Weiterführung des Entwurfs ab dem 19.2.2016

Stand: 06.03.2016

Allgemeines

Dieser Text ist noch nicht durchgesehen und noch nicht rechtschreibkorrigiert.
Dieser Text und der dazugehörige Schaltplan enthalten mit Sicherheit noch mehrere sachliche Fehler.
Beides ist daher als konzeptionelle Orientierung zu verstehen.

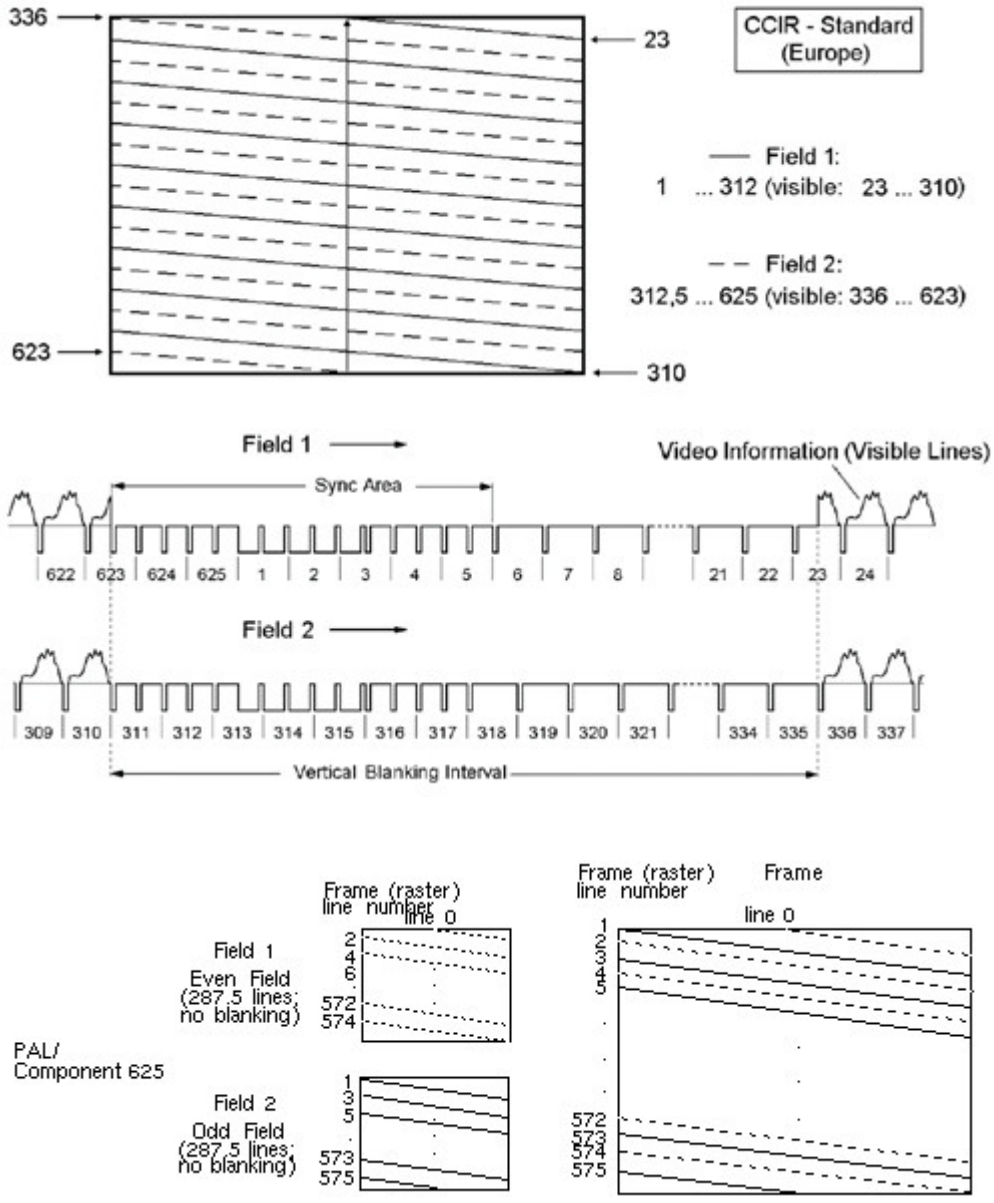
Die Durchsicht von Text und Schaltplan wird aber kurzfristig vorgenommen werden.

Neudefiniton des Video-Timings

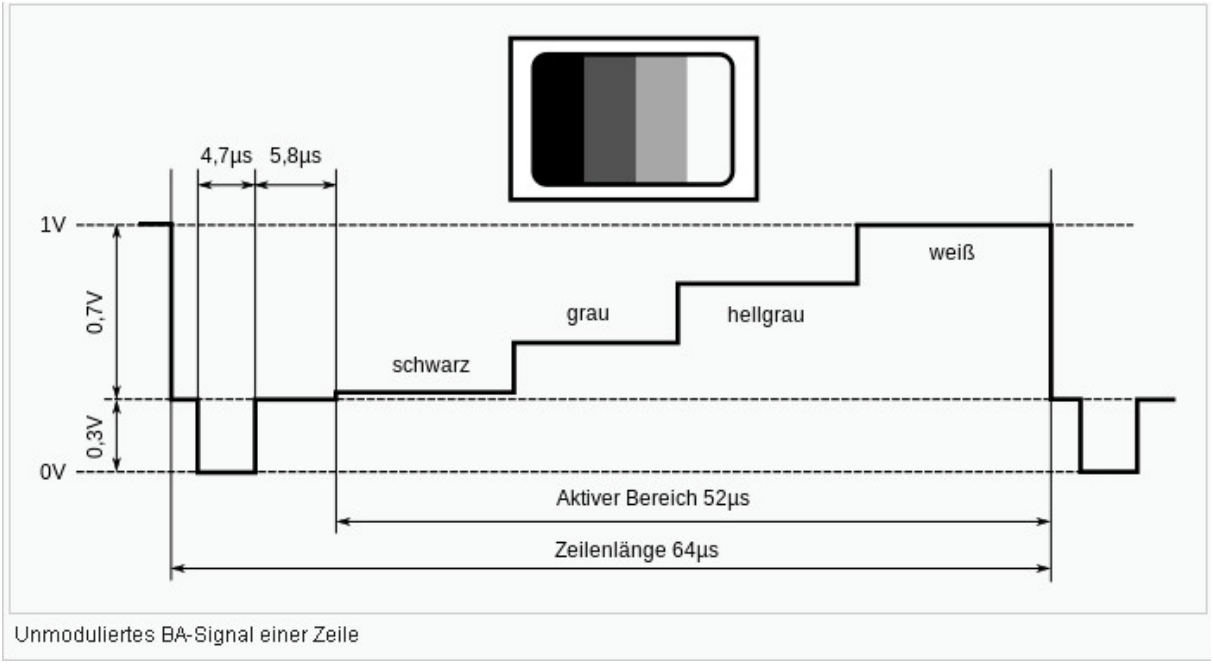
(19.02.2016)

Es zeigte sich, das die bisher genutzte Definition des Video-Signals nach Möhring (Seite 10) im Widerspruch zu anderen Quellen steht, das Timing bezüglich des ersten und des Zweiten Frames ist genau vertauscht.

Es wird nun das Timing implementiert, das in mehreren (heutigen) Quellen ünereinstimmend dargestellt ist:

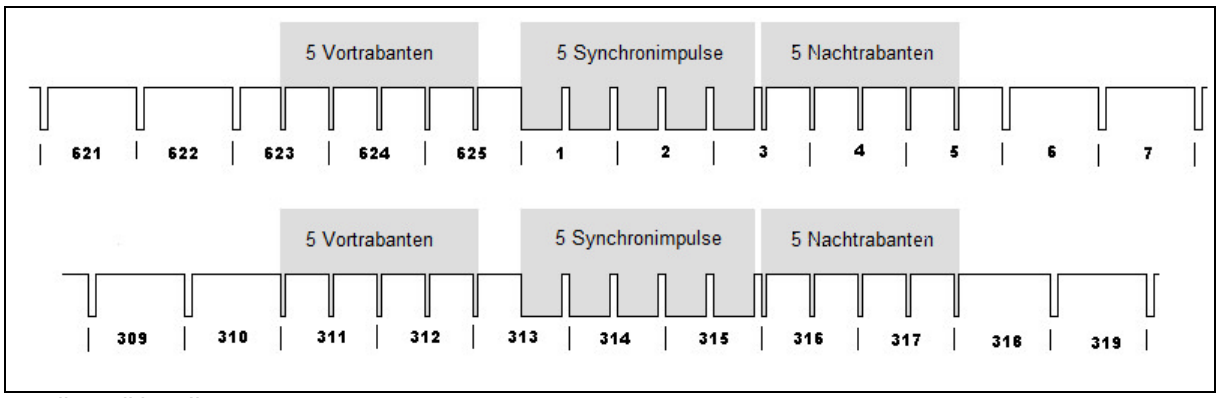


Achtung: hier andere Zählweise: die erste Zeile heisst Zeile Null !



Unmoduliertes BA-Signal einer Zeile

Quelle: Wikipedia



Quelle: Wikipedia

Es wurde die Entscheidung getroffen, das Video-Timing nach Wikipedia als Grundlage zur Realisierung zu verwenden.

Erzeugung des Video-Timings und weitere Schaltungsteile auf Video-Timing-Ebene

Grundsätzliches

Zunächst war vorgesehen, einen höher integrierten Baustein 74ACT715/LM1882 zur Erzeugung des Video-Timings zu verwenden. Im Verlauf des Entwurfsprozesses zeigte sich jedoch, das es vielfältige Verknüpfungen zwischen dem Video-Timing und den anderen Funktionsgruppen gibt, die teilweise vom Timing her sehr kritisch sind. Daher wurde die Entscheidung getroffen, auch das Video-Timing mit SSI-Bausteinen der Serie 74F zu realisieren. Dieser Weg ist zwar zunächst aufwendinger, er vermeidet jedoch mögliche Risiken bezüglich des Timings, die erst ein einem späteren Zeitpunkt des Projektverlaufs sichtbar werden könnten.

Aufgrund von Vorüberlegungen wird ein Pixeltakt von 14,7456MHz vorgesehen (PAL Square Pixel). Um das Timing der gesamten Video-Karte synchron zu gestalten wird der Pixel-Takt auch als Takt für die Steuerung des Textspeichers verwendet.

Erzeugung des Horizontal-Timings

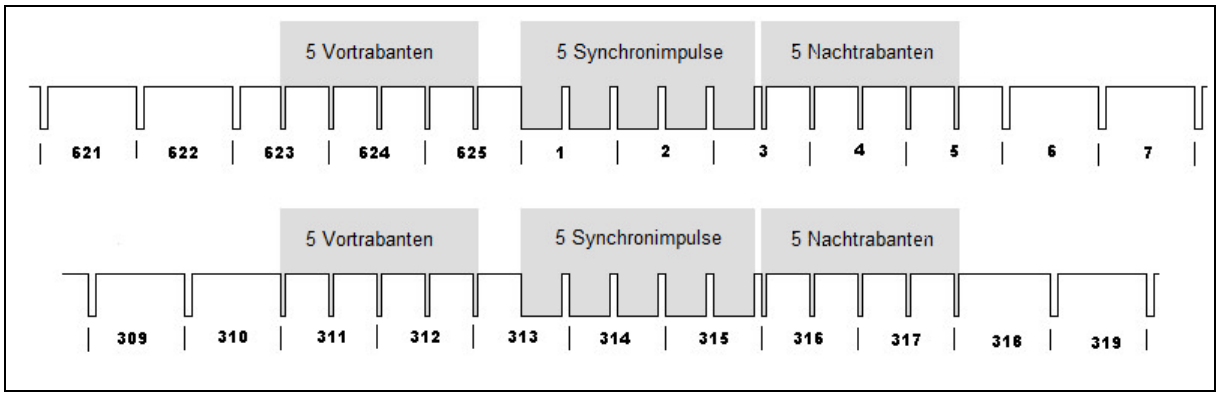
Dieser Schaltungsteil zählt den Pixeltakt hoch und decodiert aus dem Zählerstand verschiedene Steuersignale. Der Zählerstand ist der horizontalen Position des Elektronenstrahls auf der Bildröhre proportional.

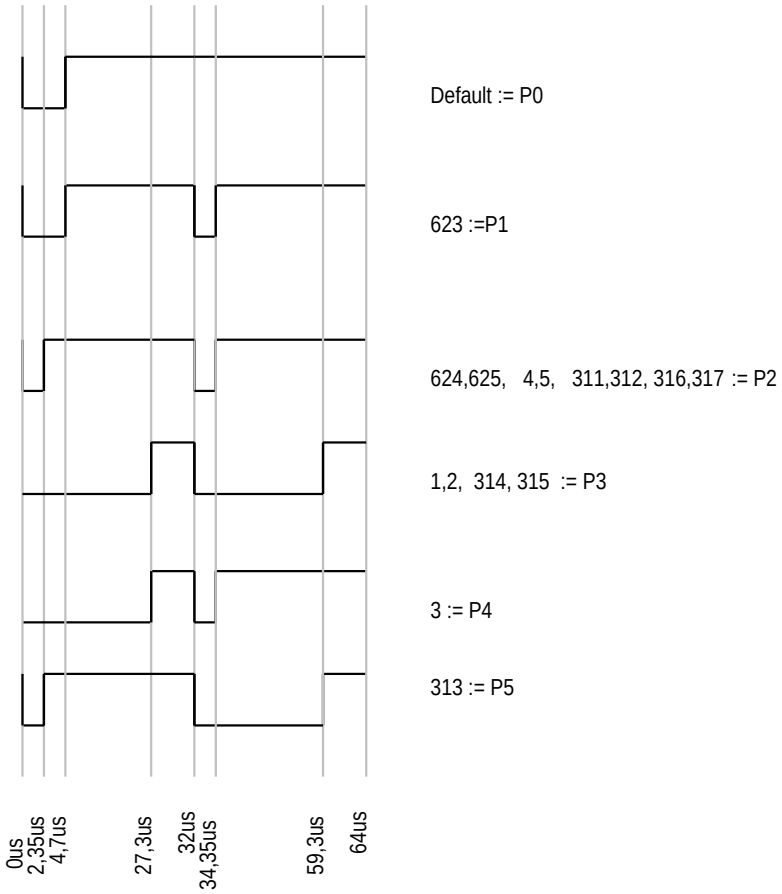
Es werden die folgenden Steuersignale ausgegeben:

hor_pix_count = 177 = 12us	markiert den Beginn des aktiven Bereichs der Zeile, in dem ein Bildinhalt dargestellt wird.
hor_pix_count = 944 = 64us	markiert das Ende der Zeile
hor_pix_count = 35 = 2,35us hor_pix_count = 69 = 4,7us hor_pix_count = 403 = 27,3us hor_pix_count = 472 = 32us hor_pix_count = 507 = 34,35us hor_pix_count = 874 = 59,3us	für Generierung der Sync-Patterns

Es soll zunächst hergeleitet werden, welche Steuersignale zum Erzeugen der veschiedenen Sync-Patterns benötigt werden.

Man erkennt, das 6 verschiedene Patterns, hier mit P0 bis P6 bezeichnet, vorkommen:





Diese Patterns sind in der obenstehenden Abbildung mit den dazugehörigen Zeiten und den Zeilen, in denen sie vorkommen, dargestellt.

Man erkennt, das sich alle 6 Pattern an einem oder mehreren der 7 Zeitpunkte 2,35us, 4,7us, 27,3us, 32us, 34,35us, 59,3us und 64us ändern.

Wenn man diese Zeitpunkte aus dem Zählerstand des Horizontal-Zählers decodiert, dann erhält man Ausgangssignale, die für die Länge einer Periode des Pixeltakts aktiv sind, wenn der entsprechende Zählerstand anliegt. Diese Signale können dann das Setzen oder Rücksetzen des Sync-Signals veranlassen. Hierbei werden diese Signale in Abhängigkeit von der aktuell anliegenden Zeile auf die Setz- bzw. Rücksetzeingänge für das Sync-Signal geschaltet.

Es soll noch der durch die Bildung der Zeiten mit dem (nicht exakt „passenden“) Pixeltakt entstehenden Quantisierungsfehler eingegangen werden:

Die Periode des Pixeltaltes 14.7456 MHz ist 0,06788168 us.

Zählerstand	tatsächliche Zeit	ideale Zeit	Abweichung	
35	2,374us	2,35us	+23,6ns	+1,0%
69	4,679us	4,7us	-20,6ns	-0,44%
177	12,00us	12,00us	+3,5ns	0%
403	27,33us	27,3us	+30,2ns	+0,1%
472	32,01us	32,00us	+9,5ns	+0,03%
507	34,38%	34,35us	+33,1ns	+0,1%
874	59,27us	59,3us	-28,1ns	-0,05%
944	64,02us	64,00us	+19,1ns	+0,03%

Man erkennt, das die beiden für die Synchronisation kritischen Zeiten, 32us und 64us mit einer besonders geringen Abweichung (gerade auch relativ zueinander) erzeugt werden. Die mit 64,02 us etwas zu höhe Zeitdauer einer Zeile ist ohne Belang, da dies durch den Monitor mühelos ausgeglichen werden kann. (Man erkennt, das eine exakte Darstellung der 32us-Zeitmarke nur mit einer geraden Anzahl an Pixeln pro Zeile möglich ist)

Mit den hintereinander geschalteten Synchronzählern U1101 und U1102 wird der Pixeltakt hochgezählt. Die Freigabeingänge /ENT und /ENP von U1101 sind fest auf L gelegt. Damit zählt der Zähler (sofern über S0 und S1 entsprechend eingestellt) mit jeder Periode des Pixeltaktes. Wenn U1101 überläuft, dann ist dessen Ausgang /RCO für eine Periode des Pixeltaktes auf L, womit dann U1102 bei der dann erfolgenden ansteigenden Flanke des Pixeltaktes aum Eins hochzählt.

Beim Erreichen des Zählerstandes 944 wird vom (bisher in diesem Text noch nicht beschriebenen) Decoder das Signal hor_pix_count=944=64us_unreg auf L gesetzt. In der Folge wird die Zählerkette aus U1101 und U1102 mit dem Wert 1 (nicht Null!) geladen. Das Rücksetzen auf Eins wurde vorgenommen, da in der Literatur die Pixel und Zeilen eines Videobildes im Allgemeinen von Eins an beginnend gezählt werden und nicht von Null an. Damit werden Verwechslungen nebeneinanderliegnder Zeilen und Pixel im Verlauf von Entwurf und Inbetriebnahme unwahrscheinlicher gemacht.

Die Steuereingänge der Zählerbausteine 74AS867 haben die folgende Wirkung:

S1	S0	
L	L	clear (asynchron)
L	H	count down (wenn gleichzeitig /ENT und /ENP auf L)
H	L	load
H	H	count up (wenn gleichzeitig /ENT und /ENP auf L)

Bei U1101 und U1102 ist der Eingang S1 fest auf H gelegt.

Damit vereinfacht sich die obige Tabelle zu:

S0	
L	load
H	count up (wenn gleichzeitig /ENT und /ENP auf L)

Im Resetzustand (#reset = L) wird der Zähler (da der Pixeltakt auch während des Resetzustandes durchäuft) mit dem Startwert 1 geladen. In diesem Fall ergibt sich am Ausgang des Gatters U1111A stets ein L-Pegel.

Bei einem Zählerstand von ungleich 944 ist das Signal hor_pix_count=944=64us_unreg auf L. Damit ergibt sich ein H-Pegel am Ausgang des Inverters U1101A und damit auch (#reset=H) am Ausgang des AND-Gatters U1111A. Damit sind die Eingänge S0 der Zählerkette auf H, womit die Zählerkette bei jeder ansteigendne Flanke des Pixeltaktes ihren Zählerstand um Eins erhöht.

Beim Erreichen des Zählerstandes 944 geht das Signal hor_pix_count=944=64us_unreg auf H. Damit geht der Ausgang des Inverters U1101A auf L, womit auch der Ausgang es AND-Gatters U1111A auf L geht. Damit sind die Eingänge S0 der Zählerkette auf L, womit die Zählerkette mit der dann folgenden ansteigenden Flanke des Pixeltaktes mit dem Wert 1 geladen wird.

Nun wird die Decoderschaltung beschrieben:

Die Schaltung ist auf einen möglichst geringen Bauteilaufwand hin ausgelegt. Mit den Decodern U1103, U1104 und U1105A werden die einzelnen „Stellen" des Zählerstandes, wenn man ihn als Hex-Zahl betrachtet, getrennt voneinander decodiert. Bei jedem Zählerstand ist von jedem der drei Decoder ist genau ein Ausgang auf L.

Die weitere Wirkungsweise des Decoders wird anhand des Signals hor_pix_count=944=64us_unreg beschrieben.
944 ist die Summe aus 3 * 256 + 11 * 16 + 0*1 = 768 + 176 + 0
Bei einem Zählerstand von 944 sind die folgenden drei Decoderausgänge auf L, während alle anderen Decoderausgänge auf H sind:

U1101 Ausgang Y0
U1104 Ausgang Y11
U1105 Ausgang Y3

Diese drei Ausgänge werden an die Eingänge eines 3-fach NOR-Gatters (U1108B) geführt. Der Ausgang dieses Gatters ist nur dann auf H, wenn alle drei Eingänge auf L sind, dies ist nur dann der Fall, wenn der Zählerstand 944 beträgt.

Die Decodierschaltungen für die anderen Steuersignale arbeiten sinngemäß mit den Gattern U1106A bis U1108A.

Die Laufzeit von der ansteigenden Flanke des Pixeltaktes bis zum Ausgang der soeben beschriebenen NOR-Gatter setzt sich wie folgt zusammen.

(Verzögerungszeiten max. über Temperatur, der jeweils größte Wert aus L zu H oder H zu L, Betrachtung des langsamsten Pfades)

74AS867	Clk = > Q	16ns max.
74F154	A..D zu Y	10,5ns max.
74F27	A zu Y	7ns max.
Summe		33,5ns max.

Eine Periode des Pixeltaktes dauert 67,8ns. Die auf die betrachtete Schaltung folgende kombinatorische Schaltung hat eine Laufzeit in ähnlicher Größenordnung wie die gerade betrachtete Decoderschaltung. Daher wird eine Registerstufe, aufgebaut mit dem Octal-Registerbaustein U1109, vorgesehen. Die damit entstandene Pipeline-Verzögerung von einem Takt muss an anderen Pfaden der Schaltung nachvollzogen werden.

Erzeugung des Vertikal-Timings

Die Zählerkette aus U1201 und U1202 sowie die nachfolgende Decoderschaltung mit U1203, U1204 und U1105B gibt Steuersignale aus, die jeweils für die Dauer einer Zeile auf H sind. Die Schaltung zählt dann um Eins hoch, wenn die die Pixel zählende Zählerkette aus U1101 und U1102 den Zählerstand 944 (also das letzte Pixel vor dem Zeilenumbruch) erreicht hat.

Mit dem Abschluss der letzten Zeile (625) wird der Zähler auf den Wert 1 zurückgesetzt. (Um mit der üblichen Notation für Zeilen im Videobild konform zu gehen ist die oberste Zeile die Zeile 1 und nicht die Zeile Null)

Um die Schaltung im Detail zu betrachten wird noch einmal die Wirkungsweise der Steuereingänge des 74AS867 dargestelt:

S1	S0	
L	L	clear (asynchron)
L	H	count down (wenn gleichzeitig /ENT und /ENP auf L)
H	L	load
H	H	count up (wenn gleichzeitig /ENT und /ENP auf L)

Bei U1201 und U1202 ist der Eingang S1 fest auf H gelegt.

Damit vereinfacht sich die obige Tabelle zu:

S0	
L	load
H	count up (wenn gleichzeitig /ENT und /ENP auf L)

Bei einem L-Pegel am Signal #reset ergibt sich am Ausgang eds Gatters U1111B stets ein L-Pegel. womit der Zähler mit dem Wert 1 geladen wird.

Für die folgende Betrachtung sei das Signasl #reset stets auf H.

Man erkennt das der Ausgang von U1214A nur dann auf L ist, wenn die Signale video_line_count=625_unreg und hor_pix_count=944=64us_ unreg gelichzeitig auf H sind. Dies ist nur dann der Fall, wenn gerade das letzte Pixel der letzten Zeile des zweiten Halbbildes (von beiden Zählerketten im Zusammenwirken) adressiert ist. Dieser L-Pegel am Ausgang von U1214A führt zu einem L-Pegel am Ausgang von U111B, womit dann der Zähler (bei der dann folgenden ansteigenden Flanke des Pixeltaktes) mit 1 geladen wird.

Wenn das Signal hor_pix_count=944=64us_unreg auf H ist, das Signal video_line_cont=625_unreg aber auf L ist, dann stellt sich am Ausgang von U1214A ein H-Pegel ein. Damit stellt sich auch an den Eingäng S0 der Zählerketten ein H-Pegel ein. Gleichzeitig werden über den Inverter U1110B die Eingänge ENT/ und ENP des ersten Zählers der Kette (U1201) und der Eingang /ENP des Zweiten Zählers der Kette (U1202) auf L gesetzt. Bei der dann folgenden ansteigenden Flanke des Pixeltaktes zählt die Zählerkette um Eins hoch.

Wenn das Signal hor_pix_count=944=64us_unreg auf L ist, dann sind die Eingänge /ENT/ENP von U1201 sowie /ENP von U1202 auf H, der Zählerstand ändert sich nicht, obwohl der Pixeltakt anliegt.

Der mit U1203, U1204,U1105B sowie U1206A..U1212B aufgebaute Decoder ist zu dem bereits beschriebenen Decoder für den Horizontalteil identisch, lediglich die Zahl der Ausgänge wurde erhöht. Das Fan-out der Bausteine 74F154 und 74F139 beträgt 33 und wird in dieser Schaltung mit hohen Reserven nicht ausgenutzt.

Erzeugung der Sync-Patterns

Die Signale #P0, P1..P5 steuern die Erzeugung der verschiedenen Pulsformen für die Erzeugung des CSYNC-Signals. Von diesen 6 Signalen ist jeweils ein Signal in Abhängigkeit von der aktuell anliegenden Zeile aktiv:

P1:	Zeile 623
P2:	Zeilen 624,625, 4,5, 311,312, 316,317
P3:	Zeilen 1,2, 314,315
P4:	Zeile 3
P5:	Zeile 313

#P0 ist immer dann aktiv, wenn keines der Signale P1..P5 aktiv ist, also in allen Zeilen mit „normalem" Sync-Pattern.

Das Signal P1 ist direkt der Decoderausgang video_line_count=623.
Das Signal P1 wird durch Veroderung der dazugehörigen Decoderausgänge erzeugt. Hierzu werden die Inverter U1101E/F, U1215A/B, U1215 E/F sowie U1216A/B und das NAND-Gatter U1218 verwendet.

Das Signal P3 wird auf identische Weise mit den Invertern U1110C/D und U1215C/D sowie dem NAND-Gatter U1219A erzeugt.

Die Signale P4 und P5 entsprechen den Decoderausgängen für die Zeilen 3 und 313.

Es soll nun die längste Laufzeit für diese Signale bestimmt werden. Diese ist für das Signal P3 (bis zum Ausgang des Gatters U1219A) wirksam:

(Verzögerungszeiten max. über Temperatur, der jeweils größte Wert aus L zu H oder H zu L, Betrachtung des langsamsten Pfades)

74AS867	Clk = > Q	16ns max.
74F154	A..D zu Y	10,5ns max.
74F27	A zu Y	7ns max.
74F04	A zu Y	6ns max.
74F20	A zu Y	6ns max.
Summe		45,5ns max.

Mit der Laufzeit der noch folgenden Logik ergäbe sich eine zu hohe Gesamtlaufzeit, die in der Größenordnung der Periode des Pixeltaktes von 67,8ns liegen würde. Daher wird eine Pipeline-Registerstufe mit dem Octal-Registerbaustein U1221 vorgesehen.

Die Set-Up-Zeit des verwendeten Registers 74F574 ist 2,0ns.

Das CSYNC-Signal wird mit der aus dem Multiplexer U1303 und dem Flipflop U1304A aufgebauten Statemachine erzeugt.

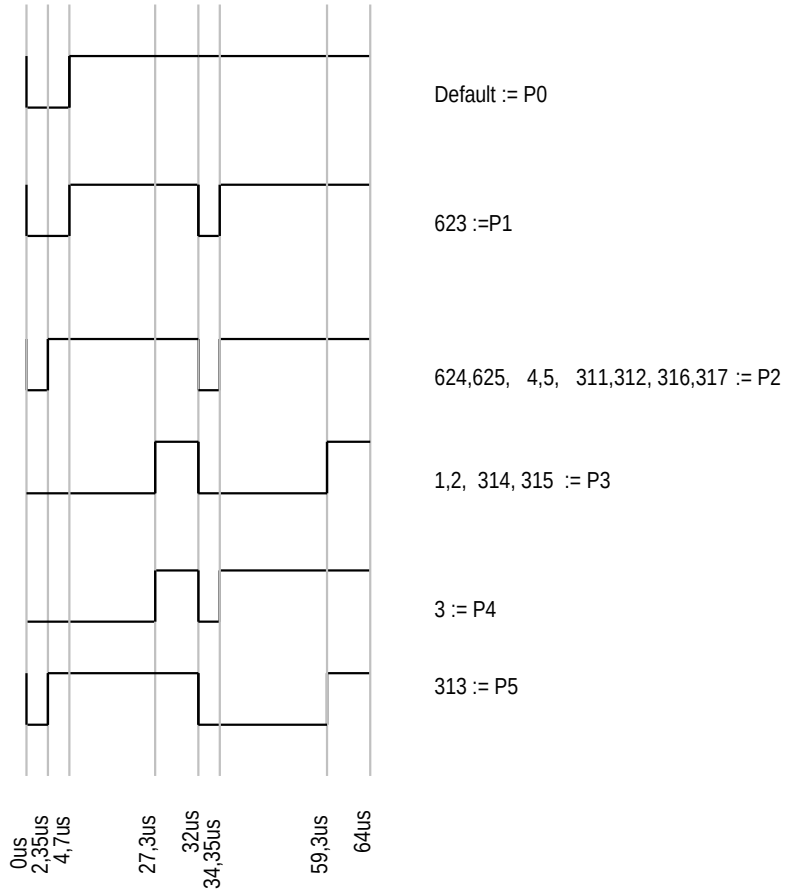
Der Eingang A des Multiplexers U1303 ist der Set-Eingang der Statemachine. Wenn dieser Eingang auf H ist, dann wird mit der folgenden ansteigenden Flanke des Pixeltakts die Statemachine gesetzt. Der Eingang A des Multiplexers U1303 ist der Reset-Eingang der Statemachine. Wenn dieser Eingang auf H ist, dann wird mit der folgenden ansteigenden Flanke des Pixeltakts die Statemachine rückgesetzt.

Die Funktion der Statemachine wird anhand der folgenden Wahrheitstabelle veranschaulicht:

U1303 B	U1303 A	gewählter Eingang U1303	U1303 Y (= Q (t+1)	Funktion
L	L	D0	Q	halten
L	H	D1	L	rücksetzen
H	L	D2	H	setzen
H	H	D3	Q	halten (Widerspruch)

Das am Multiplexerausgang U1303/Y anliegende Signal geht an den D-Eingang des Flipflops U1304A. Mit der dann folgenden ansteigenden Flanke des Pixeltaktes wird dieses Signal vom Flipflop übernommen und erscheint am Ausgang Q des Flipflops.

Aus der noch einmal wiedergegebenen Impulsdiagramm des CSYNC-Signals kann nun die Funktion der soeben beschriebenen Statemachine in Verbindung mit der ihr vorgeschalteten Logik beschrieben werden:



Es wurde bereits sichergestellt, das die Signale P0..P5 gegeneinander ausschliessend sind.

Wenn das Signal P0 aktiv ist, dann soll folgende Funktion ausgeführt werden:

Setzen der Statemachine zum Zeitpunkt 4,7us
Rücksetzen der Statemachine zum Zeitpunkt 64us.

Wenn das Signal P1 aktiv ist, dann soll folgende Funktion ausgeführt werden:

Setzen der Statemachine zum Zeitpunkt 4,7us
Rücksetzen der Statemachine zum Zeitpunkt 32us.
Setzen der Statemachine zum Zeitpunkt 34,35us
Rücksetzen der Statemachine zum Zeitpunkt 64us.

Wenn das Signal P2 aktiv ist, dann soll folgende Funktion ausgeführt werden:

Setzen der Statemachine zum Zeitpunkt 2,35us
Rücksetzen der Statemachine zum Zeitpunkt 32us.
Setzen der Statemachine zum Zeitpunkt 34,35us
Rücksetzen der Statemachine zum Zeitpunkt 64us.

Wenn das Signal P3 aktiv ist, dann soll folgende Funktion ausgeführt werden:

Setzen der Statemachine zum Zeitpunkt 27,3us
Rücksetzen der Statemachine zum Zeitpunkt 32us.
Setzen der Statemachine zum Zeitpunkt 59,3us
Rücksetzen der Statemachine zum Zeitpunkt 64us.

Wenn das Signal P4 aktiv ist, dann soll folgende Funktion ausgeführt werden:

Setzen der Statemachine zum Zeitpunkt 27,3us
Rücksetzen der Statemachine zum Zeitpunkt 32us.
Setzen der Statemachine zum Zeitpunkt 34,35us
Rücksetzen der Statemachine zum Zeitpunkt 64us.

Wenn das Signal P5 aktiv ist, dann soll folgende Funktion ausgeführt werden:

Setzen der Statemachine zum Zeitpunkt 2,35us
Rücksetzen der Statemachine zum Zeitpunkt 32us.
Setzen der Statemachine zum Zeitpunkt 59,3us
Rücksetzen der Statemachine zum Zeitpunkt 64us.

Umgekehrt kann man diese Funktionsbeschreibung nach den Zeitpunkten des Setzens und Rücksetzens auflösen:

Setzen zum Zeitpunkt 2,35us:	Wenn P2 oder wenn P5
Setzen zum Zeitpunkt 4,7us:	Wenn P0 oder wenn P1 aktiv
Setzen zum Zeitpunkt 27,3us:	Wenn P3 oder wenn P4 aktiv
Setzen zum Zeitpunkt 34,35us:	Wenn P1 oder wenn P2 oder wenn P4 aktiv
Setzen zum Zeitpunkt 59,3us:	Wenn P3 oder wenn P5 aktiv
Rücksetzen zum Zeitpunkt 32us:	Wenn P0 NICHT aktiv
Rücksetzen zum Zeitpunkt 64us	immer.

Diese Funktion wird mit der aus U1301A..D, U1302A..C und U1306 aufgebauten Logik implementiert. Die auf das Horinzontaltiming bezogenen Zeitinfomation hor_pix_count=... und die auf das Vertikaltiming bezogene Zeileninformation P... werden in der ersten Gatterebene (U1301A..D, U1302A/B) mit NAND-Gattern verundet. Beim Zusammentreffen der ausgewählten auf das Horizontale bezogenen Zeitinformation pix_count=... und der auf das Vertikel bezogenen Zeileninformation P.. geht der Asusgang des entsprechenen HAND-Gatters für die Dauer eines Pixels auf L. Wenn einer der Ausgänge der ersten Gatterebene auf L geht, dann geht das dazugehörige Ausgangssignal der zweiten Gatterebene (U1302C, U1306) auf H, womit die Statemachine in der bereits beschriebenen Weise über die Signale set_sm_csync und res_sm_csync gesetzt oder rückgesetzt wird.

Die zusammengesetzten Zeileninformationssignale #P0, P0+P1, P1+P2+P4, P2+P5,P3+P4, P3+P5 werden mit den Bausteinen U1212C, U1217B/F und U1222A..D durch einfache Veroderung erzeugt.

Es soll nun die Laufzeit durch die soeben beschriebenen Logikschaltungen und die zuvor beschriebene Statemachine betrachtet werden:

(Verzögerungszeiten max. über Temperatur, der jeweils größte Wert aus L zu H oder H zu L, Betrachtung des langsamsten Pfades)

U1221	74F574	Clk = > Q	10ns
U1212C	74F27	A = > Y	7ns
U1217F	74F04	A = > Y	6ns
U1301D	74F00	A = > Y	6ns
U1306	74F30	A = > Y	5,5ns
U1303	74F151	S = > Y	10,5ns
U1304A	74F74	tsu	3ns
Summe:			48ns

Die Taktperiode des Pixeltaktes ist 67,8ns, die Reserve ist noch hinreichend.

Weiterhin soll das Timing des CSNC-Signals an U1304A/Q in Bezug auf das ursprüngliche Horizontaltiming (Bezug: Zählerstand U1101/U1102) betrachtet werden.

1. Pipeline-Stufe: Register U1109 parallel zu Vertikalzähler U1201/U1202
2. Pipeline-Stufe: Flipflop U1304A.

Das CSYNC-Signal ist also dem als Referenz dienenden Zählerstand U1101/U1102 um 2 Pipeline-Stufen (= 2 Pixeltakte) nachfolgend.

Der exakte Zeitpunkt, zu dem die Signale P0..5 nach Zeilenbeginn umschalten ist unkritisch, da alle Sync-Patterns bis zum Zeitpunkt 2,35ns = 35 Pixeltakte miteinander identisch sind. Daher geht der Pipeline-Delay durch U1221 nicht in diese Betrachtung ein.

Zu einem späteren Zeitpunkt wird noch die Angleichung der Pipeline-Delays zwischen dem CSYNC-Signal und dem den Bildinhalt darstellendne Signal betrachtet.

Erzeugung des active_video Signals

Das active_video-Signal ist dann auf H, wenn Bildinhalte ausgegeben werden sollen, also im aktiven Bereich einer Zeile (12us..64us), dies aber nur, wenn es sich um eine zur Darstellung des Bildinhalts gedachten Zeile handelt, also eine Zeile, die sich ausserhalb der beiden Bereiche des Bildrücklaufs eines Frames befindet. Wenn das active_viedo-Signal auf L ist, dann wird die Ausgabe des Schwarzwertes erzwungen, der entsprechende Bereich der Zeile oder des Bildes also geblankt.

Die zur Bilddarstellung aktive Region innerhalb der Zeilen wird durch das Signal active_pixel angezeigt, das mit der aus U1305 und U1304B aufgebauten Statemachine gebildet wird. Der Aufbau dieser Statemachine entspricht der bereits beschriebenen Statemachine für das CSYNC-Signal.

Die Statemachine wird dann gesetzt, wenn horizontal die Zeit 12us (177 Pixel) erreicht ist (hor_pix_count = 177 = 12us) und beim Zeilenende (hor_pix_count = 944 = 64us) wieder rückgesetzt.

Die zur Bilddarstellung aktiven Zeilen innerhalb des Frames werden durch das Signal active_line angezeigt. Dieses Signal wird mit der aus U1401 und U1402A aufgebauten Statemachine erzeugt.

Die Statemachine wird dann gesetzt, wenn die Zeilen 27 oder 340 erreicht werden. (Verknüpfung mit U1213B). Die Statemachine wird dann rückgesetzt, wenn die Zeilen 307 oder 620 erreicht werden (Verknüpfung mit U1213B)

Die Signale active_Pixel und active_line werden mit dem Gatter U1111C verknüpft.

Die Signallaufzeiten für die zuletzt beschriebene Statemachine (active Line) ist:

(Verzögerungszeiten max. über Temperatur, der jeweils größte Wert aus L zu H oder H zu L, Betrachtung des langsamsten Pfades)

74AS867	Clk = > Q	16ns max.
74F154	A..D zu Y	10,5ns max.
74F27	A zu Y	7ns max.
74F32	A zu Y	6,6ns max.
74F151	S = > Y	10,5ns
74F74	tsu	3ns
Summe		53,6ns max.

Man erkennt eine gerade eben noch ausreichende Reserve von 67,8ns – 53,6ns = 14,2ns.

Die Signallaufzeit für die Statemachine für das Signal active_Pixel ist geringer.

Erzeugung des field-Signals

Hinweis: dieses Signal wird ggf. nicht benötigt, da stets 2 x 2 Pixel-Dots ausgegeben werden.

Das field-Signal wird mit der aus U1403 und U1402B aufgebauten Statemachine erzeugt. Die Funktion der Statemachine entspricht dem bereits Beschriebenen, die Signallaufzeit kürzer als die zuvor ermittelte Zeit von 53,6ns max. beim active_line-Signal.

Die Statemachine wird beim Erreichen von Zeile 312 gesetzt und beim Erreichen von Zeile 625 rückgesetzt, es ist also im zweiten Halbbild auf H. Es hat eine Pipeline-Verzögerung von 2 Pixeltakten zum Horizontal-Zählerstand.

Erzeugung des new-field-Signals

Das new_field-Signal geht dann für die Dauer einer Zeile auf H, wenn die Zeilen 312 oder 625 erreicht werden, also in der kommenden Zeile ein neues Halbbild beginnt. Das Signal entsteht durch Verknüpfung im Gatter U1213C. Es hat eine Pipeline-Verzögerung von einem Pixeltakt zum Horizontal-Zählerstand.

Erzeugung des Transfer-Signals

Das Signal transfer_en ist in der Austastlücke nach dem zweiten Halbbild, also vor Beginn eines neuen Frames, aktiv. Dieses Signal gibt das Nachladen des Textspeichers aus dem FIFO frei. Es wird mit der aus U1404 und U1405A aufgebauten Statemachine erzeugt. Es wird mit dem Erreichen der Zeile 620 gesetzt und mit dem Erreichen der Zeile 26 zurückgesetzt. Es hat eine Pipeline-Verzögerung von 2 Pixeltakten zum Horizontal-Zählerstand.

Erzeugung der horizontalen Leseadresse für den Textspeicher (pointer_char_read [5..0])

Ein Zeichen plus Abstand nimmt horizontal 9 Dots, das sind 18 Pixel, ein.
Die horizontale Leseadresse soll ab dem Zeitpunkt 12us = 177 Pixel nach jeweils dem Durchlaufen von 18 Pixeln um Eins hochgezählt werden, um das nächstfolgende Zeichen zu adressieren.

Es werden vom Character-ROM jeweils 8 Dots eines Zeichens parallel ausgegeben und ein einem Schieberegister serialisiert, wobei dann das neunte Dot (Teil des Abstandes) automatisch generiert wird. Daher wird keine horizontale Leseadresse für das Character-ROM benötigt.

Die Leseadresse pointer_char_read[5..0] wird im Zählerbaustein U1505 hochgezählt.

Die Funktion des Zählers wird durch seine Steuereingänge wie folgt bestimmt, wenn, wie es hier der Fall ist, der Eingang S1 fest an H liegt:

S0	
L	load
H	count up (wenn gleichzeitig /ENT und /ENP auf L)

Wenn das #reset-Signal auf L ist, dann wird der Zähler mit Null geladen. (Pixeltakt auch bei Reset durchlaufend) Für die weitere Betrachtung wird davon ausgegangen, das das Signal #reset stets auf H ist.

Wenn das Signal hor_pix_count = 944 = 64us auf H ist, dann wird der Zähler ebenfalls mit Null geladen. Damit wird der Zähler mit dem Ende jeder Zeile auf Null zurückgesetzt.

Wenn das Signal hor_pix_count = 944 = 64us auf L ist, dann zählt der Zähler immer dann um Eins hoch, wenn das Signal #incr_char_pointer_read auf L ist und eine ansteigende Flanke des Pixeltaktes auftritt. Dies ist vor Beginn eines neuen darzustellenden Zeichens der Fall.

Das Signal #incr_char_pointer_read wird mit dem Zählerbaustein U1501 erzeugt.

Um die Schaltung im Detail zu betrachten wird noch einmal die Wirkungsweise der Steuereingänge des 74AS867 dargestelt:

S1	S0	
L	L	clear (asynchron)
L	H	count down (wenn gleichzeitig /ENT und /ENP auf L)
H	L	load
H	H	count up (wenn gleichzeitig /ENT und /ENP auf L)

Wenn das Signal active_pixel auf L ist, also der aktive Bereich der Zeile noch nicht erreicht ist, stellt sich am Eingang S0 von U1501 ein L-Pegel ein und am Eingang S1 ein H-Pegel. Damit wird der Zähler mit dem Wert 17 geladen, der an seinen Eingängen „hardwired“ eingestellt ist. Der Ausgang RCO des Zählers geht damit auf H.

Zu diesem Zeitpunkt ist der Zähler U1505, der die Signale pointer_char_read[5..0] erzeugt, bereits definiert mit dem Erreichen des Endes der vorherigen Zeile zurückgesetzt.

Wenn der aktive Bereich der Zeile erreicht ist, dann stellt sich an active_line ein H-Pegel ein. Damit ergibt sich am Ausgang des AND-Gatters U1502A ein H-Pegel (RCO ist auf H, da Zähler mit 17 geladen), womit sich an S0 ein H-Pegel und an S1 ein L-Pegel einstellt. Damit zählt der Zähler (von 17 an beginnend) mit jeder steigenden Flanke des Pixeltakts um Eins nach unten.

Beim Erreichen des Wertes Null stellt sich ein L-Pegel am Ausgang RCO ein. Damit wird der Zähler erneut mit dem Wert 17 geladen. Der RCO-Ausgang steuert direkt das Signal #incr_char_pointer_read. Damit wird gleichzeitig die Character-Adresse mittels des Zählers U1505 um Eins erhöht.

Erzeugung der vertikalen Leseadresse für das Character-ROM (char_matrix_Y_[3..0])

Ein Zeichen plus Abstand nimmt vertikal 14 Dots, das sind 28 Zeilen pro Ganzbild, ein.
Pro Halbbild nimmt das Zeichen dann vertikal 14 Zeilen ein.

Aufgrund der Festlegung, das ein Dot aus 2 x 2 Pixeln besteht, wird in beiden Halbbildern ein identisches Videosignal ausgegeben.

Das Character-Rom wird vertikal mit den Signalen char_matrix_Y_[3..0] adressiert. Diese Signale stellen den Zählerstand des Zählers U1503 dar.

Die mit U1503 aufgebaute Zählschaltung wird nun im Detail beschrieben.

Wenn das Signal active_line auf L ist, dann ist gerade noch keine Zeile mit darzustellendem Bildinhalt anstehend. In diesem Fall stellt sich am Ausgang von U1502C ein L-Pegel ein, womit der Zähler mit dem Wert Null geladen wird. Damit wird an char_matrix_Y_[3..0] die Adresse Null ausgegeben. Weiterhin geht auch der Ausgang des Comparators U1504 auf H.

Wenn das Signal active_line auf H geht, also der darzustellende Bereich des Halbbildes beginnt, stellt sich am Ausgang des Gatters U1502C ein H-Pegel ein. Wenn nun, Am Ende der Zeile, das Signal hor_pix_count = 944 = 64us (für die Zeitdauer eines Pixeltaktes) auf H geht, erhöht sich der Zählerstand von U1503 um Eins. Die tatsächliche Umschaltung geschieht dann, wenn gerade die Austastlücke begonnen hat.

Sobald der Zählerstand 14 erreicht wird, befindet man sich schon in der obersten Zeile des nächstfolgenden Zeichens. Dies wird zum Zeitpunkt 64us (944 pixel) dieser Zeile mittels des Comparators U1504erkannt, da sich zu diesem Zeitpunkt der Zählerstand ändert. Die Eingänge B dieses Comparators liegen „hardwired“ auf dem Wert 14. Mit dem Erkennen von „14“ geht der Ausgang des Comparators auf L, womit auch der Ausgang des Gatters U1502C auf L geht und der Zähler erneut mit Null geladen wird. (Laden hat Prorität vor Zählen) Beim Erreichen des aktiven Bereichs der Zeile ist dann die Adresse „0“ stabil anstehend und es wird die oberste Zeile des zur nächsten Textzeile gehörenden Characters adressiert.

Der soeben beschriebene Comparatorausgang steuert auch das Signal #incr_text_line_pointer_read, so das beim Übergang von Matrixzeile 13 auf Matrixzeile Null die nächstfolgende Textzeile aus dem Textspeicher adressiert wird.

Das Signal #incr_text_line_pointer_read hat eine Dauer von einem Pixeltakt.

Erzeugung der vertikalen Leseadresse für den Textspeicher (pointer_text_line_read [4..0], mit Scroll-Offset)

Die Adresse pointer_text_line_read[4..0] entspricht dem Zählerstand des Zählers U1506. Die grundlegende Version der verwendeten Zählschaltung wurde bereits anhand der Schaltung zur Erzeugung der Adresse pointer_char_read[5..0] beschrieben.

Der Zähler wird mit einem H-Pegel am Signal new_field (Während der Zeilen 312 und 625 auf H) über den Tristate-Buffer U1508 mit dem Scroll-Offset geladen. Das bedeutet, das der Zähler zu Beginn jedes Halbbildes auf die Adresse des Textspeichers gesetzt wird, die in der obersten Textzeile des dargestellten Bildes erscheinen soll.

Wenn nicht gescrollt wird ist der Scroll-Offset Null und die im Textspeicher unter der Zeilenadresse Null abgelegte Zeile wird als oberste Zeile des Bildes dargestellt.

Der Zähler wird mit einem L-Pegel des Signals #incr_text_line_pointer_read um Eins hochgezählt, dieses Signal geht dann für einen Pixeltakt auf L, wenn die erste Matrixzeile einer neuen Textzeile beginnt, mit noch einigem Abstand vor dem aktiven Bereich dieser Zeile.

Am Ende der letzten Videozeile zur Darstellung der 19.Textzeile (=vorletzte Textzeile) geht der Zählerstand von 18 auf 19 über. Damit ist zu Beginn der Darstellung der ersten Videozeile der 20.Textzeile (=letzte Textzeile) der Ausgang des Comparators U1507 auf L. Am Ende der letzten Videozeile zur Datrstellung dieser Textzeile wird auch das Signal #incr_text_pointer_read L. Damit ist dann auch (für einen Pixeltakt) der Ausgang des OR-Gatters U1511A auf L. Damit wird über U1509 der Wert Null in den Zähler geladen.

Kurz darauf wird das Signal new_field aktiv und es wird der Schroll-Offset in den Zähler geladen, der ebenfalls Null ist. Damit hat dieser Vorgang in dem beschriebenen Fall keine Wirkung.

Nun wird der Fall betrachtet, das der Scroll-Offset den Wert Eins hat.

Wenn das Signal new_field aktiv wird, dann wird der Zähler mit dem Wert Eins geladen. Damit beginnt die Bilddarstellung mit der Zeile, die im Textspeicher unter der Zeilenadresse Eins abgelegt ist.
Mit dem Beginn der Darstellung der vorletzten Textzeile (19. Textzeile) ist bereits der Zählerstand 19 vorhanden. Mitd em Übergang auf die 20. Textzeile wird damit der Wert Null in den Zähler geladen, so das in dieser untersten Textzeile auf dem Bildschirm die Textzeile angezeigt wird, die im Textspeicher an der Zeilenadresse Null abgelegt ist.

Man erkennt weiterhin, das zu Beginn eines neuen fields bereits wieder der Scroll-Offset im Zähler steht, das nachladen des Scroll-Offsets, solange sich dieser nicht verändert, keine Wirkung hat.

Wenn der Schroll-Offset den Wert 2 hat, dann würde das Nullsetzen des Zähelrs bereits mit dem Beginn der vorletzten Textzeile wirksam.

Man erkennt, das das Zählmuster 0,1,2,3..18,19,0,1.. um den Scroll-Offset versetzt abläuft.

Das Widerstandsnetzwerk RN1501 verhindert, das sich dann, wenn beide Tristate-Buffer U1508 und U1509 hochohmig sind, undefinierte Pegel an den Eingängen des Zählers aufbauen.

Erzeugung des Videosignals (Gesamt-Timing)

Es wird nun die Erzeugung des Video-Signals in Bezug auf das Horizontal-Timing betrachtet.

Der Zählerstand von U1101 und U1101 (hor_pix_count) wird hierbei als Referenz genommen.

Die Flanke des Pixeltakts, die eine bestimmte Änderung an hor_pix_count verursacht wird als t=0 definiert.

Dieser Textabschnitt bezieht sich auf das Timing-Blockschaltbild im Schaltplan.

Für den Textspeicherund für den Zeichensatzspeicher wird eine Zugriffszeit von einem Pixeltakt angenommen (auch wenn die Speicher tatsächlich schneller sind)

Das active_Pixel-Signal entsteht mit einem Takt Verzögerung zum initialen Zählerstand. Die Statemachine zur Erzeugung des Signals incr_char_pointer_read bewirkt eine weitere Verzögerung um einen Takt. Ebenso entsteht im Zähler, der das Signal char_pinter_read[5..0] erzeugt eine weitere Verzögerung um einen Takt. Die aus dem Textspeicher ausgelesenen Daten sind damit um 4 Pixeltakte gegenüber hor_pix_count verzögert.

Durch die Statemachine zum Einfügen es Cursors und des Blankens auf Zeichenebene entsteht eine weitere Verzögerung um einen Pixeltakt, ebenso durch die Zugriffszeit des Zeichensatzspeichers. Das Ladesignal für das ausgangsseitige Schieberegister muss also um 6 Pixeltakte gegenüber hor_pix_count verzögert sein.

Das Videosignal (erstes Dot) seht somit 7 Pixeltakte nach hor_pix_count am Ausgang des Schieberegisters an.

Daher muss das CSYNC-Signal, das bereits eine Verzögerung von 2 Pixeltakten gegenüber hor_pix_count aufweist um weitere 5 Pixeltakte verzögert werden, um gleichzeitig mit dem Videosignal am Ausgangsbuffer einzutreffen.

Das active_video-Signal muss in gleichartiger Weise um 6 Pixeltakte verzögert werden.

Weiterhin soll der Video-Zeilenwechsel innerhalb einer Textzeile betrachtet werden:

Die Vertikaladresse am Zeichensatzspeicher ändert sich zweii Pixeltakte (t = +2p) nach dem Erreichen der Zeile 625 am Zähler U1101/U1102.

Damit wird die nächsttiefere Dot-Zeile des aktuell angezeigten Characters adressiert, aber noch nicht in das Schieberegister übernommen. Gleichzeitig mit der Übernahme ins Schieberegister erfolgt aber auch das Blanken des ausgehenden Videosignals.

Ebenfalls zum Zeitpunkt t = +3p erfolgt das Rücksetzen der Character-Adresse für den textspeicher auf den Zeilenbeginn.

Mit dem Beginn des aktiven Bereichs der kommenden Zeile wird, über die Statemachine initial load, die zur entsprechenden Videozeile gehörende Dotzeile des ersten Zeichens der Zeil ein das Schieberegister geladen.

Nun soll der Video-Zeilenwechsel bei gelichzeitigem Wechsel der Textzeile betrachtet werden:

Die Vertikaladresse am Textspeicher ändert sich zum Zeitpunkt t = +3p. Damit steht das letzte Zeichen der komemnden Zeile am Ausgang des Textspeichers an, wird aber nicht mehr angezeigt, da das Blanking-Signal zum Zeilenende aktiv wird.

Mit dem Beginn des aktiven Bereichs der kommenden Zeile wird, über die Statemachine initial load, die zur entsprechenden Videozeile gehörende Dotzeile des ersten Zeichens der Zeil ein das Schieberegister geladen.

Erzeugung des Videosignals (Schieberegister)

Das vom Zeichensatzspeicher kommende Signal dot[7..0] wird parallel in die Schieberegister U1602 und U1603 geladen und dann seriell aus diesen herausgetaktet. Dot 0 ist dabei der zuerst erscheinende Dot des Zeichens (an der linken Seite des Zeichens)

Das Zeichen selbst besteht horizontal aus 5 Dots. Der Abstand zwischen den Zeichen beträgt 4 Dots. Damit ergeben sich 9 Dots, der Zeichensatzspeicher gibt jedoch nur 8 Dots aus. Das letzte Dot wird daher gedoppelt, in dem der Ausgang QA des Schieberegisters U1602 an dessen Eingang zurückgeführt wird. (Mit dieser Methode ist auch die Darstellung größerer grafischer Elemente möglich, wenn man das „vorletzte" Dot auf hell setzt, dann wird der Abstandsraum zwischen den Zeichen vollständig überbrückt.

Die Beschaltung der Schieberegister 74F194 soll nun im Detail beschrieben werden.

Die Funktionsauswahleingänge S1 und S0 der Schieberegister werden über den Multiplexer U1601 (74F153) gesteuert. Hierbei gilt die folgende Wahrheitstabelle:

U1601 B	U1601 A	74F194 S1	74F194 S0	74F194 Funktion
L	L	L	L	hold
L	H	L	H	Shift right (SR => QA, QA => QB)
H	L	H	H	parallel load
H	H	H	H	parallel load

Wenn das Signal load_sr (an U1601/B) auf H ist, dann wird das Schieberegister (unabhängig vom Steuersignal an U1601/A) mit dem Ausgangswert des Zeichensatzspeichers geladen.

Wenn das Signal load_sr auf L ist und das Signal shift_sr an U1601/A auf H ist, dann wird der Inhalt der Schieberegister (mit der ansteigenden Flanke des Pixeltakts) um Eins nach rechts geschoben.

Da ein Dot aus zwei Pixeln besteht, soll das Schieberegister nur mit jeden zweiten Pixeltakt um Eins weitergeschoben werden. Hierzu wird der Eingang U1601/A (shift_sr) an das invertierte Signal hor_pix_count_0 gelegt.

Es wurde bereits gezwigt, das das Signal load_sr eine Pipeline-Verzögerung von 6 Pixeltakten gegenüber hor_count hat. Mit einer Verzögerung von 7 Pixeltakten gegenüber hor_count erschient dann das gerade parallel geladene erste Dot am Ausgang des Schieberegisters. Dieses Dot soll dann dort 2 Pixeltakte lang anstehen. Damit muss das Schieben des Registers wie in der Folge hergeleitet gesteuert werden

hor_count +6:	load_sr = H, shift_sr = x	letztes Dot voriges Zeichen (gedoppelt) an U1603/QD
hor_count +7:	load_sr = L, shift_sr = L	erstes Dot an U1603/QD
hor_count +8:	load_sr = L, shift_sr = H	erstes Dot an U1603/QD
hor_count + 9:	load_sr = L, shift_sr = L	zweites Dot an U1603/QD
hor_count +10:	load_sr = L, shift_sr = H	zweites Dot an U1603/QD
hor_count +11:	load_sr = L, shift_sr = L	drittes Dot an U1603/QD

Das Laden des Schieberegisters wird (mit der entsprechenden Pipeline-Verzögerung) durch das Signal #incr_char_pointer_read veranlasst. Dieses Signal wird jedoch nicht für das erste Zeichen einer Zeile erzeugt. Daher muss dieses Signal gesondert erzeugt werden und es wird mit dem (zuvor negierten) Signal #incr_char_pointer_read mit dem Gatter U1213C verodert.

Mit dem Flipflop U1410A wird das Signal active_pixel um einen Pixeltakt verzögert. Wenn das Signal active_pixel von L auf H übergeht, dann ist der Eingang D von U1410A auf H, während der Ausgang /Q von U1410A noch auf H ist, womit sich am Ausgang des AND-Gatters U1111C ein H-Impuls von der Dauer eines Pixeltakts ergibt. Dieser Impuls wird mit U1410B um einen Pixeltakt verzögert, damit er zum richtigen Zeitpunkt an den Schieberegistern eintrifft.

Die Registerstufe mit U1701A und U1701B vermeidet durch Laufzeitunterschiede verursachte Glitches am Videosignal.

Erzeugung des Videosignals (Analoge Ausgangsschaltung)

Es wird ein dreiwertiges Signal mit den Zuständen „Weiß“, „Schwarz“ und „Sync“ benötig. Deshalb benutzen wir die folgende Schaltung, die im Prinzip einem D/A-Wandler entspricht.

Verwendet wird hierbei der Baustein 74ABT125. An ihn werden die digitalen Signalausgänge SYNC und WEISS angelegt. Je nachdem ob sie HIGH oder LOW sind, schalten sie entweder ihren Ausgang auf Massepotential oder sind hochohmig.

Mit diesen Bausteinen wird nun ein steuerbarer Spannungsteiler aufgebaut. Der Zustand WEISS wird mittels des Gatters U1701A nur dann zugelassen, wenn der Zustand SYNC nicht aktiv ist.

Es ist sinnvoll, mit Hilfe des geschalteten Spannungsteilers zunächst ein Videosignal zu erzeugen, daß eine Amplitude von 2Vpp hat und Amplitudenwerte zwischen 1V und 3V hat. Damit kann dann ein mit +5V betriebener Pufferverstärker angesteuert werden. Dieser muß bei Verstärkungen von +1 stabil sein und eine ausreichende Bandbreite haben.
Das Videosignal wird mit der doppelten Amplitude des am Monitor benötigten Eingangspegels (1Vpp) ausgegeben, da hinter den Verstärkerausgang noch ein 75Ohm-Abschlusswiderstand geschaltet ist. Dieser bildet dann mit dem eingangsseitigen Abschlußwiderstand des Monitors von ebenfalls 75Ohm einen Spannungsteiler, womit sich dann der gewünschte Spannungspegel von 1Vpp ergibt. Die Ausgangsimpedanz des Operationsverstärkers ist idealisert Null, daher muß der Abschlußwiderstand in Serie zu diesem sein.

Die folgende Tabelle illustriert die möglichen Zustände:

SYNC	WEISS	Beschreibung	Spannung
0	0	Synchronimpuls	1,0V
0	1	Von Logik ausgeschlossen	
1	0	Schwarz	1,6V
1	1	Weiss	3,0V

Im Zustand „weiss“ sind beide Gatterausgänge des 74ABT125 hochohmig. Der Spannungspegel wird dabei von R1701 und R1704 bestimmt. R1704 wird auf 330 Ohm festgelegt, um dieEinflüsse von Schaltkapazitäten auf die Bandbreite hinreichend niedrig zu halten.

Damit fließt ein Querstrom von 3,0V / 330 R = 9,09 mA durch R1701 und R1704.
R1701 ist dann (5V – 3V) / 9,09mA = 220 R

Im Zustand „Schwarz“ ist der „obere“ 74ABT125 ausgangsseitig nach Masse durchgeschaltet.

Durch R1704 fließt dann ein Strom von 1,6V / 330R = 4,85 mA
Durch R1701 fließt dann ein Strom von (5V – 1,6V) / 220R = 15,45 mA.

Durch R1702 fließt dann ein Strom von 15,45mA – 4,85mA = 10,6mA
R1702 ist dann: 1,6V / 10,6mA = 150 R

Im Zustand „Sync“ sind beide 74ABT125 ausgangsseitig nach Masse geschaltet.

Durch R1704 fließt dann ein Strom von 1V / 330R = 3,03 mA
Durch R1702 fließt dann ein Strom von 1V / 150R = 6,66 mA
Durch R1701 fließt dann ein Strom von (5V – 1V) / 220R = 18,18 mA.
Durch R1703 fließt dann ein Strom von 18,18 mA – 3,03mA – 6,77 mA = 8,38mA
R1703 ist dann 1V / 8,38mA = 120 R

Text- und Zeichensatzspeicher

Allgemeines

Für den Text- und den Zeichensatzspeicher werden Dual-Port-RAMs des Typs IDT7134LA20PDG verwendet. Diese Speicher haben eine Zugriffszeit von 20ns und TTL-kompatible Eingänge.

Das Auslesen des Textspeichers

Der Textspeicher ist der Baustein U2101.
Der Speicher wird von der Seite L her aus dem FIFO heraus beschrieben und von der Seite R her synchron zum Video-Timing ausgelesen. Das Beschreiben des Speichers erfolgt nur in der Austastlücke vor dem Beginn eines neuen Frames.

Die Adressierung bezüglich der horizontalen Anordnung der Zeichen erfolgt mit den Signalen pointer_char_read[5..0], deren Erzeugung bereits beschrieben wurde. Diese Signale sind zum Video-Timing synchron.

Die Adressierung bezüglich der Textzeilen erfolgt mit den Signalen pointer_text_line_read[4..0], in denen bereits der Scroll-Offset vorhanden ist. Die Erzeugung der Signale pointer_text_line_read[4..0] wurde bereits beschrieben. Diese Signale sind zum Video-Timing synchron. Die Signale text_line_scroll_offset[4..0] werden in der Austastlücke, vor dem Beginn des neuen Frames, aktualisiert.

Die Erzeugung der Signale text_line_scroll_offset[4..0] wird in der weiteren Abfolge dieses Dokuments noch beschrieben.

Erzeugung der Signale pointer_text_line_write[5..0]

Mit diesen Signalen wird der Textspeicher für das Schreiben im Bezug auf die Textzeile adressiert.

An dieser Stelle wird eine synchrone Zählschaltung mit dem Baustein U2502 eingesetzt.

Wenn das Steuersignal #res_pointer_text_line_write aktiv ist, dann wird der Zähler mit der nächstfolgenden ansteigenden Flanke des Pixeltaktes mit Null geladen.

Wenn das Steuersignal #incr_pointer_text_line_write aktiv ist, dann wird der Zähler mit der nächstfolgenden ansteigenden Flanke des Pixeltaktes um Eins hochgezählt.

Beim Erreichen der untersten Textzeile (Zählerstand 19, enstspricht der 20-sten Zeile) geht das Signal #last_line auf L.

Die Funktion der Schaltung wird im Detail anhand der folgenden Wahrheitstabelle erklärt:

#incr	#res	dec	S1	S0	ENT/P	
0	0	0	1	1	1	idle
0	1	1	1	1	0	incr
1	0	2	1	0	1	load 0
1	1	3	1	1	1	idle

Man erkennt, das der Zähler nur dann zum Hochzählen freigegeben wird, wenn das Signal #incr_pointer_text_line_write aktiv ist und gleichzeitig das Signal #res_pointer_text_line_write inaktiv ist.

Ebenso wird der Zähler nur dann zum Laden (= rücksetzen) freigegeben, wenn wenn das Signal #res_pointer_text_line_write aktiv ist und gleichzeitig das Signal #incr_pointer_text_line_write inaktiv ist.

Das Signal #last_line wird durch Vergleichen des Zählerstandes mit dem „hardwired"-Wert 39 mit dem Comparator U2503 erzeugt.

Erzeugung der Signale pointer_char_write_[5..0]

Mit diesen Signalen wird der Textspeicher für das Schreiben im Bezug auf die Position des zu schreibenden Zeichens innerhalb der Textzeile adressiert.

An dieser Stelle wird eine synchrone Zählschaltung mit dem Baustein U2404 eingesetzt.

Wenn das Steuersignal #res_pointer_char_write aktiv ist, dann wird der Zähler mit der nächstfolgenden ansteigenden Flanke des Pixeltaktes mit Null geladen.

Wenn das Steuersignal #incr_pointer_char_write aktiv ist, dann wird der Zähler mit der nächstfolgenden ansteigenden Flanke des Pixeltaktes um Eins hochgezählt.

Wenn das Steuersignal #decr_pointer_char_write aktiv ist, dann wird der Zähler mit der nächstfolgenden ansteigenden Flanke des Pixeltaktes um Eins heruntergezählt.

Beim Erreichen der letzten Zeichenposition einer Zeile (Zählerstand 39, enstspricht dem 40-sten Zeichen) geht das Signal #last_char auf L.

Beim Erreichen der ersten Zeichenposition einer Zeile (Zählerstand 0, enstspricht dem 1-sten Zeichen) geht das Signal #first_char auf L.

Diese Signale werden mit den Comparatoren U2405 und U2406 erzeugt.

Die Funktion der Schaltung wird im Detail anhand der folgenden Wahrheitstabelle erklärt:

#incr	#decr	#res	dec	S1	S0	ENT/P	
0	0	0	0	1	1	1	idle
0	0	1	1	1	1	1	idle
0	1	0	2	1	1	1	idle
0	1	1	3	1	1	0	incr
1	0	0	4	1	1	1	idle
1	0	1	5	0	1	0	decr
1	1	0	6	1	0	1	load 0
1	1	1	7	1	1	1	idle

Die Eingangssignale werden mit dem Decoder U2401 decodiert. Das Enable-Signal wird durch eine Oder-Verknüpfung (in Bezug auf den L-Pegel) zweier Decoderausgänge mit dem AND-Gatter U2402A gebildet, damit wird der Zähler sowohl für das hinauf- als auch für das hinabzählen freigegeben.

Erzeugung der Signale text_line_scroll_offset_[3..0]

Mit diesen Signalen wird der für die Scrollfunktion benöigte Offset zwischen Lese- udn Schreibadresse generiert.

Die Schaltung mit dem Synchrinzähler U2602, dem Decoder U2601 und dem Comparator U2603 entspricht der bereits beschriebenen Schaltung für die Signale pointer_text_line_write[5..0].

Wenn das Steuersignal #res_pointer_text_line_offset aktiv ist, dann wird der Zähler mit der nächstfolgenden ansteigenden Flanke des Pixeltaktes mit Null geladen.

Wenn das Steuersignal #incr_pointer_text_line_offset aktiv ist, dann wird der Zähler mit der nächstfolgenden ansteigenden Flanke des Pixeltaktes um Eins hochgezählt.

Beim Erreichen des maximal möglichsten Offsets (Zählerstand 18, entspricht der 19 Zeilen) geht das Signal #max_offset auf L.

Einblendung von Funktionszeichen

Mit dem aus den Tristate-Buffern U2206, U2207 und U2208 gebildeten Multiplexer kann ausgewählt werden, ob der Zeichensatzspeicher mit dem aus dem Textspeicher ausgelesenem Zeichen, dem Leerzeichen (=Blanking) oder dem Cursorzeichen adressiert wird.

Wird die Notwendigkeit des Einblenden eines Blanks (von den noch im weiteren Verlauf des Textes beschriebenen Schaltungsteilen) erkannt, dann wird das Signal #display_blank auf L gesetzt. Damit sind die Ausgänge des Bausteins U2208 aktiv, an dessen Eingängenn „hardwired“ das SPACE-Zeichen (32d) anliegt. Die Ausgänge der Bausteine U2206 und U2207 sind hochohmig, da im beschriebenen Fall die Signale #display_char_mem_out und #display_Cursor auf H sind.

Wenn das Signal #display_cursor auf L ist wird über U2207 das Cursor-Zeichen (220d) durchgeschaltet.

Wenn das Signal #display_char_mem_out auf L ist, wird das am Ausgang des Textspeichers anstehende Zeichen durchgeschaltet.

Die mit dem Decoder und dem Gatter U1701B aufgebaute Schaltung stellt sicher, das immer nur eine der drei Auswahlleitungen auf L ist.

Wenn die Anforderung „#Cursor“ und „#Blank“ gleichzeitig vorhanden sind, dann wird durch die Wirkung des Gatters U1710B der Cursor angezeigt.

Erzeugung des Cursors

Es wird noch gezeigt, das die Signale zur Adressierung des Schreibvorgangs in den Textspeicher (nach Beendigung des Schreibvorgangs in der Austastlücke) immer auf die als nächstes zu beschreibende Zeichenposition zeigen.

Dies ist die Position des Cursors.

Immer dann, wenn während des Auslesens die Gleichheit der gerade ausgelesenen Zeichenposition mit der für den nächsten Schreibvorgang vorgesehenen Zeichenposition festgestellt wird, muss also das Cursor-Zeichen (Cursor-Block 220d) in den Character-Datenstrom eingeblendet werden.

Durch ein externes Taktsignal von ca. 2Hz kann weiterhin zwischen dem Cursor und dem Blank-Zeichen umgeschaltet werden, womit der Eindruck eines blinkenden Cursors entsteht.
Die Erzeugung dieses Signals wird an anderer Stelle dieses Dokuments noch beschrieben.

Die Gleichheit der Lese- und Schreibpointer auf Zeichenebene (innerhalb einer Zeile) wird mit dem Comparator U2104 festgestellt.
Die Gleichheit der Lese- und Schreibpointer auf Textzeilenebene wird mit dem Comparator U2105 festgestellt.
Nur dann, wenn beide Comparatoren Gleichheit erkennen ist auch der Ausgang des OR-Gatters U2108A auf L.

Wenn gleichzeitig das Signal cursor_on auf H ist, dann ergibt sich am Ausgang des Gatters U2108C ein L-Pegel, der (über die bereits beschriebne Schaltung mit dem Decoder U2209B) zur Anzeige des Cursor-Signals führt.

Wenn dagegen, bei gleichzeitigem L-Pegel am Signal #cursor, das Signal cursor_on auf H ist, dann ergibt sich ein L-Pegel am Ausgang des Gatters U2108B, der zum Anzeigen eines Leerzeichens führt.

Es soll nun die Generierung des Steuersignals cursor on beschrieben werden:

Wenn das vom Businterface kommende Signal cursor_en auf H ist, blinkt der Cursor mit der durch den mit U2611A aufgebauten Oszillator vorgegebenen Frequenz von ca. 2Hz. Wenn das Signal cursor_en auf L ist, dann ist der Cursor unsichtbar.

Die Ausführung des Oszillators als RC-Oszillator erlaubt eine einfache Anpassung der Frequenz an den visuellen Eindruck in tatsächlichen Einsatz.

Sowohl das Ausgangssignal des Oszillators auch das vom Businterface kommende Signal cursor_en sind zum Pixeltakt asynchron. Daher wird das verknüpfte Signal am Ausgang von U1701C mit den Flipflops U2610A/B auf den Pixeltakt synchronisiert. Um Defler durch metastabile Zustände zu vermeiden werden zwei Fliplfops kaskadiert.

Blanken von Zeichen nach der Cursorposition

Zeichenpositionen, die sich nach der Cursorposition befinden sind mit undefinierten Inhalten gefüllt und sollen daher geblankt werden.

Hierzu ist die mit den Bausteinen U2201 und U2202A aufgebaute Statemachine vorgesehen.

Diese Statemachine besitzt die folgende Wahrheitstabelle:

#cursor B	new_field A	in		Funktion
0	0	0	VCC	set
0	1	1	GND	res
1	0	2	Q	hold
1	1	3	GND	res

Man erkennt, das die Statemachine mit der auf das Erkennen der Cursorposition folgenden ansteigenden Flanke des Pixeltaktes gesetzt wird, womit dann ein Blank-Zeichen eingeblendet wird. Aufgrund der Wirkung des Gatters U1701B wird jedoch trotzdem über die gesamte Zeichenposition der Cursor angezeigt.

An allen danach folgenden Zeichenpositionen wird dann ein Leerzeichen angezeigt, bis die Statemacheine mit der auf das Aktivwerden des Signals new_field folgenden ansteigenden Flanke des Pixeltakts wieder zurückgesetzt wird. Das Signal new_field kennzeichnet den Beginn eines neuen Halbbildes mit der obersten Textzeile.

Es wird dann ein Blank-Zeichen eingeblendet, wenn der Ausgang /Q von U2202A auf L ist.

Blanken von Zeichen in unvollständig beschriebenen Zeilen.

Mit dem Comparator U2107 wird ein LF-Zeichen am Ausgang des Textspeichers erkannt (Vergleich mit hardwired 10d)

Wenn ein LF-Zeichen erkannt wird, geht das Signal #cahr = LF auf L. Mit der dann folgenden ansteigenden Flanke des Pixeltakts wird die mit U2201 und U2202B aufgebaute Statemachine gesetzt, womit die Einblendung eines Blank-Zeichens veranlasst wird.

Mit dem Beginn des Auslesens einer neuen Zeile wird die Statemachine mit der auf das Aktivwerden des Signals #incr_text_line_pointer_read folgenden ansteigenden Flanke des Pixeltaktes zurückgesetzt.

Mit dem AND-Gatter U2204A werden die drei Bedingungen zum Einblenden eines Blank-Zeichens verodert.

Steuerung des Schreibens in den Textspeicher

Die von der CPU kommenden Zeichen werden von dieser in ein FIFO (U3101) geschrieben. Der Inhalt dieses FIFOs wird in der Austastlücke (vor einem neuen Frame) in den Textspeicher übertragen.

Hierbei wird zwischen den darstellbaren Zeichen und den Steuerzeichen unterschieden.

Die Wirkung der Steuerzeichen wird im Folgenden beschrieben. Der Umfang der Steuerzeichen und die Komplexität iherer Wirkung wurde gegenüber tatsächlich existierendenden historischen Terminals stark reduziert, auf das unbedingt notwendige reduziert. Das verlagert einen Teil des Aufwands, erlaubt es aber auch die Video-Karte mit einfachen TTL-Bausteinen in noch vertretbarer Zeit aufzubauen. (Im VT-100 werden die Funktionen der Steuerzeichen mit einem 8080-Microprozessor implementiert)

Steuerzeichen LF:

Dieses Zeichen wird im Textspeicher abgelegt (um das Blanken des undefinierten auf LF folgenden Teils der Textzeile zu steuern). Es bewirkt weiterhin, das der Schreibpointer zum Anfang der Folgezeile springt. Das dann folgende Zeichen wird also an die erste Position der Folgezeile geschrieben.

Steuerzeichen CLS:

Dieses Zeichen wird nicht im Textspeicher abgelegt. Es bewirkt, das der Schreibpointer auf das erste Zeichen der obersten Textzeile gesetzt wird. Das dann folgende Zeichen wird also in der ersten Zeichenposition der obersten Zeile abgespeichert.

Aufgrund der bereits beschriebenen Blanking-Steuerung erscheint bis zum Schreiben des ersten Zeichens ein leerer Bildschirm mit dem Cursor an der ersten Zeichenposition der obersten Zeile.

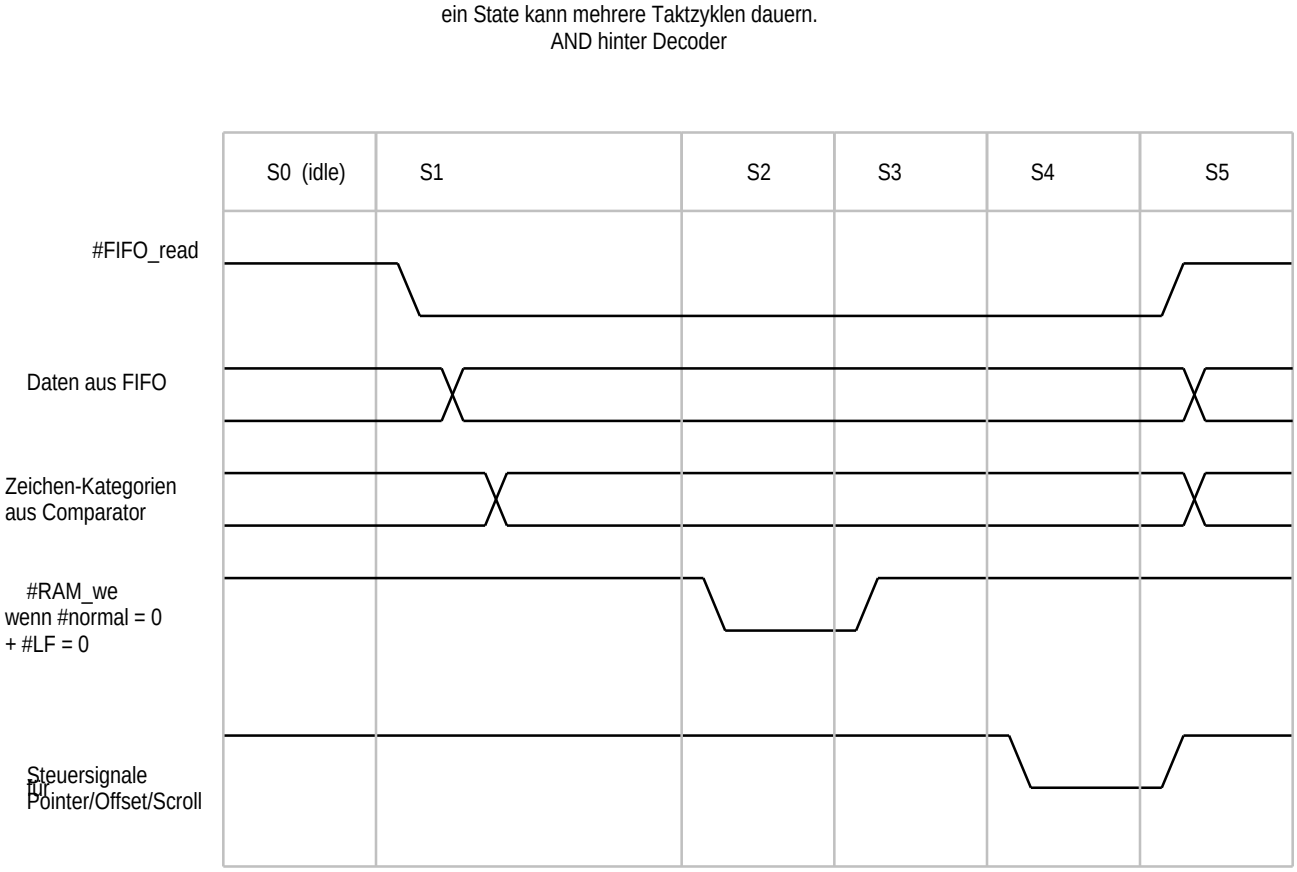
Steuerzeichen BS:

Dieses Zeichen wird nicht im Textspeicher abgelegt. Es bewirkt, das der Schreibpointer um eine Zeichenposition zurückgesetzt wird, wenn er sich nicht schon auf der ersten Zeichenposition der Zeile befindet. (Ein Zurückgehen über die Zeilengrenzen hinweg ist bewusst ausgeschlossen)

Diese Funktionen werden mit einer Statemachine implementiert.

Man kann alle benötigten Abläufe als linearen Ablauf der 6 Zustände S0 bis S5 implementieren. In einigen Abläufen geschieht dann in einigen Zuständen „nichts“, sie werden „leer“ durchlaufen. Dies stört aber nicht, da diese Abläufe nur selten vorkommen und da ohnehin ausreichend Zeit in der Austastlücke vorhanden ist, um die Zeichen aus dem FIFO in den Textspeicher zu übertragen.

Das folgende Impulsdiagramm visualiert diese Vorgehensweise:



Nomalerweise befindet sich die Statemachine im Zustand S0, in dem alle ihre Ausgangssignale den inaktiven Zustand einnehmen, es wird also nicht aus dem FIFO gelesen und nicht in den Textspeicher geschrieben.

Wenn das Signal transfer_en, das die Austastlücke vor Beginn eines neuen Frames anzeigt und das Signal #FIFO_EF (dessen H-Pegel anzeigt, das Daten im FIFO „abholbereit“ sind) gelichzeitig auf H sind, dann sollen Daten vom FIFO in den Textspeicher übertragen werden. Dazu geht die Statemachine dan in den Zustand S1 über.

Im Zustand S1 blockiert die Statemachine die weitere Wirkung der Signale #FIFO_EF und transfer_EN, um sicherzustellen, das der aktuelle Lese-Schreib-Zyklus auch dann vollständig zuendegeführt wird, wenn eine der auslösenden Bedingungen entfällt. (das ist beim Auslesen der letzen Speicherstelle im FIFO oder beim Beenden der Austastlücke durch das Video-Timing der Fall, im zweiten Fall entsteht keine gestörte Anzeige, weil das Beenden des Zyklusses wesentlich weniger Zeit als die noch geblankte Schwarzschulter des Videosignals benötigt)

Weiterhin wird das FIFO ausgelesen, womit das im FIFO stehende Zeichen an den Comparatoren für die Erkennung von Steuerzeichen und am Dateneingang des Textspeichers anliegt.
Zum Ende des Zustandes S1, der drei Pixeltakte andauert, stehen die aus den möglicherweise erkannten Steuerzeichen abgeleiteten Auswahlsignale für die in den nachfolgenden Zyklen auszuführenden Aktionen an.

Der Zustand S1 geht in den Zustand S2 über. Wenn im Zustand S1 erkant wurde, das es sich um ein Zeichen handelt, das im Textspeicher gespeichert werden soll (darstellbares Zeichen oder LF), dann geht das Schreibsignal für den Textspeicher in diesem Zustand auf L.

Wenn ein Zeichen anliegt, das nicht gespeichert werden soll (Steuerzeichen CLS oder BS), dann geschieht im Zustand S2 keine Aktion.

Der Zustand S2 geht in den Zustand S3 über. Wenn im Zustand S2 ein Zeichen in den Textspeicher gespeichert werden sollte, dann wird im Zustand S3 das Schreibsignal für den Textspeicher wieder auf H gesetzt, womit das tatsächliche Speichern ausgelöst wird.

Der Zustand S3 geht in den Zustand S4 über. Im Zustand S4 werden die Steuersignale für die Adresspointer (Zeichenposition, Textzeile, Scroll-Offset) in Abhängigkeit vom in S1 erkannten Zeichen ausgegeben.

Der Zustand S4 geht in den Zustand S5 über. Im Zustand S5 werden die in Zustand S4 aktivierten Steuersignale wieder in den Ruhezustand versetzt. Ebenso wird das Lesesignal für das FIFO wieder in den inaktiven Zustand gesetzt.

Der Zustand S5 geht in den Zusatnd S0 über. Damit erfolgt eine erneute Abfrage der Signale #FIFO_EF und transfer_en. In der Folge wird ein neuer Transfer durch den Übergang in S1 eingeleitet oder die Statemachine verbleibt im Ruhezustand S0.

Nun soll die Funktion der Steuersignale in Abhängigkeit von den erkannten Zeichentypen genauer beschrieben werden. Um dies zu vereinfachen wird das Signal #char=normal vorgesehen, das bei allen Zeichen ausser LF,BS und CLS aktiv ist.

Die folgende Beschreibung der Funktion bezieht sich auf den Zustand S4.
In allen anderen Zustaänden sind diese Signale inaktiv.

<u>#incr_pointer_char_write:</u>
Weitergehen zur nächsten Zeichenposition.

aktiv wenn char = normal und /last_char

Das in diesem Zyklus ausgelesene und abgespeicherte Zeichen ist ein normales Zeichen und es ist noch nicht die letzte Zeichenposition in der Zeile erreicht.

<u>#res_pointer_char_write:</u>
Springe zur ersten Zeichenpositin am Anfang der Zeile.

aktiv wenn char = normal und last_Char

oder char = LF

oder char = CLS

Das in diesem Zyklus ausgelesene und abgespeicherte Zeichen ist ein normales Zeichen und es ist die letzte Zeichenposition in der Zeile erreicht.

oder

Das in diesem Zyklus ausgelesene Zeichen ist LF oder CLS

#decr_pointer_char_write:
Gehe um eine Zeichenposition zurück.

aktiv wenn char = BS und /first_char

Das in diesem Zyklus ausgelesene Zeichen ist BS und man befindet sich nach er ersten Zeichenposition der Zeile

(Im Fall, das man sich auf der ersten zeichenposition der Zeile befindet soll keine Aktion erfolgen.)

#incr_pointer_text_line_write:
Gehe um eine Zeile nach unten.

aktiv wenn char = normal und last_char

oder

char = LF

Es wurde ein „normales“ Zeichen in die letzte Zeichenposition einer Zeile geschrieben, die nicht die unterste (=letzte) Zeile ist.

oder

Es wurde das Zeichen LF erkannt und in eine beliebige Zeichenposition geschrieben, die Zeile ist nicht die unterste (letzte) Zeile.

#res_pointer_text_line_write:
Gehe zur obersten (=ersten) Zeile.

aktiv wenn char = normal und last_line und last_char

oder

char = LF und last_line

oder

char = CLS

Es wurde ein „normales“ Zeichen in die letzte Zeichenposition der letzten Zeile geschrieben.

oder

Es wurde das Zeichen LF erkannt und in eine beliebige Zeichenposition der letzten Zeile geschrieben.

oder

Es wurde das Zeichen CLS erkannt.

Set_scroll:
Erkennen der Notwendigkeit des Scrollens (überlauf des Bildschrims)

char = normal *
last_char *
last_line

oder

char = LF *
last_line

Reset_scroll:
Rücksetzen des Scroll-Modus

char = CLS

#incr_pointer_text_line_offset:

Beginne das Auslesen des Textspeichers um eine Textzeile tiefer.

aktiv wenn #max_offs und (

char = normal *
last_char *
last_line

oder

char = LF *
last_line

oder

char = normal *
last_char *
scroll

oder

char = LF *
scroll

Es wurde ein „normales“ Zeichen in die letzte Zeichenposition einer Zeile geschrieben, die die unterste (=letzte) Zeile ist und der maximale Scroll-Offset wurde noch nicht erreicht.

Es wurde ein „LF Zeichen erkannt und in die die unterste (=letzte) Zeile geschrieben und der maximale Scroll-Offset wurde noch nicht erreicht.

Es wurde ein „normales“ Zeichen in die letzte Zeichenposition einer beliebigen Zeile geschrieben, das Scroll-Flag wurde gesetzt und der maximale Scroll-Offset wurde noch nicht erreicht.

Es wurde ein „LF Zeichen erkannt und in eine beliebige Zeile geschrieben, das Scroll-Flag wurde gesetzt und der maximale Scroll-Offset wurde noch nicht erreicht.

#res_pointer_text_line_offset:

Beginne das Auslesen des Textspeichers an der obersten Textzeile.

aktiv wenn max_offs und (

char = normal *
last_char *
last_line

oder

char = LF *
last_line

oder

char = normal *
last_char *
scroll

oder

char = LF *
scroll

oder

char = CLS

Nun soll die Schaltung im Detail betrachtet werden:

Der Baustein U3101 ist der FIFO mit einer Kapazität von 4096 Byte (die weit über dem an dieser Stelle Benötigten liegt)

Mit den Bausteinen U3102A/B und U3103A/B wird erreicht, das die besonderen Bedingungen für das Resetten des FIFOs eingehalten werden. Die Signale /W und /R müssen während des Resettens und noch eine defineirte Zeit nach dem Resetten auf H sein. (Ist dies nicht der Fall, ist die Initialisierung der Pointer des FIFOS nicht eindeutig gesichert)

Wenn das Signal #reset auf L ist, dann ist der Ausgang /Q des Flipflops U3102B auf H, womit dann auch über die Gatter U3103A und U3103B die Signale /W und /R des FIFOs auf H sein. Wenn das Signal #reset auf H geht, dann geht mit der dann folgenden ansteigenden Flanke des Pixeltaktes der Ausgang Q des Flipflops U3102A auf H. Mit der dann weiter folgenden Flanke des Pixeltaktes geht der Ausgang /Q des Flipflops U3102B auf L, womit dann über die Gatter U3103A/B die Möglichkeit freigeschaltet wird, das die Signale /W und /R auf L gesteuert werden können.

Die Beschaltung der Steuereungänge /XI und /FL/RT erfolgte nach den Angaben im Datenblatt für die einfachste, unkaskadierte Anwendung des Bausteins.

Mit den Comparatoren U32303, U3204 und U3205 werden die Steuerzeichen LF, CLS und BS erkannt. Hierbei werden die aus dem FIFO ausgelesenen Zeichen mit den folgenden „hardwired“ Mustern verglichen:

LF (Line-Feed) 10d
CLS (Clear Screen) = FF (Form Feed) = 12d
BS (Backspace) = 8d

Beim Erkennen eines der genannten Steuerzeichen geht der Ausgang des entsprechenden Comparators auf L. Um die Auslegung der Steuerlogik zu vereinfachen wird mit dem NAND-Gatter U3207B das Signal #char=normal generiert, das immer dann auf L ist, wenn das anliegende Zeichen weder LF, CLS oder BS ist.

Mit dem Synchronzähler U3201 werden die Zustände S0 bis S5 der Statemachine für die Steuerung des Schreibens in den Textspeicher hochgezählt. Der Zustand S1 nimmt hierbei drei Taktperioden ein. Der Zähler wird mit dem Pixeltakt getaktet. Damit ergibt sich der Vorteil, nur einen einzigen Takt im System zu haben (Ausnahme: der über das genau dafür konstruierte FIFO sicher abgetrennte asynchrone Buszugriff der CPU)

Ausserhalb der für den Transfer vom FIFO zum Textspeicher vorgesehenen Austastlücke ist das Signal transfer_en auf L. Das Signal #FIFO_EF ist dann auf L, wenn keine noch zu übertragenden Daten mehr im FIFO stehen.

In den beiden genannten Fällen ist der Ausgang des Gatters U3206A auf L.
Es wird weiter angenommen, die Statemachine befinde sich im Ruhezustand S0. (Die Begründung für diese Annahme wird noch gegeben) Dann ist das Signal #S0 auf L, womit auch am Ausgang des Gatters U3103C ein L-Pegel ansteht. Damit ergibt sich auch (unabhängig vom Pegel am Signal #S5) ein L-Pegel am Ausgang des Gatters U3206B. Damit liegt an den Eingängen S1/S0 des Zählers U3201 die Wertekombination 1/0 an. Damit wird der Zähler bei der nächstfolgenden ansteigenden Flanke des Pixeltakts mit dem „hardwired“ an seinen Eingängen anliegendne Wert Null geladen. Dies entspricht dem Zustand S0, das Ausgangssignal #S0 des den Zählerstand decodierenden Decoders U3202 ist damit auf L, womit die eingangs gemachte Annahme begründet ist.

Man erkennt, das dieser Zustand stabil ist und so lange anhält, bis sich durch die Gleichzeitigkeit eines H-Pegels an den Signalen #FIFO_EF und transfer_en („Zeichen vorhanden“ und gleichzeitig „Austastlücke“) ein H-Pegel am Ausgang des Gatters U3206A ergibt, der auch am Ausgang von U3103C ansteht. Da der Zähler den Zählerstand Null hat, ist das Signal #S5 auf H, womit sich auch am Ausgang von U3206B ein H-Pegel einstellt. Damit liegt an den Eingängen S1/S0 des Zählers U3201

die Wertekombination 1/1 an. Damit zählt der Zähler bei der nächstfolgenden ansteigenden Flanke des Pixeltakts um Eins hoch.

Damit geht das Signal #S0 auf H. Durch das sich Einstellen eines L-Pegels an den Signalen #FIFO_EF (Letztes Zeichen wurde gelesen) oder transfer_en (Austastlücke vorbei) wird der Pegel am Ausgang von U3103C nicht mehr beeinflusst, der H-Pegel bleibt anstehend, womit der begonnene Transfer-Zyklus abgeschlossen wird. (Ausreichend Zeitreserve zum Beginn der aktiven Region der ersten Zeile ist vorhanden)

Mit jeder ansteigenden Flanke des Pixeltaktes zählt der Zähler um Eins hoch. Nch 7 Takten ist der Zustand S5 erreicht, das Signal #S5 geht auf L. Damit wird, unabhängig von den Signalen #FIFO_EF und transfer_en ein L-Pegel am Ausgang von U3206B erzwungen. Damit wird der Zähler mit der dann folgenden ansteigenden Flanke des Pixeltaktes wieder mit dem Wert Null geladen.

Damit stellt sich erneut ein L-Pegel am Signal #S0 ein, womit dann der Pegel am Ausgang des Gatters U3206A (Gleichzeitigkeit von Austastlücke und Vorhandensein eines zu übertragenden Zeichens) daüber entscheidet, ob die Statemachine im Ruhezustand S0 verbleibt oder einen neuen Transferzyklus beginnt.

Ein Transfer-Zyklus dauert 8 * 67,8ns = 0,542us.
In einer Video-Zeile können daher 118 Zeichen übertragen werden.
Die Austastlücke dauert 32 Zeilen.
Damit können in der Austastlücke 3775 Zeichen übertragen werden.
Der Inhalt des Bildschirms beträgt 20 x 40 Zeichen, das sind 800 Zeichen.
Man erkennt eine erhebliche Reserve.

Der Zustand S1 dauert 3 Taktyklen an, womit die Laufzeiten der kombinatorischen Logik zur Erkennung der Steuerzeichen berücksichtigt wird.

Mit dem Gatter U2314C wird erreicht, das das Signal #FIFO_read in den Zustaänden S1,S2,S3 und S4 auf L ist.

Mit den Gattern U3214A und U4314B wird das Schreibsignal für den Textspeicher erzeugt. Das Signal #ram_we ist im Zustand S2 aktiv, wenn entweder ein „normales“ Zeichen oder ein LF-Zeichen anliegend ist.

Die Schaltungen zur Erzeugung der weiteren Steuersignale sind eine direkte Umsetzung der bereits im vorstehenden Text dargestellten Verknüpfungen.

Die Statemachine zur Generierung des Scroll-Flags ist mit den Bausteinen U4304 und U4302B aufgebaut. Das Funktionsprinzip dieser Schaltung wurde bereits beschrieben.

Es sollen nun die Laufzeiten genauer betrachtet werden.

Die Zugriffszeit des FIFOS IDT7204L12TPG ist 12ns
Die Zugriffszeit des Textspeichers 7134LA20PDG ist 20ns.
Die Taktzeit der Statemachine ist 67,8ns, so das an dieser Stelle Zeitreserven bestehen.

Es soll nun die Laufzeit vom Übergang zu S1 bis zum Anstehen des Signals #char=normal ermittelt werden:

(max.-Werte über Temperatur)

U3201	74AS867	clk = > Q	16ns
U3202	74F138	A..C = > Y	9ns
U3214C	74F00	A..Y	6ns
U3101	7204L12	/R = > Q	12ns
U3203..5	74F521	A/B = > A=B	11ns

U3207B	74F20	A = > Y	6ns
Summe:			60ns

Es soll nun die Laufzeit vom Übergang auf S2 bis zum Signal #ram_we ermittelt werden:

(Der Ausgang von U3214B ist zum Ende von S1 schon stabil)

U3201	74AS867	clk = > Q	16ns
U3202	74F138	A..C = > Y	9ns
U3214A	74F00	A = > Y	6ns

Es soll nun die Laufzeit vom Übergang zu S1 bis zum Anstehen der Steuersignale (bis auf die Bedingung S4) ermittelt werden.

Der Pfad von #char=normal zu #incr_pointer_text_line_offset ist hierbei der längste Pfad.

(max.-Werte über Temperatur)

U3201	74AS867	clk = > Q	16ns
U3202	74F138	A..C = > Y	9ns
U3214C	74F00	A..Y	6ns
U3101	7204L12	/R = > Q	12ns
U3203..5	74F521	A/B = > A=B	11ns
U3207B	74F20	A = > Y	6ns
U3215A	74F27	A = > Y	7ns
U3217A	74F32	A = > Y	6,6ns
U3216B	74F27	A = > Y	7ns
U3208F	74F04	A = > Y	6ns
U3218A	74F08	A = > Y	6,6ns
U3219A	74F00	A = > Y	6ns
Summe:			99,2ns

Damit besteht eine Reserve von 3 x 67,8ns – 99,2ns = 203,4ns – 99,2ns = 104,2ns.
Würde man S1 auf zwei Taktzyklen verkürzen bestünde noch eine Reserve von 36,4ns.

Angesichts der in der Summe recht großen Kapazitäten der vielen hintereinandergeschalteten Verbindungsleitungen in dem recht langen Pfad wird jedoch die ursprünglich geplante Dauer von S1 von drei Taktperioden beibehalten, insbesondere auch, da die zur Verfügung stehende Zeit für den Transfer mehr als aureichend ist.

Das Businterface

Mit dem Businterface werden die folgenden Funktionen ausgeführt.

1. Schreiben von Zeichen in das FIFO
2. Auslesen des Full-Flags des FIFOs

3. Aktivieren / Deaktivieren der Cursordarstellung

4. Laden des Zeichensatzes
5. Auswahl des Zeichensatzes (für Laden und Darstellen, es sind zwei Zeichensätze möglich)

Das Businterface weist keine Besonderheiten auf, es entspricht weitgehend dem bereits realisierten Businterface der Karte SP2-001 MEM_IO.

Mit den Comparatoren U5201 und U5202 wird zunächst der Adressbereich FF FF 00 00 .. FF FF FF FF ausgewählt.

Mit dem Comparator U5203 wird weiterhin der Adressbereich FF FF FF E0. .FF FF FF FF ausgewählt, aus diesem nach weiterer Decodierung mit den Decodern U5205 und U5206 die Chipselectsignale für den I/O-Port und das FIFO abgeleitet werden:

FF FF FF EC	Parallelport (Lesen und Schreiben)		
	Lesen:	Bit 0	#FIFO_FF (L wenn FIFO voll, H wenn noch Platz im FIFO)
	Schreiben:	Bit 0	cursor_en (L = Cursor aus / H = Cursor sichtbar)

Mit dem Comparator U5207 wird der Adressbereich FF FF 8 xx .. FF FF B xx für das Laden des Zeichensatzes in den Zeichensatzspeicher decodiert.