	3?5	18
? = B in Monochrome Modes and D in Color Modes		

CRT Controller Address Register

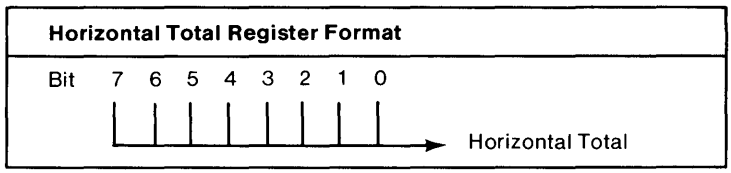
The Address register is a pointer register located at hex 3B4 or hex 3D4. If an IBM Monochrome Display is attached to the adapter, address 3B4 is used. If a color display is attached to the adapter, address 3D4 is used. This register is loaded with a binary value that points to the CRT Controller data register where data is to be written. This value is referred to as "Index" in the table above.



Bit 0-Bit 4 CRT Controller Address Bits—A binary value pointing to the CRT Controller register where data is to be written.

Horizontal Total Register

This is a write-only register pointed to when the value in the CRT Controller address register is hex 00. The processor output port address for this register is hex 3B5 or hex 3D5.

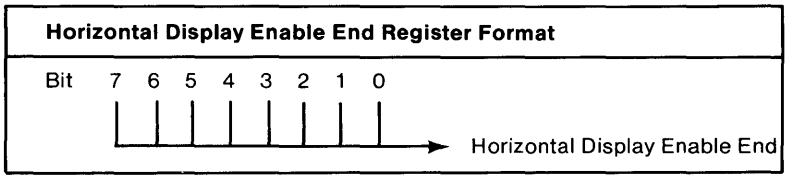


This register defines the total number of characters in the horizontal scan interval including the retrace time. The value directly controls the period of the horizontal retrace output signal. An internal horizontal character counter counts character clock inputs to the CRT Controller, and all horizontal and vertical timings are based upon the horizontal register. Comparators are used to compare register values with horizontal character values to provide horizontal timings.

Bit 0-Bit 7 Horizontal Total—The total number of characters less 2.

Horizontal Display Enable End Register

This is a write-only register pointed to when the value in the CRT Controller address register is hex 01. The processor output port address for this register is hex 3B5 or hex 3D5.

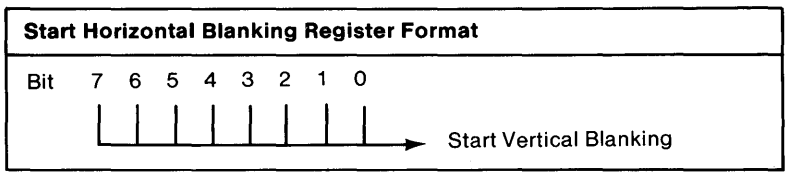


This register defines the length of the horizontal display enable signal. It determines the number of displayed character positions per horizontal line.

Bit 0-Bit 7 Horizontal display enable end —A value one less than the total number of displayed characters.

Start Horizontal Blanking Register

This is a write-only register pointed to when the value in the CRT Controller address register is hex 02. The processor output port address for this register is hex 3B5 or hex 3D5.

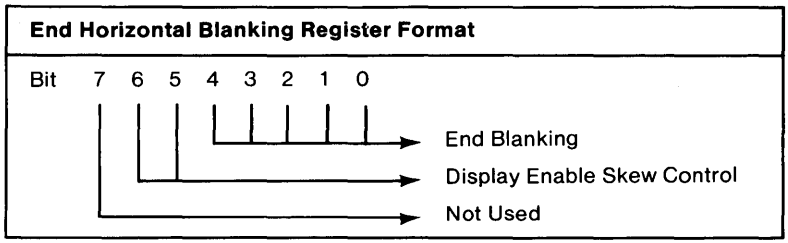


This register determines when the horizontal blanking output signal becomes active. The row scan address and underline scan line decode outputs are multiplexed on the memory address outputs and cursor outputs respectively during the blanking interval. These outputs are latched external to the CRT Controller with the falling edge of the BLANK output signal. The row scan address and underline signals remain on the output signals for one character count beyond the end of the blanking signal.

Bit 0-Bit 7 Start Horizontal Blanking—The horizontal blanking signal becomes active when the horizontal character counter reaches this value.

End Horizontal Blanking Register

This is a write-only register pointed to when the value in the CRT Controller address register is hex 03. The processor output port address for this register is hex 3B5 or hex 3D5.



This register determines when the horizontal blanking output signal becomes inactive. The row scan address and underline scan line decode outputs are multiplexed on the memory address outputs and the cursor outputs respectively during the blanking interval. These outputs are latched external to the CRT Controller with the falling edge of the BLANK output signal. The row scan address and underline signals remain on the output signals for one character count beyond the end of the blanking signal.

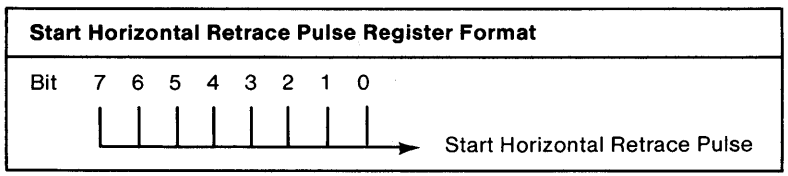
Bit 0-Bit 4 End Horizontal Blanking—A value equal to the five least significant bits of the horizontal character counter value at which time the horizontal blanking signal becomes inactive (logical 0). To obtain a blanking signal of width W, the following algorithm is used: Value of Start Blanking Register + Width of Blanking signal in character clock units = 5-bit result to be programmed into the End Horizontal Blanking Register.

Bit 5-Bit 6 Display Enable Skew Control—These two bits determine the amount of display enable skew. Display enable skew control is required to provide sufficient time for the CRT Controller to access the display buffer to obtain a character and attribute code, access the character generator font, and then go through the Horizontal Pel Panning Register in the Attribute Controller. Each access requires the display enable signal to be skewed one character clock unit so that the video output is in synchronization with the horizontal and vertical retrace signals. The bit values and amount of skew are shown in the following table:

Bits 6 5	
0 0	Zero character clock skew
0 1	One character clock skew
1 0	Two character clock skew
1 1	Three character clock skew

Start Horizontal Retrace Pulse Register

This is a write-only register pointed to when the value in the CRT Controller address register is hex 04. The processor output port address for this register is hex 3B5 or hex 3D5.

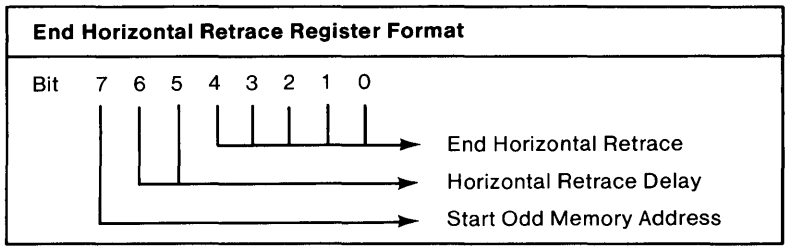


This register is used to center the screen horizontally, and to specify the character position at which the Horizontal Retrace Pulse becomes active.

Bit 0-Bit 7 Start Horizontal Retrace Pulse—The value programmed is a binary count of the character position number at which the signal becomes active.

End Horizontal Retrace Register

This is a write-only register pointed to when the value in the CRT Controller address register is hex 05. The processor output port address for this register is hex 3B5 or hex 3D5.



This register specifies the character position at which the Horizontal Retrace Pulse becomes inactive (logical 0).

Bit 0-Bit 4 End Horizontal Retrace—A value equal to the five least significant bits of the horizontal character counter value at which time the horizontal retrace signal becomes inactive (logical 0). To obtain a retrace signal of width W, the following algorithm is used: Value of Start Retrace Register + width of horizontal retrace signal in character clock units = 5-bit result to be programmed into the End Horizontal Retrace Register.

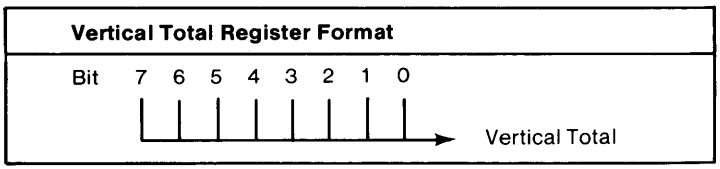
Bit 5-Bit 6 Horizontal Retrace Delay—These bits control the skew of the horizontal retrace signal. Binary 00 equals no Horizontal Retrace Delay. For some modes, it is necessary to provide a horizontal retrace signal that takes up the entire blanking interval. Some internal timings are generated by the falling edge of the horizontal retrace signal. To guarantee the signals are

latched properly, the retrace signal is started before the end of the display enable signal, and then skewed several character clock times to provide the proper screen centering.

Bit 7 Start Odd/Even Memory Address—This bit controls whether the first CRT memory address output after a horizontal retrace begins with an even or an odd address. A logical 0 selects even addresses; a logical 1 selects odd addresses. This bit is used for horizontal pel panning applications. Generally, this bit should be set to a logical 0.

Vertical Total Register

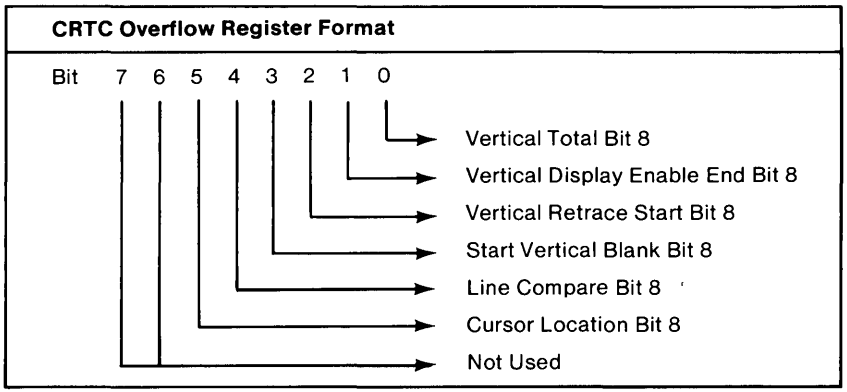
This is a write-only register pointed to when the value in the CRT Controller address register is hex 06. The processor output port address for this register is hex 3B5 or 3D5.



Bit 0-Bit 7 Vertical Total—This is the low-order eight bits of a nine-bit register. The binary value represents the number of horizontal raster scans on the CRT screen, including vertical retrace. The value in this register determines the period of the vertical retrace signal. Bit 8 of this register is contained in the CRT Controller Overflow Register hex 07 bit 0.

CRT Controller Overflow Register

This is a write-only register pointed to when the value in the CRT Controller Address Register is hex 07. The processor output port address for this register is hex 3B5 or hex 3D5.



Bit 0 Vertical Total—Bit 8 of the Vertical Total register (index hex 06).

Bit 1 Vertical Display Enable End—Bit 8 of the Vertical Display Enable End register (index hex 12).

Bit 2 Vertical Retrace Start—Bit 8 of the Vertical Retrace Start register (index hex 10).

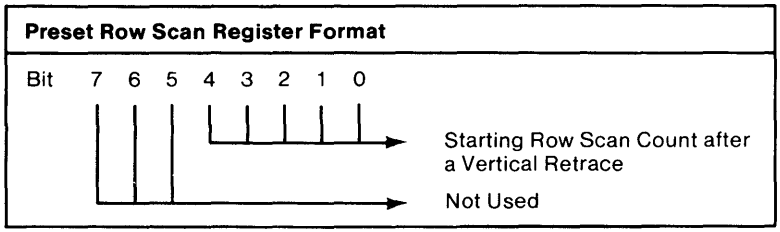
Bit 3 Start Vertical Blank—Bit 8 of the Start Vertical Blank register (index hex 15).

Bit 4 Line Compare—Bit 8 of the Line Compare register (index hex 18).

Bit 5 Cursor Location—Bit 8 of the Cursor Location register (index hex 0A).

Preset Row Scan Register

This is a write-only register pointed to when the value in the CRT Controller address register is hex 08. The processor output port address for this register is hex 3B5 or hex 3D5.

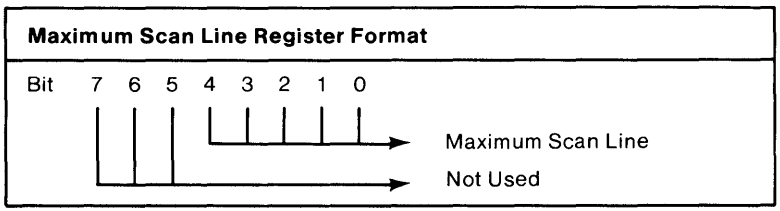


This register is used for pel scrolling.

Bit 0-Bit 4 Preset Row Scan (Pel Scrolling)—This register specifies the starting row scan count after a vertical retrace. The row scan counter increments each horizontal retrace time until a maximum row scan occurs. At maximum row scan compare time the row scan is cleared (not preset).

Maximum Scan Line Register

This is a write-only register pointed to when the value in the CRT Controller address register is hex 09. The processor output port address for this register is hex 3B5 or hex 3D5.

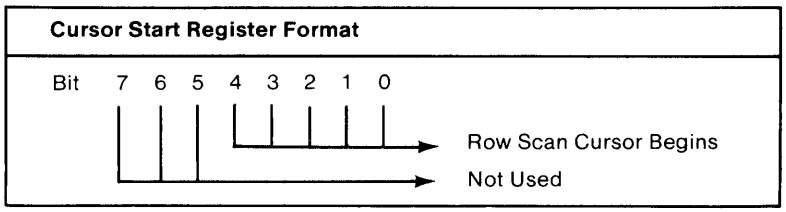


Bit 0-Bit 4 Maximum Scan Line—This register specifies the number of scan lines per character row. The number to be programmed is the maximum row scan number minus one.

Cursor Start Register

This is a write-only register pointed to when the value in the CRT Controller address register is hex 0A. The processor output port

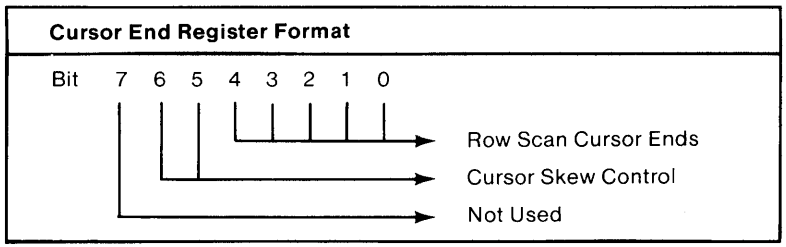
address for this register is hex 3B5 or hex 3D5.



Bit 0-Bit 4 Cursor Start—This register specifies the row scan of a character line where the cursor is to begin. The number programmed should be one less than the starting cursor row scan.

Cursor End Register

This is a write-only register pointed to when the value in the CRT Controller address register is hex 0B. The processor output port address for this register is hex 3B5 or hex 3D5.



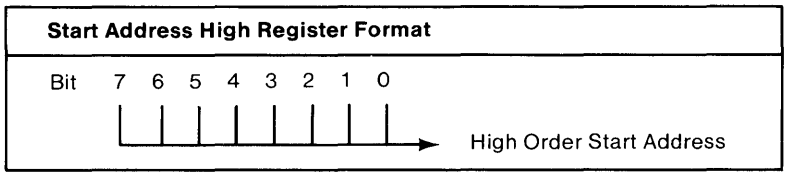
Bit 0-Bit 4 Cursor End—These bits specify the row scan where the cursor is to end.

Bit 5-Bit 6 Cursor Skew—These bits control the skew of the cursor signal.

Bits 6 5	
0 0	Zero character clock skew
0 1	One character clock skew
1 0	Two character clock skew
1 1	Three character clock skew

Start Address High Register

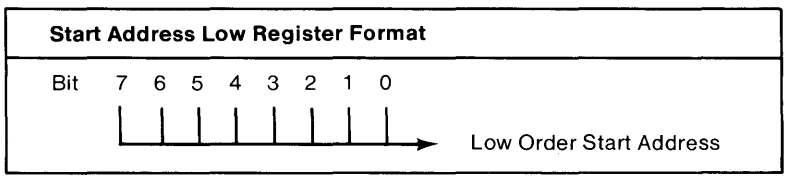
This is a read/write register pointed to when the value in the CRT Controller address register is hex 0C. The processor input/output port address for this register is hex 3B5 or hex 3D5.



Bit 0-Bit 7 Start Address High—These are the high-order eight bits of the start address. The 16-bit value, from the high-order and low-order start address registers, is the first address after the vertical retrace on each screen refresh.

Start Address Low Register

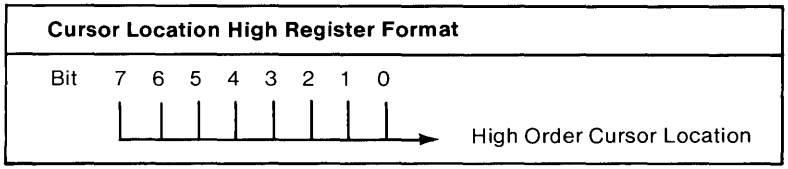
This is a read/write register pointed to when the value in the CRT Controller address register is hex 0D. The processor input/output port address for this register is hex 3B5 or hex 3D5.



Bit 0-Bit 7 Start Address Low—These are the low-order 8 bits of the start address.

Cursor Location High Register

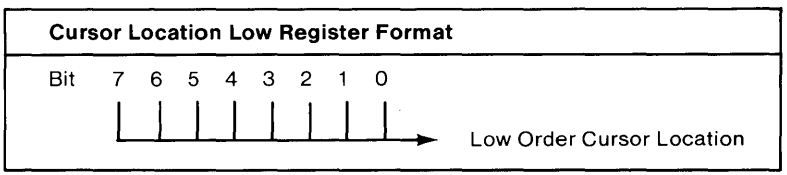
This is a read/write register pointed to when the value in the CRT Controller address register is hex 0E. The processor input/output port address for this register is hex 3B5 or hex 3D5.



Bit 0-Bit 7 Cursor Location High—These are the high-order 8 bits of the cursor location.

Cursor Location Low Register

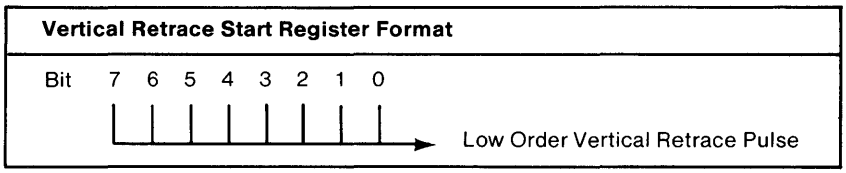
This is a read/write register pointed to when the value in the CRT Controller address register is hex 0F. The processor input/output port address for this register is hex 3B5 or Hex 3D5.



Bit 0-Bit 7 Cursor Location Low— These are the low-order 8 bits of the cursor location.

Vertical Retrace Start Register

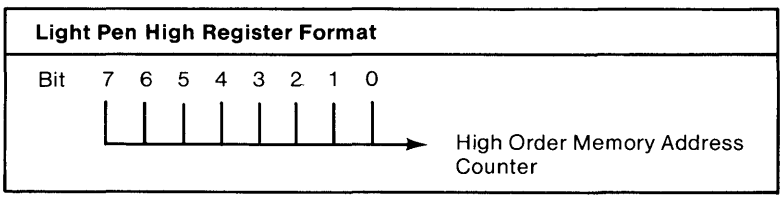
This is a write-only register pointed to when the value in the CRT Controller address register is hex 10. The processor output port address for this register is hex 3B5 or hex 3D5.



Bit 0-Bit 7 Vertical Retrace Start—This is the low-order 8 bits of the vertical retrace pulse start position programmed in horizontal scan lines. Bit 8 is in the overflow register location hex 07.

Light Pen High Register

This is a read-only register pointed to when the value in the CRT Controller address register is hex 10. The processor input port address for this register is hex 3B5 or hex 3D5.

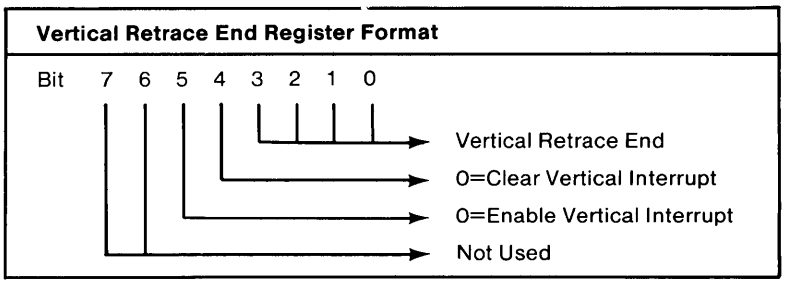


Bit 0-Bit 7 Light Pen High—This is the high order 8 bits of the memory address counter at the time the light pen was triggered.

Vertical Retrace End Register

This is a write-only register pointed to when the value in the CRT Controller address register is hex 11. The processor output port

address for this register is hex 3B5 or hex 3D5.



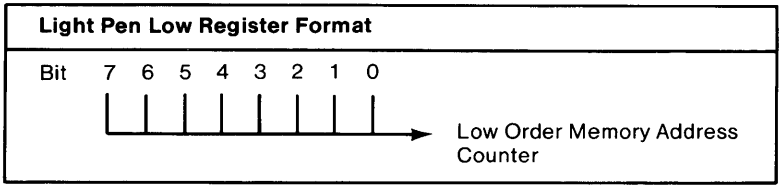
Bit 0-Bit 3 Vertical Retrace End—These bits determine the horizontal scan count value when the vertical retrace output signal becomes inactive. The register is programmed in units of horizontal scan lines. To obtain a vertical retrace signal of width W, the following algorithm is used: Value of Start Vertical Retrace Register + width of vertical retrace signal in horizontal scan units = 4-bit result to be programmed into the End Horizontal Retrace Register.

Bit 4 Clear Vertical Interrupt—A logical 0 will clear a vertical interrupt.

Bit 5 Enable Vertical Interrupt—A logical 0 will enable vertical interrupt.

Light Pen Low Register

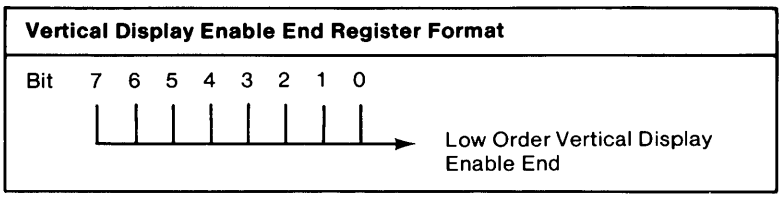
This is a read-only register pointed to when the value in the CRT Controller address register is hex 11. The processor input port address for this register is hex 3B5 or 3D5.



Bit 0-Bit 7 Light Pen Low—This is is the low-order 8 bits of the memory address counter at the time the light pen was triggered.

Vertical Display Enable End Register

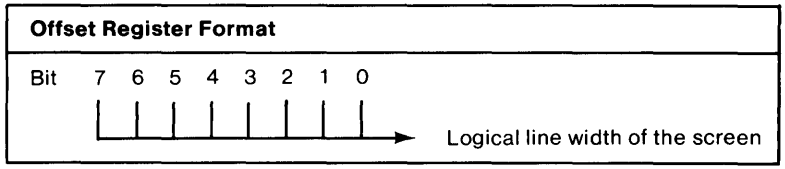
This is a write-only register pointed to when the value in the CRT Controller address register is hex 12. The processor output port address for this register is hex 3B5 or hex 3D5.



Bit 0-Bit 7 Vertical Display Enable End—These are the low-order 8 bits of the vertical display enable end position. This address specifies which scan line ends the active video area of the screen. Bit 8 is in the overflow register location hex 07.

Offset Register

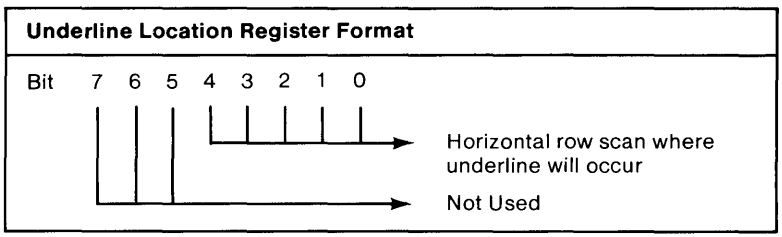
This is a write-only register pointed to when the value in the CRT Controller address register is hex 13. The processor output port address for this register is hex 3B5 or hex 3D5.



Bit 0-Bit 7 Offset—This register specifies the logical line width of the screen. The starting memory address for the next character row is larger than the current character row by this amount. The Offset Register is programmed with a word address. Depending upon the method of clocking the CRT Controller, this word address is either a word or double word address.

Underline Location Register

This is a write-only register pointed to when the value in the CRT Controller address register is hex 14. The processor output port address for this register is hex 3B5 or hex 3D5.

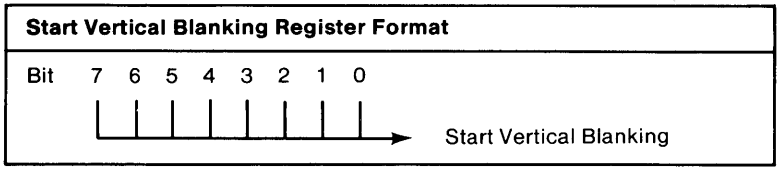


Bit 0-Bit 4 Underline Location—This register specifies the horizontal row scan on which underline will occur. The value programmed is one less than the scan line number desired.

Start Vertical Blanking Register

This is a write-only register pointed to when the value in the CRT Controller address register is hex 15. The processor output port

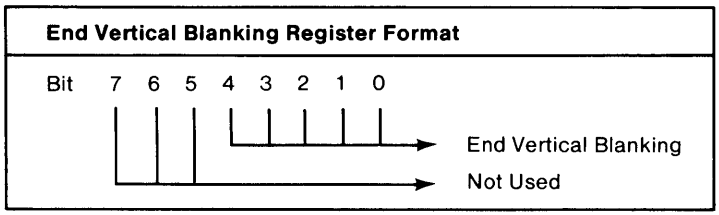
address for this register is hex 3B5 or hex 3D5.



Bit 0-Bit 7 Start Vertical Blank—These are the low 8 bits of the horizontal scan line count, at which the vertical blanking signal becomes active. Bit 8 bit is in the overflow register hex 07.

End Vertical Blanking Register

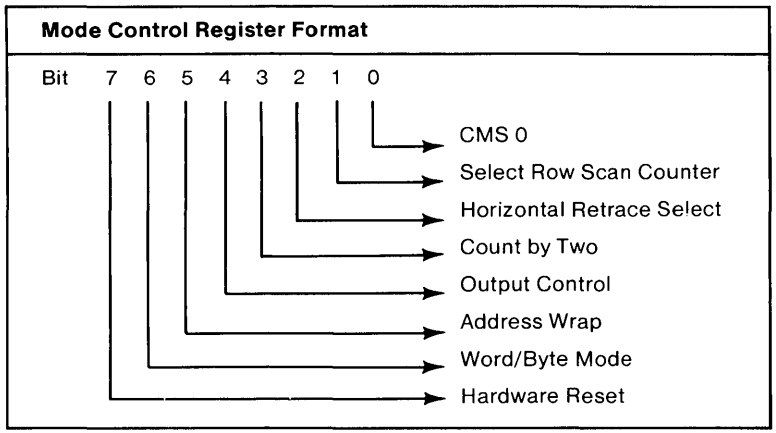
This is a write-only register pointed to when the value in the CRT Controller address register is hex 16. The processor output port address for this register is hex 3B5 or hex 3D5.



Bit 0-Bit 4 End Vertical Blank—This register specifies the horizontal scan count value when the vertical blank output signal becomes inactive. The register is programmed in units of horizontal scan lines. To obtain a vertical blank signal of width W, the following algorithm is used: Value of Start Vertical Blank Register + width of vertical blank signal in horizontal scan units = 5-bit result to be programmed into the End Vertical Blank Register.

Mode Control Register

This is a write-only register pointed to when the value in the CRT Controller address register is hex 17. The processor output port address for this register is hex 3B5 or hex 3D5.



Bit 0 Compatibility Mode Support— When this bit is a logical 0, the row scan address bit 0 is substituted for memory address bit 13 during active display time. A logical 1 enables memory address bit 13 to appear on the memory address output bit 13 signal of the CRT Controller. The CRT Controller used on the IBM Color/Graphics Monitor Adapter is the 6845. The 6845 has 128 horizontal scan line address capability. To obtain 640 by 200 graphics resolution, the CRTC was programmed for 100 horizontal scan lines with 2 row scan addresses per character row. Row scan address bit 0 became the most significant address bit to the display buffer. Successive scan lines of the display image were displaced in memory by 8K bytes. This bit allows compatibility with the 6845 and Color Graphics APA modes of operation.

Bit 1 Select Row Scan Counter—A logical 0 selects row scan counter bit 1 on MA 14 output pin. A logical 1 selects MA 14 counter bit on MA 14 output pin.

Bit 2 Horizontal Retrace Select—This bit selects Horizontal Retrace or Horizontal Retrace divided by 2 as the clock that controls the vertical timing counter. This bit can be used to effectively double the vertical resolution capability of the CRT Controller. The vertical counter has a maximum resolution of 512 scan lines due to the 9-bit wide Vertical Total Register. If the vertical counter is clocked with the horizontal retrace divided by 2 clock, then the vertical resolution is doubled to 1024 horizontal scan lines. A logical 0 selects HRTC and a logical 1 selects HRTC divided by 2.

Bit 3 Count By Two— When this bit is set to 0, the memory address counter is clocked with the character clock input. A logical 1 clocks the memory address counter with the character clock input divided by 2. This bit is used to create either a byte or word refresh address for the display buffer.

Bit 4 Output Control—A logical 0 enables the module output drivers. A logical 1 forces all outputs into high impedance state.

Bit 5 Address Wrap—This bit selects Memory Address counter bit MA 13 or bit MA 15, and it appears on the MA 0 output pin in the word address mode. If you are not in the word address mode, MA 0 counter output appears on the MA 0 output pin. A logical 1 selects MA 15. In odd/even mode, bit MA 13 should be selected when the 64K memory is installed on the board. Bit MA 15 should be selected when greater then 64K memory is installed. This function is used to implement Color Graphics Monitor Adapter compatibility.

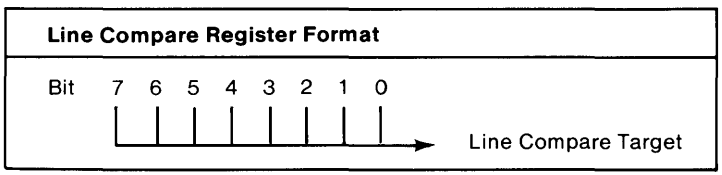
Bit 6 Word Mode or Byte Mode—When this bit is a logical 0, the Word Mode shifts all memory address counter bits down one bit, and the most significant bit of the counter appears on the least significant bit of the memory address outputs. See table below for address output details. A logical 1 selects the Byte Address mode.

	Internal Memory Address Counter Wiring to the Output Multiplexer	
CRTC Out Pin	**Byte Address Mode**	**Word Address Mode**
MA 0/RFA 0	MA 0	MA 15 or MA 13
MA 1/RFA 1	MA 1	MA 0
MA 2/RFA 2	MA 2	MA 1
MA 3/RFA 3	MA 3	MA 2
*	*	*
*	*	*
*	*	*
MA 14/RS 3	MA 14	MA 13
MA 15/RS 4	MA 15	MA 14

Bit 7 Hardware Reset—A logical 0 forces horizontal and vertical retrace to clear. A logical 1 forces horizontal and vertical retrace to be enabled.

Line Compare Register

This is a write-only register pointed to when the value in the CRT Controller address register is hex 18. The processor output port address for this register is hex 3B5 or hex 3D5.



Bit 0-Bit 7 Line Compare—This register is the low-order 8 bits of the compare target. When the vertical

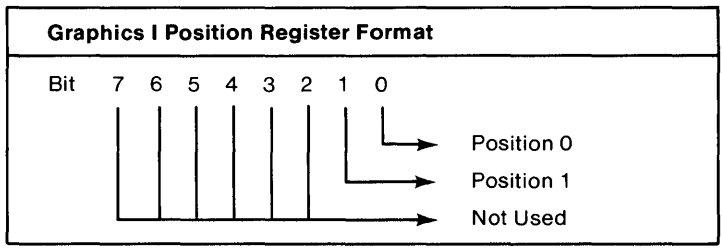
counter reaches this value, the internal start of the line counter is cleared. This allows an area of the screen to be immune to scrolling. Bit 8 of this register is in the overflow register hex 07.

Graphics Controller Registers

Name	Port	Index
Graphics 1 Position	3CC	-
Graphics 2 Position	3CA	-
Graphics 1 & 2 Address	3CE	-
Set/Reset	3CF	00
Enable Set/Reset	3CF	01
Color Compare	3CF	02
Data Rotate	3CF	03
Read Map Select	3CF	04
Mode Register	3CF	05
Miscellaneous	3CF	06
Color Don't Care	3CF	07
Bit Mask	3CF	08

Graphics 1 Position Register

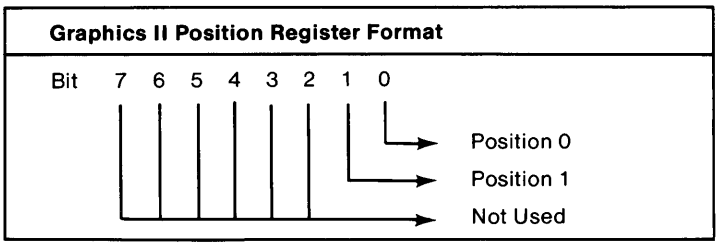
This is a write-only register. The processor output port address for this register is hex 3CC.



Bit 0-Bit 1 Position—These 2 bits are binary encoded hierarchy bits for the graphics chips. The position register controls which 2 bits of the processor data bus each chip responds to. Graphics 1 must be programmed with a position register value of 0 for this card.

Graphics 2 Position Register

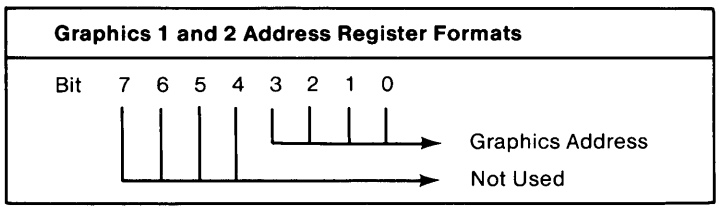
This is a write-only register. The processor output port address for this register is hex 3CA.



Bit 0-Bit 1 Position—These 2 bits are binary encoded hierarchy bits for the graphics chips. The position register controls which 2 bits of the processor data bus to which each chip responds. Graphics 2 must be programmed with a position register value of 1 for this card.

Graphics 1 and 2 Address Register

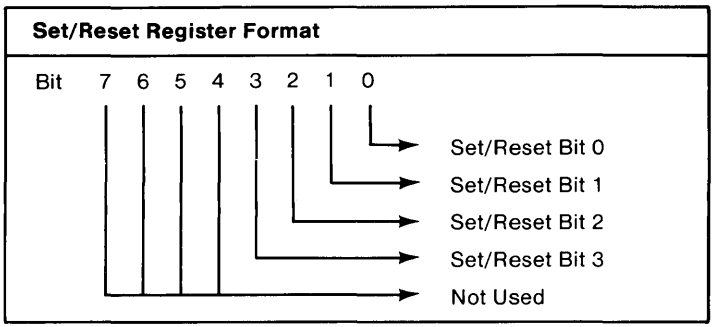
This is a write-only register and the processor output port address for this register is hex 3CE.



Bit 0-Bit 3 Graphics 1 and 2 Address Bits—This output loads the address register in both graphics chips simultaneously. This register points to the data register of the graphics chips.

Set/Reset Register

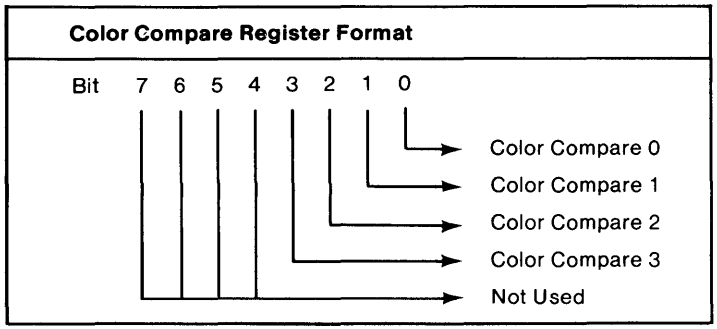
This is a write-only register pointed to by the value in the Graphics 1 and 2 address register. This value must be hex 00 before writing can take place. The processor output port address for this register is hex 3CF.



Bit 0-Bit 3 Set/Reset—These bits represent the value written to the respective memory planes when the processor does a memory write with write mode 0 selected and set/reset mode is enabled. Set/Reset can be enabled on a plane by plane basis with separate OUT commands to the Set/Reset register.

Enable Set/Reset Register

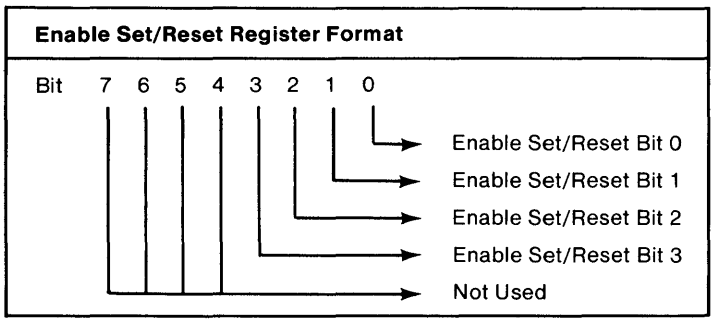
This is a write-only register and is pointed to by the value in the Graphics 1 and 2 address register. This value must be hex 01 before writing can take place. The processor output port for this register is hex 3CF.



Bit 0-Bit 3 Enable Set/Reset—These bits enable the set/reset function. The respective memory plane is written with the value of the Set/Reset register provided the write mode is 0. When write mode is 0 and Set/Reset is not enabled on a plane, that plane is written with the value of the processor data.

Color Compare Register

This is a write-only register pointed to by the value in the Graphics 1 and 2 address register. This value must be hex 02 before writing can take place. The processor output port address for this register is hex 3CF.

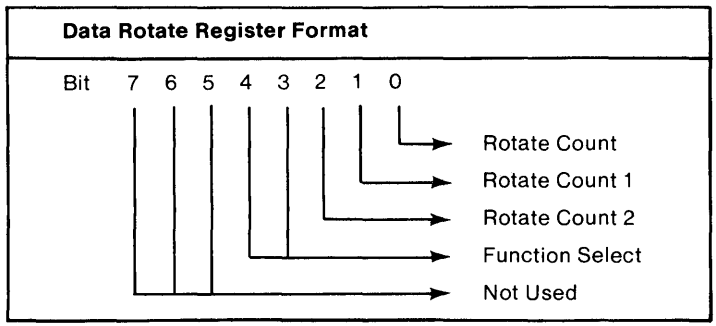


Bit 0-Bit 3 Color Compare—These bits represent a 4 bit color value to be compared. If the processor sets

read mode 1 on the graphics chips, and does a memory read, the data returned from the memory cycle will be a 1 in each bit position where the 4 bit planes equal the color compare register.

Data Rotate Register

This is a write-only register pointed to by the value in the Graphics 1 and 2 address register. This value must be hex 03 before writing can take place. The processor output port address for this register is hex 3CF.



Bit 0-Bit 2 Rotate Count—These bits represent a binary encoded value of the number of positions to rotate the processor data bus during processor memory writes. This operation is done when the write mode is 0. To write unrotated data the processor must select a count of 0.

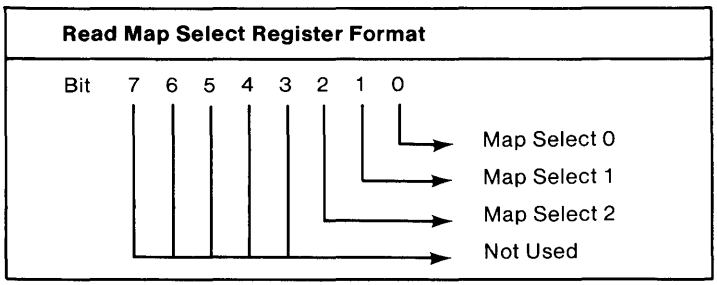
Bit 3-Bit 4 Function Select—Data written to memory can operate logically with data already in the processor latches. The bit functions are defined in the following table.

Bits 4 3	
0 0	Data unmodified.
0 1	Data AND'ed with latched data.
1 0	Data OR'ed with latched data.
1 1	Data XOR'ed with latched data.

Data may be any of the choices selected by the Write Mode Register except processor latches. If rotated data is selected, the rotate applies before the logical function.

Read Map Select Register

This is a write-only register pointed to by the value in the Graphics 1 and 2 address register. This value must be hex 04 before writing can take place. The processor output port address for this register is hex 3CF.

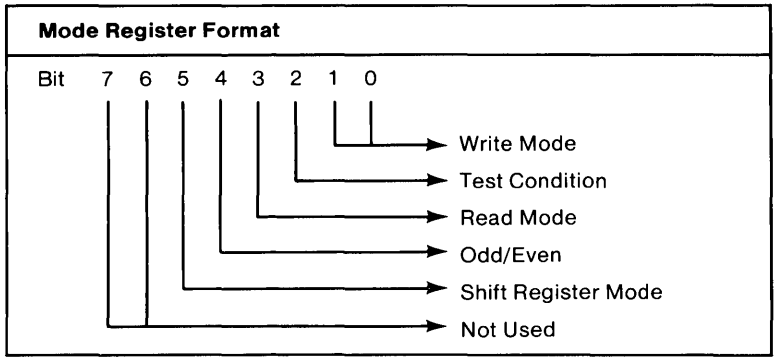


Bit 0-Bit 2 Map Select—These bits represent a binary encoded value of the memory plane number from which the processor reads data. This register has no effect on the color compare read mode described elsewhere in this section.

Mode Register

This is a write-only register pointed to by the value in the Graphics 1 and 2 address register. This value must be hex 05

before writing can take place. The processor output port address for this register is 3CF.



Bit 0-Bit 1 Write Mode

Bits 1 0	
0 0	Each memory plane is written with the processor data rotated by the number of counts in the rotate register, unless Set/Reset is enabled for the plane. Planes for which Set/Reset is enabled are written with 8 bits of the value contained in the Set/Reset register for that plane.
0 1	Each memory plane is written with the contents of the processor latches. These latches are loaded by a processor read operation.
1 0	Memory plane *n* (0 through 3) is filled with 8 bits of the value of data bit *n*.
1 1	Not Valid

The logic function specified by the function select register also applies.

Bit 2 Test Condition—A logical 1 directs graphics controller outputs to be placed in high impedance state for testing.

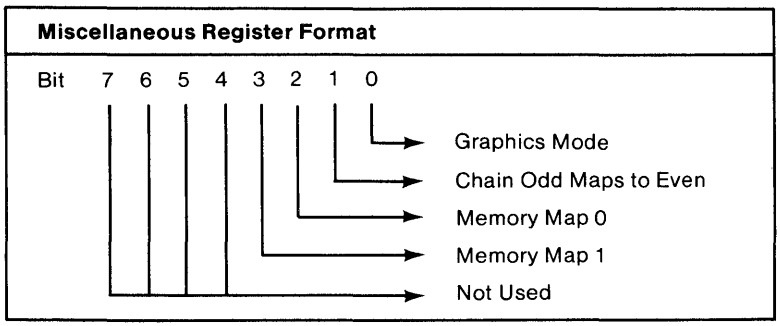
Bit 3 Read Mode—When this bit is a logical 0, the processor reads data from the memory plane selected by the read map select register. When this bit is a logical 1, the processor reads the results of the comparison of the 4 memory planes and the color compare register.

Bit 4 Odd/Even—A logical 1 selects the odd/even addressing mode, which is useful for emulation of the Color Graphics Monitor Adapter compatible modes. Normally the value here follows the value of the Memory Mode Register bit 3 of the Sequencer.

Bit 5 Shift Register—A logical 1 directs the shift registers on each graphics chip to format the serial data stream with even numbered bits on the even numbered maps and odd numbered bits on the odd maps.

Miscellaneous Register

This is a write-only register pointed to by the value in the Graphics 1 and 2 address register. This value must be hex 06 before writing can take place. The processor output port for this register is hex 3CF.



Bit 0 Graphics Mode—This bit controls alpha-mode addressing. A logical 1 selects graphics mode. When set to graphics mode, the character generator address latches are disabled.

Bit 1 Chain Odd Maps To Even Maps—When set to 1, this bit directs the processor address bit 0 to be replaced by a higher order bit and odd/even maps to be selected with odd/even values of the processor A0 bit, respectively.

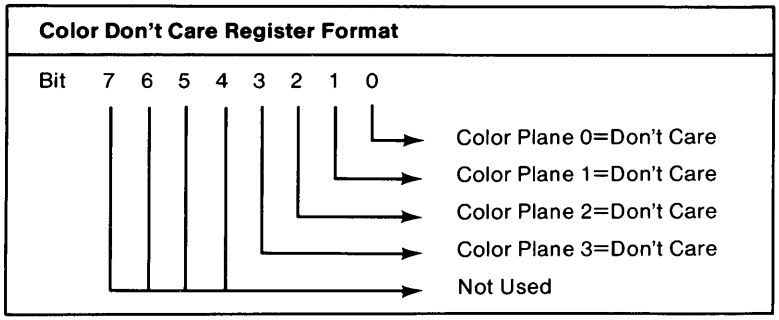
Bit 2-Bit 3 Memory Map—These bits control the mapping of the regenerative buffer into the processor address space.

Bits 3 2	
0 0	Hex A000 for 128K bytes.
0 1	Hex A000 for 64K bytes.
1 0	Hex B000 for 32K bytes
1 1	Hex B800 for 32K bytes.

If the display adapter is mapped at address hex A000 for 128K bytes, no other adapter can be installed in the system.

Color Don't Care Register

This is a write-only register and is pointed to by the value in the Graphics 1 and 2 address register. This value must be hex 07 before writing can take place. The processor output port for this register is hex 3CF.



Bit 0 Color Don't Care—Color plane 0=don't care when reading color compare when this bit is set to 1.

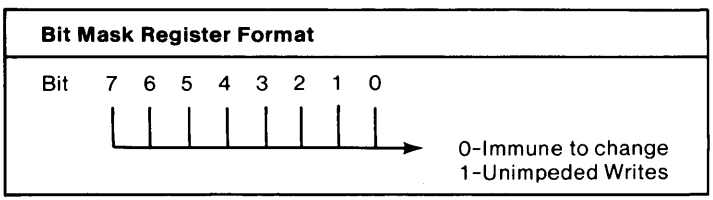
Bit 1 Color Don't Care—Color plane 1=don't care when reading color compare when this bit is set to 1.

Bit 2 Color Don't Care—Color plane 2=don't care when reading color compare when this bit is set to 1.

Bit 3 Color Don't Care—Color plane 3=don't care when reading color compare when this bit is set to 1.

Bit Mask Register

This is a write-only register and is pointed to by the value in the Graphics 1 and 2 address register. This value must be hex 08 before writing can take place. The processor output port for this register is hex 3CF.



Bit 0-Bit 7 Bit Mask—Any bit programmed to *n* causes the corresponding bit *n*in each bit plane to be immune to change provided that the location being written was the last location read by the processor. Bits programmed to a 1 allow unimpeded writes to the corresponding bits in the bit planes.

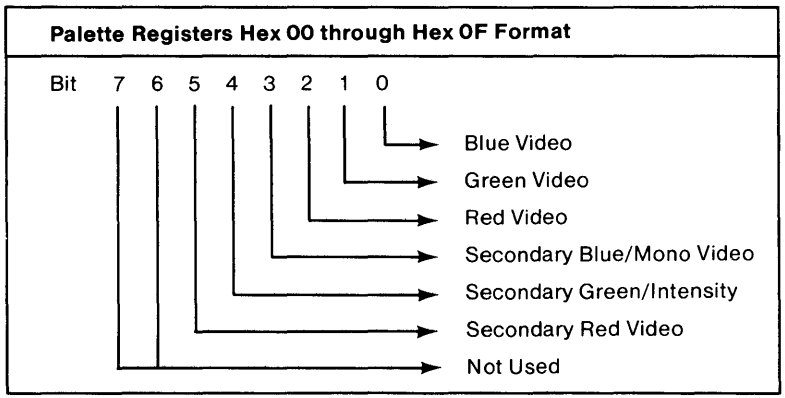
The bit mask applies to any data written by the processor (rotate, AND'ed, OR'ed, XOR'ed, DX, and S/R). To preserve bits using the bit mask, data must be latched internally by reading the location. When data is written to preserve the bits, the most current data in latches is written in those positions. The bit mask applies to all bit planes simultaneously.

Attribute Controller Registers

Name	Port	Index
Address Register	3C0	-
Palette Registers	3C0	00-0F
Mode Control Register	3C0	10
Overscan Color Register	3C0	11
Color Plane Enable Register	3C0	12
Horizontal Pel Panning Register	3C0	13

Attribute Address Register

This is a write-only register. The processor output port is hex 3C0.



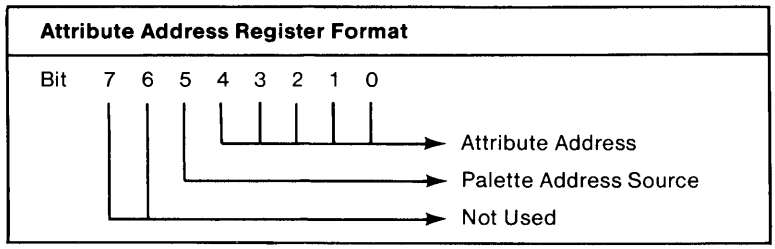
Bit 0-Bit 4 Attribute Address Bits—The Address Register is a pointer register located at hex 3C0. This register is loaded with a binary value that points to the attribute data register where data is to be written. The Attribute Controller does not have an address bit input to control selection of the address and data registers. An internal address flip-flop controls selection of either the address or data registers. To initialize the flip-flop, an IOR instruction is issued to the Attribute Controller at address 3BA or 3DA. This clears the flip-flop, and selects the Address Register. After the Address Register has been loaded, the

next OUT instruction loads the data register. The flip-flop toggles each time an OUT is issued to the Attribute Controller.

Bit 5 Palette Address Source—When loading the color palette registers, bit 5 must be cleared to 0. To enable the memory data to access the color palette, bit 5 must be set to 1.

Palette Register Hex 00 through Hex 0F

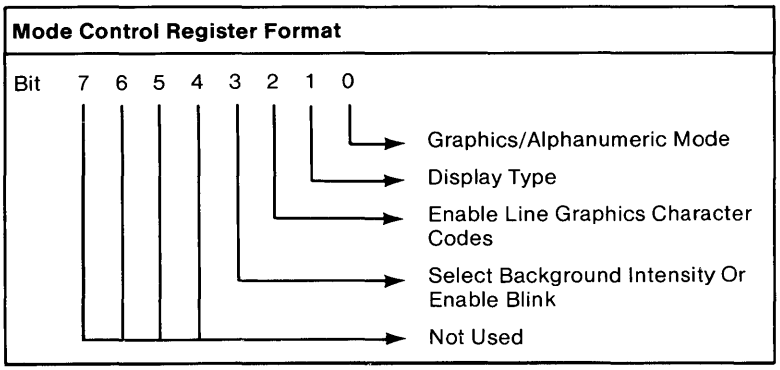
This is a write-only register. The processor output port is hex 3C0.



Bit 0-Bit 5 Palette—These 6-bit registers allow a dynamic mapping between the text attribute or graphic color input value and the display color in the CRT screen. A logical 1 selects the appropriate color. A logical 0 de-selects. The color palette register should be modified only during the vertical retrace interval to avoid glitches in the displayed image. Note that some color monitors do not have an intensity input and only a maximum of eight colors are be available. Monitors with four color inputs display sixteen colors, and monitors with six color inputs display 64 colors.

Mode Control Register

This is a write-only register pointed to by the value in the Attribute address register. This value must be hex 10 before writing can take place. The processor output port address for this register is hex 3C0.



Bit 0 Graphics/Alphanumeric Mode—A logical 0 selects selects alphanumeric mode. A logical 1 selects graphics mode.

Bit 1 Monochrome Display/Color Display—A logical 0 selects IBM monochrome display attributes. A logical 1 selects color Display attributes.

Bit 2 Enable Line Graphics Character Codes—When this bit is set to 0, the ninth dot will be the same as the background. A logical 1 enables the special line graphics character codes for the IBM Monochrome Display adapter. This bit when enabled forces the ninth dot of a line graphic character to be identical to the eighth dot of the character. The line graphics character codes for the Monochrome Display Adapter are Hex C0 through Hex DF.

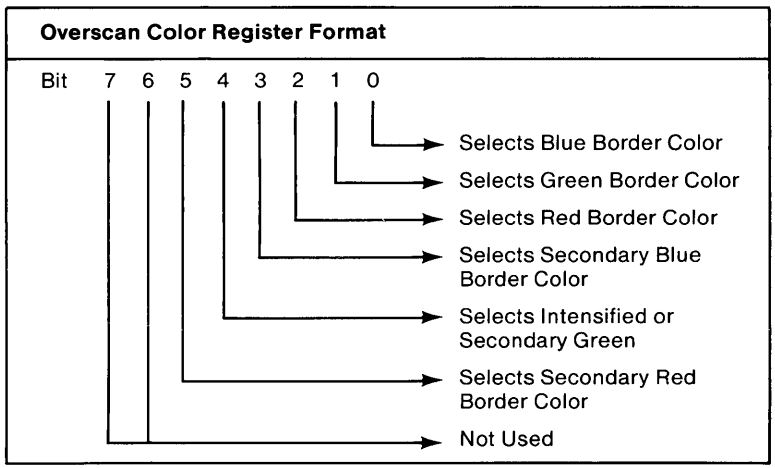
For character fonts that do not utilize the line graphics character codes in the range of Hex C0

through Hex DF, bit 2 of this register should be a logical 0. Otherwise unwanted video information will be displayed on the CRT screen.

Bit 3 Enable Blink/Select Background Intensity—A logical 0 selects the background intensity of the attribute input. This mode was available on the Monochrome and Color Graphics adapters. A logical 1 enables the blink attribute in alphanumeric modes. This bit must also be set to 1 for blinking graphics modes.

Overscan Color Register

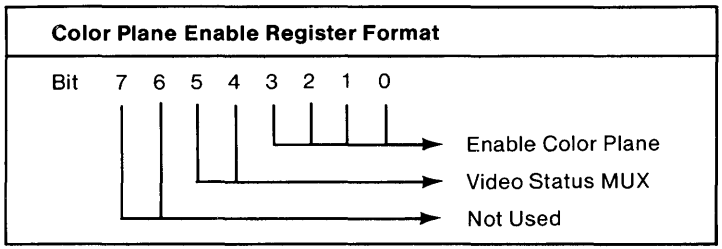
This is a write-only register pointed to by the value in the Attribute address register. This value must be hex 11 before writing can take place. The processor output port address for this register is hex 3C0.



Bit 0-Bit 5 Overscan Color—This 6-bit register determines the overscan (border) color displayed on the CRT screen. For monochrome display this register should be set to a value of 0. A logical 1 selects the appropriate color.

Color Plane Enable Register

This is a write-only register pointed to by the value in the Attribute address register. This value must be hex 12 before writing can take place. The processor output port address for this register is 3C0.



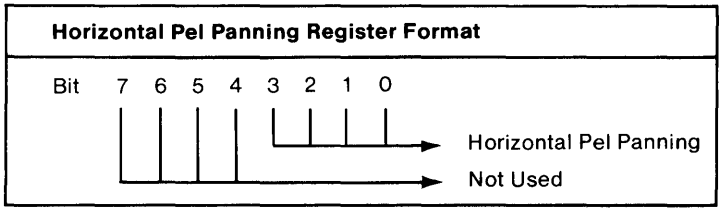
Bit 0-Bit 3 Enable Color Plane—Writing a logical 1 in any of bits 0 through 3 enables the respective display memory color plane.

Bit 4-Bit 5 Video Status MUX—Selects two of the six color outputs to be available on the status port. The following table illustrates the combinations available and the color output wiring.

COLOR PLANE ENABLE REGISTER		INPUT STATUS REGISTER ONE	
Bit 5	**Bit 4**	**Bit 5**	**Bit 4**
0	0	Red	Blue
0	1	Secondary Blue	Green
1	0	Secondary Red	Secondary Green
1	1	Not Used	Not Used

Horizontal Pel Panning Register

This is a write-only register pointed to by the value in the Attribute address register. This value must be hex 12 before writing can take place. The processor output port address for this register is hex 3C0.



Bit 0-Bit 3 Horizontal Pel Panning—This 4 bit register selects the number of picture elements (pels) to shift the video data horizontally to the left. Pel panning is available in both A/N and APA modes. In Monochrome A/N mode, the image can be shifted a maximum of 9 pels. In all other A/N and APA modes, the image can be shifted a maximum of 8 pels. The sequence for shifting the image is given below:

9 pels/character : 8, 0, 1, 2, 3, 4, 5, 6, 7 (Monochrome A/N mode only)

8 pels/character : 0, 1, 2, 3, 4, 5, 6, 7 (All other Modes)

Programming Considerations

Programming the Registers

Each of the LSI devices has an address register and a number of data registers. The address register serves as a pointer to the other registers on the LSI device. It is a write-only register that is loaded by the processor by executing an 'OUT' instruction to its I/O address with the index of the selected data register.

The data registers on each LSI device are accessed through a common I/O address. They are distinguished by the pointer (index) in the address register. To write to a data register, the address register is loaded with the index of the appropriate data register, then the selected data register is loaded by executing an 'OUT' instruction to the common I/O address.

The external registers that are not part of an LSI device and the Graphics I and II registers are not accessed through an address register; they are written to directly.

The following tables define the values that are loaded into the registers by BIOS to support the different modes of operation supported by this adapter.

Register			Mode of Operation																
Name	Port	Index	0	1	2	3	4	5	6	7	D	E	F	10	F‡	10‡	0*	1*	2*	3*
Miscellaneous	3C2	–	23	23	23	23	23	23	23	A6	23	23	A2	A7	A2	A7	A7	A7	A7	A7
Feature Cntrl	3?A	–	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
Input Stat 0	3C2	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
Input Stat 1	3?2	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
? = B in monochrome modes ? = D in color modes																				
*Values for these modes when the IBM Enhanced Color Display is attached																				
‡Values for these modes when greater than 64K Graphics Memory is installed																				

External Registers

Register			Mode of Operation																
Name	Port	Index	0	1	2	3	4	5	6	7	D	E	F	10	F‡	10‡	0*	1*	2*	3*
Seq Address	3C4	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
Reset	3C5	00	03	03	03	03	03	03	03	03	03	03	03	03	03	03	03	03	03	03
Clock Mode	3C5	01	0B	0B	01	01	0B	0B	01	00	0B	01	05	05	01	01	0B	0B	01	01
Map Mask	3C5	02	03	03	03	03	03	03	01	03	0F	0F	0F	0F	0F	0F	03	03	03	03
Char Gen Sel	3C5	03	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
Memory Mode	3C5	04	03	03	03	03	02	02	06	03	06	06	00	00	06	06	03	03	03	03
*Values for these modes when the IBM Enhanced Color Display is attached																				
‡Values for these modes when greater than 64K Graphics Memory is installed																				

Sequencer Registers

Register			Mode of Operation																	
Name	Port	Index	0	1	2	3	4	5	6	7	D	E	F	10	F‡	10‡	0*	1*	2*	3*
Address Reg	3?4	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
Horiz Total	3?5	00	37	37	70	70	37	37	70	60	37	70	60	5B	60	5B	2D	2D	5B	5B
Hrz Disp End	3?5	01	27	27	4F	4F	27	27	4F	4F	27	4F	4F	4F	4F	4F	27	27	4F	4F
Strt Hrz Blk	3?5	02	2D	2D	5C	5C	2D	2D	59	56	2D	56	56	53	56	53	2B	2B	53	53
End Hrz Blk	3?5	03	37	37	2F	2F	37	37	2D	3A	37	2D	1A	17	3A	37	2D	2D	37	37
Strt Hrz Retr	3?5	04	31	31	5F	5F	30	30	5E	51	30	5E	50	50	50	52	28	28	51	51
End Hrz Retr	3?5	05	15	15	07	07	14	14	06	60	14	06	E0	BA	60	00	6D	6D	5B	5B
Vert Total	3?5	06	04	04	04	04	04	04	04	70	04	04	70	6C	70	6C	6C	6C	6C	6C
Overflow	3?5	07	11	11	11	11	11	11	11	1F	11	11	1F	1F	1F	1F	1F	1F	1F	1F
Preset Row SC	3?5	08	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
Max Scan Line	3?5	09	07	07	07	07	01	01	01	0D	00	00	00	00	00	00	0D	0D	0D	0D
Cursor Start	3?5	0A	06	06	06	06	00	00	00	0B	00	00	00	00	00	00	0B	0B	0B	0B
Cursor End	3?5	0B	07	07	07	07	00	00	00	0C	00	00	00	00	00	00	0C	0C	0C	0C
Strt Addr Hi	3?5	0C	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
Strt Addr Lo	3?5	0D	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
? = B in monochrome modes ? = D in color modes																				
*Values for these modes when the IBM Enhanced Color Display is attached																				
‡Values for these modes when greater than 64K Graphics Memory is installed																				

CRT Controller Registers (1 of 2)

Register			Mode of Operation																	
Name	Port	Index	0	1	2	3	4	5	6	7	D	E	F	10	F‡	10‡	0*	1*	2*	3*
Cursor LC Hi	3?5	0E	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
Cursor LC Low	3?5	0F	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
Vrt Retr Strt	3?5	10	E1	E1	E1	E1	E1	E1	E0	5E	E1	E0	5E	5E	5E	5E	5E	5E	5E	5E
Light Pen Hi	3?5	10	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
Vert Retr End	3?5	11	24	24	24	24	24	24	23	2E	24	23	2E	2B	2E	2B	2B	2B	2B	2B
Light Pen Low	3?5	11	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
Vrt Disp End	3?5	12	C7	C7	C7	C7	C7	C7	C7	5D	C7	C7	5D	5D	5D	5D	5D	5D	5D	5D
Offset	3?5	13	14	14	28	28	14	14	28	28	14	28	14	14	28	28	14	14	28	28
Underline Loc	3?5	14	08	08	08	08	00	00	00	0D	00	00	0D	0F	0D	0F	0F	0F	0F	0F
Strt Vert Blk	3?5	15	E0	E0	E0	E0	E0	E0	DF	5E	E0	DF	5E	5F	5E	5F	5E	5E	5E	5E
End Vert Blk	3?5	16	F0	F0	F0	F0	F0	F0	EF	6E	F0	EF	6E	0A	6E	0A	0A	0A	0A	0A
Mode Control	3?5	17	A3	A3	A3	A3	A2	A2	C2	A3	E3	E3	8B	8B	E3	E3	A3	A3	A3	A3
Line Compare	3?5	18	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
? = B in monochrome modes ? = D in color modes																				
*Values for these modes when the IBM Enhanced Color Display is attached																				
‡Values for these modes when greater than 64K Graphics Memory is installed																				

CRT Controller Registers (2 of 2)

Register			Mode of Operation																	
Name	Port	Index	0	1	2	3	4	5	6	7	D	E	F	10	F‡	10‡	0*	1*	2*	3*
Grphx I Pos	3CC	–	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
Grphx II Pos	3CA	–	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01
Grphx I II AD	3CE	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
Set Reset	3CF	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
Enable S/R	3CF	01	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
Color Compare	3CF	02	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
Data Rotate	3CF	03	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
Read Map Sel	3CF	04	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
Mode Register	3CF	05	10	10	10	10	30	30	00	10	00	00	10	10	00	00	10	10	10	10
Miscellaneous	3CF	06	0E	0E	0E	0E	0F	0F	0D	0A	05	05	07	07	05	05	0E	0E	0E	0E
Color No Care	3CF	07	00	00	00	00	00	00	00	00	0F	0F	0F	0F	0F	0F	00	00	00	00
Bit Mask	3CF	08	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
*Values for these modes when the IBM Enhanced Color Display is attached																				
‡Values for these modes when greater than 64K Graphics Memory is installed																				

Graphics SI Registers

Register			Mode of Operation																	
Name	Port	Index	0	1	2	3	4	5	6	7	D	E	F	10	F‡	10‡	0*	1*	2*	3*
Address	3?A	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
Palette	3C0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
Palette	3C0	01	01	01	01	01	13	13	17	08	01	01	08	01	08	01	01	01	01	01
Palette	3C0	02	02	02	02	02	15	15	17	08	02	02	00	00	00	02	02	02	02	02
Palette	3C0	03	03	03	03	03	17	17	17	08	03	03	00	00	00	03	03	03	03	03
Palette	3C0	04	04	04	04	04	02	02	17	08	04	04	18	04	18	04	04	04	04	04
Palette	3C0	05	05	05	05	05	04	04	17	08	05	05	18	07	18	05	05	05	05	05
Palette	3C0	06	06	06	06	06	06	06	17	08	06	06	00	00	00	06	14	14	14	14
Palette	3C0	07	07	07	07	07	07	07	17	08	07	07	00	00	00	07	07	07	07	07
Palette	3C0	08	10	10	10	10	10	10	17	10	10	10	00	00	00	38	38	38	38	38
Palette	3C0	09	11	11	11	11	11	11	17	18	11	11	08	01	08	39	39	39	39	39
Palette	3C0	0A	12	12	12	12	12	12	17	18	12	12	00	00	00	3A	3A	3A	3A	3A
Palette	3C0	0B	13	13	13	13	13	13	17	18	13	13	00	00	00	3B	3B	3B	3B	3B
? = B in monochrome modes ? = D in color modes																				
*Values for these modes when the IBM Enhanced Color Display is attached																				
‡Values for these modes when greater than 64K Graphics Memory is installed																				

Attribute Registers (1 of 2)

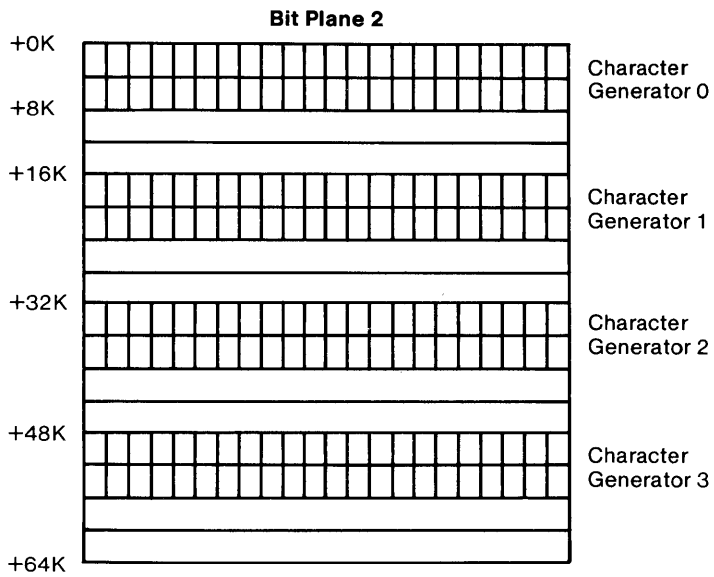
Register			Mode of Operation																	
Name	Port	Index	0	1	2	3	4	5	6	7	D	E	F	10	F‡	10‡	0*	1*	2*	3*
Palette	3C0	0C	14	14	14	14	14	14	17	18	14	14	00	04	00	3C	3C	3C	3C	3C
Palette	3C0	0D	15	15	15	15	15	15	17	18	15	15	18	07	18	3D	3D	3D	3D	3D
Palette	3C0	0E	16	16	16	16	16	16	17	18	16	16	00	00	00	3E	3E	3E	3E	3E
Palette	3C0	0F	17	17	17	17	17	17	18	17	17	00	00	00	3F	3F	3F	3F	3F	3F
Mode Control	3C0	10	08	08	08	08	01	01	01	0E	01	01	0B	0B	0B	01	08	08	08	08
Overscan	3C0	11	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
Color Plane	3C0	12	0F	0F	0F	0F	03	03	01	0F	0F	0F	05	05	05	0F	0F	0F	0F	0F
Hrz Panning	3C0	13	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
*Values for these modes when the IBM Enhanced Color Display is attached																				
‡Values for these modes when greater than 64K Graphics Memory is installed																				

Attribute Registers (2 of 2)

RAM Loadable Character Generator

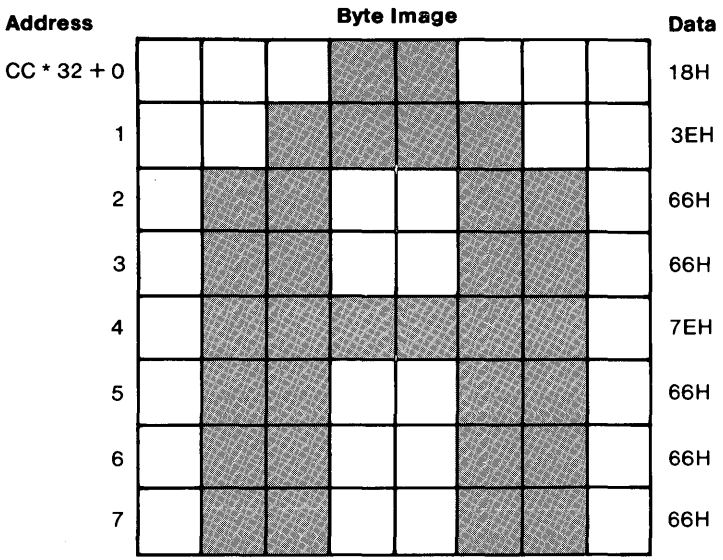
The character generator on the adapter is RAM loadable and can support characters up to 32 scan lines high. Two character generators are stored within the BIOS and one is automatically loaded into the RAM by the BIOS when an alphanumeric mode is selected. The Character Map Select Register can be programmed to define the function of bit 3 of the attribute byte to be a character generator switch. This allows the user to select between any two character sets residing in bit plane 2. This effectively gives the user access to 512 characters instead of 256. character tables may be loaded off line.The adapter must have 128K bytes of storage to support this function. Up to four tables can be loaded can be loaded with 256K of graphics memory installed.

The structure of the character tables is described in the following figure. The character generator is in bit plane 2 and must be protected using the map mask function.



The following figure illustrates the structure of each character pattern. If the CRT controller is programmed to generate *n* row

scans, then *n* bytes must be filled in for each character in the character generator. The example assumes eight row scans per character.



CC = Value of the character code. For example, 41H in the case of an ASCII "A".

Creating a 512 Character Set

This section describes how to create a 512 character set on the IBM Color Display. Note that only 256 characters can be printed on the printer. This is a special application which the Enhanced Graphics Adapter will support. The 9 by 14 characters will be displayed when attribute bit 3 is a logical 0, and the IBM Color/Graphics Monitor Adapter 8 by 8 characters will be displayed when the attribute bit 3 is a logical 1. This example is for demonstrative purposes only. The assembly language routine for creating 512 characters is given below. Debug 2.0 was used for this example. The starting assembly address is 100 and the character string is stored in location 200. This function requires 128K or more of graphics memory.

```
a100
mov ax,1102        ;load 8x8 character font in character
mov bl,02          ;generator number 2
int 10

mov ax,1103        ;select 512 character operation
mov bl,08          ;if attribute bit 3=1 use 8x8 font
int 10             ;if attribute bit 3=0 use 9x14 font

mov ax,1000        ;set color plane enable to 7H to disable
mov bx,0712        ;attribute bit 3 in the color palette
int 10             ;lookup table

mov ax,1301
mov bx,000F        ;write char. string with attribute bit 3=1
mov cx,003A        ;cx = character string length
mov dx,1600        ;write character on line 22 of display
mov bp,0200        ;pointer to character string location
push cs
pop es
int 10

mov ax ,1301
mov bx,0007        ;write char. string with attribute bit 3=0
mov cx,003A        ;cx = character string length
mov dx,1700        ;write character on line 23 of display
mov bp,0200        ;pointer to character string location
push cs
pop es
int 10
int 3

a200 db            "This character string is used to show 512
                   characters"
```

Creating an 80 by 43 Alphanumeric Mode

The following examples show how to create 80 column by 43 row, both alphanumeric and graphics, images on the IBM Monochrome Display. The BIOS Interface supports an 80 column by *n* row display by using the character generator load routine call. The print screen routine must be revectored to

handle the additional character rows on the screen. The assembly language required for both an alphanumeric and a graphics screen is shown below.

```
mov al,7          ;Monochrome alphanumeric mode
int 10            ;video interrupt call
mov ax,1112       ;character generator BIOS routine
mov bl,0          ;load 8 by 8 double dot character font
int 10            ;video interrupt call
mov ax,1200       ;alternate screen routine
move bl,20        ;select alternate print screen routine
int 10            ;video interrupt call
int 3

mov ax,f          ;Monochrome graphic mode
int 10            ;video interrupt call
mov ax,1123       ;character generator BIOS routine
mov bl,0          ;load 8 by 8 double dot character font
mov dl,2B         ;43 character rows
int 10            ;video interrupt call
mov ax,1200       ;alternate screen routine
mov bl,20         ;alternate print screen routine
int 10            ;video interrupt call
int 3
```

Vertical Interrupt Feature

The Enhanced Graphics Adapter can be programmed to create an interrupt each time the vertical display refresh time has ended. An interrupt handler routine must be written by the application to take advantage of this feature. The CRT Vertical interrupt is on IRQ2. The CPU can poll the Enhanced Graphics Adapter Input Status Register 0 (bit 7) to determine whether the CRTC caused the interrupt to occur.

The Vertical Retrace End Register (11H) in the CRT controller contains two bits which are used to control the interrupt circuitry. The remaining bits must be output as per the value in the mode table.

Bit 5 Enable Vertical Interrupt—A logical 0 will enable vertical interrupt.

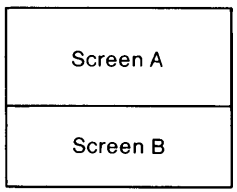
Bit 4 Clear Vertical Interrupt—A logical 0 will clear a vertical interrupt.

The sequence of events which occur in an interrupt handler are outlined below.

1. Clear IRQ latch and enable driver
2. Enable IRQ latch
3. Wait for vertical interrupt
4. Poll Interrupt Status Register 0 to determine if CRTC has caused the interrupt
5. If CRTC interrupt, then clear IRQ latch; if not, then branch to next interrupt handler.
6. Enable IRQ latch
7. Update Enhanced Graphics Adapter during vertical blanking interval
8. Wait for next vertical interrupt

Creating a Split Screen

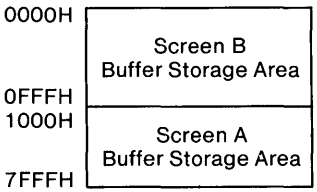
The Enhanced Graphics Adapter hardware supports an alphanumeric mode dual screen display. The top portion of the screen is designated as screen A, and the bottom portion of the screen is designated as screen B as per the following figure.



Dual Screen Definition

The following figure shows the screen mapping for a system containing a 32K byte alphanumeric storage buffer. Note that the Enhanced Graphics Adapter has a 32K byte storage buffer in alphanumeric mode. Information displayed on screen A is

defined by the start address high and low registers (0CH and 0DH) of the CRTC. Information displayed on screen B always begins at address 0000H.



Screen Mapping Within the Display Buffer Address Space

The Line Compare Register (18H) of the CRT Controller is utilized to perform the split screen function. The CRTC has an internal horizontal scan counter, and logic which compares the horizontal scan counter value to the Line Compare Register value and clears the memory address generator when a compare occurs. The linear address generator then sequentially addresses the display buffer starting at location zero, and each subsequent row address is is determined by the 16 bit addition of the start of line latch and the offset register.

Screen B can be smoothly scrolled onto the CRT screen by updating the Line compare in synchronization with the vertical retrace signal. The information on screen B is immune from scrolling operations which utilize the Start Address High and Low registers to scroll through the Screen A address map.

Compatibility Issues

The CRT Controller on the IBM Enhanced Graphics Adapter is a custom design, and is different than the 6845 controller used on the IBM Monochrome Monitor Adapter and the IBM Color/Graphics Monitor Adapter. It should be noted that several CRTC register addresses differ between the adapters. The following figure illustrates the registers which do not map directly across the two controllers.

Register	6485 Function	EGA CRTC Function
02H	Start Horiz. Retrace	Start Horiz. Blanking
03H	End Horiz. Retrace	End Horiz. Blanking
04H	Vertical Total	Start Horiz. Retrace
05H	Vertical Total Adjust	End Horiz. Retrace
06H	Vertical Displayed	Vertical Total
07H	Vertical Sync Position	Overflow
08H	Interlace Mode and Skew	Preset Row Scan

Existing applications which utilize the BIOS interface will generally be compatible with the Enhanced Graphics Adapter.

Horizontal screen centering was required on the IBM Color/Graphics Monitor Adapter in order to center the screen when generating composite video. This was done through the Horizontal Sync Position Register. Since the Enhanced Graphics Adapter does not support a composite video monitor, programs which do screen centering may cause loss of the screen image if centering is attempted.

The Enhanced Graphics Adapter offers a wider variety of displayable monochrome character attributes than the IBM Monochrome Display Adapter. Some attribute values may display differently between the two Adapters. The values listed in the table below, in any combinations with the blink and intensity attributes, will display identically.

Background R G B	Foreground R G B	Function
0 0 0	0 0 0	Non-Display
0 0 0	0 0 1	Underline
0 0 0	1 1 1	White Character/Black Background
1 1 1	0 0 0	Reverse Video

Software which explicitly addresses 3D8 (Mode Select Register) or 3D9 (Color Select Register) on the Color Graphics Monitor Adapter may produce different results on the Enhanced Graphics Adapter. For example, blinking which is disabled by writing to 3D8 on the Color Graphics Adapter will not be disabled on the Enhanced Graphics Adapter.

Interface

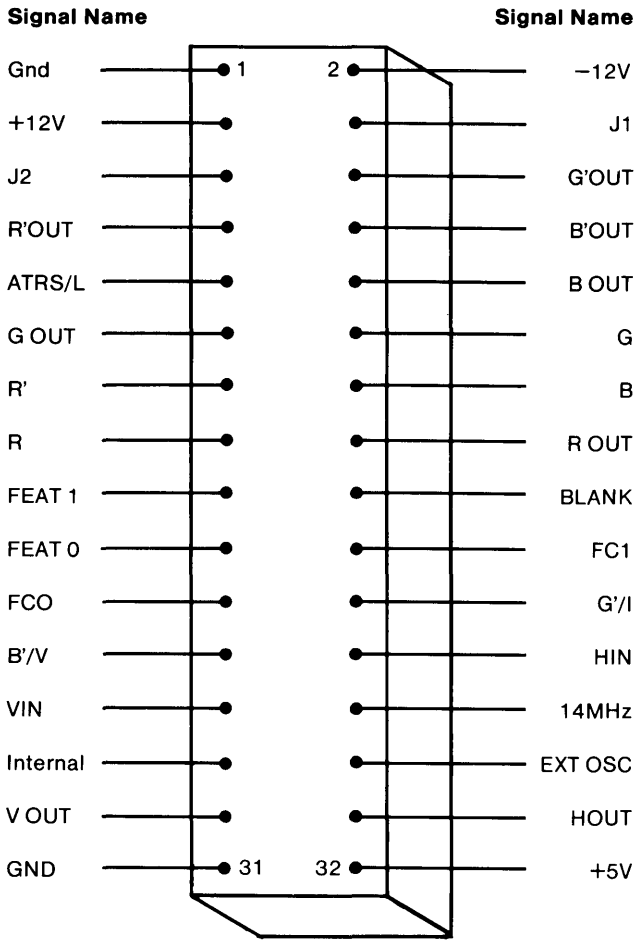
Feature Connector

The following is a description of the Enhanced Graphics Adapter feature connector. Note that signals coming from the Enhanced Graphics Adapter are labeled "inputs" and the signals coming to the Enhanced Graphics Adapter through the feature connector are labeled "outputs".

Signal	Description
J2	This pin is connected to auxiliary jack 2 on the rear panel of the adapter.
R'OUT	Secondary red output
ATRS/L	Attribute shift load. This signal controls the serialization of the video information. The shift register parallel loads at the dot clock leading edge when this signal is low.
G OUT	Primary green output
R'	Secondary red input
R	Primary red input
FC1	This signal is input from bit 1 (Feature Control Bit 1) of the Feature Control Register.
FC0	This signal is input from bit 0 (Feature Control Bit 0) of the Feature control Register.
FEAT 0	This signal is output to bit 5 (Feature Code 0) of Input Status Register 0.
B'/V	Secondary blue input/Monochrome video
VIN	Vertical retrace input

Internal	This signal is output to bit 4 (Disable Internal Video Drivers) of the Miscellaneous Output Register.
V OUT	Vertical retrace output
J1	This pin is connected to auxiliary jack 1 on the rear panel of the adapter.
G'OUT	Secondary green output
B'OUT	Secondary blue output
B OUT	Blue output
G	Green input
B	Blue input
R OUT	Red output
BLANK	This is a composite horizontal and vertical blanking signal from the CRTC.
FEAT 1	This signal is output to bit 6 (Feature Code 1) of Input Status Register 0.
G'/I	Secondary green/Intensity input
HIN	Horizontal retrace input from the CRTC
14MHZ	14 MHz signal from the system board
EXT OSC	External dot clock output
HOUT	Horizontal retrace output

The following figure shows the layout and pin numbering of the feature connector.

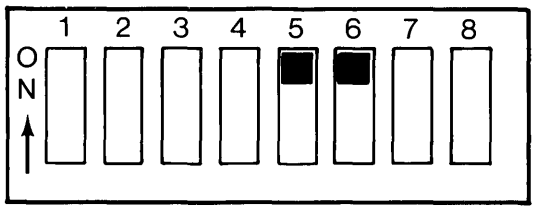


Feature Connector Diagram

Specifications

System Board Switches

The following figure shows the proper system board DIP switch settings for the IBM Enhanced Graphics Adapter when used with the Personal Computer and the Personal Computer XT. The switch block locations are illustrated in the Technical Reference Manual "System Board Component Diagram". The Personal Computer has two DIP switch blocks; the switch settings shown pertain to DIP Switch Block 1. The Personal Computer XT has one DIP switch block.

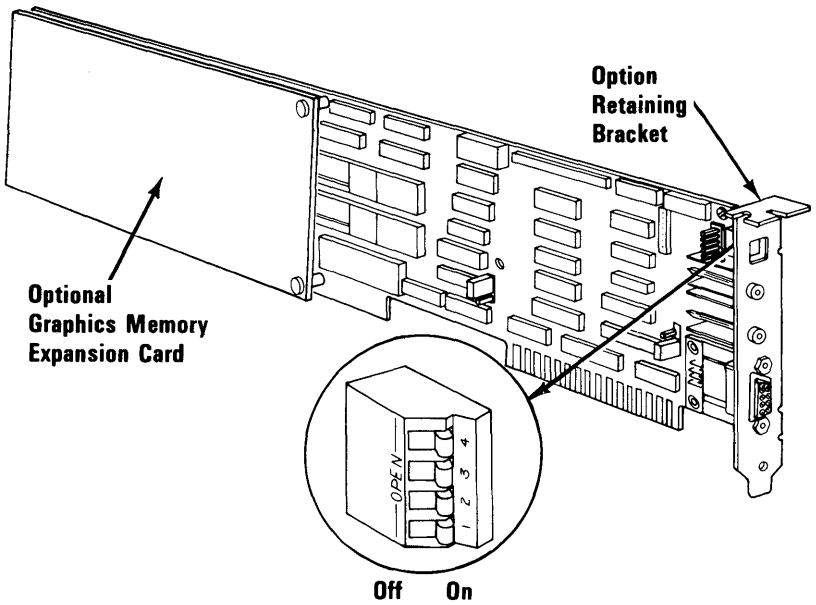


Switch Block (1)

Note: The DIP switches must be set as shown whenever the IBM Enhanced Graphics Adapter is installed, regardless of display type. This is true even when a second display adapter is installed in the system.

Configuration Switches

The following diagram shows the location and orientation of the configuration switches on the Enhanced Graphics Adapter.



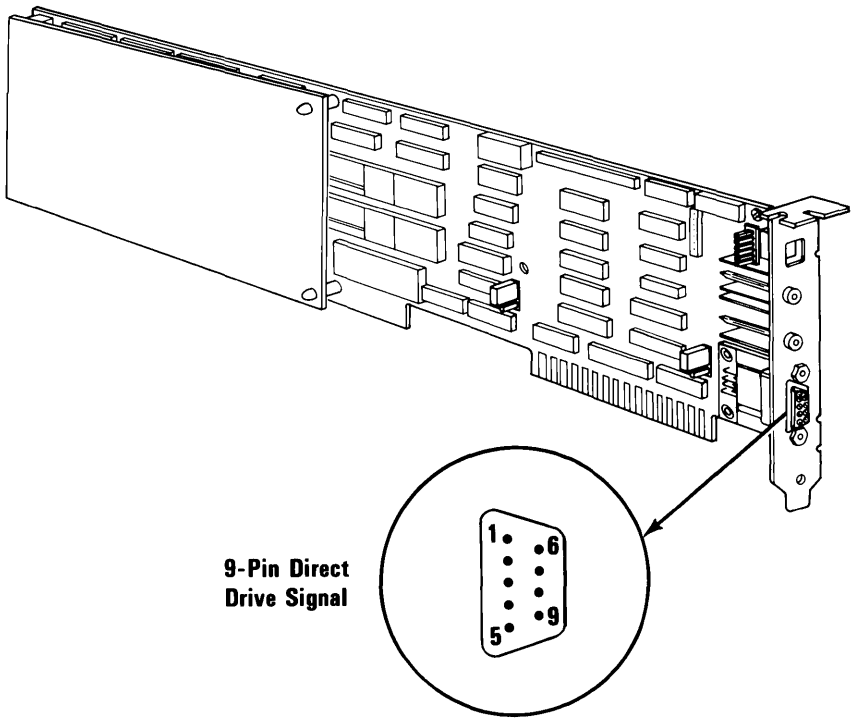
Configuration Switch Settings

The configuration switches on the Enhanced Graphics Adapter determine the type of display support the adapter provides, as follows:

Switch Settings for Enhanced Graphics Adapter as Primary Display Adapter						
				Configuration		
SW1	SW2	SW3	SW4	Enhanced Adapter	Monochrome Adapter	Color/Graphics Adapter
On	Off	Off	On	Color Display 40x25	Secondary	–
Off	Off	Off	On	Color Display 80x25	Secondary	–
On	On	On	Off	Enhanced Display Emulation Mode	Secondary	–
Off	On	On	Off	Enhanced Display Hi Res Mode	Secondary	–
On	Off	On	Off	Monochrome	–	Secondary 40x25
Off	Off	On	Off	Monochrome	–	Secondary 80x25

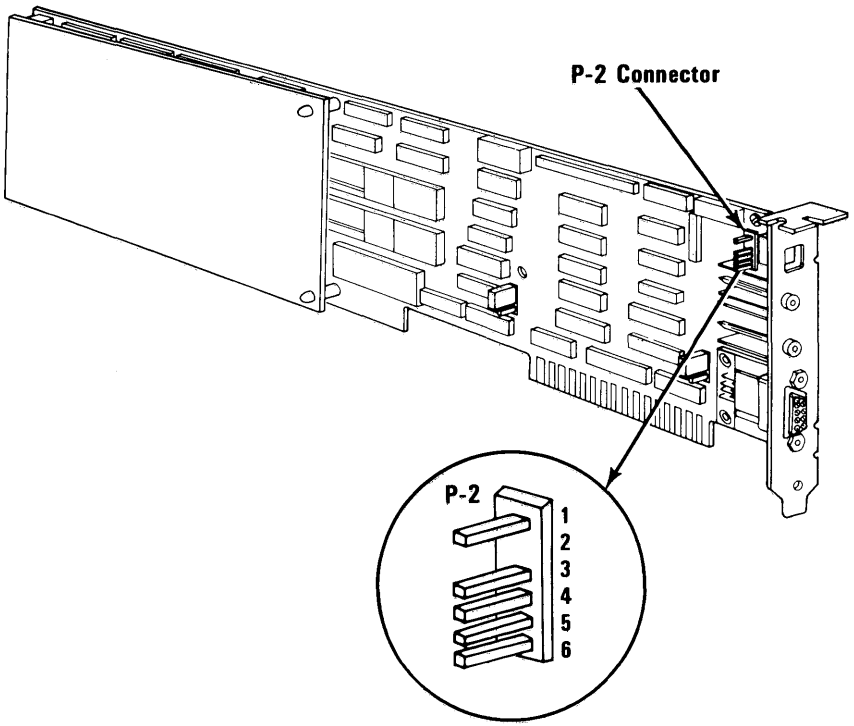
Switch Settings for Enhanced Graphics Adapter as Secondary Display Adapter						
				Configuration		
SW1	SW2	SW3	SW4	Enhanced Adapter	Monochrome Adapter	Color/Graphics Adapter
On	On	On	On	Color Display 40x25	Primary	–
Off	On	On	On	Color Display 80x25	Primary	–
On	Off	On	On	Enhanced Display Emulation Mode	Primary	–
Off	Off	On	On	Enhanced Display Hi Res Mode	Primary	–
On	On	Off	On	Monochrome	–	Primary 40x25
Off	On	Off	On	Monochrome	–	Primary 80x25

Direct Drive Connector



	Signal Name - Description	Pin	
Direct Drive Display	Ground	1	**Enhanced Graphics Adapter**
	Secondary Red	2	
	Primary Red	3	
	Primary Green	4	
	Primary Blue	5	
	Secondary Green/Intensity	6	
	Secondary Blue/Mono Video	7	
	Horizontal Retrace	8	
	Vertical Retrace	9	

Light Pen Interface



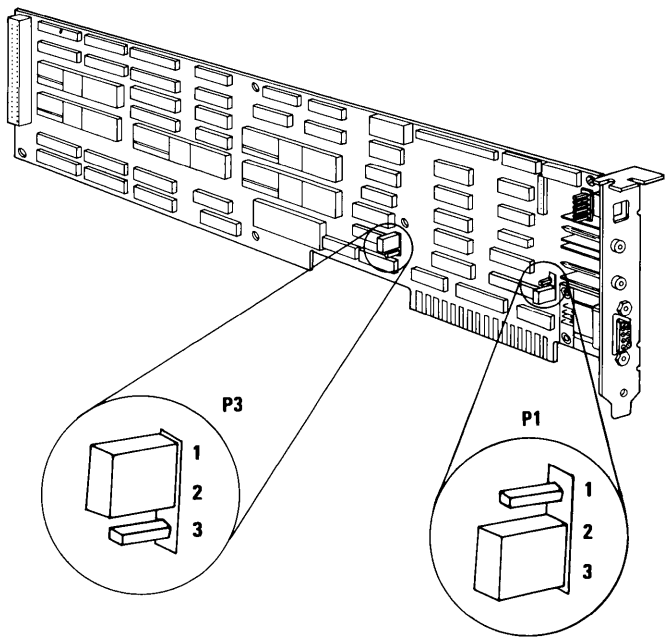
	P-2 Connector	Pin	
Light Pen Attachment	+Light Pen Input	1	**Enhanced Graphics Adapter**
	Not used	2	
	+Light Pen Switch	3	
	Ground	4	
	+5 Volts	5	
	12 Volts	6	

Jumper Descriptions

Located on the adapter are two jumpers designated P1 and P3. Jumper P1 changes the function of pin 2 on the direct drive interface. When placed on pins 2 and 3, jumper P1 selects ground as the function of direct drive interface, pin 2. This selection is for displays that support five color outputs, such as the IBM Color Display. When P1 is placed on pins 1 and 2, red prime output is placed on pin 2 of the direct drive interface connector. This supports the IBM Enhanced Color Display, which utilizes six color outputs on the direct drive interface.

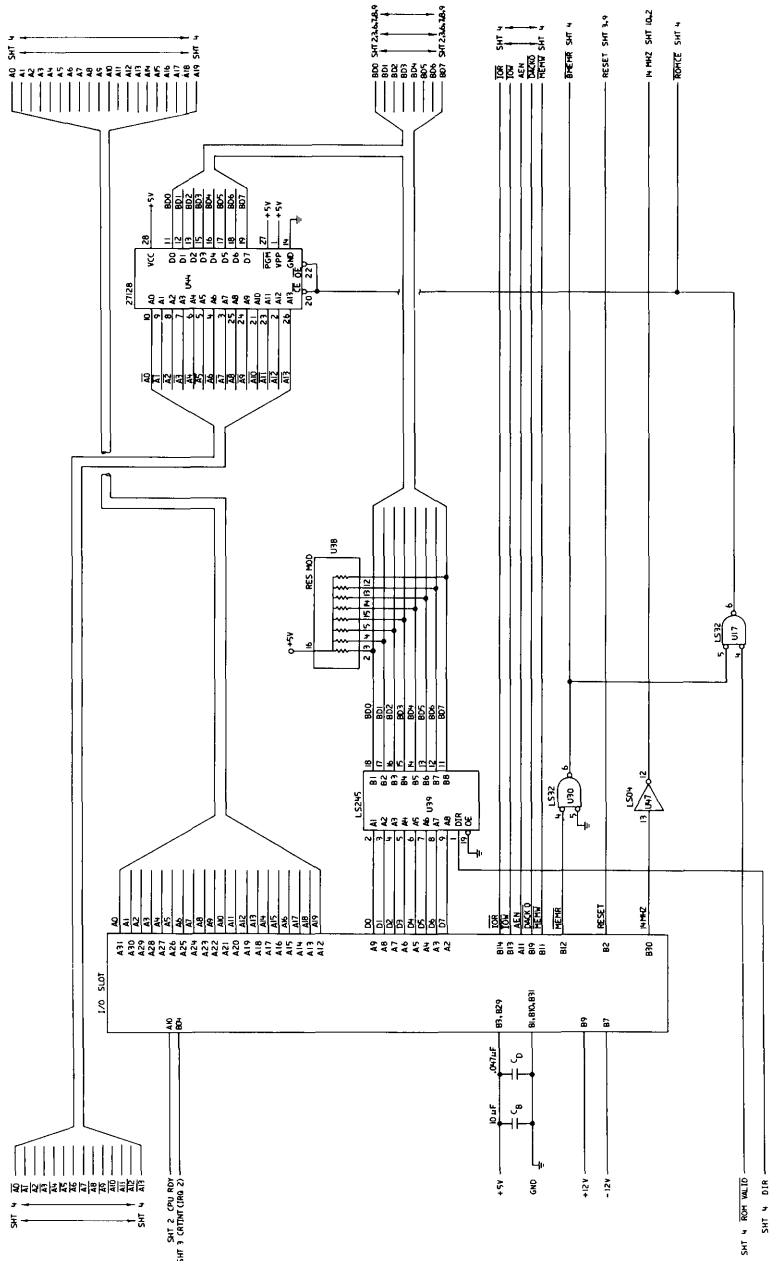
Jumper P3 changes the I/O address port of the Enhanced Graphics Adapter within the system. In its normal position, (pins 1 and 2), all Enhanced Graphics Adapter addresses are in the range 3XX. Moving jumper P3 to pins 2 and 3 changes the addresses to 2XX. Operation of the adapter in the 2XX mode is not supported in BIOS.

The following figure shows the location of the jumpers and numbering of the connectors.

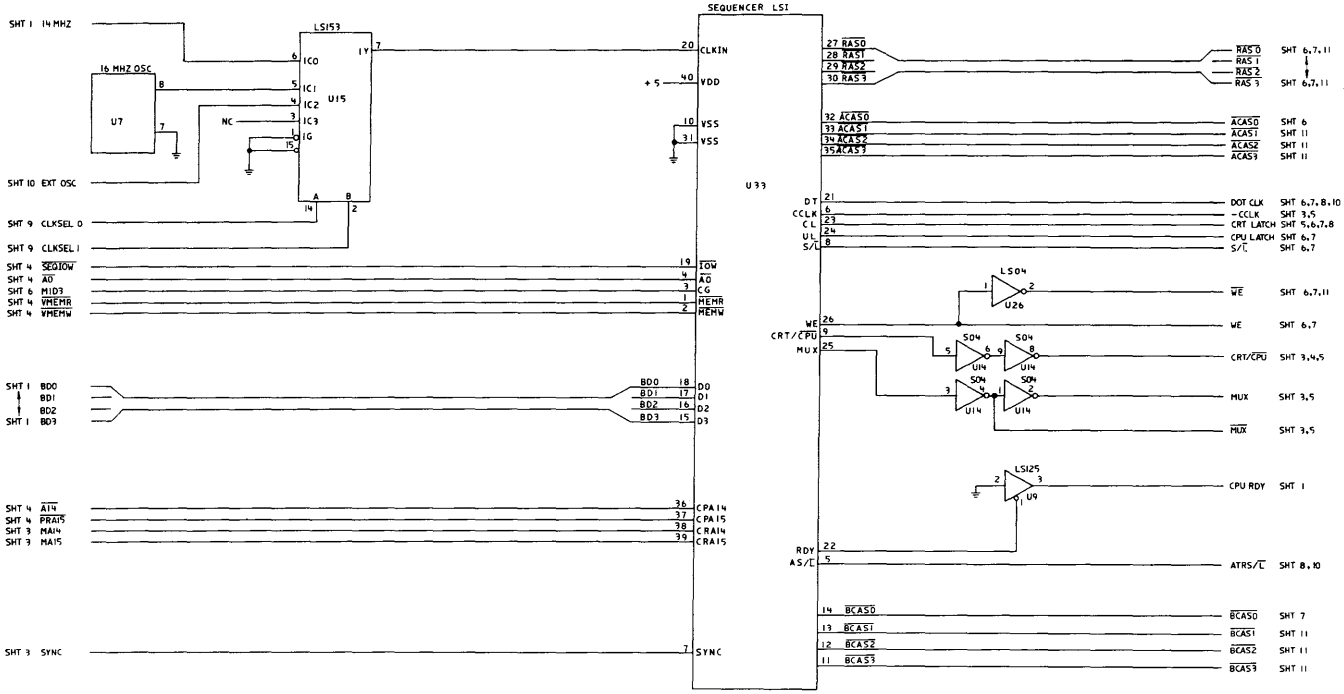


Logic Diagrams

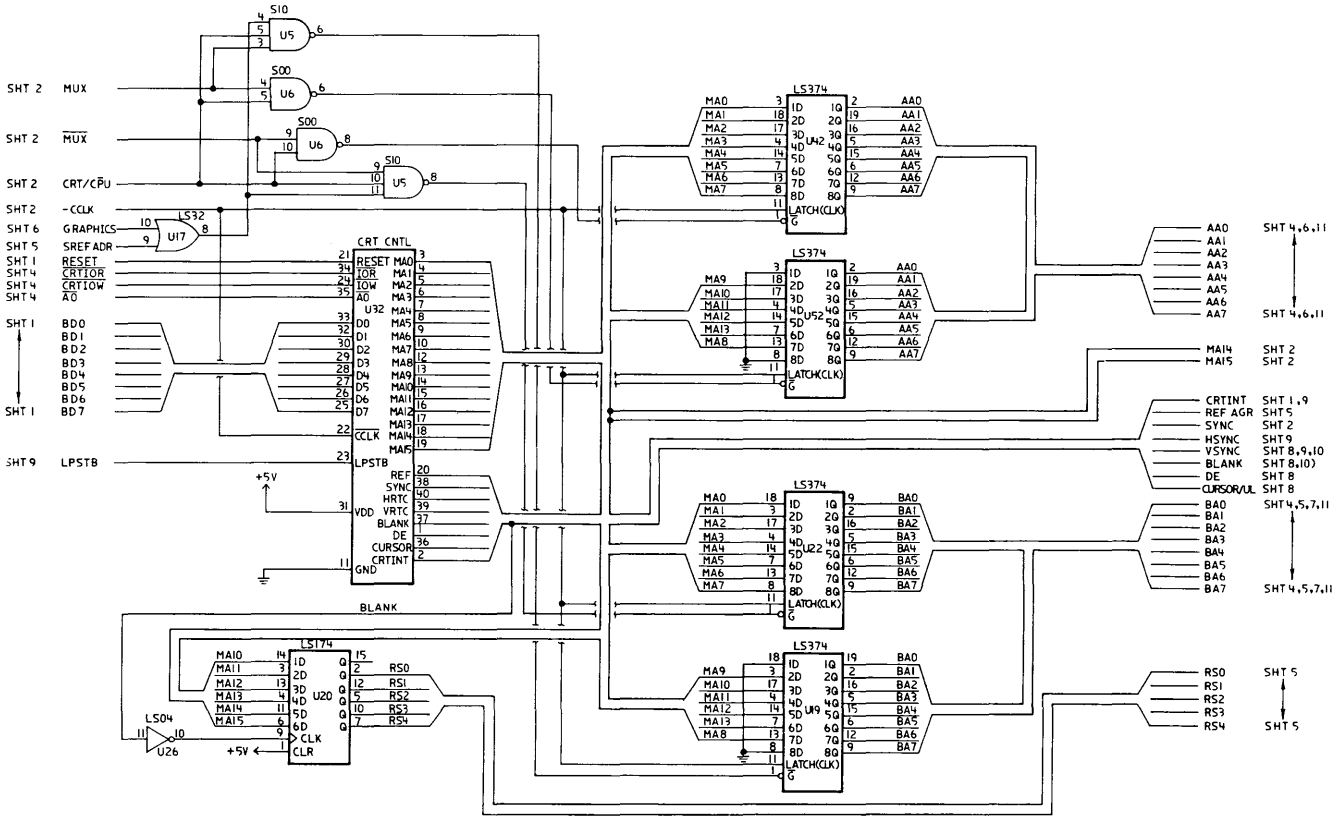
ENHANCED GRAPHICS ADAPTER



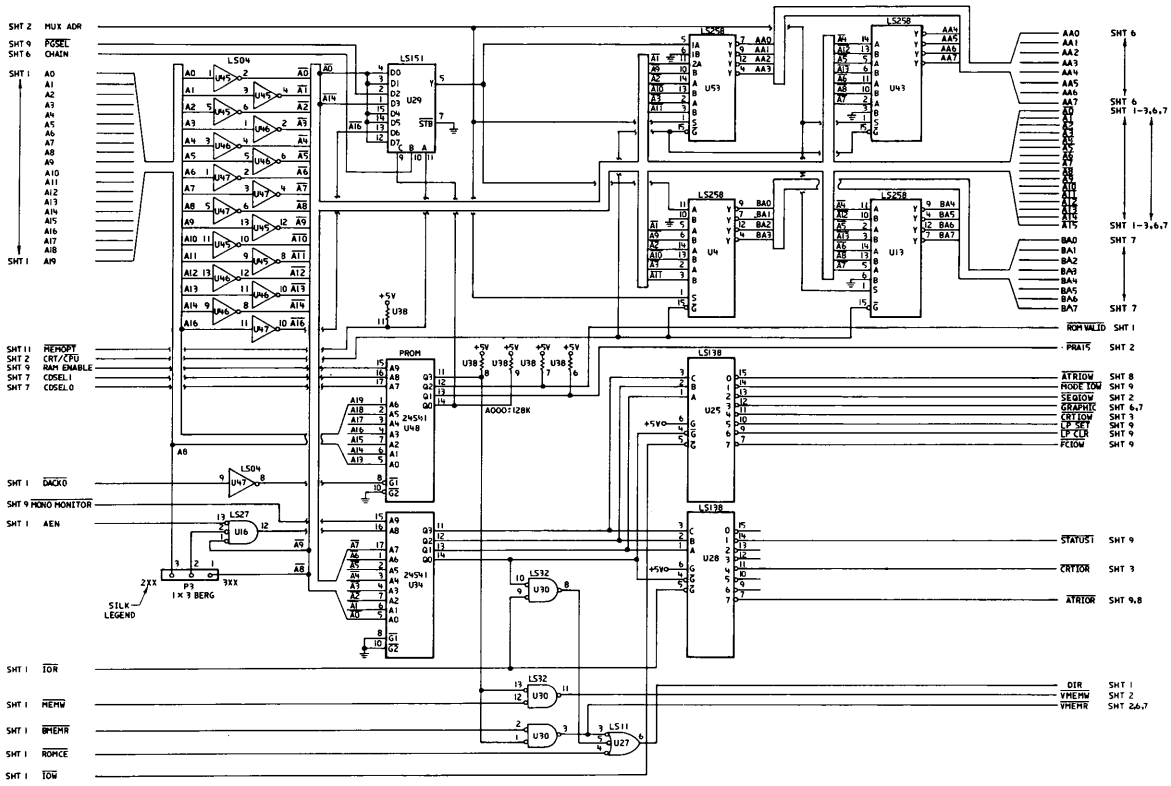
Enhanced Graphics Adapter Sheet 1 of 11



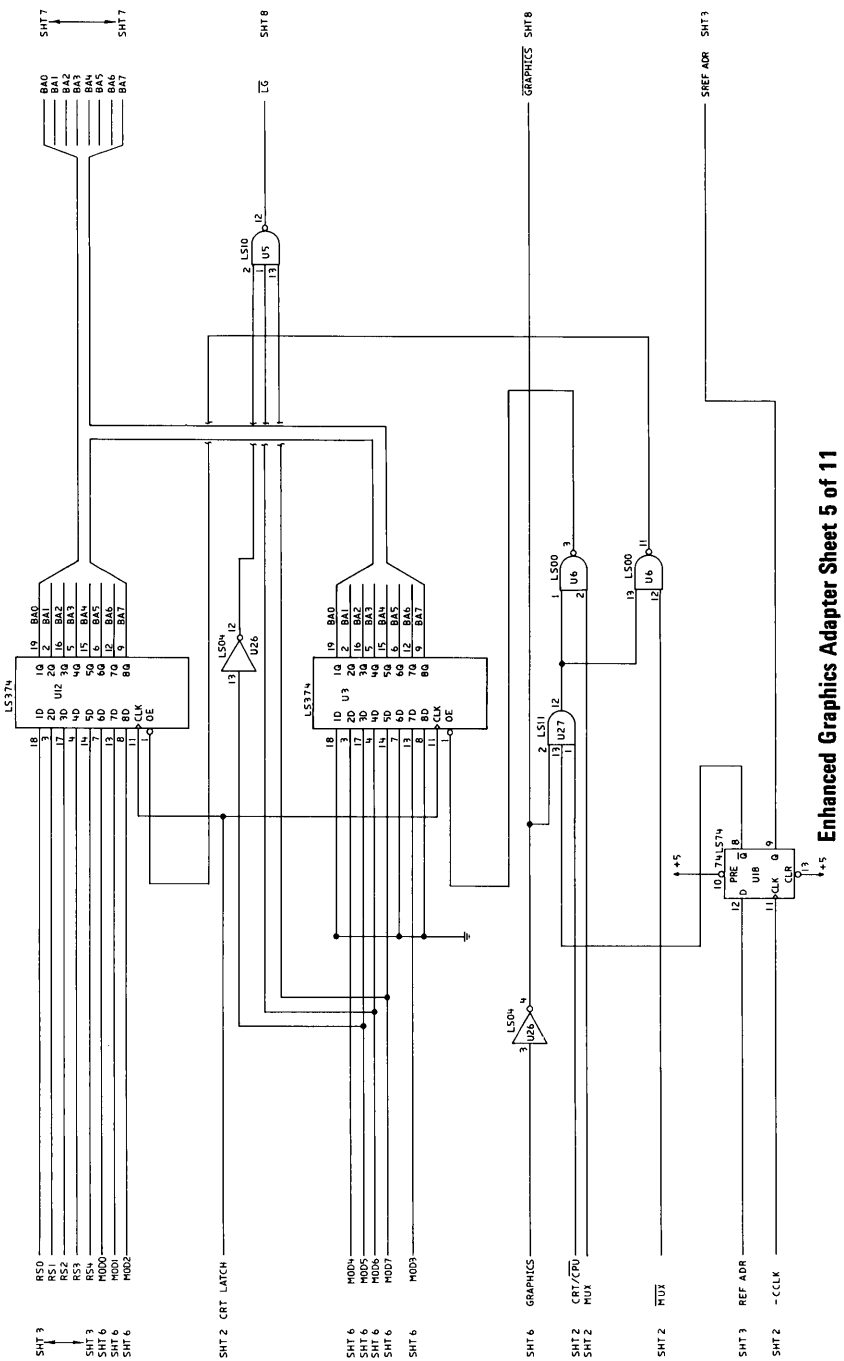
Enhanced Graphics Adapter Sheet 2 of 11



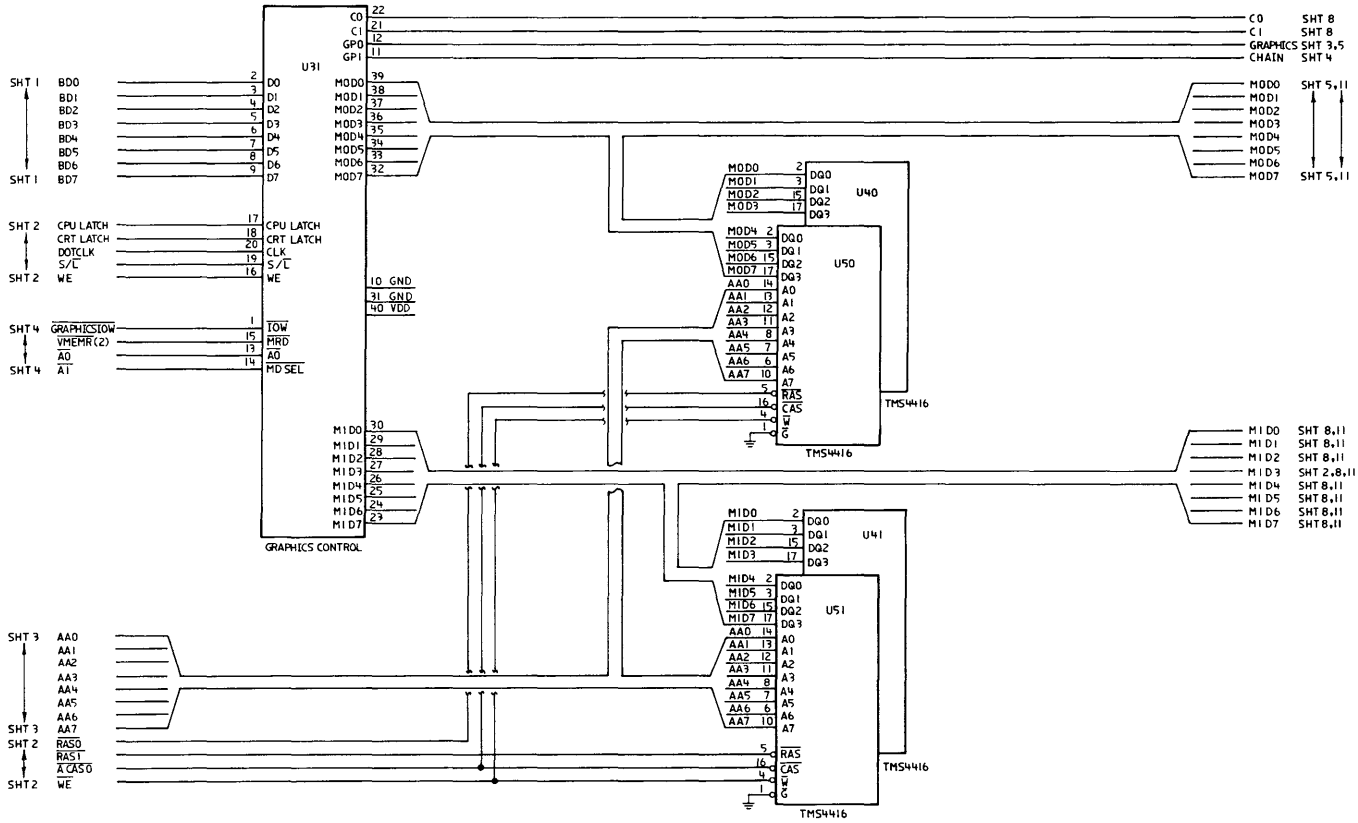
Enhanced Graphics Adapter Sheet 3 of 11



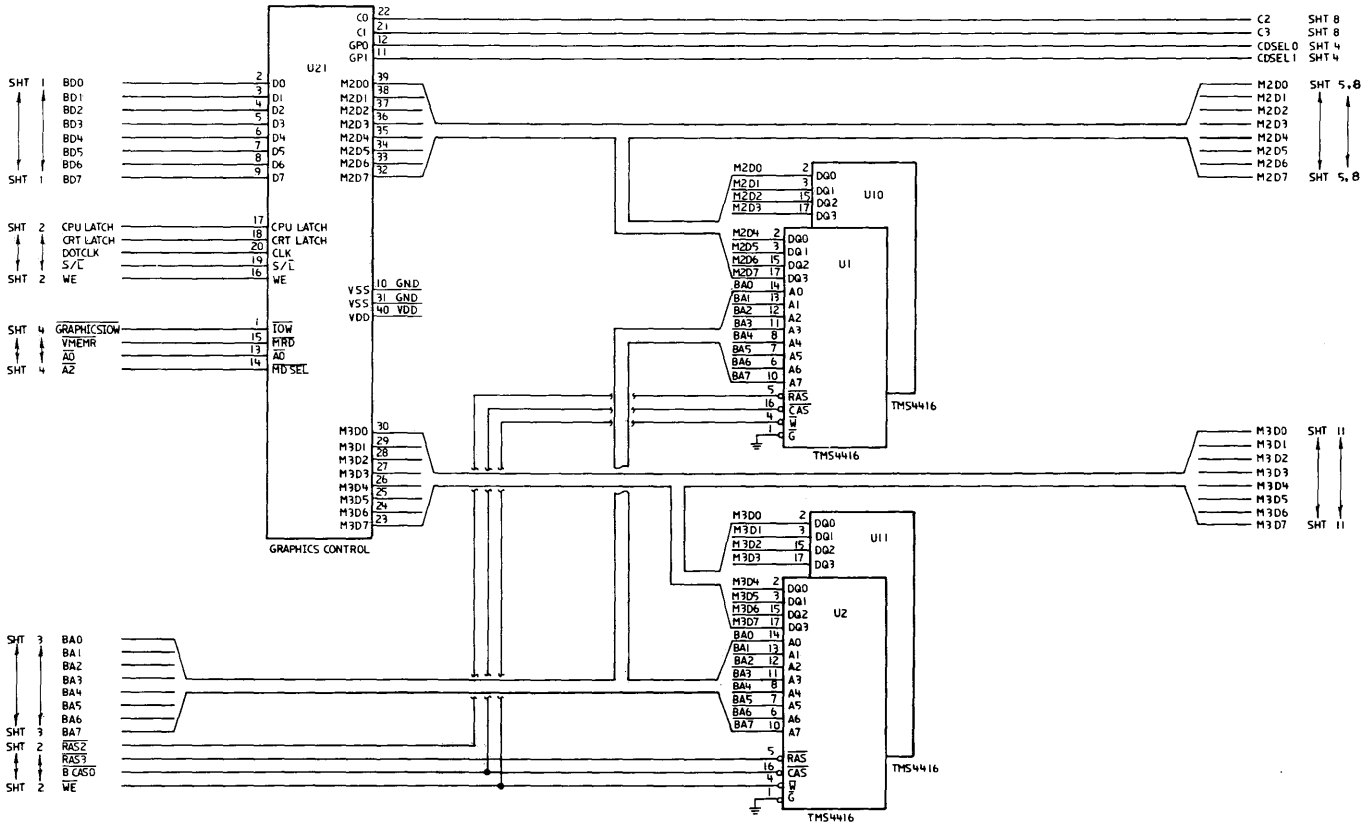
Enhanced Graphics Adapter Sheet 4 of 11



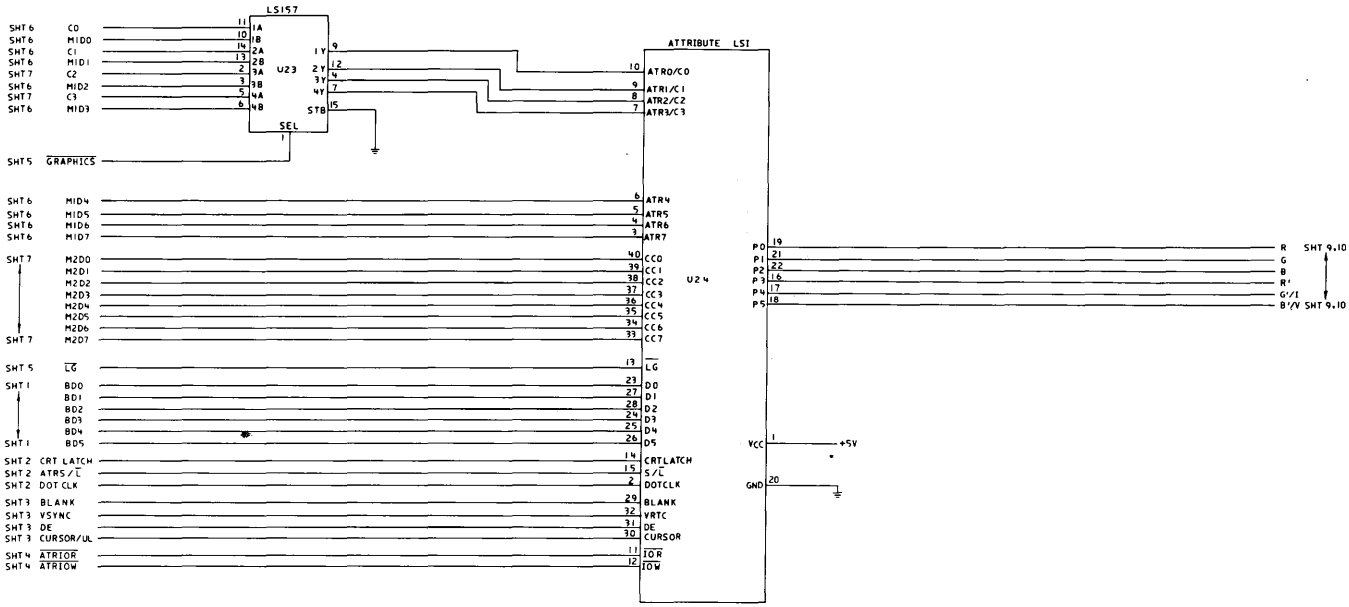
Enhanced Graphics Adapter Sheet 5 of 11



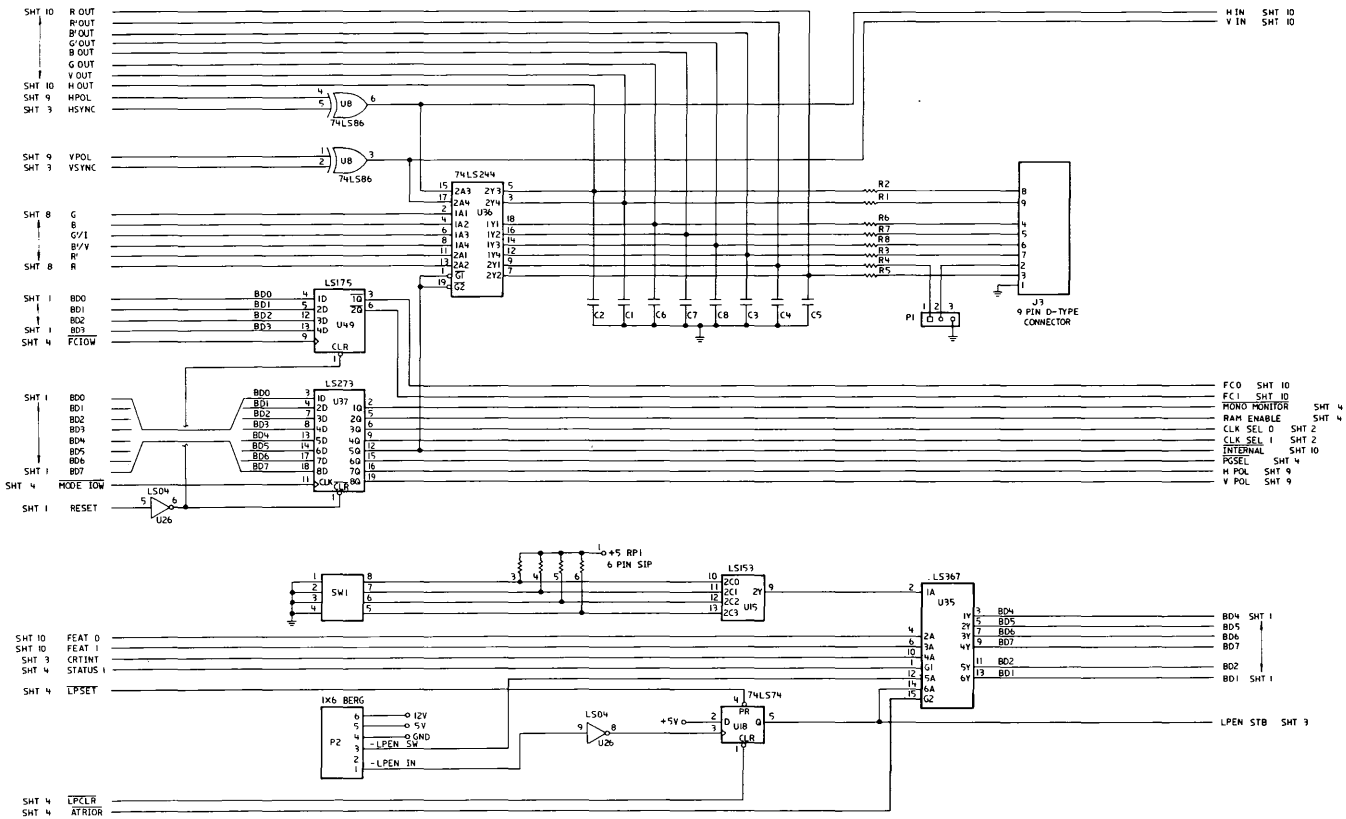
Enhanced Graphics Adapter Sheet 6 of 11



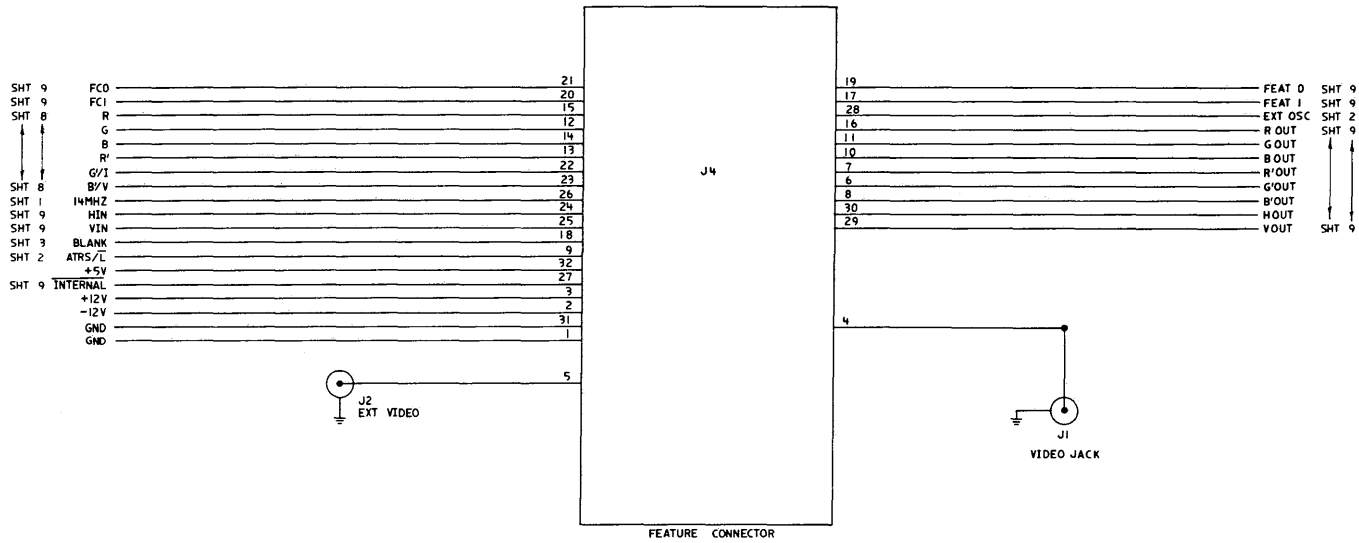
Enhanced Graphics Adapter Sheet 7 of 11



Enhanced Graphics Adapter Sheet 8 of 11



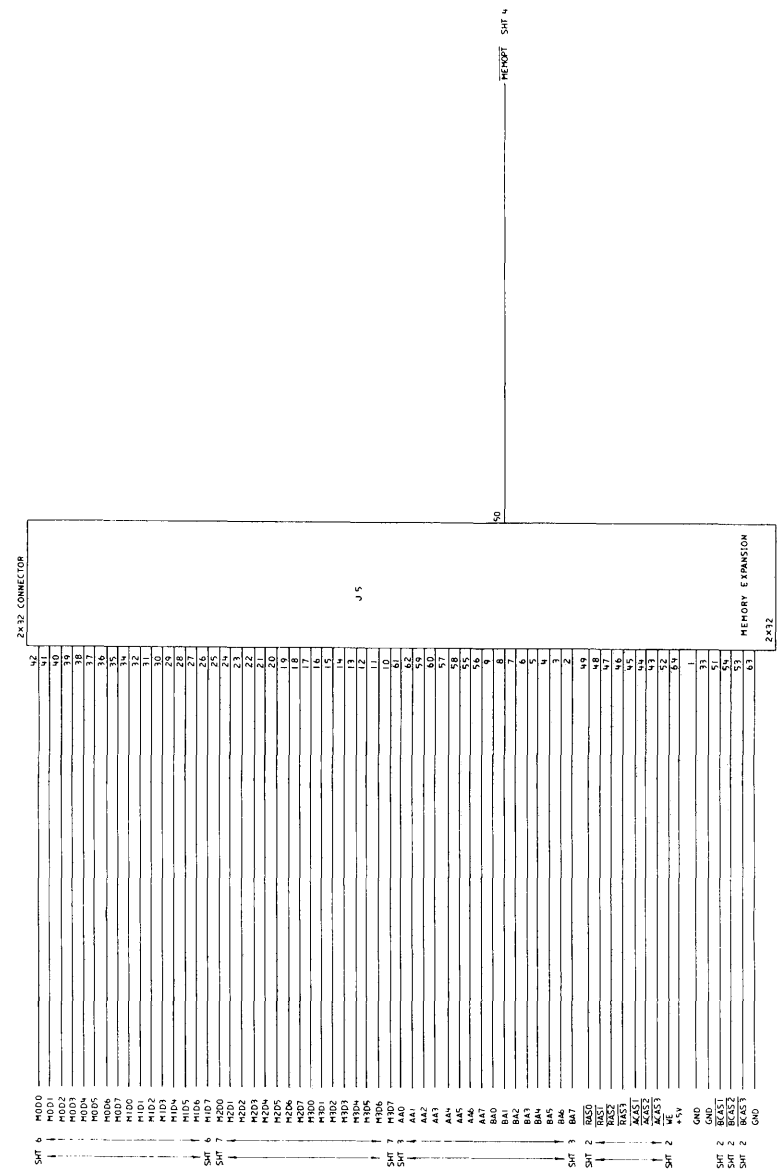
Enhanced Graphics Adapter Sheet 9 of 11



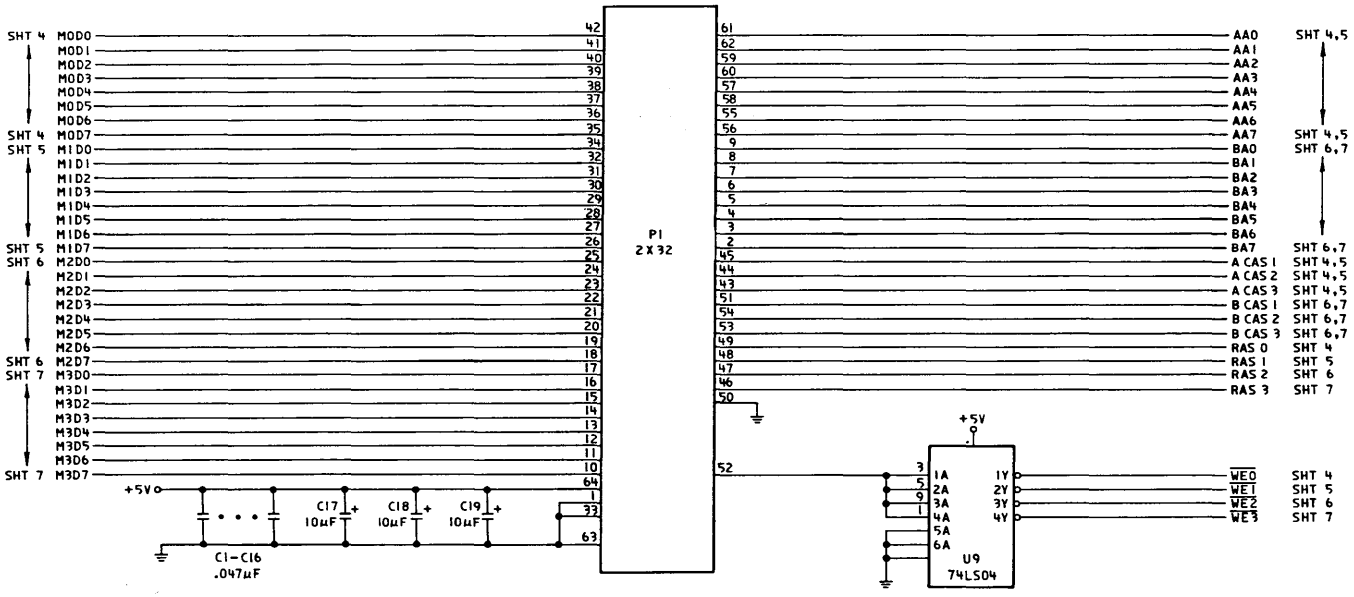
NOTE :

1 GROUNDS – ONE AT EACH END OF CONNECTOR.

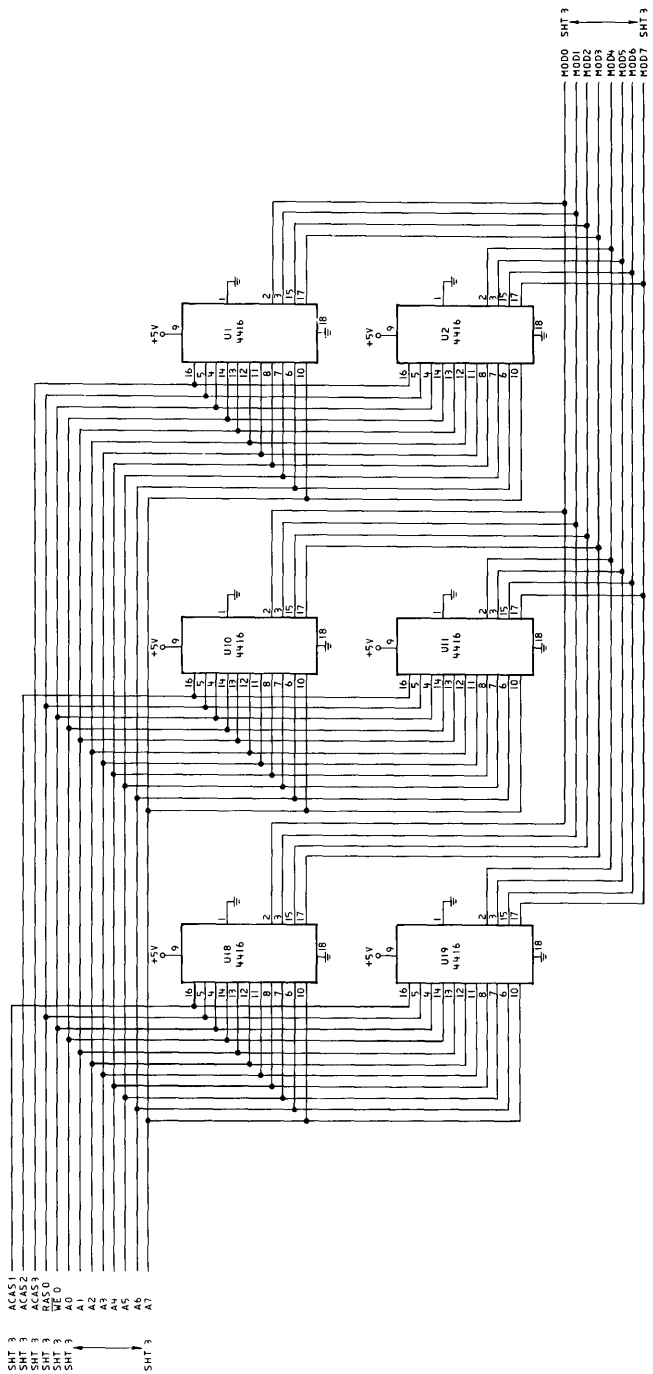
Enhanced Graphics Adapter Sheet 10 of 11



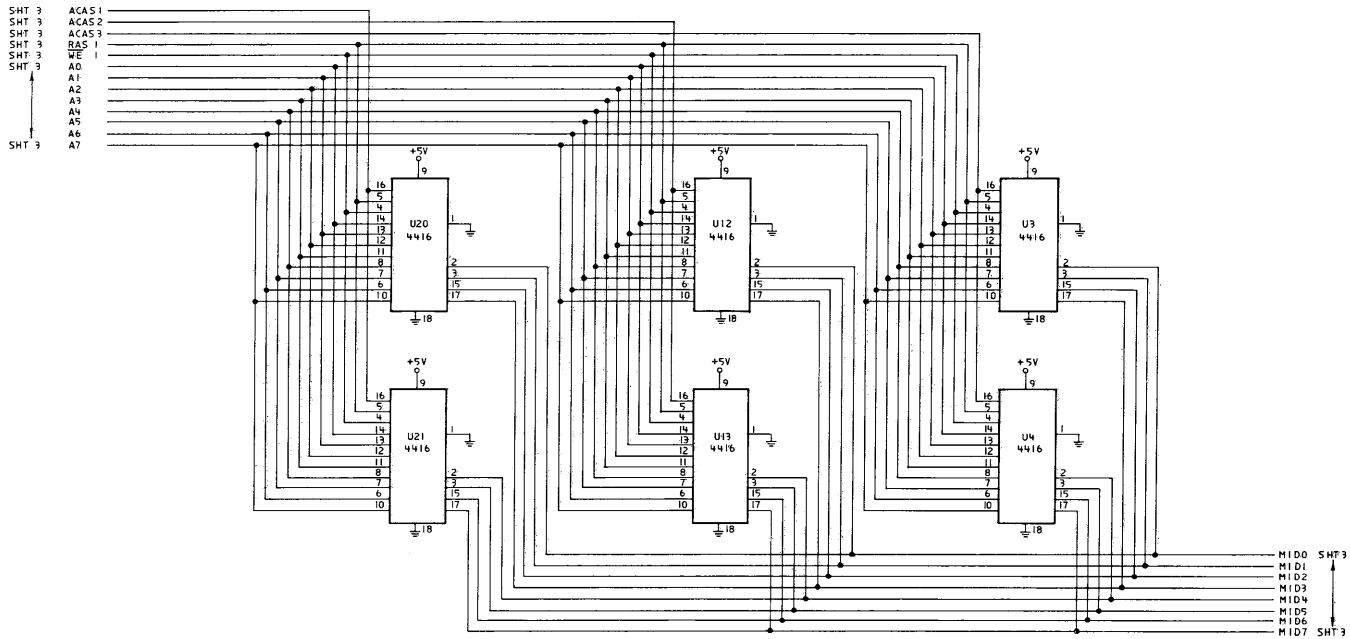
Enhanced Graphics Adapter Sheet 11 of 11



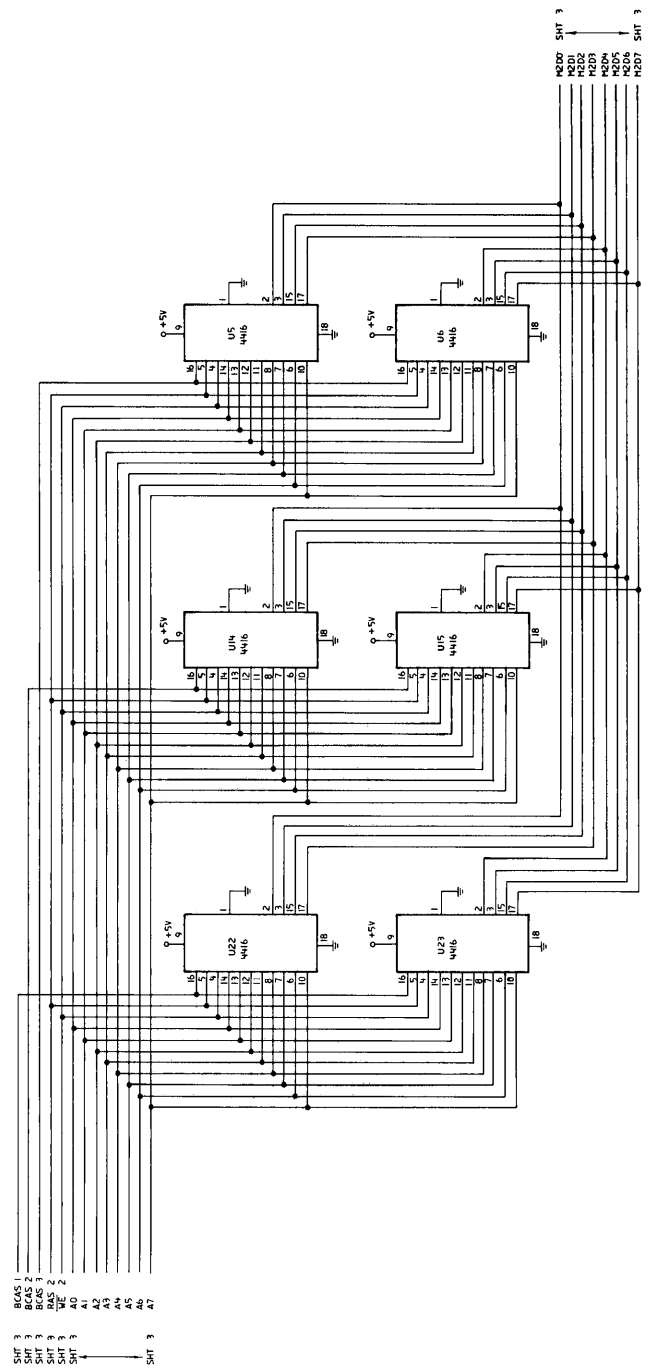
Graphics Memory Expansion Card Sheet 1 of 5



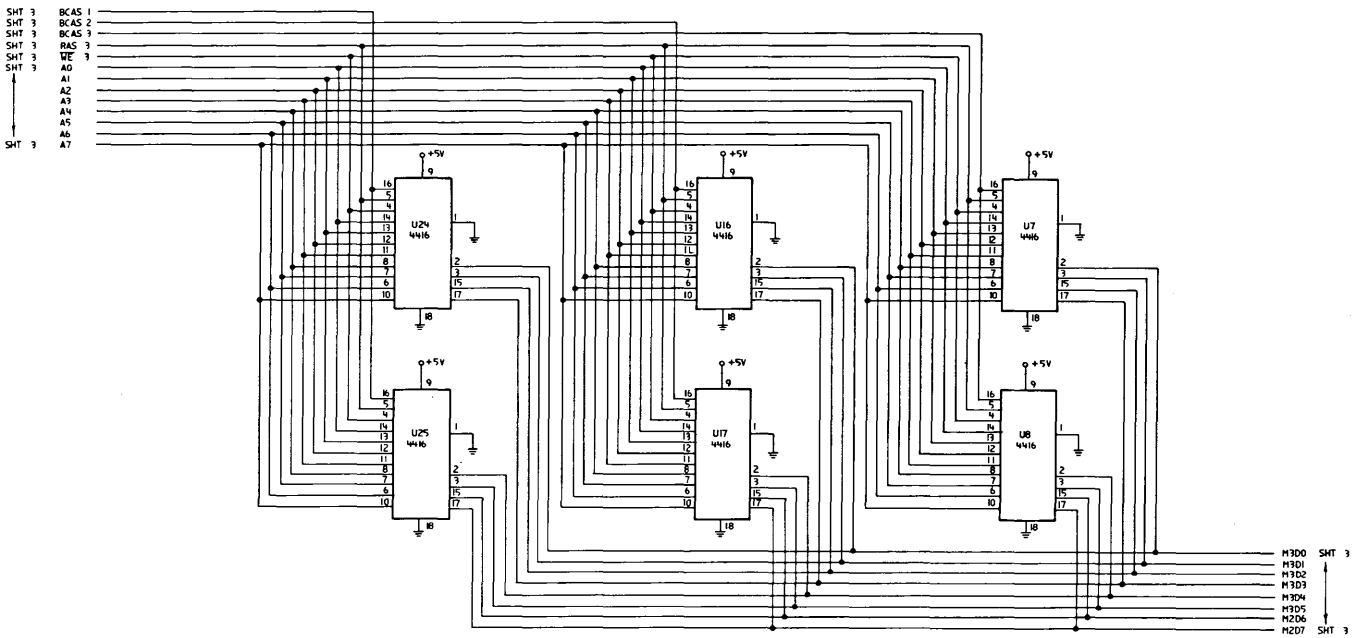
Graphics Memory Expansion Card Sheet 2 of 5



Graphics Memory Expansion Card Sheet 3 of 5



Graphics Memory Expansion Card Sheet 4 of 5



Graphics Memory Expansion Card Sheet 5 of 5

BIOS Listing

Vectors with Special Meanings

Interrupt Hex 42 - Reserved

When an IBM Enhanced Graphics Adapter is installed, the BIOS routines use interrupt 42 to revector the video pointer.

Interrupt Hex 43 - IBM Enhanced Graphics Video Parameters

When an IBM Enhanced Graphics Adapter is installed, the BIOS routines use this vector to point to a data region containing the parameters required for the initializing of the IBM Enhanced Graphics Adapter. Note that the format of the table must adhere to the BIOS conventions established in the listing. The power-on routines initialize this vector to point to the parameters contained in the IBM Enhanced Graphics Adapter ROM.

Interrupt Hex 44 - Graphics Character Table

When an IBM Enhanced Graphics Adapter is installed the BIOS routines use this vector to point to a table of dot patterns that will be used when graphics characters are to be displayed. This table will be used for the first 128 code points in video modes 4, 5, and 6. This table will be used for 256 characters in all additional graphics modes. See the appropriate BIOS interface for additional information on setting and using the graphics character table pointer.

```
                        PAGE,120
                        TITLE   ENHANCED GRAPHICS ADAPTER BIOS
                        EXTRN   CGMN:NEAR, CGDDOT:NEAR, INT_1F_1:NEAR, CGMN_FDG:NEAR
                        EXTRN   END_ADDRESS:NEAR

                ;--------------------------------------------------------------
                ;       THE  BIOS ROUTINES ARE MEANT TO BE ACCESSED THROUGH    :
                ;       SOFTWARE INTERRUPTS ONLY.  ANY ADDRESSES PRESENT IN    :
                ;       THE LISTINGS  ARE INCLUDED  ONLY FOR  COMPLETENESS,    :
                ;       NOT FOR  REFERENCE.   APPLICATIONS WHICH  REFERENCE    :
                ;       ABSOLUTE   ADDRESSES   WITHIN   THE   CODE  SEGMENT    :
                ;       VIOLATE THE STRUCTURE AND DESIGN OF BIOS.              :
                ;--------------------------------------------------------------

= 0000          WNR             EQU     0

                .list
C               INCLUDE         VFRONT.INC
C               SUBTTL  VFRONT.INC
C               PAGE
C
C       ;--- INT 10 -----------------------------------------------------------
C       ; VIDEO_IO                                                           :
C       ;       THESE ROUTINES PROVIDE THE CRT INTERFACE                     :
C       ;       THE FOLLOWING FUNCTIONS ARE PROVIDED:                        :
C       ;       (AH)=0  SET MODE (AL) CONTAINS MODE VALUE                    :
C       ;                                                                    :
C       ;       AL AD   TYPE    RES       NOTES  DF-DIM  DISPLAY      MAX PGS  :
C       ;       ------------------------------------------------------------ :
C       ;     *  0 B8   ALPHA   640X200          40X25   COLOR - BW      8     :
C       ;        1 B8   ALPHA   640X200          40X25   COLOR           8     :
C       ;     *  2 B8   ALPHA   640X200          80X25   COLOR - BW      8     :
C       ;        3 B8   ALPHA   640X200          80X25   COLOR           8     :
C       ;        4 B8   GRPHX   320X200          40X25   COLOR           1     :
C       ;        5 B8   GRPHX   320X200          40X25   COLOR - BW      1     :
C       ;        6 B8   GRPHX   640X200          80X25   COLOR - BW      1     :
C       ;     *  7 B0   ALPHA   720X350          80X25   MONOCHROME      8     :
C       ;                                                                    :
C       ;        8      RESERVED                                             :
C       ;        9      RESERVED                                             :
C       ;        A      RESERVED                                             :
C       ;        B      RESERVED - INTERNAL USE                              :
C       ;        C      RESERVED - INTERNAL USE                              :
C       ;                                                                    :
C       ;        D A0   GRPHX   320X200          40X25   COLOR           8     :
C       ;        E A0   GRPHX   640X200          80X25   COLOR           4     :
C       ;        F A0   GRPHX   640X350          80X25   MONOCHROME      2     :
C       ;       10 A0   GRPHX   640X350          80X25   hi res          2     :
C       ;                                                                    :
C       ;                                                                    :
C       ;               NOTE : HIGH BIT AL SET PREVENTS REGEN BUFFER CLEAR ON :
C       ;                      MODES RUNNING ON THE COMBO VIDEO ADAPTER       :
C       ;                                                                    :
C       ;               *** NOTE BW MODES OPERATE SAME AS COLOR MODES, BUT    :
C       ;                       COLOR BURST IS NOT ENABLED                   :
C       ;       (AH)=1  SET CURSOR TYPE                                      :
C       ;               (CH) =  BITS 4-0 = START LINE FOR CURSOR             :
C       ;                       ** HARDWARE WILL ALWAYS CAUSE BLINK          :
C       ;                       ** SETTING BIT 5 OR 6 WILL CAUSE ERRATIC     :
C       ;                          BLINKING OR NO CURSOR AT ALL              :
C       ;               (CL) =  BITS 4-0 = END LINE FOR CURSOR               :
C       ;       (AH)=2  SET CURSOR POSITION                                  :
C       ;               (DH,DL) = ROW,COLUMN  (0,0) IS UPPER LEFT            :
C       ;               (BH) = PAGE NUMBER                                   :
C       ;       (AH)=3  READ CURSOR POSITION                                 :
C       ;               (BH) = PAGE NUMBER                                   :
C       ;               ON EXIT (DH,DL) = ROW,COLUMN OF CURRENT CURSOR       :
C       ;                       (CH,CL) = CURSOR MODE CURRENTLY SET          :
C       ;       (AH)=4  READ LIGHT PEN POSITION                              :
C       ;               ON EXIT:                                             :
C       ;               (AH) = 0 -- LIGHT PEN SWITCH NOT DOWN/NOT TRIGGERED  :
C       ;               (AH) = 1 -- VALID LIGHT PEN VALUE IN REGISTERS       :
C       ;                       (DH,DL) = ROW,COLUMN OF CHARACTER LP POSN    :
C       ;                       (CH) = RASTER LINE (0-199)                   :
C       ;                       (CX) = RASTER LINE (0-NNN) NEW GRAPHICS MODES :
C       ;                       (BX) = PIXEL COLUMN (0-319,639)              :
C       ;       (AH)=5  SELECT ACTIVE DISPLAY PAGE                           :
C       ;               (AL) = NEW PAGE VALUE, SEE AH=0 FOR PAGE INFO        :
C       ;       (AH)=6  SCROLL ACTIVE PAGE UP                                :
C       ;               (AL) = NUMBER OF LINES, INPUT LINES BLANKED AT BOTTOM :
C       ;                       OF WINDOW                                    :
C       ;                       AL = 0 MEANS BLANK ENTIRE WINDOW             :
C       ;               (CH,CL) = ROW,COLUMN OF UPPER LEFT CORNER OF SCROLL  :
C       ;               (DH,DL) = ROW,COLUMN OF LOWER RIGHT CORNER OF SCROLL :
C       ;               (BH) = ATTRIBUTE TO BE USED ON BLANK LINE            :
C       ;       (AH)=7  SCROLL ACTIVE PAGE DOWN                              :
C       ;               (AL) = NUMBER OF LINES, INPUT LINES BLANKED AT TOP   :
C       ;                       OF WINDOW                                    :
C       ;                       AL = 0 MEANS BLANK ENTIRE WINDOW             :
C       ;               (CH,CL) = ROW,COLUMN OF UPPER LEFT CORNER OF SCROLL  :
C       ;               (DH,DL) = ROW,COLUMN OF LOWER RIGHT CORNER OF SCROLL :
C       ;               (BH) = ATTRIBUTE TO BE USED ON BLANK LINE            :
C       ;                                                                    :
C       ;       CHARACTER HANDLING ROUTINES                                  :
C       ;                                                                    :
C       ;       (AH) = 8 READ ATTRIBUTE/CHARACTER AT CURRENT CURSOR POSITION :
C       ;               (BH) = DISPLAY PAGE                                  :
C       ;               ON EXIT:                                             :
C       ;               (AL) = CHAR READ                                     :
C       ;               (AH) = ATTRIBUTE OF CHARACTER READ (ALPHA MODES ONLY) :
C       ;       (AH) = 9 WRITE ATTRIBUTE/CHARACTER AT CURRENT CURSOR POSITION :
C       ;               (BH) = DISPLAY PAGE                                  :
C       ;               (CX) = COUNT OF CHARACTERS TO WRITE                  :
C       ;               (AL) = CHAR TO WRITE                                 :
C       ;               (BL) = ATTRIBUTE OF CHARACTER (ALPHA)/COLOR OF CHAR  :
C       ;                       (GRAPHICS)                                   :
C       ;                       SEE NOTE ON WRITE DOT FOR BIT 7 OF BL = 1.   :
C       ;       (AH) = A  WRITE CHARACTER ONLY AT CURRENT CURSOR POSITION    :
C       ;               (BH) = DISPLAY PAGE                                  :
C       ;               (CX) = COUNT OF CHARACTERS TO WRITE                  :
C       ;               (AL) = CHAR TO WRITE                                 :
C       ;       FOR READ/WRITE CHARACTER INTERFACE WHILE IN GRAPHICS MODE, THE :
C       ;               CHARACTERS ARE FORMED FROM A CHARACTER GENERATOR IMAGE :
C       ;               MAINTAINED IN THE SYSTEM ROM.  ONLY THE 1ST 128 CHARS :
C       ;               ARE CONTAINED THERE.  TO READ/WRITE THE SECOND 128    :
C       ;               CHARS, THE USER MUST INITIALIZE THE POINTER AT        :
C       ;               INTERRUPT 1FH (LOCATION 0007CH) TO POINT TO THE 1K BYTE :
C       ;               TABLE CONTAINING THE CODE POINTS FOR THE SECOND       :
C       ;               128 CHARS (128-255).                                 :
C       ;                                                                    :
C       ;               FOR THE NEW GRAPHICS MODES 256 GRAPHICS CHARS ARE    :
C       ;               SUPPLIED IN THE SYSTEM ROM.                          :
C       ;                                                                    :
C       ;       FOR WRITE CHARACTER INTERFACE IN GRAPHICS MODE, THE REPLICATION :
C       ;               FACTOR CONTAINED IN (CX) ON ENTRY WILL PRODUCE VALID  :
C       ;               RESULTS ONLY FOR CHARACTERS CONTAINED ON THE SAME ROW. :
```

```
127     C ;             CONTINUATION TO SUCCEEDING LINES WILL NOT PRODUCE
128     C ;             CORRECTLY.
129     C ;
130     C ;     GRAPHICS INTERFACE
131     C ;     (AH) = B  SET COLOR PALETTE
132     C ;             FOR USE IN COMPATIBILITY MODES
133     C ;             (BH) = PALETTE COLOR ID BEING SET (0-127)
134     C ;             (BL) = COLOR VALUE TO BE USED WITH THAT COLOR ID
135     C ;                NOTE: FOR THE CURRENT COLOR CARD, THIS ENTRY POINT
136     C ;                      HAS MEANING ONLY FOR 320X200 GRAPHICS.
137     C ;                      COLOR ID = 0 SELECTS THE BACKGROUND COLOR (0-15)
138     C ;                      COLOR ID = 1 SELECTS THE PALETTE TO BE USED:
139     C ;                                   0 = GREEN(1)/RED(2)/BROWN(3)
140     C ;                                   1 = CYAN(1)/MAGENTA(2)/WHITE(3)
141     C ;                      IN 40X25 OR 80X25 ALPHA MODES, THE VALUE SET
142     C ;                                FOR PALETTE COLOR 0 INDICATES THE
143     C ;                                BORDER COLOR TO BE USED (VALUES 0-31,
144     C ;                                WHERE 16-31 SELECT THE HIGH INTENSITY
145     C ;                                BACKGROUND SET).
146     C ;     (AH) = C  WRITE DOT
147     C ;             (BH) = PAGE
148     C ;             (DX) = ROW NUMBER
149     C ;             (CX) = COLUMN NUMBER
150     C ;             (AL) = COLOR VALUE
151     C ;                     IF BIT 7 OF AL = 1, THEN THE COLOR VALUE IS
152     C ;                     EXCLUSIVE OR'D WITH THE CURRENT CONTENTS OF
153     C ;                     THE DOT
154     C ;     (AH) = D  READ DOT
155     C ;             (BH) = PAGE
156     C ;             (DX) = ROW NUMBER
157     C ;             (CX) = COLUMN NUMBER
158     C ;             (AL) RETURNS THE DOT READ
159     C ;
160     C ; ASCII TELETYPE ROUTINE FOR OUTPUT
161     C ;
162     C ;     (AH) = E  WRITE TELETYPE TO ACTIVE PAGE
163     C ;             (AL) = CHAR TO WRITE
164     C ;             (BL) = FOREGROUND COLOR IN GRAPHICS MODE
165     C ;             NOTE -- SCREEN WIDTH IS CONTROLLED BY PREVIOUS MODE SET
166     C ;
167     C ;     (AH) = F  CURRENT VIDEO STATE
168     C ;             RETURNS THE CURRENT VIDEO STATE
169     C ;             (AL) = MODE CURRENTLY SET  (SEE AH=0 FOR EXPLANATION)
170     C ;             (AH) = NUMBER OF CHARACTER COLUMNS ON SCREEN
171     C ;             (BH) = CURRENT ACTIVE DISPLAY PAGE
172     C ;
173     C ;     (AH) = 10  SET PALETTE REGISTERS
174     C ;
175     C ;             (AL) = 0        SET INDIVIDUAL PALETTE REGISTER
176     C ;              BL = PALETTE REGISTER TO BE SET
177     C ;              BH = VALUE TO SET
178     C ;
179     C ;             AL = 1          SET OVERSCAN REGISTER
180     C ;             BH = VALUE TO SET
181     C ;
182     C ;             AL = 2          SET ALL PALETTE REGISTERS AND OVERSCAN
183     C ;             ES:DX POINTS TO A 17 BYTE TABLE
184     C ;             BYTES 0 - 15 ARE THE PALETTE VALUES, RESPECTIVELY
185     C ;             BYTE 16 IS THE OVERSCAN VALUE
186     C ;
187     C ;             AL = 3          TOGGLE INTENSIFY/BLINKING BIT
188     C ;                     BL = 0  ENABLE INTENSIFY
189     C ;                     BL = 1  ENABLE BLINKING
190     C ;
191     C ;     (AH) = 11  CHARACTER GENERATOR ROUTINE
192     C ;             note : this call will initiate a mode set, completely
193     C ;                    resetting the video environment but maintaining
194     C ;                    the regen buffer.
195     C ;
196     C ;             AL = 00   USER ALPHA LOAD
197     C ;                     ES:BP - POINTER TO USER TABLE
198     C ;                     CX    - COUNT TO STORE
199     C ;                     DX    - CHARACTER OFFSET INTO TABLE
200     C ;                     BL    - BLOCK TO LOAD
201     C ;                     BH    - NUMBER OF BYTES PER CHARACTER
202     C ;             AL = 01   ROM MONOCHROME SET
203     C ;                     BL    - BLOCK TO LOAD
204     C ;             AL = 02   ROM 8X8 DOUBLE DOT
205     C ;                     BL    - BLOCK TO LOAD
206     C ;             AL = 03   SET BLOCK SPECIFIER
207     C ;                     BL    - CHAR GEN BLOCK SPECIFIER
208     C ;                             D3-D2  ATTR BIT 3 ONE,  CHAR GEN 0-3
209     C ;                             D1-D0  ATTR BIT 3 ZERO, CHAR GEN 0-3
210     C ;                     NOTE : WHEN USING AL = 03 A FUNCTION CALL
211     C ;                             AX = 1000H
212     C ;                             BX = 0712H
213     C ;                             IS RECOMMENDED TO SET THE COLOR PLANES
214     C ;                             RESULTING IN 512 CHARACTERS AND EIGHT
215     C ;                             CONSISTENT COLORS.
216     C ;
217     C ;     NOTE : THE FOLLOWING INTERFACE (AL=1X) IS SIMILAR IN FUNCTION
218     C ;             TO (AL=0X) EXCEPT THAT :
219     C ;                     - PAGE ZERO MUST BE ACTIVE
220     C ;                     - POINTS (BYTES/CHAR) WILL BE RECALCULATED
221     C ;                     - ROWS WILL BE CALCULATED FROM THE FOLLOWING:
222     C ;                             INT[(200 OR 350) / POINTS] - 1
223     C ;                     - CRT_LEN WILL BE CALCULATED FROM :
224     C ;                             (ROWS + 1) * CRT_COLS * 2
225     C ;                     - THE CRTC WILL BE REPROGRAMMED AS FOLLOWS :
226     C ;                             R09H = POINTS - 1       MAX SCAN LINE
227     C ;                                    R09H done only in mode 7
228     C ;                             R0AH = POINTS - 2       CURSOR START
229     C ;                             R0BH = 0                CURSOR END
230     C ;                             R12H =                  VERT DISP END
231     C ;                                    [(ROWS + 1) * POINTS] - 1
232     C ;                             R14H = POINTS           UNDERLINE LOC
233     C ;
234     C ;                     THE ABOVE REGISTER CALCULATIONS MUST BE CLOSE TO THE
235     C ;                     ORIGINAL TABLE VALUES OR UNDETERMINED RESULTS WILL
236     C ;                     OCCUR.
237     C ;
238     C ;                     NOTE : THE FOLLOWING INTERFACE IS DESIGNED TO BE
239     C ;                            CALLED ONLY IMMEDIATELY AFTER A MODE SET HAS
240     C ;                            BEEN ISSUED. FAILURE TO ADHERE TO THIS PRACTICE
241     C ;                            MAY CAUSE UNDETERMINED RESULTS.
242     C ;
243     C ;                     AL = 10   USER ALPHA LOAD
244     C ;                             ES:BP - POINTER TO USER TABLE
245     C ;                             CX    - COUNT TO STORE
246     C ;                             DX    - CHARACTER OFFSET INTO TABLE
247     C ;                             BL    - BLOCK TO LOAD
248     C ;                             BH    - NUMBER OF BYTES PER CHARACTER
249     C ;                     AL = 11   ROM MONOCHROME SET
250     C ;                             BL    - BLOCK TO LOAD
251     C ;                     AL = 12   ROM 8X8 DOUBLE DOT
252     C ;                             BL    - BLOCK TO LOAD
```

```
C  ;                                                                       :
C  ;                                                                       :
C  ;               NOTE : THE FOLLOWING INTERFACE IS DESIGNED TO BE        :
C  ;                      CALLED ONLY IMMEDIATELY AFTER A MODE SET HAS     :
C  ;                      BEEN ISSUED. FAILURE TO ADHERE TO THIS PRACTICE  :
C  ;                      MAY CAUSE UNDETERMINED RESULTS.                  :
C  ;                                                                       :
C  ;               AL = 20   USER GRAPHICS CHARS  INT 01FH (8X8)           :
C  ;                       ES:BP - POINTER TO USER TABLE                   :
C  ;               AL = 21   USER GRAPHICS CHARS                           :
C  ;                       ES:BP - POINTER TO USER TABLE                   :
C  ;                       CX    - POINTS (BYTES PER CHARACTER)            :
C  ;                       BL    - ROW SPECIFIER                           :
C  ;                                                                       :
C  ;                               BL = 0  USER                            :
C  ;                                       DL - ROWS                       :
C  ;                               BL = 1  14 (0EH)                        :
C  ;                               BL = 2  25 (19H)                        :
C  ;                               BL = 3  43 (2BH)                        :
C  ;                                                                       :
C  ;               AL = 22   ROM 8 X 14 SET                                :
C  ;                       BL    - ROW SPECIFIER                           :
C  ;               AL = 23   ROM 8 X 8 DOUBLE DOT                          :
C  ;                       BL    - ROW SPECIFIER                           :
C  ;                                                                       :
C  ;                                                                       :
C  ;               AL = 30   INFORMATION                                   :
C  ;                               CX    - POINTS                          :
C  ;                               DL    - ROWS                            :
C  ;                       BH    - 0     RETURN CURRENT INT 1FH PTR        :
C  ;                               ES:BP - PTR TO TABLE                    :
C  ;                       BH    - 1     RETURN CURRENT INT 44H PTR        :
C  ;                               ES:BP - PTR TO TABLE                    :
C  ;                       BH    - 2     RETURN ROM 8 X 14 PTR             :
C  ;                               ES:BP - PTR TO TABLE                    :
C  ;                       BH    - 3     RETURN ROM DOUBLE DOT PTR         :
C  ;                               ES:BP - PTR TO TABLE                    :
C  ;                       BH    - 4     RETURN ROM DOUBLE DOT PTR (TOP)   :
C  ;                               ES:BP - PTR TO TABLE                    :
C  ;                       BH    - 5     RETURN ROM ALPHA ALTERNATE 9X14   :
C  ;                               ES:BP - PTR TO TABLE                    :
C  ;                                                                       :
C  ;       (AH) = 12   ALTERNATE SELECT                                    :
C  ;                                                                       :
C  ;               BL = 10  RETURN EGA INFORMATION                         :
C  ;                       BH = 0 - COLOR MODE IN EFFECT <3><D><X>         :
C  ;                            1 - MONOC MODE IN EFFECT <3><B><X>         :
C  ;                       BL = MEMORY VALUE                               :
C  ;                               0 0 - 064K      0 1 - 128K              :
C  ;                               1 0 - 192K      1 1 - 256K              :
C  ;                       CH = FEATURE BITS                               :
C  ;                       CL = SWITCH SETTING                             :
C  ;                                                                       :
C  ;               BL = 20  SELECT ALTERNATE PRINT SCREEN ROUTINE          :
C  ;                                                                       :
C  ;       (AH) = 13   WRITE STRING                                        :
C  ;                       ES:BP - POINTER TO STRING TO BE WRITTEN         :
C  ;                       CX    - CHARACTER ONLY COUNT                    :
C  ;                       DX    - POSITION TO BEGIN STRING, IN CURSOR     :
C  ;                               TERMS                                   :
C  ;                       BH    - PAGE NUMBER                             :
C  ;                                                                       :
C  ;               AL = 0                                                  :
C  ;                       BL    - ATTRIBUTE                               :
C  ;                       STRING - (CHAR, CHAR, CHAR, ...)                :
C  ;                       CURSOR NOT MOVED                                :
C  ;               AL = 1                                                  :
C  ;                       BL    - ATTRIBUTE                               :
C  ;                       STRING - (CHAR, CHAR, CHAR, ...)                :
C  ;                       CURSOR  IS MOVED                                :
C  ;               AL = 2                                                  :
C  ;                       STRING - (CHAR, ATTR, CHAR, ATTR, ...)          :
C  ;                       CURSOR NOT MOVED                                :
C  ;               AL = 3                                                  :
C  ;                       STRING - (CHAR, ATTR, CHAR, ATTR, ...)          :
C  ;                       CURSOR  IS MOVED                                :
C  ;                                                                       :
C  ;               NOTE : CHAR RET, LINE FEED, BACKSPACE, AND BELL ARE     :
C  ;                       TREATED AS COMMANDS RATHER THAN PRINTABLE       :
C  ;                       CHARACTERS.                                     :
C  ;                                                                       :
C  ;-----------------------------------------------------------------------:
C
C  SRLOAD  MACRO   SEGREG,VALUE
C          IFNB <VALUE>
C          IFIDN <VALUE>,<0>
C          SUB     DX,DX
C          ELSE
C          MOV     DX,VALUE
C          ENDIF
C          ENDIF
C          IF WNR
C          MOV     AH,08FH
C          INT     15H
C          ENDIF
C          MOV     SEGREG,DX
C          ENDM
C
C  WIN     MACRO
C          IF WNR
C          MOV     AH,08EH
C          INT     15H
C          ENDIF
C          IN      AL,DX
C          ENDM
C
C  WOUT    MACRO
C          IF WNR
C          PUSH    AX
C          MOV     AH,08DH
C          INT     15H
C          ENDIF
C          OUT     DX,AL
C          IF WNR
C          POP     AX
C          ENDIF
C          ENDM
C
C  WLXS    MACRO   SEGREG,TARGREG,VALUE
C          IF      WNR
C          PUSH    DX
C          MOV     TARGREG,VALUE
C          SRLOAD  SEGREG,VALUE+2
C          POP     DX
C          ELSE
C          L&SEGREG        TARGREG,VALUE
```

```
                                        C          ENDIF
                                        C          ENDM
                                        C
                                        C ;----- LOW MEMORY SEGMENT
                                        C
0000                                    C ABSO     SEGMENT AT      0
0014                                    C          ORG     005H*4                  ; PRINT SCREEN VECTOR
0014                                    C INT5_PTR         LABEL   DWORD
0040                                    C          ORG     010H*4                  ; VIDEO I/O VECTOR
0040                                    C VIDEO            LABEL   DWORD
007C                                    C          ORG     01FH*4                  ; GRAPHIC CHARS 128-255
007C                                    C EXT_PTR          LABEL   DWORD
                                        C
0108                                    C          ORG     042H*4                  ; REVECTORED 10H*4
0108                                    C PLANAR_VIDEO     LABEL   DWORD
                                        C
010C                                    C          ORG     043H*4                  ;.GRAPHIC CHARS 0-255
010C                                    C GRX_SET          LABEL   DWORD
                                        C
0410                                    C          ORG     0410H
0410                                    C equip_low        label   byte
0410 ????                               C EQUIP_FLAG       DW      ?
                                        C
                                        C ;----- REUSE RAM FROM PLANAR BIOS
                                        C
0449                                    C          ORG     449H
0449 ??                                 C CRT_MODE         DB      ?
044A ????                               C CRT_COLS         DW      ?
044C ????                               C CRT_LEN          DW      ?
044E ????                               C CRT_START        DW      ?
0450    08 [                            C CURSOR_POSN      DW      8 DUP(?)
            ????
                 ]                      C
                                        C
                                        C
0460 ????                               C CURSOR_MODE      DW      ?
0462 ??                                 C ACTIVE_PAGE      DB      ?
0463 ????                               C ADDR_6845        DW      ?
0465 ??                                 C CRT_MODE_SET     DB      ?
0466 ??                                 C CRT_PALETTE      DB      ?
                                        C
0472                                    C          ORG     0472H
0472 ????                               C RESET_FLAG       DW      ?
0484                                    C          ORG     0484H
0484 ??                                 C ROWS     DB      ?                       ; ROWS ON THE SCREEN
0485 ????                               C POINTS   DW      ?                       ; BYTES PER CHARACTER
                                        C
0487 ??                                 C INFO     DB      ?
                                        C
                                        C ; INFO
                                        C ;        D7 - HIGH BIT OF MODE SET, CLEAR/NOT CLEAR REGEN
                                        C ;        D6 - MEMORY  D6 D5 = 0 0 - 064K    0 1 - 128K
                                        C ;        D5 - MEMORY          1 0 - 192K    1 1 - 256K
                                        C ;        D4 - RESERVED
                                        C ;        D3 - ega active monitor (0), ega not active (1)
                                        C ;        D2 - wait for display enable (1)
                                        C ;        D1 - EGA HAS A MONOCHROME ATTACHED (1)
                                        C ;        D0 - set c_type emulate active (0)
                                        C
0488 ??                                 C INFO_3   DB      ?
                                        C
                                        C ; INFO_3
                                        C ;        D7-D4   FEATURE BITS
                                        C ;        D3-D0   SWITCHES
                                        C
                                        C          org     04a8h
04A8                                    C save_ptr         label   dword
04A8                                    C
                                        C ;----- save_ptr
                                        C ;
                                        C ;        save_ptr is a pointer to a table as described as follows :
                                        C ;
                                        C ;        dword_1         video parameter table pointer
                                        C ;        dword_2         dynamic save area pointer
                                        C ;        dword_3         alpha mode auxiliary char gen pointer
                                        C ;        dword_4         graphics mode auxiliary char gen pointer
                                        C ;        dword_5         reserved
                                        C ;        dword_6         reserved
                                        C ;        dword_7         reserved
                                        C ;
                                        C ;        dword_1         Parameter Table Pointer
                                        C ;                        Initialized to BIOS EGA parameter table.
                                        C ;                        This value MUST exist.
                                        C ;
                                        C ;        dword_2         Parameter Save area pointer
                                        C ;                        Initialized to 0000:0000, this value is optional.
                                        C ;                        When non-zero, this pointer will be used as pointer
                                        C ;                        to a RAM area where certain dynamic values are to
                                        C ;                        be saved. When in EGA operation this RAM area will
                                        C ;                        hold the 16 EGA palette register values plus
                                        C ;                        the overscan value in bytes 0-16d respectively.
                                        C ;                        At least 256 bytes must be allocated for this area.
                                        C ;
                                        C ;        dword_3         Alpha Mode Auxiliary pointer
                                        C ;                        Initialized to 0000:0000, this value is optional.
                                        C ;                        When non-zero, this pointer is used as a pointer
                                        C ;                        to a tables described as follows :
                                        C ;
                                        C ;                        byte    bytes/character
                                        C ;                        byte    block to load, should be zero for normal
                                        C ;                                operation
                                        C ;                        word    count to store, should be 256d for normal
                                        C ;                                operation
                                        C ;                        word    character offset, should be zero for normal
                                        C ;                                operation
                                        C ;                        dword   pointer to a font table
                                        C ;                        byte    displayable rows
                                        C ;                                if 'FF' the maximum calculated value will be
                                        C ;                                used, else this value will be used
                                        C ;                        byte    consecutive bytes of mode values for which
                                        C ;                                this font description is to be used.
                                        C ;                                The end of this stream is indicated by a
                                        C ;                                byte code of 'FF'
                                        C ;
                                        C ;        dword_4         Graphics Mode Auxiliary pointer
                                        C ;                        Initialized to 0000:0000, this value is optional.
                                        C ;                        When non-zero, this pointer is used as a pointer
                                        C ;                        to a tables described as follows :
                                        C ;
                                        C ;                        byte    displayable rows
                                        C ;                        word    bytes per character
                                        C ;                        dword   pointer to a font table
                                        C ;                        byte    consecutive bytes of mode values for which
                                        C ;                                this font description is to be used.
                                        C ;                                The end of this stream is indicated by a
                                        C ;                                byte code of 'FF'
                                        C ;
```

```
C  ;       dword_5 thru dword_7
C  ;                       Reserved and set to 0000:0000.
C  ;
C
C         ORG     0500H
C  STATUS_BYTE    DB      ?
C  ABSO    ENDS
C
C  PORT_B          EQU     61H                     ; 8255 PORT B ADDR
C  TIMER           EQU     40H
C
C  ;----- EQUATES FOR CARD PORT ADDRESSES
C
C         SEQ_ADDR        EQU     0C4H
C         SEQ_DATA        EQU     0C5H
C         CRTC_ADDR       EQU     0D4H
C         CRTC_ADDR_B     EQU     0B4H
C         CRTC_DATA       EQU     0D5H            ; OR 0B5H
C         GRAPH_1_POS     EQU     0CCH
C         GRAPH_2_POS     EQU     0CAH
C         GRAPH_ADDR      EQU     0CEH
C         GRAPH_DATA      EQU     0CFH
C         MISC_OUTPUT     EQU     0C2H
C         IN_STAT_0       EQU     0C2H
C         INPUT_STATUS_B  EQU     0BAH
C         INPUT_STATUS    EQU     0DAH
C         ATTR_READ       EQU     0DAH
C         ATTR_WRITE      EQU     0C0H
C
C  ;----- EQUATES FOR ADDRESS REGISTER VALUES
C
C         S_RESET         EQU     00H
C         S_CLOCK         EQU     01H
C         S_MAP           EQU     02H
C         S_CGEN          EQU     03H
C         S_MEM           EQU     04H
C
C         C_HRZ_TOT       EQU     00H
C         C_HRZ_DSP       EQU     01H
C         C_STRT_HRZ_BLK  EQU     02H
C         C_END_HRZ_BLK   EQU     03H
C         C_STRT_HRZ_SYN  EQU     04H
C         C_END_HRZ_SYN   EQU     05H
C         C_VRT_TOT       EQU     06H
C         C_OVERFLOW      EQU     07H
C         C_PRE_ROW       EQU     08H
C         C_MAX_SCAN_LN   EQU     09H
C         C_CRSR_START    EQU     0AH
C         C_CRSR_END      EQU     0BH
C         C_STRT_HGH      EQU     0CH
C         C_STRT_LOW      EQU     0DH
C         C_CRSR_LOC_HGH  EQU     0EH
C         C_CRSR_LOC_LOW  EQU     0FH
C         C_VRT_SYN_STRT  EQU     10H             ; WRITE ONLY
C         C_LGHT_PEN_HGH  EQU     10H             ; READ ONLY
C         C_VRT_SYN_END   EQU     11H             ; WRITE ONLY
C         C_LGHT_PEN_LOW  EQU     11H             ; READ ONLY
C         C_VRT_DSP_END   EQU     12H
C         C_OFFSET        EQU     13H
C         C_UNDERLN_LOC   EQU     14H
C         C_STRT_VRT_BLK  EQU     15H
C         C_END_VRT_BLK   EQU     16H
C         C_MODE_CNTL     EQU     17H
C         C_LN_COMP       EQU     18H
C
C         G_SET_RESET     EQU     00H
C         G_ENBL_SET      EQU     01H
C         G_CLR_COMP      EQU     02H
C         G_DATA_ROT      EQU     03H
C         G_READ_MAP      EQU     04H
C         G_MODE          EQU     05H
C         G_MISC          EQU     06H
C         G_COLOR         EQU     07H
C         G_BIT_MASK      EQU     08H
C
C         P_MODE          EQU     10H
C         P_OVERSC        EQU     11H
C         P_CPLANE        EQU     12H
C         P_HPEL          EQU     13H
C
C         SUBTTL

      ;----- CODE SEGMENT

      CODE    SEGMENT PUBLIC

C         INCLUDE         VPOST.INC
C         SUBTTL  VPOST.INC
C         PAGE
C
C  ;----- POST
C
C         ASSUME  CS:CODE,DS:ABSO
C         ORG     0H
C         DB      055H                            ; SIGNATURE
C         DB      0AAH                            ;  BYTES
C         DB      020H                            ; LENGTH INDICATOR
C
C  ;----- NOTE : do not use the signature bytes as a presence test
C
C  ;       PLANAR VIDEO SWITCH SETTINGS
C
C  ;       0 0 - UNUSED
C  ;       0 1 - 40 X 25 COLOR
C  ;       1 0 - 80 X 25 COLOR
C  ;       1 1 - 80 X 25 MONOCHROME
C  ; NOTE : 0 0 MUST BE SET WHEN THIS ADAPTER IS INSTALLED.
C
C  ;       VIDEO ADAPTER SWITCH SETTINGS
C
C  ;       0 0 0 0 - MONOC PRIMARY, EGA COLOR, 40X25
C  ;       0 0 0 1 - MONOC PRIMARY, EGA COLOR, 80X25
C  ;       0 0 1 0 - monoc primary, ega hi res emulate (same as 0001)
C  ;       0 0 1 1 - MONOC PRIMARY, ega hi res enhanced
C  ;       0 1 0 0 - COLOR 40 PRIMARY, EGA MONOCHROME
C  ;       0 1 0 1 - COLOR 80 PRIMARY, EGA MONOCHROME
C
C  ;       0 1 1 0 - MONOC SECONDARY, EGA COLOR, 40X25
C  ;       0 1 1 1 - MONOC SECONDARY, EGA COLOR, 80X25
C  ;       1 0 0 0 - monoc secondary, ega hi res emulate (same as 0111)
C  ;       1 0 0 1 - MONOC SECONDARY, EGA hi res enhanced
C  ;       1 0 1 0 - COLOR 40 SECONDARY, EGA MONOCHROME
C  ;       1 0 1 1 - COLOR 80 SECONDARY, EGA MONOCHROME
C
C  ;       1 1 0 0 - RESERVED
C  ;       1 1 0 1 - RESERVED
```

```
                                                631      C  ;       1 1 1 0 - RESERVED
                                                632      C  ;       1 1 1 1 - RESERVED
                                                633      C
                                                634      C  ;----- SETUP ROUTINE FOR THIS MODULE
                                                635      C
0003                                            636      C  VIDEO_SETUP      PROC    FAR
0003  EB 29                                     637      C                   JMP     SHORT   L1
0005  32 34 30 30                               638      C                   DB      '2400'
0009  36 33 32 30 30 33 33                      639      C                   DB      '6320030 (C)COPYRIGHT IBM 1983'
      30 20 28 43 29 43
      4F 50 59 52 49 47
      48 54 20 49 42 4D
      20 31 39 38 33
0026  31 31 2F 30 33 2F                         640      C                   DB      '11/03/83'
      38 33                                     641      C
                                                642      C
                                                643      C  ;----- SET UP VIDEO VECTORS
                                                644      C
002E                                            645      C  L1:
002E  B6 03                                     646      C                   mov     dh,3
0030  B2 DA                                     647      C                   mov     dl,input_status
0032  EC                                        648      C                   in      al,dx
0033  B2 BA                                     649      C                   mov     dl,input_status_b
0035  EC                                        650      C                   in      al,dx
0036  B2 C0                                     651      C                   mov     dl,attr_write
0038  B0 00                                     652      C                   mov     al,0
003A  EE                                        653      C                   out     dx,al
                                                654      C
                                                655      C                   SRLOAD  DS,0
003B  2B D2                                     656      C+                  SUB     DX,DX
003D  8E DA                                     657      C+                  MOV     DS,DX
003F  FA                                        658      C                   CLI
0040  C7 06 0040 R 0CDA R                       659      C                   MOV     WORD PTR VIDEO,OFFSET COMBO_VIDEO
0046  8C 0E 0042 R                             660      C                   MOV     WORD PTR VIDEO+2, CS
004A  C7 06 0108 R F065                         661      C                   MOV     WORD PTR PLANAR_VIDEO,0F065H
0050  C7 06 010A R F000                         662      C                   MOV     WORD PTR PLANAR_VIDEO+2,0F000H
0056  C7 06 04A8 R 010F R                       663      C                   MOV     WORD PTR save_PTR,OFFSET save_tbl
005C  8C 0E 04AA R                             664      C                   MOV     WORD PTR save_PTR+2, CS
0060  C7 06 007C R 0000 E                      665      C                   MOV     WORD PTR EXT_PTR, OFFSET INT_1F_1
0066  8C 0E 007E R                             666      C                   MOV     WORD PTR EXT_PTR+2, CS
006A  C7 06 010C R 0000 E                      667      C                   MOV     WORD PTR GRX_SET, OFFSET CGDDOT
0070  8C 0E 010E R                             668      C                   MOV     WORD PTR GRX_SET+2, CS
0074  FB                                       669      C                   STI
                                                670      C
                                                671      C  ;----- POST FOR COMBO VIDEO CARD
                                                672      C
0075  C6 06 0487 R 04                          673      C                   mov     info,00000100B
007A  E8 009C R                                 674      C                   CALL    RD_SWS
007D  88 1E 0488 R                              675      C                   MOV     INFO_3,BL
0081  E8 00D1 R                                676      C                   CALL    F_BTS
0084  08 06 0488 R                              677      C                   OR      INFO_3,AL
0088  8A 1E 0488 R                              678      C                   MOV     BL,INFO_3
008C  E8 00F6 R                                 679      C                   CALL    MK_ENV
008F  E9 0247 R                                 680      C                   JMP     POST
0092                                            681      C  SKIP:
0092  CB                                        682      C                   RET
0093                                            683      C  VIDEO_SETUP      ENDP
                                                684      C
                                                685      C
0093                                            686      C  POR_1    PROC    NEAR
                                                687      C                  WOUT
0093  EE                                        688      C+                 OUT     DX,AL
0094  50                                        689      C                  PUSH    AX
0095  58                                        690      C                  POP     AX
                                                691      C                  WIN
0096  EC                                        692      C+                 IN      AL,DX
0097  24 10                                     693      C                  AND     AL,010H
0099  D0 E8                                     694      C                  SHR     AL,1
009B  C3                                        695      C                  RET
009C                                            696      C  POR_1    ENDP
                                                697      C
                                                698      C  ;----- READ THE SWITCH SETTINGS ON THE CARD
                                                699      C
009C                                            700      C  RD_SWS   PROC    NEAR
                                                701      C                   ASSUME  DS:ABS0
009C  B6 03                                     702      C                   mov     dh,3
009E  B2 C2                                     703      C                   MOV     DL,IN_STAT_0
00A0  B2 C2                                     704      C                   mov     dl,misc_output
00A2  B0 01                                     705      C                   mov     al,1
                                                706      C                   wout
00A4  EE                                        707      C+                  OUT     DX,AL
                                                708      C
                                                709      C  ;----- COULD BE 0,4,8,C
                                                710      C
00A5  B0 0D                                     711      C                   MOV     AL,0DH
00A7  E8 0093 R                                 712      C                   CALL    POR_1
00AA  D0 E8                                     713      C                   SHR     AL,1
00AC  D0 E8                                     714      C                   SHR     AL,1
00AE  D0 E8                                     715      C                   SHR     AL,1
00B0  8A D8                                     716      C                   MOV     BL,AL
                                                717      C
00B2  B0 09                                     718      C                   MOV     AL,9
00B4  E8 0093 R                                 719      C                   CALL    POR_1
00B7  D0 E8                                     720      C                   SHR     AL,1
00B9  D0 E8                                     721      C                   SHR     AL,1
00BB  0A D8                                     722      C                   OR      BL,AL
                                                723      C
00BD  B0 05                                     724      C                   MOV     AL,5
00BF  E8 0093 R                                 725      C                   CALL    POR_1
00C2  D0 E8                                     726      C                   SHR     AL,1
00C4  0A D8                                     727      C                   OR      BL,AL
                                                728      C
00C6  B0 01                                     729      C                   MOV     AL,1
00C8  E8 0093 R                                 730      C                   CALL    POR_1
00CB  0A D8                                     731      C                   OR      BL,AL
                                                732      C
00CD  80 E3 0F                                  733      C                   AND     BL,0FH
00D0  C3                                        734      C                   RET
00D1                                            735      C  RD_SWS   ENDP
                                                736      C
                                                737      C  ;----- OBTAIN THE FEATURE BITS FROM DAUGHTER CARD
                                                738      C
00D1                                            739      C  F_BTS    PROC    NEAR
00D1  B6 03                                     740      C                   mov     dh,3
00D3  B2 BA                                     741      C                   MOV     DL,0BAH
00D5  B0 01                                     742      C                   MOV     AL,1
                                                743      C                   WOUT
00D7  EE                                        744      C+                  OUT     DX,AL
00D8  B2 DA                                     745      C                   MOV     DL,0DAH
                                                746      C                   WOUT
00DA  EE                                        747      C+                  OUT     DX,AL
00DB  B2 C2                                     748      C                   MOV     DL,IN_STAT_0
                                                749      C                   WIN                             ; READ FEATURE BITS
00DD  EC                                        750      C+                  IN      AL,DX
00DE  24 60                                     751      C                   AND     AL,060H
00E0  D0 E8                                     752      C                   SHR     AL,1
```

```
00E2  8A D8                   757     C               MOV     BL,AL
00E4  B2 BA                   758     C               MOV     DL,0BAH
00E6  B0 02                   759     C               MOV     AL,2
                              760     C               WOUT
00E8  EE                      761     C+              OUT     DX,AL
00E9  B2 DA                   762     C               MOV     DL,0DAH
                              763     C               WOUT
00EB  EE                      764     C+              OUT     DX,AL
00EC  B2 C2                   765     C               MOV     DL,IN_STAT_0
                              766     C               WIN                                     ; READ FEATURE BITS
00EE  EC                      767     C+              IN      AL,DX
00EF  24 60                   768     C               AND     AL,060H
00F1  D0 E0                   769     C               SHL     AL,1
00F3  0A C3                   770     C               OR      AL,BL
00F5  C3                      771     C               RET
00F6                          772     C  F_BTS        ENDP
                              773     C
                              774     C  ;----- ESTABLISH THE VIDEO ENVIRONMENT, KEYED OFF OF THE SWITCHES
                              775     C
00F6                          776     C  MK_ENV       PROC    NEAR
                              777     C               ASSUME  DS:ABS0
00F6  2A FF                   778     C               SUB     BH,BH
00F8  80 E3 0F                779     C               AND     BL,0FH
00FB  D1 E3                   780     C               SAL     BX,1
00FD  52                      781     C               PUSH    DX
00FE  B6 03                   782     C               mov     dh,3
0100  8A E6                   783     C               MOV     AH,DH
0102  5A                      784     C               POP     DX
0103  80 E4 01                785     C               AND     AH,1
0106  FE C4                   786     C               INC     AH
0108  F6 D4                   787     C               NOT     AH
010A  2E: FF A7 012B R        788     C               JMP     WORD PTR CS:[BX + OFFSET T5]
                              789     C
010F                          790     C  save_tbl             label   dword
010F  071A R                  791     C                       dw      offset  video_parms             ; parms
0111  C000                    792     C                       dw      0c000h                          ; parms
0113  0000                    793     C                       dw      0                               ; pal save area
0115  0000                    794     C                       dw      0                               ; pal save area
0117  0000                    795     C                       dw      0                               ; alpha tables
0119  0000                    796     C                       dw      0                               ; alpha tables
011B  0000                    797     C                       dw      0                               ; graphics tables
011D  0000                    798     C                       dw      0                               ; graphics tables
                              799     C
011F  0000                    800     C                       dw      0
0121  0000                    801     C                       dw      0
0123  0000                    802     C                       dw      0
0125  0000                    803     C                       dw      0
0127  0000                    804     C                       dw      0
0129  0000                    805     C                       dw      0
                              806     C
012B                          807     C  T5           LABEL   WORD
012B  0176 R                  808     C                       DW      OFFSET PST_0
012D  0181 R                  809     C                       DW      OFFSET PST_1
012F  0181 R                  810     C                       DW      OFFSET PST_2
0131  018C R                  811     C                       DW      OFFSET PST_3
0133  0197 R                  812     C                       DW      OFFSET PST_4
0135  01AB R                  813     C                       DW      OFFSET PST_5
0137  01BF R                  814     C                       DW      OFFSET PST_6
0139  01CA R                  815     C                       DW      OFFSET PST_7
                              816     C
013B  01CA R                  817     C                       DW      OFFSET PST_8
013D  01D5 R                  818     C                       DW      OFFSET PST_9
013F  01E0 R                  819     C                       DW      OFFSET PST_A
0141  01F4 R                  820     C                       DW      OFFSET PST_B
0143  0207 R                  821     C                       DW      OFFSET PST_OUT
0145  0207 R                  822     C                       DW      OFFSET PST_OUT
0147  0207 R                  823     C                       DW      OFFSET PST_OUT
0149  0207 R                  824     C                       DW      OFFSET PST_OUT
                              825     C
014B                          826     C  ENV_X        PROC    NEAR                                    ; SET 40X25 COLOR ALPHA
014B  80 26 0410 R CF         827     C                       and     equip_low,0cfh
0150  80 0E 0410 R 10         828     C                       or      equip_low,010h
0155  B8 0001                829     C                       MOV     AX,1H
0158  CD 10                   830     C                       INT     10H
015A  C3                      831     C                       RET
015B                          832     C  ENV_X        ENDP
                              833     C
015B                          834     C  ENV_0        PROC    NEAR                                    ; SET 80X25 COLOR ALPHA
015B  80 26 0410 R CF         835     C                       and     equip_low,0cfh
0160  80 0E 0410 R 20         836     C                       or      equip_low,020h
0165  B8 0003                837     C                       MOV     AX,03H
0168  CD 10                   838     C                       INT     10H
016A  C3                      839     C                       RET
016B                          840     C  ENV_0        ENDP
                              841     C
016B                          842     C  ENV_3        PROC    NEAR                                   '; SET MONOCHROME ALPHA
016B  80 0E 0410 R 30         843     C                       or      equip_low,030h
0170  B8 0007                844     C                       MOV     AX,07H
0173  CD 10                   845     C                       INT     10H
0175  C3                      846     C                       RET
0176                          847     C  ENV_3        ENDP
                              848     C
                              849     C
0176                          850     C  PST_0:
0176  20 26 0487 R            851     C                       AND     INFO,AH
017A  E8 014B R               852     C                       CALL    ENV_X
017D  E8 016B R               853     C                       CALL    ENV_3
0180  C3                      854     C                       RET
0181                          855     C  PST_1:
0181                          856     C  PST_2:
0181  20 26 0487 R            857     C                       AND     INFO,AH
0185  E8 015B R               858     C                       CALL    ENV_0
0188  E8 016B R               859     C                       CALL    ENV_3
018B  C3                      860     C                       RET
018C                          861     C  PST_3:
018C  20 26 0487 R            862     C                       AND     INFO,AH
0190  E8 015B R               863     C                       CALL    ENV_0
0193  E8 0168 R               864     C                       CALL    ENV_3
0196  C3                      865     C                       RET
0197                          866     C  PST_4:
0197  B6 03                   867     C                       mov     dh,3
0199  B2 C2                   868     C                       mov     dl,misc_output
019B  B0 00                   869     C                       mov     al,0
                              870     C                       wout
019D  EE                      871     C+                      OUT     DX,AL
019E  F6 D4                   872     C                       NOT     AH
01A0  08 26 0487 R            873     C                       OR      INFO,AH
01A4  E8 016B R               874     C                       CALL    ENV_3
01A7  E8 014B R               875     C                       CALL    ENV_X
01AA  C3                      876     C                       RET
01AB                          877     C  PST_5:
01AB  B6 03                   878     C                       mov     dh,3
01AD  B2 C2                   879     C                       mov     dl,misc_output
01AF  B0 00                   880     C                       mov     al,0
                              881     C                       wout
01B1  EE                      882     C+                      OUT     DX,AL
```

```
01B2  F6 D4                    883     C          NOT     AH
01B4  08 26 0487 R             884     C          OR      INFO,AH
01B8  E8 016B R                885     C          CALL    ENV_3
01BB  E8 015B R                886     C          CALL    ENV_0
01BE  C3                       887     C          RET
01BF                           888     C  PST_6:
01BF  20 26 0487 R             889     C          AND     INFO,AH
01C3  E8 016B R                890     C          CALL    ENV_3
01C6  E8 014B R                891     C          CALL    ENV_X
01C9  C3                       892     C          RET
01CA                           893     C  PST_7:
01CA                           894     C  PST_8:
01CA  20 26 0487 R             895     C          AND     INFO,AH
01CE  E8 016B R                896     C          CALL    ENV_3
01D1  E8 015B R                897     C          CALL    ENV_0
01D4  C3                       898     C          RET
01D5                           899     C  PST_9:
01D5  20 26 0487 R             900     C          AND     INFO,AH
01D9  E8 016B R                901     C          CALL    ENV_3
01DC  E8 015B R                902     C          CALL    ENV_0
01DF  C3                       903     C          RET
01E0                           904     C  PST_A:
01E0  B6 03                    905     C          mov     dh,3
01E2  B2 C2                    906     C          mov     dl,misc_output
01E4  B0 00                    907     C          mov     al,0
                               908     C          wout
01E6  EE                       909     C+         OUT     DX,AL
01E7  F6 D4                    910     C          NOT     AH
01E9  08 26 0487 R             911     C          OR      INFO,AH
01ED  E8 014B R                912     C          CALL    ENV_X
01F0  E8 016B R                913     C          CALL    ENV_3
01F3  C3                       914     C          RET
01F4                           915     C  PST_B:
01F4  B6 03                    916     C          mov     dh,3
01F6  B2 C2                    917     C          mov     dl,misc_output
01F8  B0 00                    918     C          mov     al,0
                               919     C          wout
01FA  EE                       920     C+         OUT     DX,AL
01FB  F6 D4                    921     C          NOT     AH
01FD  08 26 0487 R             922     C          OR      INFO,AH
0201  E8 015B R                923     C          CALL    ENV_0
0204  E8 016B R                924     C          CALL    ENV_3
0207                           925     C  PST_OUT:
0207  C3                       926     C          RET
0208                           927     C  MK_ENV  ENDP
                               928     C
                               929     C  ;--------------------------------------------------------------------------
                               930     C  ; THIS ROUTINE TESTS THE CRT CARD INTERNAL DATA BUS AND IN A LIMITED    :
                               931     C  ; WAY TESTS THE CRTC VIDEO CHIP BY WRITING/READING FROM CURSOR REGISTER :
                               932     C  ; CARRY IS SET IF AN ERROR IS FOUND                                     :
                               933     C  ;                                                                       :
                               934     C  ; REGISTERS BX,SI,ES,DS ARE PRESERVED.                                  :
                               935     C  ; REGISTERS AX,CX,DX ARE MODIFYED.                                      :
                               936     C  ;--------------------------------------------------------------------------
0208                           937     C  CD_PRESENCE_TST PROC    NEAR
0208  53                       938     C          PUSH    BX                      ; SAVE BX
0209  BB 007F                  939     C          MOV     BX,07FH                 ; INITIAL WORD PATTERN BYTE
020C  8B FB                    940     C          mov     di,bx
020E  50                       941     C          PUSH    AX                      ; SAVE PORT ADDRESS
020F  E8 022F R                942     C          CALL    RD_CURSOR               ;
0212  8B F0                    943     C          MOV     SI,AX                   ; SAVE ORIGINAL VALUE
0214  58                       944     C          POP     AX                      ; RECOVER PORT ADDRESS
0215  50                       945     C          PUSH    AX                      ; SAVE PORT ADDRESS
0216  E8 0239 R                946     C          CALL    WR_CURSOR               ; WRITE CURSOR
0219  58                       947     C          POP     AX                      ; RECOVER PORT ADDRESS
021A  50                       948     C          PUSH    AX                      ; SAVE PORT ADDRESS
021B  E8 022F R                949     C          CALL    RD_CURSOR               ; READ IT BACK
021E  3B C7                    950     C          CMP     AX,DI                   ; SAME?
0220  58                       951     C          POP     AX
0221  75 03                    952     C          JNZ     NOT_PRESENT             ; EXIT IF NOT EQUAL
0223  E8 05 90                953     C          JMP     TST_EX
0226                           954     C  NOT_PRESENT:
0226  33 C0                    955     C          XOR     AX,AX                   ; SET NOT PRESENT
0228  5B                       956     C          POP     BX
0229  C3                       957     C          RET
022A                           958     C  TST_EX:
022A  B8 0001                  959     C          MOV     AX,1                    ; SET PRESENT ON EXIT
022D  5B                       960     C          POP     BX                      ; RESTORE BX
022E  C3                       961     C          RET
022F                           962     C  CD_PRESENCE_TST ENDP
                               963     C
                               964     C  ;--------------------------------------------------------------------------
                               965     C  ; MODULE NAME  RD_CURSOR                                                :
                               966     C  ;  READ CURSOR POSITION [ADDRESS] (FROM CRTC) TO  AX                    :
                               967     C  ;                                                                       :
                               968     C  ; REGISTER AX IS MODIFIED.                                              :
                               969     C  ;--------------------------------------------------------------------------
022F                           970     C  RD_CURSOR       PROC NEAR
022F  52                       971     C          PUSH    DX                              ; SAVE REGS USED
0230  8B D0                    972     C          MOV     DX,AX
0232  B0 0E                    973     C          MOV     AL,C_CRSR_LOC_HGH
0234  EE                       974     C          OUT     DX,AL
0235  42                       975     C          INC     DX
0236  EC                       976     C          IN      AL,DX
                               977     C                                          ; RETURN WITH CURSOR POS IN AX
                               978     C                                          ; RESTORE REGS USED
0237  5A                       979     C          POP     DX
0238  C3                       980     C          RET
0239                           981     C  RD_CURSOR       ENDP
                               982     C
                               983     C  ;--------------------------------------------------------------------------
                               984     C  ; MODULE NAME  WR_CURSOR                                                :
                               985     C  ;  WRITE CURSOR POSITION [ADDRESS] (TO CRTC) WITH CONTENTS OF AX        :
                               986     C  ;                                                                       :
                               987     C  ; ALL REGISTERS PRESERVED                                               :
                               988     C  ;--------------------------------------------------------------------------
0239                           989     C  WR_CURSOR       PROC NEAR
                               990     C                                          ; SAVE REGS USED
0239  50                       991     C          PUSH    AX
023A  52                       992     C          PUSH    DX
023B  8B D0                    993     C          MOV     DX,AX
023D  B4 0E                    994     C          MOV     AH,C_CRSR_LOC_HGH       ; CURSOR LOCATION HIGH INDEX
023F  B0 7F                    995     C          MOV     AL,07FH                 ; TEST VALUE
0241  E8 0D18 R                996     C          CALL    OUT_DX
                               997     C                                          ; RETURN WITH CURSOR POS IN AX
                               998     C                                          ; RESTORE REGS USED
0244  5A                       999     C          POP     DX
0245  58                      1000     C          POP     AX
0246  C3                      1001     C          RET
0247                          1002     C  WR_CURSOR       ENDP
                              1003     C
0247                          1004     C  POST:
                              1005     C  ;--------------------------------------------------
                              1006     C  ;       INITIALIZE AND START CRT CONTROLLER (6845)      :
                              1007     C  ;       ON COLOR GRAPHICS AND MONOCHROME CARDS          :
                              1008     C  ;       TEST VIDEO READ/WRITE STORAGE.                  :
```

```
                        1009     C  ; DESCRIPTION                                                 :
                        1010     C  ;       RESET THE VIDEO ENABLE SIGNAL.                       :
                        1011     C  ;       SELECT ALPHANUMERIC MODE, 40 * 25, B & W.             :
                        1012     C  ;       READ/WRITE DATA PATTERNS TO STG. CHECK STG            :
                        1013     C  ;       ADDRESSABILITY.                                      :
                        1014     C  ;-----------------------------------------------------------
                        1015     C          ASSUME  DS:ABS0,ES:ABS0
0247  E8 0001 R         1016     C          CALL    DDS
024A  F6 06 0487 R 02   1017     C          test    info,2
024F  75 12             1018     C          JNZ     COLOR_PRESENCE_TST
0251  B8 03B4           1019     C          MOV     AX,03B4H
0254  E8 0208 R         1020     C          CALL    CD_PRESENCE_TST
0257  3D 0001           1021     C          CMP     AX,1
025A  74 03             1022     C          JE      CONT1
025C  E9 031A R         1023     C          JMP     POD14
025F                    1024     C  CONT1:
025F  B4 30             1025     C          MOV     AH,30H                  ; MONOCHROME CARD INSTALLED
0261  EB 10             1026     C          jmp     short   over
0263                    1027     C  COLOR_PRESENCE_TST:
0263  B8 03D4           1028     C          MOV     AX,03D4H
0266  E8 0208 R         1029     C          CALL    CD_PRESENCE_TST
0269  3D 0001           1030     C          CMP     AX,1
026C  74 03             1031     C          JE      CONT2
026E  E9 031A R         1032     C          JMP     POD14
0271                    1033     C  CONT2:
0271  B4 20             1034     C          MOV     AH,20H                  ; COLOR GRAPHICS CARD INSTALLED
0273                    1035     C  OVER:
0273  50               1036     C          PUSH    AX                      ; RESAVE VALUE
0274  BB B000           1037     C          MOV     BX,0B000H               ; BEG VIDEO RAM ADDR B/W CD
0277  BA 03B8           1038     C          MOV     DX,3B8H                 ; MODE CONTROL B/W
027A  B9 1000           1039     C          MOV     CX,4096                 ; RAM BYTE CNT FOR B/W CD
027D  B0 01             1040     C          MOV     AL,1                    ; SET MODE FOR BW CARD
027F  80 FC 30          1041     C          CMP     AH,30H                  ; B/W VIDEO CARD ATTACHED?
0282  74 08             1042     C          JE      E9                      ; YES - GO TEST VIDEO STG
0284  B7 B8             1043     C          MOV     BH,0B8H                 ; BEG VIDEO RAM ADDR COLOR CD
0286  B2 D8             1044     C          MOV     DL,0D8H                 ; MODE CONTROL COLOR
0288  B5 40             1045     C          MOV     CH,40H                  ; RAM BYTE CNT FOR COLOR CD
028A  FE C8             1046     C          DEC     AL                      ; SET MODE TO 0 FOR COLOR CD
028C                    1047     C  E9:                                     ; TEST_VIDEO_STG:
028C  EE                1048     C          OUT     DX,AL                   ; DISABLE VIDEO FOR COLOR CD
                        1049     C
028D  8B 2E 0472 R      1050     C          MOV     BP,DS:RESET_FLAG        ; POD INITIALIZED BY KBD RESET
                        1051     C
0291  81 FD 1234        1052     C          CMP     BP,1234H                ; POD INITIATED BY KBD RESET?
0295  8E C3             1053     C          MOV     ES,BX                   ; POINT ES TO VIDEO RAM STG
0297  74 07             1054     C          JE      E10                     ; YES - SKIP VIDEO RAM TEST
0299  8E DB             1055     C          MOV     DS,BX                   ; POINT DS TO VIDEO RAM STG
                        1056     C          ASSUME  DS:NOTHING,ES:NOTHING
029B  E8 02E2 R         1057     C          CALL    STGTST_CNT              ; GO TEST VIDEO R/W STG
029E  75 2E             1058     C          JNE     E17                     ; R/W STG FAILURE - BEEP SPK
                        1059     C  ;-----------------------------------------------------------
                        1060     C  ;       SETUP VIDEO DATA ON SCREEN FOR VIDEO LINE TEST.      :
                        1061     C  ; DESCRIPTION                                               :
                        1062     C  ;       ENABLE VIDEO SIGNAL AND SET MODE.                    :
                        1063     C  ;       DISPLAY A HORIZONTAL BAR ON SCREEN.                  :
                        1064     C  ;-----------------------------------------------------------
02A0                    1065     C  E10:
02A0  58                1066     C          POP     AX                      ; GET VIDEO SENSE SWS (AH)
02A1  50                1067     C          PUSH    AX                      ; SAVE IT
02A2  B8 7020           1068     C          MOV     AX,7020H                ; WRT BLANKS IN REVERSE VIDEO
02A5  2B FF             1069     C          SUB     DI,DI                   ; SETUP STARTING LOC
02A7  B9 0028           1070     C          MOV     CX,40                   ; NO. OF BLANKS TO DISPLAY
02AA  F3/ AB            1071     C          REP     STOSW                   ; WRITE VIDEO STORAGE
                        1072     C  ;-----------------------------------------------------------
                        1073     C  ;       CRT INTERFACE LINES TEST                             :
                        1074     C  ; DESCRIPTION                                               :
                        1075     C  ;       SENSE ON/OFF TRANSITION OF THE VIDEO ENABLE          :
                        1076     C  ;       AND HORIZONTAL SYNC LINES.                           :
                        1077     C  ;-----------------------------------------------------------
02AC  58                1078     C          POP     AX                      ; GET VIDEO SENSE SW INFO
02AD  50                1079     C          PUSH    AX                      ; SAVE IT
02AE  80 FC 30          1080     C          CMP     AH,30H                  ; B/W CARD ATTACHED?
02B1  BA 03BA           1081     C          MOV     DX,03BAH                ; SETUP ADDR OF BW STATUS PORT
02B4  74 02             1082     C          JE      E11                     ; YES - GO TEST LINES
02B6  B2 DA             1083     C          MOV     DL,0DAH                 ; COLOR CARD IS ATTACHED
02B8                    1084     C  E11:                                    ; LINE_TST:
02B8  B4 08             1085     C          MOV     AH,8
02BA                    1086     C  E12:                                    ; OFLOOP_CNT:
02BA  2B C9             1087     C          SUB     CX,CX
02BC                    1088     C  E13:
02BC  EC                1089     C          IN      AL,DX                   ; READ CRT STATUS PORT
02BD  22 C4             1090     C          AND     AL,AH                   ; CHECK VIDEO/HORZ LINE
02BF  75 04             1091     C          JNZ     E14                     ; ITS ON - CHECK IF IT GOES OFF
02C1  E2 F9             1092     C          LOOP    E13                     ; LOOP TILL ON OR TIMEOUT
02C3  EB 09             1093     C          JMP     SHORT E17               ; GO PRINT ERROR MSG
02C5                    1094     C  E14:
02C5  2B C9             1095     C          SUB     CX,CX
02C7                    1096     C  E15:
02C7  EC                1097     C          IN      AL,DX                   ; READ CRT STATUS PORT
02C8  22 C4             1098     C          AND     AL,AH                   ; CHECK VIDEO/HORZ LINE
02CA  74 0A             1099     C          JZ      E16                     ; ITS ON - CHECK NEXT LINE
02CC  E2 F9             1100     C          LOOP    E15                     ; LOOP IF OFF TILL IT GOES ON
02CE                    1101     C  E17:                                    ;  CRT_ERR
02CE  BA 0102           1102     C          MOV     DX,102H
02D1  E8 06CB R         1103     C          CALL    ERR_BEEP                ; GO BEEP SPEAKER
02D4  EB 06             1104     C          JMP     SHORT E18
02D6                    1105     C  E16:                                    ; NXT_LINE
02D6  B1 03             1106     C          MOV     CL,3                    ; GET NEXT BIT TO CHECK
02D8  D2 EC             1107     C          SHR     AH,CL
02DA  75 DE             1108     C          JNZ     E12                     ; GO CHECK HORIZONTAL LINE
02DC                    1109     C  E18:                                    ;  DISPLAY_CURSOR:
02DC  58                1110     C          POP     AX                      ; GET VIDEO SENSE SWS (AH)
02DD  EB 3B             1111     C          jmp     short   pod14
                        1112     C
                        1113     C  ;-----------------------------------------------------------------------
                        1114     C  ;       THIS SUBROUTINE PERFORMS A READ/WRITE STORAGE TEST ON          :
                        1115     C  ;       A 16K BLOCK OF STORAGE.                                       :
                        1116     C  ; ENTRY REQUIREMENTS:                                                 :
                        1117     C  ;       ES = ADDRESS OF STORAGE SEGMENT BEING TESTED                   :
                        1118     C  ;       DS = ADDRESS OF STORAGE SEGMENT BEING TESTED                   :
                        1119     C  ;       WHEN ENTERING AT STGTST_CNT, CX MUST BE LOADED WITH            :
                        1120     C  ;       THE BYTE COUNT.                                               :
                        1121     C  ; EXIT PARAMETERS:                                                    :
                        1122     C  ;       ZERO FLAG = 0 IF STORAGE ERROR (DATA COMPARE OR PARITY CHECK.  :
                        1123     C  ;             AL = 0 DENOTES A PARITY CHECK, ELSE AL=XOR'ED BIT       :
                        1124     C  ;                    PATTERN OF THE EXPECTED DATA PATTERN VS THE      :
                        1125     C  ;                    ACTUAL DATA READ.                               :
                        1126     C  ;       AX,BX,CX,DX,DI, AND SI ARE ALL DESTROYED.                     :
                        1127     C  ;-----------------------------------------------------------------------
02DF                    1128     C  STGTST  PROC    NEAR
02DF  B9 4000           1129     C          MOV     CX,4000H                ; SETUP CNT TO TEST A 16K BLK
02E2                    1130     C  STGTST_CNT:
02E2  FC                1131     C          CLD                             ; SET DIR FLAG TO INCREMENT
02E3  8B D9             1132     C          MOV     BX,CX                   ; SAVE CNT (WK FOR VIDEO OR 16K)
02E5  B8 AAAA           1133     C          MOV     AX,0AAAAH               ; GET DATA PATTERN TO WRITE
02E8  BA FF55           1134     C          MOV     DX,0FF55H               ; SETUP OTHER DATA PATTERNS TO USE
```

```
02EB  2B FF                1135  C                SUB     DI,DI                   ; DI = OFFSET 0 RELATIVE TO ES REG
02ED  F3/ AA               1136  C                REP     STOSB                   ; WRITE STORAGE LOCATIONS
02EF                       1137  C  C3:                                            ;  STG01
02EF  4F                   1138  C                DEC     DI                      ; POINT TO LAST BYTE JUST WRITTEN
02F0  FD                   1139  C                STD                             ; SET DIR FLAG TO GO BACKWARDS
02F1                       1140  C  C4:
02F1  8B F7                1141  C                MOV     SI,DI
02F3  8B CB                1142  C                MOV     CX,BX                   ; SETUP BYTE CNT
02F5                       1143  C  C5:                                            ; INNER TEST LOOP
02F5  AC                   1144  C                LODSB                           ; READ OLD TEST BYTE [SI]+
02F6  32 C4                1145  C                XOR     AL,AH                   ; DATA READ AS EXPECTED ?
02F8  75 1E                1146  C                JNE     C7                      ;  NO - GO TO ERROR ROUTINE
02FA  8A C2                1147  C                MOV     AL,DL                   ; GET NEXT DATA PATTERN TO WRITE
02FC  AA                   1148  C                STOSB                           ; WRITE INTO LOCATION JUST READ
02FD  E2 F6                1149  C                LOOP    C5                      ;  DECREMENT COUNT AND LOOP CX
                           1150  C
02FF  22 E4                1151  C                AND     AH,AH                   ; ENDING 0 PATTERN WRITTEN TO STG?
0301  74 13                1152  C                JZ      C6X                     ; YES - RETURN TO CALLER WITH AL=0
0303  8A E0                1153  C                MOV     AH,AL                   ; SETUP NEW VALUE FOR COMPARE
0305  86 F2                1154  C                XCHG    DH,DL                   ; MOVE NEXT DATA PATTERN TO DL
0307  22 E4                1155  C                AND     AH,AH                   ; READING ZERO PATTERN THIS PASS ?
0309  75 04                1156  C                JNZ     C6                      ; CONTINUE TEST SEQUENCE TILL 0
030B  8A D4                1157  C                MOV     DL,AH                   ;  ELSE SET 0 FOR END READ PATTERN
030D  EB E0                1158  C                JMP     C3                      ;  AND MAKE FINAL BACKWARDS PASS
030F                       1159  C  C6:
030F  FC                   1160  C                CLD                             ; SET DIR FLAG TO GO FORWARD
0310  47                   1161  C                INC     DI                      ; SET POINTER TO BEG LOCATION
0311  74 DE                1162  C                JZ      C4                      ; READ/WRITE FORWARD IN STG
0313  4F                   1163  C                DEC     DI                      ; ADJUST POINTER
0314  EB D9                1164  C                JMP     C3                      ; READ/WRITE BACKWARD IN STG
0316                       1165  C  C6X:
0316  B0 00                1166  C                MOV     AL,000H                 ; AL=0 DATA COMPARE OK
0318                       1167  C  C7:
0318  FC                   1168  C                CLD                             ; SET DIRECTION FLAG BACK TO INC
0319  C3                   1169  C                RET
031A                       1170  C  STGTST  ENDP
                           1171  C
                           1172  C  ;--------------------------------------------------------
                           1173  C  ;                EGA CRT ATTACHMENT TEST                 :
                           1174  C  ;                                                        :
                           1175  C  ; 1. INIT CRT TO 40X25 - BW  ****SET TO MODE****       :
                           1176  C  ; 2. CHECK FOR VERTICAL AND VIDEO ENABLES, AND CHECK    :
                           1177  C  ;    TIMING OF SAME                                     :
                           1178  C  ; 3. CHECK VERTICAL INTERRUPT                           :
                           1179  C  ; 4. CHECK RED, BLUE, GREEN, AND INTENSIFY DOTS         :
                           1180  C  ; 5. INIT TO 40X25 - COLOR/MONO ****SET TO MODE****    :
                           1181  C  ;--------------------------------------------------------
                           1182  C
                           1183  C  ;----- NOMINAL TIME IS 8286H FOR 60 HZ.
                           1184  C  ;----- NOMINAL TIME IS A2FEH FOR 50 HZ.
                           1185  C
= AOAC                     1186  C  MAX_VERT_COLOR  EQU     0A0ACH                  ; MAX TIME FOR VERT/VERT
                           1187  C                                                  ;  (NOMINAL + 10%)
= C460                     1188  C  MIN_VERT_COLOR  EQU     0C460H                  ; MIN TIME FOR VERT/VERT
                           1189  C                                                  ;  (NOMINAL - 10%)
= 00C8                     1190  C  CENAB_PER_FRAME EQU     200                     ; NUM OF ENABLES PER FRAME
= 8D99                     1191  C  MAX_VERT_MONO   EQU     08D99H                  ; MAX TIME FOR VERT/VERT
                           1192  C                                                  ;  (NOMINAL + 10%)
= B862                     1193  C  MIN_VERT_MONO   EQU     0B862H                  ; MIN TIME FOR VERT/VERT
                           1194  C                                                  ;  (NOMINAL - 10%)
= 015E                     1195  C  eENAB_PER_FRAME equ     350                     ; enhanced enables per frame
= 015E                     1196  C  MENAB_PER_FRAME EQU     350                     ; NUM OF ENABLES PER FRAME
                           1197  C
= 0043                     1198  C  TIM_CTL         EQU     043H                    ; 8253 TIMER CONTROL PORT
= 0040                     1199  C  TIMER0          EQU     040H                    ; 8253 TIMER/CNTER 0 PORT
                           1200  C
031A                       1201  C  POD14   PROC    NEAR
031A  83 EC 0A             1202  C          SUB     SP,0AH                          ; RESERVE 5 WORDS ON STACK
031D  8B EC                1203  C          MOV     BP,SP                           ; INIT SCRATCH PAD POINTER
                           1204  C
                           1205  C          ASSUME  DS:ABS0,ES:ABS0
031F  E8 0D01 R            1206  C          CALL    DDS
0322  B0 30                1207  C          MOV     AL,00110000B                    ; SET TIMER 0 TO MODE 0
                           1208  C
0324  E6 43                1209  C          OUT     TIM_CTL,AL
0326  B0 00                1210  C          MOV     AL,00H
0328  E6 40                1211  C          OUT     TIMER0,AL                       ; SEND FIRST BYTE TO TIMER
032A  F6 06 0487 R 02      1212  C          test    info,2
032F  74 1F                1213  C          JZ      COLOR_EGA_V
0331  E8 016B R            1214  C          CALL    ENV_3                           ; SET UP IN MONOCHROME
0334  C7 46 02 015E        1215  C          MOV     WORD PTR[BP][2],MENAB_PER_FRAME ; NUM.OF FRAMES FOR MONO
0339  C7 46 04 8D99        1216  C          MOV     WORD PTR[BP][4],MAX_VERT_MONO   ; MAX TIME FOR VERT/VERT
033E  C7 46 06 B862        1217  C          MOV     WORD PTR[BP][6],MIN_VERT_MONO   ; MIN TIME FOR VERT/VERT
0343  B2 B4                1218  C          MOV     DL,CRTC_ADDR_B                  ; MONO CRTC REG
0345  B4 01                1219  C          MOV     AH,C_HRZ_DSP                    ; HORIZ. TOTAL DIPLAY
0347  B0 27                1220  C          MOV     AL,27H                          ;  TO 40 COL
0349  E8 0D18 R            1221  C          CALL    OUT_DX                          ;
034C  B2 BA                1222  C          MOV     DL,INPUT_STATUS_B               ; 3BA
034E  EB 2A                1223  C          jmp     short   common
0350                       1224  C  COLOR_EGA_V:
0350  E8 014B R            1225  C          CALL    ENV_X                           ; SET UP IN 40X25 COLOR
0353  E8 0E9C R            1226  C          call    brst_det                        ; enhanced mode
0356  73 11                1227  C          jnc     color_v                         ; no,40x25
0358  B2 D4                1228  C          mov     dl,crtc_addr                    ; brst mode only!
035A  B4 01                1229  C          mov     ah,1                            ; hrz dsp end
035C  B0 14                1230  C          mov     al,20                           ; modify for test only
035E  E8 0D18 R            1231  C          call    out_dx
0361  C7 46 02 015E        1232  C          MOV     WORD PTR[BP][2],eENAB_PER_FRAME ; NUM.OF FRAMES FOR COLOR
0366  EB 06 90             1233  C          jmp     brst_color_v
0369                       1234  C  color_V:
                           1235  C
0369  C7 46 02 00C8        1236  C          MOV     WORD PTR[BP][2],CENAB_PER_FRAME ; NUM.OF FRAMES FOR COLOR
036E                       1237  C  brst_color_v:
036E  C7 46 04 AOAC        1238  C          MOV     WORD PTR[BP][4],MAX_VERT_COLOR  ; MAX TIME FOR VERT/VERT
0373  C7 46 06 C460        1239  C          MOV     WORD PTR[BP][6],MIN_VERT_COLOR  ; MIN TIME FOR VERT/VERT
0378  B2 DA                1240  C          MOV     DL,INPUT_STATUS                 ; SET ADDRESSING TO VIDEO
                           1241  C                                                  ;  ATTR STATUS
037A                       1242  C  COMMON:
037A  B8 0500              1243  C          MOV     AX,0500H                        ; SET TO VIDEO PAGE 0
037D  CD 10                1244  C          INT     10H
037F  2B C9                1245  C          SUB     CX,CX
                           1246  C
                           1247  C  ;----- LOOK FOR VERTICAL
                           1248  C
0381                       1249  C  POD14_1:
0381  EC                   1250  C          IN      AL,DX                           ; GET STATUS
0382  A8 08                1251  C          TEST    AL,00001000B                    ; VERTICAL THERE YET?
0384  75 07                1252  C          JNE     POD14_2                         ; CONTINUE IF IT IS
0386  E2 F9                1253  C          LOOP    POD14_1                         ; KEEP LOOKING TILL COUNT
0388  B3 00                1254  C          MOV     BL,00                           ;  EXHAUSTED
038A  E9 044B R            1255  C          JMP     POD14_ERR                       ; NO VERTICAL
                           1256  C
                           1257  C  ;----- GOT VERTICAL - START TIMER
                           1258  C
038D                       1259  C  POD14_2:
038D  B0 00                1260  C          MOV     AL,0
```

```
038F  E6 40                1261  C          OUT     TIMER0,AL                       ; SEND 2ND BYTE TO TIMER TO
                           1262  C                                                  ; START IT
0391  2B DB                1263  C          SUB     BX,BX                           ; INIT. ENABLE COUNTER
                           1264  C  ;----- WAIT FOR VERTICAL TO GO AWAY
0393  33 C9                1265  C          XOR     CX,CX
                           1266  C  POD14_25:
0395  EC                   1267  C          IN      AL,DX                           ; GET STATUS
0396  A8 08                1268  C          TEST    AL,00001000B                    ; VERTICAL STILL THERE
0398  74 07                1269  C          JZ      POD14_3                         ; CONTINUE IF IT'S GONE
039A  E2 F9                1270  C          LOOP    POD14_25                        ; KEEP LOOKING TILL COUNT
039C  B3 01                1271  C          MOV     BL,01H                          ;  EXHAUSTED
039E  E9 044B R            1272  C          JMP     POD14_ERR                       ; VERTICAL STUCK ON
                           1273  C
                           1274  C  ;----- NOW START LOOKING FOR ENABLE TRANSITIONS
                           1275  C
03A1                       1276  C  POD14_3:
03A1  2B C9                1277  C          SUB     CX,CX
03A3                       1278  C  POD14_4:
03A3  EC                   1279  C          IN      AL,DX                           ; GET STATUS
03A4  A8 01                1280  C          TEST    AL,00000001B                    ; ENABLE ON YET?
03A6  74 15                1281  C          JE      POD14_5                         ; GO ON IF IT IS
03A8  A8 08                1282  C          TEST    AL,00001000B                    ; VERTICAL ON AGAIN?
03AA  75 23                1283  C          JNE     POD14_75                        ; CONTINUE IF IT IS
03AC  E2 F5                1284  C          LOOP    POD14_4                         ; KEEP LOOKING IF NOT
03AE  B3 02                1285  C          MOV     BL,02H
03B0  E9 044B R            1286  C          JMP     POD14_ERR                       ; ENABLE STUCK OFF
03B3                       1287  C  POD14_4A:
03B3  B3 03                1288  C          MOV     BL,03H
03B5  E9 044B R            1289  C          JMP     POD14_ERR                       ; VERTICAL STUCK ON
03B8                       1290  C  POD14_4B:
03B8  B3 04                1291  C          MOV     BL,04H
03BA  E9 044B R            1292  C          JMP     POD14_ERR                       ; ENABLE STUCK ON
                           1293  C
                           1294  C  ;----- MAKE SURE VERTICAL WENT OFF WITH ENABLE GOING ON
                           1295  C
03BD                       1296  C  POD14_5:
03BD  A8 08                1297  C          TEST    AL,00001000B                    ; VERTICAL OFF?
03BF  75 F2                1298  C          JNZ     POD14_4A                        ; GO ON IF IT IS
                           1299  C  ;----- NOW WAIT FOR ENABLE TO GO OFF            ;  (ERROR IF NOT)
03C1                       1300  C  POD14_6:
03C1  EC                   1301  C          IN      AL,DX                           ; GET STATUS
03C2  A8 01                1302  C          TEST    AL,00000001B                    ; ENABLE OFF YET?
03C4  E1 FB                1303  C          LOOPE   POD14_6                         ; KEEP LOOKING IF NOT
03C6  E3 F0                1304  C          JCXZ    POD14_4B                        ;  YET LOW
                           1305  C  ;----- ENABLE HAS TOGGLED, BUMP COUNTER AND TEST FOR NEXT VERTICAL
03C8                       1306  C  POD14_7:
03C8  43                   1307  C          INC     BX                              ; BUMP ENABLE COUNTER
03C9  74 04                1308  C          JZ      POD14_75                        ; IF COUNTER WRAPS,
                           1309  C                                                  ;  SOMETHING IS WRONG
03CB  A8 08                1310  C          TEST    AL,00001000B                    ; DID ENABLE GO LOW
                           1311  C                                                  ;  BECAUSE OF VERTICAL
03CD  74 D2                1312  C          JZ      POD14_3                         ; IF NOT, LOOK FOR ANOTHER
                           1313  C                                                  ;  ENABLE TOGGLE
                           1314  C  ;----- HAVE HAD COMPLETE VERTICAL-VERTICAL CYCLE, NOW TEST RESULTS
03CF                       1315  C  POD14_75:
03CF  B0 00                1316  C          MOV     AL,00                           ; LATCH TIMER0
03D1  E6 43                1317  C          OUT     TIM_CTL,AL
03D3  3B 5E 02             1318  C          CMP     BX,WORD PTR[BP][2]              ; NUMBER OF ENABLES BETWEEN
                           1319  C                                                  ;  VERTICALS O.K.?
03D6  74 04                1320  C          JE      POD14_8
03D8  B3 05                1321  C          MOV     BL,05H
03DA  EB 6F                1322  C          jmp     short pod14_err
03DC                       1323  C  POD14_8:
03DC  E4 40                1324  C          IN      AL,TIMER0                       ; GET TIMER VALUE LOW
03DE  8A E0                1325  C          MOV     AH,AL                           ; SAVE IT
03E0  90                   1326  C          NOP
03E1  E4 40                1327  C          IN      AL,TIMER0                       ; GET TIMER HIGH
03E3  86 E0                1328  C          XCHG    AH,AL
03E5  90                   1329  C          NOP
03E6  90                   1330  C          NOP
03E7  3B 46 04             1331  C          CMP     AX,WORD PTR[BP][4]              ; MAXIMUM VERTICAL TIMING
03EA  7D 04                1332  C          JGE     POD14_9
03EC  B3 06                1333  C          MOV     BL,06H
03EE  EB 5B                1334  C          jmp     short pod14_err
03F0                       1335  C  POD14_9:
03F0  3B 46 06             1336  C          CMP     AX,WORD PTR[BP][6]              ; MINIMUM VERTICAL TIMING
03F3  7E 04                1337  C          JLE     POD14_10
03F5  B3 07                1338  C          MOV     BL,07H
03F7  EB 52                1339  C          jmp     short   pod14_err
                           1340  C
                           1341  C  ;----- SEE IF RED, GREEN, BLUE AND INTENSIFY DOTS WORK
                           1342  C
                           1343  C  ;----- FIRST, SET A LINE OF REVERSE VIDEO, INTENSIFIED BLANKS INTO BUFFER
03F9                       1344  C  POD14_10:
03F9  B8 09DB              1345  C          MOV     AX,09DBH                        ; WRITE CHARS, BLANKS
03FC  BB 000F              1346  C          MOV     BX,000FH                        ; PAGE 0, REVERSE VIDEO,
                           1347  C                                                  ;  HIGH INTENSITY
03FF  B9 0050              1348  C          MOV     CX,80                           ; 80 CHARACTERS
0402  CD 10                1349  C          INT     10H
0404  EC                   1350  C          IN      AL,DX
0405  52                   1351  C          PUSH    DX                              ; SAVE INPUT STATUS
0406  B2 C0                1352  C          MOV     DL,ATTR_WRITE                   ; ATTRIBUTE ADDRESSS
0408  B4 0F                1353  C          MOV     AH,0FH                          ; PALETTE REG 'F'
040A  B0 3F                1354  C          MOV     AL,03FH                         ; TEST VALUE
040C  E8 0D18 R            1355  C          CALL    OUT_DX                          ; VIDEO STATUS MUX
040F  B8 000F              1356  C          MOV     AX,0FH                          ; START WITH BLUE DOTS
0412  5A                   1357  C          POP     DX
0413                       1358  C  POD14_13:
0413  50                   1359  C          PUSH    AX                              ; SAVE
0414  52                   1360  C          PUSH    DX                              ; SAVE INPUT STATUS
0415  B2 C0                1361  C          MOV     DL,ATTR_WRITE                   ; ATTRIBUTE ADDRESSS
0417  B4 32                1362  C          MOV     AH,32H                          ; COLOR PLANE ENABLE
0419  E8 0D18 R            1363  C          CALL    OUT_DX                          ; VIDEO STATUS MUX
041C  5A                   1364  C          POP     DX                              ; RECOVER INPUT STATUS
041D  58                   1365  C          POP     AX
041E  2B C9                1366  C          SUB     CX,CX
                           1367  C  ;----- SEE IF DOT COMES ON
0420                       1368  C  POD14_14:
0420  EC                   1369  C          IN      AL,DX                           ; GET STATUS
0421  A8 30                1370  C          TEST    AL,00110000B                    ; DOT THERE?
0423  75 09                1371  C          JNZ     POD14_15                        ; LOOK FOR DOT TO TURN OFF
0425  E2 F9                1372  C          LOOP    POD14_14                        ; CONTINUE TEST FOR DOT ON
0427  B3 10                1373  C          MOV     BL,10H
0429  0A DC                1374  C          OR      BL,AH                           ; OR IN DOT BEING TESTED
042B  EB 1E 90             1375  C          JMP     POD14_ERR                       ; DOT NOT COMING ON
                           1376  C  ;----- SEE IF DOT GOES OFF
042E                       1377  C  POD14_15:
042E  2B C9                1378  C          SUB     CX,CX
0430                       1379  C  POD14_16:
0430  EC                   1380  C          IN      AL,DX                           ; GET STATUS
0431  A8 30                1381  C          TEST    AL,00110000B                    ; IS DOT STILL ON?
0433  74 08                1382  C          JE      POD14_17                        ; GO ON IF DOT OFF
0435  E2 F9                1383  C          LOOP    POD14_16                        ; ELSE, KEEP WAITING FOR
                           1384  C                                                  ;  DOT TO GO OFF
0437  B3 20                1385  C          MOV     BL,20H
0439  0A DC                1386  C          OR      BL,AH                           ; OR IN DOT BEING TESTED
```

```
043B  EB 0E                                1387  C          jmp     short pod14_err
                                           1388  C
                                           1389  C  ;----- ADJUST TO POINT TO NEXT DOT
                                           1390  C
043D                                       1391  C  POD14_17:
043D  FE C4                                1392  C          INC     AH
043F  80 FC 30                             1393  C          CMP     AH,030H                 ; ALL 3 DOTS DONE?
0442  74 25                                1394  C          JE      POD14_18                ; GO END
0444  80 CC 0F                             1395  C          OR      AH,0FH                  ; MAKE 0F,1F,2F
0447  8A C4                                1396  C          MOV     AL,AH
0449  EB C8                                1397  C          JMP     POD14_13                ; GO LOOK FOR ANOTHER DOT
044B                                       1398  C  POD14_ERR:
044B  B9 0006                              1399  C          MOV     CX,6
044E  BA 0103                              1400  C          MOV     DX,0103H                ; ONE LONG AND THREE SHORT
0451  E8 06CB R                           1401  C          CALL    ERR_BEEP
0454  83 C4 0A                             1402  C          ADD     SP,0ah                  ; BALANCE STACK
0457  B0 36                                1403  C          MOV     AL,00110110B            ; RE-INIT TIMER 0
0459  E6 43                                1404  C          OUT     TIM_CTL,AL
045B  2A C0                                1405  C          SUB     AL,AL
045D  E6 40                                1406  C          OUT     TIMER0,AL
045F  90                                   1407  C          NOP
0460  90                                   1408  C          NOP
0461  E6 40                                1409  C          OUT     TIMER0,AL
0463  BD 0001                             1410  C          MOV     BP,1
0466  E9 0092 R                           1411  C          JMP     SKIP
                                           1412  C          ASSUME  DS:ABS0
0469                                       1413  C  POD14_18:
0469  E8 0DD1 R                           1414  C          CALL    DDS
046C  B8 0500                              1415  C          MOV     AX,0500H                ; SET TO VIDEO PAGE 0
046F  CD 10                                1416  C          INT     10H
0471  B0 36                                1417  C          MOV     AL,00110110B            ; RE-INIT TIMER 0
0473  E6 43                                1418  C          OUT     TIM_CTL,AL
0475  2A C0                                1419  C          SUB     AL,AL
0477  E6 40                                1420  C          OUT     TIMER0,AL
0479  90                                   1421  C          NOP
047A  90                                   1422  C          NOP
047B  E6 40                                1423  C          OUT     TIMER0,AL
047D  83 C4 0A                             1424  C          ADD     SP,0AH                  ; REMOVE SCRATCH PAD
0480  BD 0000                             1425  C          MOV     BP,0                    ; MAKE BP NON ZERO
0483                                       1426  C  POD14   ENDP
                                           1427  C
                                           1428  C  ;----- TEST STORAGE
                                           1429  C
0483                                       1430  C  MEM_TEST:
0483  1E                                   1431  C          PUSH    DS
0484  E8 0DD1 R                           1432  C          CALL    DDS
                                           1433  C          ASSUME DS:ABS0
0487  F6 06 0487 R 02                      1434  C          test    info,2
048C  74 12                                1435  C          JZ      D_COLOR_M
048E  80 0E 0410 R 30                      1436  C          or      equip_low,030h
0493  B8 000F                             1437  C          mov     ax,0fh
0496  80 0E 0487 R 60                      1438  C          or      info,060h
049B  B8 000F                             1439  C          mov     ax,0fh
049E  EB 00                                1440  C          jmp     short d_out_m
04A0                                       1441  C  D_COLOR_M:
04A0  80 26 0410 R CF                      1442  C          and     equip_low,0cfh
04A5  80 0E 0410 R 20                      1443  C          or      equip_low,020h
04AA  B8 000E                             1444  C          MOV     AX,0EH                  ; INTERNAL COLOR MODE
04AD                                       1445  C  D_OUT_M:                              ; TEST IN COLOR
04AD  CD 10                                1446  C          INT     10H                     ; OUT 5/4/83
04AF  83 EC 06                             1447  C          SUB     SP,6                    ; RESERVE 3 WORDS ON STACK
04B2  8B EC                                1448  C          MOV     BP,SP                   ; SET BP
04B4  B8 A000                             1449  C          MOV     AX,0A000H               ; PUT BUFFER ADDRESS IN AX
                                           1450  C          ASSUME  DS:NOTHING,ES:NOTHING
04B7  8E D8                                1451  C          MOV     DS,AX                   ; SET UP SEG REGS TO POINT
04B9  8E C0                                1452  C          MOV     ES,AX                   ; TO BUFFER AREA
04BB  C7 46 02 0000                        1453  C          MOV     WORD PTR[BP][2],0       ; INITIALIZE
04C0  C7 46 04 0000                        1454  C          MOV     WORD PTR[BP][4],0       ; INITIALIZE
04C5  B6 03                                1455  C          mov     dh,3
04C7  B2 C4                                1456  C          MOV     DL,SEQ_ADDR
04C9  B8 0201                             1457  C          mov     ax,0201h
04CC  E8 0D18 R                           1458  C          CALL    OUT_DX
04CF  B2 CE                                1459  C          MOV     DL,GRAPH_ADDR           ; ADDRESS READ MAP SELECT
04D1  B8 0400                             1460  C          mov     ax,0400h
04D4  E8 0D18 R                           1461  C          CALL    OUT_DX
04D7  52                                   1462  C          PUSH    DX
04D8  B2 DA                                1463  C          MOV     DL,ATTR_READ            ; SET UP ATTRIBUTE
04DA  EC                                   1464  C          IN      AL,DX
04DB  B2 C0                                1465  C          MOV     DL,ATTR_WRITE           ; ATTRIBUTE WRITE ADDRESS
04DD  B8 3200                             1466  C          mov     ax,3200h
04E0  E8 0D18 R                           1467  C          CALL    OUT_DX
04E3  E8 0692 R                           1468  C          CALL    HOW_BIG                 ; GO FIND AMOUNT OF MEMORY
04E6  80 FC 00                             1469  C          CMP     AH,0
04E9  74 03                                1470  C          JZ      AA1
04EB  E9 05D0 R                           1471  C          JMP     EGA_MEM_ERROR
04EE                                       1472  C  AA1:
04EE  E8 05DC R                           1473  C          CALL    MEMORY_OK               ; GO TEST IT
04F1  80 FC 00                             1474  C          CMP     AH,0
04F4  74 03                                1475  C          JZ      AA2
04F6  E9 05D0 R                           1476  C          JMP     EGA_MEM_ERROR
04F9                                       1477  C  AA2:
04F9  5A                                   1478  C          POP     DX
04FA  B2 C4                                1479  C          MOV     DL,SEQ_ADDR
04FC  B8 0202                             1480  C          mov     ax,0202h
04FF  E8 0D18 R                           1481  C          CALL    OUT_DX
0502  B2 CE                                1482  C          MOV     DL,GRAPH_ADDR           ; ADDRESS OF READ MAP
0504  B8 0401                             1483  C          mov     ax,0401h
0507  E8 0D18 R                           1484  C          CALL    OUT_DX
050A  52                                   1485  C          PUSH    DX
050B  B2 DA                                1486  C          MOV     DL,ATTR_READ            ; SET UP ATTRIBUTE
050D  EC                                   1487  C          IN      AL,DX
050E  B2 C0                                1488  C          MOV     DL,ATTR_WRITE           ; ATTRIBUTE WRITE ADDRESS
0510  B8 3200                             1489  C          mov     ax,3200h
0513  E8 0D18 R                           1490  C          CALL    OUT_DX
0516  C7 46 04 0000                        1491  C          MOV     WORD PTR [BP][4],0      ; INITIALIZE
051B  E8 0692 R                           1492  C          CALL    HOW_BIG                 ; GO FIND AMOUNT OF MEMORY
051E  80 FC 00                             1493  C          CMP     AH,0
0521  74 03                                1494  C          JZ      AA3
0523  E9 05D0 R                           1495  C          JMP     EGA_MEM_ERROR
0526                                       1496  C  AA3:
0526  E8 05DC R                           1497  C          CALL    MEMORY_OK               ; GO TEST IT
0529  80 FC 00                             1498  C          CMP     AH,0
052C  74 03                                1499  C          JZ      AA4
052E  E9 05D0 R                           1500  C          JMP     EGA_MEM_ERROR
0531                                       1501  C  AA4:
0531  5A                                   1502  C          POP     DX
0532  B2 C4                                1503  C          MOV     DL,SEQ_ADDR
0534  B8 0204                             1504  C          mov     ax,0204h
0537  E8 0D18 R                           1505  C          CALL    OUT_DX
053A  52                                   1506  C          PUSH    DX
053B  B2 CE                                1507  C          MOV     DL,GRAPH_ADDR           ; ADDRESS OF READ MAP
053D  B8 0402                             1508  C          mov     ax,0402h
0540  E8 0D18 R                           1509  C          CALL    OUT_DX
0543  B2 DA                                1510  C          MOV     DL,ATTR_READ            ; SET UP ATTRIBUTE
0545  EC                                   1511  C          IN      AL,DX
0546  B2 C0                                1512  C          MOV     DL,ATTR_WRITE           ; ATTRIBUTE WRITE ADDRESS
```

```
0548  B8 3200                1513    C           mov     ax,3200h
054B  E8 0D18 R              1514    C           CALL    OUT_DX
054E  C7 46 04 0000          1515    C           MOV     WORD PTR[BP][4],0           ; INITIALIZE
0553  E8 0692 R              1516    C           CALL    HOW_BIG                     ; GO FIND AMOUNT OF MEMORY
0556  80 FC 00               1517    C           CMP     AH,0
0559  74 03                  1518    C           JZ      AA5
055B  E8 73 90               1519    C           JMP     EGA_MEM_ERROR
055E                         1520    C   AA5:
055E  E8 05DC R              1521    C           CALL    MEMORY_OK                   ; GO TEST IT
0561  80 FC 00               1522    C           CMP     AH,0
0564  74 03                  1523    C           JZ      AA6
0566  E8 68 90               1524    C           JMP     EGA_MEM_ERROR
0569                         1525    C   AA6:
0569  5A                    1526    C           POP     DX
056A  B2 C4                  1527    C           MOV     DL,SEQ_ADDR
056C  B8 0208                1528    C           mov     ax,0208h
056F  E8 0D18 R              1529    C           CALL    OUT_DX
0572  B2 CE                  1530    C           MOV     DL,GRAPH_ADDR               ; ADDRESS OF READ MAP
0574  B8 0403                1531    C           mov     ax,0403h
0577  E8 0D18 R              1532    C           CALL    OUT_DX
057A  52                    1533    C           PUSH    DX
057B  B2 DA                  1534    C           MOV     DL,ATTR_READ                ; SET UP ATTRIBUTE
057D  EC                    1535    C           IN      AL,DX
057E  B2 C0                  1536    C           MOV     DL,ATTR_WRITE               ; ATTRIBUTE WRITE ADDRESS
0580  B8 3200                1537    C           mov     ax,3200h
0583  E8 0D18 R              1538    C           CALL    OUT_DX
0586  C7 46 04 0000          1539    C           MOV     WORD PTR[BP][4],0           ; INITIALIZE
058B  E8 0692 R              1540    C           CALL    HOW_BIG                     ; GO FIND AMOUNT OF MEMORY
058E  80 FC 00               1541    C           CMP     AH,0
0591  75 3D                  1542    C           JNZ     EGA_MEM_ERROR
0593  E8 05DC R              1543    C           CALL    MEMORY_OK                   ; GO TEST IT
0596  80 FC 00               1544    C           CMP     AH,0
0599  75 35                  1545    C           JNZ     EGA_MEM_ERROR
059B  55                    1546    C           PUSH    BP                          ; SAVE SCRATCH PAD POINTER
059C  BD 0000                1547    C           MOV     BP,0                        ; RESET BP FOR XT
059F                         1548    C   EGA_MEM_EXIT:
059F  5E                    1549    C           POP     SI                          ; RESTORE
05A0  5A                    1550    C           POP     DX
05A1  E8 0D01 R              1551    C           CALL    DDS                         ; SET DATA SEGMENT
                             1552    C           ASSUME  DS:ABS0
05A4  36: 8B 5C 02           1553    C           MOV     BX,WORD PTR SS:[SI][2]      ; GET EGA MEMORY SIZE
05A8  B1 06                  1554    C           MOV     CL,06H                      ; DIVIDE BY 64 TO GET
05AA  D3 EB                  1555    C           SHR     BX,CL                       ;  NUMBER OF 64KB BLOCKS
05AC  4B                    1556    C           dec     bx
05AD  B1 05                  1557    C           MOV     CL,05H
05AF  D3 E3                  1558    C           SHL     BX,CL
05B1  80 E3 60               1559    C           AND     BL,01100000B                ; ISOLATE BITS 5 AND 6
                             1560    C
05B4  80 26 0487 R 9F        1561    C           and     info,10011111b
                             1562    C
05B9  08 1E 0487 R           1563    C           OR      INFO,BL
                             1564    C
05BD  80 0E 0487 R 04        1565    C           OR      INFO,00000100B              ; 04H  SET 3XX ACTIVE
05C2  8A 1E 0488 R           1566    C           MOV     BL,INFO_3
05C6  E8 00F6 R              1567    C           CALL    MK_ENV
05C9  83 C4 06               1568    C           ADD     SP,6                        ; RESTORE STACK
05CC  1F                    1569    C           POP     DS
05CD  E9 0092 R              1570    C           JMP     SKIP                        ; GO TO END
05D0                         1571    C   EGA_MEM_ERROR:
05D0  BA 0103                1572    C           MOV     DX,0103H                    ; ONE LONG AND THREE SHORT
05D3  E8 06CB R              1573    C           CALL    ERR_BEEP
05D6  55                    1574    C           PUSH    BP                          ; SAVE SCRATCH PAD POINTER
05D7  BD 0001                1575    C           MOV     BP,1                        ; INDICATE ERROR FOR XT
05DA  EB C3                  1576    C           JMP     EGA_MEM_EXIT
                             1577    C
                             1578    C   ;----- THIS ROUTINE FINDS AMOUNT OF MEMORY GOOD
                             1579    C
05DC                         1580    C   MEMORY_OK      PROC    NEAR
05DC  BB A000                1581    C           MOV     BX,0A000H                   ; SET PTR. TO BUFFER SEG
05DF  8E DB                  1582    C           MOV     DS,BX                       ; SET SEG.REG.
05E1  8E C3                  1583    C           MOV     ES,BX
05E3  8B 46 04               1584    C           MOV     AX,WORD PTR[BP][4]          ; SET COUNT FOR 32K WORDS
05E6  8A E8                  1585    C           MOV     CH,AL                       ; SET AMOUNT OF BUFFER
05E8  2A C9                  1586    C           SUB     CL,CL                       ;  TO BE TESTED
05EA  D1 E1                  1587    C           SHL     CX,1                        ; MULTIPLY BY TWO
05EC  E8 05FE R              1588    C           CALL    PODSTG
05EF  80 FC 00               1589    C           CMP     AH,0                        ; TEST FOR ERROR
05F2  75 09                  1590    C           JNZ     MEMORY_OK_ERR               ; IF ERROR GO PRINT IT
05F4                         1591    C   MEMORY_OK_EX:
05F4  8B 46 04               1592    C           MOV     AX,WORD PTR[BP][4]          ; AMOUNT OF MEMORY FOUND
05F7  01 46 02               1593    C           ADD     WORD PTR[BP][2],AX          ; AMOUNT OF MEMORY GOOD
05FA  B8 0000                1594    C           MOV     AX,0
05FD                         1595    C   MEMORY_OK_ERR:
05FD  C3                    1596    C           RET
05FE                         1597    C   MEMORY_OK      ENDP
                             1598    C
                             1599    C   ;--------------------------------------------------------------
                             1600    C   ; THIS ROUTINE PERFORMS A READ/WRITE TEST ON A BLOCK OF STORAGE :
                             1601    C   ; (MAX. SIZE = 32KW). IF "WARM START", FILL BLOCK WITH 0000 AND :
                             1602    C   ; RETURN.                                                       :
                             1603    C   ; ON ENTRY:                                                    :
                             1604    C   ;   ES = ADDRESS OF STORAGE TO BE TESTED                       :
                             1605    C   ;   DS = ADDRESS OF STORAGE TO BE TESTED                       :
                             1606    C   ;   CX = WORD COUNT OF STORAGE BLOCK TO BE TESTED              :
                             1607    C   ;        (MAX. = 8000H (32K WORDS))                            :
                             1608    C   ; ON EXIT:                                                     :
                             1609    C   ;   ZERO FLAG = OFF IF STORAGE ERROR                          :
                             1610    C   ; AX,BX,CX,DX,DI,SI ARE ALL DESTROYED.                         :
                             1611    C   ;--------------------------------------------------------------
05FE                         1612    C   PODSTG  PROC    NEAR
05FE  55                    1613    C           PUSH    BP
05FF  FC                    1614    C           CLD                                 ; SET DIR TO INCREMENT
0600  2B FF                  1615    C           SUB     DI,DI                       ; SET DI=0000 REL TO START
                             1616    C                                               ;  OF SEGMENT
0602  2B C0                  1617    C           SUB     AX,AX                       ; INITIAL DATA PATTERN FOR
                             1618    C                                               ;  00-FF TEST
0604  E8 0D01 R              1619    C           CALL    DDS
                             1620    C           ASSUME  DS:ABS0
0607  8B 1E 0472 R           1621    C           MOV     BX,DS:RESET_FLAG            ; WARM START?
060B  81 FB 1234             1622    C           CMP     BX,1234H
060F  8C C2                  1623    C           MOV     DX,ES
0611  8E DA                  1624    C           MOV     DS,DX                       ; RESTORE DS
0613  74 62                  1625    C           JE      PODSTG_5                    ; GO DO FILL WITH 0000
                             1626    C                                               ;  IF WARM START
0615  81 FB 4321             1627    C           CMP     BX,4321H                    ; DCP WARM START?
0619  74 5C                  1628    C           JE      PODSTG_5                    ; DO FILL IF SO
061B                         1629    C   PODSTG_1:
061B  88 05                  1630    C           MOV     [DI],AL                     ; WRITE TEST DATA
061D  8A 05                  1631    C           MOV     AL,[DI]                     ; GET IT BACK
061F  32 C4                  1632    C           XOR     AL,AH                       ; COMPARE TO EXPECTED
0621  75 40                  1633    C           JNZ     PODSTG_ERRO                 ; ERROR EXIT IF MISCOMPARE
0623  FE C4                  1634    C           INC     AH                          ; FORM NEW DATA PATTERN
0625  8A C4                  1635    C           MOV     AL,AH
0627  75 F2                  1636    C           JNZ     PODSTG_1                    ; LOOP TILL ALL 256 DATA
                             1637    C                                               ;  PATTERNS DONE
0629  8B E9                  1638    C           MOV     BP,CX                       ; SAVE WORD COUNT
```

```
062B  B8 AA55                1639  C          MOV     AX,0AA55H                       ; LOAD DATA PATTERN
062E  8B D8                  1640  C          MOV     BX,AX
0630  BA 55AA                1641  C          MOV     DX,055AAH                       ; LOAD OTHER DATA PATTERN
0633  F3/ AB                 1642  C          REP     STOSW                           ; FILL WORDS FROM LOW TO
                             1643  C                                                  ;  HIGH WITH AAAA
0635  4F                     1644  C          DEC     DI                              ; POINT TO LAST WORD
0636  4F                     1645  C          DEC     DI                              ;  WRITTEN
0637  FD                     1646  C          STD                                     ; SET DIR FLAG TO GO DOWN
0638  8B F7                  1647  C          MOV     SI,DI                           ; SET INDEX REGS. EQUAL
063A  8B CD                  1648  C          MOV     CX,BP                           ; RECOVER WORD COUNT
                             1649  C  PODSTG_2:                                       ; GO FROM HIGH TO LOW
063C  AD                     1650  C          LODSW                                   ; GET WORD FROM MEMORY
063D  33 C3                  1651  C          XOR     AX,BX                           ; EQUAL WHAT S/B THERE?
063F  75 22                  1652  C          JNZ     PODSTG_ERR0                     ; GO ERROR EXIT IF NOT
0641  8B C2                  1653  C          MOV     AX,DX                           ; GET 55 DATA PATTERN AND
0643  AB                     1654  C          STOSW                                   ;  STORE IN LOC JUST READ
0644  E2 F6                  1655  C          LOOP    PODSTG_2                        ; LOOP TILL ALL BYTES DONE
0646  8B CD                  1656  C          MOV     CX,BP                           ; RECOVER WORD COUNT
0648  FC                     1657  C          CLD                                     ; BACK TO INCREMENT
0649  46                     1658  C          INC     SI                              ; ADJUST PTRS
064A  46                     1659  C          INC     SI
064B  8B FE                  1660  C          MOV     DI,SI
064D                         1661  C  PODSTG_3:                                       ; LOW TO HIGH DOING WORDS
064D  AD                     1662  C          LODSW                                   ; GET A WORD
064E  33 C2                  1663  C          XOR     AX,DX                           ; SHOULD COMPARE TO DX
0650  75 11                  1664  C          JNZ     PODSTG_ERR0                     ; GO ERROR IF NOT
0652  AB                     1665  C          STOSW                                   ; WRITE 0000 BACK TO LOC
                             1666  C                                                  ;  JUST READ
0653  E2 F8                  1667  C          LOOP    PODSTG_3                        ; LOOP TILL DONE
                             1668  C
0655  FD                     1669  C          STD                                     ; BACK TO DECREMENT
0656  4E                     1670  C          DEC     SI                              ; ADJUST POINTER DOWN TO
                             1671  C                                                  ;  LAST WORD WRITTEN
0657  4E                     1672  C          DEC     SI
0658  8B CD                  1673  C          MOV     CX,BP                           ; GET WORD COUNT
065A                         1674  C  PODSTG_4:
065A  AD                     1675  C          LODSW                                   ; GET WORD
065B  0B C0                  1676  C          OR      AX,AX                           ; = TO 0000
065D  75 04                  1677  C          JNZ     PODSTG_ERR0                     ; ERROR IF NOT
065F  E2 F9                  1678  C          LOOP    PODSTG_4                        ; LOOP TILL DONE
0661  EB 11                  1679  C          jmp     short podstg_err2
0663                         1680  C  PODSTG_ERR0:
0663  8B C8                  1681  C          MOV     CX,AX                           ; SAVE BITS IN ERROR
0665  32 E4                  1682  C          XOR     AH,AH
0667  0A ED                  1683  C          OR      CH,CH                           ; HIGH BYTE ERROR?
0669  74 02                  1684  C          JZ      PODSTG_ERR1
066B  B4 01                  1685  C          MOV     AH,1                            ; SET HIGH BYTE ERROR
066D                         1686  C  PODSTG_ERR1:
066D  0A C9                  1687  C          OR      CL,CL                           ; LOW BYTE ERROR?
066F  74 03                  1688  C          JZ      PODSTG_ERR2
0671  80 C4 02               1689  C          ADD     AH,2
0674                         1690  C  PODSTG_ERR2:
0674  5D                     1691  C          POP     BP
0675  FC                     1692  C          CLD                                     ; SET DIR FLAG BACK TO INC
0676  C3                     1693  C          RET                                     ; RETURN TO CALLER
0677                         1694  C  PODSTG_5:                                       ; SIMPLE FILL WITH 0000 ON
                             1695  C                                                  ;  WARM-START
0677  50                     1696  C          PUSH    AX                              ; SAVE
0678  52                     1697  C          PUSH    DX                              ; SAVE VALUE
0679  B6 03                  1698  C          mov     dh,3
067B  B2 C4                  1699  C          MOV     DL,SEQ_ADDR                     ; SEQ_ADDR  REGISTER
067D  B8 020F                1700  C          mov     ax,020fh
0680  E8 0D18 R              1701  C          CALL    OUT_DX                          ; DO IT
0683  5A                     1702  C          POP     DX                              ; RESTORE
0684  58                     1703  C          POP     AX                              ; RESTORE
0685  F3/ AB                 1704  C          REP     STOSW
0687  E8 0D01 R              1705  C          CALL    DDS
                             1706  C          ASSUME  DS:ABS0
068A  89 1E 0472 R           1707  C          MOV     DS:RESET_FLAG,BX
068E  8E DA                  1708  C          MOV     DS,DX                           ; RESTORE DS
0690  EB E2                  1709  C          JMP     PODSTG_ERR2                     ; AND EXIT
0692                         1710  C  PODSTG  ENDP
                             1711  C
                             1712  C  ;----- DETERMINE SIZE OF BUFFER
                             1713  C
0692                         1714  C  HOW_BIG          PROC NEAR
0692  8C DA                  1715  C          MOV     DX,DS                           ; SET PNTR TO BUFFER LOC
0694  2B DB                  1716  C          SUB     BX,BX                           ; BASIC COUNT OF 00K
0696                         1717  C  FILL_LOOP:
0696  8E C2                  1718  C          MOV     ES,DX                           ; SET SEG. REG
0698  2B FF                  1719  C          SUB     DI,DI
069A  B8 AA55                1720  C          MOV     AX,0AA55H                       ; TEST PATTERN
069D  8B C8                  1721  C          MOV     CX,AX
069F  26: 89 05              1722  C          MOV     ES:[DI],AX                      ; SEND TO MEMORY
06A2  B0 0F                  1723  C          MOV     AL,0FH                          ; PUT SOMETHING IN AL
06A4  26: 8B 05              1724  C          MOV     AX,ES:[DI]                      ; GET PATTERN FROM MEMORY
06A7  33 C1                  1725  C          XOR     AX,CX                           ; COMPARE PATTERNS
06A9  75 14                  1726  C          JNZ     HOW_BIG_END                     ; GO END IF NO COMPARE
06AB  B9 2000                1727  C          MOV     CX,2000H                        ; SET COUNT FOR 8K WORDS
06AE  F3/ AB                 1728  C  REP      STOSW                                   ; FILL 8K WORDS
06B0  81 C2 0400             1729  C          ADD     DX,0400H                        ; POINT TO NEXT 16K BLOCK
06B4  83 C3 10               1730  C          ADD     BX,16                           ; BUMP COUNT BY 16KB
06B7  80 FE B0               1731  C          CMP     DH,0B0H
06BA  75 DA                  1732  C          JNZ     FILL_LOOP                       ; AREA YET ?(B0000H)
06BC  EB 01 90               1733  C          JMP     HOW_BIG_END
06BF                         1734  C  HOW_BIG_END:
06BF  80 FE A0               1735  C          CMP     DH,0A0H                          ; 1ST 16KB OK
06C2  74 06                  1736  C          jz      hb_error_exit
06C4                         1737  C  RESUME:
06C4  01 5E 04               1738  C          ADD     WORD PTR[BP][4],BX              ; SAVE BUFFER FOUND
06C7  B8 0000                1739  C          MOV     AX,0
06CA                         1740  C  HB_ERROR_EXIT:
06CA  C3                     1741  C          RET
06CB                         1742  C  HOW_BIG          ENDP
                             1743  C
                             1744  C  ;--------------------------------------
                             1745  C  ;SUBROUTINES FOR POWER ON DIAGNOSTICS  :
                             1746  C  ;-------------------------------------------------------------------
                             1747  C  ;THIS PROCEDURE WILL ISSUE ONE LONG TONE (3 SEC) AND ONE OR      :
                             1748  C  ;MORE SHORT TONES (1 SEC) TO INDICATE A FAILURE ON THE PLANAR    :
                             1749  C  ;BOARD ,A BAD RAM MODULE,OR A PROBLEM WITH THE CRT.              :
                             1750  C  ;ENTRY REQUIREMENTS:                                             :
                             1751  C  ;       DH=NUMBER OF LONG TONES TO BEEP                         :
                             1752  C  ;       DL=NUMBER OF SHORT TONES TO BEEP.                       :
                             1753  C  ;-------------------------------------------------------------------
06CB                         1754  C  ERR_BEEP   PROC    NEAR
06CB  9C                     1755  C          PUSHF                                   ; SAVE FLAGS
06CC  FA                     1756  C          CLI                                     ; DISABLE SYSTEM INTS
06CD  1E                     1757  C          PUSH    DS
06CE  E8 0D01 R              1758  C          CALL    DDS
                             1759  C          ASSUME  DS:ABS0
06D1  0A F6                  1760  C          OR      DH,DH                           ; ANY LONG TONES TO BEEP
06D3  74 0B                  1761  C          JZ      G3                              ; NO, DO THE SHORT ONES
06D5                         1762  C  G1:                                             ; LONG BEEP
06D5  B3 06                  1763  C          MOV     BL,6                            ; COUNTER FOR BEEPS
06D7  E8 0D23 R              1764  C          CALL    BEEP                            ; DO THE BEEP
```

```
06DA                                    1765     C  G2:
06DA  E2 FE                             1766     C          LOOP    G2                      ; DELAY BETWEEN BEEPS
06DC  FE CE                             1767     C          DEC     DH                      ; ANY MORE TO DO
06DE  75 F5                             1768     C          JNZ     G1                      ; DO IT
06E0                                    1769     C  G3:
06E0  B3 01                             1770     C          MOV     BL,1                    ; COUNTER FOR A SHORT BEEP
06E2  E8 0D23 R                         1771     C          CALL    BEEP                    ; DO IT
06E5                                    1772     C  G4:
06E5  E2 FE                             1773     C          LOOP    G4                      ; DELAY BETWEEN BEEPS
06E7  FE CA                             1774     C          DEC     DL                      ; DONE WITH SHORT BEEPS
06E9  75 F5                             1775     C          JNZ     G3                      ; DO MORE
06EB                                    1776     C  G5:
06EB  E2 FE                             1777     C          LOOP G5                         ; DELAY BEFORE RETURN
06ED                                    1778     C  G6:
06ED  E2 FE                             1779     C          LOOP G6
06EF  1F                                1780     C          POP     DS                      ; RESTORE CONTENTS OF DS
06F0  9D                                1781     C          POPF                            ; RESTORE FLAGS
06F1  C3                                1782     C          RET
06F2                                    1783     C  ERR_BEEP        ENDP
                                        1784     C
                                        1785     C          SUBTTL
                                        1786
                                        1787
06F2                                    1788          T2    LABEL   WORD
06F2  0EB5 R                            1789          DW    OFFSET  AH0                     ; MODE SET
06F4  10EC R                            1790          DW    OFFSET  AH1                     ; SET CURSOR TYPE
06F6  1154 R                            1791          DW    OFFSET  AH2                     ; SET CURSOR POSITION
06F8  1183 R                            1792          DW    OFFSET  AH3                     ; READ CURSOR POSITION
06FA  119A R                            1793          DW    OFFSET  AH4                     ; READ LIGHT PEN POSITION
06FC  12A1 R                            1794          DW    OFFSET  AH5                     ; ACTIVE DISPLAY PAGE
06FE  150B R                            1795          DW    OFFSET  AH6                     ; SCROLL DOWN
0700  15AD R                            1796          DW    OFFSET  AH7                     ; SCROLL UP
0702  17CF R                            1797          DW    OFFSET  AH8                     ; READ CHAR/ATTRIBUTE
0704  1896 R                            1798          DW    OFFSET  AH9                     ; WRITE CHAR/ATTRIBUTE
0706  18DA R                            1799          DW    OFFSET  AHA                     ; WRITE CHARACTER ONLY
0708  1A72 R                            1800          DW    OFFSET  AHB                     ; SET COLOR PALETTE
070A  1BC8 R                            1801          DW    OFFSET  AHC                     ; WRITE DOT
070C  1C9C R                            1802          DW    OFFSET  AHD                     ; READ DOT
070E  1CFE R                            1803          DW    OFFSET  AHE                     ; WRITE TTY
0710  1D82 R                            1804          DW    OFFSET  AHF                     ; CURRENT VIDEO STATE
0712  1DC2 R                            1805          DW    OFFSET  AH10                    ; SET PALETTE REGISTERS
0714  1F95 R                            1806          DW    OFFSET  AH11                    ; CHAR GENERATOR ROUTINE
0716  20BC R                            1807          DW    OFFSET  AH12                    ; ALTERNATE SELECT
0718  2115 R                            1808          DW    OFFSET  AH13                    ; WRITE STRING
= 0028                                  1809          T2L   EQU     $-T2
                                        1810
                                        1811     C          INCLUDE         VPARMS.INC
                                        1812     C          SUBTTL  VPARMS.INC
                                        1813     C          PAGE
071A                                    1814     C  VIDEO_PARMS     LABEL   BYTE
                                        1815     C
                                        1816     C  ; Structure of this table
                                        1817     C  ;
                                        1818     C  ;       columns, rows, pels per character
                                        1819     C  ;       page length
                                        1820     C  ;       sequencer parameters
                                        1821     C  ;       miscellaneous register
                                        1822     C  ;       CRTC parameters
                                        1823     C  ;       Attribute parameters
                                        1824     C  ;       Graphics parameters
                                        1825     C
= 0000                                  1826     C  base_1  equ     $ - video_parms
071A                                    1827     C  base_1_l        label   byte
                                        1828     C
                                        1829     C  ;----- default modes
                                        1830     C
                                        1831     C  ;--0--
071A  28 18 08                          1832     C          db      40d,24d,08d
071D  0800                              1833     C          dw      00800h
                                        1834     C
= 0005                                  1835     C  tfs_len equ     $ - base_1_l
                                        1836     C
071F                                    1837     C  SEQ_PARMS       LABEL   BYTE
071F  0B 03 00 03                       1838     C          DB      00BH,003H,000H,003H
= 0004                                  1839     C  M1      EQU     $ - SEQ_PARMS
                                        1840     C
0723  23                                1841     C          DB      023H
                                        1842     C
0724                                    1843     C  CRT_PARMS       LABEL   BYTE
0724  37 27 2D 37 31 15                 1844     C          DB      037H,027H,02DH,037H,031H,015H
072A  04 11 00 07 06 07                 1845     C          DB      004H,011H,000H,007H,006H,007H
0730  00 00 00 00 E1 24                 1846     C          DB      000H,000H,000H,000H,0E1H,024H
0736  C7 14 08 E0 F0 A3                 1847     C          DB      0C7H,014H,008H,0E0H,0F0H,0A3H
073C  FF                                1848     C          DB      0FFH
= 0019                                  1849     C  M4      EQU     $-CRT_PARMS
                                        1850     C
073D                                    1851     C  ATTR_PARMS      LABEL   BYTE
073D  00 01 02 03 04 05                 1852     C          DB      000H,001H,002H,003H,004H,005H
0743  06 07 10 11 12 13                 1853     C          DB      006H,007H,010H,011H,012H,013H
0749  14 15 16 17 08 00                 1854     C          DB      014H,015H,016H,017H,008H,000H
074F  0F 00                             1855     C          DB      00FH,000H
= 0014                                  1856     C  M5      EQU     $-ATTR_PARMS
                                        1857     C
= 0037                                  1858     C  in_2    equ     $ - base_1_l
0751                                    1859     C  GRAPH_PARMS     LABEL   BYTE
0751  00 00 00 00 00 10                 1860     C          DB      000H,000H,000H,000H,000H,010H
0757  0E 00 FF                         1861     C          DB      00EH,000H,0FFH
= 0009                                  1862     C  M6      EQU     $-GRAPH_PARMS
                                        1863     C
= 0040                                  1864     C  m_tbl_len       equ     $ - base_1_l
                                        1865     C
                                        1866     C  ;--1--
075A  28 18 08                          1867     C          db      40d,24d,08d
075D  0800                              1868     C          dw      00800h
                                        1869     C
075F  0B 03 00 03                       1870     C          DB      00BH,003H,000H,003H
                                        1871     C
0763  23                                1872     C          DB      023H
                                        1873     C
0764  37 27 2D 37 31 15                 1874     C          DB      037H,027H,02DH,037H,031H,015H
076A  04 11 00 07 06 07                 1875     C          DB      004H,011H,000H,007H,006H,007H
0770  00 00 00 00 E1 24                 1876     C          DB      000H,000H,000H,000H,0E1H,024H
0776  C7 14 08 E0 F0 A3                 1877     C          DB      0C7H,014H,008H,0E0H,0F0H,0A3H
077C  FF                                1878     C          DB      0FFH
                                        1879     C
077D  00 01 02 03 04 05                 1880     C          DB      000H,001H,002H,003H,004H,005H
0783  06 07 10 11 12 13                 1881     C          DB      006H,007H,010H,011H,012H,013H
0789  14 15 16 17 08 00                 1882     C          DB      014H,015H,016H,017H,008H,000H
078F  0F 00                             1883     C          DB      00FH,000H
                                        1884     C
0791  00 00 00 00 00 10                 1885     C          DB      000H,000H,000H,000H,000H,010H
0797  0E 00 FF                         1886     C          DB      00EH,000H,0FFH
                                        1887     C
                                        1888     C  ;--2--
079A  50 18 08                          1889     C          db      80d,24d,08d
079D  1000                              1890     C          dw      01000h
```

```
                                                1891    C
079F  01 03 00 00 03                            1892    C               DB      001H,003H,000H,003H
                                                1893    C
07A3  23                                        1894    C               DB      023H
                                                1895    C
07A4  70 4F 5C 2F 5F 07                         1896    C               DB      070H,04FH,05CH,02FH,05FH,007H
07AA  04 11 00 07 06 07                         1897    C               DB      004H,011H,000H,007H,006H,007H
07B0  00 00 00 00 E1 24                         1898    C               DB      000H,000H,000H,000H,0E1H,024H
07B6  C7 28 08 E0 F0 A3                         1899    C               DB      0C7H,028H,008H,0E0H,0F0H,0A3H
07BC  FF                                        1900    C               DB      0FFH
                                                1901    C
07BD  00 01 02 03 04 05                         1902    C               DB      000H,001H,002H,003H,004H,005H
07C3  06 07 10 11 12 13                         1903    C               DB      006H,007H,010H,011H,012H,013H
07C9  14 15 16 17 08 00                         1904    C               DB      014H,015H,016H,017H,008H,000H
07CF  0F 00                                     1905    C               DB      00FH,000H
                                                1906    C
07D1  00 00 00 00 00 10                         1907    C               DB      000H,000H,000H,000H,000H,010H
07D7  0E 00 FF                                  1908    C               DB      00EH,000H,0FFH
                                                1909    C
                                                1910    C       ;--3--
07DA  50 18 08                                  1911    C               db      80d,24d,08d
07DD  1000                                      1912    C               dw      01000h
                                                1913    C
07DF  01 03 00 03                               1914    C               DB      001H,003H,000H,003H
                                                1915    C
07E3  23                                        1916    C               DB      023H
                                                1917    C
07E4  70 4F 5C 2F 5F 07                         1918    C               DB      070H,04FH,05CH,02FH,05FH,007H
07EA  04 11 00 07 06 07                         1919    C               DB      004H,011H,000H,007H,006H,007H
07F0  00 00 00 00 E1 24                         1920    C               DB      000H,000H,000H,000H,0E1H,024H
07F6  C7 28 08 E0 F0 A3                         1921    C               DB      0C7H,028H,008H,0E0H,0F0H,0A3H
07FC  FF                                        1922    C               DB      0FFH
                                                1923    C
07FD  00 01 02 03 04 05                         1924    C               DB      000H,001H,002H,003H,004H,005H
0803  06 07 10 11 12 13                         1925    C               DB      006H,007H,010H,011H,012H,013H
0809  14 15 16 17 08 00                         1926    C               DB      014H,015H,016H,017H,008H,000H
080F  0F 00                                     1927    C               DB      00FH,000H
                                                1928    C
0811  00 00 00 00 00 10                         1929    C               DB      000H,000H,000H,000H,000H,010H
0817  0E 00 FF                                  1930    C               DB      00EH,000H,0FFH
                                                1931    C
                                                1932    C       ;--4--
081A  28 18 08                                  1933    C               db      40d,24d,08d
081D  4000                                      1934    C               dw      04000h
                                                1935    C
081F  0B 03 00 02                               1936    C               DB      00BH,003H,000H,002H
                                                1937    C
0823  23                                        1938    C               DB      023H
                                                1939    C
0824  37 27 2D 37 30 14                         1940    C               DB      037H,027H,02DH,037H,030H,014H
082A  04 11 00 01 00 00                         1941    C               DB      004H,011H,000H,001H,000H,000H
0830  00 00 00 00 E1 24                         1942    C               DB      000H,000H,000H,000H,0E1H,024H
0836  C7 14 00 E0 F0 A2                         1943    C               DB      0C7H,014H,000H,0E0H,0F0H,0A2H
083C  FF                                        1944    C               DB      0FFH
                                                1945    C
083D  00 13 15 17 02 04                         1946    C               DB      000H,013H,015H,017H,002H,004H
0843  06 07 10 11 12 13                         1947    C               DB      006H,007H,010H,011H,012H,013H
0849  14 15 16 17 01 00                         1948    C               DB      014H,015H,016H,017H,001H,000H
084F  03 00                                     1949    C               DB      003H,000H
                                                1950    C
0851  00 00 00 00 00 30                         1951    C               DB      000H,000H,000H,000H,000H,030H
0857  0F 00 FF                                  1952    C               DB      00FH,000H,0FFH
                                                1953    C
                                                1954    C       ;--5--
085A  28 18 08                                  1955    C               db      40d,24d,08d
085D  4000                                      1956    C               dw      04000h
                                                1957    C
085F  0B 03 00 02                               1958    C               DB      00BH,003H,000H,002H
                                                1959    C
0863  23                                        1960    C               DB      023H
                                                1961    C
0864  37 27 2D 37 30 14                         1962    C               DB      037H,027H,02DH,037H,030H,014H
086A  04 11 00 01 00 00                         1963    C               DB      004H,011H,000H,001H,000H,000H
0870  00 00 00 00 E1 24                         1964    C               DB      000H,000H,000H,000H,0E1H,024H
0876  C7 14 00 E0 F0 A2                         1965    C               DB      0C7H,014H,000H,0E0H,0F0H,0A2H
087C  FF                                        1966    C               DB      0FFH
                                                1967    C
087D  00 13 15 17 02 04                         1968    C               DB      000H,013H,015H,017H,002H,004H
0883  06 07 10 11 12 13                         1969    C               DB      006H,007H,010H,011H,012H,013H
0889  14 15 16 17 01 00                         1970    C               DB      014H,015H,016H,017H,001H,000H
088F  03 00                                     1971    C               DB      003H,000H
                                                1972    C
0891  00 00 00 00 00 30                         1973    C               DB      000H,000H,000H,000H,000H,030H
0897  0F 00 FF                                  1974    C               DB      00FH,000H,0FFH
                                                1975    C
                                                1976    C       ;--6--
089A  50 18 08                                  1977    C               db      80d,24d,08d
089D  4000                                      1978    C               dw      04000h
                                                1979    C
089F  01 01 00 06                               1980    C               DB      001H,001H,000H,006H
                                                1981    C
08A3  23                                        1982    C               DB      023H
                                                1983    C
08A4  70 4F 59 2D 5E 06                         1984    C               DB      070H,04FH,059H,02DH,05EH,006H
08AA  04 11 00 01 00 00                         1985    C               DB      004H,011H,000H,001H,000H,000H
08B0  00 00 00 00 E0 23                         1986    C               DB      000H,000H,000H,000H,0E0H,023H
08B6  C7 28 00 DF EF C2                         1987    C               DB      0C7H,028H,000H,0DFH,0EFH,0C2H
08BC  FF                                        1988    C               DB      0FFH
                                                1989    C
08BD  00 17 17 17 17 17                         1990    C               DB      000H,017H,017H,017H,017H,017H
08C3  17 17 17 17 17 17                         1991    C               DB      017H,017H,017H,017H,017H,017H
08C9  17 17 17 17 01 00                         1992    C               DB      017H,017H,017H,017H,001H,000H
08CF  01 00                                     1993    C               DB      001H,000H
                                                1994    C
08D1  00 00 00 00 00 00                         1995    C               DB      000H,000H,000H,000H,000H,000H
08D7  0D 00 FF                                  1996    C               DB      00DH,000H,0FFH
                                                1997    C
                                                1998    C       ;--7--
08DA  50 18 0E                                  1999    C               db      80d,24d,14d
08DD  1000                                      2000    C               dw      01000h
                                                2001    C
08DF  00 03 00 03                               2002    C               DB      000H,003H,000H,003H
                                                2003    C
08E3  A6                                        2004    C               DB      0a6H
                                                2005    C
08E4  60 4F 56 3A 51 60                         2006    C               DB      060H,04FH,056H,03AH,051H,060H
08EA  70 1F 00 0D 0B 0C                         2007    C               DB      070H,01FH,000H,00DH,00BH,00CH
08F0  00 00 00 00 5E 2E                         2008    C               DB      000H,000H,000H,000H,05EH,02EH
08F6  5D 28 0D 5E 6E A3                         2009    C               DB      05DH,028H,00DH,05EH,06EH,0A3H
08FC  FF                                        2010    C               DB      0FFH
                                                2011    C
08FD  00 08 08 08 08 08                         2012    C               DB      000H,008H,008H,008H,008H,008H
0903  08 08 10 18 18 18                         2013    C               DB      008H,008H,010H,018H,018H,018H
0909  18 18 18 18 0E 00                         2014    C               DB      018H,018H,018H,018H,00EH,000H
090F  0F 08                                     2015    C               DB      00FH,008H
                                                2016    C
```

```
0911  00 00 00 00 00 00 10          2017     C          DB      000H,000H,000H,000H,000H,010H
0917  0A 00 FF                      2018     C          DB      00AH,000H,0FFH
                                    2019     C
                                    2020     C  ;--8--
091A  28 18 08                      2021     C          db      40d,24d,08d
091D  4000                          2022     C          dw      04000h
                                    2023     C
091F  00 00 00 03                   2024     C          DB      000H,000H,000H,003H
                                    2025     C
0923  23                            2026     C          DB      023H
                                    2027     C
0924  37 27 2D 37 31 15            2028     C          DB      037H,027H,02DH,037H,031H,015H
092A  04 11 00 07 06 07            2029     C          DB      004H,011H,000H,007H,006H,007H
0930  00 00 00 00 E1 24            2030     C          DB      000H,000H,000H,000H,0E1H,024H
0936  C7 14 08 E0 F0 A3            2031     C          DB      0C7H,014H,008H,0E0H,0F0H,0A3H
093C  FF                            2032     C          DB      0FFH
                                    2033     C
093D  00 01 02 03 04 05            2034     C          DB      000H,001H,002H,003H,004H,005H
0943  06 07 10 11 12 13            2035     C          DB      006H,007H,010H,011H,012H,013H
0949  14 15 16 17 08 00            2036     C          DB      014H,015H,016H,017H,008H,000H
094F  0F 00                         2037     C          DB      00FH,000H
                                    2038     C
0951  00 00 00 00 00 00 10          2039     C          DB      000H,000H,000H,000H,000H,010H
0957  0E 00 FF                      2040     C          DB      00EH,000H,0FFH
                                    2041     C
                                    2042     C  ;--9--
095A  28 18 08                      2043     C          db      40d,24d,08d
095D  4000                          2044     C          dw      04000h
                                    2045     C
095F  00 00 00 03                   2046     C          DB      000H,000H,000H,003H
                                    2047     C
0963  23                            2048     C          DB      023H
                                    2049     C
0964  37 27 2D 37 31 15            2050     C          DB      037H,027H,02DH,037H,031H,015H
096A  04 11 00 07 06 07            2051     C          DB      004H,011H,000H,007H,006H,007H
0970  00 00 00 00 E1 24            2052     C          DB      000H,000H,000H,000H,0E1H,024H
0976  C7 14 08 E0 F0 A3            2053     C          DB      0C7H,014H,008H,0E0H,0F0H,0A3H
097C  FF                            2054     C          DB      0FFH
                                    2055     C
097D  00 01 02 03 04 05            2056     C          DB      000H,001H,002H,003H,004H,005H
0983  06 07 10 11 12 13            2057     C          DB      006H,007H,010H,011H,012H,013H
0989  14 15 16 17 08 00            2058     C          DB      014H,015H,016H,017H,008H,000H
098F  0F 00                         2059     C          DB      00FH,000H
                                    2060     C
0991  00 00 00 00 00 10             2061     C          DB      000H,000H,000H,000H,000H,010H
0997  0E 00 FF                      2062     C          DB      00EH,000H,0FFH
                                    2063     C
                                    2064     C  ;--a--
099A  28 18 08                      2065     C          db      40d,24d,08d
099D  4000                          2066     C          dw      04000h
                                    2067     C
099F  00 00 00 03                   2068     C          DB      000H,000H,000H,003H
                                    2069     C
09A3  23                            2070     C          DB      023H
                                    2071     C
09A4  37 27 2D 37 31 15            2072     C          DB      037H,027H,02DH,037H,031H,015H
09AA  04 11 00 07 06 07            2073     C          DB      004H,011H,000H,007H,006H,007H
09B0  00 00 00 00 E1 24            2074     C          DB      000H,000H,000H,000H,0E1H,024H
09B6  C7 14 08 E0 F0 A3            2075     C          DB      0C7H,014H,008H,0E0H,0F0H,0A3H
09BC  FF                            2076     C          DB      0FFH
                                    2077     C
09BD  00 01 02 03 04 05            2078     C          DB      000H,001H,002H,003H,004H,005H
09C3  06 07 10 11 12 13            2079     C          DB      006H,007H,010H,011H,012H,013H
09C9  14 15 16 17 08 00            2080     C          DB      014H,015H,016H,017H,008H,000H
09CF  0F 00                         2081     C          DB      00FH,000H
                                    2082     C
09D1  00 00 00 00 00 10             2083     C          DB      000H,000H,000H,000H,000H,010H
09D7  0E 00 FF                      2084     C          DB      00EH,000H,0FFH
                                    2085     C
                                    2086     C  ;--b--
09DA  50 18 08                      2087     C          db      80d,24d,08d
09DD  1000                          2088     C          dw      01000h
                                    2089     C
09DF  01 04 00 07                   2090     C          DB      001H,004H,000H,007H
                                    2091     C
09E3  23                            2092     C          DB      023H
                                    2093     C
09E4  70 4F 5C 2F 5F 07            2094     C          DB      070H,04FH,05CH,02FH,05FH,007H
09EA  04 11 00 07 06 07            2095     C          DB      004H,011H,000H,007H,006H,007H
09F0  00 00 00 00 E1 24            2096     C          DB      000H,000H,000H,000H,0E1H,024H
09F6  C7 28 08 E0 F0 A3            2097     C          DB      0C7H,028H,008H,0E0H,0F0H,0A3H
09FC  FF                            2098     C          DB      0FFH
                                    2099     C
09FD  00 00 00 00 00 00            2100     C          DB      000H,000H,000H,000H,000H,000H
0A03  00 00 00 00 00 00            2101     C          DB      000H,000H,000H,000H,000H,000H
0A09  00 00 00 00 00 00            2102     C          DB      000H,000H,000H,000H,000H,000H
0A0F  0F 00                         2103     C          DB      00FH,000H
                                    2104     C
0A11  00 00 00 00 00 00            2105     C          DB      000H,000H,000H,000H,000H,000H
0A17  04 00 FF                      2106     C          DB      004H,000H,0FFH
                                    2107     C  ;--c--
0A1A  50 18 0E                      2108     C          db      80d,24d,14d
0A1D  1000                          2109     C          dw      01000h
                                    2110     C
0A1F  00 04 00 07                   2111     C          DB      000H,004H,000H,007H
                                    2112     C
0A23  A6                            2113     C          DB      0a6H
                                    2114     C
0A24  60 4F 56 3A 51 60            2115     C          DB      060H,04FH,056H,03AH,051H,060H
0A2A  70 1F 00 0D 0B 0C            2116     C          DB      070H,01FH,000H,00DH,00bH,00cH
0A30  00 00 00 00 5E 2E            2117     C          DB      000H,000H,000H,000H,05EH,02EH
0A36  5D 28 0D 5E 6E A3            2118     C          DB      05DH,028H,00DH,05EH,06EH,0A3H
0A3C  FF                            2119     C          DB      0FFH
                                    2120     C
0A3D  00 00 00 00 00 00            2121     C          DB      000H,000H,000H,000H,000H,000H
0A43  00 00 00 00 00 00            2122     C          DB      000H,000H,000H,000H,000H,000H
0A49  00 00 00 00 0E 00            2123     C          DB      000H,000H,000H,000H,00EH,000H
0A4F  0F 08                         2124     C          DB      00FH,008H
                                    2125     C
0A51  00 00 00 00 00 00            2126     C          DB      000H,000H,000H,000H,000H,000H
0A57  04 00 FF                      2127     C          DB      004H,000H,0FFH
                                    2128     C  ;--d--
0A5A  28 18 08                      2129     C          db      40d,24d,08d
0A5D  2000                          2130     C          dw      02000h
                                    2131     C
0A5F  0B 0F 00 06                   2132     C          DB      00BH,00FH,000H,006H
                                    2133     C
0A63  23                            2134     C          DB      023H
                                    2135     C
0A64  37 27 2D 37 30 14            2136     C          DB      037H,027H,02DH,037H,030H,014H
0A6A  04 11 00 00 00 00            2137     C          DB      004H,011H,000H,000H,000H,000H
0A70  00 00 00 00 E1 24            2138     C          DB      000H,000H,000H,000H,0E1H,024H
0A76  C7 14 00 E0 F0 E3            2139     C          DB      0C7H,014H,000H,0E0H,0F0H,0E3H
0A7C  FF                            2140     C          DB      0FFH
                                    2141     C
0A7D  00 01 02 03 04 05            2142     C          DB      000H,001H,002H,003H,004H,005H
```

```
0A83  06 07 10 11 12 13
0A89  14 15 16 17 01 00
0A8F  0F 00

0A91  00 00 00 00 00 00
0A97  05 0F FF

0A9A  50 18 08
0A9D  4000

0A9F  01 0F 00 06

0AA3  23

0AA4  70 4F 59 2D 5E 06
0AAA  04 11 00 00 00 00
0AB0  00 00 00 00 E0 23
0AB6  C7 28 00 DF EF E3
0ABC  FF

0ABD  00 01 02 03 04 05
0AC3  06 07 10 11 12 13
0AC9  14 15 16 17 01 00
0ACF  0F 00

0AD1  00 00 00 00 00 00
0AD7  05 0F FF

0ADA  50 18 0E
0ADD  8000

0ADF  05 0F 00 00

0AE3  A2

0AE4  60 4F 56 1A 50 E0
0AEA  70 1F 00 00 00 00
0AF0  00 00 00 00 5E 2E
0AF6  5D 14 00 5E 6E 8B
0AFC  FF

0AFD  00 08 00 00 18 18
0B03  00 00 00 08 00 00
0B09  00 18 00 00 0B 00
0B0F  05 00

0B11  00 00 00 00 00 10
0B17  07 0F FF

0B1A  50 18 0E
0B1D  8000

0B1F  05 0F 00 00

0B23  A7

0B24  5B 4F 53 17 50 BA
0B2A  6C 1F 00 00 00 00
0B30  00 00 00 00 5E 2B
0B36  5D 14 0F 5F 0A 8B
0B3C  FF

0B3D  00 01 00 00 04 07
0B43  00 00 00 01 00 00
0B49  04 07 00 00 01 00
0B4F  05 00

0B51  00 00 00 00 00 10
0B57  07 0F FF

= 0440

085A  50 18 0E
085D  8000

0B5F  01 0F 00 06

0B63  A2

0B64  60 4F 56 3A 50 60
086A  70 1F 00 00 00 00
0B70  00 00 00 00 5E 2E
0B76  5D 28 0D 5E 6E E3
0B7C  FF

0B7D  00 08 00 00 18 18
0B83  00 00 00 08 00 00
0B89  00 18 00 00 0B 00
0B8F  05 00

0B91  00 00 00 00 00 00
0B97  05 0F FF

0B9A  50 18 0E
0B9D  8000

0B9F  01 0F 00 06

0BA3  A7

0BA4  5B 4F 53 37 52 00
0BAA  6C 1F 00 00 00 00
0BB0  00 00 00 00 5E 2B
0BB6  5D 28 0F 5F 0A E3
0BBC  FF

0BBD  00 01 02 03 04 05
0BC3  14 07 38 39 3A 3B
0BC9  3C 3D 3E 3F 01 00
0BCF  0F 00

0BD1  00 00 00 00 00 00
0BD7  05 0F FF

= 04C0

0BDA  28 18 0E
0BDD  0800

2143  C          DB      006H,007H,010H,011H,012H,013H
2144  C          DB      014H,015H,016H,017H,001H,000H
2145  C          DB      00FH,000H
2146  C
2147  C          DB      000H,000H,000H,000H,000H,000H
2148  C          DB      005H,00FH,0FFH
2149  C
2150  C          db      80d,24d,08d
2151  C          dw      04000h
2152  C
2153  C          DB      001H,00FH,000H,006H
2154  C
2155  C          DB      023H
2156  C
2157  C          DB      070H,04FH,059H,02DH,05EH,006H
2158  C          DB      004H,011H,000H,000H,000H,000H
2159  C          DB      000H,000H,000H,0E0H,023H
2160  C          DB      0C7H,028H,000H,0DFH,0EFH,0E3H
2161  C          DB      0FFH
2162  C
2163  C          DB      000H,001H,002H,003H,004H,005H
2164  C          DB      006H,007H,010H,011H,012H,013H
2165  C          DB      014H,015H,016H,017H,001H,000H
2166  C          DB      00FH,000H
2167  C
2168  C          DB      000H,000H,000H,000H,000H,000H
2169  C          DB      005H,00FH,0FFH
2170  C  ;--f--
2171  C          db      80d,24d,14d
2172  C          dw      08000h
2173  C
2174  C          DB      005H,00FH,000H,000H
2175  C
2176  C          DB      0a2H
2177  C
2178  C          DB      060H,04FH,056H,01AH,050H,0E0H
2179  C          DB      070H,01FH,000H,000H,000H,000H
2180  C          DB      000H,000H,000H,000H,05EH,02EH
2181  C          DB      05DH,014H,00DH,05EH,06EH,08BH
2182  C          DB      0FFH
2183  C
2184  C          DB      000H,008H,000H,000H,018H,018H
2185  C          DB      000H,000H,000H,008H,000H,000H
2186  C          DB      000H,018H,000H,000H,00BH,000H
2187  C          DB      005H,000H
2188  C
2189  C          DB      000H,000H,000H,000H,000H,010H
2190  C          DB      007H,00FH,0FFH
2191  C  ;-10--
2192  C          db      80d,24d,14d
2193  C          dw      08000h
2194  C
2195  C          DB      005H,00FH,000H,000H
2196  C
2197  C          DB      0a7H
2198  C
2199  C          DB      05BH,04FH,053H,017H,050h,0baH
2200  C          DB      06CH,01FH,000H,000H,000H,000H
2201  C          DB      000H,000H,000H,000H,05EH,02BH
2202  C          DB      05DH,014H,00FH,05FH,00AH,08BH
2203  C          DB      0FFH
2204  C
2205  C          DB      000H,001h,000H,000H,004h,007h
2206  C          DB      000H,000H,000H,001h,000H,000H
2207  C          DB      004h,007h,000H,000H,001H,000H
2208  C          DB      005H,000H
2209  C
2210  C          DB      000H,000H,000H,000H,000H,010H
2211  C          DB      007H,00FH,0FFH
2212  C
2213  C  base_2  equ     $ - video_parms
2214  C
2215  C  ;-----   > 16K mode values
2216  C
2217  C  ;--f--
2218  C          db      80d,24d,14d
2219  C          dw      08000h
2220  C
2221  C          DB      001H,00FH,000H,006H
2222  C
2223  C          DB      0a2H
2224  C
2225  C          DB      060H,04FH,056H,03AH,050H,060H
2226  C          DB      070H,01FH,000H,000H,000H,000H
2227  C          DB      000H,000H,000H,000H,05EH,02EH
2228  C          DB      05DH,028H,00DH,05EH,06EH,0E3H
2229  C          DB      0FFH
2230  C
2231  C          DB      000H,008H,000H,000H,018H,018H
2232  C          DB      000H,000H,000H,008H,000H,000H
2233  C          DB      000H,018H,000H,000H,00BH,000H
2234  C          DB      005H,000H
2235  C
2236  C          DB      000H,000H,000H,000H,000H,000H
2237  C          DB      005H,00FH,0FFH
2238  C
2239  C  ;-10--
2240  C          db      80d,24d,14d
2241  C          dw      08000h
2242  C
2243  C          DB      001H,00FH,000H,006H
2244  C
2245  C          DB      0a7H
2246  C
2247  C          DB      05BH,04FH,053H,037H,052H,000H
2248  C          DB      06CH,01FH,000H,000H,000H,000H
2249  C          DB      000H,000H,000H,000H,05EH,02BH
2250  C          DB      05DH,028H,00FH,05FH,00AH,0E3H
2251  C          DB      0FFH
2252  C
2253  C          DB      000H,001H,002H,003H,004H,005H
2254  C          DB      014H,007H,038H,039H,03aH,03bH
2255  C          DB      03cH,03dH,03eH,03FH,001H,000H
2256  C          DB      00FH,000H
2257  C
2258  C          DB      000H,000H,000H,000H,000H,000H
2259  C          DB      005H,00FH,0FFH
2260  C
2261  C
2262  C  base_3  equ     $ - video_parms
2263  C
2264  C  ;----- hi res alternate values
2265  C
2266  C  ;--0--
2267  C          db      40d,24d,14d
2268  C          dw      00800h
```

```
                                                2269    C
0BDF  0B 03 00 03                               2270    C               DB      00BH,003H,000H,003H
                                                2271    C
0BE3  A7                                        2272    C               DB      0a7H
                                                2273    C
0BE4  2D 27 2B 2D 28 6D                         2274    C               DB      02dH,027H,02bH,02dH,028h,06dh
0BEA  6C 1F 00 0D 06 07                         2275    C               DB      06cH,01fH,000H,00dH,006H,007H
0BF0  00 00 00 00 5E 2B                         2276    C               DB      000H,000H,000H,000H,05eH,02bH
0BF6  5D 14 0F 5E 0A A3                         2277    C               DB      05dH,014H,00FH,05eH,00aH,0A3H
0BFC  FF                                        2278    C               DB      0FFH
                                                2279    C
0BFD  00 01 02 03 04 05                         2280    C               DB      000H,001H,002H,003H,004H,005H
0C03  14 07 3B 39 3A 3B                         2281    C               DB      014H,007H,038H,039H,03aH,03bH
0C09  3C 3D 3E 3F 08 00                         2282    C               DB      03cH,03dH,03eH,03fH,008H,000H
0C0F  0F 00                                     2283    C               DB      00FH,000H
                                                2284    C
0C11  00 00 00 00 00 10                         2285    C               DB      000H,000H,000H,000H,000H,010H
0C17  0E 00 FF                                  2286    C               DB      00EH,000H,0FFH
                                                2287    C
                                                2288    C       ;--1--
0C1A  28 18 0E                                  2289    C               db      40d,24d,14d
0C1D  0800                                      2290    C               dw      00800h
                                                2291    C
0C1F  0B 03 00 03                               2292    C               DB      00BH,003H,000H,003H
                                                2293    C
0C23  A7                                        2294    C               DB      0a7H
                                                2295    C
0C24  2D 27 2B 2D 28 6D                         2296    C               DB      02dH,027H,02bH,02dH,028h,06dh
0C2A  6C 1F 00 0D 06 07                         2297    C               DB      06cH,01fH,000H,00dH,006H,007H
0C30  00 00 00 00 5E 2B                         2298    C               DB      000H,000H,000H,000H,05eH,02bH
0C36  5D 14 0F 5E 0A A3                         2299    C               DB      05dH,014H,00FH,05eH,00aH,0A3H
0C3C  FF                                        2300    C               DB      0FFH
                                                2301    C
0C3D  00 01 02 03 04 05                         2302    C               DB      000H,001H,002H,003H,004H,005H
0C43  14 07 38 39 3A 3B                         2303    C               DB      014H,007H,038H,039H,03aH,03bH
0C49  3C 3D 3E 3F 08 00                         2304    C               DB      03cH,03dH,03eH,03fH,008H,000H
0C4F  0F 00                                     2305    C               DB      00FH,000H
                                                2306    C
0C51  00 00 00 00 00 10                         2307    C               DB      000H,000H,000H,000H,000H,010H
0C57  0E 00 FF                                  2308    C               DB      00EH,000H,0FFH
                                                2309    C
                                                2310    C       ;--2--
0C5A  50 18 0E                                  2311    C               db      80d,24d,14d
0C5D  1000                                      2312    C               dw      01000h
                                                2313    C
0C5F  01 03 00 03                               2314    C               DB      001H,003H,000H,003H
                                                2315    C
0C63  A7                                        2316    C               DB      0a7H
                                                2317    C
0C64  5B 4F 53 37 51 5B                         2318    C               DB      05bH,04FH,053H,037H,051h,05bh
0C6A  6C 1F 00 0D 06 07                         2319    C               DB      06cH,01fH,000H,00dH,006H,007H
0C70  00 00 00 00 5E 2B                         2320    C               DB      000H,000H,000H,000H,05eH,02bH
0C76  5D 28 0F 5E 0A A3                         2321    C               DB      05dH,028H,00fH,05eH,00aH,0A3H
0C7C  FF                                        2322    C               DB      0FFH
                                                2323    C
0C7D  00 01 02 03 04 05                         2324    C               DB      000H,001H,002H,003H,004H,005H
0C83  14 07 38 39 3A 3B                         2325    C               DB      014H,007H,038H,039H,03aH,03bH
0C89  3C 3D 3E 3F 08 00                         2326    C               DB      03cH,03dH,03eH,03fH,008H,000H
0C8F  0F 00                                     2327    C               DB      00FH,000H
                                                2328    C
0C91  00 00 00 00 00 10                         2329    C               DB      000H,000H,000H,000H,000H,010H
0C97  0E 00 FF                                  2330    C               DB      00EH,000H,0FFH
                                                2331    C
                                                2332    C       ;--3--
0C9A  50 18 0E                                  2333    C               db      80d,24d,14d
0C9D  1000                                      2334    C               dw      01000h
                                                2335    C
0C9F  01 03 00 03                               2336    C               DB      001H,003H,000H,003H
                                                2337    C
0CA3  A7                                        2338    C               DB      0a7H
                                                2339    C
0CA4  5B 4F 53 37 51 5B                         2340    C               DB      05bH,04FH,053H,037H,051h,05bh
0CAA  6C 1F 00 0D 06 07                         2341    C               DB      06cH,01fH,000H,00dH,006H,007H
0CB0  00 00 00 00 5E 2B                         2342    C               DB      000H,000H,000H,000H,05eH,02bH
0CB6  5D 28 0F 5E 0A A3                         2343    C               DB      05dH,028H,00fH,05eH,00aH,0A3H
0CBC  FF                                        2344    C               DB      0FFH
                                                2345    C
0CBD  00 01 02 03 04 05                         2346    C               DB      000H,001H,002H,003H,004H,005H
0CC3  14 07 38 39 3A 3B                         2347    C               DB      014H,007H,038H,039H,03aH,03bH
0CC9  3C 3D 3E 3F 08 00                         2348    C               DB      03cH,03dH,03eH,03fH,008H,000H
0CCF  0F 00                                     2349    C               DB      00FH,000H
                                                2350    C
0CD1  00 00 00 00 00 10                         2351    C               DB      000H,000H,000H,000H,000H,010H
0CD7  0E 00 FF                                  2352    C               DB      00EH,000H,0FFH
                                                2353    C
                                                2354    C
                                                2355    C               SUBTTL
                                                2356
                                                2357        ;----- VECTOR INTO <AH> SPECIFIED FUNCTION
                                                2358
0CDA                                            2359        COMBO_VIDEO     PROC    NEAR
0CDA  FB                                        2360                STI                                     ; INTERRUPTS ON
0CDB  FC                                        2361                CLD                                     ; SET DIRECTION FORWARD
0CDC  55                                        2362                PUSH    BP                              ; SAVE THE REGISTER SET
0CDD  06                                        2363                PUSH    ES
0CDE  1E                                        2364                PUSH    DS
0CDF  52                                        2365                PUSH    DX
0CE0  51                                        2366                PUSH    CX
0CE1  53                                        2367                PUSH    BX
0CE2  56                                        2368                PUSH    SI
0CE3  57                                        2369                PUSH    DI
                                                2370
0CE4  50                                        2371                PUSH    AX                              ; SAVE AX VALUE
0CE5  8A C4                                     2372                MOV     AL,AH                           ; GET INTO LOW BYTE
0CE7  32 E4                                     2373                XOR     AH,AH                           ; ZERO TO HIGH BYTE
0CE9  D1 E0                                     2374                SAL     AX,1                            ; * 2 FOR TABLE LOOKUP
0CEB  8B F0                                     2375                MOV     SI,AX                           ; PUT INTO SI FOR BRANCH
0CED  3D 0028                                   2376                CMP     AX,T2L                          ; TEST FOR WITHIN RANGE
0CF0  72 06                                     2377                JB      M2                              ; BRANCH AROUND BRANCH
0CF2  58                                        2378                POP     AX                              ; RECOVER REGISTER
0CF3  CD 42                                     2379                INT     42H                             ; PASS UNRECOGNIZED CALL
0CF5  E9 219B R                                 2380                JMP     V_RET                           ; RETURN TO CALLER
0CF8                                            2381        M2:
                                                2382                ASSUME  DS:ABS0
0CF8  E8 0D01 R                                 2383                CALL    DDS
0CFB  58                                        2384                POP     AX                              ; RECOVER
0CFC  2E: FF A4 06F2 R                          2385                JMP     WORD PTR CS:[SI + OFFSET T2]    ; JMP TO AH=0 THRU AH=XX
                                                2386
                                                2387        ;----- UTILITY ROUTINES
                                                2388
                                                2389        ;----- SET DS TO THE DATA SEGMENT
                                                2390
0D01                                            2391        DDS     PROC    NEAR
0D01  50                                        2392                PUSH    AX                              ; SAVE REGISTER
0D02  2B C0                                     2393                sub     ax,ax
0D04  8E D8                                     2394                mov     ds,ax
```

```
0D06  58                      2395            POP     AX                              ; RESTORE REGISTER
0D07  C3                      2396            RET
0D08                          2397    DDS     ENDP
                              2398
0D08                          2399    WHAT_BASE       PROC    NEAR
                              2400            ASSUME  DS:ABS0
0D08  1E                      2401            PUSH    DS                              ; SAVE DATA SEGMENT
0D09  E8 0D01 R               2402            CALL    DDS                             ; GET LOW MEMORY SEGMENT
0D0C  8B 16 0463 R            2403            MOV     DX,ADDR_6845                    ; GET CRTC ADDRESS
0D10  80 E2 F0                2404            AND     DL,0F0H                         ; STRIP OFF LOW NIBBLE
0D13  80 CA 0A                2405            OR      DL,0AH                          ; SET TO STATUS REGISTER
0D16  1F                      2406            POP     DS
0D17  C3                      2407            RET
0D18                          2408    WHAT_BASE       ENDP
                              2409
0D18                          2410    OUT_DX  PROC    NEAR                            ; AH=INDEX,AL=DATA,DX=PORT
0D18  86 C4                   2411            XCHG    AL,AH                           ; GET INDEX VALUE
                              2412            WOUT                                    ; SET INDEX REG
0D1A  EE                      2413    +       OUT     DX,AL
0D1B  42                      2414            INC     DX                              ; SET DX TO DATA REG
0D1C  86 C4                   2415            XCHG    AL,AH                           ; GET DATA VALUE
                              2416            WOUT                                    ; SET DATA REG
0D1E  EE                      2417    +       OUT     DX,AL
0D1F  4A                      2418            DEC     DX                              ; SET DX BACK TO INDEX
0D20  C3                      2419            RET
0D21                          2420    OUT_DX  ENDP
                              2421
                              2422    ;----- ROUTINE TO SOUND BEEPER
                              2423
0D21                          2424    BP_1    PROC    NEAR
                              2425            WOUT
0D21  EE                      2426    +       OUT     DX,AL
0D22  C3                      2427            RET
0D23                          2428    BP_1    ENDP
                              2429
0D23                          2430    BEEP    PROC    NEAR
0D23  52                      2431            PUSH    DX
0D24  BA 0043                 2432            MOV     DX,TIMER+3
0D27  B0 B6                   2433            MOV     AL,10110110B                    ; SEL TIM 2,LSB,MSB,BINARY
0D29  E8 0D21 R               2434            CALL    BP_1                            ; WRITE THE TIMER MODE REG
0D2C  B8 0533                 2435            MOV     AX,533H                         ; DIVISOR FOR 1000 HZ
0D2F  4A                      2436            DEC     DX
0D30  E8 0D21 R               2437            CALL    BP_1                            ; WRITE TIMER 2 CNT - LSB
0D33  8A C4                   2438            MOV     AL,AH
0D35  E8 0D21 R               2439            CALL    BP_1                            ; WRITE TIMER 2 CNT - MSB
0D38  BA 0061                 2440            MOV     DX,PORT_B
                              2441            WIN                                     ; GET SETTING OF PORT
0D3B  EC                      2442    +       IN      AL,DX
0D3C  8A E0                   2443            MOV     AH,AL                           ; SAVE THAT SETTING
0D3E  0C 03                   2444            OR      AL,03                           ; TURN SPEAKER ON
0D40  E8 0D21 R               2445            CALL    BP_1
0D43  2B C9                   2446            SUB     CX,CX                           ; SET CNT TO WAIT 500 MS
0D45                          2447    G7:
0D45  E2 FE                   2448            LOOP    G7                              ; DELAY BEFORE TURNING OFF
0D47  FE CB                   2449            DEC     BL                              ; DELAY CNT EXPIRED?
0D49  75 FA                   2450            JNZ     G7                              ; NO-CONTINUE BEEPING SPK
0D4B  8A C4                   2451            MOV     AL,AH                           ; RECOVER VALUE OF PORT
0D4D  E8 0D21 R               2452            CALL    BP_1
0D50  5A                      2453            POP     DX
0D51  C3                      2454            RET                                     ; RETURN TO CALLER
0D52                          2455    BEEP    ENDP
                              2456
                              2457    ;----- find the parameter table vector in the save table
                              2458
0D52                          2459    set_base        proc    near
                              2460            assume  ds:abs0
0D52  E8 0D01 R               2461            call    dds
                              2462            wlxs    es,bx,save_ptr                  ; get ptr to ptr table
0D55  C4 1E 04A8 R            2463    +       Les     bx,save_ptr
0D59  26: C4 1F               2464            les     bx,dword ptr es:[bx]            ; get parameter ptr
0D5C  C3                      2465            ret
0D5D                          2466    set_base        endp
                              2467
                              2468    ;----- establish addressing to the correct mode table entry
                              2469
0D5D                          2470    make_base       proc    near
                              2471            assume  ds:abs0
0D5D  51                      2472            push    cx
0D5E  52                      2473            push    dx
0D5F  E8 0D52 R               2474            call    set_base                        ; get parm tbl ptr
0D62  8A 26 0449 R            2475            mov     ah,crt_mode
0D66  F6 06 0487 R 60         2476            test    info,060h                       ; test for base card
0D6B  74 18                   2477            jz      b_m_1                           ; min memory
                              2478
                              2479    ;----- we have a memory expansion option here
                              2480
0D6D  80 FC 0F                2481            cmp     ah,0fh
0D70  75 07                   2482            jne     b_m_2
0D72  81 C3 0440              2483            add     bx,base_2 - base_1
0D76  EB 33 90                2484            jmp     b_m_out
0D79                          2485    b_m_2:
0D79  80 FC 10                2486            cmp     ah,010h
0D7C  75 07                   2487            jne     b_m_1
0D7E  81 C3 0480              2488            add     bx,base_2 + m_tbl_len - base_1
0D82  EB 27 90                2489            jmp     b_m_out
0D85                          2490    b_m_1:
0D85  80 FC 03                2491            cmp     ah,03h
0D88  77 14                   2492            ja      b_m_3                           ; skip enhanced portion
                              2493
                              2494    ;----- check the switch setting for enhancement
                              2495
0D8A  A0 0488 R               2496            mov     al,info_3
0D8D  24 0F                   2497            and     al,0fh
0D8F  3C 03                   2498            cmp     al,03h                          ; secondary emulate setting
0D91  74 07                   2499            je      brs
0D93  3C 09                   2500            cmp     al,09h                          ; primary emulate setting
0D95  74 03                   2501            je      brs
0D97  EB 05 90                2502            jmp     b_m_3
                              2503
                              2504    ;----- we will perform enhancement
                              2505
0D9A                          2506    brs:
0D9A  81 C3 04C0              2507            add     bx,base_3 - base_1              ; vector to enhancement tbl
0D9E                          2508    b_m_3:
0D9E  8A 0E 0449 R            2509            mov     cl,crt_mode
0DA2  2A ED                   2510            sub     ch,ch
0DA4  E3 05                   2511            jcxz    b_m_4
                              2512
                              2513    ;----- this loop will move the ptr to the individual mode entry
                              2514
0DA6                          2515    b_m_5:
0DA6  83 C3 40                2516            add     bx,m_tbl_len                    ; length of one mode entry
0DA9  E2 FB                   2517            loop    b_m_5
0DAB                          2518    b_m_4:
0DAB                          2519    b_m_out:
0DAB  5A                      2520            pop     dx
```

```
0F32                                    2773    ST_2A:
0F32  A2 0449 R                         2774            MOV     CRT_MODE,AL                     ; SAVE MODE VALUE
0F35  B2 B4                             2775            MOV     DL,CRTC_ADDR_B                  ; IT IS (2/3)-B-X
0F37  89 16 0463 R                      2776            MOV     ADDR_6845,DX                    ; SAVE CRTC ADDRESS
0F3B  EB 1C 90                          2777            JMP     QQ1                             ; CONTINUE THE MODE SET
                                        2778
                                        2779    ;----- COLOR SETUP TO THE ADAPTER
                                        2780
0F3E                                    2781    ST_3:
0F3E  58                                2782            POP     AX                              ; RECOVER PARAMETER VALUE
0F3F  50                                2783            PUSH    AX                              ; SAVE IT
0F40  B6 03                             2784            mov     dh,3
0F42  24 80                             2785            AND     AL,080H                         ; ISOLATE REGEN CLEAR BIT
0F44  80 26 0487 R 7F                   2786            AND     INFO,07FH                       ; PREPARE INFO BYTE
0F49  08 06 0487 R                      2787            OR      INFO,AL                         ; SET IT, OR NOT
0F4D  58                                2788            POP     AX                              ; RECOVER TRUE MODE CALL
0F4E  24 7F                             2789            AND     AL,07FH                         ; DONE WITH D7
0F50  A2 0449 R                         2790            MOV     CRT_MODE,AL                     ; SAVE THIS MODE
0F53  B2 D4                             2791            MOV     DL,CRTC_ADDR                    ; (2/3)-D-X
0F55  89 16 0463 R                      2792            MOV     ADDR_6845,DX                    ; SAVE CRTC ADDRESS
0F59                                    2793    QQ1:
0F59  C7 06 044E R 0000                 2794            MOV     CRT_START,0                     ; SAVE START ADDRESS
0F5F  C6 06 0462 R 00                   2795            MOV     ACTIVE_PAGE,0                   ; RESET PAGE VALUE TO ZERO
                                        2796            ASSUME  ES:NOTHING
0F64  B9 0008                           2797            MOV     CX,8                            ; 8 PAGES OF CURSOR VALUES
0F67  BF 0450 R                         2798            MOV     DI,OFFSET CURSOR_POSN           ; OFFSET
0F6A  1E                                2799            PUSH    DS                              ; ESTABLISH
0F6B  07                                2800            POP     ES                              ;  ADDRESSING
0F6C  2B C0                             2801            SUB     AX,AX                           ; 0 THOSE CURSOR LOCATIONS
0F6E  F3/ AB                            2802            REP     STOSW                           ; CLEAR OUT SAVED VALUES
                                        2803
0F70  E8 0D5D R                         2804            call    make_base
                                        2805
0F73  26: 8A 07                         2806            MOV     AL,ES:[bx]                      ; GET COLUMN COUNT
0F76  2A E4                             2807            SUB     AH,AH                           ; ZERO HIGH BYTE
0F78  A3 044A R                         2808            MOV     CRT_COLS,AX                     ; STORE COLUMN VALUE
                                        2809
0F7B  26: 8A 47 01                      2810            MOV     AL,ES:[bx][1]                   ; GET ROW VALUE
0F7F  A2 0484 R                         2811            MOV     ROWS,AL                         ; STORE ROW VALUE
                                        2812
0F82  26: 8A 47 02                      2813            MOV     AL,ES:[bx][2]                   ; GET THE BYTES/CHAR
0F86  2A E4                             2814            SUB     AH,AH                           ; ZERO HIGH BYTE
0F88  A3 0485 R                         2815            MOV     POINTS,AX                       ; STORE BYTES/CHAR
                                        2816
0F8B  26: 8B 47 03                      2817            MOV     AX,ES:[bx][3]                   ; GET PAGE SIZE
0F8F  A3 044C R                         2818            MOV     CRT_LEN,AX                      ; STORE PAGE LENGTH
                                        2819
0F92  2B DB                             2820            SUB     BX,BX                           ; ZERO
0F94  B0 01                             2821            MOV     AL,1                            ; MONOCHROME ALPHA CHAR GEN
0F96  8A 26 0449 R                      2822            MOV     AH,CRT_MODE                     ; GET CURRENT MODE
0F9A  80 FC 07                          2823            CMP     AH,7                            ; IS IT MONOCHROME
0F9D  74 0C                             2824            JE      ENTRY_2                         ; 9X14 FONT
0F9F  80 FC 03                          2825            cmp     ah,03h
0FA2  77 35                             2826            ja      entry_1
                                        2827
0FA4  E8 0E9C R                         2828            call    brst_det
0FA7  72 02                             2829            jc      entry_2
                                        2830
0FA9  B0 02                             2831            MOV     AL,2                            ; COLOR ALPHA CHAR GEN
0FAB                                    2832    ENTRY_2:
0FAB  E8 1EAB R                         2833            CALL    CH_GEN                          ; LOAD ALPHA CHAR GEN
0FAE  E8 0D01 R                         2834            CALL    DDS
0FB1  8A 26 0449 R                      2835            MOV     AH,CRT_MODE                     ; GET CURRENT MODE
0FB5  80 FC 07                          2836            CMP     AH,7                            ; IS IT MONOCHROME
0FB8  74 03                             2837            JE      FDG_IT                          ; 9X14 FONT
0FBA  EB 1D 90                          2838            JMP     ENTRY_1
0FBD                                    2839    FDG_IT:
0FBD  BD 0000 E                         2840            MOV     BP,OFFSET CGMN_FDG              ; TABLE POINTER
0FC0  BB 0E00                           2841            MOV     BX,0E00H                        ; 14 BYTES PER CHAR
0FC3                                    2842    FDG:
0FC3  0E                                2843            PUSH    CS                              ; GET THE ROM SEGMENT
0FC4  07                                2844            POP     ES                              ;  INTO ES
0FC5  26: 8B 56 00                      2845            MOV     DX,ES:[BP]                      ; GET THE CHAR HEX CODE
0FC9  0B D2                             2846            OR      DX,DX                           ; ZERO = NO MORE CHARS
0FCB  74 0C                             2847            JZ      ENTRY_1                         ; NO MORE
0FCD  B9 0001                           2848            MOV     CX,1                            ; DO ONE CHAR AT A TIME
0FD0  45                                2849            INC     BP                              ; MOVE TO FIRST CODE POINT
0FD1  E8 1EF3 R                         2850            CALL    DO_MAP2                         ; STORE THE CODE POINT
0FD4  83 C5 0E                          2851            ADD     BP,014D                         ; ADJUST BP TO NEXT CODE
0FD7  EB EA                             2852            JMP     FDG                             ; DO ANOTHER
0FD9                                    2853    ENTRY_1:
0FD9  E8 0DAE R                         2854            call    set_regs
0FDC  E8 0E57 R                         2855            CALL    BLANK                           ; CLEAR OUT THE BUFFER
0FDF  E8 0E98 R                         2856            CALL    PH_5
                                        2857
                                        2858            ASSUME  DS:ABS0
0FE2  E8 0D01 R                         2859            CALL    DDS
0FE5  80 3E 0449 R 0F                   2860            cmp     crt_mode,0fh
0FEA  72 06                             2861            jb      ms_1
0FEC  C7 06 010C R 0000 E               2862            MOV     word ptr GRX_SET , OFFSET CGMN
0FF2                                    2863    MS_1:
0FF2  80 3E 0449 R 07                   2864            cmp     crt_mode,7
0FF7  77 09                             2865            ja      save_grph
0FF9  74 4B                             2866            je      save_alph
0FFB  80 3E 0449 R 03                   2867            cmp     crt_mode,3
1000  76 44                             2868            jbe     save_alph
1002                                    2869    save_grph:
1002  C4 1E 04A8 R                      2870            les     bx,save_ptr
1006  83 C3 0C                          2871            add     bx,0ch
1009  26: C4 1F                         2872            les     bx,dword ptr es:[bx]
100C  8C C0                             2873            mov     ax,es
100E  0B C3                             2874            or      ax,bx
1010  74 32                             2875            jz      j4j                             ; jmp ah0_done
1012  BE 0007                           2876            mov     si,07h
1015                                    2877    sg_1:
1015  26: 8A 00                         2878            mov     al,es:[bx][si]
1018  3C FF                             2879            cmp     al,0ffh
101A  74 7A                             2880            je      ah0_done
101C  3A 06 0449 R                      2881            cmp     al,crt_mode
1020  74 03                             2882            je      sg_2
1022  46                                2883            inc     si
1023  EB F0                             2884            jmp     sg_1
1025                                    2885    sg_2:
1025  FA                                2886            cli
1026  26: 8A 07                         2887            mov     al,byte ptr es:[bx]
1029  FE C8                             2888            dec     al
102B  A2 0484 R                         2889            mov     rows,al
102E  26: 8B 47 01                      2890            mov     ax,word ptr es:[bx][1]
1032  A3 0485 R                         2891            mov     points,ax
1035  26: 8B 47 03                      2892            mov     ax,word ptr es:[bx][3]
1039  A3 010C R                         2893            mov     word ptr grx_set,ax
103C  26: 8B 47 05                      2894            mov     ax,word ptr es:[bx][5]
1040  A3 010E R                         2895            mov     word ptr grx_set + 2,ax
1043  FB                                2896            sti
1044                                    2897    j4j:
1044  EB 50                             2898            jmp     short ah0_done
```

```
1046                                    save_alph:
1046  C4 1E 04A8 R                              les     bx,save_ptr
104A  83 C3 08                                  add     bx,08h
104D  26: C4 1F                                 les     bx,dword ptr es:[bx]
1050  8C C0                                    mov     ax,es
1052  0B C3                                    or      ax,bx
1054  74 40                                    jz      ah0_done
1056  BE 000B                                  mov     si,0bh
1059                                    sa_1:
1059  26: 8A 00                                 mov     al,es:[bx][si]
105C  3C FF                                     cmp     al,0ffh
105E  74 36                                     je      ah0_done
1060  3A 06 0449 R                              cmp     al,crt_mode
1064  74 03                                    je      sa_2
1066  46                                       inc     si
1067  EB F0                                    jmp     sa_1
1069                                    sa_2:
1069  26: 8A 27                                 mov     ah,es:[bx]
106C  26: 8A 47 01                              mov     al,es:[bx][1]
1070  26: 8B 4F 02                              mov     cx,es:[bx][2]
1074  26: 8B 57 04                              mov     dx,es:[bx][4]
1078  26: 8B 6F 06                              mov     bp,es:[bx][6]
107C  26: 8E 47 08                              mov     es,es:[bx][8]
1080  53                                       push    bx
1081  8B D8                                    mov     bx,ax
1083  B8 1110                                  mov     ax,1110h
1086  CD 10                                    int     10h
1088  5B                                       pop     bx
1089  26: 8A 47 0A                             mov     al,es:[bx][0ah]
108D  3C FF                                     cmp     al,0ffh
108F  74 05                                     je      ah0_done
1091  FE C8                                    dec     al
1093  A2 0484 R                                mov     rows,al
                                        
                                        ;----- set the low ram values for compatibility (3D8 and 3D9 save bytes)
                                        
1096                                    ah0_done:
1096  E8 0D01 R                                 call    dds
1099  80 3E 0449 R 07                           cmp     crt_mode,7
109E  77 1E                                     ja      dndcs
10A0  BB 10C5 R                                 mov     bx,offset compat_mode
10A3  A0 0449 R                                 mov     al,crt_mode
10A6  2A E4                                     sub     ah,ah
10A8  03 D8                                     add     bx,ax
10AA  2E: 8A 07                                 mov     al,cs:[bx]
10AD  A2 0465 R                                 mov     crt_mode_set,al
10B0  B0 30                                     mov     al,030h
10B2  80 3E 0449 R 06                           cmp     crt_mode,6
10B7  75 02                                     jne     do_pal
10B9  B0 3F                                     mov     al,03fh
10BB                                    do_pal:
10BB  A2 0466 R                                 mov     crt_palette,al
10BE                                    dndcs:
10BE  8B 0E 0460 R                              mov     cx,cursor_mode
10C2  EB 28 90                                  jmp     ah1
                                        
10C5                                    compat_mode     label   byte
10C5  2C 28 2D 29 2A 2E                         db      02ch,028h,02dh,029h,02ah,02eh
10CB  1E 29                                     db      01eh,029h
                                        
                                        C       INCLUDE V1-5.INC
                                        C       SUBTTL V1-5.INC
                                        C       PAGE
                                        C
10CD                                    C calc_cursor   proc    near
                                        C       assume  ds:abs0
10CD  80 FD 00                          C       cmp     ch,0                    ; check for full height
10D0  75 04                             C       jne     cc_1                    ; normal check
10D2  FE C1                             C       inc     cl                      ; adjust end value
10D4  EB 0A                             C       jmp     short   calc_out
10D6                                    C cc_1:
10D6  FE C1                             C       inc     cl                      ; adjust for ega registers
10D8  3A 0E 0485 R                      C       cmp     cl,byte ptr points      ; will it wrap
10DC  72 02                             C       jb      calc_out                ; no, its ok
10DE  2A C9                             C       sub     cl,cl                   ; ega method for cursor end
10E0                                    C calc_out:
10E0  51                                C       push    cx                      ; save cursor type value
10E1  2A CD                             C       sub     cl,ch                   ; end - start
10E3  80 F9 10                          C       cmp     cl,010h                 ; low nibble equal
10E6  59                                C       pop     cx                      ; restore
10E7  75 02                             C       jne     comp_4
10E9  FE C1                             C       inc     cl                      ; add 1 for correct cursor
10EB                                    C comp_4:
10EB  C3                                C       ret                             ; back to caller
10EC                                    C calc_cursor   endp
                                        C
                                        C ;--------------------------------------------------------------------
                                        C ; SET_CTYPE     SET CURSOR TYPE                                   :
                                        C ;       THIS ROUTINE SETS THE CURSOR VALUE                        :
                                        C ; INPUT                                                           :
                                        C ;       (CX) HAS CURSOR VALUE CH-START LINE, CL-STOP LINE         :
                                        C ; OUTPUT                                                          :
                                        C ;       NONE                                                      :
                                        C ;--------------------------------------------------------------------
= 0004                                  C cut_off       equ     4
10EC                                    C AH1:
                                        C       ASSUME  DS:ABS0
10EC  B4 0A                             C       MOV     AH,C_CRSR_START         ; CRTC REG FOR CURSOR SET
10EE  89 0E 0460 R                      C       MOV     CURSOR_MODE,CX          ; SAVE IN DATA AREA
10F2  F6 06 0487 R 08                   C       test    info,8                  ; ega active bit
10F7  75 33                             C       jnz     do_set                  ; 0=ega, 1=old cards
                                        C
                                        C ;----- this section will emulate cursor off on the ega
                                        C
10F9  8A C5                             C       MOV     AL,CH                   ; GET START VALUE
10FB  24 60                             C       AND     AL,060H                 ; TURN OFF CURSOR ?
10FD  3C 20                             C       CMP     AL,020H                 ; TEST THE BITS
10FF  75 05                             C       jne     ah1_a                   ; skip cursor off
1101  B9 1E00                           C       MOV     CX,01E00H               ; EMULATE CURSOR OFF
1104  EB 26                             C       jmp     short do_set
                                        C
                                        C ;----- this section : adjust the cursor and test for enhanced operation
                                        C
1106                                    C ah1_a:
1106  F6 06 0487 R 01                   C       test    info,1                  ; cursor emulate bit
110B  75 1F                             C       jnz     do_set                  ; 0=emulate, 1=value as-is
110D  80 3E 0449 R 03                   C       cmp     crt_mode,3              ; possible emulation
1112  77 15                             C       ja      ah1_s                   ; no, set the cursor type
1114  E8 0E9C R                         C       call    brst_det                ; see if emulate mode
1117  73 10                             C       jnc     ah1_s                   ; not emulating
1119  80 FD 04                          C       cmp     ch,cut_off              ; test start
111C  76 03                             C       jbe     ah1_b                   ; skip adjust
111E  80 C5 05                          C       add     ch,5                    ; adjust
1121                                    C ah1_b:
1121  80 F9 04                          C       cmp     cl,cut_off              ; test end
1124  76 03                             C       jbe     ah1_s                   ; skip adjust
```

```
1126  80 C1 05                3025     C          add     cl,5
1129                          3026     C  AH1_S:
1129  E8 10CD R               3027     C          call    calc_cursor                     ; adjust end register
112C                          3028     C  do_set:
112C  E8 1132 R               3029     C          CALL    M16                             ; OUTPUT CX REG
112F  E9 219B R               3030     C          JMP     V_RET                           ; RETURN TO CALLER
                              3031     C
                              3032     C  ;----- THIS ROUTINE OUTPUTS THE CX REGISTER TO THE CRTC REGS NAMED IN AH
1132                          3034     C  M16:
1132  8B 16 0463 R            3035     C          MOV     DX,ADDR_6845                    ; ADDRESS REGISTER
1136  8A C5                   3036     C          MOV     AL,CH                           ; DATA
1138  E8 0D18 R               3037     C          CALL    OUT_DX                          ; OUTPUT THE VALUE
113B  FE C4                   3038     C          INC     AH                              ; NEXT REGISTER
113D  8A C1                   3039     C          MOV     AL,CL                           ; SECOND DATA VALUE
113F  E8 0D18 R               3040     C          CALL    OUT_DX                          ; OUTPUT THE VALUE
1142  C3                      3041     C          RET                                     ; ALL DONE
                              3042     C
                              3043     C  ;--------------------------------------------------------
                              3044     C  ; POSITION                                               :
                              3045     C  ;       THIS SERVICE ROUTINE CALCULATES THE REGEN BUFFER  :
                              3046     C  ;       ADDRESS OF A CHARACTER IN THE ALPHA MODE          :
                              3047     C  ; INPUT                                                  :
                              3048     C  ;       AX = ROW, COLUMN POSITION                        :
                              3049     C  ; OUTPUT                                                 :
                              3050     C  ;       AX = OFFSET OF CHAR POSITION IN REGEN BUFFER      :
                              3051     C  ;--------------------------------------------------------
1143                          3052     C  POSITION        PROC    NEAR
1143  53                      3053     C          PUSH    BX                              ; SAVE REGISTER
1144  8B D8                   3054     C          MOV     BX,AX
1146  8A C4                   3055     C          MOV     AL,AH                           ; ROWS TO AL
1148  F6 26 044A R            3056     C          MUL     BYTE PTR CRT_COLS               ; DETERMINE BYTES TO ROW
114C  32 FF                   3057     C          XOR     BH,BH                           ; ZERO OUT
114E  03 C3                   3058     C          ADD     AX,BX                           ; ADD IN COLUMN VALUE
1150  D1 E0                   3059     C          SAL     AX,1                            ; * 2 FOR ATTRIBUTE BYTES
1152  5B                      3060     C          POP     BX                              ; RESTORE REGISTER
1153  C3                      3061     C          RET
1154                          3062     C  POSITION        ENDP
                              3063     C
                              3064     C  ;--------------------------------------------------------------
                              3065     C  ; SET_CPOS      SET CURSOR POSITION                       :
                              3066     C  ;       THIS ROUTINE SETS THE CURRENT CURSOR POSITION TO THE  :
                              3067     C  ;       NEW X-Y VALUES PASSED                             :
                              3068     C  ; INPUT                                                   :
                              3069     C  ;       DX - ROW,COLUMN OF NEW CURSOR                     :
                              3070     C  ;       BH - DISPLAY PAGE OF CURSOR                       :
                              3071     C  ; OUTPUT                                                  :
                              3072     C  ;       CURSOR IS SET AT CRTC IF DISPLAY PAGE IS CURRENT  :
                              3073     C  ;       DISPLAY                                           :
                              3074     C  ;--------------------------------------------------------------
1154                          3075     C  AH2:
1154  E8 115A R               3076     C          CALL    SET_CPOS
1157  E9 219B R               3077     C          JMP     V_RET
                              3078     C
115A                          3079     C  SET_CPOS:
115A  8A CF                   3080     C          MOV     CL,BH
115C  32 ED                   3081     C          XOR     CH,CH                           ; ESTABLISH LOOP COUNT
115E  D1 E1                   3082     C          SAL     CX,1                            ; WORD OFFSET
1160  8B F1                   3083     C          MOV     SI,CX                           ; USE INDEX REGISTER
1162  89 94 0450 R            3084     C          MOV     [SI+OFFSET CURSOR_POSN],DX      ; SAVE THE POINTER
1166  38 3E 0462 R            3085     C          CMP     ACTIVE_PAGE,BH
116A  75 05                   3086     C          JNZ     M17                             ; SET_CPOS_RETURN
116C  8B C2                   3087     C          MOV     AX,DX                           ; GET ROW/COLUMN TO AX
116E  E8 1172 R               3088     C          CALL    M18                             ; CURSOR_SET
1171                          3089     C  M17:                                            ; SET_CPOS_RETURN
1171  C3                      3090     C          RET
                              3091     C
                              3092     C  ;----- SET CURSOR POSITION, AX HAS ROW/COLUMN FOR CURSOR
                              3093     C
1172                          3094     C  M18     PROC    NEAR
1172  E8 1143 R               3095     C          CALL    POSITION                        ; DETERMINE LOC IN REGEN
1175  8B C8                   3096     C          MOV     CX,AX
1177  03 0E 044E R            3097     C          ADD     CX,CRT_START                    ; ADD IN THE START ADDR
                              3098     C                                                  ;  FOR THIS PAGE
117B  D1 F9                   3099     C          SAR     CX,1                            ; / 2 FOR CHAR ONLY COUNT
117D  B4 0E                   3100     C          MOV     AH,C_CRSR_LOC_HGH               ; REGISTER NUMBER FOR CURSOR
117F  E8 1132 R               3101     C          CALL    M16                             ; SET VALUE TO CRTC
1182  C3                      3102     C          RET
1183                          3103     C  M18     ENDP
                              3104     C
                              3105     C  ;--------------------------------------------------------
                              3106     C  ; READ_CURSOR                                            :
                              3107     C  ;       THIS ROUTINE READS THE CURRENT CURSOR VALUE FROM  :
                              3108     C  ;       MEMORY AND SENDS IT BACK TO THE CALLER            :
                              3109     C  ; INPUT                                                  :
                              3110     C  ;       BH - PAGE OF CURSOR                              :
                              3111     C  ; OUTPUT                                                 :
                              3112     C  ;       DX - ROW, COLUMN OF THE CURRENT CURSOR POSITION   :
                              3113     C  ;       CX - CURRENT CURSOR MODE                         :
                              3114     C  ;--------------------------------------------------------
1183                          3115     C  AH3:
1183  8A DF                   3116     C          MOV     BL,BH                           ; PAGE VALUE
1185  32 FF                   3117     C          XOR     BH,BH                           ; ZERO UPPER BYTE
1187  D1 E3                   3118     C          SAL     BX,1                            ; WORD OFFSET
1189  8B 97 0450 R            3119     C          MOV     DX,[BX + OFFSET CURSOR_POSN]    ; GET CURSOR FOR THIS PAGE
118D  8B 0E 0460 R            3120     C          MOV     CX,CURSOR_MODE                  ; GET THE CURSOR MODE
1191  5F                      3121     C          POP     DI
1192  5E                      3122     C          POP     SI
1193  5B                      3123     C          POP     BX
1194  58                      3124     C          POP     AX                              ; DISCARD CX
1195  58                      3125     C          POP     AX                              ; DISCARD DX
1196  1F                      3126     C          POP     DS
1197  07                      3127     C          POP     ES
1198  5D                      3128     C          POP     BP
1199  CF                      3129     C          IRET
                              3130     C
                              3131     C  ;----- READ LIGHT PEN POSITION
                              3132     C
119A                          3133     C  AH4:
119A  A0 0449 R               3134     C          MOV     AL,CRT_MODE
119D  3C 07                   3135     C          CMP     AL,07H
119F  77 37                   3136     C          JA      READ_LPEN
                              3137     C
11A1  F6 06 0487 R 02         3138     C          test    info,2
11A6  74 07                   3139     C          JZ      CVA_IS_COLOR
                              3140     C
                              3141     C  ;----- MONOCHROME HERE (MONOC BIT 1)
                              3142     C
11A8  3C 07                   3143     C          CMP     AL,07H
11AA  74 2C                   3144     C          JE      READ_LPEN
11AC  EB 05 90                3145     C          JMP     OLD_LP
                              3146     C
                              3147     C  ;----- CVA IS COLOR HERE (MONOC BIT 0)
                              3148     C
11AF                          3149     C  CVA_IS_COLOR:
11AF  3C 06                   3150     C          CMP     AL,06H
```

```
11B1  76 25                    3151  C          JBE     READ_LPEN
11B3                           3152  C  OLD_LP:
11B3  CD 42                    3153  C          INT     42H                     ; CALL EXISTING CODE
11B5  5F                       3154  C          POP     DI
11B6  5E                       3155  C          POP     SI
11B7  83 C4 06                 3156  C          ADD     SP,6                    ; DISCARD SAVED BX,CX,DX
11BA  1F                       3157  C          POP     DS
11BB  07                       3158  C          POP     ES
11BC  5D                       3159  C          POP     BP
11BD  CF                       3160  C          IRET
                               3161  C
                               3162  C ;--------------------------------------------------------------------------
                               3163  C ; LIGHT PEN                                                             :
                               3164  C ;       THIS ROUTINE TESTS THE LIGHT PEN SWITCH AND THE LIGHT           :
                               3165  C ;       PEN TRIGGER. IF BOTH ARE SET, THE LOCATION OF THE LIGHT         :
                               3166  C ;       PEN IS DETERMINED. OTHERWISE, A RETURN WITH NO                  :
                               3167  C ;       INFORMATION IS MADE.                                            :
                               3168  C ; ON EXIT                                                               :
                               3169  C ;       (AH) = 0 IF NO LIGHT PEN INFORMATION IS AVAILABLE               :
                               3170  C ;            BX,CX,DX ARE DESTROYED                                     :
                               3171  C ;       (AH) = 1 IF LIGHT PEN IS AVAILABLE                              :
                               3172  C ;                (DH,DL) = ROW,COLUMN OF CURRENT LIGHT PEN              :
                               3173  C ;                          POSITION                                     :
                               3174  C ;                (CH) = RASTER POSITION (OLD MODES)                     :
                               3175  C ;                (CX) = RASTER POSITION (NEW MODES)                     :
                               3176  C ;                (BX) = BEST GUESS AT PIXEL HORIZONTAL POSITION         :
                               3177  C ;--------------------------------------------------------------------------
                               3178  C          ASSUME  CS:CODE,DS:ABS0
                               3179  C ;----- SUBTRACT_TABLE
11BE                           3180  C  V1       LABEL   BYTE
11BE  06 06 07 07 05 05        3181  C          DB      006H,006H,007H,007H,005H,005H   ; 0-5
11C4  04 05 00 00 00 00        3182  C          DB      004H,005H,000H,000H,000H,000H   ; 6-B
11CA  00 05 06 04 04 04        3183  C          DB      000H,005H,006H,004H,004H,004H   ; C-11
11D0  04 06 06 04 07 04        3184  C          DB      004H,006H,006H,004H,007H,004H   ; 12-17
11D6  07 04                    3185  C          DB      007H,004H                       ; 18-19
                               3186  C
11D8                           3187  C  READ_LPEN        PROC    NEAR
                               3188  C
                               3189  C ;----- WAIT FOR LIGHT PEN TO BE DEPRESSED
                               3190  C
11D8  8B 16 0463 R             3191  C          MOV     DX,ADDR_6845            ; GET BASE ADDRESS OF 6845
11DC  83 C2 06                 3192  C          ADD     DX,6                    ; POINT TO STATUS REGISTER
                               3193  C          WIN                             ; GET STATUS REGISTER
11DF  EC                       3194  C+         IN      AL,DX
11E0  A8 04                    3195  C          TEST    AL,4                    ; TEST LIGHT PEN SWITCH
11E2  B4 00                    3196  C          MOV     AH,0                    ; SET NO LIGHT PEN RETURN
11E4  74 03                    3197  C          JZ      V9                      ;  CODE
11E6  E9 128E R                3198  C          JMP     V6                      ; NOT SET, RETURN
                               3199  C
                               3200  C ;----- NOW TEST FOR LIGHT PEN TRIGGER
                               3201  C
11E9                           3202  C  V9:
11E9  A8 02                    3203  C          TEST    AL,2                    ; TEST LIGHT PEN TRIGGER
11EB  75 03                    3204  C          JNZ     V7A                     ; RETURN WITHOUT RESETTING
                               3205  C                                          ;  TRIGGER
11ED  E9 1298 R                3206  C          JMP     V7                      ; EXIT LIGHT PEN ROUTINE
                               3207  C
                               3208  C ;----- TRIGGER HAS BEEN SET, READ THE VALUE IN
                               3209  C
11F0                           3210  C  V7A:
11F0  B4 10                    3211  C          MOV     AH,16                   ; LIGHT PEN REGISTERS
                               3212  C
                               3213  C ;----- INPUT REGS POINTED TO BY AH, AND CONVERT TO ROW COLUMN IN DX
                               3214  C
11F2  8B 16 0463 R             3215  C          MOV     DX,ADDR_6845            ; ADDRESS REGISTER
11F6  8A C4                    3216  C          MOV     AL,AH                   ; REGISTER TO READ
                               3217  C          WOUT                            ; SET IT UP
11F8  EE                       3218  C+         OUT     DX,AL
11F9  42                       3219  C          INC     DX                      ; DATA REGISTER
11FA  50                       3220  C          PUSH    AX
                               3221  C          WIN                             ; GET THE VALUE
11FB  EC                       3222  C+         IN      AL,DX
11FC  8A E8                    3223  C          MOV     CH,AL                   ; SAVE IN CX
11FE  58                       3224  C          POP     AX
11FF  4A                       3225  C          DEC     DX                      ; ADDRESS REGISTER
1200  FE C4                    3226  C          INC     AH
1202  8A C4                    3227  C          MOV     AL,AH                   ; SECOND DATA REGISTER
                               3228  C          WOUT
1204  EE                       3229  C+         OUT     DX,AL
1205  42                       3230  C          INC     DX                      ; POINT TO DATA REGISTER
                               3231  C          WIN                             ; GET THE 2ND DATA VALUE
1206  EC                       3232  C+         IN      AL,DX
1207  8A E5                    3233  C          MOV     AH,CH                   ; AX HAS INPUT VALUE
                               3234  C
                               3235  C ;----- AX HAS THE VALUE READ IN FROM THE 6845
                               3236  C
1209  8A 1E 0449 R             3237  C          MOV     BL,CRT_MODE
120D  2A FF                    3238  C          SUB     BH,BH                   ; MODE VALUE TO BX
120F  2E: 8A 9F 11BE R         3239  C          MOV     BL,CS:V1[BX]            ; AMOUNT TO SUBTRACT
1214  2B C3                    3240  C          SUB     AX,BX                   ; TAKE IT AWAY
1216  8B 1E 044E R             3241  C          MOV     BX,CRT_START            ; SCREEN ADDRESS
121A  D1 EB                    3242  C          SHR     BX,1                    ; DIVIDE BY 2
121C  2B C3                    3243  C          SUB     AX,BX                   ; ADJUST TO ZERO START
121E  79 02                    3244  C          JNS     V2                      ; IF POSITIVE, GET MODE
1220  2B C0                    3245  C          SUB     AX,AX                   ; <0 PLAYS AS 0
                               3246  C
                               3247  C ;----- DETERMINE MODE OF OPERATION
                               3248  C
1222                           3249  C  V2:                                     ; DETERMINE_MODE
1222  B1 03                    3250  C          MOV     CL,3                    ; SET *8 SHIFT COUNT
1224  80 3E 0449 R 04          3251  C          CMP     CRT_MODE,4              ; GRAPHICS OR ALPHA
1229  72 4D                    3252  C          JB      V4                      ; ALPHA_PEN
122B  80 3E 0449 R 07          3253  C          CMP     CRT_MODE,7
1230  74 46                    3254  C          JE      V4                      ; ALPHA_PEN
                               3255  C
1232  80 3E 0449 R 06          3256  C          CMP     CRT_MODE,06H
1237  77 28                    3257  C          JA      V8
1239  75 02                    3258  C          JNE     V8X
123B  D1 E8                    3259  C          SHR     AX,1
                               3260  C
                               3261  C ;----- OLD GRAPHICS MODES
                               3262  C
123D                           3263  C  V8X:
123D  B2 28                    3264  C          MOV     DL,40                   ; DIVISOR FOR GRAPHICS
123F  F6 F2                    3265  C          DIV     DL                      ; ROW(AL) AND COLUMN(AH)
                               3266  C                                          ;  AL RANGE 0-99,
                               3267  C                                          ;  AH RANGE 0-39
                               3268  C ;----- DETERMINE GRAPHIC ROW POSITION
                               3269  C
1241  8A E8                    3270  C          MOV     CH,AL                   ; SAVE ROW VALUE IN CH
1243  02 ED                    3271  C          ADD     CH,CH                   ; *2 FOR EVEN/ODD FIELD
1245  8A DC                    3272  C          MOV     BL,AH                   ; COLUMN VALUE TO BX
1247  2A FF                    3273  C          SUB     BH,BH                   ; *8 FOR MEDIUM RES
1249  80 3E 0449 R 06          3274  C          CMP     CRT_MODE,6              ; MEDIUM OR HIGH RES
124E  75 04                    3275  C          JNE     V3                      ; NOT_HIGH_RES
1250  B1 04                    3276  C          MOV     CL,4                    ; SHIFT VALUE FOR HIGH RES
```

```
1252  D0 E4                    3277  C           SAL     AH,1                            ; COLUMN VALUE *2 FOR HIGH RES
1254                           3278  C  V3:                                              ; NOT_HIGH_RES
1254  D3 E3                    3279  C           SHL     BX,CL                           ; *16 FOR HIGH RES
                               3280  C
                               3281  C  ;----- DETERMINE ALPHA CHAR POSITION
                               3282  C
1256  8A D4                    3283  C           MOV     DL,AH                           ; COLUMN VALUE FOR RETURN
1258  8A F0                    3284  C           MOV     DH,AL                           ; ROW VALUE
125A  D0 EE                    3285  C           SHR     DH,1                            ; DIVIDE BY 4
125C  D0 EE                    3286  C           SHR     DH,1                            ;  FOR VALUE IN 0-24 RANGE
125E  EB 2C 90                 3287  C           JMP     V5                              ; LIGHT_PEN_RETURN_SET
1261                           3288  C  V8:
                               3289  C
                               3290  C  ;----- NEW GRAPHICS MODES
                               3291  C
1261  99                       3292  C           CWD                                     ; PREPARE TO DIVIDE
1262  F7 36 044A R             3293  C           DIV     CRT_COLS                        ; AX = ROW, DX = COLUMN
1266  8B DA                    3294  C           MOV     BX,DX                           ; SAVE REMAINDER
1268  D3 E3                    3295  C           SAL     BX,CL                           ; PEL COLUMN
126A  8B C8                    3296  C           MOV     CX,AX                           ; PEL ROW
126C  52                       3297  C           PUSH    DX                              ; SAVE FROM DIVIDE
126D  99                       3298  C           CWD                                     ; PREPARE TO DIVIDE
126E  F7 36 0485 R             3299  C           DIV     POINTS                          ; DIVIDE BY BYTES/CHAR
1272  5A                       3300  C           POP     DX                              ; RECOVER
1273  8A F0                    3301  C           MOV     DH,AL                           ; CHARACTER ROW
1275  EB 15 90                 3302  C           JMP     V5
                               3303  C
                               3304  C  ;----- ALPHA MODE ON LIGHT PEN
                               3305  C
1278                           3306  C  V4:                                              ; ALPHA_PEN
1278  F6 36 044A R             3307  C           DIV     BYTE PTR CRT_COLS               ; ROW,COLUMN VALUE
127C  8A F0                    3308  C           MOV     DH,AL                           ; ROWS TO DH
127E  8A D4                    3309  C           MOV     DL,AH                           ; COLS TO DL
1280  8A DC                    3310  C           MOV     BL,AH                           ; COLUMN VALUE
1282  32 FF                    3311  C           XOR     BH,BH                           ;  TO BX
1284  D3 E3                    3312  C           SAL     BX,CL
1286  F6 26 0485 R             3313  C           MUL     BYTE PTR POINTS
128A  8B C8                    3314  C           MOV     CX,AX
128C                           3315  C  V5:                                              ; LIGHT_PEN_RETURN_SET
128C  B4 01                    3316  C           MOV     AH,1                            ; INDICATE EVERTHING SET
128E                           3317  C  V6:                                              ; LIGHT_PEN_RETURN
128E  52                       3318  C           PUSH    DX                              ; SAVE RETURN VALUE
                               3319  C                                                   ;  (IN CASE)
128F  8B 16 0463 R             3320  C           MOV     DX,ADDR_6845                    ; GET BASE ADDRESS
1293  83 C2 07                 3321  C           ADD     DX,7                            ; POINT TO RESET PARM
                               3322  C           WOUT                                    ; ADDRESS, NOT DATA,
1296  EE                       3323  C+          OUT     DX,AL
                               3324  C                                                   ;  IS IMPORTANT
1297  5A                       3325  C           POP     DX                              ; RECOVER VALUE
1298                           3326  C  V7:                                              ; RETURN_NO_RESET
1298  5F                       3327  C           POP     DI
1299  5E                       3328  C           POP     SI
129A  83 C4 06                 3329  C           ADD     SP,6                            ; DISCARD SAVED BX,CX,DX
129D  1F                       3330  C           POP     DS
129E  07                       3331  C           POP     ES
129F  5D                       3332  C           POP     BP
12A0  CF                       3333  C           IRET
12A1                           3334  C  READ_LPEN        ENDP
                               3335  C
                               3336  C  ;--------------------------------------------------------------
                               3337  C  ; ACT_DISP_PAGE          SELECT ACTIVE DISPLAY PAGE          :
                               3338  C  ;       THIS ROUTINE SETS THE ACTIVE DISPLAY PAGE, ALLOWING  :
                               3339  C  ;       FOR MULTIPLE PAGES OF DISPLAYED VIDEO.               :
                               3340  C  ; INPUT                                                      :
                               3341  C  ;       AL HAS THE NEW ACTIVE DISPLAY PAGE                   :
                               3342  C  ; OUTPUT                                                     :
                               3343  C  ;       THE CRTC IS RESET TO DISPLAY THAT PAGE               :
                               3344  C  ;--------------------------------------------------------------
12A1                           3345  C  AH5:
12A1  A2 0462 R                3346  C           MOV     ACTIVE_PAGE,AL                  ; SAVE ACTIVE PAGE VALUE
12A4  8B 0E 044C R             3347  C           MOV     CX,CRT_LEN                      ; GET SAVED LENGTH OF
                               3348  C                                                   ;  REGEN BUFFER
12A8  98                       3349  C           CBW                                     ; CONVERT AL TO WORD
12A9  50                       3350  C           PUSH    AX                              ; SAVE PAGE VALUE
12AA  F7 E1                    3351  C           MUL     CX                              ; DISPLAY PAGE TIMES
                               3352  C                                                   ;  REGEN LENGTH
12AC  A3 044E R                3353  C           MOV     CRT_START,AX                    ; SAVE START ADDRESS FOR
                               3354  C                                                   ;  LATER REQUIREMENTS
12AF  8B C8                    3355  C           MOV     CX,AX                           ; START ADDRESS TO CX
12B1  8A 1E 0449 R             3356  C           MOV     BL,CRT_MODE
12B5  80 FB 07                 3357  C           CMP     BL,7                            ; DO NOT DIVIDE BY TWO FOR THE
12B8  77 02                    3358  C           ja      adp_1
12BA                           3359  C  ADP_2:
12BA  D1 F9                    3360  C           SAR     CX,1                            ; / 2 FOR CRTc HANDLING
12BC                           3361  C  ADP_1:
12BC  B4 0C                    3362  C           MOV     AH,C_STRT_HGH                   ; REG FOR START ADDRESS
12BE  E8 1132 R                3363  C           CALL    M16
12C1  5B                       3364  C           POP     BX                              ; RECOVER PAGE VALUE
12C2  D1 E3                    3365  C           SAL     BX,1                            ; *2 FOR WORD OFFSET
12C4  8B 87 0450 R             3366  C           MOV     AX,[BX + OFFSET CURSOR_POSN]    ; GET CURSOR FOR THIS PAGE
12C8  E8 1172 R                3367  C           CALL    M18                             ; SET THE CURSOR POSITION
12CB  E9 219B R                3368  C           JMP     V_RET
                               3369  C
                               3370  C           SUBTTL
                               3371
                               3372  C           INCLUDE VSCROLL.INC
                               3373  C           SUBTTL  VSCROLL.INC
                               3374  C           PAGE
                               3375  C
12CE                           3376  C  FLTA     PROC    NEAR                            ; CHECK FOR SCROLL COUNT
12CE  50                       3377  C           PUSH    AX
12CF  8A E6                    3378  C           MOV     AH,DH                           ; LOWER ROW
12D1  2A E5                    3379  C           SUB     AH,CH                           ; UPPER ROW
12D3  FE C4                    3380  C           INC     AH                              ; NUMBER TO SCROLL
12D5  3A E0                    3381  C           CMP     AH,AL                           ; SAME AS REQUESTED
12D7  58                       3382  C           POP     AX
12D8  75 02                    3383  C           JNE     LTA
12DA  2A C0                    3384  C           SUB     AL,AL                           ; YES, SET TO 0 FOR BLANK
12DC                           3385  C  LTA:
12DC  C3                       3386  C           RET
12DD                           3387  C  FLTA     ENDP
                               3388  C
12DD                           3389  C  CRANK    PROC    NEAR                            ; MOVE ROWS OF PELS UP
12DD  53                       3390  C           PUSH    BX
                               3391  C           ASSUME  DS:ABS0
12DE  1E                       3392  C           PUSH    DS                              ; SAVE DATA SEGMENT
12DF  E8 0D01 R                3393  C           CALL    DDS                             ; SET DATA SEGMENT
12E2  8B 1E 044A R             3394  C           MOV     BX,CRT_COLS
12E6  1F                       3395  C           POP     DS
12E7                           3396  C  CRANK_A:
12E7  51                       3397  C           PUSH    CX                              ; SAVE MOVE COUNT
12E8  8A CA                    3398  C           MOV     CL,DL                           ; COLUMN COUNT
12EA  2A ED                    3399  C           SUB     CH,CH                           ; CLEAR HIGH BYTE
12EC  56                       3400  C           PUSH    SI                              ; SAVE POINTERS
12ED  57                       3401  C           PUSH    DI
12EE  F3/ A4                   3402  C           REP     MOVSB                           ; MOVE THAT ROW
```

```
12F0  5F                    3403  C          POP     DI                      ; RECOVER POINTERS
12F1  5E                    3404  C          POP     SI
12F2  03 F3                 3405  C          ADD     SI,BX                   ; NEXT ROW
12F4  03 FB                 3406  C          ADD     DI,BX                   ; NEXT ROW
12F6  59                    3407  C          POP     CX                      ; RECOVER ROW COUNT
12F7  E2 EE                 3408  C          LOOP    CRANK_A                 ; DO MORE
12F9  5B                    3409  C          POP     BX
12FA  C3                    3410  C          RET                             ; RETURN TO CALLER
12FB                        3411  C  CRANK    ENDP
                            3412  C
12FB                        3413  C  CRANK_4  PROC    NEAR                    ; MOVE ROWS OF PELS DOWN
12FB  53                    3414  C          PUSH    BX
                            3415  C          ASSUME  DS:ABS0
12FC  1E                    3416  C          PUSH    DS                      ; SAVE DATA SEGMENT
12FD  E8 0D01 R             3417  C          CALL    DDS                     ; SET DATA SEGMENT
1300  8B 1E 044A R          3418  C          MOV     BX,CRT_COLS
1304  1F                    3419  C          POP     DS
1305                        3420  C  CRANK_B:
1305  51                    3421  C          PUSH    CX                      ; SAVE MOVE COUNT
1306  8A CA                 3422  C          MOV     CL,DL                   ; COLUMN COUNT
1308  2A ED                 3423  C          SUB     CH,CH                   ; CLEAR HIGH BYTE
130A  56                    3424  C          PUSH    SI                      ; SAVE POINTERS
130B  57                    3425  C          PUSH    DI
130C  F3/ A4                3426  C          REP     MOVSB                   ; MOVE THAT ROW
130E  5F                    3427  C          POP     DI                      ; RECOVER POINTERS
130F  5E                    3428  C          POP     SI
1310  2B F3                 3429  C          SUB     SI,BX                   ; NEXT ROW
1312  2B FB                 3430  C          SUB     DI,BX                   ; NEXT ROW
1314  59                    3431  C          POP     CX                      ; RECOVER ROW COUNT
1315  E2 EE                 3432  C          LOOP    CRANK_B                 ; DO MORE
1317  5B                    3433  C          POP     BX
1318  C3                    3434  C          RET                             ; RETURN TO CALLER
1319                        3435  C  CRANK_4  ENDP
                            3436  C
1319                        3437  C  PART_1   PROC    NEAR                    ; FILL ROW AFTER SCROLL
1319  52                    3438  C          PUSH    DX
131A  B6 03                 3439  C          mov     dh,3
131C  B2 C4                 3440  C          MOV     DL,SEQ_ADDR             ; SEQUENCER
131E  B8 020F               3441  C          MOV     AX,020FH                ; MAP MASK
1321  E8 0D18 R             3442  C          CALL    OUT_DX                  ; ALL MAPS ON
1324  5A                    3443  C          POP     DX
1325  2B C0                 3444  C          SUB     AX,AX                   ; ZERO
1327  8A CA                 3445  C          MOV     CL,DL                   ; COLUMN COUNT
1329  2A ED                 3446  C          sub     ch,ch
132B  57                    3447  C          PUSH    DI                      ; SAVE POINTER
132C  F3/ AA                3448  C          REP     STOSB                   ; CLEAR ONE ROW OF PELS
132E  5F                    3449  C          POP     DI                      ; RECOVER POINTER
132F  8A C6                 3450  C          MOV     AL,DH                   ; GET COLOR VALUE
1331  52                    3451  C          PUSH    DX
1332  B6 03                 3452  C          mov     dh,3
1334  B2 C4                 3453  C          MOV     DL,SEQ_ADDR             ; SEQUENCER
1336  B4 02                 3454  C          MOV     AH,02H                  ; MAP MASK
1338  E8 0D18 R             3455  C          CALL    OUT_DX                  ; SET THE COLOR
133B  5A                    3456  C          POP     DX
133C  B0 FF                 3457  C          MOV     AL,0FFH                 ; ALL BITS ON
133E  8A CA                 3458  C          MOV     CL,DL                   ; COLUMN COUNT
1340  57                    3459  C          PUSH    DI                      ; SAVE POINTER
1341  F3/ AA                3460  C          REP     STOSB                   ; TURN ON THOSE BITS IN
                            3461  C                                          ; ENABLED PLANES
1343  5F                    3462  C          POP     DI                      ; RECOVER POINTER
1344  C3                    3463  C          RET                             ; RETURN TO CALLER
1345                        3464  C  PART_1   ENDP
                            3465  C
1345                        3466  C  PART_2   PROC    NEAR
1345  B6 03                 3467  C          mov     dh,3
1347  B2 C4                 3468  C          MOV     DL,SEQ_ADDR             ; SEQUENCER
1349  B8 020F               3469  C          MOV     AX,020FH                ; MAP MASK, ALL MAPS
134C  E8 0D18 R             3470  C          CALL    OUT_DX                  ; ENABLE THE MAPS
134F  C3                    3471  C          RET                             ; RETURN TO CALLER
1350                        3472  C  PART_2   ENDP
                            3473  C
1350                        3474  C  BLNK_3   PROC    NEAR                    ; BLANK FOR SCROLL UP
1350  1E                    3475  C          PUSH    DS                      ; SAVE DATA SEGMENT
                            3476  C          ASSUME  DS:ABS0
1351  E8 0D01 R             3477  C          CALL    DDS                     ; GET LOW MEMORY SEGMENT
1354  8A F7                 3478  C          MOV     DH,BH                   ; ATTRIBUTE FOR BLANK LINE
1356  2A FF                 3479  C          SUB     BH,BH                   ; CLEAR HIGH BYTE
1358  50                    3480  C          PUSH    AX                      ; SAVE
1359  52                    3481  C          PUSH    DX                      ; SAVE BECAUSE OF MULTIPLY
135A  8B C3                 3482  C          MOV     AX,BX                   ; ROW COUNT
135C  F7 26 0485 R          3483  C          MUL     POINTS                  ; CHARACTER HEIGHT
1360  8B D8                 3484  C          MOV     BX,AX                   ; NET VALUE TO BX
1362  5A                    3485  C          POP     DX                      ; RECOVER
1363  58                    3486  C          POP     AX
                            3487  C
1364  1F                    3488  C          POP     DS
                            3489  C          ASSUME  DS:NOTHING
1365                        3490  C  S13:
1365  E8 1319 R             3491  C          CALL    PART_1                  ; BLANK OUT ROW WITH COLOR
                            3492  C          ASSUME  DS:ABS0
1368  1E                    3493  C          PUSH    DS                      ; SAVE SEGMENT
1369  E8 0D01 R             3494  C          CALL    DDS                     ; LOW MEMORY SEGMENT
136C  03 3E 044A R          3495  C          ADD     DI,CRT_COLS             ; NEXT ROW
1370  1F                    3496  C          POP     DS                      ; RECOVER
1371  4B                    3497  C          DEC     BX                      ; NEXT
1372  75 F1                 3498  C          JNZ     S13                     ; DO MORE
1374  E8 1345 R             3499  C          CALL    PART_2
1377  C3                    3500  C          RET                             ; RETURN TO CALLER
1378                        3501  C  BLNK_3   ENDP
                            3502  C
1378                        3503  C  BLNK_4   PROC    NEAR                    ; BLANK FOR SCROLL DOWN
1378  1E                    3504  C          PUSH    DS                      ; SAVE DATA SEGMENT
                            3505  C          ASSUME  DS:ABS0
1379  E8 0D01 R             3506  C          CALL    DDS                     ; GET LOW MEMORY SEGMENT
137C  8A F7                 3507  C          MOV     DH,BH                   ; ATTRIBUTE FOR BLANK LINE
137E  2A FF                 3508  C          SUB     BH,BH                   ; CLEAR HIGH BYTE
1380  50                    3509  C          PUSH    AX                      ; SAVE
1381  52                    3510  C          PUSH    DX                      ; SAVE BECAUSE OF MULTIPLY
1382  8B C3                 3511  C          MOV     AX,BX                   ; ROW COUNT
1384  F7 26 0485 R          3512  C          MUL     POINTS                  ; CHARACTER HEIGHT
1388  8B D8                 3513  C          MOV     BX,AX                   ; NET VALUE TO BX
138A  5A                    3514  C          POP     DX                      ; RECOVER
138B  58                    3515  C          POP     AX
                            3516  C
138C  1F                    3517  C          POP     DS
                            3518  C          ASSUME  DS:NOTHING
138D                        3519  C  S13_4:
138D  E8 1319 R             3520  C          CALL    PART_1                  ; BLANK OUT ROW WITH COLOR
                            3521  C          ASSUME  DS:ABS0
1390  1E                    3522  C          PUSH    DS                      ; SAVE SEGMENT
1391  E8 0D01 R             3523  C          CALL    DDS                     ; LOW MEMORY SEGMENT
1394  2B 3E 044A R          3524  C          SUB     DI,CRT_COLS             ; NEXT ROW
1398  1F                    3525  C          POP     DS                      ; RECOVER
1399  4B                    3526  C          DEC     BX                      ; NEXT
139A  75 F1                 3527  C          JNZ     S13_4                   ; DO MORE
139C  E8 1345 R             3528  C          CALL    PART_2
```

```
139F  C3                        3529  C           RET                                     ; RETURN TO CALLER
13A0                            3530  C  BLNK_4   ENDP
                                3531  C
                                3532  C  ;--------------------------------------------------------
                                3533  C  ; SCROLL UP                                              :
                                3534  C  ;       THIS ROUTINE MOVES A BLOCK OF CHARACTERS UP     :
                                3535  C  ;       ON THE SCREEN                                   :
                                3536  C  ; INPUT                                                 :
                                3537  C  ;       (AH) = CURRENT CRT MODE                         :
                                3538  C  ;       (AL) = NUMBER OF ROWS TO SCROLL                 :
                                3539  C  ;       (CX) = ROW/COLUMN OF UPPER LEFT CORNER          :
                                3540  C  ;       (DX) = ROW/COLUMN OF LOWER RIGHT CORNER         :
                                3541  C  ;       (BH) = ATTRIBUTE TO BE USED ON BLANKED LINE     :
                                3542  C  ;       (DS) = DATA SEGMENT                             :
                                3543  C  ;       (ES) = REGEN BUFFER SEGMENT                     :
                                3544  C  ; OUTPUT                                                :
                                3545  C  ;       NONE -- THE REGEN BUFFER IS MODIFIED            :
                                3546  C  ;--------------------------------------------------------
                                3547  C           ASSUME  CS:CODE,DS:ABS0,ES:NOTHING
13A0                            3548  C  SCROLL_UP        PROC    NEAR
13A0  8A D8                     3549  C           MOV     BL,AL                           ; SAVE LINE COUNT IN BL
13A2  E8 16E8 R                 3550  C           CALL    MK_ES
13A5  80 FC 04                  3551  C           CMP     AH,4                            ; TEST FOR GRAPHICS MODE
13A8  72 08                     3552  C           jb      n1                              ; handle seperately
13AA  80 FC 07                  3553  C           CMP     AH,7                            ; TEST FOR BW CARD
13AD  74 03                     3554  C           JE      N1
13AF  E9 1471 R                 3555  C           JMP     GRAPHICS_UP
13B2                            3556  C  N1:                                              ; UP_CONTINUE
13B2  53                        3557  C           PUSH    BX                              ; SAVE FILL ATTR IN BH
13B3  8B C1                     3558  C           MOV     AX,CX                           ; UPPER LEFT POSITION
13B5  E8 13EF R                 3559  C           CALL    SCROLL_POSITION                 ; DO SETUP FOR SCROLL
13B8  74 31                     3560  C           JZ      N7                              ; BLANK_FIELD
13BA  03 F0                     3561  C           ADD     SI,AX                           ; FROM ADDRESS
13BC  8A E6                     3562  C           MOV     AH,DH                           ; # ROWS IN BLOCK
13BE  2A E3                     3563  C           SUB     AH,BL                           ; # ROWS TO BE MOVED
13C0                            3564  C  N2:                                              ; ROW_LOOP
13C0  E8 142F R                 3565  C           CALL    N10                             ; MOVE ONE ROW
13C3  03 F5                     3566  C           ADD     SI,BP
13C5  03 FD                     3567  C           ADD     DI,BP                           ; NEXT LINE IN BLOCK
13C7  FE CC                     3568  C           DEC     AH                              ; COUNT OF LINES TO MOVE
13C9  75 F5                     3569  C           JNZ     N2                              ; ROW_LOOP
13CB                            3570  C  N3:                                              ; CLEAR_ENTRY
13CB  58                        3571  C           POP     AX                              ; RECOVER ATTRIBUTE IN AH
13CC  B0 20                     3572  C           MOV     AL,' '                          ; FILL WITH BLANKS
13CE                            3573  C  N4:                                              ; CLEAR_LOOP
13CE  E8 1438 R                 3574  C           CALL    N11                             ; CLEAR THE ROW
13D1  03 FD                     3575  C           ADD     DI,BP                           ; POINT TO NEXT LINE
13D3  FE CB                     3576  C           DEC     BL                              ; LINES TO SCROLL
13D5  75 F7                     3577  C           JNZ     N4                              ; CLEAR_LOOP
13D7                            3578  C  N5:                                              ; SCROLL_END
13D7  E8 0D01 R                 3579  C           CALL    DDS
13DA  80 3E 0449 R 07           3580  C           CMP     CRT_MODE,7                      ; IS THIS THE B/W CARD
13DF  74 07                     3581  C           JE      N6                              ; SKIP THE MODE RESET
13E1  A0 0465 R                 3582  C           MOV     AL,CRT_MODE_SET                 ; GET THE MODE SET
13E4  BA 03D8                   3583  C           MOV     DX,03D8H                        ; ALWAYS SET COLOR CARD
                                3584  C           WOUT
13E7 ,EE                        3585  C+          OUT     DX,AL
13E8                            3586  C  N6:                                              ; VIDEO_RET_HERE
13E8  E9 219B R                 3587  C           JMP     V_RET
13EB                            3588  C  N7:                                              ; BLANK_FIELD
13EB  8A DE                     3589  C           MOV     BL,DH                           ; GET ROW COUNT
13ED  EB DC                     3590  C           JMP     N3                              ; GO CLEAR THAT AREA
13EF                            3591  C  SCROLL_UP        ENDP
                                3592  C
                                3593  C  ;----- HANDLE COMMON SCROLL SET UP HERE
                                3594  C
13EF                            3595  C  SCROLL_POSITION PROC    NEAR
13EF  F6 06 0487 R 04           3596  C           test    info,4
13F4  74 12                     3597  C           jz      n9
                                3598  C
                                3599  C  ;----- 80X25 COLOR CARD SCROLL
                                3600  C
13F6  52                        3601  C           PUSH    DX
13F7  B6 03                     3602  C           mov     dh,3
13F9  B2 DA                     3603  C           MOV     DL,0DAH                         ; COLOR CARD HERE
13FB  50                        3604  C           PUSH    AX
13FC                            3605  C  N8:                                              ; WAIT_DISP_ENABLE
                                3606  C           WIN
13FC  EC                        3607  C+          IN      AL,DX
13FD  A8 08                     3608  C           TEST    AL,8                            ; WAIT FOR VERT RETRACE
13FF  74 FB                     3609  C           JZ      N8                              ; WAIT_DISP_ENABLE
1401  B0 25                     3610  C           MOV     AL,25H
1403  B2 D8                     3611  C           MOV     DL,0D8H                         ; DX=3D8
                                3612  C           WOUT                                    ; TURN OFF VIDEO
1405  EE                        3613  C+          OUT     DX,AL
1406  58                        3614  C           POP     AX                              ;  DURING VERTICAL RETRACE
1407  5A                        3615  C           POP     DX
1408                            3616  C  N9:
1408  E8 1143 R                 3617  C           CALL    POSITION                        ; CONVERT TO REGEN POINTER
140B  03 06 044E R              3618  C           ADD     AX,CRT_START                    ; OFFSET OF ACTIVE PAGE
140F  8B F8                     3619  C           MOV     DI,AX                           ; TO ADDRESS FOR SCROLL
1411  8B F0                     3620  C           MOV     SI,AX                           ; FROM ADDRESS FOR SCROLL
1413  2B D1                     3621  C           SUB     DX,CX                           ; DX = #ROWS, #COLS
1415  FE C6                     3622  C           INC     DH
1417  FE C2                     3623  C           INC     DL                              ; INCREMENT FOR 0 ORIGIN
1419  32 ED                     3624  C           XOR     CH,CH                           ; ZERO HIGH BYTE OF COUNT
141B  8B 2E 044A R              3625  C           MOV     BP,CRT_COLS                     ; NUM OF COLS IN DISPLAY
141F  03 ED                     3626  C           ADD     BP,BP                           ; TIMES 2 FOR ATTR BYTE
1421  8A C3                     3627  C           MOV     AL,BL                           ; GET LINE COUNT
1423  F6 26 044A R              3628  C           MUL     BYTE PTR CRT_COLS               ; OFFSET TO FROM ADDRESS
1427  03 C0                     3629  C           ADD     AX,AX                           ; *2 FOR ATTRIBUTE BYTE
1429  06                        3630  C           PUSH    ES                              ; ESTABLISH ADDRESSING
142A  1F                        3631  C           POP     DS                              ;  FOR BOTH POINTERS
142B  80 FB 00                  3632  C           CMP     BL,0                            ; 0 MEANS BLANK FIELD
142E  C3                        3633  C           RET                                     ; RETURN WITH FLAGS SET
142F                            3634  C  SCROLL_POSITION ENDP
                                3635  C
                                3636  C  ;----- MOVE_ROW
                                3637  C
142F                            3638  C  N10     PROC    NEAR
142F  8A CA                     3639  C           MOV     CL,DL                           ; GET # OF COLS TO MOVE
1431  56                        3640  C           PUSH    SI
1432  57                        3641  C           PUSH    DI                              ; SAVE START ADDRESS
1433  F3/ A5                    3642  C           REP     MOVSW                           ; MOVE THAT LINE ON SCREEN
1435  5F                        3643  C           POP     DI
1436  5E                        3644  C           POP     SI                              ; RECOVER ADDRESSES
1437  C3                        3645  C           RET
1438                            3646  C  N10     ENDP
                                3647  C
                                3648  C  ;----- CLEAR_ROW
                                3649  C
1438                            3650  C  N11     PROC    NEAR
1438  8A CA                     3651  C           MOV     CL,DL                           ; GET # COLUMNS TO CLEAR
143A  57                        3652  C           PUSH    DI
143B  F3/ AB                    3653  C           REP     STOSW                           ; STORE THE FILL CHARACTER
143D  5F                        3654  C           POP     DI
```

```
143E  C3                        3655     C          RET
143F                            3656     C  N11     ENDP
                                3657     C
                                3658     C  ;--------------------------------------------------------
                                3659     C  ; SCROLL_DOWN                                          :
                                3660     C  ;       THIS ROUTINE MOVES THE CHARACTERS WITHIN A     :
                                3661     C  ;       DEFINED BLOCK DOWN ON THE SCREEN, FILLING THE  :
                                3662     C  ;       TOP LINES WITH A DEFINED CHARACTER             :
                                3663     C  ; INPUT                                                :
                                3664     C  ;       (AH) = CURRENT CRT MODE                        :
                                3665     C  ;       (AL) = NUMBER OF LINES TO SCROLL               :
                                3666     C  ;       (CX) = UPPER LEFT CORNER OF REGION             :
                                3667     C  ;       (DX) = LOWER RIGHT CORNER OF REGION            :
                                3668     C  ;       (BH) = FILL CHARACTER                          :
                                3669     C  ;       (DS) = DATA SEGMENT                            :
                                3670     C  ;       (ES) = REGEN SEGMENT                           :
                                3671     C  ; OUTPUT                                               :
                                3672     C  ;       NONE -- SCREEN IS SCROLLED                     :
                                3673     C  ;--------------------------------------------------------
143F                            3674     C  SCROLL_DOWN     PROC    NEAR
143F  FD                        3675     C          STD                             ; SCROLL DOWN
1440  8A D8                     3676     C          MOV     BL,AL                   ; LINE COUNT TO BL
1442  E8 16E8 R                 3677     C          CALL    MK_ES
1445  53                        3678     C          PUSH    BX                      ; SAVE ATTRIBUTE IN BH
1446  8B C2                     3679     C          MOV     AX,DX                   ; LOWER RIGHT CORNER
1448  E8 13EF R                 3680     C          CALL    SCROLL_POSITION         ; GET REGEN LOCATION
144B  74 20                     3681     C          JZ      N16
144D  2B F0                     3682     C          SUB     SI,AX                   ; SI IS FROM ADDRESS
144F  8A E6                     3683     C          MOV     AH,DH                   ; GET TOTAL # ROWS
1451  2A E3                     3684     C          SUB     AH,BL                   ; COUNT TO MOVE IN SCROLL
1453                            3685     C  N13:
1453  E8 142F R                 3686     C          CALL    N10                     ; MOVE ONE ROW
1456  2B F5                     3687     C          SUB     SI,BP
1458  2B FD                     3688     C          SUB     DI,BP
145A  FE CC                     3689     C          DEC     AH
145C  75 F5                     3690     C          JNZ     N13
145E                            3691     C  N14:
145E  58                        3692     C          POP     AX                      ; RECOVER ATTRIBUTE IN AH
145F  B0 20                     3693     C          MOV     AL,' '
1461                            3694     C  N15:
1461  E8 1438 R                 3695     C          CALL    N11                     ; CLEAR ONE ROW
1464  2B FD                     3696     C          SUB     DI,BP                   ; GO TO NEXT ROW
1466  FE CB                     3697     C          DEC     BL
1468  75 F7                     3698     C          JNZ     N15
146A  E9 13D7 R                 3699     C          JMP     N5                      ; SCROLL_END
146D                            3700     C  N16:
146D  8A DE                     3701     C          MOV     BL,DH
146F  EB ED                     3702     C          JMP     N14
1471                            3703     C  SCROLL_DOWN     ENDP
                                3704     C
                                3705     C  ;--------------------------------------------------------
                                3706     C  ; SCROLL UP                                            :
                                3707     C  ;       THIS ROUTINE SCROLLS UP THE INFORMATION ON THE CRT :
                                3708     C  ; ENTRY                                                :
                                3709     C  ;       CH,CL = UPPER LEFT CORNER OF REGION TO SCROLL  :
                                3710     C  ;       DH,DL = LOWER RIGHT CORNER OF REGION TO SCROLL :
                                3711     C  ;         BOTH OF THE ABOVE ARE IN CHARACTER POSITIONS  :
                                3712     C  ;       BH = FILL VALUE FOR BLANKED LINES              :
                                3713     C  ;       AL = # LINES TO SCROLL (AL=0 MEANS BLANK THE ENTIRE :
                                3714     C  ;            FIELD)                                    :
                                3715     C  ;       DS = DATA SEGMENT                              :
                                3716     C  ;       ES = REGEN SEGMENT                             :
                                3717     C  ; EXIT                                                 :
                                3718     C  ;       NOTHING, THE SCREEN IS SCROLLED                :
                                3719     C  ;--------------------------------------------------------
1471                            3720     C  GRAPHICS_UP     PROC    NEAR
1471  8A D8                     3721     C          MOV     BL,AL                   ; SAVE LINE COUNT IN BL
1473  8B C1                     3722     C          MOV     AX,CX                   ; GET UPPER LEFT POSITION
                                3723     C                                          ;  INTO AX REG
                                3724     C
                                3725     C  ;----- USE CHARACTER SUBROUTINE FOR POSITIONING
                                3726     C  ;----- ADDRESS RETURNED IS MULTIPLIED BY 2 FROM CORRECT VALUE
                                3727     C
1475  E8 16A4 R                 3728     C          CALL    GRAPH_POSN
1478  8B F8                     3729     C          MOV     DI,AX                   ; SAVE RESULT AS
                                3730     C                                          ;  DESTINATION ADDRESS
                                3731     C
                                3732     C  ;----- DETERMINE SIZE OF WINDOW
                                3733     C
147A  2B D1                     3734     C          SUB     DX,CX
147C  81 C2 0101                3735     C          ADD     DX,101H                 ; ADJUST VALUES
1480  D0 E6                     3736     C          SAL     DH,1                    ; MULTIPLY # ROWS BY 4
                                3737     C                                          ;  SINCE 8 VERT DOTS/CHAR
1482  D0 E6                     3738     C          SAL     DH,1                    ;  AND EVEN/ODD ROWS
                                3739     C
                                3740     C  ;----- DETERMINE CRT MODE
                                3741     C
1484  80 3E 0449 R 06           3742     C          CMP     CRT_MODE,6              ; TEST FOR MEDIUM RES
1489  73 04                     3743     C          JNC     R7                      ; FIND_SOURCE
                                3744     C
                                3745     C  ;----- MEDIUM RES UP
                                3746     C
148B  D0 E2                     3747     C          SAL     DL,1                    ; * 2,
148D  D1 E7                     3748     C          SAL     DI,1                    ;  SINCE 2 BYTES/CHAR
                                3749     C
                                3750     C  ;----- DETERMINE THE SOURCE ADDRESS IN THE BUFFER
                                3751     C
148F                            3752     C  R7:                                     ; FIND_SOURCE
148F  06                        3753     C          PUSH    ES                      ; GET SEGMENTS BOTH
1490  1F                        3754     C          POP     DS                      ;  POINTING TO REGEN
1491  2A ED                     3755     C          SUB     CH,CH                   ; 0 TO HIGH OF COUNT REG
1493  D0 E3                     3756     C          SAL     BL,1                    ; NUMBER OF LINES *4
1495  D0 E3                     3757     C          SAL     BL,1
1497  74 2D                     3758     C          JZ      R11                     ; IF 0, BLANK ENTIRE FIELD
1499  8A C3                     3759     C          MOV     AL,BL                   ; NUMBER OF LINES IN AL
149B  B4 50                     3760     C          MOV     AH,80                   ; 80 BYTES/ROW
149D  F6 E4                     3761     C          MUL     AH                      ; OFFSET TO SOURCE
149F  8B F7                     3762     C          MOV     SI,DI                   ; SET UP SOURCE
14A1  03 F0                     3763     C          ADD     SI,AX                   ;  ADD IN OFFSET TO IT
14A3  8A E6                     3764     C          MOV     AH,DH                   ; NUMBER OF ROWS IN FIELD
14A5  2A E3                     3765     C          SUB     AH,BL                   ; DETERMINE NUMBER TO MOVE
                                3766     C
                                3767     C  ;----- LOOP THROUGH, MOVING ONE ROW AT A TIME, BOTH EVEN AND ODD FIELDS
                                3768     C
14A7                            3769     C  R8:                                     ; ROW_LOOP
14A7  E8 14CA R                 3770     C          CALL    R17                     ; MOVE ONE ROW
14AA  81 EE 1FB0                3771     C          SUB     SI,2000H-80             ; MOVE TO NEXT ROW
14AE  81 EF 1FB0                3772     C          SUB     DI,2000H-80
14B2  FE CC                     3773     C          DEC     AH                      ; NUMBER OF ROWS TO MOVE
14B4  75 F1                     3774     C          JNZ     R8                      ; CONTINUE TILL ALL MOVED
                                3775     C
                                3776     C  ;----- FILL IN THE VACATED LINE(S)
                                3777     C
14B6                            3778     C  R9:                                     ; CLEAR_ENTRY
14B6  8A C7                     3779     C          MOV     AL,BH                   ; ATTRIBUTE TO FILL WITH
14B8                            3780     C  R10:
```

```
14B8  E8 14E3 R           3781  C          CALL    R18                     ; CLEAR THAT ROW
14BB  81 EF 1FB0          3782  C          SUB     DI,2000H-80             ; POINT TO NEXT LINE
14BF  FE CB               3783  C          DEC     BL                      ; NUMBER OF LINES TO FILL
14C1  75 F5               3784  C          JNZ     R10                     ; CLEAR_LOOP
14C3  E9 219B R           3785  C          JMP     V_RET
14C6                      3786  C  R11:                                    ; BLANK_FIELD
14C6  8A DE               3787  C          MOV     BL,DH                   ; SET BLANK COUNT TO
                          3788  C                                          ;  EVERYTHING IN FIELD
14C8  EB EC               3789  C          JMP     R9                      ; CLEAR THE FIELD
14CA                      3790  C  GRAPHICS_UP     ENDP
                          3791  C
                          3792  C  ;----- ROUTINE TO MOVE ONE ROW OF INFORMATION
                          3793  C
14CA                      3794  C  R17      PROC    NEAR
14CA  8A CA               3795  C          MOV     CL,DL                   ; NUM OF BYTES IN THE ROW
14CC  56                  3796  C          PUSH    SI
14CD  57                  3797  C          PUSH    DI                      ; SAVE POINTERS
14CE  F3/ A4              3798  C          REP     MOVSB                   ; MOVE THE EVEN FIELD
14D0  5F                  3799  C          POP     DI
14D1  5E                  3800  C          POP     SI
14D2  81 C6 2000          3801  C          ADD     SI,2000H
14D6  81 C7 2000          3802  C          ADD     DI,2000H                ; POINT TO THE ODD FIELD
14DA  56                  3803  C          PUSH    SI
14DB  57                  3804  C          PUSH    DI                      ; SAVE THE POINTERS
14DC  8A CA               3805  C          MOV     CL,DL                   ; COUNT BACK
14DE  F3/ A4              3806  C          REP     MOVSB                   ; MOVE THE ODD FIELD
14E0  5F                  3807  C          POP     DI
14E1  5E                  3808  C          POP     SI                      ; POINTERS BACK
14E2  C3                  3809  C          RET                             ; RETURN TO CALLER
14E3                      3810  C  R17      ENDP
                          3811  C
                          3812  C  ;----- CLEAR A SINGLE ROW
                          3813  C
14E3                      3814  C  R18      PROC    NEAR
14E3  8A CA               3815  C          MOV     CL,DL                   ; NUMBER OF BYTES IN FIELD
14E5  57                  3816  C          PUSH    DI                      ; SAVE POINTER
14E6  F3/ AA              3817  C          REP     STOSB                   ; STORE THE NEW VALUE
14E8  5F                  3818  C          POP     DI                      ; POINTER BACK
14E9  81 C7 2000          3819  C          ADD     DI,2000H                ; POINT TO ODD FIELD
14ED  57                  3820  C          PUSH    DI
14EE  8A CA               3821  C          MOV     CL,DL
14F0  F3/ AA              3822  C          REP     STOSB                   ; FILL THE ODD FILELD
14F2  5F                  3823  C          POP     DI
14F3  C3                  3824  C          RET                             ; RETURN TO CALLER
14F4                      3825  C  R18      ENDP
                          3826  G
14F4                      3827  C  mem_det proc    near
                          3828  C          assume  ds:abs0
14F4  50                  3829  C          push    ax
14F5  1E                  3830  C          push    ds
14F6  E8 0D01 R           3831  C          call    dds
14F9  8A 26 0487 R        3832  C          mov     ah,info
14FD  80 E4 60            3833  C          and     ah,060h
1500  1F                  3834  C          pop     ds
1501  58                  3835  C          pop     ax
1502  74 02              3836  C          jz      min
1504  F9                  3837  C          stc
1505  C3                  3838  C          ret
1506                      3839  C  min:
1506  F8                  3840  C          clc
1507  C3                  3841  C          ret
1508                      3842  C  mem_det endp
                          3843  C
                          3844  C  ;----- SCROLL ACTIVE PAGE UP
                          3845  C
1508                      3846  C  SC_2:
1508  E9 13A0 R           3847  C          JMP     SCROLL_UP
                          3848  C
150B                      3849  C  AH6:
                          3850  C          ASSUME  DS:ABS0
150B  E8 12CE R           3851  C          CALL    FLTA
150E  8A 26 0449 R        3852  C          MOV     AH,CRT_MODE             ; GET CURRENT MODE
1512  80 FC 07            3853  C          CMP     AH,07H
1515  76 F1              3854  C          JBE     SC_2                    ; ANY OF THE OLD MODES
1517  80 FC 0D            3855  C          CMP     AH,0DH
151A  73 17              3856  C          JAE     GRAPHICS_UP_2           ; NEW GRAPHICS MODES
151C  E9 219B R           3857  C          JMP     V_RET                   ; NOT A RECOGNIZED MODE
                          3858  G
151F                      3859  C  GR_ST_1 PROC    NEAR
151F  BA A000             3860  C          MOV     DX,0A000H               ; REGEN BUFFER
1522  BD 0511             3861  C          MOV     BP,0511H                ; GRAPHICS WRITE MODE
1525  80 FC 0F            3862  C          cmp     ah,0fh
1528  72 08              3863  G          jb      vv1
152A  E8 14F4 R           3864  G          call    mem_det
152D  73 03              3865  C          jnc     vv1
152F  BD 0501             3866  C          MOV     BP,0501H                ; GRAPHICS WRITE MODE
1532                      3867  C  VV1:
1532  C3                  3868  C          RET
1533                      3869  C  GR_ST_1 ENDP
                          3870  C
1533                      3871  C  GRAPHICS_UP_2   PROC    NEAR
                          3872  C          ASSUME  DS:ABS0
1533  52                  3873  C          PUSH    DX
1534  E8 151F R           3874  C          CALL    GR_ST_1                 ; SET SEGMENT, WRITE MODE
                          3875  C          SRLOAD  ES                      ; SET REGEN
1537  8E C2               3876  C+         MOV     ES,DX
1539  5A                  3877  C          POP     DX
153A  8A D8               3878  C          MOV     BL,AL                   ; NUMBER OF LINES
153C  8B C1               3879  C          MOV     AX,CX                   ; UPPER LEFT CORNER
153E  53                  3880  C          PUSH    BX
153F  8A 3E 0462 R        3881  C          MOV     BH,ACTIVE_PAGE          ; ACTIVE PAGE FOR SCROLL
1543  E8 16C3 R           3882  C          CALL    ORX_PSN                 ; ADDRESS IN REGEN
1546  5B                  3883  C          POP     BX
1547  8B F8               3884  C          MOV     DI,AX                   ; SET POINTER
1549  2B D1               3885  C          SUB     DX,CX                   ; DETERMINE WINDOW
154B  81 C2 0101          3886  C          ADD     DX,0101H                ; ADJUST
154F  2A E4               3887  C          SUB     AH,AH                   ; ZERO HIGH BYTE
1551  8A C3               3888  C          MOV     AL,BL                   ; LINE COUNT
1553  52                  3889  C          PUSH    DX
1554  F7 26 0485 R        3890  C          MUL     POINTS                  ; BYTES PER CHARACTER
1558  F7 26 044A R        3891  C          MUL     CRT_COLS                ; COLUMNS
155C  8B F7               3892  C          MOV     SI,DI                   ; SET UP SOURCE INDEX
155E  03 F0               3893  C          ADD     SI,AX                   ; ADJUST
                          3894  C          ASSUME  DS:NOTHING
1560  06                  3895  C          PUSH    ES
1561  1F                  3896  C          POP     DS
1562  5A                  3897  C          POP     DX
1563  0A DB               3898  C          OR      BL,BL                   ; LINE COUNT
1565  74 3F               3899  C          JZ      AR9
1567  8A CE               3900  C          MOV     CL,DH
1569  2A CB               3901  C          SUB     CL,BL
156B  2A ED               3902  C          SUB     CH,CH
                          3903  C
                          3904  C          ASSUME  DS:ABS0
156D  1E                  3905  C          PUSH    DS
156E  E8 0D01 R           3906  C          CALL    DDS                     ; LOW MEMORY SEGMENT
```

```
1571  50                    
1572  52                    
1573  8B C1                 
1575  F7 26 0485 R          
1579  8B C8                 
157B  5A                    
157C  58                    

157D  1F                    

157E  52                    
157F  8B C5                 
1581  B6 03                 
1583  B2 CE                 
1585  E8 0D18 R             
1588  B2 C4                 
158A  B8 020F               
158D  E8 0D18 R             
1590  5A                    
1591  E8 12DD R             

1594  52                    
1595  4D                    
1596  8B C5                 
1598  B6 03                 
159A  B2 CE                 
159C  E8 0D18 R             
159F  5A                    
15A0                        
15A0  E8 1350 R             
15A3  E9 219B R             
15A6                        
15A6  8A DE                 
15A8  EB F6                 
15AA                        


15AA                        
15AA  E9 143F R             

15AD                        

15AD  E8 12CE R             
15B0  8A 26 0449 R          
15B4  80 FC 03              
15B7  76 F1                 
15B9  80 FC 07              
15BC  74 EC                 

15BE  80 FC 0D              
15C1  73 0C                 
15C3  80 FC 06              
15C6  77 04                 
15C8  B4 07                 
15CA  CD 42                 
15CC                        
15CC  E9 219B R             

15CF                        
15CF  FD                    
15D0  8A D8                 
15D2  52                    
15D3  E8 151F R             

15D6  8E C2                 
15D8  5A                    
15D9  8B C2                 
15DB  FE C4                 
15DD  53                    
15DE  8A 3E 0462 R          
15E2  E8 16C3 R             
15E5  5B                    
15E6  2B 06 044A R          
15EA  8B F8                 
15EC  2B D1                 
15EE  81 C2 0101            
15F2  2A E4                 
15F4  8A C3                 
15F6  52                    
15F7  F7 26 0485 R          
15FB  F7 26 044A R          
15FF  8B F7                 
1601  2B F0                 

1603  06                    
1604  1F                    
1605  5A                    
1606  0A DB                 
1608  74 40                 
160A  8A CE                 
160C  2A CB                 
160E  2A ED                 


1610  1E                    
1611  E8 0D01 R             
1614  50                    
1615  52                    
1616  8B C1                 
1618  F7 26 0485 R          
161C  8B C8                 
161E  5A                    
161F  58                    

1620  1F                    

1621  52                    
1622  8B C5                 
1624  B6 03                 
1626  B2 CE                 
1628  E8 0D18 R             
162B  B2 C4                 
162D  B8 020F               
1630  E8 0D18 R             
1633  5A                    
1634  E8 12FB R             

1637  52                    
1638  4D                    
1639  8B C5                 
163B  B6 03                 
163D  B2 CE                 
163F  E8 0D18 R             
1642  5A                    
1643                        
```

```
3907  C           PUSH    AX
3908  C           PUSH    DX
3909  C           MOV     AX,CX
3910  C           MUL     POINTS                  ; BYTES PER CHAR
3911  C           MOV     CX,AX                   ; SET THE COUNT
3912  C           POP     DX
3913  C           POP     AX
3914  C           ASSUME  DS:NOTHING
3915  C           POP     DS
3916  C
3917  C           PUSH    DX
3918  C           MOV     AX,BP
3919  C           mov     dh,3
3920  C           MOV     DL,GRAPH_ADDR           ; GRAPHICS
3921  C           CALL    OUT_DX
3922  C           MOV     DL,SEQ_ADDR             ; SEQUENCER
3923  C           MOV     AX,020FH                ; ENABLE ALL MAPS
3924  C           CALL    OUT_DX
3925  C           POP     DX
3926  C           CALL    CRANK                   ; SCROLL THE SCREEN
3927  C
3928  C           PUSH    DX
3929  C           DEC     BP
3930  C           MOV     AX,BP
3931  C           mov     dh,3
3932  C           MOV     DL,GRAPH_ADDR
3933  C           CALL    OUT_DX
3934  C           POP     DX
3935  C  AR10:
3936  C           CALL    BLNK_3
3937  C           JMP     V_RET
3938  C  AR9:
3939  C           MOV     BL,DH                   ; BLANK ENTIRE WINDOW
3940  C           JMP     AR10
3941  C  GRAPHICS_UP_2    ENDP
3942  C
3943  C  ;----- SCROLL ACTIVE DISPLAY PAGE DOWN
3944  C
3945  C  SC_3:
3946  C           JMP     SCROLL_DOWN
3947  C
3948  C  AH7:
3949  C           ASSUME  DS:ABS0
3950  C           CALL    FLTA
3951  C           MOV     AH,CRT_MODE
3952  C           CMP     AH,03H                  ; OLD COLOR ALPHA
3953  C           JBE     SC_3
3954  C           CMP     AH,07H                  ; MONOCHROME ALPHA
3955  C           JE      SC_3
3956  C
3957  C           CMP     AH,0DH                  ; NEW GRAPHICS MODES
3958  C           JAE     GRAPHICS_DN_2
3959  C           CMP     AH,06H                  ; OLD GRAPHICS MODES
3960  C           JA      M_0
3961  C           MOV     AH,07H
3962  C           INT     42H
3963  C  M_0:
3964  C           JMP     V_RET
3965  C
3966  C  GRAPHICS_DN_2    PROC    NEAR
3967  C           STD                             ; DIRECTION TO DECREMENT
3968  C           MOV     BL,AL                   ; LINE COUNT
3969  C           PUSH    DX                      ; SAVE LOWER RIGHT
3970  C           CALL    GR_ST_1
3971  C           SRLOAD  ES                      ; SET REGEN SEGMENT
3972  C+          MOV     ES,DX
3973  C           POP     DX
3974  C           MOV     AX,DX
3975  C           INC     AH                      ; MOV CHAR ROW UP BY ONE
3976  C           PUSH    BX
3977  C           MOV     BH,ACTIVE_PAGE
3978  C           CALL    GRX_PSN                 ; ADDRESS IN REGEN
3979  C           POP     BX
3980  C           SUB     AX,CRT_COLS             ; ONE SCAN OVERSHOOT
3981  C           MOV     DI,AX
3982  C           SUB     DX,CX                   ; CALCULATE WINDOW
3983  C           ADD     DX,0101H                ; ADJUST COUNT
3984  C           SUB     AH,AH
3985  C           MOV     AL,BL
3986  C           PUSH    DX
3987  C           MUL     POINTS                  ; BYTES PER CHAR
3988  C           MUL     CRT_COLS                ; BYTES PER ROW
3989  C           MOV     SI,DI
3990  C           SUB     SI,AX
3991  C           ASSUME  DS:NOTHING
3992  C           PUSH    ES                      ; SET DS TO
3993  C           POP     DS                      ;  THE REGEN SEGMENT
3994  C           POP     DX
3995  C           OR      BL,BL                   ; SCROLL COUNT
3996  C           JZ      DXR9                    ; BLANK ENTIRE WINDOW
3997  C           MOV     CL,DH
3998  C           SUB     CL,BL
3999  C           SUB     CH,CH
4000  C
4001  C           ASSUME  DS:ABS0
4002  C           PUSH    DS
4003  C           CALL    DDS
4004  C           PUSH    AX
4005  C           PUSH    DX
4006  C           MOV     AX,CX
4007  C           MUL     POINTS                  ; BYTES PER CHAR
4008  C           MOV     CX,AX
4009  C           POP     DX
4010  C           POP     AX
4011  C           ASSUME  DS:NOTHING
4012  C           POP     DS
4013  C
4014  C           PUSH    DX
4015  C           MOV     AX,BP
4016  C           mov     dh,3
4017  C           MOV     DL,GRAPH_ADDR           ; GRAPHICS
4018  C           CALL    OUT_DX
4019  C           MOV     DL,SEQ_ADDR             ; SEQUENCER
4020  C           MOV     AX,020FH                ; ENABLE ALL MAPS
4021  C           CALL    OUT_DX
4022  C           POP     DX
4023  C           CALL    CRANK_4                 ; SCROLL THE SCREEN
4024  C
4025  C           PUSH    DX
4026  C           DEC     BP
4027  C           MOV     AX,BP
4028  C           mov     dh,3
4029  C           MOV     DL,GRAPH_ADDR
4030  C           CALL    OUT_DX
4031  C           POP     DX
4032  C  DXR10:
```

```
1643  E8 1378 R                 4033  C           CALL    BLNK_4
1646  FC                        4034  C           CLD
1647  E9 219B R                 4035  C           JMP     V_RET
164A                            4036  C   DXR9:
164A  8A DE                     4037  C           MOV     BL,DH                       ; BLANK ENTIRE WINDOW
164C  EB F5                     4038  C           JMP     DXR10
164E                            4039  C   GRAPHICS_DN_2   ENDP
                                4040  C
                                4041  C           SUBTTL
                                4042
                                4043  C           INCLUDE         VGRW.INC
                                4044  C           SUBTTL  VGRW.INC
                                4045  C           PAGE
                                4046  C
                                4047  C           ASSUME  DS:ABS0
164E                            4048  C   FIND_POSITION   PROC    NEAR
164E  8A CF                     4049  C           MOV     CL,BH                       ; DISPLAY PAGE TO CX
1650  32 ED                     4050  C           XOR     CH,CH
1652  8B F1                     4051  C           MOV     SI,CX                       ; MOVE TO SI FOR INDEX
1654  D1 E6                     4052  C           SAL     SI,1                        ; * 2 FOR WORD OFFSET
1656  8B 84 0450 R              4053  C           MOV     AX,[SI+ OFFSET CURSOR_POSN] ; ROW/COLUMN OF THAT PAGE
165A  33 DB                     4054  C           XOR     BX,BX                       ; SET START ADDRESS TO 0
165C  E3 06                     4055  C           JCXZ    P5                          ; NO_PAGE
165E                            4056  C   P4:                                       ; PAGE_LOOP
165E  03 1E 044C R              4057  C           ADD     BX,CRT_LEN                  ; LENGTH OF BUFFER
1662  E2 FA                     4058  C           LOOP    P4
1664                            4059  C   P5:                                       ; NO_PAGE
1664  E8 1143 R                 4060  C           CALL    POSITION                    ; DETERMINE LOC IN REGEN
1667  03 D8                     4061  C           ADD     BX,AX                       ; ADD TO START OF REGEN
1669  C3                        4062  C           RET
166A                            4063  C   FIND_POSITION   ENDP
                                4064  C
                                4065  C   ;--------------------------------------------------------
                                4066  C   ; EXPAND_MED_COLOR                                      :
                                4067  C   ;       THIS ROUTINE EXPANDS THE LOW 2 BITS IN BL TO    :
                                4068  C   ;       FILL THE ENTIRE BX REGISTER                     :
                                4069  C   ; ENTRY                                               :
                                4070  C   ;       BL = COLOR TO BE USED ( LOW 2 BITS )            :
                                4071  C   ; EXIT                                                :
                                4072  C   ;       BX = COLOR TO BE USED ( 8 REPLICATIONS OF THE   :
                                4073  C   ;       2 COLOR BITS )                                  :
                                4074  C   ;--------------------------------------------------------
166A                            4075  C   S19     PROC    NEAR
166A  80 E3 03                  4076  C           AND     BL,3                        ; ISOLATE THE COLOR BITS
166D  8A C3                     4077  C           MOV     AL,BL                       ; COPY TO AL
166F  51                        4078  C           PUSH    CX                          ; SAVE REGISTER
1670  B9 0003                  4079  C           MOV     CX,3                        ; NUMBER OF TIMES
1673                            4080  C   S20:
1673  D0 E0                     4081  C           SAL     AL,1
1675  D0 E0                     4082  C           SAL     AL,1                        ; LEFT SHIFT BY 2
1677  0A D8                     4083  C           OR      BL,AL                       ; ANOTHER COLOR VERSION
                                4084  C                                               ;  INTO BL
1679  E2 F8                     4085  C           LOOP    S20                         ; FILL ALL OF BL
167B  8A FB                     4086  C           MOV     BH,BL                       ; FILL UPPER PORTION
167D  59                        4087  C           POP     CX                          ; REGISTER BACK
167E  C3                        4088  C           RET                                 ; ALL DONE
167F                            4089  C   S19     ENDP
                                4090  C   ;--------------------------------------------------------
                                4091  C   ; EXPAND_BYTE                                         :
                                4092  C   ;       THIS ROUTINE TAKES THE BYTE IN AL AND DOUBLES    :
                                4093  C   ;       ALL OF THE BITS, TURNING THE 8 BITS INTO        :
                                4094  C   ;       16 BITS. THE RESULT `S LEFT IN AX               :
                                4095  C   ;--------------------------------------------------------
167F                            4096  C   S21     PROC    NEAR
167F  52                        4097  C           PUSH    DX                          ; SAVE REGISTERS
1680  51                        4098  C           PUSH    CX
1681  53                        4099  C           PUSH    BX
1682  2B D2                     4100  C           SUB     DX,DX                       ; RESULT REGISTER
1684  B9 0001                  4101  C           MOV     CX,1                        ; MASK REGISTER
1687                            4102  C   S22:
1687  8B D8                     4103  C           MOV     BX,AX                       ; BASE INTO TEMP
1689  23 D9                     4104  C           AND     BX,CX                       ; USE MASK TO EXTRACT BIT
168B  0B D3                     4105  C           OR      DX,BX                       ; PUT INTO RESULT REGISTER
168D  D1 E0                     4106  C           SHL     AX,1
168F  D1 E1                     4107  C           SHL     CX,1                        ; SHIFT BASE AND MASK BY 1
1691  8B D8                     4108  C           MOV     BX,AX                       ; BASE TO TEMP
1693  23 D9                     4109  C           AND     BX,CX                       ; EXTRACT THE SAME BIT
1695  0B D3                     4110  C           OR      DX,BX                       ; PUT INTO RESULT
1697  D1 E1                     4111  C           SHL     CX,1                        ; SHIFT ONLY MASK NOW,
                                4112  C                                               ;  MOVING TO NEXT BASE
1699  73 EC                     4113  C           JNC     S22                         ; USE MASK BIT COMING OUT
                                4114  C                                               ;  TO TERMINATE
169B  8B C2                     4115  C           MOV     AX,DX                       ; RESULT TO PARM REGISTER
169D  5B                        4116  C           POP     BX
169E  59                        4117  C           POP     CX                          ; RECOVER REGISTERS
169F  5A                        4118  C           POP     DX
16A0  C3                        4119  C           RET                                 ; ALL DONE
16A1                            4120  C   S21     ENDP
                                4121  C
16A1                            4122  C   S26     PROC    NEAR
16A1  A1 0450 R                 4123  C           MOV     AX,CURSOR_POSN              ; GET CURRENT CURSOR
16A4                            4124  C   GRAPH_POSN      LABEL   NEAR
16A4  53                        4125  C           PUSH    BX                          ; SAVE REGISTER
16A5  8B D8                     4126  C           MOV     BX,AX                       ; SAVE A COPY OF CURSOR
16A7  8A C4                     4127  C           MOV     AL,AH                       ; GET ROWS TO AL
16A9  F6 26 044A R              4128  C           MUL     BYTE PTR CRT_COLS           ; MULTIPLY BY BYTES/COLUMN
16AD  D1 E0                     4129  C           SHL     AX,1                        ; *4 SINCE 4 ROWS/BYTE
16AF  D1 E0                     4130  C           SHL     AX,1
16B1  2A FF                     4131  C           SUB     BH,BH                       ; ISOLATE COLUMN VALUE
16B3  03 C3                     4132  C           ADD     AX,BX                       ; DETERMINE OFFSET
16B5  5B                        4133  C           POP     BX                          ; RECOVER POINTER
16B6  C3                        4134  C           RET                                 ; ALL DONE
16B7                            4135  C   S26     ENDP
                                4136  C
                                4137  C   ;--------------------------------------------------------
                                4138  C   ; GR_CUR                                               :
                                4139  C   ; ENTRY                                               :
                                4140  C   ;       BH = DISPLAY PAGE                               :
                                4141  C   ; EXIT                                                :
                                4142  C   ;       AX = CURSOR POSITION FOR REQUESTED PAGE         :
                                4143  C   ;--------------------------------------------------------
16B7                            4144  C   GR_CUR:
                                4145  C           ASSUME  DS:ABS0
16B7  53                        4146  C           PUSH    BX                          ; SAVE REGISTER
16B8  8A DF                     4147  C           MOV     BL,BH                       ; GET TO LOW BYTE
16BA  2A FF                     4148  C           SUB     BH,BH                       ; ZERO HIGH BYTE
16BC  D1 E3                     4149  C           SAL     BX,1                        ; *2 FOR WORD COUNT
16BE  8B B7 0450 R              4150  C           MOV     AX,[BX + OFFSET CURSOR_POSN]    ; CURSOR, REQUESTED PAGE
16C2  5B                        4151  C           POP     BX                          ; RECOVER REGISTER
                                4152  C   ;--------------------------------------------------------
                                4153  C   ; GRX_PSN                                              :
                                4154  C   ; ENTRY                                               :
                                4155  C   ;       AX = CURSOR POSITION IN DESIRED PAGE            :
                                4156  C   ;       BH = DESIRED PAGE                               :
                                4157  C   ; EXIT                                                :
                                4158  C   ;       AX = BYTE OFFSET INTO REGEN                     :
```

```
                                4159    C  ;--------------------------------------------------------
16C3                            4160    C  GRX_PSN PROC    NEAR
16C3  53                        4161    C          PUSH    BX                          ; SAVE
16C4  51                        4162    C          PUSH    CX                          ; SAVE
16C5  52                        4163    C          PUSH    DX                          ; SAVE
16C6  2A ED                     4164    C          SUB     CH,CH                       ; ZERO
16C8  8A CF                     4165    C          MOV     CL,BH                       ; PAGE NUMBER
16CA  8B D8                     4166    C          MOV     BX,AX                       ; ROW, COLUMN
16CC  8A C4                     4167    C          MOV     AL,AH                       ; ROW
16CE  F6 26 044A R              4168    C          MUL     BYTE PTR CRT_COLS           ; ROW * COLUMNS/ROW
16D2  F7 26 0485 R              4169    C          MUL     POINTS                      ; BYTES PER ROW
16D6  2A FF                     4170    C          SUB     BH,BH                       ; ZERO TO LEAVE COL VALUE
16D8  03 C3                     4171    C          ADD     AX,BX                       ; ADD IN COLUMN
16DA  8B 1E 044C R              4172    C          MOV     BX,CRT_LEN                  ; PAGE LENGTH
16DE  E3 04                     4173    C          JCXZ    GP_2                        ; NO PAGE OFFSET
16E0                            4174    C  GP_3:
16E0  03 C3                     4175    C          ADD     AX,BX                       ; ADD IN THE PAGE LENGTH
16E2  E2 FC                     4176    C          LOOP    GP_3                        ; DO FOR NUMBER OF PAGES
16E4                            4177    C
16E4  5A                        4178    C  GP_2:   POP     DX                          ; RECOVER
16E5  59                        4179    C          POP     CX                          ; RECOVER
16E6  5B                        4180    C          POP     BX                          ; RECOVER
16E7  C3                        4181    C          RET
16E8                            4182    C  GRX_PSN ENDP
                                4183    C
16E8                            4184    C  MK_ES:
16E8  8E B800                   4185    C          MOV     SI,0B800H
16EB  8B 3E 0410 R              4186    C          MOV     DI,EQUIP_FLAG
16EF  81 E7 0030                4187    C          AND     DI,030H
16F3  83 FF 30                  4188    C          CMP     DI,030H
16F6  75 03                     4189    C          JNE     P6_A
16F8  BE B000                   4190    C          MOV     SI,0B000H
16FB                            4191    C  P6_A:
16FB  8E C6                     4192    C          MOV     ES,SI
16FD  C3                        4193    C          RET
                                4194    C
                                4195    C  ;--------------------------------------------------------
                                4196    C  ; READ_AC_CURRENT                                       :
                                4197    C  ;       THIS ROUTINE READS THE ATTRIBUTE AND CHARACTER  :
                                4198    C  ;       AT THE CURRENT CURSOR POSITION AND RETURNS THEM :
                                4199    C  ;       TO THE CALLER                                   :
                                4200    C  ;INPUT                                                  :
                                4201    C  ;       (AH) = CURRENT CRT MODE                         :
                                4202    C  ;       (BH) = DISPLAY PAGE ( ALPHA MODES ONLY )        :
                                4203    C  ;       (DS) = DATA SEGMENT                             :
                                4204    C  ;       (ES) = REGEN SEGMENT                            :
                                4205    C  ;OUTPUT                                                 :
                                4206    C  ;       (AL) = CHAR READ                                :
                                4207    C  ;       (AH) = ATTRIBUTE READ                           :
                                4208    C  ;--------------------------------------------------------
                                4209    C          ASSUME  CS:CODE,DS:ABS0,ES:NOTHING
16FE                            4210    C  READ_AC_CURRENT PROC    NEAR
16FE  E8 16E8 R                 4211    C          CALL    MK_ES
1701  E8 164E R                 4212    C          CALL    FIND_POSITION
1704  8B F3                     4213    C          MOV     SI,BX                       ; ADDRESSING IN SI
                                4214    C
1706  8B 16 0463 R              4215    C          MOV     DX,ADDR_6845                ; GET BASE ADDRESS
170A  83 C2 06                  4216    C          ADD     DX,6                        ; POINT AT STATUS PORT
                                4217    C
170D  F6 06 0487 R 04           4218    C          test    info,4
                                4219    C
1712  06                        4220    C          PUSH    ES
1713  1F                        4221    C          POP     DS                          ; SEGMENT FOR QUICK ACCESS
                                4222    C
1714  74 0B                     4223    C          jz      p3a
                                4224    C
                                4225    C  ;----- WAIT FOR HORIZONTAL RETRACE
                                4226    C
1716                            4227    C  P2:                                         ; WAIT FOR RETRACE LOW
                                4228    C          WIN                                 ; GET STATUS
1716  EC                        4229    C+         IN      AL,DX
1717  A8 01                     4230    C          TEST    AL,1                        ; IS HORZ RETRACE LOW
1719  75 FB                     4231    C          JNZ     P2                          ; WAIT UNTIL IT IS
171B  FA                        4232    C          CLI                                 ; NO MORE INTERRUPTS
171C                            4233    C  P3:                                         ; WAIT FOR RETRACE HIGH
                                4234    C          WIN                                 ; GET STATUS
171C  EC                        4235    C+         IN      AL,DX
171D  A8 01                     4236    C          TEST    AL,1                        ; IS IT HIGH
171F  74 FB                     4237    C          JZ      P3                          ; WAIT UNTIL IT IS
1721                            4238    C  p3a:
1721  AD                        4239    C          LODSW                               ; GET THE CHAR/ATTR
1722  E9 219B R                 4240    C          JMP     V_RET
1725                            4241    C  READ_AC_CURRENT ENDP
                                4242    C
                                4243    C  ;--------------------------------------------------------
                                4244    C  ; MED_READ_BYTE                                         :
                                4245    C  ;       THIS ROUTINE WILL TAKE 2 BYTES FROM THE REGEN   :
                                4246    C  ;       BUFFER, COMPARE AGAINST THE CURRENT FOREGROUND  :
                                4247    C  ;       COLOR, AND PLACE THE CORRESPONDING ON/OFF BIT   :
                                4248    C  ;       PATTERN INTO THE CURRENT POSITION IN THE SAVE   :
                                4249    C  ;       AREA                                            :
                                4250    C  ; ENTRY                                                 :
                                4251    C  ;       SI,DS = POINTER TO REGEN AREA OF INTEREST       :
                                4252    C  ;       BX = EXPANDED FOREGROUND COLOR                  :
                                4253    C  ;       BP = POINTER TO SAVE AREA                       :
                                4254    C  ; EXIT                                                  :
                                4255    C  ;       BP IS INCREMENT AFTER SAVE                      :
                                4256    C  ;--------------------------------------------------------
1725                            4257    C  S23     PROC    NEAR
1725  8A 24                     4258    C          MOV     AH,[SI]                     ; GET FIRST BYTE
1727  8A 44 01                  4259    C          MOV     AL,[SI+1]                   ; GET SECOND BYTE
172A  B9 C000                   4260    C          MOV     CX,0C000H                   ; 2 BIT MASK TO TEST
                                4261    C                                              ;  THE ENTRIES
172D  B2 00                     4262    C          MOV     DL,0                        ; RESULT REGISTER
172F                            4263    C  S24:
172F  85 C1                     4264    C          TEST    AX,CX                       ; IS THIS BACKGROUND?
1731  F8                        4265    C          CLC                                 ; CLEAR CARRY IN HOPES
                                4266    C                                              ;  THAT IT IS
1732  74 01                     4267    C          JZ      S25                         ; IF 0, IT IS BACKGROUND
1734  F9                        4268    C          STC                                 ; WASN'T, SO SET CARRY
1735                            4269    C  S25:
1735  D0 D2                     4270    C          RCL     DL,1                        ; MOVE THAT BIT INTO THE
1737  D1 E9                     4271    C          SHR     CX,1                        ;  RESULT
1739  D1 E9                     4272    C          SHR     CX,1                        ; MOVE THE MASK TO THE
                                4273    C                                              ;  RIGHT BY 2 BITS
173B  73 F2                     4274    C          JNC     S24                         ; DO IT AGAIN IF MASK
                                4275    C                                              ;  DIDN'T FALL OUT
173D  88 56 00                  4276    C          MOV     [BP],DL                     ; STORE RESULT IN SAVE
1740  45                        4277    C          INC     BP                          ; ADJUST POINTER
1741  C3                        4278    C          RET                                 ; ALL DONE
1742                            4279    C  S23     ENDP
                                4280    C
                                4281    C
1742                            4282    C  GRAPHICS_READ   PROC    NEAR
1742  E8 16E8 R                 4283    C          CALL    MK_ES
1745  E8 16A1 R                 4284    C          CALL    S26                         ; CONVERTED TO OFFSET
```

```
1748  8B F0                4285     C          MOV     SI,AX                           ; SAVE IN SI
174A  83 EC 08             4286     C          SUB     SP,8                            ; ALLOCATE SPACE TO SAVE
                           4287     C                                                  ;  THE READ CODE POINT
174D  8B EC                4288     C          MOV     BP,SP                           ; POINTER TO SAVE AREA
                           4289     C
                           4290     C  ;----- DETERMINE GRAPHICS MODES
                           4291     C
174F  80 3E 0449 R 06      4292     C          CMP     CRT_MODE,6
1754  06                   4293     C          PUSH    ES
1755  1F                   4294     C          POP     DS                              ; POINT TO REGEN SEGMENT
1756  72 1A                4295     C          JC      S13P                            ; MEDIUM RESOLUTION
                           4296     C
                           4297     C  ;----- HIGH RESOLUTION READ
                           4298     C
                           4299     C  ;----- GET VALUES FROM REGEN BUFFER AND CONVERT TO CODE POINT
                           4300     C
1758  B6 04                4301     C          MOV     DH,4                            ; NUMBER OF PASSES
175A                       4302     C  S12P:
175A  8A 04                4303     C          MOV     AL,[SI]                         ; GET FIRST BYTE
175C  88 46 00             4304     C          MOV     [BP],AL                         ; SAVE IN STORAGE AREA
175F  45                   4305     C          INC     BP                              ; NEXT LOCATION
1760  8A 84 2000           4306     C          MOV     AL,[SI+2000H]                   ; GET LOWER REGION BYTE
1764  88 46 00             4307     C          MOV     [BP],AL                         ; ADJUST AND STORE
1767  45                   4308     C          INC     BP
1768  83 C6 50             4309     C          ADD     SI,80                           ; POINTER INTO REGEN
176B  FE CE                4310     C          DEC     DH                              ; LOOP CONTROL
176D  75 EB                4311     C          JNZ     S12P                            ; DO IT SOME MORE
176F  EB 17 90             4312     C          JMP     S15P                            ; GO MATCH THE SAVED CODE
                           4313     C                                                  ;  POINTS
                           4314     C
                           4315     C  ;----- MEDIUM RESOLUTION READ
                           4316     C
1772                       4317     C  S13P:                                           ; MED_RES_READ
1772  D1 E6                4318     C          SAL     SI,1                            ; OFFSET*2, 2 BYTES/CHAR
1774  B6 04                4319     C          MOV     DH,4                            ; NUMBER OF PASSES
1776                       4320     C  S14P:
1776  E8 1725 R            4321     C          CALL    S23                             ; GET PAIR BYTES
                           4322     C                                                  ;  INTO SINGLE SAVE
1779  81 C6 2000           4323     C          ADD     SI,2000H                        ; GO TO LOWER REGION
177D  E8 1725 R            4324     C          CALL    S23                             ; GET THIS PAIR INTO SAVE
1780  81 EE 1F80           4325     C          SUB     SI,2000H-80                     ; ADJUST POINTER BACK INTO
1784  FE CE                4326     C          DEC     DH                              ;  UPPER
1786  75 EE                4327     C          JNZ     S14P                            ; KEEP GOING UNTIL 8 DONE
                           4328     C
                           4329     C  ;----- SAVE AREA HAS CHARACTER IN IT, MATCH IT
                           4330     C
1788                       4331     C  S15P:                                           ; FIND_CHAR
1788  1E                   4332     C          PUSH    DS
1789  E8 0D01 R            4333     C          CALL    DDS
                           4334     C          WLXS    ES,DI,GRX_SET                   ; ESTABLISH ADDRESSING
178C  C4 3E 010C R         4335     C+         LES     DI,GRX_SET
1790  1F                   4336     C          POP     DS
1791  83 ED 08             4337     C          SUB     BP,8                            ; ADJUST POINTER TO
                           4338     C                                                  ;  BEGINNING OF SAVE AREA
1794  8B F5                4339     C          MOV     SI,BP
1796  FC                   4340     C          CLD                                     ; ENSURE DIRECTION
1797  B0 00                4341     C          MOV     AL,0                            ; CURRENT CODE POINT BEING
                           4342     C  S16P:                                           ;  MATCHED
1799  16                   4343     C          PUSH    SS                              ; ADDRESSING TO STACK
179A  1F                   4344     C          POP     DS                              ; FOR THE STRING COMPARE
179B  BA 0080              4345     C          MOV     DX,128                          ; NUMBER TO TEST AGAINST
179E                       4346     C  S17P:
179E  56                   4347     C          PUSH    SI                              ; SAVE SAVE AREA POINTER
179F  57                   4348     C          PUSH    DI                              ; SAVE CODE POINTER
17A0  B9 0008              4349     C          MOV     CX,8                            ; NUMBER OF BYTES TO MATCH
17A3  F3/ A6               4350     C          REPE    CMPSB                           ; COMPARE THE 8 BYTES
17A5  5F                   4351     C          POP     DI                              ; RECOVER THE POINTERS
17A6  5E                   4352     C          POP     SI
17A7  74 1D                4353     C          JZ      S18P                            ; IF ZERO FLAG SET,
                           4354     C                                                  ;  THEN MATCH OCCURRED
17A9  FE C0                4355     C          INC     AL                              ; NO MATCH, MOVE TO NEXT
17AB  83 C7 08             4356     C          ADD     DI,8                            ; NEXT CODE POINT
17AE  4A                   4357     C          DEC     DX                              ; LOOP CONTROL
17AF  75 ED                4358     C          JNZ     S17P                            ; DO ALL OF THEM
                           4359     C
                           4360     C  ;----- CHAR NOT MATCHED, MIGHT BE IN USER SUPPLIED SECOND HALF
                           4361     C
17B1  3C 00                4362     C          CMP     AL,0                            ; AL <> 0 IF ONLY 1ST
                           4363     C                                                  ;  HALF SCANNED
17B3  74 11                4364     C          JE      S18P                            ; IF = 0, THEN ALL HAS
                           4365     C                                                  ;  BEEN SCANNED
                           4366     C          ASSUME  DS:ABS0
17B5  E8 0D01 R            4367     C          CALL    DDS
                           4368     C          WLXS    ES,DI,EXT_PTR                   ; GET POINTER
17B8  C4 3E 007C R         4369     C+         LES     DI,EXT_PTR
17BC  8C C0                4370     C          MOV     AX,ES                           ; SEE IF THE PNTR EXISTS
17BE  0B C7                4371     C          OR      AX,DI                           ; IF ALL 0, DOESN'T EXIST
17C0  74 04                4372     C          JZ      S18P                            ; NO SENSE LOOKING
17C2  B0 80                4373     C          MOV     AL,128                          ; ORIGIN FOR SECOND HALF
17C4  EB D3                4374     C          JMP     S16P                            ; GO BACK AND TRY FOR IT
                           4375     C
                           4376     C  ;----- CHARACTER IS FOUND ( AL=0 IF NOT FOUND )
                           4377     C
17C6                       4378     C  S18P:
17C6  83 C4 08             4379     C          ADD     SP,8                            ; READJUST THE STACK,
                           4380     C                                                  ;  THROW AWAY SAVE
17C9  E9 219B R            4381     C          JMP     V_RET                           ; ALL DONE
17CC                       4382     C  GRAPHICS_READ   ENDP
                           4383     C
                           4384     C
                           4385     C  ;----- READ CHARACTER/ATTRIBUTE AT CURRENT CURSOR POSITION
                           4386     C
17CC                       4387     C  AH8S:
17CC  E9 16FE R            4388     C          JMP     READ_AC_CURRENT
                           4389     C
17CF                       4390     C  AH8:
                           4391     C          ASSUME  DS:ABS0
17CF  8A 26 0449 R         4392     C          MOV     AH,CRT_MODE                     ; GET THE CURRENT MODE
17D3  80 FC 07             4393     C          CMP     AH,07H
17D6  74 F4                4394     C          JE      AH8S
17D8  80 FC 03             4395     C          CMP     AH,03H
17DB  76 EF                4396     C          JBE     AH8S
17DD  80 FC 06             4397     C          CMP     AH,06H
17E0  77 03                4398     C          JA      Z_1
17E2  E9 1742 R            4399     C          JMP     GRAPHICS_READ
17E5                       4400     C  Z_1:
17E5  80 FC 0F             4401     C          cmp     ah,0fh
17E8  72 52                4402     C          jb      grx_rd2
17EA  E8 14F4 R            4403     C          call    mem_det
17ED  72 4D                4404     C          jc      grx_rd2
17EF  EB 0A                4405     C          jmp     short grx_rd1
17F1  80 FC 0D             4406     C          CMP     AH,0DH                          ; RANGE TEST
17F4  73 46                4407     C          JAE     GRX_RD2                         ; FOUR MAP READ
17F6  B0 00                4408     C          MOV     AL,0
17F8  E9 219B R            4409     C          JMP     V_RET
                           4410     C
```

```
17FB                                    C  GRX_RD1 PROC    NEAR
                                        C          ASSUME  DS:ABS0
                                        C          SRLOAD  ES,0A000H                       ; REGEN SEGEMNT
17FB  BA A000                           C+         MOV     DX,0A000H
17FE  8E C2                             C+         MOV     ES,DX
1800  E8 16B7 R                        C          CALL    GR_CUR                          ; BYTE OFFSET INTO REGEN
1803  8B F0                             C          MOV     SI,AX                           ; SAVE IN SI
1805  8B 1E 0485 R                      C          MOV     BX,POINTS                       ; BYTES PER CHARACTER
1809  2B E3                             C          SUB     SP,BX                           ; ALLOCATE SPACE TO SAVE
                                        C                                                  ;  THE READ CODE POINT
180B  8B EC                             C          MOV     BP,SP                           ; POINTER TO SAVE AREA
                                        C
                                        C  ;----- GET VALUES FROM REGEN BUFFER AND CONVERT TO CODE POINT
                                        C
180D  53                                C          PUSH    BX                              ; SAVE BYTES PER CHARACTER
180E  24 01                             C          AND     AL,1                            ; ODD OR EVEN BYTE
1810  8A C8                             C          MOV     CL,AL                           ; USE FOR SHIFT
1812  B0 05                             C          MOV     AL,5                            ; COLOR COMP VALUE (C0-C2)
1814  D2 E0                             C          SHL     AL,CL                           ; (C1-C3) IF ODD BYTE
1816  B4 07                             C          MOV     AH,G_COLOR                      ; COLOR COMPARE REGISTER
1818  B6 03                             C          mov     dh,3
181A  B2 CE                             C          MOV     DL,GRAPH_ADDR
181C  E8 0D18 R                        C          CALL    OUT_DX                          ; SET GRAPHICS CHIP
181F  B8 0518                           C          MOV     AX,518H                         ; READ MODE
1822  E8 0D18 R                        C          CALL    OUT_DX                          ; SET GRAPHICS CHIP
1825                                    C
1825  26: 8A 04                        C  S12_1:  MOV     AL,ES:[SI]                      ; GET FIRST BYTE
1828  F6 D0                             C          NOT     AL
182A  88 46 00                          C          MOV     SS:[BP],AL                      ; SAVE IN STORAGE AREA
182D  45                                C          INC     BP                              ; NEXT LOCATION
182E  03 36 044A R                      C          ADD     SI,CRT_COLS                     ; POINTER INTO REGEN
1832  4B                                C          DEC     BX                              ; LOOP CONTROL
1833  75 F0                             C          JNZ     S12_1                           ; DO IT SOME MORE
1835  5B                                C          POP     BX                              ; RECOVER BYTES PER CHAR
1836  B8 0510                           C          MOV     AX,510H                         ; UNDO READ MODE
1839  EB 32 90                          C          JMP     GRX_RECG                        ; CHAR REGONTION ROUTINE
183C                                    C  GRX_RD1 ENDP
                                        C
183C                                    C  GRX_RD2 PROC    NEAR
                                        C          ASSUME  DS:ABS0
                                        C          SRLOAD  ES,0A000H                       ; REGEN SEGMENT
183C  BA A000                           C+         MOV     DX,0A000H
183F  8E C2                             C+         MOV     ES,DX
1841  E8 16B7 R                        C          CALL    GR_CUR                          ; BYTE OFFSET INTO REGEN
1844  8B F0                             C          MOV     SI,AX                           ; SAVE IN SI
1846  8B 1E 0485 R                      C          MOV     BX,POINTS                       ; BYTES PER CHARACTER
184A  2B E3                             C          SUB     SP,BX                           ; ALLOCATE SPACE TO SAVE
                                        C                                                  ;  THE READ CODE POINT
184C  8B EC                             C          MOV     BP,SP                           ; POINTER TO SAVE AREA
                                        C
                                        C  ;----- GET VALUES FROM REGEN BUFFER AND CONVERT TO CODE POINT
                                        C
184E  B6 03                             C          mov     dh,3
1850  B2 CE                             C          MOV     DL,GRAPH_ADDR                   ; GRAPHICS CHIP
1852  B8 0508                           C          MOV     AX,508H                         ; COLOR COMPARE
1855  E8 0D18 R                        C          CALL    OUT_DX                          ; SET THE REGISTER
1858  53                                C          PUSH    BX                              ; SAVE BYTES PER CHARACTER
1859                                    C  S12:
1859  26: 8A 04                        C          MOV     AL,ES:[SI]                      ; GET COLOR COMPARED BYTE
185C  F6 D0                             C          NOT     AL                              ; ADJUST
185E  88 46 00                          C          MOV     SS:[BP],AL                      ; SAVE IN STORAGE AREA
1861  45                                C          INC     BP                              ; NEXT LOCATION
1862  03 36 044A R                      C          ADD     SI,CRT_COLS                     ; POINTER INTO REGEN
1866  4B                                C          DEC     BX                              ; LOOP CONTROL
1867  75 F0                             C          JNZ     S12                             ; DO IT SOME MORE
1869  5B                                C          POP     BX                              ; RECOVER BYTES PER CHAR
186A  B8 0500                           C          MOV     AX,500H                         ; UNDO READ MODE
186D                                    C  GRX_RD2 ENDP
                                        C
186D                                    C  GRX_RECG:
                                        C
                                        C  ;----- SAVE AREA HAS CHARACTER IN IT, MATCH IT
                                        C
186D  E8 0D18 R                         C          CALL    OUT_DX                          ; SET READ MODE BACK
                                        C          WLXS    ES,DI,GRX_SET                   ; GET FONT DEFINITIONS
1870  C4 3E 010C R                      C+         LES     DI,GRX_SET
1874  2B EB                             C          SUB     BP,BX                           ; ADJUST POINTER TO
                                        C                                                  ;  BEGINNING OF SAVE AREA
1876  8B F5                             C          MOV     SI,BP
1878  FC                                C          CLD                                     ; ENSURE DIRECTION
1879  B0 00                             C          MOV     AL,0                            ; CODE POINT BEING MATCHED
187B  16                                C          PUSH    SS                              ; ADDRESSING TO STACK
187C  1F                                C          POP     DS                              ;  FOR THE STRING COMPARE
187D  BA 0100                           C          MOV     DX,256D                         ; NUMBER TO TEST AGAINST
1880                                    C  S17_5:
1880  56                                C          PUSH    SI                              ; SAVE SAVE AREA POINTER
1881  57                                C          PUSH    DI                              ; SAVE CODE POINTER
1882  8B CB                             C          MOV     CX,BX                           ; NUMBER OF BYTES TO MATCH
1884  F3/ A6                            C          REPE    CMPSB                           ; COMPARE THE 8 BYTES
1886  5F                                C          POP     DI                              ; RECOVER THE POINTERS
1887  5E                                C          POP     SI
1888  74 07                             C          JZ      S18_5                           ; IF ZFL SET, THEN MATCH
                                        C                                                  ;  OCCURRED
188A  FE C0                             C          INC     AL                              ; NO MATCH, ON TO NEXT
188C  03 FB                             C          ADD     DI,BX                           ; NEXT CODE POINT
188E  4A                                C          DEC     DX                              ; LOOP CONTROL
188F  75 EF                             C          JNZ     S17_5                           ; DO ALL OF THEM
1891                                    C  S18_5:                                          ; AL=CHAR, 0 IF NOT FOUND
1891  03 E3                             C          ADD     SP,BX                           ; READJUST THE STACK
1893  E9 219B R                         C          JMP     V_RET
                                        C
                                        C  ;----- WRITE CHARACTER/ATTRIBUTE AT CURRENT CURSOR POSITION
                                        C
                                        C  ;--------------------------------------------------
                                        C  ; WRITE_AC_CURRENT                               :
                                        C  ;       THIS ROUTINE WRITES THE ATTRIBUTE        :
                                        C  ;       AND CHARACTER AT THE CURRENT CURSOR      :
                                        C  ;       POSITION                                 :
                                        C  ; INPUT                                          :
                                        C  ;       (AH) = CURRENT CRT MODE                  :
                                        C  ;       (BH) = DISPLAY PAGE                      :
                                        C  ;       (CX) = COUNT OF CHARACTERS TO WRITE      :
                                        C  ;       (AL) = CHAR TO WRITE                     :
                                        C  ;       (BL) = ATTRIBUTE OF CHAR TO WRITE        :
                                        C  ;       (DS) = DATA SEGMENT                      :
                                        C  ;       (ES) = REGEN SEGMENT                     :
                                        C  ; OUTPUT                                         :
                                        C  ;       NONE                                     :
                                        C  ;--------------------------------------------------
1896                                    C  AH9:
                                        C          ASSUME  DS:ABS0
1896  E8 0D01 R                         C          CALL    DDS
1899  8A 26 0449 R                      C          MOV     AH,CRT_MODE
                                        C
189D  80 FC 04                          C          CMP     AH,4                            ; IS THIS GRAPHICS
18A0  72 08                             C          JC      P6
```

```
18A2  80 FC 07              4537     C          CMP     AH,7                            ; IS THIS BW CARD
18A5  74 03                 4538     C          JE      P6
18A7  EB 74 90              4539     C          JMP     GRAPHICS_WRITE
18AA                        4540     C  P6:                                             ; WRITE_AC_CONTINUE
18AA  E8 16E8 R             4541     C          CALL    MK_ES
18AD  8A E3                 4542     C          MOV     AH,BL                           ; GET ATTRIBUTE TO AH
18AF  50                    4543     C          PUSH    AX                              ; SAVE ON STACK
18B0  51                    4544     C          PUSH    CX                              ; SAVE WRITE COUNT
18B1  E8 164E R             4545     C          CALL    FIND_POSITION
18B4  8B FB                 4546     C          MOV     DI,BX                           ; ADDRESS TO DI REGISTER
18B6  59                    4547     C          POP     CX                              ; WRITE COUNT
18B7  5B                    4548     C          POP     BX                              ; CHARACTER IN BX REG
18B8  8B 16 0463 R          4549     C          MOV     DX,ADDR_6845                    ; GET BASE ADDRESS
18BC  83 C2 06              4550     C          ADD     DX,6                            ; POINT AT STATUS PORT
                            4551     C
                            4552     C  ;----- WAIT FOR HORIZONTAL RETRACE
                            4553     C
18BF                        4554     C  p7:
18BF  F6 06 0487 R 04       4555     C          test    info,4
18C4  74 0B                 4556     C          jz      p9a
18C6                        4557     C  P8:
18C6  EC                    4558     C          IN      AL,DX                           ; GET STATUS
18C7  A8 01                 4559     C          TEST    AL,1                            ; IS IT LOW
18C9  75 FB                 4560     C          JNZ     P8                              ; WAIT UNTIL IT IS
18CB  FA                    4561     C          CLI                                     ; NO MORE INTERRUPTS
18CC                        4562     C  P9:
18CC  EC                    4563     C          IN      AL,DX                           ; GET STATUS
18CD  A8 01                 4564     C          TEST    AL,1                            ; IS IT HIGH
18CF  74 FB                 4565     C          JZ      P9                              ; WAIT UNTIL IT IS
18D1                        4566     C  p9a:
18D1  8B C3                 4567     C          MOV     AX,BX                           ; RECOVER THE CHAR/ATTR
18D3  AB                    4568     C          STOSW                                   ; PUT THE CHAR/ATTR
18D4  FB                    4569     C          STI                                     ; INTERRUPTS BACK ON
18D5  E2 E8                 4570     C          LOOP    P7                              ; AS MANY TIMES
18D7  E9 219B R             4571     C          JMP     V_RET
                            4572     C
                            4573     C  ;----- WRITE CHARACTER ONLY AT CURRENT CURSOR POSITION
                            4574     C
                            4575     C  ;--------------------------------------------------
                            4576     C  ; WRITE_C_CURRENT                               :
                            4577     C  ;       THIS ROUTINE WRITES THE CHARACTER AT    :
                            4578     C  ;       THE CURRENT CURSOR POSITION, ATTRIBUTE  :
                            4579     C  ;       UNCHANGED                               :
                            4580     C  ; INPUT                                         :
                            4581     C  ;       (AH) = CURRENT CRT MODE                 :
                            4582     C  ;       (BH) = DISPLAY PAGE                     :
                            4583     C  ;       (CX) = COUNT OF CHARACTERS TO WRITE     :
                            4584     C  ;       (AL) = CHAR TO WRITE                    :
                            4585     C  ;       (DS) = DATA SEGMENT                     :
                            4586     C  ;       (ES) = REGEN SEGMENT                    :
                            4587     C  ; OUTPUT                                        :
                            4588     C  ;       NONE                                    :
                            4589     C  ;--------------------------------------------------
18DA                        4590     C  AHA:
                            4591     C          ASSUME  DS:ABSO
18DA  E8 0D01 R             4592     C          CALL    DDS
18DD  8A 26 0449 R          4593     C          MOV     AH,CRT_MODE
                            4594     C
18E1  80 FC 04              4595     C          CMP     AH,4                            ; IS THIS GRAPHICS
18E4  72 08                 4596     C          JC      P10
18E6  80 FC 07              4597     C          CMP     AH,7                            ; IS THIS BW CARD
18E9  74 03                 4598     C          JE      P10
                            4599     C
18EB  EB 30 90              4600     C          JMP     GRAPHICS_WRITE
18EE                        4601     C  P10:
18EE  E8 16E8 R             4602     C          CALL    MK_ES
18F1  50                    4603     C          PUSH    AX                              ; SAVE ON STACK
18F2  51                    4604     C          PUSH    CX                              ; SAVE WRITE COUNT
18F3  E8 164E R             4605     C          CALL    FIND_POSITION
18F6  8B FB                 4606     C          MOV     DI,BX                           ; ADDRESS TO DI
18F8  59                    4607     C          POP     CX                              ; WRITE COUNT
18F9  5B                    4608     C          POP     BX                              ; BL HAS CHAR TO WRITE
                            4609     C
                            4610     C  ;----- WAIT FOR HORIZONTAL RETRACE
                            4611     C
18FA  8B 16 0463 R          4612     C          MOV     DX,ADDR_6845                    ; GET BASE ADDRESS
18FE  83 C2 06              4613     C          ADD     DX,6                            ; POINT AT STATUS PORT
1901                        4614     C  p11:
1901  F6 06 0487 R 04       4615     C          test    info,4
1906  74 0B                 4616     C          jz      p13a
1908                        4617     C  P12:
                            4618     C          WIN                                     ; GET STATUS
1908  EC                    4619     C+         IN      AL,DX
1909  A8 01                 4620     C          TEST    AL,1                            ; IS IT LOW
190B  75 FB                 4621     C          JNZ     P12                             ; WAIT UNTIL IT IS
190D  FA                    4622     C          CLI                                     ; NO MORE INTERRUPTS
190E                        4623     C  P13:
                            4624     C          WIN                                     ; GET STATUS
190E  EC                    4625     C+         IN      AL,DX
190F  A8 01                 4626     C          TEST    AL,1                            ; IS IT HIGH
1911  74 FB                 4627     C          JZ      P13                             ; WAIT UNTIL IT IS
1913                        4628     C  p13a:
1913  8A C3                 4629     C          MOV     AL,BL                           ; RECOVER CHAR
1915  AA                    4630     C          STOSB                                   ; PUT THE CHAR/ATTR
1916  FB                    4631     C          STI                                     ; INTERRUPTS BACK ON
1917  47                    4632     C          INC     DI                              ; BUMP POINTER PAST ATTR
1918  E2 E7                 4633     C          LOOP    P11                             ;  AS REQUESTED
191A  E9 219B R             4634     C          JMP     V_RET
                            4635     C
                            4636     C  ;--------------------------------------------------------------
                            4637     C  ; GRAPHICS WRITE                                             :
                            4638     C  ;       THIS ROUTINE WRITES THE ASCII CHARACTER TO THE       :
                            4639     C  ;       CURRENT POSITION ON THE SCREEN.                      :
                            4640     C  ; ENTRY                                                      :
                            4641     C  ;       AL = CHARACTER TO WRITE                              :
                            4642     C  ;       BL = COLOR ATTRIBUTE TO BE USED FOR FOREGROUND COLOR :
                            4643     C  ;        IF BIT 7 IS SET, THE CHAR IS XOR'D INTO THE REGEN   :
                            4644     C  ;        BUFFER (0 IS USED FOR THE BACKGROUND COLOR)         :
                            4645     C  ;       CX = NUMBER OF CHARS TO WRITE                        :
                            4646     C  ;       DS = DATA SEGMENT                                    :
                            4647     C  ;       ES = REGEN SEGMENT                                   :
                            4648     C  ; EXIT                                                       :
                            4649     C  ;       NOTHING IS RETURNED                                  :
                            4650     C  ;                                                            :
                            4651     C  ; GRAPHICS READ                                              :
                            4652     C  ;       THIS ROUTINE READS THE ASCII CHARACTER AT THE CURRENT :
                            4653     C  ;       CURSOR POSITION ON THE SCREEN BY MATCHING THE DOTS ON :
                            4654     C  ;       THE SCREEN TO THE CHARACTER GENERATOR CODE POINTS     :
                            4655     C  ; ENTRY                                                      :
                            4656     C  ;       NONE  (0 IS ASSUMED AS THE BACKGROUND COLOR)         :
                            4657     C  ; EXIT                                                       :
                            4658     C  ;       AL = CHARACTER READ AT THAT POSITION (0 RETURNED IF   :
                            4659     C  ;            NONE FOUND)                                     :
                            4660     C  ;                                                            :
                            4661     C  ; FOR COMPATIBILITY ROUTINES, THE IMAGES USED TO FORM CHARS ARE :
                            4662     C  ;  CONTAINED IN ROM FOR THE 1ST 128 CHARS.  TO ACCESS CHARS     :
```

```
                                4663     C ;  IN THE SECOND HALF, THE USER MUST INITIALIZE THE VECTOR AT   :
                                4664     C ;  INTERRUPT 1FH (LOCATION 0007CH) TO POINT TO THE USER         :
                                4665     C ;  SUPPLIED TABLE OF GRAPHIC IMAGES (8X8 BOXES).                :
                                4666     C ;  FAILURE TO DO SO WILL CAUSE IN STRANGE RESULTS                :
                                4667     C ;----------------------------------------------------------------
                                4668     C         ASSUME  CS:CODE,DS:ABSO,ES:NOTHING
191D                            4669     C GRAPHICS_WRITE  PROC    NEAR
191D  80 FC 07                  4670     C         CMP     AH,7
1920  72 03                     4671     C         JB      S1_A
1922  E9 19D4 R                 4672     C         JMP     GRX_WRT
                                4673     C S1_A:
1925  E8 16E8 R                 4674     C         CALL    MK_ES
1928  B4 00                     4675     C         MOV     AH,0                          ; 0 TO HIGH OF CODE POINT
192A  50                        4676     C         PUSH    AX                            ; SAVE CODE POINT VALUE
                                4677     C
                                4678     C ;----- DETERMINE POSITION IN REGEN BUFFER TO PUT CODE POINTS
                                4679     C
192B  E8 16A1 R                 4680     C         CALL    S26                           ; LOC IN REGEN BUFFER
192E  8B F8                     4681     C         MOV     DI,AX                         ; REGEN POINTER IN DI
                                4682     C
                                4683     C ;----- DETERMINE REGION TO GET CODE POINTS FROM
                                4684     C
1930  58                        4685     C         POP     AX                            ; RECOVER CODE POINT
1931  3C 80                     4686     C         CMP     AL,80H                        ; IS IT IN SECOND HALF
1933  73 06                     4687     C         JAE     S1                            ; YES
                                4688     C
                                4689     C ;----- IMAGE IS IN FIRST HALF, CONTAINED IN ROM
                                4690     C
                                4691     C         WLXS    DS,SI,GRX_SET
1935  C5 36 010C R              4692     C+        LDS     SI,GRX_SET
1939  EB 06                     4693     C         JMP     SHORT S2                      ; DETERMINE_MODE
                                4694     C
                                4695     C ;----- IMAGE IS IN SECOND HALF, IN USER RAM
                                4696     C
193B                            4697     C S1:                                           ; EXTEND_CHAR
193B  2C 80                     4698     C         SUB     AL,80H                        ; 0 ORIGIN FOR SECOND HALF
                                4699     C         WLXS    DS,SI,EXT_PTR
193D  C5 36 007C R              4700     C+        LDS     SI,EXT_PTR
                                4701     C
                                4702     C ;----- DETERMINE GRAPHICS MODE IN OPERATION
                                4703     C
1941                            4704     C S2:                                           ; DETERMINE_MODE
1941  D1 E0                     4705     C         SAL     AX,1                          ; MULTIPLY CODE POINT
1943  D1 E0                     4706     C         SAL     AX,1                          ;  VALUE BY 8
1945  D1 E0                     4707     C         SAL     AX,1
1947  03 F0                     4708     C         ADD     SI,AX                         ; SI HAS OFFSET OF
1949  1E                        4709     C         PUSH    DS                            ;  DESIRES CODES
194A  E8 0D01 R                 4710     C         CALL    DDS
194D  80 3E 0449 R 06           4711     C         CMP     CRT_MODE,6
1952  1F                        4712     C         POP     DS
1953  72 2C                     4713     C         JC      S7                            ; TEST FOR MEDIUM RES MODE
                                4714     C
                                4715     C ;----- HIGH RESOLUTION MODE
                                4716     C
1955                            4717     C S3:                                           ; HIGH_CHAR
1955  57                        4718     C         PUSH    DI                            ; SAVE REGEN POINTER
1956  56                        4719     C         PUSH    SI                            ; SAVE CODE POINTER
1957  B6 04                     4720     C         MOV     DH,4                          ; NUMBER OF TIMES THROUGH
1959                            4721     C S4:                                           ;  LOOP
1959  AC                        4722     C         LODSB                                 ; GET BYTE FROM CODE POINT
195A  F6 C3 80                  4723     C         TEST    BL,80H                        ; SHOULD WE USE THE
195D  75 16                     4724     C         JNZ     S6                            ;  FUNCTION TO PUT CHAR IN
195F  AA                        4725     C         STOSB                                 ; STORE IN REGEN BUFFER
1960  AC                        4726     C         LODSB
1961                            4727     C
1961  26: 88 85 1FFF            4728     C S5:     MOV     ES:[DI+2000H-1],AL            ; STORE IN SECOND HALF
1966  83 C7 4F                  4729     C         ADD     DI,79                         ; MOVE TO NEXT ROW IN REGEN
1969  FE CE                     4730     C         DEC     DH                            ; DONE WITH LOOP
196B  75 EC                     4731     C         JNZ     S4
196D  5E                        4732     C         POP     SI
196E  5F                        4733     C         POP     DI                            ; RECOVER REGEN POINTER
196F  47                        4734     C         INC     DI                            ; POINT TO NEXT CHAR POS
1970  E2 E3                     4735     C         LOOP    S3                            ; MORE CHARS TO WRITE
1972  E9 219B R                 4736     C         JMP     V_RET
1975                            4737     C S6:
1975  26: 32 05                 4738     C         XOR     AL,ES:[DI]                    ; XOR WITH CURRENT
1978  AA                        4739     C         STOSB                                 ; STORE THE CODE POINT
1979  AC                        4740     C         LODSB                                 ; AGAIN FOR ODD FIELD
197A  26: 32 85 1FFF            4741     C         XOR     AL,ES:[DI+2000H-1]
197F  EB E0                     4742     C         JMP     S5                            ; BACK TO MAINSTREAM
                                4743     C
                                4744     C ;----- MEDIUM RESOLUTION WRITE
                                4745     C
1981                            4746     C S7:                                           ; MED_RES_WRITE
1981  8A D3                     4747     C         MOV     DL,BL                         ; SAVE HIGH COLOR BIT
1983  D1 E7                     4748     C         SAL     DI,1                          ; OFFSET*2, 2 BYTES/CHAR
1985  E8 166A R                 4749     C         CALL    S19                           ; EXPAND BL TO FULL WORD
1988                            4750     C S8:                                           ;  OF COLOR
1988  57                        4751     C         PUSH    DI                            ; SAVE REGEN POINTER
1989  56                        4752     C         PUSH    SI                            ; SAVE THE CODE POINTER
198A  B6 04                     4753     C         MOV     DH,4                          ; NUMBER OF LOOPS
198C                            4754     C S9:
198C  AC                        4755     C         LODSB                                 ; GET CODE POINT
198D  E8 167F R                 4756     C         CALL    S21                           ; DOUBLE UP ALL THE BITS
1990  23 C3                     4757     C         AND     AX,BX                         ; CONVERT THEM TO FORE-
                                4758     C                                               ;  GROUND COLOR (0 BACK)
1992  F6 C2 80                  4759     C         TEST    DL,80H                        ; IS THIS XOR FUNCTION
1995  74 07                     4760     C         JZ      S10                           ; NO, STORE IT IN AS IT IS
1997  26: 32 25                 4761     C         XOR     AH,ES:[DI]                    ; DO FUNCTION WITH HALF
199A  26: 32 45 01              4762     C         XOR     AL,ES:[DI+1]                  ;  AND WITH OTHER HALF
199E                            4763     C S10:
199E  26: 88 25                 4764     C         MOV     ES:[DI],AH                    ; STORE FIRST BYTE
19A1  26: 88 45 01              4765     C         MOV     ES:[DI+1],AL                  ; STORE SECOND BYTE
19A5  AC                        4766     C         LODSB                                 ; GET CODE POINT
19A6  E8 167F R                 4767     C         CALL    S21
19A9  23 C3                     4768     C         AND     AX,BX                         ; CONVERT TO COLOR
19AB  F6 C2 80                  4769     C         TEST    DL,80H                        ; IS THIS XOR FUNCTION
19AE  74 0A                     4770     C         JZ      S11                           ; NO, JUST STORE THE VALUE
19B0  26: 32 A5 2000            4771     C         XOR     AH,ES:[DI+2000H]              ; FUNCTION WITH FIRST HALF
19B5  26: 32 85 2001            4772     C         XOR     AL,ES:[DI+2001H]              ; AND WITH SECOND HALF
19BA                            4773     C S11:
19BA  26: 88 A5 2000            4774     C         MOV     ES:[DI+2000H],AH
19BF  26: 88 85 2001            4775     C         MOV     ES:[DI+2000H+1],AL            ; STORE IN SECOND PORTION
19C4  83 C7 50                  4776     C         ADD     DI,80                         ; POINT TO NEXT LOCATION
19C7  FE CE                     4777     C         DEC     DH
19C9  75 C1                     4778     C         JNZ     S9                            ; KEEP GOING
19CB  5E                        4779     C         POP     SI                            ; RECOVER CODE PONTER
19CC  5F                        4780     C         POP     DI                            ; RECOVER REGEN POINTER
19CD  47                        4781     C         INC     DI                            ; POINT TO NEXT CHAR
19CE  47                        4782     C         INC     DI
19CF  E2 B7                     4783     C         LOOP    S8                            ; MORE TO WRITE
19D1  E9 219B R                 4784     C         JMP     V_RET
19D4                            4785     C GRAPHICS_WRITE  ENDP
                                4786     C
                                4787     C ;----------------------------------------------------------------
                                4788     C ; ENTRY                                                         :
```

```
                                4789     C  ;       AL = CHAR TO WRITE                                      :
                                4790     C  ;       BH = DISPLAY PAGE                                       :
                                4791     C  ;       BL = ATTRIBUTE/COLOR                                    :
                                4792     C  ;       CX = COUNT OF CHARS TO WRITE                            :
                                4793     C  ;---------------------------------------------------------------
19D4                            4794     C  GRX_WRT PROC    NEAR
                                4795     C          ASSUME  DS:ABS0, ES:NOTHING
19D4  80 FC 0F                  4796     C          cmp     ah,0fh                          ; 640x350 graphics
19D7  72 0E                     4797     C          jb      no_adj1
19D9  E8 14F4 R                 4798     C          call    mem_det                         ; base card
19DC  72 09                     4799     C          jc      no_adj1
19DE  80 E3 85                  4800     C          and     bl,10000101b                    ; 85h, xor c2 c0 mask
19E1  8A E3                     4801     C          mov     ah,bl
19E3  D0 E4                     4802     C          shl     ah,1                            ; expand c0 to c1, c2 to c3
19E5  0A DC                     4803     C          or      bl,ah                           ; build ?(80h) + (0,3,c,f)
19E7                            4804     C  no_adj1:
19E7  2A E4                     4805     C          SUB     AH,AH                           ; ZERO
19E9  F7 26 0485 R              4806     C          MUL     POINTS                          ; OFFSET FONT TABLE BASE
19ED  50                        4807     C          PUSH    AX                              ; FONT TABLE DISPLACEMENT
19EE  E8 16B7 R                 4808     C          CALL    GR_CUR                          ; GET OFFSET INTO REGEN
19F1  8B F8                     4809     C          MOV     DI,AX                           ;  INTO DESTINATION
19F3  8B 2E 0485 R              4810     C          MOV     BP,POINTS                       ; BYTES PER CHAR
                                4811     C          SRLOAD  ES,0A000H                       ; REGEN SEGEMNT
19F7  BA A000                   4812     C+         MOV     DX,0A000H
19FA  8E C2                     4813     C+         MOV     ES,DX
                                4814     C          WLXS    DS,SI,GRX_SET                   ; ADDRESSING TO FONTS
19FC  C5 36 010C R              4815     C+         LDS     SI,GRX_SET
1A00  58                        4816     C          POP     AX                              ; RECOVER OFFSET
1A01  03 F0                     4817     C          ADD     SI,AX                           ; CHARACTER IN TABLE
1A03  B6 03                     4818     C          mov     dh,3
1A05                            4819     C  S20A:
1A05  F6 C3 80                  4820     C          TEST    BL,080H                         ; TEST FOR XOR
1A08  74 0B                     4821     C          JZ      NO_XOR                          ; NO XOR
1A0A  B2 CE                     4822     C          MOV     DL,GRAPH_ADDR
1A0C  B8 0318                   4823     C          MOV     AX,0318H                        ; GRAPHICS CHIP XOR
1A0F  E8 0D18 R                 4824     C          CALL    OUT_DX                          ; SET REGISTER
1A12  EB 1E 90                  4825     C          JMP     F_2                             ; SKIP BLANK
1A15                            4826     C  NO_XOR:                                         ; BLANK BOX FOR CHAR
1A15  57                        4827     C          PUSH    DI                              ; SAVE REGEN POINTER
1A16  B2 C4                     4828     C          MOV     DL,SEQ_ADDR
1A18  B8 020F                   4829     C          MOV     AX,020FH                        ; ENABLE ALL MAPS
1A1B  E8 0D18 R                 4830     C          CALL    OUT_DX
1A1E  2B C0                     4831     C          SUB     AX,AX                           ; STORE ZERO
1A20  51                        4832     C          PUSH    CX                              ; SAVE CHARACTER COUNT
1A21  8B CD                     4833     C          MOV     CX,BP                           ; GET BYTE COUNT
1A23  1E                        4834     C          PUSH    DS
1A24  E8 0D01 R                 4835     C          CALL    DDS
1A27                            4836     C  S13A:
1A27  AA                        4837     C          STOSB                                   ; ZERO REGEN BYTE
1A28  03 3E 044A R              4838     C          ADD     DI,CRT_COLS                     ; NEXT BYTE OF BOX
1A2C  4F                        4839     C          DEC     DI                              ; ADJUST
1A2D  E2 F8                     4840     C          LOOP    S13A                            ; NEXT BYTE
1A2F  1F                        4841     C          POP     DS
1A30  59                        4842     C          POP     CX                              ; RECOVER CHARACTER COUNT
1A31  5F                        4843     C          POP     DI                              ; RECOVER REGEN POINTER
1A32                            4844     C  F_2:
1A32  B2 C4                     4845     C          MOV     DL,SEQ_ADDR
1A34  B4 02                     4846     C          MOV     AH,02H                          ; SET MAP MASK
1A36  8A C3                     4847     C          MOV     AL,BL                           ;  FOR COLOR
1A38  E8 0D18 R                 4848     C          CALL    OUT_DX                          ; SET THE CHIP
1A3B  57                        4849     C          PUSH    DI                              ; SAVE OFFSET IN REGEN
1A3C  53                        4850     C          PUSH    BX                              ; SAVE COLOR VALUE
1A3D  51                        4851     C          PUSH    CX                              ; SAVE CHACTER COUNT
1A3E  8B DD                     4852     C          MOV     BX,BP                           ; LOOP CONTROL, BYTES/CHAR
1A40  1E                        4853     C          PUSH    DS                              ; SAVE FONT SEGMENT
1A41  E8 0D01 R                 4854     C          CALL    DDS                             ; SET LOW RAM SEGMENT
                                4855     C          ASSUME  DS:ABS0
1A44  8B 0E 044A R              4856     C          MOV     CX,CRT_COLS                     ; GET COLUMN COUNT
1A48  1F                        4857     C          POP     DS                              ; RESTORE FONT SEGMENT
                                4858     C          ASSUME  DS:NOTHING
1A49                            4859     C  S1K:                                            ; WRITE OUT THE CHARACTER
1A49  8A 04                     4860     C          MOV     AL,DS:[SI]                      ; CODE POINT
1A4B  26: 8A 25                 4861     C          MOV     AH,ES:[DI]                      ; LATCH DATA
1A4E  26: 88 05                 4862     C          MOV     ES:[DI],AL                      ; WRITE ONE BYTE OF FONT
1A51  46                        4863     C          INC     SI                              ; NEXT FONT POINT
1A52  03 F9                     4864     C          ADD     DI,CX                           ; ONE ROW BELOW LAST POINT
1A54  4B                        4865     C          DEC     BX                              ; BYTES PER CHAR COUNTER
1A55  75 F2                     4866     C          JNZ     S1K                             ; DO NEXT ROW OF CHARACTER
                                4867     C
1A57  59                        4868     C          POP     CX                              ; CHARACTER COUNT
1A58  5B                        4869     C          POP     BX                              ; COLOR VALUE
1A59  2B F5                     4870     C          SUB     SI,BP                           ; ADJUST PTR TO FONT TABLE
1A5B  5F                        4871     C          POP     DI                              ; REGEN POINTER
1A5C  47                        4872     C          INC     DI                              ; NEXT CHAR POSN IN REGEN
1A5D  E2 A6                     4873     C          LOOP    S20A                            ; WRITE ANOTHER CHARACTER
                                4874     C
1A5F  B2 CE                     4875     C          MOV     DL,GRAPH_ADDR
1A61  B8 0300                   4876     C          MOV     AX,0300H                        ; NORMAL WRITE, NO ROTATE
1A64  E8 0D18 R                 4877     C          CALL    OUT_DX                          ; SET THE CHIP
1A67  B2 C4                     4878     C          MOV     DL,SEQ_ADDR
1A69  B8 020F                   4879     C          MOV     AX,020FH                        ; ENABLE ALL MAPS
1A6C  E8 0D18 R                 4880     C          CALL    OUT_DX                          ; SET THE CHIP
1A6F  E9 219B R                 4881     C          JMP     V_RET
1A72                            4882     C  GRX_WRT ENDP
                                4883     C
                                4884     C          SUBTTL
                                4885
                                4886        ;----- SET COLOR PALETTE
                                4887
1A72                            4888        AHB:
                                4889                ASSUME  DS:ABS0
1A72  80 3E 0463 R B4           4890                CMP     BYTE PTR ADDR_6845,0B4H
1A77  74 09                     4891                JE      M21_B                           ; call valid only for color
1A79  F6 06 0487 R 02           4892                test    info,2                          ; see if its the old color card
1A7E  74 05                     4893                JZ      M21_A                           ; if not, handle it here
1A80  CD 42                     4894                INT     42H                             ; old code call
1A82                            4895        M21_B:
1A82  E9 219B R                 4896                JMP     V_RET                           ; back to caller
1A85                            4897        M21_A:
1A85  2B C0                     4898                sub     ax,ax
1A87  8B E8                     4899                mov     bp,ax
1A89  C4 3E 04A8 R              4900                les     di,save_ptr
1A8D  83 C7 04                  4901                add     di,4
1A90  26: C4 3D                 4902                les     di,dword ptr es:[di]
1A93  8C C0                     4903                mov     ax,es
1A95  0B C7                     4904                or      ax,di
1A97  74 01                     4905                jz      not4ahb
1A99  45                        4906                inc     bp
1A9A                            4907        not4ahb:
1A9A  E8 1DBD R                 4908                CALL    PAL_INIT
1A9D  0A FF                     4909                OR      BH,BH
1A9F  75 65                     4910                JNZ     M20
                                4911
                                4912        ;----- HANDLE BH = 0 HERE
                                4913        ;       ALPHA MODES  => BL = OVERSCAN COLOR
                                4914        ;       GRAPHICS     => BL = OVERSCAN AND BACKGROUND COLOR
```

```
                                4915
                                4916    ;----- MOVE INTENSITY BIT FROM D3 TO D4 FOR COMPATIBILITY
                                4917
1AA1  8A FB                     4918            mov     bh,bl
1AA3  A0 0466 R                 4919            mov     al,crt_palette
1AA6  24 E0                     4920            and     al,0e0h
1AA8  80 E3 1F                  4921            and     bl,01fh
1AAB  0A C3                     4922            or      al,bl
1AAD  A2 0466 R                 4923            mov     crt_palette,al
1AB0  8A DF                     4924            mov     bl,bh
1AB2  80 E7 08                  4925            and     bh,08h
1AB5  D0 E7                     4926            shl     bh,1
1AB7  8A E8                     4927            mov     ch,al
1AB9  80 E5 EF                  4928            and     ch,0efh
1ABC  0A ED                     4929            or      ch,ch
1ABE  80 E3 0F                  4930            AND     BL,0FH
1AC1  8A FB                     4931            MOV     BH,BL
1AC3  D0 E3                     4932            SHL     BL,1
1AC5  80 E3 10                  4933            AND     BL,010H
1AC8  80 E7 07                  4934            AND     BH,07H
1ACB  0A DF                     4935            OR      BL,BH
                                4936
1ACD  A0 0449 R                 4937            MOV     AL,CRT_MODE
1AD0  3C 03                     4938            CMP     AL,3
1AD2  76 0E                     4939            JBE     M21
                                4940
                                4941    ;----- GRAPHICS MODE DONE HERE (SET PALETTE 0 AND OVERSCAN)
                                4942
1AD4  B4 00                     4943            MOV     AH,0
1AD6  8A C3                     4944            MOV     AL,BL
1AD8  E8 1D9C R                 4945            CALL    PAL_SET
                                4946
1ADB  0B ED                     4947            or      bp,bp
1ADD  74 03                     4948            jz      m21
1ADF  26: 88 1D                 4949            mov     es:[di],bl
                                4950
                                4951    ;----- ALPHA MODE DONE HERE (SET OVERSCAN REGISTER)
                                4952
1AE2                            4953    M21:
1AE2  80 3E 0449 R 03           4954            cmp     crt_mode,3                      ; check for an enhanced mode
1AE7  77 05                     4955            ja      set_ovrsc                       ; no chance
1AE9  E8 0E9C R                 4956            call    brst_det                        ; see if we are enhanced
1AEC  72 07                     4957            jc      skip_ovrsc                      ; there is no border
1AEE                            4958    set_ovrsc:
1AEE  B4 11                     4959            MOV     AH,011H                         ; overscan register
1AF0  8A C3                     4960            MOV     AL,BL
1AF2  E8 1D9C R                 4961            CALL    PAL_SET                         ; set the border
1AF5                            4962    skip_ovrsc:
1AF5  0B ED                     4963            or      bp,bp
1AF7  74 04                     4964            jz      m21y
1AF9  26: 88 5D 10              4965            mov     es:[di][16d],bl
1AFD                            4966    m21y:
1AFD  8A DD                     4967            mov     bl,ch
1AFF  80 E3 20                  4968            AND     BL,020H
1B02  B1 05                     4969            MOV     CL,5
1B04  D2 EB                     4970            SHR     BL,CL
                                4971
                                4972    ;----- HANDLE BH = 1 HERE
                                4973    ;       ALPHA MODES  => NO EFFECT
                                4974    ;       GRAPHICS     => LOW BIT OF BL = 0
                                4975    ;                       PALETTE 0 = BACKGROUND
                                4976    ;                       PALETTE 1 = GREEN
                                4977    ;                       PALETTE 2 = RED
                                4978    ;                       PALETTE 3 = BROWN
                                4979    ;                    => LOW BIT OF BL = 1
                                4980    ;                       PALETTE 0 = BACKGROUND
                                4981    ;                       PALETTE 1 = CYAN
                                4982    ;                       PALETTE 2 = MAGENTA
                                4983    ;                       PALETTE 3 = WHITE
                                4984    ;
                                4985
1B06                            4986    M20:
1B06  80 3E 0449 R 03           4987            CMP     CRT_MODE,3
1B0B  76 4A                     4988            JBE     M80
                                4989
1B0D  A0 0466 R                 4990            MOV     AL,CRT_PALETTE
1B10  24 DF                     4991            AND     AL,0DFH
1B12  80 E3 01                  4992            AND     BL,1
1B15  74 02                     4993            JZ      M22
1B17  0C 20                     4994            OR      AL,020H
1B19                            4995    M22:
1B19  A2 0466 R                 4996            MOV     CRT_PALETTE,AL
1B1C  24 10                     4997            AND     AL,010H
1B1E  0C 02                     4998            OR      AL,2
1B20  0A D8                     4999            OR      BL,AL
1B22  B4 01                     5000            MOV     AH,1
1B24  8A C3                     5001            MOV     AL,BL
1B26  E8 1D9C R                 5002            CALL    PAL_SET
                                5003
1B29  0B ED                     5004            or      bp,bp
1B2B  74 04                     5005            jz      m22y
1B2D  26: 88 5D 01              5006            mov     es:[di][1],bl
1B31                            5007    m22y:
                                5008
1B31  FE C3                     5009            INC     BL
1B33  FE C3                     5010            INC     BL
1B35  B4 02                     5011            MOV     AH,2
1B37  8A C3                     5012            MOV     AL,BL
1B39  E8 1D9C R                 5013            CALL    PAL_SET
                                5014
1B3C  0B ED                     5015            or      bp,bp
1B3E  74 04                     5016            jz      m27y
1B40  26: 88 5D 02              5017            mov     es:[di][2],bl
1B44                            5018    m27y:
                                5019
1B44  FE C3                     5020            INC     BL
1B46  FE C3                     5021            INC     BL
1B48  B4 03                     5022            MOV     AH,3
1B4A  8A C3                     5023            MOV     AL,BL
1B4C  E8 1D9C R                 5024            CALL    PAL_SET
                                5025
1B4F  0B ED                     5026            or      bp,bp
1B51  74 04                     5027            jz      m80
1B53  26: 88 5D 03              5028            mov     es:[di][3],bl
                                5029
1B57                            5030    M80:
1B57  E8 1DB4 R                 5031            CALL    PAL_ON
1B5A  E9 219B R                 5032            JMP     V_RET
                                5033
                                5034    C       INCLUDE         VDOT.INC
                                5035    C       SUBTTL  VDOT.INC
                                5036    C       PAGE
                                5037    C ;-----------------------------------------------
                                5038    C ; ENTRY                                       :
                                5039    C ;       DX = ROW                              :
                                5040    C ;       CX = COLUMN                           :
```

```
                                5041     C  ;       BH = PAGE                                       :
                                5042     C  ; EXIT                                                :
                                5043     C  ;       BX = OFFSET INTO REGEN                        :
                                5044     C  ;       AL = BIT MASK FOR COLUMN BYTE                 :
                                5045     C  ;-------------------------------------------------
1B5D                            5046     C  DOT_SUP_1        PROC    NEAR
                                5047     C
                                5048     C  ;----- OFFSET = PAGE OFFSET + ROW * BYTES/ROW + COLUMN/8
                                5049     C
1B5D  F7 26 044A R              5050     C          MUL     WORD PTR CRT_COLS               ; ROW * BYTES/ROW
1B61  51                        5051     C          PUSH    CX                              ; SAVE COLUMN VALUE
1B62  D1 E9                     5052     C          SHR     CX,1                            ; DIVIDE BY EIGHT TO
1B64  D1 E9                     5053     C          SHR     CX,1                            ;  DETERMINE THE BYTE THAT
1B66  D1 E9                     5054     C          SHR     CX,1                            ;  THIS DOT IS IN
                                5055     C                                                  ;  (8 BITS/BYTE)
1B68  03 C1                     5056     C          ADD     AX,CX                           ; BYTE OFFSET INTO PAGE
1B6A  8A DF                     5057     C          MOV     BL,BH                           ; GET PAGE INTO BL
1B6C  2A FF                     5058     C          SUB     BH,BH                           ; ZERO
1B6E  8B CB                     5059     C          MOV     CX,BX                           ; COUNT VALUE
1B70  8B 1E 044C R              5060     C          MOV     BX,CRT_LEN                      ; LENGTH OF ONE PAGE
1B74  E3 04                     5061     C          JCXZ    DS_2                            ; PAGE ZERO
1B76                            5062     C  DS_3:
1B76  03 C3                     5063     C          ADD     AX,BX                           ; BUMP TO NEXT PAGE
1B78  E2 FC                     5064     C          LOOP    DS_3                            ; DO FOR THE REST
1B7A                            5065     C  DS_2:
1B7A  59                        5066     C          POP     CX                              ; RECOVER COLUMN VALUE
1B7B  8B D8                     5067     C          MOV     BX,AX                           ; REGEN OFFSET
1B7D  80 E1 07                  5068     C          AND     CL,07H                          ; SHIFT COUNT FOR BIT MASK
1B80  B0 80                     5069     C          MOV     AL,080H                         ; MASK BIT
1B82  D2 E8                     5070     C          SHR     AL,CL                           ; POSITION MASK BIT
1B84  C3                        5071     C          RET
1B85                            5072     C  DOT_SUP_1        ENDP
                                5073     C
                                5074     C  ;--------------------------------------------------------
                                5075     C  ; THIS SUBROUTINE DETERMINES THE REGEN BYTE LOCATION    :
                                5076     C  ; OF THE INDICATED ROW COLUMN VALUE IN GRAPHICS MODE.   :
                                5077     C  ; ENTRY --                                              :
                                5078     C  ;  DX = ROW VALUE (0-199)                               :
                                5079     C  ;  CX = COLUMN VALUE (0-639)                            :
                                5080     C  ; EXIT --                                               :
                                5081     C  ;  SI = OFFSET INTO REGEN BUFFER FOR BYTE OF INTEREST   :
                                5082     C  ;  AH = MASK TO STRIP OFF THE BITS OF INTEREST          :
                                5083     C  ;  CL = BITS TO SHIFT TO RIGHT JUSTIFY THE MASK IN AH   :
                                5084     C  ;  DH = # BITS IN RESULT                                :
                                5085     C  ;--------------------------------------------------------
1B85                            5086     C  R3       PROC    NEAR
1B85  53                        5087     C          PUSH    BX                              ; SAVE BX DURING OPERATION
1B86  50                        5088     C          PUSH    AX                              ; WILL SAVE AL DURING OPERATION
                                5089     C
                                5090     C  ;----- DETERMINE 1ST BYTE IN IDICATED ROW BY MULTIPLYING ROW VALUE BY 40
                                5091     C  ;----- ( LOW BIT OF ROW DETERMINES EVEN/ODD, 80 BYTES/ROW
                                5092     C
1B87  B0 28                     5093     C          MOV     AL,40
1B89  52                        5094     C          PUSH    DX                              ; SAVE ROW VALUE
1B8A  80 E2 FE                  5095     C          AND     DL,0FEH                         ; STRIP OFF ODD/EVEN BIT
1B8D  F6 E2                     5096     C          MUL     DL                              ; AX HAS ADDRESS OF 1ST BYTE
                                5097     C                                                  ;  OF INDICATED ROW
1B8F  5A                        5098     C          POP     DX                              ; RECOVER IT
1B90  F6 C2 01                  5099     C          TEST    DL,1                            ; TEST FOR EVEN/ODD
1B93  74 03                     5100     C          JZ      R4                              ; JUMP IF EVEN ROW
1B95  05 2000                   5101     C          ADD     AX,2000H                        ; OFFSET TO LOCATION OF ODD ROWS
1B98                            5102     C  R4:                                             ; EVEN_ROW
1B98  8B F0                     5103     C          MOV     SI,AX                           ; MOVE POINTER TO SI
1B9A  58                        5104     C          POP     AX                              ; RECOVER AL VALUE
1B9B  8B D1                     5105     C          MOV     DX,CX                           ; COLUMN VALUE TO DX
                                5106     C
                                5107     C  ;----- DETERMINE GRAPHICS MODE CURRENTLY IN EFFECT
                                5108     C
                                5109     C  ;--------------------------------------------------------
                                5110     C  ; SET UP THE REGISTERS ACCORDING TO THE MODE            :
                                5111     C  ;  CH = MASK FOR LOW OF COLUMN ADDRESS ( 7/3 FOR HIGH/MED RES)  :
                                5112     C  ;  CL = # OF ADDRESS BITS IN COLUMN VALUE ( 3/2 FOR H/M)       :
                                5113     C  ;  BL = MASK TO SELECT BITS FROM POINTED BYTE (80H/C0H FOR H/M) :
                                5114     C  ;  BH = NUMBER OF VALID BITS IN POINTED BYTE ( 1/2 FOR H/M)     :
                                5115     C  ;--------------------------------------------------------
                                5116     C
1B9D  BB 02C0                   5117     C          MOV     BX,2C0H
1BA0  B9 0302                   5118     C          MOV     CX,302H                         ; SET PARMS FOR MED RES
1BA3  80 3E 0449 R 06           5119     C          CMP     CRT_MODE,6
1BA8  72 06                     5120     C          JC      R5                              ; HANDLE IF MED ARES
1BAA  BB 0180                   5121     C          MOV     BX,180H
1BAD  B9 0703                   5122     C          MOV     CX,703H                         ; SET PARMS FOR HIGH RES
                                5123     C
                                5124     C  ;----- DETERMINE BIT OFFSET IN BYTE FROM COLUMN MASK
                                5125     C
1BB0                            5126     C  R5:
1BB0  22 EA                     5127     C          AND     CH,DL                           ; ADDRESS OF PEL WITHIN BYTE TO CH
                                5128     C
                                5129     C  ;----- DETERMINE BYTE OFFSET FOR THIS LOCATION IN COLUMN
                                5130     C
1BB2  D3 EA                     5131     C          SHR     DX,CL                           ; SHIFT BY CORRECT AMOUNT
1BB4  03 F2                     5132     C          ADD     SI,DX                           ; INCREMENT THE POINTER
1BB6  8A F7                     5133     C          MOV     DH,BH                           ; GET THE # OF BITS IN RESULT TO DH
                                5134     C
                                5135     C  ;----- MULTIPLY BH (VALID BITS IN BYTE) BY CH (BIT OFFSET)
                                5136     C
1BB8  2A C9                     5137     C          SUB     CL,CL                           ; ZERO INTO STORAGE LOCATION
1BBA                            5138     C  R6:
1BBA  D0 C8                     5139     C          ROR     AL,1                            ; LEFT JUSTIFY THE VALUE
                                5140     C                                                  ;  IN AL (FOR WRITE)
1BBC  02 CD                     5141     C          ADD     CL,CH                           ; ADD IN THE BIT OFFSET VALUE
1BBE  FE CF                     5142     C          DEC     BH                              ; LOOP CONTROL
1BC0  75 F8                     5143     C          JNZ     R6                              ; ON EXIT, CL HAS SHIFT COUNT
                                5144     C                                                  ;  TO RESTORE BITS
1BC2  8A E3                     5145     C          MOV     AH,BL                           ; GET MASK TO AH
1BC4  D2 EC                     5146     C          SHR     AH,CL                           ; MOVE THE MASK TO CORRECT LOCATION
1BC6  5B                        5147     C          POP     BX                              ; RECOVER REG
1BC7  C3                        5148     C          RET                                     ; RETURN WITH EVERYTHING SET UP
1BC8                            5149     C  R3       ENDP
                                5150     C
                                5151     C  ;--------------------------------------------------------
                                5152     C  ; READ DOT  -- WRITE DOT                                :
                                5153     C  ;       THESE ROUTINES WILL WRITE A DOT, OR READ THE DOT AT     :
                                5154     C  ;       THE INDICATED LOCATION                          :
                                5155     C  ; ENTRY --                                              :
                                5156     C  ;   DX = ROW (0-199)    (THE ACTUAL VALUE DEPENDS ON THE MODE)  :
                                5157     C  ;   CX = COLUMN ( 0-639) ( THE VALUES ARE NOT RANGE CHECKED )   :
                                5158     C  ;   AL = DOT VALUE TO WRITE (1,2 OR 4 BITS DEPENDING ON MODE,   :
                                5159     C  ;        REQ'D FOR WRITE DOT ONLY, RIGHT JUSTIFIED)     :
                                5160     C  ;        BIT 7 OF AL=1 INDICATES XOR THE VALUE INTO THE LOCATION :
                                5161     C  ;   DS = DATA SEGMENT                                    :
                                5162     C  ;   ES = REGEN SEGMENT                                   :
                                5163     C  ;                                                       :
                                5164     C  ; EXIT                                                  :
                                5165     C  ;       AL = DOT VALUE READ, RIGHT JUSTIFIED, READ ONLY         :
                                5166     C  ;--------------------------------------------------------
```

```
                                5167     C
                                5168     C  ;----- WRITE DOT
                                5169     C
1BC8                            5170     C  AHC:
                                5171     C          ASSUME  DS:ABS0
1BC8  80 3E 0449 R 07           5172     C          cmp     crt_mode,7
1BCD  77 2A                     5173     C          ja      write_dot_2
                                5174     C
1BCF                            5175     C  WRITE_DOT       PROC    NEAR
                                5176     C          assume  ds:abs0,es:nothing
1BCF  52                        5177     C          push    dx
                                5178     C          srload  es,0b800h
1BD0  BA B800                   5179     C+         MOV     DX,0b800h
1BD3  8E C2                     5180     C+         MOV     es,DX
1BD5  5A                        5181     C          pop     dx
1BD6  50                        5182     C          PUSH    AX                      ; SAVE DOT VALUE
1BD7  50                        5183     C          PUSH    AX                      ;  TWICE
1BD8  E8 1B85 R                 5184     C          CALL    R3                      ; DETERMINE BYTE POSITION OF THE DOT
1BDB  D2 E8                     5185     C          SHR     AL,CL                   ; SHIFT TO SET UP THE BITS FOR OUTPUT
1BDD  22 C4                     5186     C          AND     AL,AH                   ; STRIP OFF THE OTHER BITS
1BDF  26: 8A 0C                 5187     C          MOV     CL,ES:[SI]              ; GET THE CURRENT BYTE
1BE2  5B                        5188     C          POP     BX                      ; RECOVER XOR FLAG
1BE3  F6 C3 80                  5189     C          TEST    BL,80H                  ; IS IT ON
1BE6  75 0D                     5190     C          JNZ     R2                      ; YES, XOR THE DOT
1BE8  F6 D4                     5191     C          NOT     AH                      ; SET THE MASK TO REMOVE THE
1BEA  22 CC                     5192     C          AND     CL,AH                   ;  INDICATED BITS
1BEC  0A C1                     5193     C          OR      AL,CL                   ; OR IN THE NEW VALUE OF THOSE BITS
1BEE                            5194     C  R1:                                     ; FINISH_DOT
1BEE  26: 88 04                 5195     C          MOV     ES:[SI],AL              ; RESTORE THE BYTE IN MEMORY
1BF1  58                        5196     C          POP     AX
1BF2  E9 219B R                 5197     C          jmp     v_ret
1BF5                            5198     C  R2:                                     ; XOR_DOT
1BF5  32 C1                     5199     C          XOR     AL,CL                   ; EXCLUSIVE OR THE DOTS
1BF7  EB F5                     5200     C          JMP     R1                      ; FINISH UP THE WRITING
1BF9                            5201     C  WRITE_DOT       ENDP
                                5202     C
1BF9                            5203     C  WRITE_DOT_2     PROC    NEAR
1BF9  80 3E 0449 R 0F           5204     C          cmp     crt_mode,0fh
1BFE  72 0D                     5205     C          jb      no_adj2
1C00  E8 14F4 R                 5206     C          call    mem_det                         ; base card
1C03  72 08                     5207     C          jc      no_adj2
1C05  24 85                     5208     C          and     al,10000101b            ; 85h, xor c2 c0 mask
1C07  8A E0                     5209     C          mov     ah,al
1C09  D0 E4                     5210     C          shl     ah,1                    ; expand c0 to c1, c2 to c3
1C0B  0A C4                     5211     C          or      al,ah                   ; build ?(80h) + (0,3,c,f)
1C0D                            5212     C  no_adj2:
1C0D  50                        5213     C          push    ax
1C0E  8B C2                     5214     C          MOV     AX,DX                   ; ROW VALUE
1C10  E8 1B5D R                 5215     C          CALL    DOT_SUP_1               ; BX=OFFSET, AL=BIT MASK
1C13  B6 03                     5216     C          mov     dh,3
1C15  B2 CE                     5217     C          MOV     DL,GRAPH_ADDR           ; GRAPHICS CHIP
1C17  B4 08                     5218     C          MOV     AH,G_BIT_MASK           ; BIT MASK REGISTER
1C19  E8 0D18 R                 5219     C          CALL    OUT_DX                  ; SET BIT MASK
1C1C  52                        5220     C          PUSH    DX
                                5221     C          SRLOAD  ES,0A000H               ; REGEN SEGMENT
1C1D  BA A000                   5222     C+         MOV     DX,0A000H
1C20  8E C2                     5223     C+         MOV     ES,DX
1C22  5A                        5224     C          POP     DX
1C23  58                        5225     C          POP     AX                      ; RECOVER COLOR
1C24  8A E8                     5226     C          MOV     CH,AL                   ; SAVE COLOR
1C26  F6 C5 80                  5227     C          TEST    CH,080H                 ; SEE IF XOR
1C29  74 0A                     5228     C          JZ      WD_A                    ; NO XOR
1C2B  B4 03                     5229     C          MOV     AH,G_DATA_ROT           ; DO XOR
1C2D  B0 18                     5230     C          MOV     AL,018H                 ; XOR FUNCTION
1C2F  E8 0D18 R                 5231     C          CALL    OUT_DX                  ; SET THE REGISTER
1C32  EB 12 90                  5232     C          JMP     WD_B                    ; SKIP THE BLANK
1C35                            5233     C  WD_A:                                   ; BLANK THE DOT
1C35  B2 C4                     5234     C          MOV     DL,SEQ_ADDR             ; SEQUENCER
1C37  B4 02                     5235     C          MOV     AH,S_MAP                ; MAP MASK
1C39  B0 FF                     5236     C          MOV     AL,0FFH                 ; ENABLE ALL MAPS
1C3B  E8 0D18 R                 5237     C          CALL    OUT_DX                  ; SET THE REGISTER
1C3E  26: 8A 07                 5238     C          MOV     AL,ES:[BX]              ; LATCH DATA
1C41  2A C0                     5239     C          SUB     AL,AL                   ; ZERO
1C43  26: 88 07                 5240     C          MOV     ES:[BX],AL              ; BLANK THE DOT
1C46                            5241     C  WD_B:                                   ; SET THE COLOR MAP MASK
1C46  B2 C4                     5242     C          MOV     DL,SEQ_ADDR             ; SEQUENCER
1C48  B4 02                     5243     C          MOV     AH,S_MAP                ; MAP MASK REGISTER
1C4A  8A C5                     5244     C          MOV     AL,CH                   ; COLOR VALUE
1C4C  24 0F                     5245     C          AND     AL,0FH                  ; VALUES 0-15
1C4E  E8 0D18 R                 5246     C          CALL    OUT_DX                  ; SET IT
1C51  26: 8A 07                 5247     C          MOV     AL,ES:[BX]              ; LATCH DATA
1C54  B0 FF                     5248     C          MOV     AL,0FFH                 ; WRITE VALUE
1C56  26: 88 07                 5249     C          MOV     ES:[BX],AL              ; SET THE DOT
                                5250     C
                                5251     C  ;----- NORMALIZE THE ENVIRONMENT
                                5252     C
1C59  E8 0D18 R                 5253     C          CALL    OUT_DX                  ; ALL MAPS ON
1C5C  B2 CE                     5254     C          MOV     DL,GRAPH_ADDR           ; GRAPHICS CHIPS
1C5E  B4 03                     5255     C          MOV     AH,G_DATA_ROT           ; XOR REGISTER
1C60  2A C0                     5256     C          SUB     AL,AL                   ; NORMAL WRITES
1C62  E8 0D18 R                 5257     C          CALL    OUT_DX                  ; SET IT
1C65  B4 08                     5258     C          MOV     AH,G_BIT_MASK           ; BIT MASK
1C67  B0 FF                     5259     C          MOV     AL,0FFH                 ; ALL BITS ON
1C69  E8 0D18 R                 5260     C          CALL    OUT_DX                  ; SET IT
1C6C  E9 219B R                 5261     C          JMP     V_RET                   ; WRITE DOT DONE
1C6F                            5262     C  WRITE_DOT_2     ENDP
                                5263     C
1C6F                            5264     C  RD_S    PROC    NEAR
                                5265     C          ASSUME  DS:ABS0
1C6F  50                        5266     C          PUSH    AX
1C70  52                        5267     C          PUSH    DX
                                5268     C          SRLOAD  ES,0A000H
1C71  BA A000                   5269     C+         MOV     DX,0A000H
1C74  8E C2                     5270     C+         MOV     ES,DX
1C76  5A                        5271     C          POP     DX
1C77  58                        5272     C          POP     AX
1C78  8B C2                     5273     C          MOV     AX,DX
1C7A  E8 185D R                 5274     C          CALL    DOT_SUP_1
1C7D  B5 07                     5275     C          MOV     CH,7
1C7F  2A E9                     5276     C          SUB     CH,CL
1C81  2B D2                     5277     C          SUB     DX,DX
1C83  B0 00                     5278     C          MOV     AL,0
1C85  C3                        5279     C          RET
1C86                            5280     C  RD_S    ENDP
                                5281     C
1C86                            5282     C  RD_1S   PROC    NEAR
1C86  8A CD                     5283     C          MOV     CL,CH
1C88  B4 04                     5284     C          MOV     AH,4
1C8A  52                        5285     C          PUSH    DX
1C8B  B6 03                     5286     C          mov     dh,3
1C8D  B2 CE                     5287     C          MOV     DL,GRAPH_ADDR
1C8F  E8 0D18 R                 5288     C          CALL    OUT_DX
1C92  5A                        5289     C          POP     DX
1C93  26: 8A 27                 5290     C          MOV     AH,ES:[BX]
1C96  D2 EC                     5291     C          SHR     AH,CL
1C98  80 E4 01                  5292     C          AND     AH,1
```

```
1C9B  C3                      5293    C               RET
1C9C                          5294    C  RD_1S       ENDP
                              5295    C
                              5296    C  ;----- READ DOT
                              5297    C
1C9C                          5298    C  AHD:
                              5299    C               ASSUME  DS:ABS0
1C9C  80 3E 0449 R 07         5300    C               cmp     crt_mode,7
1CA1  77 18                   5301    C               ja      r_1
                              5302    C
1CA3                          5303    C  READ_DOT     PROC    NEAR
                              5304    C               assume  ds:abs0,es:nothing
1CA3  52                      5305    C               push    dx
                              5306    C               srload  es,0b800h
1CA4  BA B800                 5307    C+              MOV     DX,0B800H
1CA7  8E C2                   5308    C+              MOV     es,DX
1CA9  5A                      5309    C               pop     dx
1CAA  E8 1B85 R               5310    C               CALL    R3                      ; DETERMINE BYTE POSITION OF DOT
1CAD  26: 8A 04               5311    C               MOV     AL,ES:[SI]              ; GET THE BYTE
1CB0  22 C4                   5312    C               AND     AL,AH                   ; MASK OFF THE OTHER BITS IN THE BYTE
1CB2  D2 E0                   5313    C               SHL     AL,CL                   ; LEFT JUSTIFY THE VALUE
1CB4  8A CE                   5314    C               MOV     CL,DH                   ; GET NUMBER OF BITS IN RESULT
1CB6  D2 C0                   5315    C               ROL     AL,CL                   ; RIGHT JUSTIFY THE RESULT
1CB8  E9 219B R               5316    C               jmp     v_ret
1CBB                          5317    C  READ_DOT     ENDP
                              5318    C
1CBB                          5319    C  R_1:
1CBB  80 3E 0449 R 0F         5320    C               cmp     crt_mode,0fh
1CC0  72 25                   5321    C               jb      read_dot_2
1CC2  E8 14F4 R               5322    C               call    mem_det
1CC5  72 20                   5323    C               jc      read_dot_2
                              5324    C
1CC7                          5325    C  READ_DOT_1   PROC    NEAR                    ; 2 MAPS
                              5326    C               ASSUME  DS:ABS0, ES:NOTHING
1CC7  E8 1C6F R               5327    C               CALL    RD_S
1CCA  E8 1C86 R               5328    C               CALL    RD_1S
1CCD  0A D4                   5329    C               OR      DL,AH
1CCF  D0 E4                   5330    C               SHL     AH,1
1CD1  0A D4                   5331    C               OR      DL,AH
1CD3  B0 02                   5332    C               MOV     AL,2
1CD5  E8 1C86 R               5333    C               CALL    RD_1S
1CD8  D0 E4                   5334    C               SHL     AH,1
1CDA  D0 E4                   5335    C               SHL     AH,1
1CDC  0A D4                   5336    C               OR      DL,AH
1CDE  D0 E4                   5337    C               SHL     AH,1
1CE0  0A D4                   5338    C               OR      DL,AH
1CE2  8A C2                   5339    C               MOV     AL,DL
1CE4  E9 219B R               5340    C               JMP     V_RET
1CE7                          5341    C  READ_DOT_1   ENDP
                              5342    C
1CE7                          5343    C  READ_DOT_2   PROC    NEAR                    ; 4 MAPS
                              5344    C               ASSUME  DS:ABS0, ES:NOTHING
1CE7  E8 1C6F R               5345    C               CALL    RD_S
1CEA                          5346    C  RD_2A:
1CEA  E8 1C86 R               5347    C               CALL    RD_1S
1CED  8A C8                   5348    C               MOV     CL,AL
1CEF  D2 E4                   5349    C               SHL     AH,CL
1CF1  0A D4                   5350    C               OR      DL,AH
1CF3  FE C0                   5351    C               INC     AL
1CF5  3C 03                   5352    C               CMP     AL,3
1CF7  76 F1                   5353    C               JBE     RD_2A
1CF9  8A C2                   5354    C               MOV     AL,DL
1CFB  E9 219B R               5355    C               JMP     V_RET
1CFE                          5356    C  READ_DOT_2   ENDP
                              5357    C
                              5358    C  ;--------------------------------------------------------------------------
                              5359    C  ; WRITE_TTY     WRITE TELETYPE TO ACTIVE PAGE                          :
                              5360    C  ;       THIS INTERFACE PROVIDES A TELETYPE LIKE INTERFACE TO THE VIDEO  :
                              5361    C  ;       CARD.    THE INPUT CHARACTER  IS WRITTEN TO THE  CURRENT CURSOR  :
                              5362    C  ;       POSITION, AND THE CURSOR IS MOVED TO THE NEXT POSITION. IF THE  :
                              5363    C  ;       CURSOR LEAVES THE LAST COLUMN OF THE FIELD,  THE COLUMN IS SET  :
                              5364    C  ;       TO ZERO,  AND THE ROW VALUE IS INCREMENTED.   IF THE ROW VALUE  :
                              5365    C  ;       LEAVES THE FIELD,  THE CURSOR IS PLACED ON THE LAST ROW, FIRST  :
                              5366    C  ;       COLUMN.  AND THE ENTIRE SCREEN IS SCROLLED UP ONE LINE.   WHEN  :
                              5367    C  ;       THE SCREEN IS SCROLLED UP, THE ATTRIBUTE FOR FILLING THE NEWLY  :
                              5368    C  ;       BLANKED LINE IS READ FROM THE  CURSOR POSITION ON THE PREVIOUS  :
                              5369    C  ;       LINE BEFORE THE SCROLL,  IN CHARACTER MODE.  IN GRAPHICS MODE,  :
                              5370    C  ;       THE 0 COLOR IS USED.                                            :
                              5371    C  ; ENTRY                                                                 :
                              5372    C  ;       (AH) = CURRENT CRT MODE                                         :
                              5373    C  ;       (AL) = CHARACTER TO BE WRITTEN                                  :
                              5374    C  ;        NOTE THAT BACK SPACE, CAR RET, BELL AND LINE FEED ARE HANDLED  :
                              5375    C  ;        AS COMMANDS RATHER THAN AS DISPLAYABLE GRAPHICS                :
                              5376    C  ;       (BL) = FOREGROUND COLOR FOR CHAR WRITE IF CURRENTLY IN A         :
                              5377    C  ;        GRAPHICS MODE                                                  :
                              5378    C  ; EXIT                                                                  :
                              5379    C  ;       ALL REGISTERS SAVED                                             :
                              5380    C  ;--------------------------------------------------------------------------
1CFE                          5381    C  AHE:
                              5382    C               ASSUME  CS:CODE,DS:ABS0
1CFE  50                      5383    C               PUSH    AX                      ; SAVE REGISTERS
1CFF  8A 3E 0462 R            5384    C               MOV     BH,ACTIVE_PAGE          ; GET THE ACTIVE PAGE
1D03  53                      5385    C               PUSH    BX                      ; SAVE
1D04  8A DF                   5386    C               MOV     BL,BH                   ; GET PAGE TO BL
1D06  32 FF                   5387    C               XOR     BH,BH                   ; CLEAR HIGH BYTE
1D08  D1 E3                   5388    C               SAL     BX,1                    ; *2 FOR WORD OFFSET
1D0A  8B 97 0450 R            5389    C               MOV     DX,[BX + OFFSET CURSOR_POSN]    ; CURSOR, ACTIVE PAGE
1D0E  5B                      5390    C               POP     BX                      ; RECOVER
                              5391    C
                              5392    C  ;----- DX NOW HAS THE CURRENT CURSOR POSITION
                              5393    C
1D0F  3C 0D                   5394    C               CMP     AL,0DH                  ; IS IT CARRIAGE RETURN
1D11  74 5C                   5395    C               JE      U9                      ; CAR_RET
1D13  3C 0A                   5396    C               CMP     AL,0AH                  ; IS IT A LINE FEED
1D15  74 5C                   5397    C               JE      U10                     ; LINE_FEED
1D17  3C 08                   5398    C               CMP     AL,08H                  ; IS IT A BACKSPACE
1D19  74 4C                   5399    C               JE      U8                      ; BACK_SPACE
1D1B  3C 07                   5400    C               CMP     AL,07H                  ; IS IT A BELL
1D1D  74 5C                   5401    C               JE      U11                     ; BELL
                              5402    C
                              5403    C  ;----- WRITE THE CHAR TO THE SCREEN
                              5404    C
1D1F  B4 0A                   5405    C               MOV     AH,10                   ; WRITE CHAR ONLY
1D21  B9 0001                 5406    C               MOV     CX,1                    ; ONLY ONE CHAR
1D24  CD 10                   5407    C               INT     10H                     ; WRITE THE CHAR
                              5408    C
                              5409    C  ;----- POSITION THE CURSOR FOR NEXT CHAR
                              5410    C
1D26  FE C2                   5411    C               INC     DL
1D28  3A 16 044A R            5412    C               CMP     DL,BYTE PTR CRT_COLS    ; TEST FOR COLUMN OVERFLOW
1D2C  75 35                   5413    C               JNZ     U7                      ; SET_CURSOR
1D2E  2A D2                   5414    C               SUB     DL,DL                   ; COLUMN FOR CURSOR
1D30  3A 36*0484 R            5415    C               CMP     DH,ROWS
1D34  75 28                   5416    C               JNZ     U6                      ; SET_CURSOR_INC
                              5417    C
                              5418    C  ;----- SCROLL REQUIRED
```

```
                                5419    C
                                5420    C  U1:
1D36                            5421    C          CALL    SET_CPOS                        ; SET THE CURSOR
1D36  E8 115A R                 5422    C
                                5423    C  ;----- DETERMINE VALUE TO FILL WITH DURING SCROLL
                                5424    C
1D39  A0 0449 R                 5425    C          MOV     AL,CRT_MODE                     ; GET THE CURRENT MODE
1D3C  3C 04                     5426    C          CMP     AL,4
1D3E  72 06                     5427    C          JB      U2                              ; READ-CURSOR
1D40  2A FF                     5428    C          SUB     BH,BH                           ; FILL WITH BACKGROUND
1D42  3C 07                     5429    C          CMP     AL,7
1D44  75 06                     5430    C          JNE     U3                              ; SCROLL-UP
1D46                            5431    C  U2:                                             ; READ-CURSOR
1D46  B4 08                     5432    C          MOV     AH,8
1D48  CD 10                     5433    C          INT     10H                             ; READ CHAR/ATTR
1D4A  8A FC                     5434    C          MOV     BH,AH                           ; STORE IN BH
1D4C                            5435    C  U3:                                             ; SCROLL-UP
1D4C  B8 0601                   5436    C          MOV     AX,601H                         ; SCROLL ONE LINE
1D4F  2B C9                     5437    C          SUB     CX,CX                           ; UPPER LEFT CORNER
1D51  8A 36 0484 R              5438    C          MOV     DH,ROWS                         ; LOWER RIGHT ROW
1D55  8A 16 044A R              5439    C          MOV     DL,BYTE PTR CRT_COLS            ; LOWER RIGHT COLUMN
1D59  FE CA                     5440    C          DEC     DL
1D5B                            5441    C  U4:                                             ; VIDEO-CALL-RETURN
1D5B  CD 10                     5442    C          INT     10H                             ; SCROLL UP THE SCREEN
1D5D                            5443    C  U5:                                             ; TTY-RETURN
1D5D  58                        5444    C          POP     AX                              ; RESTORE THE CHARACTER
1D5E  E9 219B R                 5445    C          JMP     V_RET                           ; RETURN TO CALLER
1D61                            5446    C  U6:                                             ; SET-CURSOR-INC
1D61  FE C6                     5447    C          INC     DH                              ; NEXT ROW
1D63                            5448    C  U7:                                             ; SET-CURSOR
1D63  B4 02                     5449    C          MOV     AH,2
1D65  EB F4                     5450    C          JMP     U4                              ; ESTABLISH THE NEW CURSOR
                                5451    C
                                5452    C  ;----- BACK SPACE FOUND
                                5453    C
1D67                            5454    C  U8:
1D67  0A D2                     5455    C          OR      DL,DL                           ; ALREADY AT END OF LINE
1D69  74 F8                     5456    C          JZ      U7                              ; SET CURSOR
1D6B  FE CA                     5457    C          DEC     DL                              ; NO -- JUST MOVE IT BACK
1D6D  EB F4                     5458    C          JMP     U7                              ; SET_CURSOR
                                5459    C
                                5460    C  ;----- CARRIAGE RETURN FOUND
                                5461    C
1D6F                            5462    C  U9:
1D6F  2A D2                     5463    C          SUB     DL,DL                           ; MOVE TO FIRST COLUMN
1D71  EB F0                     5464    C          JMP     U7                              ; SET_CURSOR
                                5465    C
                                5466    C  ;----- LINE FEED FOUND
                                5467    C
1D73                            5468    C  U10:
1D73  3A 36 0484 R              5469    C          CMP     DH,ROWS                         ; BOTTOM OF SCREEN
1D77  75 E8                     5470    C          JNE     U6                              ; YES, SCROLL THE SCREEN
1D79  EB BB                     5471    C          JMP     U1                              ; NO, JUST SET THE CURSOR
                                5472    C
                                5473    C  ;----- BELL FOUND
                                5474    C
1D7B                            5475    C  U11:
1D7B  B3 02                     5476    C          MOV     BL,2                            ; SET UP COUNT FOR BEEP
1D7D  E8 0D23 R                 5477    C          CALL    BEEP                            ; SOUND THE POD BELL
1D80  EB DB                     5478    C          JMP     U5                              ; TTY_RETURN
                                5479    C
                                5480    C
                                5481    C  ;----- CURRENT VIDEO STATE
                                5482    C
1D82                            5483    C  AHF:
                                5484    C          ASSUME  DS:ABS0
1D82  8A 26 044A R              5485    C          MOV     AH,BYTE PTR CRT_COLS            ; GET NUMBER OF COLUMNS
1D86  8A 3E 0462 R              5486    C          MOV     BH,ACTIVE_PAGE
1D8A  A0 0487 R                 5487    C          MOV     AL,INFO
1D8D  24 80                     5488    C          AND     AL,080H
1D8F  0A 06 0449 R              5489    C          OR      AL,CRT_MODE
1D93  5F                        5490    C          POP     DI
1D94  5E                        5491    C          POP     SI
1D95  59                        5492    C          POP     CX                              ; DISCARD BX
1D96  59                        5493    C          POP     CX
1D97  5A                        5494    C          POP     DX
1D98  1F                        5495    C          POP     DS
1D99  07                        5496    C          POP     ES
1D9A  5D                        5497    C          POP     BP
1D9B  CF                        5498    C          IRET
                                5499    C
                                5500    C          SUBTTL
                                5501
                                5502
1D9C                            5503       PAL_SET PROC    NEAR
1D9C  50                        5504               PUSH    AX
1D9D  E8 0D08 R                 5505               CALL    WHAT_BASE
1DA0  FA                        5506               CLI
1DA1                            5507       VR:
                                5508               WIN
1DA1  EC                        5509     +         IN      AL,DX
1DA2  A8 08                     5510               TEST    AL,08H                          ; VERTICAL RETRACE
1DA4  74 FB                     5511               JZ      VR
1DA6  58                        5512               POP     AX
1DA7  B2 C0                     5513               MOV     DL,ATTR_WRITE
1DA9  86 C4                     5514               XCHG    AL,AH
                                5515               WOUT
1DAB  EE                        5516     +         OUT     DX,AL
1DAC  86 C4                     5517               XCHG    AL,AH
                                5518               WOUT
1DAE  EE                        5519     +         OUT     DX,AL
1DAF  B0 20                     5520               MOV     AL,020H
                                5521               WOUT
1DB1  EE                        5522     +         OUT     DX,AL
1DB2  FB                        5523               STI
1DB3  C3                        5524               RET
1DB4                            5525       PAL_SET ENDP
                                5526
1DB4                            5527       PAL_ON  PROC    NEAR
1DB4  E8 1DBD R                 5528               CALL    PAL_INIT
1DB7  B2 C0                     5529               MOV     DL,ATTR_WRITE
1DB9  B0 20                     5530               MOV     AL,020H
                                5531               WOUT
1DBB  EE                        5532     +         OUT     DX,AL
1DBC  C3                        5533               RET
1DBD                            5534       PAL_ON  ENDP
                                5535
1DBD                            5536       PAL_INIT        PROC    NEAR
1DBD  E8 0D08 R                 5537               CALL    WHAT_BASE
                                5538               WIN
1DC0  EC                        5539     +         IN      AL,DX
1DC1  C3                        5540               RET
1DC2                            5541       PAL_INIT        ENDP
                                5542
                                5543       ;----- SET PALETTE REGISTERS
                                5544
```

```
1DC2                                    5545    AH10:
                                        5546            ASSUME  DS:ABS0
1DC2  F6 06 0487 R 02                   5547            test    info,2
1DC7  75 07                             5548            JNZ     BM_OK                   ; IN MONOCHROME MODE
                                        5549
                                        5550    ;----- HERE THE ega IS IN A COLOR MODE
                                        5551
1DC9  80 3E 0463 R 84                   5552            CMP     BYTE PTR ADDR_6845,0B4H
1DCE  74 33                             5553            JE      BM_OUT
1DD0                                    5554    BM_OK:
1DD0  8A E0                             5555            MOV     AH,AL
1DD2  0A E4                             5556            OR      AH,AH
1DD4  75 30                             5557            JNZ     BM_1
                                        5558
                                        5559    ;----- set individual register
                                        5560
1DD6  2B ED                             5561            sub     bp,bp
1DD8  C4 3E 04A8 R                      5562            les     di,save_ptr
1DDC  83 C7 04                          5563            add     di,4
1DDF  26: C4 3D                         5564            les     di,dword ptr es:[di]
1DE2  8C C0                             5565            mov     ax,es
1DE4  0B C7                             5566            or      ax,di
1DE6  74 01                             5567            jz      tlo_1
1DE8  45                                5568            inc     bp
1DE9                                    5569    tlo_1:
                                        5570
1DE9  E8 1DBD R                         5571            CALL    PAL_INIT
1DEC  8A E3                             5572            MOV     AH,BL
1DEE  8A C7                             5573            MOV     AL,BH
1DF0  E8 1D9C R                         5574            CALL    PAL_SET
1DF3  E8 1DB4 R                         5575            CALL    PAL_ON
1DF6  0B ED                             5576            or      bp,bp
1DF8  74 09                             5577            jz      bm_out
1DFA  8A C7                             5578            mov     al,bh
1DFC  2A FF                             5579            sub     bh,bh
1DFE  03 FB                             5580            add     di,bx
1E00  26: 88 05                         5581            mov     es:[di],al
1E03                                    5582    BM_OUT:
1E03  E9 219B R                         5583            JMP     V_RET
                                        5584
1E06                                    5585    BM_1:
1E06  FE CC                             5586            DEC     AH
1E08  75 2D                             5587            JNZ     BM_2
                                        5588
1E0A  2B ED                             5589            sub     bp,bp
1E0C  C4 3E 04A8 R                      5590            les     di,save_ptr
1E10  83 C7 04                          5591            add     di,4
1E13  26: C4 3D                         5592            les     di,dword ptr es:[di]
1E16  8C C0                             5593            mov     ax,es
1E18  0B C7                             5594            or      ax,di
1E1A  74 01                             5595            jz      tlo_2
1E1C  45                                5596            inc     bp
1E1D                                    5597    tlo_2:
                                        5598
                                        5599    ;----- set overscan register
                                        5600
1E1D  E8 1DBD R                         5601            CALL    PAL_INIT
1E20  B4 11                             5602            MOV     AH,011H
1E22  8A C7                             5603            MOV     AL,BH
1E24  E8 1D9C R                         5604            CALL    PAL_SET
1E27  E8 1DB4 R                         5605            CALL    PAL_ON
                                        5606
1E2A  0B ED                             5607            or      bp,bp
1E2C  74 D5                             5608            jz      bm_out
1E2E  83 C7 11                          5609            add     di,011h
1E31  26: 88 3D                         5610            mov     es:[di],bh
                                        5611
1E34  E9 219B R                         5612            JMP     V_RET
                                        5613
1E37                                    5614    BM_2:
1E37  FE CC                             5615            DEC     AH
1E39  75 40                             5616            JNZ     BM_3
                                        5617
                                        5618    ;----- set 16 palette registers and overscan register
                                        5619
1E3B  1E                                5620            push    ds
1E3C  06                                5621            push    es
                                        5622
1E3D  C4 3E 04A8 R                      5623            les     di,save_ptr
1E41  83 C7 04                          5624            add     di,4
1E44  26: C4 3D                         5625            les     di,dword ptr es:[di]            ; es:di ptr to pal save area
1E47  8C C0                             5626            mov     ax,es
1E49  0B C7                             5627            or      ax,di
1E4B  74 09                             5628            jz      tlo_3
                                        5629
1E4D  1F                                5630            pop     ds                              ; parameter es
1E4E  1E                                5631            push    ds
1E4F  8B F2                             5632            mov     si,dx                           ; parameter offset
1E51  B9 0011                           5633            mov     cx,17d
1E54  F3/ A4                            5634    rep     movsb
                                        5635
1E56                                    5636    tlo_3:
1E56  07                                5637            pop     es
1E57  1F                                5638            pop     ds
                                        5639
1E58  8B DA                             5640            MOV     BX,DX
1E5A  E8 1DBD R                         5641            CALL    PAL_INIT
1E5D  2A E4                             5642            SUB     AH,AH
1E5F                                    5643    BM_2A:
1E5F  26: 8A 07                         5644            MOV     AL,ES:[BX]
1E62  E8 1D9C R                         5645            CALL    PAL_SET
1E65  FE C4                             5646            INC     AH
1E67  43                                5647            INC     BX
1E68  80 FC 10                          5648            CMP     AH,010H
1E6B  72 F2                             5649            JB      BM_2A
1E6D  FE C4                             5650            INC     AH
1E6F  26: 8A 07                         5651            MOV     AL,ES:[BX]
1E72  E8 1D9C R                         5652            CALL    PAL_SET
1E75  E8 1DB4 R                         5653            CALL    PAL_ON
1E78  E9 219B R                         5654            JMP     V_RET
                                        5655
1E7B                                    5656    BM_3:
1E7B  FE CC                             5657            DEC     AH
1E7D  75 29                             5658            JNZ     BM_4
                                        5659
                                        5660    ;----- toggle intensify/blinking bit
                                        5661
1E7F  53                                5662            PUSH    BX
1E80  E8 0D52 R                         5663            call    set_base
1E83  83 C3 47                          5664            ADD     BX,010H + In_2
1E86  26: 8A 07                         5665            MOV     AL,ES:[BX]
1E89  5B                                5666            POP     BX
                                        5667
1E8A  0A DB                             5668            OR      BL,BL
1E8C  75 0A                             5669            JNZ     BM_6
                                        5670
```

```
                                5671            ;----- enable intensify
                                5672
1E8E  80 26 0465 R DF           5673    ,               and     crt_mode_set,11011111b
1E93  24 F7                     5674                    AND     AL,0F7H
1E95  EB 0C 90                  5675                    JMP     BM_7
1E98                            5676            BM_6:
1E98  FE CB                     5677                    DEC     BL
1E9A  75 07                     5678                    JNZ     BM_7
                                5679
                                5680            ;----- enable blink
                                5681
1E9C  80 0E 0465 R 20           5682                    or      crt_mode_set,020h
1EA1  0C 08                     5683                    OR      AL,08H
1EA3                            5684            BM_7:
1EA3  B4 10                     5685                    MOV     AH,P_MODE
1EA5  E8 1D9C R                 5686                    CALL    PAL_SET
1EA8                            5687            BM_4:
1EA8  E9 219B R                 5688                    JMP     V_RET
                                5689
                                5690    C               INCLUDE         VCHGEN.INC
                                5691    C               SUBTTL          VCHGEN.INC
                                5692    C               PAGE
                                5693    C       ;--------------------------------------------
                                5694    C       ; ENTRY                                     :
                                5695    C       ;       AL = 0  USER SPECIFIED FONT         :
                                5696    C       ;            1  8 X 14 FONT                 :
                                5697    C       ;            2  8 X 8 DOUBLE DOT            :
                                5698    C       ;       BL = BLOCK TO LOAD                  :
                                5699    C       ;--------------------------------------------
1EAB                            5700    C       CH_GEN:
1EAB  50                        5701    C               PUSH    AX                          ; SAVE THE INVOLVED REGS
1EAC  55                        5702    C               PUSH    BP
1EAD  53                        5703    C               PUSH    BX
1EAE  51                        5704    C               PUSH    CX
1EAF  52                        5705    C               PUSH    DX
1EB0  06                        5706    C               PUSH    ES
                                5707    C
                                5708    C               ASSUME  DS:ABS0
1EB1  E8 0D01 R                 5709    C               CALL    DDS                         ; SET DATA SEGMENT
1EB4  A0 0449 R                 5710    C               MOV     AL,CRT_MODE                 ; GET THE CURRENT MODE
1EB7  50                        5711    C               PUSH    AX                          ; SAVE IT
1EB8  3C 07                     5712    C               CMP     AL,7                        ; IS THIS MONOCHROME
1EBA  74 07                     5713    C               JE      H14                         ; MONOCHROME VALUES
1EBC  C6 06 0449 R 0B           5714    C               MOV     CRT_MODE,0BH                ; COLOR VALUES
1EC1  EB 05                     5715    C               JMP     SHORT H15                   ; SKIP
1EC3                            5716    C       H14:
1EC3  C6 06 0449 R 0C           5717    C               MOV     CRT_MODE,0CH                ; MONOCHROME VALUES
1EC8                            5718    C       H15:
1EC8  E8 0DAE R                 5719    C               call    set_regs
1ECB  E8 0D01 R                 5720    C               CALL    DDS                         ; RESET THE DATA SEGMENT
1ECE  58                        5721    C               POP     AX                          ; RECOVER OLD MODE VALUE
1ECF  A2 0449 R                 5722    C               MOV     CRT_MODE,AL                 ; RETURN TO LOW MEMORY
                                5723    C
1ED2  07                        5724    C               POP     ES                          ; RESTORE REGS THAT WERE
1ED3  5A                        5725    C               POP     DX                          ;  USED BY THE MODE SET
1ED4  59                        5726    C               POP     CX                          ;  ROUTINES
1ED5  5B                        5727    C               POP     BX
1ED6  5D                        5728    C               POP     BP
1ED7  58                        5729    C               POP     AX
                                5730    C
1ED8  0A C0                     5731    C               OR      AL,AL                       ; SET FLAGS
1EDA  74 17                     5732    C               JZ      DO_MAP2                     ; USER SPECIFIED FONT
1EDC  0E                        5733    C               PUSH    CS                          ; SET SEGMENT TO
1EDD  07                        5734    C               POP     ES                          ;  THIS MODULE
1EDE  2B D2                     5735    C               SUB     DX,DX                       ; ZERO OUT START OFFSET
1EE0  B9 0100                   5736    C               MOV     CX,0256D                    ; CHAR COUNT (FULL SET)
1EE3  FE C8                     5737    C               DEC     AL                          ; WHICH PARAMETER
1EE5  75 07                     5738    C               JNZ     H7                          ; MUST BE ONE
1EE7  B7 0E                     5739    C               MOV     BH,014D                     ; BYTES PER CHARACTER
1EE9  BD 0000 E                 5740    C               MOV     BP,OFFSET CGMN              ; 8 X 14 TABLE OFFSET
1EEC  EB 05                     5741    C               JMP     SHORT DO_MAP2               ; STORE IT
1EEE                            5742    C       H7:
1EEE  B7 08                     5743    C               MOV     BH,8                        ; 8 X 8 FONT
1EF0  BD 0000 E                 5744    C               MOV     BP,OFFSET CGDDOT            ; ROM 8 X 8 DOUBLE DOT
                                5745    C
                                5746    C       ;--------------------------------------------
                                5747    C       ; ALPHA CHARACTER GENERATOR LOAD            :
                                5748    C       ;                                           :
                                5749    C       ; ENTRY                                     :
                                5750    C       ;       ES:BP - POINTER TO TABLE            :
                                5751    C       ;       CX    - COUNT OF CHARS              :
                                5752    C       ;       DX    - CHAR COUNT OFFSET INTO MAP 2 :
                                5753    C       ;       BH    - BYTES PER CHARACTER         :
                                5754    C       ;       BL    - MAP 2 BLOCK TO LOAD         :
                                5755    C       ;--------------------------------------------
1EF3                            5756    C       DO_MAP2:
1EF3  06                        5757    C               PUSH    ES                          ; FONT TABLE SEGMENT
1EF4  1F                        5758    C               POP     DS                          ; ADDRESSING TO TABLE
1EF5  52                        5759    C               PUSH    DX                          ; SAVE REGISTER
                                5760    C               SRLOAD  ES,0A000H                   ; ADDRESSING TO MAP 2
1EF6  BA A000                   5761    C+              MOV     DX,0A000H
1EF9  8E C2                     5762    C+              MOV     ES,DX
1EFB  5A                        5763    C               POP     DX                          ; RECOVER REGISTER
1EFC  51                        5764    C               PUSH    CX                          ; MULTIPLY BY 020H SINCE
1EFD  B1 05                     5765    C               MOV     CL,5                        ;  MAXIMUM BYTES PER
1EFF  D3 E2                     5766    C               SHL     DX,CL                       ;  CHARACTER IS 32D=020H
1F01  59                        5767    C               POP     CX                          ; RECOVER
1F02  0A DB                     5768    C               OR      BL,BL                       ; WHICH 16K BLOCK TO LOAD
1F04  74 08                     5769    C               JZ      H3                          ; BLOCK ZERO
1F06                            5770    C       H4:
1F06  81 C2 4000                5771    C               ADD     DX,04000H                   ; INCREMENT TO NEXT BLOCK
1F0A  FE CB                     5772    C               DEC     BL                          ; ANY MORE
1F0C  75 F8                     5773    C               JNZ     H4                          ; DO ANOTHER
1F0E                            5774    C       H3:
1F0E  8A C7                     5775    C               MOV     AL,BH                       ; BYTES PER CHARACTER
1F10  2A E4                     5776    C               SUB     AH,AH                       ; ZERO
1F12  8B FA                     5777    C               MOV     DI,DX                       ; OFFSET INTO MAP
1F14  8B F5                     5778    C               MOV     SI,BP                       ; OFFSET INTO TABLE
1F16  E3 0D                     5779    C               JCXZ    LD_OVER                     ; CHARACTER COUNT
1F18                            5780    C       LD:
1F18  51                        5781    C               PUSH    CX                          ; SAVE CHARACTER COUNT
1F19  8B C8                     5782    C               MOV     CX,AX                       ; ONE ENTIRE CHARACTER
1F1B  F3/ A4                    5783    C               REP     MOVSB                       ;  AT A TIME
1F1D  2B F8                     5784    C               SUB     DI,AX                       ; ADJUST OFFSET
1F1F  83 C7 20                  5785    C               ADD     DI,020H                     ; NEXT CHARACTER POSITION
1F22  59                        5786    C               POP     CX                          ; RECOVER CHARACTER COUNT
1F23  E2 F3                     5787    C               LOOP    LD                          ; DO THE REST
1F25                            5788    C       LD_OVER:
1F25  C3                        5789    C               RET
                                5790    C
1F26                            5791    C       BRK_1:
                                5792    C               ASSUME  DS:ABS0
1F26  E8 0D01 R                 5793    C               CALL    DDS                         ; SET LOW MEMORY SEGMENT
1F29  A3 0485 R                 5794    C               MOV     POINTS,AX                   ; GET BYTES/CHARACTER
1F2C  8B 16 0463 R              5795    C               MOV     DX,ADDR_6845                ; CRTC REGISTER
1F30  80 3E 0449 R 07           5796    C               cmp     crt_mode,7
```

```
1F35  75 05                5797     C          jne     h11a
1F37  B4 14                5798     C          MOV     AH,C_UNDERLN_LOC                ; R14H
1F39  E8 0D18 R            5799     C          CALL    OUT_DX                          ; SET THE UNDERLINE LOC
1F3C                       5800     C  h11a:
1F3C  FE C8                5801     C          DEC     AL                              ; POINTS - 1
1F3E  B4 09                5802     C          MOV     AH,C_MAX_SCAN_LN                ; R09H
1F40  E8 0D18 R            5803     C          CALL    OUT_DX                          ; SET THE CHARACTER HEIGHT
1F43  FE C8                5804     C          DEC     AL                              ; POINTS - 2
                           5805     C
1F45  8A E8                5806     C          mov     ch,al                           ; cursor start
1F47  8A C8                5807     C          mov     cl,al                           ; cursor end
1F49  FE C1                5808     C          inc     cl                              ; adjust end
1F4B  B4 01                5809     C          mov     ah,1                            ; set c_type bios call
1F4D  CD 10                5810     C          int     10h                             ; set the cursor
                           5811     C
1F4F  8A 1E 0449 R         5812     C          MOV     BL,CRT_MODE                     ; GET THE CURRENT MODE
1F53  B8 015E              5813     C          MOV     AX,350D                         ; MAX SCANS ON SCREEN
1F56  80 FB 03             5814     C          CMP     BL,3                            ; 640X200 ALPHA MODES
1F59  77 08                5815     C          JA      H11                             ; MUST BE 350
1F5B  E8 0E9C R            5816     C          call    brst_det
1F5E  72 03                5817     C          jc      h11
1F60  B8 00C8              5818     C          MOV     AX,200D                         ; SET FOR 200
1F63                       5819     C  H11:
1F63  99                   5820     C          CWD                                     ; PREPARE TO DIVIDE
1F64  F7 36 0485 R         5821     C          DIV     POINTS                          ; MAX ROWS ON SCREEN
1F68  48                   5822     C          DEC     AX                              ; ADJUST
1F69  A2 0484 R            5823     C          MOV     ROWS,AL                         ; SAVE ROWS
1F6C  FE C0                5824     C          INC     AL                              ; READJUST
1F6E  2A E4                5825     C          SUB     AH,AH                           ; CLEAR
1F70  F7 26 0485 R         5826     C          MUL     POINTS                          ; ROWS*BYTES/CHAR
1F74  48                   5827     C          DEC     AX                              ; ADJUST
1F75  8B 16 0463 R         5828     C          MOV     DX,ADDR_6845                    ; CRTC ADDRESS
1F79  B4 12                5829     C          MOV     AH,C_VRT_DSP_END                ; SCANS DISPLAYED
1F7B  E8 0D18 R            5830     C          CALL    OUT_DX                          ; SET IT
1F7E  A0 0484 R            5831     C          MOV     AL,ROWS                         ; GET CHARACTER ROWS
1F81  FE C0                5832     C          INC     AL                              ; ADJUST
1F83  F6 26 044A R         5833     C          MUL     BYTE PTR CRT_COLS               ; ROWS*COLUMNS
1F87  D1 E0                5834     C          SHL     AX,1                            ; *2 FOR ALPHA MODE
1F89  05 0100              5835     C          add     ax,256d                         ; space between pages
1F8C  A3 044C R            5836     C          MOV     CRT_LEN,AX                      ; BYTES PER PAGE
1F8F  E8 0E98 R            5837     C          CALL    PH_5                            ; VIDEO ON
1F92  E9 219B R            5838     C          JMP     V_RET                           ; RETURN TO CALLER
                           5839     C
                           5840     C  ;----- LOADABLE CHARACTER GENERATOR ROUTINES
                           5841     C
1F95                       5842     C  AH11:
1F95  3C 10                5843     C          CMP     AL,010H                         ; CHECK PARAMETER
1F97  73 37                5844     C          JAE     AH11_ALPHA1                     ; NEXT STAGE
                           5845     C
                           5846     C  ;-----  ALPHA MODE ACTIVITY HERE
                           5847     C
1F99  3C 03                5848     C          CMP     AL,03H                          ; RANGE CHECK
1F9B  73 17                5849     C          JAE     H1                              ; NEXT STAGE
1F9D  E8 1EA8 R            5850     C          CALL    CH_GEN                          ; SET THE CHAR GEN
1FA0  E8 0DAE R            5851     C          call    set_regs
1FA3  E8 0E98 R            5852     C          CALL    PH_5                            ; VIDEO ON
                           5853     C          assume  ds:abs0
1FA6  E8 0D01 R            5854     C          call    dds                             ; set the data segment
1FA9  8B 0E 0460 R         5855     C          mov     cx,cursor_mode                  ; get the mode
1FAD  B4 01                5856     C          mov     ah,1                            ; set c_type
1FAF  CD 10                5857     C          int     10h                             ; emulate correct cursor
1FB1  E9 219B R            5858     C          JMP     V_RET                           ; RETURN TO CALLER
                           5859     C
                           5860     C  ;----- SET THE CHARACTER GENERATOR BLOCK SELECT REGISTER
                           5861     C
1FB4                       5862     C  H1:
1FB4  75 17                5863     C          JNE     H2                              ; NOT IN RANGE
1FB6  B6 03                5864     C          mov     dh,3
1FB8  B2 C4                5865     C          MOV     DL,SEQ_ADDR                     ; SEQUENCER
                           5866     C
1FBA  B8 0001              5867     C          mov     ax,1                            ; ah=s_reset, al=1
1FBD  E8 0D18 R            5868     C          call    out_dx
                           5869     C
1FC0  B4 03                5870     C          MOV     AH,S_CGEN                       ; CHAR BLOCK REGISTER
1FC2  8A C3                5871     C          MOV     AL,BL                           ; GET THE VALUE
1FC4  E8 0D18 R            5872     C          CALL    OUT_DX                          ; SET IT
                           5873     C
1FC7  B8 0003              5874     C          mov     ax,3                            ; ah=s_reset, al=3
1FCA  E8 0D18 R            5875     C          call    out_dx
1FCD                       5876     C  H2:
1FCD  E9 219B R            5877     C          JMP     V_RET                           ; RETURN TO CALLER
                           5878     C
1FD0                       5879     C  AH11_ALPHA1:
                           5880     C          ASSUME  DS:ABS0
1FD0  3C 20                5881     C          CMP     AL,020H
1FD2  73 26                5882     C          JAE     AH11_GRAPHICS
                           5883     C
                           5884     C  ;-----  ALPHA MODE ACTIVITY HERE
                           5885     C
1FD4  2C 10                5886     C          SUB     AL,010H                         ; ADJUST TO 0 - N
1FD6  3C 02                5887     C          CMP     AL,02H                          ; RANGE CHECK
1FD8  77 F3                5888     C          JA      H2                              ; INVALID CALL
1FDA  50                   5889     C          PUSH    AX                              ; SAVE
1FDB  53                   5890     C          PUSH    BX
1FDC  E8 1EAB R            5891     C          CALL    CH_GEN                          ; LOAD THE CHAR GEN
1FDF  E8 0DAE R            5892     C          call    set_regs
1FE2  5B                   5893     C          POP     BX
1FE3  58                   5894     C          POP     AX                              ; RESTORE
1FE4  8A E0                5895     C          MOV     AH,AL                           ; CALLING PARAMETER
1FE6  0A E4                5896     C          OR      AH,AH                           ; USER MODE
1FE8  8A C7                5897     C          MOV     AL,BH
1FEA  74 09                5898     C          JZ      H13                             ; DO NOT SET BYTES/CHAR
1FEC  B0 08                5899     C          MOV     AL,8                            ; 8 X 8 FONT
1FEE  80 FC 01             5900     C          CMP     AH,1                            ; IS THE CALL FOR MONOC
1FF1  75 02                5901     C          JNE     H13                             ; NO, LEAVE IT AT 8
1FF3  B0 0E                5902     C          MOV     AL,14D                          ; MONOC SET
1FF5                       5903     C  H13:
1FF5  2A E4                5904     C          SUB     AH,AH                           ; CLEAR UPPER BYTE
1FF7  E9 1F26 R            5905     C          JMP     BRK_1                           ; CONTINUE
                           5906     C
                           5907     C  ;-----  GRAPHICS MODE ACTIVITY HERE
                           5908     C
1FFA                       5909     C  AH11_GRAPHICS:
                           5910     C          ASSUME  DS:ABS0
1FFA  3C 30                5911     C          CMP     AL,030H
1FFC  73 6A                5912     C          JAE     AH11_INFORM
1FFE  2C 20                5913     C          SUB     AL,020H
2000  75 11                5914     C          JNZ     F10
                           5915     C
                           5916     C  ;-----  COMPATIBILITY, UPPER HALF GRAPHICS CHARACTER SET
                           5917     C
                           5918     C          ASSUME  DS:ABS0
                           5919     C          SRLOAD  DS,0
2002  2B D2                5920     C+         SUB     DX,DX
2004  8E DA                5921     C+         MOV     DS,DX
2006  FA                   5922     C          CLI
```

```
2007  89 2E 007C R          5923  C          MOV     WORD PTR EXT_PTR , BP
200B  8C 06 007E R          5924  C          MOV     WORD PTR EXT_PTR + 2 , ES
200F  FB                    5925  C          STI
2010                        5926  C  F11:
2010  E9 219B R             5927  C          JMP     V_RET
2013                        5928  C  F10:
                            5929  C          ASSUME  DS:ABS0
2013  52                    5930  C          PUSH    DX
                            5931  C          SRLOAD  DS,0
2014  2B D2                 5932  C+         SUB     DX,DX
2016  8E DA                 5933  C+         MOV     DS,DX
2018  5A                    5934  C          POP     DX
2019  3C 03                 5935  C          CMP     AL,03H                          ; RANGE CHECK
201B  77 F3                 5936  C          JA      F11
201D  FE C8                 5937  C          DEC     AL
201F  74 14                 5938  C          JZ      F19
2021  0E                    5939  C          PUSH    CS
2022  07                    5940  C          POP     ES
2023  FE C8                 5941  C          DEC     AL
2025  75 08                 5942  C          JNZ     F13
2027  B9 000E               5943  C          MOV     CX,14D
202A  BD 0000 E             5944  C          MOV     BP,OFFSET CGMN                  ; ROM 8 X 14 CHARACTER SET
202D  EB 06                 5945  C          JMP     SHORT F19
202F                        5946  C  F13:
202F  B9 0008               5947  C          MOV     CX,8
2032  BD 0000 E             5948  C          MOV     BP,OFFSET CGDDOT                ; ROM 8 X 8 DOUBLE DOT
2035                        5949  C  F19:
2035  FA                    5950  C          CLI
2036  89 2E 010C R          5951  C          MOV     WORD PTR GRX_SET , BP
203A  8C 06 010E R          5952  C          MOV     WORD PTR GRX_SET + 2 , ES
203E  FB                    5953  C          STI
                            5954  C          ASSUME  DS:ABS0
203F  E8 0D01 R             5955  C          CALL    DDS
2042  89 0E 0485 R          5956  C          MOV     POINTS,CX
2046  8A C3                 5957  C          MOV     AL,BL
2048  BB 2064 R             5958  C          MOV     BX,OFFSET RT
204B  0A C0                 5959  C          OR      AL,AL
204D  75 05                 5960  C          JNZ     DR_3
204F  8A C2                 5961  C          MOV     AL,DL
2051  EB 09 90              5962  C          JMP     DR_1
2054                        5963  C  DR_3:
2054  3C 03                 5964  C          CMP     AL,3
2056  76 02                 5965  C          JBE     DR_2
2058  B0 02                 5966  C          MOV     AL,2
205A                        5967  C  DR_2:
205A  2E: D7                5968  C          XLAT    CS:RT
205C                        5969  C  DR_1:
205C  FE C8                 5970  C          DEC     AL
205E  A2 0484 R             5971  C          MOV     ROWS,AL
2061  E9 219B R             5972  C          JMP     V_RET
                            5973  C
2064                        5974  C  RT      LABEL   BYTE
2064  00 0E 19 2B           5975  C          DB      00D,14D,25D,43D
                            5976  C
                            5977  C
                            5978  C  ;----- INFORMATION RETURN DONE HERE
                            5979  C
2068                        5980  C  AH11_INFORM:
                            5981  C          ASSUME  DS:ABS0
2068  3C 30                 5982  C          CMP     AL,030H
206A  74 03                 5983  C          JE      F6
206C                        5984  C  F5:
206C  E9 219B R             5985  C          JMP     V_RET
206F                        5986  C  F6:
206F  8B 0E 0485 R          5987  C          MOV     CX,POINTS
2073  8A 16 0484 R          5988  C          MOV     DL,ROWS
2077  80 FF 07              5989  C          CMP     BH,7
207A  77 F0                 5990  C          JA      F5
207C  80 FF 01              5991  C          CMP     BH,1
207F  77 18                 5992  C          JA      F7
                            5993  C
                            5994  C          ASSUME  DS:ABS0
2081  52                    5995  C          PUSH    DX
                            5996  C          SRLOAD  DS,0
2082  2B D2                 5997  C+         SUB     DX,DX
2084  8E DA                 5998  C+         MOV     DS,DX
2086  5A                    5999  C          POP     DX
2087  0A FF                 6000  C          OR      BH,BH
2089  75 07                 6001  C          JNZ     F9
                            6002  C          WLXS    ES,BP,EXT_PTR
208B  C4 2E 007C R          6003  C+         LES     BP,EXT_PTR
208F  EB 1A 90              6004  C          JMP     INFORM_OUT
2092                        6005  C  F9:
                            6006  C          WLXS    ES,BP,GRX_SET
2092  C4 2E 010C R          6007  C+         LES     BP,GRX_SET
2096  EB 13 90              6008  C          JMP     INFORM_OUT
                            6009  C
                            6010  C  ;----- HANDLE BH = 2  THRU  BH = 5 HERE  RETURN ROM TABLE POINTERS
                            6011  C
2099                        6012  C  F7:
                            6013  C          ASSUME  DS:ABS0
2099  80 EF 02              6014  C          SUB     BH,2
209C  8A DF                 6015  C          MOV     BL,BH
209E  2A FF                 6016  C          SUB     BH,BH
20A0  D1 E3                 6017  C          SAL     BX,1
20A2  81 C3 20B4 R          6018  C          ADD     BX,OFFSET TBL_5
20A6  2E: 8B 2F             6019  C          MOV     BP,CS:[BX]
20A9  0E                    6020  C          PUSH    CS
20AA  07                    6021  C          POP     ES
                            6022  C
20AB                        6023  C  INFORM_OUT:
20AB  5F                    6024  C          POP     DI
20AC  5E                    6025  C          POP     SI
20AD  5B                    6026  C          POP     BX
20AE  58                    6027  C          POP     AX                              ; DISCARD SAVED CX
20AF  58                    6028  C          POP     AX                              ; DISCARD SAVED DX
20B0  1F                    6029  C          POP     DS
20B1  58                    6030  C          POP     AX                              ; DISCARD SAVED ES
20B2  58                    6031  C          POP     AX                              ; DISCARD SAVED BP
20B3  CF                    6032  C          IRET
                            6033  C
                            6034  C  ;----- TABLE OF CHARACTER GENERATOR OFFSETS
                            6035  C
20B4                        6036  C  TBL_5   LABEL   WORD
20B4  0000 E                6037  C          DW      OFFSET  CGMN
20B6  0000 E                6038  C          DW      OFFSET  CGDDOT
20B8  0000 E                6039  C          DW      OFFSET  INT_1F_1
20BA  0000 E                6040  C          DW      OFFSET  CGMN_FDG
                            6041  C
                            6042  C          SUBTTL
                            6043
                            6044       ;----- ALTERNATE SELECT
                            6045
20BC                        6046       AH12:
                            6047            ASSUME  DS:ABS0
20BC  80 FB 10              6048            CMP     BL,010H                         ; RETURN ACTIVE CALL
```

```
20BF  72 51                    6049                    JB      ACT_1
20C1  74 1B                    6050                    JE      ACT_3
20C3  80 FB 20                 6051                    CMP     BL,020H                         ; ALTERNATE PRINT SCREEN
20C6  74 03                    6052                    JE      ACT_2
20C8  E9 219B R                6053                    JMP     V_RET                           ; INVALID CALL
20CB                           6054            ACT_2:                                          ; NEW PRINT SCREEN
                               6055                    SRLOAD  DS,0
20CB  2B D2                    6056            +       SUB     DX,DX
20CD  8E DA                    6057            +       MOV     DS,DX
20CF  FA                       6058                    CLI
20D0  C7 06 0014 R 21A4 R      6059                    MOV     WORD PTR INT5_PTR, OFFSET PRINT_SCREEN
20D6  8C 0E 0016 R             6060                    MOV     WORD PTR INT5_PTR+2, CS
20DA  FB                       6061                    STI
20DB  E9 219B R                6062                    JMP     V_RET
20DE                           6063            ACT_3:
20DE  8A 3E 0487 R             6064                    mov     bh,info                         ; looking for monoc bit
20E2  80 E7 02                 6065                    and     bh,2                            ; isolate
20E5  D0 EF                    6066                    shr     bh,1                            ; adjust
                               6067
20E7  A0 0487 R                6068                    mov     al,info                         ; looking for memory
20EA  24 60                    6069                    and     al,01100000b                    ; memory bits
20EC  B1 05                    6070                    mov     cl,5                            ; shift count
20EE  D2 E8                    6071                    shr     al,cl                           ; adjust mem value
20F0  8A D8                    6072                    mov     bl,al                           ; return register
                               6073
20F2  8A 0E 0488 R             6074                    MOV     CL,INFO_3                       ; FEATURE/SWITCH
20F6  8A E9                    6075                    MOV     CH,CL                           ; DUPLICATE IN CH
20F8  80 E1 0F                 6076                    AND     CL,0FH                          ; MASK OFF SWITCH VALUE
20FB  D0 ED                    6077                    SHR     CH,1                            ; MOVE FEATURE VALUE
20FD  D0 ED                    6078                    SHR     CH,1
20FF  D0 ED                    6079                    SHR     CH,1
2101  D0 ED                    6080                    SHR     CH,1
2103  80 E5 0F                 6081                    AND     CH,0FH                          ; MASK IT
                               6082
2106  5F                       6083                    POP     DI
2107  5E                       6084                    POP     SI
2108  5A                       6085                    POP     DX                              ; DISCARD BX
2109  5A                       6086                    POP     DX                              ; DISCARD CX
210A  5A                       6087                    POP     DX
210B  1F                       6088                    POP     DS
210C  07                       6089                    POP     ES
210D  5D                       6090                    POP     BP
210E  CF                       6091                    IRET
210F                           6092            AH12_X:
210F  E9 219B R                6093                    JMP     V_RET                           ; RETURN TO CALLER
2112                           6094            ACT_1:
2112                           6095            STR_OUTZ:
2112  E9 219B R                6096                    JMP     V_RET                           ; RETURN TO CALLER
                               6097
                               6098            ;----- WRITE STRING
                               6099
2115                           6100            AH13:
2115  3C 04                    6101                    CMP     AL,04                           ; RANGE CHECK
2117  73 F9                    6102                    JAE     STR_OUTZ                        ; INVALID PARAMETER
2119  E3 F7                    6103                    JCXZ    STR_OUTZ
211B  53                       6104                    PUSH    BX                              ; SAVE REGISTER
211C  8A DF                    6105                    MOV     BL,BH                           ; GET PAGE TO LOW BYTE
211E  2A FF                    6106                    SUB     BH,BH
2120  D1 E3                    6107                    SAL     BX,1                            ; *2 FOR WORD OFFSET
2122  8B B7 0450 R             6108                    MOV     SI,[BX + OFFSET CURSOR_POSN]    ; GET CURSOR POSITION
2126  5B                       6109                    POP     BX                              ; RESTORE
2127  56                       6110                    PUSH    SI                              ; CURRENT VALUE ON STACK
                               6111
2128  50                       6112                    PUSH    AX
2129  B8 0200                  6113                    MOV     AX,0200H                        ; SET THE CURSOR POSITION
212C  CD 10                    6114                    INT     10H
212E  58                       6115                    POP     AX
212F                           6116            STR_1:
212F  51                       6117                    PUSH    CX
2130  53                       6118                    PUSH    BX
2131  50                       6119                    PUSH    AX
2132  86 E0                    6120                    XCHG    AH,AL
2134  26: 8A 46 00             6121                    MOV     AL,ES:[BP]                      ; GET THE CHAR TO WRITE
2138  45                       6122                    INC     BP
2139  3C 0D                    6123                    CMP     AL,0DH                          ; CARRIAGE RETURN
213B  74 3D                    6124                    JE      STR_CR_LF
213D  3C 0A                    6125                    CMP     AL,0AH                          ; LINE FEED
213F  74 39                    6126                    JE      STR_CR_LF
2141  3C 08                    6127                    CMP     AL,08H                          ; BACKSPACE
2143  74 35                    6128                    JE      STR_CR_LF
2145  3C 07                    6129                    CMP     AL,07H                          ; BELL
2147  74 31                    6130                    JE      STR_CR_LF
2149  B9 0001                  6131                    MOV     CX,1                            ; COUNT OF CHARACTERS
214C  80 FC 02                 6132                    CMP     AH,2                            ; CHECK WHERE ATTR IS
214F  72 05                    6133                    JB      DO_STR                          ; NOT IN THE STRING
2151  26: 8A 5E 00             6134                    MOV     BL,ES:[BP]                      ; GET THE ATTRIBUTE
2155  45                       6135                    INC     BP                              ; NEXT ITEM IN STRING
2156                           6136            DO_STR:
2156  B4 09                    6137                    MOV     AH,09H                          ; WRITE THE CHAR/ATTR
2158  CD 10                    6138                    INT     10H
215A  FE C2                    6139                    INC     DL                              ; NEXT CURSOR POSITION
215C  3A 16 044A R             6140                    CMP     DL,BYTE PTR CRT_COLS            ; COLUMN OVERFLOW
2160  72 11                    6141                    JB      STR_2                           ; NOT YET
2162  3A 36 0484 R             6142                    CMP     DH,ROWS
2166  75 07                    6143                    JNE     STR_3
2168  B8 0E0A                  6144                    MOV     AX,0E0AH
216B  CD 10                    6145                    INT     10H
216D  FE CE                    6146                    DEC     DH
216F                           6147            STR_3:
216F  FE C6                    6148                    INC     DH                              ; NEXT ROW
2171  2A D2                    6149                    SUB     DL,DL                           ; COLUMN ZERO
2173                           6150            STR_2:
2173  B8 0200                  6151                    MOV     AX,0200H                        ; SET THE CURSOR
2176  CD 10                    6152                    INT     10H
2178  EB 0E                    6153                    JMP     SHORT   STR_4
217A                           6154            STR_CR_LF:
217A  B4 0E                    6155                    MOV     AH,0EH
217C  CD 10                    6156                    INT     10H
217E  8A DF                    6157                    MOV     BL,BH                           ; GET PAGE TO LOW BYTE
2180  2A FF                    6158                    SUB     BH,BH
2182  D1 E3                    6159                    SAL     BX,1                            ; *2 FOR WORD OFFSET
2184  8B 97 0450 R             6160                    MOV     DX,[BX + OFFSET CURSOR_POSN]    ; GET CURSOR POSITION
2188                           6161            STR_4:
2188  58                       6162                    POP     AX
2189  5B                       6163                    POP     BX
218A  59                       6164                    POP     CX
218B  E2 A2                    6165                    LOOP    STR_1
                               6166
218D  5A                       6167                    POP     DX                              ; RECOVER CURSOR POSITION
                               6168                                                            ;  FROM PUSH SI ABOVE
218E  3C 01                    6169                    CMP     AL,1
2190  74 09                    6170                    JE      STR_OUT
2192  3C 03                    6171                    CMP     AL,3
2194  74 05                    6172                    JE      STR_OUT
2196  B8 0200                  6173                    MOV     AX,0200H                        ; SET CURSOR POSITION
2199  CD 10                    6174                    INT     10H
```

```
219B                    6175        STR_OUT:
                        6176                                        ; ALLOW FALL THROUGH
                        6177
                        6178
219B                    6179        V_RET   PROC    NEAR            ; VIDEO BIOS RETURN
219B  5F                6180                POP     DI
219C  5E                6181                POP     SI
219D  5B                6182                POP     BX
219E  59                6183                POP     CX
219F  5A                6184                POP     DX
21A0  1F                6185                POP     DS
21A1  07                6186                POP     ES
21A2  5D                6187                POP     BP
21A3  CF                6188                IRET
21A4                    6189        V_RET   ENDP
                        6190
21A4                    6191        COMBO_VIDEO     ENDP
                        6192
                        6193    C           INCLUDE         VPRSC.INC
                        6194    C           SUBTTL  VPRSC.INC
                        6195    C           PAGE
                        6196    C   ;-------------------------------------------------------------------------------
                        6197    C   ;   INTERRUPT 5                                                              :
                        6198    C   ;       THIS LOGIC WILL BE INVOKED BY INTERRUPT 05H TO PRINT THE             :
                        6199    C   ;       SCREEN. THE CURSOR POSITION AT THE TIME THIS ROUTINE IS INVOKED      :
                        6200    C   ;       WILL BE SAVED AND RESTORED UPON COMPLETION. THE ROUTINE IS           :
                        6201    C   ;       INTENDED TO RUN WITH INTERRUPTS ENABLED. IF A SUBSEQUENT             :
                        6202    C   ;       'PRINT SCREEN' KEY IS DEPRESSED DURING THE TIME THIS ROUTINE         :
                        6203    C   ;       IS PRINTING IT WILL BE IGNORED.                                      :
                        6204    C   ;       ADDRESS 50:0 CONTAINS THE STATUS OF THE PRINT SCREEN:                :
                        6205    C   ;                                                                            :
                        6206    C   ;       50:0    =0      EITHER PRINT SCREEN HAS NOT BEEN CALLED              :
                        6207    C   ;                       OR UPON RETURN FROM A CALL THIS INDICATES        :
                        6208    C   ;                       A SUCCESSFUL OPERATION.                          :
                        6209    C   ;               =1      PRINT SCREEN IS IN PROGRESS                      :
                        6210    C   ;               =255    ERROR ENCOUNTERED DURING PRINTING                :
                        6211    C   ;-------------------------------------------------------------------------------
                        6212    C           ASSUME  CS:CODE,DS:ABS0
21A4                    6213    C   PRINT_SCREEN    PROC    FAR
21A4  FB                6214    C           STI                         ; MUST RUN WITH INTS ENABLED
21A5  1E                6215    C           PUSH    DS                  ; MUST USE 50:0 FOR DATA
21A6  50                6216    C           PUSH    AX                  ;  area storage
21A7  53                6217    C           PUSH    BX
21A8  51                6218    C           PUSH    CX                  ; USE THIS LATER FOR CURSOR LIMITS
21A9  52                6219    C           PUSH    DX                  ; WILL HOLD CURRENT CURSOR POS
21AA  E8 0D01 R         6220    C           CALL    DDS
21AD  80 3E 0500 R 01   6221    C           CMP     STATUS_BYTE,1       ; SEE IF PRINT ALREADY IN PROGRESS
21B2  74 63             6222    C           JZ      EXIT                ; JUMP IF PRINT IN PROGRESS
21B4  C6 06 0500 R 01   6223    C           MOV     STATUS_BYTE,1       ; INDICATE PRINT NOW IN PROGRESS
21B9  B4 0F             6224    C           MOV     AH,15               ; WILL REQUEST THE CURRENT MODE
21BB  CD 10             6225    C           INT     10H                 ;       [AL]=MODE (NOT USED)
                        6226    C                                       ;       [AH]=NUMBER COLUMNS/LINE
                        6227    C                                       ;       [BH]=VISUAL PAGE
                        6228    C   ;-------------------------------------------------------------------------
                        6229    C   ;       AT THIS POINT WE KNOW THE COLUMNS/LINE ARE IN                    :
                        6230    C   ;       [AX] AND THE PAGE IF APPLICABLE IS IN [BH]. THE STACK            :
                        6231    C   ;       HAS DS,AX,BX,CX,DX PUSHED. [AL] HAS VIDEO MODE                   :
                        6232    C   ;-------------------------------------------------------------------------
21BD  8A CC             6233    C           MOV     CL,AH               ; WILL MAKE USE OF [CX] REG TO
21BF  8A 2E 0484 R      6234    C           MOV     CH,ROWS             ; CONTROL ROW & COLUMNS
21C3  FE C5             6235    C           INC     CH                  ; ADJUST
21C5  E8 221D R         6236    C           CALL    CRLF                ; CAR RETURN LINE FEED ROUTINE
21C8  51                6237    C           PUSH    CX                  ; SAVE SCREEN BOUNDS
21C9  B4 03             6238    C           MOV     AH,3                ; WILL NOW READ THE CURSOR.
21CB  CD 10             6239    C           INT     10H                 ; AND PRESERVE THE POSITION
21CD  59                6240    C           POP     CX                  ; RECALL SCREEN BOUNDS
21CE  52                6241    C           PUSH    DX                  ; RECALL [BH]=VISUAL PAGE
21CF  33 D2             6242    C           XOR     DX,DX               ; SET CURSOR POSITION TO [0,0]
                        6243    C   ;-------------------------------------------------------------------
                        6244    C   ;       THE LOOP FROM PRI10 TO THE INSTRUCTION PRIOR TO PRI20    :
                        6245    C   ;       IS THE LOOP TO READ EACH CURSOR POSITION FROM THE         :
                        6246    C   ;       SCREEN AND PRINT.                                        :
                        6247    C   ;-------------------------------------------------------------------
21D1                    6248    C   PRI10:
21D1  B4 02             6249    C           MOV     AH,2                ; TO INDICATE CURSOR SET REQUEST
21D3  CD 10             6250    C           INT     10H                 ; NEW CURSOR POS ESTABLISHED
21D5  B4 08             6251    C           MOV     AH,8                ; TO INDICATE READ CHARACTER
21D7  CD 10             6252    C           INT     10H                 ; CHARACTER NOW IN [AL]
21D9  0A C0             6253    C           OR      AL,AL               ; SEE IF VALID CHAR
21DB  75 02             6254    C           JNZ     PRI15               ; JUMP IF VALID CHAR
21DD  B0 20             6255    C           MOV     AL,' '              ; MAKE A BLANK
21DF                    6256    C   PRI15:
21DF  52                6257    C           PUSH    DX                  ; SAVE CURSOR POSITION
21E0  33 D2             6258    C           XOR     DX,DX               ; INDICATE PRINTER 1
21E2  32 E4             6259    C           XOR     AH,AH               ; TO INDICATE PRINT CHAR IN [AL]
21E4  CD 17             6260    C           INT     17H                 ; PRINT THE CHARACTER
21E6  5A                6261    C           POP     DX                  ; RECALL CURSOR POSITION
21E7  F6 C4 29          6262    C           TEST    AH,029H             ; TEST FOR PRINTER ERROR
21EA  75 21             6263    C           JNZ     ERR10               ; JUMP IF ERROR DETECTED
21EC  FE C2             6264    C           INC     DL                  ; ADVANCE TO NEXT COLUMN
21EE  3A CA             6265    C           CMP     CL,DL               ; SEE IF AT END OF LINE
21F0  75 DF             6266    C           JNZ     PRI10               ; IF NOT PROCEED
21F2  32 D2             6267    C           XOR     DL,DL               ; BACK TO COLUMN 0
21F4  8A E2             6268    C           MOV     AH,DL               ; [AH]=0
21F6  52                6269    C           PUSH    DX                  ; SAVE NEW CURSOR POSITION
21F7  E8 221D R         6270    C           CALL    CRLF                ; LINE FEED CARRIAGE RETURN
21FA  5A                6271    C           POP     DX                  ; RECALL CURSOR POSITION
21FB  FE C6             6272    C           INC     DH                  ; ADVANCE TO NEXT LINE
21FD  3A EE             6273    C           CMP     CH,DH               ; FINISHED?
21FF  75 D0             6274    C           JNZ     PRI10               ; IF NOT CONTINUE
                        6275    C
2201  5A                6276    C           POP     DX                  ; RECALL CURSOR POSITION
2202  B4 02             6277    C           MOV     AH,2                ; TO INDICATE CURSOR SET REQUEST
2204  CD 10             6278    C           INT     10H                 ; CURSOR POSITION RESTORED
2206  C6 06 0500 R 00   6279    C           MOV     STATUS_BYTE,0       ; INDICATE FINISHED
220B  EB 0A             6280    C           JMP     SHORT EXIT          ; EXIT THE ROUTINE
220D                    6281    C   ERR10:
220D  5A                6282    C           POP     DX                  ; GET CURSOR POSITION
220E  B4 02             6283    C           MOV     AH,2                ; TO REQUEST CURSOR SET
2210  CD 10             6284    C           INT     10H                 ; CURSOR POSITION RESTORED
2212  C6 06 0500 R FF   6285    C           MOV     STATUS_BYTE,0FFH    ; INDICATE ERROR
2217                    6286    C   EXIT:
2217  5A                6287    C           POP     DX                  ; RESTORE ALL THE REGISTERS USED
2218  59                6288    C           POP     CX
2219  5B                6289    C           POP     BX
221A  58                6290    C           POP     AX
221B  1F                6291    C           POP     DS
221C  CF                6292    C           IRET
221D                    6293    C   PRINT_SCREEN    ENDP
                        6294    C
                        6295    C   ;------ CARRIAGE RETURN, LINE FEED SUBROUTINE
                        6296    C
221D                    6297    C   CRLF    PROC    NEAR
221D  33 D2             6298    C           XOR     DX,DX               ; PRINTER 0
221F  32 E4             6299    C           XOR     AH,AH               ; WILL NOW SEND INITIAL CR, LF
                        6300    C                                       ;  TO PRINTER
```

```
2221  B0 0D                   6301  C          MOV     AL,0DH                  ; CR
2223  CD 17                   6302  C          INT     17H                     ; SEND THE LINE FEED
2225  32 E4                   6303  C          XOR     AH,AH                   ; NOW FOR THE CR
2227  B0 0A                   6304  C          MOV     AL,0AH                  ; LF
2229  CD 17                   6305  C          INT     17H                     ; SEND THE CARRIAGE RETURN
222B  C3                      6306  C          RET
222C                          6307  C  CRLF    ENDP
                              6308  C
                              6309  C          SUBTTL
                              6310
222C                          6311       CODE    ENDS
                              6312               END
                                1                PAGE,120
                                2                SUBTTL  MONOCHROME CHARACTER GENERATOR
0000                            3       CODE    SEGMENT PUBLIC
                                4               PUBLIC  CGMN
0000                            5       CGMN    LABEL   BYTE
                                6                                                       ; BW 8*14 PATTTERN
0000  00 00 00 00 00 00         7                DB      000H,000H,000H,000H,000H,000H,000H,000H ; TOP_HALF 00
      00 00
0008  00 00 00 00 00 00 00      9                DB      000H,000H,000H,000H,000H,000H           ; BOTTOM_HALF 00
000E  00 00 7E 81 A5 81         10               DB      000H,000H,07EH,081H,0A5H,081H,081H,0BDH ; TH_01
      81 BD
0016  99 81 7E 00 00 00         12               DB      099H,081H,07EH,000H,000H,000H           ; BT_01
001C  00 00 7E FF DB FF         13               DB      000H,000H,07EH,0FFH,0DBH,0FFH,0FFH,0C3H ; TH_02
      FF C3
0024  E7 FF 7E 00 00 00         15               DB      0E7H,0FFH,07EH,000H,000H,000H           ; BT_02
002A  00 00 00 6C FE FE         16               DB      000H,000H,000H,06CH,0FEH,0FEH,0FEH,0FEH ; TH_03
      FE FE
                                17
0032  7C 38 10 00 00 00         18               DB      07CH,038H,010H,000H,000H,000H           ; BT_03
0038  00 00 00 10 38 7C         19               DB      000H,000H,000H,010H,038H,07CH,0FEH,07CH ; TH_04
      FE 7C
0040  38 10 00 00 00 00         21               DB      038H,010H,000H,000H,000H,000H           ; BT_04
0046  00 00 18 3C 3C E7         22               DB      000H,000H,018H,03CH,03CH,0E7H,0E7H,0E7H ; TH_05
      E7 E7
                                23
004E  18 18 3C 00 00 00         24               DB      018H,018H,03CH,000H,000H,000H           ; BT_05
0054  00 00 18 3C 7E FF         25               DB      000H,000H,018H,03CH,07EH,0FFH,0FFH,07EH ; TH_06
      FF 7E
                                26
005C  18 18 3C 00 00 00         27               DB      018H,018H,03CH,000H,000H,000H           ; BT_06
0062  00 00 00 00 00 18         28               DB      000H,000H,000H,000H,000H,018H,03CH,03CH ; TH_07
      3C 3C
                                29
006A  18 00 00 00 00 00         30               DB      018H,000H,000H,000H,000H,000H           ; BT_07
0070  FF FF FF FF FF E7         31               DB      0FFH,0FFH,0FFH,0FFH,0FFH,0E7H,0C3H,0C3H ; TH_08
      C3 C3
                                32
0078  E7 FF FF FF FF FF         33               DB      0E7H,0FFH,0FFH,0FFH,0FFH,0FFH           ; BT_08
007E  00 00 00 00 3C 66         34               DB      000H,000H,000H,000H,03CH,066H,042H,042H ; TH_09
      42 42
                                35
0086  66 3C 00 00 00 00         36               DB      066H,03CH,000H,000H,000H,000H           ; BT_09
008C  FF FF FF FF C3 99         37               DB      0FFH,0FFH,0FFH,0FFH,0C3H,099H,0BDH,0BDH ; TH_0A
      BD BD
                                38
0094  99 C3 FF FF FF FF         39               DB      099H,0C3H,0FFH,0FFH,0FFH,0FFH           ; BT_0A
009A  00 00 1E 0E 1A 32         40               DB      000H,000H,01EH,00EH,01AH,032H,078H,0CCH ; TH_0B
      78 CC
00A2  CC CC 78 00 00 00         42               DB      0CCH,0CCH,078H,000H,000H,000H           ; BT_0B
00A8  00 00 3C 66 66 66         43               DB      000H,000H,03CH,066H,066H,066H,03CH,018H ; TH_0C
      3C 18
                                44
00B0  7E 18 18 00 00 00         45               DB      07EH,018H,018H,000H,000H,000H           ; BT_0C
00B6  00 00 3F 33 3F 30         46               DB      000H,000H,03FH,033H,03FH,030H,030H,030H ; TH_0D
      30 30
                                47
00BE  70 F0 E0 00 00 00         48               DB      070H,0F0H,0E0H,000H,000H,000H           ; BT_0D
00C4  00 00 7F 63 7F 63         49               DB      000H,000H,07FH,063H,07FH,063H,063H,063H ; TH_0E
      63 63
                                50
00CC  67 E7 E6 C0 00 00         51               DB      067H,0E7H,0E6H,0C0H,000H,000H           ; BT_0E
00D2  00 00 18 18 DB 3C         52               DB      000H,000H,018H,018H,0DBH,03CH,0E7H,03CH ; TH_0F
      E7 3C
                                53
00DA  DB 18 18 00 00 00         54               DB      0DBH,018H,018H,000H,000H,000H           ; BT_0F
                                55
00E0  00 00 80 C0 E0 F8         56               DB      000H,000H,080H,0C0H,0E0H,0F8H,0FEH,0F8H ; TH_10
      FE F8
                                57
00E8  E0 C0 80 00 00 00         58               DB      0E0H,0C0H,080H,000H,000H,000H           ; BT_10
00EE  00 00 02 06 0E 3E         59               DB      000H,000H,002H,006H,00EH,03EH,0FEH,03EH ; TH_11
      FE 3E
                                60
00F6  0E 06 02 00 00 00         61               DB      00EH,006H,002H,000H,000H,000H           ; BT_11
00FC  00 00 18 3C 7E 18         62               DB      000H,000H,018H,03CH,07EH,018H,018H,018H ; TH_12
      18 18
                                63
0104  7E 3C 18 00 00 00         64               DB      07EH,03CH,018H,000H,000H,000H           ; BT_12
010A  00 00 66 66 66 66         65               DB      000H,000H,066H,066H,066H,066H,066H,066H ; TH-13
      66 66
                                66
0112  00 66 66 00 00 00         67               DB      000H,066H,066H,000H,000H,000H           ; BT_13
0118  00 00 7F DB DB DB         68               DB      000H,000H,07FH,0DBH,0DBH,0DBH,07BH,01BH ; TH_14
      7B 1B
                                69
0120  1B 1B 1B 00 00 00         70               DB      01BH,01BH,01BH,000H,000H,000H           ; BT_14
0126  00 7C C6 60 38 6C         71               DB      000H,07CH,0C6H,060H,038H,06CH,0C6H,0C6H ; TH_15
      C6 C6
                                72
012E  6C 38 0C C6 7C 00         73               DB      06CH,038H,00CH,0C6H,07CH,000H           ; BT_15
0134  00 00 00 00 00 00         74               DB      000H,000H,000H,000H,000H,000H,000H,000H ; TH_16
      00 00
                                75
013C  FE FE FE 00 00 00         76               DB      0FEH,0FEH,0FEH,000H,000H,000H           ; BT_16
0142  00 00 18 3C 7E 18         77               DB      000H,000H,018H,03CH,07EH,018H,018H,018H ; TH_17
      18 18
                                78
014A  7E 3C 18 7E 00 00         79               DB      07EH,03CH,018H,07EH,000H,000H           ; BT_17
0150  00 00 18 3C 7E 18         80               DB      000H,000H,018H,03CH,07EH,018H,018H,018H ; TH_18
      18 18
                                81
0158  18 18 18 00 00 00         82               DB      018H,018H,018H,000H,000H,000H           ; BT_18
015E  00 00 18 18 18 18         83               DB      000H,000H,018H,018H,018H,018H,018H,018H ; TH_19
      18 18
                                84
0166  7E 3C 18 00 00 00         85               DB      07EH,03CH,018H,000H,000H,000H           ; BT_19
016C  00 00 00 00 18 0C         86               DB      000H,000H,000H,000H,018H,00CH,0FEH,00CH ; TH_1A
      FE 0C
                                87
0174  18 00 00 00 00 00         88               DB      018H,000H,000H,000H,000H,000H           ; BT_1A
017A  00 00 00 00 30 60         89               DB      000H,000H,000H,000H,030H,060H,0FEH,060H ; TH_1B
      FE 60
                                90
0182  30 00 00 00 00 00         91               DB      030H,000H,000H,000H,000H,000H           ; BT_1B
0188  00 00 00 00 00 C0         92               DB      000H,000H,000H,000H,000H,0C0H,0C0H,0C0H ; TH_1C
      C0 C0
                                93
0190  FE 00 00 00 00 00         94               DB      0FEH,000H,000H,000H,000H,000H           ; BT_1C
0196  00 00 00 00 28 6C         95               DB      000H,000H,000H,000H,028H,06CH,0FEH,06CH ; TH_1D
      FE 6C
                                96
019E  28 00 00 00 00 00         97               DB      028H,000H,000H,000H,000H,000H           ; BT_1D
01A4  00 00 00 10 38 38         98               DB      000H,000H,000H,010H,038H,038H,07CH,07CH ; TH_1E
      7C 7C
                                99
01AC  FE FE 00 00 00 00        100               DB      0FEH,0FEH,000H,000H,000H,000H           ; BT_1E
01B2  00 00 00 FE FE 7C        101               DB      000H,000H,000H,0FEH,0FEH,07CH,07CH,038H ; TH_1F
      7C 38
                               102
01BA  38 10 00 00 00 00        103               DB      038H,010H,000H,000H,000H,000H           ; BT_1F
                               104
01C0  00 00 00 00 00 00        105               DB      000H,000H,000H,0D0H,000H,000H,000H,000H ; TH_20 SP
      00 00
                               106
01C8  00 00 00 00 00 00        107               DB      000H,000H,000H,000H,000H,000H           ; BT_20 SP
01CE  00 00 18 3C 3C 3C        108               DB      000H,000H,018H,03CH,03CH,03CH,018H,018H ; TH_21 !
      18 18
                               109
01D6  00 18 18 00 00 00        110               DB      000H,018H,018H,000H,000H,000H           ; BT_21 !
01DC  00 66 66 66 24 00        111               DB      000H,066H,066H,066H,024H,000H,000H,000H ; TH_22 "
      00 00
                               112
01E4  00 00 00 00 00 00        113               DB      000H,000H,000H,000H,000H,000H           ; BT_22 "
01EA  00 00 6C 6C FE 6C        114               DB      000H,000H,06CH,06CH,0FEH,06CH,06CH,06CH ; TH_23 #
```

```
                        6C 6C                                   115
01F2 FE 6C 6C 00 00 00                                          116             DB      0FEH,06CH,06CH,000H,000H,000H           ; BT_23 #
01F8 18 18 7C C6 C2 C0                                          117             DB      018H,018H,07CH,0C6H,0C2H,0C0H,07CH,006H ; TH_24 $
                        7C 06                                   118
0200 86 C6 7C 18 18 00                                          119             DB      086H,0C6H,07CH,018H,018H,000H           ; BT_24 $
0206 00 00 00 00 C2 C6                                          120             DB      000H,000H,000H,000H,0C2H,0C6H,00CH,018H ; TH_25 '%'
                        0C 18                                   121
020E 30 66 C6 00 00 00                                          122             DB      030H,066H,0C6H,000H,000H,000H           ; BT_25 '%'
0214 00 00 38 6C 6C 38                                          123             DB      000H,000H,038H,06CH,06CH,038H,076H,0DCH ; TH_26 &
                        76 DC                                   124
021C CC CC 76 00 00 00                                          125             DB      0CCH,0CCH,076H,000H,000H,000H           ; BT_26 &
0222 00 30 30 30 60 00                                          126             DB      000H,030H,030H,030H,060H,000H,000H,000H ; TH_27 '
                        00 00                                   127
022A 00 00 00 00 00 00                                          128             DB      000H,000H,000H,000H,000H,000H           ; BT_27 '
0230 00 00 0C 18 30 30                                          129             DB      000H,000H,00CH,018H,030H,030H,030H,030H ; TH_28 (
                        30 30                                   130
0238 30 18 0C 00 00 00                                          131             DB      030H,018H,00CH,000H,000H,000H           ; BT_28 (
023E 00 00 30 18 0C 0C                                          132             DB      000H,000H,030H,018H,00CH,00CH,00CH,00CH ; TH_29 )
                        0C 0C                                   133
0246 0C 18 30 00 00 00                                          134             DB      00CH,018H,030H,000H,000H,000H           ; BT_29 )
024C 00 00 00 00 66 3C                                          135             DB      000H,000H,000H,000H,066H,03CH,0FFH,03CH ; TH_2A *
                        FF 3C                                   136
0254 66 00 00 00 00 00                                          137             DB      066H,000H,000H,000H,000H,000H           ; BT_2A *
025A 00 00 00 00 18 18                                          138             DB      000H,000H,000H,000H,018H,018H,07EH,018H ; TH_2B +
                        7E 18                                   139
0262 18 00 00 00 00 00                                          140             DB      018H,000H,000H,000H,000H,000H           ; BT_2B +
0268 00 00 00 00 00 00                                          141             DB      000H,000H,000H,000H,000H,000H,000H,000H ; TH_2C ,
                        00 00                                   142
0270 18 18 18 30 00 00                                          143             DB      018H,018H,018H,030H,000H,000H           ; BT_2C ,
0276 00 00 00 00 00 00                                          144             DB      000H,000H,000H,000H,000H,000H,0FEH,000H ; TH_2D -
                        FE 00                                   145
027E 00 00 00 00 00 00                                          146             DB      000H,000H,000H,000H,000H,000H           ; BT_2D -
0284 00 00 00 00 00 00                                          147             DB      000H,000H,000H,000H,000H,000H,000H,000H ; TH_2E .
                        00 00                                   148
028C 00 18 18 00 00 00                                          149             DB      000H,018H,018H,000H,000H,000H           ; BT_2E .
0292 00 00 02 06 0C 18                                          150             DB      000H,000H,002H,006H,00CH,018H,030H,060H ; TH_2F /
                        30 60                                   151
029A C0 80 00 00 00 00                                          152             DB      0C0H,080H,000H,000H,000H,000H           ; BT_2F /
                                                                153
02A0 00 00 7C C6 CE DE                                          154             DB      000H,000H,07CH,0C6H,0CEH,0DEH,0F6H,0E6H ; TH_30 0
                        F6 E6                                   155
02A8 C6 C6 7C 00 00 00                                          156             DB      0C6H,0C6H,07CH,000H,000H,000H           ; BT_30 0
02AE 00 00 18 38 78 18                                          157             DB      000H,000H,018H,038H,078H,018H,018H,018H ; TH_31 1
                        18 18                                   158
02B6 18 18 7E 00 00 00                                          159             DB      018H,018H,07EH,000H,000H,000H           ; BT_31 1
02BC 00 00 7C C6 06 0C                                          160             DB      000H,000H,07CH,0C6H,006H,00CH,018H,030H ; TH_32 2
                        18 30                                   161
02C4 60 C6 FE 00 00 00                                          162             DB      060H,0C6H,0FEH,000H,000H,000H           ; BT_32 2
02CA 00 00 7C C6 06 06                                          163             DB      000H,000H,07CH,0C6H,006H,006H,03CH,006H ; TH_33 3
                        3C 06                                   164
02D2 06 C6 7C 00 00 00                                          165             DB      006H,0C6H,07CH,000H,000H,000H           ; BT_33 3
02D8 00 00 0C 1C 3C 6C                                          166             DB      000H,000H,00CH,01CH,03CH,06CH,0CCH,0FEH ; TH_34 4
                        CC FE                                   167
02E0 0C 0C 1E 00 00 00                                          168             DB      00CH,00CH,01EH,000H,000H,000H           ; BT_34 4
02E6 00 00 FE C0 C0 C0                                          169             DB      000H,000H,0FEH,0C0H,0C0H,0C0H,0FCH,006H ; TH_35 5
                        FC 06                                   170
02EE 06 C6 7C 00 00 00                                          171             DB      006H,0C6H,07CH,000H,000H,000H           ; BT_35 5
02F4 00 00 38 60 C0 C0                                          172             DB      000H,000H,038H,060H,0C0H,0C0H,0FCH,0C6H ; TH_36 6
                        FC C6                                   173
02FC C6 C6 7C 00 00 00                                          174             DB      0C6H,0C6H,07CH,000H,000H,000H           ; BT_36 6
0302 00 00 FE C6 06 0C                                          175             DB      000H,000H,0FEH,0C6H,006H,00CH,018H,030H ; TH_37 7
                        18 30                                   176
030A 30 30 30 00 00 00                                          177             DB      030H,030H,030H,000H,000H,000H           ; BT_37 7
0310 00 00 7C C6 C6 C6                                          178             DB      000H,000H,07CH,0C6H,0C6H,0C6H,07CH,0C6H ; TH_38 8
                        7C C6                                   179
0318 C6 C6 7C 00 00 00                                          180             DB      0C6H,0C6H,07CH,000H,000H,000H           ; BT_38 8
031E 00 00 7C C6 C6 C6                                          181             DB      000H,000H,07CH,0C6H,0C6H,0C6H,07EH,006H ; TH_39 9
                        7E 06                                   182
0326 06 0C 78 00 00 00                                          183             DB      006H,00CH,078H,000H,000H,000H           ; BT_39 9
032C 00 00 00 18 18 00                                          184             DB      000H,000H,000H,018H,018H,000H,000H,000H ; TH_3A :
                        00 00                                   185
0334 18 18 00 00 00 00                                          186             DB      018H,018H,000H,000H,000H,000H           ; BT_3A :
033A 00 00 00 18 18 00                                          187             DB      000H,000H,000H,018H,018H,000H,000H,000H ; TH_3B ;
                        00 00                                   188
0342 18 18 30 00 00 00                                          189             DB      018H,018H,030H,000H,000H,000H           ; BT_3B ;
0348 00 00 06 0C 18 30                                          190             DB      000H,000H,006H,00CH,018H,030H,060H,030H ; TH_3C <
                        60 30                                   191
0350 18 0C 06 00 00 00                                          192             DB      018H,00CH,006H,000H,000H,000H           ; BT_3C <
0356 00 00 00 00 00 7E                                          193             DB      000H,000H,000H,000H,000H,07EH,000H,000H ; TH_3D =
                        00 00                                   194
035E 7E 00 00 00 00 00                                          195             DB      07EH,000H,000H,000H,000H,000H           ; BT_3D =
0364 00 00 60 30 18 0C                                          196             DB      000H,000H,060H,030H,018H,00CH,006H,00CH ; TH_3E >
                        06 0C                                   197
036C 18 30 60 00 00 00                                          198             DB      018H,030H,060H,000H,000H,000H           ; BT_3E >
0372 00 00 7C C6 C6 0C                                          199             DB      000H,000H,07CH,0C6H,0C6H,00CH,018H,018H ; TH_3F ?
                        18 18                                   200
037A 00 18 18 00 00 00                                          201             DB      000H,018H,018H,000H,000H,000H           ; BT_3F ?
                                                                202
0380 00 00 7C C6 C6 DE                                          203             DB      000H,000H,07CH,0C6H,0C6H,0DEH,0DEH,0DEH ; TH_40 @
                        DE DE                                   204
0388 DC C0 7C 00 00 00                                          205             DB      0DCH,0C0H,07CH,000H,000H,000H           ; BT_40 @
038E 00 00 10 38 6C C6                                          206             DB      000H,000H,010H,038H,06CH,0C6H,0C6H,0FEH ; TH_41 A
                        C6 FE                                   207
0396 C6 C6 C6 00 00 00                                          208             DB      0C6H,0C6H,0C6H,000H,000H,000H           ; BT_41 A
039C 00 00 FC 66 66 66                                          209             DB      000H,000H,0FCH,066H,066H,066H,07CH,066H ; TH_42 B
                        7C 66                                   210
03A4 66 66 FC 00 00 00                                          211             DB      066H,066H,0FCH,000H,000H,000H           ; BT_42 B
03AA 00 00 3C 66 C2 C0                                          212             DB      000H,000H,03CH,066H,0C2H,0C0H,0C0H,0C0H ; TH_43 C
                        C0 C0                                   213
03B2 C2 66 3C 00 00 00                                          214             DB      0C2H,066H,03CH,000H,000H,000H           ; BT_43 C
03B8 00 00 F8 6C 66 66                                          215             DB      000H,000H,0F8H,06CH,066H,066H,066H,066H ; TH_44 D
                        66 66                                   216
03C0 66 6C F8 00 00 00                                          217             DB      066H,06CH,0F8H,000H,000H,000H           ; BT_44 D
03C6 00 00 FE 66 62 68                                          218             DB      000H,000H,0FEH,066H,062H,068H,078H,068H ; TH_45 E
                        78 68                                   219
03CE 62 66 FE 00 00 00                                          220             DB      062H,066H,0FEH,000H,000H,000H           ; BT_45 E
03D4 00 00 FE 66 62 68                                          221             DB      000H,000H,0FEH,066H,062H,068H,078H,068H ; TH_46 F
                        78 68                                   222
03DC 60 60 F0 00 00 00                                          223             DB      060H,060H,0F0H,000H,000H,000H           ; BT_46 F
03E2 00 00 3C 66 C2 C0                                          224             DB      000H,000H,03CH,066H,0C2H,0C0H,0C0H,0DEH ; TH_47 G
                        C0 DE                                   225
03EA C6 66 3A 00 00 00                                          226             DB      0C6H,066H,03AH,000H,000H,000H           ; BT_47 G
03F0 00 00 C6 C6 C6 C6                                          227             DB      000H,000H,0C6H,0C6H,0C6H,0C6H,0FEH,0C6H ; TH_48 H
                        FE C6                                   228
03F8 C6 C6 C6 00 00 00                                          229             DB      0C6H,0C6H,0C6H,000H,000H,000H           ; BT_48 H
03FE 00 00 3C 18 18 18                                          230             DB      000H,000H,03CH,018H,018H,018H,018H,018H ; TH_49 I
                        18 18                                   231
0406 18 18 3C 00 00 00                                          232             DB      018H,018H,03CH,000H,000H,000H           ; BT_49 I
040C 00 00 1E 0C 0C 0C                                          233             DB      000H,000H,01EH,00CH,00CH,00CH,00CH,00CH ; TH_4A J
                        0C 0C                                   234
0414 CC CC 78 00 00 00                                          235             DB      0CCH,0CCH,078H,000H,000H,000H           ; BT_4A J
041A 00 00 E6 66 6C 6C                                          236             DB      000H,000H,0E6H,066H,06CH,06CH,078H,06CH ; TH_4B K
                        78 6C                                   237
0422 6C 66 E6 00 00 00                                          238             DB      06CH,066H,0E6H,000H,000H,000H           ; BT_4B K
0428 00 00 F0 60 60 60                                          239             DB      000H,000H,0F0H,060H,060H,060H,060H,060H ; TH_4C L
                        60 60                                   240
```

```
0430  62 66 FE 00 00 00      241                DB      062H,066H,0FEH,000H,000H,000H           ; BT_4C L
0436  00 00 C6 EE FE FE      242                DB      000H,000H,0C6H,0EEH,0FEH,0FEH,0D6H,0C6H ; TH_4D M
      D6 C6                  243
043E  C6 C6 C6 00 00 00      244                DB      0C6H,0C6H,0C6H,000H,000H,000H           ; BT_4D M
0444  00 00 C6 E6 F6 FE      245                DB      000H,000H,0C6H,0E6H,0F6H,0FEH,0DEH,0CEH ; TH_4E N
      DE CE                  246
044C  C6 C6 C6 00 00 00      247                DB      0C6H,0C6H,0C6H,000H,000H,000H           ; BT_4E N
0452  00 00 38 6C C6 C6      248                DB      000H,000H,038H,06CH,0C6H,0C6H,0C6H,0C6H ; TH_4F O
      C6 C6                  249
045A  C6 6C 38 00 00 00      250                DB      0C6H,06CH,038H,000H,000H,000H           ; BT_4F O
                             251
0460  00 00 FC 66 66 66      252                DB      000H,000H,0FCH,066H,066H,066H,07CH,060H ; TH_5O P
      7C 60                  253
0468  60 60 F0 00 00 00      254                DB      060H,060H,0F0H,000H,000H,000H           ; BT_5O P
046E  00 00 7C C6 C6 C6      255                DB      000H,000H,07CH,0C6H,0C6H,0C6H,0C6H,0D6H ; TH_51 Q
      C6 D6                  256
0476  DE 7C 0C 0E 00 00      257                DB      0DEH,07CH,00CH,00EH,000H,000H           ; BT_51 Q
047C  00 00 FC 66 66 66      258                DB      000H,000H,0FCH,066H,066H,066H,07CH,06CH ; TH_52 R
      7C 6C                  259
0484  66 66 E6 00 00 00      260                DB      066H,066H,0E6H,000H,000H,000H           ; BT_52 R
048A  00 00 7C C6 C6 60      261                DB      000H,000H,07CH,0C6H,0C6H,060H,038H,00CH ; TH_53 S
      38 0C                  262
0492  C6 C6 7C 00 00 00      263                DB      0C6H,0C6H,07CH,000H,000H,000H           ; BT_53 S
0498  00 00 7E 7E 5A 18      264                DB      000H,000H,07EH,07EH,05AH,018H,018H,018H ; TH_54 T
      18 18                  265
04A0  18 18 3C 00 00 00      266                DB      018H,018H,03CH,000H,000H,000H           ; BT_54 T
04A6  00 00 C6 C6 C6 C6      267                DB      000H,000H,0C6H,0C6H,0C6H,0C6H,0C6H,0C6H ; TH_55 U
      C6 C6                  268
04AE  C6 C6 7C 00 00 00      269                DB      0C6H,0C6H,07CH,000H,000H,000H           ; BT_55 U
04B4  00 00 C6 C6 C6 C6      270                DB      000H,000H,0C6H,0C6H,0C6H,0C6H,0C6H,0C6H ; TH_56 V
      C6 C6                  271
04BC  6C 38 10 00 00 00      272                DB      06CH,038H,010H,000H,000H,000H           ; BT_56 V
04C2  00 00 C6 C6 C6 C6      273                DB      000H,000H,0C6H,0C6H,0C6H,0C6H,0D6H,0D6H ; TH_57 W
      D6 D6                  274
04CA  FE 7C 6C 00 00 00      275                DB      0FEH,07CH,06CH,000H,000H,000H           ; BT_57 W
04D0  00 00 C6 C6 6C 38      276                DB      000H,000H,0C6H,0C6H,06CH,038H,038H,038H ; TH_58 X
      38 38                  277
04D8  6C C6 C6 00 00 00      278                DB      06CH,0C6H,0C6H,000H,000H,000H           ; BT_58 X
04DE  00 00 66 66 66 66      279                DB      000H,000H,066H,066H,066H,066H,03CH,018H ; TH_59 Y
      3C 18                  280
04E6  18 18 3C 00 00 00      281                DB      018H,018H,03CH,000H,000H,000H           ; BT_59 Y
04EC  00 00 FE C6 8C 18      282                DB      000H,000H,0FEH,0C6H,08CH,018H,030H,060H ; TH_5A Z
      30 60                  283
04F4  C2 C6 FE 00 00 00      284                DB      0C2H,0C6H,0FEH,000H,000H,000H           ; BT_5A Z
04FA  00 00 3C 30 30 30      285                DB      000H,000H,03CH,030H,030H,030H,030H,030H ; TH_5B [
      30 30                  286
0502  30 30 3C 00 00 00      287                DB      030H,030H,03CH,000H,000H,000H           ; BT_5B [
0508  00 00 80 C0 E0 70      288                DB      000H,000H,080H,0C0H,0E0H,070H,038H,01CH ; TH_5C
      38 1C                  289
0510  0E 06 02 00 00 00      290                DB      00EH,006H,002H,000H,000H,000H           ; BT_5C
0516  00 00 3C 0C 0C 0C      291                DB      000H,000H,03CH,00CH,00CH,00CH,00CH,00CH ; TH_5D ]
      0C 0C                  292
051E  0C 0C 3C 00 00 00      293                DB      00CH,00CH,03CH,000H,000H,000H           ; BT_5D ]
0524  10 38 6C C6 00 00      294                DB      010H,038H,06CH,0C6H,000H,000H,000H,000H ; TH_5E
      00 00                  295
052C  00 00 00 00 00 00      296                DB      000H,000H,000H,000H,000H,000H           ; BT_5E
0532  00 00 00 00 00 00      297                DB      000H,000H,000H,000H,000H,000H,000H,000H ; TH_5F _
      00 00                  298
053A  00 00 00 00 FF 00      299                DB      000H,000H,000H,000H,0FFH,000H           ; BT_5F _
                             300
0540  30 30 18 00 00 00      301                DB      030H,030H,018H,000H,000H,000H,000H,000H ; TH_6O '
      00 00                  302
0548  00 00 00 00 00 00      303                DB      000H,000H,000H,000H,000H,000H           ; BT_6O '
054E  00 00 00 00 00 78      304                DB      000H,000H,000H,000H,000H,078H,00CH,07CH ; TH_61 LOWER_CASE A
      0C 7C                  305
0556  CC CC 76 00 00 00      306                DB      0CCH,0CCH,076H,000H,000H,000H           ; BT_61 LOWER_CASE A
055C  00 00 E0 60 60 78      307                DB      000H,000H,0E0H,060H,060H,078H,06CH,066H ; TH_62 L.C. B
      6C 66                  308
0564  66 66 7C 00 00 00      309                DB      066H,066H,07CH,000H,000H,000H           ; BT_62 L.C. B
056A  00 00 00 00 00 7C      310                DB      000H,000H,000H,000H,000H,07CH,0C6H,0C0H ; TH_63 L.C. C
      C6 C0                  311
0572  C0 C6 7C 00 00 00      312                DB      0C0H,0C6H,07CH,000H,000H,000H           ; BT_63 L.C. C
0578  00 00 1C 0C 0C 3C      313                DB      000H,000H,01CH,00CH,00CH,03CH,06CH,0CCH ; TH_64 L.C. D
      6C CC                  314
0580  CC CC 76 00 00 00      315                DB      0CCH,0CCH,076H,000H,000H,000H           ; BT_64 L.C. D
0586  00 00 00 00 00 7C      316                DB      000H,000H,000H,000H,000H,07CH,0C6H,0FEH ; TH_65 L.C. E
      C6 FE                  317
058E  C0 C6 7C 00 00 00      318                DB      0C0H,0C6H,07CH,000H,000H,000H           ; BT_65 L.C. E
0594  00 00 38 6C 64 60      319                DB      000H,000H,038H,06CH,064H,060H,0F0H,060H ; TH_66 L.C. F
      F0 60                  320
059C  60 60 F0 00 00 00      321                DB      060H,060H,0F0H,000H,000H,000H           ; BT_66 L.C. F
05A2  00 00 00 00 00 76      322                DB      000H,000H,000H,000H,000H,076H,0CCH,0CCH ; TH_67 L.C. G
      CC CC                  323
05AA  CC 7C 0C CC 78 00      324                DB      0CCH,07CH,00CH,0CCH,078H,000H           ; BT_67 L.C. G
05B0  00 00 E0 60 60 6C      325                DB      000H,000H,0E0H,060H,060H,06CH,076H,066H ; TH_68 L.C. H
      76 66                  326
05B8  66 66 E6 00 00 00      327                DB      066H,066H,0E6H,000H,000H,000H           ; BT_68 L.C. H
05BE  00 00 18 18 00 38      328                DB      000H,000H,018H,018H,000H,038H,018H,018H ; TH_69 L.C. I
      18 18                  329
05C6  18 18 3C 00 00 00      330                DB      018H,018H,03CH,000H,000H,000H           ; BT_69 L.C. I
05CC  00 00 06 06 00 0E      331                DB      000H,000H,006H,006H,000H,00EH,006H,006H ; TH_6A L.C. J
      06 06                  332
05D4  06 06 66 66 3C 00      333                DB      006H,006H,066H,066H,03CH,000H           ; BT_6A L.C. J
05DA  00 00 E0 60 60 66      334                DB      000H,000H,0E0H,060H,060H,066H,06CH,078H ; TH_6B L.C. K
      6C 78                  335
05E2  6C 66 E6 00 00 00      336                DB      06CH,066H,0E6H,000H,000H,000H           ; BT_6B L.C. K
05E8  00 00 38 18 18 18      337                DB      000H,000H,038H,018H,018H,018H,018H,018H ; TH_6C L.C. L
      18 18                  338
05F0  18 18 3C 00 00 00      339                DB      018H,018H,03CH,000H,000H,000H           ; BT_6C L.C. L
05F6  00 00 00 00 00 EC      340                DB      000H,000H,000H,000H,000H,0ECH,0FEH,0D6H ; TH_6D L.C. M
      FE D6                  341
05FE  D6 D6 C6 00 00 00      342                DB      0D6H,0D6H,0C6H,000H,000H,000H           ; BT_6D L.C. M
0604  00 00 00 00 00 DC      343                DB      000H,000H,000H,000H,000H,0DCH,066H,066H ; TH_6E L.C. N
      66 66                  344
060C  66 66 66 00 00 00      345                DB      066H,066H,066H,000H,000H,000H           ; BT_6E L.C. N
0612  00 00 00 00 00 7C      346                DB      000H,000H,000H,000H,000H,07CH,0C6H,0C6H ; TH_6F L.C. O
      C6 C6                  347
061A  C6 C6 7C 00 00 00      348                DB      0C6H,0C6H,07CH,000H,000H,000H           ; BT_6F L.C. O
                             349
0620  00 00 00 00 00 DC      350                DB      000H,000H,000H,000H,000H,0DCH,066H,066H ; TH_7O L.C. P
      66 66                  351
0628  66 7C 60 60 F0 00      352                DB      066H,07CH,060H,060H,0F0H,000H           ; BT_7O L.C. P
062E  00 00 00 00 00 76      353                DB      000H,000H,000H,000H,000H,076H,0CCH,0CCH ; TH_71 L.C. Q
      CC CC                  354
0636  CC 7C 0C 0C 1E 00      355                DB      0CCH,07CH,00CH,00CH,01EH,000H           ; BT_71 L.C. Q
063C  00 00 00 00 00 DC      356                DB      000H,000H,000H,000H,000H,0DCH,076H,066H ; TH_72 L.C. R
      76 66                  357
0644  60 60 F0 00 00 00      358                DB      060H,060H,0F0H,000H,000H,000H           ; BT_72 L.C. R
064A  00 00 00 00 00 7C      359                DB      000H,000H,000H,000H,000H,07CH,0C6H,070H ; TH_73 L.C. S
      C6 70                  360
0652  1C C6 7C 00 00 00      361                DB      01CH,0C6H,07CH,000H,000H,000H           ; BT_73 L.C. S
0658  00 00 10 30 30 FC      362                DB      000H,000H,010H,030H,030H,0FCH,030H,030H ; TH_74 L.C. T
      30 30                  363
0660  30 36 1C 00 00 00      364                DB      030H,036H,01CH,000H,000H,000H           ; BT_74 L.C. T
0666  00 00 00 00 00 CC      365                DB      000H,000H,000H,000H,000H,0CCH,0CCH,0CCH ; TH_75 L.C. U
      CC CC                  366
```

```
066E  CC CC 76 00 00 00 00      367                     DB      0CCH,0CCH,076H,000H,000H,000H,000H              ; BT_75 L.C. U
0674  00 00 00 00 00 00 66      368                     DB      000H,000H,000H,000H,000H,066H,066H,066H ; TH_76 L.C. V
      66 66                     369
067C  66 3C 18 00 00 00 00      370                     DB      066H,03CH,018H,000H,000H,000H,000H              ; BT_76 L.C. V
0682  00 00 00 00 00 00 C6      371                     DB      000H,000H,000H,000H,000H,0C6H,0C6H,0D6H ; TH_77 L.C. W
      C6 D6                     372
068A  D6 FE 6C 00 00 00 00      373                     DB      0D6H,0FEH,06CH,000H,000H,000H,000H              ; BT_77 L.C. W
0690  00 00 00 00 00 00 C6      374                     DB      000H,000H,000H,000H,000H,0C6H,06CH,038H ; TH_78 L.C. X
      6C 38                     375
0698  38 6C C6 00 00 00 00      376                     DB      038H,06CH,0C6H,000H,000H,000H,000H              ; BT_78 L.C. X
069E  00 00 00 00 00 00 C6      377                     DB      000H,000H,000H,000H,000H,0C6H,0C6H,0C6H ; TH_79 L.C. Y
      C6 C6                     378
06A6  C6 7E 06 0C F8 00 00      379                     DB      0C6H,07EH,006H,00CH,0F8H,000H,000H              ; BT_79 L.C. Y
06AC  00 00 00 00 00 00 FE      380                     DB      000H,000H,000H,000H,000H,0FEH,0CCH,018H ; TH_7A L.C. Z
      CC 18                     381
06B4  30 66 FE 00 00 00 00      382                     DB      030H,066H,0FEH,000H,000H,000H,000H              ; BT_7A L.C. Z
06BA  00 00 0E 18 18 18 18      383                     DB      000H,000H,00EH,018H,018H,018H,070H,018H ; TH_7B L BRAK
      70 18                     384
06C2  18 18 0E 00 00 00 00      385                     DB      018H,018H,00EH,000H,000H,000H,000H              ; BT_7B L BRAK
06C8  00 00 18 18 18 18 18      386                     DB      000H,000H,018H,018H,018H,018H,000H,018H ; TH_7C |
      00 18                     387
06D0  18 18 18 00 00 00 00      388                     DB      018H,018H,018H,000H,000H,000H,000H              ; BT_7C |
06D6  00 00 70 18 18 18 18      389                     DB      000H,000H,070H,018H,018H,018H,00EH,018H ; TH_7D R BRAK
      0E 18                     390
06DE  18 18 70 00 00 00 00      391                     DB      018H,018H,070H,000H,000H,000H,000H              ; BT_7D R BRAK
06E4  00 00 76 DC 00 00 00      392                     DB      000H,000H,076H,0DCH,000H,000H,000H,000H ; TH_7E TILDE
      00 00                     393
06EC  00 00 00 00 00 00 00      394                     DB      000H,000H,000H,000H,000H,000H,000H              ; BT_7E TILDE
06F2  00 00 00 00 10 38 6C      395                     DB      000H,000H,000H,000H,010H,038H,06CH,0C6H ; TH_7F DELTA
      C6 C6                     396
06FA  C6 FE 00 00 00 00 00      397                     DB      0C6H,0FEH,000H,000H,000H,000H,000H              ; BT_7F DELTA
                                398
0700  00 00 3C 66 C2 C0 C0      399                     DB      000H,000H,03CH,066H,0C2H,0C0H,0C0H,0C2H ; TH_80
      C0 C2                     400
0708  66 3C 0C 06 7C 00 00      401                     DB      066H,03CH,00CH,006H,07CH,000H,000H              ; BT_80
070E  00 00 CC CC 00 CC CC      402                     DB      000H,000H,0CCH,0CCH,000H,0CCH,0CCH,0CCH ; TH_81
      CC CC                     403
0716  CC CC 76 00 00 00 00      404                     DB      0CCH,0CCH,076H,000H,000H,000H,000H              ; BT_81
071C  00 0C 18 30 00 7C C6      405                     DB      000H,00CH,018H,030H,000H,07CH,0C6H,0FEH ; TH_82
      C6 FE                     406
0724  C0 C6 7C 00 00 00 00      407                     DB      0C0H,0C6H,07CH,000H,000H,000H,000H              ; BT_82
072A  00 10 38 6C 00 78 0C      408                     DB      000H,010H,038H,06CH,000H,078H,00CH,07CH ; TH_83
      0C 7C                     409
0732  CC CC 76 00 00 00 00      410                     DB      0CCH,0CCH,076H,000H,000H,000H,000H              ; BT_83
0738  00 00 CC CC 00 78 0C      411                     DB      000H,000H,0CCH,0CCH,000H,078H,00CH,07CH ; TH_84
      0C 7C                     412
0740  CC CC 76 00 00 00 00      413                     DB      0CCH,0CCH,076H,000H,000H,000H,000H              ; BT_84
0746  00 60 30 18 00 78 0C      414                     DB      000H,060H,030H,018H,000H,078H,00CH,07CH ; TH_85
      0C 7C                     415
074E  CC CC 76 00 00 00 00      416                     DB      0CCH,0CCH,076H,000H,000H,000H,000H              ; BT_85
0754  00 38 6C 38 00 78 0C      417                     DB      000H,038H,06CH,038H,000H,078H,00CH,07CH ; TH_86
      0C 7C                     418
075C  CC CC 76 00 00 00 00      419                     DB      0CCH,0CCH,076H,000H,000H,000H,000H              ; BT_86
0762  00 00 00 00 3C 66 60      420                     DB      000H,000H,000H,000H,03CH,066H,060H,066H ; TH_87
      60 66                     421
076A  3C 0C 06 3C 00 00 00      422                     DB      03CH,00CH,006H,03CH,000H,000H,000H              ; BT_87
0770  00 10 38 6C 00 7C C6      423                     DB      000H,010H,038H,06CH,000H,07CH,0C6H,0FEH ; TH_88
      C6 FE                     424
0778  C0 C6 7C 00 00 00 00      425                     DB      0C0H,0C6H,07CH,000H,000H,000H,000H              ; BT_88
077E  00 00 CC CC 00 7C C6      426                     DB      000H,000H,0CCH,0CCH,000H,07CH,0C6H,0FEH ; TH_89
      C6 FE                     427
0786  C0 C6 7C 00 00 00 00      428                     DB      0C0H,0C6H,07CH,000H,000H,000H,000H              ; BT_89
078C  00 60 30 18 00 7C C6      429                     DB      000H,060H,030H,018H,000H,07CH,0C6H,0FEH ; TH_8A
      C6 FE                     430
0794  C0 C6 7C 00 00 00 00      431                     DB      0C0H,0C6H,07CH,000H,000H,000H,000H              ; BT_8A
079A  00 00 66 66 00 38 18      432                     DB      000H,000H,066H,066H,000H,038H,018H,018H ; TH_8B
      18 18                     433
07A2  18 18 3C 00 00 00 00      434                     DB      018H,018H,03CH,000H,000H,000H,000H              ; BT_8B
07A8  00 18 3C 66 00 38 18      435                     DB      000H,018H,03CH,066H,000H,038H,018H,018H ; TH_8C
      18 18                     436
07B0  18 18 3C 00 00 00 00      437                     DB      018H,018H,03CH,000H,000H,000H,000H              ; BT_8C
07B6  00 60 30 18 00 38 18      438                     DB      000H,060H,030H,018H,000H,038H,018H,018H ; TH_8D
      18 18                     439
07BE  18 18 3C 00 00 00 00      440                     DB      018H,018H,03CH,000H,000H,000H,000H              ; BT_8D
07C4  00 C6 C6 10 38 6C C6      441                     DB      000H,0C6H,0C6H,010H,038H,06CH,0C6H,0C6H ; TH_8E
      C6 C6                     442
07CC  FE C6 C6 00 00 00 00      443                     DB      0FEH,0C6H,0C6H,000H,000H,000H,000H              ; BT_8E
07D2  38 6C 38 00 38 6C C6      444                     DB      038H,06CH,038H,000H,038H,06CH,0C6H,0C6H ; TH_8F
      C6 C6                     445
07DA  FE C6 C6 00 00 00 00      446                     DB      0FEH,0C6H,0C6H,000H,000H,000H,000H              ; BT_8F
                                447
07E0  18 30 60 00 FE 66 60      448                     DB      018H,030H,060H,000H,0FEH,066H,060H,07CH ; TH_90
      60 7C                     449
07E8  60 66 FE 00 00 00 00      450                     DB      060H,066H,0FEH,000H,000H,000H,000H              ; BT_90
07EE  00 00 00 00 CC 76 36      451                     DB      000H,000H,000H,000H,0CCH,076H,036H,07EH ; TH_91
      36 7E                     452
07F6  D8 D8 6E 00 00 00 00      453                     DB      0D8H,0D8H,06EH,000H,000H,000H,000H              ; BT_91
07FC  00 00 3E 6C CC CC FE      454                     DB      000H,000H,03EH,06CH,0CCH,0CCH,0FEH,0CCH ; TH_92
      CC                        455
0804  CC CC CE 00 00 00 00      456                     DB      0CCH,0CCH,0CEH,000H,000H,000H,000H              ; BT_92
080A  00 10 38 6C 00 7C C6      457                     DB      000H,010H,038H,06CH,000H,07CH,0C6H,0C6H ; TH_93
      C6 C6                     458
0812  C6 C6 7C 00 00 00 00      459                     DB      0C6H,0C6H,07CH,000H,000H,000H,000H              ; BT_93
0818  00 00 C6 C6 00 7C C6      460                     DB      000H,000H,0C6H,0C6H,000H,07CH,0C6H,0C6H ; TH_94
      C6 C6                     461
0820  C6 C6 7C 00 00 00 00      462                     DB      0C6H,0C6H,07CH,000H,000H,000H,000H              ; BT_94
0826  00 60 30 18 00 7C C6      463                     DB      000H,060H,030H,018H,000H,07CH,0C6H,0C6H ; TH_95
      C6 C6                     464
082E  C6 C6 7C 00 00 00 00      465                     DB      0C6H,0C6H,07CH,000H,000H,000H,000H              ; BT_95
0834  00 30 78 CC 00 CC CC      466                     DB      000H,030H,078H,0CCH,000H,0CCH,0CCH,0CCH ; TH_96
      CC CC                     467
083C  CC CC 76 00 00 00 00      468                     DB      0CCH,0CCH,076H,000H,000H,000H,000H              ; BT_96
0842  00 60 30 18 00 CC CC      469                     DB      000H,060H,030H,018H,000H,0CCH,0CCH,0CCH ; TH_97
      CC CC                     470
084A  CC CC 76 00 00 00 00      471                     DB      0CCH,0CCH,076H,000H,000H,000H,000H              ; BT_97
0850  00 00 C6 C6 00 C6 C6      472                     DB      000H,000H,0C6H,0C6H,000H,0C6H,0C6H,0C6H ; TH_98
      C6 C6                     473
0858  C6 7E 06 0C 78 00 00      474                     DB      0C6H,07EH,006H,00CH,078H,000H,000H              ; BT_98
085E  00 C6 C6 38 6C C6 C6      475                     DB      000H,0C6H,0C6H,038H,06CH,0C6H,0C6H,0C6H ; TH_99
      C6 C6                     476
0866  C6 6C 38 00 00 00 00      477                     DB      0C6H,06CH,038H,000H,000H,000H,000H              ; BT_99
086C  00 C6 C6 00 C6 C6 C6      478                     DB      000H,0C6H,0C6H,000H,0C6H,0C6H,0C6H,0C6H ; TH_9A
      C6 C6                     479
0874  C6 C6 7C 00 00 00 00      480                     DB      0C6H,0C6H,07CH,000H,000H,000H,000H              ; BT_9A
087A  00 18 18 3C 66 60 60      481                     DB      000H,018H,018H,03CH,066H,060H,060H,066H ; TH_9B
      60 66                     482
0882  3C 18 18 00 00 00 00      483                     DB      03CH,018H,018H,000H,000H,000H,000H              ; BT_9B
0888  00 38 6C 64 60 F0 60      484                     DB      000H,038H,06CH,064H,060H,0F0H,060H,060H ; TH_9C
      60 60                     485
0890  60 E6 FC 00 00 00 00      486                     DB      060H,0E6H,0FCH,000H,000H,000H,000H              ; BT_9C
0896  00 00 66 66 3C 18 7E      487                     DB      000H,000H,066H,066H,03CH,018H,07EH,018H ; TH_9D
      7E 18                     488
089E  7E 18 18 00 00 00 00      489                     DB      07EH,018H,018H,000H,000H,000H,000H              ; BT_9D
08A4  00 F8 CC CC F8 C4 CC      490                     DB      000H,0F8H,0CCH,0CCH,0F8H,0C4H,0CCH,0DEH ; TH_9E
      CC DE                     491
08AC  CC CC C6 00 00 00 00      492                     DB      0CCH,0CCH,0C6H,000H,000H,000H,000H              ; BT_9E
```

```
08B2  00 0E 1B 18 18 18      493                DB      000H,00EH,01BH,018H,018H,018H,07EH,018H ; TH_9F
      7E 18                  494
08BA  18 18 18 D8 70 00      495                DB      018H,018H,018H,0D8H,070H,000H           ; BT_9F
                             496
08C0  00 18 30 60 00 78      497                DB      000H,018H,030H,060H,000H,078H,00CH,07CH ; TH_A0
      0C 7C                  498
08C8  CC CC 76 00 00 00      499                DB      0CCH,0CCH,076H,000H,000H,000H           ; BT_A0
08CE  00 0C 18 30 00 38      500                DB      000H,00CH,018H,030H,000H,038H,018H,018H ; TH_A1
      18 18                  501
08D6  18 18 3C 00 00 00      502                DB      018H,018H,03CH,000H,000H,000H           ; BT_A1
08DC  00 18 30 60 00 7C      503                DB      000H,018H,030H,060H,000H,07CH,0C6H,0C6H ; TH_A2
      C6 C6                  504
08E4  C6 C6 7C 00 00 00      505                DB      0C6H,0C6H,07CH,000H,000H,000H           ; BT_A2
08EA  00 18 30 60 00 CC      506                DB      000H,018H,030H,060H,000H,0CCH,0CCH,0CCH ; TH_A3
      CC CC                  507
08F2  CC CC 76 00 00 00      508                DB      0CCH,0CCH,076H,000H,000H,000H           ; BT_A3
08F8  00 00 76 DC 00 DC      509                DB      000H,000H,076H,0DCH,000H,0DCH,066H,066H ; TH_A4
      66 66                  510
0900  66 66 66 00 00 00      511                DB      066H,066H,066H,000H,000H,000H           ; BT_A4
0906  76 DC 00 C6 E6 F6      512                DB      076H,0DCH,000H,0C6H,0E6H,0F6H,0FEH,0DEH ; TH_A5
      FE DE                  513
090E  CE C6 C6 00 00 00      514                DB      0CEH,0C6H,0C6H,000H,000H,000H           ; BT_A5
0914  00 3C 6C 6C 3E 00      515                DB      000H,03CH,06CH,06CH,03EH,000H,07EH,000H ; TH_A6
      7E 00                  516
091C  00 00 00 00 00 00      517                DB      000H,000H,000H,000H,000H,000H           ; BT_A6
0922  00 38 6C 6C 38 00      518                DB      000H,038H,06CH,06CH,038H,000H,07CH,000H ; TH_A7
      7C 00                  519
092A  00 00 00 00 00 00      520                DB      000H,000H,000H,000H,000H,000H           ; BT_A7
0930  00 00 30 30 00 30      521                DB      000H,000H,030H,030H,000H,030H,030H,060H ; TH_A8
      30 60                  522
0938  C6 C6 7C 00 00 00      523                DB      0C6H,0C6H,07CH,000H,000H,000H           ; BT_A8
093E  00 00 00 00 00 00      524                DB      000H,000H,000H,000H,000H,000H,0FEH,0C0H ; TH_A9
      FE C0                  525
0946  C0 C0 00 00 00 00      526                DB      0C0H,0C0H,000H,000H,000H,000H           ; BT_A9
094C  00 00 00 00 00 00      527                DB      000H,000H,000H,000H,000H,000H,0FEH,006H ; TH_AA
      FE 06                  528
0954  06 06 00 00 00 00      529                DB      006H,006H,000H,000H,000H,000H           ; BT_AA
095A  00 C0 C0 C6 CC D8      530                DB      000H,0C0H,0C0H,0C6H,0CCH,0D8H,030H,060H ; TH_AB
      30 60                  531
0962  DC 86 0C 18 3E 00      532                DB      0DCH,086H,0CCH,018H,03EH,000H           ; BT_AB
0968  00 C0 C0 C6 CC D8      533                DB      000H,0C0H,0C0H,0C6H,0CCH,0D8H,030H,066H ; TH_AC
      30 66                  534
0970  CE 9E 3E 06 06 00      535                DB      0CEH,09EH,03EH,006H,006H,000H           ; BT_AC
0976  00 00 18 18 00 18      536                DB      000H,000H,018H,018H,000H,018H,018H,03CH ; TH_AD
      18 3C                  537
097E  3C 3C 18 00 00 00      538                DB      03CH,03CH,018H,000H,000H,000H           ; BT_AD
0984  00 00 00 00 36 6C      539                DB      000H,000H,000H,000H,036H,06CH,0D8H,06CH ; TH_AE
      D8 6C                  540
098C  36 00 00 00 00 00      541                DB      036H,000H,000H,000H,000H,000H           ; BT_AE
0992  00 00 00 00 D8 6C      542                DB      000H,000H,000H,000H,0D8H,06CH,036H,06CH ; TH_AF
      36 6C                  543
099A  D8 00 00 00 00 00      544                DB      0D8H,000H,000H,000H,000H,000H           ; BT_AF
                             545
09A0  11 44 11 44 11 44      546                DB      011H,044H,011H,044H,011H,044H,011H,044H ; TH_B0
      11 44                  547
09A8  11 44 11 44 11 44      548                DB      011H,044H,011H,044H,011H,044H           ; BT_B0
09AE  55 AA 55 AA 55 AA      549                DB      055H,0AAH,055H,0AAH,055H,0AAH,055H,0AAH ; TH_B1
      55 AA                  550
09B6  55 AA 55 AA 55 AA      551                DB      055H,0AAH,055H,0AAH,055H,0AAH           ; BT_B1
09BC  DD 77 DD 77 DD 77      552                DB      0DDH,077H,0DDH,077H,0DDH,077H,0DDH,077H ; TH_B2
      DD 77                  553
09C4  DD 77 DD 77 DD 77      554                DB      0DDH,077H,0DDH,077H,0DDH,077H           ; BT_B2
09CA  18 18 18 18 18 18      555                DB      018H,018H,018H,018H,018H,018H,018H,018H ; TH_B3
      18 18                  556
09D2  18 18 18 18 18 18      557                DB      018H,018H,018H,018H,018H,018H           ; BT_B3
09D8  18 18 18 18 18 18      558                DB      018H,018H,018H,018H,018H,018H,018H,0F8H ; TH_B4
      18 F8                  559
09E0  18 18 18 18 18 18      560                DB      018H,018H,018H,018H,018H,018H           ; BT_B4
09E6  18 18 18 18 18 F8      561                DB      018H,018H,018H,018H,018H,0F8H,018H,0F8H ; TH_B5
      18 F8                  562
09EE  18 18 18 18 18 18      563                DB      018H,018H,018H,018H,018H,018H           ; BT_B5
09F4  36 36 36 36 36 36      564                DB      036H,036H,036H,036H,036H,036H,036H,0F6H ; TH_B6
      36 F6                  565
09FC  36 36 36 36 36 36      566                DB      036H,036H,036H,036H,036H,036H           ; BT_B6
0A02  00 00 00 00 00 00      567                DB      000H,000H,000H,000H,000H,000H,000H,0FEH ; TH_B7
      00 FE                  568
0A0A  36 36 36 36 36 36      569                DB      036H,036H,036H,036H,036H,036H           ; BT_B7
0A10  00 00 00 00 00 F8      570                DB      000H,000H,000H,000H,000H,0F8H,018H,0F8H ; TH_B8
      18 F8                  571
0A18  18 18 18 18 18 18      572                DB      018H,018H,018H,018H,018H,018H           ; BT_B8
0A1E  36 36 36 36 36 F6      573                DB      036H,036H,036H,036H,036H,0F6H,006H,0F6H ; TH_B9
      06 F6                  574
0A26  36 36 36 36 36 36      575                DB      036H,036H,036H,036H,036H,036H           ; BT_B9
0A2C  36 36 36 36 36 36      576                DB      036H,036H,036H,036H,036H,036H,036H,036H ; TH_BA
      36 36                  577
0A34  36 36 36 36 36 36      578                DB      036H,036H,036H,036H,036H,036H           ; BT_BA
0A3A  00 00 00 00 00 FE      579                DB      000H,000H,000H,000H,000H,0FEH,006H,0F6H ; TH_BB
      06 F6                  580
0A42  36 36 36 36 36 36      581                DB      036H,036H,036H,036H,036H,036H           ; BT_BB
0A48  36 36 36 36 36 F6      582                DB      036H,036H,036H,036H,036H,0F6H,006H,0FEH ; TH_BC
      06 FE                  583
0A50  00 00 00 00 00 00      584                DB      000H,000H,000H,000H,000H,000H           ; BT_BC
0A56  36 36 36 36 36 36      585                DB      036H,036H,036H,036H,036H,036H,036H,0FEH ; TH_BD
      36 FE                  586
0A5E  00 00 00 00 00 00      587                DB      000H,000H,000H,000H,000H,000H           ; BT_BD
0A64  18 18 18 18 18 F8      588                DB      018H,018H,018H,018H,018H,0F8H,018H,0F8H ; TH_BE
      18 F8                  589
0A6C  00 00 00 00 00 00      590                DB      000H,000H,000H,000H,000H,000H           ; BT_BE
0A72  00 00 00 00 00 00      591                DB      000H,000H,000H,000H,000H,000H,000H,0F8H ; TH_BF
      00 F8                  592
0A7A  18 18 18 18 18 18      593                DB      018H,018H,018H,018H,018H,018H           ; BT_BF
                             594
0A80  18 18 18 18 18 18      595                DB      018H,018H,018H,018H,018H,018H,018H,01FH ; TH_C0
      18 1F                  596
0A88  00 00 00 00 00 00      597                DB      000H,000H,000H,000H,000H,000H           ; BT_C0
0A8E  18 18 18 18 18 18      598                DB      018H,018H,018H,018H,018H,018H,018H,0FFH ; TH_C1
      18 FF                  599
0A96  00 00 00 00 00 00      600                DB      000H,000H,000H,000H,000H,000H           ; BT_C1
0A9C  00 00 00 00 00 00      601                DB      000H,000H,000H,000H,000H,000H,000H,0FFH ; TH_C2
      00 FF                  602
0AA4  18 18 18 18 18 18      603                DB      018H,018H,018H,018H,018H,018H           ; BT_C2
0AAA  18 18 18 18 18 18      604                DB      018H,018H,018H,018H,018H,018H,018H,01FH ; TH_C3
      18 1F                  605
0AB2  18 18 18 18 18 18      606                DB      018H,018H,018H,018H,018H,018H           ; BT_C3
0AB8  00 00 00 00 00 00      607                DB      000H,000H,000H,000H,000H,000H,000H,0FFH ; TH_C4
      00 FF                  608
0AC0  00 00 00 00 00 00      609                DB      000H,000H,000H,000H,000H,000H           ; BT_C4
0AC6  18 18 18 18 18 18      610                DB      018H,018H,018H,018H,018H,018H,018H,0FFH ; TH_C5
      18 FF                  611
0ACE  18 18 18 18 18 18      612                DB      018H,018H,018H,018H,018H,018H           ; BT_C5
0AD4  18 18 18 18 18 1F      613                DB      018H,018H,018H,018H,018H,01FH,018H,01FH ; TH_C6
      18 1F                  614
0ADC  18 18 18 18 18 18      615                DB      018H,018H,018H,018H,018H,018H           ; BT_C6
0AE2  36 36 36 36 36 36      616                DB      036H,036H,036H,036H,036H,036H,036H,037H ; TH_C7
      36 37                  617
0AEA  36 36 36 36 36 36      618                DB      036H,036H,036H,036H,036H,036H           ; BT_C7
```

```
0AF0 36 36 36 36 36 37     619              DB      036H,036H,036H,036H,036H,037H,030H,03FH ; TH_C8
     30 3F                 620
0AF8 00 00 00 00 00 00     621              DB      000H,000H,000H,000H,000H,000H           ; BT_C8
0AFE 00 00 00 00 00 3F     622              DB      000H,000H,000H,000H,000H,03FH,030H,037H ; TH_C9
     30 37                 623
0B06 36 36 36 36 36 36     624              DB      036H,036H,036H,036H,036H,036H           ; BT_C9
0B0C 36 36 36 36 36 F7     625              DB      036H,036H,036H,036H,036H,0F7H,000H,0FFH ; TH_CA
     00 FF                 626
0B14 00 00 00 00 00 00     627              DB      000H,000H,000H,000H,000H,000H           ; BT_CA
0B1A 00 00 00 00 00 FF     628              DB      000H,000H,000H,000H,000H,0FFH,000H,0F7H ; TH_CB
     00 F7                 629
0B22 36 36 36 36 36 36     630              DB      036H,036H,036H,036H,036H,036H           ; BT_CB
0B28 36 36 36 36 36 37     631              DB      036H,036H,036H,036H,036H,037H,030H,037H ; TH_CC
     30 37                 632
0B30 36 36 36 36 36 36     633              DB      036H,036H,036H,036H,036H,036H           ; BT_CC
0B36 00 00 00 00 00 FF     634              DB      000H,000H,000H,000H,000H,0FFH,000H,0FFH ; TH_CD
     00 FF                 635
0B3E 00 00 00 00 00 00     636              DB      000H,000H,000H,000H,000H,000H           ; BT_CD
0B44 36 36 36 36 36 F7     637              DB      036H,036H,036H,036H,036H,0F7H,000H,0F7H ; TH_CE
     00 F7                 638
0B4C 36 36 36 36 36 36     639              DB      036H,036H,036H,036H,036H,036H           ; BT_CE
0B52 18 18 18 18 18 FF     640              DB      018H,018H,018H,018H,018H,0FFH,000H,0FFH ; TH_CF
     00 FF                 641
0B5A 00 00 00 00 00 00     642              DB      000H,000H,000H,000H,000H,000H           ; BT_CF
                           643
0B60 36 36 36 36 36 36     644              DB      036H,036H,036H,036H,036H,036H,036H,0FFH ; TH_D0
     36 FF                 645
0B68 00 00 00 00 00 00     646              DB      000H,000H,000H,000H,000H,000H           ; BT_D0
0B6E 00 00 00 00 00 FF     647              DB      000H,000H,000H,000H,000H,0FFH,000H,0FFH ; TH_D1
     00 FF                 648
0B76 18 18 18 18 18 18     649              DB      018H,018H,018H,018H,018H,018H           ; BT_D1
0B7C 00 00 00 00 00 00     650              DB      000H,000H,000H,000H,000H,000H,000H,0FFH ; TH_D2
     00 FF                 651
0B84 36 36 36 36 36 36     652              DB      036H,036H,036H,036H,036H,036H           ; BT_D2
0B8A 36 36 36 36 36 36     653              DB      036H,036H,036H,036H,036H,036H,036H,03FH ; TH_D3
     36 3F                 654
0B92 00 00 00 00 00 00     655              DB      000H,000H,000H,000H,000H,000H           ; BT_D3
0B98 18 18 18 18 18 1F     656              DB      018H,018H,018H,018H,018H,01FH,018H,01FH ; TH_D4
     18 1F                 657
0BA0 00 00 00 00 00 00     658              DB      000H,000H,000H,000H,000H,000H           ; BT_D4
0BA6 00 00 00 00 00 1F     659              DB      000H,000H,000H,000H,000H,01FH,018H,01FH ; TH_D5
     18 1F                 660
0BAE 18 18 18 18 18 18     661              DB      018H,018H,018H,018H,018H,018H           ; BT_D5
0BB4 00 00 00 00 00 00     662              DB      000H,000H,000H,000H,000H,000H,000H,03FH ; TH_D6
     00 3F                 663
0BBC 36 36 36 36 36 36     664              DB      036H,036H,036H,036H,036H,036H           ; BT_D6
0BC2 36 36 36 36 36 36     665              DB      036H,036H,036H,036H,036H,036H,036H,0FFH ; TH_D7
     36 FF                 666
0BCA 36 36 36 36 36 36     667              DB      036H,036H,036H,036H,036H,036H           ; BT_D7
0BD0 18 18 18 18 18 FF     668              DB      018H,018H,018H,018H,018H,0FFH,018H,0FFH ; TH_D8
     18 FF                 669
0BD8 18 18 18 18 18 18     670              DB      018H,018H,018H,018H,018H,018H           ; BT_D8
0BDE 18 18 18 18 18 18     671              DB      018H,018H,018H,018H,018H,018H,018H,0F8H ; TH_D9
     18 F8                 672
0BE6 00 00 00 00 00 00     673              DB      000H,000H,000H,000H,000H,000H           ; BT_D9
0BEC 00 00 00 00 00 00     674              DB      000H,000H,000H,000H,000H,000H,000H,01FH ; TH_DA
     00 1F                 675
0BF4 18 18 18 18 18 18     676              DB      018H,018H,018H,018H,018H,018H           ; BT_DA
0BFA FF FF FF FF FF FF     677              DB      0FFH,0FFH,0FFH,0FFH,0FFH,0FFH,0FFH,0FFH ; TH_DB
     FF FF                 678
0C02 FF FF FF FF FF FF     679              DB      0FFH,0FFH,0FFH,0FFH,0FFH,0FFH           ; BT_DB
0C08 00 00 00 00 00 00     680              DB      000H,000H,000H,000H,000H,000H,000H,0FFH ; TH_DC
     00 FF                 681
0C10 FF FF FF FF FF FF     682              DB      0FFH,0FFH,0FFH,0FFH,0FFH,0FFH           ; BT_DC
0C16 F0 F0 F0 F0 F0 F0     683              DB      0F0H,0F0H,0F0H,0F0H,0F0H,0F0H,0F0H,0F0H ; TH_DD
     F0 F0                 684
0C1E F0 F0 F0 F0 F0 F0     685              DB      0F0H,0F0H,0F0H,0F0H,0F0H,0F0H           ; BT_DD
0C24 0F 0F 0F 0F 0F 0F     686              DB      00FH,00FH,00FH,00FH,00FH,00FH,00FH,00FH ; TH_DE
     0F 0F                 687
0C2C 0F 0F 0F 0F 0F 0F     688              DB      00FH,00FH,00FH,00FH,00FH,00FH           ; BT_DE
0C32 FF FF FF FF FF FF     689              DB      0FFH,0FFH,0FFH,0FFH,0FFH,0FFH,0FFH,000H ; TH_DF
     FF 00                 690
0C3A 00 00 00 00 00 00     691              DB      000H,000H,000H,000H,000H,000H           ; BT_DF
                           692
0C40 00 00 00 00 00 76     693              DB      000H,000H,000H,000H,000H,076H,0DCH,0D8H ; TH_E0
     DC D8                 694
0C48 D8 DC 76 00 00 00     695              DB      0D8H,0DCH,076H,000H,000H,000H           ; BT_E0
0C4E 00 00 00 00 7C C6     696              DB      000H,000H,000H,000H,07CH,0C6H,0FCH,0C6H ; TH_E1
     FC C6                 697
0C56 C6 FC C0 C0 40 00     698              DB      0C6H,0FCH,0C0H,0C0H,040H,000H           ; BT_E1
0C5C 00 00 FE C6 C6 C0     699              DB      000H,000H,0FEH,0C6H,0C6H,0C0H,0C0H,0C0H ; TH_E2
     C0 C0                 700
0C64 C0 C0 C0 00 00 00     701              DB      0C0H,0C0H,0C0H,000H,000H,000H           ; BT_E2
0C6A 00 00 00 00 FE 6C     702              DB      000H,000H,000H,000H,0FEH,06CH,06CH,06CH ; TH_E3
     6C 6C                 703
0C72 6C 6C 6C 00 00 00     704              DB      06CH,06CH,06CH,000H,000H,000H           ; BT_E3
0C78 00 00 FE C6 60 30     705              DB      000H,000H,0FEH,0C6H,060H,030H,018H,030H ; TH_E4
     18 30                 706
0C80 60 C6 FE 00 00 00     707              DB      060H,0C6H,0FEH,000H,000H,000H           ; BT_E4
0C86 00 00 00 00 00 7E     708              DB      000H,000H,000H,000H,000H,07EH,0D8H,0D8H ; TH_E5
     D8 D8                 709
0C8E D8 D8 70 00 00 00     710              DB      0D8H,0D8H,070H,000H,000H,000H           ; BT_E5
0C94 00 00 00 00 66 66     711              DB      000H,000H,000H,000H,066H,066H,066H,066H ; TH_E6
     66 66                 712
0C9C 7C 60 60 C0 00 00     713              DB      07CH,060H,060H,0C0H,000H,000H           ; BT_E6
0CA2 00 00 00 00 76 DC     714              DB      000H,000H,000H,000H,076H,0DCH,018H,018H ; TH_E7
     18 18                 715
0CAA 18 18 18 00 00 00     716              DB      018H,018H,018H,000H,000H,000H           ; BT_E7
0CB0 00 00 7E 18 3C 66     717              DB      000H,000H,07EH,018H,03CH,066H,066H,066H ; TH_E8
     66 66                 718
0CB8 3C 18 7E 00 00 00     719              DB      03CH,018H,07EH,000H,000H,000H           ; BT_E8
0CBE 00 00 38 6C C6 C6     720              DB      000H,000H,038H,06CH,0C6H,0C6H,0FEH,0C6H ; TH_E9
     FE C6                 721
0CC6 C6 6C 38 00 00 00     722              DB      0C6H,06CH,038H,000H,000H,000H           ; BT_E9
0CCC 00 00 38 6C C6 C6     723              DB      000H,000H,038H,06CH,0C6H,0C6H,0C6H,06CH ; TH_EA
     C6 6C                 724
0CD4 6C 6C EE 00 00 00     725              DB      06CH,06CH,0EEH,000H,000H,000H           ; BT_EA
0CDA 00 00 1E 30 18 0C     726              DB      000H,000H,01EH,030H,018H,00CH,03EH,066H ; TH_EB
     3E 66                 727
0CE2 66 66 3C 00 00 00     728              DB      066H,066H,03CH,000H,000H,000H           ; BT_EB
0CE8 00 00 00 00 00 7E     729              DB      000H,000H,000H,000H,000H,07EH,0DBH,0DBH ; TH_EC
     DB DB                 730
0CF0 7E 00 00 00 00 00     731              DB      07EH,000H,000H,000H,000H,000H           ; BT_EC
0CF6 00 00 03 06 7E DB     732              DB      000H,000H,003H,006H,07EH,0DBH,0DBH,0F3H ; TH_ED
     DB F3                 733
0CFE 7E 60 C0 00 60 00     734              DB      07EH,060H,0C0H,000H,000H,000H           ; BT_ED
0D04 00 00 1C 30 60 60     735              DB      000H,000H,01CH,030H,060H,060H,07CH,060H ; TH_EE
     7C 60                 736
0D0C 60 30 1C 00 00 00     737              DB      060H,030H,01CH,000H,000H,000H           ; BT_EE
0D12 00 00 00 7C C6 C6     738              DB      000H,000H,000H,07CH,0C6H,0C6H,0C6H,0C6H ; TH_EF
     C6 C6                 739
0D1A C6 C6 C6 00 00 00     740              DB      0C6H,0C6H,0C6H,000H,000H,000H           ; BT_EF
                           741
0D20 00 00 00 FE 00 00     742              DB      000H,000H,000H,0FEH,000H,000H,0FEH,000H ; TH_F0
     FE 00                 743
0D28 00 FE 00 00 00 00     744              DB      000H,0FEH,000H,000H,000H,000H           ; BT_F0
```

```
0D2E 00 00 00 18 18 7E     745                DB      000H,000H,000H,018H,018H,07EH,018H,018H ; TH_F1
     18 18                 746
0D36 00 00 FF 00 00 00     747                DB      000H,000H,0FFH,000H,000H,000H           ; BT_F1
0D3C 00 00 30 18 0C 06     748                DB      000H,000H,030H,018H,00CH,006H,00CH,018H ; TH_F2
     0C 18                 749
0D44 30 00 7E 00 00 00     750                DB      030H,000H,07EH,000H,000H,000H           ; BT_F2
0D4A 00 00 0C 18 30 60     751                DB      000H,000H,00CH,018H,030H,060H,030H,018H ; TH_F3
     30 18                 752
0D52 0C 00 7E 00 00 00     753                DB      00CH,000H,07EH,000H,000H,000H           ; BT_F3
0D58 00 00 0E 1B 1B 18     754                DB      000H,000H,00EH,01BH,01BH,018H,018H,018H ; TH_F4
     18 18                 755
0D60 18 18 18 18 18 18     756                DB      018H,018H,018H,018H,018H,018H           ; BT_F4
0D66 18 18 18 18 18 18     757                DB      018H,018H,018H,018H,018H,018H,018H,C18H ; TH_F5
     18 18                 758
0D6E D8 D8 70 00 00 00     759                DB      0D8H,0D8H,070H,000H,000H,000H           ; BT_F5
0D74 00 00 00 18 18 00     760                DB      000H,000H,000H,018H,018H,000H,07EH,000H ; TH_F6
     7E 00                 761
0D7C 18 18 00 00 00 00     762                DB      018H,018H,000H,000H,000H,000H           ; BT_F6
0D82 00 00 00 00 76 DC     763                DB      000H,000H,000H,000H,076H,0DCH,000H,076H ; TH_F7
     00 76                 764
0D8A DC 00 00 00 00 00     765                DB      0DCH,000H,000H,000H,000H,000H           ; BT_F7
0D90 00 38 6C 6C 38 00     766                DB      000H,038H,06CH,06CH,038H,000H,000H,000H ; TH_F8
     00 00                 767
0D98 00 00 00 00 00 00     768                DB      000H,000H,000H,000H,000H,000H           ; BT_F8
0D9E 00 00 00 00 00 00     769                DB      000H,000H,000H,000H,000H,000H,018H,018H ; TH_F9
     18 18                 770
0DA6 00 00 00 00 00 00     771                DB      000H,000H,000H,000H,000H,000H           ; BT_F9
0DAC 00 00 00 00 00 00     772                DB      000H,000H,000H,000H,000H,000H,000H,018H ; TH_FA
     00 18                 773
0DB4 00 00 00 00 00 00     774                DB      000H,000H,000H,000H,000H,000H           ; BT_FA
0DBA 00 0F 0C 0C 0C 0C     775                DB      000H,00FH,00CH,00CH,00CH,00CH,00CH,0ECH ; TH_FB
     0C EC                 776
0DC2 6C 3C 1C 00 00 00     777                DB      06CH,03CH,01CH,000H,000H,000H           ; BT_FB
0DC8 00 D8 6C 6C 6C 6C     778                DB      000H,0D8H,06CH,06CH,06CH,06CH,06CH,000H ; TH_FC
     6C 00                 779
0DD0 00 00 00 00 00 00     780                DB      000H,000H,000H,000H,000H,000H           ; BT_FC
0DD6 00 70 D8 30 60 C8     781                DB      000H,070H,0D8H,030H,060H,0C8H,0F8H,000H ; TH_FD
     F8 00                 782
0DDE 00 00 00 00 00 00     783                DB      000H,000H,000H,000H,000H,000H           ; BT_FD
0DE4 00 00 00 00 7C 7C     784                DB      000H,000H,000H,000H,07CH,07CH,07CH,07CH ; TH_FE
     7C 7C                 785
0DEC 7C 7C 00 00 00 00     786                DB      07CH,07CH,000H,000H,000H,000H           ; BT_FE
0DF2 00 00 00 00 00 00     787                DB      000H,000H,000H,000H,000H,000H,000H,000H ; TH_FF
     00 00                 788
0DFA 00 00 00 00 00 00     789                DB      000H,000H,000H,000H,000H,000H           ; BT_FF
0E00                       790    CODE        ENDS
                           791                END
                           1                  PAGE,120
                           2                  SUBTTL  MONOCHROME CHARACTER GENERATOR  -  ALPHA SUPPLEMENT
0000                       3      CODE        SEGMENT PUBLIC
                           4                  PUBLIC  CGMN_FDG
0000                       5      CGMN_FDG    LABEL   BYTE
                           6
                           7      ; STRUCTURE OF THIS FILE
                           8      ;       DB      XXH     WHERE XX IS THE HEX CODE FOR THE FOLLOWING CHAR
                           9      ;       DB      [BYTES 0 - 13 OF THAT CHARACTER]
                           10     ;       ...
                           11     ;       DB      00H     INDICATES NO MORE REPLACEMENTS TO BE DONE
                           12
                           13
0000 1D                    14                 DB      01DH                                    ;
0001 00 00 00 00 24 66     15                 DB      000H,000H,000H,000H,024H,066H,0FFH,066H ; TH_1D
     FF 66                 16
0009 24 00 00 00 00 00     17                 DB      024H,000H,000H,000H,000H,000H           ; BT_1D
000F 22                    18                 DB      022H                                    ;
0010 00 63 63 63 22 00     19                 DB      000H,063H,063H,063H,022H,000H,000H,000H ; TH_22 "
     00 00                 20
0018 00 00 00 00 00 00     21                 DB      000H,000H,000H,000H,000H,000H           ; BT_22 "
001E 2B                    22                 DB      02BH                                    ;
001F 00 00 00 18 18 18     23                 DB      000H,000H,000H,018H,018H,018H,0FFH,018H ; TH_2B +
     FF 18                 24
0027 18 18 00 00 00 00     25                 DB      018H,018H,000H,000H,000H,000H           ; BT_2B +
002D 2D                    26                 DB      02DH                                    ;
002E 00 00 00 00 00 00     27                 DB      000H,000H,000H,000H,000H,000H,0FFH,000H ; TH_2D -
     FF 00                 28
0036 00 00 00 00 00 00     29                 DB      000H,000H,000H,000H,000H,000H           ; BT_2D -
003C 4D                    30                 DB      04DH                                    ;
003D 00 00 C3 E7 FF DB     31                 DB      000H,000H,0C3H,0E7H,0FFH,0DBH,0C3H,0C3H ; TH_4D M
     C3 C3                 32
0045 C3 C3 C3 00 00 00     33                 DB      0C3H,0C3H,0C3H,000H,000H,000H           ; BT_4D M
004B 54                    34                 DB      054H                                    ;
004C 00 00 FF DB 99 18     35                 DB      000H,000H,0FFH,0DBH,099H,018H,018H,018H ; TH_54 T
     18 18                 36
0054 18 18 3C 00 00 00     37                 DB      018H,018H,03CH,000H,000H,000H           ; BT_54 T
005A 56                    38                 DB      056H                                    ;
005B 00 00 C3 C3 C3 C3     39                 DB      000H,000H,0C3H,0C3H,0C3H,0C3H,0C3H,0C3H ; TH_56 V
     C3 C3                 40
0063 66 3C 18 00 00 00     41                 DB      066H,03CH,018H,000H,000H,000H           ; BT_56 V
0069 57                    42                 DB      057H                                    ;
006A 00 00 C3 C3 C3 C3     43                 DB      000H,000H,0C3H,0C3H,0C3H,0C3H,0DBH,0DBH ; TH_57 W
     DB DB                 44
0072 FF 66 66 00 00 00     45                 DB      0FFH,066H,066H,000H,000H,000H           ; BT_57 W
0078 58                    46                 DB      058H                                    ;
0079 00 00 C3 C3 66 3C     47                 DB      000H,000H,0C3H,0C3H,066H,03CH,018H,03CH ; TH_58 X
     18 3C                 48
0081 66 C3 C3 00 00 00     49                 DB      066H,0C3H,0C3H,000H,000H,000H           ; BT_58 X
0087 59                    50                 DB      059H                                    ;
0088 00 00 C3 C3 C3 66     51                 DB      000H,000H,0C3H,0C3H,0C3H,066H,03CH,018H ; TH_59 Y
     3C 18                 52
0090 18 18 3C 00 00 00     53                 DB      018H,018H,03CH,000H,000H,000H           ; BT_59 Y
0096 5A                    54                 DB      05AH                                    ;
0097 00 00 FF C3 86 0C     55                 DB      000H,000H,0FFH,0C3H,086H,00CH,018H,030H ; TH_5A Z
     18 30                 56
009F 61 C3 FF 00 00 00     57                 DB      061H,0C3H,0FFH,000H,000H,000H           ; BT_5A Z
00A5 6D                    58                 DB      06DH                                    ;
00A6 00 00 00 00 00 E6     59                 DB      000H,000H,000H,000H,000H,0E6H,0FFH,0DBH ; TH_6D L.C. M
     FF DB                 60
00AE DB DB DB 00 00 00     61                 DB      0DBH,0DBH,0DBH,000H,000H,000H           ; BT_6D L.C. M
00B4 76                    62                 DB      076H                                    ;
00B5 00 00 00 00 00 C3     63                 DB      000H,000H,000H,000H,000H,0C3H,0C3H,0C3H ; TH_76 L.C. V
     C3 C3                 64
00BD 66 3C 18 00 00 00     65                 DB      066H,03CH,018H,000H,000H,000H           ; BT_76 L.C. V
00C3 77                    66                 DB      077H                                    ;
00C4 00 00 00 00 00 C3     67                 DB      000H,000H,000H,000H,000H,0C3H,0C3H,0DBH ; TH_77 L.C. W
     C3 DB                 68
00CC DB FF 66 00 00 00     69                 DB      0DBH,0FFH,066H,000H,000H,000H           ; BT_77 L.C. W
00D2 91                    70                 DB      091H                                    ;
00D3 00 00 00 00 6E 3B     71                 DB      000H,000H,000H,000H,06EH,03BH,01BH,07EH ; TH_91
     1B 7E                 72
00DB D8 DC 77 00 00 00     73                 DB      0D8H,0DCH,077H,000H,000H,000H           ; BT_91
00E1 9B                    74                 DB      09BH                                    ;
00E2 00 18 18 7E C3 C0     75                 DB      000H,018H,018H,07EH,0C3H,0C0H,0C0H,0C3H ; TH_9B
     C0 C3                 76
00EA 7E 18 18 00 00 00     77                 DB      07EH,018H,018H,000H,000H,000H           ; BT_9B
00F0 9D                    78                 DB      09DH                                    ;
00F1 00 00 C3 66 3C 18     79                 DB      000H,000H,0C3H,066H,03CH,018H,0FFH,018H ; TH_9D
```

```
                        FF 18
00F9 FF 18 18 00 00 00
00FF 9E
0100 00 FC 66 66 7C 62
     66 6F
0108 66 66 F3 00 00 00
010E F1
010F 00 00 18 18 18 FF
     18 18
0117 18 00 FF 00 00 00
011D F6
011E 00 00 18 18 00 00
     FF 00
0126 00 18 18 00 00 00
012C 00
012D
```

```
                        80
                        81              DB      0FFH,018H,018H,000H,000H,000H           ; BT_9D
                        82              DB      09EH                                    ;
                        83              DB      000H,0FCH,066H,066H,07CH,062H,066H,06FH ; TH_9E
                        84
                        85              DB      066H,066H,0F3H,000H,000H,000H           ; BT_9E
                        86              DB      0F1H                                    ;
                        87              DB      000H,000H,018H,018H,018H,0FFH,018H,018H ; TH_F1
                        88
                        89              DB      018H,000H,0FFH,000H,000H,000H           ; BT_F1
                        90              DB      0F6H                                    ;
                        91              DB      000H,000H,018H,018H,000H,000H,0FFH,000H ; TH_F6
                        92
                        93              DB      000H,018H,018H,000H,000H,000H           ; BT_F6
                        94              DB      000H                                    ; NO MORE
                        95      CODE    ENDS
                        96              END
                        1               PAGE,120
                        2               SUBTTL  DOUBLE DOT CHARACTER GENERATOR
0000                    3       CODE    SEGMENT PUBLIC
                        4               PUBLIC  CGDDOT,INT_1F_1
0000                    5       CGDDOT  LABEL   BYTE
                        6                                                               ; DOUBLE DOT
```

```
0000 00 00 00 00 00 00  7               DB      000H,000H,000H,000H,000H,000H,000H,000H ; D_00
     00 00
0008 7E 81 A5 81 BD 99  9               DB      07EH,081H,0A5H,081H,0BDH,099H,081H,07EH ; D_01
     81 7E
0010 7E FF DB FF C3 E7  11              DB      07EH,0FFH,0DBH,0FFH,0C3H,0E7H,0FFH,07EH ; D_02
     FF 7E
0018 6C FE FE FE 7C 38  13              DB      06CH,0FEH,0FEH,0FEH,07CH,038H,010H,000H ; D_03
     10 00
0020 10 38 7C FE 7C 38  15              DB      010H,038H,07CH,0FEH,07CH,038H,010H,000H ; D_04
     10 00
0028 38 7C 38 FE FE 7C  17              DB      038H,07CH,038H,0FEH,0FEH,07CH,038H,07CH ; D_05
     38 7C
0030 10 10 38 7C FE 7C  19              DB      010H,010H,038H,07CH,0FEH,07CH,038H,07CH ; D_06
     38 7C
0038 00 00 18 3C 3C 18  21              DB      000H,000H,018H,03CH,03CH,018H,000H,000H ; D_07
     00 00
0040 FF FF E7 C3 C3 E7  23              DB      0FFH,0FFH,0E7H,0C3H,0C3H,0E7H,0FFH,0FFH ; D_08
     FF FF
0048 00 3C 66 42 42 66  25              DB      000H,03CH,066H,042H,042H,066H,03CH,000H ; D_09
     3C 00
0050 FF C3 99 BD BD 99  27              DB      0FFH,0C3H,099H,0BDH,0BDH,099H,0C3H,0FFH ; D_0A
     C3 FF
0058 0F 07 0F 7D CC CC  29              DB      00FH,007H,00FH,07DH,0CCH,0CCH,0CCH,078H ; D_0B
     CC 78
0060 3C 66 66 66 3C 18  31              DB      03CH,066H,066H,066H,03CH,018H,07EH,018H ; D_0C
     7E 18
0068 3F 33 3F 30 30 70  33              DB      03FH,033H,03FH,030H,030H,070H,0F0H,0E0H ; D_0D
     F0 E0
0070 7F 63 7F 63 63 67  35              DB      07FH,063H,07FH,063H,063H,067H,0E6H,0C0H ; D_0E
     E6 C0
0078 99 5A 3C E7 E7 3C  37              DB      099H,05AH,03CH,0E7H,0E7H,03CH,05AH,099H ; D_0F
     5A 99
```

```
0080 80 E0 F8 FE F8 E0  40              DB      080H,0E0H,0F8H,0FEH,0F8H,0E0H,080H,000H ; D_10
     80 00
0088 02 0E 3E FE 3E 0E  42              DB      002H,00EH,03EH,0FEH,03EH,00EH,002H,000H ; D_11
     02 00
0090 18 3C 7E 18 18 7E  44              DB      018H,03CH,07EH,018H,018H,07EH,03CH,018H ; D_12
     3C 18
0098 66 66 66 66 66 00  46              DB      066H,066H,066H,066H,066H,000H,066H,000H ; D_13
     66 00
00A0 7F DB DB 7B 1B 1B  48              DB      07FH,0DBH,0DBH,07BH,01BH,01BH,01BH,000H ; D_14
     1B 00
00A8 3E 63 38 6C 6C 38  50              DB      03EH,063H,038H,06CH,06CH,038H,0CCH,078H ; D_15
     CC 78
00B0 00 00 00 00 7E 7E  52              DB      000H,000H,000H,000H,07EH,07EH,07EH,000H ; D_16
     7E 00
00B8 18 3C 7E 18 7E 3C  54              DB      018H,03CH,07EH,018H,07EH,03CH,018H,0FFH ; D_17
     18 FF
00C0 18 3C 7E 18 18 18  56              DB      018H,03CH,07EH,018H,018H,018H,018H,000H ; D_18
     18 00
00C8 18 18 18 18 7E 3C  58              DB      018H,018H,018H,018H,07EH,03CH,018H,000H ; D_19
     18 00
00D0 00 18 0C FE 0C 18  60              DB      000H,018H,00CH,0FEH,00CH,018H,000H,000H ; D_1A
     00 00
00D8 00 30 60 FE 60 30  62              DB      000H,030H,060H,0FEH,060H,030H,000H,000H ; D_1B
     00 00
00E0 00 00 C0 C0 C0 FE  64              DB      000H,000H,0C0H,0C0H,0C0H,0FEH,000H,000H ; D_1C
     00 00
00E8 00 24 66 FF 66 24  66              DB      000H,024H,066H,0FFH,066H,024H,000H,000H ; D_1D
     00 00
00F0 00 18 3C 7E FF FF  68              DB      000H,018H,03CH,07EH,0FFH,0FFH,000H,000H ; D_1E
     00 00
00F8 00 FF FF 7E 3C 18  70              DB      000H,0FFH,0FFH,07EH,03CH,018H,000H,000H ; D_1F
     00 00
```

```
0100 00 00 00 00 00 00  73              DB      000H,000H,000H,000H,000H,000H,000H,000H ; SP D_20
     00 00
0108 30 78 78 30 30 00  75              DB      030H,078H,078H,030H,030H,000H,030H,000H ; ! D_21
     30 00
0110 6C 6C 6C 00 00 00  77              DB      06CH,06CH,06CH,000H,000H,000H,000H,000H ; " D_22
     00 00
0118 6C 6C FE 6C FE 6C  79              DB      06CH,06CH,0FEH,06CH,0FEH,06CH,06CH,000H ; # D_23
     6C 00
0120 30 7C C0 78 0C F8  81              DB      030H,07CH,0C0H,078H,00CH,0F8H,030H,000H ; $ D_24
     30 00
0128 00 C6 CC 18 30 66  83              DB      000H,0C6H,0CCH,018H,030H,066H,0C6H,000H ; PER CENT D_25
     C6 00
0130 38 6C 38 76 DC CC  85              DB      038H,06CH,038H,076H,0DCH,0CCH,076H,000H ; & D_26
     76 00
0138 60 60 C0 00 00 00  87              DB      060H,060H,0C0H,000H,000H,000H,000H,000H ; ' D_27
     00 00
0140 18 30 60 60 60 30  89              DB      018H,030H,060H,060H,060H,030H,018H,000H ; ( D_28
     18 00
0148 60 30 18 18 18 30  91              DB      060H,030H,018H,018H,018H,030H,060H,000H ; ) D_29
     60 00
0150 00 66 3C FF 3C 66  93              DB      000H,066H,03CH,0FFH,03CH,066H,000H,000H ; * D_2A
     00 00
0158 00 30 30 FC 30 30  95              DB      000H,030H,030H,0FCH,030H,030H,000H,000H ; + D_2B
     00 00
0160 00 00 00 00 00 30  97              DB      000H,000H,000H,000H,000H,030H,030H,060H ; , D_2C
     30 60
0168 00 00 00 FC 00 00  99              DB      000H,000H,000H,0FCH,000H,000H,000H,000H ; - D_2D
     00 00
0170 00 00 00 00 00 30  101             DB      000H,000H,000H,000H,000H,030H,030H,000H ; . D_2E
     30 00
0178 06 0C 18 30 60 C0  103             DB      006H,00CH,018H,030H,060H,0C0H,080H,000H ; / D_2F
     80 00
```

```
0180 7C C6 CE DE F6 E6  106             DB      07CH,0C6H,0CEH,0DEH,0F6H,0E6H,07CH,000H ; 0 D_30
     7C 00
0188 30 70 30 30 30 30  108             DB      030H,070H,030H,030H,030H,030H,0FCH,000H ; 1 D_31
     FC 00
```

```
0190  78 CC 0C 38 60 CC      110                DB      078H,0CCH,00CH,038H,060H,0CCH,0FCH,000H ; 2 D_32
      FC 00                   111
0198  78 CC 0C 38 0C CC      112                DB      078H,0CCH,00CH,038H,00CH,0CCH,078H,000H ; 3 D_33
      78 00                   113
01A0  1C 3C 6C CC FE 0C      114                DB      01CH,03CH,06CH,0CCH,0FEH,00CH,01EH,000H ; 4 D_34
      1E 00                   115
01A8  FC C0 F8 0C 0C CC      116                DB      0FCH,0C0H,0F8H,00CH,00CH,0CCH,078H,000H ; 5 D_35
      78 00                   117
01B0  38 60 C0 F8 CC CC      118                DB      038H,060H,0C0H,0F8H,0CCH,0CCH,078H,000H ; 6 D_36
      78 00                   119
01B8  FC CC 0C 18 30 30      120                DB      0FCH,0CCH,00CH,018H,030H,030H,030H,000H ; 7 D_37
      30 00                   121
01C0  78 CC CC 78 CC CC      122                DB      078H,0CCH,0CCH,078H,0CCH,0CCH,078H,000H ; 8 D_38
      78 00                   123
01C8  78 CC CC 7C 0C 18      124                DB      078H,0CCH,0CCH,07CH,00CH,018H,070H,000H ; 9 D_39
      70 00                   125
01D0  00 30 30 00 00 30      126                DB      000H,030H,030H,000H,000H,030H,030H,000H ; : D_3A
      30 00                   127
01D8  00 30 30 00 00 30      128                DB      000H,030H,030H,000H,000H,030H,030H,060H ; ; D_3B
      30 60                   129
01E0  18 30 60 C0 60 30      130                DB      018H,030H,060H,0C0H,060H,030H,018H,000H ; < D_3C
      18 00                   131
01E8  00 00 FC 00 00 FC      132                DB      000H,000H,0FCH,000H,000H,0FCH,000H,000H ; = D_3D
      00 00                   133
01F0  60 30 18 0C 18 30      134                DB      060H,030H,018H,00CH,018H,030H,060H,000H ; > D_3E
      60 00                   135
01F8  78 CC 0C 18 30 00      136                DB      078H,0CCH,00CH,018H,030H,000H,030H,000H ; ? D_3F
      30 00                   137
                             138
0200  7C C6 DE DE DE C0      139                DB      07CH,0C6H,0DEH,0DEH,0DEH,0C0H,078H,000H ; @ D_40
      78 00                   140
0208  30 78 CC CC FC CC      141                DB      030H,078H,0CCH,0CCH,0FCH,0CCH,0CCH,000H ; A D_41
      CC 00                   142
0210  FC 66 66 7C 66 66      143                DB      0FCH,066H,066H,07CH,066H,066H,0FCH,000H ; B D_42
      FC 00                   144
0218  3C 66 C0 C0 C0 66      145                DB      03CH,066H,0C0H,0C0H,0C0H,066H,03CH,000H ; C D_43
      3C 00                   146
0220  F8 6C 66 66 66 6C      147                DB      0F8H,06CH,066H,066H,066H,06CH,0F8H,000H ; D D_44
      F8 00                   148
0228  FE 62 68 78 68 62      149                DB      0FEH,062H,068H,078H,068H,062H,0FEH,000H ; E D_45
      FE 00                   150
0230  FE 62 68 78 68 60      151                DB      0FEH,062H,068H,078H,068H,060H,0F0H,000H ; F D_46
      F0 00                   152
0238  3C 66 C0 C0 CE 66      153                DB      03CH,066H,0C0H,0C0H,0CEH,066H,03EH,000H ; G D_47
      3E 00                   154
0240  CC CC CC FC CC CC      155                DB      0CCH,0CCH,0CCH,0FCH,0CCH,0CCH,0CCH,000H ; H D_48
      CC 00                   156
0248  78 30 30 30 30 30      157                DB      078H,030H,030H,030H,030H,030H,078H,000H ; I D_49
      78 00                   158
0250  1E 0C 0C 0C CC CC      159                DB      01EH,00CH,00CH,00CH,0CCH,0CCH,078H,000H ; J D_4A
      78 00                   160
0258  E6 66 6C 78 6C 66      161                DB      0E6H,066H,06CH,078H,06CH,066H,0E6H,000H ; K D_4B
      E6 00                   162
0260  F0 60 60 60 62 66      163                DB      0F0H,060H,060H,060H,062H,066H,0FEH,000H ; L D_4C
      FE 00                   164
0268  C6 EE FE FE D6 C6      165                DB      0C6H,0EFH,0FFH,0FFH,0D6H,0C6H,0C6H,000H ; M D_4D
      C6 00                   166
0270  C6 E6 F6 DE CE C6      167                DB      0C6H,0E6H,0F6H,0DEH,0CEH,0C6H,0C6H,000H ; N D_4E
      C6 00                   168
0278  38 6C C6 C6 C6 6C      169                DB      038H,06CH,0C6H,0C6H,0C6H,06CH,038H,000H ; O D_4F
      38 00                   170
                             171
0280  FC 66 66 7C 60 60      172                DB      0FCH,066H,066H,07CH,060H,060H,0F0H,000H ; P D_50
      F0 00                   173
0288  78 CC CC CC DC 78      174                DB      078H,0CCH,0CCH,0CCH,0DCH,078H,01CH,000H ; Q D_51
      1C 00                   175
0290  FC 66 66 7C 6C 66      176                DB      0FCH,066H,066H,07CH,06CH,066H,0E6H,000H ; R D_52
      E6 00                   177
0298  78 CC E0 70 1C CC      178                DB      078H,0CCH,0E0H,070H,01CH,0CCH,078H,000H ; S D_53
      78 00                   179
02A0  FC B4 30 30 30 30      180                DB      0FCH,0B4H,030H,030H,030H,030H,078H,000H ; T D_54
      78 00                   181
02A8  CC CC CC CC CC CC      182                DB      0CCH,0CCH,0CCH,0CCH,0CCH,0CCH,0FCH,000H ; U D_55
      FC 00                   183
02B0  CC CC CC CC CC 78      184                DB      0CCH,0CCH,0CCH,0CCH,0CCH,078H,030H,000H ; V D_56
      30 00                   185
02B8  C6 C6 C6 D6 FE EE      186                DB      0C6H,0C6H,0C6H,0D6H,0FEH,0EEH,0C6H,000H ; W D_57
      C6 00                   187
02C0  C6 C6 6C 38 38 6C      188                DB      0C6H,0C6H,06CH,038H,038H,06CH,0C6H,000H ; X D_58
      C6 00                   189
02C8  CC CC CC 78 30 30      190                DB      0CCH,0CCH,0CCH,078H,030H,030H,078H,000H ; Y D_59
      78 00                   191
02D0  FE C6 8C 18 32 66      192                DB      0FEH,0C6H,08CH,018H,032H,066H,0FEH,000H ; Z D_5A
      FE 00                   193
02D8  78 60 60 60 60 60      194                DB      078H,060H,060H,060H,060H,060H,078H,000H ; [ D_5B
      78 00                   195
02E0  C0 60 30 18 0C 06      196                DB      0C0H,060H,030H,018H,00CH,006H,002H,000H ; BACKSLASH D_5C
      02 00                   197
02E8  78 18 18 18 18 18      198                DB      078H,018H,018H,018H,018H,018H,078H,000H ; ] D_5D
      78 00                   199
02F0  10 38 6C C6 00 00      200                DB      010H,038H,06CH,0C6H,000H,000H,000H,000H ; CIRCUMFLEX D_5E
      00 00                   201
02F8  00 00 00 00 00 00      202                DB      000H,000H,000H,000H,000H,000H,000H,0FFH ; _ D_5F
      00 FF                   203
                             204
0300  30 30 18 00 00 00      205                DB      030H,030H,018H,000H,000H,000H,000H,000H ; ' D_60
      00 00                   206
0308  00 00 78 0C 7C CC      207                DB      000H,000H,078H,00CH,07CH,0CCH,076H,000H ; LOWER CASE A D_61
      76 00                   208
0310  E0 60 60 7C 66 66      209                DB      0E0H,060H,060H,07CH,066H,066H,0DCH,000H ; L.C. B D_62
      DC 00                   210
0318  00 00 78 CC C0 CC      211                DB      000H,000H,078H,0CCH,0C0H,0CCH,078H,000H ; L.C. C D_63
      78 00                   212
0320  1C 0C 0C 7C CC CC      213                DB      01CH,00CH,00CH,07CH,0CCH,0CCH,076H,000H ; L.C. D D_64
      76 00                   214
0328  00 00 78 CC FC C0      215                DB      000H,000H,078H,0CCH,0FCH,0C0H,078H,000H ; L.C. E D_65
      78 00                   216
0330  38 6C 60 F0 60 60      217                DB      038H,06CH,060H,0F0H,060H,060H,0F0H,000H ; L.C. F D_66
      F0 00                   218
0338  00 00 76 CC CC 7C      219                DB      000H,000H,076H,0CCH,0CCH,07CH,00CH,0F8H ; L.C. G D_67
      0C F8                   220
0340  E0 60 6C 76 66 66      221                DB      0E0H,060H,06CH,076H,066H,066H,0E6H,000H ; L.C. H D_68
      E6 00                   222
0348  30 00 70 30 30 30      223                DB      030H,000H,070H,030H,030H,030H,078H,000H ; L.C. I D_69
      78 00                   224
0350  0C 00 0C 0C 0C CC      225                DB      00CH,000H,00CH,00CH,00CH,0CCH,0CCH,078H ; L.C. J D_6A
      CC 78                   226
0358  E0 60 66 6C 78 6C      227                DB      0E0H,060H,066H,06CH,078H,06CH,0E6H,000H ; L.C. K D_6B
      E6 00                   228
0360  70 30 30 30 30 30      229                DB      070H,030H,030H,030H,030H,030H,078H,000H ; L.C. L D_6C
      78 00                   230
0368  00 00 CC FE FE D6      231                DB      000H,000H,0CCH,0FEH,0FEH,0D6H,0C6H,000H ; L.C. M D_6D
      C6 00                   232
0370  00 00 F8 CC CC CC      233                DB      000H,000H,0F8H,0CCH,0CCH,0CCH,0CCH,000H ; L.C. N D_6E
      CC 00                   234
0378  00 00 78 CC CC CC      235                DB      000H,000H,078H,0CCH,0CCH,0CCH,078H,000H ; L.C. O D_6F
```

```
                78 00                                   236
                                                        237
0380  00 00 DC 66 66 7C                                238              DB      000H,000H,0DCH,066H,066H,07CH,060H,0F0H ; L.C. P D_70
      60 F0                                             239
0388  00 00 76 CC CC 7C                                240              DB      000H,000H,076H,0CCH,0CCH,07CH,00CH,01EH ; L.C. Q D_71
      0C 1E                                             241
0390  00 00 DC 76 66 60                                242              DB      000H,000H,0DCH,076H,066H,060H,0F0H,000H ; L.C. R D_72
      F0 00                                             243
0398  00 00 7C C0 78 0C                                244              DB      000H,000H,07CH,0C0H,078H,00CH,0F8H,000H ; L.C. S D_73
      F8 00                                             245
03A0  10 30 7C 30 30 34                                246              DB      010H,030H,07CH,030H,030H,034H,018H,000H ; L.C. T D_74
      18 00                                             247
03A8  00 00 CC CC CC CC                                248              DB      000H,000H,0CCH,0CCH,0CCH,0CCH,076H,000H ; L.C. U D_75
      76 00                                             249
03B0  00 00 CC CC CC 78                                250              DB      000H,000H,0CCH,0CCH,0CCH,078H,030H,000H ; L.C. V D_76
      30 00                                             251
03B8  00 00 C6 D6 FE FE                                252              DB      000H,000H,0C6H,0D6H,0FEH,0FEH,06CH,000H ; L.C. W D_77
      6C 00                                             253
03C0  00 00 C6 6C 38 6C                                254              DB      000H,000H,0C6H,06CH,038H,06CH,0C6H,000H ; L.C. X D_78
      C6 00                                             255
03C8  00 00 CC CC CC 7C                                256              DB      000H,000H,0CCH,0CCH,0CCH,07CH,00CH,0F8H ; L.C. Y D_79
      0C F8                                             257
03D0  00 00 FC 98 30 64                                258              DB      000H,000H,0FCH,098H,030H,064H,0FCH,000H ; L.C. Z D_7A
      FC 00                                             259
03D8  1C 30 30 E0 30 30                                260              DB      01CH,030H,030H,0E0H,030H,030H,01CH,000H ; L BRAK D_7B
      1C 00                                             261
03E0  18 18 18 00 18 18                                262              DB      018H,018H,018H,000H,018H,018H,018H,000H ; | D_7C
      18 00                                             263
03E8  E0 30 30 1C 30 30                                264              DB      0E0H,030H,030H,01CH,030H,030H,0E0H,000H ; R BRAK D_7D
      E0 00                                             265
03F0  76 DC 00 00 00 00                                266              DB      076H,0DCH,000H,000H,000H,000H,000H,000H ; TILDE D_7E
      00 00                                             267
03F8  00 10 38 6C C6 C6                                268              DB      000H,010H,038H,06CH,0C6H,0C6H,0FEH,000H ; DELTA D_7F
      FE 00                                             269
                                                        270
0400                                                    271  INT_1F_1    LABEL   BYTE
                                                        272
0400  78 CC C0 CC 78 18                                273              DB      078H,0CCH,0C0H,0CCH,078H,018H,00CH,078H ;  D_80
      0C 78                                             274
0408  00 CC 00 CC CC CC                                275              DB      000H,0CCH,000H,0CCH,0CCH,0CCH,07EH,000H ;  D_81
      7E 00                                             276
0410  1C 00 78 CC FC C0                                277              DB      01CH,000H,078H,0CCH,0FCH,0C0H,078H,000H ;  D_82
      78 00                                             278
0418  7E C3 3C 06 3E 66                                279              DB      07EH,0C3H,03CH,006H,03EH,066H,03FH,000H ;  D_83
      3F 00                                             280
0420  CC 00 78 0C 7C CC                                281              DB      0CCH,000H,078H,00CH,07CH,0CCH,07EH,000H ;  D_84
      7E 00                                             282
0428  E0 00 78 0C 7C CC                                283              DB      0E0H,000H,078H,00CH,07CH,0CCH,07EH,000H ;  D_85
      7E 00                                             284
0430  30 30 78 0C 7C CC                                285              DB      030H,030H,078H,00CH,07CH,0CCH,07EH,000H ;  D_86
      7E 00                                             286
0438  00 00 78 C0 C0 78                                287              DB      000H,000H,078H,0C0H,0C0H,078H,00CH,038H ;  D_87
      0C 38                                             288
0440  7E C3 3C 66 7E 60                                289              DB      07EH,0C3H,03CH,066H,07EH,060H,03CH,000H ;  D_88
      3C 00                                             290
0448  CC 00 78 CC FC C0                                291              DB      0CCH,000H,078H,0CCH,0FCH,0C0H,078H,000H ;  D_89
      78 00                                             292
0450  E0 00 78 CC FC C0                                293              DB      0E0H,000H,078H,0CCH,0FCH,0C0H,078H,000H ;  D_8A
      78 00                                             294
0458  CC 00 70 30 30 30                                295              DB      0CCH,000H,070H,030H,030H,030H,078H,000H ;  D_8B
      78 00                                             296
0460  7C C6 38 18 18 18                                297              DB      07CH,0C6H,038H,018H,018H,018H,03CH,000H ;  D_8C
      3C 00                                             298
0468  E0 00 70 30 30 30                                299              DB      0E0H,000H,070H,030H,030H,030H,078H,000H ;  D_8D
      78 00                                             300
0470  C6 38 6C C6 FE C6                                301              DB      0C6H,038H,06CH,0C6H,0FEH,0C6H,0C6H,000H ;  D_8E
      C6 00                                             302
0478  30 30 00 78 CC FC                                303              DB      030H,030H,000H,078H,0CCH,0FCH,0CCH,000H ;  D_8F
      CC 00                                             304
                                                        305
0480  1C 00 FC 60 78 60                                306              DB      01CH,000H,0FCH,060H,078H,060H,0FCH,000H ;  D_90
      FC 00                                             307
0488  00 00 7F 0C 7F CC                                308              DB      000H,000H,07FH,00CH,07FH,0CCH,07FH,000H ;  D_91
      7F 00                                             309
0490  3E 6C CC FE CC CC                                310              DB      03EH,06CH,0CCH,0FEH,0CCH,0CCH,0CEH,000H ;  D_92
      CE 00                                             311
0498  78 CC 00 78 CC CC                                312              DB      078H,0CCH,000H,078H,0CCH,0CCH,078H,000H ;  D_93
      78 00                                             313
04A0  00 CC 00 78 CC CC                                314              DB      000H,0CCH,000H,078H,0CCH,0CCH,078H,000H ;  D_94
      78 00                                             315
04A8  00 E0 00 78 CC CC                                316              DB      000H,0E0H,000H,078H,0CCH,0CCH,078H,000H ;  D_95
      78 00                                             317
04B0  78 CC 00 CC CC CC                                318              DB      078H,0CCH,000H,0CCH,0CCH,0CCH,07EH,000H ;  D_96
      7E 00                                             319
04B8  00 E0 00 CC CC CC                                320              DB      000H,0E0H,000H,0CCH,0CCH,0CCH,07EH,000H ;  D_97
      7E 00                                             321
04C0  00 CC 00 CC CC 7C                                322              DB      000H,0CCH,000H,0CCH,0CCH,07CH,00CH,0F8H ;  D_98
      0C F8                                             323
04C8  C3 18 3C 66 66 3C                                324              DB      0C3H,018H,03CH,066H,066H,03CH,018H,000H ;  D_99
      18 00                                             325
04D0  CC 00 CC CC CC CC                                326              DB      0CCH,000H,0CCH,0CCH,0CCH,0CCH,078H,000H ;  D_9A
      78 00                                             327
04D8  18 18 7E C0 C0 7E                                328              DB      018H,018H,07EH,0C0H,0C0H,07EH,018H,018H ;  D_9B
      18 18                                             329
04E0  38 6C 64 F0 60 E6                                330              DB      038H,06CH,064H,0F0H,060H,0E6H,0FCH,000H ;  D_9C
      FC 00                                             331
04E8  CC CC 78 FC 30 FC                                332              DB      0CCH,0CCH,078H,0FCH,030H,0FCH,030H,030H ;  D_9D
      30 30                                             333
04F0  F8 CC CC FA C6 CF                                334              DB      0F8H,0CCH,0CCH,0FAH,0C6H,0CFH,0C6H,0C7H ;  D_9E
      C6 C7                                             335
04F8  0E 1B 18 3C 18 18                                336              DB      00EH,01BH,018H,03CH,018H,018H,0D8H,070H ;  D_9F
      D8 70                                             337
                                                        338
0500  1C 00 78 0C 7C CC                                339              DB      01CH,000H,078H,00CH,07CH,0CCH,07EH,000H ;  D_A0
      7E 00                                             340
0508  38 00 70 30 30 30                                341              DB      038H,000H,070H,030H,030H,030H,078H,000H ;  D_A1
      78 00                                             342
0510  00 1C 00 78 CC CC                                343              DB      000H,01CH,000H,078H,0CCH,0CCH,078H,000H ;  D_A2
      78 00                                             344
0518  00 1C 00 CC CC CC                                345              DB      000H,01CH,000H,0CCH,0CCH,0CCH,07EH,000H ;  D_A3
      7E 00                                             346
0520  00 F8 00 F8 CC CC                                347              DB      000H,0F8H,000H,0F8H,0CCH,0CCH,0CCH,000H ;  D_A4
      CC 00                                             348
0528  FC 00 CC EC FC DC                                349              DB      0FCH,000H,0CCH,0ECH,0FCH,0DCH,0CCH,000H ;  D_A5
      CC 00                                             350
0530  3C 6C 6C 3E 00 7E                                351              DB      03CH,06CH,06CH,03EH,000H,07EH,000H,000H ;  D_A6
      00 00                                             352
0538  38 6C 6C 38 00 7C                                353              DB      038H,06CH,06CH,038H,000H,07CH,000H,000H ;  D_A7
      00 00                                             354
0540  30 00 30 60 C0 CC                                355              DB      030H,000H,030H,060H,0C0H,0CCH,078H,000H ;  D_A8
      78 00                                             356
0548  00 00 00 FC C0 C0                                357              DB      000H,000H,000H,0FCH,0C0H,0C0H,000H,000H ;  D_A9
      00 00                                             358
0550  00 00 00 FC 0C 0C                                359              DB      000H,000H,000H,0FCH,00CH,00CH,000H,000H ;  D_AA
      00 00                                             360
0558  C3 C6 CC DE 33 66                                361              DB      0C3H,0C6H,0CCH,0DEH,033H,066H,0CCH,00FH ;  D_AB
```

```
                        CC OF                                   362
0560  C3 C6 CC DB 37 6F CF 03                                   363                     DB      0C3H,0C6H,0CCH,0DBH,037H,06FH,0CFH,003H ;  D_AC
                                                                364
0568  18 18 00 18 18 18 18 00                                   365                     DB      018H,018H,000H,018H,018H,018H,018H,000H ;  D_AD
                                                                366
0570  00 33 66 CC 66 33 00 00                                   367                     DB      000H,033H,066H,0CCH,066H,033H,000H,000H ;  D_AE
                                                                368
0578  00 CC 66 33 66 CC 00 00                                   369                     DB      000H,0CCH,066H,033H,066H,0CCH,000H,000H ;  D_AF
                                                                370
                                                                371
0580  22 88 22 88 22 88 22 88                                   372                     DB      022H,088H,022H,088H,022H,088H,022H,088H ;  D_B0
                                                                373
0588  55 AA 55 AA 55 AA 55 AA                                   374                     DB      055H,0AAH,055H,0AAH,055H,0AAH,055H,0AAH ;  D_B1
                                                                375
0590  DB 77 DB EE DB 77 DB EE                                   376                     DB      0DBH,077H,0DBH,0EEH,0DBH,077H,0DBH,0EEH ;  D_B2
                                                                377
0598  18 18 18 18 18 18 18 18                                   378                     DB      018H,018H,018H,018H,018H,018H,018H,018H ;  D_B3
                                                                379
05A0  18 18 18 18 F8 18 18 18                                   380                     DB      018H,018H,018H,018H,0F8H,018H,018H,018H ;  D_B4
                                                                381
05A8  18 18 F8 18 F8 18 18 18                                   382                     DB      018H,018H,0F8H,018H,0F8H,018H,018H,018H ;  D_B5
                                                                383
05B0  36 36 36 36 F6 36 36 36                                   384                     DB      036H,036H,036H,036H,0F6H,036H,036H,036H ;  D_B6
                                                                385
05B8  00 00 00 00 FE 36 36 36                                   386                     DB      000H,000H,000H,000H,0FEH,036H,036H,036H ;  D_B7
                                                                387
05C0  00 00 F8 18 F8 18 18 18                                   388                     DB      000H,000H,0F8H,018H,0F8H,018H,018H,018H ;  D_B8
                                                                389
05C8  36 36 F6 06 F6 36 36 36                                   390                     DB      036H,036H,0F6H,006H,0F6H,036H,036H,036H ;  D_B9
                                                                391
05D0  36 36 36 36 36 36 36 36                                   392                     DB      036H,036H,036H,036H,036H,036H,036H,036H ;  D_BA
                                                                393
05D8  00 00 FE 06 F6 36 36 36                                   394                     DB      000H,000H,0FEH,006H,0F6H,036H,036H,036H ;  D_BB
                                                                395
05E0  36 36 F6 06 FE 00 00 00                                   396                     DB      036H,036H,0F6H,006H,0FEH,000H,000H,000H ;  D_BC
                                                                397
05E8  36 36 36 36 FE 00 00 00                                   398                     DB      036H,036H,036H,036H,0FEH,000H,000H,000H ;  D_BD
                                                                399
05F0  18 18 F8 18 F8 00 00 00                                   400                     DB      018H,018H,0F8H,018H,0F8H,000H,000H,000H ;  D_BE
                                                                401
05F8  00 00 00 00 F8 18 18 18                                   402                     DB      000H,000H,000H,000H,0F8H,018H,018H,018H ;  D_BF
                                                                403
                                                                404
0600  18 18 18 18 1F 00 00 00                                   405                     DB      018H,018H,018H,018H,01FH,000H,000H,000H ;  D_C0
                                                                406
0608  18 18 18 18 FF 00 00 00                                   407                     DB      018H,018H,018H,018H,0FFH,000H,000H,000H ;  D_C1
                                                                408
0610  00 00 00 00 FF 18 18 18                                   409                     DB      000H,000H,000H,000H,0FFH,018H,018H,018H ;  D_C2
                                                                410
0618  18 18 18 18 1F 18 18 18                                   411                     DB      018H,018H,018H,018H,01FH,018H,018H,018H ;  D_C3
                                                                412
0620  00 00 00 00 FF 00 00 00                                   413                     DB      000H,000H,000H,000H,0FFH,000H,000H,000H ;  D_C4
                                                                414
0628  18 18 18 18 FF 18 18 18                                   415                     DB      018H,018H,018H,018H,0FFH,018H,018H,018H ;  D_C5
                                                                416
0630  18 18 1F 18 1F 18 18 18                                   417                     DB      018H,018H,01FH,018H,01FH,018H,018H,018H ;  D_C6
                                                                418
0638  36 36 36 36 37 36 36 36                                   419                     DB      036H,036H,036H,036H,037H,036H,036H,036H ;  D_C7
                                                                420
0640  36 36 37 30 3F 00 00 00                                   421                     DB      036H,036H,037H,030H,03FH,000H,000H,000H ;  D_C8
                                                                422
0648  00 00 3F 30 37 36 36 36                                   423                     DB      000H,000H,03FH,030H,037H,036H,036H,036H ;  D_C9
                                                                424
0650  36 36 F7 00 FF 00 00 00                                   425                     DB      036H,036H,0F7H,000H,0FFH,000H,000H,000H ;  D_CA
                                                                426
0658  00 00 FF 00 F7 36 36 36                                   427                     DB      000H,000H,0FFH,000H,0F7H,036H,036H,036H ;  D_CB
                                                                428
0660  36 36 37 30 37 36 36 36                                   429                     DB      036H,036H,037H,030H,037H,036H,036H,036H ;  D_CC
                                                                430
0668  00 00 FF 00 FF 00 00 00                                   431                     DB      000H,000H,0FFH,000H,0FFH,000H,000H,000H ;  D_CD
                                                                432
0670  36 36 F7 00 F7 36 36 36                                   433                     DB      036H,036H,0F7H,000H,0F7H,036H,036H,036H ;  D_CE
                                                                434
0678  18 18 FF 00 FF 00 00 00                                   435                     DB      018H,018H,0FFH,000H,0FFH,000H,000H,000H ;  D_CF
                                                                436
                                                                437
0680  36 36 36 36 FF 00 00 00                                   438                     DB      036H,036H,036H,036H,0FFH,000H,000H,000H ;  D_D0
                                                                439
0688  00 00 FF 00 FF 18 18 18                                   440                     DB      000H,000H,0FFH,000H,0FFH,018H,018H,018H ;  D_D1
                                                                441
0690  00 00 00 00 FF 36 36 36                                   442                     DB      000H,000H,000H,000H,0FFH,036H,036H,036H ;  D_D2
                                                                443
0698  36 36 36 36 3F 00 00 00                                   444                     DB      036H,036H,036H,036H,03FH,000H,000H,000H ;  D_D3
                                                                445
06A0  18 18 1F 18 1F 00 00 00                                   446                     DB      018H,018H,01FH,018H,01FH,000H,000H,000H ;  D_D4
                                                                447
06A8  00 00 1F 18 1F 18 18 18                                   448                     DB      000H,000H,01FH,018H,01FH,018H,018H,018H ;  D_D5
                                                                449
06B0  00 00 00 00 3F 36 36 36                                   450                     DB      000H,000H,000H,000H,03FH,036H,036H,036H ;  D_D6
                                                                451
06B8  36 36 36 36 FF 36 36 36                                   452                     DB      036H,036H,036H,036H,0FFH,036H,036H,036H ;  D_D7
                                                                453
06C0  18 18 FF 18 FF 18 18 18                                   454                     DB      018H,018H,0FFH,018H,0FFH,018H,018H,018H ;  D_D8
                                                                455
06C8  18 18 18 18 F8 00 00 00                                   456                     DB      018H,018H,018H,018H,0F8H,000H,000H,000H ;  D_D9
                                                                457
06D0  00 00 00 00 1F 18 18 18                                   458                     DB      000H,000H,000H,000H,01FH,018H,018H,018H ;  D_DA
                                                                459
06D8  FF FF FF FF FF FF FF FF                                   460                     DB      0FFH,0FFH,0FFH,0FFH,0FFH,0FFH,0FFH,0FFH ;  D_DB
                                                                461
06E0  00 00 00 00 FF FF FF FF                                   462                     DB      000H,000H,000H,000H,0FFH,0FFH,0FFH,0FFH ;  D_DC
                                                                463
06E8  F0 F0 F0 F0 F0 F0 F0 F0                                   464                     DB      0F0H,0F0H,0F0H,0F0H,0F0H,0F0H,0F0H,0F0H ;  D_DD
                                                                465
06F0  0F 0F 0F 0F 0F 0F 0F 0F                                   466                     DB      00FH,00FH,00FH,00FH,00FH,00FH,00FH,00FH ;  D_DE
                                                                467
06F8  FF FF FF FF 00 00 00 00                                   468                     DB      0FFH,0FFH,0FFH,0FFH,000H,000H,000H,000H ;  D_DF
                                                                469
                                                                470
0700  00 00 76 DC C8 DC 76 00                                   471                     DB      000H,000H,076H,0DCH,0C8H,0DCH,076H,000H ;  D_E0
                                                                472
0708  00 78 CC F8 CC F8 C0 C0                                   473                     DB      000H,078H,0CCH,0F8H,0CCH,0F8H,0C0H,0C0H ;  D_E1
                                                                474
0710  00 FC CC C0 C0 C0 C0 00                                   475                     DB      000H,0FCH,0CCH,0C0H,0C0H,0C0H,0C0H,000H ;  D_E2
                                                                476
0718  00 FE 6C 6C 6C 6C 6C 00                                   477                     DB      000H,0FEH,06CH,06CH,06CH,06CH,06CH,000H ;  D_E3
                                                                478
0720  FC CC 60 30 60 CC FC 00                                   479                     DB      0FCH,0CCH,060H,030H,060H,0CCH,0FCH,000H ;  D_E4
                                                                480
0728  00 00 7E D8 D8 D8 70 00                                   481                     DB      000H,000H,07EH,0D8H,0D8H,0D8H,070H,000H ;  D_E5
                                                                482
0730  00 66 66 66 66 7C 60 C0                                   483                     DB      000H,066H,066H,066H,066H,07CH,060H,0C0H ;  D_E6
                                                                484
0738  00 76 DC 18 18 18 18 00                                   485                     DB      000H,076H,0DCH,018H,018H,018H,018H,000H ;  D_E7
                                                                486
0740  FC 30 78 CC CC 78 00 00                                   487                     DB      0FCH,030H,078H,0CCH,0CCH,078H,030H,0FCH ;  D_E8
```

```
                30 FC
0748  38 6C C6 FE C6 6C
                38 00
0750  38 6C C6 C6 6C 6C
                EE 00
0758  1C 30 18 7C CC CC
                78 00
0760  00 00 7E DB DB 7E
                00 00
0768  06 0C 7E DB DB 7E
                60 C0
0770  38 60 C0 F8 C0 60
                38 00
0778  78 CC CC CC CC CC
                CC 00
0780  00 FC 00 FC 00 FC
                00 00
0788  30 30 FC 30 30 00
                FC 00
0790  60 30 18 30 60 00
                FC 00
0798  18 30 60 30 18 00
                FC 00
07A0  0E 1B 1B 18 18 18
                18 18
07A8  18 18 18 18 18 D8
                D8 70
07B0  30 30 00 FC 00 30
                30 00
07B8  00 76 DC 00 76 DC
                00 00
07C0  38 6C 6C 38 00 00
                00 00
07C8  00 00 00 18 18 00
                00 00
07D0  00 00 00 00 18 00
                00 00
07D8  0F 0C 0C 0C EC 6C
                3C 1C
07E0  78 6C 6C 6C 6C 00
                00 00
07E8  70 18 30 60 78 00
                00 00
07F0  00 00 3C 3C 3C 3C
                00 00
07F8  00 00 00 00 00 00
                00 00

0800

0000

0000
0000
```

```
488
489                DB      038H,06CH,0C6H,0FEH,0C6H,06CH,038H,000H ;  D_E9
490
491                DB      038H,06CH,0C6H,0C6H,06CH,06CH,0EEH,000H ;  D_EA
492
493                DB      01CH,030H,018H,07CH,0CCH,0CCH,078H,000H ;  D_EB
494
495                DB      000H,000H,07EH,0DBH,0DBH,07EH,000H,000H ;  D_EC
496
497                DB      006H,00CH,07EH,0DBH,0DBH,07EH,060H,0C0H ;  D_ED
498
499                DB      038H,060H,0C0H,0F8H,0C0H,060H,038H,000H ;  D_EE
500
501                DB      078H,0CCH,0CCH,0CCH,0CCH,0CCH,0CCH,000H ;  D_EF
502
503
504                DB      000H,0FCH,000H,0FCH,000H,0FCH,000H,000H ;  D_F0
505
506                DB      030H,030H,0FCH,030H,030H,000H,0FCH,000H ;  D_F1
507
508                DB      060H,030H,018H,030H,060H,000H,0FCH,000H ;  D_F2
509
510                DB      018H,030H,060H,030H,018H,000H,0FCH,000H ;  D_F3
511
512                DB      00EH,01BH,01BH,018H,018H,018H,018H,018H ;  D_F4
513
514                DB      018H,018H,018H,018H,018H,0D8H,0D8H,070H ;  D_F5
515
516                DB      030H,030H,000H,0FCH,000H,030H,030H,000H ;  D_F6
517
518                DB      000H,076H,0DCH,000H,076H,0DCH,000H,000H ;  D_F7
519
520                DB      038H,06CH,06CH,038H,000H,000H,000H,000H ;  D_F8
521
522                DB      000H,000H,000H,018H,018H,000H,000H,000H ;  D_F9
523
524                DB      000H,000H,000H,000H,018H,000H,000H,000H ;  D_FA
525
526                DB      00FH,00CH,00CH,00CH,0ECH,06CH,03CH,01CH ;  D_FB
527
528                DB      078H,06CH,06CH,06CH,06CH,000H,000H,000H ;  D_FC
529
530                DB      070H,018H,030H,060H,078H,000H,000H,000H ;  D_FD
531
532                DB      000H,000H,03CH,03CH,03CH,03CH,000H,000H ;  D_FE
533
534                DB      000H,000H,000H,000H,000H,000H,000H,000H ;  D_FF
535
536  CODE          ENDS
537                END
1
2                 PAGE,120
                  SUBTTL  END ADDRESS
3    CODE         SEGMENT PUBLIC
4    PUBLIC       END_ADDRESS
5    END_ADDRESS  LABEL   BYTE
6    CODE         ENDS
7                 END
```

Index

F

G

H

I

L

M

O

P

R

S

U

V

Personal Computer
Hardware Reference
Library

Extended Monochrome Graphics Adapter

Contents

Description

Introduction

The Extended Monochrome Graphics Display Adapter attaches to the system I/O channel and drives a 60 Hz noninterlaced monochrome monitor (1024 × 768 picture elements-pels). The adapter has a graphic operation processor which provides bit manipulation facilities that enhance the management and presentation of text, graphics, and image information.

Note: Programming and addressing considerations preclude the use of an Extended Monochrome Graphics Display Adapter in PC-AT systems.

Advanced features include:

- A fast graphic operation processor which can do bit block transfers, line draw, image copy/merge, and rotate
- Graphic operation command queue with synchronization and branch control for animation
- Hardware controlled cursor which provides instant cursor movement with no obstruction to graphic operations
- An I/O interface which supports direct bit addressability in both horizontal and vertical (orthogonal) access directions
- DMA as an alternate controller for image and character transfers from system memory.

Operation

The Extended Monochrome Graphics Display Adapter is characterized by the following key functional elements.

BAMDA

The display buffer (bit map = 1024 × 1024) is a Bit Addressable Multidimensional Array (BAMDA) which stores the picture elements (pels) for refreshing the monitor. The buffer is arranged so that both horizontal and vertical accesses to the array are accomplished on a bit addressable X , Y field. In addition, the bit length for write operations is variable (from 1 bit up to 16 bits) in either the X or Y direction. Also logical operators can be specified on write operations such as 'XOR', 'AND' and 'OR'.

Graphic Operation Processor

A graphic operation processor is used to:

- Expedite transfers of arbitrary data block sizes from one location to another Bit wise Block Transfer (Bit BLT) and in addition, perform Boolean operations on the data as it is being moved
- Draw geometric objects in the display buffer
- Transfer image information stored in I/O channel memory.

Asynchronous Graphic Operation Queueing

A graphic operation queue mechanism is implemented in hardware and stored in the hidden (nondisplayed) area of the bit map. Graphic operation commands are asynchronously linked to one side of the queue list. The list acts as a first in – first out (FIFO) queue. The graphic operation processor processes the commands one-by-one from the other side of the queue. Interrupts, branching, and synchronization are additional control features of the command queue.

Hardware Cursor

A 48 × 64 bit hardware cursor is combined simultaneously with the video output for each displayed frame. There are 2 patterns which make up the cursor; the first is 'ANDed' with the image whereas the second is 'XORed' with the image. These two images, when combined with the video, allow for a programmable cursor pattern to appear on the screen in the cursor position.

An X,Y pair of on-board registers enables cursor positioning at an X,Y pel location through program control.

Extended Monochrome Graphics Display Adapter Diagram

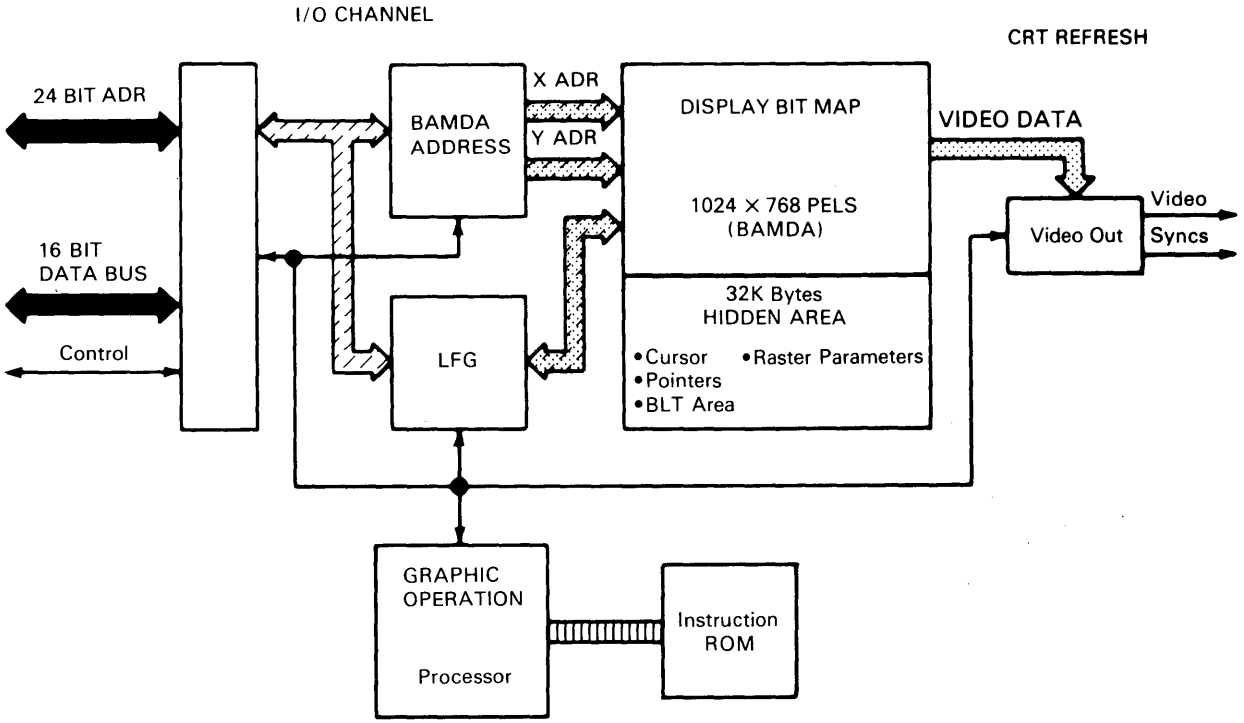


Figure 1. Extended Monochrome Graphics Display Adapter Block Diagram

The Extended Monochrome Graphics Display Adapter consists of 7 major components:

1. The I/O channel interface logic which communicates with the I/O channel and allows the system to control the display functions.
2. The BAMDA address logic which controls access to the bit map and converts system addresses to bit map X and Y coordinates.
3. The BAMDA bit map memory which is divided into a display bit map and a hidden area memory.

4. The LFG (Logic Function Generator) which performs data conversion functions on the data stream to and from BAMDA.
5. The video data out logic which transfers the display bit map data image from BAMDA to the video output connectors and provides video synchronization signals for the monitor.
6. The graphic operation processor which controls all of the adapter functions.
7. The instruction ROM (Read Only Memory) which contains the microcode instructions for the graphic operation processor.

Command Queue

The command queue is a list of graphic operation commands executed by the graphic operation processor. The command queue is loaded by the system processor, usually in a linked list. The processor retrieves these commands and executes them asynchronously from the system.

In its simplest form, the linked list of commands forms a FIFO buffer, with the system loading commands at one end and the graphic processor executing them at the other end.

Additional features of the command and queue formats allow for branching and interrupting, thus considerably increasing control flexibility over the simple FIFO approach.

Direct Memory Access (DMA)

The adapter's hardware supports DMA on system DMA channel 7. The DMA graphic operations are performed during transfers from system memory. Two-dimensional DMA (TDDMA) cut and paste operations are supported as graphic operations types.

Hardware Cursor

The adapter supports a 48x64 bit hardware cursor which 'AND's and 'XOR's two different patterns simultaneously to the screen. The cursor patterns are stored in the hidden portion of the bit map. Memory writes to the appropriate cursor X and Y positioning registers moves the cursor display position.

Programming Interface

This section describes the programming interface and the key architectural features of the Extended Monochrome Graphics Display Adapter.

Memory Map

The adapter provides the system processor with direct access to a 1M-pel bit map. A portion of this bit map is logically dual-ported to the monitor as a video frame buffer.

Bit Addressability

Hardware assist allows the system to address down to the bit level. Bit mask logic allows selective writing of from 1 to 16 bits automatically. Logical functions such as 'AND', 'OR' or 'XOR' are also specified during the writing of selected bits. Bit alignment, masking and rotation (barrel shifting) are all performed by hardware within the adapter, thus freeing the system processor and increasing I/O channel availability.

Orthogonal Access

A unique memory organization and support logic, called Bit Addressable Multidimensional Array (BAMBA), implements the X, Y bit map. BAMBA supports a mode which directly addresses 1 to 16 bits in the vertical and the conventional horizontal direction. The vertical mode also supports bit addressability. All masking and rotation are performed by hardware, thus allowing high speed graphic operations.

Programming Model

The Extended Monochrome Graphics Display Adapter appears as two separate memory maps to the system:

- A memory map
- A I/O Map.

Memory Map

A 128K-block of memory is addressable as a part of the I/O channel memory map. A contiguous 96K-byte segment of this block is used as the adapter video frame buffer. The remainder is hidden memory in the sense that it is not displayable on the monitor. It is, however, otherwise indistinguishable from the video frame buffer at the programming interface level.

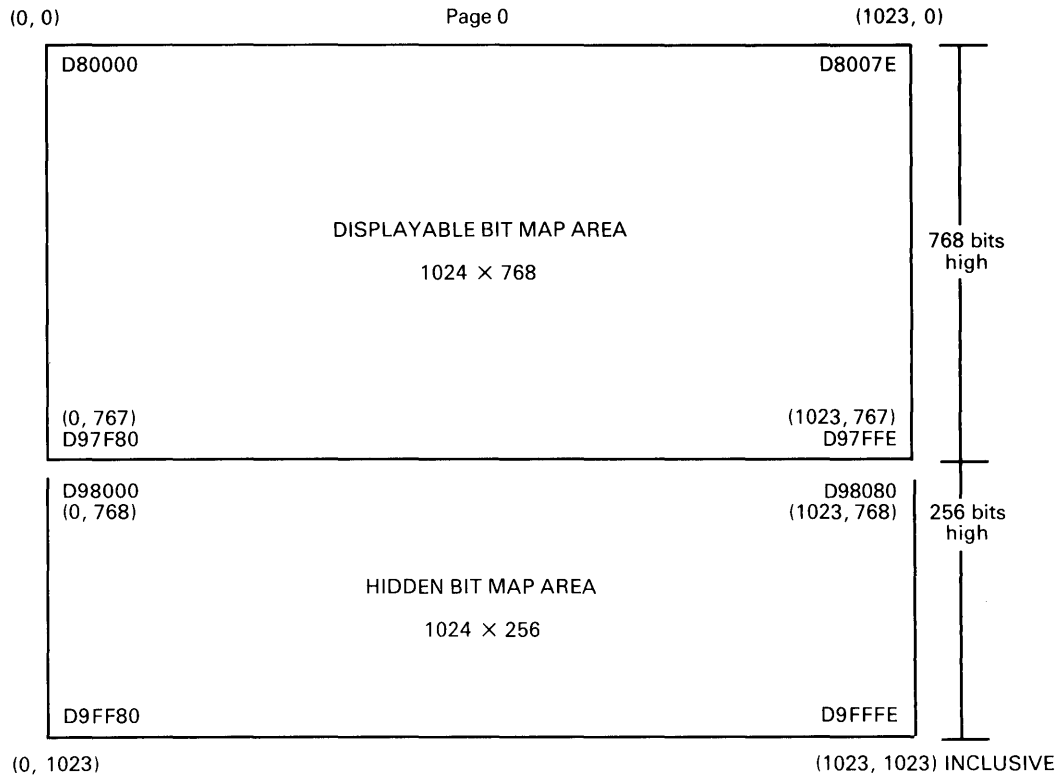


Figure 2. 1M-Pel Bit Map Frame Buffer

I/O Map

A small amount of read/write memory (20 bytes) is implemented in the I/O space as control registers. The I/O registers contain various control and status bits that allow the system to specify a variety of adapter operating modes.

While all memory implemented in the memory map can be written or read by the system, some I/O registers and bits within those registers are read only or write only. Read-only registers (or bits) can be set or reset by the adapter, but can only be interrogated for status by the system. Other bits can be set or reset by the system, but cannot be directly read.

Addressing

The Extended Monochrome Graphics Display Adapter attaches to the system I/O channel as a 16-bit device. Therefore, the least significant bit (LSB) of the address field is assumed to be 0 and is ignored by the adapter logic. When the adapter accesses (via DMA) system memory, it always aligns on even byte addresses (LSB is always driven zero).

The adapter is accessed via two separate address ranges. A 128K-byte range of I/O channel memory addresses, beginning at X'D80000', implements the bit map and is accessed via I/O channel memory read/write operations. This area is called 'memory mapped' memory. Ten control registers are implemented within a 64-byte space called 'I/O mapped memory'. These locations are accessed via I/O channel read/write operations.

Memory Mapped Memory Addressing

The high order seven bits of the system address select the 128K-byte region of the I/O channel memory map assigned to the adapter. This assigned area is X'D80000' to X'D9FFFF'. The remaining 16 bits directly address words on even byte boundaries (the LSB address bit is always zero).

Note: The bit naming notation used in the Extended Monochrome Graphics Display Adapter documentation follows the IBM convention of using bit 0 as the most significant bit. This convention is used for both data and addresses.

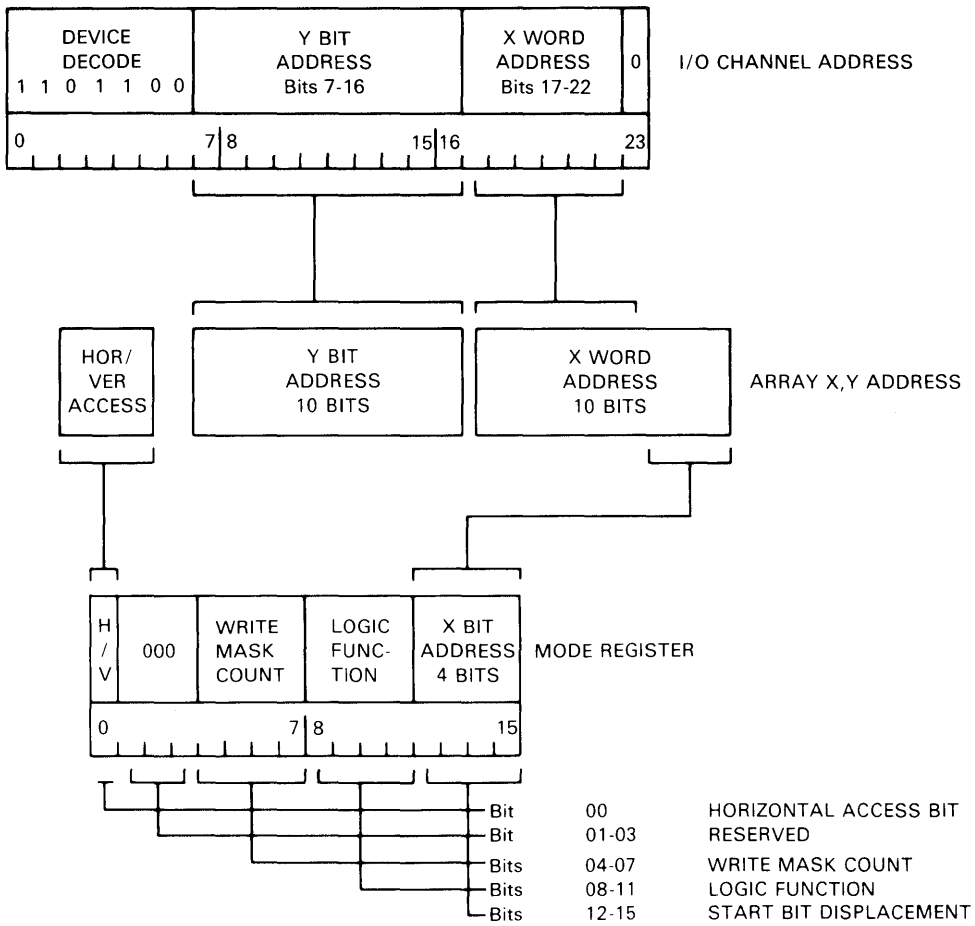


Figure 3. I/O Channel DMA

Access to the bit map is controlled by specifying an even byte address via the I/O channel address and the mode register contents. The mode register bits are set first via an I/O write operation.

Word And Bit Addressability

There are 32 modes (16 horizontal modes and 16 vertical modes) of addressing data within the bit map. One horizontal mode is on a conventional word (16 bit) boundary. The conventional method accesses 16 bits beginning with bit 0 at the left to bit 15 on the right. The other 15 horizontal modes are arbitrarily bit aligned. Thus, the access overlaps into the next word as required to reach a full 16 bit access. The addressing mode is determined by the contents of the start bit displacement field in the mode register. These bits are set via an I/O write to the mode register (see "Mode Register" on page 24). If this 4-bit field is equal to binary 0, then the access is on the conventionally aligned even byte boundary. Any other binary value causes the alignment to be offset by that number of bits to the right within the the addressed word.

Note that with the start bit displacement field set to all zeros, that bit map access is equivalent to conventional even word (16 bits) boundary addressing and the bit map appears to the system as a conventional area of I/O channel memory.

Orthogonal Access

Besides the 16 horizontal addressing modes, the access direction may be specified horizontally or vertically. The previous discussion on the addressing mode assumed horizontal access because this is the conventional access direction in bit-mapped display adapters. Setting the mode register horizontal access bit to off directs the adapter to access the bit map in the vertical orientation (down).

Masking

Since all memory accesses are I/O channel limited to 16 bit quantities, regardless of alignment, masking is provided to allow for a variable number of bits to be written in either access direction. The mode register contains a field called the write mask count, consisting of a 4-bit binary value. If the mask count equals X'0', then 16 bits are written. If the mask count equals X'1', then only one bit is written (the start X, Y bit). If the mask count equals X'2', then two bits are written.

Logic Functions

During an I/O channel memory write operation, the incoming write data from the I/O data channel is logically 'ANDed', 'ORed' or 'XORed' with the bit map data. The logical function is specified in the mode register. See "Mode Register" on page 24 for the logical functions performed.

Memory Mapped Memory Areas

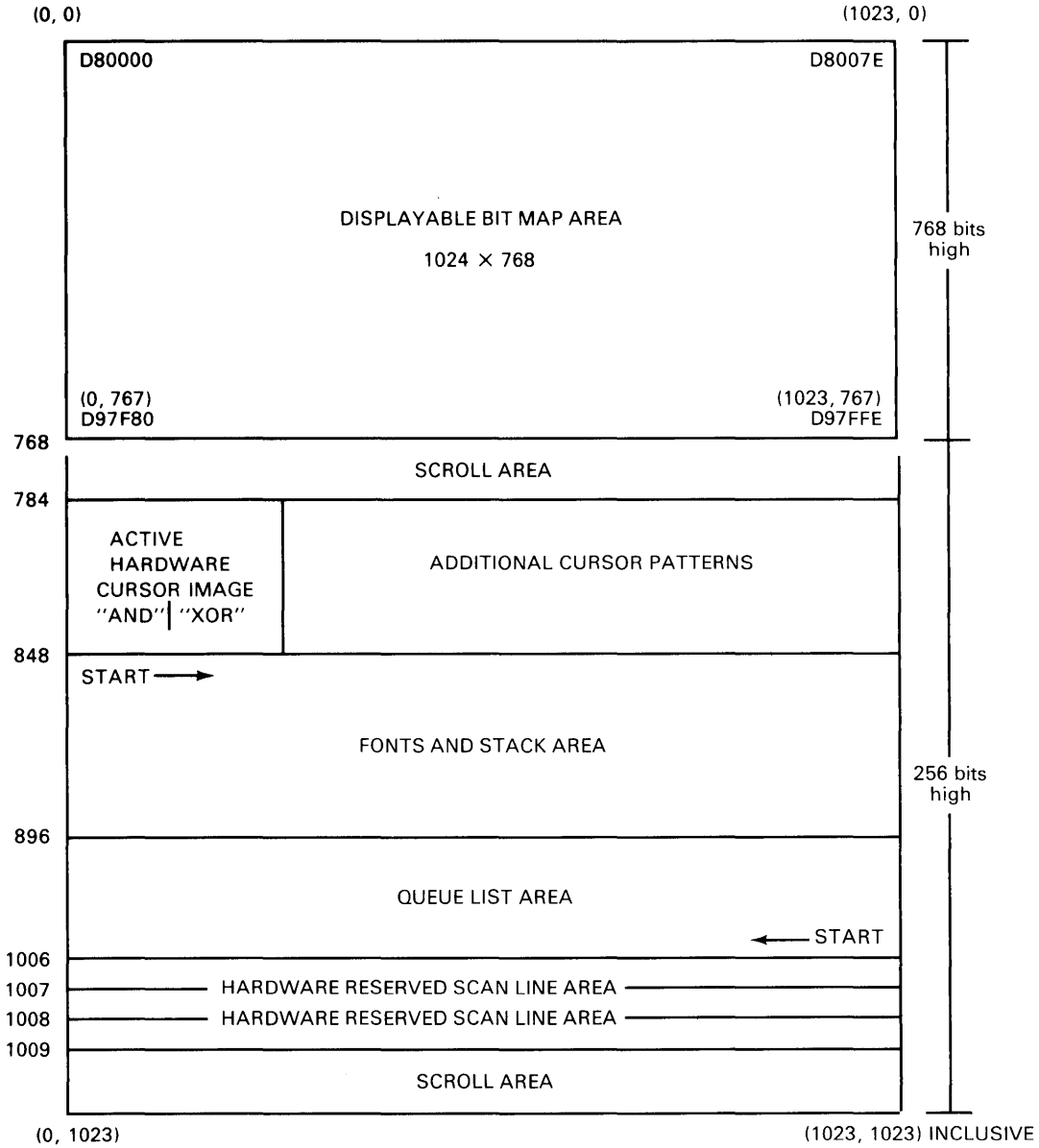


Figure 4. 1M-Pel Bit Map Frame Buffer (Suggested Area Usage)

Video Frame Buffer

A 1024 × 768 bit area is displayable from the bit map (1024 x 1024) and is located at the first 768 scan lines. The last 256 scan lines is a hidden area that contains queue lists, cursor patterns, reserved registers, vertical scroll areas, font areas, and stack areas. The channel addresses associated with the displayable portion of the bit map start at X'D80000' through X'D97FFF' (96K-bytes).

Cursor Area

The cursor is 48 × 64 bits and is built from two cursor patterns (an 'AND' pattern and an 'XOR' pattern) which are stored in the hidden bit map area. A reserved cursor save area saves the original bit map during active display of the cursor. The reserved area is located on scan line 1008 (see "Reserved Register Area" on page 13).

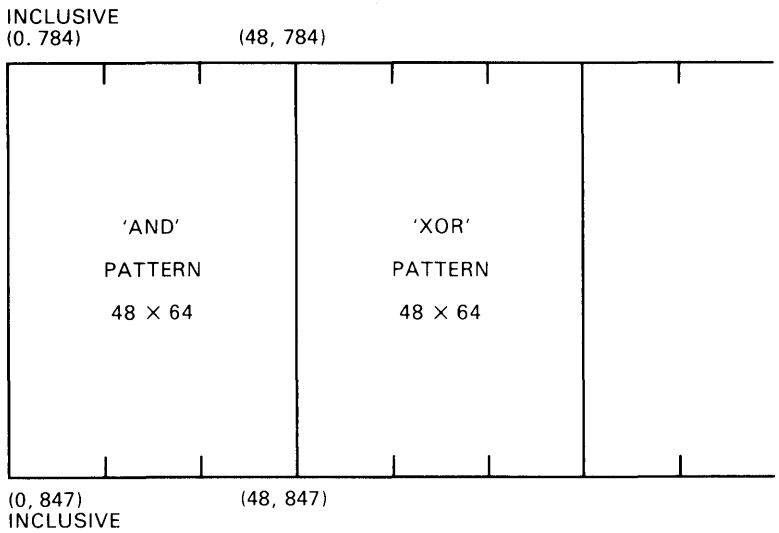


Figure 5. Hidden Cursor Patterns

Reserved Register Area

X Cursor Register

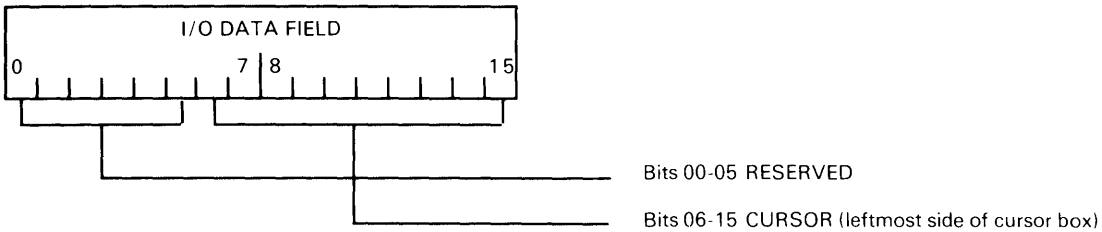


Figure 6. X Cursor Register (Address X'D9F800')

Note: This register is initialized to X'0000' with a POR or reset adapter command.

This register is loaded by the system to specify the X position in pels over which the hardware supported cursor is positioned. The valid range is from 0 (decimal) to $1024 - 48 = 976$ (decimal). Out of range values may cause distortion of the displayed image on those scan lines overlayed by the cursor pattern.

Y Cursor Register

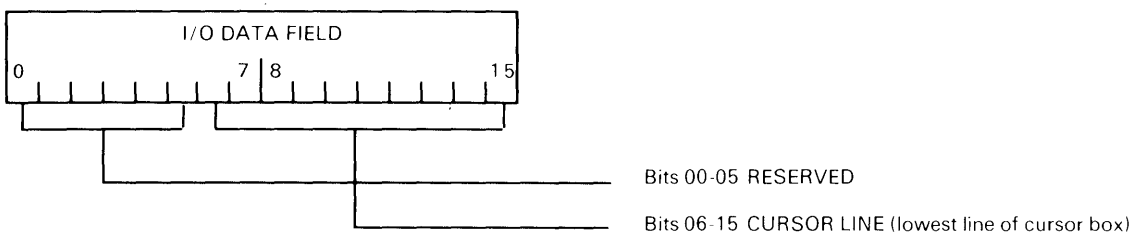


Figure 7. Y Cursor Register (Address X'D9F802')

Note: This register is initialized to X'FFFF' with a POR or reset adapter command.

This register is loaded by the system to specify the bottom scan line for the hardware supported cursor. When the Y cursor register equals 0, the bottom most scan line of the cursor is displayed on the first scan line on the screen (scan line 0). When the Y cursor register equals $767 + 63 = 830$, then the top most scan line of the cursor is displayed on the last displayable scan line of the screen (scan line 767). All other values of the Y cursor register between the top and bottom numbers, position the cursor in all the other Y bit positions on the screen. If the Y cursor register is greater than $831 = 767 + 64$, then the cursor is not displayed.

Queue Counter Register

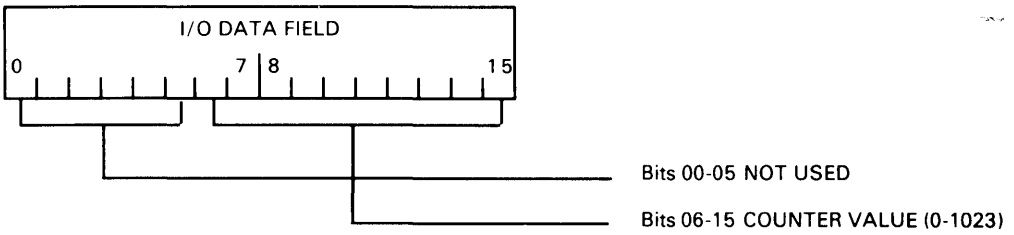


Figure 8. Queue Counter Register (Address X'D9F804')

Note: This register is initialized to X'0000' with a POR or reset adapter command.

This register contains the unexecuted 'queue loads' count. The counter is normally incremented by the system after a new command sequence has been loaded and is ready for execution by the adapter. It is decremented, conditionally by the adapter, at completion of a command sequence. As a part of the memory map, it is also fully read/writable by the system. The queue counter register should only be written when graphic operation processing is stopped (queue counter equals 0).

Queue Pointer Register

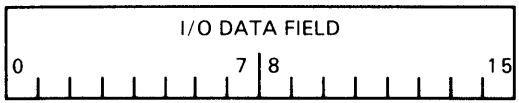
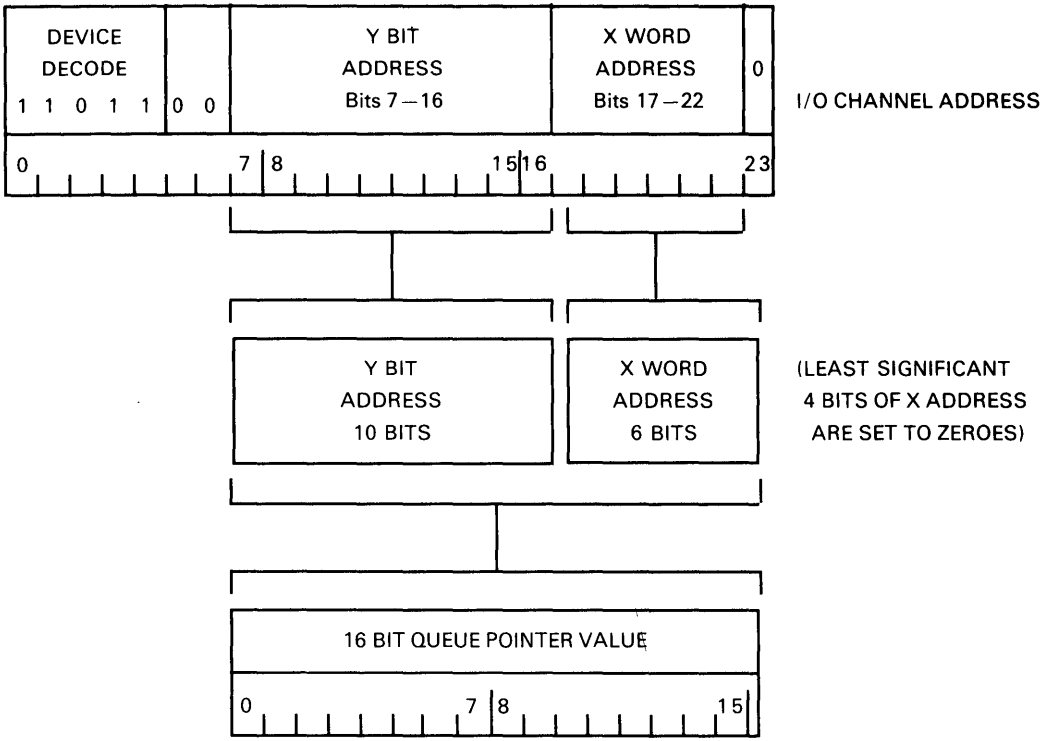


Figure 9. Queue Pointer Register (Address X'D9F806')

Note: This register is not initialized with a POR or reset adapter command.

The next queued command word location is stored in this register and may be read or modified via an adapter memory read/write channel operation. During normal graphic operation processing, this register is updated and used by the graphic operations processor as a program pointer into the graphic operations command queue. Care should be taken when updating this pointer.

I/O Address to Equivalent Queue Pointer Value Conversion



Examples of queue pointer values:

I/O Channel Address Value	Equivalent Queue Pointer Value
D988AE	C457
D9A000	D000
D9BFD0	DFE8
D9C000	E000
D9C02A	E015
D9E000	F000
D9F000	F800
D9F7FE	FBFF
D9FFFE	FFFF

Scan Line Register

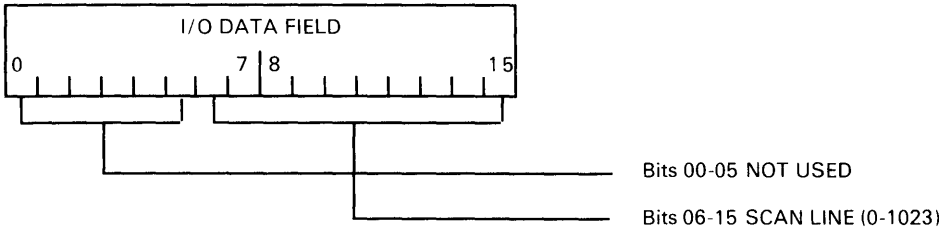


Figure 10. Scan Line Register (Address X'D9F808')

Note: This register is initialized to X'0000' with a POR or reset adapter command.

This register contains the next scan line pointer that will be serialized by the video output circuits. The table below shows the scan line value in relation to the vertical cycle.

Scan Line Values	Relative Vertical Period
000 - 767	Active display
None	Vertical front porch
768 - 770	Vertical Sync
771 - 791	Vertical back porch
792 - 1023	Not valid

Writing the scan line register should be avoided as it causes the vertical cycle to glitch, which may cause a vertical roll on the display.

Cursor Save Register

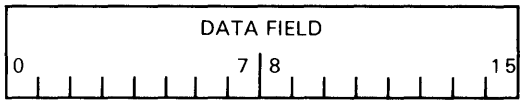


Figure 11. Cursor Save Registers (Address X'D9F80A', 'D9F80C', 'D9F80E')

Note: These registers are not initialized with a POR or reset adapter command.

The cursor save registers are used by the graphic operation processor when the display cursor is active. Bit map data may be lost if these registers are accidentally written.

Mode Shadow Register

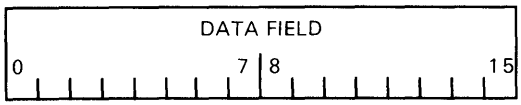


Figure 12. Mode Shadow Register (Address X'D9F812')

The mode shadow register is updated each time the mode register is written (see "Mode Register" on page 24). The programmer can read this register to determine the most recent data written to the mode register. Since the current mode register state affects the reading of the mode shadow register, the mode register should be set to horizontal access with start bit displacement set to zero. The mode shadow register is only useful in determining the current state of logic function bits and the write mask count.

Queue Link Pointer Register

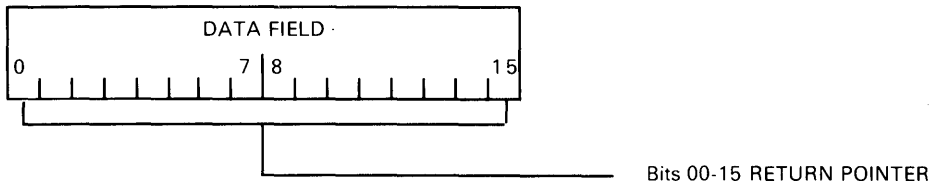


Figure 13. Queue Link Pointer Register (Address X'D9F814')

The graphic operations processor uses the queue link register during a branch and link queue graphic operation. The queue return point is stored in this register. Care should be taken when writing to this register. See graphic operation command queue for details.

Queue Mode Register

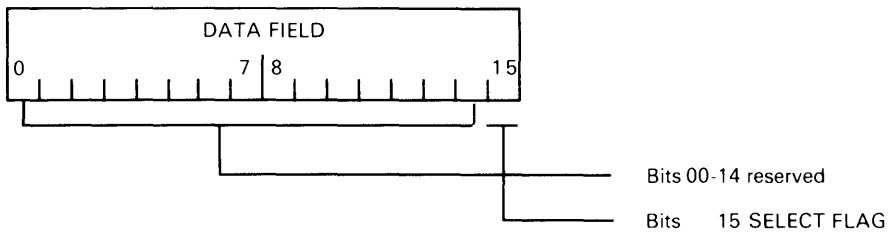


Figure 14. Queue Mode Register (Address X'D9F816')

The graphic operations processor uses the queue mode register for temporary memory. Care should be taken when writing to this register. See graphic operation command queue for details.

I/O Mapped Register Addressing

The Extended Monochrome Graphics Display Adapter responds to I/O addresses ranging from X'0D10' to X'0D2F' and X'06F3'.

I/O Address	Access	Function Performed
H'06F3'	Write	Interrupt Level 11 Acknowledge
H'0D10'	Write	Write Mode Register
X'0D12'	Write	Write Control Register
X'0D14'	Write	Increment Queue Counter
X'0D16'	Write	Diagnostic Promlevel Check
X'0D18'	Write	Reserved
X'0D1A'	Write	Enable Video Data Output
X'0D1C'	Write	Reserved
X'0D1E'	Write	Reserved
X'0D20'	Write	Reset Adapter
X'0D22'	Write	Reset Frame Sync Interrupt
X'0D24'	Write	Reset Raster Operation Interrupt
X'0D26'	Write	Disable DMA Processing
X'0D28'	Write	Enable DMA Processing
X'06F3'	Read	Reserved
X'0D10'	Read	Reserved
X'0D12'	Read	Read Status Register
X'0D14'	Read	Reserved
X'0D16'	Read	Reserved
X'0D18'	Read	Reserved

Figure 15 (Part 1 of 2). I/O Address Function Table

I/O Address	Access	Function Performed
X'0D1A'	Read	Disable Video Data Output
X'0D1C'	Read	Reserved
X'0D1E'	Read	Reserved
X'0D20'	Read	Reserved
X'0D22'	Read	Reserved
X'0D24'	Read	Reserved
X'0D26'	Read	Reserved
H'0D28'	Read	Reserved

Figure 15 (Part 2 of 2). I/O Address Function Table

Interrupt Acknowledge Register (Address X'06F3')

This is a write-only register and the data is of no significance. This address signals that a pending interrupt was serviced and re-enables the adapter interrupt request pulse generating circuit.

Reset Frame Sync Interrupt Register (Address X'0D22')

This is a write-only register and the data is of no significance. This address resets any pending frame sync interrupt (interrupt level 11). Interrupt 11 is a shared interrupt.

Reset Graphic Operation Interrupt Register (Address X'0D24')

This is a write-only register and the data is of no significance. This address resets any pending graphic operation interrupt (interrupt level 11). Interrupt 11 is a shared interrupt.

Mode Register

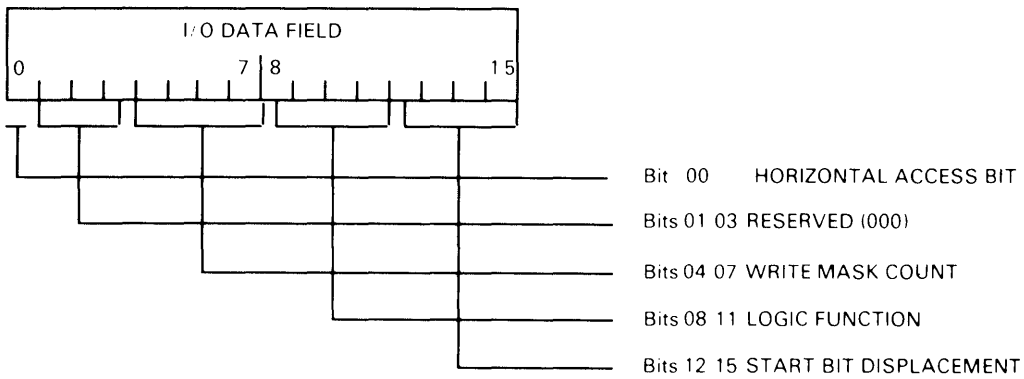


Figure 16. Mode Register (Address X'0D10')

Note: This register is initialized to X'8090' with a POR or reset adapter command.

This is a write-only hardware register. Its shadow is stored in mode shadow register which is described in "Mode Shadow Register" on page 20. The mode register is loaded to X'8090' when powered on or reset. This mode permits the adapter to accept 16-bit horizontal BAMBA accesses on a 16-bit boundary. The logic function only effects write operations and replaces the destination data bits with the I/O channel data bits.

With the horizontal access bit in an active state, a horizontal access starting at the X, Y address represented by the channel address and the start bit displacement (the 'start bit') occurs. The access is a string of 1 to 16 bits to the right of the 'start bit'. With the horizontal access bit in an inactive state, the access is a string of 1 to 16 bits down from the 'start bit'. During a write operation, the write mask count specifies the number of bits as shown in the chart below. The logical function, also shown below, specifies the action taken between the write data applied on the I/O channel and the existing data in the bit map.

The write mask count and logic function bits only affect write operations. The horizontal access and start bit displacement bits affect both read and write operations on the adapter bit map including the reading and writing of the various control registers. The programmer must be aware that, to correctly read the X cursor, Y cursor, queue counter, queue pointer, scan line, and mode shadow registers, the horizontal access bit must be set to 1 and the start bit displacement bits must all be set to 0. To write the appropriate registers, the mode register should be set to the reset state which is X'8090'.

Write Count (Hex)	Number of Bits Written
0	16
1	1
2	2
3	3
4	4
5	5
6	6
7	7
8	8
9	9
A	10
B	11
C	12
D	13
E	14
F	15

Figure 17. Write Count Mask

Logical Functional Value	Write Function Performed
0	Replace destination with zeros
1	I/O data 'AND' destination
2	¬I/O data 'AND' destination
3	Reserved
4	Reserved
5	Reserved
6	Reserved
7	Reserved
8	I/O data 'AND' ¬destination
9	Replace destination with I/O data
A	I/O data 'XOR' destination
11	I/O data 'OR' destination
12	Reserved
13	Reserved
14	¬I/O data 'OR' ¬destination
15	Replace destination with ones

Note: Destination refers to the bit map data.

Figure 18. Logical Functions

Control/Status Register

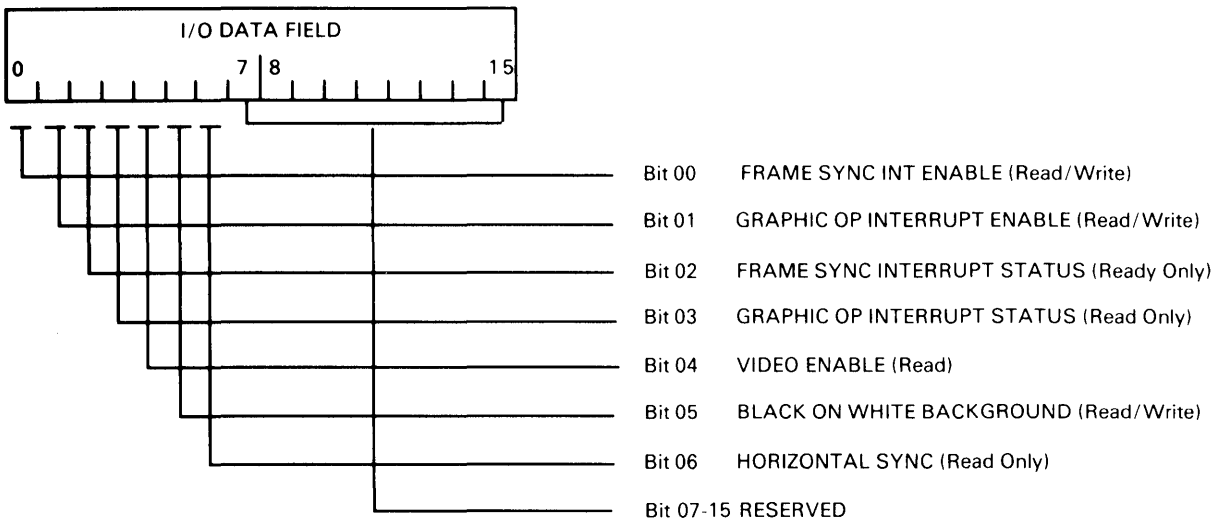


Figure 19. Control/Status Register (Address X'0D12')

Note: This register is initialized to X'00XX' with a POR or reset adapter command.

This control/status register may be read or written by the system processor. The control/status register contains the bits which enables processing, interrupts, and video. All bits in this register are normally initialized at power on to the off or 0 state.

Frame Sync Int Enable (bit 0)
This bit, when active, places any frame sync interrupt on the channel interface interrupt level 11. When inactive, no frame sync interrupts are placed on the channel and the frame sync interrupt latch is reset.

Graphic Operation Interrupt Enable, (bit 1)
This bit, when active, places any graphic operations interrupt on the channel interface interrupt level 11. When inactive, graphic operations interrupts are not placed on the channel and the graphic operations interrupt latch is reset.

Status (bits 2 and 3)
The frame sync interrupt status and the graphic operations interrupt status bits are read-only bits and indicate if these interrupts are active.

Video Enable (bit 4)
This bit, when active, indicates that video data from the bit map is being supplied to the monitor. If the bit is inactive, the video data from the bit map is not being supplied to the monitor. The video enable signal is manipulated via graphic operation types and I/O commands. The video enable signal should be activated after a POR or a reset adapter command.

Black on White Background (bit 5)
When active, inverts all video data going from the bit map to the video drive circuits. This gives an effect of a white background. Thus a clear screen operation (set bit off) or a disable video turns the screen white. When this bit is inactive, a white on black background is displayed. Therefore an off pel in the bit map corresponds to an off pel on the screen and the disabled video turns the screen black.

Horizontal Sync (bit 6)
This bit is for diagnostics and indicates the horizontal cycle is working.

Increment Queue Counter Register (Address X'0D14')

This is a write only register and the data is of no significance. A write to this register increments the command queue counter by one. The graphic operation processor decrements the queue counter with a flag bit in the execute command word. The graphic operation processor polls the queue counter to see if any graphic operations are pending execution on the queue. If the queue counter is not zero, processing of the queue begins or continues. The maximum number of increments before a decrement occurs is 1023. If count is incremented past 1023, then 1024 graphic operations queue loads are not executed.

Disable Video Data Output (Address X'0D1A')

A read operation from this I/O address disables the adapter video data output.

Enable Video Data Output (Address X'0D1A')

A write operation to this I/O address enables the adapter video data output to the monitor. Data written to this address is ignored.

Reset Adapter Register (Address X'0D20')

Note: After turning system power on, a write to this address is necessary to reset the adapter.

This is a write-only register and the data is of no significance. A write to this register resets the adapter and acts just like the I/O channel signal 'Reset Drive'. This command initializes the following registers:

Register Name	Register Addr	Register Data
X cursor register	D9F800	0000
Y cursor register	D9F802	FFFF
Queue counter register	D9F804	0000
Scan line register	D9F808	0000
Mode shadow register	D9F812	8090
Mode register	0D10 I/O	8090
Control/status register	0D12 I/O	0000

Graphic operation processing and command queue processing are stopped when this command is executed. Also, the vertical and horizontal sync cycles are interrupted and video is disabled.

Disable DMA Processing (Address X'0D26')

This I/O address allows the system to disable DMA processing by the adapter. Data written to this address is ignored.

Enable DMA Processing Register (Address X'0D28')

This I/O address allows the system to enable DMA processing by the adapter. Data written to this address is ignored.

Graphic Operation Commands

Besides allowing direct bit manipulation of the video frame buffer by the system processor, the Extended Monochrome Graphics Display Adapter can process commands which are received from the system and stored in a command queue. These commands are loaded into the adapter hidden bit map area as a linked-list structure. The graphic operation processor, which controls the command list execution, recognizes three basic command word types:

- Load register command
- Branch command
- Execute command.

The load register command sets values into the various graphic operation processor registers.

Normally, commands are fetched and processed from sequential 16-bit memory addresses. The branch command allows the list to be scattered throughout memory. Notice all command words are aligned on 16-bit word boundaries.

The execute command identifies the graphic operation type and contains flag bits that control and synchronize the queue operation.

Graphic Operation Command Queue

Commands for the graphic operations processor are loaded into a command queue by the system and sequentially executed (subject to branching types of commands) by the adapter. The command queue is represented by three elements:

- The actual memory locations used to hold the list of commands and referred as the queue
- The queue pointer which serves to hold the address within the queue of the next command word to be processed
- The queue counter which provides the synchronization and interlocking between the graphic operation processor and the system processor.

The queue consists of memory locations within the hidden bit map and is set up by the system processor as a linked list of graphic operation commands via memory write cycles over the I/O channel. The contents of any location within the queue may be accessed via a memory read cycle.

The queue pointer appears as a memory location within the I/O channel memory map and may be updated via a I/O channel memory write cycle.

The queue counter also appears as a memory location within the I/O channel memory map and may be updated via a I/O channel memory write cycle.

Normal operation of the command queue begins with the system loading a command or list of commands into the queue. This group is called a queue load. The last command is an 'Execute' instruction which contains a bit known as the decrement queue counter flag. This flag tells the graphic operation processor to decrement the queue counter register by one after the execution of the queue load is complete. If this is the first or only queue load, the queue pointer points to the first queue load command word.

Initially assume that the queue counter register is 0. Following the complete loading of the queue load (and the queue pointer if necessary), the system processor performs an I/O Write cycle to the increment queue counter register, causing the queue counter to be incremented by one. Once the queue counter has incremented past 0, the graphic operation processor begins fetching command words from the queue, starting at the address designated by the queue pointer register. As each command is processed, the queue pointer register is either decremented by one or loaded with a new queue location (branch address). The queue pointer always points to the next command word in the queue.

If the processed command word is an 'execute' type and the decrement queue counter flag is set, then the queue counter is decremented on completion of the graphic operation. If the processed command word is either a register load or branch type, or if the decrement queue counter flag is not set, then the queue counter is not changed by the graphic operation processor.

As queue loads are added to the queue by the system processor, the queue counter is incremented. As queue loads are completed, the counter is decremented. When the queue counter is decremented to zero, graphic operation processing is halted.

Graphic Commands

Execute Command

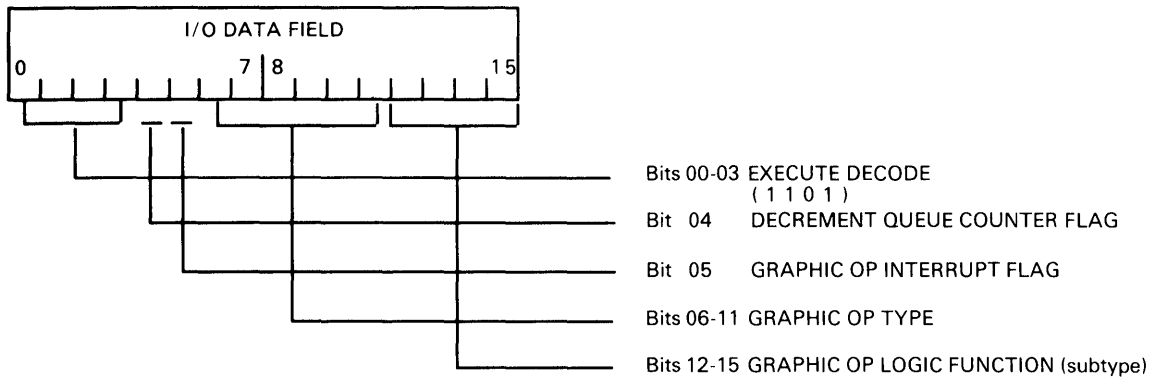


Figure 20. Execute Command Word (Address X'D9CXXX')

For each graphic operation there is one execute command word. It is designated via the execute decode B'1101'.

The execute command word contains a decrement queue counter flag. When this flag is active, it designates queue load sequence end. If the flag is inactive, more graphic operations must follow. Before incrementing the queue counter, a complete queue load sequence (decrement queue counter flag) must be loaded somewhere in the queue list structure.

The graphic op interrupt flag bit, if active, sets the graphic operation interrupt latch at the completion of the graphic operation. This action can be observed via the graphic operation interrupt status (bit 3) in the status register. The only way to reset the graphic operation interrupt latch is via an I/O channel write of a reset graphic operations command or a reset adapter command. If the graphic operation interrupt enable bit (bit 1) is active in the control register and the graphic operation interrupt latch is active, then an active interrupt is driven on the I/O channel. Disabling graphic operation interrupts does not reset the interrupt latch.

The graphic operation type field specifies different graphic operation types. The graphic operation logic function field specifies a subtype within a graphic operation sequence. See graphic operations definitions for specifics about each graphic operations type and subtype.

Bits 12-15	Logic Function (Subtype)
0	Set all bits to 0.
1	+ Operand A and + Operand B
2	+ Operand A and -Operand B
3	Pass through Operand A
4	All zero's
5	+ Operand A and + Operand B
6	+ Operand A and -Operand B
7	Pass through Operand A
8	-Operand A and + Operand B
9	Pass through Operand B (test)
A	+ Operand A xor + Operand B
B	+ Operand A or + Operand B
C	-Operand A or + Operand B
D	Pass through Operand B (test)
E	-Operand A or -Operand B
F	Set all bits to 1.

Figure 21. Graphic Operation Functions

Branch Command

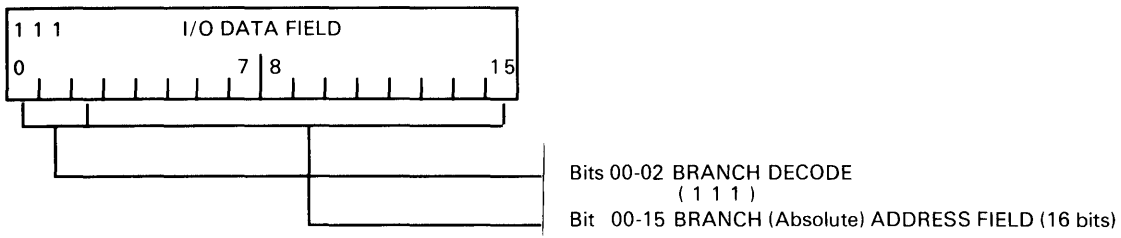


Figure 22. Branch Command Word

The branch address is 16 bits. The 10 high order bits (including the 3 branch decode bits) of the branch command word correspond to a Y address.

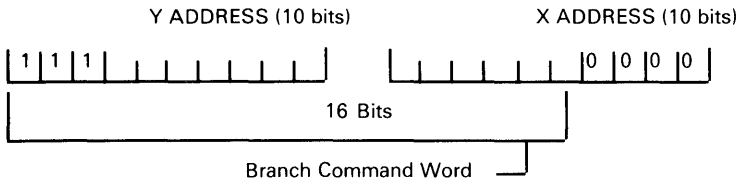


Figure 23. Branch Command Word Address Field

The other 6 (low order) bits of the branch command word form the most significant 6 bits of a 10-bit X address with the 4-least significant bits assumed to be all 0's. Therefore, branching is restricted to 16-bit word boundaries (4 LSB's of X address are all 0's) and the lower 1/8 of the bit map (Y address greater than or equal X'380', scan lines 896-1023).

Register Load

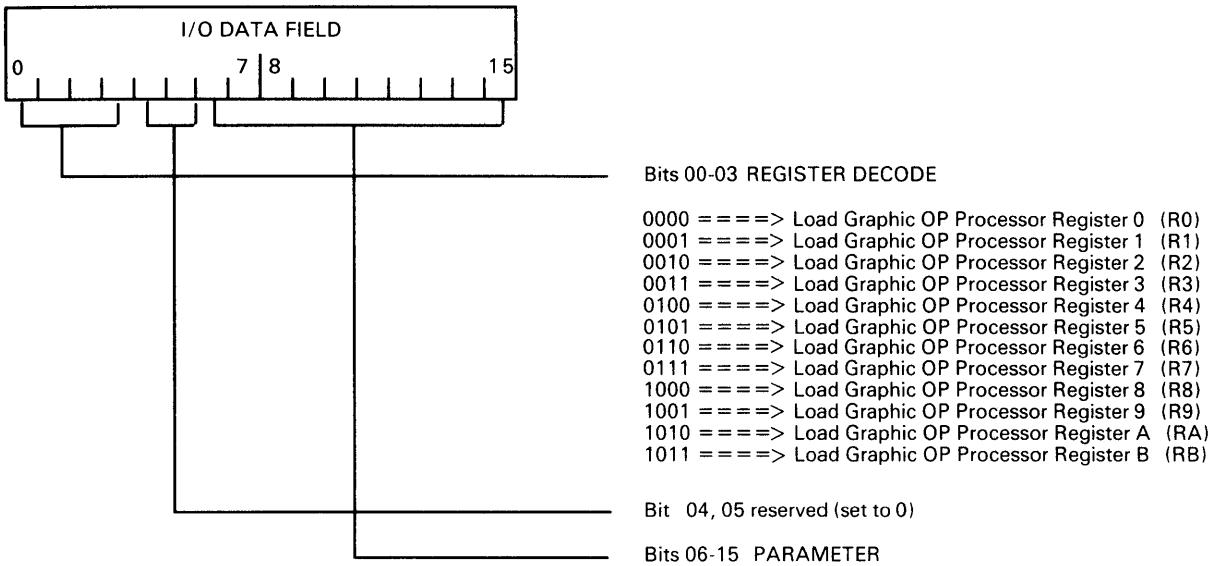


Figure 24. Register Load Command Word

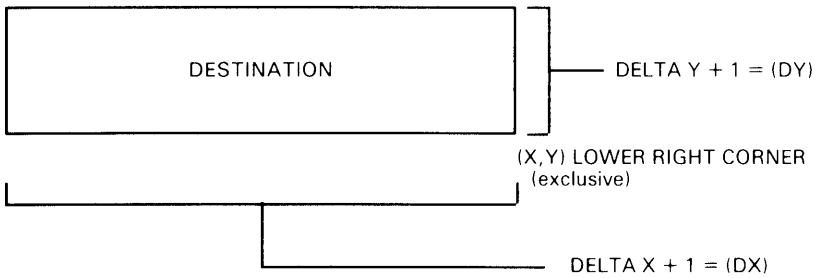
Bits 0-3 specify 1 of 12 graphic operation registers. Different graphic operations require different registers to be loaded.

Rectangular Destination Only Graphic Operation Types

A rectangular region of pels (destination) is operated on by the graphic operation processor in the bit map. The graphic operation either writes or read-modify-writes the destination specified.

The parameters specifying the rectangular destination are the coordinates of the lower right corner point (X, Y exclusive) and the size (DX, DY) of the rectangular region.

Note: The microcode register ending conditions are defined for graphic operation types X'2F','35','36','2C','2D', and '2E' only. All other graphic operation types have register ending conditions which are not defined.



Notes:

1. The DX & DY variables represent the actual number of pels written.
2. The X & Y variables are the right-bottom corner pel exclusive.

The X, Y, DX and DY parameters are specified in the queue load. Not all parameters need be specified for each graphic operation. Only the parameters that change from graphic operation to the next need be specified. See register contents at completion of graphic operation. An example of a queue load that clears a 1024 x 768 pel screen is shown below. Note that a hex value of '000' represents a value of 1024 pels both for X and DX. The same is true for Y and DY.

Queue Address	Queue Data	Action By Queue Processing	Register Start	Register Min Max	Reg End
Que Ptr	X'7000'	Load X into R7	X = 1024	1 1024	X
Que Ptr-1	X'6300'	Load Y into R6	Y = 768	1 1024	Y
Que Ptr-2	X'9000'	Load DX into R9	DX = 1024	1 DX = X	DX
Que Ptr-3	X'8300'	Load DY into R8	DY = 768	1 DY = Y	DY
Que Ptr-4	X'D2F0'	Execute Graphic Operation			

Note: Unpredictable results might occur if DX > X or DY > Y.

Write Destination (X'2F')

This graphic operation type writes a rectangular area of pels in the bit map. Three subtypes are specified via the LF (graphic op logic function) bits as listed below.

Rectangular fill replace off:	LF = 0, Set destination to 0
Rectangular XOR destination:	LF = 9, Msk = destination (Reverse video)
Rectangular fill replace on:	LF = F, Set destination to 1

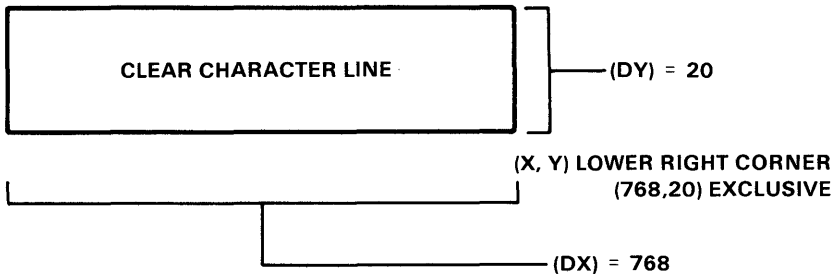


Figure 25. Clear Character Line (X = 768,Y = 20,DX = 768,DY = 20)

Read Modify Write Destination (X'2F')

This graphic operation type read-modify-writes a rectangular area of pels in the bit map. Only one subtype is specified via the LF bits as listed below.

Rectangular XOR Destination LF = 9, ¬Destination = Result Destination

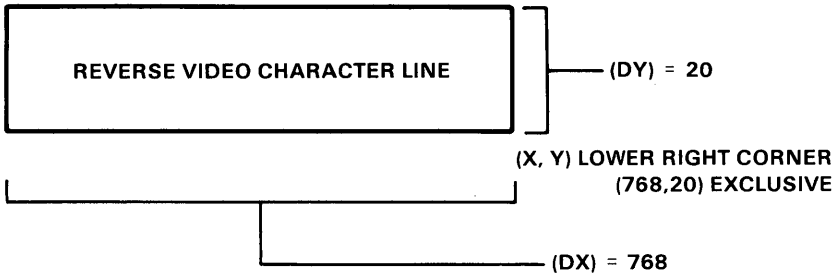


Figure 26. Reverse Video Character Line (X = 768,Y = 20,DX = 768,DY = 20)

Rectangular Source and Destination Graphic Operation Types

Two rectangular regions of pels (source-destination) are operated on by the graphic operation processor. The graphic operation reads the source and either read-modify-writes or writes the destination. The source data is to either replace the destination data, as in a copy operation or, merge data as in an AND, OR or XOR merge operations.

These graphic operation types require both a source and a destination X, Y address pair. The address pair depicts the source corner point (XS,YS) and the destination corner point (XD,YD). The size (DX,DY) of the rectangle must also be specified.

The queue load for these graphic operation types is up to seven 16-bit words as shown below. The queue load below represents a rectangular area of pels being moved down by one pel. The rectangular area defined is 1024 x 767.

Queue Address	Queue Data	Action By Queue Processing	Register Start	Register Min	Register Max
Que Ptr	X'7000'	Load XD into R7	XD = 1024	1	1024
Que Ptr-1	X'A300'	Load YD into RA	YD = 768	1	1024
Que Ptr-2	X'5000'	Load XS into R5	XS = 1024	1	1024
Que Ptr-3	X'42FF'	Load YS into R4	YS = 767	1	1024
Que Ptr-4	X'9000'	Load DX into R9	DX = 1024	1	DX = X
Que Ptr-5	X'02FF'	Load DY into R0	DY = 767	1	DY = Y
Que Ptr-6	X'D300'	Execute copy/scroll			

Note: Unpredictable results might occur if DX > XD,DY > YD,DX > XS or DY > YS.

Rectangular Copy/Merge (X'30')

The rectangular copy/merge graphic operation reads a source rectangle defined by XS, YS, DX and DY writes a destination rectangle defined by XD, YD, DX and DY as shown below. Any type of overlapping source and destination combinations are allowed. The YD register is preserved during graphic operation type X'30'.

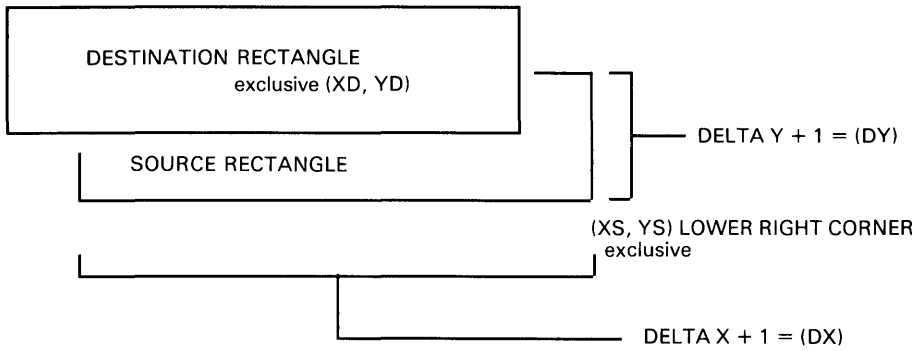


Figure 27. Copy Overlapped Character Line

To achieve a rectangular copy, the logical function (subtype) must be set to X'9'.

In consecutive repeated copy operations (graphic operations type 30) to locations with a constant destination Y coordinate, the destination Y coordinate need only be specified in the first queue load since its entry value is preserved. This speeds up the copy execution since only 6 parameters instead of the usual 7 need to be processed. An example of a general use for this preservation of the destination Y value is in the copying of character images from the display hidden area to a single horizontal character line in the active display area.

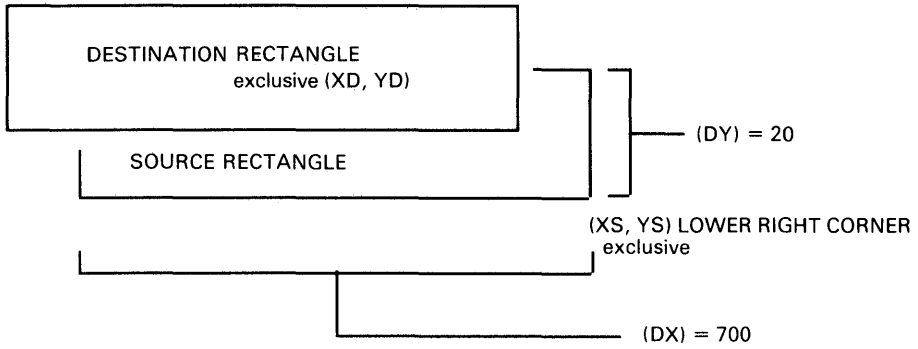


Figure 28. Copy Character Line (XD = 768, YD = 20, XS = 800, YS = 30, DX = 768, YD = 16)

To achieve a merge operation while copying the source data one of these subtypes must be specified.

Rectangular Merge And:	LF = 1, Source & destination = Result destination
Rectangular Merge And¬:	LF = 2, Source & ¬ destination = Result destination
Rectangular Merge ¬And:	LF = 8, ¬ Source & destination = Result destination
Rectangular Merge XOR:	LF = A, Source XOR destination = Result destination
Rectangular Merge OR:	LF = B, Source OR destination = Result destination
Rectangular merge ¬OR:	LF = C, ¬ Source OR destination = Result destination
Rectangular merge ¬OR¬:	LF = E, ¬ Source OR ¬ destination = Result destination

While copying the source, data is merged with the destination via subtype specification.

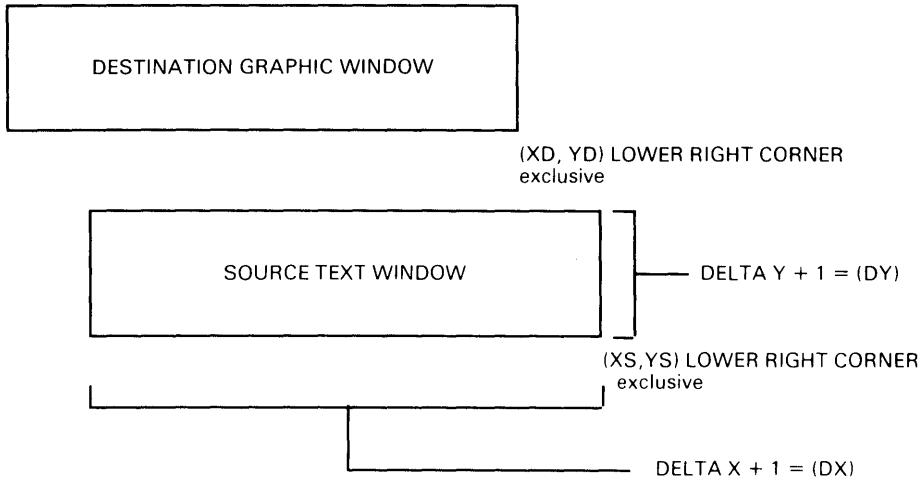


Figure 29. 'And' a Test Source Window to a Graphic Destination Window.

Rectangular Copy/Merge Rotate +90 (X'31')

This graphic operation type is similar to the rectangular copy/merge graphic operation. The destination and source rectangular areas should not overlap.

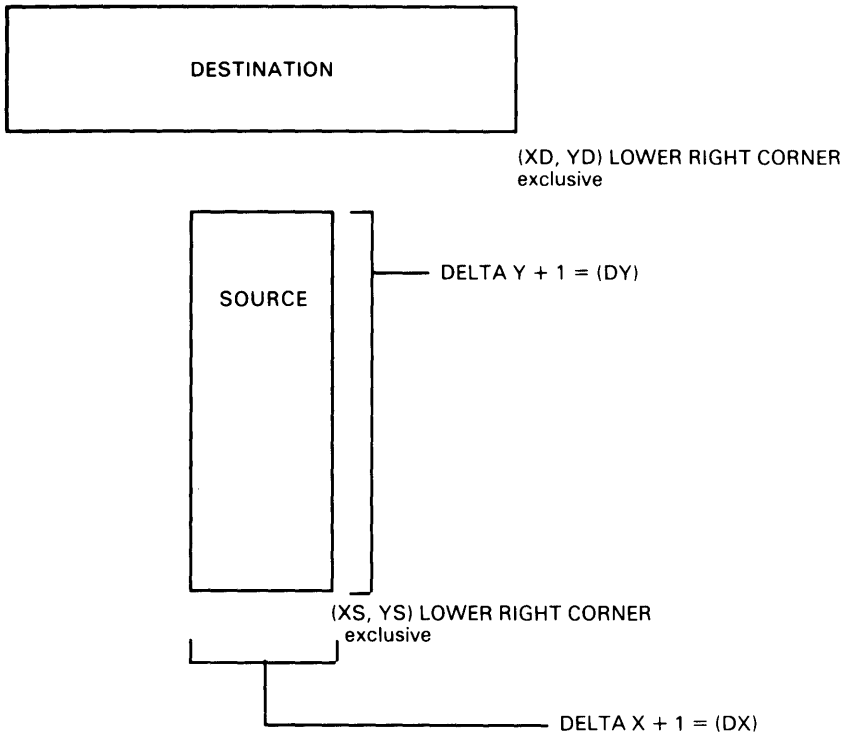


Figure 30. Rotate Window +90 Degrees

Rectangular Copy/Merge Rotate -90 (X'32')

This graphic operation type is similar to the rectangular copy/merge graphic operation. The destination and source rectangular areas should not overlap.

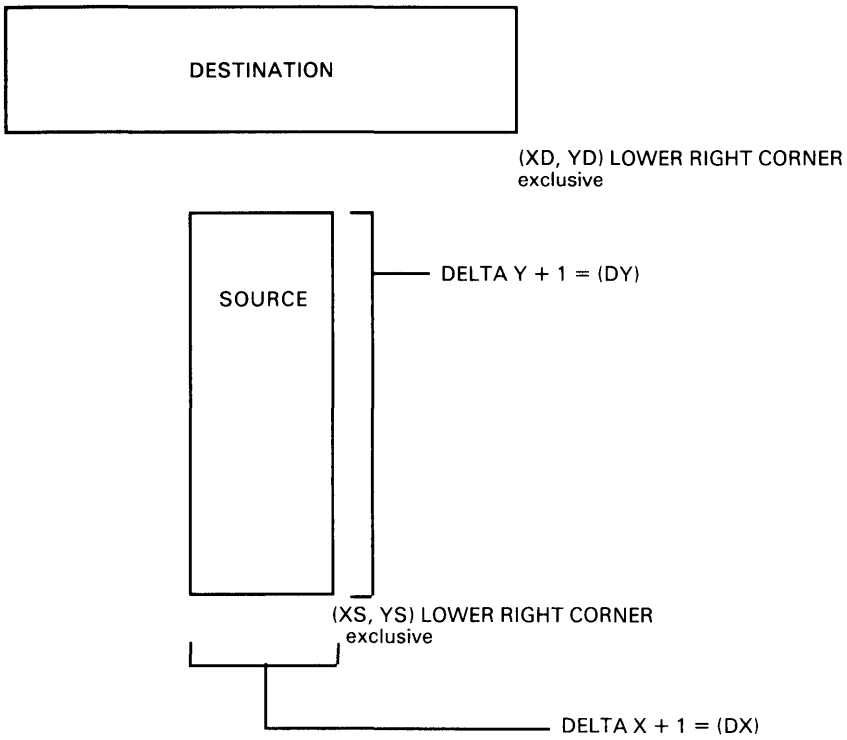


Figure 31. Rotate Window -90 Degrees

Rectangular Copy/Merge 180 X Axis Symmetry (X'33')

This graphic operation type is similar to the rectangular copy/merge graphic operation. The destination and source rectangular areas should not overlap.

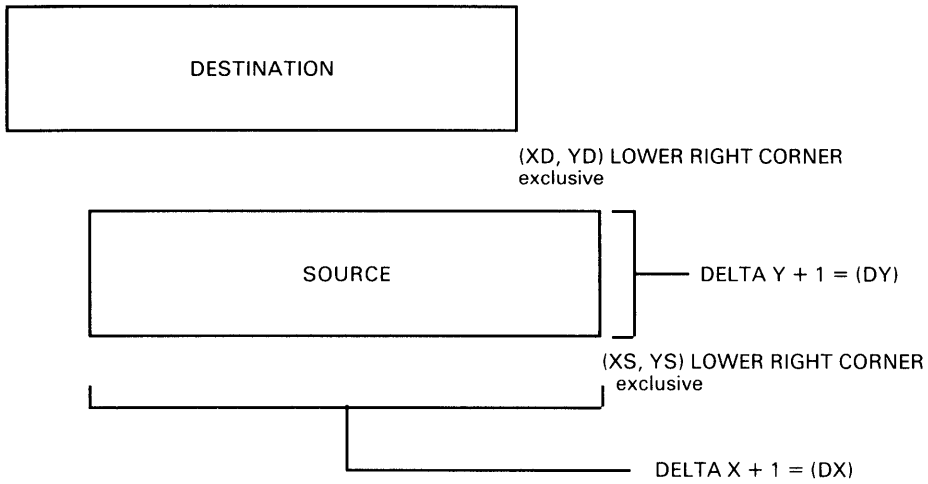


Figure 32. Flip Window 180 Degrees Along the X Axis

Rectangular Copy/Merge 180 Y Axis Symmetry (X'34')

This graphic operation type is similar to the rectangular copy/merge graphic operation. The destination and source rectangular areas should not overlap.

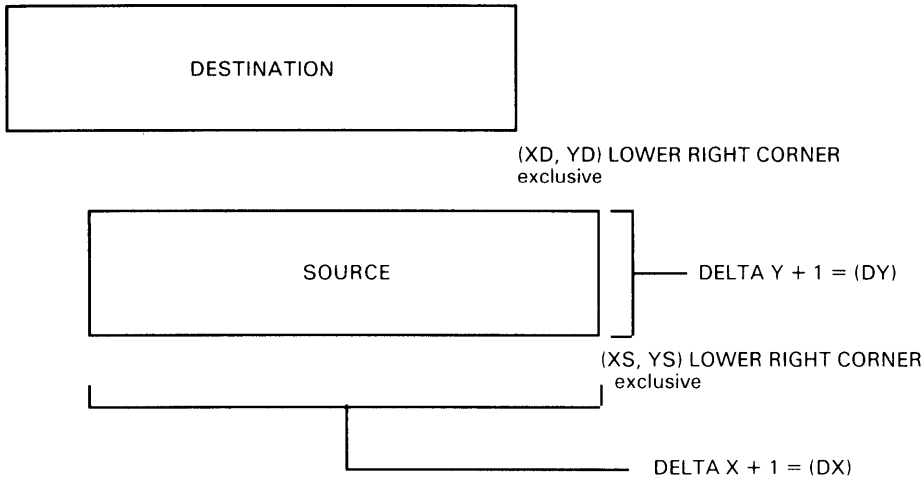


Figure 33. Flip Window 180 Degrees Along the Y Axis

Vector Draw Graphic Operations

Vector Draw (X'35')

This graphic operation draws a straight line contained within the bit map (1024 x 1024). The line is defined by four parameters, *X from, Y from, X to, and Y to.* The logical operators on the bit map for the line draw are set line on (F), set line off(0), and 'XOR' line (A).

Example: Draw a Line From (48,256) To (16,512).

Queue Address	Queue Data	Action By Queue Processing	Register Start	Register Min Max	Reg End
Que Ptr	X'1030'	Load X from R1	X1 = 48	0 1023	X to
Que Ptr-1	X'0100'	Load Y from R0	Y1 = 256	0 1023	Y to
Que Ptr-2	X'5010'	Load X to R5	X2 = 16	0 1023	
Que Ptr-3	X'6200'	LOAD Y to R6	Y2 = 512	0 1023	
Que Ptr-8	X'D35F'	Execute Line Draw	Set line on	No decrement	

At the completion of the line draw, the X,Y 'to' coordinate is saved in the X,Y 'from' coordinate registers. This permits the next graphic operation to use this saved X,Y 'to' coordinate information for the next X,Y 'from' coordinate with out having to reload (polyline) as shown below.

Example: Continue Drawing a New Line From (16,512) to (256,512)

Queue Address	Queue Data	Action By Queue Processing	Register Start	Register Min Max
Que Ptr	X'5100'	Load X2 into R5	X2 = 256	0 1023
Que Ptr-1	X'6200'	Load into R6	Y2 = 512	0 1023
Que Ptr-2	X'D35F'	Execute line draw	Set line on	No decrement

Note: X,Y 'to' coordinate is saved in the X,Y 'from' registers.

Vector Draw With Ending Null (X'36')

This graphic operation functions in exactly the same way and with the same parameters as the preceding vector draw graphic operation type X'35'. The only exception being that the ending pel of the line is not written to the bit map. Used for polyline XOR.

Relative Draw With Ending Null (X'2C')

The relative draw with ending null graphic operation allows the user to specify the next vector to be drawn as a delta or increment from the last end point specified and end with a null point. Logic functions are the same as type X'35'.

Example: Relative Draw with Ending Null (X1 and Y1 values are from current contents of R1 and R0).

Queue Address	Queue Data	Action By Queue Processing	Function
Que Ptr	X'5100'	Load DX into R5	
Que Ptr-1	X'6200'	Load DY into R6	
Que Ptr-2	X'D2C0'	Execute relative draw with ending null.	R5 = X1 + DX R6 = Y1 + DY
		Draw vector from (X1,Y1) to (X1 + DX,Y1 + DY) w/END NULL	

Relative Draw (X'2D')

The relative draw graphic operation operates in the same manner as the graphic operation type X'2C' above, except the drawn vector does not end with a null.

Example: Relative Draw (X1 and Y1 values are from the current contents of R1 and R0).

Queue Address	Queue Data	Action By Queue Processing	Function
Que Ptr	X'5100'	Load DX into R5	
Que Ptr-1	X'6200'	Load DY into R6	
Que Ptr-2	X'D2D0'	Execute relative draw.	R5 = X1 + DX R6 = Y1 + DY
		Draw vector from (X1,Y1) to (X1 + DX,Y1 + DY)	

Relative Move to (X'2E')

The relative move to graphic operation allows the user to add a delta or increment to the last start point coordinates specified rather than to require that new absolute coordinates to be specified.

Example: Relative Move To (Update the contents of R1 and R0 for new X1 and Y1).

Queue Address	Queue Data	Action By Queue Processing	Function
Que Ptr	X'2100'	Load DX into R2	R2 = X1 + DX
Que Ptr-1	X'6200'	Load DY TO > R6	R6 = Y1 + DY
Que Ptr-2	X'D2E0'	Execute relative move to	
		No vector is drawn, only X1 and Y1 are updated for later use.	

TDDMA I/O Copy Graphic Operation

TDDMA Cut/Paste From I/O Channel Memory (X'37')

Valid subtypes for graphic operation type X'37' are 1, 2, 8, 9, A, B, C, and E.

Example: Copy an Image From I/O Channel Memory Starting at Address X'1012'.

Bit 4 of the starting data word (X'1012') is the upper left corner of the rectangular copied. The image array starts at address X'1000'and ends at X'107E'. The destination rectangle is located at X = 11 and Y = 12 (upper left hand corner X,Y). The width (DX) = 31 bits. The height (DY) = 3 bits.

The skip value is the number of data words in each image array width line that are within the height of the desired rectangle, but do not contain data within the width of the desired rectangle and thus are skipped in the DMA transfer process.

The Skip value is calculated as follows:

In this example: AS = 128 x 8 = 1024, DW = DX = 31, and Bit Offset = 4.

```
Thus Skip = (Ceil( (AS   - 15) - (DW - (16 - Bit Offset ) ) ) /16) )-1
     Skip = (Ceil( (1024 - 15) - (31 - (16 -     4    ) ) ) /16) )- 1
     Skip = (Ceil(   990                              ) /16) )- 1
     Skip = (Ceil(    61.875                             ) )- 1
     Skip =          62                                   - 1
     Skip =          61   = X'3D'
```

AS Array Size or DX Array x 8: the source image width in bits.

DW Result Destination Width: the width of the unrotated destination rectangle in bits.

Ceil Ceiling Function: least integer greater than function.

Skip (Ceil((AS - 15) - (DW - (16 - Bit Offset)) /16)) - 1

Note: If Bit Offset = 0, the above formula simplifies to Skip = Ceil((AS - DW)/16)

Queue Load

Reg	Function	Value of Register	Comment
RA	A23-A17 = Source high	X'A000'	Upper left start word address
RB	A16-A9 = Source mid	X'B008'	Upper left start word address
R4	A8-A1 = Source low	X'4009'	Upper left start word address
R3	Bit offset	X'3004'	Starting bit of the starting word
R2	Skip (16 bit words)	X'203D'	Array width = X'080' bytes
R7	X destination	X'702A'	Lower right corner exclusive
R6	Y destination	X'600F'	Lower right corner exclusive
R5	Delta X + 1	X'501F'	Number of pels in width
R0	Delta Y + 1	X'0003'	Number of pels in height
	Execute Command	X'D379'	

Note: The registers used by the DMA graphic operations have the following ending conditions:

Reg	Value of Register	Change From Initial Conditions
RA	X'AXXX'	A23-A17 = Address of last I/O channel memory word read
RB	X'BXXX'	A16-A9 = Address of last I/O channel memory word read
R4	X'4XXX'	A8-A1 = Address of last I/O channel memory word read
R3	X'3XXX'	Unchanged
R2	X'2XXX'	Unchanged
R7	X'7XXX'	Unchanged
R6	X'6XXX'	Unchanged
R5	X'5XXX'	Unchanged
R0	X'0000'	Set to zero

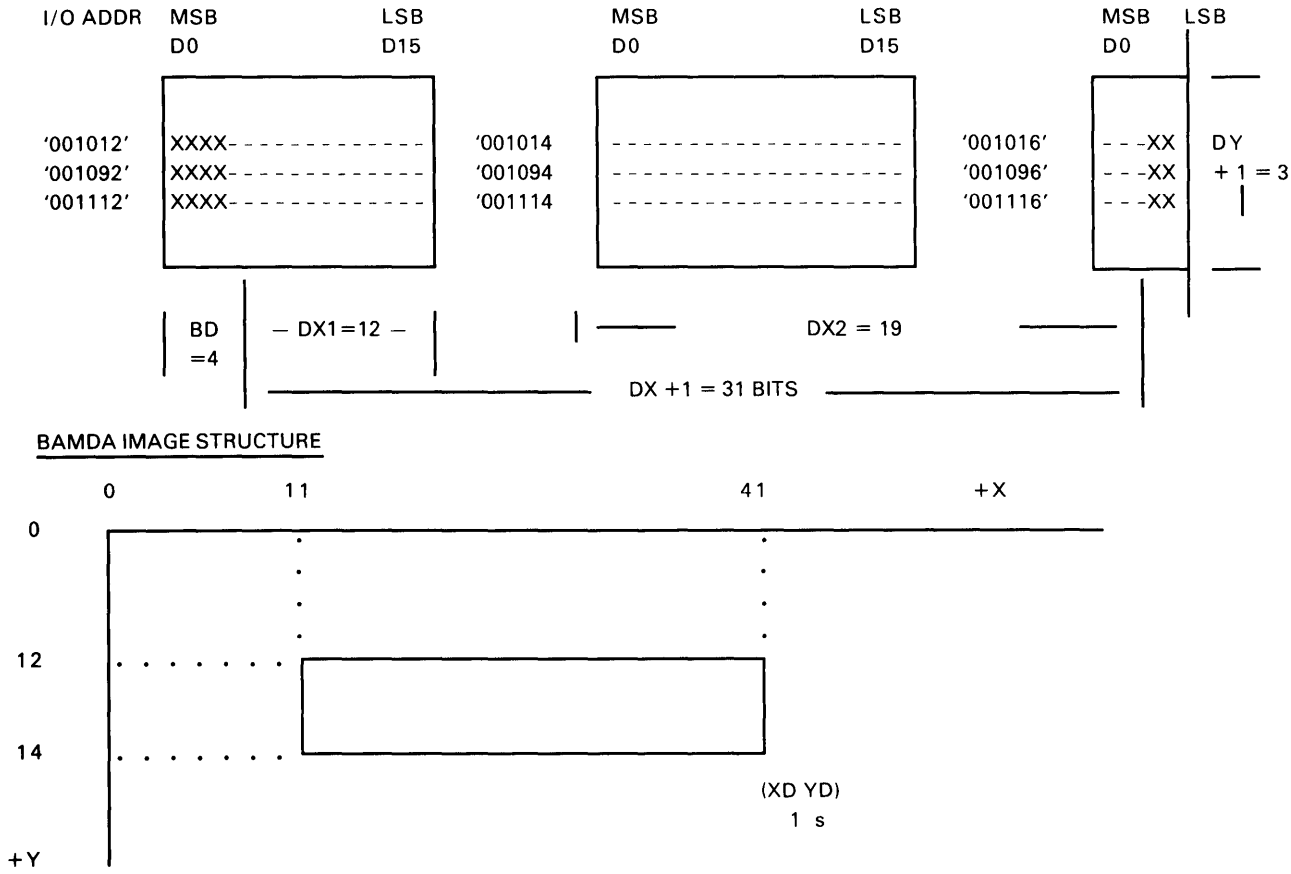


Figure 34. TDDMA I/O Copy Example

TDDMA Cut/Paste From I/O Channel Memory +90 Degree Rotate (X'38')

This graphic operation performs in the same manner as graphic operation type X'37' except that the destination is rotated 90 degrees counterclockwise.

Note: *Notice the change in the queue load parameters.

Queue Load

Reg	Function	Value of Register	Comment
RA	A23-A17 = Source high	X'A000'	Upper left start word address
RB	A16-A9 = Source mid	X'B008'	Upper left start word address
R4	A8-A1 = Source low	X'4009'	Upper left start word address
R3	Bit offset	X'3004'	Starting bit of the starting word
R2	Skip (16 bit words)	X'203D'	Array width = X'080' bytes
R6 *	X destination	X'600A'	Lower left corner exclusive (XD-(DX + 1)-1)
R7 *	Y destination	X'700F'	Lower left corner exclusive (YD)
R0 *	10 bit 2's complement of Delta X + 1	X'03E1'	
R5 *	Delta Y + 1	X'5003'	
	Execute Command	X'D389'	

TDDMA Cut/Paste From I/O Channel Memory 45 Degree Diag Flip (X'39')

This graphic operation performs in the same manner as graphic operation type X'37' except that the destination is rotated 180 degrees about the 45 degree diagonal.

Note: * Notice the change in the queue load parameters.

Queue Load

Reg	Function	Value of Register	Comment
RA *	A23-A17 = Source high	X'A000'	Upper left start word address
RB *	A16-A9 = Source mid	X'B008'	Upper left start word address
R4 *	A8-A1 = Source low	X'4009'	Upper left start word address
43	Bit offset	X'3004'	Starting bit of the starting word
R2	Skip (16 bit words)	X'203D'	Array width = X'080' bytes
R6	X destination	X'600F'	Lower right corner exclusive
R7	Y destination	X'700B'	Lower right corner exclusive
R0	Delta X + 1	X'001F'	
R5	Delta Y + 1	X'5003'	
	Execute Command	X'D399'	

TDDMA Cut/Paste From I/O Channel Memory 180 Degree X Axis Symmetry (X'3A')

This graphic operation performs in the same manner as graphic operation type X'37' except that the destination is rotated 180 degrees about the X axis.

Note: Notice the change in the queue load parameters.

Queue Load

Reg	Function	Value of Register	Comment
RA *	A23-A17 = Source high	X'A000'	Upper left start word address
RB *	A16-A9 = Source mid	X'B008'	Upper left start word address
R4 *	A8-A1 = Source low	X'4009'	Upper left start word address
R3	Bit offset	X'3004'	Starting bit of the starting word
R2	Skip (16 bit words)	X'203D'	Array width = X'080' bytes
R7	X destination	X'700B'	Upper right corner exclusive (XD)
R6	Y destination	X'600F'	Upper right corner exclusive (YD-(Dx + 1)-1)
R5	Delta X + 1	X'5013'	
R0 *	10 bit 2's Complement of Delta Y + 1	X'03FD'	
	Execute Command	X'D3A9'	

Control Graphic Operation Types

Branch and Link Queue (X'3B')

The branch and link queue graphic operation allows branching to a specified queue location and saves the queue pointer value for the next-in-line location for later use by the return queue graphic operation.

Example: Branch and Link Queue, branch to queue address X'E015', and save the next queue address X'EFFD' in the queue link pointer register for later use by the return queue operation.

Queue Address	Queue Data	Action By Queue Processing
X'F000'	X'A0E0	Load RA with 8 high order bits of branch target queue pointer value
X'EFFF'	X'B015	Load RA with 8 low order bits of branch target queue pointer value
X'EFFE'	X'D3B0	Execute branch and link queue graphic operation
X'EFFD'		Link address for next operation after return queue.

Return Queue (X'3C')

The return queue graphic operation returns from a branch and link queue operation to the queue address, which was saved in the queue link pointer register. In the previous example, the return queue graphic operation causes queue processing to return to the queue address X'EFFD'.

Scan Line Sync (X'3D')

The scan line sync graphic operation provides a means of stopping queue processing until a specified scan line number (SLN) appears in the scan line counter register. The scan line complement (SLC) is the input data to the scan line sync graphic operation and is the 10 bit 2's complement of the scan line number.

Note: Valid SLN values are X'000' and X'001' to X'317. The valid SLC values are X'000' and X'3FF' to X'0E9'. Out of range values cause the adapter queue processing to hang in such a manner that the adapter must be reset to restore queue processing operation.

Example: Scan Line Sync, stop queue processing until the scan line counter reaches a value of X'1F5'.

Queue Address	Queue Data	Action By Queue Processing
		Calculate SLC = 10 bit 2's complement of X'1F5' = X'20B'
Que Ptr	X'A20B'	Load RA with 10 bit scan line complement
Que Ptr-1	X'D3D0'	Execute the scan line sync graphic operation

Graphic Operation Type X'3E' Set Video On

The set video on graphic operation enables the video data line to the display monitor. When the video data line is enabled, the information contained in the displayable bit map area is displayed on the monitor.

Example: Set Video On

Queue Address	Queue Data	Action By Queue Processing
Que Ptr	X'D3E0'	Execute set video on - Enable the video data line.

Set Video Off (X'3F')

The set video off graphic operation disables the video data line to the display monitor. When the video data line is disabled, the information contained in the displayable bit map area is not displayed on the monitor. The monitor instead produces the background screen type, either an all white screen or an all black screen, depending on the current setting of the black on white background bit in the control/status register.

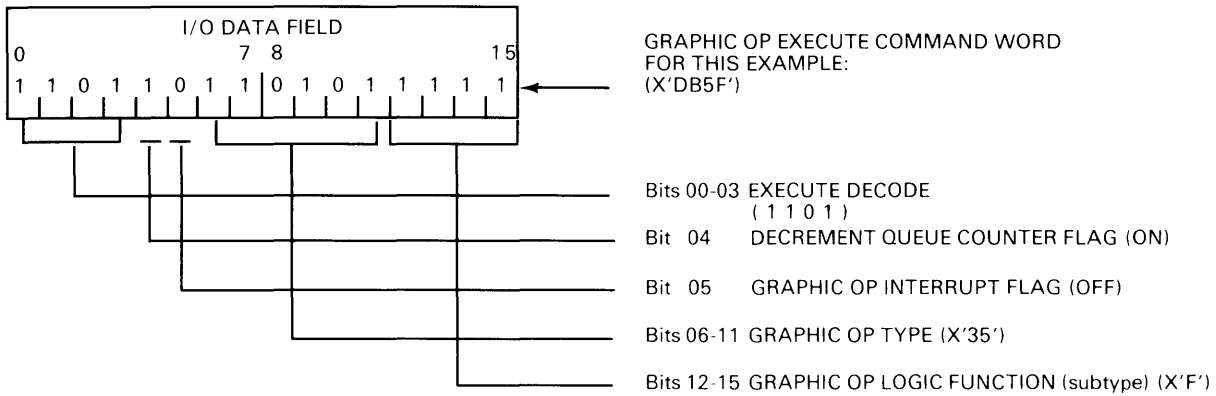
Example: Set Video Off

Queue Address	Queue Data	Action By Queue Processing
Que Ptr	X'D3F0'	Execute set video off - Disable the video data line.

Graphic Operation Queue Load

The queue load data example shown for each graphic operation type demonstrates what a queue load might look like for that operation. For instance, to draw a line in BAMDA from the decimal coordinates (X1,Y1 = 100,200) to (X2,Y2 = 512,384) with all pels on the line set to 1, you could use the graphic operation type X'35' (Vector Draw) with the following queue load data.

System Addr. (low 24 bits)	Queue Load Data	Parameter Name/Decimal Value
X'D9F7FE'	X'1064'	X1 / 100
X'D9F7FC'	X'00C8'	Y1 / 200
X'D9F7FA'	X'5200'	X2 / 512
X'D9F7F8'	X'6180'	Y2 / 384
X'D9F7F6'	X'DB5F'	Execute Vector Draw subtype F (Replace On) with dec Q flag on



To execute the graphic operation, write the queue pointer register with the queue pointer value corresponding to the queue load starting address (X'FBFF' corresponds to X'D9F7FE') and then increment the queue counter. Do a I/O memory write X'FBFF' to I/O channel memory address X'D9F806' (queue pointer register location) and then a write to I/O address X'0D14' (increment queue counter). The complete operation to write the queue load and execute it requires a total of six memory write operations and one I/O write.

Graphic Operation Queue Load Format Table

Definitions of symbols used in this table are:

- xxx = 10 bit hex data (max value = X'3FF' = 1023)
- 0xx = 8 bit hex data (max value = X'0FF = 255)
- s = subtype
- lre = lower right exclusive point
- X1, Y1 represent the start point coordinates of a line or the upper left corner point of a rectangle.
- X2, Y2 represent the end point coordinates of a line or the lower right corner point of a rectangle.
- DX, DY represent (X2-X1) and (Y2-Y1) respectively.

Type	Graphic Operation Name and Queue Load Data Example
2C	RELATIVE DRAW WITH ENDING NULL Data Reg Parameter 1xxx R1 X1 0xxx R0 Y1 5xxx R5 DX 6xxx R6 DY D2Cs Execute command
2D	RELATIVE DRAW Data Reg Parameter 1xxx R1 X1 0xxx R0 Y1 5xxx R5 DX 6xxx R6 DY D2Ds Execute command

Type	Graphic Operation Name and Queue Load Data Example
2E	RELATIVE MOVE TO Data Reg Parameter 2xxx R2 DX 6xxx R6 DY D2Es Execute command
2F	RECTANGULAR FILL (HOR/VER OPTIMIZED) Data Reg Parameter 7xxx R7 XD lre 6xxx R6 YD lre 9xxx R9 DX + 1 8xxx R8 DY + 1 D2Fs Execute command
30	RECTANGULAR COPY/MERGE Data Reg Parameter 5xxx R5 Xsource lre 4xxx R4 Ysource lre 7xxx R7 Xdest lre Axxx RA Ydest lre 9xxx R9 DX + 1 0xxx R0 DY + 1 D30s Execute command

Type	Graphic Operation Name and Queue Load Data Example
31	RECTANGULAR COPY/MERGE +90 DEGREE ROTATE Data Reg Parameter 5xxx R5 Xsource lre 4xxx R4 Ysource lre Axxx RA Xdest lre 7xxx R7 Ydest lre 9xxx R9 DX + 1 0xxx R0 DY + 1 D31s Execute command
32	RECTANGULAR COPY/MERGE -90 DEGREE ROTATE Data Reg Parameter 4xxx R4 Xsource lre 5xxx R5 Ysource lre 7xxx R7 Xdest lre Axxx RA Ydest lre 0xxx R0 DX + 1 9xxx R9 DY + 1 D32s Execute command
33	RECTANGULAR COPY/MERGE 180 DEGREES X AXIS SYMMETRY Data Reg Parameter 5xxx R5 Xsource lre 4xxx R4 Ysource lre 7xxx R7 Xdest lre Axxx RA (Ydest-(DY + 1)-1) lre 9xxx R9 DX + 1 0xxx R0 DY + 1 D33s Execute command

Type	Graphic Operation Name and Queue Load Data Example
34	RECTANGULAR COPY/MERGE 180 DEGREES Y AXIS SYMMETRY Data Reg Parameter 4xxx R4 Xsource lre 5xxx R5 Ysource lre Axxx RA (Xdest-(DX + 1)-1) lre 7xxx R7 Ydest lre 0xxx R0 DX + 1 9xxx R9 DY + 1 D34s Execute command
35	VECTOR DRAW Data Reg Parameter 1xxx R1 X1 0xxx R0 Y1 5xxx R5 X2 6xxx R6 Y2 D35s Execute command
35	POLYLINE DRAW (CONNECTED CONTINUATION) Data Reg Parameter 5xxx R5 X2 6xxx R6 Y2 D35s Execute command

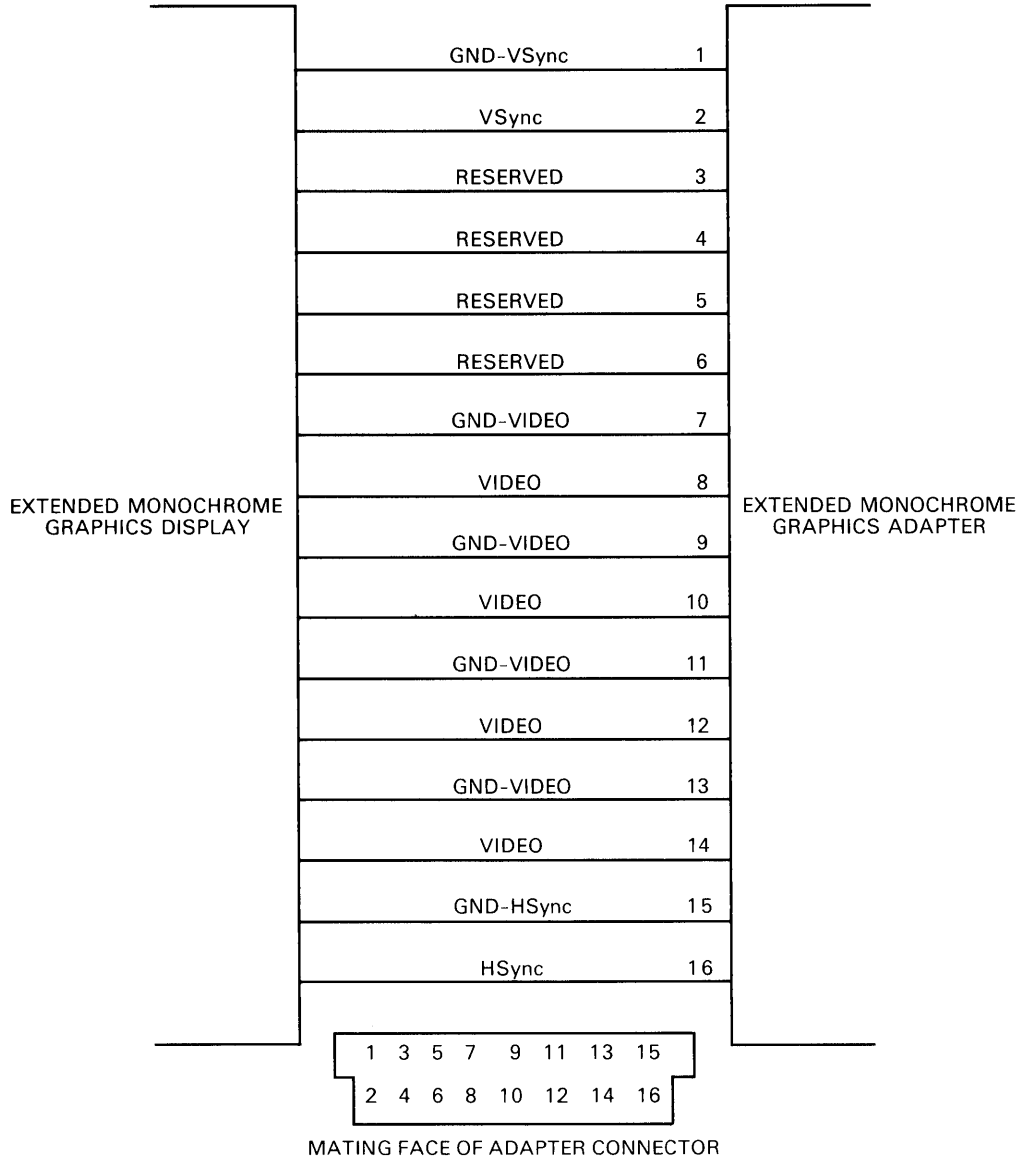
Type	Graphic Operation Name and Queue Load Data Example
36	VECTOR DRAW WITH ENDING NULL Data Reg Parameter 1xxx R1 X1 0xxx R0 Y1 5xxx R5 X2 6xxx R6 Y2 D36s Execute command
36	POLYLINE DRAW WITH ENDING NULL (CONNECTED CONTINUATION) Data Reg Parameter 5xxx R5 X2 6xxx R6 Y2 D36s Execute command
37	TDDMA CUT/PASTE FROM SYSTEM MEMORY Data Reg Parameter A0xx RA DMA addr HI(0-6) B0xx RB DMA addr MI(7-14) 40xx R4 DMA addr LO(15-22) 3xxx R3 Bit offset 2xxx R2 Skip amount 7xxx R7 Xdest lre 6xxx R6 Ydest lre 5xxx R5 DX + 1 0xxx R0 DY + 1 D37s Execute command

Type	Graphic Operation Name and Queue Load Data Example
38	TDDMA CUT/PASTE FROM SYSTEM MEMORY +90 DEGREE ROTATE Data Reg Parameter A0xx RA DMA addr HI(0-6) B0xx RB DMA addr MI(7-14) 40xx R4 DMA addr LO(15-22) 3xxx R3 Bit offset 2xxx R2 Skip amount 7xxx R7 Xdest lre 6xxx R6 (Xdest-(DX + 1)-1) lre 5xxx R5 DX + 1 0xxx R0 2's comp of (DX + 1) D38s Execute command
39	TDDMA CUT/PASTE FROM SYSTEM MEMORY 45 DEGREES DIAGONAL FLIP Data Reg Parameter A0xx RA DMA addr HI(0-6) B0xx RB DMA addr MI(7-14) 40xx R4 DMA addr LO(15-22) 3xxx R3 Bit offset 2xxx R2 Skip amount 7xxx R7 Xdest lre 6xxx R6 Xdest lre 5xxx R5 DY + 1 0xxx R0 DX + 1 D39s Execute command

Type	Graphic Operation Name and Queue Load Data Example
3A	TDDMA CUT/PASTE FROM SYSTEM MEMORY 45 DEGREES DIAGONAL FLIP Data Reg Parameter A0xx RA DMA addr HI(0-6) B0xx RB DMA addr MI(7-14) 40xx R4 DMA addr LO(15-22) 3xxx R3 Bit offset 2xxx R2 Skip amount 7xxx R7 Xdest lre 6xxx R6 (Ydest-(DY + 1)-1) lre 5xxx R5 DX + 1 0xxx R0 2's Comp of DY + 1 D3As Execute command
3B	BRANCH AND LINK QUEUE (one level deep) Data Reg Parameter A0xx RA BR addr HI(0-7) B0xx RB BR addr LO(8-15) D3Bs Execute command
3C	RETURN QUEUE (one level deep) Data D3C0 Execute command
3D	SCAN LINE LINK Data Reg Parameter A0xx RA Scan line complement D3D0 Execute command

Type	Graphic Operation Name and Queue Load Data Example
3E	SET VIDEO ON Data D3E0 Execute command
3F	SET VIDEO OFF Data D3F0 Execute command

Connector Specifications



IBM

Personal Computer Hardware Reference Library

512 KB Memory Expansion Option

Contents

Notes:

Description

This adapter has 36 RAM modules (128K x 1) for a total capacity of 512Kb.

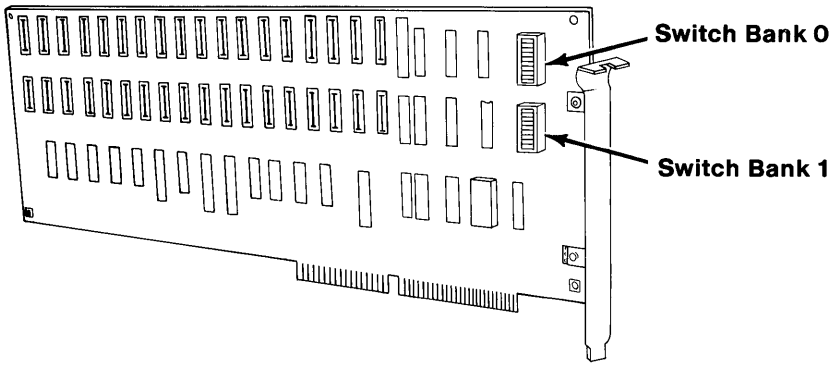
Memory Cycles

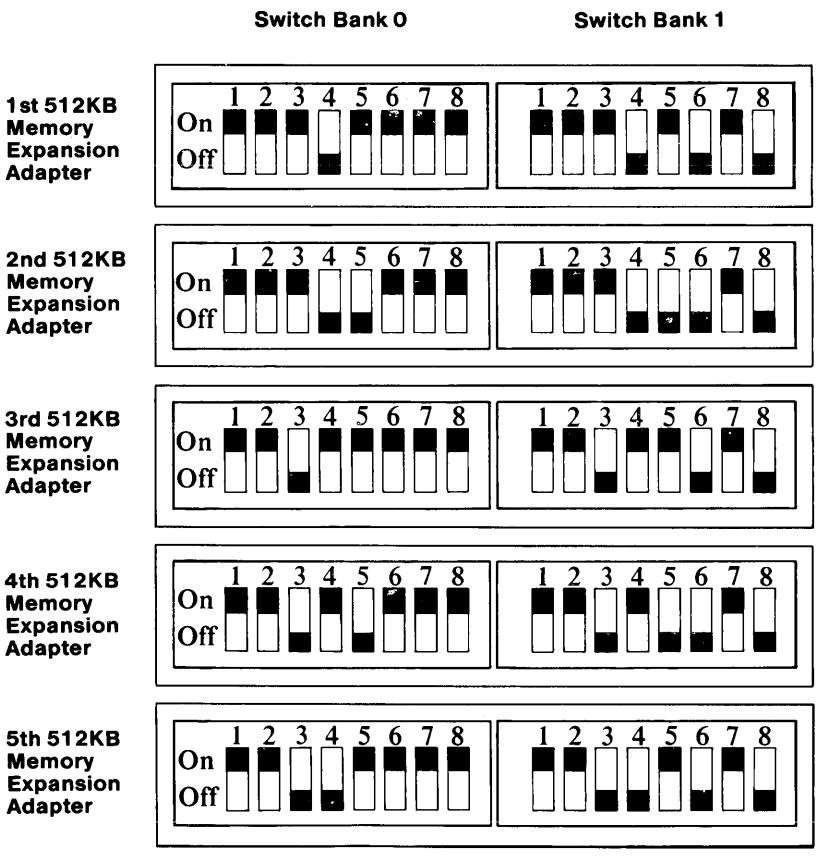
MEMR and MEMW commands require a 1-wait-state, 3-clock memory cycle. Data moves as a byte (8 data bits and 1 parity bit) or as a word (16 data bits and 2 parity bits) and is parity-checked on the adapter. A parity error causes an I/O channel check (non-maskable interrupt) to the system.

Memory Address Switches

There are two banks of memory address switches on each memory adapter. These switches are set to values for the first, second, third, etc. memory adapter in the system. The following figure shows the switch configuration for each adapter.

The first memory expansion adapter must start at address space hex 100000. If more than one adapter is installed, no gaps between memory are allowed. All expansion memory must be one contiguous block starting at address hex 100000.


Switch Bank 0
Switch Bank 1


Switch Bank 0
Switch Bank 1
1st 512KB Memory Expansion Adapter
2nd 512KB Memory Expansion Adapter
3rd 512KB Memory Expansion Adapter
4th 512KB Memory Expansion Adapter
5th 512KB Memory Expansion Adapter
On
Off
1 2 3 4 5 6 7 8

I/O Channel Check

When the I/O channel check occurs, a non-maskable interrupt (NMI) results, and the status bits determine the source (one status bit is I/O channel check and the other is system-board parity check). Writing to the failing card will clear the status bit.

Specifications

Voltage Tolerances

The maximum variation of the +5 Vdc is ±5% at the adapter pins.

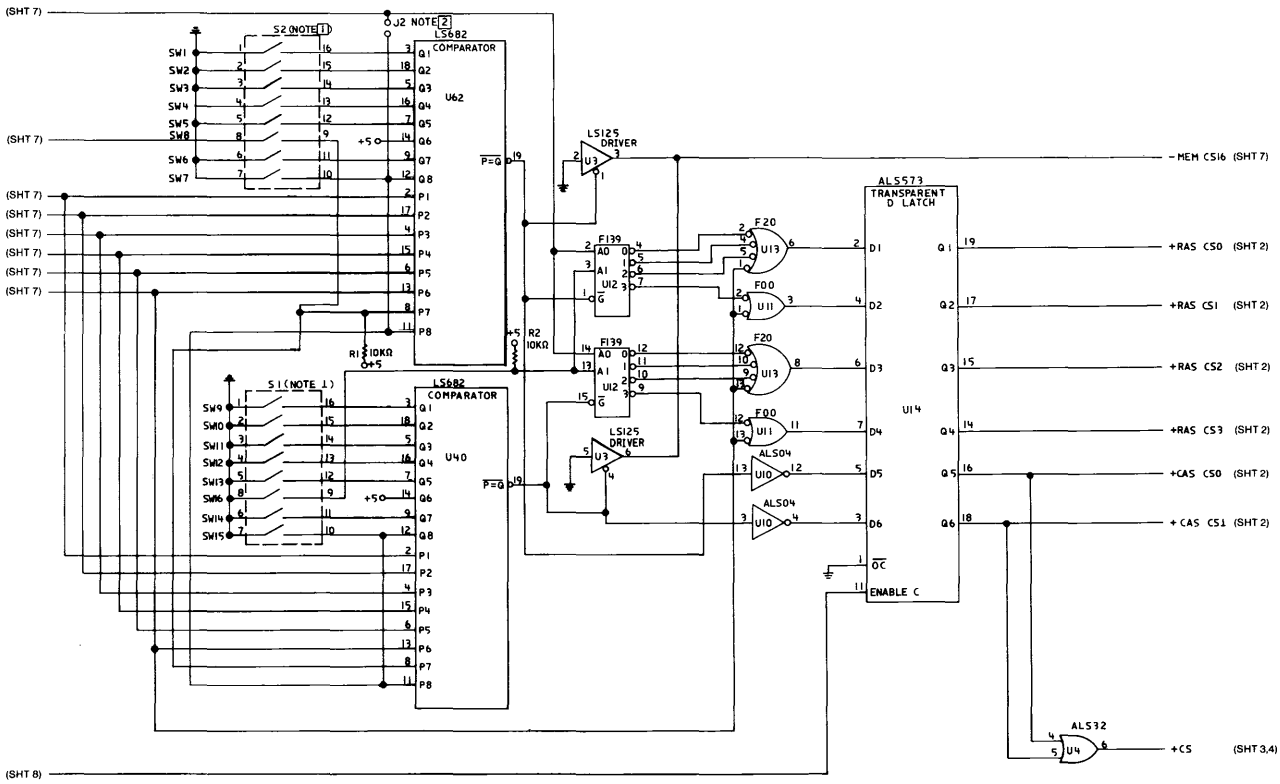
Power Dissipation

The +5-Vdc power used by the adapter is a maximum of 5.25 watts, and the maximum current used is 1 ampere.

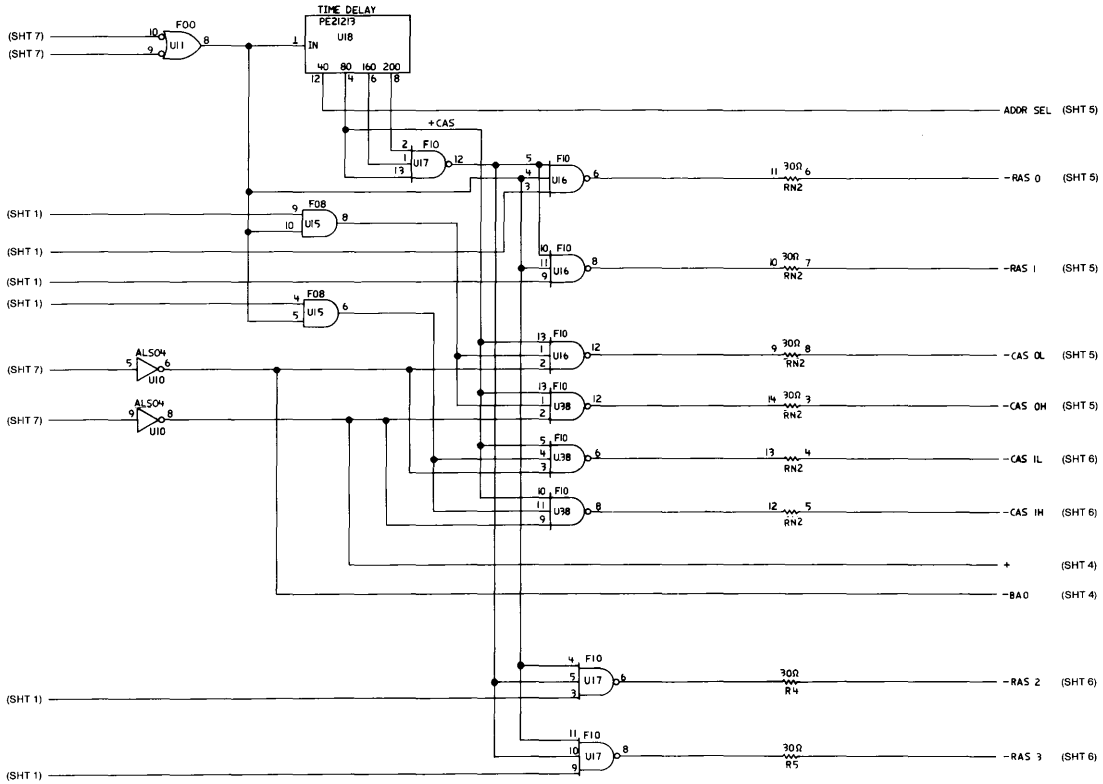
Temperature Variation

The adapter will operate between 10 and 50 degrees Celsius (50 and 122 degrees Farenheit).

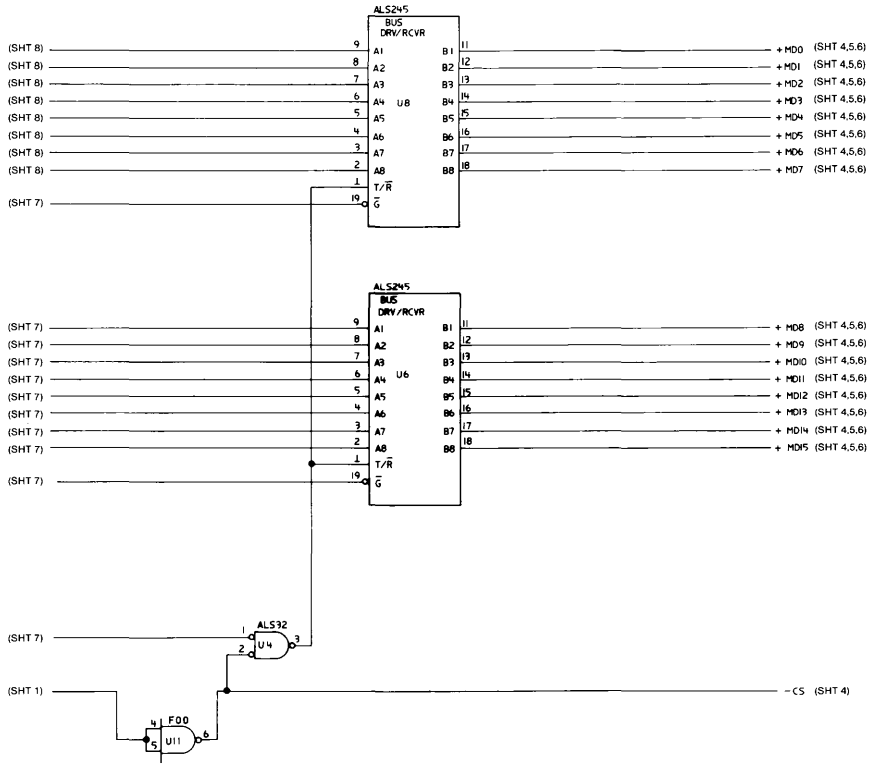
Notes:



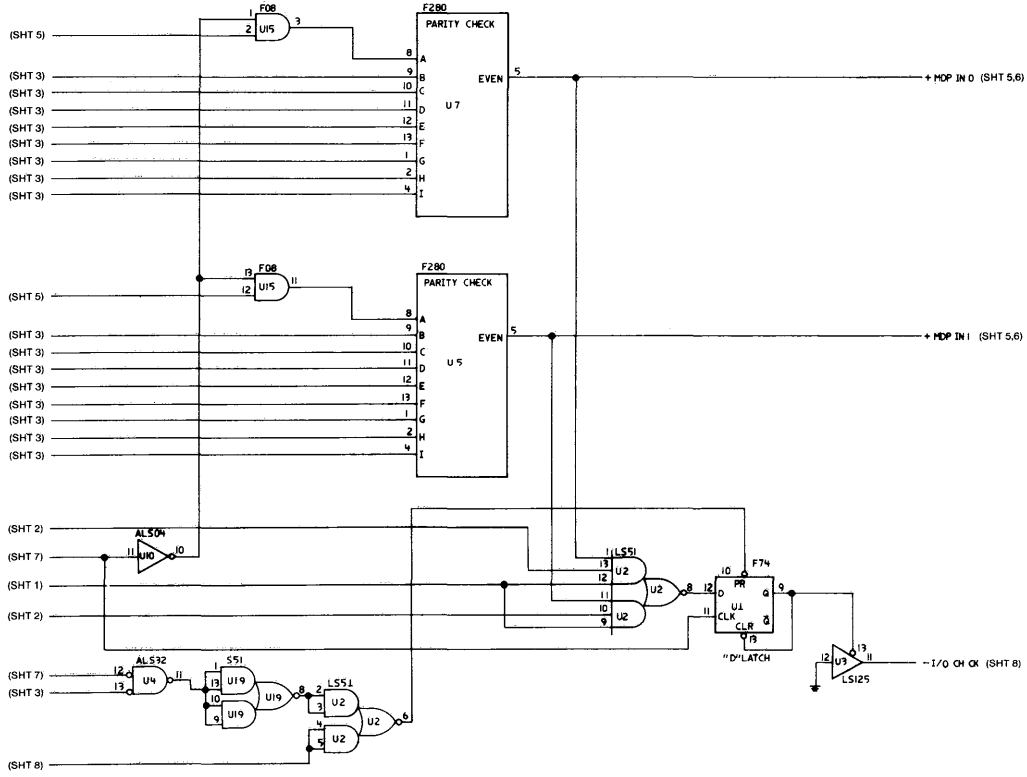
512 KB Memory Expansion Option (Sheet 1 of 8)



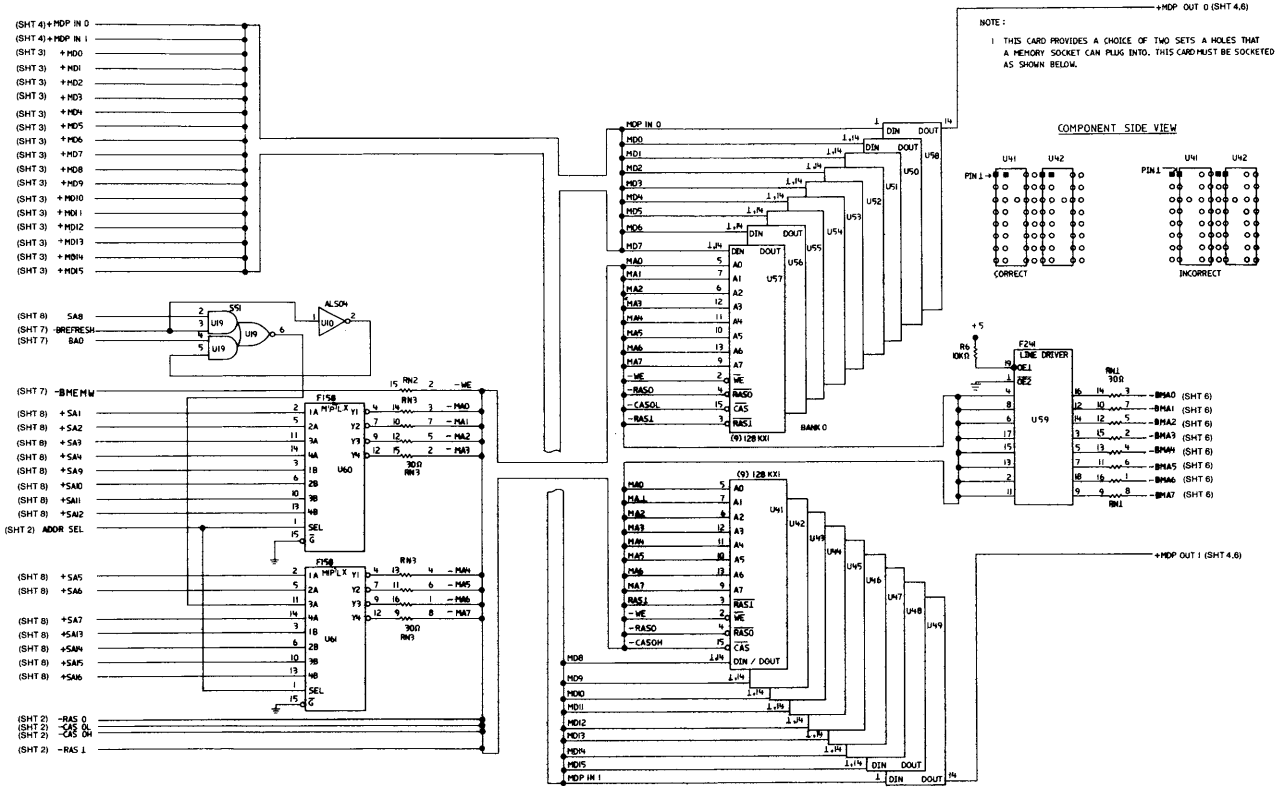
512 KB Memory Expansion Option (Sheet 2 of 8)



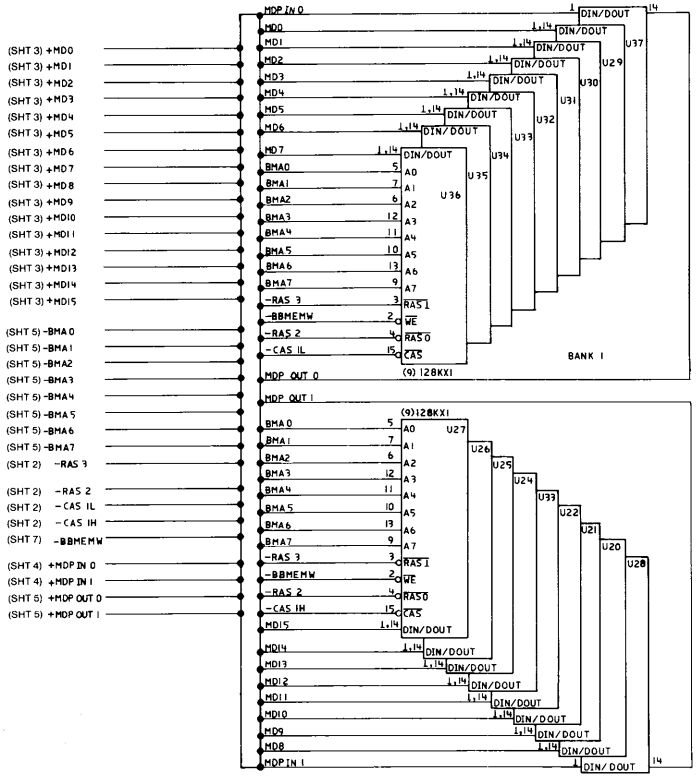
512 KB Memory Expansion Option (Sheet 3 of 8)



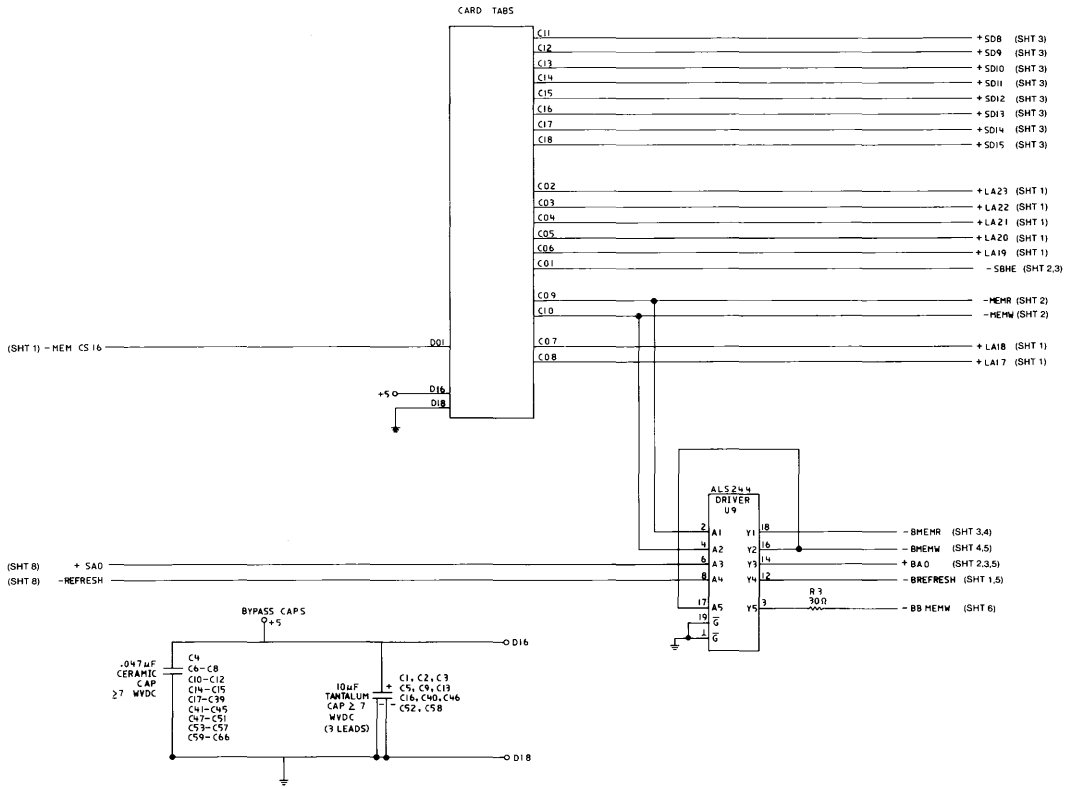
512 KB Memory Expansion Option (Sheet 4 of 8)



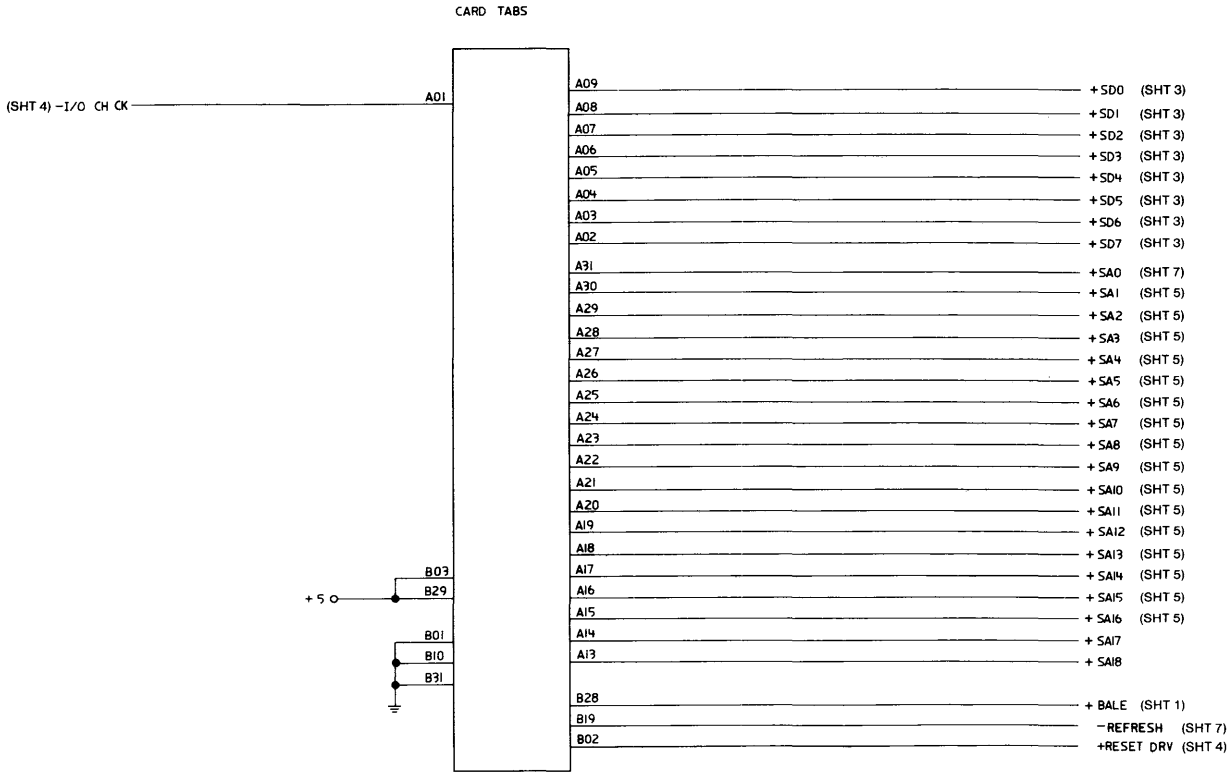
512 KB Memory Expansion Option (Sheet 5 of 8)



512 KB Memory Expansion Option (Sheet 6 of 8)



512 KB Memory Expansion Option (Sheet 7 of 8)



512 KB Memory Expansion Option (Sheet 8 of 8)

IBM

Personal Computer Hardware Reference Library

Monochrome Display and Printer Adapter

Contents

Introduction

The IBM Monochrome Display and Printer Adapter has two functions. The first is to provide an interface to the IBM Monochrome Display. The second is to provide a parallel interface for the IBM Printers. We will discuss this adapter by function.

Monochrome Display Adapter Function

Description

The IBM Monochrome Display and Printer Adapter is designed around the Motorola 6845 CRT Controller module. There are 4K bytes of RAM on the adapter that are used for the display buffer. This buffer has two ports to which the system unit's microprocessor has direct access. No parity is provided on the display buffer.

Two bytes are fetched from the display buffer in 553 ns, providing a data rate of 1.8M bytes/second.

The adapter supports 256 different character codes. An 8K-byte character generator contains the fonts for the character codes. The characters, values, and screen characteristics are given in "Of Characters, Keystrokes, and Colors" in your *Technical Reference* system manual.

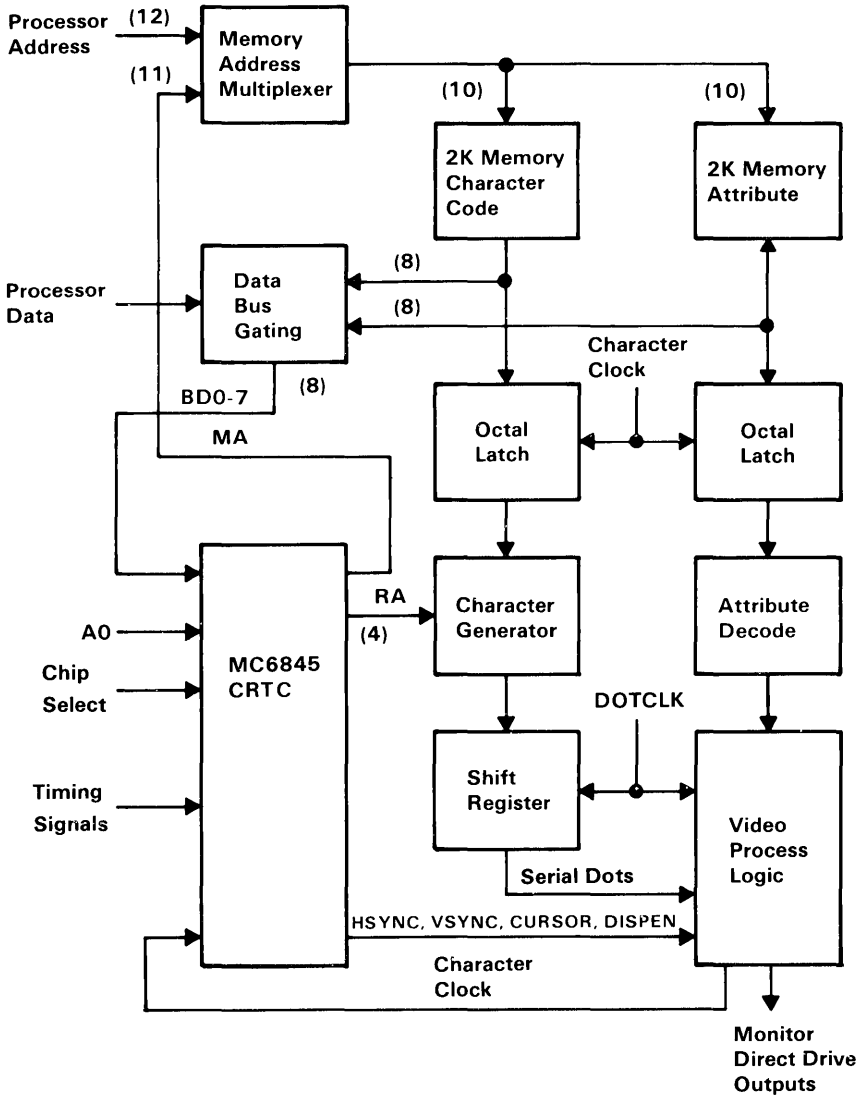
This adapter, when used with a display containing P39 phosphor, does not support a light pen.

Where possible, only one low-power Schottky (LS) load is present on any I/O slot. Some of the address bus lines have two LS loads. No signal has more than two LS loads.

Characteristics of the adapter are:

- Supports 80-character by 25-line screen
- Has direct-drive output
- Supports 9-PEL by 14-PEL character box
- Supports 7-PEL by 9-PEL character
- Has 18-kHz monitor
- Has character attributes

The following is a block diagram of the monochrome display adapter portion of the IBM Monochrome Display and Printer Adapter.



IBM Monochrome Display Adapter Block Diagram

4 Monochrome Adapter

Programming Considerations

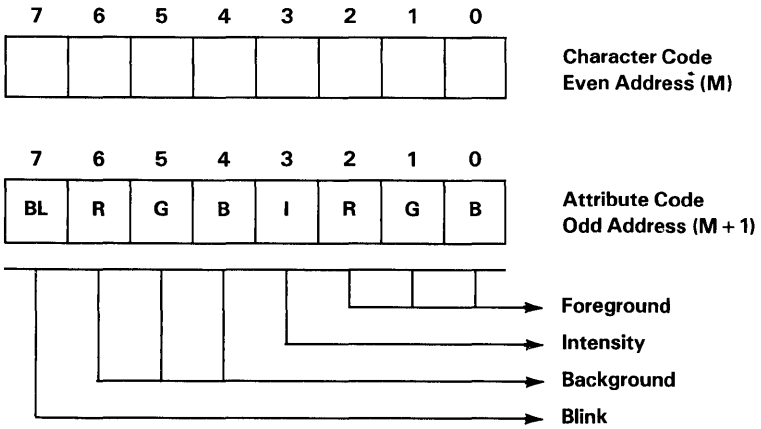
The following table summarizes the 6845 controller module's internal data registers, their functions, and their parameters. For the IBM Monochrome Display, the values must be programmed into the 6845 to ensure proper initialization of the display.

Register Number	Register File	Program Unit	IBM Monochrome Display (Address in hex)
R0	Horizontal Total	Characters	61
R1	Horizontal Displayed	Characters	50
R2	Horizontal Sync Position	Characters	52
R3	Horizontal Sync Width	Characters	F
R4	Vertical Total	Character Rows	19
R5	Vertical Total Adjust	Scan Line	6
R6	Vertical Displayed	Character Row	19
R7	Vertical Sync Position	Character Row	19
R8	Interlace Mode	---	02
R9	Maximum Scan Line Address	Scan Line	D
R10	Cursor Start	Scan Line	B
R11	Cursor End	Scan Line	C
R12	Start Address (H)	---	00
R13	Start Address (L)	---	00
R14	Cursor (H)	---	00
R15	Cursor (L)	---	00
R16	Reserved	---	--
R17	Reserved	---	--

To ensure proper initialization, the first command issued to the IBM Monochrome Display and Printer Adapter must be sent to the CRT control port 1 (hex 3B8), and must be a hex 01, to set the high-resolution mode. If this bit is not set, the system unit's microprocessor's access to the adapter must never occur. If the high-resolution bit is not set, the system unit's microprocessor will stop running.

System configurations that have both an IBM Monochrome Display and Printer Adapter, and an IBM Color/Graphics Monitor Adapter, must ensure that both adapters are properly initialized after a power-on reset. Damage to either display may occur if not properly initialized.

The IBM Monochrome Display and Printer Adapter supports 256 different character codes. In the character set are alphanumerics and block graphics. Each character in the display buffer has a corresponding character attribute. The character code must be an even address, and the attribute code must be an odd address in the display buffer.



The adapter decodes the character attribute byte as defined above. The blink and intensity bits may be combined with the foreground and background bits to further enhance the character attribute functions listed below:

Background R G B	Foreground R G B	Function
0 0 0	0 0 0	Non-Display
0 0 0	0 0 1	Underline
0 0 0	1 1 1	White Character/Black Background
1 1 1	0 0 0	Reverse Video

The 4K display buffer supports one screen of the 25 rows of 80 characters, plus a character attribute for each display character. The starting address of the buffer is hex B0000. The display buffer can be read using direct memory access (DMA); however, at least one wait state will be inserted by the system unit's microprocessor. The duration of the wait state will vary, because the microprocessor/monitor access is synchronized with the character clock on this adapter.

Interrupt level 7 is used on the parallel interface. Interrupts can be enabled or disabled through the printer control port. The interrupt is a high-level active signal.

The following table breaks down the functions of the I/O address decode for the adapter. The I/O address decode is from hex 3B0 through hex 3BF. The bit assignment for each I/O address follows:

I/O Register Address	Function
3B0	Not Used
3B1	Not Used
3B2	Not Used
3B3	Not Used
3B4	6845 Index Register
3B5	6845 Data Register
3B6	Not Used
3B7	Not Used
3B8	CRT Control Port 1
3B9	Reserved
3BA	CRT Status Port
3BB	Reserved
3BC	Parallel Data Port
3BD	Printer Status Port
3BE	Printer Control Port
3BF	Not Used

I/O Address and Bit Map

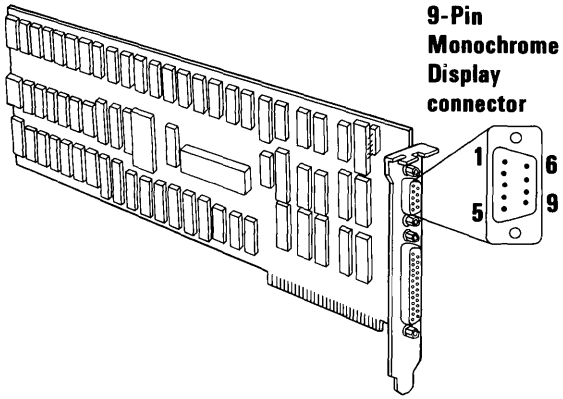
Bit Number	Function
0	+ High Resolution Mode
1	Not Used
2	Not Used
3	+ Video Enable
4	Not Used
5	+ Enable Blink
6,7	Not Used

6845 CRT Control Port 1 (Hex 3B8)

Bit Number	Function
0	+ Horizontal Drive
1	Reserved
2	Reserved
3	+ Black/White Video

6845 CRT Status Port (Hex 3BA)

Specifications



At Standard TTL Levels

IBM Monochrome Display			IBM Monochrome Display and Printer Adapter
	Ground	1	
	Ground	2	
	Not Used	3	
	Not Used	4	
	Not Used	5	
←	+ Intensity	6	
←	+ Video	7	
←	+ Horizontal	8	
←	– Vertical	9	

Note: Signal voltages are 0.0 to 0.6 Vdc at down level and + 2.4 to 3.5 Vdc at high level.

Connector Specifications

Printer Adapter Function

Description

The printer adapter portion of the IBM Monochrome Display and Printer Adapter is specifically designed to attach printers with a parallel-port interface, but it can be used as a general input/output port for any device or application that matches its input/output capabilities. It has 12 TTL-buffer output points, which are latched and can be written and read under program control using the microprocessor In or Out instruction. The adapter also has five steady-state input points that may be read using the microprocessor's In instructions.

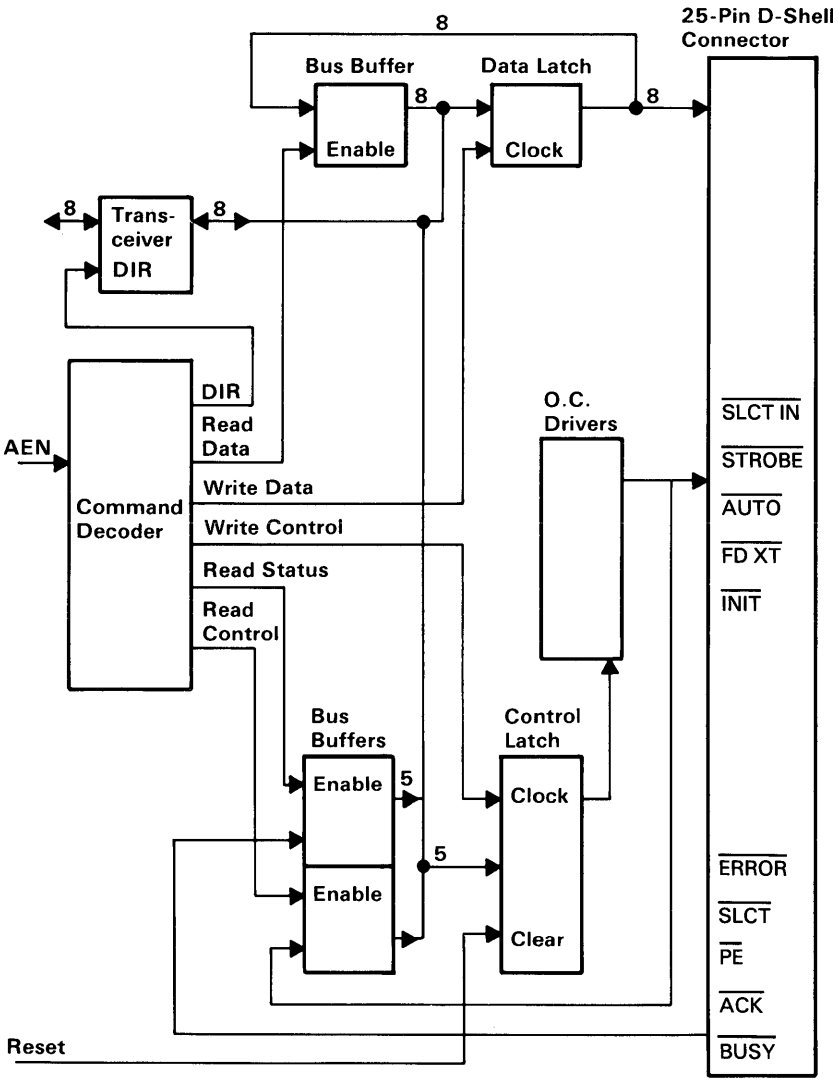
In addition, one input can also be used to create a microprocessor interrupt. This interrupt can be enabled and disabled under program control. A reset from the power-on circuit is also ORed with a program output point, allowing a device to receive a 'power-on reset' when the system unit's microprocessor is reset.

The input/output signals are made available at the back of the adapter through a right-angle, printed-circuit-board-mounted, 25-pin, D-shell connector. This connector protrudes through the rear panel of the system unit or expansion unit, where a cable may be attached.

When this adapter is used to attach a printer, data or printer commands are loaded into an 8-bit, latched, output port, and the strobe line is activated, writing data to the printer. The program then may read the input ports for printer status indicating when the next character can be written, or it may use the interrupt line to indicate "not busy" to the software.

The output ports may also be read at the card's interface for diagnostic loop functions. This allows faults to be isolated to the adapter or the attaching device.

The following is a block diagram of the printer adapter portion of the Monochrome Display and Printer Adapter.



Printer Adapter Block Diagram

Programming Considerations

The printer adapter portion of the IBM Monochrome Display and Printer Adapter responds to five I/O instructions: two output and three input. The output instructions transfer data into 2 latches whose outputs are presented on pins of a 25-pin D-shell connector.

Two of the three input instructions allow the system unit's microprocessor to read back the contents of the two latches. The third allows the system unit's microprocessor to read the real-time status from a group of pins on the connector.

A description of each instruction follows.

IBM Monochrome Display & Printer Adapter			
Output to address hex 3BC			
Bit 7 Pin 9	Bit 6 Pin 8	Bit 5 Pin 7	Bit 4 Pin 6

The instruction captures data from the data bus and is present on the respective pins. Each of these pins is capable of sourcing 2.6 mA and sinking 24 mA.

It is essential that the external device does not try to pull these lines to ground.

IBM Monochrome Display & Printer Adpater	
Output to address hex 3BE	
	Bit 4 IRQ Enable

This instruction causes the latch to capture the five least significant bits of the data bus. The four least significant bits present their outputs, or inverted versions of their outputs, to the

respective pins as shown in the previous figure. If bit 4 is written as a 1, the card will interrupt the system unit's microprocessor on the condition that pin 10 changes from high to low.

These pins are driven by open-collector drivers pulled to +5 Vdc through 4.7 kΩ resistors. They can each sink approximately 7 mA and maintain 0.8 volts down-level.

IBM Monochrome Display & Printer Adapter
Input from address hex 3BC

This instruction presents the system unit's microprocessor with data present on the pins associated with the output to hex 3BC. This should normally reflect the exact value that was last written to hex 3BC. If an external device should be driving data on these pins at the time of an input (in violation of usage ground rules), this data will be ORed with the latch contents.

IBM Monochrome Display & Printer Adapter
Input from address hex 3BD

This instruction presents the real-time status to the system unit's microprocessor from the pins as follows.

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Pin 11	Pin 10	Pin 12	Pin 13	Pin 15	—	—	—

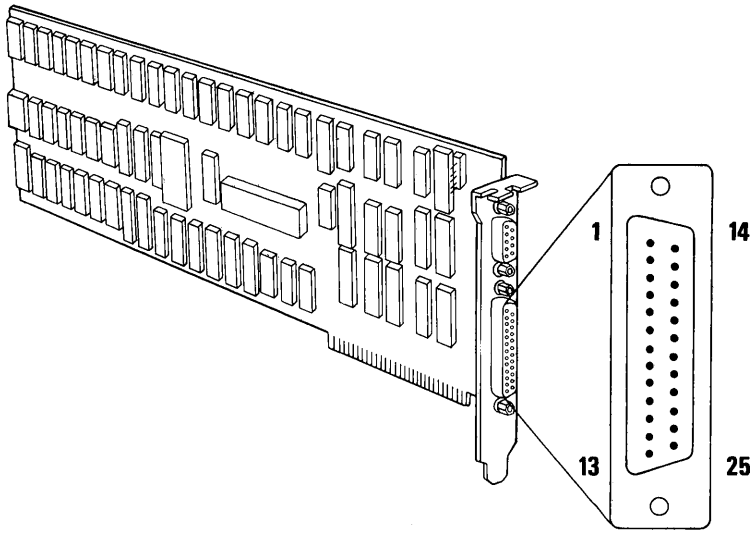
IBM Monochrome Display & Printer Adapter
Input from address hex 3BE

This instruction causes the data present on pins 1, 14, 16, 17, and the IRQ bit to be read by the system unit's microprocessor. In the absence of external drive applied to these pins, data read by the system unit's microprocessor will match data last written to hex 3BE in the same bit positions. Notice that data bits 0–2 are not included. If external drivers are dotted to these pins, that data will be ORed with data applied to the pins by the hex 3BE latch.

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
			IRQ Enable	$\overline{\text{Pin 17}}$	Pin 16	$\overline{\text{Pin 14}}$	$\overline{\text{Pin 1}}$
			Por = 0	Por = 1	Por = 0	Por = 1	Por = 1

These pins assume the states shown after a reset from the system unit's microprocessor.

Specifications

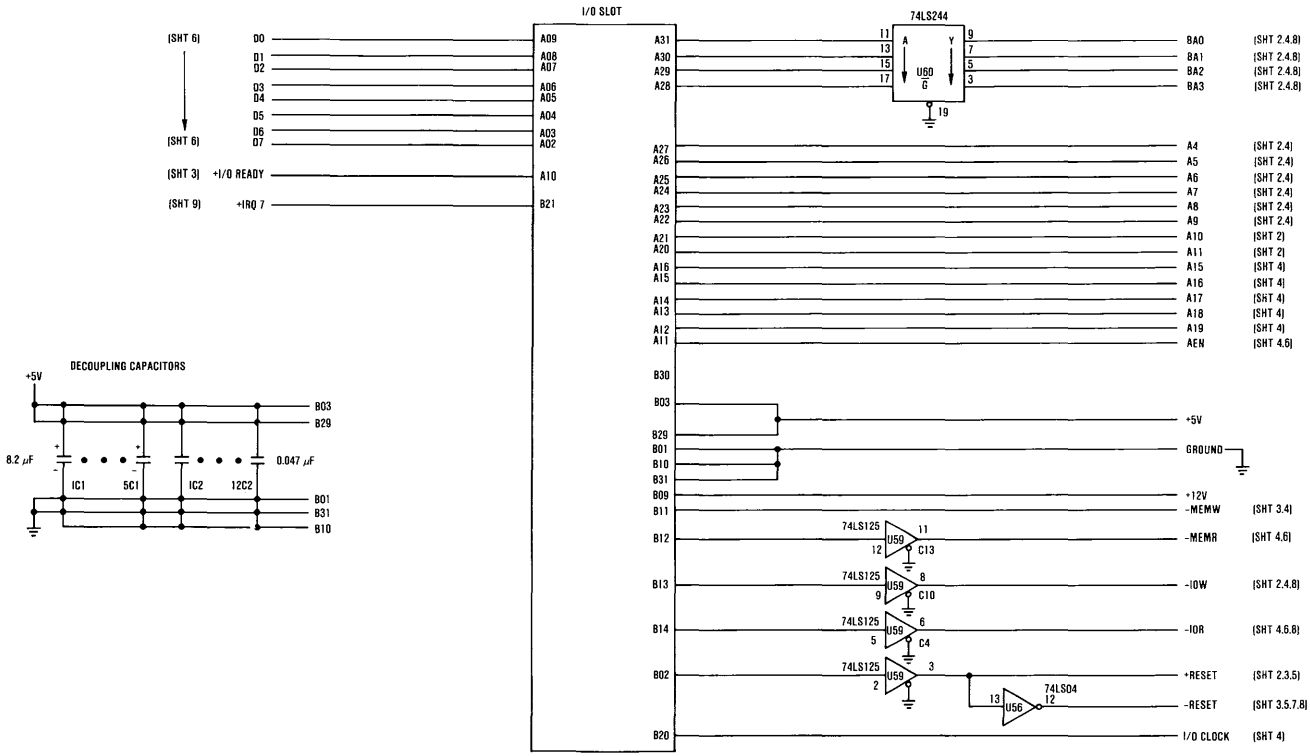


At Standard TTL Levels

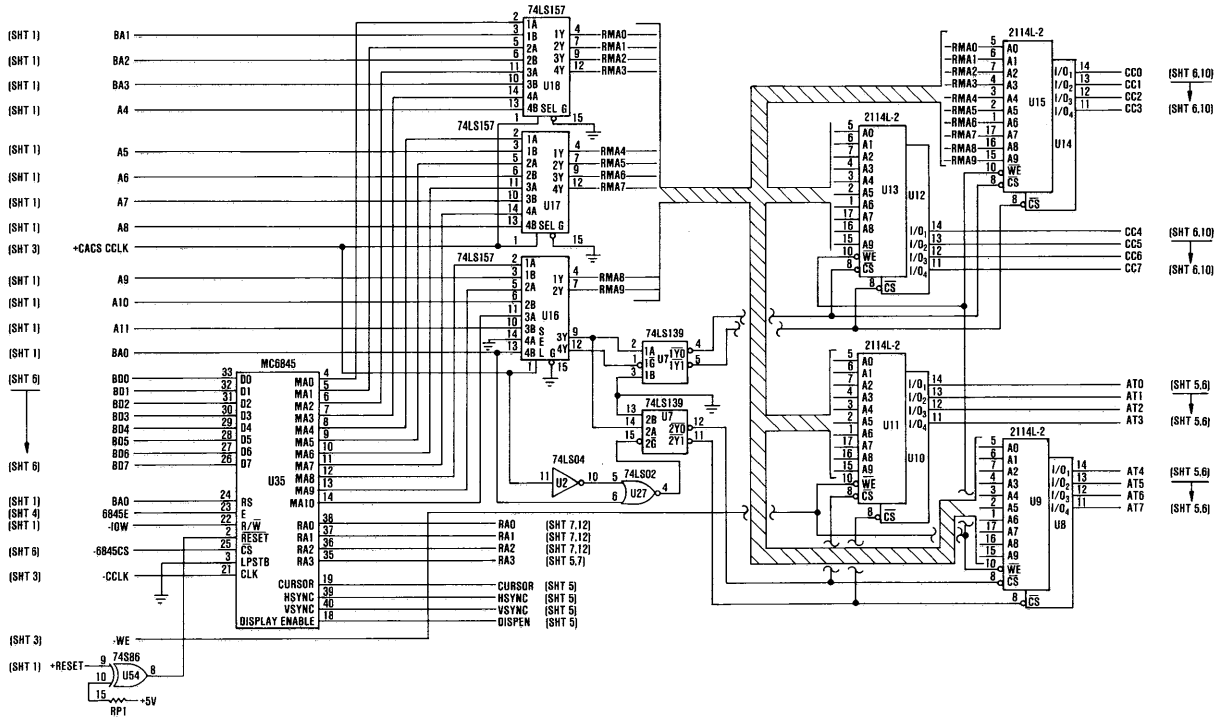
Signal Name	Adapter Pin Number	
– Strobe	1	
+ Data Bit 0	2	
+ Data Bit 1	3	
+ Data Bit 2	4	
+ Data Bit 3	5	
+ Data Bit 4	6	
+ Data Bit 5	7	IBM
+ Data Bit 6	8	Monochrome
+ Data Bit 7	9	Display and
– Acknowledge	10	Printer
+ Busy	11	Adapter
+ P. End (out of paper)	12	
+ Select	13	
– Auto Feed	14	
– Error	15	
– Initialize Printer	16	
– Select Input	17	
Ground	18-25	

Printer

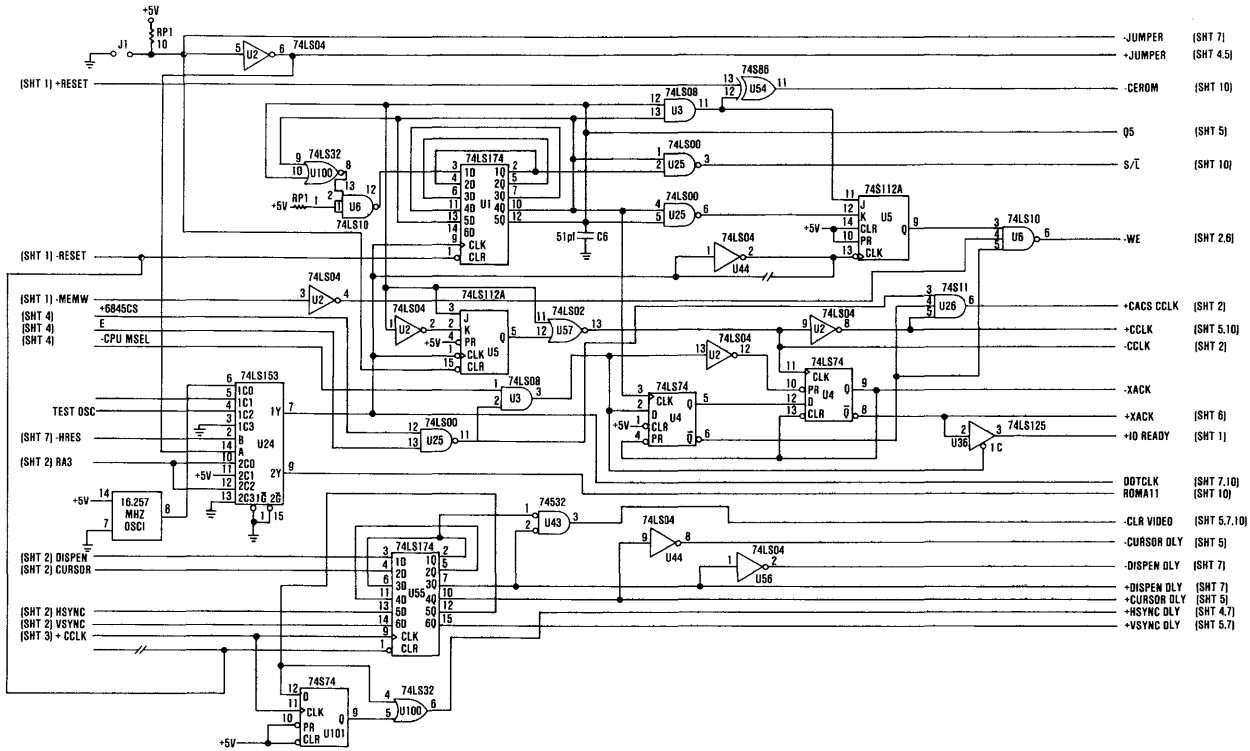
Connector Specifications



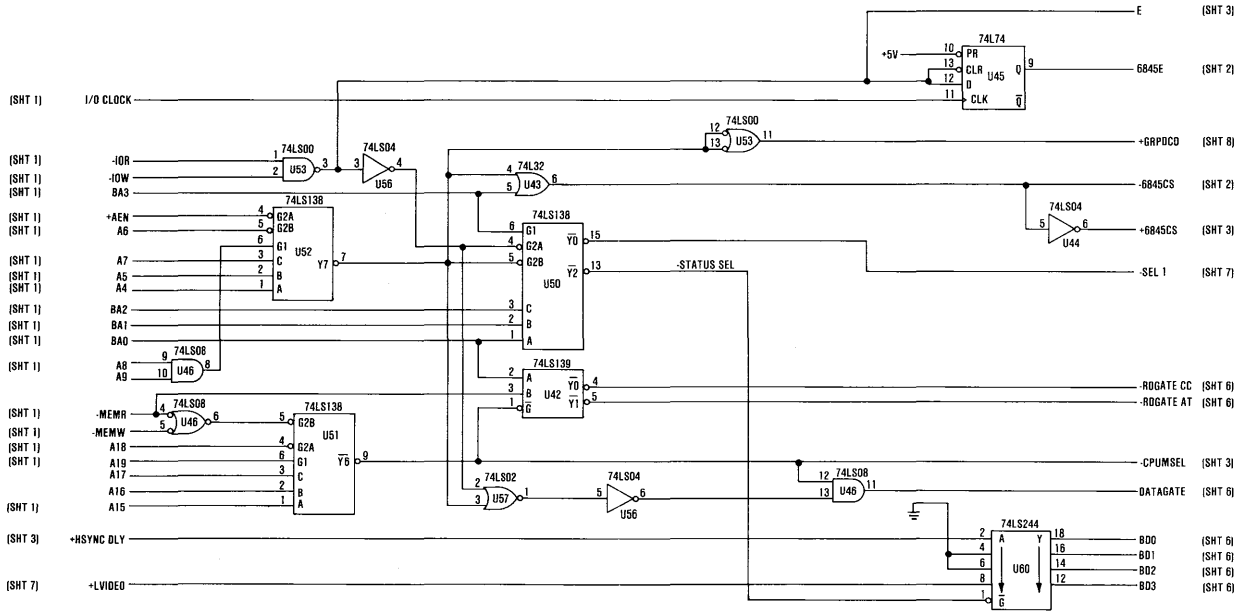
Monochrome Display Adapter (Sheet 1 of 10)



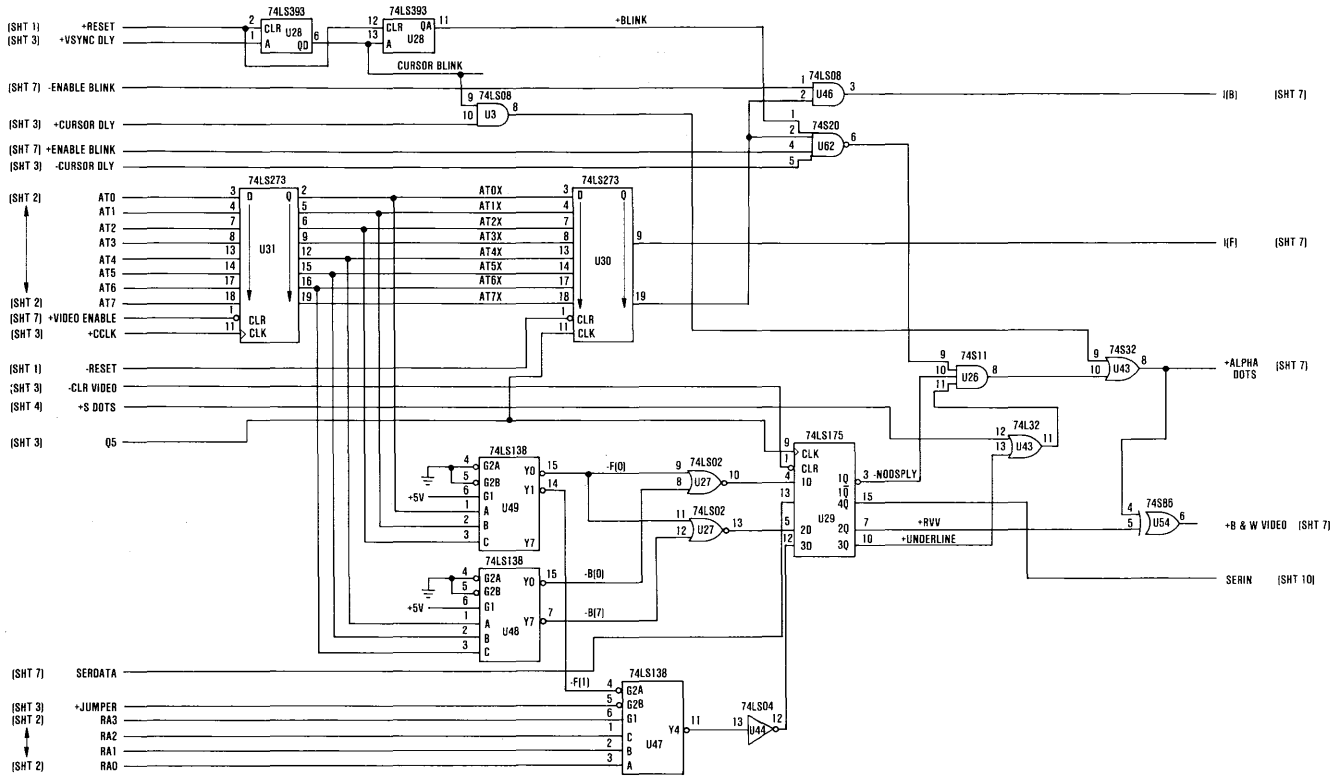
Monochrome Display Adapter (Sheet 2 of 10)



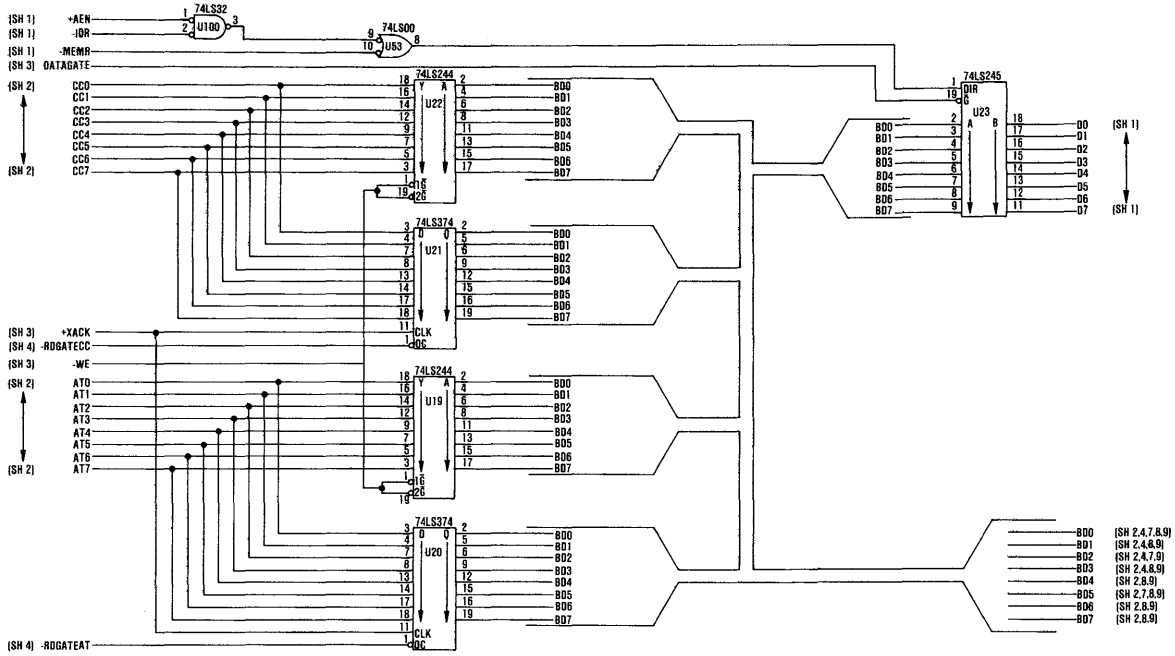
Monochrome Display Adapter (Sheet 3 of 10)



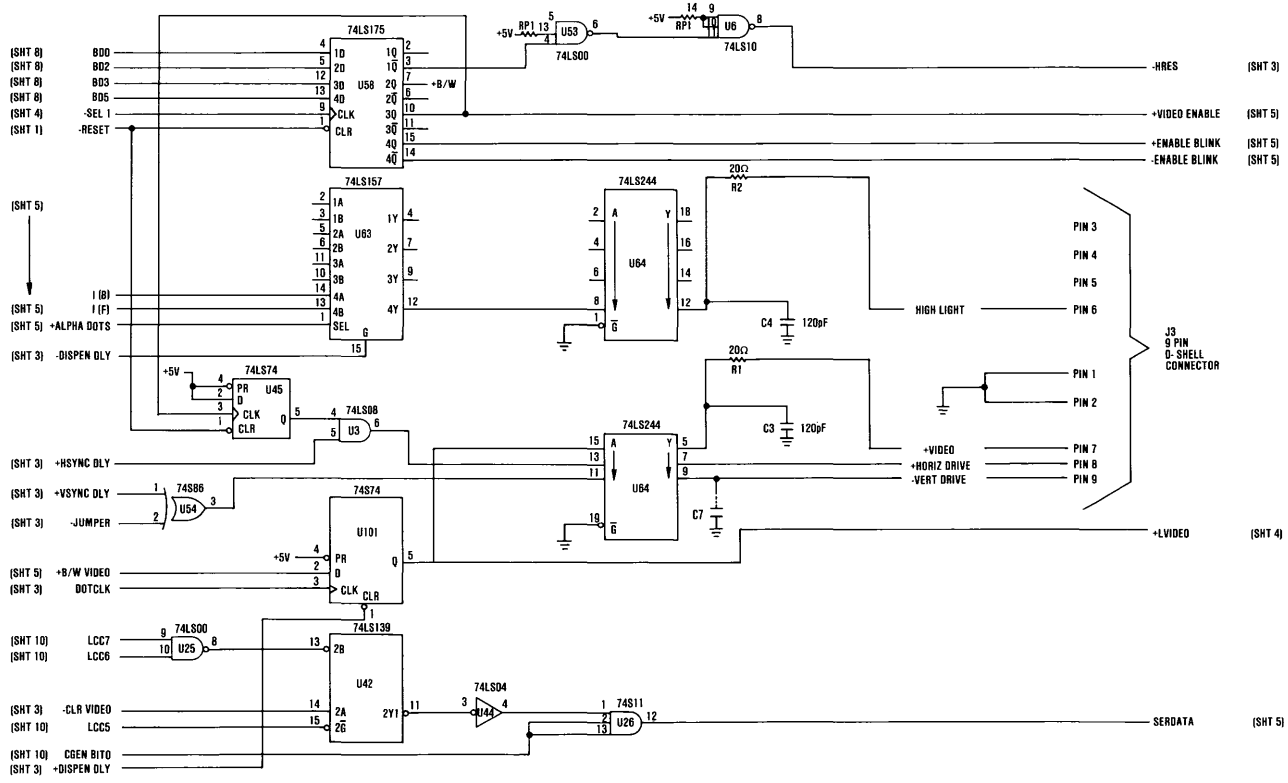
Monochrome Display Adapter (Sheet 4 of 10)



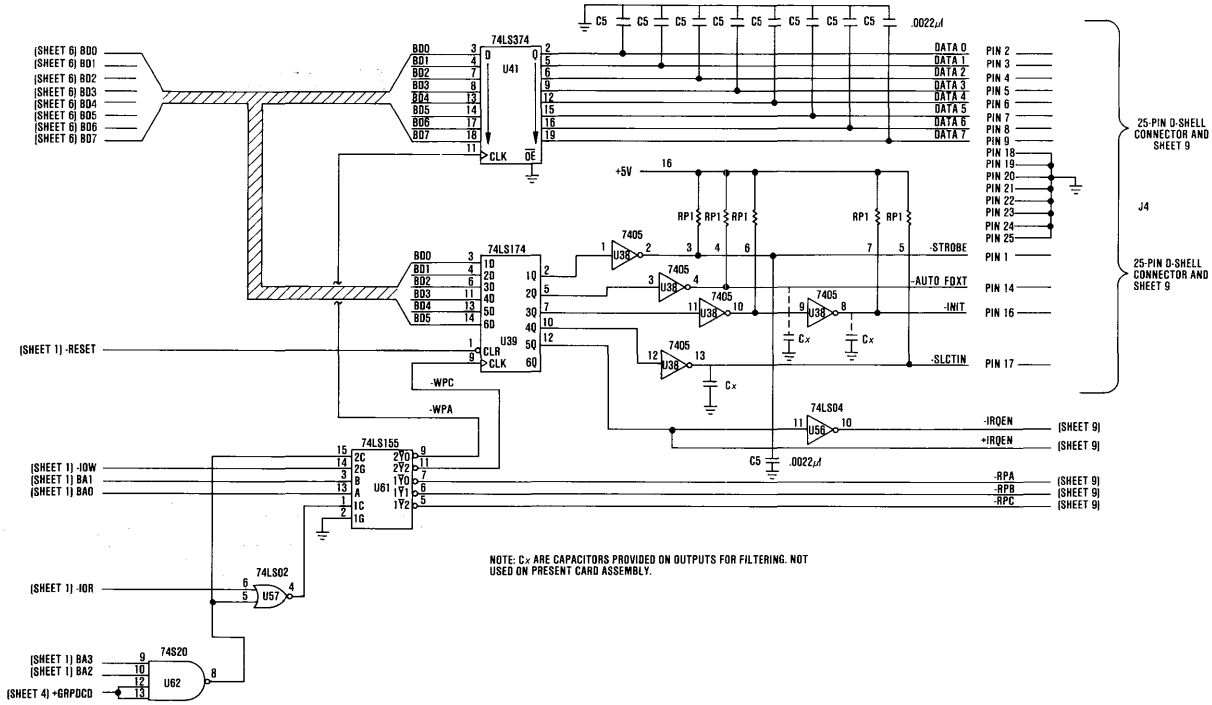
Monochrome Display Adapter (Sheet 5 of 10)



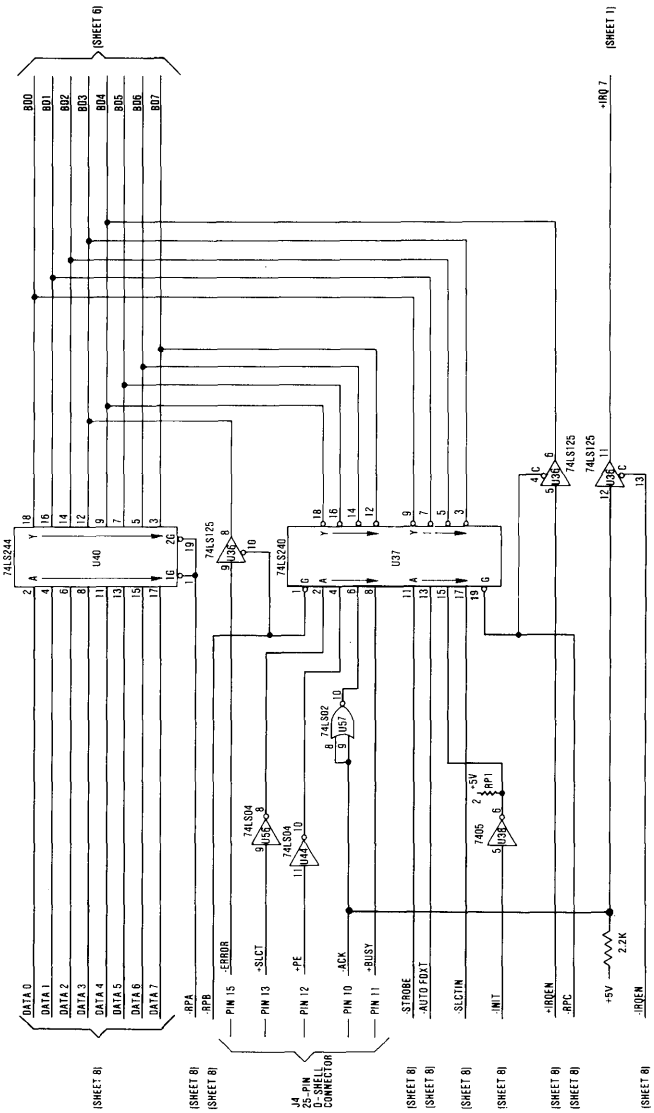
Monochrome Display Adapter (Sheet 6 of 10)



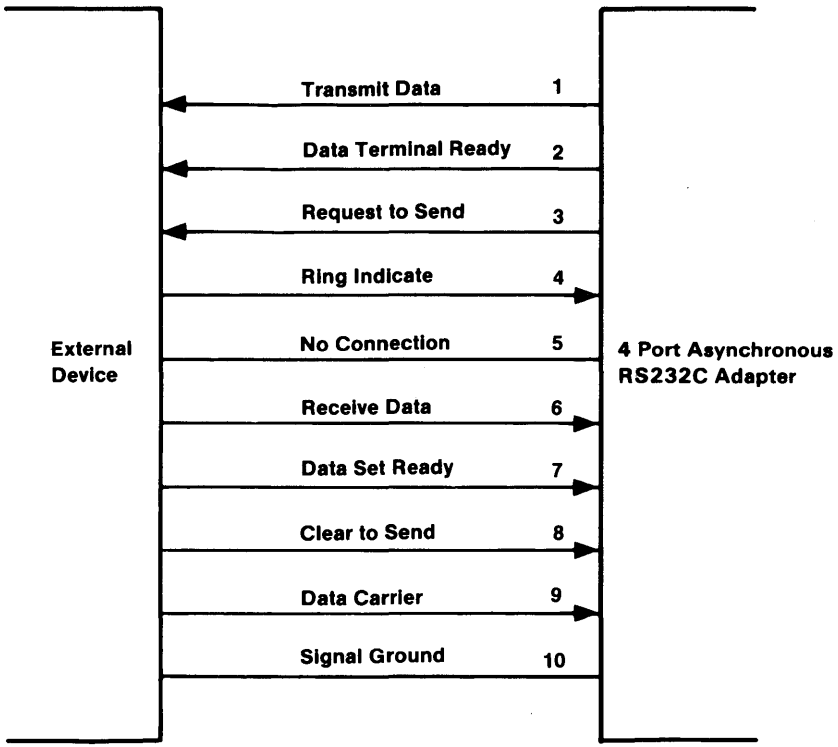
Monochrome Display Adapter (Sheet 7 of 10)



Monochrome Display Adapter (Sheet 8 of 10)



Monochrome Display Adapter (Sheet 9 of 10)


External Device
Transmit Data 1
Data Terminal Ready 2
Request to Send 3
Ring Indicate 4
No Connection 5
Receive Data 6
Data Set Ready 7
Clear to Send 8
Data Carrier 9
Signal Ground 10
4 Port Asynchronous RS232C Adapter

IBM

Personal Computer
Hardware Reference
Library

Serial/Parallel Adapter

Contents

Notes:

IBM Personal Computer AT Serial/Parallel Adapter

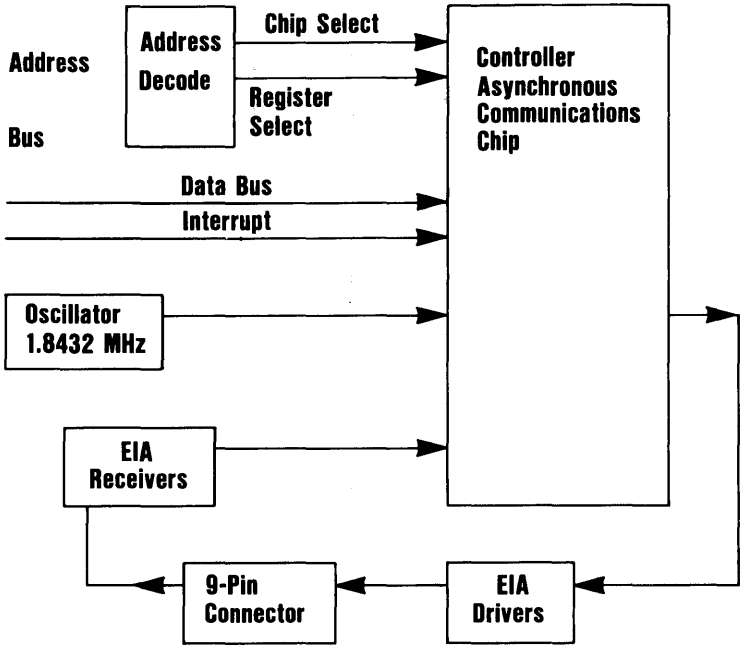
The IBM Personal Computer AT Serial/Parallel Adapter provides a parallel port and a serial port. It plugs into a system-board expansion slot. All system-control signals and voltage requirements are provided through a 2- by 31-position card edge connector.

Serial Portion of the Adapter

The serial portion of the adapter is fully programmable and supports asynchronous communications. It will add and remove start, stop, and parity bits. A programmable baud-rate generator allows operation from 50 baud to 9600 baud. Five-, six-, seven- and eight-bit characters with 1, 1.5, or 2 stop bits are supported. A prioritized interrupt system controls transmit, receive, error, and line status as well as data-set interrupts.

The rear of the adapter has a 9-pin D-shell connector that is classified as an RS-232C port. When the optional IBM Communications Cable (9-Pin), which has a 9-pin D-shell connector on one end and a 25-pin D-shell connector on the other end, is connected to the adapter, the 25-pin end of the cable has all the signals of a standard EIA RS-232C interface.

The following figure is a block diagram of the serial portion of the adapter.



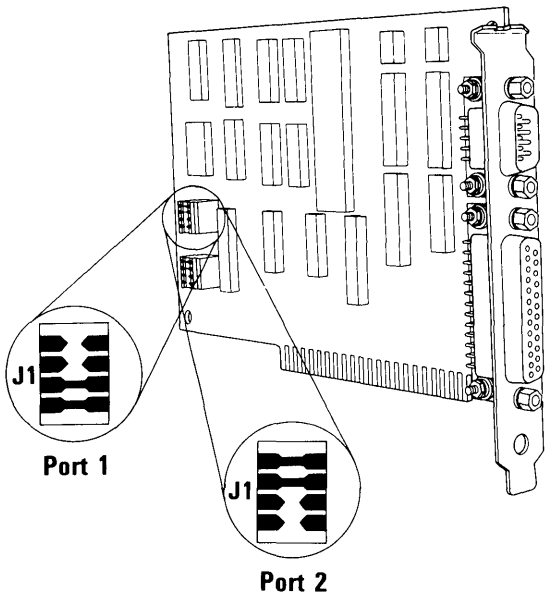
Serial Portion Block Diagram

The serial portion of the adapter has a controller that provides the following functions:

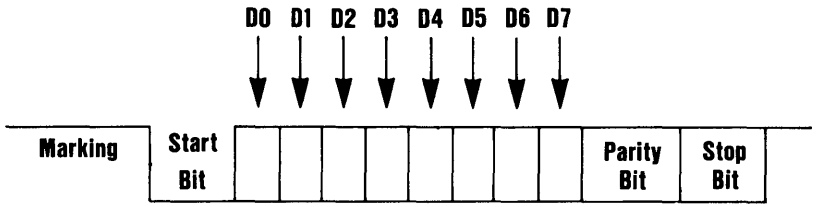
- Adds or deletes standard, asynchronous-communications bits to or from a serial data stream.
- Provides full, double buffering, which eliminates the need for precise synchronization.
- Provides a programmable baud-rate generator.
- Provides modem controls (CTS, RTS, DSR, DTR, RI, and CD).

Communications Application

The serial output port may be addressed as either communications port 1 or communications port 2 as defined by jumper J1 (see the following figure). In this section hex addresses begin with an X which can be either a 3 for communications port 1 (interrupt level 4) or a 2 for communications port 2 (interrupt level 3).



The data format will be as follows:



Data bit 0 is the first bit to be sent or received. The controller automatically inserts the start bit, the correct parity bit (if programmed to do so), and the stop bit (1, 1.5, or 2, depending on the command in the line-control register).

Controller Specifications

The following describes the function of controller input/output signals.

-Clear to Send: (-CTS), Pin 36—The '-CTS' signal is a modem-control function input, the condition of which can be tested by the processor by reading bit 4 (CTS) of the modem status register. Bit 0 (DCTS) of the modem status register indicates if the '-CTS' input has changed state since the previous reading.

> **Note:** Whenever the CTS bit of the modem status register changes state, an interrupt is generated if the modem-status interrupt is enabled.

-Data Set Ready: (-DSR), Pin 37—When low, indicates the modem or data set is ready to establish the communications link and transfer data with the controller. The '-DSR' signal is a modem-control function input, the condition of which can be tested by the processor reading bit 5 (DSR) of the modem status register. Bit 1 (DDSR) of the modem status register indicates if the '-DSR' input has changed since the previous reading.

> **Note:** Whenever the DSR bit of the modem status register changes state, an interrupt is generated if the modem-status interrupt is enabled.

-Data Carrier Detect: (-DCD), Pin 38—When low, indicates the modem or data set detected a data carrier. The '-DCD' signal is a modem-control function input, the condition of which can be tested by the processor reading bit 7 (DCD) of the modem status register. Bit 3 (DCD) of the modem status register indicates if the '-DCD' input has changed state since the previous reading.

> **Note:** Whenever the DCD bit of the modem status register changes state, an interrupt is generated if the modem status interrupt is enabled.

-Ring Indicator: (-RI), Pin 39—When low, indicates the modem or data set detected a telephone ringing signal. The '-RI' signal is a modem-control function input, the condition of which can be

tested by the processor reading bit 6 (RI) of the modem status register. Bit 2 (TERI) of the modem status register indicates if the '-RI' input has changed from an active to an inactive state since the previous reading.

> **Note:** Whenever the RI bit of the modem status register changes from an inactive to an active state, an interrupt is generated if the modem-status interrupt is enabled.

VCC Pin 40—+5 Vdc supply

VSS Pin 20—Ground (0 Vdc) reference

Output Signals

-Data Terminal Ready: (-DTR), Pin 33—When active, informs the modem or data set that the controller is ready to communicate. The 'DTR' output signal can be set to an active level by programming bit 0 (DTR) of the modem control register to an active level. The '-DTR' signal is set inactive upon a master reset operation.

-Request to Send: (-RTS), Pin 32—When active, informs the modem or data set that the controller is ready to send data. The '-RTS' output signal can be set to an active level by programming bit 1 (RTS) of the modem control register to an active level. The '-RTS' signal is set inactive upon a master reset operation.

-Output 1: (-OUT 1), Pin 34—User-designated output that can be set to an active level by programming bit 2 (-OUT 1) of the modem control register to an inactive level. The '-OUT 1' signal is set inactive upon a master reset operation. Pin 34 is connected to an active source.

-Output 2: (-OUT 2), Pin 31—User-designated output that can be set to an active level by programming bit 3 (-OUT 2) of the modem control register to an inactive level. The '-OUT 2' signal is set inactive upon a master reset operation. Pin 31 controls interrupts to the system.

Controller-Accessible Registers

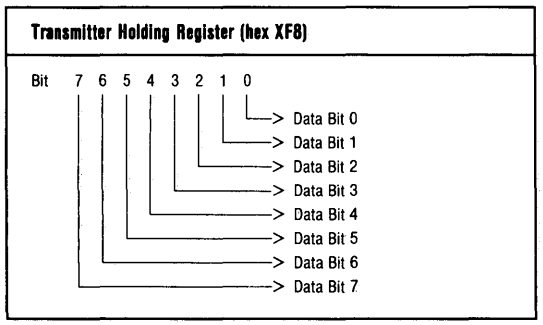
The controller has a number of accessible registers. The system programmer may gain access to or control any of the controller registers through the microprocessor. These registers are used to control the controller's operations and to transmit and receive data. The X in the register address determines the the port selected; 3 is for port 1 and 2 is for port 2.

Specific registers are selected according to the following figure:

I/O Address	Register Selected	DLAB State
XF8	TX buffer	0 (write)
XF8	RX buffer	0 (read)
XF8	Divisor Latch LSB	1
XF9	Divisor Latch MSB	1
XF9	Interrupt Enable Register	0
XFA	Interrupt Identification Register	
XFB	Line Control Register	
XFC	Modem Control Register	
XFD	Line Status Register	
XFE	Modem Status Register	
XFF	Reserved	

Controller-Accessible Registers

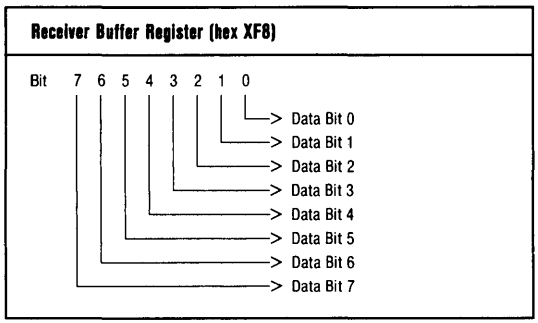
Transmitter Holding Register (Hex XF8): The transmitter holding register (THR) contains the character to be sent.



Transmitter Holding Register

Bit 0 is the least-significant bit and the first bit sent serially.

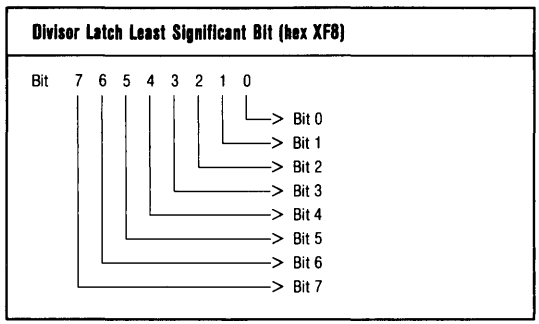
Receiver Buffer Register (Hex XF8): The receiver buffer register (RBR) contains the received character.



Receiver Buffer Register

Bit 0 is the least-significant bit and the first bit received serially.

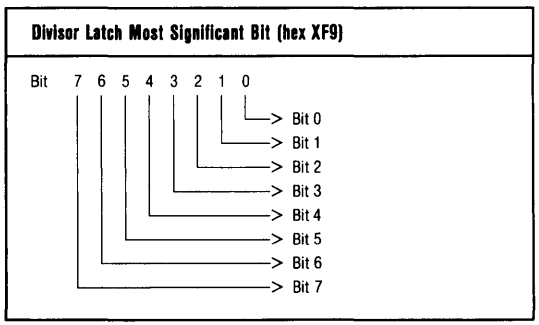
Divisor Latch LSB (Hex XF8)



Divisor Latch Least Significant Bit

Information about this register may be found under "Programmable Baud Rate Generator" later in this section.

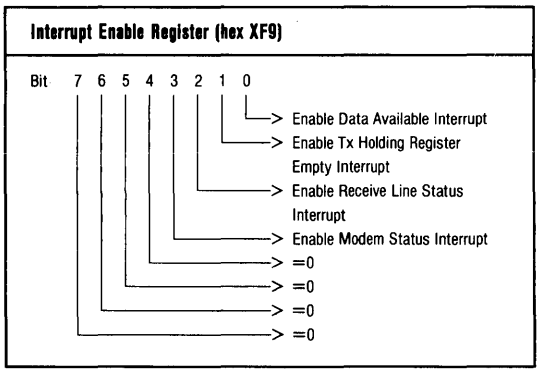
Divisor Latch MSB (Hex XF9)



Divisor Latch Most Significant Bit

Information about this register may be found under "Programmable Baud Rate Generator" later in this section.

Interrupt Enable Register (Hex XF9): This 8-bit register allows the four types of controller interrupts to separately activate the 'chip-interrupt' (INTRPT) output signal. The interrupt system can be totally disabled by resetting bits 0 through 3 of the interrupt enable register (IER). Similarly, by setting the appropriate bits of this register to logical 1, selected interrupts can be enabled. Disabling the interrupt system inhibits the 'IER' and the active 'INTRPT' output from the chip. All other system functions operate normally, including the setting of the line-status and modem-status registers.



Interrupt Enable Register

Bit 0 When set to logical 1, enables the received-data-available interrupt.

Bit 1 When set to logical 1, enables the transmitter-holding-register-empty interrupt.

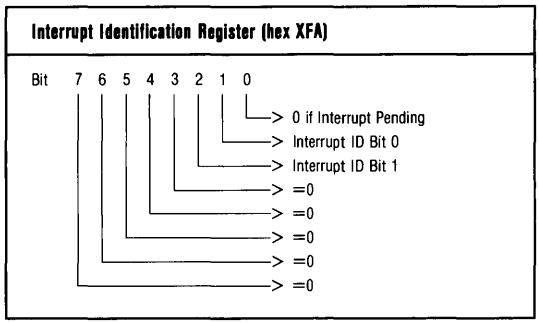
Bit 2 When set to logical 1, enables the receiver-line-status interrupt.

Bit 3 When set to logical 1, enables the modem-status interrupt.

Bits 4–7 These four bits are always logical 0.

Interrupt Identification Register (Hex XFA): The controller has an on-chip interrupt capability that makes communications possible with all of the currently popular microprocessors. In order to minimize programming overhead during data character transfers, the controller prioritizes interrupts into four levels: receiver line status (priority 1), received data ready (priority 2), transmitter holding register empty (priority 3), and modem status (priority 4).

Information about a pending prioritized interrupt is stored in the interrupt identification register (IIR). (See the figure "Interrupt Control Functions," later.) The IIR, when addressed during chip-select time, stops the pending interrupt with the highest priority, and no other interrupts are acknowledged until the processor services that particular interrupt.



Interrupt Identification Register

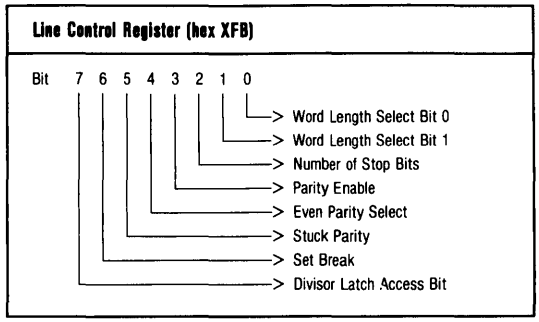
Bit 0 This bit can be used in either hard-wired, prioritized, or polled conditions to indicate if an interrupt is pending. When bit 0 is logical 0, an interrupt is pending, and the IIR contents may be used as a pointer to the appropriate interrupt service routine. When bit 0 is logical 1, no interrupt is pending, and polling (if used) continues.

Bits 1–2 These two bits identify the pending interrupt that has the highest priority interrupt pending, as shown in the following figure.

Bits 3–7 These five bits are always logical 0.

Interrupt ID Register			Interrupt Set And Reset Functions			
Bit 2	**Bit 1**	**Bit 0**	**Priority Level**	**Interrupt Type**	**Interrupt Source**	**Interrupt Reset Control**
0	0	0	—	None	None	—
1	1	0	Highest	Receiver Line Status	Overrun Error or Parity Error or Framing Error or Break Interrupt	Reading the Line Status Register
1	0	0	Second	Received Data Available	Receiver Data Available	Reading the Receiver Buffer Register
0	1	0	Third	Transmitter Holding Register Empty	Transmitter Holding Register Empty	Reading the IIR (if source of interrupt) or writing into the THR
0	0	0	Fourth	Modem Status	Clear to Send or Data Set Ready or Ring Indicator or Received Line Signal Detect	Reading the Modem Status Register

Line-Control Register (Hex XFB): The system programmer specifies the format of the asynchronous data communications exchange through the line control register. In addition to controlling the format, the programmer may retrieve the contents of the line control register for inspection. This feature simplifies system programming and eliminates the need to store line characteristics separately in system memory.



Line Control Register

Bits 0, 1 These two bits specify the number of bits in each serial character that is sent or received. The encoding of bits 0 and 1 is as follows:

Bit 1	Bit 2	Word Length (Bits)
0	0	5
0	1	6
1	0	7
1	1	8

Word Length

Bit 2 This bit specifies the number of stop bits in each serial character that is sent or received. If bit 2 is a logical 0, one stop bit is generated or checked in the data sent or received. If bit 2 is logical 1 when a 5-bit word length is selected through bits 0 and 1, 1-1/2 stop bits are generated or checked. If bit 2 is logical 1 when either a 6-, 7-, or 8-bit word length is selected, two stop bits are generated or checked.

Bit 3 This bit is the parity-enable bit. When bit 3 is logical 1, a parity bit is generated (transmit data) or checked (receive data) between the last data word and stop bit of the serial data. (The parity bit is used to produce an even or odd number of 1's when the data-word bits and parity bit are summed.)

Bit 4 This bit is the even-parity-select bit. When bit 3 is a logical 1 and bit 4 is a logical 0, an odd number of

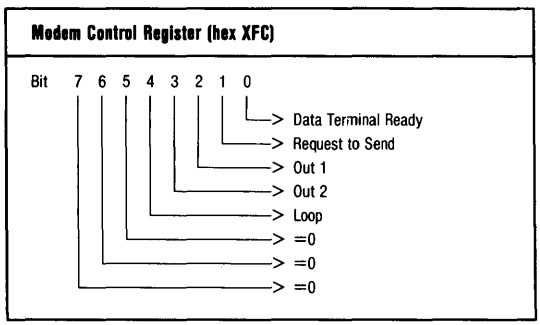
logical 1's is sent or checked in the data word bits and parity bit. When both bit 3 and bit 4 are a logical 1, an even number of bits is sent or checked.

Bit 5 This bit is the stuck-parity bit. When bit 3 is a logical 1 and bit 5 is a logical 1, the parity bit is sent and then detected by the receiver as a logical 0, if bit 4 is a logical 1, or as a logical 1 if bit 4 is a logical 0.

Bit 6 This bit is the set-break control bit. When bit 6 is set to a logical 1, the serial output (SOUT) is forced to the spacing (logical 0) state and remains there regardless of other transmitter activity. The set-break is disabled by setting bit 6 to logical 0. This feature enables the microprocessor to select a specific terminal in a computer communications system.

Bit 7 This bit is the divisor-latch access bit (DLAB). It must be set high (logical 1) to gain access to the divisor latches of the baud-rate generator during a read or write operation. It must be set low (logical 0) to gain access to the receiver buffer, the transmitter holding register, or the interrupt enable register.

Modem Control Register (Hex XFC): This 8-bit register controls the data exchange with the modem or data set (an external device acting as a modem).



Modem Control Register

Bit 0 This bit controls the '-data terminal ready' (-DTR) output. When bit 0 is set to logical 1, the -DTR output is forced active. When bit 0 is reset to logical 0, the '-DTR' output is forced inactive.

Bit 1 This bit controls the '-request-to-send' (-RTS) output. Bit 1 affects the '-RTS' output in the same way bit 0 affects the '-DTR' output.

Bit 2 This bit controls the '-Output 1' (-OUT 1) signal, which is a spare the programmer can use. Bit 2 affects the '-OUT 1' output in the same way bit 0 affects the '-DTR' output.

Bit 3 This bit controls the '-Output 2' (-OUT 2) signal, which is a spare the programmer can use. Bit 3 affects the '-OUT 2' output in the same way bit 0 affects the '-DTR' output.

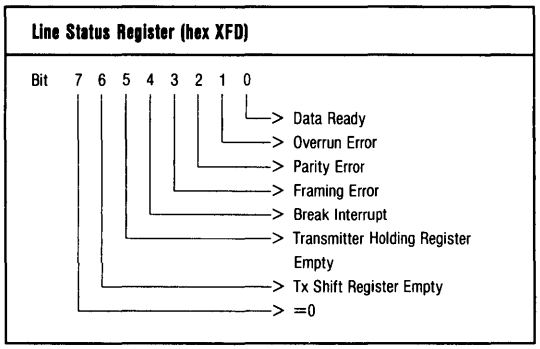
Bit 4 This bit provides a loopback feature for diagnostic testing of the controller. When bit 4 is set to logical 1, the following occur: the 'transmitter serial output' (SOUT) is set to the active state; the 'receiver serial input' (SIN) is disconnected; the output of the transmitter shift register is "looped back" to the receiver shift register input; the four modem-control inputs ('-CTS', '-DSR', '-RLSD', and '-RI') are disconnected; and the four modem-control outputs ('-DTR', '-RTS', '-OUT 1' and '-OUT 2') are internally connected to the four modem control inputs. In the diagnostic mode, data sent is immediately received. This feature allows the processor to verify the transmit- and receive-data paths of the controller.

In the diagnostic mode, the receiver and transmitter interrupts are fully operational, as are the modem-control interrupts. But the interrupts' sources are now the lower four bits of the modem control register (MCR) instead of the four modem-control inputs. The interrupts are still controlled by the interrupt enable register.

The controller's interrupt system can be tested by writing to the lower six bits of the line status register and the lower four bits of the modem status register. Setting any of these bits to logical 1 generates the appropriate interrupt (if enabled). Resetting these interrupts is the same as for normal controller operation. To return to normal operation, the registers must be reprogrammed for normal operation, and then bit 4 of the MCR must be reset to logical 0.

Bits 5–7 These bits are permanently set to logical 0.

Line Status Register (Hex XFD): This 8-bit register provides the processor with status information about the data transfer.



Line Status Register

Bit 0 This bit is the receiver data ready (DR) indicator. It is set to logical 1 whenever a complete incoming character has been received and transferred into the receiver buffer register. Bit 0 may be reset to logical 0 by the processor either reading the data in the receiver's buffer register or writing logical 0 in it.

Bit 1 This bit is the overrun error (OE) indicator. It indicates that data in the receiver's buffer register was not read by the processor before the next character was transferred into the register, thereby destroying the previous character. The OE indicator is reset whenever the processor reads the contents of the line status register.

Bit 2 This bit is the parity error (PE) indicator and indicates the received data character does not have the correct even or odd parity, as selected by the even-parity-select bit. The PE bit is set to logical 1 upon detection of a parity error, and is reset to logical 0 whenever the processor reads the contents of the line status register.

Bit 3 This bit is the framing error (FE) indicator. It indicates the received character did not have a valid stop bit. Bit 3 is set to logical 1 whenever the stop bit following the last data bit or parity bit is detected as a zero bit (spacing level).

Bit 4 This bit is the break interrupt (BI) indicator. It is set to logical 1 whenever the received data input is held in the spacing state (logical 0) for longer than a fullword transmission time (that is, the total time of start bit + data bits + parity stop bits).

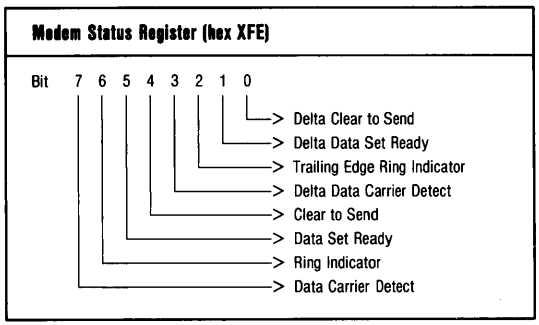
> **Note:** Bits 1 through 4 are error conditions that produce a receiver line-status interrupt whenever any of the corresponding conditions are detected.

Bit 5 This bit is the transmitter holding register empty (THRE) indicator. It indicates the controller is ready to accept a new character for transmission. In addition, this bit causes the controller to issue an interrupt to the processor when the TRHE interrupt enable is set active. The THRE bit is set to logical 1 when a character is transferred from the transmitter holding register into the transmitter shift register. It is reset to logical 0 when the processor loads the transmitter holding register.

Bit 6 This bit is the transmitter empty (TEMT) indicator. It is set to logical 1 whenever the transmitter holding request (THR) and the transmitter shift request (TSR) are both empty. It is reset to logical 0 whenever THR or TSR contains a data character.

Bit 7 This bit is permanently set to logical 0.

Modem Status Register (Hex XFE): The 8-bit MSR provides the current state of the control lines from the modem (or external device) to the processor. In addition, four bits of the MSR provide change information. These four bits are set to logical 1 whenever a control input from the modem changes state. They are reset to logical 0 whenever the processor reads this register.



Modem Status Register

Bit 0 This bit is the delta clear-to-send (DCTS) indicator. It indicates the '-CTS' input to the chip has changed state since the last time it was read by the processor.

Bit 1 This bit is the delta data-set-ready (DDSR) indicator. It indicates the '-DSR' input to the chip has changed state since the last time it was read by the processor.

Bit 2 This bit is the trailing-edge ring-indicator (TERI) detector. It indicates the '-RI' input to the chip has changed from an active condition to an inactive condition.

Bit 3 This bit is the delta data-carrier-detect (DDCD) indicator. It indicates the '-DCD' input to the chip has changed state.

> **Note:** Whenever bit 0, 1, 2, or 3 is set to a logical 1, a modem status interrupt is generated.

Bit 4 This bit is the opposite of the '-clear-to-send' (-CTS) input. If bit 4 of the MCR loop is set to a logical 1, this bit is equivalent to RTS of the MCR.

Bit 5 This bit is the opposite of the '-data-set-ready' (-DSR) input. If bit 4 of the MCR is set to a logical 1, this bit is equivalent to DTR of the MCR.

Bit 6 This bit is the opposite of the '-ring-indicator' (-RI) input. If bit 4 of the MCR is set to a logical 1, this bit is equivalent to OUT 1 of the MCR.

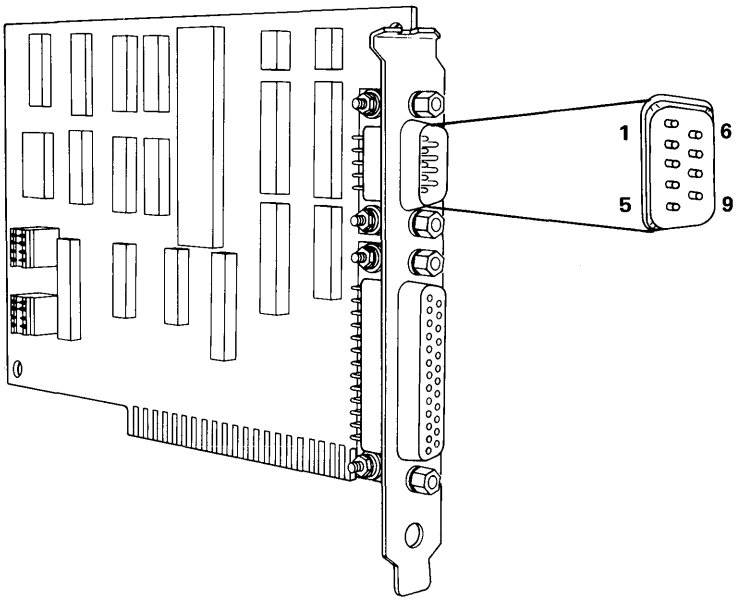
Bit 7 This bit is the opposite of the '-data-carrier-detect' (-DCD) input. If bit 4 of the MCR is set to a logical 1, this bit is equivalent to OUT 2 of the MCR.

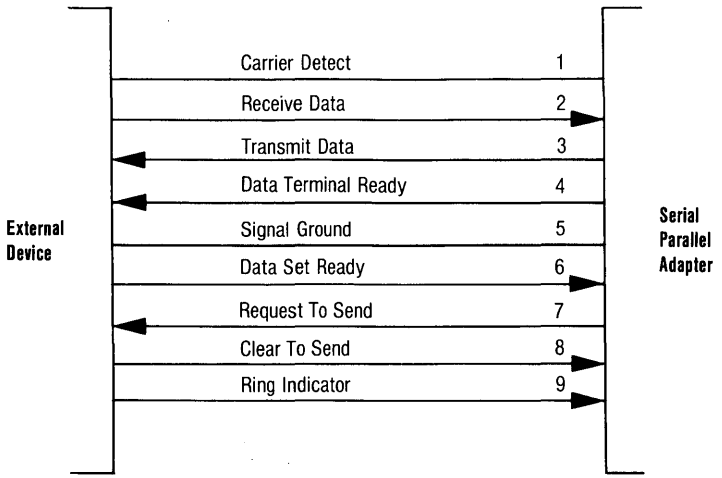
Programmable Baud-Rate Generator

The controller has a programmable baud-rate generator that can divide the clock input (1.8432 MHz) by any divisor from 1 to 655,535 or 2^{16}-1. The output frequency of the baud-rate generator is the baud rate multiplied by 16. Two 8-bit latches store the divisor in a 16-bit binary format. These divisor latches must be loaded during setup to ensure desired operation of the baud-rate generator. When either of the divisor latches is loaded, a 16-bit baud counter is immediately loaded. This prevents long counts on the first load.

Pin Assignment for Serial Port

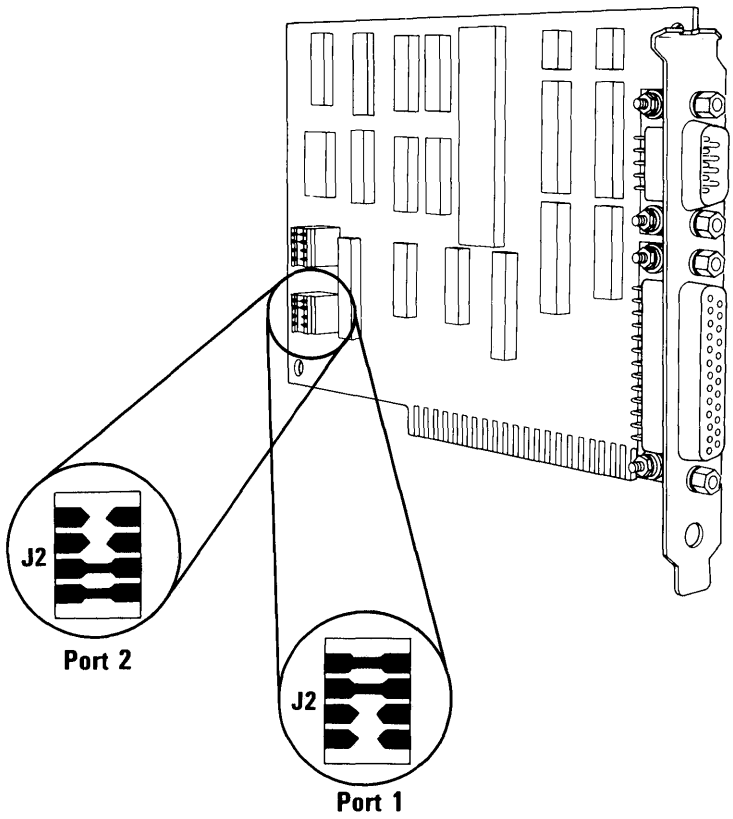
The following figure shows the pin assignments for the serial port in a communications environment.



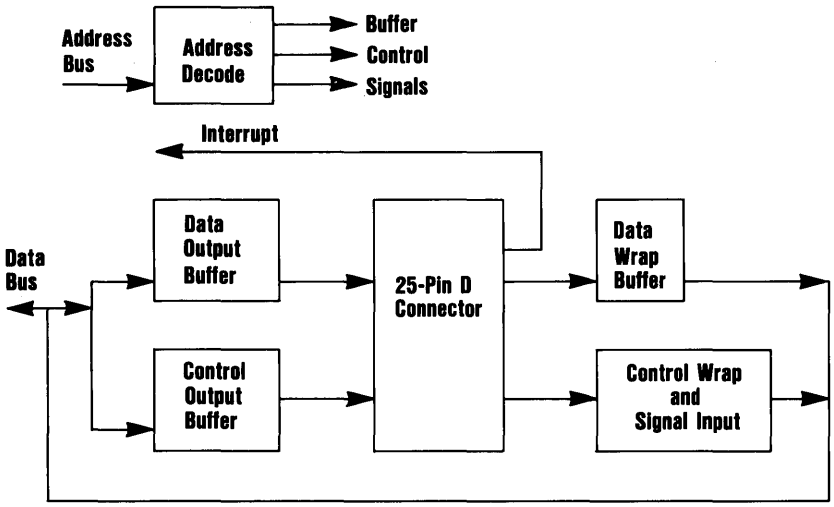


Parallel Portion of the Adapter

The parallel portion of the adapter makes possible the attachment of various devices that accept eight bits of parallel data at standard TTL levels. The rear of the adapter has a 25-pin, D-shell connector. This port may be addressed as either parallel port 1 or 2. The port address is determined by the position of jumper J2, as shown in the following figure.



The following figure is a block diagram of the parallel portion of the adapter.



Parallel Portion Block Diagram

Printer Application

The following discusses the use of the parallel portion of the adapter to connect to a parallel printer. Hexidecimal addresses in this section begin with an X, which is replaced with a 3 to indicate port 1, or a 2 to indicate port 2.

Data Latch (Hex X78, X7C)

Writing to this address causes data to be stored in the printer's data buffer. Reading this address sends the contents of the printer's data buffer to the system microprocessor.

Printer Controls (hex X7A, X7E)

Printer control signals are stored at this address to be read by the system microprocessor. The following are bit definitions for this byte.

Bit 7 Not used

Bit 6 Not used

Bit 5 Not used

Bit 4 +IRQ Enable—A 1 in this position allows an interrupt to occur when '-ACK' changes from true to false.

Bit 3 +SLCT IN—A 1 in this bit position selects the printer.

Bit 2 -INIT—A 0 starts the printer (50-microsecond pulse, minimum).

Bit 1 +AUTO FD XT—A 1 causes the printer to line-feed after a line is printed.

Bit 0 +STROBE—A 0.5-microsecond minimum, high, active pulse clocks data into the printer. Valid data must be present for a minimum of 0.5 microsecond before and after the strobe pulse.

Printer Status - Address X79, X7D

Printer status is stored at this address to be read by the microprocessor. The following are bit definitions for this byte.

Bit 7 -BUSY—When this signal is active, the printer is busy and cannot accept data. It may become active during data entry, while the printer is offline, during printing, when the print head is changing positions, or while in an error state.

Bit 6 -ACK—This bit represents the current state of the printer's '-ACK' signal. A 0 means the printer has received the character and is ready to accept another. Normally, this signal will be active for approximately 5 microseconds before '-BUSY' stops.

Bit 5 +PE—A 1 means the printer has detected the end of paper.

Bit 4 +SLCT—A 1 means the printer is selected.

Bit 3 -Error—A 0 means the printer has encountered an error condition.

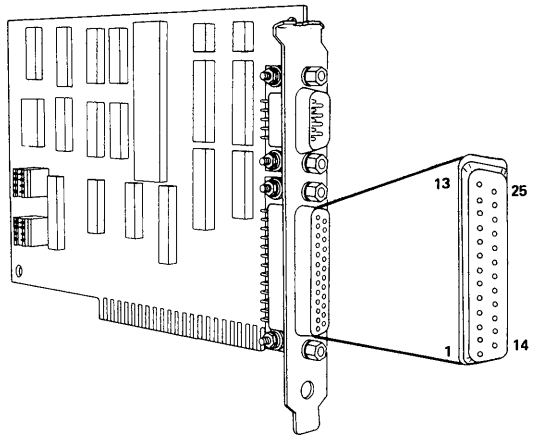
Bit 2 Unused.

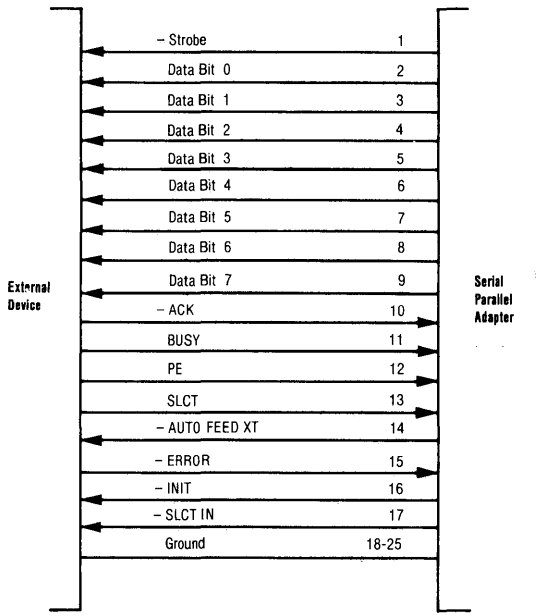
Bit 1 Unused.

Bit 0 Unused.

Parallel Interface

The adapter has a 25-pin, D-shell connector at the rear of the adapter. The following figure shows the signals and their pin assignments. Typical printer input signals also are shown.





Specifications

The following figures list characteristics of the output driver.

Sink current	24 mA	Max
Source current	-2.6 mA	Max
High-level output voltage	2.4 Vdc	Min
Low-level output voltage	0.5 Vdc	Max

Parallel Data and Processor IRQ

Sink current	16 mA	Max
Source current	0.55 mA	Max
High level output voltage	5 Vdc	Minus pull-up
Low level output voltage	0.4 Vdc	Max

Parallel Control

Sink current	24 mA	Max
Source current	-15 mA	Max
High level output voltage	2.0 Vdc	Min
Low level output voltage	0.5 Vdc	Max

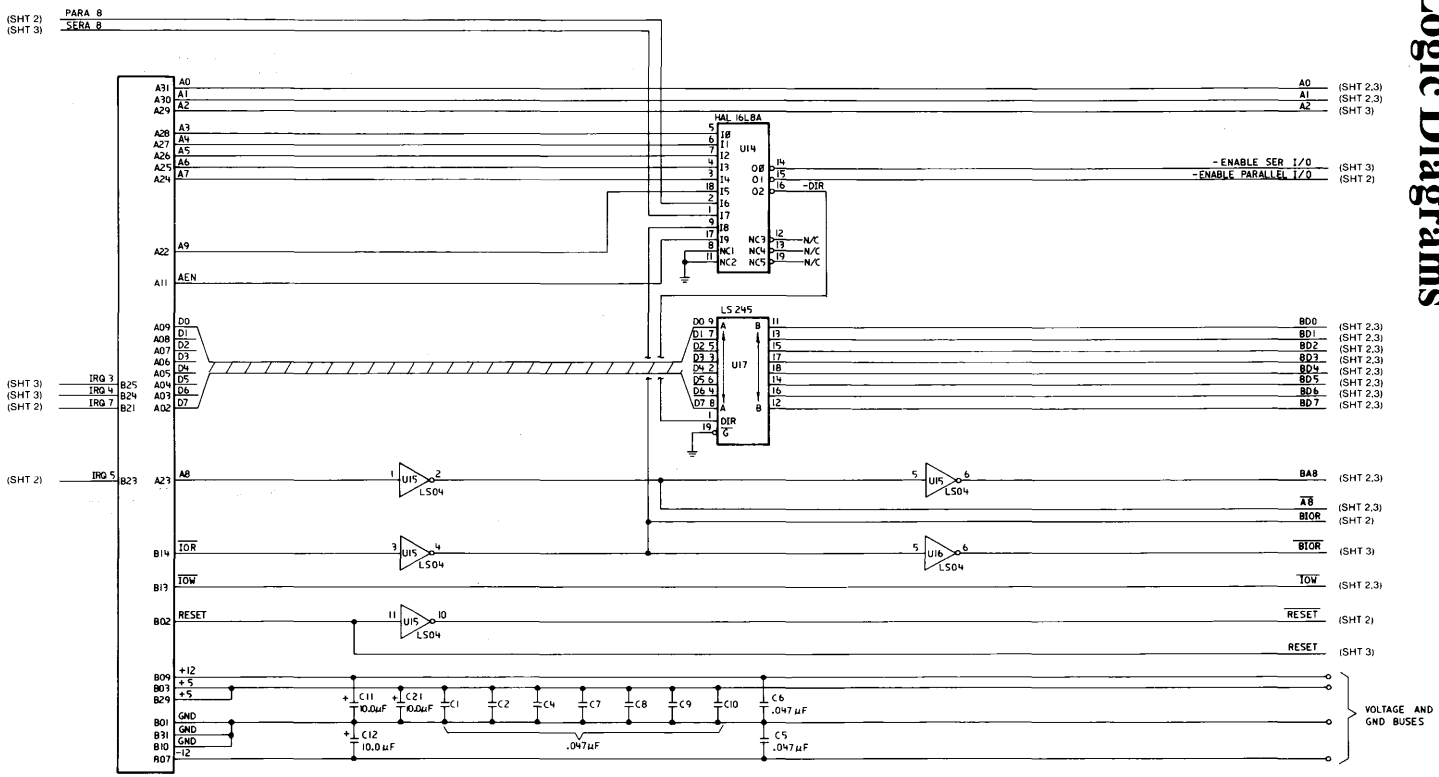
Parallel Processor Interface (except IRQ)

The following are the specifications for the serial interface.

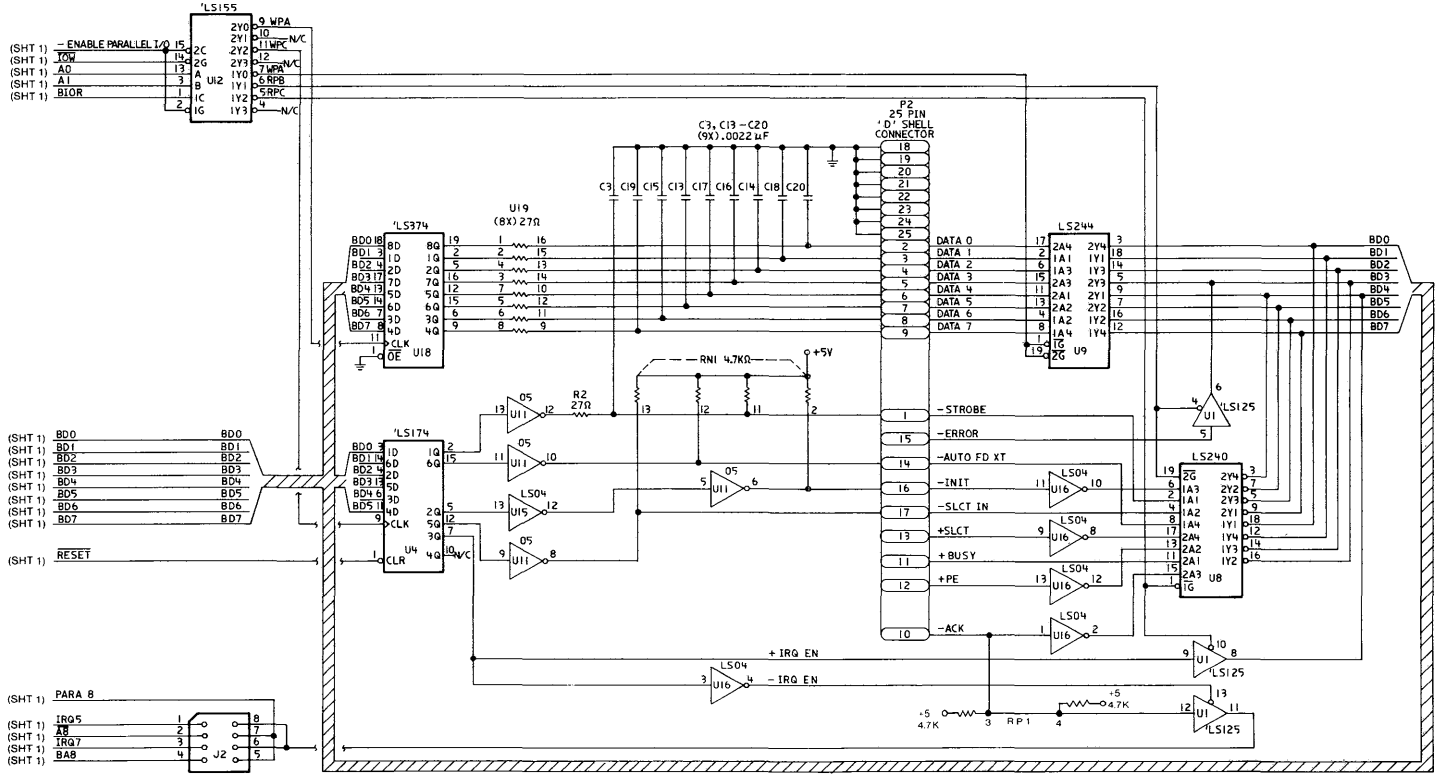
Function	Condition
On	Spacing condition (binary 0, positive voltage).
Off	Marking condition (binary 1, negative voltage).

Voltage	Function
above +15 Vdc	Invalid
+3 Vdc to +15 Vdc	On
-3 Vdc to +3 Vdc	Invalid
-3 Vdc to -15 Vdc	Off
Below -15 Vdc	Invalid

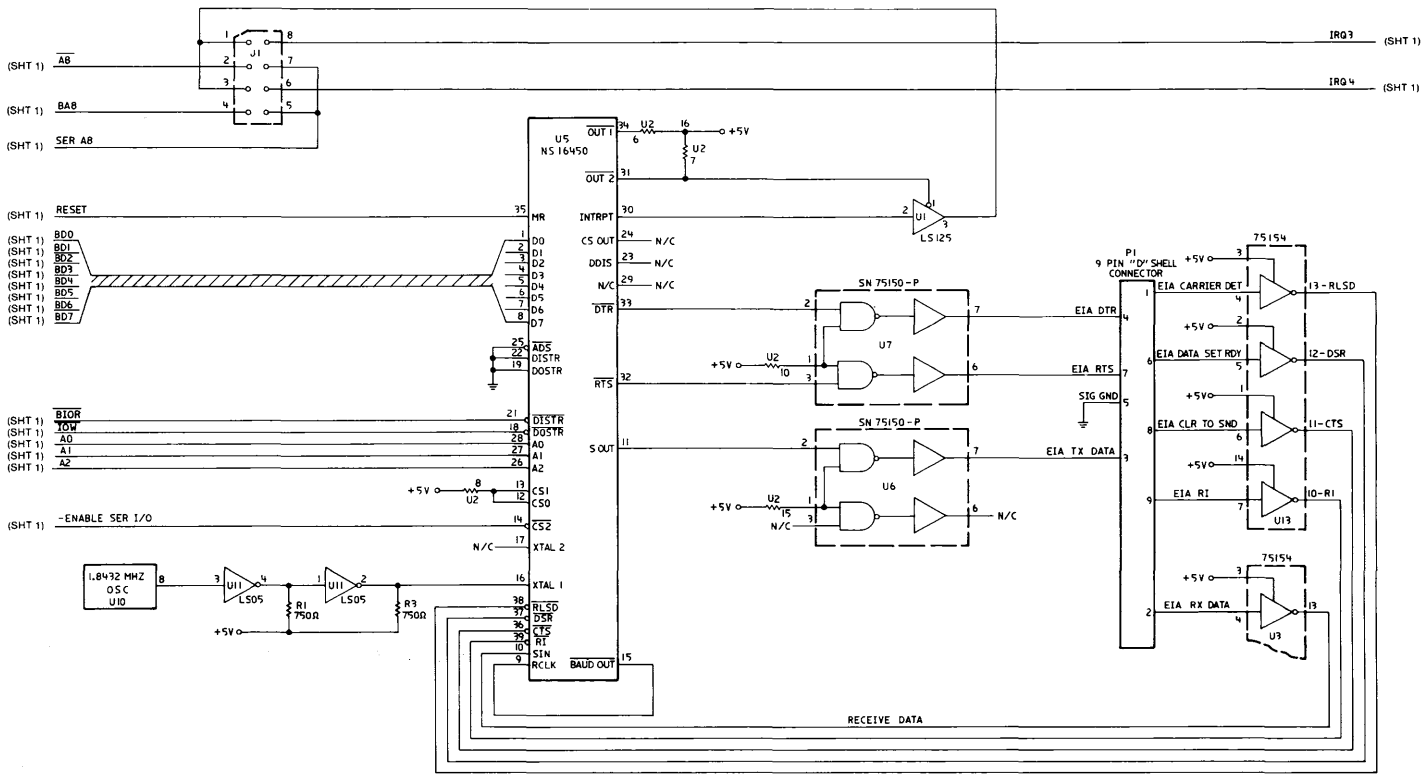
Serial Port Functions



Serial/Printer Adapter (Sheet 1 of 3)



Serial/Printer Adapter (Sheet 2 of 3)



Serial/Printer Adapter (Sheet 3 of 3)

Notes:

IBM TECHNICAL NEWSLETTER

for the

IBM RT PC

Hardware Technical Reference

—OVER—

SN20-9844
75X1073
June 1987

84X0875
Printed in U.S.A.

Update Kit Contents

This kit contains:

1. Volume III binder
2. Updates to Volumes I and II
3. Information for new RT PC adapters.

Notes

Replace the following pages in volume I with TNL pages supplied.

1. Replace pages iii through xiv with new TNL pages iii through xiv in the front of volume 1.
2. Replace Section 2. Processor Board with new TNL Section 2. Processor Board
3. Replace Section 3. Memory Boards with new TNL Section 3. Memory Boards
4. Replace Section 4. Floating-Point Accelerator with new TNL Section 4. Floating-Point Accelerator.
5. Replace pages 5-27 through 5-28 with new TNL pages 5-27 through 5-28.
6. Replace pages 5-63 through 5-64 with new TNL pages 5-63 through 5-64.
7. Replace pages 6-17 through 6-20 with new TNL pages 6-17 through 6-20.
8. Replace pages 6-47 through 6-62 with new TNL pages 6-48 through 6-64.
9. Replace Section 7. System IPL ROM with new TNL Section 7. System IPL ROM.
10. Replace Section 8. System Compatibility with new TNL Section 8. System Compatibility.
11. Replace pages 11-13 through 11-16, 11-81 through 11-82, and 11-87 through 11-88 with new TNL pages 11-13 through 11-16, 11-81 through 11-82 , and 11-87 through 11-88.
12. Replace Appendix A. with new TNL Appendix A.
13. Replace Glossary and Index with new TNL Glossary and Index.

Add Appendix B. Advanced Processor Board to Volume I.

Replace the following adapters in Volume II.

1. Replace ESDI Magnetic Media Adapter with new TNL ESDI Magnetic Media Adapter.
2. Replace Extended Monochrome Graphics Adapter with new TNL Extended Monochrome Graphics Adapter.
3. Replace RS-422A Adapter with new TNL RS-422A Adapter.
4. Replace RS-232C Adapter with new TNL RS-232C Adapter.

Add the following adapters to Volume II or Volume III.

1. Extended Enhanced Small Device Interface (HESDI) Adapter
2. Small Computer System Interface (SCSI) Adapter.
3. Extended Monochrome Graphics Display.

Replace pages 27 and 28 of the Monochrome Display and Printer Adapter in Volume II with new TNL pages 27 and 28.

New divider tabs are provided for Volumes II and III. You may want to organize the information using these dividers and redistribute the adapters in Volumes II and III.

Please file this cover letter at the back of the manual to provide a record of changes.

SN20-9844
75X1073
June 1987

84X0875
Printed in U.S.A.

Printed in the
United States of America

References in this
publication to IBM
products or services do not
imply that IBM intends
to make them available
outside the United States.

84X0873