

oettle & reichler
datentechnik

Schießgrabenstr. 28 a
8900 Augsburg 1

Tel.: (0821) 15 46 32

R G B

Farbgrafik Video Controller

H A N D B U C H

Inhaltsverzeichnis

Schaltungsbeschreibung	4
Busanschluß	4
I/O Prom Pinbelegung	5
Video-Memory-Management	5
Control Logik	5
Takt und Pixelfrequenz	5
Steuersignale	6
Hardware Zoom	7
Video-Ram Adressierung	8
Farbe	8
Verdoppelung des Adreßbereichs	8
Einsatz von 256 kBit Rams	8
Synchron-Signale	9
Versorgungs-Spannungen	10
I/O Interface	10
Adreßbelegung	11
Portbelegungen	11
PIO Port A und Eprom	11
PIO Port B	11
PIO Port C	12
Farb-Selektion	12
Hardware-Zoom	13
Konfigurationen	13
Video-Speicher-Konfigurationen	13
256 kBit	13
512 kBit	14
Dot-Clock-Frequenz	14
Einstellungen	14
12 V Versorgung	14
Synchron-Signale	14
Speicheraufteilung	15
256 kBit	15
512 kBit	15

Einstellen der Video-Mischstufe	15
Monochrom 256 kBit	15
Monochrom 512 kBit	16
Farbe 3 x 256 kBit	16
Konfigurationsübersicht	16
Light-Pen-Anschluß	17
Ausgangssignale	17
Programmier Hinweise	17
Allgemeine Hinweise	17
Videoformate	18
Direkter Videospeicher-Zugriff	19
Charakter-Eprom Ansteuerung	19
Eprom Standard-Belegung	22
Programm-Beispiel	23
Datenblatt uPD 7220	29
Allgemeines	29
Das Zeichnen von Figuren	31
Parameter-Ram	32
Kommando Beschreibungen	33
AC-Charakteristik	36
Timing-Diagramm	37
Anhang	41
Stückliste	41
Bestückungsplan	42
I/O-Interface und Videomischstufe	43
Control-Logik	44
Video-Memory-Interface	45

Kein Teil dieser Veröffentlichung darf reproduziert, vervielfältigt, gespeichert oder übersetzt werden, ohne die ausdrückliche schriftliche Zustimmung von DATENTECHNIK oettle & reichler. Wir behalten uns das Recht vor, Änderungen, die einer Verbesserung einer Schaltung oder unserer Produkte dienen, ohne besondere Hinweise vorzunehmen. Für die Richtigkeit der hier gegebenen Daten, Schaltpläne, Programme und Beschreibungen wird keine Haftung übernommen.

Übersicht:

Das RGB-Grafiksystem ist ein kompletter Grafik-Videocontroller für Farbe oder Schwarz/Weiß aufgebaut auf einer einzigen ECB-Bus kompatiblen Europakarte. Es ermöglicht die Erstellung und Verarbeitung von hochauflösenden Bildschirmgrafiken (max. 3x1024x1024) und dient gleichzeitig als komfortable Text-Ausgabeeinheit. Beim Einsatz des RGB-Grafiksystems entfällt daher die Notwendigkeit von separaten grafischen und alphanumerischen Videoausgabeeinheiten. Zur Farberweiterung müssen keine teuren Zusatzboards zugekauft werden. Alternativ kann das RGB-Modul mit 16k x 4 oder 64k x 4 Speicherbausteinen bestückt werden.

Schaltungsbeschreibung:

Kern der RGB-Grafik Einheit bildet der "Graphic Display Controller" uPD 7220 von NEC, der zwischen Bus und Bildspeicher angeordnet ist. Sein leistungsfähiger Befehlssatz, seine Möglichkeit zur selbstständigen Erzeugung von Figuren und seine DMA-Fähigkeit bedeuten eine wesentliche Entlastung des zentralen Prozessors. Zum Setzen eines Bildpunktes sind lediglich vier Taktzyklen erforderlich. Das Ausfüllen des 3/4 MBit großen Bildspeichers erfolgt bei 16 Mhz Taktrate mit 24 Millionen Punkten/s. Das Setzen eines einzelnen Punktes im RMW-Zyklus dauert 2 us.

I/O-Interface:

Gesteuert wird das RGB-Modul direkt über den ECB-Bus als reine I/O-Karte. Ein externes Prozessor-Modul übergibt die Daten direkt über den Bus, wodurch die Datenübertragungsraten gegenüber seriell über RS-232 gesteuerte Einheiten stark erhöht werden. Über einen bidirektionlen LS-Buffer sind die Datenleitungen des Systembus mit denen der drei I/O Bausteine uPD 7220, PIO 8255 und Eprom 2716 verbunden. Als Steuersignale dienen /RD, /WR und /IORQ. Als reine I/O-Karte werden nur die Adressen A0-A7 benötigt. Zur Adreßselektion dient der Prom IC 05, der es ermöglicht, die I/O-Bausteine frei in den 256 Byte Adreßraum zu legen. A2 - A7 vom Hostprozessor sind direkt mit A2 - A7 des Proms 24SA10 (256x4) verbunden. A0 und A1 des Proms sind unbenutzt und liegen auf 0 Volt. Die Open-Collector-Ausgänge des Proms dienen zur Freigabe der jeweiligen I/O-Einheit:

I/O Prom IC 05 Pinbelegung:

D0: Enable uPD 7220, active low

D1: Enable PIO 8255, active low

D2: Enable Eprom 2716, active low

D3: Enable Datentreiber IC 06 (74 LS 245), active low

Standardmäßig (I/O Adressen 30h - 3Fh) sind folgende Werte einprogrammiert:

30h 0110b - 6h - Select uPD 7220

34h 0101b - 5h - Select PIO 8255

38h 0011b - 3h - Select Eprom 2716

3Ch 0011b - 3h - Select Eprom 2716

Video-Memory-Management:

Zur Ansteuerung des Video-Rams stellt der GDP einen gemultiplexten 16 Bit breiten Datenbus und einen 18 Bit breiten Adreßbus zur Verfügung (AD0-AD15, A16, A17). Daraus ergibt sich die Forderung, pro Farbebene jeweils 16-Bit-breite Datenbuffer, Speicherbausteine und Video-Schieberegister zu verwenden. Auf Grund des Einsatzes dynamischer Speicherbausteine (pro Ebene 4x TMS 4416 = 16k x 4 oder 4x TMS 4464 = 64k x 4) müssen die Speicher-Adressen gemultiplext werden. Dazu dienen die Multiplexer 74 LS 157 (IC 19/20). In den Display-Phasen liefert der GDP die Adressen für die Speicher, deren Inhalt jeweils über zwei 8-bit Schieberegister in einen seriellen Datenstrom umgewandelt werden. In den Blanking-Phasen, also während BLANK aktiv ist, findet der Datenaustausch zwischen Speicher und GDP zur Datenmanipulation sowie der notwendige Refresh beim Hardwarzoom statt.

Control Logik:**Takt und Pixelfrequenz:**

Ein Quarzoszillator aufgebaut aus IC 03 und Q bestimmt direkt die Pixelfrequenz, d.h. die Frequenz, mit der die Daten aus den Schieberegistern geschoben und dem Monitor zugeführt werden. Der synchrone Zähler (IC 01) teilt die Pixelfrequenz durch 2, 4 und 8. Daraus wird 2xWCLK abgeleitet, das ein 8tel der Frequenz des Pixelclocks entspricht. Das Taktsignal 2xWCLK wird dem GDP zuge-

führt und dient zur Synchronisation. $2xWCLK$ besitzt die doppelte Frequenz eines Display-Zyklus. In einem Display Zyklus werden die 16 Bit breiten Schieberegister neu geladen und die 16 Datenbits seriell ausgelesen. Ein RMW-Zyklus, d.h. ein Datenaustausch zwischen Videospeicher und GDP, benötigt vier $2xWCLK$ Zyklen.

Beispiel:

Pixelfrequenz - 16 MHz, 62.5 ns

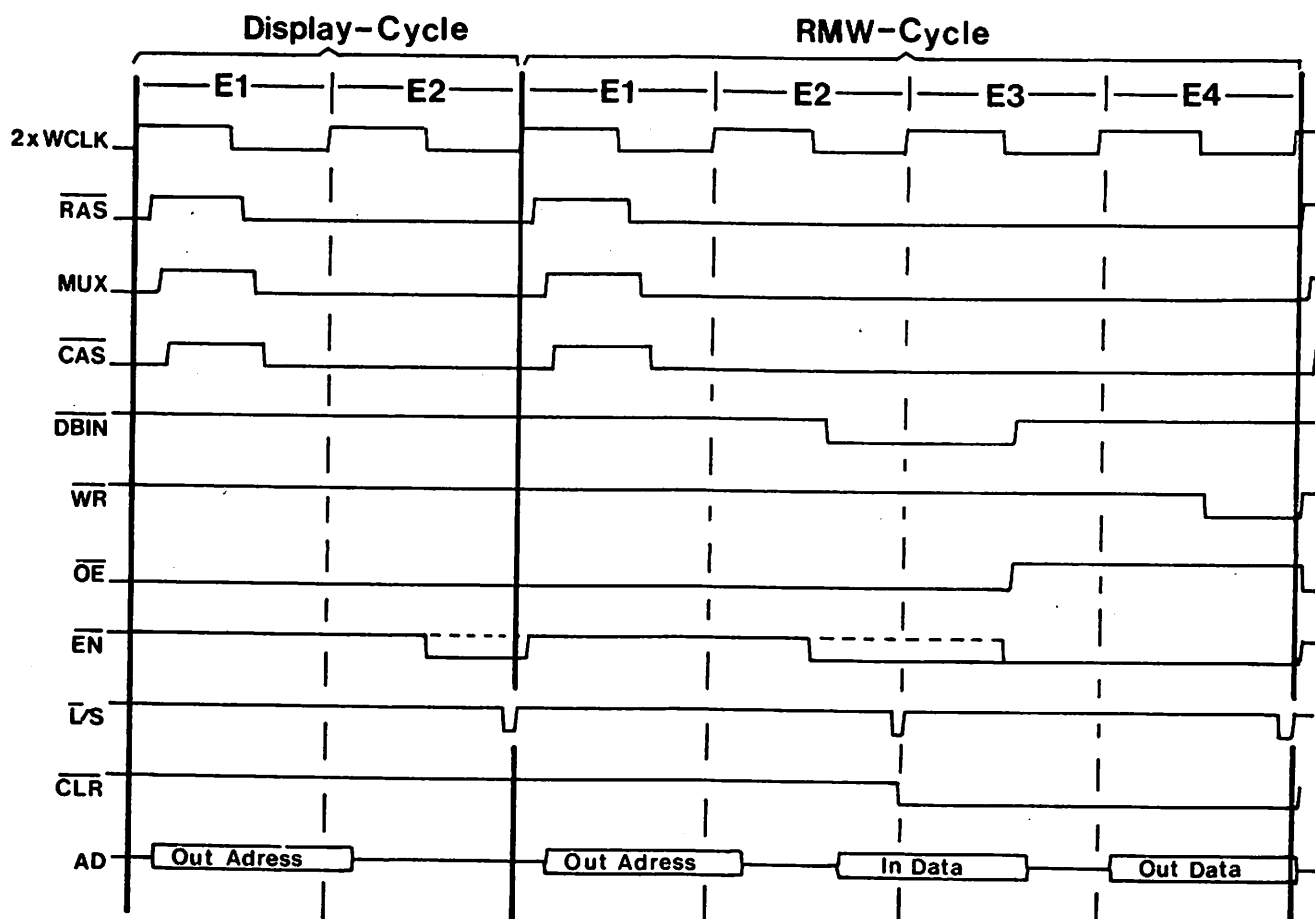
$2xWCLK$ - 2 MHz = Pixelfrequenz/2, 500 ns

Displayzyklus - 1 MHz = $2xWCLK / 2 = WCLK$ (Writeclock), 1 μ s

RMW-Zyklus - 0.5 MHz = $2xWCLK / 4$, 2 μ s

Steuersignale:

Der Zusammenhang der Steuersignale ergibt sich aus folgendem Diagramm:



Der GDP stellt aus dem 2xWCLK Takt die beiden Signale ALE (Adress Latch Enable) und DBIN (Data Bus In) zur Verfügung. ALE signalisiert dabei, daß die Adressen ADO - A17 gültig sind und daher ein RAS-CAS Zyklus zur Adressierung der Speicher durchgeführt werden soll. Das DBIN-Signal dient zur Unterscheidung, ob es sich um einen normalen Display Zyklus handelt (2 Taktphasen) oder ein RMW- Zyklus (4 Taktphasen). Das Blank Signal dient zur Dunkel-Steuerung der Schieberegister außerhalb der activen Display-Phasen, z.B. zum Beschreiben des Videospeichers und zum Refresh.

Aus diesen drei Signalen und dem 2xWCLK Takt werden sämtliche Steuersignale erzeugt:

Video-Speicher:	RAS, CAS, WR_r , WR_g , WR_b , OE
Adreßmultiplexer:	MUX
Databuffer:	Enable E_r , E_g , E_b , Direction R/W
Shift-Register:	Load/Shift, Clear

Hardware Zoom:

Ein Zoomen des Bildes (Display-Zoom) wird durch ein Verlängern der entsprechenden Steuersignale erreicht.

1. Horizontale Streckung: Die Schieberegister werden mit niedrigerer Frequenz betrieben und der GDP längt das ALE-Signal entsprechend.

2. Vertikale Streckung: Der GDP adressiert den Video-Ram zeilenweise mehrfach.

Der Multiplexer (IC 00) dient zur Bestimmung des Shift-Clocks der Schieberegister. Er wird durch den Teiler IC 01 mit den entsprechenden Vielfachen der nominellen Pixelfrequenz beliefert. Dabei sind Zoomfaktoren von 1-fach, 2-fach, 4-fach und 8-fach möglich.

Video-Ram Adressierung:**1. Farbe:**

Um jeweils eine der drei 16k x 16 (64k x 16) großen Farbebenen zu adressieren werden AD0-AD13 (AD0-AD15) vom GDP herangezogen. Über Multiplexer werden diese den Speichern zugeführt. Das Auslesen der Speicher in die Video-Schieberegister während der Display-Phasen geschieht normalerweise für alle drei Ebenen parallel, d.h. alle Speicherebenen bekommen ein OE-Signal (Output Enable). Die von der PIO gelieferten Farbwertsignale geben die jeweiligen Farbebenen zum Lesen durch den GDP frei.

Während der RMW-Zyklen (Modifikation des Bildspeichers) müssen jedoch gezielt einzelne Farbebenen zum Lesen und Schreiben vom GDP angesprochen werden. D.h. die Signale Memory-WR und Data-buffer-Enable müssen für jede Ebene getrennt gesteuert werden.

Auch die Selektion mehrerer Ebenen auf einmal zum gleichzeitigen Schreiben ist möglich. Beim Lesen darf jedoch nur gezielt eine Ebene angesprochen werden. Diesen Vorgang steuert der Prom IC-14, der in Abhängigkeit von den Farbwertsignalen, dem Read/Write Signal und 'Load Mask' die Freigabe der Datenpuffer für die jeweilige Farbebene steuert.

2. Verdoppelung des Adreßbereichs bei Schwarz/Weiß:

Zusätzlich besteht die Möglichkeit der Verkettung zweier Speicherebenen zu einer großen Ebene, um einen doppelt so großen Video-Ram zu erhalten. Nun dient zur Selektion der Speicherebenen nicht mehr das Farbwertsignal der PIO, sondern die nächst höhere Adresse AD14 des GDP. Diese Option läßt sich per Jumper einstellen.

3. Einsatz von 256 kBit Rams:

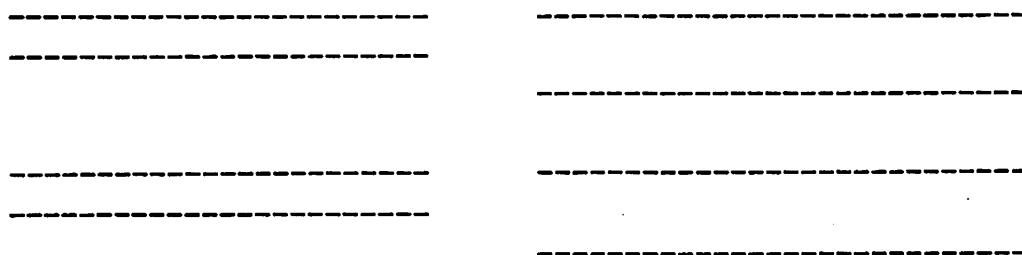
Der Einsatz von 256 kBit Rams mit einer Organisation von 4 x 64k (TMS 4464) ist auf dem RGB-Modul durch bloßen Austausch der Speicherchips möglich. Alle dafür notwendigen Adresserweiterungen sind auf der Karte schon vorgenommen und an die RAM-Bausteine geführt. Werden diese Bausteine eingesetzt, so ergibt dies eine Vervierfachung der RAM-Kapazität. Die Verdoppelung des Adressbereichs durch Aneinanderkettung zweier Speicherebenen

(siehe 2) ist dabei nicht mehr möglich. Doch dürfte die hierbei erzielbare Auflösung von 1024 x 1024 Bildpunkten ohnehin den meisten Anforderungen genügen.

Synchron-Signale bei Interlaced:

Bei Interlaced Anwendungen (Zweibild-Verfahren) sind die vom GDP gelieferten Synchronsignale Vsync und Hsync nur unzureichend aufeinander abgestimmt. Es kommt zu einer unerwünschten teilweisen Überlappung der beiden Halbbilder. Beim Halbbildverfahren wird der Einsatz der Vertical-Synchron-Signale für gerade und ungerade Zeilen um eine halbe Zeile zueinander verschoben. Der Einsatz des geraden Halbbilds beginnt in der Mitte einer Bildzeile. Der Einsatz des ungeraden Halbbilds am Ende der Bildzeile, also am Anfang der vorderen Schwarzscher horizontal. Daher beträgt beim uPD 7220 dieser Versatz der beiden Halbbilder nur ca. 2 : 3 einer Zeile, wodurch sich die ungeraden und geraden Zeilen teilweise überlappen. Dies wirkt sich sehr störend bei der Bild-Darstellung aus, da der vertikale Abstand zwischen zwei Bildpunkten unterschiedliche Werte annimmt. Beim RGB-Modul wird der Zeitpunkt des ungeraden Vertical-Synchronsignals bis zum nächsten HSync-Signal hinausgezögert. Dadurch beträgt das Verhältniss zwischen ungeraden und geraden Zeilen die erforderlichen 50 %. Dazu dienen IC 04, IC 10 und diverse Gatterbausteine. Zudem wird ein negatives Composite-Sync Signal aus Vsync und Hsync erzeugt.

Folgendes Diagramm verdeutlicht diesen Zusammenhang. Links ist stark vergrößert die Zeilenanordnung ohne Verzögerung dargestellt. Gerade und ungerade Zeilen rücken zu nahe zusammen, wodurch es zu einer Überlappung der Halbbilder kommt. Dazwischen entstehen große Lücken.



Ohne Verzögerung:

Mit Verzögerung von VSync odd:

Findet kein Halbildverfahren Anwendung, so ist diese Zusatz-Schaltung ohne Bedeutung. Der Vertikal-Synchron-Impuls setzt hierbei immer beim Start der vorderen horizontalen Schwarzschulter ein.

Versorgungs-Spannungen:

Alle Bausteine des RGB-Moduls benötigen nur eine 5 V Versorgungsspannung. Auch die Video-Endstufe ist für eine 5 V Versorgung ausgelegt. Doch steile Schaltflanken im System können sich auf die 5 V Versorgung übertragen, was sich mindernd auf die Qualität des Video-Bilds auswirken kann (evt. leichte Schatten in großen Weiß-Bereichen). Daher ist eine getrennt Spannungsversorgung für die Video-Endstufe vorgesehen, die allerdings aus der 12 V Leitung des ECB-Bus gespeist wird. Der Spannungsregler IC 43 erzeugt eine hochstabilisierte 5 Volt Spannung die zur Versorgung der Videoendstufe dient.

I/O - Interface:

Aus der Sicht des Programmierers teilt sich das RGB-Modul in drei verschiedene I/O-Funktionsblöcke auf:

GDP uPD 7220:	Manipuliert Video Ram Datenaustausch zwischen System und Videoram
PIO 8255:	Adressiert Charakter Eprom Einstellen des Hardware Zooms Anwählen des Farbwertsignals Lesen und Beschalten der DMA-Signale des GDP
EPROM:	Auslesen des über PIO adressierten Charakters

Vom RGB-Modul werden dafür 16 I/O-Adressen belegt, die sich in 8 Byte-Blöcken beliebig im I/O-Adressraum verteilen lassen. Die I/O-Adressebelegung bestimmt der PROM IC 05.

Adreßbelegung:

Standardmäßig belegt das RGB-Modul folgende 16 I/O-Adressen:

30H GDP Status Read, Parameter into FIFO
31H GDP FIFO Read, Command into FIFO
32H GDP , entspricht 30H
33H GDP , entspricht 31H

34H PIO Port A: Output, adresiert EPROM A3 - A10
35H PIO Port B: Input, Bit 0 liest DRQ von GDP
36H PIO Port C: Output, select Zoom-Colour, DACK
37H PIO Control: Bestimmt Betriebsart der PIO

38H-3FH EPROM 2716: Lesen einer Charakter-Row

Portbelegungen:**PIO Port A Belegung und Eprom:**

Port A der PIO adressiert direkt die höherwertigen Adressen A3 - A10 des Charaktereproms. Dieser Port muß dabei als Ausgang geschaltet werden. Das gewünschte ASCII-Zeichen kann direkt an Port A übergeben werden, danach läßt sich der Inhalt des EPROM's wie ein I/O-Kanal auslesen. Dabei entspricht jede Zeile (Row) eines Charakters einer I/O-Adresse des Eproms. Der Eprom belegt 8 aufeinanderfolgende I/O-Adressen (A0 - A7). Damit ist auf einfachste Weise eine Charakter-Matrix von 8 x 8 auslesbar, die direkt in das Parameter-Ram des GDP's zum Schreiben übergeben werden kann. Vor der Übergabe kann der Prozessor das Charakter noch manipulieren (z.B. Unterstreichen, Invertieren usw.). Sind höhere Matrix-Auflösungen erwünscht, so muß über Port A eine neue Adresse angewählt werden, um auf die nächsten Elemente des Charakters zugreifen zu können.

PIO Port B Belegung:

Über Bit 0 kann der DRQ-Anschluß des GDP gelesen werden. Dieser Port muß dabei als Eingang geschaltet sein. Das DRQ Signal des GDP's ist activ high. Bit 1-7 von Port B sind unbelegt und frei als User-Input.

PIO Port C Belegung:

Port C der PIO 8255 ist als Ausgang geschaltet und dient zur Steuerung des Farbwerts, des Hardwarezooms und zur Freigabe des DACK-Signals an den GDP bei DMA-Zugriffen:

Bit 0 SELBLUE: Blau Ebene frei zum Lesen und Schreiben, active low
BIT 1 SELGREN: Grün Ebene frei, active low
BIT 2 SELRED : Rot Ebene frei, active low

BIT 3 ZOOMA: Zoom Faktor low Bit, active high
BIT 4 ZOOMB: Zoom Faktor high Bit, active high

BIT 5 DACK: Gibt Dack an GDP frei, active low

BIT 6/7: Frei als Ausgang

Farb Selektion:

Über Bit 0-2 Port C werden die jeweiligen Farbebenen zum Lesen und Beschreiben durch den GDP freigegeben werden. Auch das gleichzeitige Anwählen mehrerer Farben ist gestattet. Somit können in einem einzigen Schreibdurchgang mehrere Farbebenen gesetzt oder rückgesetzt werden (z.B. Löschen des Video-Rams oder belegen mit einer Hintergrundfarbe).

Beim Lesen des Video-Speichers sorgt IC 14 dafür, daß bei Selektion mehrerer Farbebenen nur eine priorisierte Ebene im RMW-Cyclus gelesen wird. Dabei hat Rot die höchste Priorität, Blau die niedrigste. Ist z.B. Ebene Rot, Blau und Grün zum Schreiben freigegeben (= weiß), so wird im Read-Modify-Write Cyclus des GDP nur die Rot-Ebene gelesen und modifiziert, jedoch in alle drei Ebenen wieder geschrieben. Somit wird die Rotinformation in die Blau- und Grün-Ebene übertragen. Ist die Übertragung der Farbinformation von einer Ebene in die andere unerwünscht, so sind die Farbebenen getrennt zu selektieren.

Port C:			GDP:		VIDEO:
BIT 2:	1:	0:	READ:	WRITE:	
0	0	0	rot	rot/grün/blau	= Weiß
0	0	1	rot	rot/grün	= Gelb
0	1	0	rot	rot/blau	= Purpur
0	1	1	rot	rot	= Rot
1	0	0	grün	grün/blau	= Indigo
1	0	1	grün	grün	= Grün
1	1	0	blau	blau	= Blau
1	1	1	-	-	= Schwarz

Hardware-Zoom:

In Verbindung mit dem Zoom Kommando an den GDP müssen BIT 3/4 von Port C entsprechend programmiert werden:

Port C:		
BIT 4:	3:	
0	0	Zoom Faktor 1, dh. natürliche Darstellung
0	1	Zoom Faktor 2
1	0	Zoom Faktor 4
1	1	Zoom Faktor 8

Video-Speicher Konfigurationen:

Der Video Speicher läßt sich in drei verschiedenen Konfigurationen aufteilen.

1. 256 kBit (1 MBit):

Zur Darstellung von Farb- und Schwarz/Weiß Bildern wird der Speicher in drei identische Ebenen von 256 kBit (1 MBit) aufgeteilt. Somit kann z.B. eine Auflösung von 3 x 512 x 512 Pixel (3 x 1024 x 1024) erreicht werden. Das Auslesen des Video-Rams zum Monitor geschieht dabei parallel für alle drei Ebenen. Schreib- und Lese-Vorgänge durch den GDP steuern die Farb-Select Signale PIO Port C Bit 0-2. Der Adreßbereich des Video-Rams erstreckt sich dabei von 0000H-3FFFH (0FFFFh) zu 16 Bit Worten (A0 bis A13/A15).

2. 512 kBit:

Wird eine noch höhere Auflösung gewünscht, so läßt sich bei Einsatz von 64 k - Speicherbausteinen die Rot und Grün Ebene zu einem großen Speicherblock hintereinanderschalten. Somit wird eine Auflösung von 512 kBit erzielt z.B. 1024 x 512 bei 64 k - Rams. Die beiden Ebenen werden getrennt voneinander Ausgelesen und Beschrieben, zur Selektion dient GDP-A14. Dadurch wird der Adreßraum auf 0000h - 7FFFh aufgeweitet. Diese Betriebsart ist nur für monochrome Anwendungen geeignet, die Blau Ebene braucht dabei nicht bestückt zu werden (IC 35-41 entfallen, R_p entfällt), die Farbsteuersignale PIO Port C Bit 0-2 sind wirkungslos.

Dot-Clock:

Die Video-Frequenz des RGB-Moduls bestimmt Quartz Q in Verbindung mit C1. Die maximale Pixel-Frequenz wird durch die Schieberegister (74 LS 166) bestimmt. Bei normaler Umgebungstemperatur beträgt F_{max} 24 MHz. Bei zu hohen Frequenzen werden die Schieberegister unsauber geladen, da die vorgeschriebene Setup-Time für das Load-Signal der Schieberegister nicht eingehalten wird. Bei höheren Frequenzen als 16 MHz ist C1 zu erniedrigen, bei niedrigeren Frequenzen ist umgekehrt zu verfahren:

Beispiele: 16 MHz: C1: 1,2 nF - 680 pF; 24 MHz: C1: 470 pF

Steckbrücken und Einstellarbeiten:

12 V Versorgung:

Steht eine 12 V Versorgung im System zur Verfügung, so ist IC 43 einzulöten, B2 entfällt. Beim reinen 5 V Betrieb entfällt IC 43, dafür ist B2 einzusetzen.

Synchron Signale:

Die Polarität der Signale HSync und VSync kann über 6 zwischen IC 08 und IC 09 gelegene Brücken unabhängig voneinander eingestellt werden. Die Einstellung ist nur bei Anschluß eines Parallel-Monitors erforderlich, der die Synch Signale getrennt verarbeitet. Der mit V bzw. H (Vsync bzw. Hsync) gekennzeichnete Anschluß muß mit dem rechts bzw. links davon liegenden Anschluß verbunden werden. '+' entspricht dabei positiver, '-' negativer

Polarität der Sync-Signale. Standardmäßig sind die Brücken für negative Synchrosignale gesetzt.

Speicher Aufteilung:

Um die Speicheraufteilung zu bestimmen, sind die unter IC 08 und unter IC 09 mit (1) bis (6) gekennzeichneten Lötunkte entsprechend zu verbinden.

Die bei IC 44 gelegenen Bauteile und Brücken (B0, B1, R13, R14, D2 und D3) sind ebenfalls entsprechend einzusetzen.

Standardmäßig sind alle erforderlichen Brücken für den Einsatz drei getrennter Speicherebenen (3x256k) gesetzt. Beim Koppeln von zwei Speicherebenen (512 kBit) ist es erforderlich diese Standardbrücken zu unterbrechen und umzusetzen.

256 kBit: Lötunkt 2-3 und 5-6 verbinden (Standardeinstellung)
B1 und B0 entfallen
R13 und R14 entfallen
D2 und D3 durch Brücke ersetzen (Standardeinstellung)

512 kBit: Lötunkt 2 mit 1 und Lötpt. 5 mit 4 verbinden
(Brücke 2-3 und 5-6 unterbrechen !)
B1 und B0 einsetzen
R13 und R14 einsetzen
D2 und D3 einsetzen

Einstellen der Video-Mischstufe:

Die über IC 43 gelegenen Widerstände R_r , R_g , R_b , R_s und R_v bestimmen die Zusammensetzung und Amplitude des Videosignals: R_s bestimmt dabei den prozentualen Anteil des Synch-Signals im BAS-Signal. Dabei erhöhen niedrigere Widerstandswerte von R_s den Anteil des Synch-Signals.

Folgende Werte sind dabei einzusetzen:

Monochrom 256 kBit:

Speicherebene Rot bestücken (IC 17, 18, 21-26). Speicherebene Grün und Blau entfällt (IC 27 - IC 42). $R_r = 270 \text{ Ohm}$, R_v und R_s jeweils 270 Ohm, R_g und R_b entfallen.

Monochrom 512 kBit:

Speicherebene Blau entfällt (IC 35 - IC 42). Speicherebenen Rot und Grün bestücken. R_r und R_g 300 Ohm, R_s 390 Ohm, R_v 240 Ohm, R_b entfällt.

Farbe 3 x 256 kBit:

Alle drei Speicherebenen sind hierbei zu bestücken. R_s entfällt, R_v durch Brücke ersetzen. R_r , R_g und R_b jeweils 270 Ohm (150 - 470 Ohm). Die besten Auflösungsergebnisse ergeben sich, falls die Abschluß-Widerstände R_r , R_g und R_b für die jeweiligen Farbsignale erst am Ende des Video Kabels, also im Farbmonitor selbst eingesetzt werden. Voraussetzung hierzu ist allerdings, daß im Monitor eine 5 V Spannung zur Verfügung steht. Die Leitungen RED, GREEN und BLUE sind hierzu jeweils über 270 Ohm (150 - 470 Ohm) an 5 V zu legen. Dafür entfallen R_g , R_r und R_b . Großen Einfluß auf die Qualität des Videobildes hat dabei das Videokabel. Übersprechen zwischen den einzelnen Signalen ist durch Abschirmung der einzelnen Leitungen zu verhindern. Sehr gute Ergebnisse lassen sich mit einem Flachbandkabel erzielen, wobei jede zweite Leitung Masse führt.

Konfigurations-Übersicht:

	3 x 256k Farbe	1 x 256k S/W	1 x 512k S/W	Stand. Brücken
Lötpkte. 1,2,3	2-3	2-3	2-1	2-3
Lötpkte. 4,5,6	5-6	5-6	5-4	5-6
Brücken B 0/1	-	-	x	-
Wid. R13 / R14	-	-	1k2	-
Dioden D2 / D3	*	*	x	*
Wid. R_g	270R	-	300R	
Wid. R_r	270R	270R	300R	
Wid. R_b	270R	-	-	
Wid. R_v	*	270R	240R	
Wid. R_s	-	270R	390R	
IC 27 - 34	x	-	x	
IC 35 - 42	x	-	-	

- : Bauteil entfällt

x : Bauteil einsetzen

* : Bauteil durch Brücke ersetzen

Light-Pen Anschluß:

Der uPD 7220 ist für den Einsatz eines Lightpens geeignet. Dazu dient Anschluß 8 am 9 poligen D-Stecker S2. Dieser Anschluß führt an IC 16 Pin 21 (Lpen) und ist standardmäßig durch die Brücke B3 auf 0 Volt gelegt, d.h. inaktiviert. Wird ein Lightpen eingesetzt, so ist die GND-Brücke B3 unterhalb von IC 16 zu unterbrechen.

Ausgangssignale:

Am 9 poligen D-Stecker S2 werden folgende Signale zum Anschluß eines Farb- oder Schwarz/Weiß Monitors zur Verfügung gestellt:

- 1 HSYNC : Polarität über Steckbrücke wählbar (Std. neg.), TTL
- 2 VIDEO : Composite Video (BAS) neg. Synch 68 Ohm
- 3 GREEN : Grün Signal (TTL open Collector) positive
- 4 RED : Rot Signal (TTL open Collector) positive
- 5 BLUE : Blau Signal (TTL open Collector) positive
- 6 VSYNC : Polarität über Steckbrücke wählbar (Std. neg.), TTL
- 7 GND : Masse 0 V
- 8 LPEN : Lightpen Anschluß (Std. über B3 auf 0 V)
- 9 SYNC : Composite Sync (V-& HSYNC) negativ, TTL

Als Option wird am Video Miniatur Stecker S1 ein CCIR-Norm gerechtes BAS Ausgangssignal bereitgestellt (0,5 - 1,5 V V_{ss} , 75 Ohm).

Allgemeine Hinweise:

1. Der GDP ist grundsätzlich im 'Graphics Mode' zu betreiben. Mixed Graphics & Character Mode ist auch denkbar; diese Betriebsart kann jedoch auch nur mit Graphic Areas belegt werden.
2. Muß nur der Inhalt des Videorams erhalten bleiben, der auch tatsächlich auf dem Bildschirm dargestellt wird, so kann auf den Refresh der dynamischen Speicher verzichtet werden (Dynamic Refresh Cycles Disable), da dies die normalen Display-Zyklen übernehmen. Dies bringt leichte Geschwindigkeitsvorteile mit sich, schließt jedoch die Zoom-Funktion aus (1-facher Zoom).
3. Nach Möglichkeit sollte die Non-Interlaced Darstellung verwendet werden, da nur sie ein absolut flimmerfreies Bild bei 50 Hz Bildfrequenz gewährleistet. 'Interlaced Repeat Field for

Character Display' zeigt bei großen Weiß-Feldern schon geringe Flimmererscheinungen. Bei Interlaced-Anwendungen sollte eine lang-nachleuchtende Bildröhre eingesetzt werden.

4. Die Werte für das CCHAR Kommando sollten dem unten aufgeführten Programmbeispiel entnommen werden

5. Der maximale Adreßbereich des Videospeichers beträgt bei 256k Anordnung 0000-3FFFh (bei 256 kRams 0-0FFFFh). Bei 4000H (10000h) wird wieder der Anfang des Rams adressiert. Bei der 512k Anordnung verdoppelt sich der Adreßbereich auf 0000-7FFFh.

6. Der von uns gelieferte Charakter-Eprom erzeugt waagrechte Alphanumerik bei einer Zeichen-Richtung (DIR) 02. Senkrechte Schrift wird durch DIR = 04 erzeugt. Der Inhalt des Eproms ist dadurch direkt in das Parameter-Ram des GDP zu übertragen.

Videoformate:

Der GDP erlaubt es jedes beliebige Videoformat frei zu programmieren. Zeichen pro Zeile, Zeilen pro Bild, Länge der vorderen und hinteren Schwarzschnur und der Synchronisationssignale sind frei wählbar. Damit läßt sich nahezu jedes beliebige Bildschirmformat exakt auf die gewählte Pixelfrequenz und den Anforderungen des Monitors anpassen. Folgende Werte sind dabei bei der Standardfrequenz von 16 MHz empfehlenswert, um ein CCIR-kompatibles Video-Signal mit 64 µs Zeilenfrequenz und 20 ms Bildwechselfrequenz zu erzeugen:

	Non-Interlaced:	Interlaced:
Punkte horizontal:	704	608
Punkte vertikal:	288	429
Vordere Schwarz-Schulter horiz.:	6	8
Hintere Schwarz-Schulter horiz.:	10	13
Horizontal Synch.:	4	5
16 Bit Worte pro Zeile:	44	38
Vordere Schwarz-Schulter vert.:	1	45
Hintere Schwarz-Schulter vert.:	18	47
Vertikal Synch.:	5	7
Zeilen pro Bild:	288	214

Alle horizontal Werte beziehen sich auf die Länge eines 16 Bit Worts (bei 16 MHz = 1 us), alle vertikalen Werte beziehen sich auf die Länge einer Zeile (CCIR Norm = 64 us).

Direkter Video-Speicher Zugriff:

Schnelle Transfers vom Videospeicher zum System (z.B. Massenspeicher) und umgekehrt erfolgen unter Ausnutzung der DMA-Steuersignale des GDP. Dieser ist mit Hilfe der DMA-Steuersignale in der Lage direkt mit einem DMA-Baustein zusammenzuarbeiten. Aus Platzgründen ist auf dem RGB-Modul der Einsatz einer eigenen DMA unmöglich. Stattdessen wird dem GDP ein DMA-Betrieb simuliert, indem die CPU den Transfer unter Beschaltung der Signale DRQ und DACK übernimmt:

Nach dem Kommando DMAR oder DMAW an den GDP legt die CPU Bit 5 von PIO Port C auf low, und gibt damit das Signal DACK an den GDP frei. Jeder nun folgende I/O-Befehl an den GDP (Read oder Write) aktiviert nun auch das DACK-Signal. Ob der GDP zum Datenaustausch bereit ist, läßt sich über den Anschluß DRQ feststellen, der von der CPU über PIO Port B Bit 0 getestet werden kann. Ist DRQ aktiv, kann ein Lese- oder Schreib Befehl an den GDP, der dabei DACK bekommt, erfolgen. Nach Beendigung des Transfers muß die CPU das DACK Signal wieder sperren, indem sie Bit 5 von PIO Port C wieder auf high legt. Andere Möglichkeiten des Datentransfers mit dem Videoram bestehen nicht, da z.B. die evt. denkbaren Befehle RDATA und WDATA im Graphics Mode nur Bit0-Transfers zulassen.

Charakter-Eprom Ansteuerung:

Der von uns standardmäßig gelieferte Eprom besitzt einen deutschen/internationalen Zeichensatz mit einer Auflösung von 8x12 Pixel pro Charakter. Das Auslesen und Einprogrammieren eines Zeichens soll an Hand des Zeichens 'j' (hex 6A) erläutert werden.

1. Der Hex-Wert des Ascii-Zeichens (6AH) wird direkt an PIO Port A übergeben, womit A3-A10 des Eproms beschaltetet sind:

A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0	
0	1	1	0	1	0	1	0	X	X	X	-> Eprom-Adress
!-----!											350H - 357h
ASCII (6A) an A10 - A3											ROW7 - ROW0

2. Auslesen der 8 Werte des Eproms und übertragen in Parameter-Ram GDP. Dabei entspricht die niedrigste Adresse des Eproms (38H) der obersten Reihe eines Charakters, die höchste Adresse des Eproms (3FH) der untersten Reihe eines Charakters. Zudem stehen die Bitfolgen Spiegelverkehrt im Eprom. Dies trägt entscheidend zur einfachen Programmierung des GDP bei, da diese Werte direkt in das Parameter-Ram des GDP geladen werden können.

3. Nach der Übergabe der 8 Reihen des Charakters an das Parameter-Ram des GDP wird ein FIGS-Kommando an den GDP zum Schreiben des Parameter-Rams in den Video-Speicher gegeben. Dabei müssen nur 2 Parameter (P1: Figure Type; P2: DC low) übergeben werden, da die notwendigen restlichen (P3: DC high = 0, P4: D low = 8, P5: D high = 0) automatisch auf den richtigen Wert gesetzt werden (Initial Value):

FIGS-Kommando: 4CH

FIGS-Parameter 1: 12H -> Graphic Character Drawing, Direction 2
2: 07H -> DC low -1 : 7+1 Rows schreiben

GCHRD-Kommando: 68H -> Video-Ram wird geschrieben

4. Nun müssen die restlichen 4 oberen Reihen des Charakters gelesen und in den GDP übertragen werden. Die neue Adresse des Eproms ermittelt sich folgendermaßen: Ascii-Zeichen um eine Stelle nach rechts rotieren, Bit 7 setzen, Carry Flag testen.

7	6	5	4	3	2	1	0	C	RLA	7	6	5	4	3	2	1	0	C
0	1	1	0	1	0	1	0	x	----->	1	0	1	1	0	1	0	1	0
									SETB 7									
6AH										0B5H								

Das somit gewonnene Zeichen dient als neue Adresse A10-A3 für den Eprom. Das Carry Bit entscheidet dabei ob die restlichen 4 Reihen aus den oberen (Carry = 1) oder unteren (Carry = 0) 4 Eprom I/O-Ports gelesen werden soll. Im vorliegenden Fall ist das Carry-Bit 0, somit wird die untere Hälfte des Eproms (38H - 3BH) gelesen.

A10	A9	A8.	A7	A6	A5	A4.	A3	A2	A1	A0		
1	0	1	1	0	1	0	1	0	X	X	-->	Eprom-Adress
-----											^	5A8H - 5ABH
											!	ROW11 - ROW8
											!	
											!-Carry = 0 : Read Port 38H -	
											Port 3BH	

5. Nun verfährt man wie in Schritten (2) und (3), mit dem Unterschied, daß nur 4 Werte (entsprechend Row 8 - Row 11) in den Parameter Ram geladen und in den Video-Ram geschrieben werden.

ow- Port	I/O	EPROM		In PRAM	FIGS-
Nr.: A:	Port:	Adress:	Inhalt:	GCHR Nr.:	Para. P2:
12	- 38H	5A8H	-00H = -----	4	-
11	B5H ! 39H	5A9H	-00H = -----	3	! DC = 3
10	! 3AH	5AAH	-10H = ---X----	2	!
9	- 3BH	5ABH	-00H = -----	1	-
8	- 38H	350H	-18H = ---XX----	8	-
7	! 39H	351H	-10H = ---X----	7	!
6	! 3AH	352H	-10H = ---X----	6	!
5	! 3BH	353H	-10H = ---X----	5	! DC = 7
4	6AH ! 3CH	354H	-10H = ---X----	4	!
3	! 3DH	355H	-10H = ---X----	3	!
2	! 3EH	356H	-11H = ---X---X	2	!
1	- 3FH	357H	-0EH = ----XXX-	1	-

Eprom Standard Belegung:

Ascii: Row 0-7		Row 8-11				
00-	000	400	29-	)	148	4A4
01-	008	404	2A-	*	150	4A8
02-	010	408	2B-	+	158	4AC
03-	018	40C	2C-	,	160	4B0
04-	020	410	2D-	-	168	4B4
05-	028	414	2E-	.	170	4B8
06-	030	418	2F-	/	178	4BC
07-	038	41C	30-	0	180	4C0
08-	040	420	31-	1	188	4C4
09-	048	424	32-	2	190	4C8
0A-	050	428	33-	3	198	4CC
0B-'Ae'	058	42C	34-	4	1A0	4D0
0C-'Oe'	060	430	35-	5	1A8	4D4
0D-'Ue'	068	434	36-	6	1B0	4D8
0E-	070	438	37-	7	1B8	4DC
0F-	078	43C	38-	8	1C0	4E0
10-'Pa'	080	440	39-	9	1C8	4E4
11-	088	444	3A-	:	1D0	4E8
12-	090	448	3B-	;	1D8	4EC
13-	098	44C	3C-	<	1E0	4F0
14-	0A0	450	3D-	=	1E8	4F4
15-	0A8	454	3E-	>	1F0	4F8
16-	0B0	458	40-	@	1F8	4FC
17-	0B8	45C	41-	A	200	500
18-	0C0	460	42-	B	208	504
19-	0C8	464	43-	C	210	508
1A-	0D0	468	44-	D	218	50C
1B-'ae'	0D8	46C	45-	E	220	510
1C-'oe'	0E0	470	46-	F	228	514
1D-'ue'	0E8	474	47-	G	230	518
1E-'ss'	0F0	478	48-	H	238	51C
1F-	0F8	47C	49-	I	240	520
20-' '	100	480	4A-	J	248	524
21-	108	484	4B-	K	250	528
22-	110	488	4C-	L	258	52C
23-	118	48C	4D-	M	260	530
24-	120	490	4E-	N	268	534
25-	128	494	4F-	O	270	538
26-	130	498	50-	P	278	53C
27-	138	49C	51-	Q	280	540
28-	140	4A0			288	544

52-	R	290	548
53-	S	298	54C
54-	T	2A0	550
55-	U	2A8	554
56-	V	2B0	558
57-	W	2B8	55C
58-	X	2C0	560
59-	Y	2C8	564
5A-	Z	2D0	568
5B-	[	2D8	56C
5C-	\	2E0	570
5D-	]	2E8	574
5E-	^	2F0	578
5F-	_	2F8	57C
60-	`	300	580
61-	a	308	584
62-	b	310	588
63-	c	318	58C
64-	d	320	590
65-	e	328	594
66-	f	330	598
67-	g	338	59C
68-	h	340	5A0
69-	i	348	5A4
6A-	j	350	5A8
6B-	k	358	5AC
6C-	l	360	5B0
6D-	m	368	5B4
6E-	n	370	5B8
6F-	o	378	5BC
70-	p	380	5C0
71-	q	388	5C4
72-	r	390	5C8
73-	s	398	5CC
74-	t	3A0	5D0
75-	u	3A8	5D4
76-	v	3B0	5D8
77-	w	3B8	5DC
78-	x	3C0	5E0
79-	y	3C8	5E4
7A-	z	3D0	5E8
7B-	{	3D8	5EC
7C-		3E0	5F0
7D-	}	3E8	5F4
7E-	~	3F0	5F8
7F-	DEL	3F8	5FC

'Pa' = Paragraph Zeichen
'ss' = Scharfes 's'

Programm Beispiel:

Die im Folgendem aufgeführten Programmbeispiele sollen die prinzipielle Programmierung des RGB-Moduls verdeutlichen. Besondere Beachtung ist der Initialisierungsroutine, die die Programmierung der I/O Bausteine und den Aufbau eines CCIR-Norm gerechten Video-Formats zeigt, und der CHAR-Routine, die den Umgang mit dem Charakter Eprom erläutert, zu schenken. Ein Anspruch auf Vollständigkeit kann aus diesem Beispiel (!) nicht erhoben werden. Sollen eigene Routinen erstellt werden, so ist ein genaues Studium des Datenblatts des GDP 7220 erforderlich.

```

0100                                ORG      100H
                                MACLIB   Z80

                                ;*****
                                ;***** PORTADDRESS *****
                                ;*****

                                ;      GDP-MODUL 30H - 3FH

0030 =      P$GDP$7220      EQU      30H
0034 =      P$GDP$ADR      EQU      34H
0038 =      P$GDP$CHAR      EQU      38H

                                ;PIO 8255: PORT A, OUTPUT ADDRESS
                                ;      PORT B, INPUT DO = DRQ-GDP
                                ;      PORT C, OUTPUT CONTROL

0035 =      P$GDP$DRQ      EQU      P$GDP$ADR+1
0036 =      P$GDP$CTRL      EQU      P$GDP$ADR+2
0037 =      P$GDP$MODE      EQU      P$GDP$ADR+3
0031 =      P$GDP$CMD      EQU      P$GDP$7220+1H      ;WR-COMMAND INTO FIFO
0030 =      P$GDP$PARAM      EQU      P$GDP$7220+0H      ;WR-PARAMETER INTO FIFO
0030 =      P$GDP$STATUS      EQU      P$GDP$7220+0H      ;RD-STATUS REGISTER
0031 =      P$GDP$DCRD      EQU      P$GDP$7220+1H      ;RD-DATA FROM FIFO

                                ;UTILITY SUBROUTINES CMDOUT, PAROUT, FIFRDY

                                ;***** CMDOUT *****
                                ; SEND COMMAND IN <A> TO GDP COMMAND REG.
                                ; SAVE REGISTERS
                                ;*****

0100 CDOC01      CMDOUT: CALL      FIFRDY      ; GDP READY ?
0103 D331      OUT      P$GDP$CMD
0105 C9      RET

                                ;***** PAROUT *****
                                ; SEND PARAMETER IN <A> TO GDP
                                ; SAVE REGISTERS
                                ;*****

0106 CDOC01      PAROUT: CALL      FIFRDY      ; GDP READY ?
0109 D330      OUT      P$GDP$PARAM
010B C9      RET

```

```

;***** FIFRDY *****
; GDP FIFO CHECK TILL READY (NOT FULL)
; SAVE REGISTERS
;*****

010C F5      FIFRDY: PUSH    PSW
010D DB30    FIF1:  IN      P$GDP$STATUS
              BIT      1,A
010F+CB4F    DB      0CBH,1*8+A+40H
              JRNZ     FIF1      ; NOT RDY !
0111+20FA    DB      20H,FIF1-$-1
0113 F1      POP      PSW
0114 C9      RET

;***** SELECT COLOUR *****
; COLOUR IN <A> BIT 0 - BIT 2
; CHANGES <A>
;*****

0115 F5      SELCOLR: PUSH PSW
              ; CHECK IF DRAWING IN PROGRESS

0116 DB30    DRRDY1: IN      P$GDP$STATUS
0118 E60C    ANI      00001100B      ; MASK FIFO EMPTY/DRAWING
              ; IN PROGRESS
011A FE04    CPI      04H            ; FIFO EMPTY ? DRAWING FINISHED ?
              JRNZ     DRRDY1        ; NO -> WAIT
011C+20F8    DB      20H,DRRDY1-$-1

011E F1E607  POP PSW! ANI 7          ;MASK BIT 0- BIT 2
0121 C5      PUSH     B
0122 47      MOV      B,A
0123 DB36    IN      P$GDP$CTRL      ;READ PORTC
0125 E6F8    ANI      0F8H          ;MASK ZOOM, DACK, A11-A12
0127 B0      ORA      B
0128 C1D336  POP B!  OUT P$GDP$CTRL ;SELECT COLOUR
012B C9      RET

;***** WRITE CURSOR *****
;SET CURSOR TO HL = EAD, A = DADDR
;*****

012C F5      CURS:  PUSH    PSW
012D 3E49    MVI      A,49H          ; CURS COMMAND
012F CD0001  CALL     CMDOUT
0132 7D      MOV      A,L            ; EAD-L
0133 CD0601  CALL     PAROUT
0136 7C      MOV      A,H            ; EAD-H
0137 CD0601  CALL     PAROUT
013A F1      POP      PSW
013B CD0601  CALL     PAROUT          ; DADDR
013E C9      RET

```

```
;***** INIT GDP - VRAM *****
; CREATE CCIR-NORM VIDEO TIMING
; CLEAR SCREEN , HOME CURSOR
;*****
```

INITVRAM:

```
013F 3E82      MVI      A,82H          ; PGM PIO: A=OUT,B=IN,C=OUT
0141 D337      OUT      P$GDP$MODE
0143 3E20      MVI      A,0010$0000B   ; /DACK=H, ZOOM=1X,
                                           ; ALL COLOURS SELECT

0145 D336      OUT      P$GDP$CTRL
0147 215001    LXI      H,RESET
014A CD9D01    CALL     PARA          ; RESET GDP
014D C37901    JMP      INIT1         ; CLEAR SCREEN
```

```
;*****
; COMMAND AND PARAMETER CONSTANT TABLES
;*****
```

```
0150 06      RESET:  DB      6          ; 6 COMMANDS
```

; RESET COMMAND

```
0151 0008      DB      0,8          ; FIRST COMMAND RESET
                                           ; 8 PARAMETER FOLLOWS
```

```
;DISPALY FIELD FOR 704 X 288 PIXEL
;PIXEL FREQUENCY 16 MHZ
;LINE FREQUENCY 15,7 KHZ
;FRAME FREQUENCY 50 HZ
```

```
0153 12      DB      0001$0010B      ; P1: GRAPHIC MODE,
                                           ; DRAWING DURING RETRACE
                                           ; BLANKING, NON-INTERLACE,
                                           ; NO REFRESH
0154 2A      DB      2AH              ; P2: ACTIVE DISPLAY WORDS
                                           ; PER LINE - 2, 44 WORDS
0155 A3      DB      101$00011B      ; P3: VSYNC - 1, HSYNC -1
0156 14      DB      000101$00B      ; P4: HFP - 1, VSYNCH H
0157 09      DB      09              ; P5: HORIZONTAL BACK PORCH - 1
0158 02      DB      02              ; P6: VERTICAL FRONT PORCH -1
0159 20      DB      20H             ; P7: ACTIVE DIPLAY LINES LOW
015A 45      DB      010001$01B      ; P8: VBP, AD LINES HIGH = 288 LINES
```

; VSYNCH COMMAND

```
015B 6F00      DB      6FH,0          ; 2ND COMMAND VSYNCH,
                                           ; MASTER MODE, NO PARAMETER
```

; CCHAR COMMAND

```
015D 4B03      DB      4BH,3          ; 3D COMMAND CCHAR,
                                           ; 3 PARAMETERS FOLLOWS
015F 00      DB      0              ; P1: MUST BE SET TO ZERO
0160 60      DB      01$1$00000B     ; P2: BR MAY NOT BE ZERO,
                                           ; STEADY CURSOR
0161 00      DB      0              ; P3: HAS NO MEANING
```

; PITCH COMMAND

```
0162 4701      DB      47H,1          ; 4TH COMMAND PITCH, 1 PARAMETER
```

```

0164 2C          DB      2CH          ; P1: NUMBER OF WORD
                                           ; ADRESSES HORIZONTAL
                                           ; ZOOM COMMAND

0165 460100      DB      46H,1,0      ; 5TH COMMAND ZOOM,
                                           ; 1 PARAMETER = NO ZOOM
                                           ; START DISPLAY

0168 6B00        DB      6BH,0        ; 6TH COMMAND START DISPLAY
                                           ; & END IDLE MODE, NO PARAMETER

;*****
; PRAMLOAD:
;*****

; LOAD PARAMETER RAM RA-0 - RA3 : DEFINE STARTING ADDRESS
; AND LENGTH OF DISPALY AREAS

016A 01          DB      1            ; 1 COMMAND
016B 70          DB      70H          ; COMMAND PRAM-LOAD STARTING
                                           ; WITH RA0
016C 04          DB      4            ; 4 PARAMETERS FOLLOW
016D 00000F3F    DB      0,0,0FH,03FH ; STARTING ADDRESS 0, LENGTH, GRAPHIC

;*****
; HIGH SPEED:
;*****

; SET UP GDP TO CLEAR AREA IN HIGH SPEED MODE BY
; MANIPULATING ONE WORD (16 BITS) PER CYCLE -> MASK = 0FFFFH

0171 02          DB      2            ; 2 COMMANDS FOLLOWS

0172 4A02        DB      4AH,2        ; SET MASK REGISTER , 2 PARAMETERS
0174 FFFF        DB      0FFH,0FFH    ; ALL MASK BITS SET TO ONE

0176 4C0102      DB      4CH,1,2      ; FIGS COMMAND, 1 PARAMETER,
                                           ; SET DRAW DIR 02

;*****
; CLEAR SCREEN , FORM FEED
;*****

0179 216A01      INIT1: LXI H,PRAMLOAD
017C CD9D01      CALL PARA
017F 21FF3F      LXI      H,3FFFH      ; HL = MAX. EAD.-ADDRESS

;***** CLEAR AREA *****
;LOESCHT AREA DER WORTGROESSE HL+1
;*****

CLRAREA:

0182 E5          PUSH     H
0183 217101      LXI      H,HIGHSPEED
0186 CD9D01      CALL     PARA
0189 E1          POP      H
018A 7D          MOV      A,L          ; PARAMETERS FOLLOWS
                                           ; THE FIGS COMMAND
                                           ; P2: LENGTH L

018B CD0601      CALL     PAROUT
018E 7C          MOV      A,H

```

```

018F CD0601      CALL    PAROUT          ; P3: LENGTH H
0192 3E32        MVI     A,32H          ; WDAT COMMAND RESET
0194 CD0001      CALL    CMDOUT         ; WDAT
0197 3EFF        MVI     A,0FFH        ; PARAMETER
0199 CD0601      CALL    PAROUT
019C C9          RET

```

```

;***** PARA-ROUTINE *****
; SEND COMMAND GROUPS AND ASSOCIATED PARAMETERS FROM TABLE
; ADDRESSED BY <HL> TO GDP
; CHANGES <HL> , <B> = 0
;*****

```

```

019D 46          PARA:  MOV     B,M      ; FIRST BYTE NUMBER OF COMMANDS
019E C5          NCMD   PUSH     B        ; SAVE IT
019F 237E        INX H! MOV A,M      ; FIRST COMMAND
01A1 CD0001      CALL    CMDOUT
01A4 237E        INX H! MOV A,M      ; NUMBER OF PARAMETERS FOLLOWS
01A6 B7          ORA     A
01A7+2808        JRZ     NOPA        ; TEST FOR ZERO PARAMETER
01A9 47          DB      28H,NOPA-$-1
01AA 237E        MOV     B,A        ; B=PARAMETER COUNTER
01AC CD0601      PAR    INX H! MOV A,M
01AF+10F9        CALL    PAROUT
01B1 C1          DJNZ    PAR        ; ANY PARAMETERS REMAIN ?
01B2+10EA        DB      10H,PAR-$-1
01B4 C9          NOPA  POP     B
01B5 D334        DJNZ    NCMD        ; MOVE TO NEXT COMMAND
01B6 C9          DB      10H,NCMD-$-1
01B7 C9          RET

```

```

;***** CHARACTER OUT *****
; WRITE ASCCI CHARACTER IN <A> INTO VIDEO RAM
; READ CHARACTER EPROM, MATRIX 8 X 12 PIXELS
;*****

```

```

01B5 D334        CHAR:  OUT     P$GDP$ADR ; ADRESS EPROM VIA PORTC
01B7 F5          PUSH    PSW           ; WITH ASCII CHARACTER
01B8 3E20        MVI     A,20H        ; SAVE IT
01BA CD0001      CALL    CMDOUT        ; WDAT COMMAND REPLACE
01BB C9          ; SET WDAT

; WRITE FIRST 8 ROWS INTO GDP, STARTING WITH ROW 0

01BD 0608        MVI     B,8          ; FIRST 8 ROWS TO WRITE
01BF 0E38        MVI     C,P$GDP$CHAR ; C= PORT ADDRESS CHARRD (EPROM
01C0 C9          ; PORT LOW)
01C1 CDD201      CALL    LOADP         ; 8 PARAMETER INTO GDP

;NEXT 4 ROWS FOLLOWS -> HAS TO SEND NEW ADDRESS TO EPROM

01C4 F1          POP     PSW           ; ASCII IN A
01C5 0F          RRC                 ; TEST IF ASCCI EVON OR ODD
01C6 CBFF        SETB    7,A          ; -> DIVIDE BY 2 -> BIT 0 IN CARRY
01C7 C9          ; SET BIT 7 OF ASCII TO
01C8 0E38        SETB    7,A          ; SELECT NEW ADDRESS
01C9 C9          DB      0CBH,7*8+A+0COH
01CA C9          MVI     C,P$GDP$CHAR
01CB C9          JRNC    EVEN         ; NO CARRY -> ASCII EVEN

```

```

                                ; -> READ 4 LOW BYTES OF EPROM
01CA+3002                      DB      30H,EVEN-$-1
01CC 0E3C                      MVI     C,P$GDP$CHAR+4 ; CARRY -> READ 4 HIGH BYTES

01CE D334                      EVEN:   OUT      P$GDP$ADR ; ADDRESS EPROM WITH ASCII-H
01D0 0604                      MVI     B,04 ; WRITE NEXT 4 ROWS

;***** LOADPARAMETER *****
;B = ROW COUNTER, C = PORTADDRESS TO READ CHARCTERS
;*****

LOADP:
01D2 3E80                      MVI     A,80H
01D4 90                        SUB      B ; REGISTER B GIVES STARTING
                                ; ADDRESS TO LOAD PARAM.
01D5 CD0001                    CALL    CMDOUT ; PARAM. LOAD AND STARTING ADDRESS
01D8 58                        MOV      E,B
01D9 1D                        DCR      E ; DC-PARAMETER 4 OR 8 ROWS -1
                                ; FOR FUTURE USE

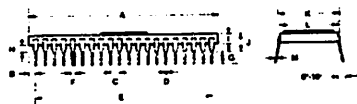
CHAOUT1:INP                    A ; READ CHARACTER FROM EPROM
01DA+ED78                      DB      0EDH,A*8+40H
01DC CD0601                    CALL    PAROUT ; WRITE IT INTO PRAM
01DF 0C                        INR      C ; NEXT ROW -> INC EPROM ADR IN C
                                ; ANY ROWS REMAINS ?
01E0+10F8                      DB      10H,CHAOUT1-$-1

01E2 3E4C                      MVI     A,4CH ; FIGS COMMAND
01E4 CD0001                    CALL    CMDOUT
01E7 3E12                      MVI     A,12H ; DIR 2 , AREA FILL
01E9 CD0601                    CALL    PAROUT
01EC 7B                        MOV      A,E ; DC-L, HIGHT OF CHARACTER
                                ; 8 OR 4 ROWS -1
01ED CD0601                    CALL    PAROUT ; NO NEED TO GIVE DC-H AND D
01F0 3E68                      MVI     A,68H
01F2 CD0001                    CALL    CMDOUT ; GCHRD COMMAND -> DRAW CHARACTER
                                ; INTO VIDEO RAM
01F5 C9                        RET
                                END

```

μPD7220/GDC
μPD7220-1/μPD7220-2
GRAPHICS DISPLAY
CONTROLLER

Package Outline
μPD7220D (Ceramic)



Ceramic		
Item	Millimeters	Inches
A	51.5 Max	2.03 Max
B	1.62 Min	0.06 Min
C	2.54 ± 0.1	0.1 ± 0.004
D	0.5 ± 0.1	0.02 ± 0.004
E	40.25 ± 0.1	1.59 ± 0.004
F	1.62 Min	0.06 Min
G	2.2 Min	0.12 Min
H	1.0 Min	0.04 Min
I	3.5 Min	0.14 Min
J	4.5 Min	0.18 Min
K	15.24 Typ	0.6 Typ
L	14.93 Typ	0.59 Typ
M	0.25 ± 0.05	0.01 ± 0.0019

CORRECTIONS:

AC Characteristics for 7220, 7220-1, 7220-2

Read Cycle:
 RD Pulse width t_{RD}
 add $t_{DCE} = \frac{1}{2} t_{CLK}$ to t_{DR} (max.)
Write Cycle:
 WR Pulse width t_{WR}
 add $3 t_{CLK}$ to t_{WR} (max.)
DMA Cycle:
 add DAC Low Level width t_{DAL}
 + $2 t_{CLK}$ (min.)
R/W Cycle:
 ALE width t_{AU}
 change t_{AU} (min.) to $1/3 t_{CLK}$
 add Addr. Setup to RAS t_{ASL}
 + 30 ns (min.)

AC Characteristics for 7220-2

MA Cycle:
 change t_{DCE} delay from $t_{DCE} + t_{SIG}$
 to 110 ns (max.)
 change t_{DCE} delay from t_{DCE}
 t_{RD} (R) and t_{RD} (W)
 to $t_{CLK} + 110$ ns (max.)
R/W Cycle:
 change Address Data Delay from
 $2 t_{MCLK}$ t_{AD} to 115 ns (max.)
 change Address Data Floating from
 $2 t_{MCLK}$ t_{OFF} to 115 ns (max.)
AC Characteristics for 7220, 7220-1
Write Cycle:
 change Address Hold from WR t_{AH}
 to 10 ns (min.)
 change Data Hold from WR t_{DH}
 to 10 ns (min.)

Description

The μPD7220 Graphics Display Controller (GDC) is an intelligent microprocessor peripheral designed to be the heart of a high-performance raster-scan computer graphics and character display system. Positioned between the video display memory and the microprocessor bus, the GDC performs the tasks needed to generate the raster display and manage the display memory. Processor software overhead is minimized by the GDC's sophisticated instruction set, graphics figure drawing, and DMA transfer capabilities. The display memory supported by the GDC can be configured in any number of formats and sizes up to 256K 16-bit words. The display can be zoomed and panned, while partitioned screen areas can be independently scrolled. With its light pen input and multiple controller capability, the GDC is ideal for advanced computer graphics applications.

For a more detailed description of the GDC's operation, please refer to the GDC Design Manual.

System Considerations

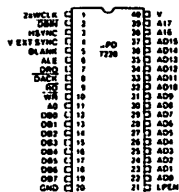
The GDC is designed to work with a general purpose microprocessor to implement a high-performance computer graphics system. Through the division of labor established by the GDC's design, each of the system components is used to the maximum extent through six-level hierarchy of simultaneous tasks. At the lowest level, the GDC generates the basic video raster timing, including sync and blanking signals. Partitioned areas on the screen and zooming are also accomplished at this level. At the next level, video display memory is modified during the figure drawing operations and data moves. Third, display memory addresses are calculated pixel by pixel as drawing progresses. Outside the GDC at the next level, preliminary calculations are done to prepare drawing parameters. At the fifth level, the picture must be represented as a list of graphics figures drawable by the GDC. Finally, this representation must be manipulated, stored, and communicated. By handling the first three levels, the GDC takes care of the high-speed and repetitive tasks required to implement a graphics system.

Features

- Microprocessor Interface
 - DMA transfers with 8257- or 8237-type controllers
 - FIFO Command Buffering
- Display Memory Interface
 - Up to 256K words of 16 bits
 - Read-Modify-Write (RMW) Display Memory cycles in under 800ns
 - Dynamic RAM refresh cycles for nonaccessed memory
- Light Pen Input
- External video synchronization mode
- Graphics Mode
 - Four megabit, bit-mapped display memory
 - Character Mode
 - 8K character code and attributes display memory
 - Mixed Graphics and Character Mode
 - 64K if all characters
 - 1 megapixel if all graphics
- Graphics Capabilities
 - Figure drawing of lines, arcs/circles, rectangles, and graphics characters in 800ns per pixel
 - Display 1024-by-1024 pixels with 4 planes of color or grayscale
 - Two independently scrollable areas
- Character Capabilities
 - Auto cursor advance
 - Four independently scrollable areas
 - Programmable cursor height
 - Characters per row: up to 256
 - Character rows per screen: up to 100
- Video Display Format
 - Zoom magnification factors of 1 to 16
 - Panning
 - Command-settable video raster parameters
- Technology
 - Single +5 volt, NMOS, 40-pin DIP
- DMA Capability
 - Bytes or word transfers
 - 4 clock periods per byte transferred

NEC cannot assume any responsibility for any circuits shown or represent that they are free from patent infringement.
 NEC reserves the right to make changes any time without notice in order to improve design and supply the best product possible.

Pin Configuration



Pin Identification

No.	Pin	Symbol	Direction	Function
1	ZWCLK	IN		Clock Input
2	DREQ	OUT		Display Memory Read Input Flag
3	HSYNC	OUT		Horizontal Video Sync Output
4	V-EST SYNC	IN/OUT		Vertical Video Sync Output or External VSYNC Input
5	BLANK	OUT		CRT Blanking Output
6	ALE (RST)	OUT		Address Latch Enable Output
7	DREQ	OUT		DMA Request Output
8	DREQ	IN		DMA Acknowledge Input
9	RG	IN		Read Strobe Input for Microprocessor Interface
10	WR	IN		Write Strobe Input for Microprocessor Interface
11	AD	IN		Address Select Input for Microprocessor Interface
12-18	DQS to F	IN/OUT		Bidirectional Data Bus to Host Microprocessor
19	GND			Ground
21	LPEH	IN		Light Pen Detect Input
23-24	AD16 to 17	IN/OUT		Address and Data Lines to Display Memory
25-27	AD13 to 15	IN/OUT		Utilization Varies with Mode of Operation
28	A16	OUT		Utilization Varies with Mode of Operation
29	A17	OUT		Utilization Varies with Mode of Operation
30	V			+5V - 10%

Character Mode Pin Utilization

No.	Pin	Symbol	Direction	Function
25-27	AD13 to 15	IN/OUT		Line Counter Bits 0 to 2 Outputs
28	A16	OUT		Line Counter Bit 3 Output
29	A17	OUT		Cursor Output and Line Counter Bit 4

Mixed Mode Pin Utilization

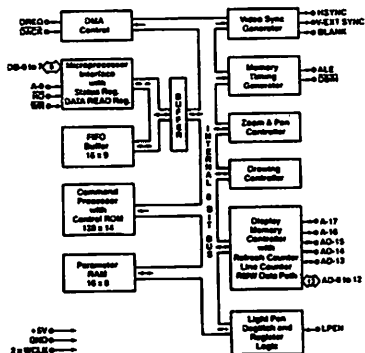
No.	Pin	Symbol	Direction	Function
25-27	AD13 to 15	IN/OUT		Address and Data Bits 13 to 15
28	A16	OUT		Attribute Blank and Clear Line Counter Output
29	A17	OUT		Cursor and 80-Step Area Flag Output

*Output 10 clock cycles after trailing edge of HSYNC. See figure for timing example.

Graphics Mode Pin Utilization

No.	Pin	Symbol	Direction	Function
25-27	AD13 to 15	IN/OUT		Address and Data Bits 13 to 15
28	A16	OUT		Address Bit 16 Output
29	A17	OUT		Address Bit 17 Output

Block Diagram



GDC Components

Microprocessor Bus Interface

Control of the GDC by the system microprocessor is achieved through an 8-bit bidirectional interface. The status register is readable at any time. Access to the FIFO buffer is coordinated through flags in the status register and operates independently of the various internal GDC operations, due to the separate data bus connecting the interface and the FIFO buffer.

Command Processor

The contents of the FIFO are interpreted by the command processor. The command bytes are decoded, and the succeeding parameters are distributed to their proper destinations within the GDC. The command processor yields to the bus interface when both access the FIFO simultaneously.

DMA Control

The DMA control circuitry in the GDC coordinates transfers over the microprocessor interface when using an external DMA controller. The DMA Request and Acknowledge handshake lines directly interface with a μ PDB257 or μ PDB237 DMA controller, so that display data can be moved between the microprocessor memory and the display memory.

Parameter RAM

The 16-byte RAM stores parameters that are used repetitively during the display and drawing processes. In character mode, this RAM holds four sets of partitioned display area parameters; in graphics mode, the drawing pattern and graphics character take the place of two of the sets of parameters.

Video Sync Generator

Based on the clock input, the sync logic generates the raster timing signals for almost any interlaced, non-interlaced, or "repeat field" interlaced video format. The generator is programmed during the idle period following a reset. In video sync slave mode, it coordinates timing between multiple GDCs.

Memory Timing Generator

The memory timing circuitry provides two memory cycle types: a two-clock period refresh cycle and the read-modify-write (RMW) cycle which takes four clock periods. The memory control signals needed to drive the display memory devices are easily generated from the GDC's ALE and DQS inputs.

Zoom & Pan Controller

Based on the programmable zoom display factor and the display area entries in the parameter RAM, the zoom and pan controller determines when to advance to the next memory address for display refresh and when to go on to the next display area. A horizontal zoom is produced by slowing down the display refresh rate while maintaining the video sync rates. Vertical zoom is accomplished by repeatedly accessing each line a number of times equal to the horizontal repeat. Once the line count for a display area is exhausted, the controller accesses the starting address and line count of the next display area from the parameter RAM. The system microprocessor, by modifying a display

area starting address, can pan in any direction, independently of the other display areas.

Drawing Controller

The drawing processor contains the logic necessary to calculate the addresses and positions of the pixels of the various graphics figures. Given a starting point and the appropriate drawing parameters, the drawing controller needs no further assistance to complete the figure drawing.

Display Memory Controller

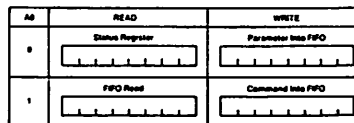
The display memory controller's tasks are numerous. Its primary purpose is to multiplex the address and data information in and out of the display memory. It also contains the 16-bit logic unit used to modify the display memory contents during RMW cycles, the character mode line counter, and the refresh counter for dynamic RAMs. The memory controller apportions the video field time between the various types of cycles.

Light Pen Deglitcher

Only if two rising edges on the light pen input occur at the same point during successive video fields are the pulses accepted as a valid light pen detection. A status bit indicates to the system microprocessor that the light pen register contains a valid address.

Programmer's View of GDC

The GDC occupies two addresses on the system microprocessor bus through which the GDC's status register and FIFO are accessed. Commands and parameters are written into the GDC's FIFO and are differentiated based on address bit A0. The status register or the FIFO can be read as selected by the address line.



GDC Microprocessor Bus Interface Registers

Commands to the GDC take the form of a command byte followed by a series of parameter bytes as needed for specifying the details of the command. The command processor decodes the commands, unpacks the parameters, loads them into the appropriate registers within the GDC, and initiates the required operations.

The commands available in the GDC can be organized into five categories as described in the following section.

GDC Command Summary**Video Control Commands**

1. RESET Resets the GDC to its idle state.
2. SYNC Specifies the video display format.
3. VSYNC Selects master or slave video synchronization mode.
4. CCHAR Specifies the cursor and character row heights.

Display Control Commands

1. START Ends idle mode and unblanks the display.
2. BCTRL Controls the blanking and unblanking of the display.
3. ZOOM Specifies zoom factors for the display and graphics characters writing.
4. CURS Sets the position of the cursor in display memory.
5. PRAM Defines starting addresses and lengths of the display areas and specifies the eight bytes for the graphics character.
6. PITCH Specifies the width of the X dimension of display memory.

Drawing Control Commands

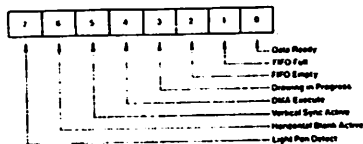
1. WDAT Writes data words or bytes into display memory.
2. MASK Sets the mask register contents.
3. FIGS Specifies the parameters for the drawing controller.
4. FIGD Draws the figure as specified above.
5. GCHRD Draws the graphics character into display memory.

Data Read Commands

1. RDAT Reads data words or bytes from display memory.
2. CURD Reads the cursor position.
3. LPRD Reads the light pen address.

DMA Control Commands

1. DMAR Requests a DMA read transfer.
2. DMAW Requests a DMA write transfer.

Status Register Flags**Status Register (SR)****SR-7: Light Pen Detect**

When this bit is set to 1, the light pen address (LAD) register contains a digitized value that the system microprocessor may read. This flag is reset after the 3-byte LAD is moved into the FIFO in response to the light pen read command.

SR-6: Horizontal Blanking Active

A 1 value for this flag signifies that horizontal retrace blanking is currently underway.

SR-5: Vertical Sync

Vertical retrace sync occurs while this flag is a 1. The vertical sync flag coordinates display format modifying commands to the blanked interval surrounding vertical sync. This eliminates display disturbances.

SR-4: DMA Execute

This bit is a 1 during DMA data transfers.

SR-3: Drawing in Progress

While the GDC is drawing a graphics figure, this status bit is a 1.

SR-2: FIFO Empty

This bit and the FIFO Full flag coordinate system microprocessor accesses with the GDC FIFO. When it is 1, the Empty flag ensures that all the commands and parameters previously sent to the GDC have been interpreted.

SR-1: FIFO Full

A 1 at this flag indicates a full FIFO in the GDC. A 0 ensures that there is room for at least one byte. This flag needs to be checked before each write into the GDC.

SR-0: Data Ready

When this flag is a 1, it indicates that a byte is available to be read by the system microprocessor. This bit must be tested before each read operation. It drops to a 0 while the data is transferred from the FIFO into the microprocessor interface data register.

FIFO Operation & Command Protocol

The first-in, first-out buffer (FIFO) in the GDC handles the command dialogue with the system microprocessor. This flow of information uses a half-duplex technique, in which the single 16-location FIFO is used for both directions of data movement, one direction at a time. The FIFO's direction is controlled by the system microprocessor through the GDC's command set. The host microprocessor coordinates these transfers by checking the appropriate status register bits.

The command protocol used by the GDC requires differentiation of the first byte of a command sequence from the succeeding bytes. The first byte contains the operation code and the remaining bytes carry parameters. Writing into the GDC causes the FIFO to store a flag value alongside the data byte to signify whether the byte was written into the command or the parameter address. The command processor in the GDC tests this bit as it interprets the entries in the FIFO.

The receipt of a command byte by the command processor marks the end of any previous operation. The number of parameter bytes supplied with a command is cut short by the receipt of the next command byte. A read operation from the GDC to the microprocessor can be terminated at any time by the next command.

The FIFO changes direction under the control of the system microprocessor. Commands written into the GDC always put the FIFO into write mode if it wasn't in it already.

If it was in read mode, any read data in the FIFO at the time of the turnaround is lost. Commands which require a GDC response, such as RDAT, CURD and LPRD, put the FIFO into read mode after the command is interpreted by the GDC's command processor. Any commands and parameters behind the read-evoking command are discarded when the FIFO direction is reversed.

Read-Modify-Write Cycle

Data transfers between the GDC and the display memory are accomplished using a read-modify-write (RMW) memory cycle. The four clock period timing of the RMW cycle is used to: 1) output the address, 2) read data from the memory, 3) modify the data, and 4) write the modified data back into the initially selected memory address. This type of memory cycle is used for all interactions with display memory including DMA transfers, except for the two clock period display and RAM refresh cycles.

The operations performed during the modify portion of the RMW cycle merit additional explanation. The circuitry in the GDC uses three main elements: the Pattern register, the Mask register, and the 16-bit Logic Unit. The Pattern register holds the data pattern to be moved into memory. It is loaded by the WDAT parameters or, during drawing, from the parameter RAM. The Mask register contents determine which bits of the read data will be modified. Based on the contents of these registers, the Logic Unit performs the selected operations of REPLACE, COMPLEMENT, SET, or CLEAR on the data read from display memory.

The Pattern register contents are ANDed with the Mask register contents to enable the actual modification of the memory read data, on a bit-by-bit basis. For graphics drawing, one bit at a time from the Pattern register is combined with the Mask. When ANDed with the bit set to a 1 in the Mask register, the proper single pixel is modified by the Logic Unit. For the next pixel in the figure, the next bit in the Pattern register is selected and the Mask register bit is moved to identify the pixel's location within the word. The Execution word address pointer register, EAD, is also adjusted as required to address the word containing the next pixel.

In character mode, all of the bits in the Pattern register are used in parallel to form the respective bits of the modify data word. Since the bits of the character code word are used in parallel, unlike the one-bit-at-a-time graphics drawing process, this facility allows any or all of the bits in a memory word to be modified in one RMW memory cycle. The Mask register must be loaded with 1s in the positions where modification is to be permitted.

The Mask register can be loaded in either of two ways. In graphics mode, the CURS command contains a four-bit dAD field to specify the dot address. The command processor converts this parameter into the one-of-16 format used in the Mask register for figure drawing. A full 16 bits can be loaded into the Mask register using the MASK command. In addition to the character mode use mentioned above, the 16-bit MASK load is convenient in graphics mode when all of the pixels of a word are to be set to the same value.

The Logic Unit combines the data read from display memory, the Pattern Register, and the Mask register to generate the data to be written back into display memory. Any one of four operations can be selected: REPLACE,

COMPLEMENT, CLEAR or SET. In each case, if the respective Mask bit is 0, that particular bit of the read data is returned to memory unmodified. If the Mask bit is 1, the modification is enabled. With the REPLACE operation, the Pattern Register data simply takes the place of the read data for modification enabled bits. For the other three operations, a 0 in the modify data allows the read data bit to be returned to memory. A 1 value causes the specified operation to be performed in the bit positions with set Mask bits.

Figure Drawing

The GDC draws graphics figures at the rate of one pixel per read-modify-write (RMW) display memory cycle. These cycles take four clock periods to complete. At a clock frequency of 5MHz, this is equal to 800ns. During the RMW cycle the GDC simultaneously calculates the address and position of the next pixel to be drawn.

The graphics figure drawing process depends on the display memory addressing structure. Groups of 16 horizontally adjacent pixels form the 16-bit words which are handled by the GDC. Display memory is organized as a linearly addressed space of these words. Addressing of individual pixels is handled by the GDC's internal RMW logic.

During the drawing process, the GDC finds the next pixel of the figure which is one of the eight nearest neighbors of the last pixel drawn. The GDC assigns each of these eight directions a number from 0 to 7, starting with straight down and proceeding counterclockwise.

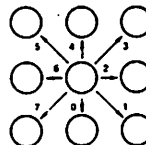
**Drawing Directions**

Figure drawing requires the proper manipulation of the address and the pixel bit position according to the drawing direction to determine the next pixel of the figure. To move to the word above or below the current one, it is necessary to subtract or add the number of words per line in display memory. This parameter is called the pitch. To move to the word to either side, the Execute word address cursor, EAD, must be incremented or decremented as the dot address pointer bit reaches the LSB or the MSB of the Mask register. To move to a pixel within the same word, it is necessary to rotate the dot address pointer register to the right or left.

The table below summarizes these operations for each direction.

Dir	Operations to Address the Next Pixel
000	EAD ← P ← EAD
001	EAD ← P ← EAD
010	EAD(15:8) ← 1 EAD + 1 ← EAD EAD ← LR EAD(7:0) ← 1 EAD + 1 ← EAD EAD ← LR
011	EAD ← P ← EAD EAD(15:8) ← 1 EAD + 1 ← EAD EAD ← LR EAD ← P ← EAD
100	EAD ← P ← EAD EAD(15:8) ← 1 EAD - 1 ← EAD EAD ← RR EAD(7:0) ← 1 EAD - 1 ← EAD EAD ← RR
101	EAD ← P ← EAD EAD(15:8) ← 1 EAD - 1 ← EAD EAD ← RR EAD ← P ← EAD
110	EAD ← P ← EAD EAD(15:8) ← 1 EAD - 1 ← EAD EAD ← RR
111	EAD ← P ← EAD EAD(15:8) ← 1 EAD - 1 ← EAD EAD ← RR

Where P: Pattern LR: Left Register RR: Right Register
EAD: E-Address Word Address, and
EAD: Dot Address stored in the Mask Register

Whole word drawing is useful for filling areas in memory with a single value. By setting the Mask register to all 1s with the MASK command, both the LSB and MSB of the dAD will always be 1, so that the EAD value will be incremented or decremented for each cycle regardless of direction. One RMW cycle will be able to effect all 16 bits of the word for any drawing type. One bit in the Pattern register is used per RMW cycle to write all the bits of the word to the same value. The next Pattern bit is used for the word, etc.

For the various figures, the effect of the initial direction upon the resulting drawing is shown below:

Dir	Line	Arc	Character	Slant Char	Rectangle	DMA
000						
001						
010						
011						
100						
101						
110						
111						

Note that during line drawing, the angle of the line may be anywhere within the shaded octant defined by the DIR value. Arc drawing starts in the direction initially specified by the DIR value and veers into an arc as drawing proceeds. An arc may be up to 45 degrees in length. DMA transfers are done on word boundaries only, and follow the arrows indicated in the table to find successive word addresses. The slanted paths for DMA transfers indicate the GDC changing both the X and Y components of the word address when moving to the next word. It does not follow a 45 degree diagonal path by pixels.

Drawing Parameters

In preparation for graphics figure drawing, the GDC's Drawing Processor needs the figure type, direction and drawing parameters, the starting pixel address, and the pattern from the microprocessor. Once these are in place within the GDC, the Figure Draw command, FIGD, initiates the drawing operation. From that point on, the system microprocessor is not involved in the drawing process. The GDC Drawing Controller coordinates the RMW circuitry and address registers to draw the specified figure pixel by pixel.

The algorithms used by the processor for figure drawing are designed to optimize its drawing speed. To this end, the specific details about the figure to be drawn are reduced by the microprocessor to a form conducive to high-speed address calculations within the GDC. In this way the repetitive, pixel-by-pixel calculations can be done quickly, thereby minimizing the overall figure drawing time. The table below summarizes the parameters.

Drawing Type	DC	B	B2	B1	B0
Initial Value	0	0	0	-1	-1
Line	1:101	2:101	3:101	4:101	5:101
Arc	1:101	2:101	3:101	4:101	5:101
Rectangle	3	A-1	B-1	-1	A-1
Area Fill	B-1	A	A	-1	-1
Graphic Character	B-1	A	A	-1	-1
Write Data	W-1	-1	-1	-1	-1
Drawn	D-1	C-1	-1	-1	-1
DMA	D-1	C-2	0	-1	-1
Read Data	W	-1	-1	-1	-1

*Values shown for the various parameters remain as each drawing process ends.
**Circles are drawn with 8 arcs, each of which span 45°, so that an arc of 1, 1.2 and an arc of 0.
***Graphic characters are a special case of bit-map area filling in which B and A = 0. B & A = 0 means no need to read D and B2.

Where:

1 of ONE 5 value

All numbers are shown in base 10 for convenience. The GDC accepts base 2 numbers (2's complement) when more appropriate.

The parameter bytes sent to GDC for this parameter

1) The larger of 2n or 3n

2) The smaller of 2n or 3n

3) Radius of curvature in pixels

4) Angle from major axis to end of the arc, A = 45°

5) Angle from major axis to start of the arc, A = 45°

6) Round up to the next higher integer

7) Round down to the next lower integer

8) Number of pixels in the initial specified direction

9) Number of pixels in the direction at right angles to the initial specified direction

10) Number of words to be addressed

11) Number of pixels to be addressed in the initially specified direction

12) Number of pixels to be addressed in the initially specified direction

13) Number of pixels to be addressed in the initially specified direction

14) Number of pixels to be addressed in the initially specified direction

15) Number of pixels to be addressed in the initially specified direction

16) Number of pixels to be addressed in the initially specified direction

17) Number of pixels to be addressed in the initially specified direction

18) Number of pixels to be addressed in the initially specified direction

19) Number of pixels to be addressed in the initially specified direction

20) Number of pixels to be addressed in the initially specified direction

21) Number of pixels to be addressed in the initially specified direction

22) Number of pixels to be addressed in the initially specified direction

23) Number of pixels to be addressed in the initially specified direction

24) Number of pixels to be addressed in the initially specified direction

25) Number of pixels to be addressed in the initially specified direction

26) Number of pixels to be addressed in the initially specified direction

27) Number of pixels to be addressed in the initially specified direction

28) Number of pixels to be addressed in the initially specified direction

29) Number of pixels to be addressed in the initially specified direction

30) Number of pixels to be addressed in the initially specified direction

31) Number of pixels to be addressed in the initially specified direction

32) Number of pixels to be addressed in the initially specified direction

33) Number of pixels to be addressed in the initially specified direction

34) Number of pixels to be addressed in the initially specified direction

35) Number of pixels to be addressed in the initially specified direction

36) Number of pixels to be addressed in the initially specified direction

37) Number of pixels to be addressed in the initially specified direction

38) Number of pixels to be addressed in the initially specified direction

39) Number of pixels to be addressed in the initially specified direction

40) Number of pixels to be addressed in the initially specified direction

Graphics Character Drawing

Graphics characters can be drawn into display memory pixel-by-pixel. The up to 8-by-8 character display is loaded into the GDC's parameter RAM by the system microprocessor. Consequently, there are no limitations on the character set used. By varying the drawing parameters and drawing direction, numerous drawing options are available. In area fill applications, a character can be written into display memory as many times as desired without reloading the parameter RAM.

Once the parameter RAM has been loaded with up to eight graphics character bytes by the appropriate PRAM command, the GCHRD command can be used to draw the bytes into display memory starting at the cursor. The zoom magnification factor for writing, set by the zoom command, controls the size of the character written into the display memory in integer multiples of 1 through 16. The bit values in the PRAM are repeated horizontally and vertically the number of times specified by the zoom factor.

The movement of these PRAM bytes to the display memory is controlled by the parameters of the FIGS command. Based on the specified height and width of the area to be drawn, the parameter RAM is scanned to fill the required area.

For an 8-by-8 graphics character, the first pixel drawn uses the LSB of RA-15, the second pixel uses bit 1 of RA-15, and so on, until the MSB of RA-15 is reached.

The GDC jumps to the corresponding bit in RA-14 to continue the drawing. The progression then advances toward the LSB of RA-14. This snaking sequence is continued for the other 6 PRAM bytes. This progression matches the sequence of display memory addresses calculated by the drawing processor as shown above. If the area is narrower than 8 pixels wide, the snaking will advance to the next PRAM byte before the MSB is reached. If the area is less than 8 lines high, fewer bytes in the parameter RAM will be scanned. If the area is larger than 8 by 8, the GDC will repeat the contents of the parameter RAM in two dimensions, as required to fill the area with the 8-by-8 mosaic. (Fractions of the 8-by-8 pattern will be used to fill areas which are not multiples of 8 by 8.)

Parameter RAM Contents: RAM Address RA 0 to 15

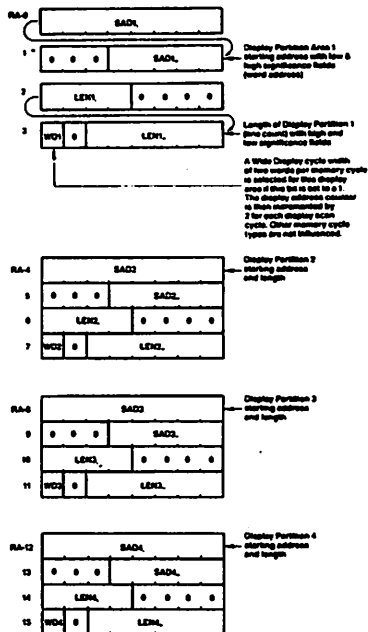
The parameters stored in the parameter RAM, PRAM, are available for the GDC to refer to repeatedly during figure drawing and raster-scanning. In each mode of operation the values in the PRAM are interpreted by the GDC in a predetermined fashion. The host microprocessor must load the appropriate parameters into the proper PRAM locations. PRAM loading command allows the host to write into any location of the PRAM and transfer as many bytes as desired. In this way any stored parameter byte or bytes may be changed without influencing the other bytes.

The PRAM stores two types of information. For specifying the details of the display area partitions, blocks of four bytes are used. The four parameters stored in each block include the starting address in display memory of each display area, and its length. In addition, there are two mode bits for each area which specify whether the area is a bit-mapped graphics area or a coded character area, and whether a 16-bit or a 32-bit wide display cycle is to be used for that area.

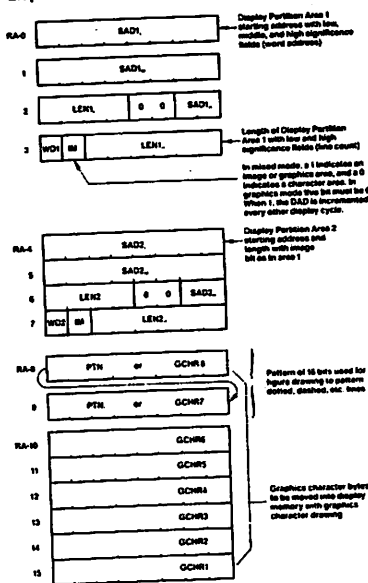
The other use for the PRAM contents is to supply the pattern for figure drawing when in a bit-mapped graphics area or mode. In these situations, PRAM bytes 8 through 16 are reserved for this patterning information. For line, arc, and rectangle drawing (linear figures) locations 8 and 9 are loaded into the Pattern Register to allow the GDC to draw dotted, dashed, etc. lines. For area filling and graphics bit-mapped character drawing locations 8 through 15 are referenced for the pattern or character to be drawn.

Details of the bit assignments are shown for the various modes of operation.

Character Mode



Graphics and Mixed Graphics and Character Modes

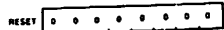


Command Bytes Summary

RESET	0 0 0 0 0 0 0 0
SYNC	0 0 0 0 1 1 1 DE
VSYNC	0 1 1 0 1 1 1 M
CHAR	0 1 0 0 1 0 1 1
START	0 1 1 0 1 0 1 1
BCTRL	0 0 0 0 1 1 0 DE
ZOOM	0 1 0 0 0 1 1 0
CURS	0 1 0 0 1 0 0 1
PRAM	0 1 1 1 1 SA
PITCH	0 1 0 0 0 1 1 1
WOAT	0 0 1 TYPE 0 MOD
MASK	0 1 0 0 1 0 1 0
FIGS	0 1 0 0 1 1 0 0
FIGD	0 1 1 0 1 1 0 0
GCHWD	0 1 1 0 1 0 0 0
ROAT	1 0 1 TYPE 0 MOD
CURD	1 1 1 0 0 0 0 0
LPRD	1 1 0 0 0 0 0 0
DMAH	1 0 1 TYPE 1 MOD
DMAV	0 0 1 TYPE 1 MOD

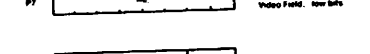
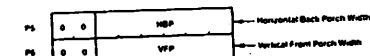
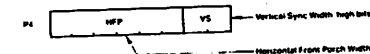
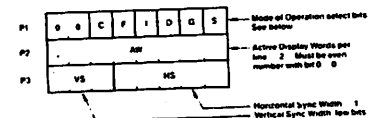
Video Control Commands

Reset



Blank the display, enter idle mode, and enable within the GDC.
 -- FIFO
 -- Command Processor
 -- Interval Counters

This command can be executed at any time and does not modify any of the parameters already loaded into the GDC. If followed by parameter bytes, this command also sets the sync generator parameters as described below. Idle mode is exited with the START command.



In graphics mode, a word is a group of 16 pixels. In character mode, a word is one character code and its attributes, if any. The number of active words per line must be an even number from 2 to 256. An all-zero parameter value selects a count equal to 2ⁿ where n = number of bits in the parameter field for vertical parameters. All horizontal widths are counted in display words. All vertical intervals are counted in lines.

Horizontal Back Porch Constraints

- In general:
HBP > 3 Display Word Cycles (6 clock cycles).
- If the (IMAGE or WD modes change within one video field:
HBP > 5 Display Word Cycles (10 clock cycles).
- If interlace or mixed mode is used:
HBP > 5 Display Word Cycles (10 clock cycles).

Horizontal Front Porch Constraints

- If the display ZOOM function is used at other than 1X:
HFP > 2 Display Word Cycles (4 clock cycles).
- If the GDC is used in the video sync Slave mode:
HFP > 4 Display Word Cycles (8 clock cycles).
- If the Light Pen is used:
HFP > 6 Display Word Cycles (12 clock cycles).
- If interlace mode is used:
HFP > 3 Display Word Cycles (6 clock cycles).

Horizontal SYNC Constraints

- If interlaced display mode is used:
HS > 5 Display Word Cycles (10 clock cycles).

Modes of Operation Bits

C, G	Display Mode
0 0	Mixed Graphics & Character
0 1	Graphics Mode
1 0	Character Mode
1 1	Invalid
I, S	Video Framing
0 0	Noninterlaced
0 1	Invalid
1 0	Interlaced Repeat Field for Character Displays
1 1	Interlaced

Repeat Field Framing: 2 Field Sequence with 1/2 line offset between otherwise identical fields.

Interlaced Framing: 2 Field Sequence with 1/2 line offset. Each field displays alternate lines.

Noninterlaced Framing: 1 field brings all of the information to the screen.

Total scanned lines in interlace mode is odd. The sum of VFP - VS - VBP - AL should equal one less than the desired odd number of lines.

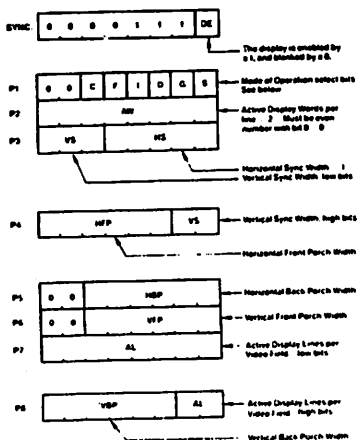
D	Dynamic RAM Refresh Cycles Enable
0	No Refresh - STATIC RAM
1	Refresh - Dynamic RAM

Dynamic RAM refresh is important when high display zoom factors or DMA are used in such a way that not all of the rows in the RAMs are regularly accessed during display raster generation and for otherwise inactive display memory.

F	Drawing Time Window
0	Drawing during active display time and retrace blanking
1	Drawing only during retrace blanking

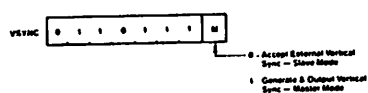
Access to display memory can be limited to retrace blanking intervals only, so that no disruptions of the image are seen on the screen.

SYNC Format Specify



This command also loads parameters into the sync generator. The various parameter fields and bits are identical to those at the RESET command. The GDC is not reset nor does it enter idle mode.

Vertical Sync Mode



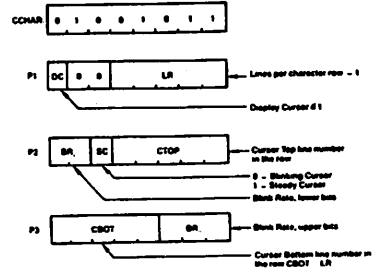
When using two or more GDCs to contribute to one image, one GDC is defined as the master sync generator, and the others operate as its slaves. The VSYNC pins of all GDCs are connected together.

A few considerations should be observed when synchronizing two or more GDCs to generate overlaid video via the VSYNC INPUT OUTPUT pin. As mentioned above, the Horizontal Front Porch (HFP) must be 4 or more display cycles wide. This is equivalent to eight or more clock cycles. This gives the slave GDCs time to initialize their internal video sync generators to the proper point in the video field to match the incoming vertical sync pulse (VSYNC). This resetting of the generator occurs just after the end of the incoming VSYNC pulse, during the HFP interval. Enough time during HFP is required to allow the slave GDC to complete the operation before the start of the HSYNC interval.

Once the GDCs are initialized and set up as Master and Slaves, they must be given time to synchronize. It is a good idea to watch the VSYNC status bit of the Master GDC and

wait until after one or more VSYNC pulses have been generated before the display process is started. The START command will begin the active display of data and will end the video synchronization process, so be sure there has been at least one VSYNC pulse generated for the Slaves to synchronize to.

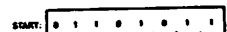
Cursor & Character Characteristics



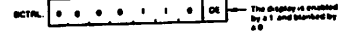
In graphics mode, LR should be set to 0. The blink rate parameter controls both the cursor and attribute blink rates. The cursor blink-on time = blink-off time = $2 \times BR$ (video frames). The attribute blink rate is always $1/2$ the cursor rate but with a $1/4$ on- $1/4$ off duty cycle. All three parameter bytes must be output for interface displays, regardless of mode. For interface displays in graphics mode, the parameter $BR_1 = 3$.

Display Control Commands

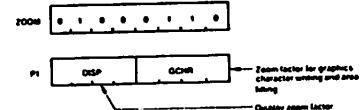
Start Display & End Idle Mode



Display Blanking Control

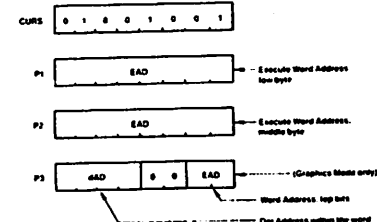


Zoom Factors Specify



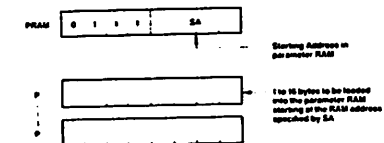
Zoom magnification factors of 1 through 16 are available using codes 0 through 15, respectively.

Cursor Position Specify



In character mode, the third parameter byte is not needed. The cursor is displayed for the word time in which the display scan address (DAD) equals the cursor address. In graphics mode, the cursor word address specifies the word containing the starting pixel of the drawing; the dot address value specifies the pixel within that word.

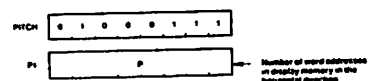
Parameter RAM Load



From the starting address, SA, any number of bytes may be loaded into the parameter RAM at incrementing addresses, up to location 15. The sequence of parameter bytes is terminated by the next command byte entered into

the FIFO. The parameter RAM stores 16 bytes of information in predefined locations which differ for graphics and character modes. See the parameter RAM discussion for bit assignments.

Pitch Specification

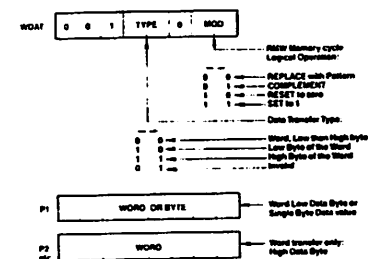


This value is used during drawing by the drawing processor to find the word directly above or below the current word, and during display to find the start of the next line.

The Pitch parameter (width of display memory) is set by two different commands. In addition to the PITCH command, the RESET (or SYNC) command also sets the pitch value. The "active words per line" parameter, which specifies the width of the raster-scan display, also sets the Pitch of the display memory. Note that the AW value is two less than the display window width. The PITCH command must be used to set the proper memory width larger than the window width.

Drawing Control Commands

Write Data into Display Memory



Upon receiving a set of parameters (two bytes for a word transfer, one for a byte transfer), one RMW cycle into Video Memory is done at the address pointed to by the cursor EAD. The EAD pointer is advanced to the next word, according to the previously specified direction. More parameters can then be accepted.

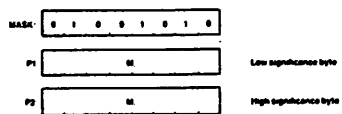
For byte writes, the unspecified byte is treated as all zeros during the RMW memory cycle.

In graphics bit-map situations, only the LSB of the WDAT parameter bytes is used as the pattern in the RMW operations. Therefore it is possible to have only all ones or all zeros pattern. In coded character applications all the bits of the WDAT parameters are used to establish the drawing pattern.

The WDAT command operates differently from the other commands which initiate RMW cycle activity. It requires

parameters to set up the Pattern register while the other commands use the stored values in the parameter RAM. Like all of these commands, the WDAT command must be preceded by a FIGS command and its parameters. Only the first three parameters need be given following the FIGS opcode, to set up the type of drawing, the DIR direction, and the DC value. The DC parameter + 1 will be the number of RMW cycles done by the GDC with the first set of WDAT parameters. Additional sets of WDAT parameters will see a DC value of 0 which will cause only one RMW cycle to be executed per set of parameters.

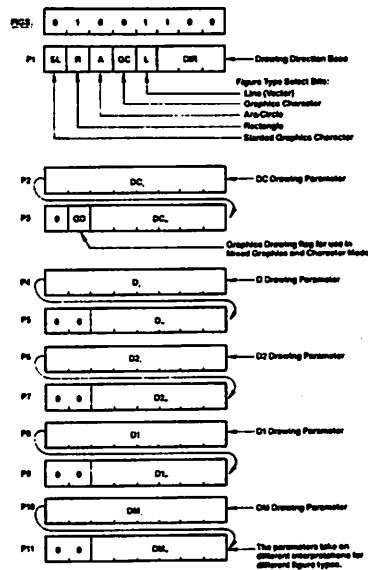
Mask Register Load



This command sets the value of the 16-bit Mask register of the figure drawing processor. The Mask register controls which bits can be modified in the display memory during a read-modify-write cycle.

The Mask register is loaded both by the MASK command and the third parameter byte of the CURS command. The MASK command accepts two parameter bytes to load a 16-bit value into the Mask register. All 16 bits can be individually one or zero, under program control. The CURS command on the other hand, puts a "1 of 16" pattern into the Mask register based on the value of the Dot Address value, dAD. If normal single-pixel-at-a-time graphics figure drawing is desired, there is no need to do a MASK command at all since the CURS command will set up the proper pattern to address the proper pixels as drawing progresses. For coded character DMA, and screen setting and clearing operations using the WDAT command, the MASK command should be used after the CURS command if its third parameter byte has been output. The Mask register should be set to all "ONES" for any "word-at-a-time" operation.

Figure Drawing Parameters Specify

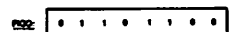


Valid Figure Type Select Combinations

SL	R	A	DC	L	Operation
0	0	0	0	0	Character Display Mode Drawing, Individual Dot Drawing, DMA, WDAE, and RDAE
0	0	0	0	1	Straight Line Drawing
0	0	0	1	0	Graphics Character Drawing and Area filling with graphics character patterns
0	0	1	0	0	Area and Circle Drawing
0	1	0	0	0	Rectangle Drawing
1	0	0	1	0	Stenciled Graphics Character Drawing and Stenciled Area Filling

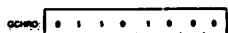
Only these bit combinations assure correct drawing operation.

Figure Draw Start



On execution of this instruction, the GDC loads the parameters from the parameter RAM into the drawing processor and starts the drawing process at the pixel pointed to by the cursor, EAD, and the dot address, dAD.

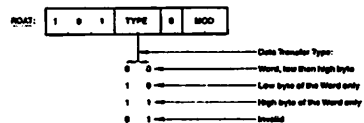
Graphics Character Draw and Area Filling Start



Based on parameters loaded with the FIGS command, this command initiates the drawing of the graphics character or area filling pattern stored in Parameter RAM. Drawing begins at the address in display memory pointed to by the EAD and dAD values.

Data Read Commands

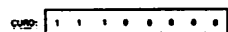
Read Data from Display Memory



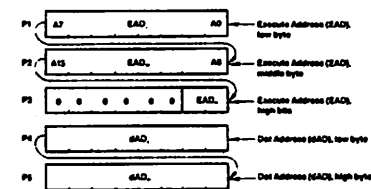
Using the DIR and DC parameters of the FIGS command to establish direction and transfer count, multiple RMW cycles can be executed without specification of the cursor address after the initial load (DC = number of words or bytes).

As this instruction begins to execute, the FIFO buffer direction is reversed so that the data read from display memory can pass to the microprocessor. Any commands or parameters in the FIFO at this time will be lost. A command byte sent to the GDC will immediately reverse the buffer direction back to write mode, and all RDAT information not yet read from the FIFO will be lost. MOD should be set to 00 if no modification to video buffer is desired.

Cursor Address Read



The following bytes are returned by the GDC through the FIFO:

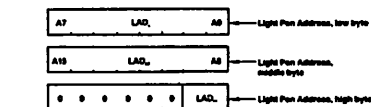


The Execute Address, EAD, points to the display memory word containing the pixel to be addressed. The Dot Address, dAD, within the word is represented as a 1-of-16 code for graphics drawing operations.

Light Pen Address Read



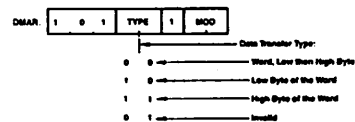
The following bytes are returned by the GDC through the FIFO:



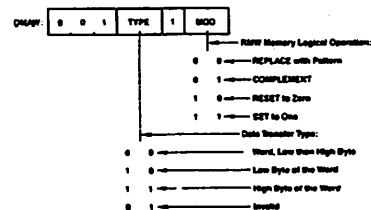
The light pen address, LAD, corresponds to the display word address, dAD, at which the light pen input signal is detected and digitized.

The light pen may be used in graphics, character, or mixed modes but only indicates the word address of light pen position.

DMA Read Request



DMA Write Request



AC Characteristics, $\mu P072200$ (Cont'd)

Display Cycle (QDC - Display Memory)									
Parameter	Symbol	Min	Max	72200 Limits	72200-1 Limits	72200-2 Limits	Unit	Test Conditions	
Video Signal Delay from STROBE	t_{VD}		150		100	100	ns	$C_L = 50\text{ pF}$	
Input Cycle (QDC - Display Memory)									
Parameter	Symbol	Min	Max	72200 Limits	72200-1 Limits	72200-2 Limits	Unit	Test Conditions	
Input Signal Setup to STROBE	t_{SI}		20		20	15	ns		
Input Signal Hold after STROBE	t_{SH}		t_{SH}				ns		
Clock (STROBE)									
Parameter	Symbol	Min	Max	72200 Limits	72200-1 Limits	72200-2 Limits	Unit	Test Conditions	
Clock Rise Time	t_{CR}		20		20	20	ns		
Clock Fall Time	t_{CF}		20		20	20	ns		
Clock High Pulse Width	t_{CH}		100		60	70	ns		
Clock Low Pulse Width	t_{CL}		100		60	70	ns		
Clock Cycle	t_{CC}		250		200	180	ns		

Absolute Maximum Ratings* (Tentative)

Ambient Temperature under Bias	0°C to 70°C
Storage Temperature	-55°C to 150°C
Voltage on Any Pin with Respect to Ground	-0.5V to +7V
Power Dissipation	1.5 W

*COMMENT: Exposing the device to stresses above those listed in Absolute Maximum Ratings could cause permanent damage. The device is not meant to be operated under conditions outside the limits described in the operational sections of this specification. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

DC Characteristics

$T_a = 0^\circ\text{C}$ to 70°C ; $V_{CC} = 5V \pm 10\%$; $GND = 0V$

Parameter	Symbol	Min	Typ	Max	Unit	Test Conditions
Input Low Voltage	V_{IL}	0	2.2	0.8	V	$V_{CC} = 5V$
Input High Voltage	V_{IH}	2.2	5.0	5.0	V	$V_{CC} = 5V$
Output Low Voltage	V_{OL}	0	0.4	0.4	V	$I_{OL} = 2.2\text{ mA}$
Output High Voltage	V_{OH}	2.4	4.5	4.5	V	$I_{OH} = -40\text{ }\mu\text{A}$
Input Low Current	I_{IL}	-10	-10	-10	μA	$V_{IL} = 0.8V$
Input High Current	I_{IH}	10	10	10	μA	$V_{IH} = 5.0V$
Output Low Current	I_{OL}	-10	-10	-10	μA	$V_{OL} = 0.4V$
Output High Current	I_{OH}	10	10	10	μA	$V_{OH} = 4.5V$
Clock Input Low Voltage	V_{CL}	0	0.8	0.8	V	
Clock Input High Voltage	V_{CH}	2.4	4.5	4.5	V	
V_{CC} Supply Current	I_{CC}		3.5	3.5	mA	$V_{CC} = 5.0V$

Capacitance

$T_a = 25^\circ\text{C}$; $V_{CC} = GND = 0V$

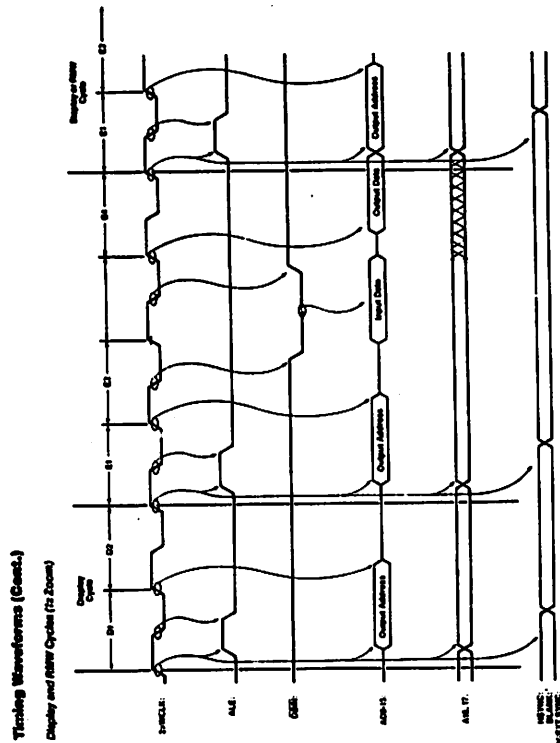
Parameter	Symbol	Min	Typ	Max	Unit	Test Conditions
Input Capacitance	C_i		10	10	pF	
Output Capacitance	C_o		20	20	pF	$f = 1\text{ MHz}$
Clock Capacitance	C_{CL}		20	20	pF	Continuous = 0V

Notes:
1. STROBE: $V_{CC} = 5.0V$, $V_{IL} = 0.8V$
2. STROBE: $V_{CC} = 5.0V$, $V_{OH} = 4.5V$

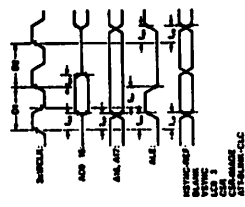
AC Characteristics, $\mu P072200$

$T_a = 0^\circ\text{C}$ to 70°C ; $V_{CC} = 5.0V \pm 10\%$; $GND = 0V$

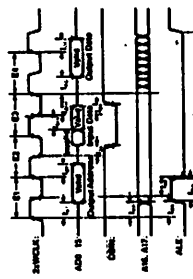
Read Cycle (QDC - CPU)									
Parameter	Symbol	Min	Max	72200 Limits	72200-1 Limits	72200-2 Limits	Unit	Test Conditions	
Address Setup to RD ¹	t_{AS}		0		0	0	ns		
Address Hold from RD ¹	t_{AH}		0		0	0	ns		
RD Pulse Width	t_{PW}		$t_{PW} + 20$		$t_{PW} + 20$	$t_{PW} + 20$	ns	$C_L = 50\text{ pF}$	
RD Delay from RD ¹	t_{DR}		120		20	70	ns		
RD Pulse Setup to RD ¹	t_{SR}		120		100	50	ns		
RD Pulse Cycle	t_{PC}		$4 t_{CC}$		$4 t_{CC}$	$4 t_{CC}$	ns		
Write Cycle (QDC - CPU)									
Parameter	Symbol	Min	Max	72200 Limits	72200-1 Limits	72200-2 Limits	Unit	Test Conditions	
Address Setup to WR ¹	t_{AS}		0		0	0	ns		
Address Hold from WR ¹	t_{AH}		120		100	70	ns		
WR Pulse Width	t_{PW}		100		60	70	ns		
WR Delay from WR ¹	t_{DR}		100		60	70	ns		
WR Pulse Setup to WR ¹	t_{SR}		100		60	70	ns		
WR Pulse Cycle	t_{PC}		$4 t_{CC}$		$4 t_{CC}$	$4 t_{CC}$	ns		
DMA Read Cycle (QDC - CPU)									
Parameter	Symbol	Min	Max	72200 Limits	72200-1 Limits	72200-2 Limits	Unit	Test Conditions	
DAC Setup to RD ¹	t_{AS}		0		0	0	ns		
DAC Hold from RD ¹	t_{AH}		0		0	0	ns		
RD Pulse Width	t_{PW}		$t_{PW} + 20$		$t_{PW} + 20$	$t_{PW} + 20$	ns	$C_L = 50\text{ pF}$	
RD Delay from RD ¹	t_{DR}		120		20	70	ns		
DAC Delay from RD ¹	t_{DR}		120		20	70	ns		
DAC Pulse Setup to RD ¹	t_{SR}		120		100	50	ns		
DAC Pulse Cycle	t_{PC}		$4 t_{CC}$		$4 t_{CC}$	$4 t_{CC}$	ns		
DAC High Level Width	t_{CH}		$t_{CH} + 100$		$t_{CH} + 100$	$t_{CH} + 100$	ns	$C_L = 50\text{ pF}$	
DAC Low Level Width	t_{CL}		$t_{CL} + 100$		$t_{CL} + 100$	$t_{CL} + 100$	ns	$C_L = 50\text{ pF}$	
DAC Delay from RD ¹	t_{DR}		120		20	70	ns		
DAC Pulse Setup to RD ¹	t_{SR}		120		100	50	ns		
DAC Pulse Cycle	t_{PC}		$4 t_{CC}$		$4 t_{CC}$	$4 t_{CC}$	ns		
DMA Write Cycle (QDC - CPU)									
Parameter	Symbol	Min	Max	72200 Limits	72200-1 Limits	72200-2 Limits	Unit	Test Conditions	
DAC Setup to WR ¹	t_{AS}		0		0	0	ns		
DAC Hold from WR ¹	t_{AH}		120		100	70	ns		
WR Pulse Width	t_{PW}		100		60	70	ns		
WR Delay from WR ¹	t_{DR}		100		60	70	ns		
WR Pulse Setup to WR ¹	t_{SR}		100		60	70	ns		
WR Pulse Cycle	t_{PC}		$4 t_{CC}$		$4 t_{CC}$	$4 t_{CC}$	ns		
BIOS Cycle (QDC - Display Memory)									
Parameter	Symbol	Min	Max	72200 Limits	72200-1 Limits	72200-2 Limits	Unit	Test Conditions	
Address Setup to RD ¹	t_{AS}		0		0	0	ns		
Address Hold from RD ¹	t_{AH}		120		100	70	ns		
RD Pulse Width	t_{PW}		100		60	70	ns		
RD Delay from RD ¹	t_{DR}		100		60	70	ns		
RD Pulse Setup to RD ¹	t_{SR}		100		60	70	ns		
RD Pulse Cycle	t_{PC}		$4 t_{CC}$		$4 t_{CC}$	$4 t_{CC}$	ns		
BIOS Cycle (QDC - Display Memory)									
Parameter	Symbol	Min	Max	72200 Limits	72200-1 Limits	72200-2 Limits	Unit	Test Conditions	
Address Setup to RD ¹	t_{AS}		0		0	0	ns		
Address Hold from RD ¹	t_{AH}		120		100	70	ns		
RD Pulse Width	t_{PW}		100		60	70	ns		
RD Delay from RD ¹	t_{DR}		100		60	70	ns		
RD Pulse Setup to RD ¹	t_{SR}		100		60	70	ns		
RD Pulse Cycle	t_{PC}		$4 t_{CC}$		$4 t_{CC}$	$4 t_{CC}$	ns		



Display Memory Display Cycle Timing

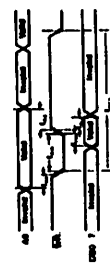


Display Memory Raster Timing

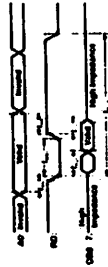


Timing Waveforms

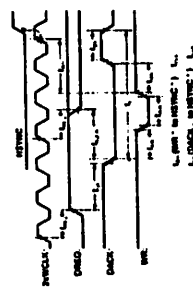
Microprocessor Interface Write Timing



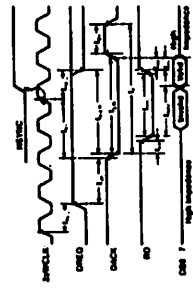
Microprocessor Interface Read Timing



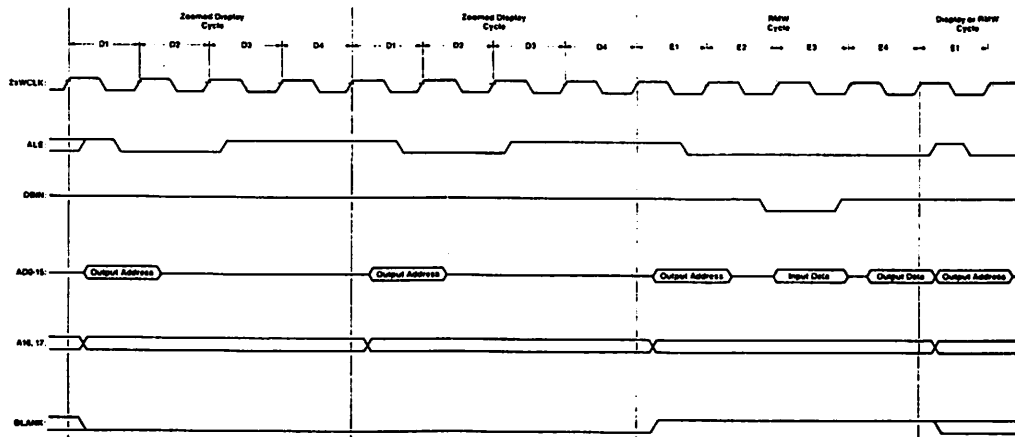
Microprocessor Interface DMA Write Timing



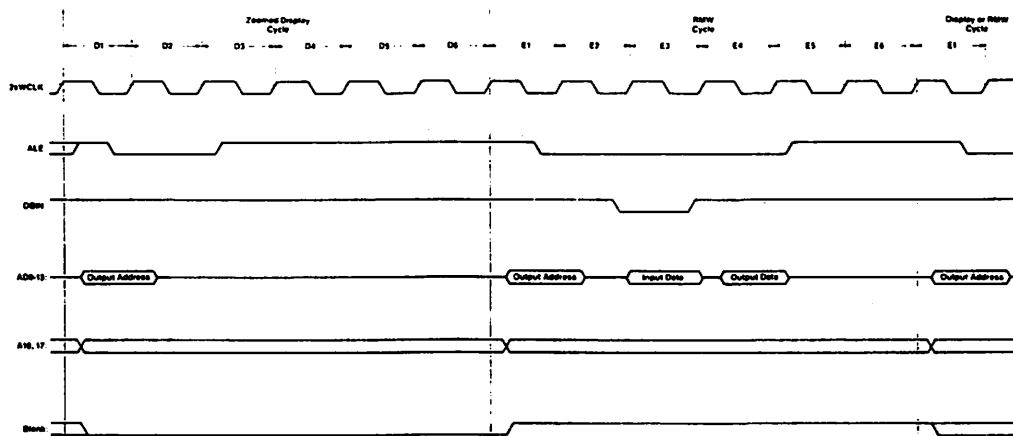
Microprocessor Interface DMA Read Timing



Display and RMW Cycles (2x Zoom)



Zoomed Display Operation with RMW Cycle (3x Zoom)

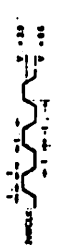


Timing Waveforms (Cont.)

Light Pen and Exit and Spec Input Timing



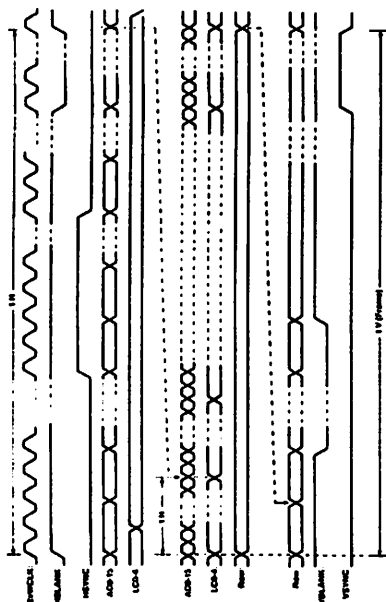
Clock Timing (2XINVCLEO)



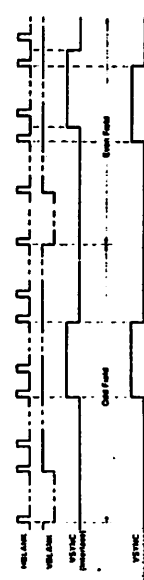
Test Level (for AC Tests, except ZTCCLK)



Video Sync Signals Timing

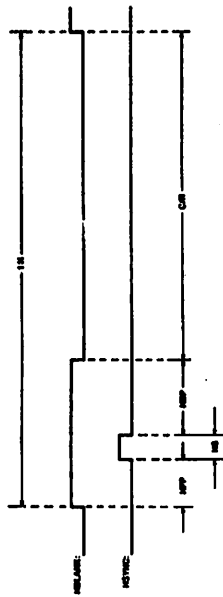


Interlaced Video Timing

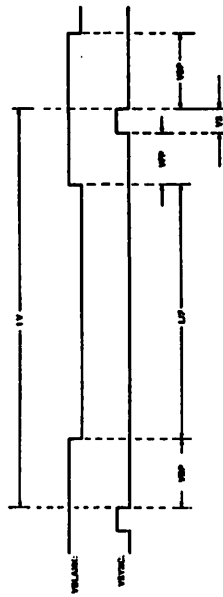


Thinking Wonders (Cont.)

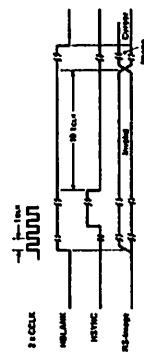
Video Horizontal Span Converter Parameters

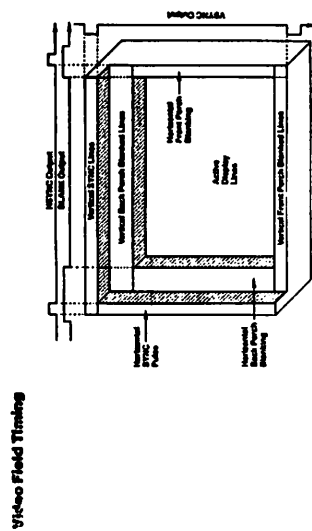
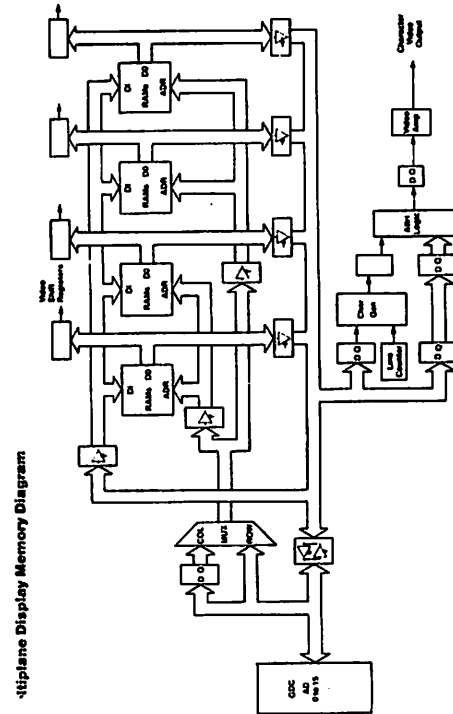
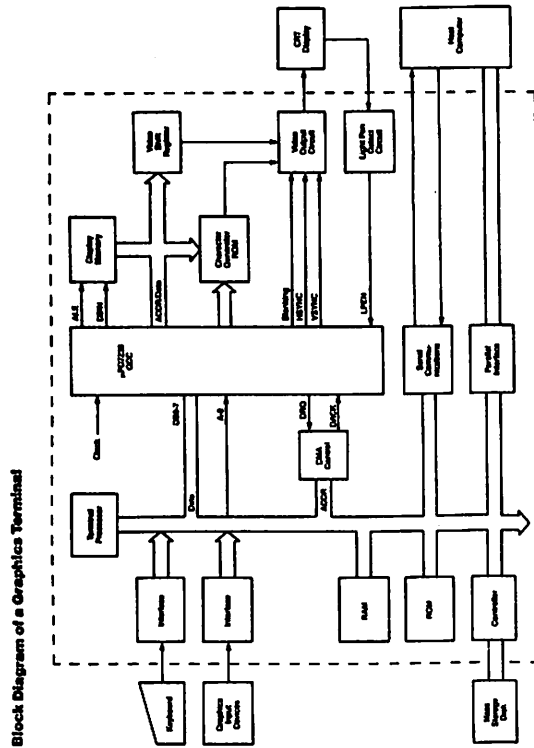


Video Vertical Sync Generator Parameters

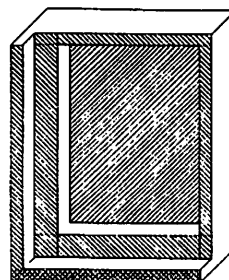


Cursor - Image Edit Flag

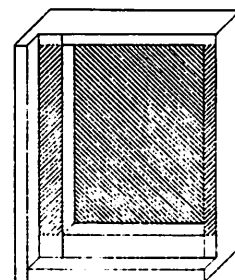




Drawing Intervals



DMA Request Intervals



Stückliste:**IC's:**

IC 00	74 LS 151
IC 01	74 S 163
IC 02	74 LS 14
IC 03	74 S 04
IC 04	74 LS 122
IC 05/14	TBP 24 SA 10 (I/O- und Bus-Steuer-Prom)
IC 06	74 LS 245
IC 07/10	74 LS 74
IC 08	74 S 175
IC 09	4929 / 49829
IC 11	74 S 32
IC 12	2716 (Charakter-Eprom)
IC 13	8255A-5
IC 15	74 S 32
IC 16	uPD 7220
IC 17,27,35	74 LS 245
IC 19,20	74 LS 157
IC 18,28,36	74 LS 166
IC 21-23	TMS 4416-20 oder TMS 4464-20
IC 25,33,41	74 LS 166
IC 26,34,42	74 LS 245
IC 29-32	TMS 4416-20 oder TMS 4464-20
IC 37-40	TMS 4416-20 oder TMS 4464-20
IC 43	78 L 05
IC 44	7407

Quarze:

Q Quarz $f_{typ} = 16 \text{ MHz}$

Widerstände:

R 0,15	1K2 Ohm		
R 1-4	680 Ohm		
R 5	1K2 Ohm		
R 6	3K9 Ohm		
R 7	1K2 Ohm		
R 8-10	680 Ohm		
R 11	68 Ohm		
R 12	1K2 Ohm		
R 13-14	1k2 Ohm		
R 16-17	1k2 Ohm		
R V	240 Ohm	(1024x512)	270 Ohm (512x512)
R S	390 Ohm	"	270 Ohm "
R R	300 Ohm	"	270 Ohm "
R G	300 Ohm	"	270 Ohm "
R B	-		270 Ohm "

Andere Halbleiter:

D 0-3	1 N 4148 o.ä.
T	BCY 58 VIII

Kondensatoren:

K	15 x 100 nF
C 0,2	4,7 uF 16 V Tantal
C 1	820 pF
C 3	47 nF
C 6-10	4,7 uF 16 V Tantal

Stecker:

S 0	64 polige VG-Steckerleiste a+c
S 1	Miniatur Video HF Buchse
S 2	9-poliger D-Stecker

