

Inbetriebnahme “GDC“

des Modularen Experimental-Computers

1. [Einleitung und Systemvoraussetzungen](#)
2. [Prüfung der exakten Adressierung](#)
 - 2.1. [Gruppendresse](#)
 - 2.2. [Einzeladressen](#)
3. [I/O-Datenbus Transfer](#)
 - 3.1. [Einzeladressen I/O](#)
 - 3.1.1. [OUT GDC](#)
 - 3.1.2. [IN GDC](#)
4. [Laden mit “BS-EXPR“](#)
5. [Test Synchronisation](#)
6. [Kontrolle der abgeleiteten Signale](#)
 - 6.1. [Programmierung der Bild-Synchron-Parameter](#)
 - 6.2. [Berechnung der PARATAB-Werte](#)
 - 6.2.1. [Betriebsstrukturbyte](#)
 - 6.2.2. [Worte / Zeile](#)
 - 6.2.3. [Der Horizontal-Synchronimpuls \$HSYN\$](#)
 - 6.2.4. [Die hintere vertikale Schwarzwertschulter \$hvSS\$](#)
 - 6.2.5. [Die hintere horizontale Schwarzwertschulter \$hhSS\$](#)
 - 6.2.6. [Gesamt-Zeilendauer](#)
 - 6.2.7. [Zeilen / Bild](#)
 - 6.2.8. [Der Vertikal-Synchronimpuls \$vsyn\$](#)
 - 6.2.9. [Die vertikale vordere und hintere Schwarzsulter \$vvSS, vhSS\$](#)
 - 6.3. [PARATAB laden](#)
 - 6.4. [Kontrolle der abgeleiteten Signale](#)
7. [Neue \$MUX\$ / \$stb\$ – Steuerung](#)
8. [Versuch mit Monitor](#)
 - 8.1. [PARATAB neu](#)

Anhang - Tabellen

© Copyright by Peter Salomon, Berlin – Januar 2019/2020

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.

1. Einleitung und Systemvoraussetzungen

Diese Dokumentation ist aus ca. 20 A4-Seiten handschriftlicher Aufzeichnungen (Bleistift) rekonstruiert worden. Aus den zunächst etwas verwirrend erscheinenden Aufzeichnungen kann ermessen werden, wie kompliziert das richtige Schaltungs-Design rund um den GDC U82720D werden kann, wenn alle Funktions-Features ausgeschöpft werden sollen.

Try & Error – Versuch und Irrtum – war der ständige Begleiter bei der zeitaufwändigen Entwicklung.

Und – um es gleich vornweg zu nehmen – leider ist des ganze Projekt nicht bis zur vollständigen, durch den GDC-IC vorgegeben möglichen Funktionalität fertig geworden ...

Wie aus der Hardware-Beschreibung zu ersehen ist, fehlt noch die Bearbeitung von Softscroll, dem Zeichengenerator, der endgültige Video-Signal-Erzeugung, dem Zoom-Ink-Register, dem CTL-Register, dem Palettenspeicher, -Pointer und -Register, sowie die DRAM-Speichererweiterung auf mehrere Ebenen für eine Farb-Bild-Wiedergabe.

Zum Hardwaretest und Entwicklung der GDC-Grundroutinen wird der Betrieb an der K1520-Bus-Schnittstelle des SEW (KC85/3-Zusatzgerät) vorgesehen.

Dazu sind folgende Parameter einzustellen:

- Adresse 24,
- Steuerwort 45.

Zur 5V-Stromversorgung der GDC-Baugruppe sind die internen Ressourcen des SEW nicht ausreichend. Deshalb wird dafür eine externe Netzteil-Baugruppe von Robotron vorgesehen.

2. Prüfung der exakten Adressierung

2.1. Gruppendresse

Die Gruppenadresse der GDC-Baugruppe ist: 0CXH

DIL-Schalter: HHLL

2.2. Einzeladressen

Die Einzeladressen der GDC-Register sind:	0/1	(GDC)
	2/3	-
	4/5	CTR
	6/7	ZIR
	8/9	PAR
	A/B	PPT
	C/D	PSP
	E/F	ASCI

3. I/O-Datenbus Transfer

3.1. Einzeladressen I/O

3.1.1. OUT GDC

Für die einfache Überprüfung des Datenverkehrs auf dem I/O-Bus zum GDC wird folgende Code-Sequenz verwendet (KC85/3-spezifisch):

```
                DEFW 7F7FH
                DEFM "OUT"
                DEFB 1
Loop:           LD A, 0           ; "RESET" GDC
                OUT 0C1H, A      ; Befehlsdaresse
                CALL 0F003H
                DEFB 02AH       ; Abfrage "BREAK"-Taste
                JRC Ende
                JR Loop
Ende:           RET
```

Nachweis der Pegel "0" auf dem GDC-Datenbus: ok

Bei gleichzeitiger Adressierung

A0 = H
/WR = L
/RD = H

3.1.2 IN GDC

Für die Überprüfung des des Datenverkehrs auf dem I/O-Bus vom GDC wird folgende Code-Sequenz verwendet (KC85/3-spezifisch):

```
                DEFW 7F7FH
                DEFM "IN"
                DEFB 1
Loop:           IN A, 0C0H           ; "Status-Register" GDC
                CALL 0F003H
                DEFB 0             ; Ausgabe Bildschirm
                CALL 0F003H
                DEFB 02AH         ; Abfrage "BREAK"-Taste
                JRC Ende
                JR Loop
Ende:           RET
```

Nachweis der Pegel auf dem GDC-Datenbus: L | H | L | L | L | H | L | L (D)

 L | H | H | L | L | H | L | L (d)

Bei gleichzeitiger Adressierung A0 = 0
 /WR = H
 /RD = L

4. Laden mit "BS-EXPR"

Das Programm wird mit ORG: 3000H geladen.

Ein schrittweiser Test kann mit dem KC-Programm "TEMO" vorgenommen werden.

Es wurde sofort ein Fehler festgestellt, der dann auch gleich beseitigt wurde

(-> Rand-Ergänzungen im Quelltext).

Danach mußten einige Code-Ergänzungen nach "Weiter" eingefügt werden, weil sonst keine

- Synchronimpuls-Ausgabe erfolgt,

- der Bildschirm dunkel getastet ist, d.h. BLANK = H.

Notwendig sind die Befehle:

- VSYNC-Angabe,
- START,
- BCTRL ("BS Ein"),

d.h. ab 035H bis 050H

5. Test Synchronisation

Dazu muß vorher PARATAB mit den dazu notwendigen Werten geladen werden.

Die Kontrolle wird mit dem Oszi EO213 durch Ausmessen vorgenommen:

- Abstand Zeilen-Synchronimpulse,
- Abstand Bild-Synchronimpulse,
- Synchronimpuls-Breite Zeilen- und Bildsynchronimpulse,
- Breite vordere und hintere Schwarz-Schulter (v_{hSS} und h_{hSS}).

Eine Kontrolle mit dem Monitor (PC1715-Monitor, oder BWG 1.0) wäre zwar möglich, aber es ist mangels Bild-Inhalt nichts zu sehen.

6. Kontrolle der abgeleiteten Signale

Mit dem Oszi können auch die aus dem Taktschema abgeleiteten Signale kontrolliert werden.

Es geht dabei um:

- ALE
- /RAS ,
- /CAS ,
- STB ,
- MUX1 ,
- MUX2 ,
- RG ,
- E/O ,
- RCT ,
- SRLOAD ,
- WR .

Wenn PARATAB mit den notwendigen (ursprünglich aus Literaturangaben) Werten geladen wurde, oder auch mit den GKS-Werten, ergaben sich nicht zu definierende Ergebnisse – vor allem das ALE-Signal war diskontinuierlich.

Als nächsten Schritt zur Eingrenzung des Fehlers wurde folgendes vorgenommen:

- alle Werte außer 12H, ... FFH werden auf 0 gesetzt.
- 2BH (oder 26) wird zur besseren Auszählbarkeit auf 0FH gesetzt, was 16 Worte / Zeile bedeutet.

In der Realität sind es aber 18 Worte / Zeile, d.h. immer +2!

H-Synchr. im Minimum = 1 Wort, beginnend mit der steigenden Flanke von ALE, eine Zeile (18 W) zählt von fallender Flanke bis steigender Flanke von H-Synchr.

Nächster Schritt:

- H-Synchr auf 3 gesetzt sollte einen H-Synchronimpuls von $2\mu\text{s}$ ergeben, dazu
- $v_{hSS} = +1W$ (eingetragen 0!)
- $h_{hSS} = +1W$

Ergebnis war ein $v_{hSS} = 3\mu\text{s}$ bei einem ALE-Impuls $150\text{ns}/350\text{ns}$.

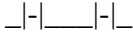
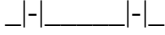
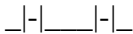
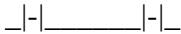

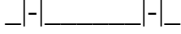
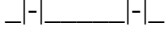

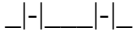
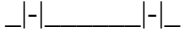
Der nächste Versuch mit H-Synchr auf 0 gesetzt ergibt H-Synchr = $1W = 0,5\mu\text{s}$.

Worte / Zeile auf 0 gesetzt -> geht nicht. – Im Minimum ist $4W$ ($2\mu\text{s}$) zulässig und dann müssen immer gerade Zahlen (4, 6, 8 usw.) eingesetzt werden.

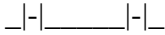
Beim H-Synchr-Wert sind alle Zahlen zulässig.

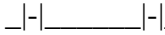
6.1. Programmierung der Bild-Synchron-Parameter

Da die Programmierung der Bild-Synchron-Parameter des GDC recht kompliziert und auf den ersten Blick sehr undurchsichtig ist, wurde auf Basis dieser Grenzwert-Untersuchungen einige Parameter-Beispiele versucht:

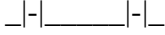
(1)	(Vorr.: alles außer 12H auf 0)	Zeilensynchronimpuls	
			0,5 2 0,5µs
		Bildsynchronimpuls	
			2,6 7,7 2,6ms
(2)	Zeichen / Zeile: 01	Zeilensynchronimpuls	keine Änderung!
		Bildsynchronimpuls	keine Änderung
(3)	Zeichen / Zeile: 02	Zeilensynchronimpuls	
			0,5 3 0,5µs
		Bildsynchronimpuls	
			3,6 10,8 3,6ms
(4)	Zeichen / Zeile: 0FH (16)	Zeilensynchronimpuls	
			0,5 9 0,5µs
		Bildsynchronimpuls	
			9,8 29,4 3,6ms
(5)	Zeichen / Zeile: FFH (255)	Zeilensynchronimpuls	
			0,5 129 0,5µs
		Bildsynchronimpuls	
			133,3 392,2 133,3ms
(6)	Zeichen / Zeile: 0FH	Zeilensynchronimpuls	
	zSynchr: 01		
		Bildsynchronimpuls	
			10,3 30,8 10,3ms

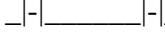
(7) Zeichen / Zeile: 0FH
 zSynchr: 1FH
 (d.h. B0 bis B4 gesetzt = 31)

Zeilensynchronimpuls 
 16 9 16μs

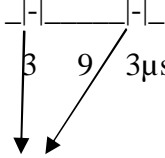
Bildsynchronimpuls 
 25,6 77 25,6ms


(8) Zeichen / Zeile: 0FH
 zSynchr: 5

Zeilensynchronimpuls 
 3 9 3μs

Bildsynchronimpuls 
 12,2 37 12,2ms


(9) wie (8) aber zusätzlich
 vSynchr: 1
 (d.h. B5 gesetzt, B6, B7 = 0) = 25

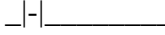
Zeilensynchronimpuls 
 3 9 3μs

Bildsynchronimpuls 
 12,2μs! 37ms 12,2μs


Das funktioniert nur bei vSynch = 1, d.h. min. B5 gesetzt!

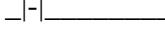
(10) wie (8) aber
 vSynchr: 14
 (d.h. B5, B6, B7 gesetzt) = E5


Zeilensynchronimpuls 
 3 9 3μs

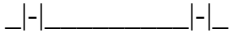
Bildsynchronimpuls 
 82μs 37ms 82μs

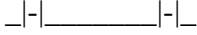
(11) wie (10) aber zusätzlich
 Zeilen / Bild: 1

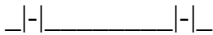
Zeilensynchronimpuls 
 3 9 3μs

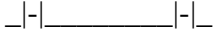
Bildsynchronimpuls 
 82μs 24,7ms 82μs

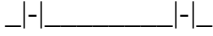
(das kann nicht stimmen!) 

(12) wie (11) aber
 Zeilen / Bild: FFH (255) Bildsynchronimpuls 
 82μs 27,7ms 82μs

(13) wie (11) aber vhSS: 1 d.h. B2 gesetzt -> 04
 Bildsynchronimpuls 
 82μs 12,3ms 82μs

(14) wie (13) aber vvSS: 1 d.h. B0 gesetzt -> 01
 Bildsynchronimpuls 
 82μs 36μs! 82μs
 vvSS = 1, d.h. min. B0 gesetzt

(15) wie (14) aber hhSS: 1 d.h. B0 gesetzt
 Bildsynchronimpuls 
 37,4μs 86,8μs! 37,4μs

(16) wie (14) aber hvSS: 1 d.h. B2 gesetzt
 Bildsynchronimpuls 
 39,1μs 86,8μs! 39,1μs

=> Resüme: vvSS und hvSS min. 1! – sonst arbeitet der Synchron-Generator nicht richtig.

6.2. Berechnung der PARATAB-Werte

6.2.1. Betriebsstrukturbyte

Das Betriebsstrukturbyte ist auf 12H festgelegt:

B0 = 0	(kein Zeilensprung)
B1 = 1	(Grafik-Betriebsart)
B2 = 0	(kein Refresh, noch unklar!)
B3 = 0	(ZSp, WH-Feld ?)
B4 = 1	(Zeichnen nur bei Dunkelast.)
B5 = 0	(Zeichen-Betriebart)
B6/7 = 0	

6.2.2. Worte / Zeile

Ausgangspunkt ist folgende Überlegung (unter Berücksichtigung von Monitor-Betriebsdaten):

- 80 Zeichen aktiv pro Zeile (1 Zeichen = 8 Bit = 1 Byte)
- 80 Zeichen sind 40 Worte / Zeile – 2 = 38 (26H),
d.h. $38 \times 0,5\mu\text{s} = 19\mu\text{s} = t_{\text{Hak}}$

Die Länge des Horizontal-Synchronimpulses ist im Wert für Worte / Zeile enthalten.

6.2.3. Der Horizontal-Synchronimpuls t_{HSYN}

Der Horizontal-Synchronimpuls t_{HSYN} soll ca. 4 bis 5 μs betragen. Damit wird:

$$\begin{array}{rcl} t_{\text{HSYN}} = 5\mu\text{s} / 0,5\mu\text{s} = 10 & & \begin{array}{l} \text{B0} = 0 \text{)} \\ \text{B1} = 1 \text{)} \\ \text{B2} = 0 \text{)} = 0\text{AH (10dez.)} \\ \text{B3} = 1 \text{)} \\ \text{B4} = 0 \text{)} \\ \hline \text{VSYN(LB)} \text{ B5} = ? \text{)} \\ \text{- " -} \text{ B6} = ? \text{) noch offen, wird weiter unten} \\ \text{- " -} \text{ B7} = ? \text{) bestimmt} \end{array} \end{array}$$

6.2.4. Die horizontal vordere Schwarzwertschulter t_{hvSS}

Die horizontal vordere Schwarzwertschulter t_{hvSS} soll ca. 3 μs betragen. Damit wird:

$$\begin{array}{rcl} t_{\text{hvSS}} = 3\mu\text{s} / 0,5\mu\text{s} = 6 & & \begin{array}{l} \text{B0} = ? \text{) VSYN}_{(\text{HB})}, \text{ noch offen,} \\ \text{B1} = ? \text{) wird weiter unten bestimmt} \\ \text{B2} = 0 \text{) Fehler?! } \neq 6 \\ \text{B3} = 1 \text{)} \\ \text{B4} = 1 \text{)} \\ \text{B5} = 0 \text{)} \\ \text{B6} = 0 \text{)} \end{array} \end{array}$$

6.2.5. Die horizontale hintere Schwarzwertschulter t_{hhSS}

Die horizontale hintere Schwarzwertschulter t_{hhSS} soll ca. 5 μs betragen. Damit wird:

$$\begin{array}{rcl} t_{\text{hhSS}} = 5\mu\text{s} / 0,5\mu\text{s} = 10 & & \begin{array}{l} \text{B0} = 0 \text{)} = 0\text{AH} \\ \text{B1} = 1 \text{)} \\ \text{B2} = 0 \text{)} \\ \text{B3} = 1 \text{)} \\ \text{B4} = 0 \text{)} \\ \text{B5} = 0 \text{)} \\ \text{B6} = 0 \text{) (fest)} \\ \text{B7} = 0 \text{) " - " } \end{array} \end{array}$$

6.2.6. Gesamt-Zeilendauer

Die Gesamt-Zeilendauer errechnet sich zu: $t_{\text{Hges}} = t_{\text{Hak}} + t_{\text{HSYN}} + t_{\text{hvSS}} + t_{\text{hhSS}}$

$$= 19\mu\text{s} + 5\mu\text{s} + 3\mu\text{s} + 5\mu\text{s} = 32\mu\text{s} = t_{\text{Hges}} \text{ oder } 31\text{kHz}$$

6.2.7. Zeilen / Bild

Der Einfachheit halber werden zunächst 255 Zeilen pro Bild festgelegt, d.h. 0FFH,

bzw. $255 \times 32 \mu\text{s} = 8,16\text{ms} = t_{vak}$

Wie unter 3.5.1.4. der Hardware-Beschreibung aufgeführt, sollten eigentlich 512 Zeilen / Bild realisiert werden.

6.2.8. Der Vertikal-Synchronimpuls v_{SYN}

Der Vertikal-Synchronimpuls v_{SYN} soll ca. 1ms betragen. Damit errechnet sich:

$$t_{v_{SYN}} = 1000\mu\text{s} / 32\mu\text{s} = 31, \text{ oder } 1FH \quad \begin{array}{l} B5 = 1) \\ B6 = 1) \\ B7 = 1) \\ B0 = 1) \\ B1 = 1) \end{array} \text{ d.h. max. möglich}$$

6.2.9. Die vertikale vordere und hintere Schwarzschulter v_{VSS} , v_{HSS}

Die vertikale vordere Schwarzschulter v_{VSS} soll ebenfalls ca. 1ms sein.

Daraus ergibt sich ebenfalls für:

$$t_{v_{VSS}} = 1000\mu\text{s} / 32\mu\text{s} = 31, \text{ oder } 1FH.$$

Die vertikale hintere Schwarzschulter v_{HSS} soll ca. 2ms betragen. Damit wird:

$$t_{v_{HSS}} = 2000\mu\text{s} / 32\mu\text{s} = 62.$$

6.2.10. Die Verhältnisse $t_{v_{SYN}} / t_{h_{SYN}}$ und $t_{v_{SYN}} / t_{h_{VSS}}$

Aus (8) und (3) ergibt sich EAH, bzw. 1BH.

Die Gesamt-Bilddauer kann somit errechnet werden:

$$t_{v_{Ges}} = t_{vak} + t_{v_{SYN}} + t_{v_{VSS}} + t_{v_{HSS}} = 8,16 + 1 + 1 + 2 = 11,16\text{ms, oder } 90\text{Hz}$$

Zusammenfassung: 12H; 26H; EAH; 1BH; 0AH; 1FH; FFH; F8H
(Korrekturen) E8H F9H

6.3. PARATAB laden

Als nächstes wird das Tabellen-Array **PARATAB** mit den berechneten Werten geladen.

Die sich damit und der GDC-Hardware ergebenden Werte mit dem Oszi vergleichen:

	berechnete Werte	gemessenen Werte
t_{Hak}	19 μ s	20,1 μ s
t_{Hges} (VON HSYN ZU HSYN)	32 μ s	34,5 μ s
t_{vak}	8,16 μ s	8,8 μ s
t_{Vges} (VON VSYN ZU VSYN)	11,16ms	13,1ms

BLANK-Signal im Bereich t_{Hak} und t_{vak} kontrollieren: o.k.

Dann mit dem Videosignal (S-Reg) verbinden -> da gibt's Leitungs-Probleme ...

Zum Vergleich werden die Monitor-Signale vom PC1715, bzw. KC85/3 geprüft (mit dem Oszi ausmessen) -> mit dem Ergebnis, dass die vom PC1715 nicht passen!

Das BLANK-Signal könnte zwar als Konsequenz mit /OE des S-Reg verbunden werden – ist aber falsch!

In den BLANK-Zeiten liegt dann H-Pegel auf Video, was ein Helltasten bedeuten würde. Besser ist es mit CLEAR – dann ergibt sich ein L-Pegel, der muß dann aber mit /BLANK verknüpft werden.

6.4. Kontrolle der abgeleiteten Signale

/RAS	o.k., evtl. Treiber 5.3 entfernen, weil H/L mit MUX zusammenfällt, was vielleicht trotzdem geht wegen Δt des Adresstreibers
/CAS	nicht o.k.! – es fehlt der 2.Zugriff -> siehe E/O
STB	o.k.
MUX1	o.k., evtl. zusätzliches Δt mit 5.3 (? - siehe auch SLOAD)
MUX2	- “ -
RG	noch zu prüfen!
E/O	nicht o.k.! – FF 6.2 arbeitet nicht!
RCT	nicht o.k. – zu kurz! -> siehe MUX
SLOAD	- “ -
WR	noch zu prüfen!

Für die restlichen Steuersignale noch Clips setzen und anzeichnen! – (wie /RAS und /CAS)

- Problematik /CAS: 6.2 schaltet manchmal nicht, weil Anfangsbedingung (beim Einschalten) fehlt, d.h. es ist mit $C = 100\text{pF}$, $R = 4,7\text{k}\Omega$ am IC eine Anfangsbedingung zu schaffen.
C an /CAS: mit der ersten H/L-Flanke wird 6.2 gesetzt, d.h. $E/O = H$
- Problematik /RAS - /MUX: evtl. zwischen 2.5 und 2.6 mit $C = 100\text{pF}$ gegen Masse oder 5P eine Verlängerung um 10 bis 20ns erreichen, oder zusätzlich 5.3 ??
- Schaltfehler: an 1.4/9 liegt Ltg. 176 (MUX2), nach Impulsdigramm soll aber MUX1 anliegen -> ändern, auch im Schaltbild
- Designfehler: Funktion MUX (12.1/2 – 9.1/2) immer noch unklar!
- AD0 im Wide-Betrieb immer "L" (nur gerade Adressen!)
-> noch mal nachlesen!
 - Ltg.176 / 178 an 9.1 / 9.2 vertauschen, weil das Umschalten der LOW-Adresse mit 9.1 erfolgt, aber wenn AD0 immer LOW, dann 12.1 weglassen? – 1.4 / 9.1 – Funktion prüfen!
 - E/O-Generierung funktioniert nicht! ->
/CAS-Generierung ist nach Taktdiagramm falsch! – weil
-> RG zu falschen Zeitpunkt gebildet, da bei $1 \times \text{WC1k}$ L/H-Bereich vertauscht ist (Zähler-Ausgang invertiert)
-> 4.3 weglassen – war schon raus, also wieder rein ??
-> RG wird in Abhängigkeit von ALE zufällig erzeugt, da der interne Zähler im GDC mal auf der 1. oder 2. L/H-Flanke ALE erzeugt ...
-> WC1k muß in Abhängigkeit vom Zustand ALE erzeugt werden (nicht "parallel" !)
 - ALE-Signal nach 5.4 und 2.6/12 (invertiert!) mit $C = 100\text{pF}$ (?) am Lade-Eingang von CT2/11, Zähler-Eingänge A, B, C, D alle auf H (der Zähler zählt rückwärts !); damit wird WC1k , $2 \times \text{WC1k}$, $4 \times \text{WC1k}$ zwangsweise auf H gesetzt, wenn ALE auch H ist.
-> das funktioniert leider so nicht

-> neue Variante: U_R von CT2 auf Masse, A, B auf 5P und C mit NOR (002, vom 1.4 ?) aus $4 \times WClk$ und MUX2 ??? ...

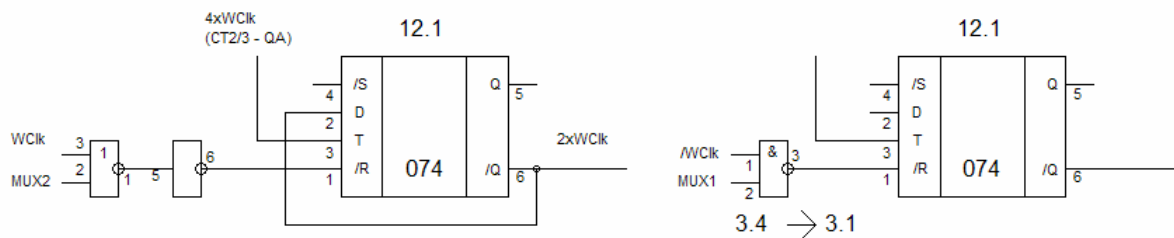
Nein! – Das Problem ist anders:

- Es ist notwendig den internen Zähler des GDC mit CT2 so zu synchronisieren, daß sich der vorgegebene Impulsablauf ergibt.

Wie soll das gemacht werden?

- Da 12.1 offensichtlich nicht mehr gebraucht wird, kann das FF als "Parallelteiler" zu CT2 ($4 \times WClk \rightarrow 2 \times WClk$) eingesetzt werden.

(siehe S13, aber Gatter 9.1 bleibt bestehen! – Anschluß 2 -> H



Es wird die rechte Variante verwendet, da durch neue St_B -Generierung (siehe weiter unten) Gatter 3.1 frei wird.

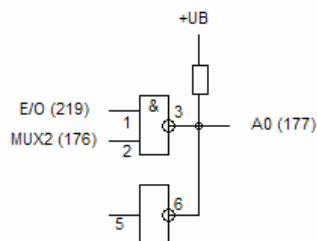
Es gibt einen weiteren Design-Fehler!

- Da MUX – A8 nur bei MUX1 = H freigegeben werden soll, muß an 9.2/5 -> Ltg.175.

In anderen Zeiten ist MUX1 = L und damit das Wired-OR-Teil 9.2 = H,

d.h. der Zustand ist nunmehr nur noch von 9.1 abhängig.

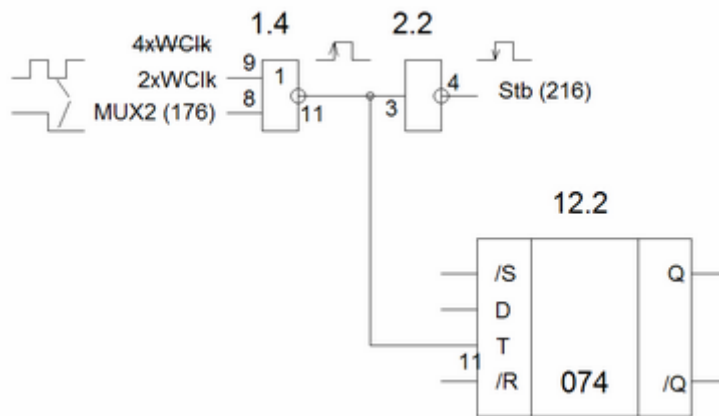
Dazu an 9.1 folgende Anschlüsse, d.h. Gatter 1.4 entfällt:



Dafür gibt es eine neue Design-Variante zur Bildung von St_B .

- Da die 8-fach-Register 8283 mit der H/L-Flanke speichern, das FF 12.2 (für A8) jedoch mit der L/H-Flanke, wäre die Zeit zu kurz, bis mit ALE -H/L eine gültige Adresse übernommen werden kann.

Daher neues Design:



Gatter 3.1 entfällt, bzw. kann anstelle 3.4 eingesetzt werden.

Impulsüberprüfung

-> 2xWC1k (GDC) – Bildung (Synchronisierung): o.k.

-> Stb – Generierung nicht o.k.! -> noch mal Design-Änderung??

Nein! -

1.4/9 wird anstelle 4xWC1k (CT2/3) an

2xWC1k (CT2/2) gelegt,

damit wird die L/H-Flanke ca. 100ns nach L/H-Flanke ALE.

-> Stb: ca. 40ns vor H/L von ALE

ca. 60ns vor H/L von RAS

aber ca. 50ns nach MUX2 (DRAM-High-Adresse des GDC),

d.h. ins DRAM würden ungültige Adressen gegeben!

-> Stb muß für die Zeit kurz vor MUX1 bis kurz nach der 2. H/L-Flanke CAS

(ungerade Adresse) LOW sein!

Sonst bleibt die Adresse nicht gespeichert und es werden ungültige Adressen

durchgeschaltet (Treiber-Modus des 8283).

Außerdem ist die Zuordnung MUX1 / MUX2 zu den Adressen falsch!

Folgender Ablauf ist zu realisieren:

1. kurz nach L/H von ALE ist die GDC-RAM-Adresse gültig, d.h. sie ist mit Stb (H/L) in MUX_H / MUX_L zu speichern. Stb muß dabei LOW bleiben! -
2. kurz vor H/L von ALE ist mit MUX_H die High-Adresse zum DRAM zu schalten, damit mit H/L von RAS die Row-Adresse im DRAM gespeichert werden kann. -

3. dann ist mit MUX_L die Low-Adresse an den DRAM zu legen, damit danach mit H/L von CAS die *gerade* DRAM-Adresse angesprochen werden kann.
4. mit E/O ist danach zwangsweise A0 vom DRAM auf H zu legen, bzw. zu invertieren, damit mit dem 2. CAS-Zugriff die *ungerade* Adresse angesprochen werden kann.

Fazit: -> Das stb -Signal für die MUX muß neu designed und $MUX1/MUX2$ richtig zugeordnet werden!

7. Neue MUX / stb – Steuerung

Als Ergebnis der o.g. Überlegungen ist eine neue stb-Steuerung entworfen worden.

- /OE – Freigabe von MUX_H : an MUX2 (Ltg.176),
- /OE - Freigabe von MUX_L : an MUX1 (Ltg. 175),
- stb für beide : mit dem MUX1-Signal (Ltg.216 entfällt!)
 - MUX1 = H -> Treiber-Modus des 8283,
mit H/L-Flanke Speicherung der H-Adresse
und dann Low-Adresse,
wobei H-Adresse weiterhin aktiv bis Wechsel
zur Low-Adresse
- stb für 12.1 (FF) wird weiterhin wie gehabt mit NOR 1.3 gebildet.
Damit wird die Adresse gehalten für ca. 100ns nach L/H von ALE.
- DBIN (Ltg.133) ist noch nicht verdrahtet,
sowohl auf der GDC-Platine, als auch auf der Speicher-Platte!
- RG (Ltg.517) ist noch nicht verdrahtet!

Die Signale sind jetzt soweit o.k.

8. Versuch mit Monitor

Als nächstes soll der Versuch mit einem Computer-Monitor erfolgen.

Dazu muß das Programm "Init" vom GDC geladen werden, damit definierte Synchron-Signale erzeugt werden können. (siehe 6.2. [Berechnung der PARATAB-Werte](#)).

Das Ergebnis war leider negativ, weil es ein Problem mit dem Taktsignal gibt.

Im Display-Zyklus wird durch das Synchr.-FF die H-Phase des Taktes verlängert.

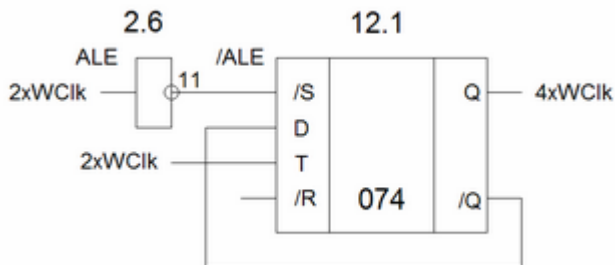
Dadurch stimmen alle (!) PARATAB-Werte nicht, z.B. für den 10H-Wert ergibt sich statt 16 Taktzyklen nur 8!

-> Taktsynchronisation noch mal prüfen!

Es darf nicht der GDC-Takt mit dem $1 \times \text{WC1k}$ -Signal synchronisiert werden, sondern das $1 \times \text{WC1k}$ -Signal für die CAS / E/O-Generierung mit dem ALE-Signal.

Das bedeutet, daß mit CT2 das $1 \times \text{WC1k}$ nicht mehr generiert werden kann, d.h. dafür wird das FF 12.1 verwendet.

Der GDC-Takt $2 \times \text{WC1k}$ wird wieder mit CT2 erzeugt. FF 12.1 wird ebenfalls mit $2 \times \text{WC1k}$ getaktet, aber mit dem ALE-Signal so gesetzt, daß mit jeder H-Phase von ALE $4 \times \text{WC1k}$ ebenfalls H ist.



-> Sp neu zeichnen! (d.h. Änderungen einarbeiten)

-> Hardware scheint jetzt zu gehen, jedoch muß das BLANK-Signal noch eingefügt werden.

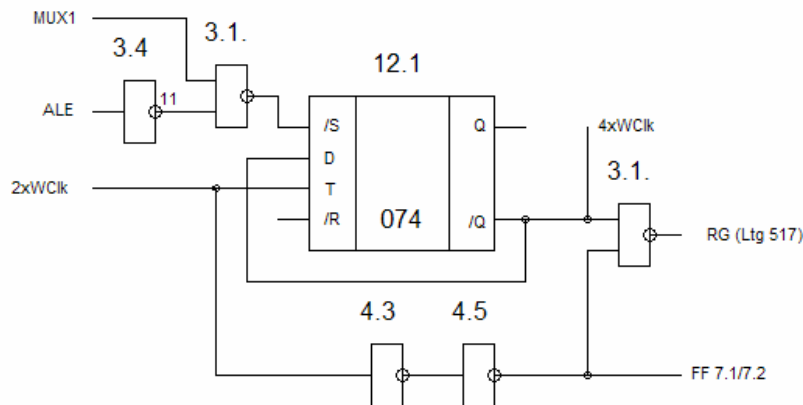
Einige Ergänzungen zur neuen $4 \times \text{WC1k}$ -Erzeugung:

- Da mit der Gesamtlänge von $/ALE$ offensichtlich die nächste L/H-Flanke für die Taktung von 12.1 überschritten wird, muß der Ladeimpuls verkürzt werden. Dazu wird das WC1k -Signal mit dem freien Gatter 3.1 und $/ALE$ verknüpft.

Es zeigte sich jedoch, daß zwar jetzt die CAS / E/O-Signale richtig gebildet werden, aber das RG-Signal einen unerwünschten Spike aufwies.

Dies konnte durch eine kurze Verzögerung des $2 \times \text{WC1k}$ -Signals mit den beiden noch freien Invertoren 4.3 und 4.5 beseitigt werden.

Der Sp-Teil sieht nun wie folgt aus:



Es wird das verzögerte, aber sonst nicht invertierte MUX1-Signal anstelle /ALE verwendet (wegen NAND).

- Das BLANK-Signal (High-aktiv, d.h. in den H-Zeiten keine Bildausgabe!) wird mit dem /OE-Eingang der Schieberegister (DL299D) verbunden.

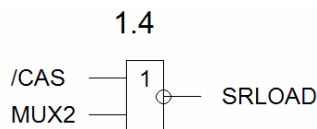
-> Die Schieberegister werden nicht richtig geladen!

Der Ladeimpuls SRLOAD muß L/H sein! (siehe DL299D)

Da Q_A kein TriState-Ausgang ist, kann über /OE das BLANK-Signal nicht einführt werden.

-> neues Design!

Realisierung:



Das ursprünglich vorgesehene MUX2-Signal musste gegen das 4xWClk-Signal (FF) ausgetauscht werden, weil SRLOAD sonst zu kurz wird und die SR nicht laden!

Außerdem muß der Pixeltakt noch phasenverschoben werden, damit die L/H-Flanke in das SRLOAD-Signal fällt, d.h. der Inverter 2.1 entfällt.

Als nächstes ist die richtige Initialisierung des GDC zu prüfen.

Siehe dazu die neue PARATAB in 8.1.

Dazu wird das Programm "BS_EXPR" geladen (Quelle), assemblieren und das Teilprogramm "A" mit dem TEMO durchgehen.

Alternativ: UP "Befehl" mit TEMO durchgehen,
zuvor (!) jedoch Reg L mit der Befehls-Nr. entspr. EQU-Anweisung laden.

Prüfen, ob:

- richtiger Befehls-Code übermittelt wird,
- die Anzahl und die richtigen Parameter.

Für alle zur Initialisierung notwendigen Befehle (siehe Listing).

- Auswirkungen auf den GDC:
- RAM-Adressierung 0 bis FFFFH,
gebraucht werden zunächst 20480 Byte für 255 Zeilen.

Letztes Byte im Video-RAM = 50FFH (?) - Nein! –

Durch die spez. SR-Ansteuerung werden für 1 Byte auf dem Bildschirm (Farbe)

2 Worte Bildschirmspeicher benötigt, ist A0FFH die letzte Adresse im Video-RAM.

Programm-Test ergab Fehler im Listing:

-> sind aber beseitigt, d.h. müsste neu ausgedruckt werden (wegen Adressierung)

Test der UP schwierig, da das aufgerufene UP OUTKO, bzw. OUTPA in Schleife geht (Bit 1)

-> günstiger wäre es vielleicht für den Test am 1.Byte von OUTKO, bzw. OURPA temporär ein RET
(C9) einzutragen und somit die Schleife zu umgehen (???)

Nein, ist nicht notwendig, die Routine funktioniert!

Quelle ist ergänzt mit ORG-Anweisung für PARATAB und neu abgespeichert. Die PARATAB-Einträge
sind in jedem Fall noch zu ergänzen (im RAM ???) ...

Achtung! –

Möglicherweise wird je nach Prog.-Größe das Endes von TABINI überschrieben ...

-> PARATAB neu anweisen (weiter hinten ?) und ausdrucken ...

Ein neuer Ausdruck ist erfolgt mit ORG = 3100H für PARATAB (50Byte frei zwischen TABINI und
PARATAB.

PARATAB-Einträge nicht vergessen!

Als nächstes Parameter für FIGSINI bestimmen und in PRAM eintragen – welche Funktion ???

(hier enden die handschriftlichen Aufzeichnungen, d.h. zu Weiterem ist es dann nicht mehr
gekommen ...)

8.1. PARATAB neu

verschiedene Varianten ...

(1)	12H			
(2)	2BH	40W / Zeile +5W /-Rücklauf - 2		26
(3)	E3H	1 1 1 0 0 0 1 1 B ----- Vsynch Hsynch nwBits	E8H -> Hsyn= 4μs	
(4)	08	0 0 0 0 1 0 0 0 B ----- hvSS (= 2) Vsynch (hwBits)	1BH -> Vsyn = 15μs hvSS = 6μs	1AH (1,6μs)
(5)	05	0 0 0 0 0 1 0 1 B ----- hhSS (= 3)	0AH -> hhSS = 9μs	
(6)	07	0 0 0 0 0 1 1 1 B ----- vvSS	1FH -> vvSS = 15μs	
(7)	0FFH	255 Zeilen / Bild (nwBits)	512 ?	
(8)	040H	0 1 0 0 0 0 0 0 B ----- vhSS Zeilen / Bild (hwBits)	F9H -> vhSS = 15μs	48

1 Zeile = 36μs

1 Bild = 11ms

Anhang – Tabellen

1. Initialisierungsprogramm für den GDC

Basis: 1. Tabelle (INITAB1) - Kommandos
2. Tabelle (INITAB2) - Parameter

Prinzip:	INITAB1:	00	; Reset	(8)
		0EH	; Sync – Display - dunkel (BLANK)	(8)
		0FH	; - “ - - freigegeben	(8)
		6EH	; VSYNC – ext. Synchr.	(0)
		6FH	; - “ - intern Synchr.-Imp,	(0)
		4BH	; CCHAR	(3)
		6AH	; START (ohne Param.!)	(0)
		0CH	; BCTRL - Bildfläche dunkel	(0)
		0DH	; - “ - Bildfläche freigegeben	
		46H	; ZOOM	(1)
		49H	; CURS - Cursorposition	(3)
		7X	; PRAM - Startadresse	(0 bis 15)
		47H	; PITCH	(1)
		2X	; WDAT	(2)
		3X	; WDAT	(2)
		4AH	; MASK	(2)
		4CH	; FIGS	(0)
		6CH	; FIGD	(0)
		68H	; GCHRD	(0)
		AX	; RDAT	(0)
		BX	; RDAT	(0)
		E0H	; CURD	(5)
		C0H	; LPRD	(5)
		AX	; DMAR	(0)
		BX	; DMAR	(0)
		2X	; DMAW	(0)
		3X	; DMAW	(0)

2. GDC - Speicherabbild

a) Display-Zyklus = 500ns

A0

D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
-----	-----	-----	-----	-----	-----	----	----	----	----	----	----	----	----	----	----

A1

D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
-----	-----	-----	-----	-----	-----	----	----	----	----	----	----	----	----	----	----

weitere Display-Zykl.

A3

D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
-----	-----	-----	-----	-----	-----	----	----	----	----	----	----	----	----	----	----

A4

D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
-----	-----	-----	-----	-----	-----	----	----	----	----	----	----	----	----	----	----

usw.

b) Schieberegister-Ausgang

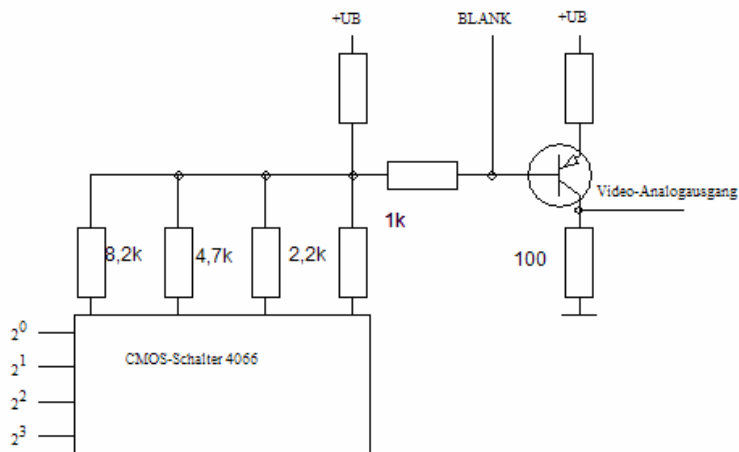
A0D0	A0D1	A0D2	A0D3	A1D0	A1D1	A1D2	A1D3	A2D0	A2D1	A2D2	A2D3	A3D0	A3D1	A3D2	A3D3
A0D4	A0D5	A0D6	A0D7	A1D4	A1D5	A1D6	A1D7	A2D4	A2D5	A2D6	A2D7	A3D4	A3D5	A3D6	A3D7
A0D8	A0D9	A0D10	A0D11	A1D8	A1D9	A1D10	A1D11	A2D8	A2D9	A2D10	A2D11	A3D8	A3D9	A3D10	A3D11
A0D12	A0D13	A0D14	A0D15	A1D12	A1D13	A1D14	A1D15	A2D12	A2D13	A2D14	A2D15	A3D12	A3D13	A3D14	A3D15

usw.

Ein Display-Zyklus = 500ns

2. Einfacher Video-DA-Wandler (4-Bit)

Zum Testen am Monitor wird ein einfacher Video-DA-Wandler gebraucht, der das 4Bit-Digitalsignal in eine Analog-Spannung unsetzt:



Funktionsprobe mit Treppenfunktion:

- geringfügige Verzerrungen im oberen Bereich, die wahrscheinlich nicht relevant sind