

# Experimentier-Rechner

1. Überblick
2. Aufbau
3. Baugruppen
  - 3.1. [CPU-1 \(Z80A = UA 880 D\)](#)
    - 3.1.1. [Bauelemente-Basis](#)
    - 3.1.2. [Takterzeugung](#)
  - 3.2. [CPU-2 \(U 8000-System\)](#)
  - 3.3. CPU-3 (8086-System)
  - 3.4. RAM
  - 3.5. [GDC \( \$\mu\$ PD 7220PC = U 82720 D\)](#)
    - 3.5.1. [Konzept – Schaltungsbeschreibung](#)
      - 3.5.1.1. [Koppelinterface mit K1520-Bus](#)
      - 3.5.1.2. [Das I/O-Interface des GDC](#)
      - 3.5.1.3. [Takterzeugung für den GDC](#)
      - 3.5.1.4. [RAM-Interface für den GDC](#)
      - 3.5.1.5. Softscroll
      - 3.5.1.5. Softscroll
      - 3.5.1.6. Zeichengenerator
      - 3.5.1.7. Video-Signal-Erzeugung
      - 3.5.1.8. Das Zoom-Ink-Register
      - 3.5.1.9. Das CTL-Register
      - 3.5.1.10. Palettenspeicher, -Pointer und -Register
    - 3.5.2. GDC-Software – Grundroutinen
  - 3.6. Dual-Port-RAM
  - 3.7. [I/O-Baugruppe](#)
    - 3.7.1. [Funktionsumfang](#)
    - 3.7.2. [Bauelemente-Basis](#)
    - 3.7.3. [Realisierte Basisfunktionen](#)
      - 3.7.3.1. [Tastatur-Takterzeugung](#)

### 3.8. Interface für externe Speicher

### 3.9. Stromversorgung

## 4. Externe Speicher

## 5. Peripherie-Geräte

© Copyright by Peter Salomon, Berlin – Januar 2019

Die vorliegende Publikation ist urheberrechtlich geschützt. Alle Rechte, Irrtum und Änderungen vorbehalten.

Eine auch auszugsweise Vervielfältigung bedarf in jedem Fall der Genehmigung des Herausgebers.

Die hier wiedergegebenen Informationen, Dokumente, Schaltungen, Verfahren und Programmmaterialien wurden sorgfältig erarbeitet, sind jedoch ohne Rücksicht auf die Patentlage zu sehen, sowie mit keinerlei Verpflichtungen, noch juristischer Verantwortung oder Garantie in irgendeiner Art verbunden. Folglich ist jegliche Haftung ausgeschlossen, die in irgendeiner Art aus der Benutzung dieses Materials oder Teilen davon entstehen könnte.

Für Mitteilung eventueller Fehler ist der Autor jederzeit dankbar.

Es wird darauf hingewiesen, dass die erwähnten Firmen- und Markennamen, sowie Produktbezeichnungen in der Regel gesetzlichem Schutz unterliegen.

## 3. Baugruppen

### 3.1. CPU-1 (Z80A = UA 880 D)

#### 3.1.1. Bauelemente-Basis

Hauptbestandteile:

- CPU: UA880D
- Bustreiber: DS8286D, DS8282D, DL571D
- Reset-Logik
- Takterzeugung (U8127D, max. 8 MHz)
- Monitor-/Boot-EPROM (8k – U2764C)
- Monitor-/Boot-CMOS (8k – U6264D, Batterie-gestützt)
- Bank-/Boot-Baugruppe zur Adressierung von max. 16 MB,  
Bank-/Boot-IO-Adresse: 00H (vorläufige Festlegung)

#### 3.1.2. Takterzeugung

erfolgt auf Basis die programmierbaren Taktgenerators U8127D,  
modifiziert für:

- Einzelschrittbetrieb,
- verschiedene Taktfrequenzen,
- Reset-Logik,
- WAIT-Steuerung

#### Literatur zum U8127D:

Mikroprozessortechnik 1987, H7, S213 bis 215

Applikative Information 1987, H1, S24 bis 29

Rolf-Dieter Klein: Die Z80-CPU voll ausgebaut, MC (?), (?), S74-78 (liegt als Kopie vor)

### 3.2. CPU-2 (U 8000-System)

#### Literatur:

- [1] Hinweise zur Anwendung  
Handbuch U8000  
Mikroprozessor CPU U8001/U8002 - Technische Beschreibung  
VEB Mikroelektronik "Karl Marx" Erfurt, 100S
- [2] Programmierung des 16-Bit-Mikroprozessorsystems U8000  
VEB Verlag Technik 1987, von Dr.-Ing. Heiz Brennenstuhl  
(mit Assembler-Listing Monitorprogramm), 222S
- [3] Z8000 - Mikroprozessor-System  
te-wi-Verlag 1980, von Peter Stuhlmüller  
(viel Hardwareprobleme, Multiprozessoranwendungen u.a.), ca.450S
- [4] Frei programmierbarer Arithmetikmodul für den K1520 (mit U8002D)  
rfe 1984, H8, S483-485, 492, von Fey, P.; Kriesten, S.; Rieken, R.
- [5] Leistungsfähige 16-bit-ZRE-Karte für K1520  
rfe 1983, H10, S629-631, 636, von Rehm, W.; Fey, P.
- [6] Testhilfen für 16-bit-Mikrorechner  
rfe 1985, H4, S221-224, von Dipl.-Ing. Ralf Rieken, Dipl.-Ing. Mattias Kirchhoff
- [7] Universelles 16-bit-System USS 8000  
rfe 1985, H5, S282-285, von Dipl.-Ing. Wolfgang Rehm  
(u.a. Bussignal-Belegung der ZRE.UNI)
- [8] Erweiterung des K1520 durch Integration eines 16-bit-Mikrorechners (mit U8001/2)  
rfe 1985, H8, S495-497, von Dipl.-Ing. Nobert Braumüller, Dr.-Ing. Heinz Brennenstuhl, u.a.
- [9] Das 16-bit-Mikroprozessorsystem U8000  
rfe 1985, H11, S687-691 (T1); H12, S760-763, von Dipl.-Ing. Frank Meinecke
- [10] Festplattensteuerung für U8000-Rechner  
rfe 1988, H8, S516-517, von Dr.-Ing. Ralf Rieken
- [11] Einplatinen-Computer mit Z8000 (Z8002)  
ELEKTRONIK 1980, H16, S44-50, von Rolf-Dieter Klein  
(sehr gute Hardwarebeschreibung mit Schaltbild und Monitor-Listing)

### 3.5. GDC

Als Herzstück des GDC kommt die IS 72720 ( uPD 7220 AD ) zum Einsatz. Es handelt sich hierbei um eine hochintegrierte Schaltung mit komplexen Wirkungsmechanismen in Form eines eigenen Grafik-Prozessors. Näheres ist den einschlägigen Produktbeschreibungen zu entnehmen (siehe Literatur).

#### Literatur:

Literaturzusammenstellung für den GDC

- [1] "Hinweise zur Anwendung Grafik-Display-Controller GDC U82720"  
Applikation Technische Beschreibung  
VEB Mikroelektronik "Karl-Marx" Erfurt Stammbetrieb, A4, 114S
- [2] Bankel, Mario; Brückner, Peter:  
"Grafik-Interface mit dem U82720",  
Mikroprozessortechnik 1987, H4, S99-103  
(grundlegende Schaltungen)
- [3] Quednow, Wilfried, Dr.: "Videosteuerung VIS 3A mit GDC U82720D",  
Mikroprozessortechnik 1988, H3, S66-68 (Teil 1)  
H11, S339-341 (Teil 2)
- [4] Fischer, Matthias, Dipl.-Ing.; Wolf, Thomas, Dr.-Ing.:  
"Farbgrafikmodul mit Grafik-Display-Controller",  
rfe 1988, H4, S253-256  
(GDC 1.1)
- [5] Fischer, Matthias, Dipl.-Ing.; Wolf, Thomas, Dr.-Ing.:  
"Farbgrafiksteckereinheit GDC-2"  
rfe 1989, H4, S209-213
- [6] Sänger, Peter, Dr.-Ing.; Dittmar, Klaus, Ing.:  
"Farbgrafiksteckereinheit GDC-3",  
rfe 1990, H8. S529-532  
(... der von mir versuchte Nachbau,  
angepasste Variante auf 2x K1520-Steckereinheitengröße)
- [7] "Mikroelektronik in der Amateurpraxis", Bd. 4, Brandenburgisches Verlagshaus, 1990,  
1. Kapitel: "Bilder im Computer", von Bernd Hübler  
(Detail-Schaltbilder Grafik-BG mit GDC U82720 und Grundsoftware)
- [8] "Betriebssystem CP/M - Aufbau und Anwendung", Brandenburgisches Verlagshaus, 1990,  
von Manfred Kramer  
(hier auf S115-125 ein Z80-Assembler-Listing für GDC U82720)
- [9] Oettle und Reichler: "Die Farbgrafik-Karte",  
mc 1985, H4, S136-144  
(Hardware-Aufbau)
- [10] Guido Dampf: "Grafik mit dem 7220 von NEC",  
mc 1986, H11, S54-65  
(Software-Programme in Turbo-Pascal)

- [11] Johannes Assenbaum: "Flott gepixelt",  
c't 1990, H5, S260-276  
H6, S285-289  
H7, S200-208  
H8, S286-295  
(Video-Interface zum c't68020, mit uPD7220A)
- [12] INTEL Microsystem Components Handbook 1984  
Volume II, 8-106, DATA-SHEETS 82720  
8-91, AR-255,  
("Dedicated VLSI Chip Lightens Graphic Display Design Load")  
8-99, AR-298  
("Graphics Chip Makes Low Cost High Resolution Color Displays Possible")  
(leider nur als Microfish)
- [13] Hamid Assarpour, NEC Electronics Inc.:  
"Graphics controller chip raises video data rate, is simpler to program"  
Electronic Design 1984, July 12, S135-142, ,  
(betr. uPD7220A)
- [14] P. Prabhaker Rao and S. Srinivasan:  
"New design for an 82720-based colour graphics generator",  
microprocessors and microsystems Vol 10, 1986, No 7, S386-391
- [15] S. Manohar, N. Murali and L. M. Patnaik:  
"Design of a Multibus-compatible colour graphics subsystem",  
microprocessors and microsystems Vol 10, 1986, No 8, S284-289  
(betr. NEC 7220)

### 3.5.1. Konzept - Schaltungsbeschreibung

Die GDC-StE wird auf zwei K1520-Lochrasterplatten aufgebaut, die mit einem 58-pol. Steckverbinder verbunden. Demzufolge beansprucht diese StE 2 Steckplätze!

Es wird versucht die Variante nach GDC 3 [6] zu realisieren.

I/O-Basis-Adresse:	0CxH	(DIL-Schalter: HLL)		
	0/1	GDC		
	2/3	?		
	4/5	CTR-Register		
	6/7	ZIR-Register	ZOOM	INK
	8/9	PAR-Register	X X	RF P2 P1 P0
	A/B	PPT-Register	X A6 A6 A4 A3 A2 A1 A0	
	C/D	PSP-Register	D7 D6 D5 D4 D3 D2 D1 D0	+ A8 bis A11!
	E/F	ASCI-Register	D7 D6 D5 D4 D3 D2 D1 D0	

#### 3.5.1.1 Koppelinterface mit K1520-Bus

Der 8-Bit breite K1520-Datenbus D0 - D7 (Ltg. 101 - 108) wird über einen bidirektionalen Bus-Treiber 8286 an den internen Datenbus (Ltg 122 - 129) angekoppelt. Die Datentransfer-Richtung ist normalerweise in Richtung GDC. Nur wenn über den /DIR-Eingang das LOW-aktive Bus-Signal /RD ( Ltg. 118 ) anliegt, wird die Datentransfer-Richtung umgedreht.

Die TriState-Ausgänge werden über die Ltg. 109 ( LOW-aktiv ) nur bei einer gültigen IO-Adresse angesteuert. Diese wiederum wird vom 8-Bit-Komperator 8121 aus den IO-Adressen A4 - A7 (Ltg. 114 - 117) entsprechend der DIP-Schalterstellung gebildet, wobei die Freigabe über /STR nur beim Anliegen von /IORQ (Ltg 120) erfolgt.

Zusätzlich verhindert die Einbindung von /M1, daß bei Interrupt-Signalen eine gültige IO-Adresse gebildet werden kann.

Bei einer DIP-Schalterstellung

B0 = H

B1 = H

B2 = L

$$B3 = L$$

ergibt das am Komparator z.B. die Gruppenadresse 0CxH.

Die weitere Decodierung der IO-Adressen erfolgt mit dem Decoder 8205. Hiermit sind folgende Einzeladdress-Paare erreichbar. Die paarweise Adressierung ergibt sich aus der Eigenart des *I/O-Interfacings des GDC*, welcher keinen Chip-Select-Eingang hat und somit mit A0 direkt adressiert werden muß ( weitere Beschreibung zu diesem Schaltungsteil, siehe dort )

0/1	-	GDC ( Ltg. 229 )
2/3	-	(z.Zt. noch offen)
4/5	-	CTR ( Ltg. 140 )
6/7	-	ZIR ( Ltg. 142 )
8/9	-	PAR ( Ltg. 144 )
A/B	-	PPT ( Ltg. 146 )
C/D	-	PSP ( Ltg. 147 )
E/F	-	ASCI ( Ltg. 148 )

Des Weiteren wird das Bus-Signal /RD ( Ltg. 118 ) und /WR ( Ltg. 119 ) für das *I/O-Interfacings des GDC* ( weitere Beschreibung zu diesem Schaltungsteil, siehe dort ) benutzt.

### 3.5.1.2. Das I/O-Interface des GDC

Für die bidirektionale Kommunikation des GDC mit dem Systembus ist das Signal A0 ( Ltg. 229 ) in Verbindung mit den Signalen /RD ( Ltg. 131 ) und /WR ( Ltg. 132 ) zuständig. Dabei gilt folgende Zuordnung:

<b>A0</b>	<b>/RD-aktiv</b>	<b>/WR-aktiv</b>	<b>RD-/WR-inaktiv</b>
	( Leseoperation )	( Schreiboperation )	( beide "High" = keine Operation)



“0”	GDC-Statusregister	GDC-FIFO	“Parameter”
“1”	GDC-FIFO	GDC-FIFO	“Kommandos”

Die Adressierung des GDC mit A0 wird jedoch noch mit dem NOR-Gatter 1.2 beeinflusst, d.h. wenn der *Zeichengenerator* ( weitere Beschreibung zu diesem Schaltungsteil, siehe dort ) ein Byte-Satz für ein Zeichen in das Parameter-RAM des GDC schreiben will, wird A0 zwangsweise auf “LOW” gesetzt ( Adresse “0” ). Durch das Gatter 4.6 muß deshalb A0 ( Ltg. 110 ) zunächst invertiert werden, weil das NOR-Gatter 1.2 das Signal wiederum invertiert.

Der /RD-Signal des GDC ( Ltg. 131 ) wird über das NOR-Gatter 1.1 aus dem Bus-Signal /RD ( Ltg. 118) und dem GDC-Select-Signal ( Ltg. 136 ) gebildet, wobei mit Gatter 4.1 die durch Gatter 1.1 hervorgerufene Invertierung des Signal wieder aufgehoben wird. Je nach Zustand A0 werden die Daten aus dem GDC-FIFO oder GDC-Status-Register dann bei /RD-Anforderung über den 8286 auf den K1520-Bus gelegt.

### 3.5.1.3. Takterzeugung für den GDC

Der Grundtakt wird mit einem integriertem 16-Mhz Quarztaktgenerator ( Ltg. 209 ) erzeugt. Dieser wird mit dem Gatter 2.1 invertiert ( Ltg. 210 ).

Mit dem rückwärtszählenden Binärzählers CT1 wird entsprechend des ladbaren Zoom-Faktors ( Ltg. 205 - 208 ) das Video-Schieberegister-Taktsignal  $SR_{CLK}$  ( Ltg. ? ) gebildet, d.h. je nach Ladewert an den Ladeeingängen A - D des Zählers wird der Übertragsimpuls nach Erreichen des Zählerwertes “0” gebildet. Gleichzeitig wird der Zähler wieder neu mit dem Zoom-Faktor (Hardware-Zoom) aus dem *Zoom-Ink-Register* geladen ( weitere Beschreibung zu diesem Schaltungsteil, siehe dort ).

Zur Synchronisierung mit dem Ladevorgang der Video-Schieberegister (  $SR_{LOAD}$  ) wird mit dem Signal  $R_{CT}$  ( Ltg. 235 ) der Zähler

rückgesetzt.

Z.Zt. erfolgt die Erprobung ohne Hardware-Zoom, d.h. die Video-Schieberegister werden mit dem vollen 16x/WCLK ( Ltg. 210 ) getaktet, die Zoomfaktorsteuerung mit CT1 ist wirkungslos.

Für die Ablaufsteuerung, insbesondere des *RAM-Interface* ( weitere Beschreibung zu diesem Schaltungsteil, siehe dort ) wurde zunächst eine Lösung mit dem Binärzähler CT2 - ebenfalls im Rückwärts-Zählbetrieb - vorgesehen, die Signale 8x WCLK ( Ltg. 211 ), 4x WCLK ( Ltg. 212 ) und

*Achtung!*

*Hardwareänderung an  
CT2 (QA-QD) -  
noch nicht realisiert!*

2x WCLK ( Ltg. 213 ) abzuleiten.

( siehe auch Taktdiagramm GDC - "BWS-Lesen" )

*Achtung!*

*Hardwareänderung -  
ist noch nicht realisiert!  
Funktion prüfen!*

Dabei ergab sich das besondere Problem der Synchronisierung des internen Taktzählers des

GDC (davon werden alle weiteren GDC-Steuersignale des RAM-Interface abgeleitet) mit dem CT2. Dazu wurde mit dem zusätzlichen Gatter 16/1 das ALE-Signal

*nachträglicher Einbau  
über CT 3*

mit dem invertierten Signal verknüpft. Durch die Laufzeitverzögerung der Gatter 5.4 und 2.6 inclusive dessen zusätzlicher kapazitiver Belastung mit ca. 50pF ergibt sich eine sehr kurze L-Nadel ( Ltg 215 ), die als Ladeimpuls für CT2 dient.

CT2 wird entsprechend seiner Eingangsbeschaltung  $A-D=U_B$  mit "H" zum Zeitpunkt der L/H-Flanke des ALE-Signals mit Zählerwert "15" geladen und zählt dann rückwärts.

Nur mit dieser Synchronisation ist der weitere Taktablauf entsprechend Taktdiagramm möglich. Leider ergibt sich damit aber auch eine ungünstige Taktverlängerung des H-Teils des 1. Taktes GDC-Taktes, was wiederum negative Auswirkungen auf das weitere Taktgeschehen hatte.

Diese Schaltungsvariante wurde verworfen, d.h. die „Ladesynchronisation“ wurde wieder rückgängig gemacht und somit können aus den o.g. Gründen die abgeleiteten Signale 4xWCLK und 8x WCLK nicht verwendet werden.

Die in der Literatur [ ] [ ] angegebene Schaltungsausführung mit einem Schieberegister war so ohne weiteres nicht einsetzbar, da ja hier auch die besondere Betriebsart "Wide-Display-Mode" realisiert werden sollte. *gehört zu RAM-Interface!*

Mit dem Taktsignal 2x WCLK ( Ltg. 135 = Ltg. 212 ) wird der GDC getaktet.

*Vereinheitlichen!*

Davon abgeleitet erzeugt der GDC das ALE-Signal ( Ltg. 134 ), d.h. ca. 30 bis max. 120ns nach der steigenden Taktflanke wird ALE solange HIGH, bis die fallende Taktflanke ALE wieder zurückschaltet ( Verzögerung ebenfalls ca. 30 bis max. 120ns ).

ALE zeigt an, daß eine gültige 16-Bit-Adresse auf dem gemultiplexten Adress-/Datenbus des GDC anliegt. Diese muß mit der H/L-Flanke von ALE zwischengespeichert werden, da bereits nach 30 bis 160ns nach der nächsten L/H-Takt-Flanke der Adress-/Datenbus undefiniert ist. Beginnend zu diesem Zeitpunkt bis spätestens zum Ende des zweiten GDC-Taktes müssen dann die RAM-Daten bereitstehen. (siehe dazu auch weitere Erklärung zum *RAM-Interface*)

Nach Ende des zweiten GDC-Taktes, d.h. der L/H-Flanke des nächsten Taktes wird wiederum mit 30 - 160ns Verzögerung das nächste ALE-Signal gebildet und somit der nächste BWS-Zyklus eingeleitet.

Der Bildwiederholpeicher wird also während dieser Zyklen nur lesend angesprochen.

Um jedoch Information in den BWS einschreiben zu können, bedarf es des sogenannten "Read-Modify-Wright" ( RMW )-Zyklus

Das GDC-Signal /DBIN zeigt an, daß ein RMW-Zyklus stattfindet, d.h. der GDC legt zunächst wie beim BWS-Zyklus eine 16-Bit-Adresse auf den gemultiplexten Adress-/Datenbus, jedoch wird ca. 30 - 160ns nach der H/L-Flanke des 2.Taktes das GDC-Signal /DBIN (Ltg. 133) Low-aktiv. Damit signalisiert der GDC, daß die aus dem BWS ausgelesenen Daten spätestens ca. 30 - 160ns nach der H/L-Flanke des 3.Taktes an den nunmehr auf Eingang geschalteten Adress-/Datenbus des GDC bereitstehen müssen. Der GDC liest sie ein modifiziert sie und stellt sie ca. 30 - 160ns nach der L/H-Flanke des 3.Taktes ( 2x WCLK ) wieder an dem gemultiplexten Adress-/Datenbus bereit, um sie in das RAM zurück schreiben zu können.

#### **3.5.1.4. RAM-Interface für den GDC**

Bei der Konzipierung des RAM-Interface sind entsprechend den Anforderungen und Möglichkeiten des GDC folgende Ausgangspunkte bzw. Bedingungen zu beachten:

- a) Der gemultiplexter Adress-/Datenbus des GDC-RAM's ist 16Bit breit, wobei der Adressbus nur für 1 Taktimpuls an den GDC-Ausgängen AD0 - AD15 zur Verfügung steht. Dieser Taktzustand wird durch das GDC-Signal ALE = HIGH ( Ltg. 134 ) angezeigt.
- b) Der volle Adressbereich des GDC erstreckt sich entsprechend den Adressleitungen A(D)0 -A(D)15 und den zusätzlichen Adressleitungen

A16, A17 von 0.0000H bis 3.FFFFH, also 4MBit oder 2x 256kByte adressierbarer Speicherraum

- c) Es sollen sowohl 64k\*1-DRAM's wie auch 256k\*1-DRAM's zum Einsatz kommen können  
(entsprechend der 16 Datenleitungen jeweils 16 Stk)
- d) Durch den DRAM-Einsatz bedingt muß ein Refresh-Regim gesichert werden.  
Ein Hidden-Refresh soll nicht zur Anwendung kommen, da damit wertvolle GDC-Arbeitszeit verbraucht würde.
- e) Als besonderes Feature des GDC soll beim sog. "Wide-Diplay-Mode" im BWS-Zyklus ein zweiter Zugriff auf die nächst höhere BWS-Adresse erfolgen, ohne daß dazu eine konkrete Adressausgabe des GDC erfolgt.

Der Betrieb von DRAM's als BWS zeichnet sich dadurch aus, daß durch das im Bildaufbau festgelegten Zeitregim eine bestimmte Anzahl Speicher-Adressen ständig inkrementierend nacheinander angesprochen, d.h. ausgelesen werden.

Typische Bildschirmmodi sind dabei z.B.:

- 40 Zeichen zu 8Bit = 320 x 256 Zeilen,  
wobei 8 Zeilen wiederum eine Zeichenzeile sind,  
d.h. damit ergeben sich 16 Zeichenzeilen  
(“Heimcomputerstandard”)
- 80 Zeichen zu 8Bit = 640 x 512 Zeilen,  
d.h. 32 Zeichenzeilen und damit ähnlich dem “PC-Standard (DOS)”,  
welcher hier realisiert werden soll,

wobei zusätzliche Farbmodi zunächst hier nicht betrachtet werden.

(siehe dazu 3.3.1.7 - 10)

Das Zeitregim richtet sich dabei nach dem zu verwendenden Monitor. Normale FS-Monitore arbeiten mit einer Zeilendauer von ca. 64  $\mu$ s und 50Hz (=20ms) Bildwechselfrequenz, moderne Computer-Monitore hingegen mit Bildwechselfrequenzen von über 70Hz.

DRAM's benötigen für den Erhalt ihrer Daten regelmäßige Refresh-Zyklen. Beim BWS müssen diese im Gegensatz zu normalen Speicherbänken nicht extra eingefügt werden, wenn gesichert ist, daß durch die fortlaufende Adressierung alle notwendigen Seiten-Adressen innerhalb der Refreshzeit angesprochen werden.

Die maximale Refreshzeit beträgt normalerweise 2ms bei 64k DRAMs und 4ms bei 256k DRAMs. Nun werden aber die Seitenadressen mit dem /RAS-Zyklus angesprochen und dabei wird nur dann fortlaufend adressiert, wenn mit der üblichen Adressierungsfolge LOW-Byte (/RAS)- HIGH-Byte (/CAS) gearbeitet wird.

Das grundlegende Problem der Realisierung des notwendigen Refresh-Regims (d) für die DRAM's läßt sich also damit lösen, daß innerhalb der Refresh-Zeit nach DRAM-Spezifikation (ca. 2ms) wenigstens alle DRAM-Seiten ( nur mit /RAS zu adressieren! ) angesprochen werden. Bei den modernen DRAM's sind das lediglich 128 Adressen ( ist hier bei 64k\*1 bzw. 256k\*1 gleich ) bzw. die 7 DRAM- Adressleitungen A0 bis A6.

Bei den 640 Bit/Zeile und einem Wort-Zugriff von jeweils 16Bit ergibt das also 40 Zugriffe/Zeile in ca. 64  $\mu$ s. Nach Auslesen von 4 Zeilen zu insgesamt 160 Zugriffen wäre also innerhalb von 256 $\mu$ s die Refresh-Zeit sehr gut eingehalten. Stehen durch Aufteilung des DRAM's in Farb-Ebenen nur noch z.B. 4Bit pro Zugriff zur Verfügung, so stellt das insofern kein Problem dar, weil dieses lediglich mehr Zugriffe/Zeile erfordert und somit die Refresh-Zeitbedingung noch eher erfüllt werden.

Wie ist (sind) jedoch in diesem Zusammenhang das "Wide-Display-Mode"-Problem (e) zu lösen?.

Wird der GDC in diese Betriebsart programmiert ( Setzen des WD-Bit ), so wird bei der BWS-Adressierung - und nur dabei (!) - die BWS-Adresse jeweils um 2 inkrementiert, d.h. wenn sichergestellt ist, daß die Anfangsadresse des BWS ( ebenfalls programmierbar ) auf einen geradzahligen Wert festgelegt ist, ändert sich A0 nicht mehr, sondern bleibt immer auf "0"!

Damit besteht die Möglichkeit innerhalb eines BWS-Zyklus anstatt der normalerweise möglichen 16 Bit Video-Information nunmehr 32 Bit zu erzeugen. Es muß "lediglich" innerhalb eines BWS-Zyklus durch einen weiteren DRAM-Zugriff auf der benachbarten ungeraden Adresse weitere 16 Bit ausgelesen werden. Dies gilt aber nur für den normalen BWS-Zyklus. Soll der GDC durch Abarbeitung von GDC-eigenen Befehlen einen RMW-Zyklus durchführen müssen ( Signal /DBIN ), muß natürlich auch der Zugriff auf die ungerade Adresse möglich sein. Die Arbeitszugriffe des GDC ( RMW-Zyklus ) auf den BWS sind deshalb davon nicht berührt, diese erfolgen nach wie vor auf alle BWS-Adressen (\*).

*Einschränkungen bzgl. phys.  
DRAM und prog. BWS  
müssen noch geklärt werden!  
( Fehlerbehandlung ?)*

Der doppelte DRAM-Zugriff muß deshalb innerhalb eines BWS-Zyklus stattfinden.

Das ist jedoch mit einer vollständigen RAS/CAS-Adressierung innerhalb des vorgegebenen Zeitregims von 2 GDC-Takten kaum möglich. Kurze DRAM-Zugriffe lassen sich jedoch im sog. Page-Mode vornehmen. Dazu kann nach einer RAS-Adressierung mit nachfolgender mehrmaliger ( hier 2x ) CAS-Adressierung zugegriffen werden. Die Grenze der Zugriffs-Anzahl wird dabei durch die Refresh-Bedingungen gesetzt, was hier aber nicht relevant ist.

Daraus schlußfolgernd müßten die niederwertigen DRAM-Zugriffe

( A0 - A7 bzw. A8 ) CAS-gesteuert sein. Das widerspricht aber dem o.g. hinsichtlich der Refresh-Bedingungen, es sei denn, man beschränkt die niederwertige CAS-Adressierung auf A0.

Damit ergibt sich zwar die doppelte Refresh-Zeit von ca. 0,5ms, diese ist aber immer noch weit unter der 2ms-Grenze.

Zur schaltungstechnischen Realisierung im Detail siehe weiter unten.

Nun zu dem Problem der 64k\*1/256k\*1-DRAM-Kompatibilität (c).

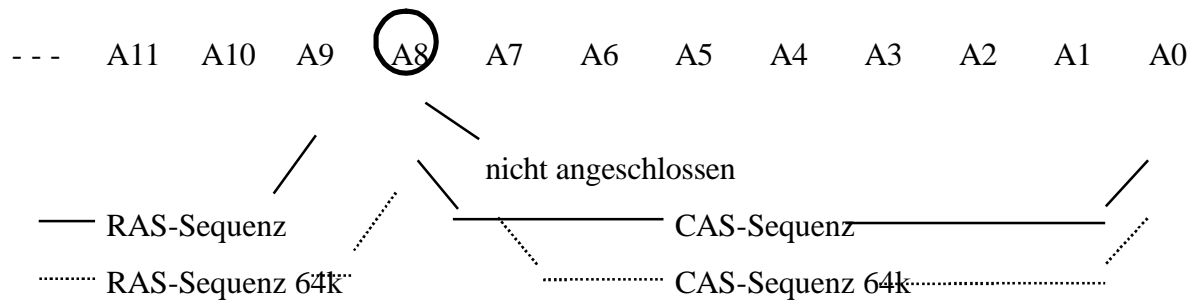
Mit den Adress-/Datenleitungen AD0 - AD15 ( Ltg. 149 - 164 ) würde sich ein Adressraum von 64k ansprechen lassen, d.h. von Adr. 0000H - 0FFFFH, wohlgemerkt in Worten zu 16 Bit.

Wenn 64k-DRAMs verwendet werden, so benötigen diese 2x 8Bit für die Adressierung, z.B. A0 - A7 für /RAS und danach A8 - A15 für /CAS. Schaltungstechnisch kann das z.B. mit TTL-Multiplexern, z.B. 74157, 74257 realisiert werden, oder wie hier, mit den 8283-Bustreiber-IC's. Sollen jedoch für den gesamten, für den GDC zu adressierenden Adressbereich ( 4 MBit mit den Adressen AD0 - AD15 + A16, A17 ) in der gleichen Beschaltung auch 256k-DRAM's eingesetzt werden können, so müßte der Multiplexer A0 - A8 für /RAS und danach A9 - A17 für /CAS an die DRAM-Adresseingänge A0 - A8 schalten. Beim Einsatz von 64k-DRAM's ist dann aber in dieser Beschaltung A8 nicht belegt!

Es würden somit beim Einsatz von 64k-DRAM's im o.g.

Adressierungsmodus nur folgende Adressbereiche des linearen GDC-Adressraumes angesprochen werden können:



**GDC-Adressraum**

0000H - 00FFH

0100H - 01FFH sind nicht adressierbar,

d.h. es werden wieder die 0000 - 00FFH-Adressen angesprochen

0200H - 02FFH sind wieder adressierbar,

0300H - 03FFH wieder nicht, usw.

Für die max. Ausrüstung mit 256k-DRAMs werden aber die neun Multiplex-Adress-Leitungen ( A0 - A8 / A9 - A17 ) benötigt. Für den Einsatz von 64k-DRAMs werden jedoch nur acht benötigt, die neunte Ltg. würde frei bleiben.

Bei linearer Multiplex-Adressierung, d.h. A0 - A8 / A9 -A17, würden sich also bei eingesetzten 64k-DRAMs die o.g. "Adress-Lücken" bei der durch den GDC verlangten linearen Adressraum ergeben.

Eine Lösung des Problems ist, daß die DRAM-Adressleitung A8, welche nur beim 256k-DRAM vorkommt, mit A16 (RAS) bzw. A17 (CAS) belegt wird. Damit ist die lineare GDC-Adressierung auch mit 64k-DRAM's möglich. Die Ablage-Reihenfolge im DRAM ist dabei völlig unerheblich ( immer unter Beachtung der Refresh-Bedingungen ).

Im Ergebnis dieser Vorbetrachtungen ergibt sich nunmehr folgende Multiplexer-Adress-Zuordnung:

*muß hardwaremäßig noch geändert werden!*

<u>GDC</u>	<u>RAS</u>	<u>CAS</u>
A(D)0		A0
A(D)1	A0	
A(D)2	A1	
A(D)3	A2	
A(D)4	A3	
A(D)5	A4	
A(D)6	A5	
A(D)7	A6	
A(D)8	A7	
A(D)9		A1
A(D)10		A2
A(D)11		A3
A(D)12		A4
A(D)13		A5
A(D)14		A6
A(D)15		A7
A16	A8	
A17		A8

Ein weiteres Problem muß noch gelöst werden.

Mit der hier eingesetzten 8283-Bustreiber-Multiplexschaltung lassen sich nur acht Multiplex-Adress-Leitungen ( z.B. A1-A8 ) für die DRAM's erzeugen.

Die noch fehlende DRAM-Adress-Leitung, hier also A0, wird deshalb mit der Wired-OR-Gatterschaltung ( 14.1/14.2 auf der RAM-LP) realisiert.

*(Bestand, Hardware muß noch geändert werden!)*

Für den Zugriff mit /RAS werden damit wie o.g. die GDC-Adresse A1 durchgeschaltet, für /CAS-Zugriff hingegen GDC-Adresse A0.

Außerdem besteht es noch ein weiterer Grund diese speziellen Schaltungsanordnung ausgerechnet für A0/A1 vorzusehen und nicht z.B.

für A8/A17 der GDC-Adressen. Dieser ist wegen des optional zu betreibenden "Wide-Display-Betrieb" notwendig und wird schaltungstechnisch weiter unten erläutert.

Als nächstes soll das Timing-Problem der Ansteuerung der Bustreiber-Multiplexer, der RAS/CAS-Steuerung, sowie des Video-Schieberegisters erläutert werden.

Dazu siehe das Taktablauf-Diagramm.

Dreh- und Angelpunkt des Timing-Problems des RAM-Interface ist das GDC-Signal ALE ( Ltg. 134 ). Wie schon o.g. wird dieses mit seiner L/H-Flanke ca. 30 - 120ns nach einer L/H-Flanke der GDC-Taktes 2x WCLK ( Ltg. 135 ) erzeugt und schaltet nach ca. 30 - 100ns nach der folgenden H/L-Flanke des GDC-Taktes wieder zurück.

**(Achtung:**  
*nachfolgender Absatz ist noch Bestand -> neue RAS-MUX-CAS-Steuerung;*  
*Hardware muß auch noch geändert werden!)*

Da die zulässigen TTL-Lasten an den Ausgängen des GDC (MOS-Schaltkreis), hier ALE, auf wenige mA (= 1x TTL-Last) begrenzt ist, wird zunächst ein als nichtinvertierender Treiber geschalteten AND-Gatter 5.4 nachgesetzt.

Damit erhöht sich die Laufzeit von ALE (Ltg. 214) um weitere 20ns (> 50 -140ns nach L/H-Flanke GDC-Takt). Da die 16x /RAS-Eingänge der DRAM's eine große kapazitive Last darstellen und mit den relativ langen Zuleitungen bis auf die RAM-Lp die Gefahr von durch Leitungsreflektion bedingten Störeinflüsse undefinierte Timing-Zustände möglich sind, wird auf der RAM-Lp weitere AND-Gatter 13.1 als Treiber für /RAS (Ltg. 224), sowie 13.2 für /CAS (Ltg. 228) vorgesehen.

Mit den Gattern 2.5 und 2.6 wird das Signal doppelt invertiert. Durch die zusätzliche kapazitive Belastung von 2.6 (siehe auch *Taktsignalerzeugung*) wird von dem ALE-Signal dadurch mit einer Laufzeit von ca. 60ns das Strobe-Signal ( Ltg. 179 ) für die als Adress-Multiplexer geschalteten Bustreiber-D-Latches  $RG_L$  MUX1 und  $RG_H$  MUX2 ( 8283 ) gebildet. D.h. nach weiteren ca. 60ns nach der L/H-Flanke von ALE werden sowohl MUX1 als auch MUX2 vorerst in den durchgängigen Treibermodus geschaltet werden ( IN = OUT ). Da das STB-Signal ( Ltg. 179 ) gleichzeitig das /OE-Signal für MUX2 (Ltg. 175) ist, wird jedoch  $RG_H$  in den TriState-Zustand versetzt, d.h. nur der L-Teil der Adresse ( A1 - A8,A16 ) liegt ca. 100ns nach Adress-Ausgabe des GDC als DRAM-Adresse an. Damit hat der Adressbus genügend Zeit stabile Pegel anzunehmen.

Jetzt fehlt noch die DRAM-Adresse A0.

Zunächst müssen die verbleibenden Adressen A0, A1 noch zwischengespeichert werden. Dazu wird das D-FF 12.1/2 vorgesehen. Da diese im Gegensatz zu MUX1/MUX2 mit einer L/H-Flanke nach dem Zeitpunkt des Vorliegens stabiler GDC-Adressen zu takten sind, wurde nicht mit MUX1 getaktet, sondern mittels der Gatterschaltung 2.2, 5.3 eine um ca. 280ns zu ALE verzögerte L/H-Flanke erzeugt. (siehe dazu Taktdiagramm)

Die zwischengespeicherten Adresslinien A0/A1 (Ltg. 249/250) werden zu der auf der RAM-Lp liegenden Wired-OR-Gatterschaltung 14.1/2 geschaltet, welche dort mit MUX1/MUX2 gemultiplext werden.

Die DRAM-Adresse A0 ist dann Ltg 177. Wegen der hohen Anforderung an kurze Schaltzeiten ist die +UB-Anbindung mit R1 ist so niederohmig wie möglich (330 Ohm) gestaltet.

Das /CAS-Signal wird mit dem D-FF 6.1 aus dem MUX1-Signal abgeleitet. Mit dem durch Gatter 2.4 invertiertem 1xWCLK (Ltg. 213) wird FF 6.1 getaktet, so daß ca. 100ns nach /RAS und MUX1,

Anliegen von H-RAM-Adresse, diese in die DRAM's übernommen wird. Nach der Zugriffszeit der DRAM's von ca. 120-150ns liegen nunmehr die Daten an deren Ausgänge solange an, bis /CAS wieder auf HIGH-Potential geht. Da die nächste L/H-Flanke des Taktsignals 1xWCLK zu spät kommt, wird dies zwangsweise vorzeitig durch H-Setzen von FF 6.1 mit MUX2 vorgenommen, wenn ein neuer BWS- bzw. auch RMW-Zyklus eingeleitet wird.

Die BWS-Daten werden mit einem kurzen Implus in die

(... und hier endet abrupt der Entwicklungsbericht – leider!

Zu den weiteren GDC-Baugruppen gibt es keine Beschreibungen – Berichte ... dazu ist es in der Bearbeitung leider nicht mehr gekommen ...)

#### **3.5.1.5. Softscroll**

#### **3.5.1.6. Zeichengenerator**

#### **3.5.1.7. Video-Signal-Erzeugung**

#### **3.5.1.8. Das Zoom-Ink-Register**

#### **3.5.1.9. Das CTL-Register**

#### **3.5.1.10. Palettenspeicher, -Pointer und -Register**

### **3.5.2. GDC-Software – Grundroutinen**

Es existieren bereits einige Grundroutinen für den EXPR, insbesondere für die Initialisierung und weiterer Hardware- und Programmentwicklung für den GDC u.a. StE.

Host-Rechner war: KC85/3 am K1520-Port -> Siehe "BS\_EXPR1.txt".

## 3.7. I/O-Baugruppe

### 3.7.1. Funktionsumfang

Die I/O-Baugruppe (StE) soll für folgende Funktionen realisieren:

- Tastatur-Interface,
- V24,
- Centronics-Schnittstelle,
- EPROMMER (??),
- ggf. auch noch ein Dualport-RAM (wie im KC85-Entwicklungssystem)

### 3.7.2. Bauelemente-Basis

Folgende System-Schaltkreise sollen zum Einsatz kommen:

I/O-Adressen: (Basisadresse = 2xH, d.h. A5 = H, A6 = L, A7 = L)

- PIO UA855D (2x)	28H	29H	2AH	2BH
	2CH	2DH	2EH	2FH
- SIO UA856D	24H	25H	26H	27H
- CTC UA857D	20H	21H	22H	23H
- 8k SRAM U6264D	30H	31H	32H	33H

### 3.7.3. Realisierte Basisfunktionen

Tastatur-Interface Interrupt-gesteuert (höchste Priorität), SIO – Kanal B

#### 3.7.3.1. Tastatur-Takterzeugung

wird über den CTC, Kanal 2 realisiert – mit DFFH = f/2

Kanal 1: Tastaturempfangstakt

Kanal 0: V24 Sende-/Empfangstakt

-> 1. Versuch:  $f_T = (???)$

(... hier enden die Notizen ...)