

**Integrierte  
Schaltkreise**



**Hinweise  
zur Anwendung**

**Grafik – Display – Controller  
GDC U 82 720**

**Applikation**

Die technischen Angaben dieses Handbuchs tragen reinen Informationscharakter.  
Verbindliche technische Liefer- und Reklamationsgrundlage sind ausschließlich die  
Typstandards.  
Die vorliegende Dokumentation gibt keine Auskunft über Liefermöglichkeiten und beinhaltet  
keine Verbindlichkeiten zur Produktion.



## Inhalt

1.	Eigenschaften des Grafik-Display-Controllers U82720 .....	7
1.1.	Einführung .....	7
1.1.1.	Spezifikationen des GDC U82720 .....	7
1.2.	Blockschaltbild .....	9
1.3.	Erzeugung der Informationen im Bildwiederholtspeicher .....	11
2.	Betriebsarten des GDC U82720 .....	13
2.1.	Grafik-Betriebsart .....	13
2.2.	Zeichen-Betriebsart .....	14
2.3.	Misch-Betriebsart .....	14
3.	Funktionsbeschreibung des GDC U82720 .....	15
3.1.	Einführung .....	15
3.2.	Grafik-Betriebsart .....	15
3.2.1.	Speicherorganisation .....	15
3.2.2.	Zeitverhalten der Bilddarstellung .....	15
3.2.3.	Spezielle Darstellungsfunktionen .....	16
3.2.3.1.	Ausschnittdarstellung .....	16
3.2.3.2.	Vergrößerung .....	16
3.2.3.3.	Schwenkfunktion .....	17
3.2.4.	Zeichnen und Schreiben .....	17
3.2.4.1.	Veränderung des Speichers .....	17
3.2.4.2.	Befehle zum Lesen und Schreiben .....	18
3.3.	Zeichen-Betriebsart .....	21
3.3.1.	Speicherorganisation .....	21
3.3.2.	Zeitverhalten der Bilddarstellung .....	21
3.3.3.	Spezielle Darstellungsfunktionen .....	22
3.3.3.1.	Ausschnittdarstellung .....	22
3.3.3.2.	Vergrößerung und Schwenkfunktion .....	22
3.3.4.	Zeichnen und Schreiben .....	22
3.4.	Misch-Betriebsart .....	22
3.4.1.	Speicherorganisation .....	23
3.4.2.	Zeitverhalten der Bilddarstellung .....	23
3.4.3.	Spezielle Darstellungsfunktionen .....	24
3.4.3.1.	Ausschnittdarstellung .....	24
3.4.3.2.	Vergrößerung und Schwenkfunktion .....	24
3.4.4.	Zeichnen und Schreiben .....	24
4.	Programmierung des GDC .....	25
4.1.	Allgemeines .....	25
4.2.	Statusregisterflags .....	25
4.3.	FIFO-Operation und Kommandoprotokoll .....	26
4.4.	Read-Modify-Write-Zyklus .....	28
4.5.	Ablauf der Bilddarstellung .....	29
4.6.	Zeichnungsparameter .....	31
4.7.	Darstellung grafischer Zeichen .....	32
4.8.	Funktion des Parameter-RAM .....	33
4.9.	Image-Flag .....	36
4.9.1.	Image-Flag in der Misch-Betriebsart .....	36

4.9.1.1.	Zeitverhalten .....	37
4.9.2.	Image-Flag in der Grafik-Betriebsart .....	37
5.	Beschreibung der Kommandos des GDC U82720 .....	39
5.1.	Video-Steuerkommandos .....	39
5.1.1.	RESET-Kommando .....	39
5.1.2.	Kommandos zur Synchronisation .....	40
5.1.2.1.	Synchronisationsparameter .....	40
5.1.2.2.	SYNC-Kommando .....	42
5.1.2.3.	VSNC-Kommando .....	42
5.1.2.4.	CCHAR-Kommando .....	43
5.1.3.	Display-Steuerkommandos .....	44
5.1.3.1.	ZOOM-Kommando .....	45
5.1.3.2.	CURS-Kommando .....	45
5.1.3.3.	START-Kommando .....	45
5.1.3.4.	BCTRL-Kommando .....	45
5.1.3.5.	PRAM-Kommando .....	45
5.1.3.6.	PITCH-Kommando .....	46
5.1.4.	Zeichnungs-Steuerkommandos .....	46
5.1.4.1.	WDAT-Kommando .....	46
5.1.4.2.	MASK-Kommando .....	47
5.1.4.3.	FIGS-Kommando .....	47
5.1.4.4.	FIGD-Kommando .....	49
5.1.4.5.	GCHRD-Kommando .....	49
5.1.5.	Daten-Lesekommandos .....	49
5.1.5.1.	RDAT-Kommando .....	49
5.1.5.2.	CURD-Kommando .....	49
5.1.5.3.	LPRD-Kommando .....	50
5.1.5.4.	DMAR-Kommando .....	50
5.1.5.5.	DMAW-Kommando .....	51
6.	Zeichnen von Darstellungen .....	52
6.1.	Einführende Bemerkungen .....	52
6.2.	Adressierung der Bildpunkte beim Zeichnen der Darstellungen ....	52
6.3.	Architektur des BildwiederholSpeichers .....	53
6.3.1.	Überblick .....	53
6.3.2.	Betriebsarten .....	53
6.3.3.	Das Konzept des linearen Adreßraums im GDC U82720 .....	54
6.4.	Inhalt des BildwiederholSpeichers .....	55
6.5.	Kennzeichnung einer Pixeladresse im BildwiederholSpeicher .....	56
6.5.1.	Definition von EAD und dAD .....	56
6.5.2.	Umwandlung der x,y-Koordinaten in eine Speicheradresse .....	57
6.5.2.1.	Die 'obere linke Ecke' .....	57
6.5.2.2.	Wortadressen im BildwiederholSpeicher .....	58
6.5.2.3.	Das Finden von Wortadressen gegebener x,y-Koordinaten .....	58
6.5.2.4.	Das Finden der Bildpunktadressen .....	59
6.5.2.5.	Andere Koordinatensysteme .....	59
6.5.2.6.	Organisation des BildwiederholSpeichers .....	60
6.6.	Der digitale Differential Generator .....	61
6.7.	Zeichnungsprozess einer Darstellung .....	61
6.8.	Vorbereitung des GDC für das Zeichnen einer Darstellung .....	62
6.9.	Anwendung der FIGS-Parameter für das Zeichnen von Darstellungen .	63

6.10.	Parameterarten des FIGS-Kommandos .....	64
6.10.1.	Einführung .....	64
6.10.2.	Startkommando für die RMW-Operation .....	64
6.10.3.	Korrelation zwischen Kommando und Darstellungsart .....	64
6.11.	Das Zeichnen von Vektoren .....	65
6.12.	Beispiel für das Zeichnen eines Vektors .....	68
6.13.	Zeichenalgorithmus für Vektoren .....	70
6.14.	Zeichnen von Kreisbögen und Kreisen .....	72
6.15.	Positionierung des Cursors beim Zeichnen von Kreisbögen .....	72
6.16.	Berechnung der FIGS-Parameter für Kreisbögen .....	73
6.17.	Das Zeichnen von Rechtecken .....	75
6.18.	Ausfüllung von Flächen und Darstellung von grafischen Zeichen ...	76
6.19.	DMA-Übertragungen .....	80
6.19.1.	Einführung .....	80
6.19.2.	Vorbereitung für die DMA-Übertragung .....	80
6.19.3.	Beispiele für DMA-Übertragungen .....	81
7.	Video-Interface .....	84
7.1.	Berechnung des Videozeitverhaltens .....	84
7.1.1.	Einführende Bemerkungen .....	84
7.1.2.	Berechnung der Parameter .....	85
7.1.2.1.	Beispiel für eine Zeitrelationsberechnung .....	91
7.1.3.	Besonderheiten beim Zeilensprungverfahren .....	94
7.2.	Cursordarstellung .....	95
7.2.1.	Gedehnte Darstellungsart .....	95
7.2.2.	Cursorprogrammierung .....	96
A	Anhang .....	97
A-1.	statische Kennwerte .....	97
A-2.	dynamische Kennwerte .....	98
A-2.1.	Datenbus-Zeitverhalten Lesezyklus .....	98
A-2.2.	Datenbus-Zeitverhalten Schreibzyklus .....	98
A-2.3.	Bildwiederholpeicher-Zeitverhalten .....	99
A-2.4.	Lichtstiftsynchronisation .....	100
A-2.5.	Taktsignal-Zeitverhalten .....	100
A-2.6.	DMA-Zeitverhalten .....	101
A-3.	Zeitdiagramme .....	102
A-3.1.	Datenbus-Zeitverhalten Lesezyklus .....	102
A-3.2.	Datenbus-Zeitverhalten Schreibzyklus .....	102
A-3.3.	Zeitverhalten Bildwiederholpeicher (RMW-Zyklus) .....	103
A-3.4.	Lese-Zyklus .....	103
A-3.5.	Lichtstiftsynchronisation .....	104
A-3.6.	Taktzeitverhalten .....	104
A-3.7.	Zeitverhalten Cursor Image-Flag .....	104
A-3.8.	DMA-Zeitverhalten Lesen .....	105
A-3.9.	DMA-Zeitverhalten Schreiben .....	105
A-3.10.	Displayzyklen und RMW-Zyklen (1xZOOM) .....	106
A-3.11.	Displayzyklen und RMW-Zyklen (2xZOOM) .....	106
A-3.12.	Displayzyklen und RMW-Zyklen (3xZOOM) .....	107



A-3.13.	Zeitverhalten Video-Synchronisationssignale .....	108
A-3.14.	Zeitverhalten der Bilddarstellung mit Zeilensprungverfahren .....	108
A-3.15.	Parameter des Horizontal-Synchronisationsgenerators .....	109
A-3.16.	Parameter des Vertikal-Synchronisationsgenerators .....	109
A-3.17.	Videofeld-Zeitverhalten .....	110
A-3.18.	Zeichnungsintervalle .....	111
A-3.19.	DMA-Anforderungsintervalle .....	111

## **1. Eigenschaften des Grafik-Display-Controllers U82720**

### **1.1. Einführung**

Der U82720 Graphics Display Controller (GDC) ist ein für Mikrorechneranwendungen angepasster Controllerschaltkreis zur Steuerung von Rastergrafik- bzw. alphanumerischen Displays. Der GDC wird dabei zwischen Bildwiederholpeicher (Display Memory) und System-Mikroprozessor im Display angeordnet. Der GDC übernimmt die Verwaltung des Bildwiederholpeichers und die Erzeugung der Steuersignale für das Video-Interface.

Der GDC U82720 ermöglicht, insbesondere bei Systemen mit mehreren GDC's im Master-Slave-Betrieb, den Aufbau hochauflösender grafischer Displays mit dem Prinzip der Bilddarstellung nach dem Rasterdarstellungsverfahren.

Beim Rasterdarstellungsverfahren wird der Elektronenstrahl einer Katodenstrahlröhre in horizontaler und vertikaler Richtung zeilenförmig periodisch abgelenkt, das heißt Zeile für Zeile über den Bildschirm geführt und jeweils dort angeschaltet (hellgetastet), wo ein heller Punkt entstehen soll. Striche, Buchstaben und grafische Zeichen werden immer aus Punkten zusammengesetzt.

Der Aufwand für die Erarbeitung der Steuersoftware für ein grafisches Display kann durch den komfortablen Befehlssatz des GDC und die verfügbaren Zeichnungsalgorithmen zum Zeichnen grafischer Darstellungen klein gehalten werden.

Der vom GDC verwaltete Bildwiederholpeicher kann in den verschiedensten Formaten und Größen bis hin zu 256 KWorten zu je 16 Bit konfiguriert werden. Die Darstellungen auf dem Bildschirm können vergrößert (zooming) werden. Das Bewegen, bzw. Schwenken einer Darstellung (panning) kann softwaregesteuert ablaufen. Es können gleichzeitig mehrere Teile des Bildwiederholpeichers als sog. Ausschnitte (Fenster) auf dem Bildschirm dargestellt werden (windowing).

Ein Lichtstifteingang eignet sich besonders zur interaktiven Benutzerkommunikation am Grafik-Display.

#### **1.1.1. Spezifikationen des GDC U82720**

- > Mikroprozessor-Interface; DMA-Transfer in Verbindung mit DMA-Controllern möglich
- > Kommandospeicherung in GDC-internem FIFO-Puffer
- > Bildwiederholpeicher, adressierbar 256 KWorte zu je 16 Bit; Zugriffsmöglichkeit über Read-Modify-Write-Zyklus (RMW) und Display-Zyklus ohne RMW
- > Eingang für Lichtstift
- > Möglichkeit der externen Video-Synchronisation
- > Grafik-Betriebsart (Graphics Mode) programmierbar; Verwaltung von 4 MBit Bildwiederholpeicher
- >> Möglichkeiten bei der Grafikdarstellung
  - Zeichnen von Geraden, Bögen und Kreisen, Rechtecken und grafischen Zeichen mit einer Punktfolgefrequenz von 32 MHz in der normalen bzw. 64 MHz in der 'gedehnten' Darstellungsart (sog. Wide Display Mode);
  - Bildgröße bis zu 1024 \* 1024 Pixel in vier Bildebenen in Farbe oder Graustufen; zwei unabhängig voneinander 'rollbare' Bildfenster

- > Zeichen-Betriebsart (Character Mode) programmierbar; Verwaltung eines Bildwiederholerspeichers mit 8K \* 13 Bit Charakter- und Attributspeicher
- >> Möglichkeiten bei der Zeichendarstellung
  - automatische Verschiebung des Cursors; vier unabhängig voneinander rollbare Flächen; programmierbare Cursorgröße; 256 Zeichen pro Zeile; bis zu 100 Zeilen je Bildebene
- > Gemischte Grafik- und Zeichen-Betriebsart (Mixed Graphics and Character Mode) programmierbar; Verwaltung von 64K Zeichen-Bildwiederholerspeicher und ein Megapixel Grafik-Bildwiederholerspeicher
- > Vergrößerungsfaktor (Zoom) 1...16fach; Schwenkfunktion (Panning); programmierbare Parameter für das Video-Raster
- > DMA-Eigenschaften
  - Übertragung von Byte oder Wort in vier Taktperioden je Byte-Übertragung
- > Betriebsspannung +5 V; 40-Pin-DIL-Gehäuse (2,54 mm-Raster)

Im Bild 1.1 erfolgt die Darstellung der Anschlußbelegung des GDC U82720. Das zu verwendende Schaltbild ist im Bild 1.2 angegeben.

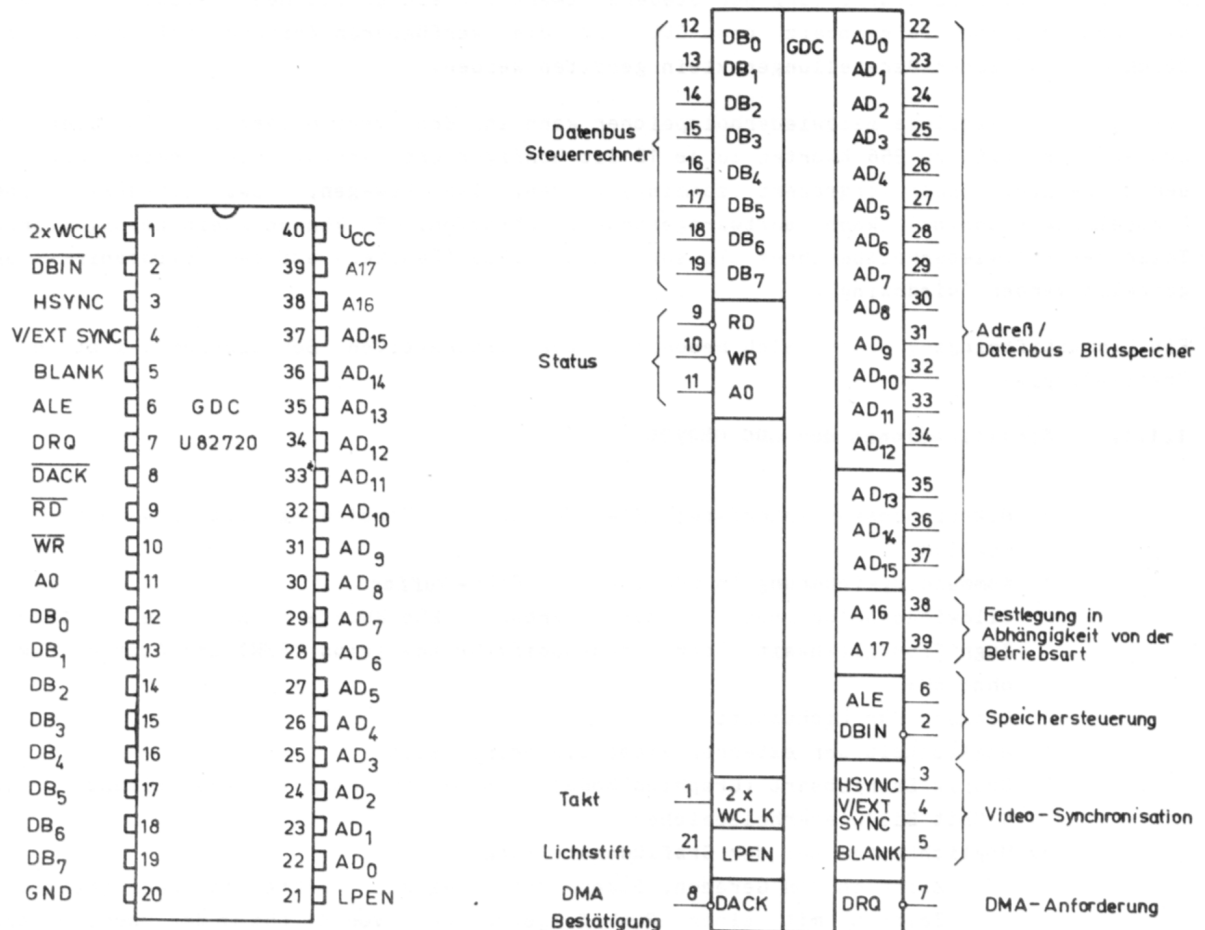


Bild 1.1: Anschlußbelegung

Bild 1.2: Schaltbild

Auf der Tafel 1.1 (s.a. Seite 12) wird eine zusammengefaßte Funktionskurzbeschreibung der Anschlüsse des GDC U82720 gegeben.



## 1.2. Blockschaltbild

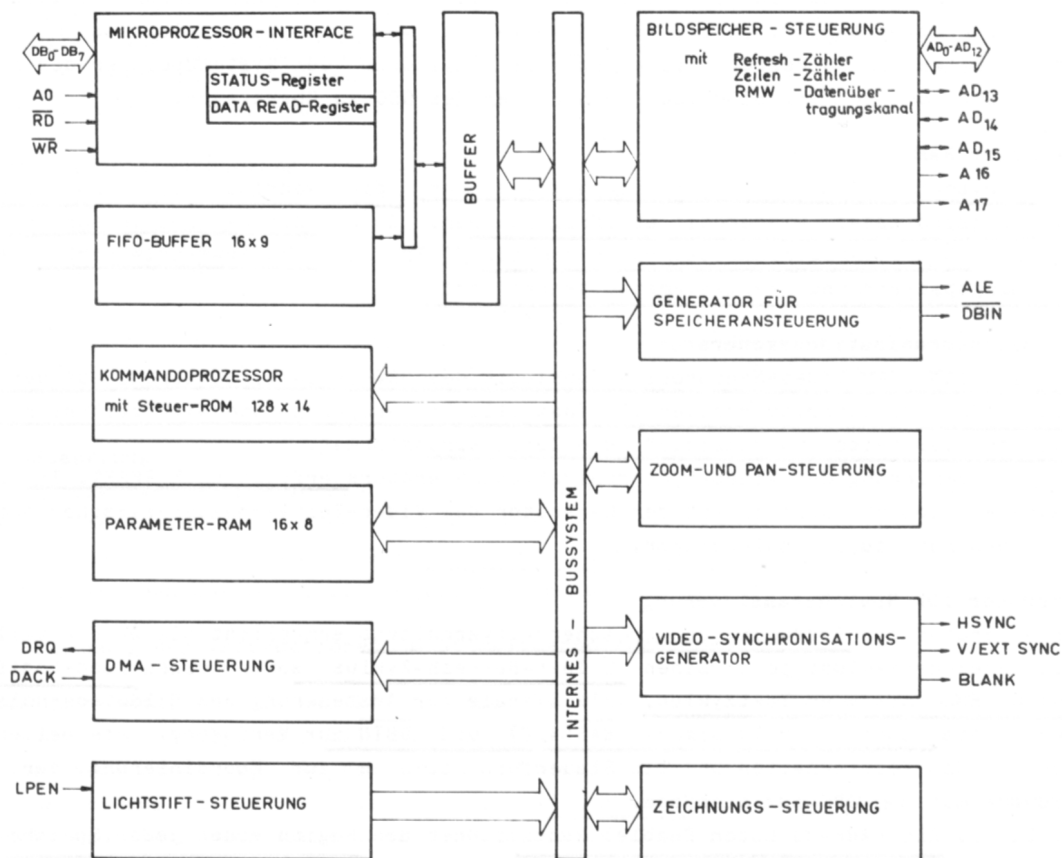


Bild 1.3: Blockschaltbild des GDC U82720

Nachfolgend werden die Elemente des Blockschaltbilds beschrieben:

### Mikroprozessor-Interface

Das Mikroprozessor-Interface mit den Datenbus-Ein-/Ausgängen  $DB_0 \dots DB_7$  ist acht Bit breit und bidirektional ausgeführt. Zum Interface gehören das Statusregister und das Data-Read-Register. Das Statusregister ist zu jedem beliebigen Zeitpunkt lesbar. Ein zusätzlicher FIFO-Buffer hat die Organisation  $9 * 16$  Bit. Der Zugriff zum FIFO-Buffer wird mit den Flags des Statusregisters sowie in Abhängigkeit von den verschiedenen internen GDC-Operationen koordiniert.

Nachfolgend werden einzelne Bits eines Registers, die eine besondere Bedeutung haben, einheitlich als Flag und mit ihrem Namen bezeichnet (Beispiel: Image Flag).

### Kommandoprozessor

Die Inhalte des FIFO-Puffer werden mit dem Kommandoprozessor (Command Processor) interpretiert. Dabei erfolgt eine Dekodierung der im FIFO befindlichen Kommandobytes und die Verteilung der Befehlsparameter an ihre Bestimmungsorte im GDC. Bei gleichzeitigem Zugriff von Businterface und Kommandoprozessor auf den FIFO-Buffer hat das Businterface die höhere Priorität.

### DMA-Steuerung

Die DMA-Steuerung (DMA Control) des GDC übernimmt die Datenübertragung am Mikroprozessor-Businterface, wenn mit einem externen DMA-Controller kommuniziert wird.

Die Steuersignale für DMA-Anforderung, DREQ (DMA-Request) sowie DMA-Bestätigung DACK (DMA-Acknowledge) können im Handshake-Betrieb für die Ansteuerung beliebiger DMA-Controller (z.B. UA858D, i8257, i8237) verwendet werden. Damit ist es möglich, die Display-Daten im direkten Speicherzugriff zwischen dem Systemspeicher des steuernden Mikroprozessors und dem Bildwiederholtspeicher des GDC zu übertragen.

#### **Parameter-RAM**

Der 16 Byte tiefe RAM speichert Befehlsparameter, die während des Darstellungs- und Zeichnungsprozesses wiederholt abgefragt werden. In der Zeichen-Betriebsart enthält der RAM die Display-Flächenaufteilungsparemeter. In der Grafik-Betriebsart werden die Zeichnungspattern und Grafikelemente gespeichert.

#### **Video-Synchronisationssgenerator**

Der Generator der Video-Synchronisationslogik des GDC erzeugt, basierend auf der Taktfrequenz, das Rasterzeitverhalten für fast alle Zeilensprung- und Nichtzeilensprungformate sowie für das "Wiederholungsfeld" beim Zeilensprungformat.

Die Programmierung des Video-Synchronisationsgenerators erfolgt nach RESET. In der sog. Slave-Betriebsart koordiniert der Generator das Video-Zeitverhalten zwischen dem GDC und einem anderen Videosignalerzeuger.

#### **Generator für Speicheransteuerung**

Ein Generator zur Erzeugung des Speicherzeitverhaltens ermöglicht die Bereitstellung von zwei Speicherzyklustypen: Einen 2-Takt-Refresh-Zyklus sowie einen Read-Modify-Write-Zyklus (RMW) mit vier Taktzyklen. Als Signale zur Ansteuerung des Bildwiederholtspeichers stehen die Speicherzyklussignale RAS(ALE) und DBIN zur Verfügung. Die beiden Signale erzeugen das Zeitverhalten und die Steuerinformationen zur Koordinierung der externen Hardware mit dem GDC-Businterface.

Das Signal ALE (Adress Latch Enable) kennzeichnet den Beginn eines jeden Speicherzyklus. Mit der Aktivierung des DBIN -Ausgangs kennzeichnet der GDC die Zeit innerhalb eines RMW-Zyklus, in der die an den RAMs des Bildwiederholtspeichers anliegenden Daten akzeptiert werden.

#### **Zoom- und Pan-Steuerung**

Der GDC ermöglicht unter Einsatz externer Hardware die Anwendung der programmierbaren Vergrößerung der Darstellung auf dem Bildschirm. Gleichfalls ist eine Schwenkbewegung über die für die Bilddarstellung benutzte Display-Bildfläche möglich.

Entsprechend der programmierten und im Parameter-RAM gespeicherten Informationen zum Vergrößerungsfaktor und zur Bildfläche des Displays, bestimmt die GDC-interne Zoom- und Pan-Steuerung, wann zur nächsten Speicheradresse für den Bildaufbau bzw. zur nächsten Display-Bildfläche übergegangen wird.

Das Grundprinzip für die Realisierung von Vergrößerungen besteht in der wiederholten hinter- bzw. untereinanderfolgenden Abbildung von Punkten mit gleichen Daten.

Durch das wiederholte Abbilden gleicher Bildpunkte in einer Zeile bei unveränderter Punktfolgefrequenz wird dabei eine Bildvergrößerung in horizontaler Richtung erreicht. Die entsprechende Bildvergrößerung in vertikaler Richtung erfolgt durch wiederholte Abbildung von Punkten mit den Daten der vorangegangenen Zeilen.

Falls die Zeilenzahl für eine Display-Bildfläche erschöpft ist, entnimmt die Zoom-Steuerung die Startadresse und die Anzahl der Zeilen für die nächste Display-Bildfläche vom Parameter-RAM.

Der System-Mikroprozessor kann durch Veränderung der Startadresse einer Display-Bildfläche das Schwenken in jede Richtung unabhängig von anderen Display-Bildflächen erreichen.

### **Zeichnungs-Steuerung**

Der Zeichnungsprozessor dieser Steuerung enthält die für die Berechnung der Pixeladressen und -positionen von diversen grafischen Darstellungen notwendige Logik. Außer einem gegebenen Startpunkt und entsprechenden Zeichnungsparametern benötigt der Zeichnungsprozessor keine weitere Unterstützung, um die Darstellung vollständig zu zeichnen.

### **Bildwiederholtspeicher-Steuerung**

Die Aufgaben der Steuerung des Bildwiederholtspeichers sind vielfältig. Am wichtigsten ist das Multiplexen der Adreß- und Dateninformationen von und zum Bildwiederholtspeicher des GDC. Die Steuerung enthält 16-Bit-Logikeinheiten, um den Inhalt des Bildwiederholtspeichers während der RMW-Zyklen ändern zu können. Weiterhin erfolgt die Ansteuerung des Zeilenzählers in der Zeichen-Betriebsart und des Refresh-Zählers für dynamische RAMs im Bildwiederholtspeicher. Gleichfalls bemißt die Steuerung die Video-Rasterzeit in den verschiedenen Zyklusarten.

### **Lichtstift-Steuerung**

Für den Fall, daß zwei steigende Flanken am Lichtstifteingang dem selben Punkt zweier aufeinanderfolgender Video-Raster entsprechen, werden die Impulse als gültige Lichtstiftinformationen anerkannt. Ein Statusbit des GDC zeigt danach dem System-Mikroprozessor an, daß das Lichtstiftregister eine gültige Adresse enthält.

### **1.3. Erzeugung der Informationen im Bildwiederholtspeicher**

Ein besonderes Merkmal des GDC U82720 ist die Benutzung eines eigenen Bildwiederholtspeichers zur Speicherung der Bildinformationen. Dabei können bis 512 KByte des Bildwiederholtspeichers direkt vom GDC adressiert werden.

Dies ist ausreichend, um ein Bild mit maximal 2048 x 2048 Pixel bei einem Bit Information pro Bildpunkt (nur hell oder dunkel, keine Zusatzinformationen!) aufzulösen.

Obwohl die GDC-interne Steuerung des Bildwiederholtspeichers während der Rasterabtastung die Adressen für den Bildwiederholtspeicher erzeugt, ist sie nicht unmittelbar an den Pixeloperationen zum Bildaufbau beteiligt. Die dafür notwendige Frequenz ist für Schaltkreise in MOS-Technologie viel zu hoch. Aus diesem Grund führt eine externe Hardware mittels eines Video-Schieberegisters eine Parallel-/Serienwandlung der Worte des Bildwiederholtspeichers in Pixel (Bildelemente) durch.

Die Bildinformationen können im Bildwiederholtspeicher (und damit praktisch auf der Display-Bildfläche) mit drei verschiedenen Möglichkeiten erzeugt werden:

- > Zeichnungs-Steuerkommandos
- > programmgesteuerter Übertragung
- > DMA-Übertragungen aus dem Speicher des System-Mikroprozessors

Die Systemarbeitsweise des GDC ist durch eine effektive Pipeline-Architektur gekennzeichnet. Die Erzeugung des Video-Signals erfolgt entsprechend einer festen Hierarchie: Auf der untersten Stufe erzeugt der GDC das grundlegende Zeitverhalten für das Rasterdarstellungsverfahren, einschließlich der Steuersignale zur Synchronisation und zum Dunkeltasten (blanking) des Elektronenstrahls. Gleichfalls kann die Flächeneinteilung des Bildschirms in Ausschnitte (Fenster) vorgenommen werden. Die Bildvergrößerung wird durch Programmierung der Zoom-Steuerung vorgenommen. In der nächsten Stufe erfolgt während des Darstellungsprozesses und der Datenübertragung eine Veränderung des Bildwiederholtspeicherinhalts. In der dritten Stufe wird während des Zeichnungsprozesses die pixelweise Berechnung der Bildwiederholtspeicheradressen durchgeführt. Die nachfolgende vierte Stufe umfaßt vorläufige Berechnungen außerhalb des GDC zur Vorbereitung von Zeichnungsparametern. Bei der fünften Stufe muß das Bild als eine Liste von



Grafikdarstellungen vorliegen, die vom GDC gezeichnet werden können. Zuletzt kann diese Darstellung verändert, gespeichert und weitergeleitet werden.

In Ergänzung zum Blockschaltbild (s.a. Seite 9) wird in nachfolgender Tafel eine Funktionskurzbeschreibung der Anschlüsse des GDC U82720 gegeben.

Symbol	Art	Benennung und Funktionskurzbeschreibung
2xWCLK	Eingang	Takteingang
DBIN	Ausgang	Buseingangssignal als Strobesignal zum Einlesen der Daten des BildwiederholSpeichers in den GDC
HSYNC	Ausgang	Horizontalsynchronisationssignal zur Erzeugung des Zeilenrücklaufs auf dem Display
V/EXT SYNC	Ein-/Ausgang	Vertikalsynchronisationssignal zur Erzeugung des Bildrücklaufs auf dem Display (Ausgangssignal); in der GDC-Slave-Betriebsart als Eingangssignal zur Synchronisation des Bildrasterzeitverhaltens mit einem Master-GDC
BLANK	Ausgang	Video-Austastsignal
RAS (ALE)	Ausgang	Speicherzyklussignal für Zeilenadreßübernahme (RAS-Signal) für dynamische RAMs oder Demultiplexsignal (ALE-Signal - Address Latch Enable) für den Adreß-/Datenbus mit statischen RAMs im BildwiederholSpeicher
DRQ	Ausgang	DMA-Anforderungssignal (DMA Request) zur Steuerung eines externen DMA-Controllers
DACK	Eingang	DMA-Bestätigungssignal (DMA Acknowledge) zur Steuerung eines externen DMA-Controller
RD	Eingang	Strobesignal (Read) zum Einlesen von GDC-Daten in den System-Mikroprozessor
WR	Eingang	Strobesignal (Write) zum Einlesen von Mikroprozessor-Daten in den GDC
AO	Eingang	Auswahl der Adresse zur Unterscheidung von Kommandos und Lese-/Schreibdaten
DB <sub>0</sub> ...DB <sub>7</sub>	Ein-/Ausgänge	bidirektionaler Mikroprozessor-Datenbus des GDC; Bussteuerung erfolgt mit den Signalen WR für Eingabe und RD für Ausgabe
GND	-	Masseanschluß
LPEN	Eingang	Lichtstiftsignal (Light Pen Detect)
AD <sub>0</sub> ...AD <sub>12</sub>	Ein-/Ausgänge	Adreß-/Datenbus des BildwiederholSpeichers (Bit 0...12)
AD <sub>13</sub> ...AD <sub>15</sub>	Ein-/Ausgänge	Funktion wird entsprechend der Betriebsart des GDC festgelegt: - bei Grafik-Betriebsart --> Adreß-Datenbus des BildwiederholSpeichers (Bit 13...15) - bei Zeichen-Betriebsart --> Zeilenzählerausgänge (Bit 0...2) - bei Misch-Betriebsart --> Adreß-/Datenbus des BildwiederholSpeichers (Bit 13...15)
A16	Ausgang	Funktion wird entsprechend der Betriebsart des GDC festgelegt: - bei Grafik-Betriebsart --> Adreßbus des BildwiederholSpeichers (Bit 16) - bei Zeichen-Betriebsart --> Zeilenzählerausgang (Bit 3) - bei Misch-Betriebsart --> Attributblinksignal und Rücksetzen des Zeilenzählers
A17	Ausgang	Funktion wird entsprechend der Betriebsart des GDC festgelegt: - bei Grafik-Betriebsart --> Adreßbus des BildwiederholSpeichers (Bit 17) - bei Zeichen-Betriebsart --> Cursor- und Zeilenzählerausgang (Bit 4) - bei Misch-Betriebsart --> Cursorausgang und Flaganzeige für Bildbetriebsart
U <sub>CC</sub>	-	Betriebsspannungsanschluß +5V

Tafel 1.1: Funktionskurzbeschreibung der Anschlüsse des U82720

## 2. Betriebsarten des GDC U82720D

Der GDC U82720 bietet die Möglichkeit des Einsatzes bei unterschiedlichen Anforderungen an die Auflösung des Bildes und die Art der Darstellung.

Drei Betriebsarten können prinzipiell aufgerufen werden:

- Grafik-Betriebsart - ermöglicht schnelle Darstellung grafischer Bildelemente
- Zeichen-Betriebsart - ermöglicht Darstellung von Zeichenketten in alphanumerischen Displays
- Misch-Betriebsart - ermöglicht die komfortable Anwendung von beiden vorstehend genannten Betriebsarten zur Darstellung von Zeichen und grafischen Bildelementen

Nachfolgend werden die einzelnen Betriebsarten des GDC vorgestellt.

### 2.1. Grafik-Betriebsart

Die bereits erwähnte Verwendung eines eigenen Bildwiederholerspeichers am GDC U82720 ermöglicht eine hohe Flexibilität bei der Lösung von Problemen grafischer Darstellungen. Die Organisation des Bildwiederholerspeichers ist von der Anzahl der Bits je Pixel (d.h. je Bildelement) und der Größe der Display-Bildfläche (d.h. der maximalen bzw. gewünschten Anzahl von Bildpunkten) abhängig.

Als einfachste Variante für den Einsatz des GDC in einem grafischen System kann die Aufteilung des Bildwiederholerspeichers in  $2048 * 2048 * 1$  Bit erfolgen. Dabei wird je Pixel nur ein Bit verwendet. Die Darstellung einer Bildsektion mit  $1024 * 1024 * 1$  Bit kann mit einer maximalen Punktfrequenz von 32 MHz, bzw. 64 MHz bei Verwendung des Wide-Display-Zyklus erfolgen.

Bei Programmierung dieses 'gedehnten' Display-Zyklus arbeitet der GDC mit doppeltem Wortzugriff im Bildwiederholerspeicher. In der Grafik-Betriebsart erfolgt dabei ein gleichzeitiger Zugriff auf 32 Bit (anstatt 16 Bit) der Daten im Bildwiederholerspeicher. Innerhalb eines RMW-Zyklus bearbeitet der GDC jedoch immer nur ein einzelnes 16-Bit-Wort. Vom GDC wird dieser Doppelwort-Display-Zyklus mit der Bereitstellung der Adresse jedes zweiten darzustellenden Speicherplatzes im Bildwiederholerspeicher unterstützt. Dabei wird, im Gegensatz zum Normalbetrieb, bei der Bestimmung der nächsten Adresse der GDC-interne Wortadresszähler immer um zwei erhöht. Mit einer externen Hardware ist sowohl der Zugriff als auch die Video-Serialisierung der unter geraden und ungeraden Adressen gespeicherten Worte im Bildwiederholerspeicher abzusichern.

Für den Wide-Display-Zyklus ist es generell erforderlich, in jeder Ebene des Bildwiederholerspeichers separate RAM-Bänke für beide 16-Bit-Worte zu verwenden, um die 32 Bit gleichzeitig in ein 32-Bit-Video-Schieberegister laden zu können. Dieser Zyklus wird in der Grafik-Betriebsart durch ein Bit des Parameter-RAM für jede darzustellende Display-Bildfläche programmiert.

Mit einer in der Grafik-Betriebsart wie auch in den anderen Betriebsarten möglichen Aufteilung des Bildwiederholerspeichers in mehrere Ebenen können für Displays mit entsprechend hoher Video-Bandbreite Farben oder Graustufen dargestellt werden. Der Bildwiederholerspeicher kann dabei maximal in vier gleichgroße Ebenen zu je  $1024 * 1024$  Bit aufgeteilt werden. Bei der Verwendung aller vier Ebenen zur Bilddarstellung können dann gleichzeitig vier Bit je Pixel bereitgestellt werden. Jede Ebene liefert ein Bit je Pixel des darzustellenden Bildes. Diese Konfiguration des Bildwiederholerspeichers ist für spezielle Farb-Displays geeignet, die eine maximale Punktfolgefrequenz von 32 MHz (bzw. 64 MHz) verarbeiten können.

Darüber hinaus gehende Anforderungen können mit mehreren GDC-Schaltkreisen des Typs U82720 erfüllt werden. Folgende Eigenschaften sind dann realisierbar:

- Mehrfach-Bildausschnitte
- Erhöhung der Zeichnungsgeschwindigkeit
- Vergrößerung der Bit-Anzahl je Pixel

Bei der Darstellung mehrerer Bildausschnitte von Mehrfach-Bildfenstern wird je GDC ein eigener Ausschnitt (Fenster) verwaltet. Zur Erhöhung der Geschwindigkeit der Darstellung arbeiten mehrere GDC parallel. Die Vergrößerung der Anzahl der Bit je Pixel wird dadurch erreicht, daß jeder GDC einen Teil der für die Darstellung eines Pixels notwendigen Bits beisteuert.

## **2.2. Zeichen-Betriebsart**

Ogbleich der U82720D vorrangig für Grafik im Rasterdarstellungsverfahren geeignet ist, kann auch eine Anwendung als Controller in alphanumerischen Displays erfolgen. In der Zeichen-Betriebsart verwaltet der GDC einen eigenen 8K \* 13 Bit großen Bildwiederhol-speicher. Dabei ist ein Zeichen 13 Bit breit. Der Bildwiederhol-speicher ist dabei ausreichend, vier Ebenen mit 25 Zeilen zu jeweils 80 Zeichen zu speichern. Eine Darstellung bis zu 256 Zeichen je Zeile ist möglich. Sanftes vertikales Scrollen wird in jeder der vier unabhängigen Bildebenen ermöglicht.

## **2.3. Misch-Betriebsart**

Der GDC U82720 unterstützt in dieser Betriebsart die gleichzeitige Darstellung von grafischen Elementen und Zeichen auf einer Display-Bildfläche. Dies wird durch folgende Eigenschaft des GDC in dieser Betriebsart ermöglicht: Die Display-Bildfläche wird in Teilflächen zur jeweiligen Darstellung von Grafik- und Zeichendaten zerlegt. Dabei ist es möglich, eine Umschaltung zwischen einer Grafik- und einer Zeichen-Teilfläche mit Raster-Zeilenauflösung durchzuführen.

In der Misch-Betriebsart können von einem GDC maximal 256 KBytes des Bildwiederhol-speichers adressiert werden. Dabei steht jeweils die Hälfte des Speicherraums für Grafik- bzw. Zeichendaten zur Verfügung.

Beim Grafik-Betrieb kann ein Megapixel Bilddaten gespeichert und zur Darstellung gebracht werden. Die Zeichen-Betriebsart erlaubt die Verwendung eines 64K-Bildwiederhol-speichers, wobei 16 Bit je Zeichen zur Verfügung stehen.



### 3. Funktionsbeschreibung des GDC U82720

#### 3.1. Einführung

Der GDC U82720 kann, wie bereits im Abschnitt 2. beschrieben, in drei grundlegenden Betriebsarten angewendet werden:

- > Grafik-Betriebsart (Graphics Mode)
- > Zeichen-Betriebsart (Character Mode)
- > Misch-Betriebsart (Mixed Mode)

Von der jeweiligen Betriebsart sind folgende Eigenschaften abhängig:

- >> Speicherorganisation
- >> Zeitverhalten der Bilddarstellung
- >> spezielle Funktionen bei der Bilddarstellung
- >> Zeichnungs- und Schreibfunktionen

Nachfolgend werden die o.g. Eigenschaften des GDC in den drei Betriebsarten beschrieben.

#### 3.2. Grafik-Betriebsart

##### 3.2.1. Speicherorganisation

Der Bildwiederholtspeicher ist in Worten zu jeweils 16 Bit organisiert. Bei programmiertem Wide-Display-Zyklus werden 32 Bit je Wort verwendet. Falls der Bildwiederholtspeicher größer als die maximal auf der Display-Bildfläche darzustellende Fläche ist, müssen die folgenden zwei Parameter definiert werden: Breite des Bildwiederholtspeichers und Breite der Display-Bildfläche.

Die Breite des Bildschirms wird mit einem Parameter des RESET-Kommandos ausgewählt. Mit einem Parameter des PITCH-Kommandos erfolgt die Auswahl der Breite des Bildwiederholtspeichers (s.a. Abschn. 5.1.).

Die Größe des Bildwiederholtspeichers kann größer als die Höhe der Display-Schirmfläche sein, die gleichfalls mit einem Parameter des RESET-Kommandos ausgewählt wird.

Ein U82720-GDC kann in der Grafik-Betriebsart bis zu vier Megabit Bildwiederholtspeicher-RAM adressieren.

##### 3.2.2. Zeitverhalten der Bilddarstellung

Das gesamte Zeitverhalten des GDC zur Dunkeltastung des Zeilenrasters und zur Darstellung des Bildes ist eine Funktion der Eingangstaktfrequenz  $2 \times WCLK$ . Innerhalb von zwei Taktperioden des Display-Zyklus werden jeweils 16 oder 32 Bit der Daten aus dem Bildwiederholtspeicher in das externe Video-Schieberegister geladen. Folglich beträgt bei einer Eingangstaktfrequenz von 4 MHz ein Display-Zyklus 500 ns.

Der Adreß-/Datenbus  $AD_0 \dots AD_{15}$  des GDC arbeitet im Multiplex-Verfahren. Unter Nutzung der

Ausgänge A16 und A17 kann eine 18 Bit Adresse zur Adressierung des Bildwiederholtspeichers gebildet werden.

Im ersten Teil des o.g. Display-Zyklus muß die Adresse des im Bildwiederholtspeicher zu lesenden Wortes in einem externen Demultiplexer zwischengespeichert werden. Im weiteren Teil dieses Zyklus werden dann die Daten aus dem Bildwiederholtspeicher ausgelesen und in das Video-Schieberegister geladen. Da sämtliche 16 (bzw. 32) Datenbits dargestellt werden müssen, berechnet sich die Punktfrequenz (dot clock) zu  $8 \text{ (bzw. 16)} * \text{GDC-Taktfrequenz oder } 16 \text{ (bzw. 32)} * \text{Lesezyklusrate.}$

Parameter des RESET- bzw. SYNC-Kommandos legen das Zeitverhalten der horizontalen (HFP) und vertikalen (VFP) vorderen Schwarzschilder, den Synchronimpuls und das Zeitverhalten der hinteren horizontalen und vertikalen Schwarzschilder (HBP, VBP) des Video-Signals (BAS-Signal) fest. Die horizontalen Parameter werden durch ein Vielfaches eines Display-Zyklus, die vertikalen Parameter durch ein Vielfaches der Zeilenzeit festgelegt. Mit einem weiteren Parameter des RESET-Kommandos wird festgelegt, ob in der Grafik-Betriebsart mit oder ohne Zeilensprungverfahren gearbeitet wird.

Ein Bit im Parameter-RAM -Flag (z.B. in RA-3, s.a. Abschn. 5.1.) kann den sog. Wide-Display-Zyklus auswählen. Bei dieser Darstellungsart wird während der Datenübertragung zum Bildwiederholtspeicher der Adreßzähler anstatt um eins um zwei erhöht. Dies ermöglicht bei Vorhandensein entsprechender externer Hardware, bei jedem Display-Lesezyklus, die Bereitstellung von 32 Bit aus dem Bildwiederholtspeicher.

Das VSYNC-Kommando (s.a. Abschn. 5.1.2.3.) legt die Funktion des V/EXT SYNC-Ein-/Ausgangs fest. Falls V/EXT SYNC (Pin 4) als Ausgang arbeitet, erzeugt der GDC das Rasterzeitverhalten für eine Display-Bildfläche sowie für andere im Master-Slave-Betrieb zu synchronisierende CRT-Controller. Ist der V/EXT SYNC-Anschluß ein Eingang, dann kann der GDC von jedem externen Vertikalsynchronimpuls synchronisiert werden.

### **3.2.3. Spezielle Darstellungsfunktionen**

Im Grafik-Betrieb können verschiedene Darstellungsfunktionen programmiert werden. Ziel ist dabei die Veränderung der Größe, Ausdehnung und Position der Darstellung auf dem Bildschirm des Displays. Nachfolgend werden die drei möglichen Darstellungsfunktionen erläutert.

#### **3.2.3.1. Ausschnittdarstellung (Windowing)**

Die Display-Bildfläche kann in der Grafik-Betriebsart in zwei Ausschnitte (bzw. Fenster), einen oberen und einen unteren, aufgeteilt werden. Diese Ausschnitte werden mit Parametern im Parameter-RAM des GDC definiert. Jeder Ausschnitt wird mit einer Startadresse und einer Ausschnittlänge (Angabe in Zeilen) spezifiziert.

Falls der zweite Ausschnitt nicht benötigt wird, sollten die Parameter des ersten Ausschnitts mit der aktiven Länge der verwendeten Display-Bildfläche übereinstimmen.

#### **3.2.3.2. Vergrößerung (Zooming)**

Ein Parameter im ZOOM-Kommando des GDC ermöglicht eine Vergrößerung der Darstellung durch wirksame Vergrößerung der Punktgröße auf dem Bildschirm des Displays. In vertikaler Richtung wird dies durch Wiederholung der gleichen Bildzeile erreicht. Die Anzahl der

Wiederholungen ist vom Parameter des Bildvergrößerungsfaktors abhängig. Gleichfalls in Abhängigkeit von diesem Parameter wird die Vergrößerung in horizontaler Richtung mit einem 'gedehnten' Display-Wortzyklus und Darstellung von weniger Worten je Zeile erreicht.

Die Taktschaltung des Schieberegisters muß vom System-Mikroprozessor so gesteuert werden, daß eine Schiebetaktfrequenz entsprechend dem Vergrößerungsfaktor erzeugt wird.

Für einen Vergrößerungsfaktor von Eins oder Zwei werden je Display-Zyklus vier Taktperioden von  $2 \times \text{WCLK}$  benötigt. Für Vergrößerungsfaktoren größer zwei werden die RMW-Zyklen auf die gleiche Länge wie die verlängerten Zoom-Display-Zyklen ausgedehnt. Die mit Zoom verlängerten Display-Zyklen werden dabei um zwei zusätzliche  $2 \times \text{WCLK}$ -Taktperioden je Stufe des Vergrößerungsfaktors länger. Die Taktfrequenz an  $2 \times \text{WCLK}$  (Pin 1) darf nicht verändert werden. Der Vergrößerungsfaktor muß innerhalb der Initialisierungsphase des GDC festgelegt werden.

Zoom

### 3.2.3.3. Schwenkfunktion (Panning)

Das Schwenken eines Bildausschnitts auf der Display-Bildfläche wird durch Veränderung der Startadresse dieses Ausschnitts erreicht. Auf diese Weise kann eine Bewegung in jede Richtung erfolgen: Vertikal auf zeilenweiser und horizontal auf wortweiser Grundlage.

### 3.2.4. Zeichnen und Schreiben

Der GDC kann durchgezogene und unterbrochene (d.h. gestrichelte, bzw. gepunktete) Geraden, Bögen, Kreise, Rechtecke, schrägstehende Rechtecke, Zeichen, schrägstehende Zeichen und ausgefüllte Rechtecke darstellen. Über DMA-Kommandos und die Lese- bzw. Schreibkommandos des GDC kann ein direkter Zugriff auf die Bit-Darstellung im Bildwiederholtspeicher erfolgen.

#### 3.2.4.1. Veränderung des Speichers

Sämtliche Zeichnungs- und Schreibprozesse werden im Bildwiederholtspeicher auf der vom Cursor festgelegten Position plazierte. In der Grafik-Betriebsart wird jedoch der Cursor nicht angezeigt. Die Position des Cursors wird durch die Ausführung der Kommandos zum Zeichnen, Schreiben sowie Lesen verändert. Der Cursor wird zu dem Bit bewegt, welches dem jeweils letzten angesprochenen Bit folgt.

Jedes Bit wird durch die Ausführung eines Read-Modify-Write-Zyklus (RMW) in den Bildwiederholtspeicher eingegeben. Normalerweise erfordern diese RMW-Zyklen vier Takt- ( $2 \times \text{WCLK}$ )-Zyklen. Falls der Vergrößerungsfaktor größer als zwei ist, erfolgt eine Erweiterung eines jeden RMW-Zyklus auf die Breite eines Display-Zyklus.

Zur Prüfung oder Veränderung von einem bis 16 Bit eines jeden Datenwortes, können während des RMW-Zyklus die Kommandos zum Schreiben (WDAT), Lesen (RDAT), DMA-Schreiben (DMAW) und DMA-Lesen (DMAR) verwendet werden. Alle anderen grafischen Zeichnungs-Steuerkommandos verändern ein Bit je RMW-Zyklus.

Das GDC-interne MASK-Register legt fest, welches Bit bzw. welche Bits des angesprochenen Wortes verändert werden sollen. Eine Eins im MASK-Register erlaubt im RMW-Zyklus die Änderung des korrespondierenden Bits im Bildwiederholtspeicher durchzuführen. Eine Null in diesem Register hindert den GDC an einer Änderung des korrespondierenden Bits. Die Maske muß mit dem MASK-Kommando und mit höherer Priorität vor der Verwendung des WDAT-

oder DMAW-Kommandos gesetzt werden. Das MASK-Register wird automatisch beim CURS-Kommando gesetzt und durch Grafikbefehle beeinflusst.

Die Änderung der Bits im Bildwiederholtspeicher kann in einer der vier nachfolgend genannten Weisen erfolgen:

1. auf Eins setzen
2. auf Null setzen
3. Bildung des Komplements
4. Ersatz durch Pattern

Falls Ersatz durch Pattern gewählt ist, werden Geraden, Bögen und Rechtecke unter Nutzung der 16-Bit-Pattern im Parameter-RAM (s.a. Abschn. 5.1.) gezeichnet.

Falls Setzen (SET), Rücksetzen (RESET) oder Komplementbildung erfolgt, wirken diese Bytes des Parameter-RAM als zusätzliche Maskierung beim Zeichnen eines Bogens, einer Geraden sowie eines Rechtecks. Jedes der 16-Bit-Segmente einer gezeichneten Geraden oder eines Bogens wird dabei mit den Pattern im Parameter-RAM verglichen. Falls dabei ein Pattern-Bit auf Eins steht, wird in Abhängigkeit der festgelegten Bedingungen das korrespondierende Bit im Bildwiederholtspeicher entweder gesetzt, rückgesetzt oder komplementiert. Ist dagegen das Pattern-Bit im Parameter-RAM auf Null gesetzt, erfolgt keine Veränderung der Bits im Bildwiederholtspeicher.

Falls der Ersatz der Bits durch den Aufruf von Pattern erfolgt (Pattern Mode), veranlassen die grafischen Zeichnungs- und Ausfüllbefehle, daß die 8 \* 8-Pattern (im Parameter-RAM Bytes 8...15) direkt in den Bildwiederholtspeicher an die richtige Stelle geschrieben werden. Beim Setzen, Rücksetzen oder bei der Komplementbildung wirken die 8 \* 8-Pattern im Parameter-RAM wieder als zusätzliche Maske für grafische Zeichen oder für Kommandos zum Ausfüllen von Flächen. Ist das entsprechende Bit im Parameter-RAM auf Eins gesetzt, kann eine Änderung des Bits im Bildwiederholtspeicher erfolgen. Falls das Bit im Parameter-RAM auf Null gesetzt ist, wird das Bit im Bildwiederholtspeicher nicht verändert.

Diese Betriebsarten werden durch ausgeben eines WDAT-Kommandos ohne Parameter ausgewählt, bevor Grafik-Kommandos aufgerufen werden. Die Pattern des Parameter-RAM haben keinen Einfluß auf die Arbeitsweise der WDAT-, RDAT-, DMAW- oder DMAR-Kommandos.

#### **3.2.4.2. Befehle zum Lesen und Schreiben**

Nachdem die grundsätzlichen Eigenschaften des GDC in der Grafik-Betriebsart festgelegt sind und das Parameter-RAM geladen wurde, sind die letzten Zeichnungsparameter mit dem FIGS-Kommando (s.a. Abschn. 5.1.) zu laden. Dabei legt der erste Parameter die Richtung in die gezeichnet werden soll sowie den darzustellenden Zeichnungstyp fest. Diesem Parameter folgen dann ein bis fünf weitere Parameter, die von der Art des darzustellenden Zeichens abhängen. Der Richtungsparameter (direction parameter) legt einen der acht Oktanten fest, in dem gelesen oder gezeichnet werden soll.

Die Wirkung der Zeichnungsrichtung auf die verschiedenen möglichen Darstellungsarten sind in Tafel 4.3 wiedergegeben. Der Bewegungsablauf im Bildwiederholtspeicher bei RDAT-, WDAT-, DMAR- und DMAW-Kommandos ist in der Spalte >DMA< dargestellt.

Die anderen zum Lesen oder Zeichnen von Darstellungen notwendigen Parameter sind in Tafel 3.1 zu finden.

Nachdem die oben beschriebene Parameterfestlegung erfolgt ist, werden die Zeichnungsprozesse für Gerade, Bogen, Kreis, Rechteck sowie schrägstehendes Rechteck mit

dem FIGD-Kommando (s.a. Abschn. 5.1.) ausgelöst.

Die Zeichnungsprozesse für Zeichen, schrägstehendes Zeichen, Ausfüllung von Flächen sowie Ausfüllung von schrägstehenden Flächen werden mit dem GCHRD-Kommando (s.a. Abschn. 5.1.) gestartet.

DMA-Übertragungen werden mit den Kommandos für DMA-Lesebetrieb (DMAR-Kommando) und DMA-Schreibbetrieb (DMAW-Kommando) durchgeführt.

Die 'gewöhnlichen' Operationen zum Lesen und Schreiben der Daten in den Bildwiederhol-speicher werden mit dem Aufruf des RDAT-Kommandos sowie durch des Einschreiben eines Parameters nach dem WDAT-Kommando (s.a. Abschn. 5.1.) ausgelöst.

Bei der Ausfüllung einer Rechteckfläche wird wiederholt das 8 \* 8-Pattern des grafischen Zeichens abgetastet, was dem Ausfüllen einer Rechteckfläche entspricht. Ist dabei die Größe des Rechtecks kein ganzzahliges Vielfaches eines 8 \* 8-Pixels, verkürzt der GDC automatisch an der Seite des Pattern, die am weitesten vom Startpunkt entfernt ist.

Die Eigenschaften der grafischen Zeichendarstellung kann durch den Vergrößerungsfaktor im GCHRD-Kommando (s.a. Abschn. 5.1.) verändert werden. Dieser Zoom-Faktor (zoom write factor) kann zwischen 1...16 (d.h. für den Parameter zwischen 0...15) festgelegt werden. Jeder Punkt (d.h. jedes Bit eines Wortes im Bildwiederhol-speicher) wird im Speicher in Abhängigkeit der festgelegten Zeichnungsrichtung horizontal und ver-tikal wiederholt. Die Anzahl der Wiederholungen ist dabei durch den Vergrößerungsfaktor festgelegt.

Das WDAT-Kommando kann zur schnellen Ausfüllung großer Flächen des Bildwiederhol-speichers mit dem gleichen Wert benutzt werden. Dabei wird die Maske generell auf Eins gesetzt und das niederwertigste Bit des WDAT-Parameters ersetzt alle Bits des geschriebenen Wortes.

Zeichnungsart	DC	D	D2	D1	DM
Anfangswert *	0	8	8	-1	-1
Gerade	$ \Delta I $	$2 \Delta D  -  \Delta I $	$2( \Delta D  -  \Delta I )$	$2 \Delta D $	-
Bogen **	$r \sin \phi \uparrow$	$r-1$	$2(r-1)$	-1	$r \sin \theta \downarrow$
Rechtecke	3	A-1	B-1	-1	A-1
Flächenfüllung	B-1	A	A	-	-
Grafisches Zeichen ***	B-1	A	A	-	-
Datenschreiben	W-1	-	-	-	-
DMAW	D-1	C-1	-	-	-
DMAR	D-1	C-2	$(C-2)/2^{++}$	-	-
Datenlesen	W	-	-	-	-

#### Zeichenerklärung:

- \* Anfangswerte für die verschiedenen Parameter werden während des Ablaufs des FIGS-Kommandos geladen
- \*\* Kreise werden mit acht Bögen gezeichnet; jeder Bogen spannt über  $45^\circ$ , d.h.  $\sin \phi = 1/\sqrt{2}$  und  $\sin \theta = 0$ .
- \*\*\* Grafikzeichen sind ein Spezialfall einer bitorientierten Flächenfüllung, bei der B und A  $\leq 8$  ist; falls A = 8, brauchen D und D2 nicht geladen zu werden
- 1 Entspricht einem "Eins"-Wert  
Alle Zahlen sind im Dezimalsystem dargestellt. Der GDC verarbeitet nur Binärwerte; wo benötigt auch Werte in Zweier-Komplementdarstellung.
- Für diese Parameter werden keine Parameterbytes an den GDC gesendet
- I Größerer Teil der X- oder Y-Achse
- D Kleinerer Teil der X- oder Y-Achse
- r Kurvenradius; Angabe mit Anzahl der Pixel
- $\phi$  Winkel von der Hauptachse zum Ende des Bogens ( $\phi \leq 45^\circ$ )
- $\theta$  Winkel von der Hauptachse zum Anfang des Bogens ( $\theta \leq 45^\circ$ )
- $\uparrow \downarrow$  Aufrundung oder Abrundung auf nächsthöhere oder -niedrigere ganze Zahl
- A Anzahl der Pixel in die festgelegte initialisierte Richtung
- B Anzahl der Pixel in Richtung des rechten Winkels zur festgelegten initialisierten Richtung
- W Anzahl der zuzugreifenden Worte
- C Anzahl der Bytes, die in die festgelegte initialisierte Richtung übertragen werden sollen; (bei Wortübertragung werden zwei Bytes pro Wort übertragen)
- D Anzahl der Worte, auf die in Richtung des rechten Winkels zur festgelegten initialisierten Richtung zugegriffen werden soll
- DC Zeichnungszählparameter (Drawing Count Parameter); dieser Parameter ist immer die um eins verringerte Zykluszahl der im Bildwiederhol-speicher auszuführenden RMW-Zyklen
- DM Punkte, die während des Zeichnungsprozesses einer Bogendarstellung maskiert werden
- ++ wird nur beim Lesen des Wortes benötigt

Tafel 3.1: Zeichnungsparameter des GDC U82720



### 3.3. Zeichen-Betriebsart

Die Zeichen-Betriebsart (character mode) ermöglicht die komfortable Darstellung von Zeichen, vorrangig in alphanumerischen Displays. Nachfolgend werden Funktion und Eigenschaften dieser Betriebsart erläutert.

#### 3.3.1. Speicherorganisation

In der Zeichen-Betriebsart ist der Bildwiederholtspeicher für eine  $8K \times 13$  Bit-Zeichendarstellung organisiert. Die Benutzung des 'gedehnten' Zyklus (Wide Display Cycle) ist für Zeichen bis 26 Bit möglich. Die Funktion des Wide-Display-Zyklus ist ansonsten identisch zur Funktion in der Grafik-Betriebsart (s.a. Abschn. 3.2.1.). Mit dem WD-Flag des Parameter-RAM erfolgt für die jeweilige Display-Bildfläche der Aufruf dieses Zyklus. Der Bildwiederholtspeicher kann größer als die eigentliche Display-Bildfläche sein. Die Breite des Displays (d.h. die Anzahl der darzustellenden Zeichen) ist ein Parameter des RESET-Kommandos. Die Breite des Bildwiederholtspeichers (Anzahl der zu speichernden Zeichen) ist ein Parameter des PITCH-Kommandos (s.a. Abschn. 5.1.). Die Höhe der Display-Bildfläche (d.h. die Anzahl der darzustellenden Zeilen) ist ebenfalls ein Parameter des RESET-Kommandos. Die Größe des zugehörigen Bildwiederholtspeichers ist entsprechend dem Quotienten aus der Anzahl der insgesamt darzustellenden Worte und der Wortanzahl in horizontaler Richtung (Pitch Factor) festzulegen.

In der Zeichen-Betriebsart arbeitet das Display im wesentlichen genauso, wie in der Grafik-Betriebsart. In der Zeichen-Betriebsart werden jedoch die vom Bildwiederholtspeicher gelesenen Daten sowohl zur Ansteuerung einer Attribut-Logik als auch zur Ansteuerung eines Zeichengenerators benutzt. In der Zeichen-Betriebsart werden die Adreßbits  $AD_{13} \dots AD_{15}$  und A16 zu Ausgängen des Zeilenzählers. Damit wird dann die richtige Zeile im Zeichengenerator ausgewählt. Der Ausgang A16 liefert das höchstwertigste Bit des Zeilenzählers. A17 wird als Ausgang für den Cursor und als höchstwertiges Bit des Zeilenzählers benutzt.

#### 3.3.2. Zeitverhalten der Bilddarstellung

Das Zeitverhalten der Bilddarstellung ist in der Zeichen-Betriebsart identisch zur Grafik-Betriebsart. Zusätzlich wird die Adresse A17 als Cursor-Ausgang verwendet.

Die Eigenschaften des Cursors in der Zeichen-Betriebsart werden durch die Parameter des CCHAR-Kommandos (s.a. Abschn. 5.1.) festgelegt. Dabei legt ein Bit die Freigabe (enable) bzw. Sperrung (disable) des Cursors fest (DC-Flag in P1 - nicht zu verwechseln mit dem DC-Wert !!). Die Höhe des Cursors ist durch die Auswahl der oberen und unteren Zeile, in welcher der Cursor erscheinen soll, programmierbar. Gleichfalls kann dabei die Blinkfrequenz programmiert werden.

Der Parameter wählt dabei die Anzahl der Raster-Zeiteinheiten, für die der Cursor aktiv oder inaktiv sein wird, aus. Dabei ergibt sich bei einem 50%-Taktzyklus das Blinken des Cursors mit der doppelten Periode, die mit dem Parameter festgelegt wurde.

Der Cursor-Ausgang (d.h. A17 am GDC) liefert dem Zeilenzähler das vierte Bit, das zehn Taktzyklen nach der fallenden Flanke von HSYNC gültig ist.



### **3.3.3. Spezielle Darstellungsfunktionen**

#### **3.3.3.1. Ausschnittdarstellung (Windowing)**

Eine Display-Bildfläche kann bei Verwendung eines GDC U82720 in der Grafik-Betriebsart in ein bis vier Ausschnitte (Fenster) aufgeteilt werden. Die Festlegung der Ausschnitte erfolgt mit Parametern, die in das Parameter-RAM des GDC eingegeben werden.

Jeder Ausschnitt wird durch eine Startadresse und die Anzahl der Zeilen für die Länge des Ausschnitts definiert. Falls die Ausschnittdarstellung nicht benötigt wird, sollte die Länge des ersten Ausschnitts mit der Länge der Display-Bildfläche übereinstimmen.

#### **3.3.3.2. Vergrößerung und Schwenkfunktion**

Die Durchführung der Vergrößerung (Zooming) und das Bewegen bzw. Schwenken (Panning) des Bildes weist keine Unterschiede zur Grafik-Betriebsart (s.a. Abschn. 3.2.3.) auf.

#### **3.3.4. Zeichnen und Schreiben**

Der GDC kann Zeichen, kodiert bis zu 13 Bit, in oder aus dem Bildwiederholtspeicher Schreiben oder Lesen. Sämtliche Lese- und Schreiboperationen finden an der, durch den Cursor festgelegten Position im Bildwiederholtspeicher statt. Die Position, an der der Cursor plazierte werden soll, wird durch Aufruf des CURD-Kommandos (s.a. Abschn. 5.1.) festgelegt.

Die Bewegung des Cursors an beliebige Stellen im Bildwiederholtspeicher-RAM erfolgt mit dem CURS-Kommando (s.a. Abschn. 5.1.5.2.). Dabei wird die Plazierung des Cursors vorgenommen. Die Plazierung des Cursors wird auch mit der Ausführung von Lese- bzw. Schreibkommandos für Zeichen verändert.

Jedes Zeichen wird innerhalb eines RMW-Zyklus geschrieben bzw. gelesen. Die Inhalte des Mask-Registers legen dabei fest, welche Bits des Zeichens verändert werden. Dieses Register kann ebenso dazu benutzt werden, die Kodierungen der Zeichen ohne Veränderung der Attribut-Bits (und umgekehrt), zu verändern. Der Ersatz durch Pattern, das Setzen bzw. Rücksetzen sowie die Komplementbildung entsprechen dem Ablauf in der Grafik-Betriebsart (s.a. Abschn. 3.2.).

Eine Ausnahme gilt es zu beachten: Die Pattern des Parameter-RAM werden nicht verwendet. Die benötigten Pattern werden durch die Pattern des WDAT-Kommandos (s.a. Abschn. 5.1.) gebildet. Das FIGS-Kommando (s.a. Abschn. 5.1.) muß in der Zeichen-Betriebsart zur Richtungsfestlegung der Cursorbewegung bei Lese- oder Schreiboperationen benutzt werden. Das FIGD- und das GCHRD-Kommando werden nicht benutzt.

### **3.4. Misch-Betriebsart**

Die Misch-Betriebsart bietet die häufig benötigte Möglichkeit der gleichzeitigen Darstellung von Zeichen und grafischen Elementen. Die Besonderheiten und Eigenschaften dieser speziellen Betriebsart des GDC U82720 werden nachfolgend vorgestellt.

### 3.4.1. Speicherorganisation

In der Misch-Betriebsart (Mixed Mode) ist der Bildwiederholtspeicher in zwei Bänke zu je 64K-Worte, jedes Wort zu 16 Bit, organisiert. Die Festlegung einer 32 Bit-Wortbreite ist mit programmiertem Wide-Display-Zyklus (Programmierung des WD-Flag im Parameter-RAM) möglich. Dabei ist der Wide-Display-Zyklus im 'gedehnten' Betrieb der Misch-Betriebsart identisch zur Grafik-Betriebsart (s.a. Abschn. 3.2.1.).

Die Höhe und Breite der Display-Bildfläche ist mit den gleichen Parametern des RESET- bzw. SYNC-Kommandos programmierbar wie in der Grafik- bzw. Zeichen-Betriebsart. Die in Worten anzugebende Breite des Bildwiederholtspeichers ist ein Parameter des PITCH-Kommandos. Die Größe des Bildwiederholtspeichers wird auch bei der Zeichen-Betriebsart vom Quotienten aus der Anzahl der insgesamt darzustellenden Worte und der Wortanzahl pro Zeile in horizontaler Richtung (Pitch Factor) bestimmt.

Das sog. Bild-Betriebsartensignal (Image Mode Signal) wird dann zur Umschaltung der externen Hardware des GDC zwischen Grafik- und Zeichen-Betriebsart genutzt.

In einem Ausschnitt (bzw. Fenster) zur Grafikdarstellung auf der Display-Bildfläche arbeitet der GDC genauso wie in der normalen Grafik-Betriebsart. Allerdings ist dabei der zu Verfügung stehende Bildwiederholtspeicher wesentlich kleiner: Es stehen nur 64 KWorte (gegenüber 256 KWorte in der normalen Grafik-Betriebsart) zur Verfügung.

Im Zeichen-Ausschnitt der Display-Bildfläche vollführt der GDC die gleichen Funktionen wie in der normalen Zeichen-Betriebsart, wenn ein externer Zeilenzähler vorhanden ist.

Der GDC U82720 ermöglicht in der Misch-Betriebsart sowohl die Bereitstellung eines Cursors als auch eines Blinksignals für Attribute.

### 3.4.2. Zeitverhalten der Bilddarstellung

In der Misch-Betriebsart wird auf jedes Wort der Grafik-Bildfläche nacheinander zweimal zugegriffen. Die Anzahl der aktiven Worte je Zeile (AW-Parameter), die mit dem RESET- bzw. SYNC-Kommando festgelegt wird, ist dabei auf das doppelte seines Normwertes zu setzen. Das Signal zum Laden des externen Video-Schieberegisters muß während des zusätzlichen Zugriffszyklus unterdrückt werden.

Mit A17 wird während der Dauer des Horizontalsynchronimpulses (HSYNC) signalisiert, ob die nachfolgende Rasterzeile als bitweise Grafikdarstellung oder als Zeichen interpretiert werden soll. Während des Display-Zyklus gibt A17 für die Zeichenfläche die Cursorposition aus, während A16 das Blinkzeitverhalten für das Attribut ausgibt.

A16 liefert ein H-aktives Zeilenzähler-Rücksetzsignal, das zehn Taktzyklen nach der fallenden Flanke des Horizontalsynchronsignals (HSYNC) gültig ist. Während des aktiven Display-Zyklus einer Zeile liefert A16 einen Blinktakt für eine externe Attributschaltung. Die Frequenz dieses Signals entspricht der Hälfte der Blinkfrequenz des Cursors. Dabei ist ein Tastverhältnis des Hell-/Dunkelzyklus von Drei zu Eins festgelegt.

A17 erzeugt ein Signal zur Auswahl der Fläche mit Grafik- bzw. Zeichen-Darstellungen, das ebenfalls zehn Taktzyklen nach der fallenden Flanke des HSYNC-Signals gültig ist. Innerhalb des aktiven Display-Zyklus liefert A17 das Cursor-Signal. Das Zeitverhalten und die Eigenschaften des Cursors werden wie in der Zeichen-Betriebsart festgelegt.

### **3.4.3. Spezielle Darstellungsfunktionen**

#### **3.4.3.1. Ausschnittdarstellung**

Der GDC ermöglicht in der Misch-Betriebsart die Darstellung von zwei Ausschnitten auf der Display-Bildfläche. Diese Ausschnitte können entsprechend der Festlegung zweier Flags im Parameter-RAM (Image Mode Flag in RA-3 und RA-7) unabhängig voneinander entweder in der Zeichen- oder in der Grafik-Betriebsart angesteuert werden. Die Position der Ausschnitte im Bildwiederholpeicher sowie deren Größe sind ebenfalls von Parametern (z.B. RA-0...RA-3) des Parameter-RAM abhängig. Der Aufbau des Parameter-RAMs für die Misch-Betriebsart ist im Bild 4.5 beschrieben.

#### **3.4.3.2. Vergrößerung und Schwenkfunktion**

Die Funktionen der Kommandos in der Misch-Betriebsart für Vergrößerung und Schwenkbewegung (Zoom und Pan) einer Display-Darstellung entsprechen den Befehlen in der Zeichen- und Grafik-Betriebsart.

#### **3.4.4. Zeichnen und Schreiben**

Das Zeichnen und Schreiben in der Misch-Betriebsart erfolgt auf die gleiche Weise wie in der Zeichen- bzw. Grafik-Betriebsart. Als Erweiterung in der Misch-Betriebsart ermöglicht die Programmierung eines Parameters des FIGS-Kommando (s.a. Abschn. 5.1) die Anzeige grafischer Zeichnungsprozesse durch die Programmierung des Flags für grafische Zeichnung (Graphics Drawing Flag in P3). Damit wird das Bild-Betriebsartensignal zur Selektion der entsprechenden RAM-Speicherbank gesetzt.

## 4. Programmierung des GDC U82720

### 4.1. Allgemeines

Für den Zugriff auf das Status-Register und den FIFO-Buffer des GDC werden zwei Adressen des System-Mikroprozessors belegt. Die Unterscheidung der Kommandos und Parameter, die in den FIFO des GDC geschrieben werden, erfolgt mit dem Adreßbit A0. Gleichfalls erfolgt mit dieser Adreßleitung die Selektion für das Lesen der Kommandos und Parameter im FIFO bzw. im Status-Register. Im Bild 4.1 ist der Aufbau der Mikroprozessor-Businterface-Register dargestellt.

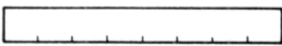
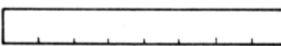
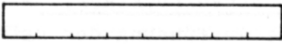
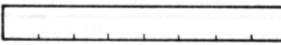
A0	Lese-Operation	Schreib - Operation
0	Status- Register 	Parameter in FIFO 
1	FIFO-lesen 	Kommando in FIFO 

Bild 4.1: Aufbau der Mikroprozessor-Businterface-Register des GDC

Die Kommandos für den GDC werden in der Form eines Kommandobytes, gefolgt von einer für die Festlegung der auswählbaren Details des Kommandos notwendigen Serie von Parameter-Bytes, gesendet. Der Kommandoprozessor dekodiert die Kommandos, untersucht die Parameter, lädt diese in die entsprechenden Register des GDC und löst die erforderlichen Operationen aus. Die Kommandos lassen sich in fünf Kategorien einteilen. Tafel 4.1 zeigt eine zusammenfassende Übersicht.

### 4.2. Statusregisterflags

Im Bild 4.2 werden die Flags SR-0...SR-7 des Statusregisters dargestellt. Nachfolgend wird die Wirkung der einzelnen Flags im Statusregister beschrieben.

#### **SR-7 : Lichtstifterkennung** (Light Pen Detect)

Für den Fall, daß dieses Flag auf Eins steht, befindet sich eine eingeschlungene Adresse des Lichtstifts im Lichtstiftadreßregister (LAD-Register), welches vom System-Mikroprozessor gelesen werden kann.

Das SR-7-Flag wird im Ergebnis des Lichtstift-Lesekommandos (LPRD) wieder rückgesetzt, nachdem die drei Bytes der Lichtstiftadresse in den FIFO-Buffer übergeben worden sind.

#### **SR-6 : Horizontalaustastung** (Horizontal Blanking Activ)

Ein gesetztes SR-6-Flag bedeutet, daß der Zeilenrücklauf gerade dunkelgetastet wird.

## Status - Register-Flags

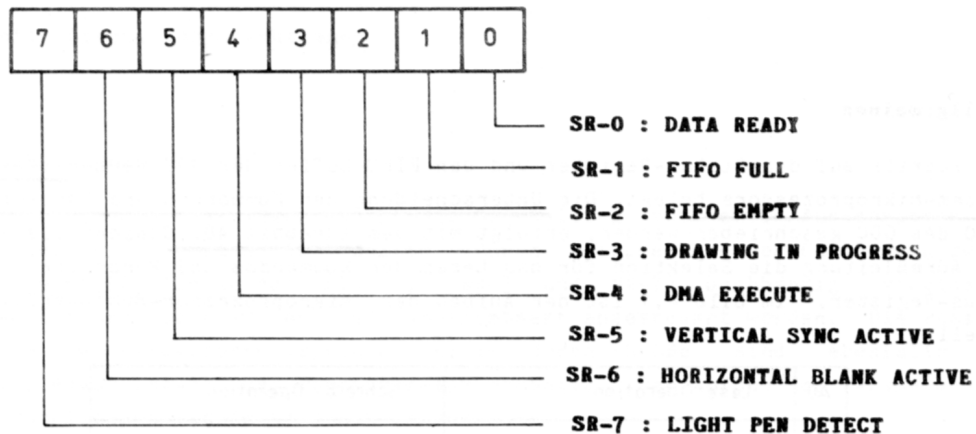


Bild 4.2: Statusregister (SR)

### **SR-5 : Vertikalsynchronisation (Vertical Sync)**

Während der Vertikalsynchronisation für den Strahlrücklauf, wird dieses Flag auf Eins gesetzt. Mittels dieses Flags wird die Ausführung der Befehle zur Änderung des Bildformats mit der Vertikal- und Horizontalaustastdauer koordiniert. Damit wird das Auftreten von Bildstörungen verhindert.

### **SR-4 : DMA-Ausführung (DMA Execute)**

Dieses Flag steht während einer Datenübertragung im DMA-Betrieb auf Eins.

### **SR-3 : Zeichnungsvorgang (Drawing in Progress)**

Während der GDC eine grafische Darstellung zeichnet, steht SR-3 auf Eins.

### **SR-2 : FIFO-Leer (FIFO Empty)**

Eine Eins bei diesem Flag zeigt an, daß der FIFO-Buffer des GDC die vorher gesendeten Kommandos und Parameter verarbeitet hat. Dieses Flag und das SR-1-Flag koordinieren den Zugriff des System-Mikroprozessors auf den FIFO des GDC.

### **SR-1 : FIFO-Voll (FIFO Full)**

Eine Eins zeigt den gefüllten FIFO-Buffer des GDC an. Eine Null bedeutet, daß noch Platz für mindestens ein Byte im FIFO vorhanden ist. Dieses Flag muß immer getestet werden, bevor in den FIFO des GDC geschrieben wird.

### **SR-0 : Daten-Bereitschaft (Data Ready)**

Eine Eins dieses Flags bedeutet, daß ein Byte durch den System-Mikroprozessor eingelesen werden kann. Dieses Flag muß vor jeder Leseoperation getestet werden. SR-0 kippt während der Datenübertragung vom FIFO zum Mikroprozessor-Interfaceregister auf Null.

## 4.3. FIFO-Operation und Kommandoprotokoll

Der FIFO-Buffer des GDC ermöglicht den Kommandodialog mit dem System-Mikroprozessor. Der Informationsfluß erfolgt mit einer Halb-Duplex-Technik. Dabei wird der 16stufige FIFO-Puffer für den Datenaustausch in beiden Richtungen benutzt. Die Richtung des FIFO-Puffer wird durch den System-Mikroprozessor mit den Kommandos des GDC festgelegt. Der System-

<u>VIDEO-STEUERKOMMANDOS (Video Control Commands)</u>	
<u>RESET</u>	(Reset Command) Rücksetzen des GDC in den Freilaufbetrieb (Idle State)
<u>SYNC</u>	(Sync Format Specify Command) Festlegung des Formats der Displayfläche
<u>CCHAR</u>	(Cursor and Character Characteristics Command) Festlegung der Spaltenhöhe für Cursor und Zeichen
<u>DARSTELLUNGS-STEUERKOMMANDOS (Display Control Commands)</u>	
<u>START</u>	(Start Command) Beendigung des Freilaufbetriebs und Freigabe des Displayschirms
<u>BCTRL</u>	(Display Blanking Control Command) Steuerung der Freigabe zur Dunkelastung des Displays
<u>ZOOM</u>	(Zoom Factor Specify Command) Festlegung des Vergrößerungsfaktors für die grafische Darstellung
<u>CURS</u>	(Cursor Position Specify Command) Steuerung der Position des Cursors im Bildwiederhol-speicher
<u>PRAM</u>	(Parameter RAM Load Command) Definition der Startadresse, der Länge(n) der Displayfläche(n), sowie der acht Bytes für ein grafisches Zeichen
<u>PITCH</u>	(Pitch Specification Command) Festlegung der Breite des Bildwiederhol-speichers in X-Richtung
<u>ZEICHNUNGS-STEUERKOMMANDOS (Drawing Control Commands)</u>	
<u>WDAT</u>	(Write Data Command) Einschreiben von Datenwörtern bzw. -bytes in den Bildwiederhol-speicher
<u>MASK</u>	(Mask Register Load Command) Festlegung des Inhalts des Maskenregisters
<u>FIGS</u>	(Figure Drawing Parameters Specify Command) Festlegung der Parameter für den Zeichnungsprozessor
<u>FIGD</u>	(Figure Draw Start Command) Zeichnung der vorher festgelegten Darstellung
<u>GCHRD</u>	(Graphics Character Draw and Area Filling Start Command) Zeichnung eines grafischen Zeichens auf den Displayschirm
<u>SPEICHERDATEN-LESEKOMMANDOS (Memory Data Read Commands)</u>	
<u>RDAT</u>	(Read Data from Display Memory) Lesen von Datenwörtern bzw. -bytes aus dem Bildwiederhol-speicher
<u>CURD</u>	(Cursor Address Read Command) Lesen der aktuellen Position des Cursors
<u>LPRD</u>	(Light Pen Address Read Command) Lesen der Lichtstiftadresse
<u>DMA-STEUERKOMMANDOS (DMA Control Commands)</u>	
<u>DMAR</u>	(DMA Read Request Command) Anforderung einer DMA-Lese-Übertragung aus dem Bildwiederhol-speicher des GDC
<u>DMAW</u>	(DMA Write Request Command) Anforderung einer DMA-Schreib-Übertragung in den Bildwiederhol-speicher des GDC

Tafel 4.1: Kommandoübersicht

Mikroprozessor koordiniert diese Übertragungen durch Testen der entsprechenden Flags im Status-Register (SR-0...SR-7).

Das vom GDC benutzte Kommandoprotokoll erfordert die Unterscheidung des ersten Bytes einer Kommandofolge von den nachfolgenden Bytes. Das erste Byte enthält den Operationscode, die restlichen Bytes die Parameter.

Das Einschreiben in den GDC veranlaßt den FIFO einen Flag-Wert parallel zum Datenbyte zu speichern, um zu kennzeichnen, ob das Byte in die Kommando- oder in die Parameteradresse geschrieben wurde. Der Kommandoprozessor im GDC testet das Bit, wenn er die Eingaben in den FIFO auswertet.

Der Empfang eines Kommandobytes durch den Kommandoprozessor des GDC markiert das Ende der vorausgegangenen Operation. Die Folge der mit einem Kommando gelieferten Parameterbytes wird sofort beim Empfang des nachfolgenden Kommandobytes abgebrochen. Eine Leseoperation vom GDC zum System-Mikroprozessor kann jederzeit durch das nachfolgende Kommando beendet

werden.

Der System-Mikroprozessor steuert die Richtungsänderung des FIFO-Buffers. Kommandos, die in den GDC geschrieben werden sollen, versetzen den FIFO in den Schreibbetrieb, sofern der noch nicht eingenommen war. Falls der FIFO sich im Lesebetrieb befand, sind irgendwelche FIFO-Lesedaten zum Zeitpunkt der Änderung der Betriebsart verloren.

Kommandos, die eine Antwort des GDC erforderlich machen (z.B. beim RDAT-, CURD- und LPRD-Kommando) versetzen den FIFO in den Lesebetrieb, nachdem der Kommando-Prozessor des GDC das Kommando interpretiert hat. Kommandos und Parameter, die nach einem den Lesebetrieb aktivierenden Kommando folgen, werden ignoriert, sobald die FIFO-Richtung umgekehrt ist.

#### **4.4. Read-Modify-Write-Zyklus**

Die Datenübertragung zwischen dem GDC und dem Bildwiederholtspeicher erfolgt unter Verwendung des RMW-Zyklus (Read-Modify-Write-Memory-Cycle). Die dabei notwendigen vier Taktzyklen werden zu nachfolgenden Operationen verwendet:

- Ausgabe der Adresse
- Lesen der Daten vom Speicher
- Veränderung der Daten
- veränderte Daten auf die ursprüngliche Speicheradresse zurückschreiben

Der RMW-Speicherzyklus wird für alle Wechselwirkungen mit dem Bildwiederholtspeicher, einschließlich der DMA-Übertragung, benutzt. Ausnahmen sind die Bilddarstellung in zwei Taktperioden des Display-Zyklus und die Refresh-Zyklen der dynamischen RAMs. Die Vorgänge, die während der Veränderung der Daten im RMW-Zyklus innerhalb des dritten Taktzyklus stattfinden, werden nachfolgend erläutert.

Die interne Logik des GDC verfügt über drei Hauptelemente:

- Patternregister (Pattern Register)
- Maskenregister (Mask Register)
- 16-Bit-Logikeinheit (Logic Unit)

Das Patternregister enthält die Datenpattern, die in den Bildwiederholtspeicher übertragen werden sollen. Es wird durch das WDAT-Kommando oder während des Zeichnungsprozesses vom Parameter-RAM geladen. Der Inhalt des Maskenregisters (MASK-Register) bestimmt, welche Bits der gelesenen Daten verändert werden sollen.

In Abhängigkeit vom Inhalt dieser beiden Register führt die 16-Bit-Logikeinheit des GDC die ausgewählten Operationen, wie Patternersatz, Komplementierung, Setzen oder Löschen an Daten aus, die aus dem Bildwiederholtspeicher gelesen wurden.

Die Inhalte des Patternregisters werden mit den Inhalten des Maskenregisters über eine UND-Funktion verknüpft. Dies erfolgt bitweise zur Freigabe der aktuellen Veränderung der vom Speicher gelesenen Daten. Für das Zeichnen einer Grafik wird zu einem Zeitpunkt ein Bit des Patternregisters mit dem Inhalt des Maskenregisters verknüpft. Ergibt die UND-Verknüpfung dieses Bits im Maskenregister eine Eins, erfolgt durch die Logikeinheit des GDC eine Änderung des zugehörigen einzelnen Pixels.

Für das nächste Pixel in der Darstellung wird das nachfolgende Bit des Patternregisters ausgewählt und das Bit des Maskenregisters zur Positionserkennung des Pixels im Datenwort weiterbewegt.

Der Pointer mit der auszuführenden Wortadresse (EAD) ist immer so festzulegen, wie es die Adressierung des Wortes erfordert, welches das nächste Pixel enthält. In der Zeichen-Betriebsart werden sämtliche Bits des Patternregisters parallel benutzt, um die



entsprechenden Bits des zu verändernden Datenwortes zu bilden. Weil die Zeichen-Codeworte, entgegen der bitweisen Verwendung in der Grafik-Betriebsart, parallel benutzt werden, ist die Änderung von einzelnen oder allen Bits eines Speicherwortes innerhalb eines einzigen RMW-Zyklus möglich. Das Maskenregister muß dabei in den Positionen mit Eins geladen werden, bei denen eine Änderung gestattet ist.

Das Laden des Maskenregisters kann auf zwei verschiedenen Wegen erfolgen: In der Grafik-Betriebsart, enthält das CURS-Kommando zur Festlegung der Punktadresse ein vier Bit großes Feld. Der Kommandoprozessor des GDC konvertiert aus diesem Parameter ein 1-aus-16-Format und benutzt das Maskenregister zur Zeichnung einer Darstellung. Vollständige 16 Bit können unter Benutzung des MASK-Kommandos in das Maskenregister geladen werden (s.a. Abschn. 5.1.).

In der Grafik-Betriebsart erweist sich das Laden der 16 Bit mit dem MASK-Kommando besonders dann als vorteilhaft, wenn alle Pixel eines Wortes auf den gleichen Wert gesetzt werden sollen. Die Logikeinheit des GDC kombiniert die vom Bildwiederhol-speicher gelesenen Daten mit dem Patternregister und dem Maskenregister, um die Daten zu generieren, die in den Bildwiederhol-speicher zurückgeschrieben werden sollen. Dabei kann eine von vier Operationen ausgewählt werden: Ersetzen, Komplementbildung, Löschen und Setzen. Falls das betreffende Bit der Maske auf Null gesetzt ist, wird das zugehörige Bit der Lesedaten unverändert zum Speicher zurückgesendet. Wenn jedoch das Bit der Maske auf Eins steht, kann die Änderung erfolgen.

Mit der Operation für Ersetzen übernehmen die geänderten Daten, entsprechend den zur Veränderung freigegebenen Bits, direkt die Plätze der Lesedaten.

Bei der Ausführung der Operationen für Komplementbildung, Löschen und Setzen der Daten bedeutet eine Null in den zu verändernden Daten, daß eine Zurücksendung des gelesenen Datenbits in den Bildwiederhol-speicher erfolgt. Die Eins bedeutet dagegen, daß die o.g. Operationen ausgelöst werden und dann auf die Bitpositionen einwirken können, die mit den entsprechend gesetzten Bits im Maskenregister zur Änderung freigegeben sind.

#### **4.5. Ablauf der Bilddarstellung**

Der GDC U82720 zeichnet grafische Darstellungen mit einer Rate von einem Pixel je RMW-Zyklus im Bildwiederhol-speicher. Ein vollständiger RMW-Zyklus benötigt dafür vier Taktzyklen der Taktfrequenz  $2 \times \text{WCLK}$ . Bei einer Taktfrequenz von 4 MHz entspricht dies einer Zeit von 1000 ns. Während des RMW-Zyklus berechnet der GDC gleichzeitig die Adresse und die Position des nächsten zu zeichnenden Pixels.

Der Ablauf der grafischen Darstellung hängt von der Art der Adressierung des Bildwiederhol-speichers ab. Gruppen von 16 horizontal benachbarten Pixeln bilden die vom GDC benutzten 16-Bit-Worte. Der Bildwiederhol-speicher ist als linearer Adreßraum für diese Worte zu organisieren. Die Adressierung von einzelnen Pixeln erfolgt mit der GDC-internen RMW-Logik.

Während des Zeichnungsprozesses findet der GDC das nächste zu zeichnende Pixel der Darstellung in einem von acht der am nächsten zum letzten gezeichneten Pixel liegenden Nachbar-Pixel. Der GDC kennzeichnet jede der acht Richtungen, beginnend mit der Geraden nach unten und im Gegenuhrzeigersinn weiterzählend, mit einer Zahl zwischen Null und Sieben.

Die Zeichnung einer Darstellung erfordert die richtige Manipulation der Adresse und der Position des Pixel-Bits, um das nächste Pixel, Übereinstimmend mit der Zeichnungsrichtung, festzulegen. Um zu einem Wort zu gelangen, das sich oberhalb oder unterhalb der augenblicklichen Position befindet, muß die Anzahl der Worte pro Zeile im Bildwiederhol-speicher subtrahiert werden. Der dazu erforderliche Parameter wird im Pitch-Faktor

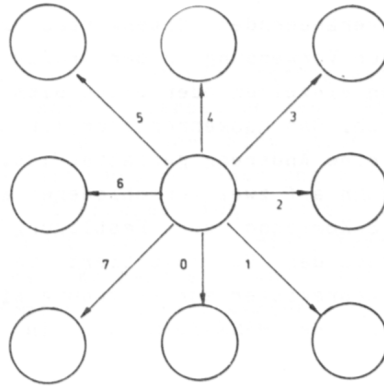


Bild 4.3: Zeichnungsrichtungen

festgelegt.

Die Bewegung zu einem benachbarten Wort erfolgt, sobald der Punktheadpointer dAD das niederwertigste oder höherwertigste Bit des Maskenregisters erreicht hat und die Bewegung in gleicher Richtung fortgesetzt werden soll. Die auszuführende Wortadresse (EAD) wird inkrementiert bzw. dekrementiert und der dAD-Pointer rotiert nach rechts bzw. links. Die Bewegung zu einem Pixel innerhalb des gleichen Wortes erfordert die Rotation des dAD-Pointers nach rechts oder links.

In Tafel 4.2 werden diese Operationen in einer Übersicht zusammengefasst.

Das Zeichnen von ganzen Worten ist zur Ausfüllung von Flächen mit dem gleichen Wert im Bildwiederholtspeicher geeignet.

Beim Setzen sämtlicher Bits des Maskenregisters mit dem MASK-Kommando auf Eins ist sowohl das LSB als auch das MSB der Punktheadadresse dAD ebenfalls auf Eins gesetzt. Damit wird der Wert der auszuführenden Wortadresse EAD für jeden Zyklus, unabhängig von der Richtung, inkrementiert oder dekrementiert werden. Ein RMW-Zyklus ist in der Lage, sämtliche 16 Bit eines Wortes für eine beliebige Zeichnungsart zu beeinflussen. Ein Bit im Patternregister wird je RMW-Zyklus genutzt, um sämtliche Bits eines Wortes auf den gleichen Wert zu setzen. Das nachfolgende Patternbit wird für das nächste Wort verwendet usw.

Die im Ergebnis der Zeichnung in die festgelegte Richtung möglichen Darstellungen des GDC U82720 sind im Bild 4.4 in Abhängigkeit des Richtungsparameters zusammengefasst.

**Zur Beachtung :** Während der Zeichnung einer Geraden kann in Folge der Festlegung des DIR-Wertes (s.a Abschn. 5.1.) der Winkel der Geraden im Inneren des überdeckten Oktanten beliebig definiert sein.

Die Zeichnung eines Bogens beginnt mit dem Start in die mit dem DIR-Wert festgelegte Richtung und der Drehung in einen Bogen, entsprechend der Zeichnung im Bild 4.4. Ein Bogen kann bis zu 45 Grad betragen.

DMA-Übertragungen werden nur in vollständigen Worten gemäß den Pfeilen in der DMA-Spalte durchgeführt, um die Wortadressen zu finden. Die schrägen Wege für die DMA-Übertragungen geben dem GDC während der Bewegung zum nächsten Wort die Änderung der Achsen in X- und Y-Richtung an. Dabei erfolgen die Bewegungen zwischen den Pixeln nicht im Winkel von 45 Grad.

Richtung	Kodierung	Adreßberechnung(en)
0	000	$EAD = EAD + P$
1	001	$EAD = EAD + P$ wenn $dAD.MSB=1$ , dann $EAD = EAD + 1$
2	010	wenn $dAD.MSB=1$ , dann $EAD = EAD + 1$ $dAD = LR(dAD)$
3	011	$EAD = EAD - P$ wenn $dAD.MSB=1$ , dann $EAD = EAD + 1$ $dAD = LR(dAD)$
4	100	$EAD = EAD - P$
5	101	$EAD = EAD - P$ wenn $dAD.LSB=1$ , dann $EAD = EAD - 1$
6	110	wenn $dAD.LSB=1$ , dann $EAD = EAD - 1$ $dAD = RR(dAD)$
7	111	$EAD = EAD + P$ wenn $dAD.LSB=1$ , dann $EAD = EAD - 1$ $dAD = RR(dAD)$

Erklärung der Abkürzungen :

P	-->	Teilfaktor (Pitch)
LR	-->	Rotieren nach links (Left Rotate)
RR	-->	Rotieren nach rechts (Right Rotate)
EAD	-->	ausführbare Wortadresse (Execute Word Address)
dAD	-->	Punktadresse (Dot Address) im Maskenregister
LSB	-->	niederwertiges Bit (Least Significant Bit)
MSB	-->	höchstwertiges Bit (Most Significant Bit)

Tafel 4.2: Einzelheiten der Adreßberechnungen des GDC

#### 4.6. Zeichnungsparameter

Zur Vorbereitung einer grafischen Zeichnung benötigt der Zeichnungsprozessor (drawing prozessor) des GDC die Darstellungsart, die Richtungs- und Zeichnungsparameter, die Pixel-Startadresse sowie die Pattern der Zeichnung vom System-Mikroprozessor. Sobald diese Voraussetzungen erfüllt sind, leitet das FIGD-Kommando (s.a. Abschn. 5.1.4.4.) die Zeichnung ein. Ab diesem Moment ist der System-Mikroprozessor nicht mehr am Zeichnungsprozeß beteiligt. Der Zeichnungsprozessor des GDC koordiniert die RMW-Logik und adressiert die Register, um das Bild pixelweise zu zeichnen. Der dabei vom Zeichnungsprozessor benutzte Algorithmus ermöglicht ein Optimum an Zeichnungsgeschwindigkeit. Die spezifischen Einzelaktivitäten beim Ablauf der zeichnerischen Darstellung werden durch den System-Mikroprozessor auf eine Form reduziert, die der schnellen Berechnung der Adressen im inneren des GDC dienlich ist. Auf diese Art erfolgen die sich ständig wiederholenden Pixel-Berechnungen und ermöglichen dadurch eine Minimierung der Zeit zur Zeichnung der Darstellung. In Tafel 3.1 wurden die Zeichnungsparameter tabellarisch zusammengefasst.

Richtung	Gerade	Bogen	Zeichen	Schrägzeichen	Rechteck	DMA
000						
001						
010						
011						
100						
101						
110						
111						

Tafel 4.3: Zeichnungsrichtungen des GDC

#### 4.7. Darstellung grafischer Zeichen

Grafische Zeichen können pixelweise in den Bildwiederholtspeicher gezeichnet werden. Vom System-Mikroprozessor sind bis zu acht mal acht Zeichen in das Parameter-RAM des GDC ladbar. Konsequenterweise werden keine Einschränkungen des zu benutzenden Zeichensatzes vorgenommen. Durch Veränderung der Zeichnungsparameter und der Zeichnungsrichtung sind

zahlreiche frei wählbare Darstellungsarten möglich. Bei Anwendungen zur Ausfüllung von Flächen kann ein grafisches Zeichen beliebig oft, ohne das Parameter-RAM erneut zu laden, in den Bildwiederholpeicher geschrieben werden. Für den Fall, daß in das Parameter-RAM mit dem dafür bestimmten PRAM-Kommando (s.a. Abschn. 5.1.3.5.) bis zu acht grafische Zeichenbytes geladen wurden, kann das GCHRD-Kommando (s.a. Abschn. 5.1.2.3.), beginnend an der Cursorposition, die Bytes in den Bildwiederholpeicher einzeichnen.

Der mit dem ZOOM-Kommando gewählte Vergrößerungsfaktor ermöglicht die Festlegung der Größe des in den Bildwiederholpeicher zu schreibenden grafischen Zeichens, in ganzzahligen Vielfachen zwischen Eins und 16. Die Bitwerte im Parameter-RAM werden sooft horizontal und vertikal wiederholt, wie es durch den ZOOM-Faktor spezifiziert wurde. Die Bewegung dieser PRAM-Bytes zum Bildwiederholpeicher wird über die Parameter des FIGS-Kommandos (s.a. Abschn. 5.1.) gesteuert. Entsprechend der festgelegten Höhe und Breite der zu zeichnenden Fläche wird das Parameter-RAM abgefragt, um die geforderte Fläche auszufüllen.

Für ein 8 \* 8-Zeichen, wird das erste Pixel unter Verwendung des niederwertigsten Bit (LSB) von RA-15 gezeichnet. Das zweite Pixel benutzt das nachfolgende Bit (Bit 1) von RA-15. Dies wiederholt sich, bis das höchstwertigste Bit (MSB) von RA-15 erreicht wird. Danach springt der GDC zum korrespondierenden Bit in RA-14, um die Zeichnung fortzusetzen. Der Vorgang dauert bis zum LSB von RA-14 und setzt sich schlangenförmig über die anderen sechs PRAM-Bytes fort. Dieser Vorgang gleicht dem Ablauf der Berechnung der Adressen des Bildes durch den Zeichnungsprozessor (s.a. Bild 4.4.).

Falls die Fläche weniger als acht Pixel breit ist, wird vor dem Erreichen des MSB zum nächsten PRAM-Byte übergegangen. Für den Fall, daß die Fläche weniger als acht Zeilen hoch ist, werden entsprechend weniger Bytes des Parameter-RAM abgefragt. Für Flächen, die größer als ein 8 \* 8-Pixelfeld sind, wiederholt der GDC den Inhalt des Parameter-RAM im doppeltem Ausmaß.

#### **4.8. Funktion des Parameter-RAM**

Die im Parameter-RAM gespeicherten Parameter sind für den GDC zum wiederholten Zugriff während der Bilddarstellung verfügbar und können während des Zeichnungsprozesses und der Rasterabtastung wiederholt übergeben werden. In jedem Zustand einer Operation werden die Werte des PRAM in einer vorgegebenen Form durch den GDC interpretiert.

Der System-Mikroprozessor muß die passenden Parameter auf den richtigen Platz im PRAM laden. Ein spezielles Kommando zum Laden des PRAM (s.a. Abschn. 5.1.3.5.) erlaubt dem System-Mikroprozessor auf einen beliebigen Platz im PRAM zuzugreifen und so viele Bytes zu übertragen, wie erforderlich sind. Auf diesem Weg können ein oder mehrere beliebige Parameterbytes ohne Beeinflussung anderer PRAM-Bytes verändert werden.

Das PRAM speichert zwei Informationsarten. Zur Festlegung von Einzelheiten der Display-Bildfläche werden Blöcke zu vier Bytes verwendet. Die vier in jedem Block gespeicherten Parameterbytes enthalten die Startadresse im Bildwiederholpeicher sowie die Länge der Display-Bildflächen.

Zusätzlich werden für jede Fläche zwei Betriebsartenbits (mode bits) definiert. Das Image-Flag (IM-Flag in RA-3, s.a. Abschn. 4.9.) legt fest, ob die Fläche zur bitweisen Grafikdarstellung oder zur kodierten Zeichendarstellung verwendet wird. Weiterhin wird festgelegt, ob für die jeweilige Fläche ein normaler oder erweiterter Display-Zyklus (WD-Flag z.B. in RA-3) benutzt wird.

Die anderen PRAM-Bytes dienen der Bereitstellung der Pattern für die grafische Bilddarstellung. Die PRAM-Bytes RA-8 bis RA-15 sind für die Pattern der Bildmusterinformation reserviert. Für das Zeichnen einer Geraden, eines Bogens oder eines Rechtecks werden die

PRAM-Bytes RA-8 und RA-9 in das Pattern-Register geladen. Dies ermöglicht dem GDC, Punkt- oder Strichlinien zu zeichnen. Für Flächenausfüllung und zur bitweisen Zeichnung grafischer Zeichen enthalten die PRAM-Byte RA-8 bis RA-15 die darzustellenden Pattern oder den alphanumerischen Zeichencode.

Die nachfolgenden Bilder 4.4 und 4.5 zeigen jeweils den Aufbau des Parameter-RAM für die Grafik- bzw. Misch-Betriebsart.

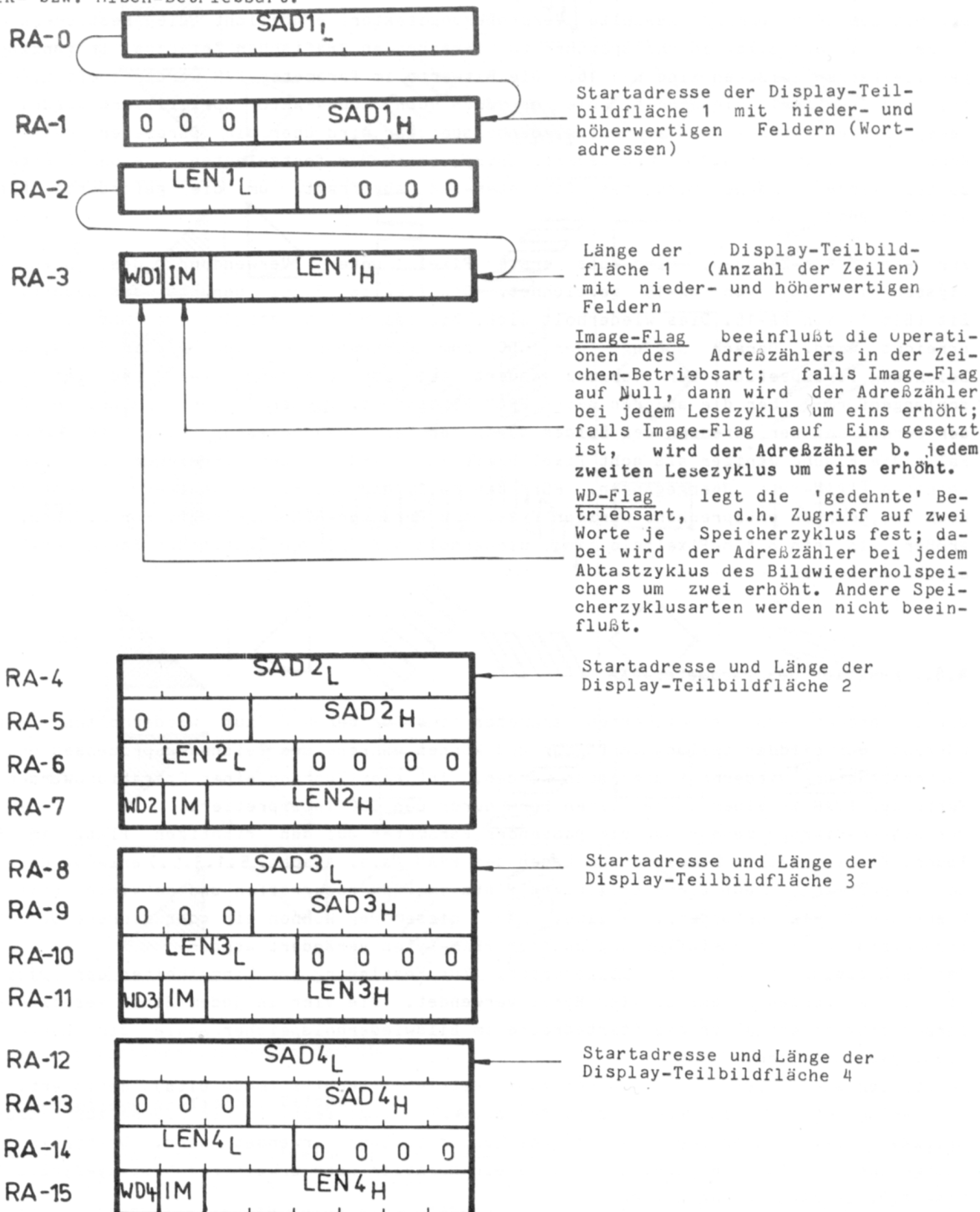


Bild 4.4: Inhalt des Parameter-RAM in der Zeichen-Betriebsart

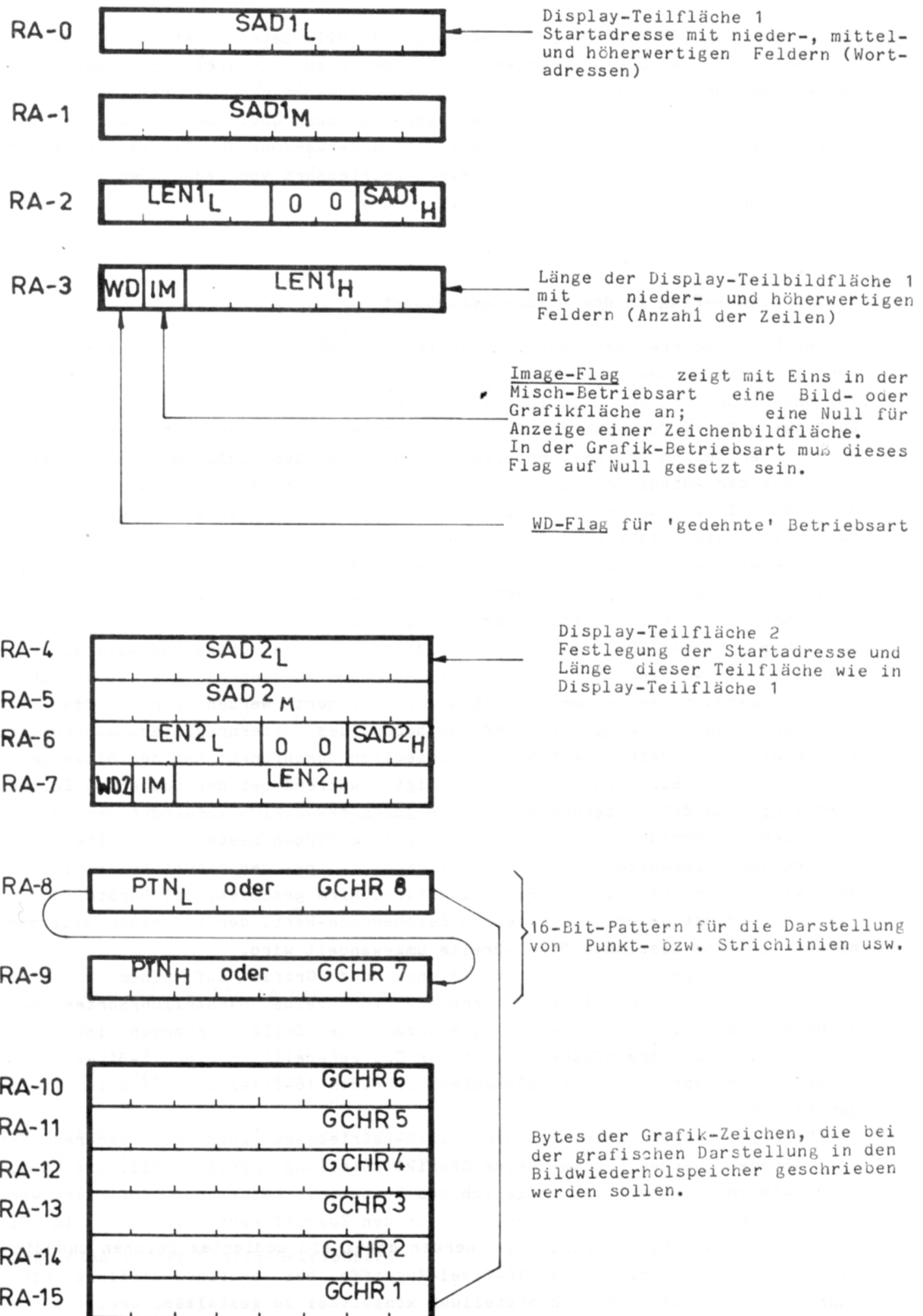


Bild 4.5: Inhalt des Parameter-RAM im Grafik- und in der Misch-Betriebsart



#### 4.9. Image-Flag

Das Image-Flag (IM-Flag) wird zur Anzeige der Bild-Betriebsart für jede Bildfläche im Parameter-RAM des GDC gespeichert (Bit 6 des PRAM-Byte drei und sieben). Die Festlegung des Wertes für das IM-Flag erfolgt mit dem PRAM-Kommando. Generell erfolgt die Festlegung dieses Flags in der Initialisierungsphase des GDC. Der Wert des IM-Flag bleibt unverändert, solange sich die Aufteilung und Verwendung der Bildflächen nicht ändert. Das IM-Flag ist hauptsächlich in der Misch-Betriebsart von Bedeutung; spielt jedoch auch in den anderen Betriebsarten eine Rolle.

##### 4.9.1. Image-Flag in der Misch-Betriebsart

In der Misch-Betriebsart bestimmt das IM-Flag, ob eine Bildfläche für bitweise Grafik- oder codierte Zeichendarstellung vorgesehen ist.

Zur Unterscheidung der Bildflächen sind mehrere Aktionen vom GDC vorzunehmen.

Zunächst erfolgt für eine Bildfläche mit codierter Zeichendarstellung die Festlegung der Anzahl der Zeilen pro Zeichenzeile (LR-Parameter des CCHAR-Kommandos). Damit erfolgt die Steuerung der Anzahl der Zeitintervalle für eine Wortzeile, die während der Inkrementierung des Zeilenzählers abgetastet wird. In der bitweisen Grafik-Betriebsart erfolgt automatisch die Abtastung einzelner Zeilen.

Vorausgesetzt wird dabei, daß für Zeichendarstellungen die externe Hardware einen Zeilenzähler und einen Zeichengenerator enthält. Für bitweise Grafikdarstellungen ist nur ein externes Video-Schieberegister erforderlich.

Weiterhin gibt der GDC das zum IM-Flag zugehörige Image-Mode-Signal (Bild-Betriebsartensignal) während der HSYNC-Zeit auf dem Ausgang A17 aus, wo es zur Steuerung der externen Hardware in einem Flip-Flop gespeichert werden kann. Gleichfalls ist zu beachten, daß der GDC die Art der Handhabung des Patternregisters während des RMW-Zyklus in den einzelnen Betriebsarten unterschiedlich durchführt: Bei der bitweisen Darstellung wird nur ein Bit pro Zyklus benötigt, während bei der codierten Zeichendarstellung sämtliche Bits des Patternregisters für jeden RMW-Zyklus verwendet werden.

Bei Anwendung der unterschiedlichen Darstellungsarten besteht ein weiterer Unterschied in der Art des Informationsflusses zum Bildschirm. Bei der bitweisen Grafikdarstellung werden bei jedem Wortzugriff 16 Pixel zum Bildschirm gesendet; bei Darstellung von codierten Zeichen bedingt jeder Zugriff einen Zeichencode-Wert, der von einem externen Zeichengenerator in eine bestimmte Pixel-Breite umgewandelt wird.

Zur gleichzeitigen Darstellung von Zeichen und Grafik auf einem Bildschirm ist die Nutzung der gleichen Pixeltaktfrequenz für beide Darstellungsarten empfehlenswert, unabhängig davon, ob die gleiche Pixelanzahl je Zeile vorhanden ist. Um die Misch-Darstellung zu vereinfachen, geht der GDC generell von einem 8-Bit-Zeichenfenster aus. Demzufolge entsprechen zwei Zeichenfenster einem 16-Pixel-Zugriff bei bitweiser Grafikdarstellung.

(Hinweis: Der GDC kann in der Misch-Betriebsart auch mit anderen Fensterbreiten betrieben werden, wenn keine Grafikdarstellung erfolgen soll. Die Festlegung des 8-Bit-Zeichenfensters ist lediglich aus Kompatibilitätsgründen definiert worden.)

Der GDC benötigt zwei 2xWCLK-Perioden für den Zugriff auf ein Wort im Bildwiederhol-speicher. In der Misch-Betriebsart werden dabei ein codiertes Zeichen und die dazugehörigen Attribute aufgerufen. Um die 16-Pixel-Zugriffe bei Grafikdarstellung mit den 8-Bit-Zugriffen bei der Zeichendarstellung kompatibel zu gestalten, greift der GDC innerhalb von vier (anstatt von zwei!) Taktzyklen von 2xWCLK auf ein neues Grafikwort zu. Damit

entspricht, unabhängig vom Typ der verwendeten Bildflächen, jedes 2xWCLK-Paar acht Pixeln. Zur korrekten Initialisierung des GDC muß der AW-Parameter des RESET-Kommandos (s. Bild 5.2 auf die Anzahl der abzubildenden 2xWCLK-Paare gesetzt werden und nicht auf die Anzahl der Grafikworte.

#### 4.9.1.1. Zeitverhalten

Das nachfolgende Zeitdiagramm (Bild 4.6) ist für beide Möglichkeiten der Darstellung in der Misch-Betriebsart anzuwenden. Sie sind auf die gleichen zwei 2xWCLK-Zyklen bezogen. Im ersten Fall nutzt eine Fläche zur codierten Zeichendarstellung diese Intervalle zur Durchführung eines Zugriffs auf ein codiertes Zeichen. Die Wortadresse (DAD) wird für die nächste Zeichenadresse inkrementiert. Im anderen Fall benutzt eine Bildfläche zur bitweisen Grafikdarstellung zwei dieser 2xWCLK Intervalle (bzw. vier 2xWCLK-Zyklen), wobei innerhalb dieser Intervalle die Wortadresse inkrementiert wird. In beiden Fällen generiert der GDC aller zwei 2xWCLK-Zyklen einen ALE-Zyklus. Im Bild 4.6 ist das zugehörige Zeitverhalten dargestellt.

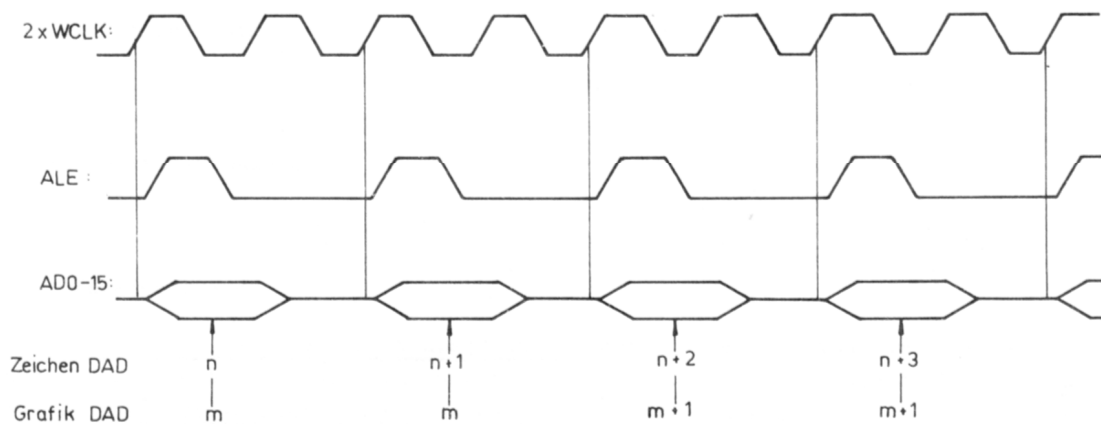


Bild 4.6: Adressierung von DAD in der Misch-Betriebsart

In einer Anwendung, die sowohl die codierte Zeichendarstellung, als auch die bitweise Grafikdarstellung verwenden soll, ist es notwendig die zur Darstellung zu bringenden Flächen für Bilder und Grafiken zu unterscheiden und eine zweite, vierte usw. Ladung des Video-Schieberegisters für Grafik zu unterdrücken. Damit wird die richtige Serialisierung aller 16 Bit des ersten Speicherzugriffs gesichert.

#### 4.9.2. Image-Flag in der Grafik-Betriebsart

Das auf den Wert Eins gesetzte Image-Flag erzwingt in der Grafik-Betriebsart, ähnlich wie in der Misch-Betriebsart, das Weiterücken zu einem anderen Display-Zyklus der verwendeten Bildfläche. Während der HSYNC-Zeit findet keine Ausgabe dieses Flags über den Ausgang A17 statt. Damit sind hinsichtlich dieses Verhaltens sämtliche Bildflächen gleich.

Es gibt in der Grafik-Betriebsart keine Möglichkeit der Anzeige für das Ende und den Anfang einer Bildfläche.

Wenn normale Operationen in der Grafik-Betriebsart erforderlich sind, ist das Image-Flag auf Null zu setzen. Damit wird die Inkrementierung der Wortadresse für das nächste Paar von 2xWCLK-Zyklen (entspricht einem Display-Zyklus) erzwungen.

## 5. Beschreibung der Kommandos des GDC U82720

Nachfolgend wird im Bild 5.1 eine Übersicht der Kommandos des GDC U82720 gegeben.

RESET:	0 0 0 0 0 0 0 0	WDAT:	0 0 1 TYP 0 MOD
SYNC :	0 0 0 0 1 1 1 DE	MASK:	0 1 0 0 1 0 1 0
VSUNC:	0 1 1 0 1 1 1 M	FIGS :	0 1 0 0 1 1 0 0
CCHAR:	0 1 0 0 1 0 1 1	FIGD :	0 1 1 0 1 1 0 0
START:	0 1 1 0 1 0 1 1	GCHRD:	0 1 1 0 1 0 0 0
BCTRL:	0 0 0 0 1 1 0 DE	RDAT:	1 0 1 TYP 0 MOD
ZOOM :	0 1 0 0 0 1 1 0	CURD:	1 1 1 0 0 0 0 0
CURS :	0 1 0 0 1 0 0 1	LPRD:	1 1 0 0 0 0 0 0
PRAM :	0 1 1 1 SA	DMAR:	1 0 1 TYP 1 MOD
PITCH:	0 1 0 0 0 1 1 1	DMAW:	0 0 1 TYP 1 MOD

Bild 5.1: Kommandoübersicht

### 5.1. Video-Steuerkommandos

#### 5.1.1. RESET-Kommando

Das RESET-Kommando kann jederzeit ausgeführt werden, ohne bereits im GDC gespeicherten Parameter zu verändern. Wenn Parameterbytes folgen, werden mit diesem Kommando auch die Parameter des Video-Synchronisationsgenerators gesetzt, wie später beschrieben wird. Der Freilaufbetrieb (Idle Mode) wird mit dem START-Kommando verlassen.

In der Grafik-Betriebsart besteht ein Wort aus einer Gruppe von 16 Pixeln. In der Zeichen-Betriebsart enthält ein Wort immer den Zeichencode und die dazugehörigen Attribute.

Eine Zeile besteht aus einer geraden Anzahl von 2 bis 256 aktiven Worten. Ein Parameter welcher ausschließlich Nullen enthält, wählt eine Zahl der Größe  $2^n$  aus, wobei n der Anzahl der Bits im Parameterfeld für Vertikalparameter entspricht.

Die Abstände in horizontaler Richtung werden in 16-Bit-Worten, die Abstände in vertikaler Richtung in Zeilen gezählt. Im Bild 5.2 ist der Aufbau des RESET-Kommandos dargestellt.

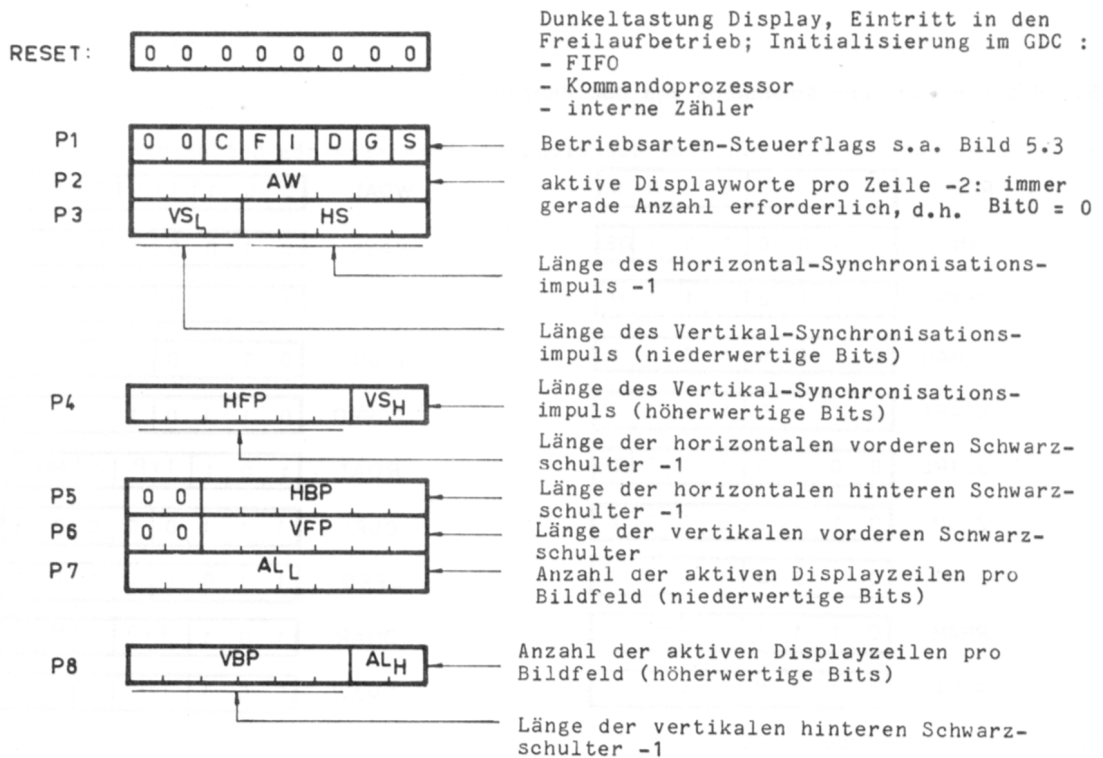


Bild 5.2: RESET-Kommando und zugehörige optionale Parameter

## 5.1.2. Kommandos zur Synchronisation

### 5.1.2.1. Synchronisationsparameter

Nachfolgend werden die Parameter zur Programmierung des Video-Synchronisationsgenerators für die verschiedenen Betriebsarten angegeben:

#### - horizontale vordere Schwarzscher (HFP)

1. generell gilt:

HFP  $\rightarrow \geq 2$  Worte

2. Falls im DMA-Betrieb gearbeitet wird oder wenn bei Verwendung des Zeilensprungverfahrens der Zoomfaktor größer als Eins gewählt wird:

HFP  $\rightarrow \geq 3$  Worte

3. Falls der GDC im Slave-Betrieb betrieben wird:

HFP  $\rightarrow \geq 4$  Worte

4. Falls eine Nutzung des Lichtstifteingangs erfolgt:

HFP  $\rightarrow \geq 6$  Worte

#### - Horizontal-Synchronisationssignal (HS)

1. Falls Refresh-Zyklen für dynamische RAMs gebraucht werden:

HS  $\rightarrow \geq 2$  Worte

2. Falls Anwendung des Zeilensprungverfahrens erfolgt:

HS  $\rightarrow \geq 5$  Worte

- horizontale hintere Schwarzschiiter (HBP)

1. Generell gilt:

HBP  $\rightarrow \geq 3$  Worte

2. Falls Anwendung des Zeilensprungverfahrens erfolgt oder Wechsel der Bild-Betriebsart (IM-Flag, s.a. Abschn. 4.9.) sowie Programmierung des Wide-Display-Zyklus (WD-Flag, s.a. Abschn. 4.8.) innerhalb eines Bildfeldes durchgeführt wird:

HBP  $\rightarrow \geq 5$  Worte

- Betriebsarten-Steuerbits

Für Betriebsart mit "Wiederholungsfeld" : Zwei Bildfelder mit einer Verschiebung von einer halben Zeile und gleichem Bildinhalt folgen nacheinander.

Für Betriebsart mit Zeilensprung : Zwei Bildfelder werden mit einer Verschiebung von einer halben Zeile übertragen. In den aufeinanderfolgenden Feldern wird jeweils ein Halbbild dargestellt.

Für Betriebsart ohne Zeilensprung : In einem Feld wird die gesamte Information auf dem Bildschirm dargestellt.

Beim Zeilensprungverfahren ist die Gesamtanzahl der Zeilen ungerade. Die Summe aus VFP + VS + VBP + AL sollte um eins niedriger als die ungerade Anzahl der gewünschten Zeilen sein. Die Betriebsarten-Steuerbits sind im Bild 5.3 zusammenfassend dargestellt.

Bit	Funktion	
C	G	Display- Betriebsart
0	0	Misch-Betriebsart
0	1	Grafik-Betriebsart
1	0	Zeichen-Betriebsart
1	1	ungültig
I	S	VIDEO - Eigenschaften
0	0	ohne Zeilensprung
0	1	ungültig
1	0	Zeilensprung-Wiederholungsfeld
1	1	mit Zeilensprung
D	Freigabe d RAM-Refresh	
0	kein Refresh	
1	Refresh	
F	Zeichnungszeitpunkt	
0	Displayzeit + Dunkeltastung	
1	nur bei Dunkeltastung	

Bild 5.3: Betriebsarten-Steuerbits

Erläuterung der Abkürzungen:

VFP vertikale vordere Schwarzschiiter

VS Vertikalsynchronimpulsdauer

VBP vertikale hintere Schwarzschiiter

AL Anzahl der aktiven Zeilen einer Bildfläche

Die Sicherung des Refreshs dynamischer RAMs im Bildwiederholpeicher gewinnt besondere Bedeutung, wenn mit großem Vergrößerungsfaktor oder im DMA-Betrieb gearbeitet wird, da während der Erzeugung des Video-Rasters nicht auf alle Reihen der RAMs zugegriffen wird. Dies gilt es auch zu beachten, wenn der Bildwiederholpeicher anderweitig im inaktiven Zustand gehalten wird.

Der Zugriff auf den Bildwiederholpeicher kann auf die dunkelgetasteten Rücklaufintervalle beschränkt werden. Damit entstehen keine Unregelmäßigkeiten in der Abbildung auf dem Display.

#### 5.1.2.2. SYNC-Kommando

Mit diesem Kommando werden Parameter zur Festlegung des SYNC-Formats in den Synchronisationsgenerator geladen. Die verschiedenen Parameterfelder und Bits entsprechen denen des RESET-Kommandos. Der GDC wird beim SYNC-Kommando jedoch nicht zurückgesetzt und erreicht auch nicht den Freilaufbetrieb. Im Bild 5.4 ist der Aufbau des SYNC-Kommandos angegeben.

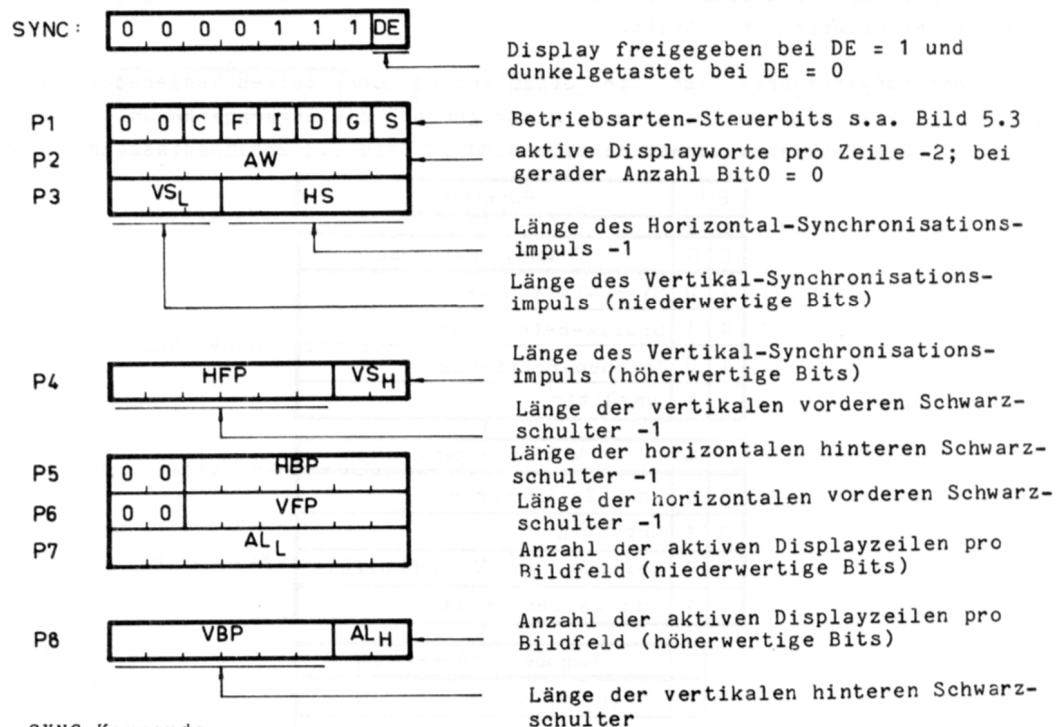


Bild 5.4: SYNC-Kommando

#### 5.1.2.3. VSYNC-Kommando

Falls zwei oder mehr GDCs zur Erzeugung eines Bildes verwendet werden, dann arbeitet ein GDC als Master-Synchronisationsgenerator, während die anderen GDCs im Slave-Betrieb arbeiten. Die VSYNC-Anschlüsse aller GDCs sind dabei miteinander verbunden.

Nachfolgend einige Bemerkungen zur o.g. Slave-Betriebsart :

Die Synchronisation von zwei oder mehr GDCs über den VSYNC-Ein-/Ausgang zur Erzeugung von mehreren überlagerten Videosignalen erfordert die Beachtung einiger Besonderheiten : Die horizontale vordere Schwarzscher (HFP) muß mindestens vier



Display-Zyklen lang sein. Das entspricht acht oder mehr Taktzyklen. In dieser Zeit erfolgt bei den Slave-GDCs die Initialisierung der internen Video-Synchronisationsgeneratoren auf den richtigen Punkt im Bildfeld, um den eintreffenden Vertikal-Synchronisationsimpuls (VSYNC) anzupassen. Dieses Rücksetzen des Generators findet gleich nach dem Ende des VSYNC-Impulses innerhalb des HFP-Intervalls statt. Vom Slave-GDC wird während des HFP-Intervalls eine bestimmte Zeit benötigt, um die Operation vor Beginn des HSYNC-Intervalls abzuschließen.

Für den Fall, daß sämtliche GDCs initialisiert sind sowie der Master und die Slaves definiert wurden, ist einige Zeit zur Synchronisation erforderlich. Zur Kontrolle dessen kann man das VSYNC-Statusflag (M-Flag) des Master-GDCs überwachen und vor Beginn eines Darstellungsprozesses solange warten, bis ein oder mehrere VSYNC-Impulse generiert wurden.

Mit dem START-Kommando beginnt die aktive Darstellung der Daten auf der Bildfläche des Displays und die Vorbereitung der Video-Synchronisation der GDCs untereinander wird beendet. Dafür muß mindestens ein VSYNC-Impuls des Masters zur Synchronisation der Slave-GDCs generiert worden sein.

Der Aufbau des VSYNC-Kommandos ist im Bild 5.5 angegeben.

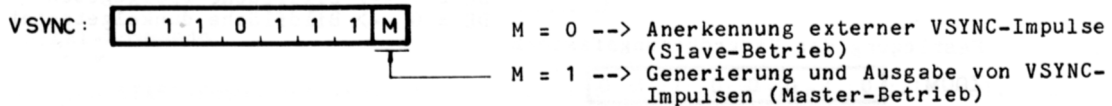


Bild 5.5: Vertikalsynchronisations-Kommando

#### 5.1.2.4. CCHAR-Kommando

In der Grafik-Betriebsart wird der LR-Parameter (Zeilen pro Zeichenreihe - 1) in P1 des CCHAR-Kommando auf Null gesetzt. Bei Darstellungen in der Grafik-Betriebsart mit Zeilensprungverfahren sollte der Blinkfolgeparameter (BR) auf den Wert Drei gesetzt sein. Der Blinkfolgeparameter steuert für den Cursor und für die Attribute die Blinkfrequenz. Die Aktiv-Zeit des Cursors ist gleich der Passiv-Zeit und entspricht  $2 \times BR$  (Blinkfrequenz). Die Blinkfrequenz der Attribute ist gleich der halben Blinkfrequenz des Cursors, die Aktiv-Zeit beträgt jedoch 75%, die Passiv-Zeit 25%. Im Bild 5.6 ist der Aufbau des CCHAR-Kommandos dargestellt.

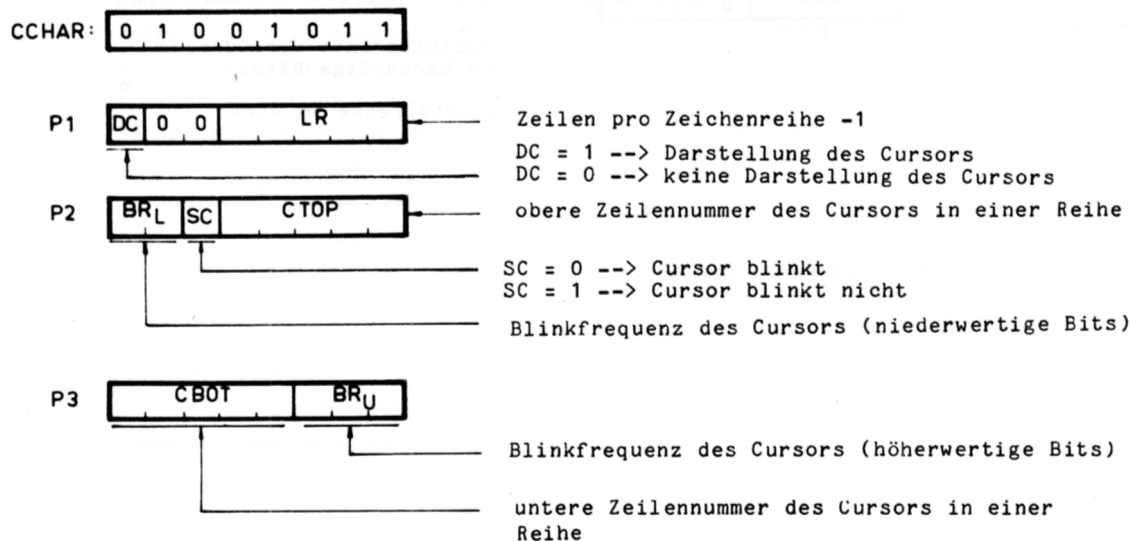


Bild 5.6: Aufbau des CCHAR-Kommandos

### 5.1.3. Display-Steuerkommandos

Die nachfolgend beschriebenen Display-Steuerkommandos START, BCTRL, ZOOM und CURS sind im Bild 5.7 zusammenfassend dargestellt.

Start der Darstellung auf der Bildfläche des Displays und Ende des Freilaufbetriebes

START: 

0	1	1	0	1	0	1	1
---	---	---	---	---	---	---	---

Steuerung der Dunkeltastung

BCTRL: 

0	0	0	0	1	1	0	DE
---	---	---	---	---	---	---	----

DE = 1 --> Bildfläche freigegeben  
DE = 0 --> Bildfläche dunkelgetastet

Festlegung des Vergrößerungsfaktors

ZOOM: 

0	1	0	0	0	1	1	0
---	---	---	---	---	---	---	---

P1 

DISP	GCHR
------	------

Vergrößerungsfaktor -1 für das Schreiben von Grafik-Zeichen

Vergrößerungsfaktor - 1

Festlegung der Cursorposition

CURS: 

0	1	0	0	1	0	0	1
---	---	---	---	---	---	---	---

P1 

EAD
-----

auszuführende Wortadresse (niederwertiges Byte)

P2 

EAD
-----

auszuführende Wortadresse (mittleres Byte)

dAD	0	0	EAD
-----	---	---	-----

(nur in Grafik-Betriebsart)

auszuführende Wortadresse (höherwertige Bits)

Punktadresse im Wort

Bild 5.7: Display-Steuerkommandos

#### 5.1.3.1. ZOOM-Kommando

Der mit dem ZOOM-Kommando programmierbare Vergrößerungsfaktor kann in einem Bereich von 1...16 gewählt werden. Das entspricht einer Kodierung von 0...15.

#### 5.1.3.2. CURS-Kommando

In der Zeichen-Betriebsart wird das dritte Parameterbyte nicht benötigt. Die Darstellung des Cursors erfolgt während der Darstellungszeit des Wortes, in der die Adresse auf der Bildfläche (DAD) gleich der Adresse des Cursors ist. In der Grafik-Betriebsart legt die Wortadresse des Cursors das Wort fest, welches das Startpixel eines Zeichnungsprozesses enthält. Die Punktadresse kennzeichnet dieses Pixel innerhalb des Wortes.

#### 5.1.3.3. START-Kommando

Nach Empfang des START-Kommandos verläßt der GDC den Freilaufbetrieb (Idle Mode). Dem Kommando folgen keine Parameterbytes.

#### 5.1.3.4. BCTRL-Kommando

Mit dem BCTRL-Kommando wird in Abhängigkeit vom niederwertigsten Bit (DE-Flag) im Operationscode die Display-Bildfläche freigegeben oder dunkelgetastet.

#### 5.1.3.5. PRAM-Kommando

Beginnend mit der Startadresse SA kann eine Anzahl von Bytes in das Parameter-RAM geschrieben werden, dessen Adresse bis zu fünfzehn Mal inkrementiert werden kann. Die Sequenz der Parameterbytes wird beendet, sobald in den FIFO-Buffer ein Kommando-Byte eingeschrieben wird. Das Parameter-RAM speichert mit einer Tiefe von 16 Bytes Informationen in vorbestimmter Anordnung, welche sich für die Grafik- und Zeichen-Betriebsart unterscheiden. Der Aufbau des PRAM-Kommandos ist im Bild 5.8 wiedergegeben.

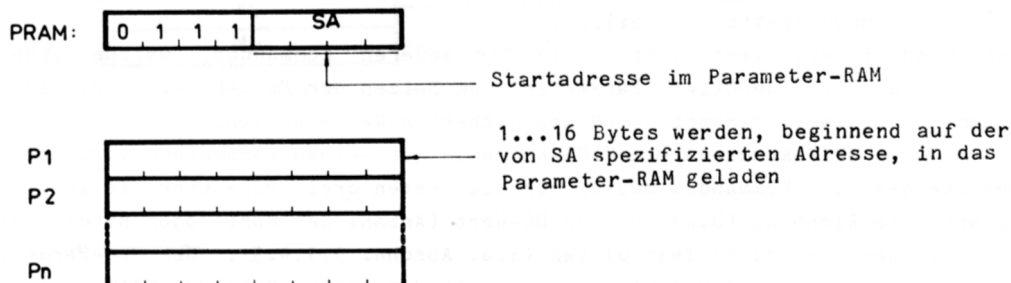


Bild 5.8: Ladekommando für das Parameter-RAM

#### 5.1.3.6. PITCH-Kommando

Der mit dem PITCH-Kommando festlegbare Wert wird während der Darstellung der Zeichnung vom Zeichnungsprozessor genutzt, um das unmittelbar über oder unter dem aktuellen Wort angeordnete Wort oder den Anfang der nächsten Zeile zu finden. Der PITCH-Parameter (Breite des Bildwiederholerspeichers) kann mit zwei verschiedenen Kommandos gesetzt werden. Neben dem PITCH-Kommando bewirken RESET (oder auch SYNC) ebenfalls die Festlegung des Pitch-Wertes. Mit dem Parameter P2 des SYNC-Kommandos (s.a. Abschn. 5.1.2.1) wird die Anzahl der aktiven Worte je Zeile, d.h. die Breite der abzutastenden Display-Bildfläche, bestimmt. Damit wird auch die Breite des Bildwiederholerspeichers festgelegt. Wenn beide Größen gleich sind, braucht kein gesondertes PITCH-Kommando gegeben werden. Im Bild 5.9 ist der Aufbau des PITCH-Kommandos angegeben.

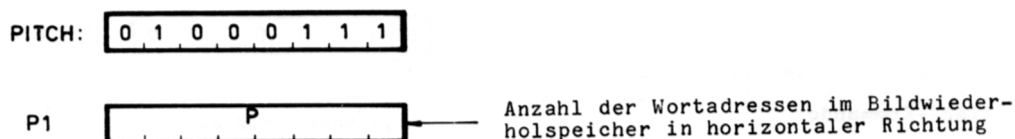


Bild 5.9: PITCH-Kommando

#### 5.1.4. Zeichnungs-Steuerkommandos

##### 5.1.4.1. WDAT-Kommando

Auf den Empfang eines Parametersatzes (zwei Bytes für eine Wortübertragung, ein Byte für Byteübertragung) erfolgt im Bildwiederholerspeicher auf der vom Cursor angezeigten auszuführenden Wortadresse (EAD) ein RMW-Zyklus. Der EAD-Pointer zeigt gemäß der vorher festgelegten Richtung auf das nächste Wort. Danach können weitere Parameter akzeptiert werden.

Beim Schreiben eines Bytes wird ein nicht spezifiziertes Byte während des RMW-Zyklus mit dem Wert Null auf allen Bitpositionen interpretiert. In der bitweisen Grafik-Betriebsart wird nur das niederwertigste Bit (LSB) der WDAT-Parameterbytes als Pattern in den RMW-Operationen genutzt. Daher besteht die Möglichkeit, nur-Null oder nur-Eins-Pattern zu erhalten. Bei kodierten Zeichenanwendungen sind alle Bits der WDAT-Parameter an der Erzeugung der Zeichnungspattern beteiligt.

Das WDAT-Kommando funktioniert nicht so wie die anderen Kommandos, welche einen RMW-Zyklus aktivieren. Es benötigt Parameter zum Setzen der Patternregister, während die anderen Kommandos, die im Parameter-RAM gespeicherten Werte nutzen. Identisch zu diesen Kommandos muß dem WDAT-Kommando ein FIGS-Kommando mit seinen Parametern vorausgehen. Dem Operationscode des FIGS-Kommandos müssen nur die ersten drei Parameter folgen, um die Zeichnungsart, die Richtung (DIR) und den DC-Wert (Anzahl der Worte oder Bytes - nicht zu verwechseln mit dem DC-Flag !) festzulegen (s.a. Abschn. 5.1.4.3). Der DC-Parameter + 1 verkörpert die Anzahl der RMW-Zyklen, welche mit dem ersten Satz der WDAT-Parameter vom GDC durchgeführt werden. Weitere WDAT-Parameter benötigen einen DC-Wert gleich Null, um die Durchführung nur eines RMW-Zyklus zu gewährleisten. Der Aufbau des WDAT-Kommandos ist im Bild 5.10 dargestellt.

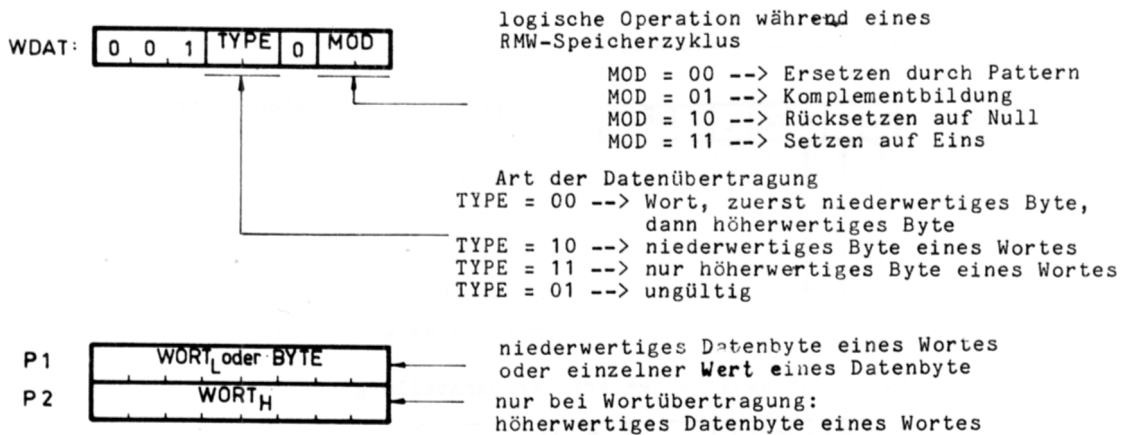


Bild 5.10: Nutzung des WDAT-Kommandos

#### 5.1.4.2. MASK-Kommando

Mit diesem Kommando wird der Inhalt des 16-Bit-Maskenregisters des Zeichnungsprozessors gesetzt. Durch das Maskenregister werden jene Bits bestimmt, die während eines RMW-Zyklus im Bildwiederholtspeicher verändert werden können.

Das Maskenregister kann sowohl durch das MASK-Kommando als auch durch das dritte Parameterbyte des CURS-Kommandos (s.a. Abschn. 5.1.3.2) geladen werden. Das MASK-Kommando akzeptiert zwei Parameterbytes, um einen 16-Bit-Wert in das Maskenregister zu laden. Sämtliche der 16 Bits können über Programmsteuerung unabhängig voneinander den Wert Null oder Eins annehmen. Das CURS-Kommando legt jedoch, abhängig von der Größe der Punktadresse (dAD), ein 1-aus-16-Pattern in das Maskenregister. Wenn normale Zeichnungsdarstellung mit einem Pixel pro Zeiteinheit verwendet wird, ist der Gebrauch des MASK-Kommandos nicht notwendig, weil das CURS-Kommando die entsprechenden Pattern zur Adressierung der richtigen Pixel im Zeichnungsprozeß setzt.

Im DMA-Betrieb mit codierten Zeichen sowie bei Operationen zum Setzen und Löschen der Bits im Bildwiederholtspeicher unter Nutzung des WDAT-Kommandos, sollte das MASK-Kommando nach dem dritten Parameterbyte des CURS-Kommandos gegeben werden. Das Maskenregister ist auch bei jeder Operation, in der ein Wort pro Zeiteinheit bearbeitet wird, zu setzen. Im Bild 5.11 ist der Aufbau des MASK-Kommandos angegeben.

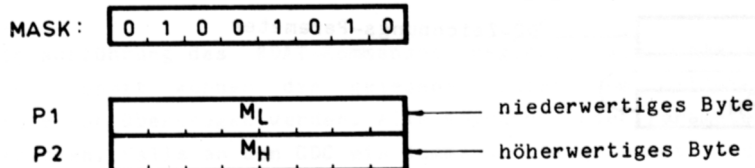
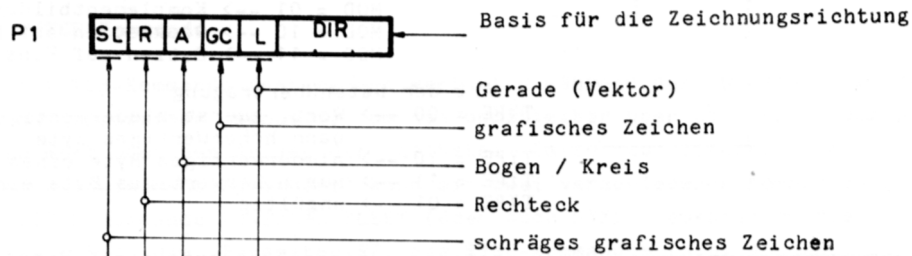


Bild 5.11: Ladekommando für das Maskenregister

#### 5.1.4.3. FIGS-Kommando

Im Anschluß an das FIGS-Kommando werden die zur Zeichnungsdarstellung notwendigen Parameter geladen. Das FIGS-Kommando und die funktionelle Bedeutung der Flags dieses Kommandos ist im Bild 5.12 dargestellt.

FIGS: 0 1 0 0 1 1 0 0



Flags zur Auswahl der Art der Darstellung

SL	R	A	GC	L	Funktion
0	0	0	0	0	Zeichen-Betriebsart, individuelle Punkt-Zeichnung, DMA, WDAT und RDAT
0	0	0	0	1	Zeichnung von Geraden
0	0	0	1	0	grafische Zeichen und Flächenausfüllung mit Grafik-Zeichenpattern
0	0	1	0	0	Zeichnung Bogen+Kreis
0	1	0	0	0	Zeichnung Rechtecke
1	0	0	1	0	Zeichnung schräger graf. Zeichen + Ausfüllung von Flächen

Nur diese Bitkombinationen sind für eine korrekte Betriebsweise des GDC U82720 zulässig.

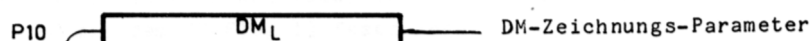
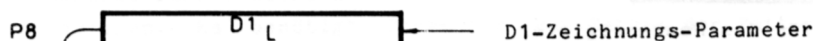
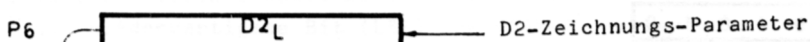
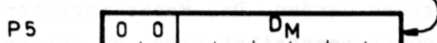
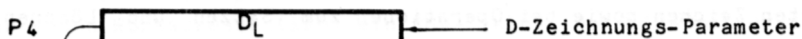
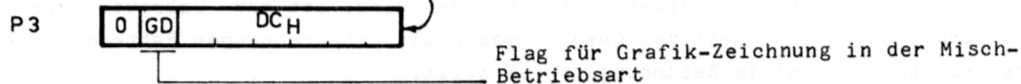
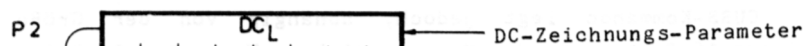


Bild 5.12: Aufbau des FIGS-Kommandos

#### 5.1.4.4. FIGD-Kommando

Bei der Ausführung dieses Kommandos lädt der GDC die Parameter aus dem Parameter-RAM in den Zeichnungsprozessor und beginnt den Zeichnungsprozeß im Bildwiederholtspeicher an der vom Cursor gekennzeichneten auszuführenden Wortadresse (EAD) und der Punktadresse (dAD). Im Bild 5.13 ist der Aufbau des FIGD-Kommandos dargestellt.

FIGD: 

0	1	1	0	1	1	0	0
---	---	---	---	---	---	---	---

Bild 5.13: FIGD-Kommando

#### 5.1.4.5. GCHRD-Kommando

Entsprechend der mit dem FIGS-Kommando geladenen Parameter initialisiert das GCHRD-Kommando die Zeichnung grafischer Zeichen oder die Ausfüllung von Flächen. Die Pattern sind im Parameter-RAM gespeichert.

Der Zeichnungsvorgang beginnt im Bildwiederholtspeicher auf der auszuführenden Wortadresse (EAD), sowie der Punktadresse (dAD). Im Bild 5.14 ist der Aufbau des GCHRD-Kommandos dargestellt.

GCHRD: 

0	1	1	0	1	0	0	0
---	---	---	---	---	---	---	---

Bild 5.14: GCHRD-Kommando

#### 5.1.5. Daten-Lesekommandos

##### 5.1.5.1. RDAT-Kommando

Unter Nutzung des mit dem FIGS-Kommando festzulegenden Richtungsparameters (DIR) und des DC-Parameters zur Festlegung der Richtung und der Anzahl der Übertragungen können mehrere RMW-Zyklen ohne erneute Festlegung der Cursoradresse nach der Initialisierung ausgeführt werden.

Wenn die Ausführung des RDAT-Kommandos beginnt, wird die Richtung des FIFO-Buffers umgekehrt. Damit können die gelesenen Daten vom Bildwiederholtspeicher zum System-Mikroprozessor übertragen werden. Alle Kommandos oder Parameter im FIFO gehen zu dieser Zeit verloren. Falls an den GDC ein Kommandobyte gesendet wird, erfolgt sofort die Umkehr der Richtung des FIFO-Buffers in die Schreib-Betriebsart. Damit sind alle RDAT-Informationen, die noch nicht aus dem FIFO gelesen wurden, verloren. Die MOD-Bits des RDAT-Kommandos sollten auf den Wert "00" gesetzt werden. Im Bild 5.15 ist der Aufbau des RDAT-Kommandos mit Erläuterung der Bits zur Datenübertragung dargestellt.

##### 5.1.5.2. CURD-Kommando

Das CURD-Kommando dient zum Lesen der Cursoradresse. Die auszuführende Wortadresse (EAD) zeigt auf das Wort im Bildwiederholtspeicher, welches das zu adressierende Pixel enthält. Die Punktadresse dAD innerhalb des Wortes wird durch einen 1-aus-16 Code gebildet. Der Aufbau des CURD-Kommandos ist im Bild 5.16 dargestellt.



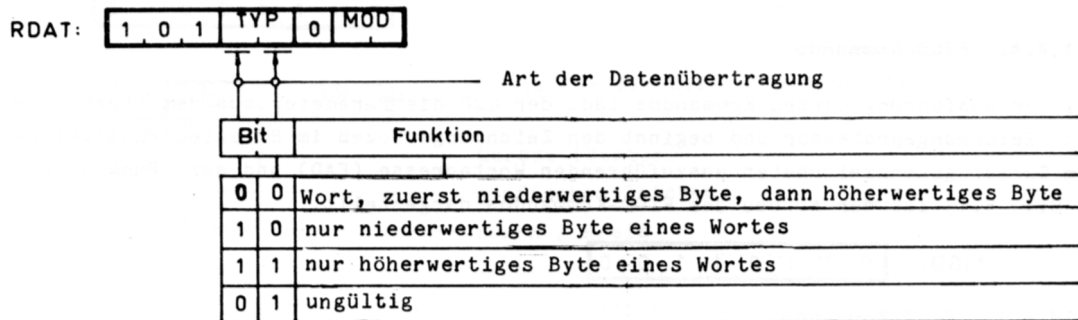


Bild 5.15: RDAT-Kommando

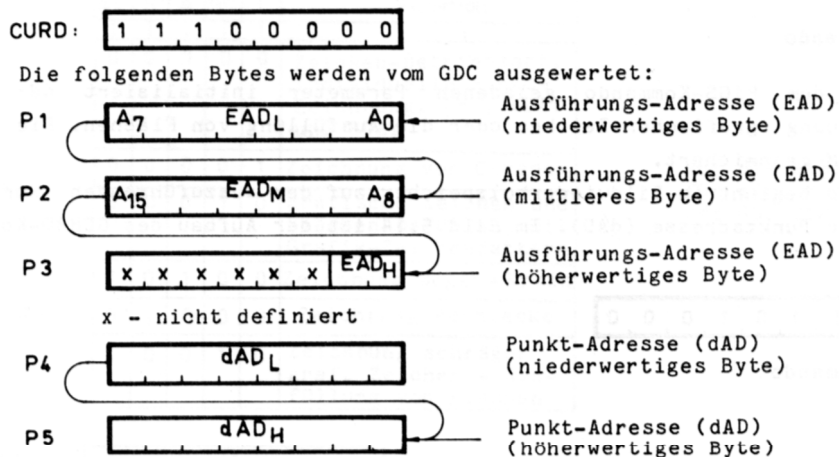


Bild 5.16: CURD-Kommando

### 5.1.5.3. LPRD-Kommando

Mit dem LPRD-Kommando wird die aktuelle Adresse des Lichtstifts gelesen. Die Lichtstiftadresse LAD korrespondiert mit der Wortadresse auf der Bildfläche (DAD), auf der das Lichtstifteingangssignal erkannt und "entstört" wurde. Das Bild 5.17 zeigt den Aufbau des LPRD-Kommandos.

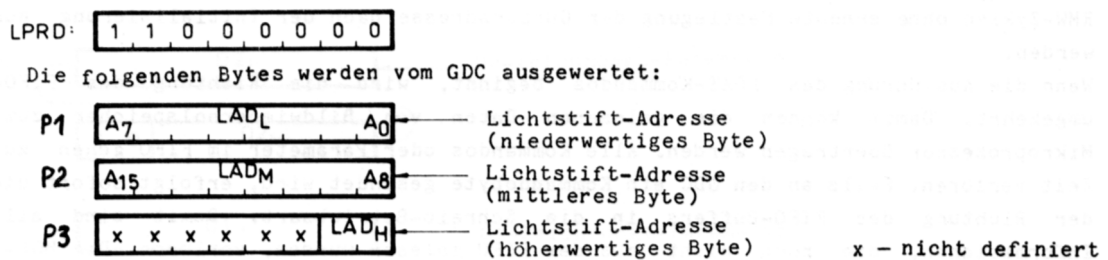


Bild 5.17: LPRD-Kommando

### 5.1.5.4. DMAR-Kommando

Mit dem DMAR-Kommando werden die Leseoperationen im Bildwiederholpeicher bei DMA-Verkehr eingeleitet. Im Bild 5.18 ist der Aufbau des DMAR-Kommandos angegeben.



## **6. Zeichnen von Darstellungen**

### **6.1. Einführende Bemerkungen**

Der GDC besitzt die Möglichkeit, selbständig mehrere Arten der Darstellung durch einfache Befehle des steuernden System-Mikroprozessors im Bildwiederholtspeicher zu zeichnen. Um das zu erreichen, hat der GDC ein speziell zur Ausführung der Zeichnungsalgorithmen entworfene interne Hardware. Das Hauptaugenmerk beim Entwurf des GDC wurde auf die Geschwindigkeit gelegt, mit der die Adressen der einzelnen Bildpunkte berechnet werden, die im Bildwiederholtspeicher beim Zeichnen von Darstellungen modifiziert werden.

Die im GDC U82720 realisierte Schaltung erfordert nur vier Taktzyklen pro Read-Modify-Write-Zyklus (RMW) für jedes zu zeichnende Pixel. Damit ist es möglich eine aus 100 Pixel bestehende Darstellung, z.B. einen Vektor, ein Rechteck, einen Kreisbogen oder ein grafisches Zeichen innerhalb von 400 Taktzyklen zu zeichnen. Bei einer Taktfrequenz von 4 MHz dauert diese Zeichenoperation nur 100  $\mu$ s.

Um diese hohe Zeichengeschwindigkeit zu erreichen, besteht die interne Struktur des GDC U82720, welche die Steuerung des Zeichenprozesses enthält, aus mehreren Addern, Schieberegistern und Zählern. Diese Netzwerke sind in einem digitalen Differentialanalysator (DDA) zusammengefaßt. Die Steuerungsparameter, welche die Darstellung beschreiben, müssen zwar vom System-Mikroprozessor aufbereitet und in den GDC geladen werden; aber von diesem Zeitpunkt an benötigt der GDC zum Zeichnen der angegebenen Darstellung keine weitere Unterstützung durch den System-Mikroprozessor.

Der höchstmögliche Systemdurchsatz wird dadurch erreicht, daß die Parameter für die nächste Darstellung bereits bestimmt und in den GDC geladen werden können, während dieser noch zeichnet.

### **6.2. Adressierung der Bildpunkte beim Zeichnen der Darstellungen**

Beim Zeichnen müssen die einzelnen Punkte (Pixel) im Bildwiederholtspeicher modifiziert werden, um die gewünschte Darstellung zu formen. Das Cursor-Kommando (CURS) ermöglicht es, die Startadresse des Bildpunktes (dAD) innerhalb der Adresse des Wortes (EAD), in dem er sich befindet, anzugeben.

Intern wird die binäre 4-Bit Bildpunktadresse (dot address, dAD) zu einem 1 aus 16-Wert erweitert, der in das MASK-Register geschrieben wird. Eine Modifizierung der Daten, die während des RMW-Zyklus aus dem Bildwiederholtspeicher gelesen werden, findet nur auf der Bitposition statt, auf der sich im MASK-Register eine logische Eins befindet. Bei der Zeichnung einer Darstellung wird durch den 1 aus 16-Wert gesichert, daß nur genau das eine Bit während des RMW-Zyklus modifiziert wird. Die anderen 15 Bit werden unverändert in den Bildwiederholtspeicher zurück geschrieben.

Zum Zeichnen von Darstellungen fordert der GDC verschiedene Voraussetzungen über die Art, wie der Bildwiederholtspeicher strukturiert und wie er dargestellt ist. Erstens muß die Wortbreite des Bildwiederholtspeichers 16 Bit betragen, damit die Darstellungen korrekt gezeichnet werden können. Zweitens muß das niederwertigste Bit (Least Significant Bit, LSB) jedes Wortes das erste Pixel sein, welches serialisiert und an das Display ausgesendet wird. Das höchstwertigste Bit (Most Significant Bit, MSB) jedes Wortes muß das letzte Pixel sein, das ausgesendet wird. Auf diese Weise ist, das übliche Nummerierungsschema (LSB hat die Nummer Null) vorausgesetzt, die Bitnummer gleich

der Pixeladresse auf dem Bildschirm, wenn man von links nach rechts zählt. Mit anderen Worten, das am weitesten links angezeigte Pixel jedes Wortes ist das LSB des Wortes im Bildwiederholtspeicher. Im Gegensatz dazu wird ein Speicherwort so dargestellt, daß das am weitesten rechts stehende Bit das LSB (Bit 0) ist.

Der GDC U82720 enthält eine innere logische Struktur, um das MASK-Register nach links oder rechts rotieren zu lassen, um auf das nächste Pixel der Darstellung zu zeigen.

Falls die äußerste rechte oder linke Bitposition erreicht wird, ist die Wortadresse (EAD) so eingestellt, daß der Zeichenprozess im benachbarten Wort fortgesetzt werden kann. Falls eine Darstellung "senkrecht" durch ein Wort geht, überträgt das MASK-Register die richtige Bildpunktadresse in das entsprechende Wort.

Durch das MASK-Kommando kann das MASK-Register eine andere Konfiguration als 1 aus 16 erhalten. Diese Möglichkeit wird hauptsächlich benutzt, wenn der GDC in der codierten Zeichen-Betriebsart arbeitet; dies kann aber auch in der Grafik-Betriebsart verwendet werden.

Modifiziertes Zeichnen von Darstellungen kann erreicht werden, wenn alle Bits des MASK-Registers auf logisch Eins gesetzt werden. In diesem Fall erkennt die Steuerlogik immer eine logische Eins in den äußeren Bitpositionen, unabhängig davon, wieviele Schiebeoperationen durchgeführt wurden. Deshalb wird die EAD immer zur nächsten Wortadresse weitergeschoben. Gleichfalls werden alle Bits des Bildwiederholtspeicherwortes modifiziert, da die Bits des MASK-Registers alle gesetzt sind.

Als Auswirkung davon wird eine Darstellung "Wort für Wort" und nicht, wie es normal wäre, "Pixel für Pixel" gezeichnet. Andere Belegungen des MASK-Registers als 1 aus 16 oder alle Bits auf logisch Eins gesetzt, ergeben beim Zeichnen sinnlose Muster.

### **6.3. Architektur des Bildwiederholtspeichers**

#### **6.3.1. Überblick**

Das Aufbereiten der Kommandos zum Zeichnen einer Darstellung mit dem GDC U82720 setzt beim Programmierer ein gründliches Verständnis der Architektur des Bildwiederholtspeichers einschließlich der Größe und der Adreßstruktur voraus. Der wirkliche Bezug zum angezeigten Bild muß klar sein. Zusätzlich müssen mehrere Terme unzweideutig definiert werden.

#### **6.3.2. Betriebsarten**

Grafische Darstellungen können durch den GDC nur selbständig gezeichnet werden, wenn der Bildwiederholtspeicher, in dem die Darstellung gezeichnet werden soll, als ein bitorientierter (bit-mapped) Grafikbereich aufgebaut ist. Das Zeichnen von Darstellungen in einen Bereich für alphanumerische Zeichen wird keine Standard-Grafikdarstellungen erzeugen.

Im der Grafik-Betriebsart stehen 18 Adreßbits zur Adressierung von bis zu 256 K Worten (vier Megapixel  $\rightarrow$  4 M) im Bildwiederholtspeicher zur Verfügung. Jedes Wort enthält 16 horizontal nebeneinanderliegende Pixel.

Im der Misch-Betriebsart (Mixed Mode) werden zwei dieser Bits für andere Zwecke benutzt, so daß nur 16 Adreßbits für die Adressierung von bis zu 64K Worten (ein Megapixel) zur

Verfügung stehen. In der Misch-Betriebsart muß darauf geachtet werden, daß keine Darstellungen in den Bereich für alphanumerische Zeichen des Bildwiederholtspeichers gezeichnet werden, da beide Bereiche zur gleichen Zeit im gleichen Bildwiederholtspeicher benutzt werden können.

### 6.3.3. Das Konzept des linearen Adreßraums im GDC U82720

Der Bildwiederholtspeicher ist wie jeder normale Programmspeicher eines Computers organisiert, d.h. vom ersten bis zum letzten Speicherplatz gibt es keine Sprünge oder fehlende Speicherplätze.

Im Unterschied zu einem in x,y-Koordinaten organisierten Speicher ist der lineare Adreßraum leicht zu überblicken und sehr flexibel.

Der GDC tastet den linearen eindimensionalen Speicher ab und erzeugt ein zweidimensionales Bild auf dem Bildschirm einer Kathodenstrahlröhre, ohne das tatsächlich eine zweidimensionale Adressierung (Zeile und Pixel) im Bildwiederholtspeicher notwendig ist.

Im Bild 6.1 wird ein zweidimensionales Feld DM mit neun Elementen, angeordnet in drei Zeilen und drei Spalten, vorgestellt.

FELD DM: Spalte 0 Spalte 1 Spalte 2

Reihe 0	DM <sub>0,0</sub>	DM <sub>1,0</sub>	DM <sub>2,0</sub>
Reihe 1	DM <sub>0,1</sub>	DM <sub>1,1</sub>	DM <sub>2,1</sub>
Reihe 2	DM <sub>0,2</sub>	DM <sub>1,2</sub>	DM <sub>2,2</sub>

Bild 6.1: Aufbau des zweidimensionalen Feldes DM

Die Indizes der Elementkennzeichnung stellen in geordneten Paaren erst die Spalte eines bestimmten Elementes und dann die Zeile (in x,y-Koordinaten) dar. Es ist zu beachten, daß durch die Richtung, in die die Zeilen- und Spaltennummern anwachsen, der Ursprung (0,0) in der linken oberen Ecke liegt. Das entspricht dem vierten Quadranten des im Bild 6.2 dargestellten Kartesischen Koordinatensystems.

Das og. Feld DM kann in einem konventionellen linearen Speicher nur abgespeichert werden, in dem die Zeilen (oder Spalten) aufeinanderfolgend im Speicher stehen.

Im Bild 6.3 ist ein derartiger Aufbau angegeben.

Dieser Aufbau hat einen entscheidenden Vorteil: Um die Größe oder Dimension des Feldes zu ändern, ist es nicht notwendig die Speicherstruktur zu ändern, solange der Speicher groß genug ist, um alle Elemente des Feldes aufzunehmen. Wenn wir zum Ausgangsfeld zurückgehen, können wir die tatsächliche Speicheradresse anstelle der geordneten Paare eintragen. Der sich damit ergebende Aufbau des Speicherfeldes ist im Bild 6.4 angegeben.

Da das Feld zeilenweise abgespeichert ist, ist die Zahl der Elemente des Feldes in dieser Ausdehnung (P oder Pitch) von grundlegender Bedeutung.

Jede horizontale Bewegung läßt sich durch einfaches inkrementieren oder dekrementieren ausführen. Vertikale Bewegungen erfordern eine Addition oder Subtraktion von P. Diagonale Bewegungen machen eine Kombination beider Operationen notwendig. Der GDC speichert das zweidimensionale Feld, das als Bild auf dem Bildschirm abgebildet werden soll, in seinem linearen Speicher in der gleichen Art und Weise wie im vorgestellten Beispiel.

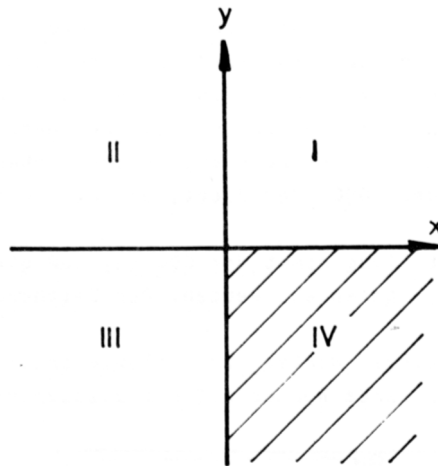


Bild 6.2: Lage der Quadranten im Kartesischen Koordinatensystem

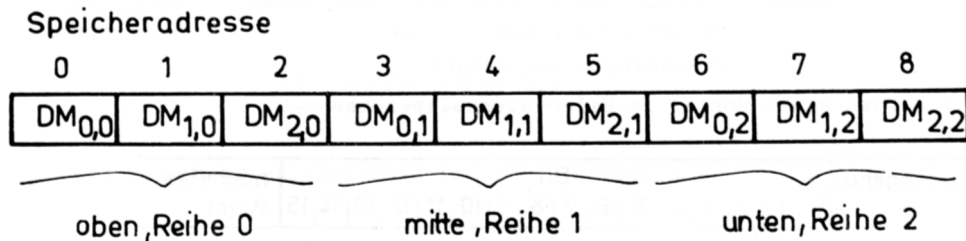


Bild 6.3: Aufbau eines konventionellen linearen Speicherraums

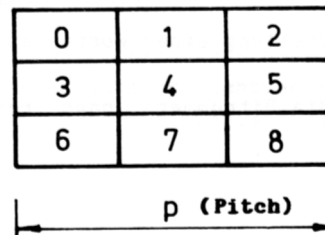


Bild 6.4: Aufbau des Speicherfeldes DM

Durch die Wahl des Quadranten für das Nummerierungsschema, welches mit dem Abtastprinzip einer Kathodenstrahlröhre übereinstimmt, läßt sich durch die Zeile auf Zeile-Speicherung der Bildwiederholtspeicher leicht abtasten, um ein Fernsehbildraster auf dem Bildschirm zu erzeugen.

#### 6.4. Inhalt des Bildwiederholtspeichers

Da der GDC U82720 einen 16-Bit-Datenbus benutzt, um den Bildwiederholtspeicher zu adressieren, wird jedes Wort in Abhängigkeit davon, ob es als grafische Fläche oder als codiertes Zeichen definiert wurde, unterschiedlich interpretiert.

In der Grafik-Betriebsart werden die 16 Bit für 16 horizontal nebeneinanderliegende Bildelemente (Pixel) benutzt, die seriell an die Kathodenstrahlröhre des Displays ausgegeben werden.

In der Zeichen-Betriebsart werden einige der 16 Bit zu einem Zeichengenerator gesendet. Dort werden sie in getrennte Pixelzeilen umgewandelt, aus denen dann die verschiedenen

Zeichen gebildet werden. Die Zeilenzahl wird dem Zeichengenerator geliefert, damit er auswählen kann, welche Zeile des Zeichens angezeigt werden muß. Die Bits, die nicht zum Zeichengenerator gesendet werden können, werden benutzt, um verschiedene Attribute auszuwählen, die das Erscheinungsbild des angezeigten Zeichens beeinflussen.

Typische Attribute sind Blinken, Unterstreichen, halbe Intensität, Farbe usw. Diese Attribute werden außerhalb des GDC dekodiert, so daß sie vom Nutzer gesteuert werden können.

Bei Bildflächen zur Grafikdarstellung nimmt der GDC an, daß die 16 Bit jedes Wortes Bit für Bit seriell zum Display gesendet werden. Der Zeichenalgorithmus geht von dieser Voraussetzung aus.

Im Bild 6.5 ist der Aufbau eines Wortes im Bildwiederholtspeicher dargestellt. Im Bild 6.6 ist das Sichtbarwerden eines Wortes auf dem Bildschirm dargestellt.

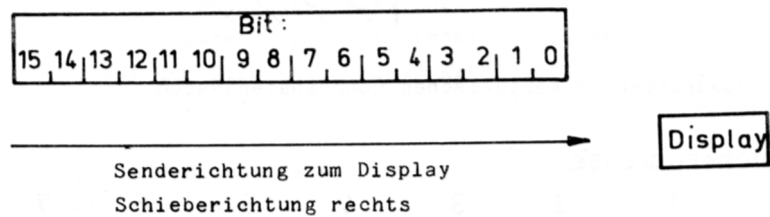


Bild 6.5: Aufbau eines Wortes im Bildwiederholtspeicher

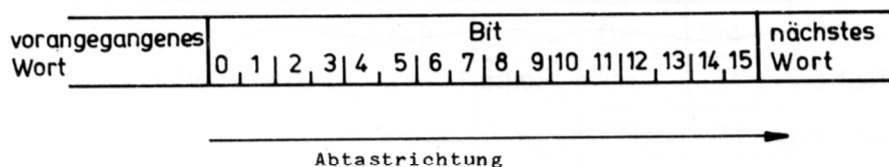


Bild 6.6: Darstellung eines Wortes auf dem Bildschirm

Beachte, daß die Bitnummern nach rechts, also in Richtung steigender Wortadressen, ansteigen.

## 6.5. Kennzeichnung einer Pixeladresse im Bildwiederholtspeicher

### 6.5.1. Definition von EAD und dAD

Der Bildwiederholtspeicher kann bis zu  $2^{22}$  (oder 4M wobei  $M = 1048576 = 2^{20}$ ) Pixel, organisiert in  $2^{18}$  (256K) Worten zu 16 Bit, enthalten. Die Adresse eines dieser Pixel wird dem GDC in zwei Teilen angegeben. Zuerst wählt eine 18 Bit-Adresse das Wort im Bildwiederholtspeicher aus, in dem sich das Pixel befindet. Als zweites zeigt ein vier Bit-Wert auf das einzelne Pixel innerhalb des Wortes. Die Wortadresse wird auch auszuführende Adresse (Execute Address, EAD) genannt und wirkt wie ein 'Cursor' im GDC. Die Startadresse des Pixels wird Punktadresse (Dot Address, dAD) genannt.



### 6.5.2. Umwandlung der x,y-Koordinaten in eine Speicheradresse

Da es wahrscheinlich ist, daß viele Anwendungen des GDC zur Aufstellung der Vektorliste die x,y-Darstellung benutzen werden, soll hier ein Beispiel für die Umwandlung der x,y-Koordinaten in eine lineare Adresse des BildwiederholSpeichers folgen.

Zur Vereinfachung soll der Ursprung (0,0) der gedachten Bildfläche in der linken oberen Ecke liegen. In diesem Punkt beginnt auch der Strahl der Kathodenstrahlröhre mit der Zeichnung des Bildfeldes. Die üblicherweise als Ursprung benutzte linke untere Ecke des Bildschirms kann durch eine einfache Substitution in den Gleichungen, die wir ableiten werden, angewendet werden.

Um eine klare Vorstellung von der Umwandlung der x,y-Koordinaten in eine lineare Adresse zu bekommen, müssen die im Bild 6.7 angegebenen Terme definiert werden.

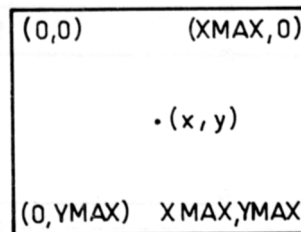


Bild 6.7: Aufbau des Bildwiederholtspeichers für eine Darstellung

Dabei ist (XMAX + 1) ein Vielfaches von 16 Pixeln bis zu 4096  
und (YMAX + 1) eine beliebige Zahl (bis zur Gesamtanzahl von 4 M Pixel im  
Bildwiederholpeicher)

Das tatsächlich dargestellte Bild kann nicht größer als der Bildwiederholtspeicher sein und muß ein geradzahliges Vielfaches von 16 Pixeln in horizontaler Richtung sein. In vertikaler Richtung dürfen es nicht mehr als 1024 Zeilen (2048 mit Zeilensprungverfahren) sein.

#### 6.5.2.1. Die 'obere linke Ecke'

Eine 'Großaufnahme' der oberen linken Ecke im Bild 6.8 enthält die Beziehungen zwischen Punkten und Worten, wie sie auf dem Bildschirm zu sehen sind.

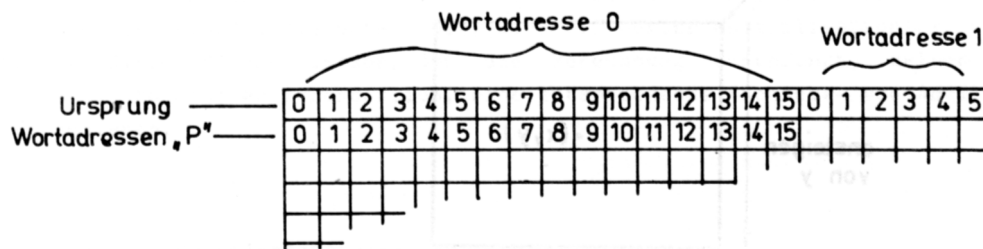


Bild 6.8: Punktdressen und ihre zugehörigen Worte

### 6.5.2.2. Wortadressen im Bildwiederholtspeicher

Unter einem globaleren Blickwinkel stellt sich die Anordnung der Wortadressen wie im Bild 6.9 dar.

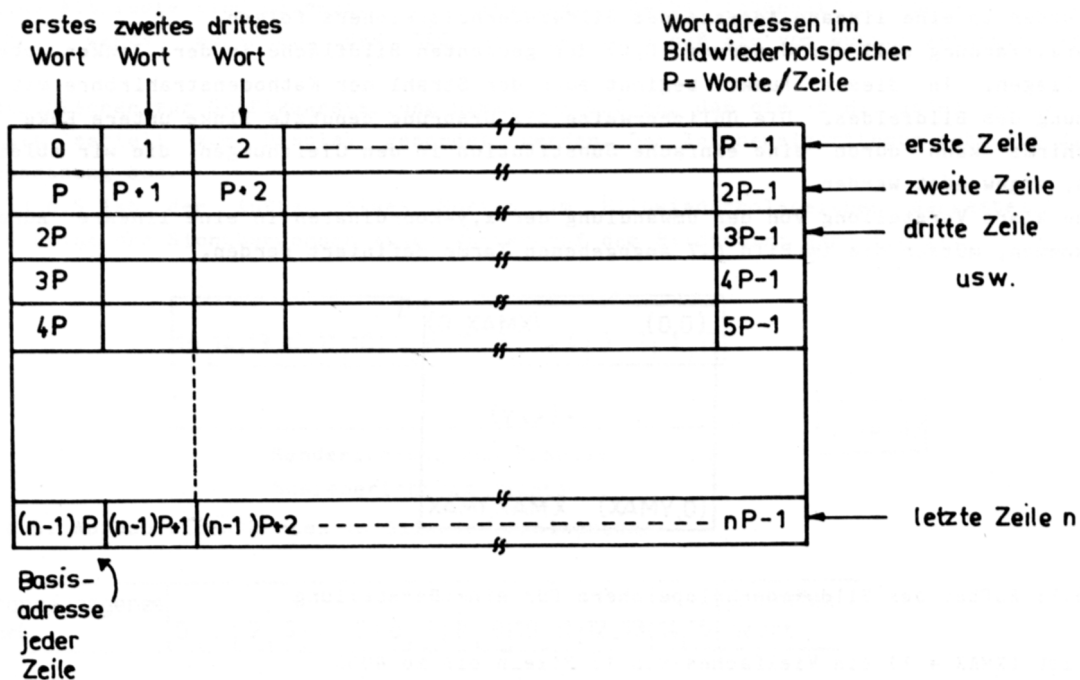


Bild 6.9: Anordnung der Wortadressen im Bildwiederholtspeicher

Dabei ist  $P$  gleich der Anzahl der 16 Bit-Worte entlang einer Zeile, mit

$$P = (X_{\text{MAX}} + 1) / 16$$

### 6.5.2.3. Das Finden von Wortadressen gegebener x,y-Koordinaten

Im Bild 6.10 sind die Richtungen für die wachsenden x- und y-Werte angegeben.

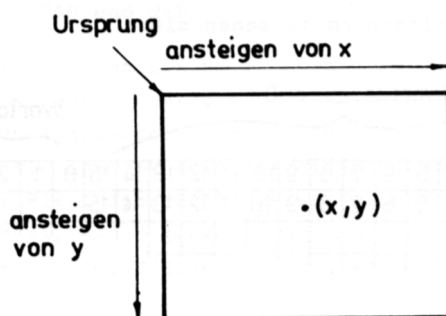


Bild 6.10: Richtungen der x- und y-Werte

Beginnend mit der y-Koordinate des Punktes, für den wir die absolute Speicheradresse finden müssen, können wir die Basisadresse, mit der die Zeile beginnt in der das Pixel liegt, finden.

Basiszeilenadresse :  $LBA = P * y$

Dazu müssen wir die Anzahl der Worte entlang der Zeile bis zum Wert der x-Koordinate addieren, um den tatsächlichen Wert der EAD zu finden:

$$EAD = LBA + \text{INTEGER}(x / 16) = Py + \text{INTEGER}(x / 16)$$

Dabei liefert die INTEGER-Operation nur den ganzzahligen Anteil der  $x/16$  Division ohne den Rest des Divisionsergebnisses.

#### 6.5.2.4. Das Finden der Bildpunktadressen

Um die Punktadresse innerhalb des Wortes, das durch die EAD-Gleichung bestimmt ist zu finden, nimmt man den Rest der oben durchgeführten Division und behandelt ihn als INTEGER-Wert der Punktadresse:

$$dAD = \text{Rest}(x / 16) * 16$$

Es folgt ein Beispiel unter Verwendung tatsächlicher Werte. Der Bildwiederholpeicher soll 512 Punkte \* 512 Zeilen groß sein ( $XMAX = YMAX = 511$ ). Der Punkt soll die Koordinaten  $(x,y) = (231,475)$  haben.

$$\begin{aligned} P &= (XMAX + 1) / 16 = 512 / 16 = 32 \text{ Worte/Zeile} \\ EAD &= Py + \text{INTEGER}(x / 16) \\ &= 32 * 475 + \text{INTEGER}(231 / 16) \\ &= 15200 + \text{INTEGER}(14,4375) \\ &= 15214 \\ dAD &= \text{Rest}(x / 16) * 16 = \text{Rest}(14,4375) * 16 \\ &= 0,4375 * 16 = 7 \end{aligned}$$

Diese Zahlen (umgewandelt in Binärzahlen) können dem GDC übergeben werden, um das interessierende Pixel zu kennzeichnen.

Es ist zu beachten, daß alle Werte positive ganze Zahlen sind. Das ist zweifellos das einfachste Format mit dem man arbeiten kann. Außerdem können alle Multiplikationen und Divisionen (mit der möglichen Ausnahme der Multiplikation mit "P" bei der Berechnung von dAD) als einfache binäre Schiebeoperationen ausgeführt werden. Ist der Bildwiederholpeicher  $2^n$  Pixel breit, ist es "P" auch. Damit wird die Multiplikation mit "P" ebenfalls nur eine einfache arithmetische Schiebeoperation sein. Falls der Bildwiederholpeicher nicht die Breite einer Zweierpotenz hat, wird ein echter Multiplikationsbefehl benötigt. Besitzt der System-Mikroprozessor, der die Berechnung durchführt, keinen Multiplikationsbefehl, so ist es erforderlich, eine entsprechende "Schiebe- und Additionsroutine" zu programmieren.

#### 6.5.2.5. Andere Koordinatensysteme

Setzt man den Nullpunkt des Koordinatensystem auf einen anderen Punkt des Bildschirms, so wird die Berechnung der physikalischen Adresse nicht komplizierter. Nachfolgend ist ein System mit dem Nullpunkt in der linken unteren Ecke im Bild 6.11 dargestellt.

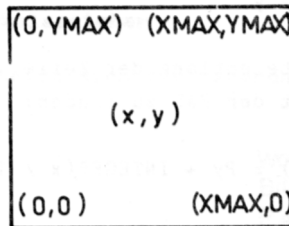


Bild 6.11: Beispiel für ein Koordinatensystem

Eine einfache Koordinaten-Transformationsbeziehung läßt sich zwischen diesen beiden Systemen definieren:

$$\begin{aligned} X &= X' \\ Y &= YMAX - Y' \end{aligned}$$

Werden diese Ausdrücke in die EAD- und dAD-Gleichungen eingesetzt, ergibt sich nur ein Unterschied:

$$\begin{aligned} \text{EAD} &= P(YMAX - Y') + \text{INTEGER}(X' / 16) \\ \text{dAD} &= \text{Rest}(X' / 16) * 16 \end{aligned}$$

Jedes andere Koordinatensystem kann genauso definiert werden.

#### 6.5.2.6. Organisation des Bildwiederholerspeichers

Bevor erläutert wird, wie die Parameter für das Zeichnen einer Darstellung berechnet werden, wird zunächst der Bildwiederholerspeicher betrachtet.

In der Grafik-Betriebsart liefert der GDC 18 Adreßleitungen, von denen 16 auch als bidirektionaler Datenbus verwendet werden. Diese Schnittstelle zum Bildwiederholerspeicher erlaubt die direkte Ansteuerung aller 262144 oder 256K Adressen, wobei jede Adresse 16 Datenbits für insgesamt 4096K Speicher enthält. Stimmt jedes Bit des Bildwiederholerspeichers mit einem Pixel auf dem Bildschirm überein (Speziellfall S/W-Bildschirm), kann das System auf unterschiedliche Art und Weise aufgebaut werden, z.B.: 2048 \* 2048, 4096 \* 4096 oder irgendeine andere Kombination. Da die gegenwärtige Bildröhrentechnologie im allgemeinen eine solch hohe Auflösung nicht ohne sehr hohe Kosten erlaubt, wird die Bildschirmgröße in den meisten Anwendungsfällen kleiner sein, wie z.B.: 512 \* 512 oder 768 \* 576.

In diesem Fall enthält der Bildwiederholerspeicher entweder mehrere Bildschirmseiten oder es wird nur ein kleiner Ausschnitt aus einer größeren Bildfläche auf dem Bildschirm dargestellt.

Der Entwurf eines Farb-Grafikdisplays erfordert spezielle Überlegungen. Der allgemein übliche Weg Farbe zu implementieren, besteht im Aufbau mehrerer Ebenen im Bildwiederholerspeicher. Dabei ist jede Ebene für eine der Grundfarben vorgesehen. Durch das gleichzeitige Senden der Videodaten aller Ebenen kann eine Vielzahl von Farben erzeugt werden.

Zum Beispiel können mit drei Ebenen (rot, blau, grün) sechs Farben sowie schwarz und weiss erzeugt werden. Zur Generierung einfarbiger Bilder (Graustufen) kann die gleiche Methode benutzt werden.

## 6.6. Der digitale Differentialanalysator

Der digitale Differentialanalysator (DDA) im GDC erzeugt mit einer fest zugeordneten internen Struktur bei sehr hoher Geschwindigkeit Darstellungen aus ihren Differentialgleichungen. Der DDA arbeitet parallel mit der RMW-Logik in einem Pipelinesystem, um die Darstellung rechentechnisch aufzubauen.

Es können Geraden und Kreise, sowie davon abgeleitete Formen, wie z.B. Rechtecke und grafische Zeichen gezeichnet werden. Weiterhin sind auch DMA-Übertragungen möglich. Die Darstellungen werden unter Verwendung des RMW-Zeichenzyklus in den Bildwiederholtspeicher gezeichnet, während der DDA die Adresse des nächsten Pixels der Darstellung berechnet.

Der im GDC verwendete Analysator benutzt eine der Richtungen der Koordinatenachsen in den Oktanten (entweder die x- oder die y-Achse) als unabhängige Achse der Darstellung. Die Anzahl der Pixel entlang dieser Richtung ist gleich der Gesamtanzahl der Pixel in diesem Darstellungssegment, da sich der Analysator entlang dieser Achse bewegt.

Die andere Achsenrichtung ist die abhängige Richtung und der Pixelabstand in dieser Richtung ist immer kleiner oder gleich dem Abstand entlang der unabhängigen Achse. Wenn ein Pixel gezeichnet wurde, kann sich die abhängige Achse vorwärtsbewegt haben oder auch nicht, was von der Art der Darstellung und der Position innerhalb der Darstellung abhängt.

Im Ergebnis liegen die Pixel der Darstellung immer innerhalb eines halben Pixels der mathematischen Idealposition der Darstellung. In Abhängigkeit vom mit DIR festgelegten Oktanten kann die unabhängige Achse in +x, -x, +y oder -y-Richtung voranschreiten.

Für die gewünschte Darstellung muß der Systemprozessor die passenden Parameter bestimmen und den DIR-Parameter benutzen, um die Korrelation zur tatsächlichen Darstellungsorientierung zu liefern.

## 6.7. Zeichnungsprozess einer Darstellung

Sobald der Kommandoprozessor des GDC alle Kommandos und Parameter interpretiert hat und der Zeichenprozess beginnt, arbeiten DDA und RMW-Logik des GDC zusammen, um die Darstellungen zu zeichnen (oder um DMA-Übertragung durchzuführen usw.). Sie sind so konfiguriert worden, um parallel miteinander arbeiten zu können, so daß der DDA schon die Adresse des nächsten Pixels bestimmt, während die RMW-Logik noch ein Pixel der Darstellung modifiziert (Pipeline-Prinzip). Damit erfolgt das Zeichnen mit der für diese internen Strukturen höchstmöglichen Geschwindigkeit.

Es gibt zwei Ebenen in dieser Pipeline-Struktur: Der DDA und die RMW-Logik.

Wenn alle Taktzyklen im GDC genutzt werden sollen, muß die erste Pixeladresse von einer Quelle außerhalb des DDA geliefert werden, damit die RMW-Logik während des ersten RMW-Zyklus etwas zu modifizieren hat.

Das ist kein Problem, da der System-Mikroprozessor in jedem Fall die erste Pixeladresse (Cursoradresse) liefern muß.

Es ist dabei wichtig zu beachten, daß der Punktezählerstand (eigentlich ein Adreßberechnungs-Zählerstand) der dem DDA geliefert wird, den ersten Zyklus, der das erste Pixel zeichnet, nicht enthält. Wenn der Punktezählerstand das Ende erreicht hat und dabei das letzte Pixel in den Bildwiederholtspeicher geschrieben wird, berechnet der DDA die Adresse des nächsten Pixels, das gezeichnet werden würde, wenn der Punktezählerbereich größer wäre. Der DDA verläßt dabei die Cursorwerte EAD und dAD und zeigt auf dieses Pixel.

## 6.8. Vorbereitung des GDC für das Zeichnen von Darstellungen

Um den GDC auf die Zeichnung von Darstellungen vorzubereiten, müssen die Einzelheiten der gewünschten Zeichenoperation durch das Kommando- und Parameter-FIFO in den GDC geladen werden. Die Art der RMW-Zyklus-Modifikation kann durch das WDAT-Kommando ohne nachfolgende Parameter ausgewählt werden. Der Cursor kann durch Anwendung des CURD-Kommandos auf die Wort- und Punktadresse (EAD und dAD) des Anfangspixels der Darstellung positioniert werden. Die Bytes 8 und 9 des Parameter-RAMs müssen mit den Zeichenpattern für durchgehende, gepunktete, gestrichelte usw. Darstellungslinien durch Verwendung des PRAM-Kommandos geladen werden.

Für das Zeichnen von grafischen Zeichen und die Ausfüllung von Flächen können die Bytes 8 bis 15 des Parameter-RAMs mit den gewünschten Pattern oder Zeichen geladen werden. In beiden Fällen wird das FIGS-Kommando benutzt, um den Differentialanalysator auf die gewünschte Darstellung einzustellen.

Ein Kommando zum Start der Darstellung (FIGD- oder GCHRD-Kommando) wird zu Beginn des eigentlichen Zeichnungsprozesses ausgesendet.

Während der RMW-Speicherzyklen können die folgenden, für das Zeichnen der gewünschten Darstellungen benötigten logischen Operationsarten ausgewählt werden: Setzen, Löschen, Komplementbildung und Ersetzen.

Das WDAT-Kommando kann benutzt werden, um den GDC für die gewünschten Operationen zu konfigurieren. Das TYPE-Feld des WDAT-Kommandos (s.a. Bild 5.10) sollte den Wortübertragungsmodus (00-Code) anwählen.

Die nachfolgende Tafel 6.1 verdeutlicht die Funktion des MOD-Feldes im WDAT-Kommando:

MOD		Operation
Bit1	Bit0	
0	0	Ersetzt das adressierte Pixelbit durch den Wert der im Patternregister ausgewählten Bitposition (REPLACE)
0	1	Bildet das Komplement des adressierten Pixelbits, wenn das ausgewählte Bit im Patternregister eine "1" enthält (COMPLEMENT)
1	0	Setzt das adressierte Pixelbit auf "0" zurück, wenn das ausgewählte Bit im Patternregister eine "1" enthält (RESET)
1	1	Setzt das adressierte Pixelbit auf "1", wenn das ausgewählte Bit im Patternregister eine "1" enthält (SET)

Tafel 6.1: Funktion des MOD-Feldes im WDAT-Kommando

Dem WDAT-Kommando müssen keine Parameter folgen, wenn es nur benutzt wird, um die RMW Operation einzustellen. Wenn dennoch Parameter folgen, würde sie der GDC in den Bildwiederholtspeicher schreiben, wie er es auch tun würde, wenn eine Datenblockübertragung geplant wäre. Die WDAT-Kommando-Sequenz ohne Parameter wird mit dem nächsten Kommandobyte beendet.

Die Parameterbytes 8 und 9 müssen die Pattern, die als Darstellung gezeichnet werden sollen, enthalten. Wenn der Zeichnungsvorgang beginnt, wird der Inhalt dieser RAM-Bytes in das Patternregister geladen und von der RMW-Hardware benutzt, um die Modifizierung der

einzelnen Pixel der Darstellung freizugeben oder zu sperren, sobald deren Adressen während des Zeichnens berechnet werden.

In der Grafik-Betriebsart werden die Bits der Patternregister so benutzt, wie sie durch einen Zähler, der nach jedem RMW-Zyklus inkrementiert wird, ausgewählt werden. Das LSB des Patternregisters wird als erstes verwendet. Der zweite RMW-Zyklus benutzt das zweite Bit usw.

Das Pattern, das der Darstellung aufgeprägt wird, wird alle 16 RMW-Zyklen wiederholt. Da vor dem Zeichnen irgendein 16 Bit Pattern in den Parameter-RAM geladen werden kann, ist es möglich, jedes beliebige modulo-16-Pattern entlang der Darstellung zu erzeugen. Das Patternregister wird immer in der selben Reihenfolge abgetastet, unabhängig von der Art und der Richtung der zu zeichnenden Darstellung.

### **6.9. Anwendung der FIGS-Parameter für das Zeichnen von Darstellungen**

Die Parameter des FIGS-Kommandos steuern die zu zeichnende Darstellung. Das erste Parameterbyte konfiguriert den Differentialanalysator (DDA) des GDC für die Grundabbildungsart. Die restlichen Parameterbytes enthalten die verschiedenen Variablen, um den DDA während des Zeichnens der Figur zu steuern. Jede Variable ist ein ganzzahliger 14-Bit-Wert der unter Verwendung von zwei Parameterbytes durch den Kommandoprozessor des GDC in das entsprechende Register geladen wird. Da die Parameter sowohl positiv als auch negativ sein können, sollte die Zweierkomplementschreibweise benutzt werden.

Die Interpretation dieser Werte durch den DDA ändert sich für die unterschiedlichen Darstellungsarten. Die fünf Zeichenvariablen werden DC, D, D2, D1 und DM genannt. Sie werden nur in dieser Anordnung durch den GDC akzeptiert.

Bei der Ausführung jedes FIGS-Kommandos haben die verschiedenen DDA-Register spezifische Anfangswerte: DC = 0, D = 8, D2 = -1 und DM = -1.

Um diese Werte zu ändern, müssen dem FIGS-Kommandobyte die entsprechenden Parameterbytes folgen. Bei manchen Darstellungsarten brauchen einige dieser Anfangswerte nicht geändert werden und folglich brauchen die entsprechenden Parameter auch nicht an den GDC gesendet werden. Der interne Kommandoprozessor des GDC nimmt die Parameter in der Reihenfolge, wie sie kommen und legt sie in fester Reihenfolge in die passenden DDA-Register ab.

Dieser Ladeprozess wird durch jedes Kommandobyte, wie z.B. das FIGD-Kommando, beendet. Deshalb ist es möglich, nur die ersten zwei oder ersten drei Variablen usw. zu laden. Es ist nicht möglich, z.B. die erste Variable zu laden, die zweite zu überspringen und dann die dritte zu laden. Das erste Parameterbyte beschreibt die Darstellung, die gezeichnet werden soll. Die oberen fünf Bits werden entsprechend gesetzt, um die verschiedenen Darstellungsarten zu kennzeichnen.

Mit Ausnahme des Slant-Flags SL (für schräge Darstellungen), das nur in Verbindung mit dem GC-Flag zur Flächenausfüllung verwendet werden kann, darf immer nur eines dieser Bits gesetzt sein. Man darf ausschließlich nur die im Bild 5.12 angegebenen Bitkombinationen benutzen, da nur diese einen fehlerfreien Zeichenbetrieb gewährleisten.

Die DIR-Bits kennzeichnen die Anfangsrichtung für den DDA, um die Orientierung der sich ergebenden Darstellung zu steuern. Üblicherweise sind die Bits für die Darstellungsart bekannt, bevor als erster Schritt irgendwelche Zeichnungsparameter bei der Bestimmung der anderen DDA-Parameter berechnet werden.



## **6.10. Parameterarten des FIGS-Kommandos**

### **6.10.1. Einführung**

Das GDC-Kommando zur Kennzeichnung einer Darstellung (bzw. Figur) (FIGS) setzt den digitalen Differentialanalysator (DDA), um eine von mehreren (RMW)-Operationen auszuführen. Der DDA kann sechs verschiedene Betriebsarten bei großer Parametervielfalt in jedem der acht Oktanten erzeugen. Nach dem das FIGS-Kommando den DDA auf die genaue Berechnung der Adressen vorbereitet hat, wird die RMW-Operation von einem der RMW-Startkommandos ausgelöst. Die fünf Bits des ersten FIGS-Parameters, die den Darstellungstyp angeben, konfigurieren den DDA für die allgemeine Betriebsart (Vektor, Kreisbogen, Rechteck usw.), die ausgeführt werden soll. Die anderen FIGS-Parameter und das einzelne zusätzlich gegebene Aufrufkommando geben die innerhalb der Darstellung durchzuführenden Operationen an. Es kann aus sechs gültigen Darstellungsarten ausgewählt werden. Nur diese angegebenen Operationen gewährleisten ein fehlerfreies Zeichnen der Darstellung.

### **6.10.2. Startkommandos für die RMW-Operation**

Nachdem das FIGS-Kommando und seine Parameter dem GDC übergeben wurden, muß der Systemprozessor die RMW-Operation mit einem der folgenden Kommandos auslösen:

- FIGD = Start des Zeichnens der Darstellung
- GCHRD = Start der Ausfüllung von Flächen und der Zeichnung von Zeichen
- WDAT = Einschreiben von Daten in den Bildwiederholtspeicher
- RDAT = Lesen von Daten aus dem Bildwiederholtspeicher
- DMAW = Auslösen der DMA-Schreibsequenz
- DMAR = Auslösen der DMA-Lesesequenz

Jedes dieser Kommandos (s.a. Abschn. 5.) kann bei einer oder verschiedenen Darstellungsarten benutzt werden.

### **6.10.3. Korrelation zwischen Kommando und Darstellungsart**

In einer Zusammenfassung sind in Tafel 6.2 die möglichen verschiedenen Darstellungsarten den sie aufrufenden Startkommandos gegenübergestellt.

Kommando	Erlaubte RMW-Operationen	SL	R	A	GC	L
FIGD	Zeichnen einer Geraden oder eines Vektors	0	0	0	0	1
	Zeichnen von Bogen oder Kreis	0	0	1	0	0
	Zeichnen eines Rechtecks	0	1	0	0	0
	Zeichnen eines einzelnen Punktes	0	0	0	0	0
GCHRD	Ausfüllen von Flächen	0	0	0	1	0
	Ausfüllen von schiefen Flächen	1	0	0	1	0
WDAT	Schreiben eines einzelnen Wortes	0	0	0	0	0
	Schreiben aufeinanderfolgender Worte	0	0	0	0	0
	Zeichenmode-Schreiben	0	0	0	0	0
RDAT	Lesen eines einzelnen Wortes	0	0	0	0	0
	Lesen aufeinanderfolgender Worte	0	0	0	0	0
	Zeichenmode-Lesen	0	0	0	0	0
DMAW	DMA Schreibsequenz	0	0	0	0	0
DMAR	DMA Lesesequenz	0	0	0	0	0

Tafel 6.2: Erlaubte RMW-Operationen

### 6.11. Das Zeichnen von Vektoren

Das Zeichnen von Vektoren erfordert neun Parameterbytes, die dem FIGS-Kommandobyte folgen. Im ersten Parameterbyte muß das "L"-Bit gesetzt und die Richtung des Oktanten festgelegt sein. Dann müssen vier Parameterbyte-Paare folgen, welche die in die DDA Register des GDC geladen werden: DC, D, D2 und D1. Diese Werte müssen durch den System-Mikroprozessor in Abhängigkeit von den gewünschten Anfangs- und Endpunkten des Vektors bestimmt werden.

Beim Zeichnen von Vektoren arbeitet der GDC mit einer Methode, die auf abhängigen und unabhängigen Achsen basiert. Der Systemmikroprozessor muß in Abhängigkeit von der Richtung des Oktanten, in dem der Vektor liegt, bestimmen, ob die x- oder die y-Achse die abhängige bzw. unabhängige Achse ist. Durch Überprüfung der Endpunktkoordinaten des Vektors läßt sich die abhängige und die unabhängige Achse ermitteln.

Die Ergebnisse davon können dann benutzt werden, um die erforderlichen Zeichenparameter sehr schnell vorzubereiten. Der erste Schritt in diesem Prozess besteht darin, die Anzahl der durch den Vektor entlang jeder Achse überspannten Pixelpositionen zu berechnen:

$$\text{DELTA X} = \text{X(Ende)} - \text{X(Anfang)}$$

$$\text{DELTA Y} = \text{Y(Ende)} - \text{Y(Anfang)}$$

Dabei sind die Anfangswerte die x,y-Koordinaten des Anfangspunktes und die Endwerte die x,y-Koordinaten des Endpunktes des Vektors. Die Deltawerte können sowohl positiv als auch negativ sein. Man beachte dabei, daß die Deltawerte gleich der Anzahl der Pixel entlang der jeweiligen Achse, aber nicht gleich der Anzahl der zu zeichnenden Pixel sind. Da das

erste Pixel in die Berechnung dieser Gleichung nicht mit einbezogen wird, sind die Deltawerte um ein Pixel kürzer als es notwendig ist, den Abstand zu überdecken. Im ersten Zeichenzyklus zeichnet der GDC das Pixel, auf das der Cursor zeigt, während der DDA gleichzeitig die Adresse des zweiten Pixel berechnet. Die Punktzahl, die dem DDA als DC-Parameter übergeben wird, bezieht sich auf die Anzahl der Pixeladressen, die berechnet und gezeichnet werden müssen, nach dem das erste Pixel gezeichnet wurde. Wenn das letzte Pixel gezeichnet wurde (d.h. der Zählerstand von DC hat den Wert "0" erreicht), geht der Cursor weiter zu dem Punkt, der als nächstes Pixel der Darstellung gezeichnet werden würde, wenn der DC-Wert größer gewesen wäre.

Dabei muß man beachten, daß die positive Richtung der y-Achse nach unten gerichtet ist, wenn man den Bildwiederholtspeicher auf der Bildfläche sieht. Das spiegelt die Tatsache wieder, daß die linke obere Ecke die Position "0" des Bildwiederholtspeichers ist. Andere Koordinatensysteme mit abweichender Ursprungsposition können leicht in dieses Format umgewandelt werden.

Im Bild 6.12 ist die Richtungsangabe für die einzelnen Quadranten angegeben.

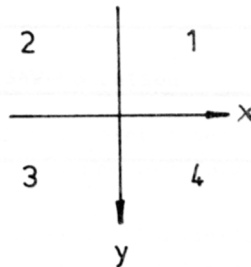


Bild 6.12: Richtungsdefinition der Quadranten

Der erste Schritt zur Bestimmung der Oktantenrichtung des Vektors besteht darin, die Richtung des Quadranten zu finden. Diese kann leicht durch die Untersuchung der Vorzeichen der beiden Deltawerte ermittelt werden:

Vorzeichen DELTA X	Vorzeichen DELTA Y	Quadrant
positiv	negativ	1
negativ	negativ	2
negativ	positiv	3
positiv	positiv	4

Im Bild 6.13 ist die Definition der Oktantenrichtung angegeben.

Innerhalb des Quadranten kann die Oktantenrichtung durch Vergleich der absoluten Größe der beiden Deltawerte bestimmt werden:

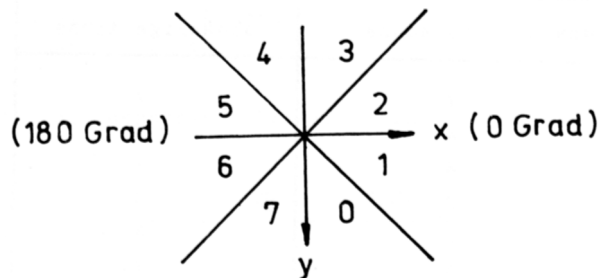


Bild 6.13: Richtungsdefinition der Oktanten

Wenn  $ABS(DELTA\ X) > ABS(DELTA\ Y)$

und der Quadrant ist:

1  
2  
3  
4

dann ist der Oktant:

2  
5  
6  
1

Wenn  $ABS(DELTA\ X) < ABS(DELTA\ Y)$

und der Quadrant ist:

1  
2  
3  
4

dann ist der Oktant:

3  
4  
7  
0

Falls  $ABS(DELTA\ X) = ABS(DELTA\ Y)$ , liegt der Vektor auf der diagonalen Oktantengrenze innerhalb des Quadranten. In diesem Fall kann der ungerade Oktant des Quadranten ausgewählt werden, um die Symmetrie der Zuordnung der Vektorenrichtung entlang der Koordinatenachse nicht zu stören. Diese Tests lassen sich mit hoher Geschwindigkeit durchführen, wenn die Vorzeichen der beiden DELTA-Terme und ihre Unterschiede in einer Softwaretabelle positioniert werden.

Falls die Vektorenrichtung innerhalb eines einzelnen Oktanten bestimmt wurde, besteht der nächste Schritt in der Bestimmung der unabhängigen Zeichnungsachse und der Berechnung der Zeichnungsparameter.

Die unabhängige Achse ist die x-Achse für die Oktanten, in denen

$ABS(DELTA\ X) > ABS(DELTA\ Y)$  gilt,

und sie ist die y-Achse für die Oktanten, in denen

$ABS(DELTA\ X) \leq ABS(DELTA\ Y)$  gilt.

Wie aus der Zeichnungsrichtung des Oktanten entnommen werden kann, ist die y-Achse die unabhängige Achse für die Oktanten, in denen die Gerade in y-Richtung länger als in x-Richtung ist.

Die unabhängigen Achsen in den einzelnen Oktanten sind nachfolgend angegeben:

Oktant	unabhängige Achse	abhängige Achse
0	y	x
1	x	y
2	x	y
3	y	x
4	y	x
5	x	y
6	x	y
7	y	x

Die Zeichnungsparameter für den DDA des GDC können aus den Absolutwerten von DELTA X und DELTA Y berechnet werden. Der Parameter für die Oktantenrichtung (DIR) steuert den DDA, in Abhängigkeit von der angegebenen Oktantenrichtung entlang der unabhängigen Achse durch Inkrementieren oder Dekrementieren der x- oder y-Adressen voranzuschreiten. Ebenso sind die erforderlichen Manipulationen, um entlang der abhängigen Achse voranzuschreiten, implizit im DIR-Parameter angegeben.

Die Gleichungen zur Bestimmung der Zeichnungsparameter einer Geraden:

(I = unabhängige Achse, D = abhängige Achse)

$DC = \text{ABS}(\text{DELTA } I)$   
 $D = [2 * \text{ABS}(\text{DELTA } D)] - \text{ABS}(\text{DELTA } I)$   
 $D2 = 2 * [\text{ABS}(\text{DELTA } D) - \text{ABS}(\text{DELTA } I)]$   
 $D1 = 2 * \text{ABS}(\text{DELTA } D)$

Es ist notwendig, in diesen Gleichungen die DELTA I- und DELTA D-Terme durch die entsprechenden DELTA X und DELTA Y Werte zu ersetzen (entsprechend der Zeichnungsrichtung des Oktanten). Die Berechnung von D und D2 erfordert eine 16-Bit-Integer-Arithmetik (2er Komplement). Die Werte, die dem GDC übergeben werden, sollten nur die unteren 14 Bit enthalten.

Sowohl DC als auch D1 sind positive ganze Zahlen aus dem Bereich von 0 bis  $(2^{14})-1$ . Beachte dabei, daß der Parameter DC die Anzahl der Pixel minus 1 enthält, da die Zählung das erste Pixel nicht enthält. Die anderen drei Parameter enthalten codierte Werte für den Anstieg der Geraden, die vom DDA benutzt werden um zu entscheiden, wann er beim fortschreitenden Zeichenprozess entlang der abhängigen Achse vorrücken muß.

## 6.12. Beispiel für das Zeichnen eines Vektors

Ein Beispiel für die Berechnung der Zeichnungsparameter eines Vektors soll helfen, das oben angegebene Verfahren verständlich zu machen.

Beachte dabei, daß alle Zahlen als vorzeichenbehaftete Dezimalzahlen dargestellt werden, während sie in der Realität dem GDC als Zweierkomplement oder ohne Vorzeichen in binärer Form übergeben werden müssen. Die Startadresse wird dem GDC als vorzeichenfreier ganzzahliger Wert durch das CURS-Kommando übergeben. Dabei muß die lineare Adressierungstechnik des BildwiederholSpeichers benutzt werden.

Im Beispiel sollen die Anfangs- und Endpunkte, ausgedrückt als geordnete (x,y)-Paare, folgende Werte haben:

Anfangspunkt = (100,100)  
 Endpunkt = (78,34)

Auf dem Bildschirm einer Kathodenstrahlröhre würde das wie in Bild 6.14 aussehen.

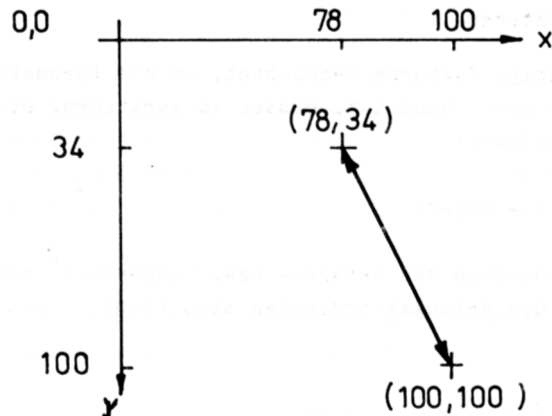


Bild 6.14: Beispiel für eine Vektordarstellung

Der erste Schritt zur Bestimmung der Richtung des Vektors besteht in der Berechnung der DELTA X- und DELTA Y-Werte:

$$\text{DELTA X} = \text{X(Ende)} - \text{X(Anfang)} = 78 - 100 = -22$$

$$\text{DELTA Y} = \text{Y(Ende)} - \text{Y(Anfang)} = 34 - 100 = -66$$

Beide Werte sind negativ. Deshalb liegt der Vektor im zweiten Quadranten. Um die Richtung des Oktanten innerhalb des Quadranten bestimmen zu können, ist festzuhalten, daß der Absolutwert von DELTA Y größer als der Absolutwert von DELTA X ist, woraus sich die Richtung des Vektors im vierten Oktanten ergibt, d.h.: DIR = 4.

Im vierten Oktanten ist die y-Achse die unabhängige und die x-Achse die abhängige Achse. Deshalb können wir definieren:

$$\text{DELTA I} = \text{DELTA Y} = -66$$

$$\text{DELTA D} = \text{DELTA X} = -22$$

Die Zeichnungsparameter des FIGS-Kommandos für eine Gerade können leicht wie folgt berechnet werden:

$$\text{DC} = \text{ABS}(\text{DELTA I}) = 66$$

$$\begin{aligned} \text{D} &= [2 * \text{ABS}(\text{DELTA D})] - \text{ABS}(\text{DELTA I}) \\ &= [2 * 22] - 66 = -22 \end{aligned}$$

$$\begin{aligned} \text{D2} &= 2 * [\text{ABS}(\text{DELTA D}) - \text{ABS}(\text{DELTA I})] \\ &= 2 * 22 - 66 = -22 \end{aligned}$$

$$\text{D1} = 2 * \text{ABS}(\text{DELTA D}) = 2 * 22 = 44$$

Wenn diese Werte in geeigneter binärer Form ausgedrückt und zusammen mit den anderen Setz-Kommandos und Parametern in den GDC geladen werden, zeichnet der GDC den geforderten Vektor in 67 RMW-Zyklen. Sowohl DC als auch D1 sind beim Zeichnen von Vektoren immer

positive ganze Zahlen. Sie können den ganzen 14 Bit breiten Wertebereich von 0 bis  $(16384 - 1)$  einnehmen. Andererseits muß für D und D2 die Zweierkomplementschreibweise für den Bereich von  $(8192 - 1)$  bis  $-8192$  verwendet werden.

### 6.13. Zeichenalgorithmus für Vektoren

Nachfolgend werden einige beliebige Vektoren betrachtet, um die Parameter die der GDC während des Zeichnens von Vektoren benötigt, besser zu verstehen. Die Gleichung einer Geraden läßt sich wie folgt ausdrücken:

$$Y2 = (dY/dX) * (X2 - X1) + Y1$$

wobei  $X1$ ,  $X2$ ,  $Y1$  und  $Y2$  die Koordinaten des Anfangs- bzw. Endpunktes des Vektors sind. Normiert man den Vektor, d.h. die Anfangskoordinaten sind  $(0,0)$ , nimmt die Gleichung die folgende Form an:

$$Y = (ADY/ADX) * X$$

Dabei sind  $X$  und  $Y$  immer die Koordinaten des folgenden Pixels des Vektors. Da in einem Rasterzeichnungssystem die  $X$ - und  $Y$ -Werte nur positive ganze Zahlen sein können, läßt sich die Gleichung folgendermaßen ausdrücken:

$$Y = (ADY/ADX) * X = I + F$$

Dabei sind  $X$  die aufeinanderfolgenden Koordinaten  $(0,1,2,...)$ ,  $I$  der ganzzahlige Anteil der  $Y$ -Koordinaten  $(0,1,2,...)$  und  $F$  der Bruchteil der  $Y$ -Koordinate  $(0 < F < 1)$ . Der Wert  $F$  in dieser Gleichung muß auf die nächste ganze Zahl gerundet werden. Für den Fall, daß  $F = 0,5$  ist, wird der  $I$ -Wert um eins inkrementiert, d.h. es wird aufgerundet. Um zu entscheiden, ob auf- oder abgerundet wird, muß der Wert  $F$  durch den GDC an jeder Pixeladresse ausgewertet werden. Stellt man die Gleichung nach dem Bruchteil  $F$  um, erhält man folgende Form:

$$F = (ADY/ADX) * X - I$$

Aufgerundet wird, wenn:  $F > 0,5$   
oder:  $(ADY/ADX) * X - I > 0,5$  ist.

Durch Multiplikation mit zwei und Subtraktion von eins ergibt sich:

$$2 * (ADY/ADX) * X - 2I - 1 > 0$$

Nach Multiplikation mit  $ADX$  erhält man:

$$D = 2ADY * X - (2I + 1) * ADX > 0$$

Die innere logische Struktur des GDC U82720 benutzt diese Gleichung, um jede aufeinanderfolgende Pixeladresse zu berechnen. Da sie immer nur das nächste Pixel berechnet, das gezeichnet werden muß, kann die Gleichung weiter vereinfacht werden. Das erste Pixel des Vektors wird an dem Punkt gezeichnet, der durch das CURS-Kommando als Anfangspunkt des Vektors gekennzeichnet wurde. An diesem Punkt ist  $X = 0$  und  $I = 0$ . Für



das nächste Pixel wird der X-Wert inkrementiert und I bleibt noch 0.

Da  $ADX > ADY$  ist, muß der Y-Wert nur inkrementiert werden wenn  $F > 0,5$  ist. Um den Wert von F abzuschätzen, muß die og. Gleichung abgeschätzt werden.

$$D = 2ADY * X - (2I + 1)ADX$$

für  $X = 1$  und  $I = 0$  ergibt sich

$$D = 2ADY - ADX$$

Auf Grund des Vorzeichens von D wird der nächste Wert der Y-Koordinate (I) entweder auf- oder abgerundet. Wenn  $D > 0$  wird I inkrementiert.

Ist  $D < 0$  wird I nicht verändert. In Abhängigkeit vom Vorzeichen des Parameters D kann das nächste Pixel entweder über oder unter der idealen Geraden gezeichnet werden.

Fall 1:  $D < 0$

Der einzige Parameter der zwischen zwei aufeinanderfolgenden Pixel inkrementiert wird ist der X-Wert und die og. Gleichung nimmt folgende Form an:

$$D1 = \underset{\text{nächstes Pixel}}{2ADY(X+1)} - \underset{\text{aktuelles Pixel}}{(2I+1)ADX} - 2ADY * X - (2I+1)ADX = 2ADY$$

Dieser Wert wird zum Wert von D für das nächste Pixel addiert, wenn D negativ war.

Fall 2:  $D > 0$

Sowohl X als auch I werden inkrementiert und die o.g. Gleichung nimmt folgende Form an:

$$D2 = \underset{\text{nächstes Pixel}}{(2ADY(X+1) - (2(I+1)+1) * ADX)} - \underset{\text{aktuelles Pixel}}{(2ADY * X - 2 * (I+1)ADX)} \\ = 2(ADY-ADX)$$

Dieser Wert wird zu D addiert, wenn D positiv oder Null war.

Die Parameter D, D1 und D2 müssen dem GDC geliefert werden, bevor der Zeichnungsprozess beginnt. Der einzige weitere Parameter der noch notwendig ist, ist die Gesamtanzahl der Pixel ohne das erste Pixel. Diese Zahl ist immer gleich ADX. Die sieben Parameter, die der System-Mikroprozessor dem GDC liefern muß, sind:

1. EAD = Wortadresse des ersten Pixels
2. dAD = Adresse des Pixel innerhalb der EAD
3. DIR = eine der acht möglichen Richtungen
4. DC = ADX oder Gesamtzahl der zu zeichnenden Pixel - 1
5. D =  $2ADY - ADX$
6. D1 =  $2ADY$
7. D2 =  $2(ADY-ADX)$

Die letzten drei Parameter müssen als vorzeichenbehaftete ganze Zahlen in Zweierkomplementschreibweise in den GDC geladen werden.

#### 6.14. Zeichnen von Kreisbögen und Kreisen

Der DDA des GDC kann Kreisbögen bis zu 45 Grad innerhalb eines Oktanten zeichnen. Kreise können unter Verwendung von acht Bögen, die alle den selben Krümmungsradius und Mittelpunkt haben und die jeweils 45 Grad des Vollkreises überdecken, gezeichnet werden. Da das gleiche Prinzip von unabhängigen und abhängigen Achsen für das Zeichnen von Vektoren benutzt wird, ist das Zeichnen von Kreisbögen auf einen Oktanten je Aufruf begrenzt. Damit können die X- und Y-Pixeladressen immer in die selbe Richtung voranschreiten.

Um einen Kreisbogen zu zeichnen, wird dem GDC folgendes übergeben:

- eine Startadresse durch das CURS-Kommando
- Pattern, die durch das PRAM-Kommando in die Bytes 8 und 9 geladen werden (falls die alten Pattern anders waren oder darüber Zweifel bestehen)
- eine RMW-Operation wird durch WDAT-Kommando ausgewählt (falls die vorhergegangene Auswahl anders war oder Zweifel darüber bestehen)
- ein FIGS-Kommando, um den DDA auf Kreisbögen einzustellen
- und ein FIGD-Kommando, um den Zeichenprozess auszulösen

Der einzige Unterschied zur Kommandofolge beim Zeichnen von Vektoren besteht in den Parametern des FIGS-Kommandos.

#### 6.15. Positionierung des Cursors beim Zeichnen von Kreisbögen

Das Kommando zur Positionierung des Cursors (CURS) wird zum Zeichnen eines Kreisbogens benutzt. Dabei wird der Cursor auf dem Punkt des zu zeichnenden Bogens positioniert, wo die x- oder y-Achse des jeweiligen Oktanten geschnitten wird.

Der Bogen wird weg von dieser Achse in Richtung der diagonalen Oktantengrenze gezeichnet. Wenn Kreise gezeichnet werden gibt es, obwohl acht Bögen notwendig sind um einen Vollkreis zu zeichnen nur vier Startpunkte, von denen jeder für zwei Bögen genutzt wird. Im Bild 6.15 sind die Richtungen der Winkel angegeben und Startpunkte angegeben.

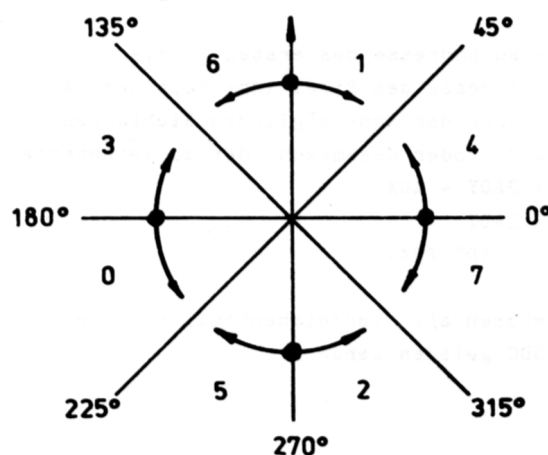


Bild 6.15: Richtungsdefinitionen der Winkel

Entsprechend Bild 6.15 ist zu beachten, daß es vier Startrichtungen gibt, von denen jede in Abhängigkeit vom festgelegten DIR-Wert im Uhrzeigersinn oder gegen den Uhrzeigersinn gekrümmt sein kann. Die Beziehungen der verschiedenen DIR-Werte zur Zeichnungsrichtung sind im Bild 6.16 deutlicher dargestellt.

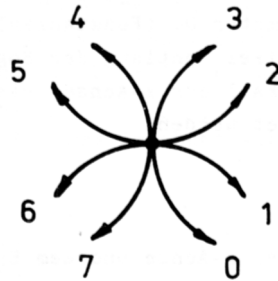


Bild 6.16: Zeichnungsrichtungen der Winkel gegenüber einem Punkt

Kreisbögen die über die Oktantengrenzen gezeichnet werden sollen, müssen in Segmenten gezeichnet werden, die die Oktantengrenzen nicht überschreiten. Wenn die Punktzahl (DC), die dem DDA mit dem FIGS-Kommando übergeben wurde, dazu führt, daß die Oktantengrenze überschritten wird, wird die daraus resultierende Figur nur bis zu dem Punkt gezeichnet, an dem sie den Anfangsoktant verläßt.

Die letzten drei Parameter müssen als vorzeichenbehaftete ganze Zahlen in Zweierkomplementschreibweise in den GDC geladen werden.

#### 6.16. Berechnung der FIGS-Parameter für Kreisbögen

Die Startwerte, die der Softwaresteuerung des GDC für das Zeichnen von Kreisen und Kreisbögen übergeben werden, sind oft der Mittelpunkt des Kreises (oder der Mittelpunkt des Krümmungsradius eines Bogens), der Radius und für Kreisbögen der Anfangs- und der Endwinkel, die jeweils vom Mittelpunkt des Krümmungsradius gemessen werden.

Anhand eines einfachen Beispiels im ersten Quadranten soll die Spezifikation eines Kreisbogens im Bild 6.17 gezeigt werden.

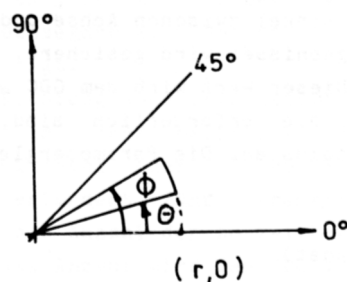


Bild 6.17: Darstellung eines Winkels in einem Oktanten

Im Bild 6.17 ist als Anfangszeichenrichtung DIR = 4 festgelegt und die y-Achse ist die unabhängige Achse. Dabei ist zu beachten, daß ein Vektor, der vom Mittelpunkt des Krümmungsradius des Kreisbogens in dieses Gebiet gezeichnet wird, die Richtung DIR = 2 hat. Ein Kreisbogen hat eine Anfangsrichtung, die senkrecht zum Radius des Bogens liegt, d.h. seine Anfangsrichtung ist senkrecht zu der des Vektors. Eine 90 Grad-Drehung der

Richtung entspricht einer Addition oder Subtraktion von zwei zum bzw. vom DIR-Wert. Der Bogen muß durch den DDA vom Startpunkt auf der Achse aus berechnet werden, auch wenn der geforderte Bogen die Achse gar nicht berührt. Deshalb muß eine Punktzahl angegeben werden, die alle Punkte zwischen der Achse und dem Endpunkt des Bogens enthält. Da die Anzahl der Pixel (Punkte) entlang der unabhängigen Achse gleich der Gesamtzahl der zu berechnenden Pixel ist, kann der Parameter DC (Punktzahl) mittels der Sinusfunktion berechnet werden. Die Anzahl der Pixel entlang der Bogenlänge ist gleich der DELTA Y-Pixelanzahl von der Achse (in diesem Fall der x-Achse) bis zum Ende des Kreisbogens. Dieses DELTA Y kann wie folgt berechnet werden:

$$\text{DELTA Y} = \text{Radius} * \sin \Theta$$

Dabei ist Theta der Winkel zwischen der x-Achse und dem Ende des Bogens; der Radius wird in Anzahl der Pixel gemessen.

Das Bild 6.18 ist die dazugehörige Darstellung angegeben.

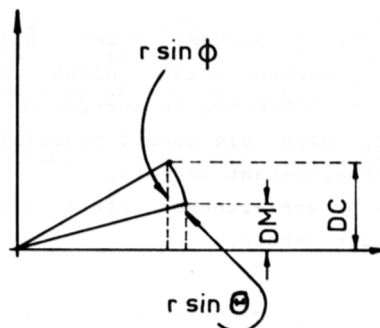


Bild 6.18: Festlegung der Bogenlänge

Das Ergebnis sollte auf die nächst größere ganze Zahl aufgerundet werden, da der GDC nur eine ganzzahlige Menge von Punkten zeichnen kann.

Dieser Wert kann als DC-Parameter verwendet werden. Die andere Größe, die berechnet werden muß, ist die Anzahl der Pixel, die berechnet, aber nicht gezeichnet werden müssen, wenn der DDA von der entsprechenden Achse zum Anfangspunkt des Bogens unterwegs ist. Dieser Wert wird in der selben Art und Weise wie die Gesamtpunktzahl berechnet. Der Unterschied besteht nur darin, daß der Winkel zwischen Achse und Anfang des Bogens (Phi) verwendet wird. Durch abrunden des Ergebnisses wird gesichert, daß der gezeichnete Bogen alle Pixel überdeckt, die er berührt. Dieser Wert wird dem GDC als Parameter DM übergeben. Die anderen beiden Parameter, die erforderlich sind, um einen Kreisbogen zu zeichnen, geben dem DDA den Krümmungsradius an. Die Parametergleichungen für alle Parameter sehen folgendermaßen aus:

$$\begin{aligned} \text{DC} &= \text{Radius} * \sin \Theta && \text{(aufgerundet)} \\ \text{D} &= \text{Radius} - 1 \\ \text{D2} &= 2 * (\text{Radius} - 1) \\ \text{D1} &= -1 && \text{(in Zweierkomplementschreibweise sind das alles "1")} \\ \text{DM} &= \text{Radius} * \sin \phi \end{aligned}$$

Dabei ist der Radius ein ganzzahliger Wert von Pixeln. Man beachte dabei, daß der Parameter D1 gleich seinem Ausgangswert ist. Da aber beim Laden der Parameter keine Parameter übersprungen werden dürfen und DM gesetzt werden muß, muß auch der Ausgangswert von D1 in der Ladesequenz enthalten sein. Beim Zeichnen eines Kreises ist Theta = 45 Grad

und  $\Phi = 0$  Grad. Damit wird  $DC = \text{Radius} * 1/\sqrt{2}$  und  $DM = 0$ .

In diesem Zusammenhang ist es interessant zu erwähnen, daß beim Zeichnen von Kreisen oder aneinanderliegenden Kreisbögen der gemeinsame Anfangspunkt auf der Achse zweimal geschrieben wird, einmal für jedes angrenzende Bogensegment. Wenn bei den RMW-Operationen die Komplementbildung (Complement Mode) benutzt wird, wird beim ersten Bogen das gemeinsame Bit gesetzt und der zweite Bogen wird es wieder zurücksetzen.

In diesem Fall wird die Verwendung von  $DM = 1$  für den zweiten Bogen verhindern, daß dieses Pixel zweimal gezeichnet wird.

Die vier möglichen Anfangspunkte für das Zeichnen von Kreisbögen sind als geordnete x,y-Paare dargestellt:

```
( Radius,0 )  
( 0,Radius )  
( -Radius,0 )  
( 0,-Radius )
```

Dabei wird angenommen, daß der Mittelpunkt des Kreises bzw. des Krümmungsradius auf 0 liegt. Um ihn zu Koordinaten außerhalb des Ursprungs zu verschieben, ist es notwendig, die Koordinaten des versetzten Mittelpunktes zu den oben beschriebenen Koordinaten zu addieren:

```
( r + Xm,Ym )  
( Xm,r + Ym )  
( Xm - r,Ym )  
( Xm,Ym - r )
```

wobei  $r$  der Radius,  $X_m$  die x-Koordinate und  $Y_m$  die y-Koordinate des Mittelpunktes ist.

#### 6.17. Das Zeichnen von Rechtecken

Der GDC kann vier Geradensegmente in Form eines Rechtecks durch eine einfache CURS-, FIGS- und FIGD-Kommandofolge zeichnen. Dabei wird angenommen, daß die Zeichnungspattern für die RMW-Operation bereits gesetzt sind. Beginnend an der Cursorposition zeichnet der DDA nacheinander vier Geradensegmente, wobei nach jedem Schnittpunkt eine 90 Grad Drehung gegen den Uhrzeigersinn erfolgt.

Die am Anfang festgelegte Richtung kann irgendeiner der acht DIR-Werte sein, so daß das Zeichnen der Darstellung in einer beliebigen Ecke des Rechtecks beginnen kann. Das Rechteck kann entweder horizontal oder um 45 Grad gedreht dargestellt werden. Das Zeichnen von Rechtecken erfordert fünf Parameter für das FIGS-Kommando. Der System-Mikroprozessor braucht nur die Anzahl der zu zeichnenden Pixel der beiden Seiten des Rechtecks zu berechnen. Diese Anzahl wird dem GDC in Form zweier Werte übergeben.

Der erste Wert gibt die Anzahl der Pixel in Richtung des DIR-Parameters an, der entlang der ersten Seite des zu zeichnenden Rechtecks zeigt. Der Zweite gibt die Länge der zweiten Seite an, die senkrecht zur ersten Seite steht. Diese Werte kann man sich als Länge und Breite des Rechtecks denken. Die genauen Werte, die dem GDC als Parameter des FIGS-Kommandos übergeben werden müssen, sind:

DC = 3

D = (Anzahl der Pixel in der mit DIR festgelegten Richtung) - 1

D2 = (Anzahl der Pixel senkrecht zur Anfangsrichtung) - 1

D1 = -1

DM = D

Das Beispiel-Rechteck ist im Bild 6.19 dargestellt.

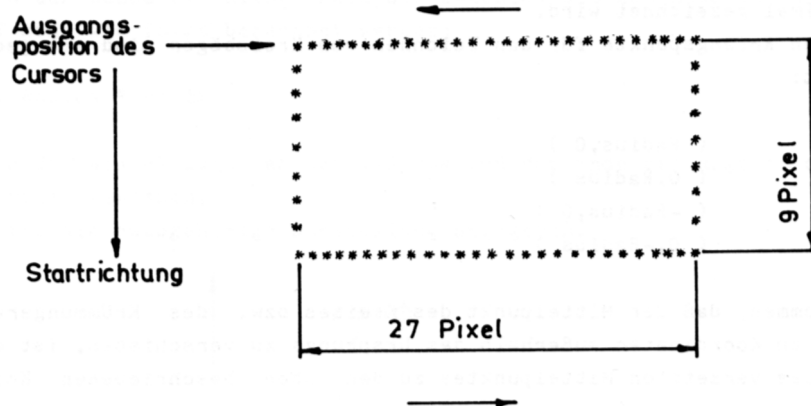


Bild 6.19: Beispiel für Darstellung eines Rechtecks

Die 'Sterne' stellen die während des Zeichnens modifizierten Pixel dar. Die Pfeile zeigen die Richtungen an, die während des Zeichenprozesses mit folgenden Parametern benutzt werden:

DIR = 0

DC = 3

D = 8

D2 = 26

D1 = -1

DM = 8

#### 6.18. Ausfüllung von Flächen und Darstellung von grafischen Zeichen

Der Inhalt der Bytes 8 bis 15 des Parameter-RAM kann an jeder beliebigen Pixelposition in den Bildwiederholpeicher übertragen werden. Wenn der Parameter-RAM mit den gewünschten Pattern oder Zeichen gefüllt ist, arbeitet er als temporärer Auffangpuffer. Dieser kann, falls er nicht überschrieben wird, so oft es gewünscht wird an jeder Pixelposition und in jeder der acht um 45 Grad gedrehten und durch den DIR-Wert ausgewählten Richtungen in den Bildwiederholpeicher geschrieben werden.

Zusätzlich können die Pattern während des Einschreibens durch den Zoom-Faktor von 1 bis 16 vergrößert werden. Die Vergrößerung erfolgt durch Pixelwiederholung in x- und y-Richtung. Bei dieser Zeichenart werden alle Pixel innerhalb einer rechteckigen oder einer schrägen rechteckigen Fläche mit dem Inhalt des Parameter-RAM ausgefüllt. Der RAM wird vom LSB des Byte 15 bis zu dessen MSB, und dann weiter zum MSB des Byte 14 bis zu dessen LSB, von dort aus weiter zum LSB des Byte 13 usw. abgetastet, bis das MSB von Byte 8 ausgegeben wurde. Ist die auszufüllende Fläche größer als 8 \* 8 Pixel, wird der Abtastprozess so lange fortgesetzt, bis die Fläche ausgefüllt ist. Falls die Fläche kein

ganzzahliges Vielfaches (oder kleiner!) als  $8 * 8$  Pixel ist, wird nur ein Teil des Parameter-RAMs ausgelesen und zum Bildwiederholpeicher gesendet, so daß das richtige Pattern, soweit wie nötig erhalten bleibt.

Die Pattern können durch den Zoom-Faktor bis zu 16fach vergrößert gezeichnet werden. Jede ganze Zahl zwischen 1 und 16 kann als Vergrößerungsfaktor verwendet werden. Die Vergrößerung erfolgt durch Pixelwiederholung, d.h. bei einem Faktor von Zwei wird jedes Bit im Parameter-RAM als vier Pixel-Block der Größe  $2 * 2$  gezeichnet. Sobald das Pattern mit einem bestimmten Zoom-Faktor in den Bildwiederholpeicher geschrieben wurde, beeinflussen danach folgende Änderungen dieses Zoom-Faktors die vorher geschriebenen Pattern nicht.

Diese Fähigkeit wird durch das Zoom-Kommando gesteuert. Das Zoom-Kommando steuert ebenfalls den Zoom-Vergrößerungsfaktor für die Rasterdarstellung. Im ersten (und einzigen) Parameter des Zoom-Kommandos werden die oberen vier Bits für den Display-Zoom-Faktor (DISP) und die unteren vier Bits für den Schreib-Zoom-Faktor (GCHR) benutzt (s.a. Bild 5.7).

Durch den Display-Zoom-Faktor wird alles, was auf dem Bildschirm dargestellt ist, vergrößert, ohne den Inhalt des Bildwiederholspeichers zu verändern. Im Gegensatz dazu wirkt sich der Schreib-Zoom-Faktor nur auf die grafischen Zeichen aus, während sie in den Bildwiederholpeicher geschrieben werden. Der Display-Zoom-Faktor kann zu jedem Zeitpunkt in jeden Wert geändert werden. Diese Änderung wird der ganze Bildschirm sofort wieder spiegeln. Der Schreib-Zoom-Faktor für grafische Zeichen legt die tatsächliche Größe der Zeichen im Bildwiederholpeicher fest.

Sobald ein Zeichen gezeichnet (in den Bildwiederholpeicher geschrieben) wurde, bleibt es im Bildwiederholpeicher immer in der gleichen Größe, unabhängig von weiteren Änderungen dieses Schreib-Zoom-Faktors.

Die Parameter des FIGS-Kommandos für das Ausfüllen von Flächen sind:

DC = (Anzahl der Pixel in senkrechter Richtung) -1  
D = Anzahl der Pixel in der Ausgangsrichtung  
D2 = D

Die restlichen Parameter können auf ihren Ausgangswerten belassen werden, wenn keine weiteren Parameter angegeben werden, bevor das GCHRD-Kommando gegeben wird.

Geeignete grafische Zeichen und ausgefüllte Flächen können benutzt werden, um einen Kursivschrift-Effekt zu erzielen. Diese Betriebsart wird durch ein Flag im ersten Parameterbyte des FIGS-Kommandos gesteuert.

Dieser Effekt kann am besten an einem Beispiel erläutert werden. Dazu wird zunächst das  $8 * 8$  Pattern für den Buchstaben "E" (s.a. Bild 6.20) in den Parameter-RAM geladen und dann ohne Vergrößerung sowohl in normaler Form als auch schräg in den Bildwiederholpeicher geschrieben.



Bild 6.20: Normales und schräges grafisches Zeichen

Ein 'Stern' repräsentiert ein einzelnes modifiziertes Pixel. Wenn das gleiche PRAM-Pattern mit einem Zoom-Vergrößerungsfaktor von Zwei eingeschrieben wird, erscheint es wie im Bild 6.21.

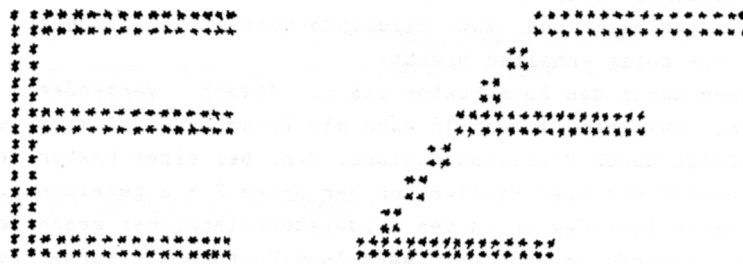


Bild.6.21: Darstellung eines grafischen Zeichens mit Zoom-Faktor Zwei

Wenn die Größe der zu füllenden Fläche kleiner als der  $8 * 8$  PRAM-Puffer ist, wird die passende Fläche des Parameter-RAM in den Bildwiederholtspeicher gezeichnet. Ausgehend vom LSB des PRAM Byte 15 werden soviel Bits in den Bildwiederholtspeicher (entlang der DIR-Richtung) gezeichnet, wie die Parameter D und D2 des FIGS-Kommandos erlauben. Es wird auf soviele Bytes zugegriffen, wie der Parameter DC festlegt. Auch diese Bytes werden nur soweit abgetastet, wie die Parameter D und D2 erlauben.

Die Auswirkung davon ist, daß nur die Ecke des PRAM-Puffer in den Bildwiederholtspeicher gezeichnet wird, die das LSB von Byte 15 enthält. Das folgende Beispiel soll diese Gedanken veranschaulichen:

Inhalt des Parameter-RAM für ein  $5 * 7$  Zeichen

Bit		7	6	5	4	3	2	1	0
<hr/>									
PRAM-Byte	15:	0	0	0	1	1	1	1	1
	14:	0	0	0	1	0	0	0	0
	13:	0	0	0	1	0	0	0	0
	12:	0	0	0	1	1	1	1	1
	11:	0	0	0	1	0	0	0	0
	10:	0	0	0	1	0	0	0	0
	9:	0	0	0	1	1	1	1	1
	8:	0	0	0	0	0	0	0	0

Dabei ist DIR = 6, so daß Byte 15 das obere Ende des Zeichens ist und Bit 0 sich in der rechten oberen Ecke befindet. Die modifizierten Pixel sehen dann wie im Bild 6.22 aus.



Bild 6.22: Darstellung eines  $5 * 7$ -Zeichens

Die 'Sterne' stellen die während des Zeichnens modifizierten Pixel dar. Wenn die auszufüllende Fläche größer als der Parameter-RAM-Puffer ist, wird der Puffer in x- und y-Richtung wiederholt, um die Fläche auszufüllen.



**Beispiel:**

**Pattern im Parameter-RAM**

Bit		7	6	5	4	3	2	1	0
<hr/>									
PRAM-Byte	15:	0	0	0	0	1	1	1	1
	14:	0	0	0	0	0	0	1	1
	13:	0	0	0	0	0	1	0	1
	12:	0	0	0	0	1	0	0	0
	11:	0	0	0	1	0	0	0	0
	10:	0	0	1	0	0	0	0	0
	9:	1	1	0	0	0	0	0	0
	8:	0	1	0	0	0	0	0	0

Dieses Pattern ist asymmetrisch, um die Orientierung der Pattern und die Wiederholung beim Zeichnen in den Bildwiederholtspeicher deutlich zu machen.

Im Bild 6.23 ist die Darstellung auf einer Bildfläche mit einer Breite von 19 und einer Höhe von 21 Pixel abgebildet.

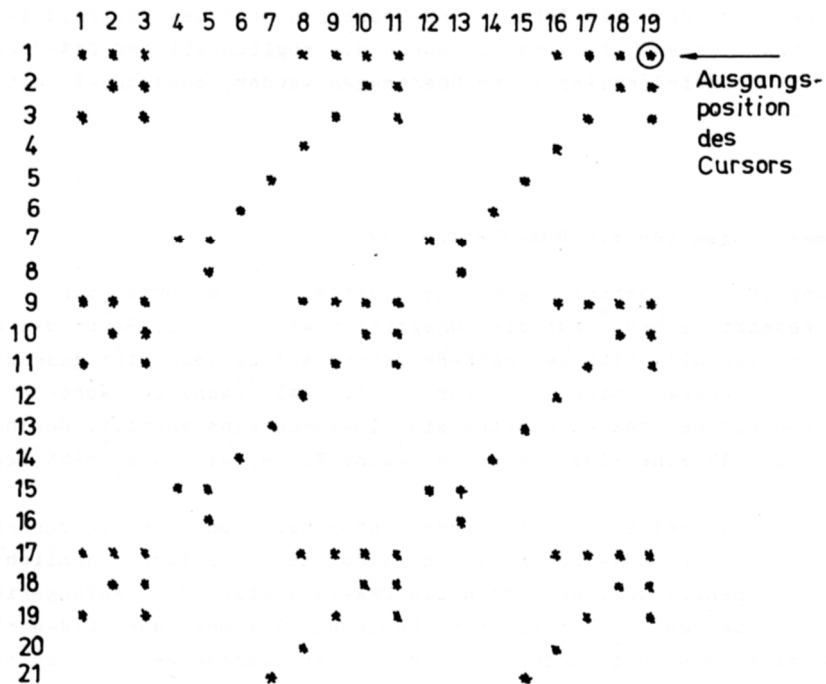


Bild 6.23: Zeichnungs-Pattern auf einer 19 \* 21-Fläche

## 6.19. DMA-Übertragungen

### 6.19.1. Einführung

DMA-Übertragungen in und aus dem Bildwiederholtspeicher werden unter Verwendung fast der selben Kommandofolgen durchgeführt, wie sie bei Zeichnungsoperationen von Darstellungen benutzt werden.

Die Unterschiede bestehen in den Parametern, die mit dem FIGS-Kommando gegeben werden und dem Austausch des Startkommandos für das Zeichnen (FIGD oder GCHRD) durch die DMA-Lese- oder DMA-Schreibkommandos (DMAR oder DMAW).

Bei DMA-Übertragungen wird ein rechteckiges Gebiet übertragen, dessen Höhe, Breite und Orientierung in den Parametern des FIGS-Kommandos angegeben sind. Solche Übertragungen arbeiten ausschließlich auf Wortadreßbasis und ignorieren die Punktsadresse (dAD). Pro RMW-Zyklus wird ein Byte übertragen. Die DMAR- und DMAW-Kommandobytes enthalten ein Feld, das angibt, ob gerade auf das Wort, bzw. nur auf das höherwertige oder niederwertige Byte des Wortes im Bildwiederholtspeicher zugegriffen werden soll (s.a. Bild 5.18 und 5.19).

Für eine DMA-Operation kann die Art der RMW-Operation über das MOD-Feld in den DMAW- und DMAR-Kommandobytes ausgewählt werden. Durch die Möglichkeit der DMA-Übertragung können Daten von und zum Bildwiederholtspeicher übertragen werden, ohne den Inhalt des FIFO im GDC zu verändern.

### 6.19.2. Vorbereitungen für die DMA-Übertragung

Zur Vorbereitung der DMA-Übertragung muß der Cursor auf die Wortadresse des Bildwiederholtspeichers gesetzt werden, auf die zugegriffen werden soll. Bevor die DMA-Übertragung beginnen kann, müssen alle Bits des MASK-Registers auf logisch Eins gesetzt werden.

Der DDA des GDC bewegt sich nur zur horizontal nächsten Wortadresse, wenn das entsprechende Randbit des MASK-Registers eine logische Eins enthält. Befindet sich in den Bitpositionen 0 und 15 eine Eins, spielt es keine Rolle, wie das MASK-Register rotiert wurde.

Das geforderte Voranschreiten der EAD findet unabhängig von der Zeichenrichtung statt. Die FIGS-Parameter für DMA-Übertragungen sind sehr einfach. Ähnlich wie bei den Parametern zur Flächenfüllung, erfordern die DMA-Parameter die Anfangsrichtung DIR und die Länge und Breite des zu übertragenden Gebietes. Bei ungeraden DIR-Werten wird das zu übertragende Gebiet um 45 Grad bezüglich der x- oder y-Achse gedreht. Um nach unten und zur Seite zu gehen, wird in diesem Fall die Adresse des nächsten RMW-Zyklus durch Addition von (Pitch + 1) oder (Pitch - 1) berechnet. Will man nach oben und seitwärts gehen, muß einer dieser Werte subtrahiert werden.

Zum Beispiel würde der Weg quer über den Bildschirm bei einer DMA-Übertragung in Richtung DIR = 3 folgendermaßen aussehen:

DIR = 3

drittes Wort: [-----]  
zweites Wort: [-----]  
erstes Wort: [-----]

Dabei ist [---] ein 16 Bit breites Wort im Bildwiederholtspeicher, wie es auf dem Bildschirm erscheinen würde.

Die Parameter DC, D und D2 (nur beim DMA-Wortlesen) legen die Ausdehnung des Rechtecks (in Worten) fest, auf die zugegriffen wird. Der DC-Wert wählt die Anzahl der Wortadressen aus, auf die senkrecht zur festgelegten Anfangsrichtung zugegriffen wird. Sein Wert ist um eins kleiner als die Anzahl der Wortadressen und er bleibt gleich, ob nun auf beide Bytes oder nur das obere oder nur das untere Byte zugegriffen wird.

Der Parameter D steuert die Anzahl der Bytes, die in die festgelegte Anfangsrichtung DIR übertragen werden. Wenn es gefordert wird, die oberen Bytes von vier Worten zu übertragen, sollte D auf 4 - 1 oder 3 gesetzt werden.

Wird es gewünscht, jeweils beide Bytes der gleichen vier Worte zu übertragen, sollte D auf  $(4 * 2) - 1$  oder 7 gesetzt werden, da bei vier Worten acht Bytes übertragen werden müssen. Wenn beide Bytes eines Wortes übertragen werden, ist D natürlich immer eine ungerade Zahl ( $2n - 1$ ), während D sowohl gerade als auch ungerade sein kann, wenn nur ein Byte jedes Wortes übertragen werden soll.

Die FIGS-Parameter lassen sich folgendermaßen zusammenfassen:

DC = (Anzahl der Wortadressen in der senkrecht zur Anfangsrichtung DIR  
liegenden Richtung) - 1  
D = (Anzahl der Bytes, die in die Anfangsrichtung DIR übertragen  
werden sollen) - 1

Bei Wortübertragungen wird unabhängig von der Zeichenrichtung zunächst das niederwertige Byte und dann das höherwertige Byte übertragen, bevor der GDC zur nächsten Wortadresse weitergeht.

Die DMA-Leseoperation eines Wortes wird anders durchgeführt. Zusätzlich zu den Parametern DC und D wird der Parameter D2 benötigt. Der Parameter D wird geringfügig anders berechnet. Er ist um zwei kleiner (und nicht um eins!) als die Anzahl der Bytes in Anfangsrichtung. Der Parameter D2 ist einfach die Hälfte des Wertes von D.

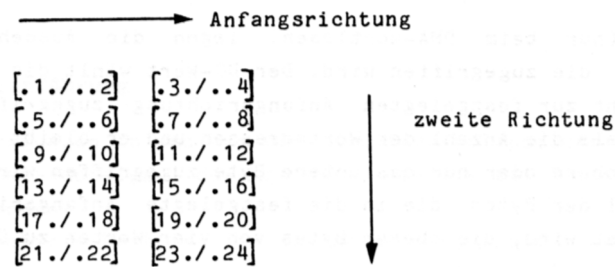
DC = (Anzahl der Wortadressen in der senkrecht zur Anfangsrichtung DIR  
liegenden Richtung) - 1  
D = (Anzahl der Bytes die in die Anfangsrichtung DIR übertragen  
werden sollen) - 2  
D2 = D/2

### 6.19.3. Beispiele für DMA-Übertragungen

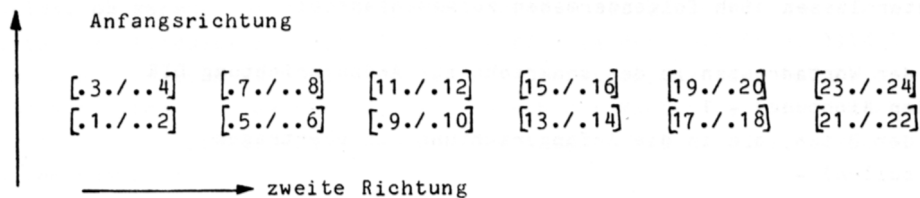
Die folgenden Beispiele einer DMA-Übertragung zeigen die Worte im Bildwiederholtspeicher in eckigen Klammern, wobei ein Schrägstrich das niederwertige Byte auf der linken Seite vom höherwertigen Byte auf der rechten Seite trennt.

Die in eckigen Klammern stehenden Zahlen kennzeichnen die Bytes in der Reihenfolge, in der sie übertragen wurden.

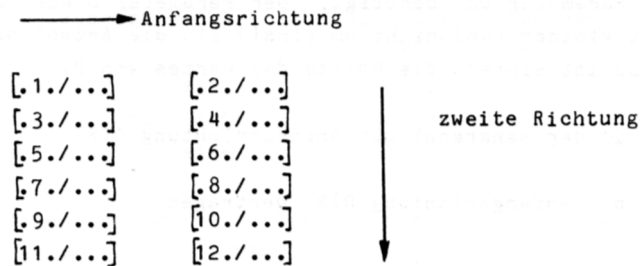
Wortschreiben in den Bildwiederholtspeicher mit DIR = 2, DC = 5 und D = 3:



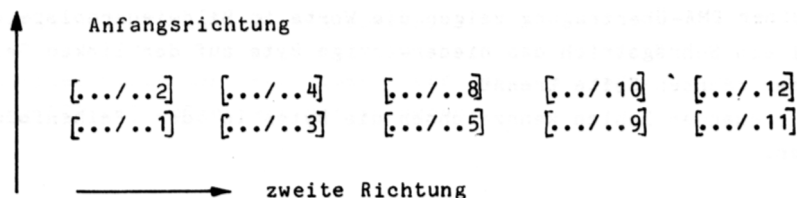
Wortschreiben in den Bildwiederholtspeicher mit DIR = 4, DC = 5 und D = 3:



Übertragung der niederwertigen Bytes mit DIR = 2, DC = 5 und D = 1:

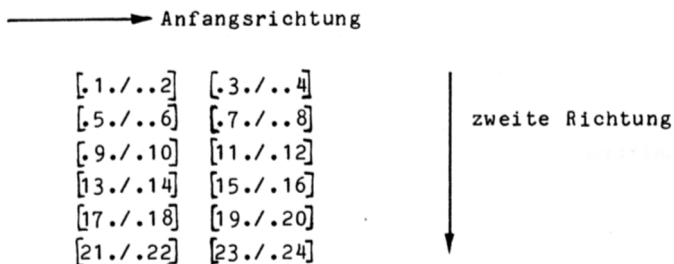


Übertragung der höherwertigen Bytes mit DIR = 4, DC = 5 und D = 1 :

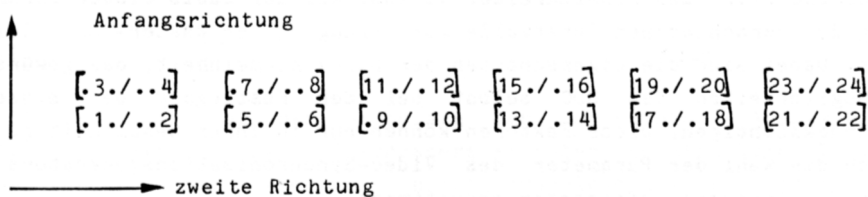


Wenn der DC-Parameter den Wert Null hat, ist die DMA-Übertragung eine eindimensionale Folge von Speicherzugriffen. In diesem Fall steuert D die Anzahl der zu übertragenden Bytes bis zu  $2^{14}$  bzw. 16384 Byte.

Wortlesen aus dem Speicher mit DIR = 2, DC = 5, D = 2 und D2 = 1:



Wortlesen aus dem Speicher mit DIR = 4, DC = 5, D = 2 und D2 = 1:



## **7. Video-Interface**

### **7.1. Berechnung des Video-Zeitverhaltens**

#### **7.1.1. Einführende Bemerkungen**

Alle Anwendungen des GDC U82720 setzen die Berechnung einer Anzahl von Parametern für den Video-Synchronisationsgenerator des Schaltkreises voraus. Auf der Basis dieser Parameter erfolgt die Steuerung der verschiedenen Intervalle zur Erzeugung der Raster- und Zeilenstruktur des Bildes. Dabei sind die Eigenschaften der Bildschirmeinheit, das gewünschte Bildformat und die Möglichkeiten des GDC selbst bei der Festlegung der einzelnen Intervalllängen zu berücksichtigen. Diese Faktoren können nur in ihrer Gesamtheit gesehen werden. Sie sind durch die Wahl der Parameter des Video-Synchronisationsgenerators und mittels Variation der Taktfrequenz aufeinander abzustimmen.

Das Bildformat wird vom Video-Synchronisationsgenerator ausgeählt. Ein Displayzyklus bildet dabei die grundlegende Zeiteinheit. Jeder Displayzyklus benötigt zwei Zyklen des Taktes  $2 \times \text{WCLK}$ . Bei der Anwendung der Bildvergrößerung werden die Zugriffszyklen auf den Bildwiederholpeicher um ein Vielfaches dieses Intervalls verlängert.

Jede Zeile im Bildraster besteht aus vier Teilen (Intervallen), die ihrerseits ganzzahlige Vielfache von Displayzyklen darstellen. Eine vollständige Zeile hat folgenden Aufbau:

- HFP : horizontale vordere Schwarzschiiter
- HS : Horizontal-Synchronisationsimpuls
- HBP : horizontale hintere Schwarzschiiter
- AW : aktive Displayperiode (aktive Worte)

Während der ersten drei Intervalle ist der Bildschirm dunkelgetastet, im vierten Intervall wird der Inhalt des Bildwiederholpeichers in Form gültiger Daten zum Bildschirm gesendet und im aktiven Teil der Zeile abgebildet.

Ein Bildfeld besteht aus vier Gruppen von Zeilen, die mit den horizontalen Intervallen korrespondieren:

- VFP : vordere Vertikal-Schwarzschiiter
- VS : Vertikal-Synchronisationssignal
- VBP : hintere Vertikal-Schwarzschiiter
- AL : aktive Zeilen

Bis auf das AW-Intervall innerhalb der aktiven Zeilen werden alle Bildsignale dunkelgetastet. Ein Vertikalintervall enthält stets ein ganzzahliges Vielfaches von Zeilen. Beim Zeilensprungverfahren besteht ein vollständiges Bild aus zwei aufeinanderfolgenden Bildfeldern, wobei jedes nur die Hälfte der insgesamt darzustellenden Bildpunkte auf dem Schirm abbildet. Ein Bildfeld enthält in dieser Betriebsart die geraden, das nächste die ungeraden Zeilen. Auf diese Weise wird sowohl die Punktfolgefrequenz als auch die Videodaten-Rate auf den halben Wert im Vergleich zur Darstellung ohne Zeilensprung reduziert. Der GDC U82720 ermöglicht die Programmierung von Videoformaten im Zeilensprungverfahren. Die Besonderheiten dieser Betriebsart werden später beschrieben.

### 7.1.2. Berechnung der Parameter

Durch eine entsprechende Festlegung der Parameter für den Synchronisationsgenerator kann in Verbindung mit der Wahl der Taktfrequenz nahezu jedes Video-Synchronisationsformat erzeugt werden.

Im folgenden werden die Abläufe zur korrekten Berechnung der Parameter und der Taktfrequenz auf der Basis gegebener Ansteuerbedingungen für die Bildschirmeinheit sowie des gewählten Video-Formats dargestellt.

**Zur Beachtung :** Für die Darstellung codierter Zeichen und bei Anwendung eines Zeichengenerators, ersetzt ein einzelnes Zeichen ein Wort aus 16 Pixeln, wie sie in der bitweisen Grafikdarstellung benutzt werden.

#### SCHRITT 1

Das Bildformat wird auf der Grundlage der vorgesehenen Applikation festgelegt. Die 'aktiven Intervalle' des Videorasters erzeugen den sichtbaren Teil des Bildfeldes. Als aktives Intervall wird immer ein nicht dunkelgetasteter Bereich verstanden, während dessen eine gültige Bildinformation aus dem Bildwiederholtspeicher gelesen und auf dem Schirm dargestellt wird. Die vollständige Videosequenz enthält auch dunkelgetastete Intervalle für den Strahlrücklauf nach jeder Zeile und nach Erreichen des unteren Bildfeldrandes. Ein Ziel der Berechnungen besteht in der Festlegung der Länge dieser dunkelgetasteten Bereiche. Bei der Bestimmung dieser Größen muß das Bildformat den Gegebenheiten des GDC und der Bildschirmeinheit angeglichen werden.

- Die Anzahl der pro Zeile aktiv darzustellenden Pixel muß ein Vielfaches von 32 betragen. Der GDC ermöglicht in der Grafik-Betriebsart die Darstellung von maximal 4096 Pixeln oder in der Zeichen-Betriebsart die Abbildung von 256 Zeichen.
- Die Anzahl der aktiven Videozeilen kann einen ganzzahligen Wert bis zu 1024 Zeilen pro Bildfeld annehmen. Im Zeilensprungverfahren können insgesamt 2048 Zeilen mit 1024 Zeilen pro Halbbild generiert werden.
- Mit der Festlegung der Zeitdauer eines Bildfeldes wird die Bildwiederholfrequenz bestimmt. Im allgemeinen werden 50 Hz verwendet. Der GDC kann aber auch andere Werte realisieren.

Die Auswahl dieser Größen setzt die Berücksichtigung zweier Aspekte voraus: Die Bildwiederholfrequenz muß hoch genug sein, um ein Flimmern des Bildes auszuschließen, die obere Grenzfrequenz der einzelnen Systemkomponenten darf aber nicht überschritten werden. Der GDC kann im Zeilensprungverfahren arbeiten, wobei in einem einzelnen Bildfeld die Hälfte der Zeilen des vollständigen Bildes dargestellt werden. Die Parameter des Video-Synchronisationsgenerators steuern die Erzeugung der einzelnen Bildfelder. Ein vollständiges Bild besteht aus zwei Bildfeldern plus einer Zeile, die vom GDC automatisch addiert wird. Die Anzahl der abgebildeten Zeilen pro Bildfeld entspricht immer der Hälfte der zum vollständigen Bild gehörenden Zeilenzahl:

- Bildfeld (Videofeld): Intervall zwischen zwei aufeinanderfolgenden Vertikal-Synchronisationsimpulsen.
- Vollständiges Bild : Nur im Zeilensprungverfahren, Intervall zu Darstellung aller zum Bild gehörenden Daten, d.h. der geraden und ungeraden Zeilen.

- DL : Anzahl der im vollständigen Bild darzustellenden aktiven Zeilen
- AL : Anzahl der aktiven Zeilen pro Bildfeld

Berechnung für Zeilensprungverfahren:

$$AL = DL / 2 \quad (7.1)$$

Berechnung ohne Zeilensprung:

$$AL = DL \quad (7.1a)$$

## SCHRITT 2

In diesem Schritt werden die Ansteuerbedingungen für die Bildschirmeinheit festgelegt. Das Display muß die technischen Voraussetzungen zur Abbildung des gewünschten Formats aufweisen. Während der Berechnungen sind die Ergebnisse den Möglichkeiten der Geräte gegenüberzustellen, um mögliche Inkompatibilitäten zu finden.

Von den technischen Parametern der Bildschirmeinheit bestimmte Größen:

- tVB : Zeitdauer der Vertikal-Dunkeltastung,  $tVB_{\min} \dots tVB_{\max}$
- tHB : Zeitdauer der Horizontal-Dunkeltastung,  $tHB_{\min} \dots tHB_{\max}$
- fHOR: horizontale Zeilenablenkfrequenz,  $fHOR_{\min} \dots fHOR_{\max}$

## SCHRITT 3

Das Videofeld besteht aus einer ganzzahligen Menge von Zeilen. Die Zeilen werden zunächst in aktive Bildzeilen und dunkelgetastete Vertikalrücklaufzeilen eingeteilt. Später erfolgt eine weitere Untergliederung der dunkelgetasteten Zeilen in o.g. VFP-, VS-, und VBP-Intervalle. Diese müssen den Gegebenheiten der Bildschirmeinheit angeglichen sein. Die Gesamtzahl der Zeilen eines Videofeldes wird zur Berechnung der Zeilenablenkfrequenz benötigt.

Zur Bestimmung der notwendigen Zeilenanzahl für den dunkelgetasteten Vertikalbereich ist die maximale Zeitdauer des aktiven Intervalls festzulegen. Die Division dieser Intervallzeit durch die bekannte Anzahl der aktiven Zeilen ergibt die maximale Zeitdauer einer Zeile. Dieser Wert wird später präzisiert. Dabei wird in Übereinstimmung mit den technischen Daten des Displays eine möglichst niedrige Zeilenablenkfrequenz angestrebt. Alle Zeilen müssen die gleiche Länge haben. Die Division des minimal dunkelzutastenden Vertikalintervalls durch die maximale Zeilenzeit ergibt die notwendige Anzahl der Zeilen für dieses Intervall. Da die Rechnung i.a. keine ganze Zahl liefert, wird auf den nächsten ganzen Wert aufgerundet. Das treibt die Zeilenablenkfrequenz geringfügig über ihren möglichen Minimalwert, damit wird aber eine korrekte Zeit für die Vertikaldunkeltastung gesichert. Die Kenntnis der aktuellen Anzahl der Zeilen im Videofeld gestattet im folgenden die exakte Berechnung der Zeilenzeit.

- AL : Anzahl der aktiven Zeilen im Videofeld
- BL : Anzahl der dunkelgetasteten Zeilen im Videofeld
- TL : Gesamtzahl der Zeilen im Videofeld
- tVA : aktive Displayzeit, vertikal
- tVB : dunkelgetastete Displayzeit, vertikal



- tZEILE : Zeitdauer (Näherungswert) einer vollständigen Zeile
- tFELD : Zeitdauer eines Videofeldes

Folgende Berechnungen werden durchgeführt:

$$tVA = tFELD - tVB \quad (7.2)$$

$$tVA_{\max} = tFELD - tVB_{\min} \quad (7.3)$$

$$tZEILE_{\text{app}} = tVA_{\max} / AL \quad (7.4)$$

(Näherungswert  $tZEILE_{\text{app}}$ )

$$BL = tVB_{\min} / tZEILE_{\text{app}} \quad (7.5)$$

(BL ist auf einen ganzzahligen Wert aufzurunden)

$$TL = AL + BL \quad (7.6)$$

#### SCHRITT 4

Wenn die Gesamtzahl der Zeilen eines Bildfeldes fixiert ist, kann der im SCHRITT 3 bestimmte Näherungswert für die Dauer einer Zeile unter Nutzung der bekannten Zeit für ein Bildfeld exakt berechnet werden. Der resultierende Wert für die horizontale Zeilenablenkfrequenz muß den technischen Parametern des Displays entsprechen.

$$tZEILE = tFELD / TL \quad (7.7)$$

(exakter Wert)

$$fHOR = 1 / tZEILE \quad (7.8)$$

#### SCHRITT 5

Mit der Bestimmung der exakten Zeit  $tZEILE$  und der bereits ermittelten Anzahl der dunkelgetasteten Zeilen während des Vertikalrücklaufs sind die Voraussetzungen zur genauen Berechnung der Intervalllänge der Vertikaldunkeltastung gegeben. Dieser Wert ist ebenfalls mit den Gegebenheiten der Bildschirmseinheit zu vergleichen.

$$tVB = BL * tZEILE \quad (7.9)$$

#### SCHRITT 6

Nach der Festlegung der Vertikal-Parameter können die Variablen innerhalb einer Zeile berechnet werden. Ebenso wie das Vertikal-Format besteht jede Zeile aus einem dunkelgetasteten und einem aktiven Teil. Der Horizontal-Rücklauf des Elektronenstrahls erfolgt während des dunkelgetasteten Intervalls der Zeile. Im aktiven Teil einer Zeile gelangen die Videodaten auf den Bildschirm, sofern die Zeile selbst eine aktive Zeile des Vertikal-Formats ist. Im Bereich des Vertikalrücklaufs werden die Zeilen vollständig dunkelgetastet.

- tHA : Zeitdauer des aktiven Zeilenbereichs
- AW : Anzahl der aktiven Worte (Displayzyklen) pro Zeile
- tDC : Zeitdauer eines Displayzyklus

- BW : Anzahl der dunkelgetasteten Worte pro Zeile
- TW : Gesamtzahl der Worte pro Zeile

Nachfolgend werden die weiteren Gleichungen zur Berechnung angegeben:

$$t_{HA_{max}} = t_{ZEILE} - t_{HB_{min}} \quad (7.10)$$

$$t_{DC_{app}} = t_{HA_{max}} / AW \quad (7.11)$$

(Näherungswert  $t_{DC_{app}}$ )

$$BW = t_{HB_{min}} / t_{DC_{app}} \quad (7.12)$$

(BW ist auf einen ganzzahligen Wert aufzurunden.)

$$TW = AW + BW \quad (7.13)$$

$$t_{DC} = t_{ZEILE} / TW \quad (7.14)$$

(exakter Wert von  $t_{DC}$ )

#### SCHRITT 7

Mit der exakten Länge eines Displayzyklus wird die aktuelle Länge des dunkelgetasteten Anteils einer Zeile bestimmt. Die bisher durchgeführten Berechnungen liefern eine Größe, die i.a. um einen Teil eines Displayzyklus länger ist als der Zielwert. Anstatt der seitens der Geräteeigenschaften möglichen minimalen Zeitdauer des dunkelgetasteten Zeilenbereiches kann natürlich ein neuer Zielwert benutzt werden. In jedem Fall muß die Länge des dunkelgetasteten Intervalls exakt bestimmt und mit den ursprünglichen Vorgaben verglichen werden. Wenn durch das Display sehr enge Grenzen für diese Größen vorgegeben sind, kann die Festlegung auf ganzzahlige Werte von Displayzyklen schwierig werden.

$$t_{HB} = t_{DC} * BW \quad (7.15)$$

#### SCHRITT 8

In der normalen Grafik-Betriebsart des GDC werden pro Displayzyklus 16 Pixel zum Bildschirm gesendet. Deshalb muß die Frequenz des Pixeltaktes den 16fachen Wert der Displayzyklusfrequenz bzw. den 8fachen Wert der Taktfrequenz  $2xWCLK$  aufweisen.

Zwei weitere Betriebsarten können anstatt der normalen Betriebsart mit 16 Pixeln pro Wort verwendet werden. In der Betriebsart mit gedehntem Speicherzugriff (Wide Display Mode) weist die Frequenz des Pixeltaktes den 32fachen Wert der Displayzyklusfrequenz (d.h. die 16fache Frequenz von  $2xWCLK$ ) auf, so daß gleichzeitig auf ein Wortpaar mit gerader und ungerader Adresse und damit auf 32 Pixel zugegriffen wird.

In der Misch-Betriebsart werden im Grafikbereich vier Taktzyklen  $2xWCLK$  für jeden aktiven Displayzyklus verwendet, d.h. innerhalb der im Vergleich zur Grafik-Betriebsart doppelten Anzahl von Taktzyklen  $2xWCLK$  wird auf 16 Pixel zugegriffen. Die Pixelfrequenz entspricht daher anstatt der 8fachen nur der 4fachen Frequenz von  $2xWCLK$ . Der GDC gibt die Speicheradresse für jedes Wort zweimal aus, so daß der AW-Parameter für die gleiche Anzahl von Pixeln den doppelten Wert verglichen mit der Grafik-Betriebsart aufweisen muß. Während des Horizontalrücklaufs besteht, wie in der Grafik-Betriebsart, ein Displayzyklus aus zwei Takten  $2xWCLK$ . Wird in der Misch-Betriebsart gearbeitet, ist der AW-Parameter den vorstehenden Überlegungen anzupassen.

- tPIX : Zeitdauer einer Pixeldarstellung (normale Grafik-Betriebsart)
- fPIX : Frequenz des Pixeltaktes
- 2xWCLK : Taktfrequenz des GDC
- t2xWCLK : Periodendauer der Taktfrequenz

$$tPIX = tDC / 16 \quad (7.16)$$

$$fPIX = 1 / tPIX \quad (7.17)$$

$$t2xWCLK = tDC / 2 \quad (7.18)$$

$$2xWCLK = 1 / t2xWCLK \quad (7.19)$$

Die berechnete Taktfrequenz 2xWCLK muß innerhalb des zugelassenen Bereichs für den GDC liegen, da sonst die korrekte Funktion des Schaltkreises nicht gewährleistet ist. Wenn die Berechnungen eine Taktfrequenz außerhalb der Grenzen für 2xWCLK ergaben, muß ein Display für ein anderes Video-Format oder eine andere Betriebsart gewählt werden.

#### SCHRITT 9

Der innerhalb einer Zeile dunkelgetastete Bereich wird in drei Intervalle gegliedert. Die Längenangabe dafür erfolgt im Parameterfeld in Worten. Die Gesamtzahl der dunkelgetasteten Worte einer Zeile wurde bereits mit BW bezeichnet.

- HFP : vordere horizontale Schwarzschiiter
- HS : Horizontal-Synchronisationssignal
- HBP : hintere horizontale Schwarzschiiter

$$BW = HFP + HS + HBP \quad (7.20)$$

Die Anzahl der Worte pro Intervall hängt von verschiedenen Bedingungen (s.a. Abschn. 5.1.) ab. Die Forderungen für die Intervalllängen stellen Minimalwerte dar, die für den Kommandoprozessor des GDC während des dunkelgetasteten Bereichs einer Zeile genügend Zeit für die Ausführung seiner Operationen sichern. Wenn für die dynamischen RAMs des Bildwiederholerspeichers Refresh freigegeben wurde, muß während des Horizontal-Synchronisationsimpulses und unter Berücksichtigung der Zeilenablenkfrequenz die notwendige Refresh-Rate gesichert sein.

Innerhalb des Horizontal-Synchronisationsimpulses werden die Speicheradressen von einem internen Refresh-Zähler erzeugt. Die Refresh-Zyklen haben die höchste Priorität innerhalb der Speicherzyklen des GDC, um Verzögerungen oder Unterbrechungen zu vermeiden.

Die Länge sowie Position des Horizontal-Synchronisationsimpulses muß den technischen Parametern der Bildschirmeinheit entsprechen. Treten Widersprüche zwischen den einzelnen Forderungen auf, sind sie zu beseitigen. Dazu kann in einigen Fällen externe Hardware eingesetzt werden. Weiterhin besteht die Möglichkeit der Veränderung der Taktfrequenz des GDC während des Rücklaufintervalls.

Die Refresh-Rate für dynamische RAMs im Bildwiederholerspeicher muß so festgelegt werden, daß die Refresh-Bedingungen für die Speicher erfüllt sind. Deshalb muß das Horizontal-Synchronisationssignal lang genug sein, um eine minimale RAM-Refreshrate bei der

benötigten horizontalen Ablenkfrequenz zu sichern. Während der aktiven Phase des o.g. Horizontal-Synchronisationssignals in einem Display-Zyklus kommt, sofern Refresh freigegeben wurde, jede Adresse von einem GDC-internen Refresh-Zähler.

Bei der Anwendung von 64K-DRAMs können z.B. 128 Refresh-Zyklen in 2000 µs erforderlich sein. Wenn die Refresh-Zyklen nur während des Horizontal-Intervalls einer jeden Zeile durchgeführt werden, muß der Zeilenaufbau die Ausführung von 128 Refresh-Zyklen in 2000 µs erlauben. Die Anzahl der Refresh-Zyklen und deren Zeitintervalle sind durch die Typeigenschaften der zur Anwendung kommenden dynamischen RAMs bestimmt.

Nachfolgende Größen sind für die Festlegung des Zeitverhaltens für die Refresh-Zyklen relevant:

- tREFRESH : Zeitdauer eines Refresh-Intervalls
- RCC : Anzahl der notwendigen Refresh-Zyklen
- tZEILE : Zeitdauer einer vollständigen Zeile
- LRI : Anzahl der Zeilen pro Refresh-Intervall
- RCL<sub>min</sub> : minimale Anzahl von Refresh-Zyklen pro Zeile
- HS : Zeitdauer des Horizontal-Synchronisationsimpulses

$$LRI = tREFRESH / tZEILE \quad (7.21)$$

(LRI ist auf einen ganzzahligen Wert abzurunden)

$$RCL_{min} = RCC / LRI \quad (7.22)$$

(RCL<sub>min</sub> ist auf einen ganzzahligen Wert aufzurunden)

$$HS \geq RCL_{min} \quad (7.23)$$

#### SCHRITT 10

Das Vertikal-Rücklaufintervall wird wie das Horizontal-Rücklaufintervall in drei Bereiche geteilt. Die Länge der einzelnen Intervalle, d.h. die Anzahl der Zeilen pro Intervall, ist nicht an spezielle Bedingungen gebunden, jedoch muß jedes Intervall mindestens eine Zeile lang sein.

- VFP : vordere vertikale Schwarzschiene
- VS : Vertikal-Synchronisationsimpuls
- VBP : hintere Vertikal-Schwarzschiene
- BL : Anzahl der insgesamt dunkelgetasteten Zeilen

$$BL = VFP + VS + VBP \quad (7.24)$$

Im Parameterfeld wird die Zeitdauer der einzelnen Komponenten in Anzahl von Zeilen angegeben (s.a. Abschn. 5.1.).

### 7.1.2.1. Beispiel für eine Zeitrelationsberechnung

Im Beispiel werden durch die technischen Daten der Bildschirmseinheit folgende Parameter vorgegeben:

fZEILE	= 22,700 kHz	horizontale Zeilenablenkfrequenz
tZEILE	= 44,053 $\mu$ s	entspricht H (eine vollständige Zeile)
fVERT	= 43...50,7 Hz	Bildwiederholffrequenz
tVERT	= 20,000 ms	bei fVERT = 50,000 Hz
tBW <sub>min</sub>	= 9,0 $\mu$ s	Horizontal-Dunkeltastung *
tBW <sub>typ</sub>	= 12,0 $\mu$ s	
tBL <sub>min</sub>	= 1000 $\mu$ s	Vertikal-Dunkeltastung *
tBL <sub>typ</sub>	= 2112 $\mu$ s	
tHFP <sub>typ</sub>	= 4 $\mu$ s	vordere Schwarzscher *
tHS <sub>typ</sub>	= 2 $\mu$ s	Horizontal-Synchronisationsimpuls *
tHBP <sub>typ</sub>	= 6 $\mu$ s	hintere Schwarzscher *
tVFP <sub>typ</sub>	= 528 $\mu$ s (12 Zeilen)	vordere Vertikal-Schwarzscher *
tVS <sub>typ</sub>	= 528 $\mu$ s (12 Zeilen)	Vertikal-Synchronisationsimpuls *
tVBP <sub>typ</sub>	= 1056 $\mu$ s (24 Zeilen)	hintere Vertikal-Schwarzscher *
tHA <sub>typ</sub>	= 32 $\mu$ s	

\* **Erläuterung** : Das Vorsetzen eines 't' vor das Symbol zeigt an, daß hier die Zeitdauer zunächst nicht in Worten oder Zeilen, wie später im Parameterfeld, sondern in einer Zeiteinheit angegeben wird.

Nachfolgend die weitere Berechnung:

$$TL = fZEILE / fVERT$$

$$TL = 22,700 \text{ kHz} / 50,000 \text{ Hz} = 454 \text{ Zeilen pro Bildfeld}$$

Anzahl der Zeilen innerhalb der Vertikal-Dunkeltastung:

$$BL = VFP + VS + VBP$$

$$BL = 12 + 12 + 24 = 48 \text{ Zeilen}$$

Anzahl der aktiven Zeilen pro Bildfeld:

$$AL = TL - BL$$

$$AL = 454 - 48 = 406 \text{ Zeilen}$$

Das gewünschte Verhältnis der Anzahl der Bildpunkte in horizontaler Richtung zur Anzahl der Bildpunkte in vertikaler Richtung sei 4 : 3. Es ergibt sich:

$$406 \text{ Zeilen} * 4/3 \text{ Pixel je Zeile} = 539,98$$

Der GDC U82720 verarbeitet geradzahlige Mengen von Worten je Zeile, d.h. Vielfache von

16 Pixeln:

$$AW = 539,98 / 16 = 33,75 \text{ Worte}$$

Der GDC kann 32 Worte für 512 Pixel oder 34 Worte für 544 Pixel erzeugen. Bei der Verwendung von 544 Pixeln ergibt sich ein Verhältnis Anzahl horizontaler zu vertikaler Bildpunkte:

$$544 \text{ Pixel} / 405 \text{ Zeilen} = 1,34$$

Zeitdauer eines Displayzyklus (eines Wortes):

$$t_{DC_{app}} = t_{HA} / AW$$

$$t_{DC_{app}} = 32 \mu s / 34 \text{ Worte} = 941,176 \text{ ns (Näherungswert)}$$

Anzahl der Worte innerhalb der Horizontaldunkeltastung:

$$BW = t_{HB} / t_{DC_{app}}$$

$$BW = 12 \mu s / 941,2 \text{ ns} = 12,75 \text{ Worte}$$

Der erhaltene Wert wird auf 13 Worte aufgerundet. Die Gesamtzahl der Worte je Zeile ergibt sich somit:

$$TW = AW + BW$$

$$TW = 34 \text{ Worte} \rightarrow \text{aktiv} + 13 \text{ Worte} \rightarrow \text{dunkelgetastet} = 47 \text{ Worte}$$

Mit diesem Ergebnis sind die exakte Zeitdauer eines Displayzyklus, sowie die daraus abgeleiteten Größen zu berechnen:

$$t_{DC} = t_{ZEILE} / TW$$

$$t_{DC} = 44,053 \mu s / 47 \text{ Worte} = 937,295 \text{ ns}$$

$$t_{2xWCLK} = t_{DC} / 2$$

$$t_{2xWCLK} = 937,295 \text{ ns} / 2 = 468,647 \text{ ns}$$

$$2xWCLK = 1 / t_{2xWCLK}$$

$$2xWCLK = 1 / 468,647 \text{ ns} = 2,1338 \text{ MHz}$$

$$t_{PIX} = t_{DC} / 16$$

$$t_{PIX} = 937,295 \text{ ns} / 16 = 58,581 \text{ ns}$$

$$f_{PIX} = 1 / t_{PIX}$$

$$f_{PIX} = 1 / 58,581 \text{ ns} = 17,0704 \text{ MHz}$$

Intervall der Horizontaldunkeltastung:

$$BW = 13 \text{ Worte}$$

$$BW = HFP + HS + HBP$$

Die typischen Werte für HFP, HS, HBP waren vorgegeben:

$$t_{HFP_{typ}} = 4 \mu s \quad t_{HFP_{typ}} / t_{DC} = 4,26 \quad HFP = 4 \text{ Worte} \\ t_{HFP} = 3,75 \mu s$$

$$t_{HS_{typ}} = 2 \mu s \quad t_{HS_{typ}} / t_{DC} = 2,13 \quad HS = 3 \text{ Worte} \\ t_{HS} = 2,81 \mu s$$

$$t_{HBP_{typ}} = 6 \mu s \quad t_{HBP_{typ}} / t_{DC} = 6,4 \quad HBP = 6 \text{ Worte} \\ t_{HBP} = 5,62 \mu s$$

Die Intervalllänge der Vertikaldunkeltastung ist durch die Vorgabe der dunkelgetasteten Zeilen berechenbar:

$$VFP = 12 \text{ Zeilen} \quad t_{VFP} = 12 * t_{ZEILE} = 528,6 \mu s$$

$$VS = 12 \text{ Zeilen} \quad t_{VS} = 12 * t_{ZEILE} = 528,6 \mu s$$

$$VBP = 24 \text{ Zeilen} \quad t_{VBP} = 24 * t_{ZEILE} = 1057,3 \mu s$$

Zusammenfassung der Video-Parameter für den Vergleich mit den Vorgaben:

Oszillatorfrequenz für die Video-Schieberegister (Pixel-Frequenz)	$f_{PIX}$	= 17,0704 MHz
Anzahl der Worte je Zeile	TW	= 47
Anzahl der Zeilen je Bildfeld	TL	= 454
Zeitdauer eines Displayzyklus	tDC	= 937,295 ns
Zeitdauer einer vollständigen Zeile	tZEILE	= 44,053 $\mu s$
	fZEILE	= 22,700 kHz
Zeitdauer eines Bildfeldes	tFELD	= 20,000 ms
	fFELD	= 50,000 Hz
Zeitdauer der Horizontaldunkeltastung	tBW	= 12,18 $\mu s$
Zeitdauer der Vertikaldunkeltastung	tBL	= 2114,5 $\mu s$
Zeitdauer der vorderen Schwarzschiiter	tHFP	= 3,75 $\mu s$

Zeitdauer des Horizontal-Synchronisationsimpulses tHS	= 2,81 $\mu$ s
Zeitdauer der hinteren Schwarzscherter tHBP	= 5,62 $\mu$ s

#### 7.1.2.2. Besonderheiten beim Zeilensprungverfahren

Beim Zeilensprungverfahren werden zwei Bildfelder (Halbbilder) erzeugt und nacheinander auf dem Schirm abgebildet. Die Gesamtzahl der Zeilen beider Felder ist ungerade. Der Vertikal-Synchronisationsimpuls zwischen beiden Feldern wird genau um die Hälfte der Zeitdauer einer Zeile verschoben, wodurch die Zeilen des zweiten Feldes zwischen den Zeilen des ersten abgebildet werden. Dadurch kann ein vollständiges Bild ohne Flimmererscheinungen bei niedriger Bandbreite des Videosignals generiert werden. Das Zeilensprungverfahren benutzt man, wenn bei Bilddarstellungen über Videokamera (bzw. auch bei rechnergestützter Bildverarbeitung) Einschränkungen der verfügbaren Bandbreite zu beachten sind. Die Bandbreite wird dann sowohl in horizontaler Richtung bei der Zeilendarstellung als auch in vertikaler Richtung bei der Darstellungsfolge der Zeilen reduziert.

Das Zeilensprungverfahren eignet sich, falls keine Einschränkung der Videobandbreite zu beachten ist, grundsätzlich weniger für grafische Darstellungen in der Grafik-Betriebsart des GDC U82720 und auch nicht bei jedem alphanumerischen Display zur Textverarbeitung. Diese Einschränkung resultiert aus folgendem: Wenn eine Zeile im Bildraster dunkel ist und die nächste in ihrer gesamten Länge hell, so sind bei der Darstellung mit Zeilensprung in den aufeinanderfolgenden Bildfeldern vom menschlichen Auge Flimmer- bzw. Vibrationserscheinungen wahrnehmbar. Diese Erscheinung kann mit speziellen Leuchtstoffen der Bildröhre reduziert werden.

Hinsichtlich der Programmierung des Bildaufbaus mit dem Zeilensprungverfahren bestehen beim GDC U82720 alle programmtechnischen Voraussetzungen. Die Signale zur Horizontal- und Vertikalsynchronisation können von jeder Videoeinheit, die separate Synchronisationssignale anerkennt, verarbeitet werden. Die Zeilenzahl beider Halbbilder, d.h. die eines vollständigen Bildes, ist um eins größer als die doppelte Zeilenzahl eines Halbbildes, für das der GDC programmiert wurde. Der Video-Synchronisationsgenerator des GDC addiert selbständig eine zusätzliche Zeile und generiert jedes zweite Vertikal-Synchronisationssignal (VSYNC) um etwa eine halbe Zeile zeitverzögert.

Bei typischen Anwendungen des Zeilensprungverfahrens liegt das zweite Feld mit weniger als 15% Abweichung von der idealen Position zwischen den Zeilen des ersten Feldes.

Übergänge am VSYNC-Ausgang finden beim Zeilensprungverfahren unter Berücksichtigung der HSYNC-Impulse an zwei verschiedenen Punkten statt. Die Übergänge für das erste Feld (sowohl für steigende, als auch für fallende Flanken) treten simultan zur steigenden Flanke von BLANK auf. Im zweiten Feld finden diese Übergänge drei 2xWCLK-Zyklen vor der Mitte der aktiven Displayperiode AW statt. Damit ist unter Beachtung der Zeit für die Dunkelastung beim Rücklauf keine zwangsläufige Positionierung der VSYNC-Übergänge genau auf die Zeilenmitte verbunden. Das Bild 7.1 verdeutlicht diesen Zusammenhang:

Die Intervalle A und B werden mit nachfolgenden Gleichungen berechnet:

$$\text{Intervall } A^* = 2 * (HFP + HS + HBP + AW/2) - 3 \quad (7.25)$$

$$\text{Intervall } B^* = 2 * (AW/2) + 3 \quad (7.26)$$

A bzw. B  $\xrightarrow{*}$  Angabe in 2xWCLK-Zyklen



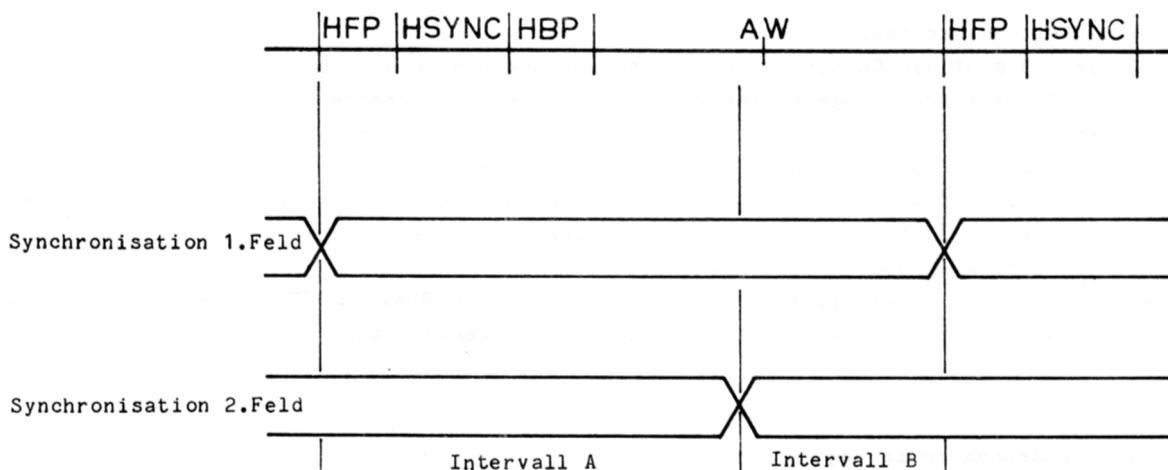


Bild 7.1: Horizontal-Zeitverhalten bezogen auf VSYNC-Intervalle

Die Anzahl der während des Horizontalrücklaufs dunkelgetasteten Worte berechnet sich nach Gleichung 7.20 :

$$BW = HFP + HS + HBP$$

Für die Gesamtzahl der Takte TC von  $2 \times WCLK$  in einer Zeile erhält man :

$$TC = 2 * (HFP + HS + HBP + AW) \quad (7.27)$$

Der Fehler LPE der Positionierung der Zeilenpaare im Zeilensprungverfahren, bezogen auf eine halbe Zeile Verschiebung, errechnet sich wie folgt :

$$LPE = (Intervall A - Intervall B) / (TC / 2) * 100\% \quad (7.28)$$

$$LPE = ((2 * BW + AW - 3) - (AW + 3)) / (TC / 2) * 100\% \quad (7.29)$$

$$LPE = (4 * BW - 12) / TC * 100\% \quad (7.30)$$

## 7.2. Cursorsdarstellung

Die Anzeige für die Cursorsdarstellung kann in der Zeichen-Betriebsart sowie für die Bildfläche zur Zeichendarstellung in der Misch-Betriebsart am Ausgang A17 des GDC erfolgen. Dieser Ausgang liegt innerhalb der Displayintervalle des Wortes, welches durch die EAD-Adresse des Cursors angezeigt wird, auf H-Potential (s.a. Abschn.5.1.). Das Ausgangssignal kann von einer externen Hardware zur Darstellung des Cursors als Negativbild, Blinken usw. genutzt werden.

### 7.2.1. Gedehte Darstellungsart

Bei der Anwendung der gedehnten Darstellungsart (Wide Display Mode) werden die Operationen mit dem Cursor aufwendiger. In dieser Betriebsart erhöht der GDC die Adressen in jedem Display-Zyklus um zwei. Die Cursoranzeige muß diesen Umstand bei der Darstellung

kodierter Zeichen berücksichtigen.

Der GDC U82720 gibt die Cursoranzeige in der entsprechenden Hälfte des Displayzyklus aus, in der auf das Zeichen zugegriffen wird. Falls die Startadresse SAD der darzustellenden Fläche gerade ist, erzeugt der GDC nur gerade Adressen, da er in jedem Displayzyklus seinen Adreßzähler jeweils um zwei erhöht. In der ersten Hälfte des Displayzyklus wird der Cursorstatus für das erste zu serialisierende Zeichen ausgegeben, welches in diesem Fall ein Zeichen mit gerader Adresse wäre. Durch externe Hardware wird der Zugriff auf die nächsthöhere ungerade Adresse gewährleistet.

In der zweiten Hälfte des Displayzyklus erfolgt über den Ausgang A17 die Anzeige des Cursorstatus für das Zeichen auf dieser nächsthöheren ungeraden Adresse.

### **7.2.2. Cursorprogrammierung**

Für den Fall, daß der Cursor nicht als Block (d.h. der Cursor ist nicht über die gesamte Reihenhöhe aktiv) dargestellt wird, ist für eine korrekte Darstellung die Programmierung auf der untersten Zeile zu vermeiden. Wenn sich der Cursor in der oberen Zeichenreihe befindet, wird unter Umständen für den CTOP-Parameter (Festlegung mit dem CCHAR-Kommando) der Wert Null anstatt des programmierten Wertes vorausgesetzt. Dabei würde dann der Cursor von der oberen Zeile einer Zeichenreihe bis zur CBOT-Zeile (Festlegung ebenfalls mit CCHAR-Kommando) dargestellt. In den anderen Zeichenreihen wird der CTOP-Parameter richtig verwendet.

Bei Verwendung des Zeilensprungverfahrens müssen, unabhängig von der Betriebsart, alle Parameter des Cursors während der Initialisierungsphase des GDC programmiert werden. Dies trifft gleichermaßen für die Grafik-, Misch- und Zeichen-Betriebsart zu. Bei der Anwendung des GDC ohne Zeilensprung braucht in der Grafik-Betriebsart nur das erste Parameterbyte des Cursors programmiert zu werden.

Tafel A-1: STATISCHE KENNWERTE

Betriebstemperaturbereich: 0...70°C,  $U_{CC} = 4,75...5,25V$ 

Symbol	Parameter	Wert min.	max.	Ein- heit	Bedingungen
$U_{CC}$	Betriebsspannung	4,75	5,25	V	
$U_{IL}$	Eingangsspannung Low	-0,5	0,8	V	
$U_{IH}$	Eingangsspannung High	2,0	$U_{CC}$	V	
$U_{ICL}$	Takteingangsspannung Low	-0,5	0,6	V	
$U_{ICH}$	Takteingangsspannung High	3,5	$U_{CC}$	V	
$U_{OL}$	Ausgangsspannung Low		0,45	V	$I_{OL} = 2,2mA$
$U_{OH}$	Ausgangsspannung High		2,4	V	$I_{OH} = -400\mu A$
$I_{OZ}$	Ausgangsreststrom		$\pm 10$	$\mu A$	$U_{SS} + 0,45V \leq U_I \leq U_{CC}$
$I_{IL}$	Eingangsreststrom		$\pm 10$	$\mu A$	$U_{SS} \leq U_I \leq U_{CC}$
$I_{CC}$	Betriebsstrom		270	mA	

Tafel A-2: DYNAMISCHE KENNWERTE

Betriebstemperaturbereich: 0...70 °C,  $U_{CC} = 4,75...5,25$  V

A-2.1: Datenbus-Zeitverhalten Lesezyklus

Nr.	Symbol		Parameter	min.	max.
	neu	alt		ns	ns
1	TsAO(RDf)	T <sub>AR</sub>	Setzzeit A0 vor $\overline{RD}$ (HL)	0	
2	ThAO(RDr)	T <sub>RA</sub>	Haltezeit A0 nach $\overline{RD}$ (LH)	0	
3	TwRD1	T <sub>RR</sub>	Impulsbreite $\overline{RD}$	<4>+20	
4	TdRDf(DO)	T <sub>RD</sub>	Verzögerungszeit $\overline{RD}$ (HL) bis Ausgabe gültiger Daten		120
5	TdRDr(DOz)	T <sub>DF</sub>	Verzögerungszeit $\overline{RD}$ (LH) bis Floaten des Datenbus	0	120
6	TrRD	T <sub>RV</sub>	Erholzeit $\overline{RD}$	4<24>	

A-2.2: Datenbus-Zeitverhalten Schreibzyklus

Nr.	Symbol		Parameter	U82720D	
	neu	alt		min. ns	max. ns
7	TsAO(WR)	T <sub>AW</sub>	Setzzeit A0 vor $\overline{WR}$ (HL)	0	
8	ThAO(WR)	T <sub>WA</sub>	Haltezeit A0 nach $\overline{WR}$ (LH)	0	
9	TwWR	T <sub>WW</sub>	Impulsbreite $\overline{WR}$	120	
10	TsDI(WRr)	T <sub>DW</sub>	Setzzeit Daten vor $\overline{WR}$ (LH)	100	
11	ThDI(WRr)	T <sub>WD</sub>	Haltezeit Daten nach $\overline{WR}$ (LH)	0	
6	TrWR	T <sub>WV</sub>	Erholzeit $\overline{WR}$	4<24>	

### A-2.3: Bildwiederholpeicher-Zeitverhalten

Nr.	Symbol		Parameter	U82720D	
	neu	alt		min. ns	max. ns
12	TdCr(ADO)	T <sub>CA</sub>	Verzögerungszeit 2xWCLK(LH) bis Ausgabe gültiger Adressen od. Daten	30	160
13	ThADO(Cr)	T <sub>AC</sub>	Haltezeit Adressen od. Daten nach 2xWCLK(LH)	30	160
14	TsDI(Cf)	T <sub>DC</sub>	Setzzeit Daten vor 2xWCLK(HL)	0	
15	ThDI(Cf)	T <sub>CD</sub>	Haltezeit Daten nach 2xWCLK(HL)	<16>+20	
16	TdCf(DBIN)	T <sub>IE</sub>	Verzögerungszeit 2xWCLK(HL) zu <u>DBIN</u>	30	120
17	TdCr(ALEr)	T <sub>CAH</sub>	Verzögerungszeit 2xWCLK(LH) bis ALE(LH)	30	125
18	TdCf(ALEf)	T <sub>CAL</sub>	Verzögerungszeit 2xWCLK(HL) bis ALE(HL)	30	100
19	TwALEl	T <sub>AL</sub>	Low-Breite ALE	<24>+30	
20	TwALEh	T <sub>AH</sub>	High-Breite ALE	<25>-20	
21	TdCr(V)	T <sub>CO</sub>	Verzögerungszeit 2xWCLK(LH) bis Videosignale		150

#### A-2.4: Lichtstiftsynchronisation

Nr.	Symbol neu	alt	Parameter	U82720D	
				min. ns	max. ns
22	TsLPVS(Cr)	T <sub>PC</sub>	Setzzeit LPEN oder VSYNC vor 2xWCLK(LH)	30	
23	TwLPVS	T <sub>PP</sub>	Impulsbreite LPEN oder VSYNC	<24>	

#### A-2.5: Taktsignal-Zeitverhalten

Nr.	Symbol neu	alt	Parameter	U82720D	
				min. ns	max. ns
24	TcC	T <sub>CY</sub>	Taktperiode	250	2000
25	TwCh	T <sub>CH</sub>	High-Breite Takt	105	
26	TwCl	T <sub>CL</sub>	Low-Breite Takt	105	
27	TrC	T <sub>R</sub>	Taktanstiegszeit		20
28	TfC	T <sub>F</sub>	Taktabfallzeit		20

# A-2.6: DMA-Zeitverhalten

Nr.	Symbol		Parameter	U82720D	
	neu	alt		min. ns	max. ns
29	TdCr(DREQ)	T <sub>KQ</sub>	Verzögerungszeit 2xWCLK(LH) bis DREQ		150
30	TsDREQ(DACK)	T <sub>RQAK</sub>	Setzzeit DREQ vor $\overline{\text{DACK}}(\text{HL})$	0	
31	TdDACKf(DREQf)	T <sub>AKRQ</sub>	Verzögerungszeit $\overline{\text{DACK}}(\text{HL})$ bis DREQ(HL)		<24> +150
32	TsDACK(RDf)	T <sub>ACC</sub>	Setzzeit $\overline{\text{DACK}}$ bis $\overline{\text{RD}}(\text{HL})$ oder $\overline{\text{WR}}(\text{HL})$	0	
33	TwRDDMA1	T <sub>RR1</sub>	Impulsbreite $\overline{\text{RD}}$ (DMA-Betrieb)	<34>+20	
34	TdRDf(DBO)	T <sub>RD1</sub>	Verzögerungszeit $\overline{\text{RD}}(\text{HL})$ bis Datenausgang		1,5<24> +120
35	ThDACK(RDr)	T <sub>CAC</sub>	Haltezeit $\overline{\text{DACK}}$ nach $\overline{\text{RD}}(\text{LH})$ oder $\overline{\text{WR}}(\text{LH})$	0	
36	TcDACKW	T <sub>AK1</sub>	Zykluszeit $\overline{\text{DACK}}$ in der Wort-Betriebsart	4<24>	
37	TcDACKB	T <sub>AK2</sub>	Zykluszeit $\overline{\text{DACK}}$ in der Byte-Betriebsart	5<24>	
38	TwDACKh	T <sub>AKH</sub>	High-Breite $\overline{\text{DACK}}$	<24>	

Erläuterung der Symbole:

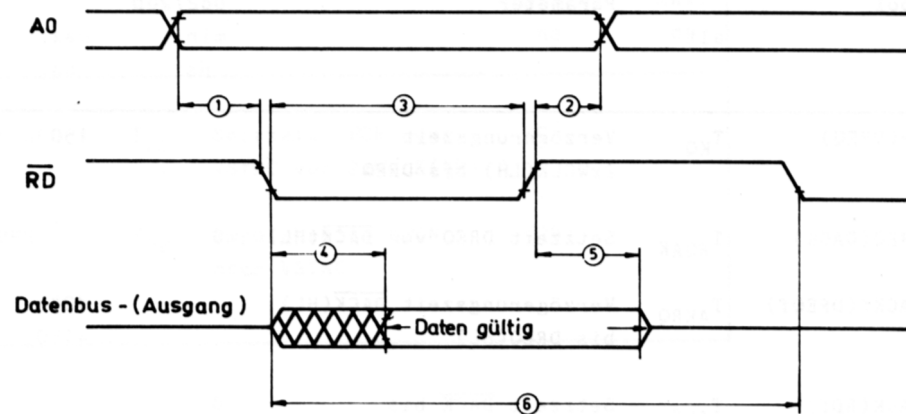
- <x> - Wert entspricht dem unter lfd. Nummer (Nr.) x angegebenen Wert
- n<x> - Wert entspricht dem n-fachen des unter lfd. Nummer (Nr.) x angegebenen Wertes

# A-3: ZEITDIAGRAMME

A-3.1:

Datenbus - Zeitverhalten

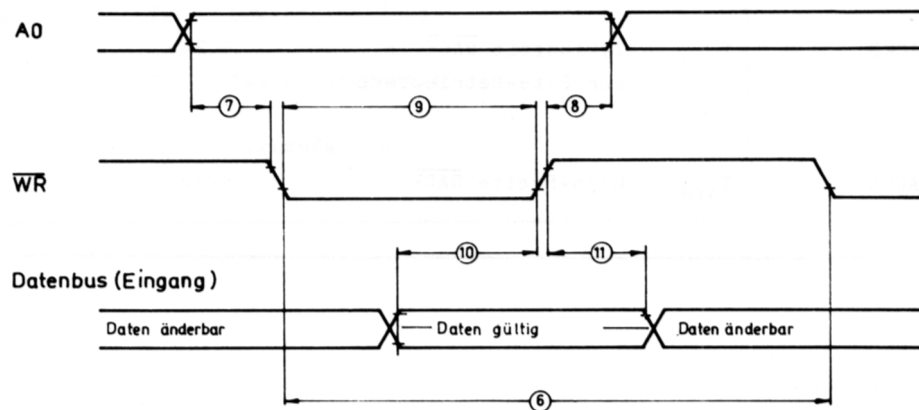
Lesezyklus



A-3.2:

Datenbus - Zeitverhalten

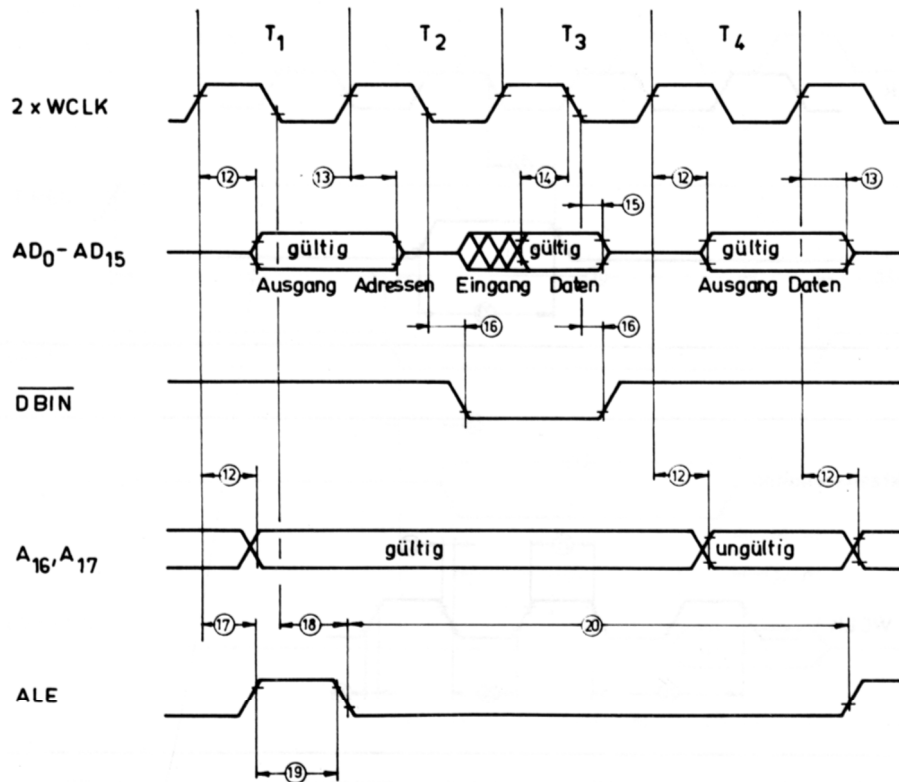
Schreibzyklus



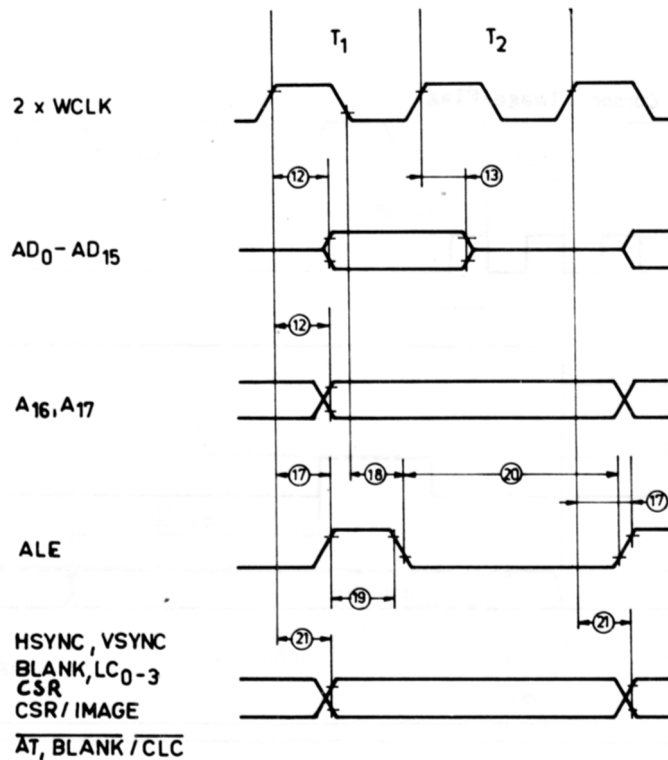


### A-3.3: Zeitverhalten Bildwiederholungspeicher

Lese / Verändern / Schreibzyklus (RMW - Zyklus)



### A-3.4: Lese-Zyklus



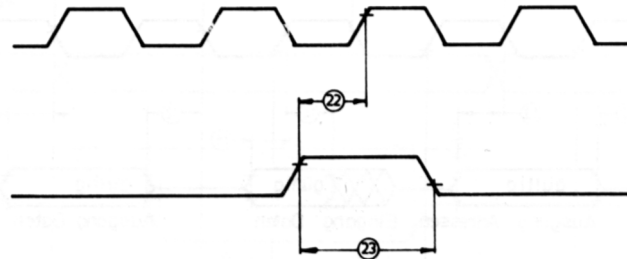
A-3.5:

Lichtstiftsynchronisation

2 x WCLK

L PEN

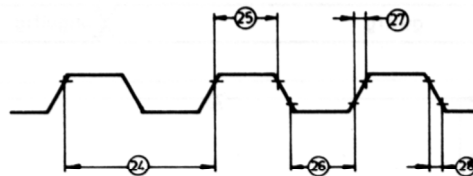
VS YNC



A-3.6:

Taktzeitverhalten

2 x WCLK



A-3.7:

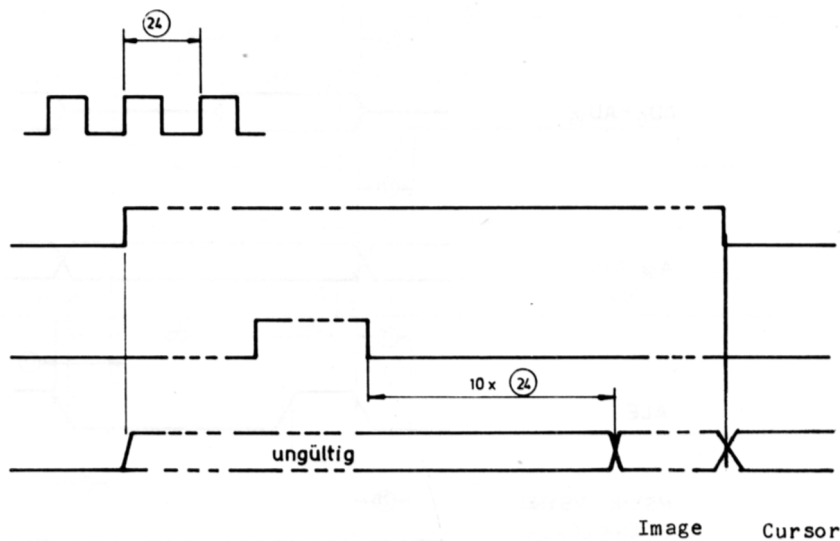
Zeitverhalten Cursor Image-Flag

2 x WCLK

HBLANK

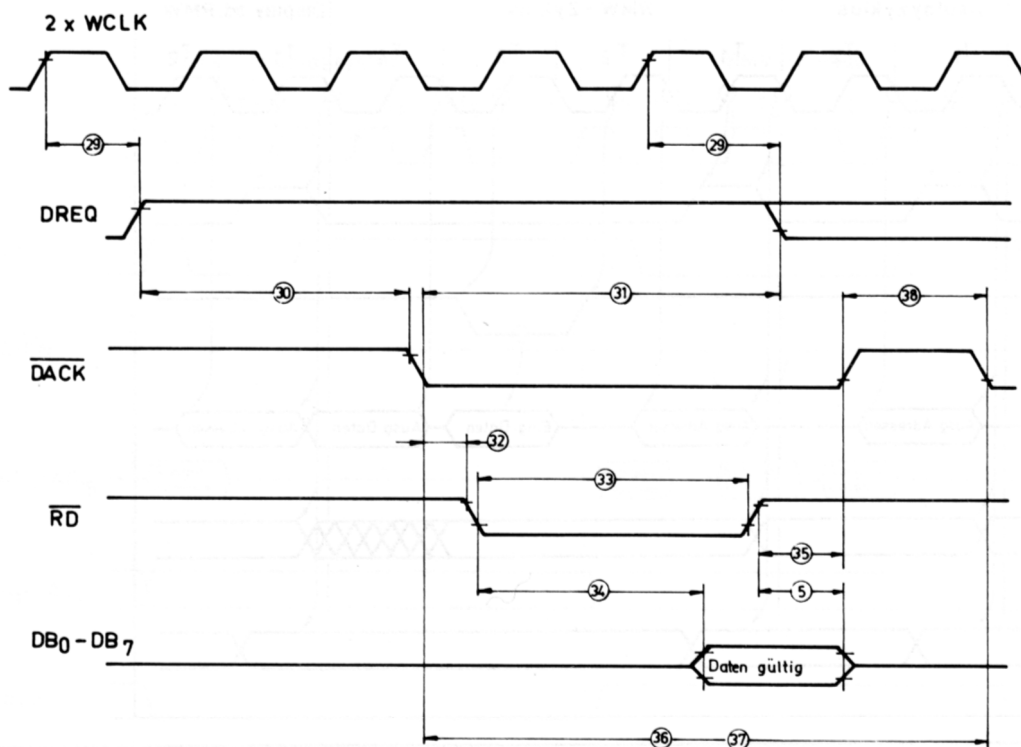
HSYNC

A 17



### A-3.8: DMA - Zeitverhalten

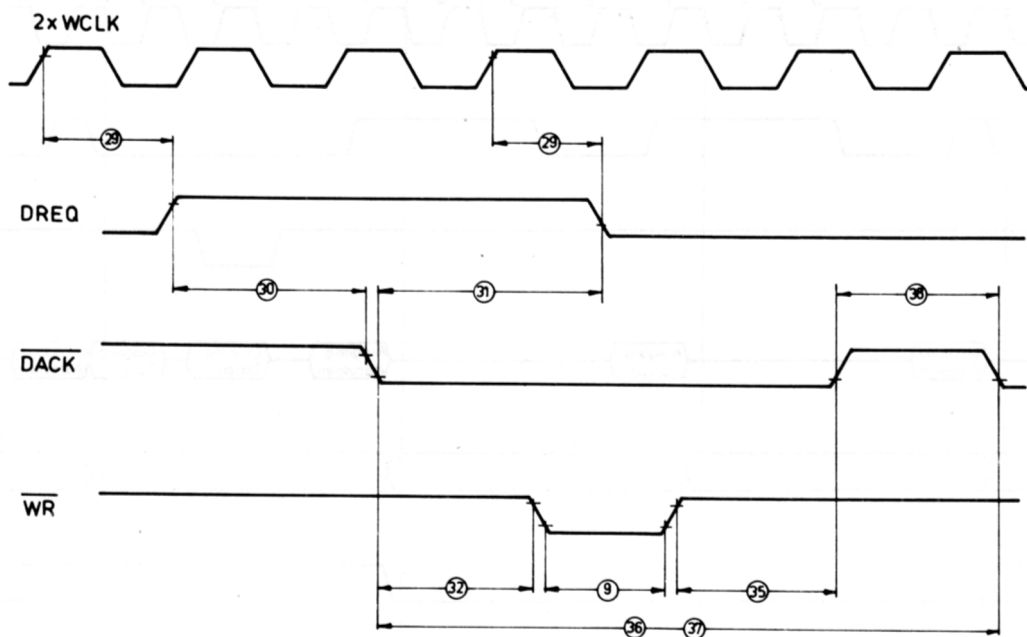
Lesen



### A-3.9:

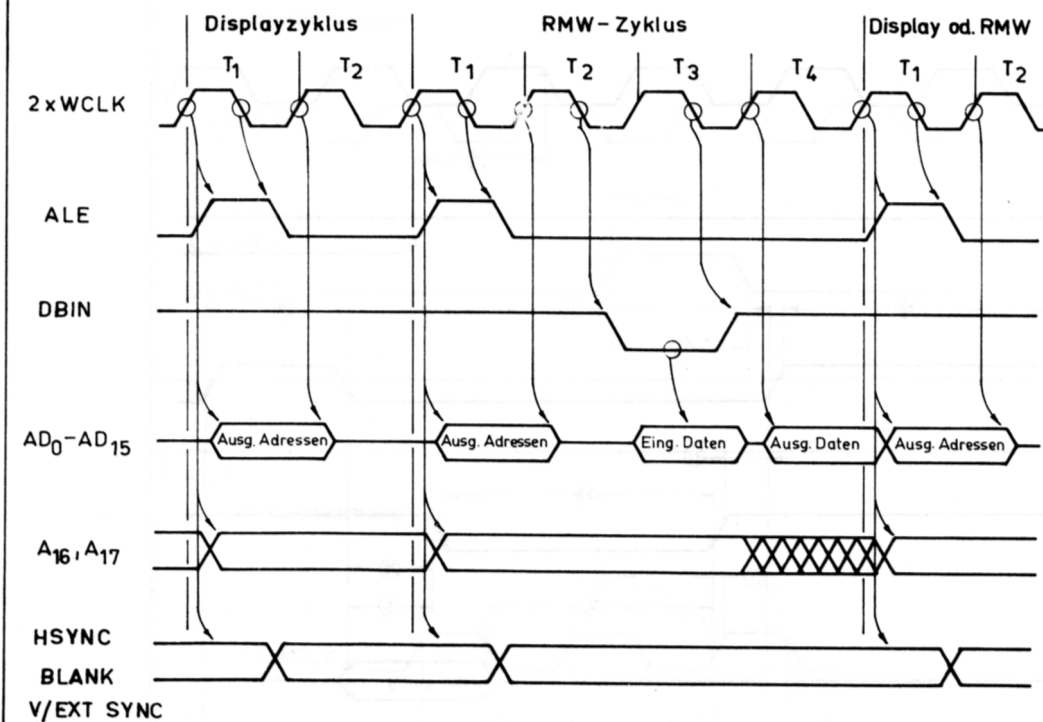
DMA - Zeitverhalten

Schreiben



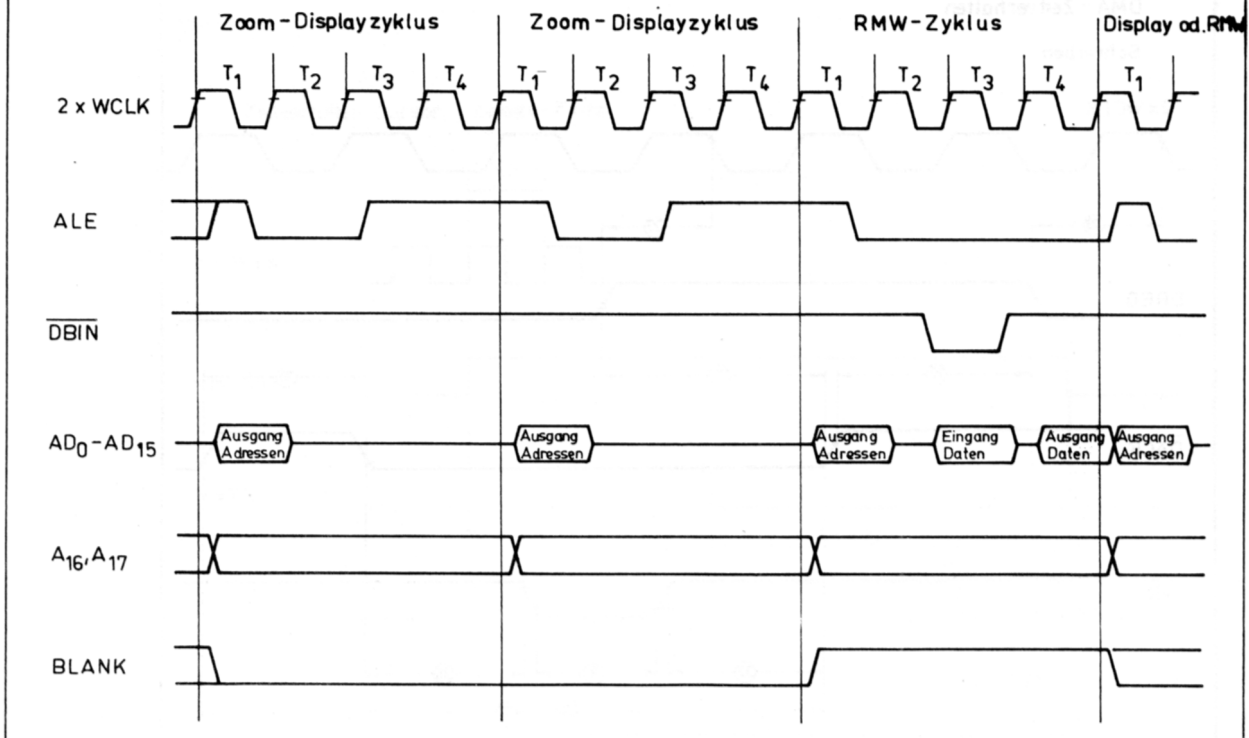
A-3.10:

Displayzyklen und RMW-Zyklen (1x ZOOM)



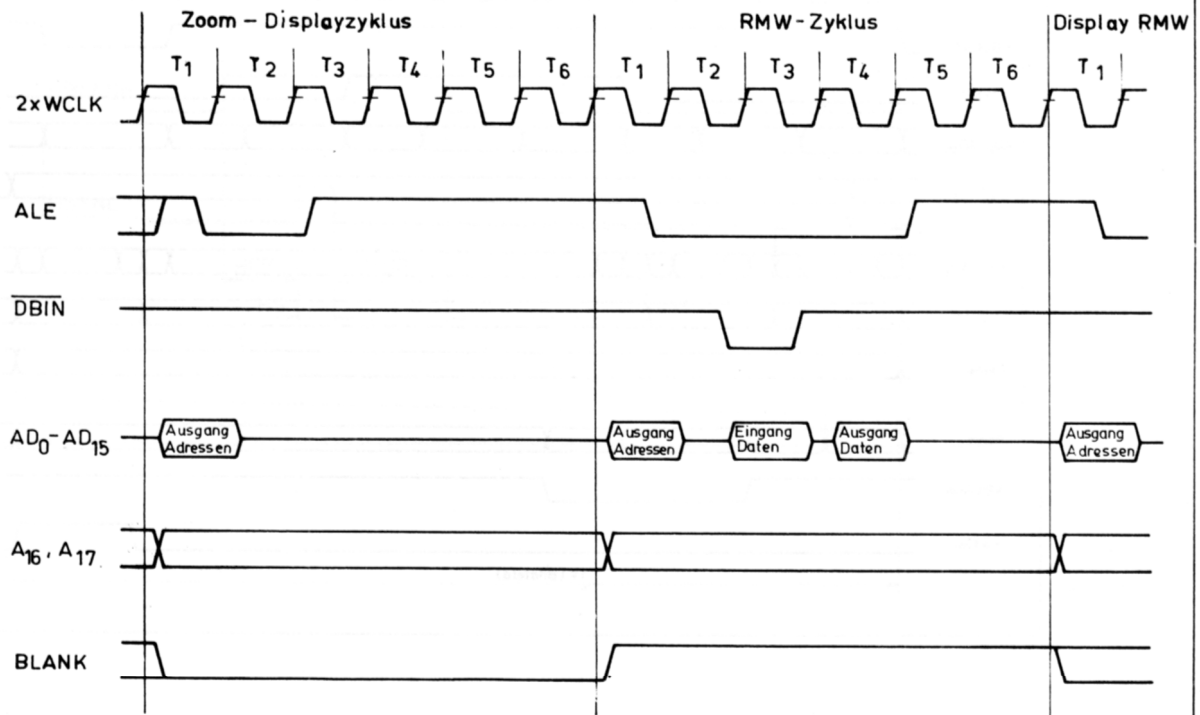
A-3.11:

Displayzyklen und RMW Zyklen (2x ZOOM)

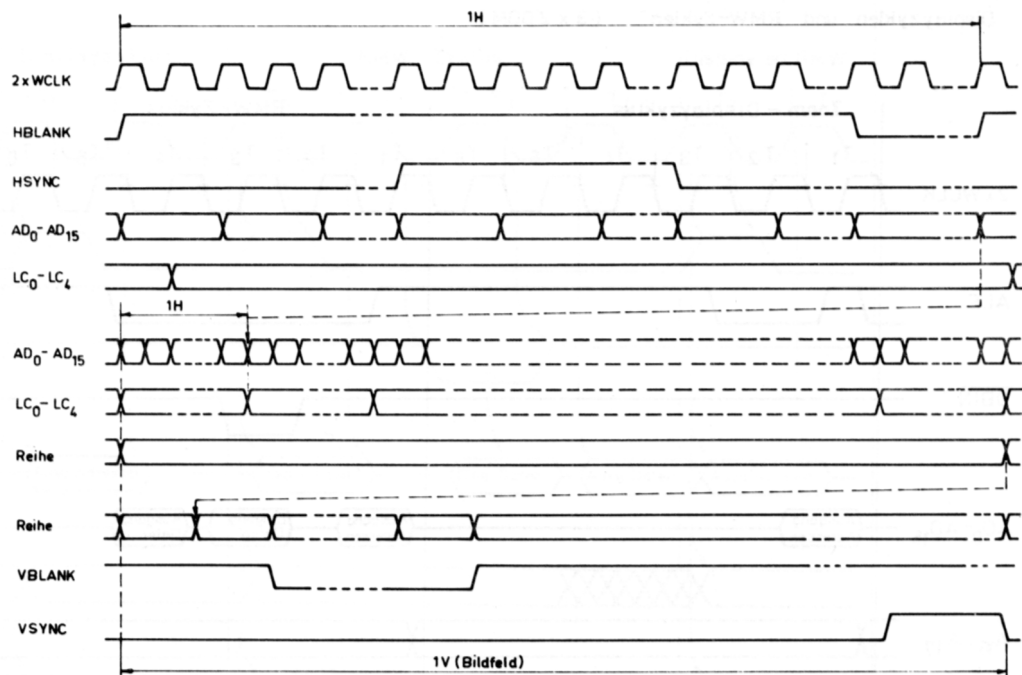


A-3.12:

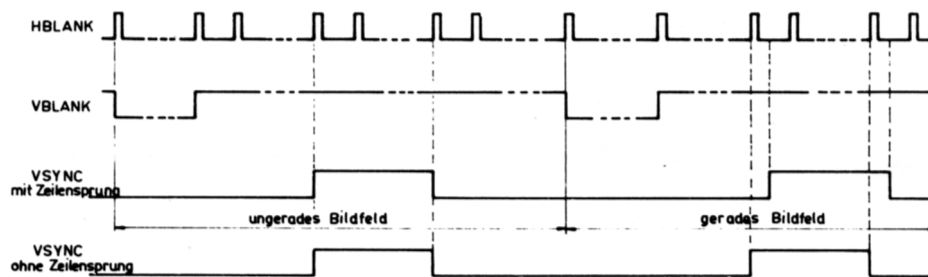
# Displayzyklen und RMW-Zyklen (3 x ZOOM)



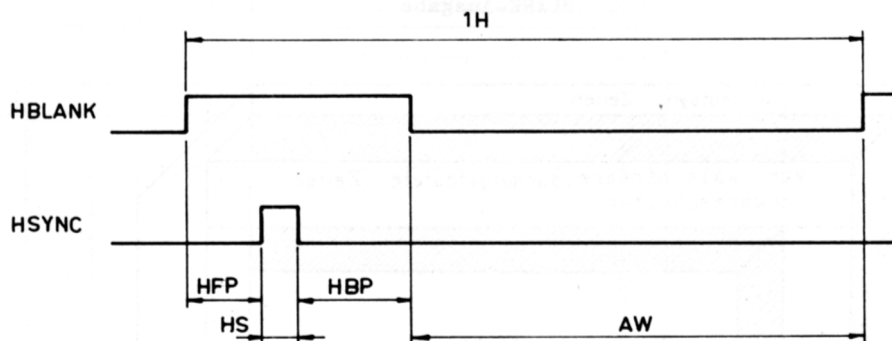
A-3.13: Zeitverhalten Video Synchronisationssignale



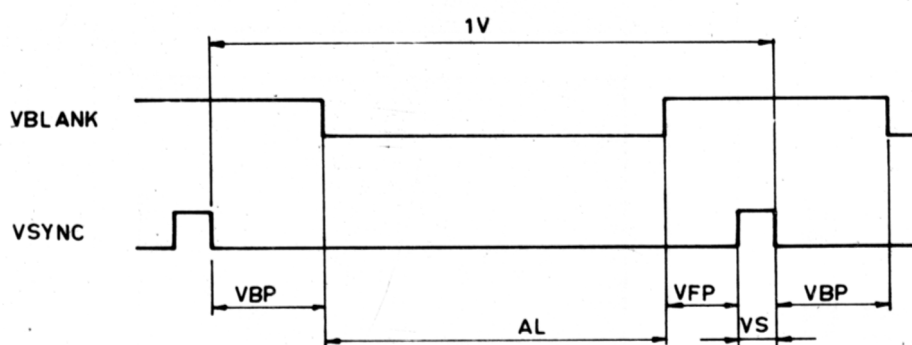
A-3.14: Zeitverhalten der Bilddarstellung mit Zeilensprungverfahren



A-3.15: Parameter des Horizontal-Synchronisationsgenerators

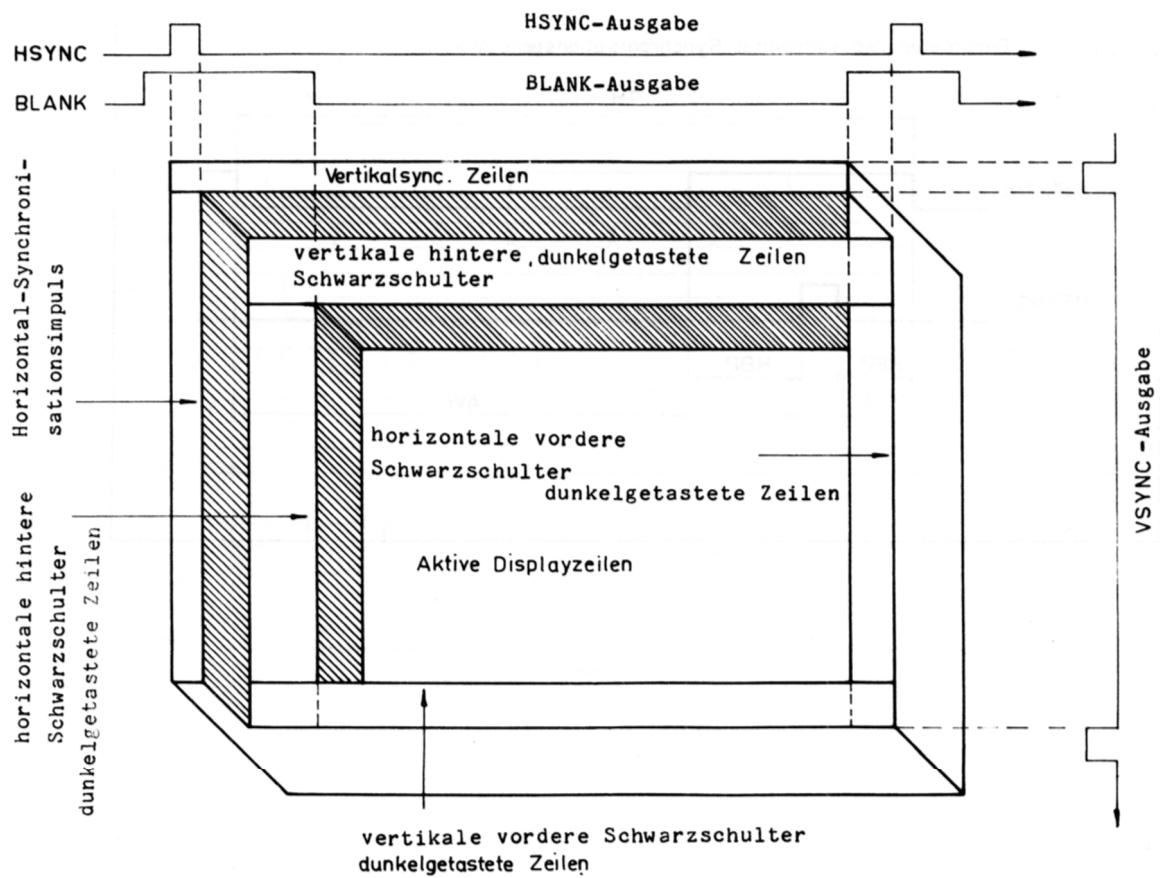


A-3.16: Parameter des Vertikal-Synchronisationsgenerators



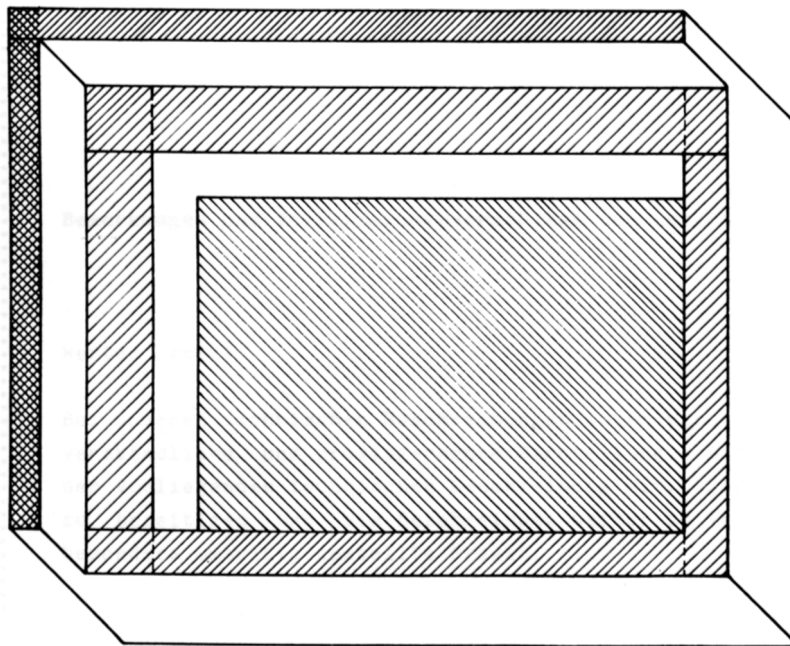
A-3.17:




# VIDEOFELD - ZEITVERHALTEN





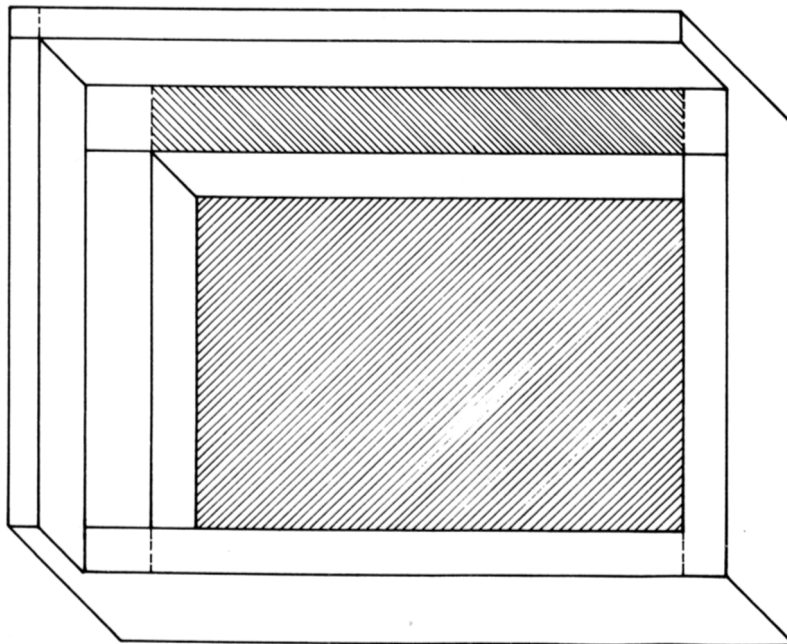
A-3.18: ZEICHNUNGSINTERVALLE





-  Zeichnungsintervall
-  zusätzl. Zeichnungsintervall,
-  dynam. RAM Refresh sonst weitere Zeichnungsintervalle

A-3.19:

DMA - ANFORDERUNGSINTERVALLE

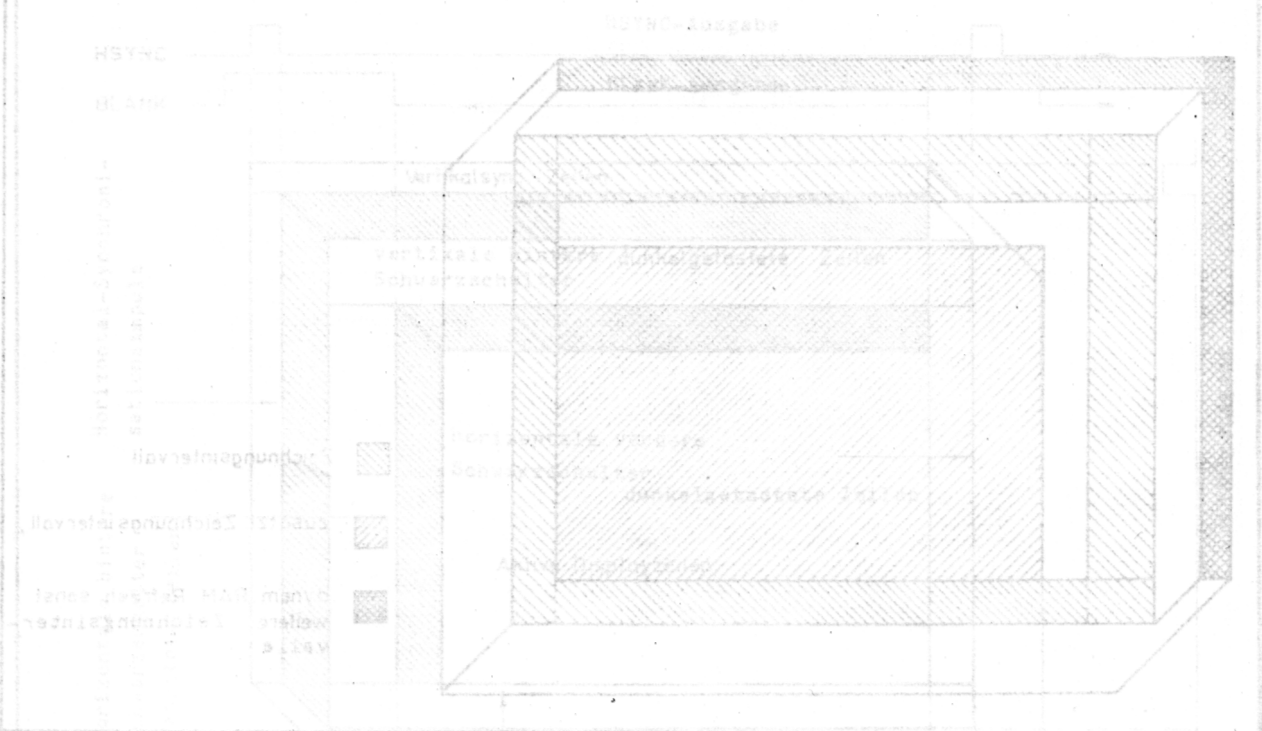


-  DMA - Anforderungsintervall
-  zusätzl. DMA Anforderungsintervall

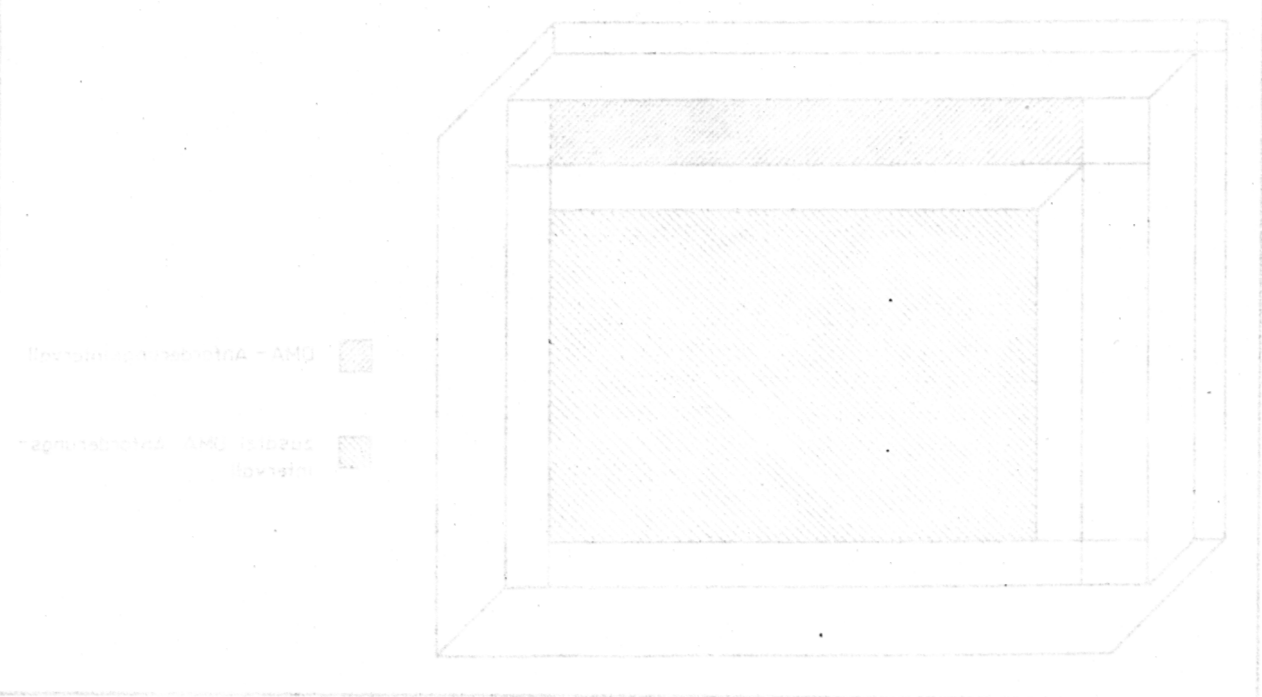
A-3.17

# Notizen

ZEICHNUNGINTERVALLE



## OMA - ANORDNUNGINTERVALLE



## **Bemerkungen und Verbesserungsvorschläge des Lesers**

Werter Leser!

Bei unseren technischen Beschreibungen legen wir Wert auf korrekte, vollständige, verständliche und praxisgerechte Darstellung. Bitte teilen Sie uns ihre Meinung zu der vorliegenden technischen Beschreibung mit. Sie helfen damit, eventuelle Fehler zu beseitigen und die Beschreibung weiter zu verbessern. Dabei interessieren uns besonders umseitig gestellte Fragen.

Ihre Hinweise richten sie bitte an:

veb mikroelektronik "karl marx" erfurt  
applikation bauelemente  
5020 Erfurt Rudolfstr. 47

**Bemerkungen und Verbesserungsvorschläge des Lesers**

Beschreibung:

Titel: Technische Beschreibung U82720

Ausgabe: 7/86

Hinweise des Lesers:

1. Sind Ihnen Druckfehler oder inhaltliche Fehler aufgefallen?
2. Enthält die Beschreibung die von Ihnen erwarteten und benötigten Informationen?  
Haben Sie Ergänzungsvorschläge?
3. Ist die Beschreibung für Sie verständlich? Wenn Sie Verständnisprobleme hatten,  
teilen Sie uns bitte das Kapitel und kurz den Sachverhalt mit.
4. Entsprechen Niveau, Umfang und Form dieser Beschreibung Ihren Anforderungen?

Absender:

Betrieb: .....  
Anschrift: .....  
Leser: .....  
Telefon: .....  
Datum: .....



**veb mikroelektronik › karl marx ‹ erfurt**  
**stammbetrieb**

DDR – 5010 Erfurt, Rudolfstraße 47 Telefon 580 Telex 061306

**elektronik**  
**export·import**

Volkseigener Außenhandelsbetrieb der  
Deutschen Demokratischen Republik  
DDR - 1026 Berlin, Alexanderplatz 6  
Telex: BLN 114721 elei, Telefon: 2180