

## Description

The μPD72120 Advanced Graphics Display Controller (AGDC) displays characters and graphics on a raster scan device from commands and parameters received from a host processor or CPU. Features of the AGDC include high-speed graphics drawing capabilities, video timing signal generation, large capacity display memory control (including video RAMs), and a versatile CPU interface. These features allow the AGDC to control graphics drawing and display of bit-mapped systems.

## Features

- High-speed graphics drawing functions
  - Graphics drawing: dot, straight line, rectangle, circle, arc, sector, segment, ellipse, ellipse arc, ellipse sector, and ellipse segment
  - Maximum drawing speed
    - 500 ns/pixel (8 MHz, pixel mode)
    - 500 ns/dot (8 MHz, plane mode)
  - Area filling (high-speed processing in word units): triangle, trapezoid, circle, ellipse, and rectangle
  - Painting: filling of any arbitrary enclosed area (bit boundary retrieval)
  - Data transfers in display memory: multiplane transfers; data transformation (90°/180°/270° rotation and reversal); multiwindow transfers; maximum transfer speed of 500 ns/word
  - Image processing: slant, arbitrary angle rotation, 16/N enlargement, and N/16 shrinkage (N any integer from 1 to 16)
  - Position specification by X-Y coordinates
  - Logical operations between planes
- Video timing signal generation
  - High-speed processing by two system clocks: display (for video sync signal generation) and graphics drawing clocks
  - External synchronization capability
- Large-capacity display memory
  - Display memory bus interface: 24-bit address and 16-bit data bus for addressing up to 16M words, 16 bits/word
  - Video RAM (VRAM) control
  - Display memory bus arbitration
- Host processor (CPU) interface
  - System bus interface: 20-bit address bus, 8- or 16-bit data bus
  - Data transfer with external DMA controller: from system memory to display memory (PUT); from display memory to system memory (GET)
  - High-speed pipeline processing with preprocessor before drawing processor
  - CPU memory or I/O mapping of internal registers and display memory for efficient system interface
- 8-MHz system clock
- CMOS technology
- Single +5-volt power supply
- Packages: 84-pin PLCC, 94-pin plastic miniflat

## Ordering Information

Part No	Package
μPD72120L	84-pin PLCC
μPD72120GJ-5BG	94-pin plastic miniflat

**Pin Identification**

Symbol	I/O	Signal Function															
<b>Clock Pins</b>																	
CLK	In	Clock supplied to circuits other than the sync signal generator and display processor. The drawing processor and preprocessor speed depend on this clock frequency.															
SCLK	In	Clock supplied to the sync signal generator and the display processor. This clock frequency is determined by the CRT timing requirements: horizontal sync frequency, number of dots per line, etc.															
<b>System Bus Control Pins</b>																	
AD <sub>0</sub> -AD <sub>15</sub>	I/O	I/O bus to the CPU consisting of multiplexed 16-bit address and a bidirectional data bus.															
A <sub>16</sub> -A <sub>19</sub>	In	Upper four address bits of the 20-bit address.															
ASTB	In	Latches the address on A <sub>16</sub> -A <sub>19</sub> and AD <sub>0</sub> -AD <sub>15</sub> on the falling edge.															
UBE	In	Together with AD <sub>0</sub> , defines the data access format as shown below. UBE should be tied high when connected to an 8-bit CPU.															
		<table border="1"> <thead> <tr> <th>AD<sub>0</sub></th> <th>UBE</th> <th>Data Access Format</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>Even-address word</td> </tr> <tr> <td>0</td> <td>1</td> <td>Even-address byte</td> </tr> <tr> <td>1</td> <td>0</td> <td>Odd-address byte</td> </tr> <tr> <td>1</td> <td>1</td> <td>Odd-address byte</td> </tr> </tbody> </table>	AD <sub>0</sub>	UBE	Data Access Format	0	0	Even-address word	0	1	Even-address byte	1	0	Odd-address byte	1	1	Odd-address byte
AD <sub>0</sub>	UBE	Data Access Format															
0	0	Even-address word															
0	1	Even-address byte															
1	0	Odd-address byte															
1	1	Odd-address byte															
RD	In	Performs a read of data from the AGDC by the host CPU.															
WR	In	Performs a write of data to the AGDC from the host CPU.															
CSIR	In	Enables reading/writing of the AGDC internal registers by the host CPU. The register is selected by the address input on AD <sub>0</sub> -AD <sub>7</sub> .															
CSDM	In	Enables reading/writing of display memory through the AGDC by the host CPU. The display memory address is generated by the address input on A <sub>16</sub> -A <sub>19</sub> and AD <sub>0</sub> -AD <sub>15</sub> and by the bank register.															
READY	Out	Activated by the data access request (RD/WR) for the AGDC. During the access, the signal may be low. RESET will set the READY line high.															
INT	Out	Signals an interrupt from the AGDC.															
DMARQ	Out	Indicates a request for data transfer (PUT/GET) to an external DMA controller. DMARQ will be low after RESET.															
DMAAK	In	Acknowledgment of DMA request to the AGDC by the DMA controller.															
RESET	In	Initializes operation of the AGDC. The internal parameter register is not cleared by RESET (it is initialized by setting data).															

**Display Memory Control Pins**

DAD <sub>0</sub> -DAD <sub>15</sub>	I/O	I/O pins for display memory; 16-bit address multiplexed with data.
-------------------------------------	-----	--

Symbol	I/O	Signal Function																								
DA <sub>16</sub> -DA <sub>23</sub>	Out	Upper 8 bits of display memory address (the lower 16 bits of the 24-bit address are output on DAD <sub>0</sub> -DAD <sub>15</sub> ).																								
DASTB	Out	Indicates that a display memory address is present on the falling edge.																								
DUBE, DLBE	Out	Defines the data format for accessing the display. RESET sets both pins low.																								
		<table border="1"> <thead> <tr> <th></th> <th>DUBE</th> <th>DLBE</th> <th>Data Access Format</th> </tr> </thead> <tbody> <tr> <td>AGDC</td> <td>0</td> <td>0</td> <td>Word</td> </tr> <tr> <td>16-bit CPU</td> <td>0</td> <td>0</td> <td>Word</td> </tr> <tr> <td>8/16-bit CPU 0</td> <td>0</td> <td>1</td> <td>High (odd) byte</td> </tr> <tr> <td>8/16-bit CPU 1</td> <td>0</td> <td>0</td> <td>Low (even) byte</td> </tr> <tr> <td>8-bit CPU</td> <td>1</td> <td>1</td> <td>High (odd) byte</td> </tr> </tbody> </table>		DUBE	DLBE	Data Access Format	AGDC	0	0	Word	16-bit CPU	0	0	Word	8/16-bit CPU 0	0	1	High (odd) byte	8/16-bit CPU 1	0	0	Low (even) byte	8-bit CPU	1	1	High (odd) byte
	DUBE	DLBE	Data Access Format																							
AGDC	0	0	Word																							
16-bit CPU	0	0	Word																							
8/16-bit CPU 0	0	1	High (odd) byte																							
8/16-bit CPU 1	0	0	Low (even) byte																							
8-bit CPU	1	1	High (odd) byte																							
DRD	Out	Controls reading of the display memory by the AGDC. Set high by RESET.																								
DWR	Out	Controls writing to the display memory by the AGDC. Set high by RESET.																								
HLDRQ	In	Requests control of the display memory bus by an external device to transfer display data.																								
HLDAK	Out	Indicates that the AGDC memory bus (DAD <sub>0</sub> -DAD <sub>15</sub> and DA <sub>16</sub> -DA <sub>23</sub> ) is in high-impedance state so that an external device can have access to the display memory bus. Set high by RESET.																								

**Video Timing Signal Related Pins**

VS/EXVS	I/O	When the AGDC operates as the master, VS is the vertical sync signal output. When the AGDC operates as a slave, the EXVS input initializes the internal vertical sync signal on the rising edge.
HS/EXHS	I/O	When the AGDC operates as the master, HS is the horizontal sync signal output. When the AGDC operates as a slave, EXHS initializes the internal horizontal sync signal on the rising edge.

**Display Signal Related Pins**

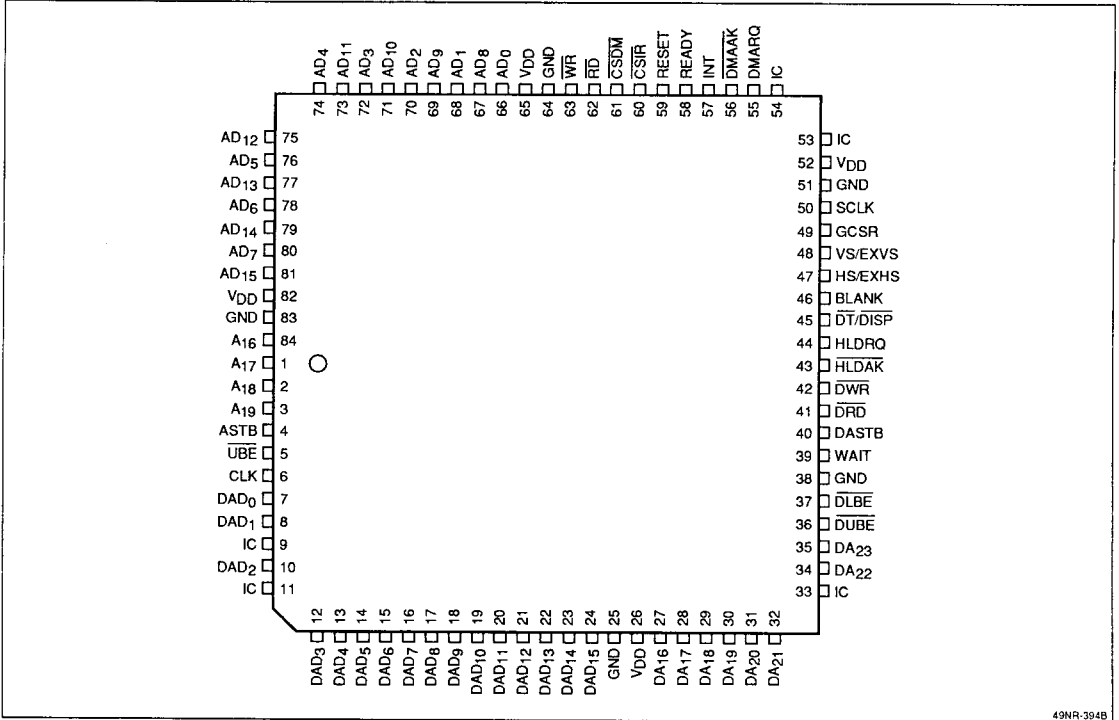
BLANK	Out	Used to blank the display.
DT/DISP	Out	Set to DT in the DT mode (when using VRAMs) and specifies the data transfer. In the cycle steal mode (VRAMs not used), indicates the display cycle.
GCSR	Out	Specifies the display of the graphics cursor
GWAIT	Out	Graphics wait signal

**Other Pins**

V <sub>DD</sub>	+5-volt power supply
GND	Ground
IC	Internally connected; leave unconnected

## Pin Configurations

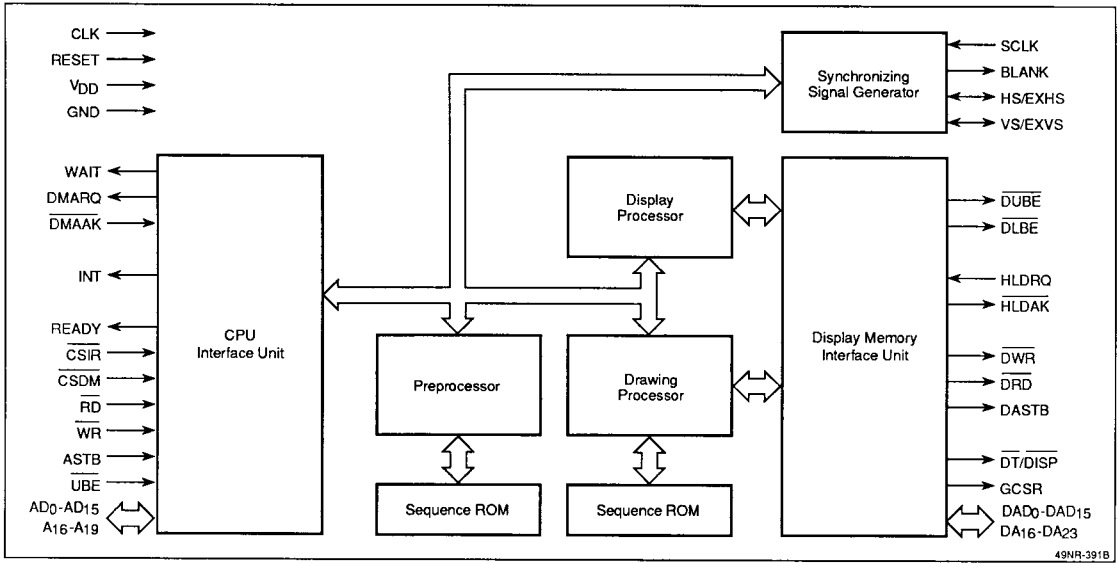
### 84-Pin PLCC



49NR-394B

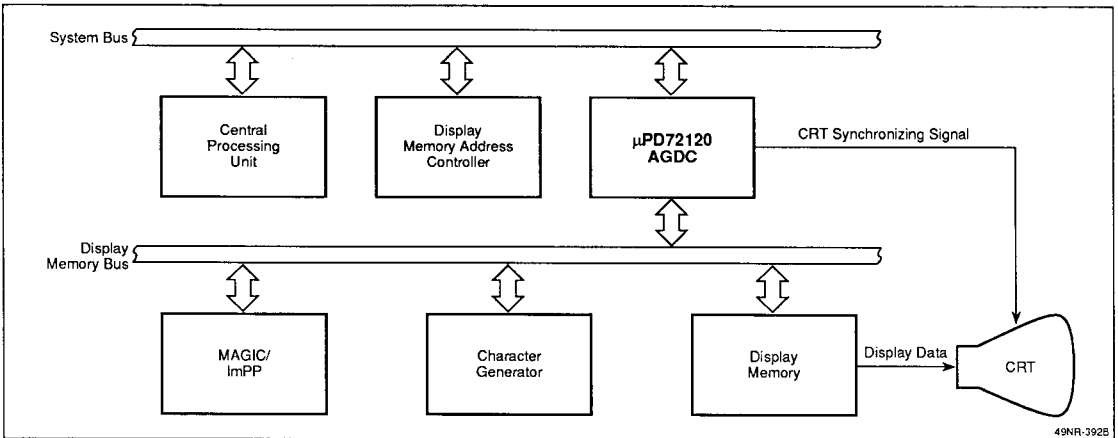


## μPD72120 Block Diagram

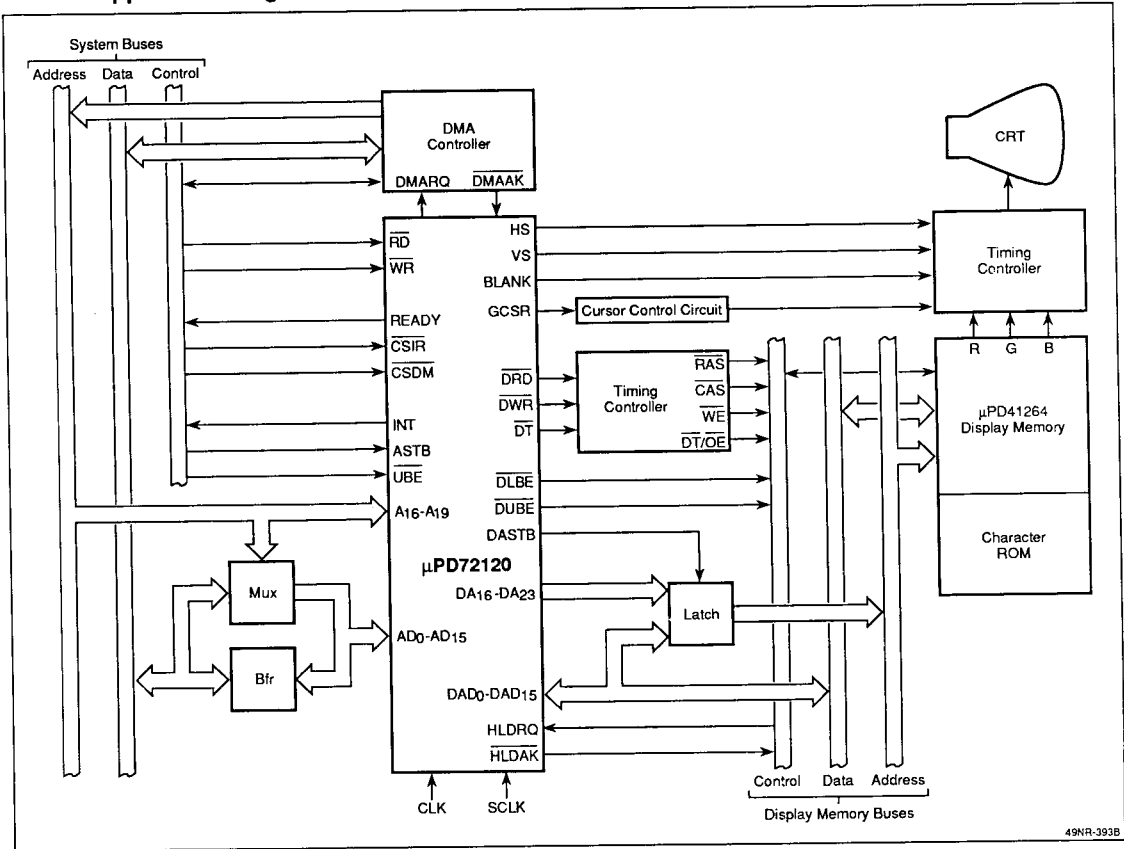


3

## System Configuration Example



General Application Diagram



## ELECTRICAL SPECIFICATIONS

### Absolute Maximum Ratings

$T_A = +25^\circ\text{C}$

Supply voltage, $V_{DD}$	-0.5 to +7.0 V
Input voltage, $V_I$	-0.5 to +7.0 V
Output voltage, $V_O$	-0.5 to +7.0 V
Operating temperature, $T_{OPT}$	-10 to +70°C
Storage temperature, $T_{STG}$	-65 to +150°C
Power dissipation, $P_D$	1.1 W

### Capacitance

$T_A = +25^\circ\text{C}; V_{DD} = \text{GND} = 0 \text{ V}$

Parameter	Symbol	Min	Max	Unit	Condition
Input	$C_I$		10	pF	$f = 1 \text{ MHz};$ unmeasured pins returned to 0 V
Output	$C_O$		20	pF	
Input/ output	$C_{I/O}$		20	pF	
Clock input	$C_C$		20	pF	

### DC Characteristics

$T_A = -10 \text{ to } +70^\circ\text{C}; V_{DD} = +5.0 \text{ V} \pm 10\%$

Parameter	Symbol	Min	Max	Unit	Condition
Low-level input voltage	$V_{IL}$	-0.5	0.8	V	Except CLK or SCLK
		-0.5	0.6	V	CLK, SCLK
High-level input voltage	$V_{IH}$	2.2	$V_{DD} + 0.5$	V	Except CLK or SCLK
		3.5	$V_{DD} + 1.0$	V	CLK, SCLK
Low-level output voltage	$V_{OL}$		0.45	V	$I_{OL} = 2.2 \text{ mA}$
High-level output voltage	$V_{OH}$	2.4		V	$I_{OH} = -400 \mu\text{A}$
Low-level input leakage current	$I_{LIL}$		-10	$\mu\text{A}$	$V_I = 0 \text{ V}$
High-level input leakage current	$I_{LIH}$		10	$\mu\text{A}$	$V_I = V_{DD}$
Low-level output leakage current	$I_{LOL}$		-10	$\mu\text{A}$	$V_O = 0 \text{ V}$
High-level output leakage current	$I_{LOH}$		10	$\mu\text{A}$	$V_O = V_{DD}$
Supply current	$I_{DD}$		200	mA	

### AC Characteristics

$T_A = -10$  to  $+70^\circ\text{C}$ ;  $V_{DD} = +5.0\text{ V} \pm 10\%$ ; see figure 1

Parameter	Figure	Symbol	Min	Max	Unit	Condition	
<b>Clock (CLK, SCLK)</b>							
Clock period	CLK	2	$t_{CYK}$	125	600	ns	
	SCLK	2	$t_{CYSK}$	125	600	ns	$t_{CYK} \leq t_{CYSK}$
High-level clock width	CLK	2	$t_{WKH}$	52		ns	
	SCLK	2	$t_{WSKH}$	52		ns	
Low-level clock width	CLK	2	$t_{WKL}$	52		ns	
	SCLK	2	$t_{WSKL}$	52		ns	
Clock rise time	CLK	2	$t_{KR}$		15	ns	
	SCLK	2	$t_{SKR}$		15	ns	
Clock fall time	CLK	2	$t_{KF}$		15	ns	
	SCLK	2	$t_{SKF}$		15	ns	
<b>Reset, Interrupt</b>							
Reset pulse width	3	$t_{RST}$	5			$t_{CYSK}$	
CLK $\uparrow$ to INT $\uparrow$ delay time	3	$t_{DKI}$		50		ns	$C_L = 50\text{ pF}$
$\overline{RD} \downarrow$ to INT $\downarrow$ delay time	3	$t_{DRI}$		$3 t_{CYK} + 50$		ns	STATUS read
<b>HLDRQ, HLDAR</b>							
CLK $\uparrow$ to HLDAR delay time	4	$t_{DKHA}$		50		ns	$C_L = 50\text{ pF}$
HLDRQ setup time to CLK $\uparrow$	4	$t_{SKHQ}$	20			ns	
HLDRQ hold time from CLK $\uparrow$	4	$t_{HKHQ}$	20			ns	
<b>DMA Read/Write Cycle</b>							
CLK $\uparrow$ to DMARQ output delay time	5,6	$t_{DKMQ}$		50		ns	$C_L = 50\text{ pF}$
DMARQ setup time to $\overline{DMAAK} \downarrow$	5,6	$t_{SMAMQ}$	0			ns	
$\overline{DMAAK}$ setup time to $\overline{RD} \downarrow$	5	$t_{SRMA}$	0			ns	
$\overline{DMAAK}$ hold time from $\overline{RD} \uparrow$	5	$t_{HRMA}$	0			ns	
$\overline{DMAAK}$ setup time to $\overline{WR} \downarrow$	6	$t_{SWMA}$	0			ns	
$\overline{DMAAK}$ hold time from $\overline{WR} \uparrow$	6	$t_{HWMA}$	0			ns	
<b>Display Memory Bus Read Cycle</b>							
CLK $\uparrow$ to address or data output delay time	4,7,8	$t_{DKA}$		30		ns	$C_L = 50\text{ pF}$
Input data setup time to CLK $\uparrow$	7	$t_{SKD}$	20			ns	
Input data hold time from CLK $\uparrow$	7	$t_{HKD}$	0			ns	
CLK $\uparrow$ to DASTB $\uparrow$ delay time	7,8	$t_{DKDSH}$		30		ns	$C_L = 50\text{ pF}$
CLK $\downarrow$ to DASTB $\downarrow$ delay time	7,8	$t_{DKDSL}$		30		ns	
CLK $\uparrow$ to $\overline{DRD}$ delay time	7	$t_{DKDR}$		30		ns	
CLK $\uparrow$ to $\overline{DWR}$ delay time	8	$t_{DKDW}$		30		ns	
<b>System Bus Read Cycle</b>							
$\overline{CS}$ setup time to $\overline{RD} \downarrow$	9	$t_{SRC}$	0			ns	
$\overline{CS}$ hold time from $\overline{RD} \uparrow$	9	$t_{HRC}$	0			ns	
$\overline{RD}$ width, high	5,9	$t_{WRH}$	50			ns	
ASTB pulse width	5,6,9,10	$t_{WAS}$	30			ns	

## AC Characteristics (cont)

Parameter	Figure	Symbol	Min	Max	Unit	Condition
ASTB setup time to RD ↓	5,9	t <sub>SRAS</sub>	0		ns	
Address setup time to ASTB ↓	5,6,9,10	t <sub>SASA</sub>	20		ns	
Address hold time from ASTB ↓	5,9	t <sub>HASA</sub>	0		ns	
Data setup time to READY ↑	5,9	t <sub>SRVD</sub>	0		ns	
Data float delay time from RD ↑	5,9	t <sub>FRD</sub>	0	40	ns	
RD ↓ to READY ↓ delay time	5,9	t <sub>DRRY</sub>		30	ns	C <sub>L</sub> = 50 pF
RD hold time from READY ↑	5,9	t <sub>HRYR</sub>	0		ns	
CLK ↑ to READY ↑ delay time	5,9	t <sub>DKRY</sub>		40	ns	C <sub>L</sub> = 50 pF
RD ↑ to ASTB ↑ delay time	5,9	t <sub>DRAS</sub>	0		ns	

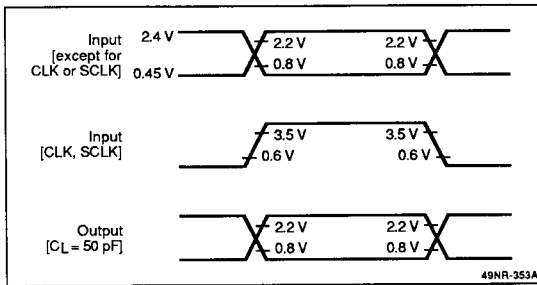
## System Bus Write Cycle

CS setup time to WR ↓	10	t <sub>SWC</sub>	0		ns	
CS hold time from WR ↑	10	t <sub>HWC</sub>	0		ns	
WR width, low	6,10	t <sub>WWL</sub>	50		ns	
WR width, high	6,10	t <sub>WWH</sub>	50		ns	
Data setup time to WR ↑	6,10	t <sub>SWD</sub>	50		ns	
Data hold time from WR ↑	6,10	t <sub>HWD</sub>	0		ns	
WR ↓ to READY ↓ delay time	6,10	t <sub>DWRY</sub>		30	ns	C <sub>L</sub> = 50 pF
WR hold time from READY ↑	6,10	t <sub>HRYW</sub>	50		ns	
CLK ↑ to READY ↑ delay time	6,10	t <sub>DKRY</sub>		40	ns	C <sub>L</sub> = 50 pF
ASTB setup time to WR ↓	6,10	t <sub>SWAS</sub>	0		ns	
WR ↑ to ASTB ↑ delay time	6,10	t <sub>DWAS</sub>	0		ns	

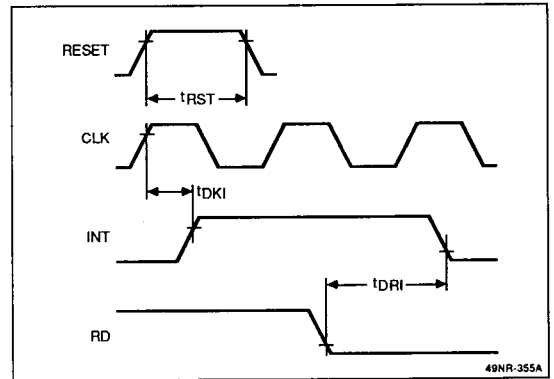
## Display Cycle

SCLK ↑ to DASTB ↑ delay time	11,12,13	t <sub>DSKDASH</sub>		30	ns	C <sub>L</sub> = 50 pF
SCLK ↓ to DASTB ↓ delay time	11,12,13	t <sub>DSKDASL</sub>		30	ns	
SCLK ↑ to D <sub>T</sub> /DISP delay time	11,12,13	t <sub>DSKDT</sub>		30	ns	
SCLK ↑ to address delay time	11,12,13	t <sub>DSKA</sub>		30	ns	
SCLK ↑ to output signal delay time (HS, VS, BLANK, or GCSR)	11,12,13	t <sub>DSKO</sub>		50	ns	
SCLK ↑ to WAIT delay time	11,12	t <sub>DSKWT</sub>		70	ns	
WAIT pulse width	11	t <sub>WWT</sub>	4t <sub>CYSK</sub> - 70		ns	
EXVS setup time to SCLK ↑	11	t <sub>SSKEV</sub>	20		ns	
EXHS setup time to SCLK ↑	11	t <sub>SSKEH</sub>	20		ns	
EXVS hold time from SCLK ↑	11	t <sub>HSKEV</sub>	20		ns	
EXHS hold time from SCLK ↑	11	t <sub>HSKEH</sub>	20		ns	

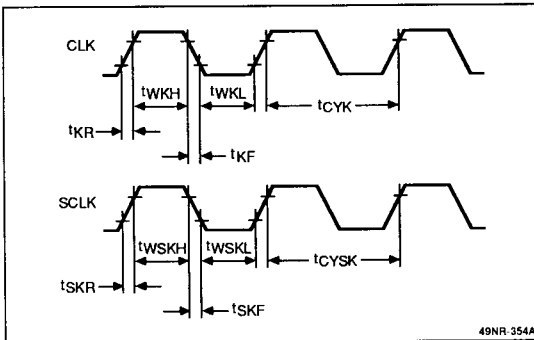
**Figure 1. Voltage Thresholds for Timing Measurements**



**Figure 3. Reset and Interrupt Waveforms**



**Figure 2. Clock Waveforms**



**Figure 4. HLDRQ and HLDK Waveforms**

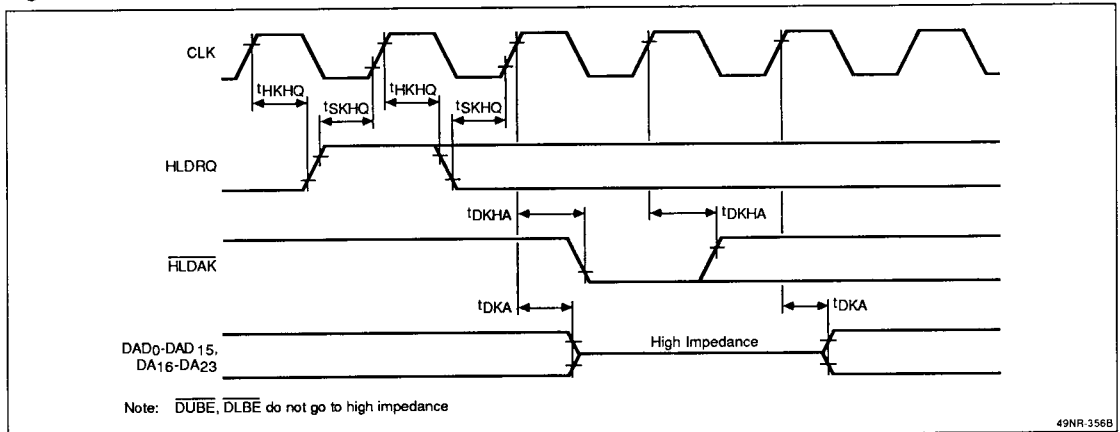


Figure 5. DMA Read Cycle

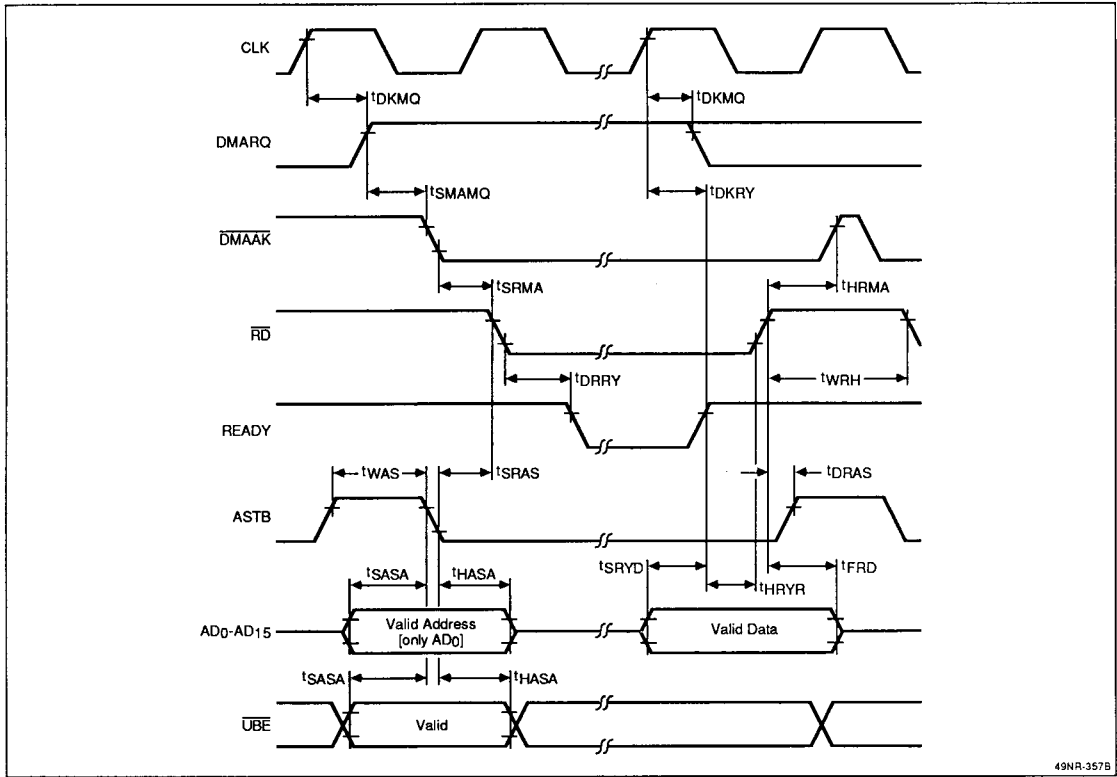
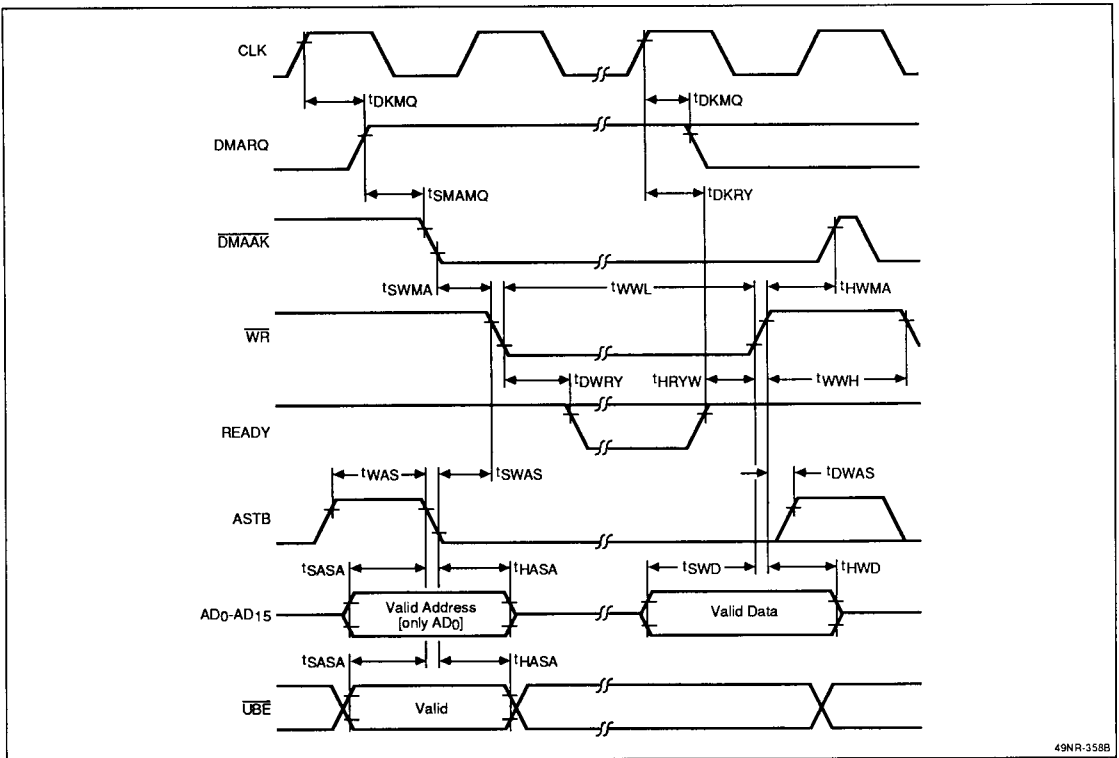
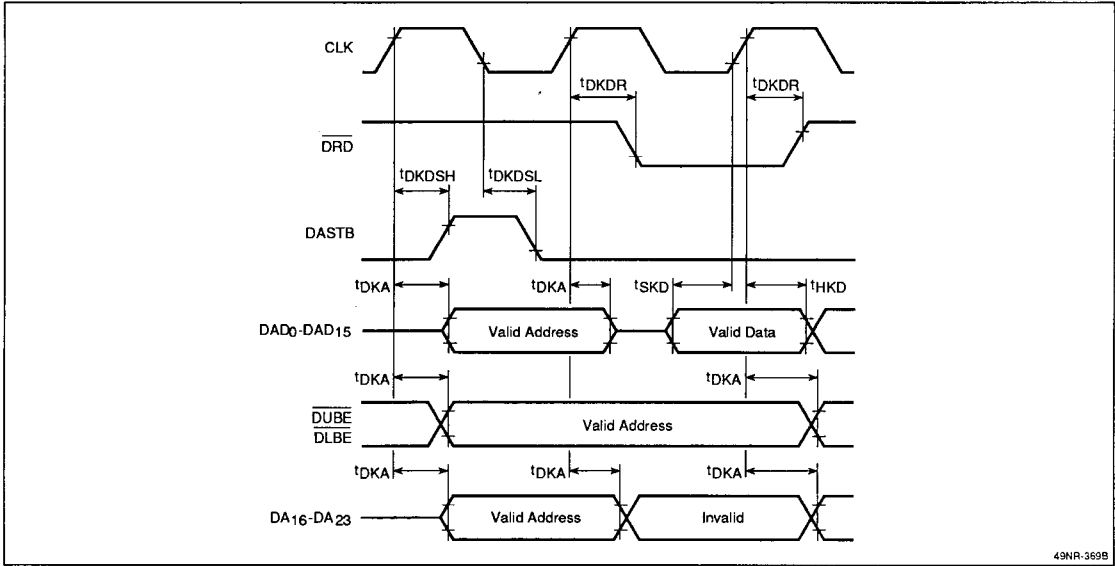


Figure 6. DMA Write Cycle

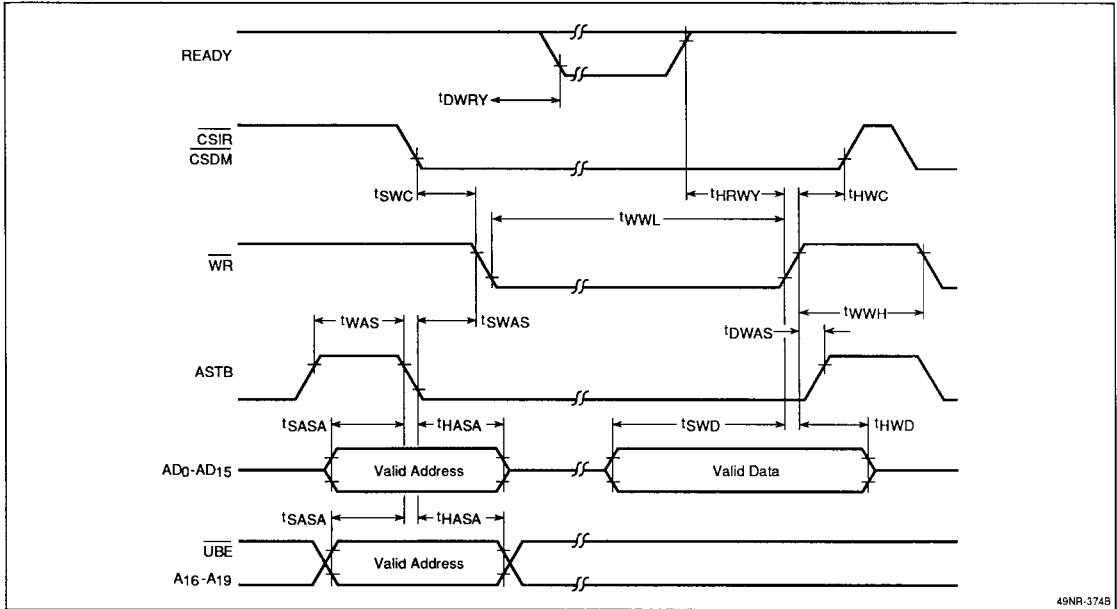


**Figure 7. Display Memory Bus Read Cycle**



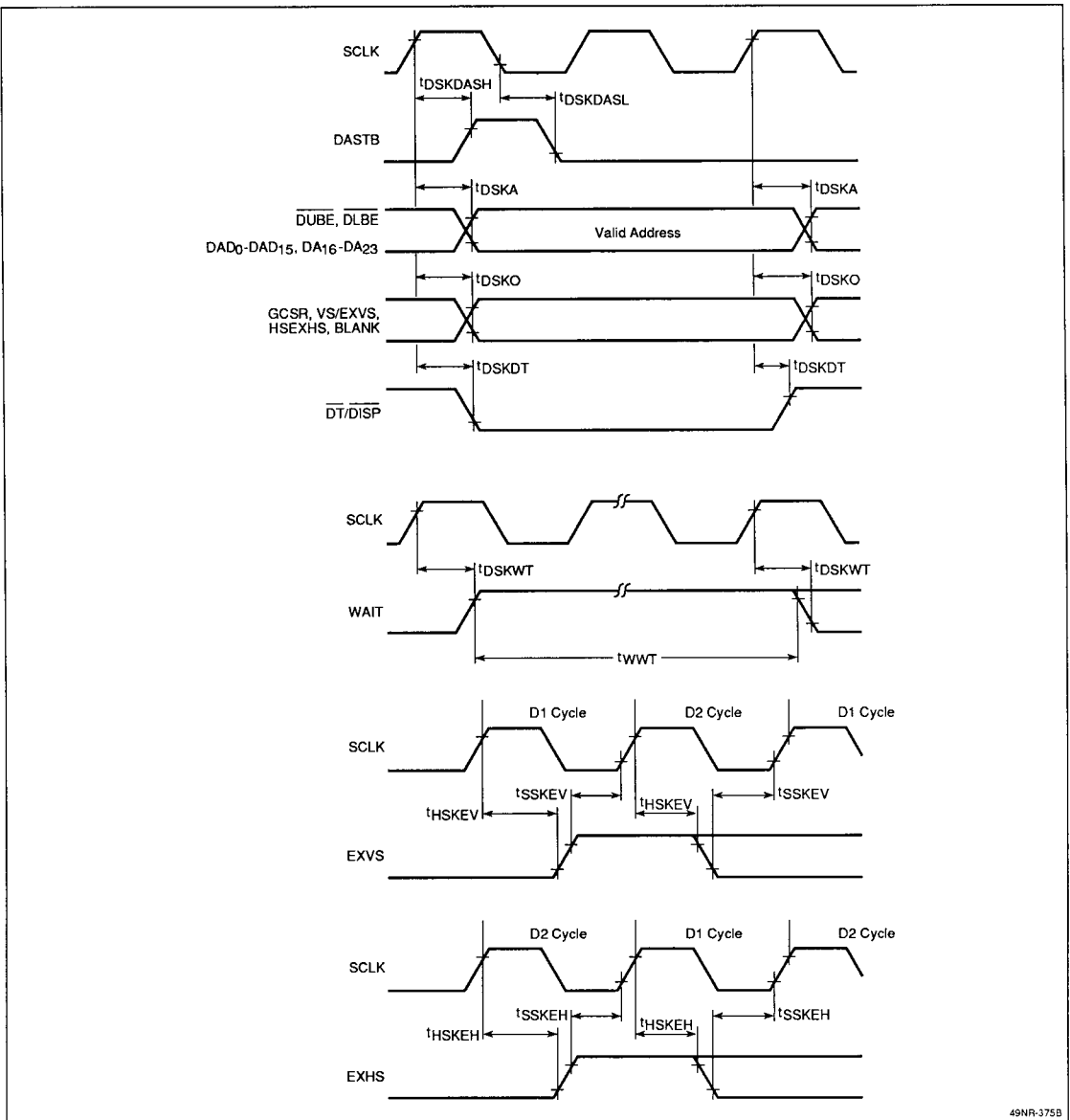


**Figure 10. System Bus Write Cycle**

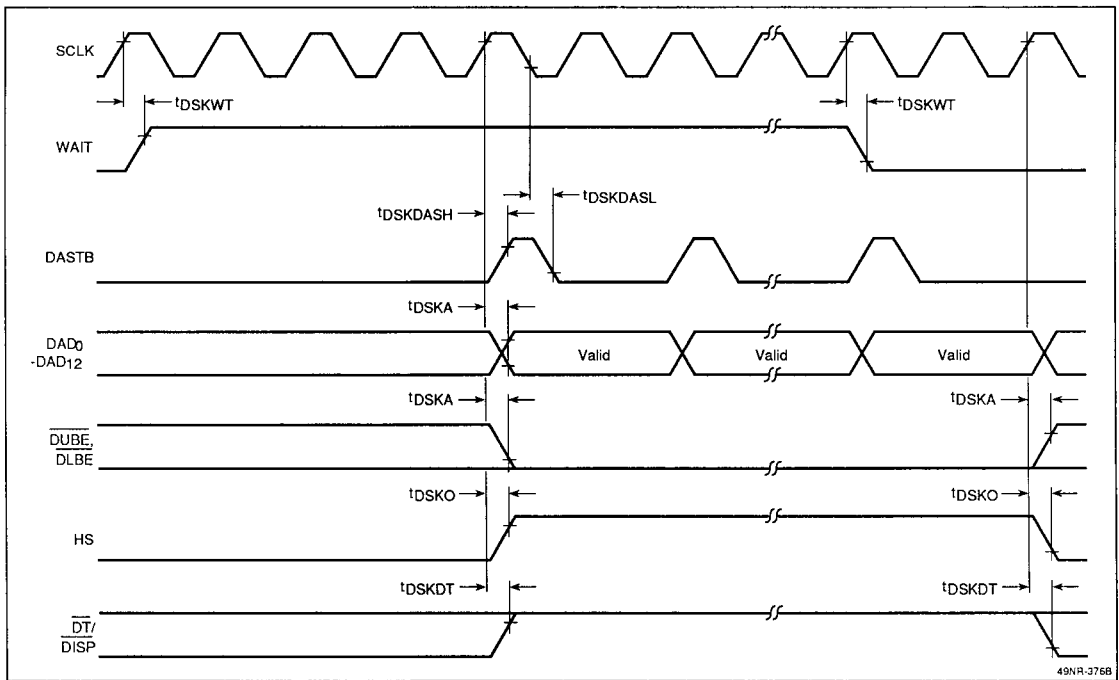


3

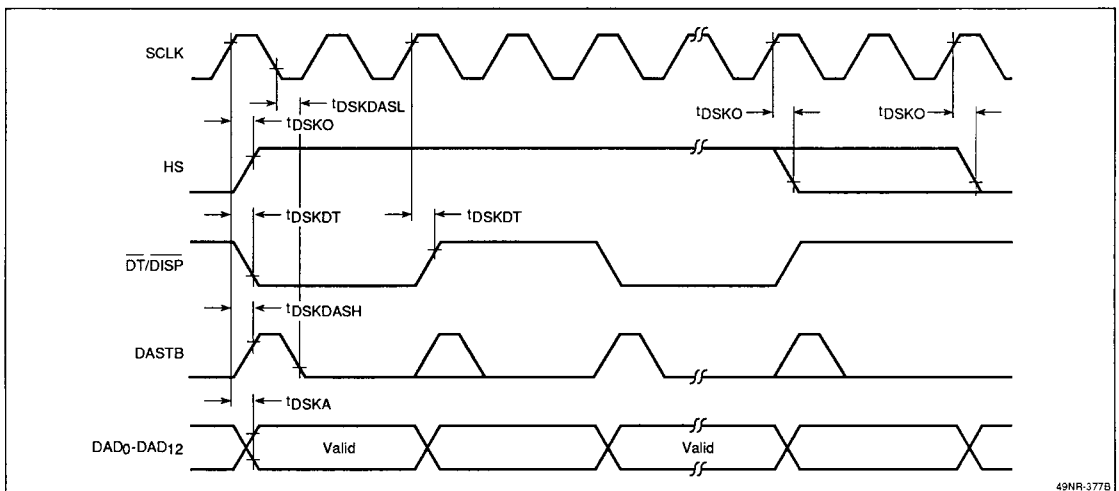
**Figure 11. Display Cycle**



**Figure 12. Display Refresh Cycle (DT Mode)**



**Figure 13. Display Refresh Cycle (CS Mode)**



### FUNCTIONAL DESCRIPTION

#### Preprocessor

The preprocessor includes a 56-word parameter RAM, an arithmetic logic unit, and a general-purpose register. It carries out the following drawing preprocessing by microprogram control.

- Conversion between coordinate and physical addresses
- Command interpretation
- Drawing parameter generation
- Calculation of tiling pattern position
- Sorting of vertex coordinates for triangular fill command
- Error checking on user-defined parameters
- Data passing with drawing processor
- Drawing processor initiation

Along with the drawing processor, the preprocessor forms part of a three-stage pipeline to improve throughput.

#### Drawing Processor

The drawing processor carries out the drawing operations on the display memory with the commands and parameters generated by the preprocessor. The drawing processor includes various arithmetic units, a general-purpose register, an arithmetic logic unit, and mask generating circuitry. In addition, it contains a 32-bit barrel shifter for high-speed bit-boundary processing operations and a 90-degree rotation data buffer. These components are controlled by a horizontal-type microprogram that can execute five types of instructions simultaneously in a single step.

#### Display Processor

An external dot-shifter for parallel-to-serial conversion is generally necessary to create scan line information for display on a CRT. The display processor generates display addresses to supply the image data to the dot shifter. This processor includes a DRAM refresh controller to generate refresh addresses during the horizontal sync active period. The display controller also controls the generation of refresh and display addresses for dual-port DRAMs (video RAMs), DRAMs, and SRAMs.

#### Sync Signal Generator

The sync signal generator produces horizontal and vertical sync signals and blank signals according to the parameters set by the user. This circuitry also generates the graphics cursor signal that can be used (with external circuitry) to generate a screen cursor.

#### CPU Interface Unit

The CPU interface unit includes a DMA interface (DMARQ, DMAAK) and an interrupt (INT) control circuit. The unit controls timing for system bus communications.

#### Display Memory Interface Unit

This interface unit controls the drawing, display, and refresh address outputs. It also controls the display memory bus arbitration for direct access to the display memory by other processors.

### REGISTERS

Table 1 lists the registers according to four classifications: control, display, drawing, and data port. Figure 14 shows the register configurations in numerical order by register address from 00H to 7FH.

Also in numerical order by address are the register descriptions in table 2. Figures listed below supplement the descriptions.

Figure	Title
15	Raster Operations; Replace and XOR
16	Raster Operations; AND and OR
17	Status Register Configuration
18	Display Memory Address Generation
19	Control Register Configuration
20	Definition of Clipping Rectangle
21	Display Control Register Configuration
22	Cursor Position Registers
23	Horizontal and Vertical Sync Timing Diagram

### DRAWING OPERATIONS

The DRAW command is written to the COMMAND register at address 6EH-6FH. The opcode in register 6FH determines the type of drawing. Various combinations of the command are selected by flags in register 6EH.

Table 3 lists the commands in five categories: data read, graphics drawing, fill, copy, and PUT/GET. Table 4 describes the commands and shows the register configuration.

Figures listed below give examples of DRAW commands.

Figure	Title
24	Graphics Drawing Commands
25	Fill and Paint Commands
26	Copy Commands; Copy, Rotate, Slant
27	Copy Commands; Enlarge/Shrink, Rotate

Table 5 summarizes the DRAW commands. Table 6 describes the 20 operation flags that can be set in register 6EH.

**Table 1. Register Classifications**

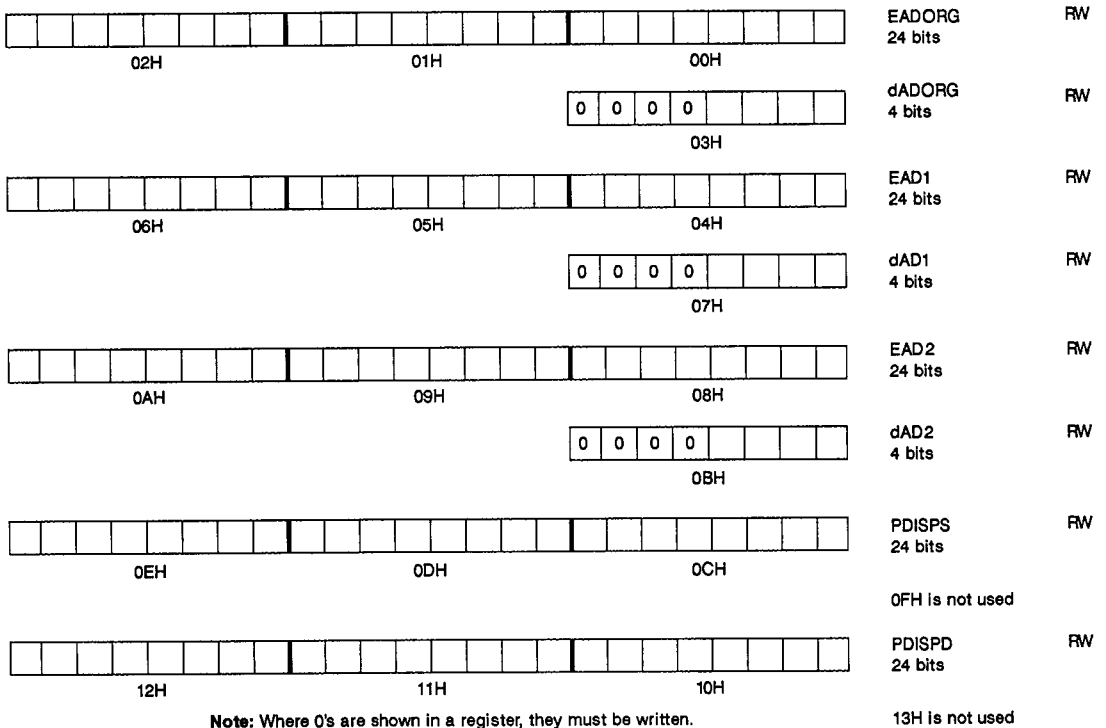
Classification	Application	Register Name	Address (Hex)	Bits
μPD72120 AGDC control registers	Status	STATUS	3C-3D	9
	Control	CTRL	3D	8
	Higher 8 bits of address in display memory direct access	BANK	3C	8
Display-related registers	Display status setting	DISPLAY CTRL	70-71	16
	Display area setting	DISPLAY PITCH	72-73	12
		AC	73	3
		DAD	74-76	24
		WC(L)	77	8
		WC(H)	7D	4
	Cursor setting	CRS	79	1
		CE	79	1
		GCSRX	78-79	12
		GCSRYS	7A-7B	12
		GCSRYS	7C-7D	12
Horizontal sync signal setting	HS, HBP HH, HD, HFP	7E-7F	12	
Vertical sync signal setting	VS, VBP, L/F, VFP	7E-7F	12	
Drawing-related registers	Logical address zero point setting	EADORG	00-02	24
		dADORG	03	4
	Logical address setting	PITCHS	58-59	16
		PITCHD	5A-5B	16
	Plane setting	PDISPS	0C-0E	24
		PDISPD	10-12	24
		PMAX	14-15	16
	Interplane logical operation setting	MOD0	16	4
		MOD1	16	4
		PLANES	5E-5F	16
	Clipping setting	XCLMIN	62-63	16
		YCLMIN	64-65	16
		XCLMAX	66-67	16
		YCLMAX	68-69	16
		CLIP	6D	2
	Enlarge/shrink coefficient setting	MAGH	6C	4
		MAGV	6C	4
Painting pattern setting	PTNP	18-1A	24	
	PTNCNT	60-61	16	
AGDC work area setting	STACK	1C-1E	24	
	STMAX	5C-5D	16	
Physical address (word address) value setting	EAD1	04-06	24	
	EAD2	08-0A	24	

**Table 1. Register Classifications (cont)**

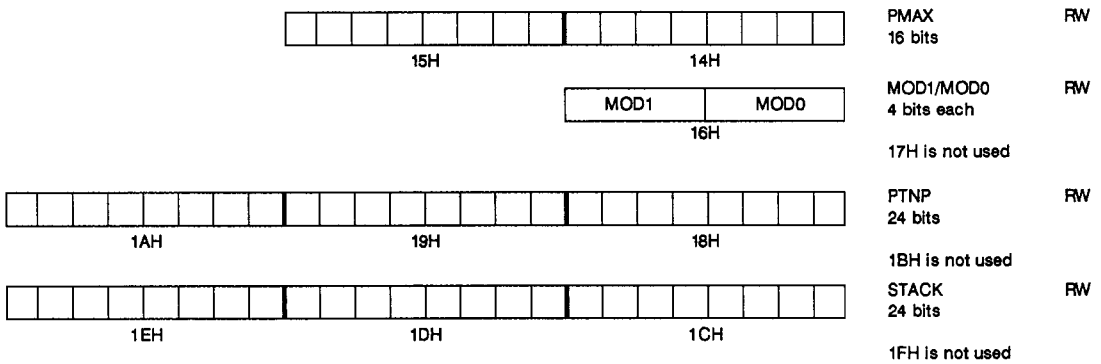
Classification	Application	Register Name	Address (Hex)	Bits
Drawing related-registers (cont)	Physical address (dot address) value setting	dAD1	07	4
		dAD2	0B	4
	Logical address (X coordinate) value setting	X	40-41	16
		DX*	44-45	16
		XS	48-49	16
		XE	4C-4D	16
		XC	50-51	16
		DH	54-55	16
	Logical address (Y coordinate) value setting	Y	42-43	16
		DY	46-47	16
		YS	4A-4B	16
		YE	4E-4F	16
YC		52-53	16	
DV		56-57	16	
Data port registers	Command	COMMAND	6E-6F	16
	Data port during execution of PUT/GET	PGPORT	3E-3F	16
	Data port during execution of READ DP/READ COL	DX*	44-45	16

\* The DX register is used as the logical address (X coordinate) value setting register and at the same time as the data port during the execution of a READ DP or READ COL command.

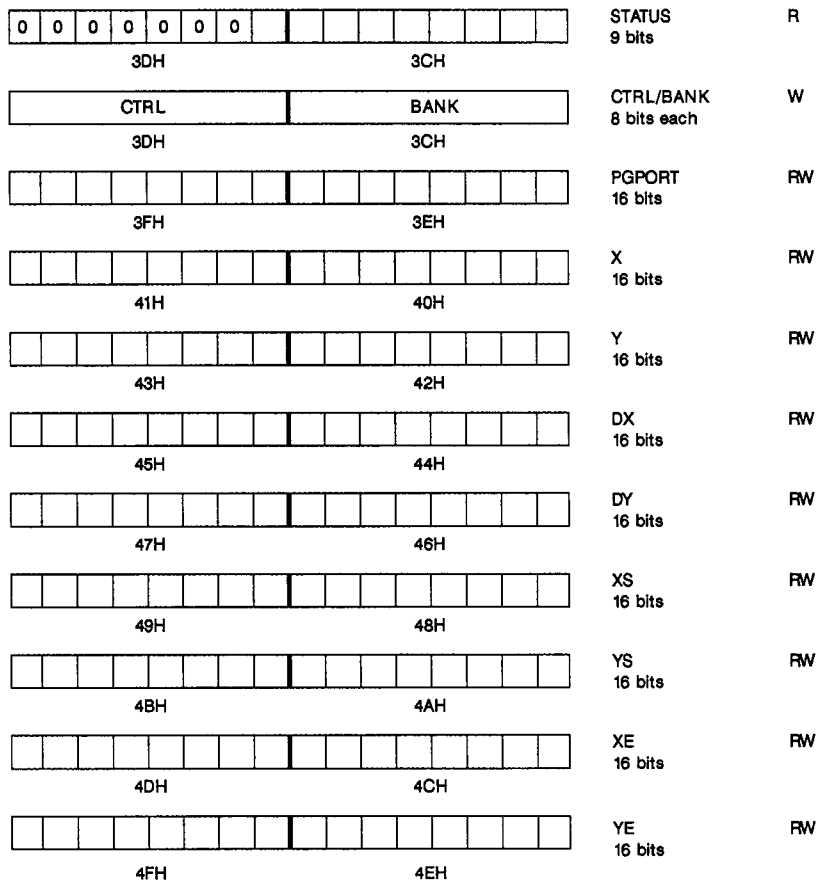
**Figure 14. Register Configurations**



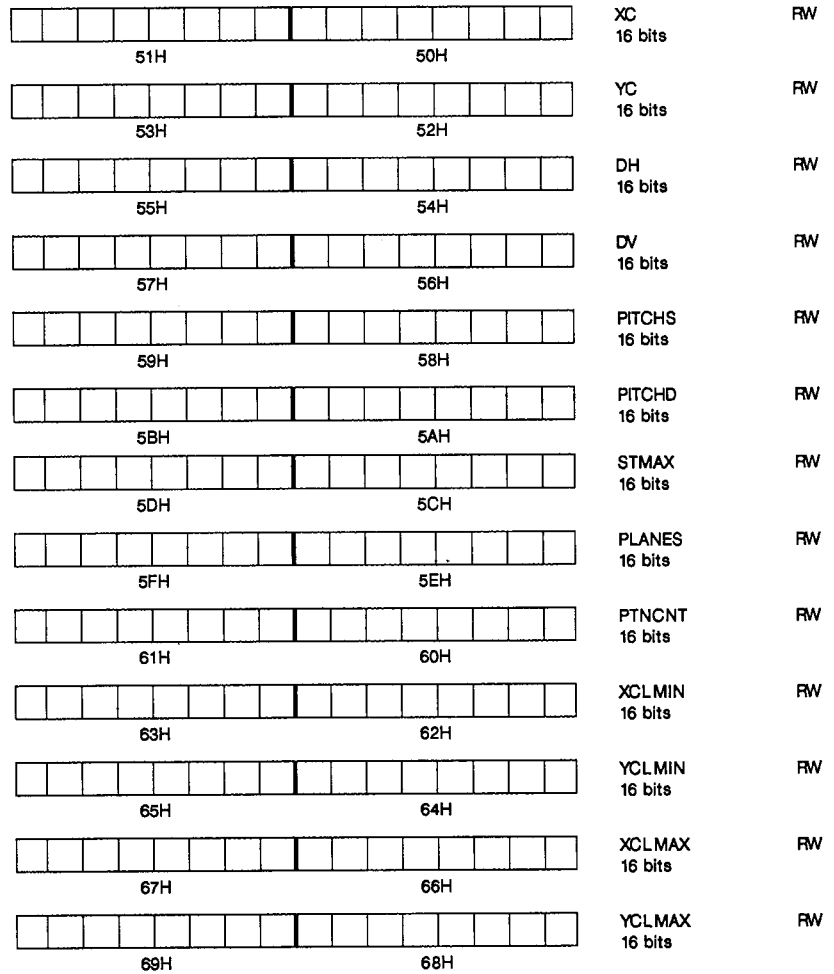
**Figure 14. Register Configurations (cont)**



Addresses 20H-3BH are used as internal working registers. These addresses are not available to the user.



**Figure 14. Register Configurations (cont)**



Addresses 6AH-6BH are used as internal working registers. They are not available to the user.

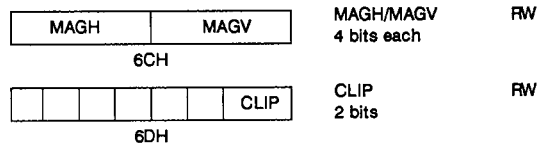
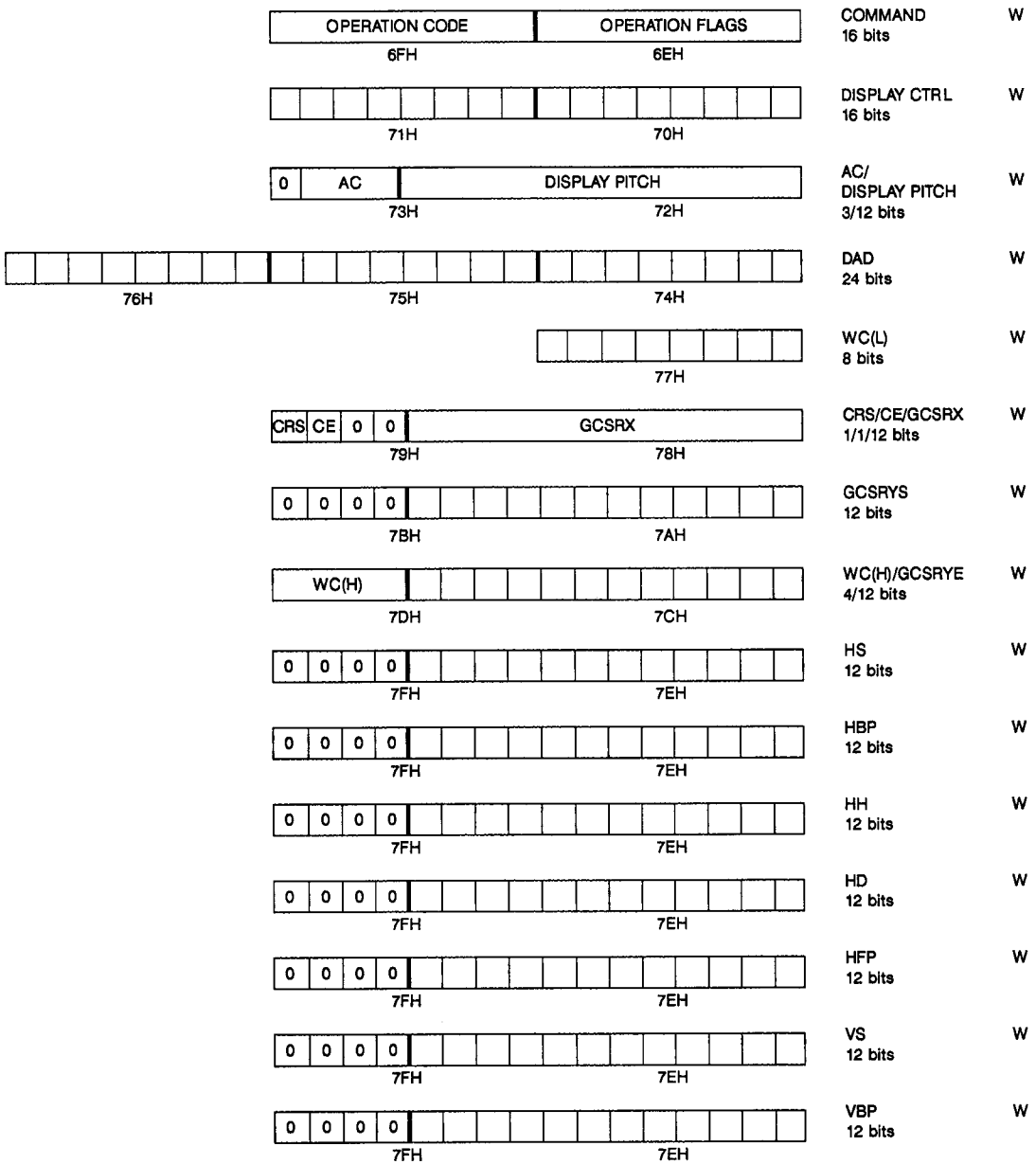
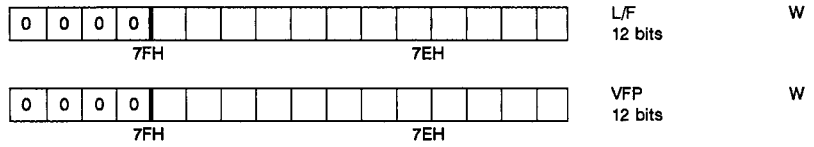


Figure 14. Register Configuration (cont)



**Figure 14. Register Configurations (cont)**



HS, HBP, HH, HD, HFP, VS, VBP, L/F and VFP are all at address 7EH-7FH and must be written in the order listed.

**Table 2. Register Descriptions**

Address (Hex)	Bits	Name	Description														
00H-02H	24	EADORG Execution Address Origin	Sets the physical address (effective address) in the display memory corresponding to the origin (0,0) on the logical plane (the X-Y coordinate plane).														
03H	4	dADORG Dot Address Origin	Sets the dot position in the physical address (effective address) in the display memory corresponding to the origin (0,0) on the logical plane (the X-Y coordinate plane).														
04H-06H	24	EAD1 Execution Address 1	Sets the drawing start physical address value in the drawing processor when the drawing start position is given by the physical address.														
07H	4	dAD1 Dot Address 1	Sets the dot position in the display memory when the drawing start position is given by the physical address														
08H-0AH	24	EAD2 Execution Address 2	Sets the drawing start physical address value in the drawing processor when the drawing start position is given by the physical address.														
0BH	4	dAD2 Dot Address 2	Sets the dot position in the display memory when the drawing start position is given by the physical address.														
0CH-0EH	24	PDISPS Plane Displacement Source	Sets the number of words that occupy one memory plane when the memory is configured with two or more planes. In the case of a COPY command, sets the number of words per source plane. In the case of a PAINT command, sets the number of words per plane containing the tiling pattern.														
10H-12H	24	PDISPD Plane Displacement Destination	Sets the number of words that occupy one memory plane when the memory is configured with two or more planes. In the case of a COPY command, sets the number of words per destination plane. In the case of a PAINT command, sets the number of words per painting plane.														
14H-15H	16	PMAX Plane Maximum	Sets the number of planes (up to 16) in the display memory to be drawn, as shown in the following table: <table border="1" style="margin-left: 20px;"> <thead> <tr> <th>PMAX</th> <th>Planes</th> </tr> </thead> <tbody> <tr> <td>0000 0000 0000 0001</td> <td>1</td> </tr> <tr> <td>0000 0000 0000 0010</td> <td>1-2</td> </tr> <tr> <td>0000 0000 0000 0100</td> <td>1-3</td> </tr> <tr> <td>0010 0000 0000 0000</td> <td>1-14</td> </tr> <tr> <td>0100 0000 0000 0000</td> <td>1-15</td> </tr> <tr> <td>1000 0000 0000 0000</td> <td>1-16</td> </tr> </tbody> </table>	PMAX	Planes	0000 0000 0000 0001	1	0000 0000 0000 0010	1-2	0000 0000 0000 0100	1-3	0010 0000 0000 0000	1-14	0100 0000 0000 0000	1-15	1000 0000 0000 0000	1-16
PMAX	Planes																
0000 0000 0000 0001	1																
0000 0000 0000 0010	1-2																
0000 0000 0000 0100	1-3																
0010 0000 0000 0000	1-14																
0100 0000 0000 0000	1-15																
1000 0000 0000 0000	1-16																
16H	4	MOD0 Drawing Mode 0	Defines the type of logical operation to be performed during drawing or copying. When the bit in the PLANES register corresponding to the memory plane is 0, the logical operation defined by MOD0 is performed. See figures 15 and 16.														
16H	4	MOD1 Drawing Mode 1	Defines the type of logical operation to be performed during drawing or copying. When the bit in the PLANES registers corresponding to the memory plane is 1, the logical operation defined by MOD1 is performed. See figures 15 and 16.														
18H-1AH	24	PTNP Pattern Pointer	Sets the first physical address in the display memory area containing the tiling (painting or filling) pattern.														

**Table 2. Register Descriptions (cont)**

Address (Hex)	Bits	Name	Description
1CH-1EH	24	STACK Stack Pointer	Sets the first physical address in the display memory area to save data such as coordinates, etc., during retrieval of the boundary points during the PAINT command (arbitrary area fill). It may be considered as the working area of the AGDC during execution of the PAINT command.
3CH-3DH	9	STATUS Status	Contains the internal status of the AGDC. The format is shown in figure 17.
3CH	8	BANK Bank	The AGDC interface to the CPU accommodates up to a 20-bit address. The AGDC can address 16M words (32M bytes) of display memory (24-bit addressing). When the CPU addresses display memory directly (through the AGDC), the lower 16 or 20 bits provided by the CPU are combined with the 8 bits from the BANK register to form the 24-bit display memory address. The address combination is shown in figure 18.
3DH	8	CTRL Control	Controls Internal AGDC processing. See figure 19.
3EH-3FH	16	PGPORT Put/Get Port	During a PUT operation, data is written to this register by the host CPU or system DMA controller. The AGDC then places the data into display memory. During a GET operation, the host CPU or DMA controller reads the data from this register that was retrieved from the display memory by the AGDC.
40H-57H	16 each	X, Y, DX, DY, XS, YS, XE, YE, XC, YC, DH, DV	Set the coordinate parameters for various drawing operations. The DX register is also used for reading the data during the READ COL command. The DH register is also used for storing half the line pattern when a 32-bit line pattern is used.
58H-59H	16	PITCHS Pitch Source	Sets the number of words in the horizontal direction of the source display memory area to be transferred.
5AH-5BH	16	PITCHD Pitch Destination	Sets the number of words in the horizontal direction of the display memory for drawing or as the destination of display memory transfer.
5CH-5DH	16	STMAX Stack Maximum	Sets the size of the display memory area in words for the STACK (used during the arbitrary area fill PAINT command). Each boundary point found during the PAINT command requires six words of memory in the STACK area.
5EH-5FH	16	PLANES Plane Select	Selects the type of logical operation to be performed on each plane during drawing or copying. Each bit in this register corresponds to a display memory plane. The least significant bit (bit 0) corresponds to the first plane, the most significant bit (bit 15) to the 16th plane. A 0 in the bit position for a plane indicates that the logical operation specified by MOD0 is to be performed and a 1, the operation specified by MOD1.
60H-61H	16	PTNCNT Pattern Count	Sets the line pattern for drawing straight and curved lines. During filling or painting operations, the function of this register depends on the TL bit as follows. TL = 1 PTNCNT specifies the length (in words) of the tiling pattern in display memory. The starting address is contained in the PTNP register. TL = 0 PTNCNT contains the actual 16-bit pattern to be used as the tiling pattern.
62H-69H	16 each	XCLMIN, YCLMIN, XCLMAX, YCLMAX X and Y Clipping, Minimum/Maximum Values	Defines the rectangular clipping region. An example is shown in figure 20.
6CH	4	MAGH Horizontal Magnification	Sets the horizontal enlarge/shrink factor.
6CH	4	MAGV Vertical Magnification	Sets the vertical enlarge/shrink factor.

**Table 2. Register Descriptions (cont)**

Address (Hex)	Bits	Name	Description																								
6DH	2	CLIP Clipping Mode	Sets the clipping mode to select one of the following operations.																								
			<table border="1"> <thead> <tr> <th>CLIP</th> <th>Function</th> </tr> </thead> <tbody> <tr> <td>00</td> <td>Draws within the clipping rectangle. Must be in this mode for PAINT.</td> </tr> <tr> <td>01</td> <td>No clipping operation</td> </tr> <tr> <td>10</td> <td>Draws outside the clipping rectangle</td> </tr> <tr> <td>11</td> <td>Prohibited</td> </tr> </tbody> </table>	CLIP	Function	00	Draws within the clipping rectangle. Must be in this mode for PAINT.	01	No clipping operation	10	Draws outside the clipping rectangle	11	Prohibited														
CLIP	Function																										
00	Draws within the clipping rectangle. Must be in this mode for PAINT.																										
01	No clipping operation																										
10	Draws outside the clipping rectangle																										
11	Prohibited																										
6EH-6FH	16	COMMAND	Commands to be executed by the AGDC are written to this register. The lower byte (bits 0-7) consists of operation flags and the upper byte (bits 8-15), an operation code. Processing begins when an operation code is written to the COMMAND register.																								
70H-71H	16	DISPLAY CTRL Display Control	Sets the operation of the display processor and sync signal generation. The format and function are shown in figure 21.																								
72H-73H	12	DISPLAY PITCH	Sets the total number of words in the horizontal direction (width) of a plane.																								
			<table border="1"> <thead> <tr> <th>Display Pitch</th> <th>Number of addresses (words)</th> </tr> </thead> <tbody> <tr> <td>0000 0000 0000</td> <td>4096</td> </tr> <tr> <td>0000 0000 0001</td> <td>1</td> </tr> <tr> <td>0000 0000 0010</td> <td>2</td> </tr> <tr> <td>0000 0000 0011</td> <td>3</td> </tr> <tr> <td>:</td> <td>:</td> </tr> <tr> <td>1111 1111 1110</td> <td>4094</td> </tr> <tr> <td>1111 1111 1111</td> <td>4095</td> </tr> </tbody> </table>	Display Pitch	Number of addresses (words)	0000 0000 0000	4096	0000 0000 0001	1	0000 0000 0010	2	0000 0000 0011	3	:	:	1111 1111 1110	4094	1111 1111 1111	4095								
Display Pitch	Number of addresses (words)																										
0000 0000 0000	4096																										
0000 0000 0001	1																										
0000 0000 0010	2																										
0000 0000 0011	3																										
:	:																										
1111 1111 1110	4094																										
1111 1111 1111	4095																										
73H	3	AC Address Control	Defines which address bus signal lines should be used to output the refresh address.																								
			<table border="1"> <thead> <tr> <th>AC</th> <th>Refresh address output pins</th> <th>Conditions for setting DT active</th> </tr> </thead> <tbody> <tr> <td>000</td> <td>DAD<sub>0</sub>-DAD<sub>12</sub></td> <td>DAD<sub>0</sub>-DAD<sub>7</sub> = 0</td> </tr> <tr> <td>001</td> <td>Disabled</td> <td>Disabled</td> </tr> <tr> <td>010</td> <td>Disabled</td> <td>Disabled</td> </tr> <tr> <td>011</td> <td>Disabled</td> <td>Disabled</td> </tr> <tr> <td>100</td> <td>DAD<sub>1</sub>-DAD<sub>12</sub></td> <td>DAD<sub>1</sub>-DAD<sub>8</sub> = 0</td> </tr> <tr> <td>101</td> <td>DAD<sub>2</sub>-DAD<sub>12</sub></td> <td>DAD<sub>2</sub>-DAD<sub>9</sub> = 0</td> </tr> <tr> <td>110</td> <td>DAD<sub>3</sub>-DAD<sub>12</sub></td> <td>DAD<sub>3</sub>-DAD<sub>10</sub> = 0</td> </tr> <tr> <td>111</td> <td>DAD<sub>4</sub>-DAD<sub>12</sub></td> <td>DAD<sub>4</sub>-DAD<sub>11</sub> = 0</td> </tr> </tbody> </table>	AC	Refresh address output pins	Conditions for setting DT active	000	DAD <sub>0</sub> -DAD <sub>12</sub>	DAD <sub>0</sub> -DAD <sub>7</sub> = 0	001	Disabled	Disabled	010	Disabled	Disabled	011	Disabled	Disabled	100	DAD <sub>1</sub> -DAD <sub>12</sub>	DAD <sub>1</sub> -DAD <sub>8</sub> = 0	101	DAD <sub>2</sub> -DAD <sub>12</sub>	DAD <sub>2</sub> -DAD <sub>9</sub> = 0	110	DAD <sub>3</sub> -DAD <sub>12</sub>	DAD <sub>3</sub> -DAD <sub>10</sub> = 0
AC	Refresh address output pins	Conditions for setting DT active																									
000	DAD <sub>0</sub> -DAD <sub>12</sub>	DAD <sub>0</sub> -DAD <sub>7</sub> = 0																									
001	Disabled	Disabled																									
010	Disabled	Disabled																									
011	Disabled	Disabled																									
100	DAD <sub>1</sub> -DAD <sub>12</sub>	DAD <sub>1</sub> -DAD <sub>8</sub> = 0																									
101	DAD <sub>2</sub> -DAD <sub>12</sub>	DAD <sub>2</sub> -DAD <sub>9</sub> = 0																									
110	DAD <sub>3</sub> -DAD <sub>12</sub>	DAD <sub>3</sub> -DAD <sub>10</sub> = 0																									
111	DAD <sub>4</sub> -DAD <sub>12</sub>	DAD <sub>4</sub> -DAD <sub>11</sub> = 0																									
74H-76H	24	DAD Display Address	Sets the display starting address for the screen																								
77H (Lower 8 bits), 7DH (Upper 4 bits)	12	WC Word Count	Sets the number of displayed words during a horizontal scan line (while BLANK low or inactive)																								
			<table border="1"> <thead> <tr> <th>WC</th> <th>Number of displayed words</th> </tr> </thead> <tbody> <tr> <td>0000 0000 0000</td> <td>1</td> </tr> <tr> <td>0000 0000 0001</td> <td>2</td> </tr> <tr> <td>:</td> <td>:</td> </tr> <tr> <td>1111 1111 1110</td> <td>4095</td> </tr> <tr> <td>1111 1111 1111</td> <td>4096</td> </tr> </tbody> </table>	WC	Number of displayed words	0000 0000 0000	1	0000 0000 0001	2	:	:	1111 1111 1110	4095	1111 1111 1111	4096												
WC	Number of displayed words																										
0000 0000 0000	1																										
0000 0000 0001	2																										
:	:																										
1111 1111 1110	4095																										
1111 1111 1111	4096																										
78H-79H	12	GCSRX Graphics Cursor X Coordinate	Sets the X (horizontal) coordinate start for the graphics cursor output pin. It is given as the number of display cycles from the start of each horizontal scan line																								
			<table border="1"> <thead> <tr> <th>GCSRX</th> <th>Starting position on each horizontal line</th> </tr> </thead> <tbody> <tr> <td>0000 0000 0000</td> <td>Disabled</td> </tr> <tr> <td>0000 0000 0001</td> <td>1st display cycle</td> </tr> <tr> <td>0000 0000 0010</td> <td>2nd display cycle</td> </tr> <tr> <td>:</td> <td>:</td> </tr> <tr> <td>1111 1111 1110</td> <td>4094th display cycle</td> </tr> <tr> <td>1111 1111 1111</td> <td>4095th display cycle</td> </tr> </tbody> </table>	GCSRX	Starting position on each horizontal line	0000 0000 0000	Disabled	0000 0000 0001	1st display cycle	0000 0000 0010	2nd display cycle	:	:	1111 1111 1110	4094th display cycle	1111 1111 1111	4095th display cycle										
GCSRX	Starting position on each horizontal line																										
0000 0000 0000	Disabled																										
0000 0000 0001	1st display cycle																										
0000 0000 0010	2nd display cycle																										
:	:																										
1111 1111 1110	4094th display cycle																										
1111 1111 1111	4095th display cycle																										

**Table 2. Register Descriptions (cont)**

Address (Hex)	Bits	Name	Description																						
79H	1	CRS Cursor Configure Select	Determines whether the horizontal and vertical cursor position registers are ANDed or ORed together. See figure 22.  <table border="0"> <tr> <td><u>CRS</u></td> <td><u>Function</u></td> </tr> <tr> <td>0</td> <td>AND</td> </tr> <tr> <td>1</td> <td>OR</td> </tr> </table>	<u>CRS</u>	<u>Function</u>	0	AND	1	OR																
<u>CRS</u>	<u>Function</u>																								
0	AND																								
1	OR																								
79H	1	CE Cursor Display Enable	Enables the graphics cursor signal to be output on the GCSR pin.  <table border="0"> <tr> <td><u>CE</u></td> <td><u>Function</u></td> </tr> <tr> <td>0</td> <td>Disabled</td> </tr> <tr> <td>1</td> <td>Enabled</td> </tr> </table>	<u>CE</u>	<u>Function</u>	0	Disabled	1	Enabled																
<u>CE</u>	<u>Function</u>																								
0	Disabled																								
1	Enabled																								
7AH-7BH	12	GCSRYS Graphics Cursor Y Coordinate Start	Determines the starting Y (vertical) coordinate of the graphics cursor, counting display lines from the top down.  <table border="0"> <tr> <td><u>GCSRYS</u></td> <td><u>Vertical starting line</u></td> </tr> <tr> <td>0000 0000 0000</td> <td>Invalid</td> </tr> <tr> <td>0000 0000 0001</td> <td>1st display line</td> </tr> <tr> <td>0000 0000 0010</td> <td>2nd display line</td> </tr> <tr> <td>:</td> <td>:</td> </tr> <tr> <td>1111 1111 1110</td> <td>4094th display line</td> </tr> <tr> <td>1111 1111 1111</td> <td>4095th display line</td> </tr> </table>	<u>GCSRYS</u>	<u>Vertical starting line</u>	0000 0000 0000	Invalid	0000 0000 0001	1st display line	0000 0000 0010	2nd display line	:	:	1111 1111 1110	4094th display line	1111 1111 1111	4095th display line								
<u>GCSRYS</u>	<u>Vertical starting line</u>																								
0000 0000 0000	Invalid																								
0000 0000 0001	1st display line																								
0000 0000 0010	2nd display line																								
:	:																								
1111 1111 1110	4094th display line																								
1111 1111 1111	4095th display line																								
7CH-7DH	12	GCSR YE Graphics Cursor Y Coordinate End	Determines the ending Y (vertical) coordinate of the graphics cursor, counting display lines from the top down.  <table border="0"> <tr> <td><u>GCSR YE</u></td> <td><u>Vertical ending line</u></td> </tr> <tr> <td>0000 0000 0000</td> <td>Invalid</td> </tr> <tr> <td>0000 0000 0001</td> <td>1st display line</td> </tr> <tr> <td>0000 0000 0010</td> <td>2nd display line</td> </tr> <tr> <td>:</td> <td>:</td> </tr> <tr> <td>1111 1111 1110</td> <td>4094th display line</td> </tr> <tr> <td>1111 1111 1111</td> <td>4095th display line</td> </tr> </table>	<u>GCSR YE</u>	<u>Vertical ending line</u>	0000 0000 0000	Invalid	0000 0000 0001	1st display line	0000 0000 0010	2nd display line	:	:	1111 1111 1110	4094th display line	1111 1111 1111	4095th display line								
<u>GCSR YE</u>	<u>Vertical ending line</u>																								
0000 0000 0000	Invalid																								
0000 0000 0001	1st display line																								
0000 0000 0010	2nd display line																								
:	:																								
1111 1111 1110	4094th display line																								
1111 1111 1111	4095th display line																								
7EH-7FH	12	HS (Horizontal Sync), HBP (Horizontal Back Porch), HH (HBP to Midpoint Between Consecutive HSs), HD (Horizontal Drawing Period), HFP (Horizontal Front Porch)	Sets the horizontal video sync (timing) parameters. See figure 23. <table border="0"> <tr> <td>HS</td> <td>Horizontal sync high-level period (horizontal retrace)</td> </tr> <tr> <td>HBP</td> <td>Horizontal back porch (non-displayed portion on left side of screen)</td> </tr> <tr> <td>HH</td> <td>Rising/falling timing for even field synchronization during interlaced display</td> </tr> <tr> <td>HD</td> <td>Horizontal display period (active display time)</td> </tr> <tr> <td>HFP</td> <td>Horizontal front porch (non-displayed portion on right side of screen)</td> </tr> </table> <table border="0"> <tr> <td><u>HS, HBP, HH, HD, HFP</u></td> <td><u>* SCLK periods</u></td> </tr> <tr> <td>0000 0000 0000</td> <td>2 clocks</td> </tr> <tr> <td>0000 0000 0001</td> <td>4 clocks</td> </tr> <tr> <td>:</td> <td>:</td> </tr> <tr> <td>1111 1111 1110</td> <td>8190 clocks</td> </tr> <tr> <td>1111 1111 1111</td> <td>8192 clocks</td> </tr> </table> <p>*One display cycle is equal to two SCLK periods  Setting requirements  For display control by AGDC: HS, HBP, HH, HD, HBP ≥ 4 SCLK periods  For interlace display: HBP ≥ 6 SCLK periods  For AGDC in slave mode: HS ≥ 10 SCLK periods</p>	HS	Horizontal sync high-level period (horizontal retrace)	HBP	Horizontal back porch (non-displayed portion on left side of screen)	HH	Rising/falling timing for even field synchronization during interlaced display	HD	Horizontal display period (active display time)	HFP	Horizontal front porch (non-displayed portion on right side of screen)	<u>HS, HBP, HH, HD, HFP</u>	<u>* SCLK periods</u>	0000 0000 0000	2 clocks	0000 0000 0001	4 clocks	:	:	1111 1111 1110	8190 clocks	1111 1111 1111	8192 clocks
HS	Horizontal sync high-level period (horizontal retrace)																								
HBP	Horizontal back porch (non-displayed portion on left side of screen)																								
HH	Rising/falling timing for even field synchronization during interlaced display																								
HD	Horizontal display period (active display time)																								
HFP	Horizontal front porch (non-displayed portion on right side of screen)																								
<u>HS, HBP, HH, HD, HFP</u>	<u>* SCLK periods</u>																								
0000 0000 0000	2 clocks																								
0000 0000 0001	4 clocks																								
:	:																								
1111 1111 1110	8190 clocks																								
1111 1111 1111	8192 clocks																								

**Table 2. Register Descriptions (cont)**

Address (Hex)	Bits	Name	Description														
7EH-7FH	12 each	VS (Vertical Sync), VBP (Vertical Back Porch), L/F (Lines per Field), VFP (Vertical Front Porch)	Sets the vertical sync (timing) parameters. See figure 23. VS Vertical sync (retrace) high-level period VBP Vertical back porch (non-displayed portion on upper part of screen) L/F Lines per field (number of horizontal scan lines displayed) VFP Vertical front porch (non-displayed portion on lower part of screen)														
			<table border="0"> <thead> <tr> <th>VS, VBP, L/F, VFP</th> <th>*Horizontal scan lines</th> </tr> </thead> <tbody> <tr> <td>0000 0000 0000</td> <td>4096</td> </tr> <tr> <td>0000 0000 0001</td> <td>1</td> </tr> <tr> <td>0000 0000 0010</td> <td>2</td> </tr> <tr> <td>:</td> <td>:</td> </tr> <tr> <td>1111 1111 1110</td> <td>4094</td> </tr> <tr> <td>1111 1111 1111</td> <td>4095</td> </tr> </tbody> </table>	VS, VBP, L/F, VFP	*Horizontal scan lines	0000 0000 0000	4096	0000 0000 0001	1	0000 0000 0010	2	:	:	1111 1111 1110	4094	1111 1111 1111	4095
VS, VBP, L/F, VFP	*Horizontal scan lines																
0000 0000 0000	4096																
0000 0000 0001	1																
0000 0000 0010	2																
:	:																
1111 1111 1110	4094																
1111 1111 1111	4095																
			* Vertical timing parameters are set as multiples of the horizontal scan line period.														

Figure 15. Raster Operations: Replace and XOR (MOD0/MOD1)

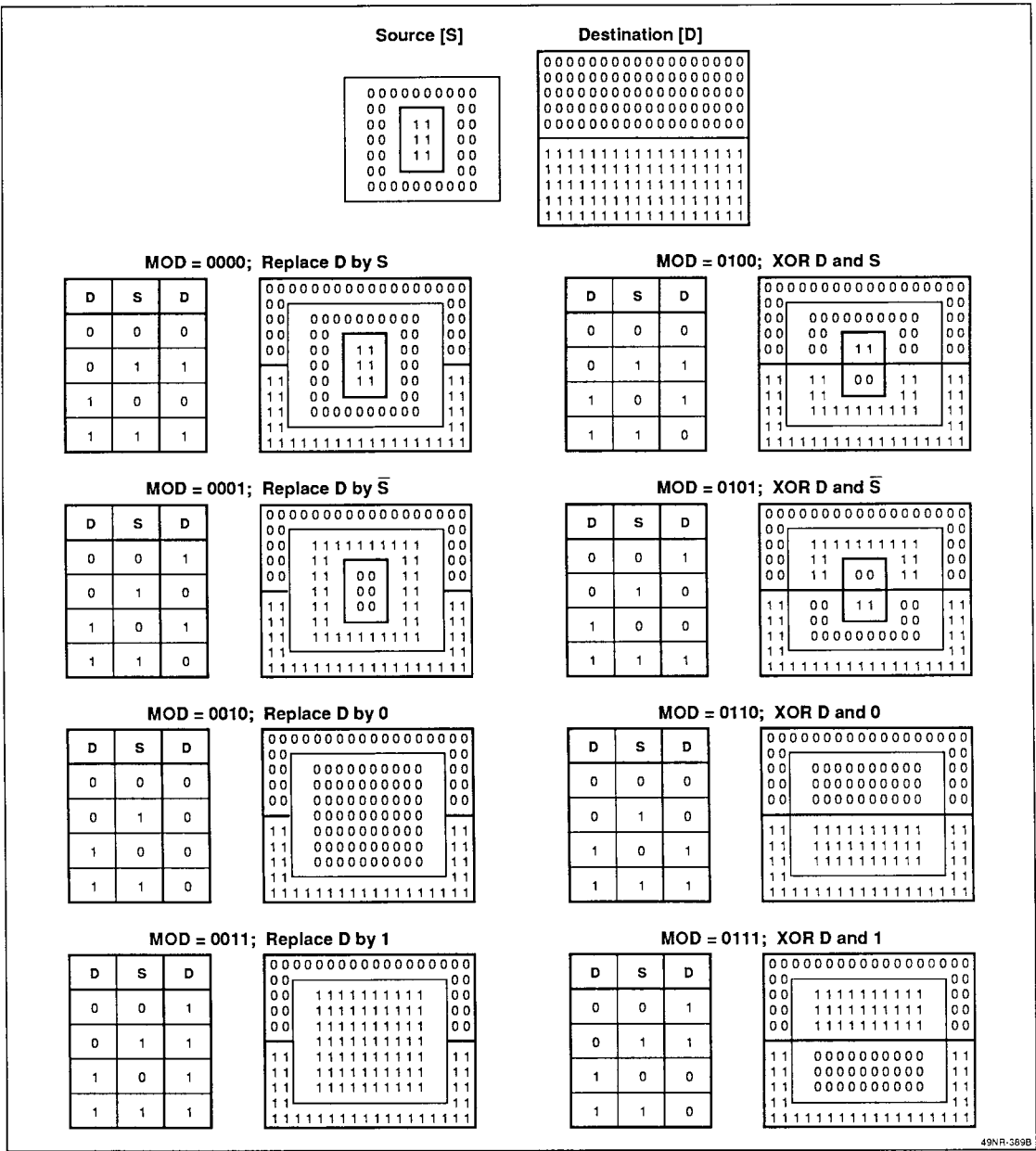
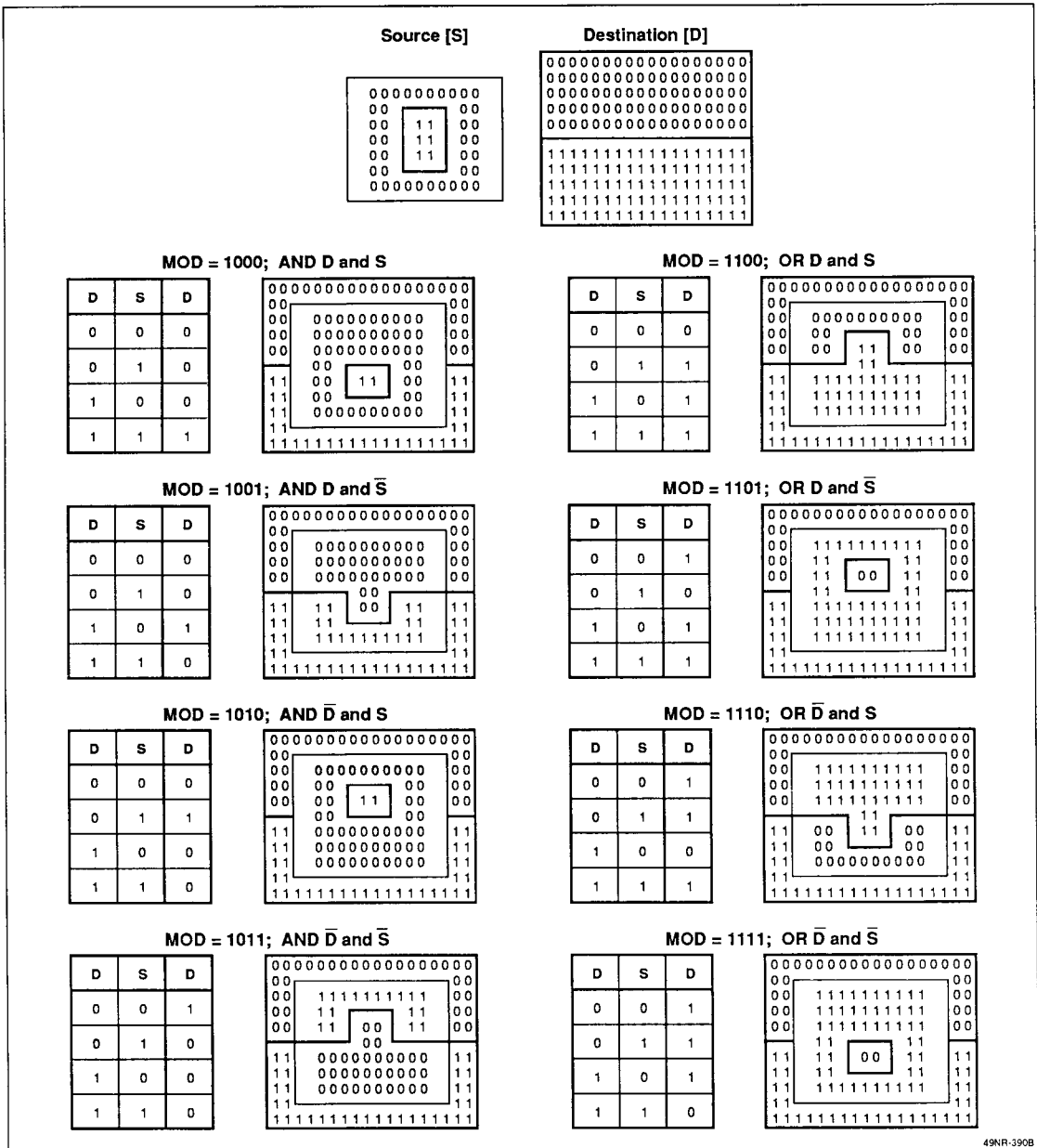
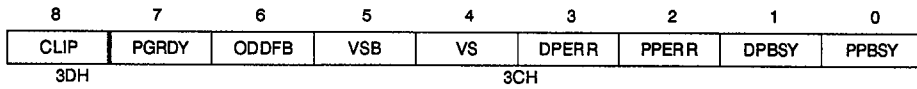


Figure 16. Raster Operations; AND and OR (MOD0/MOD1)

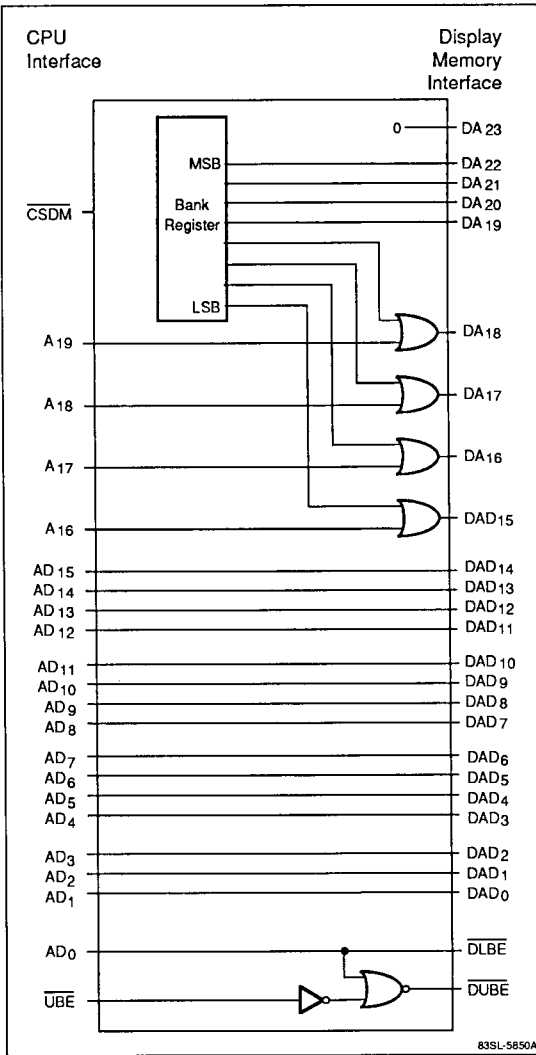


**Figure 17. Status Register Format**



Bit	Flag Name	Abbreviation	Meaning When Bit = 1
0	Preprocessor Busy	PPBSY	The preprocessor is executing a command.
1	Drawing Processor Busy	DPBSY	The drawing processor is executing a command.
2	Preprocessor Error	PPER	An error was detected during the execution of a command by the preprocessor.
3	Drawing Processor Error	DPERR	An error was detected during execution of a command by the drawing processor.
4	Vertical Sync Period	VS	Indicates vertical sync period.
5	Vertical Blanking Period	VSB	Indicates vertical blanking period.
6	Odd Field	ODDFD	Indicates odd field during interlaced operation.
7	Put/Get Ready	PGRDY	Indicates that data can be transferred during a PUT or GET command.
8	Clipping	CLIP	Picking or object detected.

**Figure 18. Display Memory Addressing**



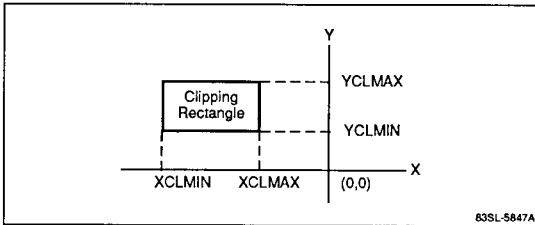
**Figure 19. Register Format**

	7	6	5	4	3	2	1	0
	DBIE	PBIE	CIE	0	0	0	ABORT	RESET

Bit	Flag Name	Abbreviation	Meaning When Bit = 1
0	Software Reset	RESET	Initializes μPD72120.
1	Processor Abort	ABORT	Stops any processing being performed and clears the processor BUSY status.
2	Not used		Must be set to 0.
3	Not used		
4	Not used		
5	Clipping Interrupt Enable	CIE	Enables the INT signal when picking (drawing in the clipped region).
6	Preprocessor Busy Interrupt Enable	PBIE	Enables the INT signal when the preprocessor status changes from BUSY to NOT BUSY.
7	Drawing Processor Busy Interrupt Enable	DBIE	Enables the INT signal when the drawing processor status changes from BUSY to NOT BUSY.

**Figure 20. Rectangular Clipping Region**



**Figure 21. Display Control Register**

	MSB	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	LSB
		DTM	DTT	DAD+		IN	RE	SC	FCCL	TCCL	MASK	M/S	SD	LFI	SPST	SVS		

Bit	Flag Name	Abbreviation	Function
0	Slave Sync	SVS	When the AGDC is in the slave mode, SVS determines the initialization of the internal horizontal and vertical counters. SVS is ignored in the master mode. <u>SVS</u> 0 Initializes the vertical and horizontal counters at the rising edge of EXVS and EXHS, respectively. 1 Initializes the vertical and horizontal counters at the rising edge of EXVS.
1	Sync Parameter Setting	SPST	Enables the writing of the sync timing parameters (HS, HBP, HH, HD, HFP, VS, VBP, L/F and VFP) to address 7EH-7FH. The writing should take place after SPST is set to 0, then to 1. <u>SPST</u> 0 Disables writing of sync parameters 1 Enables writing of sync parameters
2	Display Lines per Frame in Interlace Mode	LFI	Defines whether there is an even or odd number of lines per frame in interlaced mode. LFI is ignored in non-interlaced mode. <u>LFI</u> 0 Even total number of lines for the sum of even and odd fields (one frame). 1 Odd total number of lines for sum of even and odd fields.

3

**Figure 21. Display Control Register (cont)**

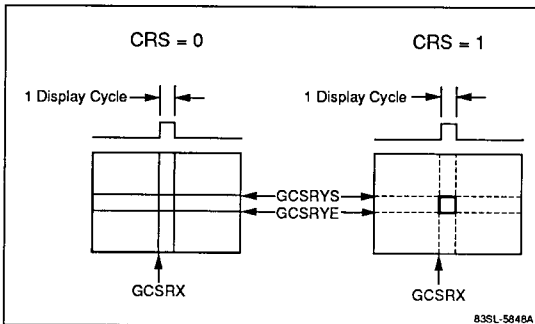
Bit	Flag Name	Abbreviation	Function
3	Stop Display	SD	<p>Defines the state of the BLANK output signal. SD is set to 1 by a high level on the RESET pin.</p> <p><u>SD</u></p> <p>0 BLANK signal active (high) only for the non-display period defined by the video sync signals.</p> <p>1 BLANK signal active for display and non-display periods (on continuously).</p>
4	Master/Slave	M/S	<p>Defines whether the AGDC is a master or a slave in terms of video sync signal generation.</p> <p><u>M/S</u></p> <p>0 Sets the AGDC to slave mode (video sync signals input through EXVS and EXHS).</p> <p>1 Sets the AGDC to master mode (generates video sync signals and outputs them through VS and HS).</p>
5	Mask	MASK	<p>Defines the VS signal output timing in the master mode. In the slave mode, defines the validity of the EXHS and EXVS sync timing input.</p> <p><u>MS</u>    <u>MASK</u></p> <p>0    0 Accepts EXHS and EXVS sync timing input.</p> <p>0    1 Ignores EXHS and EXVS sync timing input.</p> <p>1    0 Only the VS signal of the even field in interlace mode is output.</p> <p>1    1 The VS signal is output normally.</p>
6	Timing Counter Clear	TCCL	<p>Defines the timing for initializing the internal display cycle counter when the AGDC is in slave mode. TCCL is ignored when the AGDC is in master mode.</p> <p><u>TCCL</u></p> <p>0 Does not initialize the display cycle counter on the rising edge of EXVS.</p> <p>1 Initializes the display cycle counter on the rising edge of EXVS (sets the counter to the D1 cycle).</p>
7	Field Counter Clear	FCCL	<p>Defines the timing for initializing the internal field counter when using interlaced display in the slave mode. When the AGDC is in master mode or non-interlaced display, FCCL is ignored.</p> <p><u>FCCL</u></p> <p>0 Does not initialize the field counter on the rising edge of EXVS.</p> <p>1 Initializes the field counter on the rising edge of EXVS, setting the counter to the even field.</p>
8	Steal Control	SC	<p>Defines the relationship between the CLK and SCLK signals when the AGDC is in the DT mode (using video RAMs). If the AGDC is in cycle steal mode, SC is ignored.</p> <p><u>SC</u></p> <p>0 CLK does not equal SCLK.</p> <p>1 CLK and SCLK are the same</p>
9	Refresh Enable	RE	<p>Defines whether the AGDC is to generate DRAM refresh addresses.</p> <p><u>RE</u></p> <p>0 The AGDC does not generate DRAM refresh addresses</p> <p>1 The AGDC generates DRAM refresh address while HS is active (high)</p>
10	Interlace	IN	<p>Defines whether interlaced or non-interlaced display mode is to be used.</p> <p><u>IN</u></p> <p>0 Non-interlaced display</p> <p>1 Interlaced display</p>

**Figure 21. Display Control Register (cont)**

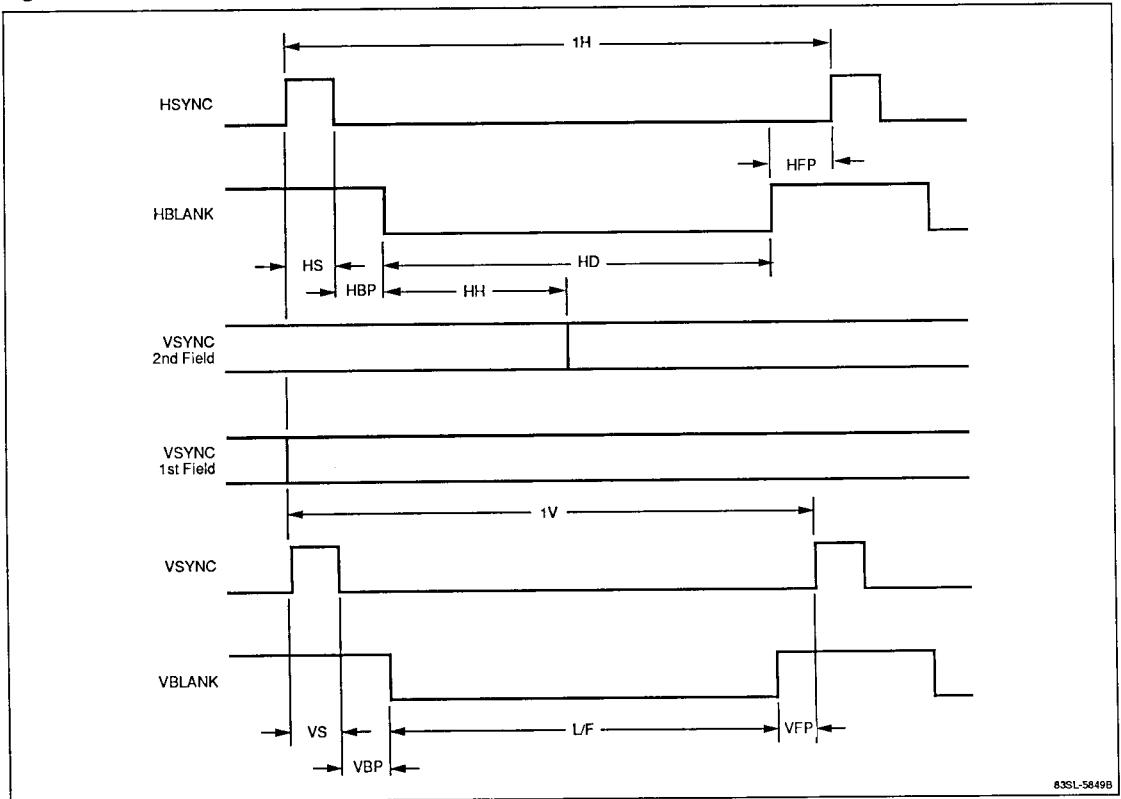
Bit	Flag Name	Abbreviation	Function
11, 12, 13	Display Address Proceedings	DAD +	<p>Defines how the AGDC's 24-bit display address register is to be incremented during each display cycle. The register is not incremented while BLANK is active. It is incremented ast each display cycle (two SCLK periods) in the DT (VRAM) mode or each time a display cycle is started in the CS (cycle steal) mode.</p> <p><b>DAD + Increment</b></p> <p>000 DAD +1 DAD → DAD +1 → DAD +2 → DAD +3 → DAD +4 . . .</p> <p>001 DAD +2 DAD → DAD +2 → DAD +4 → DAD +6 → DAD +8 . . .</p> <p>010 DAD +4 DAD → DAD +4 → DAD +8 → DAD +12 → DAD +16 . . .</p> <p>011 DAD +8 DAD → DAD +8 → DAD +16 → DAD +24 → DAD +32 . . .</p> <p>100 DAD +16 DAD → DAD +16 → DAD +32 → DAD +48 → DAD +64 . . .</p> <p>101 DAD +32 DAD → DAD +32 → DAD +64 → DAD +96 → DAD +128 . . .</p> <p>110 DAD +1/4 DAD → DAD → DAD → DAD → DAD +1 → DAD +1 → . . .</p> <p>111 DAD +1/2 DAD → DAD → DAD +1 → DAD +1 → DAD +2 → DAD +2 . . .</p>
14	Data Transfer Timing	DTT	<p>Defines the output timing for the DT (data transfer) signal when using VRAMs. DTT is ignored in the cycle steal mode.</p> <p><b>DTT</b></p> <p>0 DT is generated (active low) when any of the following conditions is true.</p> <ul style="list-style-type: none"> <li>(a) At the start of the screen display (at the first rising edge of the BLANK signal in a frame)</li> <li>(b) At the start of each horizontal scan line (at the falling edge of BLANK)</li> <li>(c) When all 8 AC register-defined bits of the 24-bit display address are 0 (when the lower 8 bits are 00H).</li> </ul> <p>1 DT is generated when any of the following conditions is true.</p> <ul style="list-style-type: none"> <li>(a) At the start of the screen display (at the first rising edge of the BLANK signal in a frame)</li> <li>(b) When all 8 AC register-defined bits of the 24-bit display address are 0.</li> </ul>
15	Data Transfer Mode	DTM	<p>Defines the display cycle generation timing. Data transfer mode is normally used with video RAMs and cycle steal mode with other types of memories.</p> <p><b>DTM</b></p> <p>0 Sets the cycle steal (SC) mode. The DT/DISP pin outputs the DISP signal (active low). Display and drawing cycles alternate in this mode.</p> <p>1 Sets the data transfer (DT) mode. The DT/DISP pin outputs the DT signal (active low).</p>

3

**Figure 22. Cursor Position Select**



**Figure 23. Horizontal and Vertical Timing Parameters**



**Table 3. List of DRAW Commands**

Commands		Absolute Coordinates	Relative Coordinates
Data Read Commands	Coordinate value read	READ_DP	
	Color information read	READ_COOL	
Graphics Drawing Commands	Dot	DOT_D	
	Straight line	A_DOT_M	R_DOT_M
		A_LINE_M0	R_LINE_M0
		A_LINE_M1	R_LINE_M1
		A_LINE_M2	R_LINE_M2
		A_LINE_D0	R_LINE_D0
		A_LINE_D1	R_LINE_D1
		A_LINE_D2	R_LINE_D2
	A_LINE_D3		
	Rectangle	A_REC	R_REC
	Circle	CRL	
	Arc	CARC	
	Circle sector	CSEC	
	Circle segment (bow)	CSEG	
	Ellipse	ELPS	
Ellipse arc	EARC		
Ellipse sector	ESEC		
Ellipse segment (bow)	ESEG		
Fill Commands	Arbitrary area fill	PAINT	
	Triangle fill	A_TRI_FILL	
	Trapezoid fill	A_TRA_FILL	
	Rectangle fill	A_REC_FILL_C	
		A_REC_FILL_A	R_REC_FILL
	Circle fill	CRL_FILL	
	Ellipse fill	ELPS_FILL	
Copy Commands	Physical address to physical address	A_COPY_AA	
	Coordinate to physical address	A_COPY_CA	
	Physical address to coordinate	A_COPY_AC	
	Coordinate to coordinate	A_COPY_CC	
	Copy function extensions	90°_COPY	
		SL_COPY	
FR_ES_COPY			
ES_COPY			
PUT/GET Commands	System memory to display memory	PUT_A	
		PUT_C	
	Display memory to system memory	GET_A	
		GET_C	
	GET function extensions	90°_GET	

**Table 4. DRAW Command Descriptions**

Commands	Name	Description
Data Read Commands	<b>READ_DP</b> Read Drawing Pointer	The current drawing pointer coordinates (X#, Y#) are output to the X and Y registers to be read by the host CPU.
	6FH	6EH
	0 0 0 0 0 1 0 0 0 0 0 0 0 0 0	
	<b>READ_COL</b> Read Color	The color information in each memory plane corresponding to the coordinates (X, Y) pointed to by the X and Y registers is placed in the DX register to be read by the host CPU. The least significant bit corresponds to the first plane, the most significant bit to the 16th plane.
	6FH	6EH
	1 0 0 1 1 1 0 0 0 0 0 0 0 0 0	
Graphics Drawing Commands	<b>DOT_D</b> Dot Direct	A dot is drawn at the current drawing pointer coordinates (X#, Y#). The drawing pointer (X#, Y#) remains unchanged. The bit pointer of the PNTCNT register shifts from the LSB by 1 bit toward the MSB .
	6FH	6EH
	0 0 0 0 1 0 0 0 0 0 IP 0 PXEN BPPX 0 0	
	<b>A_DOT_M</b> Absolute Dot with Move	A dot is drawn at the (X, Y) coordinates pointed to by the X and Y registers, respectively. The drawing pointer (X#, Y#) changes to (X, Y). The bit pointer of the PNTCNT register shifts from the LSB by 1 bit toward the MSB .
	6FH	6EH
	0 0 0 0 1 1 0 0 0 0 IP 0 PXEN BPPX 0 0	
	<b>R_DOT_M</b> Relative Dot with Move	A dot is drawn at the (X+DX, Y+DY) defined by the X, DX, Y, and DY registers, respectively. The drawing pointer (X#, Y#) changes to (X+DX, Y+DY). The bit pointer of the PNTCNT register shifts from the LSB by 1 bit toward the MSB .
	6FH	6EH
	0 0 0 1 0 0 0 0 0 0 0 IP 0 PXEN BPPX 0 0	
	Absolute Line with Move 0, 1, 2	A straight line is drawn from coordinates (X, Y) pointed to by the X and Y registers to (XE, YE) pointed to by the XE and YE registers. WEP determines whether the end point (XE, YE) is drawn. The drawing pointer (X#, Y#) changes to (XE, YE). The commands differ as follows.
	<b>A_LINE_M0</b>	6FH
		6EH
0 0 0 1 0 1 0 0 0 ED IP ES PXEN BPPX ESH WEP		

(PL)

**Table 4. DRAW Command Descriptions (cont)**

Commands	Name	Description																																													
Graphics Drawing Commands (cont)	A_LINE_M1	The X, Y, XE, YE, XS, and YS registers do not change value.																																													
	<table border="1" style="width: 100%; text-align: center;"> <tr> <td colspan="8">6FH</td> <td colspan="8">6EH</td> </tr> <tr> <td>0</td><td>0</td><td>0</td><td>1</td><td>1</td><td>0</td><td>0</td><td>0</td> <td>ED</td><td>IP</td><td>ES</td><td>PXEN</td><td>BPPX</td><td>ESH</td><td>WEP</td> </tr> <tr> <td colspan="14" style="text-align: right;">(PL)</td> </tr> </table>		6FH								6EH								0	0	0	1	1	0	0	0	ED	IP	ES	PXEN	BPPX	ESH	WEP	(PL)													
6FH								6EH																																							
0	0	0	1	1	0	0	0	ED	IP	ES	PXEN	BPPX	ESH	WEP																																	
(PL)																																															
	A_LINE_M2	The XS and YS registers change to the values in the X and Y registers. The X and Y registers change to the values in the XE and YE registers. The XE and YE registers do not change value.																																													
	<table border="1" style="width: 100%; text-align: center;"> <tr> <td colspan="8">6FH</td> <td colspan="8">6EH</td> </tr> <tr> <td>0</td><td>0</td><td>0</td><td>1</td><td>1</td><td>1</td><td>0</td><td>0</td> <td>ED</td><td>IP</td><td>ES</td><td>PXEN</td><td>BPPX</td><td>ESH</td><td>WEP</td> </tr> <tr> <td colspan="14" style="text-align: right;">(PL)</td> </tr> </table>		6FH								6EH								0	0	0	1	1	1	0	0	ED	IP	ES	PXEN	BPPX	ESH	WEP	(PL)													
6FH								6EH																																							
0	0	0	1	1	1	0	0	ED	IP	ES	PXEN	BPPX	ESH	WEP																																	
(PL)																																															
	Absolute Line Direct 0, 1, 2, 3	A straight line is drawn from the current drawing pointer (X#, Y#) to the coordinates (XE, YE) pointed to by the XE and YE registers, respectively. The values in the X and Y registers should be equal to the drawing pointer (X#, Y#) in order to execute these commands. The drawing of the end point (XE, YE) is determined by WEP. The commands differ as follows.																																													
	A_LINE_D0	The drawing pointer (X#, Y#) and X and Y register values change to XE and YE. The values in the XE, YE, XS, and YS registers do not change.																																													
	<table border="1" style="width: 100%; text-align: center;"> <tr> <td colspan="8">6FH</td> <td colspan="8">6EH</td> </tr> <tr> <td>0</td><td>0</td><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td> <td>ED</td><td>IP</td><td>ES</td><td>PXEN</td><td>BPPX</td><td>ESH</td><td>WEP</td> </tr> <tr> <td colspan="14" style="text-align: right;">(PL)</td> </tr> </table>		6FH								6EH								0	0	1	0	0	0	0	0	ED	IP	ES	PXEN	BPPX	ESH	WEP	(PL)													
6FH								6EH																																							
0	0	1	0	0	0	0	0	ED	IP	ES	PXEN	BPPX	ESH	WEP																																	
(PL)																																															
	A_LINE_D1	The values in the X, Y, XE, YE, XS, and YS registers do not change. The drawing pointer (X#, Y#) changes to (XE, YE).																																													
	<table border="1" style="width: 100%; text-align: center;"> <tr> <td colspan="8">6FH</td> <td colspan="8">6EH</td> </tr> <tr> <td>0</td><td>0</td><td>1</td><td>0</td><td>0</td><td>1</td><td>0</td><td>0</td> <td>ED</td><td>IP</td><td>ES</td><td>PXEN</td><td>BPPX</td><td>ESH</td><td>WEP</td> </tr> <tr> <td colspan="14" style="text-align: right;">(PL)</td> </tr> </table>		6FH								6EH								0	0	1	0	0	1	0	0	ED	IP	ES	PXEN	BPPX	ESH	WEP	(PL)													
6FH								6EH																																							
0	0	1	0	0	1	0	0	ED	IP	ES	PXEN	BPPX	ESH	WEP																																	
(PL)																																															
	A_LINE_D2	The values in the XS and YS registers change to those in the X and Y registers. The X and Y register values change to those in the XE and YE registers. The XE and YE register values do not change. The drawing pointer (X#, Y#) changes to (XE, YE).																																													
	<table border="1" style="width: 100%; text-align: center;"> <tr> <td colspan="8">6FH</td> <td colspan="8">6EH</td> </tr> <tr> <td>0</td><td>0</td><td>1</td><td>0</td><td>1</td><td>0</td><td>0</td><td>0</td> <td>ED</td><td>IP</td><td>ES</td><td>PXEN</td><td>BPPX</td><td>ESH</td><td>WEP</td> </tr> <tr> <td colspan="14" style="text-align: right;">(PL)</td> </tr> </table>		6FH								6EH								0	0	1	0	1	0	0	0	ED	IP	ES	PXEN	BPPX	ESH	WEP	(PL)													
6FH								6EH																																							
0	0	1	0	1	0	0	0	ED	IP	ES	PXEN	BPPX	ESH	WEP																																	
(PL)																																															

**Table 4. DRAW Command Descriptions (cont)**

Commands	Name	Description																															
Graphics Drawing Commands (cont)	A_LINE_D3	The values in the XS and YS registers are used for the end point of the line. The drawing pointer changes to (XS, YS). The values in the X and Y registers change to those in the XS and YS registers. The XE, YE, XS, and YS register values do not change.																															
<table border="1" style="margin-left: auto; margin-right: auto;"> <tr> <td colspan="8" style="text-align: center;">6FH</td> <td colspan="8" style="text-align: center;">6EH</td> </tr> <tr> <td>0</td><td>0</td><td>1</td><td>0</td><td>1</td><td>1</td><td>0</td><td>0</td> <td>ED</td><td>IP</td><td>ES</td><td>PXEN</td> <td>BPPX</td><td>ESH</td><td>WEP</td> </tr> </table>		6FH								6EH								0	0	1	0	1	1	0	0	ED	IP	ES	PXEN	BPPX	ESH	WEP	(PL)
6FH								6EH																									
0	0	1	0	1	1	0	0	ED	IP	ES	PXEN	BPPX	ESH	WEP																			
	Relative Line with Move 0, 1, 2	A straight line is drawn from coordinates (X, Y) pointed to by the X and Y registers to the point (X+DX, Y+DY) with DX and DY contained in their respective registers. Drawing of the end point is determined by the WEP bit.																															
	R_LINE_M0	The drawing pointer (X#, Y#) changes to (X+DX, Y+DY). The X and Y registers change to (X+DX, Y+DY). The DX, DY, XS, and YS register values do not change.																															
<table border="1" style="margin-left: auto; margin-right: auto;"> <tr> <td colspan="8" style="text-align: center;">6FH</td> <td colspan="8" style="text-align: center;">6EH</td> </tr> <tr> <td>0</td><td>0</td><td>1</td><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td> <td>ED</td><td>IP</td><td>ES</td><td>PXEN</td> <td>BPPX</td><td>ESH</td><td>WEP</td> </tr> </table>		6FH								6EH								0	0	1	1	0	0	0	0	ED	IP	ES	PXEN	BPPX	ESH	WEP	(PL)
6FH								6EH																									
0	0	1	1	0	0	0	0	ED	IP	ES	PXEN	BPPX	ESH	WEP																			
	R_LINE_M1	The drawing pointer (X#, Y#) changes to (X+DX, Y+DY). The X, Y, DX, DY, XS, and YS register values do not change.																															
<table border="1" style="margin-left: auto; margin-right: auto;"> <tr> <td colspan="8" style="text-align: center;">6FH</td> <td colspan="8" style="text-align: center;">6EH</td> </tr> <tr> <td>0</td><td>0</td><td>1</td><td>1</td><td>0</td><td>1</td><td>0</td><td>0</td> <td>ED</td><td>IP</td><td>ES</td><td>PXEN</td> <td>BPPX</td><td>ESH</td><td>WEP</td> </tr> </table>		6FH								6EH								0	0	1	1	0	1	0	0	ED	IP	ES	PXEN	BPPX	ESH	WEP	(PL)
6FH								6EH																									
0	0	1	1	0	1	0	0	ED	IP	ES	PXEN	BPPX	ESH	WEP																			
	R_LINE_M2	The drawing pointer (X#, Y#) changes to (X+DX, Y+DY). The XS, and YS registers change to (X, Y). The X and Y registers change to (X+DX, Y+DY). The DX and DY register values do not change.																															
<table border="1" style="margin-left: auto; margin-right: auto;"> <tr> <td colspan="8" style="text-align: center;">6FH</td> <td colspan="8" style="text-align: center;">6EH</td> </tr> <tr> <td>0</td><td>0</td><td>1</td><td>1</td><td>1</td><td>0</td><td>0</td><td>0</td> <td>ED</td><td>IP</td><td>ES</td><td>PXEN</td> <td>BPPX</td><td>ESH</td><td>WEP</td> </tr> </table>		6FH								6EH								0	0	1	1	1	0	0	0	ED	IP	ES	PXEN	BPPX	ESH	WEP	(PL)
6FH								6EH																									
0	0	1	1	1	0	0	0	ED	IP	ES	PXEN	BPPX	ESH	WEP																			
	R Line Direct 0, 1, 2	A straight line is drawn from the drawing pointer (X#, Y#) to the coordinates (X+DX, Y+DY) pointed to by the DX and DY registers. The drawing of the end point is determined by the WEP bit. The drawing pointer changes to (X+DX, Y+DY).																															
	R_LINE_D0	The X and Y registers change to (X+DX, Y+DY). The DX, DY, XS, and YS register values do not change.																															
<table border="1" style="margin-left: auto; margin-right: auto;"> <tr> <td colspan="8" style="text-align: center;">6FH</td> <td colspan="8" style="text-align: center;">6EH</td> </tr> <tr> <td>0</td><td>0</td><td>1</td><td>1</td><td>1</td><td>1</td><td>0</td><td>0</td> <td>ED</td><td>IP</td><td>ES</td><td>PXEN</td> <td>BPPX</td><td>ESH</td><td>WEP</td> </tr> </table>		6FH								6EH								0	0	1	1	1	1	0	0	ED	IP	ES	PXEN	BPPX	ESH	WEP	(PL)
6FH								6EH																									
0	0	1	1	1	1	0	0	ED	IP	ES	PXEN	BPPX	ESH	WEP																			

**Table 4. Drawing Command Descriptions (cont)**

Commands	Name	Description																																													
Graphics Drawing Commands (cont)	R_LINE_D1	The X Y, DX, DY, XS, and YS register values do not change .																																													
<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td colspan="8" style="text-align: center;">6FH</td> <td colspan="7" style="text-align: center;">6EH</td> </tr> <tr> <td style="text-align: center;">0</td><td style="text-align: center;">1</td><td style="text-align: center;">0</td><td style="text-align: center;">0</td><td style="text-align: center;">0</td><td style="text-align: center;">0</td><td style="text-align: center;">0</td><td style="text-align: center;">0</td><td style="text-align: center;">ED</td><td style="text-align: center;">IP</td><td style="text-align: center;">ES</td><td style="text-align: center;">PXEN</td><td style="text-align: center;">BPPX</td><td style="text-align: center;">ESH</td><td style="text-align: center;">WEP</td> </tr> <tr> <td colspan="15" style="text-align: right;">(PL)</td> </tr> </table>		6FH								6EH							0	1	0	0	0	0	0	0	ED	IP	ES	PXEN	BPPX	ESH	WEP	(PL)															
6FH								6EH																																							
0	1	0	0	0	0	0	0	ED	IP	ES	PXEN	BPPX	ESH	WEP																																	
(PL)																																															
	R_LINE_D2	The XS and YS registers change to (X, Y). The X and Y registers change to (X+DX, Y+DY). The DX and DY register values do not change.																																													
<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td colspan="8" style="text-align: center;">6FH</td> <td colspan="7" style="text-align: center;">6EH</td> </tr> <tr> <td style="text-align: center;">0</td><td style="text-align: center;">1</td><td style="text-align: center;">0</td><td style="text-align: center;">0</td><td style="text-align: center;">0</td><td style="text-align: center;">1</td><td style="text-align: center;">0</td><td style="text-align: center;">0</td><td style="text-align: center;">ED</td><td style="text-align: center;">IP</td><td style="text-align: center;">ES</td><td style="text-align: center;">PXEN</td><td style="text-align: center;">BPPX</td><td style="text-align: center;">ESH</td><td style="text-align: center;">WEP</td> </tr> <tr> <td colspan="15" style="text-align: right;">(PL)</td> </tr> </table>		6FH								6EH							0	1	0	0	0	1	0	0	ED	IP	ES	PXEN	BPPX	ESH	WEP	(PL)															
6FH								6EH																																							
0	1	0	0	0	1	0	0	ED	IP	ES	PXEN	BPPX	ESH	WEP																																	
(PL)																																															
	A_REC Absolute Rectangle	A rectangle with horizontal and vertical sides parallel to the X and Y axes is drawn with the diagonal vertices at coordinates (X, Y) and (XS, YS) pointed to by the X, Y, XS, and YS registers, respectively. The drawing pointer (X#, Y#) changes to (X, Y).																																													
<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td colspan="8" style="text-align: center;">6FH</td> <td colspan="7" style="text-align: center;">6EH</td> </tr> <tr> <td style="text-align: center;">0</td><td style="text-align: center;">1</td><td style="text-align: center;">0</td><td style="text-align: center;">0</td><td style="text-align: center;">1</td><td style="text-align: center;">0</td><td style="text-align: center;">0</td><td style="text-align: center;">0</td><td style="text-align: center;">0</td><td style="text-align: center;">IP</td><td style="text-align: center;">ES</td><td style="text-align: center;">PXEN</td><td style="text-align: center;">BPPX</td><td style="text-align: center;">ESH</td><td style="text-align: center;">0</td> </tr> </table>		6FH								6EH							0	1	0	0	1	0	0	0	0	IP	ES	PXEN	BPPX	ESH	0																
6FH								6EH																																							
0	1	0	0	1	0	0	0	0	IP	ES	PXEN	BPPX	ESH	0																																	
	R_REC Relative Rectangle	A rectangle with horizontal and vertical sides parallel to the X and Y axes is drawn with the diagonal vertices at coordinates (X, Y) and (X+DX, Y+DY). The drawing pointer (X#, Y#) changes to (X, Y).																																													
<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td colspan="8" style="text-align: center;">6FH</td> <td colspan="7" style="text-align: center;">6EH</td> </tr> <tr> <td style="text-align: center;">0</td><td style="text-align: center;">1</td><td style="text-align: center;">0</td><td style="text-align: center;">0</td><td style="text-align: center;">1</td><td style="text-align: center;">1</td><td style="text-align: center;">0</td><td style="text-align: center;">0</td><td style="text-align: center;">0</td><td style="text-align: center;">IP</td><td style="text-align: center;">ES</td><td style="text-align: center;">PXEN</td><td style="text-align: center;">BPPX</td><td style="text-align: center;">ESH</td><td style="text-align: center;">0</td> </tr> </table>		6FH								6EH							0	1	0	0	1	1	0	0	0	IP	ES	PXEN	BPPX	ESH	0																
6FH								6EH																																							
0	1	0	0	1	1	0	0	0	IP	ES	PXEN	BPPX	ESH	0																																	
	CRL Circle	A circle is drawn counterclockwise with the center at (XC, YC) pointed to by the XC and YC registers and with radius DX defined by the DX register. The drawing pointer (X#, Y#) changes to (XC, YC+DX). The circle is started from (XC, YC+DX). DX must be > 0.																																													
<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td colspan="8" style="text-align: center;">6FH</td> <td colspan="7" style="text-align: center;">6EH</td> </tr> <tr> <td style="text-align: center;">0</td><td style="text-align: center;">1</td><td style="text-align: center;">0</td><td style="text-align: center;">1</td><td style="text-align: center;">0</td><td style="text-align: center;">0</td><td style="text-align: center;">0</td><td style="text-align: center;">0</td><td style="text-align: center;">0</td><td style="text-align: center;">IP</td><td style="text-align: center;">0</td><td style="text-align: center;">PXEN</td><td style="text-align: center;">BPPX</td><td style="text-align: center;">0</td><td style="text-align: center;">0</td> </tr> </table>		6FH								6EH							0	1	0	1	0	0	0	0	0	IP	0	PXEN	BPPX	0	0																
6FH								6EH																																							
0	1	0	1	0	0	0	0	0	IP	0	PXEN	BPPX	0	0																																	
	CARC Circle Arc	A circular arc is drawn from coordinates (XS, YS) to (XE, YE) with the center of the circle at (XC, YC) and radius DX. These are pointed to by the XS, YS, XE, YE, XC, YC, and DX registers, respectively. The drawing pointer changes to (XE, YE). DX must be > 0.																																													
<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td colspan="8" style="text-align: center;">6FH</td> <td colspan="7" style="text-align: center;">6EH</td> </tr> <tr> <td style="text-align: center;">0</td><td style="text-align: center;">1</td><td style="text-align: center;">0</td><td style="text-align: center;">1</td><td style="text-align: center;">0</td><td style="text-align: center;">1</td><td style="text-align: center;">0</td><td style="text-align: center;">0</td><td style="text-align: center;">CF</td><td style="text-align: center;">IP</td><td style="text-align: center;">0</td><td style="text-align: center;">PXEN</td><td style="text-align: center;">BPPX</td><td style="text-align: center;">0</td><td style="text-align: center;">WEP</td> </tr> </table>		6FH								6EH							0	1	0	1	0	1	0	0	CF	IP	0	PXEN	BPPX	0	WEP																
6FH								6EH																																							
0	1	0	1	0	1	0	0	CF	IP	0	PXEN	BPPX	0	WEP																																	
	CSEC Circle Sector	A circular sector is drawn with the center at (XC, YC), DX the radius, (XS, YS) the starting point, and (XE, YE) the ending point. The drawing pointer changes to (XS, YS). DX must be > 0.																																													
<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td colspan="8" style="text-align: center;">6FH</td> <td colspan="7" style="text-align: center;">6EH</td> </tr> <tr> <td style="text-align: center;">0</td><td style="text-align: center;">1</td><td style="text-align: center;">0</td><td style="text-align: center;">1</td><td style="text-align: center;">1</td><td style="text-align: center;">0</td><td style="text-align: center;">0</td><td style="text-align: center;">0</td><td style="text-align: center;">CF</td><td style="text-align: center;">IP</td><td style="text-align: center;">0</td><td style="text-align: center;">PXEN</td><td style="text-align: center;">BPPX</td><td style="text-align: center;">0</td><td style="text-align: center;">0</td> </tr> </table>		6FH								6EH							0	1	0	1	1	0	0	0	CF	IP	0	PXEN	BPPX	0	0																
6FH								6EH																																							
0	1	0	1	1	0	0	0	CF	IP	0	PXEN	BPPX	0	0																																	



**Table 4. DRAW Command Descriptions (cont)**

Commands	Name	Description
Fill and Paint Commands	PAINT	A boundary-point search is carried out starting from coordinates (X, Y) and the resulting enclosed area is filled with a solid or tiling pattern. When PMOD = 0, the boundary colors are set into the DX register. The area to be painted must be enclosed within the clipping rectangle and the CLIP register must be set to 00.
	6FH	6EH
	0 1 1 0 1 0 0 0	TL 0 1 SS 0 PMOD 0 0
A_TRI_FILL Absolute Triangle Fill	A triangular region with vertices at (X, Y), (XS, YS), and (XC, YC) is filled with the tiling pattern. Y, YS, and YC must not be equal to each other.	
	6FH	6EH
	0 1 1 0 1 1 0 0	TL 0 1 SS WL WR 0 0
A_TRA_FILL Absolute Trapezoid Fill	A trapezoidal area with its parallel sides (upper and lower) defined by line segments connecting (X, Y) to (XS, Y) and (YS, YE) to (XE, YE), where YS is an X-axis value, is filled with the tiling pattern.	
	6FH	6EH
	0 1 1 1 0 0 0 0	TL 0 1 SS WL WR 0 0
R_TRA_FILL Relative Trapezoid Fill	A trapezoidal area with its upper parallel side defined by the line segment connecting (X, Y) to (XS, Y), a height of DV + 1 dots above the lower side line segment connecting X + DX and XS + XC, is filled with the tiling pattern.	
	6FH	6EH
	0 1 1 1 0 1 0 0	TL 0 1 SS WL WR 0 0
A_REC_FILL_C Absolute Rectangle Fill by Coordinates	A rectangle with vertical and horizontal sides parallel to the coordinate axes is filled with the tiling pattern. The diagonal vertices of the rectangle are (X, Y) and (XS, YS).	
	6FH	6EH
	1 0 0 0 1 1 0 0	TL 0 1 SS WL WR FAST 0
A_REC_FILL_A Absolute Rectangle Fill by Address	A rectangle with vertical and horizontal sides parallel to the coordinate axes is filled with the tiling pattern. The rectangle is defined by the number of dots in the horizontal direction DH + 1, the number of dots in the vertical direction DV + 1, the starting address (physical address) EAD1, and the bit position in the starting address dAD1.	
	6FH	6EH
	1 0 0 0 1 1 1 0	0 0 1 1 1 1 0

**Table 4. DRAW Command Descriptions (cont)**

Commands	Name	Description
Fill and Paint Commands (cont)	R_REC_FILL Relative Rectangle Fill by Coordinates	A rectangle with vertical and horizontal sides parallel to the coordinate axes is filled with the tiling pattern. The rectangle is defined by the starting point (X, Y), the horizontal width DX, and the vertical height DY. The diagonal vertices are at (X, Y) and (X+DX, Y+DY).
	6FH	6EH
	1 0 0 1 0 0 0 0	TL 0 1 SS WL WR FAST 0
	CRL_FILL Circle Fill	A circle with its center at (XC, YC) and a radius of DX is filled with the tiling pattern. Points on the circumference are filled. The filling starts from the top of the circle and proceeds downward.
	6FH	6EH
	0 1 0 1 0 0 0 0	TL 0 1 SS 1 1 0 0
	ELPS_FILL Ellipse Fill	An ellipse with its major and minor axes parallel to the coordinate axes, center at (XC, YC), Y-direction radius DY, and ratio of the squares of the X- and Y-direction radii $DX^2/DY^2 = DH/DV$ is filled with the tiling pattern. The filling starts from the top of the ellipse and proceeds downward.
	6FH	6EH
	0 1 0 1 1 1 0 0	TL 0 1 SS 1 1 0 0
Copy Commands	A_COPY_AA Absolute Copy Address to Address	A rectangular area of memory starting from physical location EAD2 and bit position dAD2, with horizontal size DH + 1 dots and vertical size DV + 1 dots, is transferred to the rectangular area of memory starting from EAD1 and bit position dAD1.
	6FH	6EH
	0 1 1 1 1 0 0 0	ESE REV ROT 0 SD_SEL FAST 0
	A_COPY_CA Absolute Copy Coordinate to Address	A rectangular area of display memory starting from (XS, YS), with horizontal size DH + 1 dots and vertical size DV + 1 dots, is transferred to the rectangular area of memory starting from physical address EAD1 and bit position dAD1.
	6FH	6EH
	0 1 1 1 1 0 0 0	ESE REV ROT 0 SD_SEL FAST 0
	A_COPY_AC Absolute Copy Address to Coordinate	A rectangular area of display memory starting from physical address EAD2 and bit position dAD2, with horizontal size DH + 1 dots and vertical size DV + 1 dots, is transferred to the rectangular area of memory starting from (X, Y).
	6FH	6EH
	1 0 0 0 0 0 0 0	ESE REV ROT 0 SD_SEL FAST 0

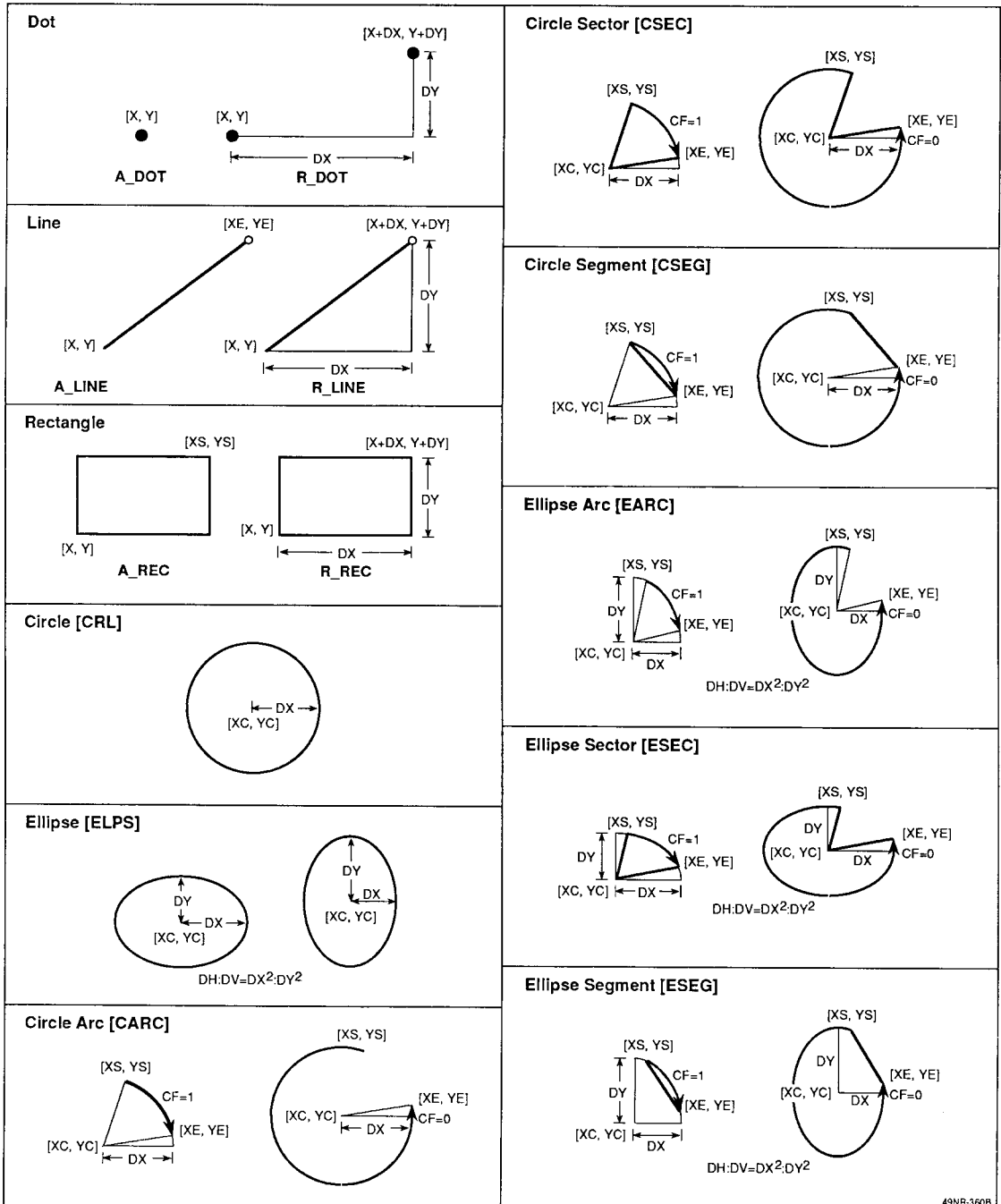
**Table 4. DRAW Command Descriptions (cont)**

Commands	Name	Description														
Copy Commands (cont)	A_COPY_CC Absolute Copy Coordinate to Coordinate	A rectangular area of display memory starting from (XS, YS), with horizontal size DH + 1 dots and vertical size DV + 1 dots, is transferred to the rectangular area of memory starting at (X, Y).														
	<div style="display: flex; justify-content: space-around;"> <span>6FH</span> <span>6EH</span> </div> <table border="1" style="margin: auto;"> <tr> <td>1</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>0</td><td>0</td><td>ESE</td><td>REV</td><td>ROT</td><td>0</td><td>SD_SEL</td><td>FAST</td><td>0</td> </tr> </table>		1	0	0	0	0	1	0	0	ESE	REV	ROT	0	SD_SEL	FAST
1	0	0	0	0	1	0	0	ESE	REV	ROT	0	SD_SEL	FAST	0		
Copy Function Extensions		The function of each COPY command can be extended by changing the lower 2 bits of the command code. This extension is defined in the lower byte (6EH) of the command register.														
	90°_COPY 90° Rotation Copy	The transferred memory area is rotated 90° counterclockwise.														
	<div style="display: flex; justify-content: center;"> <span>6EH</span> </div> <table border="1" style="margin: auto;"> <tr> <td>ESE</td><td>REV</td><td>ROT</td><td>1</td><td>SD_SEL</td><td>0</td><td>0</td> </tr> </table>		ESE	REV	ROT	1	SD_SEL	0	0							
ESE	REV	ROT	1	SD_SEL	0	0										
	SL_COPY Slant Copy	The data in a rectangular area of display memory is slanted by DX in the X-direction to the change in the Y-direction														
	<div style="display: flex; justify-content: center;"> <span>6EH</span> </div> <table border="1" style="margin: auto;"> <tr> <td>ESE</td><td>REV</td><td>ROT</td><td>0</td><td>SD_SEL</td><td>0</td><td>1</td> </tr> </table>		ESE	REV	ROT	0	SD_SEL	0	1							
ESE	REV	ROT	0	SD_SEL	0	1										
	FR_ES_COPY Free Angle Rotation, Enlarge/Shrink Copy	The rectangular data from the source area is transferred to a parallelogram at the destination area in display memory. DY and DX determine the angle for the horizontal side, XE and YE for the vertical side. MAGH and MAGV determine the horizontal and vertical enlargement or shrink factors.														
	<div style="display: flex; justify-content: center;"> <span>6EH</span> </div> <table border="1" style="margin: auto;"> <tr> <td>ESH</td><td>ESV</td><td>FS</td><td>1</td><td>SD_SEL</td><td>1</td><td>0</td> </tr> </table>		ESH	ESV	FS	1	SD_SEL	1	0							
ESH	ESV	FS	1	SD_SEL	1	0										
	ES_COPY Enlarge/Shrink Copy	The rectangular data from the source area is transferred to a rectangular area at the destination in display memory and enlarged or shrunk in the horizontal and/or vertical direction. MAGH and MAGV determine the horizontal and vertical scale factors.														
	<div style="display: flex; justify-content: center;"> <span>6EH</span> </div> <table border="1" style="margin: auto;"> <tr> <td>ESH</td><td>REV</td><td>ROT</td><td>ESV</td><td>SD_SEL</td><td>1</td><td>1</td> </tr> </table>		ESH	REV	ROT	ESV	SD_SEL	1	1							
ESH	REV	ROT	ESV	SD_SEL	1	1										

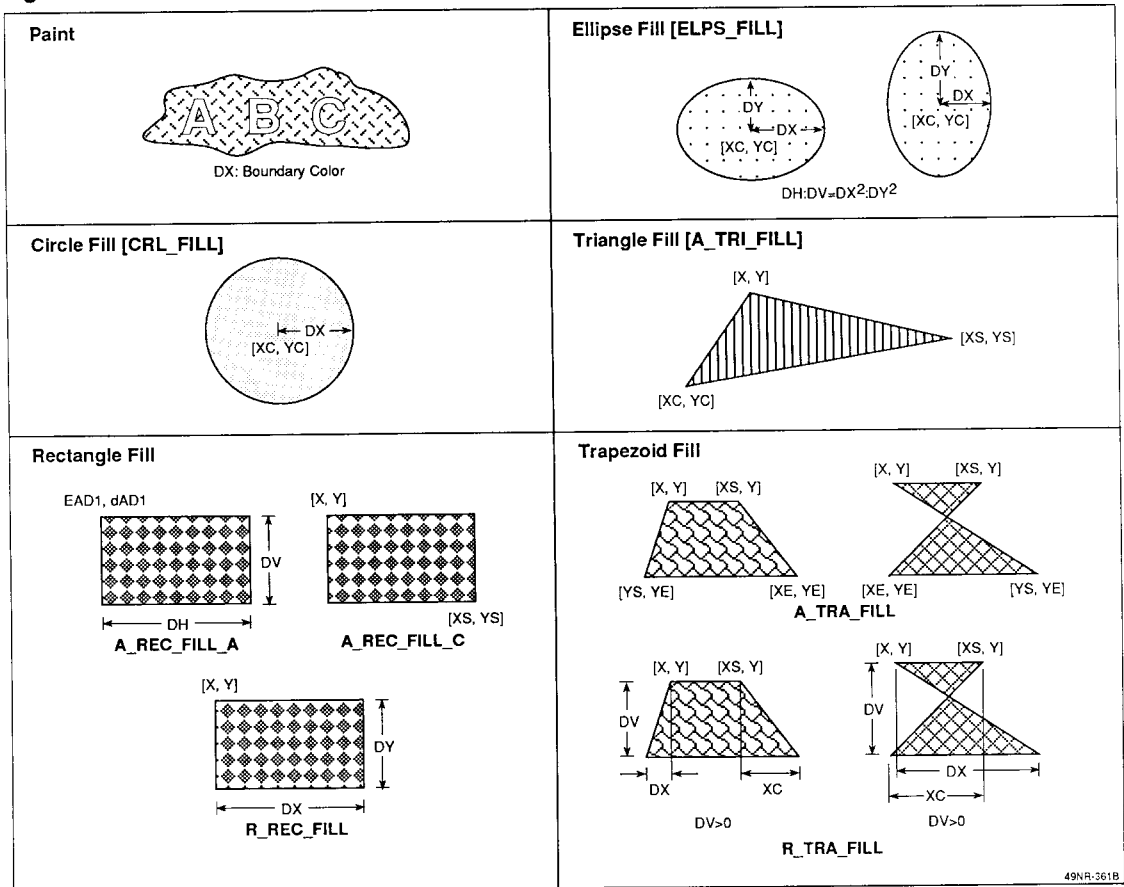
**Table 4. DRAW Command Descriptions (cont)**

Commands	Name	Description														
PUT/GET Commands	PUT_A Put Data to Address Field	Transfers data from the PGPORT register to a rectangular area of display memory starting from word address EAD1 and bit position dAD1 with horizontal width DH+1 dots and vertical height DV+1 dots.														
	<div style="display: flex; justify-content: space-between;"> <span>6FH</span> <span>6EH</span> </div> <table border="1" style="margin: auto;"> <tr> <td>1</td><td>0</td><td>0</td><td>1</td><td>0</td><td>1</td><td>0</td><td>0</td><td>0</td><td>REV</td><td>ROT</td><td>0</td><td>SD_SEL</td><td>1</td><td>1</td> </tr> </table>		1	0	0	1	0	1	0	0	0	REV	ROT	0	SD_SEL	1
1	0	0	1	0	1	0	0	0	REV	ROT	0	SD_SEL	1	1		
	PUT_C Put Data to Coordinate Field	Transfers data from the PGPORT register to a rectangular area of display memory starting from (X, Y) with horizontal width DH+1 dots and vertical height DV+1 dots.														
	<div style="display: flex; justify-content: space-between;"> <span>6FH</span> <span>6EH</span> </div> <table border="1" style="margin: auto;"> <tr> <td>1</td><td>0</td><td>0</td><td>1</td><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td><td>REV</td><td>ROT</td><td>0</td><td>SD_SEL</td><td>1</td><td>1</td> </tr> </table>		1	0	0	1	1	0	0	0	0	REV	ROT	0	SD_SEL	1
1	0	0	1	1	0	0	0	0	REV	ROT	0	SD_SEL	1	1		
	GET_A Get Data from Address Field	Transfers data to the PGPORT register from a rectangular area of display memory starting from word address EAD1 and bit position dAD1 with horizontal width DH+1 dots and vertical height DV+1 dots.														
	<div style="display: flex; justify-content: space-between;"> <span>6FH</span> <span>6EH</span> </div> <table border="1" style="margin: auto;"> <tr> <td>1</td><td>0</td><td>0</td><td>1</td><td>0</td><td>1</td><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>SD_SEL</td><td>1</td><td>0</td> </tr> </table>		1	0	0	1	0	1	1	0	0	0	0	0	SD_SEL	1
1	0	0	1	0	1	1	0	0	0	0	0	SD_SEL	1	0		
	GET_C Get Data from Coordinate Field	Transfers data to the PGPORT register from a rectangular area of display memory starting from (X, Y) with horizontal width DH+1 dots and vertical height DV+1 dots.														
	<div style="display: flex; justify-content: space-between;"> <span>6FH</span> <span>6EH</span> </div> <table border="1" style="margin: auto;"> <tr> <td>1</td><td>0</td><td>0</td><td>1</td><td>1</td><td>0</td><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>SD_SEL</td><td>1</td><td>1</td> </tr> </table>		1	0	0	1	1	0	1	0	0	0	0	0	SD_SEL	1
1	0	0	1	1	0	1	0	0	0	0	0	SD_SEL	1	1		
Get Function Extensions	90°_COPY	Data in the rectangular area of display memory is rotated through 90° and transferred to the PGPORT register.														
<div style="display: flex; justify-content: space-between;"> <span>6EH</span> </div> <table border="1" style="margin: auto;"> <tr> <td>0</td><td>REV</td><td>ROT</td><td>1</td><td>SD_SEL</td><td>1</td><td>0</td> </tr> </table>		0	REV	ROT	1	SD_SEL	1	0								
0	REV	ROT	1	SD_SEL	1	0										

Figure 24. Graphics Drawing Commands



**Figure 25. Fill and Paint Commands**



49NR-361B

**Figure 26. Copy Commands; Copy, Rotate, Slant**

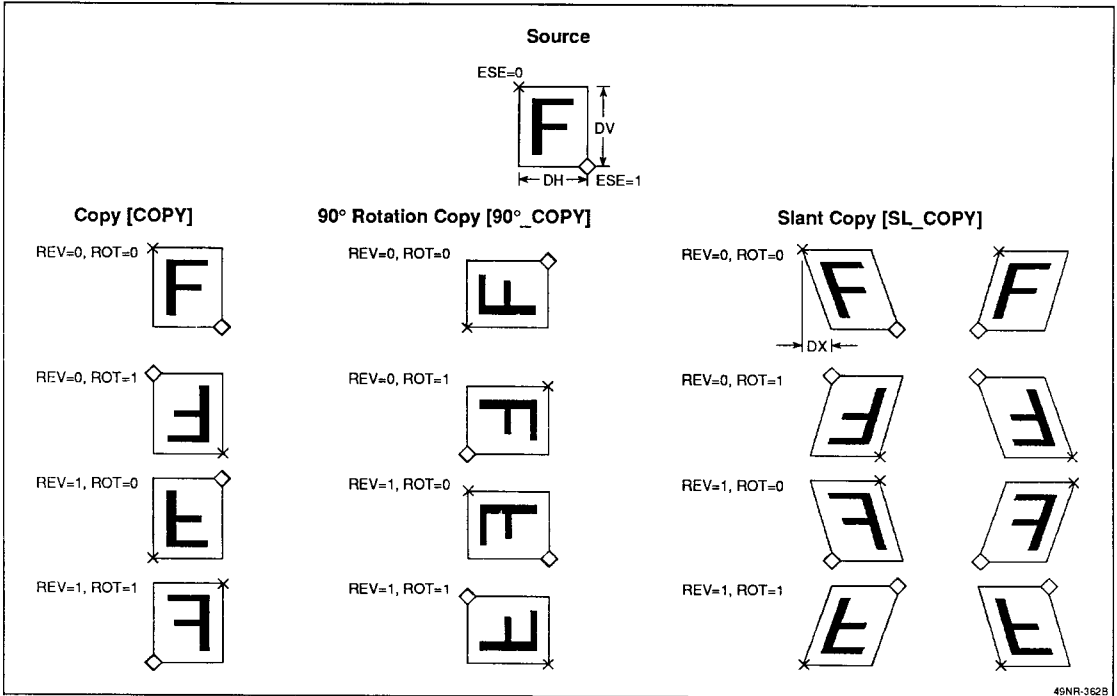
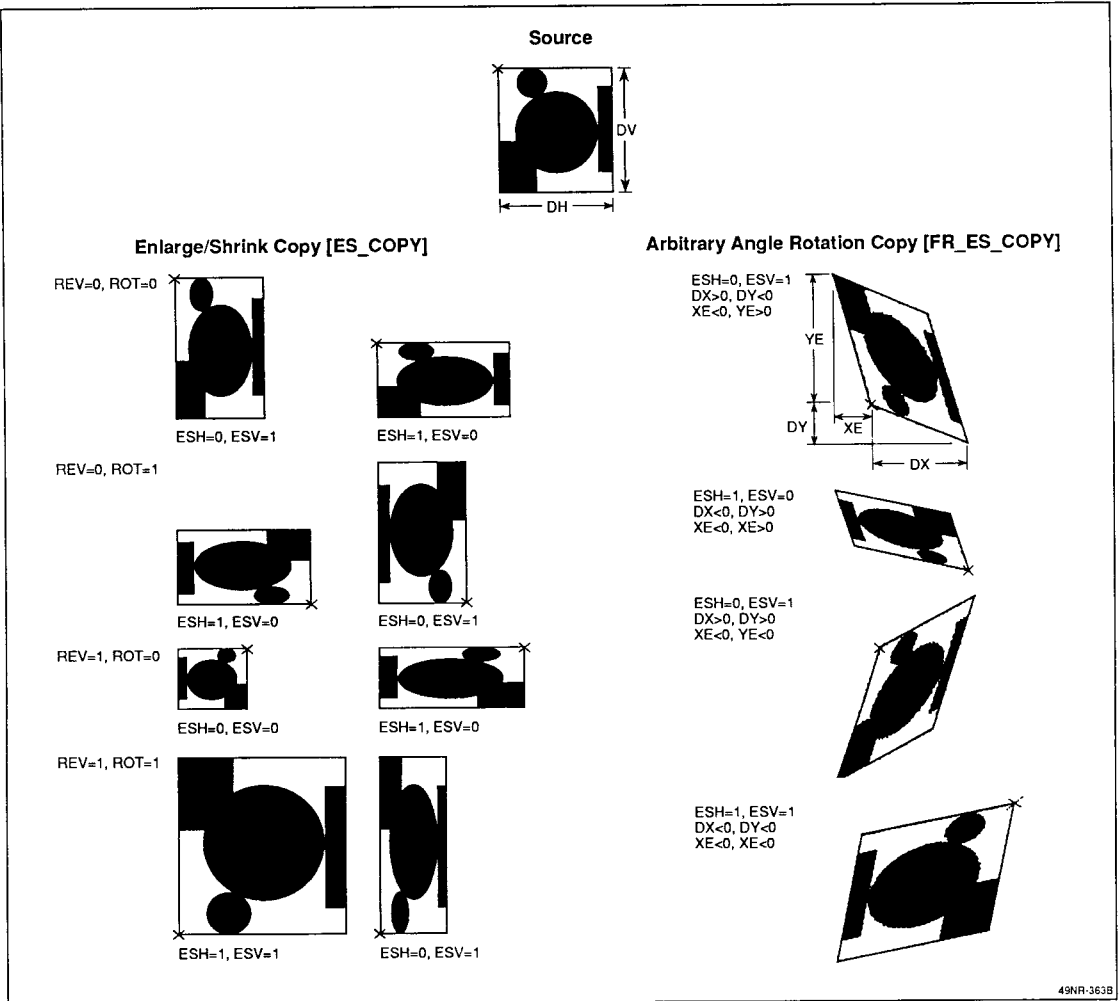


Figure 27. Copy Commands; Enlarge/Shrink, Rotate





**Table 5. DRAW Command Summary (cont)**

Command	Opcode (Hex)	Parameters																								
CRL_FILL	50	XC, YC, DX																								
ELPS_FILL	5C	XC, YC, DY, DH, DV																								
<table border="1" style="width: 100%; text-align: center;"> <tr> <td colspan="4">B7</td> <td colspan="4">Operation Flags</td> <td colspan="4">B0</td> </tr> <tr> <td>TL</td><td>0</td><td>1</td><td>SS</td><td>1</td><td>1</td><td>0</td><td>0</td><td colspan="4">6EH</td> </tr> </table>			B7				Operation Flags				B0				TL	0	1	SS	1	1	0	0	6EH			
B7				Operation Flags				B0																		
TL	0	1	SS	1	1	0	0	6EH																		
A_COPY_AA	78	EAD1, dAD1, EAD2, dAD2, DH, DV																								
_CA	7C	XS, YS, EAD1, dAD1, DH, DV																								
_AC	80	EAD2, dAD2, DH, DV, X, Y																								
_CC	84	XS, YS, X, Y, DH, DV																								
<table border="1" style="width: 100%; text-align: center;"> <tr> <td colspan="4">B7</td> <td colspan="4">Operation Flags</td> <td colspan="4">B0</td> </tr> <tr> <td>ESE</td><td>REV</td><td>ROT</td><td>0</td><td>SD_SEL</td><td>FAST</td><td>0</td><td>0</td><td colspan="4">6EH</td> </tr> </table>			B7				Operation Flags				B0				ESE	REV	ROT	0	SD_SEL	FAST	0	0	6EH			
B7				Operation Flags				B0																		
ESE	REV	ROT	0	SD_SEL	FAST	0	0	6EH																		
A_90°_COPY_AA	78	EAD1, dAD1, EAD2, dAD2, DH, DV																								
_CA	7C	XS, YS, EAD2, dAD2, DH, DV																								
_AC	80	EAD2, dAD2, X, Y, DH, DV																								
_CC	84	XS, YS, X, Y, DH, DV																								
<table border="1" style="width: 100%; text-align: center;"> <tr> <td colspan="4">B7</td> <td colspan="4">Operation Flags</td> <td colspan="4">B0</td> </tr> <tr> <td>ESE</td><td>REV</td><td>ROT</td><td>1</td><td>SD_SEL</td><td>0</td><td>0</td><td>0</td><td colspan="4">6EH</td> </tr> </table>			B7				Operation Flags				B0				ESE	REV	ROT	1	SD_SEL	0	0	0	6EH			
B7				Operation Flags				B0																		
ESE	REV	ROT	1	SD_SEL	0	0	0	6EH																		
A_SL_COPY_AA	78	EAD1, dAD1, EAD2, dAD2, DH, DV, DX																								
_CA	7C	XS, YS, EAD1, dAD1, DH, DV, DX																								
_AC	80	EAD2, dAD2, X, Y, DH, DV, DX																								
_CC	84	XS, YS, X, Y, DH, DV, DX																								
<table border="1" style="width: 100%; text-align: center;"> <tr> <td colspan="4">B7</td> <td colspan="4">Operation Flags</td> <td colspan="4">B0</td> </tr> <tr> <td>ESE</td><td>REV</td><td>ROT</td><td>0</td><td>SD_SEL</td><td>0</td><td>1</td><td>1</td><td colspan="4">6EH</td> </tr> </table>			B7				Operation Flags				B0				ESE	REV	ROT	0	SD_SEL	0	1	1	6EH			
B7				Operation Flags				B0																		
ESE	REV	ROT	0	SD_SEL	0	1	1	6EH																		
A_FR_																										
ES_COPY_AA	78	EAD1, dAD1, EAD2, dAD2, DH, DV, DX, DY, XE, YE																								
_CA	7C	XS, YS, EAD1, dAD1, DH, DV, DX, DY, XE, YE																								
_AC	80	EAD2, dAD2, X, Y, DH, DV, DX, DY, XE, YE																								
_CC	84	XS, YS, X, Y, DH, DV, DX, DY, XE, YE																								
<table border="1" style="width: 100%; text-align: center;"> <tr> <td colspan="4">B7</td> <td colspan="4">Operation Flags</td> <td colspan="4">B0</td> </tr> <tr> <td>ESH</td><td>ESV</td><td>FS</td><td>1</td><td>SD_SEL</td><td>1</td><td>0</td><td>0</td><td colspan="4">6EH</td> </tr> </table>			B7				Operation Flags				B0				ESH	ESV	FS	1	SD_SEL	1	0	0	6EH			
B7				Operation Flags				B0																		
ESH	ESV	FS	1	SD_SEL	1	0	0	6EH																		

Command	Opcode (Hex)	Parameters																								
A_ES_COPY_AC	80	EAD2, dAD2, X, Y, DH, DV																								
_CC	84	XS, YS, X, Y, DH, DV																								
<table border="1" style="width: 100%; text-align: center;"> <tr> <td colspan="4">B7</td> <td colspan="4">Operation Flags</td> <td colspan="4">B0</td> </tr> <tr> <td>ESH</td><td>REV</td><td>ROT</td><td>ESV</td><td>SD_SEL</td><td>1</td><td>1</td><td>1</td><td colspan="4">6EH</td> </tr> </table>			B7				Operation Flags				B0				ESH	REV	ROT	ESV	SD_SEL	1	1	1	6EH			
B7				Operation Flags				B0																		
ESH	REV	ROT	ESV	SD_SEL	1	1	1	6EH																		
PUT_A	94	EAD1, dAD1, DH, DV																								
_C	98	X, Y, DH, DV																								
<table border="1" style="width: 100%; text-align: center;"> <tr> <td colspan="4">B7</td> <td colspan="4">Operation Flags</td> <td colspan="4">B0</td> </tr> <tr> <td>0</td><td>REV</td><td>ROT</td><td>0</td><td>SD_SEL</td><td>1</td><td>1</td><td>1</td><td colspan="4">6EH</td> </tr> </table>			B7				Operation Flags				B0				0	REV	ROT	0	SD_SEL	1	1	1	6EH			
B7				Operation Flags				B0																		
0	REV	ROT	0	SD_SEL	1	1	1	6EH																		
GET_A	96	EAD1, dAD1, DH, DV																								
_C	9A	X, Y, DH, DV																								
<table border="1" style="width: 100%; text-align: center;"> <tr> <td colspan="4">B7</td> <td colspan="4">Operation Flags</td> <td colspan="4">B0</td> </tr> <tr> <td>0</td><td>0</td><td>0</td><td>0</td><td>SD_SEL</td><td>1</td><td>0</td><td>0</td><td colspan="4">6EH</td> </tr> </table>			B7				Operation Flags				B0				0	0	0	0	SD_SEL	1	0	0	6EH			
B7				Operation Flags				B0																		
0	0	0	0	SD_SEL	1	0	0	6EH																		
90°_GET_A	96	EAD1, dAD1, DH, DV																								
_C	9A	X, Y, DH, DV																								
<table border="1" style="width: 100%; text-align: center;"> <tr> <td colspan="4">B7</td> <td colspan="4">Operation Flags</td> <td colspan="4">B0</td> </tr> <tr> <td>0</td><td>REV</td><td>ROT</td><td>1</td><td>SD_SEL</td><td>1</td><td>0</td><td>0</td><td colspan="4">6EH</td> </tr> </table>			B7				Operation Flags				B0				0	REV	ROT	1	SD_SEL	1	0	0	6EH			
B7				Operation Flags				B0																		
0	REV	ROT	1	SD_SEL	1	0	0	6EH																		

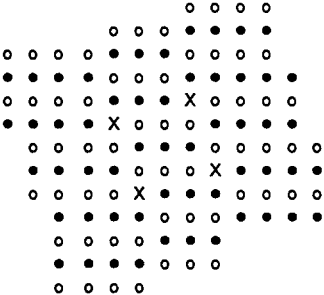
**Table 6. Operation Flag Descriptions**

Name	Description																																																			
PXEN (Pixel Drawing Enable) BPPX (Bits per Pixel)	The plane or packed pixel display memory configuration is selected by PXEN and the number of bits in one pixel is defined by BPPX. The μPD72120 display memory data width is 16 bits. For plane configuration, PXEN = 0.																																																			
	<table border="1"> <thead> <tr> <th>BPPX</th> <th>PXEN</th> <th>Bits/Pixel</th> </tr> </thead> <tbody> <tr> <td>xx</td> <td>0</td> <td>1</td> </tr> <tr> <td>00</td> <td>1</td> <td>2</td> </tr> <tr> <td>01</td> <td>1</td> <td>4</td> </tr> <tr> <td>10</td> <td>1</td> <td>8</td> </tr> <tr> <td>11</td> <td>1</td> <td>16</td> </tr> </tbody> </table>	BPPX	PXEN	Bits/Pixel	xx	0	1	00	1	2	01	1	4	10	1	8	11	1	16																																	
BPPX	PXEN	Bits/Pixel																																																		
xx	0	1																																																		
00	1	2																																																		
01	1	4																																																		
10	1	8																																																		
11	1	16																																																		
ES (Enlarge/Shrink) ESH (Enlarge/Shrink Horizontally) ESV (Enlarge/Shrink Vertically)	Select the enlarge and shrink options.																																																			
	<table border="1"> <thead> <tr> <th>ES</th> <th>ESH</th> <th>ESV</th> <th>Copy Operation</th> <th>Drawing Operation</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>X</td> <td>X</td> <td>No enlarge/shrink</td> <td>No enlarge/shrink</td> </tr> <tr> <td>1</td> <td>0</td> <td>X</td> <td>Horizontal shrink</td> <td>Horizontal pattern shrink</td> </tr> <tr> <td>1</td> <td>1</td> <td>X</td> <td>Horizontal enlarge</td> <td>Horizontal pattern enlarge</td> </tr> <tr> <td>1</td> <td>X</td> <td>0</td> <td>Vertical shrink</td> <td>—</td> </tr> <tr> <td>1</td> <td>X</td> <td>1</td> <td>Vertical enlarge</td> <td>—</td> </tr> </tbody> </table>	ES	ESH	ESV	Copy Operation	Drawing Operation	0	X	X	No enlarge/shrink	No enlarge/shrink	1	0	X	Horizontal shrink	Horizontal pattern shrink	1	1	X	Horizontal enlarge	Horizontal pattern enlarge	1	X	0	Vertical shrink	—	1	X	1	Vertical enlarge	—																					
ES	ESH	ESV	Copy Operation	Drawing Operation																																																
0	X	X	No enlarge/shrink	No enlarge/shrink																																																
1	0	X	Horizontal shrink	Horizontal pattern shrink																																																
1	1	X	Horizontal enlarge	Horizontal pattern enlarge																																																
1	X	0	Vertical shrink	—																																																
1	X	1	Vertical enlarge	—																																																
	Enlargement/Shrinkage factors.																																																			
	<table border="1"> <thead> <tr> <th>MAGH/MAGV</th> <th>ESH/ESV = 0</th> <th>ESH/ESV = 1</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>1/16</td> <td>16/1</td> </tr> <tr> <td>1</td> <td>2/16</td> <td>16/2</td> </tr> <tr> <td>2</td> <td>3/16</td> <td>16/3</td> </tr> <tr> <td>3</td> <td>4/16</td> <td>16/4</td> </tr> <tr> <td>4</td> <td>5/16</td> <td>16/5</td> </tr> <tr> <td>5</td> <td>6/16</td> <td>16/6</td> </tr> <tr> <td>6</td> <td>7/16</td> <td>16/7</td> </tr> <tr> <td>7</td> <td>8/16</td> <td>16/8</td> </tr> <tr> <td>8</td> <td>9/16</td> <td>16/9</td> </tr> <tr> <td>9</td> <td>10/16</td> <td>16/10</td> </tr> <tr> <td>10</td> <td>11/16</td> <td>16/11</td> </tr> <tr> <td>11</td> <td>12/16</td> <td>16/12</td> </tr> <tr> <td>12</td> <td>13/16</td> <td>16/13</td> </tr> <tr> <td>13</td> <td>14/16</td> <td>16/14</td> </tr> <tr> <td>14</td> <td>15/16</td> <td>16/15</td> </tr> <tr> <td>15</td> <td>16/16</td> <td>16/16</td> </tr> </tbody> </table>	MAGH/MAGV	ESH/ESV = 0	ESH/ESV = 1	0	1/16	16/1	1	2/16	16/2	2	3/16	16/3	3	4/16	16/4	4	5/16	16/5	5	6/16	16/6	6	7/16	16/7	7	8/16	16/8	8	9/16	16/9	9	10/16	16/10	10	11/16	16/11	11	12/16	16/12	12	13/16	16/13	13	14/16	16/14	14	15/16	16/15	15	16/16	16/16
MAGH/MAGV	ESH/ESV = 0	ESH/ESV = 1																																																		
0	1/16	16/1																																																		
1	2/16	16/2																																																		
2	3/16	16/3																																																		
3	4/16	16/4																																																		
4	5/16	16/5																																																		
5	6/16	16/6																																																		
6	7/16	16/7																																																		
7	8/16	16/8																																																		
8	9/16	16/9																																																		
9	10/16	16/10																																																		
10	11/16	16/11																																																		
11	12/16	16/12																																																		
12	13/16	16/13																																																		
13	14/16	16/14																																																		
14	15/16	16/15																																																		
15	16/16	16/16																																																		
ED (Enlargement Direction)	Defines the direction of enlargement for line drawing.																																																			
	<table border="1"> <thead> <tr> <th>ED</th> <th>Enlargement Direction</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>To the right of the line in the direction of drawing.</td> </tr> <tr> <td>1</td> <td>To the left of the line in the direction of drawing.</td> </tr> </tbody> </table>	ED	Enlargement Direction	0	To the right of the line in the direction of drawing.	1	To the left of the line in the direction of drawing.																																													
ED	Enlargement Direction																																																			
0	To the right of the line in the direction of drawing.																																																			
1	To the left of the line in the direction of drawing.																																																			
IP (Initialize Pattern Pointer)	Initializes the line pattern pointer to the first bit of the pattern register.																																																			
	<table border="1"> <thead> <tr> <th>IP</th> <th>Function</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>Pointer not initialized</td> </tr> <tr> <td>1</td> <td>Pointer initialized</td> </tr> </tbody> </table>	IP	Function	0	Pointer not initialized	1	Pointer initialized																																													
IP	Function																																																			
0	Pointer not initialized																																																			
1	Pointer initialized																																																			
CF (Clockwise Flag)	Defines the drawing direction for circular and elliptical arcs, sectors, and segments.																																																			
	<table border="1"> <thead> <tr> <th>CF</th> <th>Function</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>Counterclockwise</td> </tr> <tr> <td>1</td> <td>Clockwise</td> </tr> </tbody> </table>	CF	Function	0	Counterclockwise	1	Clockwise																																													
CF	Function																																																			
0	Counterclockwise																																																			
1	Clockwise																																																			

**Table 6. Operation Flag Descriptions (cont)**

Name	Description																	
TL (Tiling Pattern)	Defines the use of a tiling pattern in filling.																	
SS (Single Source Pattern)	<table border="1"> <thead> <tr> <th>TL</th> <th>SS</th> <th>Function</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>Not used</td> </tr> <tr> <td>0</td> <td>1</td> <td>The pattern in the PTNCNT register is used for all planes</td> </tr> <tr> <td>1</td> <td>0</td> <td>The patterns stored in display memory are used for each plane.</td> </tr> <tr> <td>1</td> <td>1</td> <td>The same pattern stored in display memory is used for all planes.</td> </tr> </tbody> </table>	TL	SS	Function	0	0	Not used	0	1	The pattern in the PTNCNT register is used for all planes	1	0	The patterns stored in display memory are used for each plane.	1	1	The same pattern stored in display memory is used for all planes.	<p>To quickly clear all planes to zero, set TL = 0 and SS = 1. When it is necessary to paint with a different color for each bit, set TL = 1 and SS = 0.</p>	
TL	SS	Function																
0	0	Not used																
0	1	The pattern in the PTNCNT register is used for all planes																
1	0	The patterns stored in display memory are used for each plane.																
1	1	The same pattern stored in display memory is used for all planes.																
PMOD (Paint Mode)	<p>Selects the arbitrary boundary area for the PAINT command.</p> <table border="1"> <thead> <tr> <th>PMOD</th> <th>Function</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>Boundary colors are defined by the DX register.</td> </tr> <tr> <td>1</td> <td>Boundaries are all the points with colors different than the starting point (X, Y).</td> </tr> </tbody> </table>			PMOD	Function	0	Boundary colors are defined by the DX register.	1	Boundaries are all the points with colors different than the starting point (X, Y).									
PMOD	Function																	
0	Boundary colors are defined by the DX register.																	
1	Boundaries are all the points with colors different than the starting point (X, Y).																	
WL (Write Left)	Defines whether the boundary points are drawn during a FILL command.																	
WR (Write Right)	<table border="1"> <thead> <tr> <th>WL</th> <th>Function</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>Points on left boundary are not drawn</td> </tr> <tr> <td>1</td> <td>Points on left boundary are drawn</td> </tr> </tbody> </table>	WL	Function	0	Points on left boundary are not drawn	1	Points on left boundary are drawn	<table border="1"> <thead> <tr> <th>WR</th> <th>Function</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>Points on right boundary are not drawn</td> </tr> <tr> <td>1</td> <td>Points on right boundary are drawn</td> </tr> </tbody> </table>	WR	Function	0	Points on right boundary are not drawn	1	Points on right boundary are drawn				
WL	Function																	
0	Points on left boundary are not drawn																	
1	Points on left boundary are drawn																	
WR	Function																	
0	Points on right boundary are not drawn																	
1	Points on right boundary are drawn																	
FAST (Fast)	<p>Specifies the normal or fast mode for drawing.</p> <table border="1"> <thead> <tr> <th>FAST</th> <th>Function</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>Normal speed</td> </tr> <tr> <td>1</td> <td>Fast speed</td> </tr> </tbody> </table> <p>However, FAST mode cannot be used for all drawing operations.</p> <p>REC_FILL The FAST mode cannot be used if clipping or painting with a tiling pattern. It can only be used for replacing data.</p> <p>COPY The FAST mode can be used only for ordinary COPY with replace. It cannot be used with other COPY operation or with multiple sources.</p>			FAST	Function	0	Normal speed	1	Fast speed									
FAST	Function																	
0	Normal speed																	
1	Fast speed																	
ESE (Exchange Start With End)	<p>Defines the reading order of the source data during COPY.</p> <table border="1"> <thead> <tr> <th>ESE</th> <th>Reading Order</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>Upper left to lower right (left to right on each row)</td> </tr> <tr> <td>1</td> <td>Lower right to upper left (right to left on each row)</td> </tr> </tbody> </table>			ESE	Reading Order	0	Upper left to lower right (left to right on each row)	1	Lower right to upper left (right to left on each row)									
ESE	Reading Order																	
0	Upper left to lower right (left to right on each row)																	
1	Lower right to upper left (right to left on each row)																	
REV (Reverse)	<p>Defines the reverse drawing direction during COPY</p> <table border="1"> <thead> <tr> <th>REV</th> <th>Drawing Direction</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>Left to right, top to bottom</td> </tr> <tr> <td>1</td> <td>Right to left, top to bottom</td> </tr> </tbody> </table>			REV	Drawing Direction	0	Left to right, top to bottom	1	Right to left, top to bottom									
REV	Drawing Direction																	
0	Left to right, top to bottom																	
1	Right to left, top to bottom																	
ROT (Rotation)	<p>Defines 180° rotation drawing during COPY.</p> <table border="1"> <thead> <tr> <th>ROT</th> <th>Function</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>Normal</td> </tr> <tr> <td>1</td> <td>180° rotation drawing</td> </tr> </tbody> </table>			ROT	Function	0	Normal	1	180° rotation drawing									
ROT	Function																	
0	Normal																	
1	180° rotation drawing																	
SD_SEL (Source Destination Mode Select)	<p>Selects the transfer mode between planes.</p> <table border="1"> <thead> <tr> <th>SD_SEL</th> <th>Transfer Mode</th> <th>Logical Operation By</th> </tr> </thead> <tbody> <tr> <td>00</td> <td>Multiple sources and single destination</td> <td>MOD1 during read of the sources; MOD0 during write to the destination</td> </tr> <tr> <td>01</td> <td>Multiple sources and single destination</td> <td>MOD0 or MOD1 during read of the sources; REPLACE during write to the destination</td> </tr> <tr> <td>10</td> <td>Single source and multiple destinations</td> <td>MOD0 or MOD1 during write to each of the destinations.</td> </tr> <tr> <td>11</td> <td>Multiple sources and multiple destinations</td> <td>MOD0 or MOD1 during write to each of the destinations.</td> </tr> </tbody> </table>			SD_SEL	Transfer Mode	Logical Operation By	00	Multiple sources and single destination	MOD1 during read of the sources; MOD0 during write to the destination	01	Multiple sources and single destination	MOD0 or MOD1 during read of the sources; REPLACE during write to the destination	10	Single source and multiple destinations	MOD0 or MOD1 during write to each of the destinations.	11	Multiple sources and multiple destinations	MOD0 or MOD1 during write to each of the destinations.
SD_SEL	Transfer Mode	Logical Operation By																
00	Multiple sources and single destination	MOD1 during read of the sources; MOD0 during write to the destination																
01	Multiple sources and single destination	MOD0 or MOD1 during read of the sources; REPLACE during write to the destination																
10	Single source and multiple destinations	MOD0 or MOD1 during write to each of the destinations.																
11	Multiple sources and multiple destinations	MOD0 or MOD1 during write to each of the destinations.																

**Table 6. Operation Flag Descriptions (cont)**

Name	Description									
FS (Fill Shortage)	<p>When the coordinate conversion is made during the arbitrary angle rotate copy, some points may not be drawn. FS specifies whether to draw these points.</p> <table border="0"> <thead> <tr> <th data-bbox="365 343 387 360">FS</th> <th data-bbox="443 343 516 360">Function</th> </tr> </thead> <tbody> <tr> <td data-bbox="365 363 378 380">0</td> <td data-bbox="443 363 561 380">X Points drawn</td> </tr> <tr> <td data-bbox="365 384 378 401">1</td> <td data-bbox="443 384 589 401">X Points not drawn</td> </tr> </tbody> </table> 	FS	Function	0	X Points drawn	1	X Points not drawn			
FS	Function									
0	X Points drawn									
1	X Points not drawn									
PL (Pattern Line Length)	<p>Specifies whether a 16-bit or 32-bit pattern is to be used for line drawing.</p> <table border="0"> <thead> <tr> <th data-bbox="378 770 400 787">PL</th> <th data-bbox="430 770 548 787">Pattern Length</th> <th data-bbox="597 770 651 787">Pattern</th> </tr> </thead> <tbody> <tr> <td data-bbox="378 795 391 813">0</td> <td data-bbox="430 795 484 813">16 bits</td> <td data-bbox="597 795 883 813">PNTCNT contains the 16-bit pattern.</td> </tr> <tr> <td data-bbox="378 821 391 838">1</td> <td data-bbox="430 821 484 838">32 bits</td> <td data-bbox="597 821 1154 867">PNTCNT contains the first 16 bits of the pattern; DH contains the next 16 bits. The pattern cannot be initialized by setting IP = 0.</td> </tr> </tbody> </table>	PL	Pattern Length	Pattern	0	16 bits	PNTCNT contains the 16-bit pattern.	1	32 bits	PNTCNT contains the first 16 bits of the pattern; DH contains the next 16 bits. The pattern cannot be initialized by setting IP = 0.
PL	Pattern Length	Pattern								
0	16 bits	PNTCNT contains the 16-bit pattern.								
1	32 bits	PNTCNT contains the first 16 bits of the pattern; DH contains the next 16 bits. The pattern cannot be initialized by setting IP = 0.								