

Ingo Eickmann

Laßt Bilder leuchten

Die Programmierung des CRT-Controllers 6845

Der universelle CRT-Controller 6845 ist das Herz jeder CGA-, MDA- und jeder Hercules-Grafikkarte. Auch bei den neueren Videoadaptoren EGA, VGA und MCGA ist seine Funktionsweise in den LSI-Chip übernommen worden. Welche Möglichkeiten bietet er und wie muß er programmiert werden?

Der 6845 ist tot. So hieß es bereits vor Jahren, denn als er für die ersten PC-Videokarten verwendet wurde, war er schon einige Zeit auf dem Markt und entsprach nicht mehr dem neuesten Stand der Technik. Er zählte jedoch schon damals wegen seiner universellen Eigenschaften zu den meistverwendeten Cathode Ray Tube Controllern, abgekürzt CRTC.

Der 6845 lebt

Auf originalen CGA-, MDA- und Hercules-Karten ist er als einziger 40poliger DIL-Chip leicht zu lokalisieren. Bei den modernen EGA-, VGA- und MCGA-Videokarten ist er zwar nicht mehr körperlich anwesend, seine Funktionsweise wurde beim Design der LSI-Chipsätze für Grafikkarten jedoch prinzipiell beibehalten. Hierdurch wurde sowohl weitgehend Kompatibilität zu den alten Videostandards erreicht, als auch die Grundvoraussetzung geschaffen, frühere Grafikmodi zu emulieren. Doch jede Kompatibilität bei einer abwärtskompatiblen Grafikkarte mit erweiterten Fähigkeiten hat ihre Grenzen.

Ein Maß für die Kompatibilität des CRTC-Controllers ist heute noch immer sein Verhalten bei Spielen, die den CRTC oft recht radikal neu programmieren. Als Kompatibilitätsbeweis wird selbst von Vertriebsleuten die Lauffähigkeit des Flugsimulators genannt. Ein gegenüber dem Original-6845 leicht modifizierter CRTC einer EGA- oder VGA-Karte kann aufgrund unterschiedlicher Registerbelegungen leicht zu Inkompatibilitäten führen.

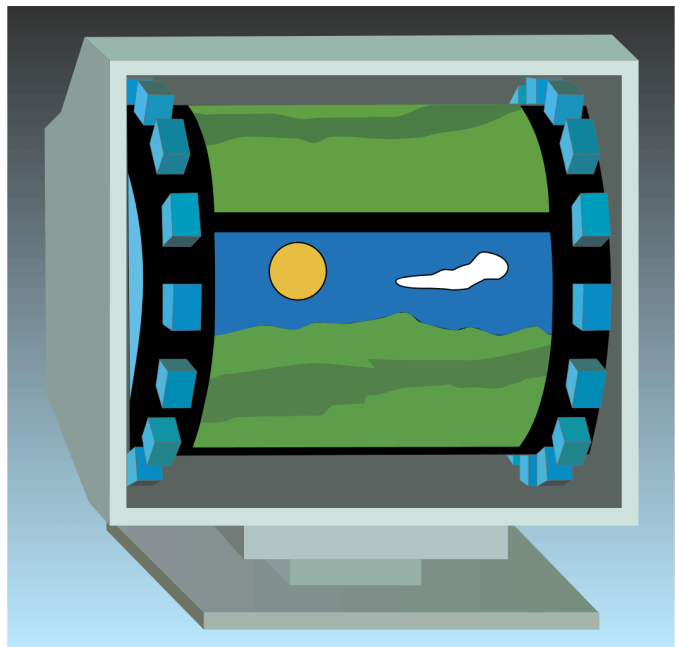
Jeder Monitor benötigt zur Darstellung eines stehenden Bildes neben dem Pixelsignal (Videosignal) horizontale und vertikale Synchronisationsimpulse zur Rückführung des Kathodenstrahls. Der CRTC hat die Aufgabe, beide Synchronisationssignale über einen externen Taktgenerator zu erzeugen und während der anschließenden Bewegung des Kathodenstrahls die richtigen Pixelinformationen aus dem Videospeicher zu holen.

Er muß hierzu so programmiert werden, daß bei gegebener Videobandbreite (Pixelrate) die horizontalen und vertikalen Frequenzanforderungen des Monitors erfüllt werden. Im Zeitalter der Multisync- und Multiscan-Monitore ist diese Programmierung nicht mehr so kritisch wie früher, da diese neuen Monitore einen ganzen Frequenzbereich verarbeiten können.

Der Monitor bebzt

Der CRT-Controller ist kein beliebig und gefahrlos programmierbarer Baustein, an dem man rumspielen kann. Viermal hat ein TTL-Monitor meine Experimente wegen falscher und bis an die Grenzen gehender Programmierung nicht überlebt, davon dreimal gewollt und reproduzierbar. Durch besonderes Timing der Synchronisationssignale können im Zeilentransformator des Monitors so hohe Spannungen entstehen, daß die Isolation zwischen den Wicklungen durchbrennt. Übrigens hält sich seit dem legendären Tandy Model II das Gerücht, daß es auch möglich sein soll, mit dem CRT-Controller die Bildröhre implodieren zu lassen.

Soweit zum Gebiet der destruktiven Programmierung. Dieser Artikel liefert in erster Linie Hintergrundinformationen über den Controller, und nur Profis sollten das Timing des 6845 selbst programmieren, oder den CRT-Controller einer PC-Videokarte für andere Auflösungen einstellen.



Abzählreim für Videocontroller

Bild 1 erläutert wichtige Parameter für die Programmierung des 6845. Intern teilt der CRTC eine Zeile in einen sichtbaren und einen unsichtbaren Bereich. Im letzteren wird der Kathodenstrahl dunkel geschaltet (Overscan) und – nach einem horizontalen Synchronisationsimpuls – zum Anfang der nächsten Zeile geführt (Retrace). Vertikal ist die gleiche Einteilung vorgenommen worden. Intern werden insgesamt mehr Zeilen hochgezählt, als auf dem Bildschirm sichtbar werden (Vertical Displayed). Als Ausgleich sind vertikal noch Rasterzeilen, auch ScanLines genannt, einzelne angehängt (Vertical Total Adjust). Der memory address pointer des CRTC, der auf die Adresse des gerade verarbeiteten Zeichens im Videospeicher zeigt, wird lediglich im sichtbaren Bildschirmbereich hochgezählt.

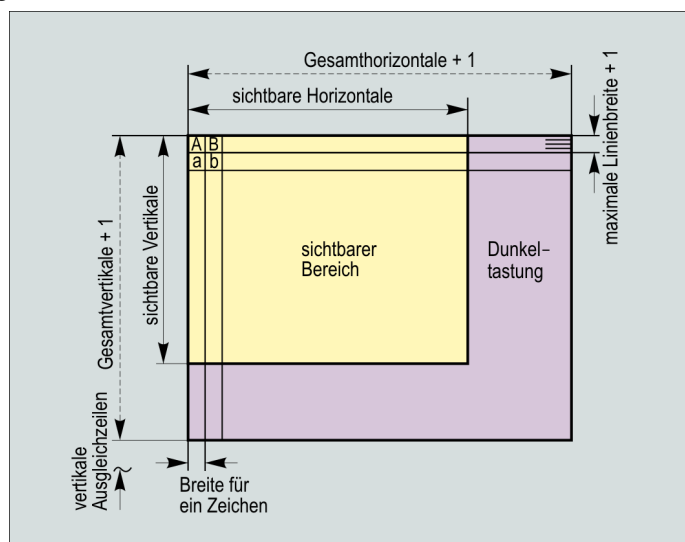


Bild 1. Der CRTC 6845 arbeitet intern mit einem sichtbaren Bereich des Bildschirms und nicht sichtbaren Teilen für die Rückführung des Kathodenstrahls

Ein Zeichen besteht aus mehreren Rasterzeilen, die vom Kathodenstrahl in aufeinanderfolgenden Horizontalbewegungen geschrieben werden. Der Memory Address Pointer muß nach jeder Rasterzeile einer Zeichenzeile wieder auf den Anfang der Zeile springen und darf erst nach der Darstellung aller Rasterzeilen dieser Character Line auf die Adresse des nächsten Zeilenanfangs erhöht werden.

In den Textmodi holt der Memory Address Pointer den ASCII-Code des darzustellenden Zeichens aus dem Videospeicher, der wiederum zusammen mit der Spaltenadresse vom CRTC die aktuelle Rasterzeile in einem Character Generator auswählt. Im Gegensatz dazu wird der CRTC in den Grafikmodi mit nur einer Rasterzeile pro Zeichen programmiert. Also erhöht er nach jeder Rasterzeile den memory address pointer. Die Daten des Videospeichers werden durch die externe Beschaltung direkt zum Monitor geschickt.

PC-Videoadapter arbeiten in den Grafikmodi teilweise auch mit zwei oder vier ScanLines pro Zeichen und benutzen zusätzlich zur Memory Address die Spaltenadresse des CRTC zur Adressierung des Videospeichers. Da Grafikmodi aus der Sicht des CRT-Controllers spezielle Textmodi mit einer sehr flachen Zeichenmatrix sind, wird im Folgenden nur auf den allgemeinen Fall wirklicher Textmodi eingegangen. Die Zeitbasis für die Programmierung des 6845 ist ein character clock, die Zeit, in der der Kathodenstrahl eine Rasterzeile der Zeichenmatrix schreibt. Alle Register des 6845, die Positionen und Zeiten angeben, beziehen sich auf diese Einheit.

Registriertes Timing

Die Tabelle 1 enthält alle Register des CRTC 6845. Die Register R0 bis R9 legen das horizontale und vertikale Timing fest. Die Register R0 bis R13 sind nur beschreibbar (w), R14/R15 sind auch lesbar (r/w) und R16/R17 nur lesbar (r). In der rechten Spalte der Tabelle sind die im 6845 für die Speicherung vorhandenen Bits aufgeführt. Der 6845 enthält insgesamt 19 Register, ein Adreßregister zur Anwahl eines Datenregisters und die ersten 18 Datenregister der Tabelle (R0 bis R17). Zusätzlich enthält diese Tabelle die Änderungen und Erweiterungen für die neueren EGA-, VGA- und MCGA-Videoadapter. Sie sind dort vermerkt, wo die Registerinhalte des 6845 nicht übernommen wurden. Dies betrifft bei der EGA/VGA besonders den Teil der Timing-Informationen (R0 bis R9). Vergleicht man die neuen Registerinhalte mit denen des 6845, dann wird sofort klar, daß bei der Programmierung einiger EGA-/VGA-Register mit Werten für einen 6845 diese völlig anders interpretiert werden. Die Änderungen bei den Registersätzen moderner Grafikkarten wurden notwendig, um neue Modi und Auflösungen zu ermöglichen. Die CRT-Controller der MCGA und EGA wurden beide vom Registersatz des 6845 abgeleitet, wobei Veränderungen notwendig waren, um 16 Rasterzeilen pro Zeichen und mehr als 256 Rasterzeilen pro Bild verarbeiten zu können. Der CRTC der VGA wurde weitgehend vom Controller der EGA übernommen, die Light Pen Funktionen blieben jedoch nicht mehr implementiert. Bei beiden PS/2-Grafikadaptern, VGA und MCGA, sind alle Register - im Gegensatz zu ihren Vorgängern - auch lesbar.

Registerinhalte berechnen

Die Berechnung der Registerinhalte erfolgt in Abhängigkeit von den Frequenzanforderungen des Monitors, also der maximalen Videobandbreite und der horizontalen und vertikalen Synchronisation sowie von der Zeichenmatrix und der Anzahl der sichtbaren Spalten und Zeilen. Die Register *R10*, *R11*, *R12/R13* und *R14/ R15* legen den Cursor und die Bildstartadresse (Page 0: 0000h) fest. *R17/R18* geben die letzte Positionsmeldung des Light Pen wieder. Diese Register sind bei der Programmierung unkritisch. Wesentlich anspruchsvoller sind die Timing Register *R0* bis *R9*. Alle für ihre Berechnung notwendigen Parameter stehen in *Tabelle 2*. Die *Tabelle 3* enthält den Rechenweg für die Programmierung des 6845 aus jenen Parametern. Wenn ein Videoadapter neu dimensioniert wird, dann erfolgt zunächst eine Abschätzung der Videobandbreite *f*. Soll jedoch ein PC-Videoadapter für eine neue Auflösung umprogrammiert werden, dann liegt die Videobandbreite durch das bereits existierende Taktsignal für den CRTC fest. Diese Frequenzen sind für die PC-Standardmodi in *Tabelle 4* zusammengestellt. Alle äußeren Beschaltungen des CRTC werden von den Mode Control Registern der Grafikkarte selektiert [2].

Nach der Berechnung der Registerinhalte *R0* bis *R9* müssen als Probe die resultierenden Rückführzeiten (Horizontal/Vertical Retrace Time) mit den Anforderungen des Monitors verglichen werden. Nur wenn sie größer als die Minimalwerte aus dessen Spezifikation sind, kann der Monitor mit dieser Ansteuerung betrieben werden. Ist dies nicht der Fall, dann muß die Berechnung mit veränderten Parametern (*Tabelle 2*) wiederholt werden. Einige Variationsmöglichkeiten bieten die Startpositionen und die Impulslängen der horizontalen und vertikalen Synchronisationssignale. Als brauchbare Faustregel hat sich ein horizontaler Synchronisationsimpuls während zwei Drittel des Overscan und eine vertikale Synchronisation unmittelbar nach der letzten Rasterzeile erwiesen.

Anhand der *Tabelle 1* können die Berechnungen auch für die leicht modifizierten CRTC-Register einer EGA, VGA oder MCGA durchgeführt werden. *Tabelle 4* enthält die Parameter für die Standardmodi der PC- und PS/2-Grafiksysteme. Hieraus wird deutlich, daß mit der Anzahl der verfügbaren Auflösungen einer Grafikkarte die Zahl der benötigten Frequenzen wächst. Lediglich die 14,318 MHz für den CGA-Mode werden vom Motherboard an B30 (OSC) des PC-Slot angeboten. Die MCGA bietet einen zusätzlichen Service. Ein gesetztes Bit 3 im Memory-Controller-Mode-Control-Register (*R16*) veranlaßt die interne Berechnung der horizontalen Timing-Parameter für den jeweiligen Videomode, sofern die Register *R0* bis *R3* die entsprechenden Parameter für Videomode 0 (40x25 color) enthielten. Warum sollte man die Registerinhalte mühsam nach *Tabelle 3* von Hand ausrechnen? *Bild 2* enthält das Turbo-Pascal-Programm *6845regs.pas* zur Berechnung der Timing-Registerinhalte. Die Registerinhalte *R0* bis *R9* des CRTC werden berechnet, angezeigt und anhand der Retrace-Zeiten überprüft. Die Datenregister des 6845 wählt man bei PC-Grafikkarten über ein Adreßregister an. Bei allen monochromen Adaptern, sowie der Hercules Incolor Card und den monochromen Modi der EGA und VGA befindet sich das Adreßregister bei Port 3B4h und das angesprochene Datenregister bei 3B5h. CGA, MCGA und alle Farbmodi der EGA und VGA haben das Adreßregister bei Port 3D4h und das Datenregister bei 3D5h. Das PC-BIOS verwaltet die Portadresse des primären Videoadapters (3B4h oder 3D4h) im Systemsegment (0040:0063h).

Beim Beschreiben oder Lesen der Datenregister muß grundsätzlich gewährleistet sein, daß zwischen dem Beschreiben des Adreßregisters und der Bearbeitung des Datenregisters keine Veränderung des Adreßregisters durch eine Interrupt-Service-Routine erfolgt. Daher sollten bei sauberer Programmierung die Interrupts ausgeschaltet werden:

```
mov dx,3B4h    ; Adreßport
mov al,1      ; Register 1
cli           ; Interrupts ausschalten
out dx,al     ; Adreßregister belegen
inc dx       ; Datenport adressieren
mov al,80     ; Wert für Register 1
out dx,al     ; Register 1 beschreiben
sti          ; Interrupts wieder
             ; zulassen
```

EGA-, VGA-, MCGA- und Hercules-Adapter lassen auch das Beschreiben mit einem 16 Bit OUT-Befehl zu, hier kann das Ausschalten der Interrupts entfallen:

```
mov ax,100h + 80
out dx,ax
```

Bevor nach jedem Einschalten des Computers der Monitor synchronisiert und die ersten vernünftigen Zeichen auf dem Bildschirm erscheinen, muß der Controller initialisiert werden. Daher sind die Parameter für den 6845 im ROM abgelegt, bei CGA-Systemen im BIOS-ROM auf der Hauptplatine, bei neueren Grafikkarten im Video-BIOS auf der jeweiligen Karte. *Tabelle 5* enthält die Parameter für die CRTC-Einstellungen unterschiedlicher Videoadapter. Wer eine Super-VGA betreibt, kann diese Tabelle auch für seine Super-Modi leicht erweitern, indem er die Register des CRTC im jeweiligen Mode ausliest. Leider lassen die älteren Grafikkarten oft keine höheren Textauflösungen zu. So kann der CRTC einer Hercules-Karte nicht für eine 132x25 Auflösung programmiert werden, da im Videospeicher maximal 2048 Character für den Bildaufbau adressiert werden können. Ein weiterer Nachteil dieser Grafikkarte – und auch der CGA – ist, daß zwar der CRTC unterschiedliche

Zeichenmatrizen verarbeiten kann, der Character Generator jedoch fest auf 8x14 oder 8x8 Pixel pro Zeichen programmiert ist. Um die Zeichen in einer anderen Größe auf dem Monitor darzustellen, müßte das Character-Generator-EPROM gegen einen selbst programmierten Chip ausgetauscht werden.

Um die Baugruppe des CRTC vollständig zu beschreiben, ist in *Tabelle 6* die unterschiedliche Belegung des CRTC-Statusregisters der Standardgrafikkarten wiedergegeben. Ein Tip am Rande: Nur die Hercules-Karte toggelt das Bit 7, hierdurch läßt sie sich leicht von einer anderen monochromen Karte unterscheiden. Nun haben Sie eine Fülle von Informationen über Ihren CRTC erhalten. Ich habe an dieser Stelle absichtlich kein fertiges Programm vorgestellt, das das Umprogrammieren eines CRT-Controllers auf einer bestimmten PC-Grafikkarte zuläßt. Die Gefahr, daß es blind eingehackt wird, und man die Hardware durch spielerisches Ausprobieren und falsche Programmierung beschädigt, ist zu groß. Ganz abgesehen vom materiellen Schaden – nicht jeder kann sich einen Zeilentrafo neu wickeln – ist es für den Benutzer selbst nicht ganz ungefährlich, vor einem plötzlich dahinscheidenden Monitor zu sitzen.

Wer genügend Know-how besitzt, um den CRT-Controller neu zu programmieren, dem fällt es anhand der *Tabelle 1* leicht, das Programm *6845regs.pas* mit wenigen Befehlen so umzuschreiben, daß es einen Videoadapter direkt mit den neu berechneten Werten programmiert. Am angenehmsten kann man damit auf einem Computer mit zwei Videosystemen arbeiten; auf dem einen Display erscheint das Menü des Programms und auf dem Bildschirm des Probanden ein Testbild.

Literatur

[1] MC6845 Data Sheet, Motorola Inc., 1984.

[2] *Wilton, R.*: The programmer's guide to PC and PS/2 Video Systems, Microsoft Press, 1987

Tabelle 1. Register des 6845 und modifizierter Controller

R#	Name	Inhalt	r/w	Bits des 6845
Horizontal				
0	Horizontal Total	Anzahl der Zeichen pro Zeile	w	xxxx xxxx
1	Horizontal Displayed EGA: Horizontal Display Enable End	Angezeigte Zeichen $R1 < R0$ Anzeige unterdrücken	w	xxxx xxxx
2	Horizontal Sync. Position EGA: Start Horizontal Blanking	Startposition für Horizontalrücklauf, höherer Wert schiebt den Bildschirminhalt nach links. $R2 > R1$ Start der Dunkeltastung	w	xxxx xxxx
3	Horizontal Sync. Pulse Width EGA: End Horizontal Blanking	Dauer des horizontalen Synchronisationsimpulses in Character Clocks. $R3 > (R0 - R2)$ Bit 4...0: Beginn und Dauer der Dunkeltastung Bit 6, 5: Anzeige ermöglichen	w	___xxxx
Vertical				
4	Vertical Total EGA: Start Horizontal Retrace	Anzahl aller Zeichenzeilen – 1 Start des Horizontal-Rücklaufs	w	_xxx xxxx
5	Vertikal Total Adjust EGA: End Horizontal Retrace	Vertikale Summenanpassung Bit 4...0: Beginn und Dauer des Zeilenrücklaufes Bit 6, 5: Horizontale Dunkeltastung Bit 7: ungerade (1) / gerade (0) Start-Adresse für Pixel Panning	w	__x xxxx
6	Vertical Displayed EGA: Vertical Total	Anzahl der angezeigten Zeichenzeilen $R6 < R4$ Bildschirmgesamthöhe	w	_xxx xxxx
7	Vertical Sync. Position EGA: Overflow (sichtb. Rand)	Startposition für vertikalen Rücklauf in Character Clocks, höherer Wert schiebt Bildschirminhalt nach oben. Bit 0: Gesamthöhe Bit 1: Anzeige unterdrücken Bit 2: Beginn des Vertikal-Rücklaufs Bit 3: Beginn der Vertikal-Dunkeltastung Bit 4: Linienzahlenvergleich mit Bit 8 Bit 5: Cursorposition	w	_xxx xxxx
8	Interlace Mode EGA: Preset Row Scan MCGA: reserviert	Bit 0 = 0: Kein Interlace (normal) Interlace: Bit 1 (0/1): Sync-Modus/Sync- und Video-Modus Bit 4: Erste Linie oben, wichtig für vertikales Pixel Panning	w	___xx
9	Maximum ScanLine MCGA: ScanLines per Char.	Anzahl der Rasterzeilen – 1 EGA: Bit 6: Linienzahlenvergleich mit Bit 9 Anzahl der Zeilen pro Zeichen / 2	w	__x xxxx

R#	Name	Inhalt	r/w	Bits des 6845
	Cursor			
10	Cursor Start	Bit 4...0: Startadresse des Cursorblocks innerhalb der Zeichenmatrix Bit 6, 5: 0: Cursor blinkt nicht 1: Cursor aus 2: blinkt mit $f_v/16$ 3: blinkt mit $f_v/32$ MCGA: Cursor-Start / 2	w	_xxx xxxx
11	Cursor End	Endadresse des Cursorblocks innerhalb der Zeichenmatrix EGA: Bit 6, 5: Cursorform MCGA: Cursorende / 2	w	__x xxxx
12	Start Address (High)	Speicheradresse des ersten sichtbaren Zeichens links oben, automatisch B000h (monochrom); B800h (color)	w	__xx xxxx
13	Start Address (Low)	niedrigwertiger Teil der Startadresse	w	xxxx xxxx
14	Cursor Location (High)	Refresh Adresse für den Cursor	r/w	__xx xxxx
15	Cursor Location (Low)	niedrigwertiger Teil der Cursorposition	r/w	xxxx xxxx
16	Light Pen (High) EGA/VGA: Vertical Retrace Start MCGA: Memory Controller Mode Control	Refresh Adresse des CRTC bei ansteigender Flanke eines Light-Pen-Impulses. Dieser Wert muß abhängig von Verzögerungen im Light Pen und Synchronizer des CRTC softwareseitig korrigiert werden (nicht bei VGA). Startposition für Bildwechsel (Vertikalrücklauf) Bit 0: 320x200, 256 Farben Bit 1: 640x480, 2 Farben Bit 2: reserviert Bit 3: Zeilenfrequenz Bit 4: Punkt-Clock ermöglichen Bit 5: reserviert Bit 6: invertiertes Bit 8 aus Register 6 Bit 7: Register 0...7 schreibschützen	r w r/w	__xx xxxx
17	Light Pen (Low) EGA: Vertical Retrace End MCGA: Interrupt Control	niedrigwertiger Teil der der Adresse (R16) Beginn und Dauer des Bildwechsels Bit 4 = 0: Vertikal-Interrupt löschen Bit 5 = 0: Vertikal-Interrupt (IRQ 2) zulassen Register 0...7 schreibschützen	r	__xx xxxx

R#	Name	Inhalt	r/w	Bits des 6845
	nur bei LSI-Chips:			
18	EGA: Vertical Display Enable End MCGA: Character Generator, Sync. Polarity	Vertikale: Anzeige unterdrücken Zeichengenerator	w	
19	EGA: Offset (Logical Line Width) MCGA: Character Generator Pointer	Startadresse Zeiger auf Zeichengenerator	w	
20	EGA: Underline Location MCGA: Character Generator Count	untere Rasterlinie der Zeichenlinie Zeichenzähler	w	
21	EGA: Start Vertical Blanking	Beginn der vertikalen Dunkeltastung	w	
22	EGA: End Vertical Blanking	Bit 4...0: Start und Dauer der Dunkeltastung	w	
23	EGA: Mode control	Bit 0: CGA-kompatibel, 8K-Offset zwischen geraden und ungeraden Zeilen Bit 1: Spaltenzähler auswählen Bit 2: Horizontal-Rücklauf / 2 Bit 3: Memory-Adresse durch 2 teilen zur Berechnung der Refresh-Adresse Bit 4: hochohmiger Output Bit 5: Adreßumleitung bei CGA-Kompatibilität Bit 6: Adreß-Modus 0 = Wort 1 = Byte Bit 7: Hardware-Reset		
24	EGA: Line Compare	Ist diese Zeile erreicht, wird der Memory-Adreß-Zähler auf 0000h gesetzt.	w	
25	EGA: Double Scan Control	Bit 0: Double Scan gestattet Bit 1: Test Bit 2: Flag für Liniensummenausgleich	w	
32 : 63	MCGA: reserviert			

Tabelle 2. Wichtige Parameter

Video-Bandbreite (falls für den CRT bereits vorgegeben):	f
Zeilenfrequenz:	f_H
Bildfrequenz:	f_V
minimale Rückführzeit des Strahls (Monitorabhängig, ca. 0,01 ms)	t_{RHmin}
minimale Zeit für Bildaufbau (Monitorabhängig, 1...3 ms):	t_{RVmin}
Anzahl der Zeichen pro Zeile	N
Anzahl der Zeilen:	R
Punkte pro Zeichen (horizontal):	D
Punkte pro Zeichen (vertikal):	S

Tabelle 3. Formeln für Register

Video-Bandbreite (Abschätzung):

$$f \approx f' = \frac{N \cdot D}{\frac{1}{f_H} - t_{RHmin}} \cdot 1,05$$

Horizontal total:

$$R0 = \frac{f'}{f_H \cdot D} \text{ (abrunden)}$$

bei gegebener Videobandbreite:

$$R0 = \frac{f}{f_H \cdot D} \text{ (abrunden)}$$

Video-Bandbreite: $f = f_H \cdot D \cdot (R0 + 1)$ Horizontal sichtbar: $R1 = N$

Horizontal-Synchronisation S-Pulsbreite:

$$R3 = 2 \cdot \frac{R0 - N}{3} \text{ (aufrunden)}$$

Horizontal-Synchronisation-Position:

$$R2 = N + \frac{R3}{6}$$

Anzahl der Zeichenzeilen: $R_T = \frac{f_H}{S \cdot f_V}$ Vertikal total: $R4 = R_T - 1$

Vertikale Ausgleichszeilen:

$$R5 = \frac{f_H}{f_V} - S \cdot R_T \text{ (abrunden)}$$

Vertikal sichtbar: $R6 = R$ Vertikal-Synchronisation-Position: $R7 = R$

(unmittelbar nach letzter ScanLine)

Interlace Mode (non interlace): $R8 = 2$ max. Linienanzahl: $R9 = S - 1$ **Probe:**

Horizontale Rückführzeit:

$$t_{RH} = (R0 + 1 - R2) \cdot \frac{D}{f}$$

Vertikale Rückführzeit:

$$t_{RV} = \frac{1}{f_V} - \frac{R \cdot S}{f_H}$$

Es muß gelten: $t_{RH} \geq t_{RHmin}$ und $t_{RV} \geq t_{RVmin}$

Tabelle 4. Frequenzen für die PC-Standardmodi

Adapter	MDA,HGC	CGA	EGA	VGA		
Mode	720x350	640x200	640x350	640x400	720x400	640x480
f in MHz	16,257	14,318	16,257	25,175	28,322	25,175
f_H in kHz	18,43	15,75	21,85	31,5	31,5	31,5
f_V in Hz	50	60	60	70	70	60
N	80	80	80	80	80	Grafikmode
R	25	25	25	25	25	
D	9	8	8	8	9	
S	14	8	14	16	16	

Tabelle 5. Parameter für die CRTC-Einstellungen

Adapter	MDA,HGC	CGA	CGA	CGA	EGA	VGA	HGC
Videofreq.	50 Hz	60 Hz	60 Hz	60 Hz	60 Hz	60 Hz	50 Hz
Mode	mono	co80	co40	Grafik	co80	co80	Grafik
R0	97	113	56	56	95	95	53
R1	80	80	40	40	79	79	45
R2	82	90	45	45	80	80	46
R3	15	10	10	10	2	2	7
R4	25	31	31	127	85	85	91
R5	6	6	6	6	129	129	2
R6	25	25	25	100	191	191	87
R7	25	28	28	112	31	31	87
R8	2	2	2	2	0	0	2
R9	13	7	7	1	77	79	3

Tabelle 6. Belegung des CRTC-Statusregisters

Adapter	MDA	HGC	CGA, EGA	VGA, MCGA
Bit 0	Horiz. Sync.	Horiz. Sync.	Anzeige erlaubt	Anzeige erlaubt
Bit 1		Light Pen Trigger	Light Pen Trigger	
Bit 2			Light Pen Switch	
Bit 3	Bildlauf	Bildlauf	Vertikal Sync.	Vertikal Sync.
Bit 7		Vertical Sync.		

```

-----
6845 Register Calculations                                     release 1.0
Copyright (c) Ingo Eickmann, 1989                            10/30/89
-----
Compilieren mit Turbo Pascal 4.0/5.x                         I. Eickmann
Getestet unter MS-DOS 3.3                                    Im Leuchterbruch 8
                                                            5000 Köln 80
-----
Die Inhalte der Timing-Register R0 bis R9 des 6845 werden in
Abhängigkeit der horizontalen und vertikalen Ablenkfrequenzen,
der Zeichenmatrix, der Spalten- und Zeilenzahl berechnet und
mit den resultierenden Retrace Zeiten ausgegeben.
-----

Program _6845Regs;

uses CRT;

var R      : array[0..9] of Integer;
    B      : array[1..9] of Real;
    BC     : array[1..9] of String;
    RC     : array[0..9] of String;
    i,j    : Integer;
    f,f_approx,t_horzRet,t_vertRet : Real;

Procedure CalcRegisters;
begin
    f_approx:=B[5]*B[7]/(1/B[1]-B[3])*1.05;
    R[0]:=trunc(f_approx/B[1]/B[7])-1;
    f:=B[1]*B[7]*(R[0]+1);
    R[1]:=trunc(B[5]);
    R[3]:=trunc((R[0]-R[1])*2/3)+1;
    R[2]:=trunc(R[1]+R[3]/6);
    R[4]:=trunc(B[1]/B[2]/B[8])-1;
    R[5]:=trunc(B[1]/B[2]-B[8]*(R[4]+1));
    R[6]:=trunc(B[6]);
    R[7]:=trunc(B[9]);
    R[9]:=trunc(B[8])-1;
    t_horzRet:=(R[0]+1-R[2])*B[7]/f;
    t_vertRet:=1/B[2]-B[6]*B[8]/B[1];
end;

Begin
    BC[1]:='Horizontal Frequency:  ';      B[1]:=18432;      (* Werte      *)
    BC[2]:='Vertical Frequency:  ';      B[2]:=50;        (* für Hercules *)
    BC[3]:='Min horz retrace time:  ';    B[3]:=10e-6;    (* Textmode   *)
    BC[4]:='Min vert retrace time:  ';    B[4]:=1e-3;
    BC[5]:='# of disp Char/Row:  ';      B[5]:=80;
    BC[6]:='# of disp Rows:  ';          B[6]:=25;
    BC[7]:='# of Dots/CharRow:  ';       B[7]:=9;
    BC[8]:='# of ScanLines/Char:  ';     B[8]:=14;
    BC[9]:='vert Sync Position:  ';      B[9]:=int(B[6]);
    R[8]:=2;                               (* Interlace Mode *)

    RC[0]:='Horizontal total:  ';
    RC[1]:='Horizontal displayed:  ';
    RC[2]:='Horizontal Sync Pos:  ';
    RC[3]:='Horizontal Sync Width:  ';
    RC[4]:='Vertical total:  ';
    RC[5]:='Vertical total adjust:  ';
    RC[6]:='Vertical displayed:  ';
    RC[7]:='Vertical Sync Pos:  ';
    RC[8]:='Interlace Mode:  ';
    RC[9]:='Max ScanLine Address:  ';

    R[8]:=2;                               { Interlace Mode }

    repeat
        clrscr;
        CalcRegisters;
        for i:=0 to 9 do
            writeln('R',i:1,' ',RC[i],' ',R[i]);
        writeln('Video Bandwidth:  ',f);
        write('t_horzRet: ',t_horzRet:9:8,' >= ',BC[3],' ',B[3]:9:8);
        writeln(' ',t_horzRet >= B[3]);
        write('t_vertRet: ',t_vertRet:9:8,' >= ',BC[4],' ',B[4]:9:8);
        writeln(' ',t_vertRet >= B[4]);
        writeln('');
        writeln('Menü: 0 Exit');
        for i:=1 to 9 do
            writeln(' ',i:2,' ',BC[i],' ',B[i]);
        write('Auswahl: '); readln(i);
        if i>0 then
            begin
                write('Wert: '); readln(B[i]);
            end
        until (i<=0) or (B[i]<0);
    End.

```

Bild 2. Das Programm 6845regs.pas berechnet die Inhalte der Timing-Register R0 bis R9 des 6845