

paperware 3

Universele terminal

Paperware 3 bevat aanvullende informatie over de Universele Terminal, onder gebruikmaking van de in *Elektuur*, september 1983 gepubliceerde VDU-kaart en de in *Elektuur*, november 1983 gepubliceerde CPU-kaart. Als CRT-controller (beeldsturing) wordt een IC van het type 6845 (Motorola) of 6545 (Rockwell/Synertek) gebruikt. Met dit IC kan op eenvoudige wijze een nabouwzekere video-interface worden gebouwd die tot véél in staat is.

De hoofdstukindeling is als volgt:

- * I Beschrijving van de 6845/6545.
- * II Programmeren van de 6845/6545.
- * III Beschrijving en programmering van de ACIA 6551.
- * IV Beschrijving van de diverse interfaces:
V24/RS 232, parallel, Centronics.
- * V Source-listing van de video-routines voor de Universele Terminal.

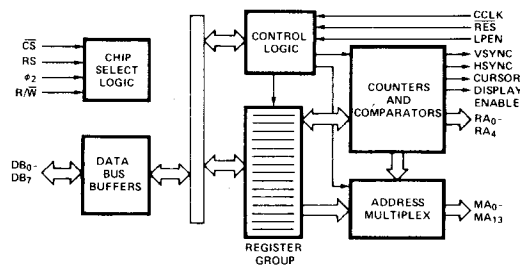
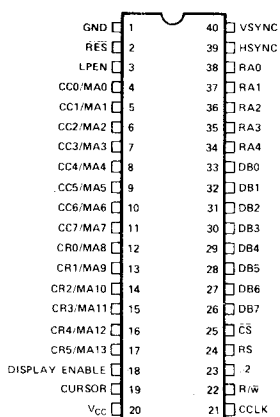


Uitgeversmij. Elektuur B.v					
Library	<table border="1"><tr><td></td><td></td><td></td><td></td></tr></table>				
Filenumber	<table border="1"><tr><td></td><td></td><td></td><td></td></tr></table>				

* I Beschrijving van de 6845/6545

Met het beeldstuurings-IC (CRT-controller) type 6845/6545 kun je alle microcomputers op simpele wijze aansluiten op een video-monitor. Het IC is programmeerbaar; via software kunnen diverse beeldformaten worden gerealiseerd zonder ook maar één wijziging in de hardware van de video-interface. Daar komt nog bij dat het IC goedkoop is. Aangevuld met een paar TTL-IC's levert het IC alle beeldschrijfmiddelen, als onderdeel van de totale video-interface. Figuur 1 toont de plattegrond en de interne opbouw (architectuur). Er zijn tellerschakelingen ten behoeve van de horizontale en de verticale opbouw van het beeld(scherm). Verder een lineaire adresteller en een registergroep (bestaande uit een aantal programmeerbare registers). De tellerschakelingen zorgen eveneens voor de stuursignalen voor de video-monitor: HSYNC, VSYNC, DISPLAY ENABLE en de raster-adreslijnen RA0... RA4. Met de 14 adreslijnen MA0... MA13 kan beeldgeheugen tot maximaal 16 K worden geadresseerd. De systeem-processor waarop de CRT-controller is aangesloten is in staat om de registergroep te lezen of er data in te schrijven. Dit levert de al genoemde mogelijkheid tot programmeren op. Men kan ook vastleggen hoeveel karakters een karakterrij omvat, uit hoeveel video-lijnen een karakterrij is opgebouwd, hoeveel karakterrijen, dus regels er op het scherm komen, of hoe breed de boven-, onder-, linker en rechter beeldrand moeten worden, of welke afmetingen de cursor (= "aanwijspunt") heeft. Of vastleggen dat de cursor knippert of juist niet. Zelfs de knipperfrequentie van de cursor kan worden geprogrammeerd. Omdat er verschillende video-frekquenties, horizontaal en vertikaal, bestaan, afhankelijk van het land waarin men verblijft, zijn ook de tijdsafstanden tussen opeenvolgende horizontale, respectievelijk verticale synchronisatie-pulsen programmeerbaar.

1



83903-1

De aansluitingen van de 6845/6545

Pen 1 (massa) en pen 20 (+5 V)

Aansluitingen voor de voedingsspanning. De tolerantie in de voedingsspanning bedraagt $\pm 10\%$, dus het voedingsspanningsbereik loopt van 4,5 V tot 5,5 V.

Pen 2 (RES)

Dit is de reset-ingang van de CRT-controller. Is actief in de toestand "0". Tijdens het resetten wordt de inhoud van de display-startregisters R12 en R13 op de adresbus MA0... MA13 gezet. Verder worden alle interne tellers gereset en alle overige uitgangen "0" gemaakt. De inhoud van de interne registers blijft ongewijzigd.

Pen 3 (LPEN)

De lichtpen-ingang. Indien de lichtpen op een letter op het beeldscherm wordt geplaatst, dan geeft deze lichtpen een impuls af op tijdstippen dat de elektronestraal van de monitor bezig is met het "schrijven" van die letter. De ingang LPEN reageert op een positieve lichtpen-impuls (dus "0"/"1"-overgang). Het bij de letter behorende adres wordt in het lichtpenregister (R16/R17) gekopieerd. De processor is vervolgens in staat om het adres waarop de lichtpen staat gericht, te lezen.

Pennen 4... 17 (MA0... MA13)

Dit zijn de adreslijnen voor de adressering van video-RAM. Via deze adresbus worden opeenvolgende karakters in video-RAM aangewezen om op het scherm te worden weergegeven. Het momentane adres op deze adresbus bepaalt welk karakter (opgeslagen in de geadresseerde RAM-geheugenplaats) op dat moment op het scherm wordt weergegeven.

Pen 18 (DISPLAY ENABLE)

Het via de CRT-controller gerealaiseerde beeld omvat een actief gedeelte

en een niet-actief gedeelte. Indien de elektronestraal bezig is met het schrijven van een beeldrand, dan spreken we van het niet-actieve gedeelte. Dan is DISPLAY ENABLE "0". Is er daarentegen sprake van het schrijven van een karakterrij, dan betreft dit de actieve fase en is DISPLAY ENABLE "1". Dit uitgangssignaal wordt gebruikt voor de "donkersturing" van het beeld en verder voor de sturing van de perifere logica van de CRT-controller.

Pen 19 (CURSOR)

Een uitgang, die "1" wordt zodra de CRT-controller die plaats op het beeldscherm adresseert, waar de cursor zich bevindt. De positie van de cursor op het scherm is vrij programmeerbaar. De inhoud van het cursor-register (R14/R15) is bepalend voor die positie. Het CURSOR-signaal wordt als ingang gebruikt voor de perifere logica ten behoeve van de cursor-sturing.

Pen 21 (CCLK)

Ingang voor het karakter-kloksignaal. De frekwentie van deze klok bepaalt de karakterfrekwentie op de video-monitor. Alle interne tellers gebruiken CCLK als tijdbasis. De puntfrekwentie (dot-frekwentie) is 8 x zo hoog als de karakterfrekwentie. Op de Elektuur-VDU-kaart is de karakterfrekwentie via deling herleid uit de puntfrekwentie. Hierdoor ontstaat een scherp en stabiel video-beeld.

Pen 22 (R/W)

Deze ingang wordt gestuurd uit de R/W-leiding van het processorsysteem. Indien data uit één van de interne registers moet worden gelezen, dan moet R/W "1" zijn. Een "0" op deze ingang maakt het voor de processor mogelijk om data in één van de interne registers te schrijven.

Pen 23 ($\Phi 2$ of E)

Dit is een ingang die wordt gestuurd uit de $\Phi 2$ -systeemklok. Aangezien de Z80 een asynchrone busstructuur bezit wordt deze pen op de VDU-kaart via een op de print aanwezige interface gestuurd. Alle data van of naar de CRT-controller wordt aan de hand van dit signaal gestuurd.

Pen 24 (RS)

Deze register-selektie-ingang is verbonden met adreslijn A0 van het processorsysteem. Via RS zijn alle interne registers toegankelijk. Indien RS "0" is wordt het adresregister van de CRT-controller geadresseerd. Het adresregister wijst het door de CPU te adresseren register aan. Als de processor dus een bepaald register binnen de 6845/6545 wil adresseren, dan moet eerst passende data in het adresregister worden geschreven en wel zodanig dat het adresregister op het gewenste register wijst. Zoals vermeld

moet RS "0" zijn tijdens deze schrijfoperatie.

Nu is het gewenste register van de registergroep vastgelegd; in dit vastgelegde register kan de processor nu lezen of schrijven. Dan moet RS "1" zijn.

Pen 25 (\overline{CS})

De Chip Select-ingang \overline{CS} is via een adresdekoder met de adresbus van het processorsysteem verbonden. Is dit signaal "0", dan wordt de CRT-controller door de CPU geselecteerd.

Pennen 26 ... 33 (DB7 ... DB0)

Deze acht leidingen zijn met de databus van het processorsysteem doorverbonden. Zolang de CRT-controller (via \overline{CS}) niet is geadresseerd zijn deze aansluitingen "tri state", dus feitelijk ontkoppeld.

Pennen 34 ... 37 (RA4 ... RA0)

Vier raster-adreslijnen, uitgangen waar op de adressen voor de "raster scan" staan. De elektronenstraal op de videomonitor moet meerdere video-lijnen schrijven om één karakterrij op het scherm weer te geven. Het adres op deze vier lijnen geeft uitsluitend over die video-lijnen die in het kader van de weergave van een karakterrij op een bepaald tijdstip wordt geschreven. Het aantal video-lijnen per karakterrij is vrij te kiezen, c.q. te programmeren. Dat geldt ook voor de afstand (in een aantal video-lijnen uitgedrukt) tussen twee opeenvolgende karakterrijen.

Pen 39 (HSYNC)

Het gaat hierbij om de horizontale synchronisatie-uitgang van de CRT-controller. De breedte (= tijdsduur) van de horizontale synchronisatiepuls en het tijdsverschil tussen opeenvolgende synchronisatiepulsen zijn vrij te programmeren.

Pen 40 (VSYNC)

Dat kan niet missen: de verticale synchronisatie-uitgang van de CRT-controller. De breedte (= tijdsduur) van de verticale synchronisatiepuls en het tijdsverschil tussen opeenvolgende verticale synchronisatiepulsen zijn vrij te programmeren.

Figuur 1 toont u eveneens de interne opbouw van de CRT-controller. De chip-select-logica maakt het lezen en schrijven rond de registergroep mogelijk. De databusbuffers (ont)koppelen de interne 6845/6545-databus van/aan de externe databus. De tellers en de komparatoren realiseren, in samenwerking met de data in de diverse registers van de registergroep, de diverse video-uitgangssignalen die nodig zijn voor de opbouw van het uiteindelijke video-signaal. De adres-multiplexer zorgt voor de adressen van de op de videomonitor weergegeven karakters.

* II De programmering van de 6845/6545

De registergroep is van bijzonder belang voor de programmeur. De van de groep deel uitmakende registers bevatten alle informatie, die voor de opbouw van het video-beeld van belang zijn. Bijvoorbeeld: het aantal karakters per regel, het aantal karakterrijen (= regels) per beeld, het aantal door de elektronenstraal te schrijven video-lijnen voor de weergave van één regel. Indien er een lichtpen is aangesloten, dan kan de processor door het lezen van het lichtpenregister (dat deel uitmaakt van de registergroep) nagaan op welk beeldadres het karakter staat, waarmee de lichtpen "kontakt maakt". Dit soort informatie en nog veel meer andere informatie vindt u in de registers van de registergroep.

Figuur 2 laat zien hoe een (video-) beeld wordt opgebouwd. Horizontaal wordt het tijdsverloop in karakters geteld. Vertikaal telt "men" karakterrijen, dus regels. In het voorbeeld is sprake van een beeldformaat van 80 x 24 karakters. De karakters 0 ... 79 op de eerste rij van figuur 2 worden op het scherm weergegeven. De "karakters" 80 ... 89 vertegenwoordigen de linker en rechter (donkere) beeldrand. De tijd, gedurende welke de CRT-controller van 0 tot 89 telt is vastgelegd op 64 μ s. De puntfrequentie is 8 x zo hoog als de karakterfrequentie. Beide frequenties zijn op de VDU-kaart gekoppeld via een gemeenschappelijke klokgenerator.

Een regel omvat meerdere, door de elektronenstraal te schrijven video-lijnen. Op de VDU-kaart is uitgegaan van 9 video-lijnen per regel, de donkere lijnen tussen twee opeenvolgende regels meegerekend. Een regel neemt derhalve $9 \times 64 \mu s = 576 \mu s$ in beslag. Vertikaal in figuur 2 worden regels, dus karakterrijen geteld. De rijen 0 ... 23 betreffen het actieve gedeelte van het beeld, de rijen 24 ... 31 vormen de donkere boven- en onderrand

van het beeld. De verticale synchronisatiepuls VSYNC valt samen met het midden van het inactieve regelgedeelte, de horizontale synchronisatiepuls HSYNC valt samen met het midden van het inactieve karaktergedeelte.

Het beeldformaat en de positie van horizontale en verticale synchronisatiepulsen kunnen vrij eenvoudig worden vastgelegd via het schrijven van data in registers van de registergroep. Daarom moet direct na een systeemreset ($RES = "0"$) door de processor data uit een opzoektabel (look up table) worden gelezen en in de registers van de registergroep worden gekopieerd. Over het programmeren van die registers gaan we het nu hebben.

De registergroep van de 6845/6545 omvat 21 registers, te weten R0 ... R19 en R31. Alle registers zijn hooguit acht bits breed. Alle registers worden gekozen via één enkele adreslijn.

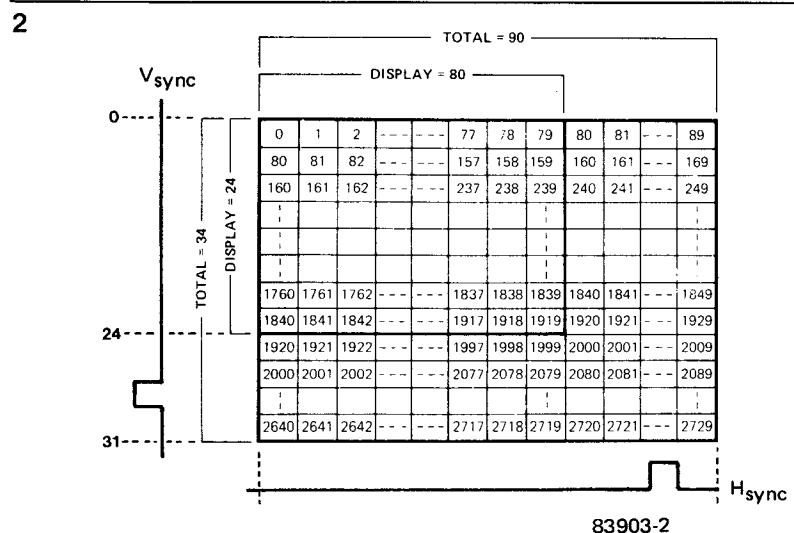
Hoe gaat de registerkeuze in zijn werk? Eerst moet, via schrijven, het adresregister worden geladen met data die samenhangt met het gewenste register. In aansluiting daarop wordt de adreslijn van de CRT-controller (RS is aangesloten op A0) "1" gemaakt en nu kan de processor via de databus data schrijven in het gewenste register. Voor de absolute adressen van het adresregister en van de registergroep wordt u verwezen naar de source-listing van de UT-software.

Register R0 (totaal horizontaal)

Dit register bevat in binaire vorm de optelsom van actieve en niet-actieve karakters binnen een regel, min één. Actieve karakters zijn zichtbaar, niet-actieve karakters stellen de linker en rechter beeldrand voor. De inhoud van R0 bepaalt de frequentie van de horizontale synchronisatiepulsen.

Register R1 (horizontaal weergegeven)

Dit register bevat het aantal actieve,



dus weergegeven karakters per regel. Een inhoud \$50 duidt op 80 (aktieve) karakters per regel. De inhoud is nooit groter dan de inhoud van R0.

Register R2 (positie horizontale syncpuls HSYNC)

De inhoud van dit register is bepalend voor de ligging van de horizontale synchronisatiepuls in een periode met niet-aktieve karakters, dus bepalend voor de breedte van de linker, respectievelijk rechter beeldrand. De inhoud van R2 ligt tussen die van R1 en die van R0.

Register R3 (breedte van HSYNC & VSYNC)

De bits b3 . . . b0 van R3 bepalen hoeveel (maximaal 16) karakters de horizontale synchronisatiepuls breed is. De verticale synchronisatiepuls neemt altijd 16 video-lijntijden in beslag. Met de SY 6545E van de firma Synertek is het mogelijk om via de bits b7 . . . b4 van R3 de breedte van de verticale synchronisatiepuls VSYNC (maximaal 16 video-lijntijden) te programmeren.

Register R4 (totaal vertikaal)

Dit register bevat het aantal actieve plus niet-actieve karakterrijen (regels), min één. Een inhoud \$1F van R4 duidt op een totaal-aantal van 32 rijen. De tijd die dan verstrijkt voordat alle 32 rijen, actief of niet-actief, van een beeld zijn geschreven, bedraagt $32 \times 9 \times 64 \mu s = 18,432$ millisekonden. Aangezien de gebruikelijke tijd tussen twee opeenvolgende verticale synchronisatiepulsen 20 millisekonden is, is een kleine correctie van 20 minus 18,432 millisekonden noodzakelijk. Er ontbreekt $1,568 \text{ ms} = 1568 \mu s$. N.B. Met R5 vindt de aanvulling plaats in stappen van $64 \mu s$.

Register R5 (aanpassing totaal vertikaal) (fijnafstemming)

Dit 5-bits register, dat niet kan worden gelezen, bepaalt het aantal lijntijden dat, opgeteld bij de inhoud van R4, zorg draagt voor een beeld-herhalingsfrequentie van 50 Hz, dus een beeld-tijd van 20 millisekonden.

Register R6 (vertikaal weergegeven)

Een 7-bits register dat het aantal actieve karakterregels bevat. De inhoud van R5 is kleiner dan die van R4.

Register R7 (positie verticale syncpuls VSYNC)

De inhoud van dit register bepaalt de positie van de verticale synchronisatiepuls VSYNC binnen een aantal niet-actieve karakterrijen (zie figuur 2). De positie wordt uitgedrukt in karakter-tijden. Een en ander is van invloed op de hoogte van de bovenrand en van de onderrand van het beeld.

Register R8 (werkwijze van de CRT-controller)

Dit register omvat slechts twee bits. In figuur 3a is aangegeven hoe de bits de compositie van het video-signaal beïnvloeden. Figuur 3c laat zien wat dat in de beeldpraktijk inhoudt. De R8 van de SY 6545E van Synertek biedt méér mogelijkheden. Hier is R8 acht bits breed. Zie hiervoor figuur 3b. De belangrijkste bits van dit uitgebreide R8-register zijn de "skew"-bits voor de cursor en voor het signaal DISPLAY ENABLE. Met deze twee bits kunnen de signalen CURSOR en DISPLAY ENABLE gedurende een karaktertijd worden vertraagd. Op de VDU-kaart is deze vertraging met flipflops, extern gerealiseerd. Voordeel: alle XX45 CRT-controllers kunnen op deze Elektuur-print worden ingezet!

Register R9 (aantal video-lijnen per regel)

De inhoud van dit register bepaalt met hoeveel video-lijnen (minus 1) één karakterregel wordt beschreven door de elektronenstraal. Op de VDU-kaart is uitgegaan van 9 video-lijnen per regel, te weten acht actieve lijnen en één "lege" video-lijn.

Let op! Indien grafische symbolen moeten worden weergegeven, moet worden uitgegaan van acht video-lijnen per grafisch teken. Dit omdat de grafische symbolen zijn opgebouwd volgens een 8×8 -matrix. De "lege" video-lijn komt nu te vervallen, er ontstaat dan immers een aaneengesloten geheel van grafische symbolen.

Registers R10 en R11 (cursor-informatie)

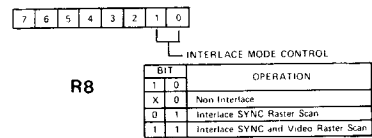
De rechter vijf bits van R10 bepalen, in welke video-lijn de cursor begint en de rechter vijf bits van R11 stellen vast in welke video-lijn de cursor eindigt (zie figuur 4b). Op deze wijze zijn dus de verticale afmetingen van de cursor vastgelegd. De bits b6 en b5 van R10 geven antwoord op de vraag of de cursor knippert (met een bepaalde frequentie) of niet knippert, en of de weergave van de cursor al dan niet moet plaatsvinden. Zie verder de waarheidstabel van figuur 4a.

Registers R12 (H) en R13 (L) (Display-startadres)

Deze twee registers vormen tezamen een 14-bits adres, dat hoort bij de beginpositie van het eerste karakter ("home position", linksboven op het scherm). Op de VDU-kaart worden uitsluitend de bits b0 . . . b10 gebruikt omdat de video-RAM 2 K geheugenruimte omvat (000 . . . 7FF).

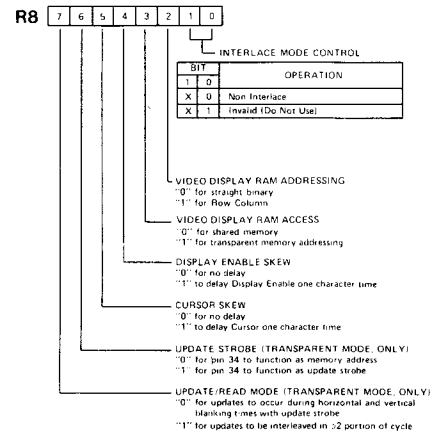
Via manipulatie van de inhoud van R12 en R13 is het mogelijk om het beeld één regel omhoog of omlaag te laten verspringen, of om het beeld een met het aantal regels per beeld overeenkomend aantal regels omhoog of omlaag te laten verspringen. Men

3a



83903-3a

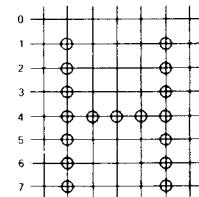
b



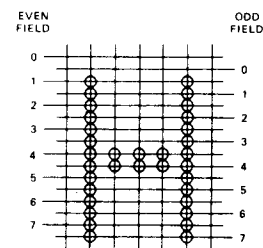
83903-3b

c

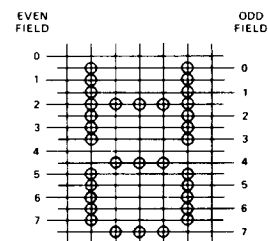
Display Modes



NON INTERLACED



INTERLACED SYNC



INTERLACED SYNC AND VIDEO

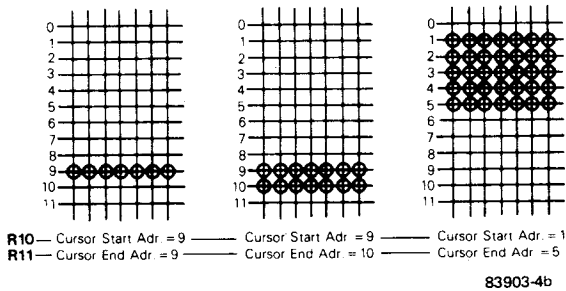
83903-3c

4a

R10	BIT		CURSOR MODE
	6	5	
0	0	0	No Blinking
0	0	1	No Cursor
1	0	0	Blink at 16x field rate (fast)
1	1	1	Blink at 32x field rate (slow)

83903-4a

b



spreekt van "scroll up" en "scroll down". In figuur 5 is de opbouw van de video-RAM geschetst. Dit geheugen kent een ringvormige opzet ("wrap around memory"). De sturing van alle karakters op het scherm geschiedt onder invloed van een hardware-adreswijzer (pointer), te weten de display-startadreswijzer, en drie software-adreswijzers: FLN, CLN en LLN. Alle vier genoemde adreswijzers zijn virtuele adreswijzers, m.a.w. hun inhoud komt niet overeen met het effectieve karakteradres in video-RAM. Het effectieve karakteradres bereken je als volgt:

effectieve karakteradres =
 inhoud van CLN
 (momentane cursor-regel)
 plus
 positienummer van de cursor
 plus
 absoluut beginadres van de video-RAM
 (\$D000)

Definities:
 FLN = "First LiNe pointer". Staat altijd gericht op het begin van de eerste regel. Inhoud varieert van 000...7FF.
 CLN = "Current LiNe pointer". Staat gericht op het begin van de regel waarin zich de cursor bevindt. De cursorpositie volgt uit de optelsom van inhoud CLN en de inhoud van COL, welke laatste het positienummer (kolomnummer) van de cursor bevat. De inhoud van CLN varieert van 000...7FF.

LLN = "Last LiNe pointer". Staat altijd gericht op het begin van de onderste regel. Inhoud varieert van 000...7FF.
 De display-startadreswijzer (in de CRT-controller) staat net als FLN gericht op het eerste karakter van de bovenste regel. Ook hier varieert de inhoud van 000...7FF.
 De inhoud van de vier adreswijzers wijzigt zich als volgt:

1. Verhoog telkens met 1 tot maximaal \$7FF. Bij overschrijding van de maximale inhoud begint het opnieuw bij nul.
2. Verlaag telkens met 1 tot minimaal

nul. Begin vervolgens opnieuw bij \$7FF.

Scroll:

Het gebruikelijke beeldformaat rond de Elektuur-VDU-kaart bedraagt 80 x 24. In verticale richting kan de cursor bewegen tussen regel 1 en regel 24. Beweegt de cursor omlaag en voorbij de onderste regel, dan vindt een "scroll up" plaats. De video-software doet in dat geval de inhoud van het beeld een regel omhoog schuiven; de tot voor kort bovenste regel verdwijnt. Beweegt de cursor omhoog en voorbij de bovenste regel, dan is er sprake van een "scroll down". In dat geval wordt de beeldinhoud een regel omlaag verschoven; de tot voor kort onderste regel verdwijnt.

Er zijn twee soorten scroll:
 1. Geheugen-scroll. De totale inhoud van de video-RAM wordt in het geheugen verplaatst; de verplaatsing komt overeen met het aantal karakters op een regel. Het display-startadres blijft ongewijzigd. In het geval van een 80 x 24-beeldformaat moet 80 x 24 = 1920 x data worden gekopieerd naar de nieuwe positie in video-RAM. Dat

kost tijd en het houdt de processor teveel bezig. Deze methode wordt dan ook op de VDU-kaart niet gebruikt.
 2. De adreswijzer-scroll. Bij deze gang van zaken wordt alleen de inhoud van het display-startadres (in de CRT-controller) gewijzigd. In het geval van een scroll up wordt de inhoud van het display-startadres met het aantal karakters per regel verhoogd, bij een scroll down met eenzelfde aantal verlaagd. Dit vergt slechts één schrijfoperatie voor de processor. Vandaar dat deze methode veel sneller verloopt.

Registers R14 (H) en R15 (L) (cursorspositie)

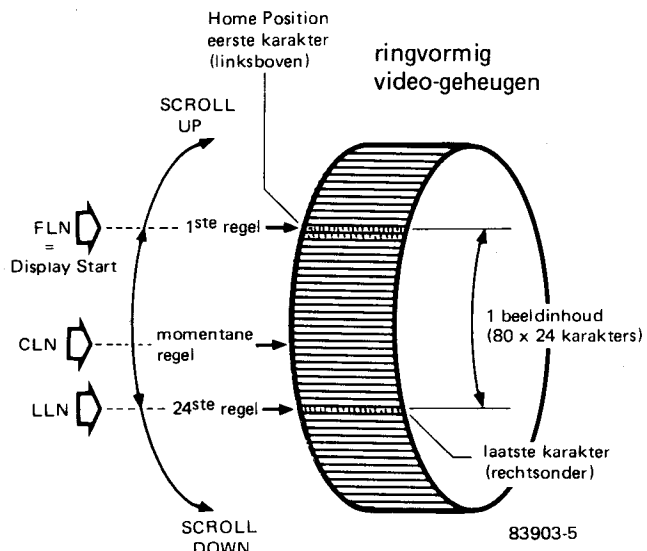
De inhoud is bepalend voor de positie van de cursor op het scherm. Zodra de data in R14/R15 gelijk is aan de door de adresbus MA0...MA10 voorgestelde data wordt de uitgang CURSOR actief. Verder worden automatisch de nivo's op de leidingen RA0...RA4 vergeleken met de inhoud van de cursorregisters R10/R11. Deze vergelijking levert de hoogte op van de cursor (uitgedrukt in een aantal video-lijnen).

Registers R16 (H) en R17 (L) (lichtpenregisters)

Deze registers bevatten het adres van het karakter waarop de lichtpen staat gericht. De opname van dit adres gebeurt tijdens een nivo-overgang van "0" naar "1" op de ingang LPEN van de CRT-controller, gevolgd door een nivo-overgang van "1" naar "0" van het karakterkloksignaal.

De 6845/6545 bezit nog een paar andere registers, te weten R18, R19 en R31. Deze registers zullen hier niet worden besproken omdat ze niet dezelfde betekenis hebben voor al de diverse fabrikaten 6845/6545. Er wordt dan ook geen gebruik gemaakt van deze registers in de video-software.

5



CS	RS	Address Reg.					Reg. No.	Register Name	Stored Info.	RD	WR	Register Bit													
		4	3	2	1	0						7	6	5	4	3	2	1	0						
1	-	-	-	-	-	-	-					/	/	/	/	/	/	/	/	/	/	/	/		
0	0	-	-	-	-	-	-	Address Reg	Reg. No.		\	/	/	/	/	/	/	/	/	/	A ₄	A ₃	A ₂	A ₁	A ₀
0	0	-	-	-	-	-	-	Status Req.		\		U	L	V	/	/	/	/	/	/	/	/	/	/	
0	1	0	0	0	0	0	R0	Horiz. Total - 1	# Charac.		\	●	●	●	●	●	●	●	●	●	●	●	●	●	
0	1	0	0	0	0	1	R1	Horiz. Displayed	# Charac.		\	●	●	●	●	●	●	●	●	●	●	●	●	●	
0	1	0	0	0	1	0	R2	Horiz. Sync Position	# Charac.		\	●	●	●	●	●	●	●	●	●	●	●	●	●	
0	1	0	0	0	1	1	R3	VSYNC, HSYNC Widths	# Scan Lines and # Char. Times		\	V ₃	V ₂	V ₁	V ₀	H ₃	H ₂	H ₁	H ₀	/	/	/	/	/	
0	1	0	0	1	0	0	R4	Vert. Total - 1	# Charac. Row		\	/	●	●	●	●	●	●	●	●	●	●	●	●	
0	1	0	0	1	0	1	R5	Vert. Total Adjust	# Scan Lines		\	/	/	/	/	/	/	/	/	/	/	/	/	/	
0	1	0	0	1	1	0	R6	Vert. Displayed	# Charac. Rows		\	●	●	●	●	●	●	●	●	●	●	●	●	●	
0	1	0	0	1	1	1	R7	Vert. Sync Position	# Charac. Rows		\	/	●	●	●	●	●	●	●	●	●	●	●	●	
0	1	0	1	0	0	0	R8	Mode Control			\	●	●	●	●	●	●	●	●	●	●	●	●	●	
0	1	0	1	0	0	1	R9	Scan Lines - 1	# Scan Lines		\	/	/	/	/	/	/	/	/	/	/	/	/	/	
0	1	0	1	0	1	0	R10	Cursor Start	Scan Line No.		\	B ₁	B ₀	●	●	●	●	●	●	●	●	●	●	●	
0	1	0	1	0	1	1	R11	Cursor End	Scan Line No.		\	/	/	/	/	/	/	/	/	/	/	/	/	/	
0	1	0	1	1	0	0	R12	Display Start Addr (H)			\	/	●	●	●	●	●	●	●	●	●	●	●	●	
0	1	0	1	1	0	1	R13	Display Start Addr (L)			\	●	●	●	●	●	●	●	●	●	●	●	●	●	
0	1	0	1	1	1	0	R14	Cursor Position (H)		\	\	/	/	/	/	/	/	/	/	/	/	/	/	/	
0	1	0	1	1	1	1	R15	Cursor Position (L)		\	\	●	●	●	●	●	●	●	●	●	●	●	●	●	
0	1	1	0	0	0	0	R16	Light Pen Req (H)		\	/	●	●	●	●	●	●	●	●	●	●	●	●	●	
0	1	1	0	0	0	1	R17	Light Pen Req (L)		\	/	●	●	●	●	●	●	●	●	●	●	●	●	●	
0	1	1	0	0	1	0	R18	Update Address Reg (H)		\	/	/	/	/	/	/	/	/	/	/	/	/	/	/	
0	1	1	0	0	1	1	R19	Update Address Reg (L)		\	/	●	●	●	●	●	●	●	●	●	●	●	●	●	
0	1	1	1	1	1	1	R31	Dummy Location			/	/	/	/	/	/	/	/	/	/	/	/	/	/	

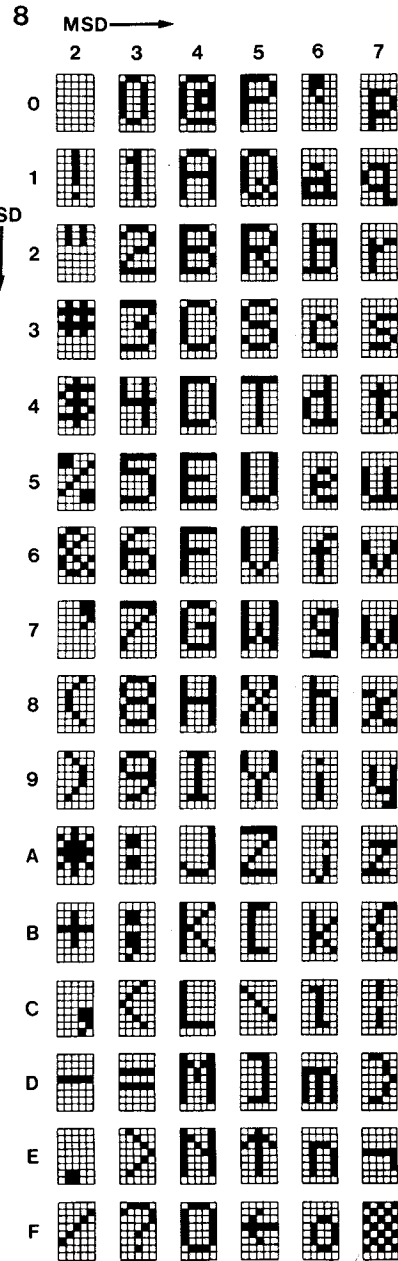
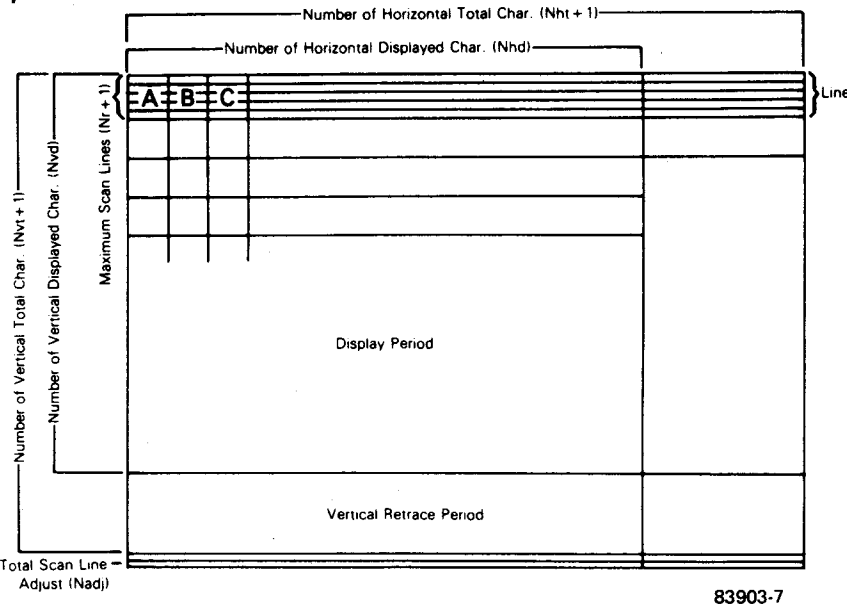
Notes: ● Designates binary bit
 / Designates unused bit. Reading this bit is always "0", except for R31, which does not drive the data bus at all, and for CS = "1" which operates likewise.

83903-6

Een samenvattend overzicht van de registergroepen treft u aan in figuur 6. Figuur 7 verschaft een samenvattend overzicht van de beeldopbouw, met inbegrip van de diverse gebruikte begrippen.

Voor de omzetting van een karakter in het overeenkomende puntenpatroon zorgt de karaktergenerator. In een EPROM (2732) is het puntenpatroon vastgelegd van alle karakters en grafische symbolen. Aangezien Ohio Scientific het een en ander aan spelletjes-software te bieden heeft, zijn de grafische tekens compatibel met Ohio.

De karaktergenerator bestaat uit twee helften, elk 2 K groot. De karakters beslaan het adresbereik 00X...7FX, de grafische tekens het adresbereik 80X...FFX (X = 0...F). Figuur 8 toont de inhoud van de karaktergenerator voor de Amerikaanse versie en figuur 9 toont voor de volledigheid de wijzigingen ten behoeve van de Duitse versie (Umlauten en sz = ß).





128 \$B0



129 \$B1



130 \$B2



131 \$B3



132 \$B4



133 \$B5



134 \$B6



135 \$B7



136 \$B8



137 \$B9



138 \$B8A



139 \$B8B



140 \$B8C



141 \$B8D



142 \$B8E



143 \$B8F



144 \$B90



145 \$B91



146 \$B92



147 \$B93



148 \$B94



149 \$B95



150 \$B96



151 \$B97



152 \$B98



153 \$B99



154 \$B9A



155 \$B9B



156 \$B9C



157 \$B9D



158 \$B9E



159 \$B9F



160 \$BA0



161 \$BA1



162 \$BA2



163 \$BA3



164 \$BA4



165 \$BA5



166 \$BA6



167 \$BA7



168 \$BA8



169 \$BA9



170 \$BAA



171 \$BAB



172 \$BAC



173 \$BAD



174 \$BAE



175 \$BAF



176 \$BB0



177 \$BB1



178 \$BB2



179 \$BB3



180 \$BB4



181 \$BB5



182 \$BB6



183 \$BB7



184 \$BB8



185 \$BB9



186 \$BBA



187 \$BBB



188 \$BBC



189 \$BBD



190 \$BBE



191 \$BBF



192 \$BB0



193 \$BC1



194 \$BC2



195 \$BC3



196 \$BC4



197 \$BC5



198 \$BC6



199 \$BC7



200 \$BC8



201 \$BC9



202 \$BCA



203 SCB



204 SCC



205 SCD



206 SCE



207 SCF



208 SD0



209 SD1



210 SD2



211 SD3



212 SD4



213 SD5



214 SD6



215 SD7



216 SD8



217 SD9



218 SDA



219 SDB



220 SDC



221 SDD



222 SDE



223 SDF



224 SE0



225 SE1



226 SE2



227 SE3



228 SE4



229 SE5



230 SE6



231 SE7



232 SE8



233 SE9



234 SEA



235 SEB



236 SEC



237 SED



238 SEE



239 SEF



240 SF0



241 SF1



242 SF2



243 SF3



244 SF4



245 SF5



246 SF6



247 SF7



248 SF8



249 SF9



250 SFA



251 SFB



252 SFC



253 SFD



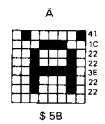
254 SFE



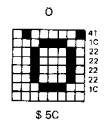
255 SFF

83903-8

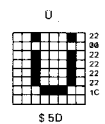
9



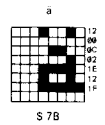
\$5B



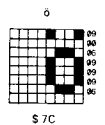
\$5C



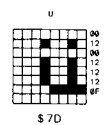
\$5D



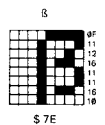
\$7B



\$7C



\$7D



\$7E

83903-9

* III Beschrijving en programmering van de ACIA 6551

Het seriële dataverkeer tussen de computer en de Universele Terminal wordt geregeld door een ACIA, een asynchrone interface-adaptor, van het type 6551. Dit IC herbergt een interne baudrate-generator, twee besturingsregisters, een statusregister, een data-verzendregister en een data-ontvangstregister. Extern zijn slechts een kristal en een nivo-aanpassing nodig. Figuur 10 toont het blokschema van de 6551. Alle handshake-leidingen, nodig voor de direkte aansluiting op een modem, zijn aanwezig.

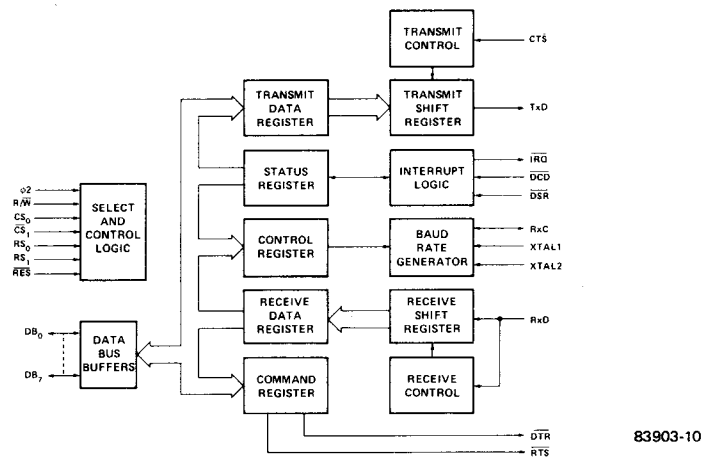
Vijf interne registers van de 6551 worden geselecteerd door de signalen RS0 en RS1 (registerselectie), in sommige gevallen aangevuld met het signaal R/W; figuur 11 geeft de details. De processor schrijft de te verzenden data in het data-verzendregister. De snelheid waarmee seriële data bit voor bit de ACIA in of uit gaat wordt bepaald door een programmeerbare baudrate-generator. Data die op het beeldscherm zichtbaar moet worden gemaakt ontvangt de Universele Terminal van de computer door het lezen van het data-ontvangstregister. De baudrate-generator stuurt de zender en de ontvanger binnen de ACIA. Het is echter ook mogelijk om de ontvanger van de baudrate-generator te ontkoppelen en een externe baudrate-generator aan te sluiten. De zender en de ontvanger kunnen in dat geval met onderling verschillende transmissiesnelheden werken. Voordat de tot de Universele Terminal behorende processor data in het data-verzendregister kan schrijven, of voordat-ie het data-ontvangstregister kan lezen, moet het statusregister worden gelezen. De bits van het statusregister geven uitsluitel over de vraag of het data-verzendregister al leeg is, en of het data-ontvangstregister vol is. Andere bits van het statusregister stellen vast of er in de transmissie fouten zijn opgetreden of niet. Het kommandoregister en het stuurregister zijn lees/schrijfregisters. Ze bepalen de werkwijze van de ACIA. Na een automatische of een manuele systeem-reset (RES = 0) krijgen de beide registers via software een bepaalde inhoud toegekend; de werkwijze ligt daarmee vast.

De bits van het kommandoregister:

b0:
 0 = blokkeer een ontvanger-interrupt en alle andere interrupts.
 1 = sta een ontvanger-interrupt en alle andere interrupts toe.

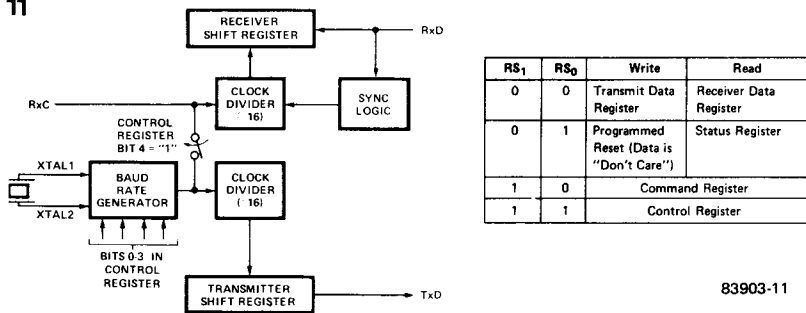
b1:
 0 = sta een interrupt ten gevolge van een vol data-ontvangstregister toe.
 1 = blokkeer een interrupt ten gevolge van een vol data-ontvangstregister.

10



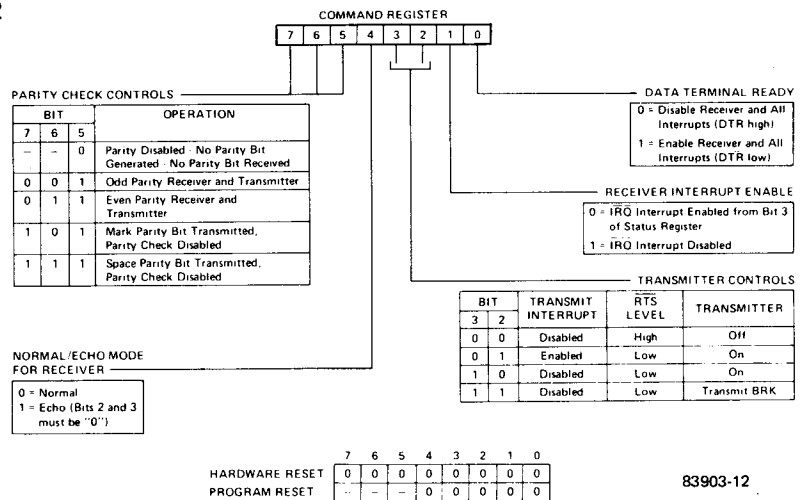
83903-10

11



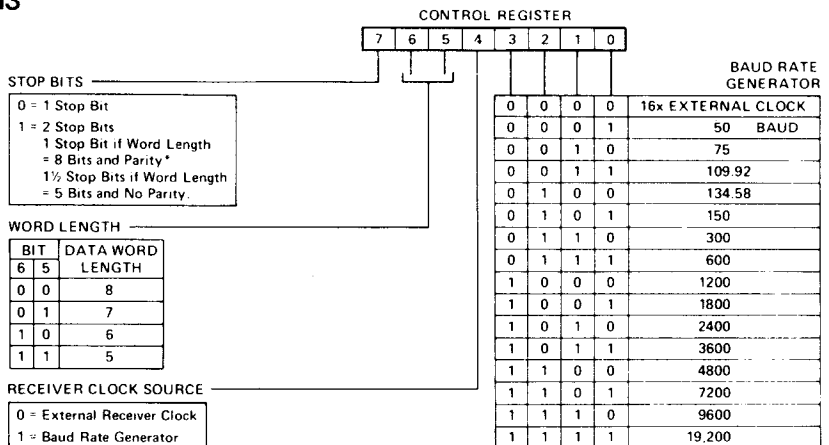
83903-11

12



83903-12

13



*This allows for 9-bit transmission (8 data bits plus parity).

83903-13

- b3 b2:
- 0 0 = blokkeer een zender-interrupt. De RTS-leiding is "1".
 - 0 1 = sta een zender-interrupt toe. De RTS-leiding is "0".
 - 1 0 = blokkeer een zender-interrupt, met RTS "0".
 - 1 1 = blokkeer een zender-interrupt, met RTS "0". Verzend een BRK.

- b4:
- 0 = normale werkwijze
 - 1 = echo-karakters. Omschakeling op half en full duplex. De bits b3 en b2 moeten "0" zijn.

- b7 b6 b5:
- X X 0 geen pariteitsbit verzonden of ontvangen.
 - 0 0 1 ontvang en verzend met oneven pariteit.
 - 0 1 1 ontvang en verzend met even pariteit
 - 1 0 1 verzend een mark-pariteitsbit; geen pariteitskontrolle.
 - 1 1 1 verzend een space-pariteitsbit; geen pariteitskontrolle.

De diverse bits van het stuurregister (figuur 13):
 b3 b2 b1 b0:
 Via deze vier bits wordt de baudrate van de ACIA geprogrammeerd. Indien alle vier bits nul zijn wordt er uitgegaan van een externe baudrate-generator, aangesloten op pen 5 (R x C). Er zijn 15 verschillende baudrates mogelijk van 50 tot en met 19200 baud.

- b4:
- 0 = ontvanger wordt met externe baudrate-generator geklokt, met pen 5 van de 6551 als klokkingang.
 - 1 = ontvanger en zender worden gestuurd vanuit de interne baudrate-generator, waarbij pen 5 van de 6551 de klokuitgang is. Omdat de interne baudrate-generator andere ACIA's kan sturen is het mogelijk om voor die andere ACIA's geen dure kristallen in te zetten.

- b6 b5:
- 0 0 = serieel signaal met 8 databits.
 - 0 1 = serieel signaal met 7 databits.
 - 1 0 = serieel signaal met 6 databits.
 - 1 1 = serieel signaal met 5 databits (Baudot-transmissie).
- Bit b7 bepaalt het aantal stopbits.

Figuur 14 toont de bits van het statusregister van de 6551. De belangrijkste bits zijn de ontvangstvlag b3 en de verzendvlag b4. Deze vlaggen vertellen of er een nieuw karakter mag worden verzonden of ontvangen. In de sourcelisting die van deze paperware deel uitmaakt vindt u typische verzenden- en ontvangstroutines. De ACIA-verzendroutine heet ACIOUT, de ACIA-ontvangstroutine ACIIN. In figuur 15 is de aansluitplaattegrond van de 6551 te zien, alsmede een ASCII-tabel. Programmering van de ACIA:

De Universele Terminal van Elektuur bestaat uit de 6502-CPU-kaart, beschreven in Elektuur, november 1983, en de VDU-kaart, beschreven in Elektuur, september 1983. Beide kaarten zijn uitgevoerd in euroformaat en kunnen direct op de Elektuurbus worden geplaatst. Opdat de Universele Terminal op alle computersystemen kan worden aangesloten, is op de CPU-kaart voorzien in aanvullende hardware ten behoeve van de eenvoudige programmering van het ACIA-stuurregister en van het ACIA-kommandoregister. In figuur 16 is te zien dat PL3 het ACIA-kommandoregister programmeert en dat PL4 het ACIA-stuurregister programmeert. De programmering bestaat uit het aanbrengen respectievelijk achterwege laten van draadbruggen.

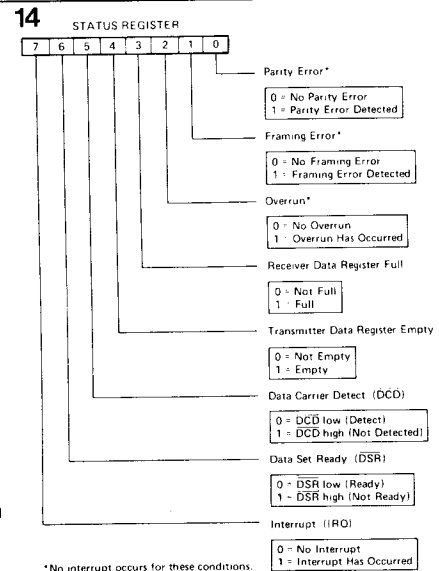
Let op!
 1 = draadbrug aangebracht;
 0 = kontakten open laten.
 De al dan niet aangebrachte draadbruggen in PL3 en PL4 corresponderen met de afzonderlijke bits van het kommandoregister en van het stuurregister. Na een RESET leest de terminal het gekozen bitpatroon op PL3 en op PL4 en programmeert daarmee de werkwijze van de ACIA. Hoe programmeer je het video-formaat?
 Door in figuur 16 rond PL6 draadbruggen aan te brengen, respectievelijk niet aan te brengen is het mogelijk om het video-formaat vast te leggen, dus het aantal karakters per regel en het aantal regels per scherm. Er zijn acht video-formaten in EPROM vastgelegd; één daarvan is naar keuze van de gebruiker. De volgende formaten staan ter beschikking:

- 0 0 0 80 x 24
- 0 0 1 80 x 25
- 0 1 0 64 x 16
- 0 1 1 64 x 24
- 1 0 0 90 x 22
- 1 0 1 48 x 12
- 1 1 0 24 x 24
- 1 1 1 vrij te programmeren

* IV Beschrijving van de diverse interfaces

a. RS 232/V24-interface

De seriële interface (= koppeling) van de Universele Terminal is in figuur 17 te zien. De totale seriële data-uitwisseling vindt plaats via PL7. Uit deze figuur wordt ook duidelijk hoe de 25-polige konektor (mannetje oftevel male) op PL7 moet worden aangesloten, via een platte meeraderige verbinding (flat cable). Zolang er geen sprake is van de aansluiting op een modem moeten de pennen 4, 5, 6 en 8 op de konektor onderling worden doorverbonden. Het Data-terminal-ready-signaal ligt aan pen 20

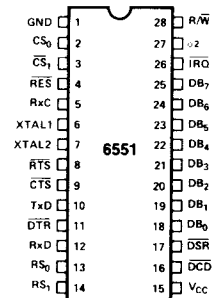


*No interrupt occurs for these conditions.

HARDWARE RESET	7	6	5	4	3	2	1	0
PROGRAM RESET	0	-	-	1	0	0	0	0

83903-14

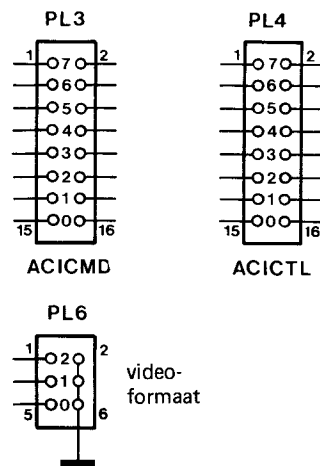
15



83903-15

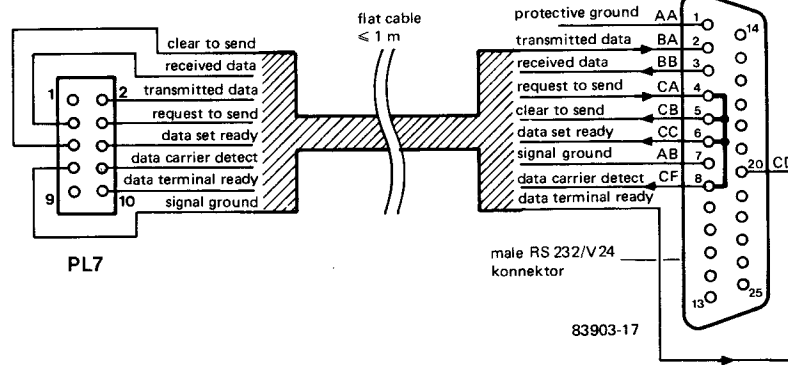
	MSD	0	1	2	3	4	5	6	7
LSD	000	001	010	011	100	101	110	111	
0	0000	NUL	DLE	SP	0	@	P	p	
1	0001	SOH	DC1	"	1	A	Q	a	q
2	0010	STX	DC2	"	2	B	R	b	r
3	0011	ETX	DC3	"	3	C	S	c	s
4	0100	EOT	DC4	"	4	D	T	t	
5	0101	ENG	NAK	%	5	E	U	e	u
6	0110	ACK	SYN	&	6	F	V	f	v
7	0111	BEL	ETB	'	7	G	W	g	w
8	1000	BS	CAN	(8	H	X	h	x
9	1001	HT	EM)	9	I	Y	i	y
A	1010	LF	SUB	*	J	Z	j	z	
B	1011	VT	ESC	.	K	[k		
C	1100	FF	FS	<	L	\	l	l	
D	1101	CR	GS	=	M]	m		
E	1110	SO	RS	>	N	^	n	~	
F	1111	SI	VS	/	O	~	o	DEL	

16

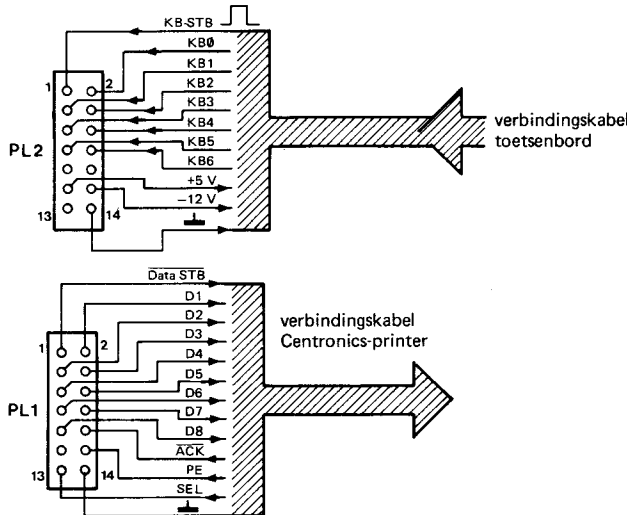


83903-16

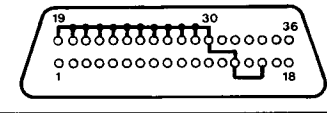
17



18



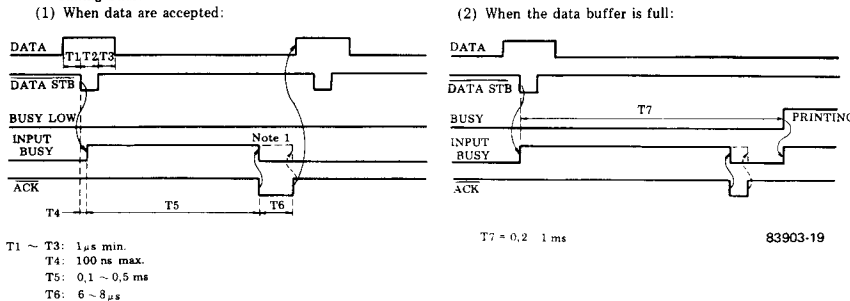
Pin No.	Signal Name	Pin No.	Signal Name
1	DATA STB	19	TWISTED PAIR GND (PIN 1)
2	DATA 1	20	TWISTED PAIR GND (PIN 2)
3	DATA 2	21	TWISTED PAIR GND (PIN 3)
4	DATA 3	22	TWISTED PAIR GND (PIN 4)
5	DATA 4	23	TWISTED PAIR GND (PIN 5)
6	DATA 5	24	TWISTED PAIR GND (PIN 6)
7	DATA 6	25	TWISTED PAIR GND (PIN 7)
8	DATA 7	26	TWISTED PAIR GND (PIN 8)
9	DATA 8	27	TWISTED PAIR GND (PIN 9)
10	ACK	28	TWISTED PAIR GND (PIN 10)
11	INPUT BUSY	29	TWISTED PAIR GND (PIN 11)
12	PE	30	TWISTED PAIR GND (PIN 31)
13	SELECT	31	INPUT-PRIME
14	0 V	32	FAULT
15	NC	33	0 V
16	0 V	34	NC
17	CHASSIS GND	35	NC
18	+5 V DC	36	INPUT-BUSY



83903-18

19

Timing Chart



T1 ~ T3: 1 μs min.
T4: 100 ns max.
T5: 0.1 ~ 0.5 ms
T6: 6 ~ 8 μs

83903-19

en dient voor de melding van de terminal aan de computer, dat hij, d.w.z. de terminal, niet bereid is om op dat moment data te ontvangen. Dat verandert zodra pin 20 een +12 V-nivo aanneemt (negatieve logica).

b. parallel-interface

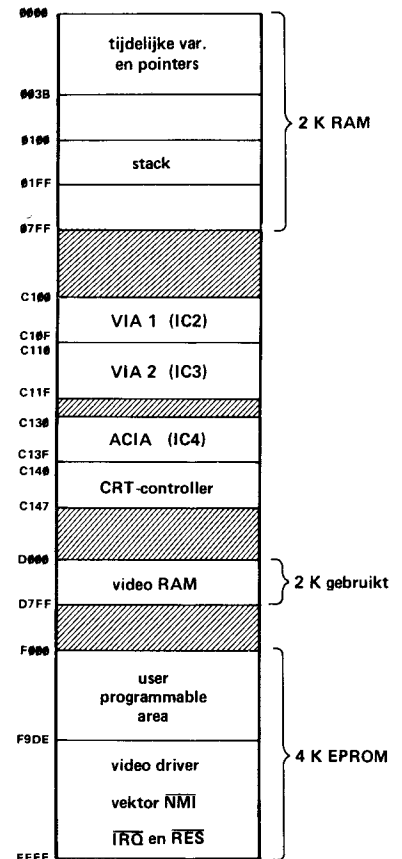
Konnektor PL2 in figuur 18 is bedoeld voor de aansluiting van een parallel toetsbord, met behulp van meeraderig "flat cable". De handshake-leiding CA1 van IC2 is doorverbonden met de keyboard-strobe (KB-STB). In het interrupt-vlagregister van de 6522-VIA wordt er een vlag "1" zodra een positieve KB-STB-puls binnenkomt. De toetsbord-data KB0...KB6 wordt in het ingangs-tussenregister van de VIA opgeslagen. Dat heeft het voordeel dat de (ASCII-) waarde van een ingedrukte toets behouden blijft na het loslaten van de

toets. Aan PL2 zijn de spanningen +5 V en -12 V toegevoerd. Het is dan ook mogelijk om uw toetsbord uit de Universele Terminal te voeden.

c. Centronics-interface (printer)

Een printer, uitgerust met een Centronics-interface, kan via PL1 (zie figuur 18) direkt op de Universele Terminal worden aangesloten. Een garantie voor het probleemloos functioneren van het dataverkeer biedt het aan massa leggen van de zogenaamde retourleidingen ("twisted ground"). Figuur 19 toont het tijdsverloop van de diverse handshake-signalen op de Centronics-stuurbus; deze signalen staan onder supervisie van de software van de Universele Terminal. Figuur 20 geeft de geheugenindeling ("memory map") van de Universele Terminal weer.

20



83903-20

```

0005:
0010: F9DE          ORG   $F9DE
0015:
0020:
0025:          SOURCE LISTING OF A VT-52 COMPATIBLE TERMINAL
0030:
0035:          WRITTEN BY A. NACHTMANN
0040:
0045:
0050:          COPYRIGHT C 1983 ELEKTOR/ELEKTUUR
0055:
0060:
0065:          PROCESSOR:      6502 OR 65C02
0070:          CRT CONTROLLER: 6845 MOTOROLA, ROCKWELL, SYNERTEC
0075:
0080:
0085:          *VT-52 COMMANDS*
0090:
0095:          OTHER COMMANDS SEE COMMAND LOOKUP TABLES
0100:
0105:
0110:          CARRIAGE RETURN
0115:          LINE FEED
0120:          ESC H CURSOR HOME
0125:          ESC A CURSOR UP
0130:          ESC B CURSOR DOWN
0135:          ESC C CURSOR RIGHT
0140:          ESC D CURSOR LEFT
0145:          ESC K ERASE TO END OF LINE
0150:          ESC J ERASE TO END OF SCREEN
0155:
0160:          *POINTERS AND TEMPS*
0165:
0170: F9DE          RAMPTR *      $0000  RAMPONTER
0175: F9DE          AHOLD  *      $0002  SAVE THE CHARACTER HERE
0180: F9DE          FLN    *      $0004  FIRST LINE POINTER
0185: F9DE          CLN    *      FLN    +02  CURRENT LINE POINTER
0190: F9DE          LLN    *      CLN    +02  LAST LINE POINTER
0195: F9DE          SCRPT *      LLN    +02  SLAVE SCREEN POINTER
0200: F9DE          CURSOR *      SCRPT +02  CURRENT CURSOR
0205: F9DE          INLINE *      CURSOR +02  IN LINE WITH WINDOW
0210: F9DE          COL    *      INLINE +01  CURRENT COLUMN
0215: F9DE          ESCFLG *      COL    +01  ESC FLAG
0220: F9DE          TEMCOL *      ESCFLG +01  SLAVE COLUMN
0225: F9DE          RAMBEG *      TEMCOL +01  THE REFRESH RAM STARTS HERE
0230: F9DE          CHAPLN *      RAMBEG +02  CHARACTER/LINE
0235: F9DE          LPSCR  *      CHAPLN +01  LINES/SCREEN
0240: F9DE          FORMAT *      LPSCR  +01  SCREEN FORMAT
0245: F9DE          INDEX  *      FORMAT +01  SOFTWARE STACK FOR CHAR. INPUT
0250: F9DE          BUFFER *      INDEX  +01
0255: F9DE          TABLE *      BUFFER +08  CRT FORMAT TABLE
0260: F9DE          JMPVEC *      TABLE +12  COMMAND ADDRESS POINTER
0265: F9DE          CENFLG *      JMPVEC +02  CENTRONICS FLAG
0270: F9DE          DUPLEX *      CENFLG +01  0=FULL; 1=HALF DUPLEX
0275: F9DE          AUTOLF *      DUPLEX +01  0=NO; 1=AUTO CRLF TO CENTRONICS
0280: F9DE          NMIVEC *      AUTOLF +01  (NMI VECTOR)

```

```

0285: F9DE      IRQVEC *      NMIVEC +02 (IRQ VECTOR)
0290:
0295:
0300:          *SPECIAL CHARACTERS*
0305:
0310: F9DE      SPACE *      $0020
0315: F9DE      ESC   *      $001B
0320:
0325:
0330:          *JUNIOR'S SERIAL I/O*
0335:
0340:
0345:
0350:
0355:          *VIA 1 ADDRESSES*
0360:
0365: F9DE      VAPBD *      $C100  PORT B DATA
0370: F9DE      VAPAD *      VAPBD  +01 PORT A DATA
0375: F9DE      VAPBDD *     VAPBD  +02 PORT B DATA DIRECTION
0380: F9DE      VAPADD *     VAPBD  +03 PORT A DATA DIRECTION
0385: F9DE      VATACL *     VAPBD  +04 T1, LATCH-LOW, COUNTER-LOW
0390: F9DE      VATACH *     VAPBD  +05 T1, COUNTER-HIGH
0395: F9DE      VATALL *     VAPBD  +06 T1, LATCH-LOW
0400: F9DE      VATALH *     VAPBD  +07 T1, LATCH-HIGH
0405: F9DE      VATBCL *     VAPBD  +08 T2, LATCH-LOW, COUNTER-LOW
0410: F9DE      VATBCH *     VAPBD  +09 T2, COUNTER-HIGH
0415: F9DE      VASR   *     VAPBD  +0A SHIFT REGISTER
0420: F9DE      VAACR *     VAPBD  +0B AUXILIARY CONTROL REGISTER
0425: F9DE      VAPCR *     VAPBD  +0C PERIPHERAL CONTROL REGISTER
0430: F9DE      VAIFR *     VAPBD  +0D INTERRUPT FLAG REGISTER
0435: F9DE      VAIER *     VAPBD  +0E INTERRUPT ENABLE REGISTER
0440: F9DE      VAPADN *    VAPBD  +0F PORT A DATA, NO HANDSHAKE
0445:
0450:          *VIA 2 ADDRESSES*
0455:
0460: F9DE      VBPBD *      $C110  PORT B DATA
0465: F9DE      VBPAD *      VBPBD  +01 PORT A DATA
0470: F9DE      VBPBDD *     VBPBD  +02 PORT B DATA DIRECTION
0475: F9DE      VBPADD *     VBPBD  +03 PORT A DATA DIRECTION
0480: F9DE      VBTACL *     VBPBD  +04 T1, LATCH-LOW, COUNTER-LOW
0485: F9DE      VBTACH *     VBPBD  +05 T1, COUNTER-HIGH
0490: F9DE      VBTALL *     VBPBD  +06 T1, LATCH-LOW
0495: F9DE      VBTALH *     VBPBD  +07 T1, LATCH-HIGH
0500: F9DE      VBTBCL *     VBPBD  +08 T2, LATCH-LOW, COUNTER-LOW
0505: F9DE      VBTBCH *     VBPBD  +09 T2, COUNTER-HIGH
0510: F9DE      VBSR   *     VBPBD  +0A SHIFT REGISTER
0515: F9DE      VBACR *     VBPBD  +0B AUXILIARY CONTROL REGISTER
0520: F9DE      VBPCR *     VBPBD  +0C PERIPHERAL CONTROL REGISTER
0525: F9DE      VBIFR *     VBPBD  +0D INTERRUPT FLAG REGISTER
0530: F9DE      VBIER *     VBPBD  +0E INTERRUPT ENABLE REGISTER
0535: F9DE      VBPADN *    VBPBD  +0F PORT A DATA, NO HANDSHAKE
0540:
0545:          *ACIA ADDRESSES*
0550:
0555: F9DE      RECREG *     $C130  RECEIVER REGISTER
0560: F9DE      TRAREG *     RECREG  TRANSMITTER REGISTER

```

```

0565: F9DE      ACIASR *      RECREG +01 STATUS REGISTER
0570: F9DE      ACICMD *      RECREG +02 COMMAND REGISTER
0575: F9DE      ACICTL *      RECREG +03 CONTROL REGISTER
0580:
0585:          *CRT ADDRESSES*
0590:
0595: F9DE      AR      *      $C140 ADDRESS REGISTER OF THE CRT
0600: F9DE      RFILE *      AR      +01 REGISTER FILE OF CRT
0605:
0610:
0615:
0620:
0625:          *XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX*
0630:          * VT 52 COMPATIBLE TERMINAL *
0635:          *XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX*
0640:
0645:
0650:          *INTERNAL 6845 REGISTER FILE*
0655:
0660: F9DE      HORTOT *      $0000 HORIZONTAL TOTAL-1
0665: F9DE      HORDIS *      $0001 HORIZONTAL DISPLAYED
0670: F9DE      HSYPOS *      $0002 HORIZONTAL SYNC POSITION
0675: F9DE      VHSYWI *      $0003 VERTICAL, HORIZONTAL SYNC WIDTH
0680: F9DE      VERTOT *      $0004 VERTICAL TOTAL-1
0685: F9DE      VTOTAJ *      $0005 VERTICAL TOTAL ADJUST
0690: F9DE      VERDIS *      $0006 VERTICAL DISPLAYED
0695: F9DE      VSYPOS *      $0007 VERTICAL SYNC POSITION
0700: F9DE      MODE   *      $0008 INTERLACE MODE REGISTER
0705: F9DE      SCANLN *      $0009 CHARACTER SCAN LINES-1
0710: F9DE      CURSTA *      $000A CURSOR START
0715: F9DE      CUREND *      $000B CURSOR END
0720: F9DE      DSPSTH *      $000C DISPLAY START HIGH
0725: F9DE      DSPSTL *      $000D DISPLAY START LOW
0730: F9DE      CURPOH *      $000E CURSOR POSITION HIGH
0735: F9DE      CURPOL *      $000F CURSOR POSITION LOW
0740: F9DE      LIPENH *      $0010 LIGHT PEN HIGH
0745: F9DE      LIPENL *      $0011 LIGHT PEN LOW
0750: F9DE      UPDATA *      $0012 UPDATE HIGH (ROCKWELL, SYNERTEK)
0755: F9DE      UPDATL *      $0013 UPDATE LOW
0760: F9DE      DUMMY  *      $001F DUMMY REGISTER
0765:
0770:
0775:
0780:
0785:
0790:          ***VIDEO HANDLER***
0795:
0800: F9DE 48      VIDEO PHA          SAVE THE REGISTERS
0805: F9DF 8A          TXA
0810: F9E0 48          PHA
0815: F9E1 98          TYA
0820: F9E2 48          PHA
0825: F9E3 A5 02      LDA AHOLD GET THE CHARACTER
0830: F9E5 C9 1B      CMPIM ESC IS IT THE ESC CHAR.?
0835: F9E7 D0 15      BNE CHECK
0840: F9E9 85 10      STA ESCFLG SET THE ESC FLAG

```

```

0845:
0850: F9EB 68          VIDEND PLA          RESTORE THE REGISTERS
0855: F9EC A8          TAY
0860: F9ED 68          PLA
0865: F9EE AA          TAX
0870: F9EF 68          PLA
0875: F9F0 EA          NOP                $2C
0880: F9F1 EA          NOP                $80 BIT SPAD CHECK FOR BREAK
                                DURING OUTPUT
0885: F9F2 EA          NOP                $FA OR $1A
0890: F9F3 EA          NOP                $10 BPL BRKTST
0895: F9F4 EA          NOP                $01
0900: F9F5 60          RTS
0905:
0910: F9F6 EA          BRKTST NOP         $2C BIT SPAD WAIT TILL KEY
                                IS RELEASED
0915: F9F7 EA          NOP                $80
0920: F9F8 EA          NOP                $FA OR $1A
0925: F9F9 EA          NOP                $10 BPL BRKTST
0930: F9FA EA          NOP                $FB
0935: F9FB EA          NOP                $6C JMP ($FA7C) GOTO BREAK HANL
0940: F9FC EA          NOP                $7C
0945: F9FD EA          NOP                $FA OR $1A
0950:
0955: F9FE 20 E7 FC    CHECK JSR          COMCOM COMPUTE THE COMMAND INDEX
0960: FA01 B0 0E          BCS          VALVEC THERE WAS NO COMMAND
0965: FA03 BD 4F FD    LDAX        COMADR SET THE COMMAND VECTOR
0970: FA06 85 32          STA          JMPVEC
0975: FA08 E8          INX
0980: FA09 BD 4F FD    LDAX        COMADR
0985: FA0C 85 33          STA          JMPVEC +01
0990: FA0E 6C 32 00    JMI          JMPVEC
0995:
1000: FA11 4C 7F FA    VALVEC JMP          VALID
1005:
1010:
1015:                ***EXECUTE A VIDEO COMMAND***
1020:
1025:
1030:                *CARRIAGE RETURN*
1035:
1040: FA14 A2 00          RETURN LDXIM $00
1045: FA16 86 0F          STX          COL          COL=0
1050: FA18 20 EE FA    JSR          ADJUST ADJUST THE CURSOR
1055: FA1B 4C EB F9    JMP          VIDEND
1060:
1065:                *LINE FEED*
1070:
1075: FA1E 20 52 FC    FEED JSR          CURDN
1080: FA21 4C EB F9    JMP          VIDEND
1085:
1090:                *CLEAR SCREEN & HOME CURSOR*
1095:
1100: FA24 20 97 FC    CLRHOM JSR          HOME
1105: FA27 20 D9 FB    JSR          ERTEOS
1110: FA2A 4C EB F9    JMP          VIDEND
1115:
1120:                *CURSOR LEFT*

```



```

1125:
1130: FA2D 20 8D FC LEFT JSR CURLFT
1135: FA30 4C EB F9 JMP VIDEND
1140:
1145: *HOME CURSOR*
1150:
1155: FA33 20 97 FC HOCU JSR HOME
1160: FA36 4C EB F9 JMP VIDEND
1165:
1170:
1175:
1180: *CURSOR UP*
1185:
1190: FA39 20 71 FC UP JSR CURUP
1195: FA3C 4C EB F9 JMP VIDEND
1200:
1205: *CURSOR DOWN*
1210:
1215: FA3F 20 52 FC DOWN JSR CURDN
1220: FA42 4C EB F9 JMP VIDEND
1225:
1230: *CURSOR RIGHT*
1235:
1240: FA45 20 92 FC RIGHT JSR CURRGT
1245: FA48 4C EB F9 JMP VIDEND
1250:
1255: *ERASE TO END OF LINE*
1260:
1265: FA4B 20 AE FB ERLNX JSR ERTEOL
1270: FA4E 4C EB F9 JMP VIDEND
1275:
1280: *ERASE TO END OF SCREEN*
1285:
1290: FA51 20 D9 FB ERSCRX JSR ERTEOS
1295: FA54 4C EB F9 JMP VIDEND
1300:
1305:
1310:
1315:
1320: *DELETE THE CURRENT LINE*
1325:
1330: FA57 A2 00 DELLIN LDXIM $00
1335: FA59 86 0F STX COL
1340: FA5B 20 35 FB JSR ADJCUR MOVE CURSOR TO COL 0
1345: FA5E 20 AE FB JSR ERTEOL
1350: FA61 4C EB F9 JMP VIDEND
1355:
1360: *TOGGLE THE CENTRONICS FLAG*
1365:
1370: FA64 A5 34 TOGGLE LDA CENFLG
1375: FA66 49 01 EORIM $01
1380: FA68 85 34 STA CENFLG
1385: FA6A 4C EB F9 JMP VIDEND
1390:
1395: *TOGGLE THE AUTO CRLF FLAG*
1400:

```

```

1405: FA6D A5 36      TOGLF  LDA   AUTOLF
1410: FA6F 49 01      EORIM $01
1415: FA71 85 36      STA   AUTOLF
1420: FA73 4C EB F9   JMP   VIDEND
1425:
1430:                *TOGGLE THE DUPLEX FLAG*
1435:
1440: FA76 A5 35      TOGDUP LDA   DUPLEX
1445: FA78 49 01      EORIM $01
1450: FA7A 85 35      STA   DUPLEX
1455: FA7C 4C EB F9   JMP   VIDEND
1460:
1465:
1470:
1475:                *FILTER < SPACE*
1480:
1485: FA7F A5 02      VALID  LDA   AHOLD
1490: FA81 C9 20      CMPIM /
1495: FA83 B0 03      BCS   TOSCR
1500: FA85 4C EB F9   JMP   VIDEND
1505:
1510:                *TRANSFER TO SCREEN*
1515:
1520: FA88 20 29 FC   TOSCR  JSR   TPUT
1525: FA8B 4C EB F9   JMP   VIDEND
1530:
1535:
1540:
1545:                ***SUBROUTINES***
1550:
1555:
1560:                *FIRST LINE UP*
1565:
1570: FA8E 38          FLNUP  SEC
1575: FA8F A5 04          LDA   FLN
1580: FA91 E5 14          SBC   CHAPLN  FLN=FLN-CHAPLN
1585: FA93 85 04          STA   FLN
1590: FA95 A5 05          LDA   FLN      +01
1595: FA97 E9 00          SBCIM $00
1600: FA99 29 07          ANDIM $07      MAX IS $7FF
1605: FA9B 85 05          STA   FLN      +01
1610: FA9D 60          RTS
1615:
1620:                *CURRENT LINE UP*
1625:
1630: FA9E 38          CLNUP  SEC
1635: FA9F A5 06          LDA   CLN
1640: FAA1 E5 14          SBC   CHAPLN
1645: FAA3 85 06          STA   CLN
1650: FAA5 A5 07          LDA   CLN      +01
1655: FAA7 E9 00          SBCIM $00
1660: FAA9 29 07          ANDIM $07      MAX IS $7FF
1665: FAAB 85 07          STA   CLN      +01
1670: FAAD 60          RTS
1675:
1680:                *LAST LINE UP*

```

```

1685:
1690: FAAE 38          LLNUP  SEC
1695: FAAF A5 08      LDA    LLN
1700: FAB1 E5 14      SBC   CHAPLN
1705: FAB3 85 08      STA   LLN
1710: FAB5 A5 09      LDA   LLN      +01
1715: FAB7 E9 00      SBCIM $00
1720: FAB9 29 07      ANDIM $07      MAX IS $7FF
1725: FABB 85 09      STA   LLN      +01
1730: FABD 60          RTS
1735:
1740:                *FIRST LINE DOWN*
1745:
1750: FABE 18          FLNDN  CLC
1755: FABF A5 04      LDA   FLN
1760: FAC1 65 14      ADC   CHAPLN
1765: FAC3 85 04      STA   FLN      FLN=FLN+CHAPLN
1770: FAC5 A5 05      LDA   FLN      +01
1775: FAC7 69 00      ADCIM $00
1780: FAC9 29 07      ANDIM $07      MAX IS $7FF
1785: FACB 85 05      STA   FLN      +01
1790: FACD 60          RTS
1795:
1800:                *CURRENT LINE DOWN*
1805:
1810: FACE 18          CLNDN  CLC
1815: FACF A5 06      LDA   CLN
1820: FAD1 65 14      ADC   CHAPLN
1825: FAD3 85 06      STA   CLN      CLN=CLN+CHAPLN
1830: FAD5 A5 07      LDA   CLN      +01
1835: FAD7 69 00      ADCIM $00
1840: FAD9 29 07      ANDIM $07      MAX IS $7FF
1845: FADB 85 07      STA   CLN      +01
1850: FADD 60          RTS
1855:
1860:                *LAST LINE DOWN*
1865:
1870: FADE 18          LLNDN  CLC
1875: FADF A5 08      LDA   LLN
1880: FAE1 65 14      ADC   CHAPLN
1885: FAE3 85 08      STA   LLN      LLN=LLN+CHAPLN
1890: FAE5 A5 09      LDA   LLN      +01
1895: FAE7 69 00      ADCIM $00
1900: FAE9 29 07      ANDIM $07      MAX IS $7FF
1905: FAEB 85 09      STA   LLN      +01
1910: FAED 60          RTS
1915:
1920:                *ADJUST THE LINE POINTERS*
1925:
1930: FAEE A4 0F      ADJUST LDY    COL      IS COLUMN NEGATIVE?
1935: FAF0 10 1F      BPL   ADSA    BRANCH ON NO
1940: FAF2 C6 0E      DEC   INLINE  GO BACK ONE LINE
1945: FAF4 10 10      BPL   ADJU    BRANCH IF STILL ON SCREEN
1950: FAF6 20 8E FA   JSR   FLNUP   FIRST LINE UP
1955: FAF9 20 79 FB   JSR   FLNCRT  ADJUST DISPLAY START
1960: FAFC 20 07 FC   JSR   ERAFLN  ERASE THE FIRST LINE

```

1965:	FAFF	20	AE	FA		JSR	LLNUP	LAST LINE UP
1970:	FB02	A0	00			LDYIM	\$00	RESET INLINE
1975:	FB04	84	0E			STY	INLINE	
1980:								
1985:	FB06	20	9E	FA	ADJU	JSR	CLNUP	CURRENT LINE UP
1990:	FB09	A4	14			LDY	CHAPLN	
1995:	FB0B	88				DEY		COL=CHAPLN-1
2000:	FB0C	84	0F			STY	COL	
2005:	FB0E	4C	35	FB		JMP	ADJCUR	
2010:								
2015:								
2020:	FB11	C4	14		ADSA	CPY	CHAPLN	IS COL>=CHAPLN?
2025:	FB13	90	20			BCC	ADJCUR	BRANCH ON NO
2030:	FB15	A0	00			LDYIM	\$00	
2035:	FB17	84	0F			STY	COL	COL=0
2040:	FB19	E6	0E			INC	INLINE	MOVE DOWN 1 LINE
2045:	FB1B	A4	0E			LDY	INLINE	
2050:	FB1D	C4	15			CPY	LPSCR	STILL ON SCREEN?
2055:	FB1F	90	11			BCC	ADJV	IS INLINE>=LPSCR?
2060:	FB21	20	BE	FA		JSR	FLNDN	FIRST LINE DOWN
2065:	FB24	20	79	FB		JSR	FLNCRT	ADJUST DISPLAY START
2070:	FB27	20	DE	FA		JSR	LLNDN	LAST LINE DOWN
2075:	FB2A	A4	15			LDY	LPSCR	
2080:	FB2C	88				DEY		
2085:	FB2D	84	0E			STY	INLINE	INLINE=LPSCR-1
2090:	FB2F	20	18	FC		JSR	ERALLN	ERASE LAST LINE
2095:								
2100:	FB32	20	CE	FA	ADJV	JSR	CLNDN	CURRENT LINE DOWN
2105:								
2110:	FB35	A2	00		ADJCUR	LDXIM	\$00	
2115:	FB37	86	0C			STX	CURSOR	RESET CURSOR
2120:	FB39	86	0D			STX	CURSOR	+01
2125:	FB3B	A6	0E			LDX	INLINE	IS INLINE=0?
2130:	FB3D	F0	0E			BEQ	ACURX	BRANCH ON YES
2135:								
2140:	FB3F	18			ACURA	CLC		
2145:	FB40	A5	14			LDA	CHAPLN	
2150:	FB42	65	0C			ADC	CURSOR	
2155:	FB44	85	0C			STA	CURSOR	
2160:	FB46	90	02			BCC	ACURB	
2165:	FB48	E6	0D			INC	CURSOR	+01 CURSOR=INLINE*CHAPLN
2170:								
2175:	FB4A	CA			ACURB	DEX		
2180:	FB4B	D0	F2			BNE	ACURA	
2185:								
2190:	FB4D	18			ACURX	CLC		
2195:	FB4E	A5	04			LDA	FLN	
2200:	FB50	65	0C			ADC	CURSOR	
2205:	FB52	85	0C			STA	CURSOR	CURSOR=CURSOR+FLN
2210:	FB54	A5	05			LDA	FLN	+01
2215:	FB56	65	0D			ADC	CURSOR	+01
2220:	FB58	85	0D			STA	CURSOR	+01
2225:								
2230:	FB5A	18				CLC		
2235:	FB5B	A5	0F			LDA	COL	
2240:	FB5D	65	0C			ADC	CURSOR	CURSOR=CURSOR+COL

```

2245: FB5F 85 0C          STA  CURSOR
2250: FB61 90 02          BCC  ACURC
2255: FB63 E6 0D          INC  CURSOR +01
2260:
2265: FB65 A2 0E          ACURC LDXIM CURPOH
2270: FB67 A5 0D          LDA  CURSOR +01
2275: FB69 8E 40 C1       STX  AR
2280: FB6C 8D 41 C1       STA  RFILE  CURSOR--->CRT CONTROLLER
2285: FB6F E8              INX
2290: FB70 A5 0C          LDA  CURSOR
2295: FB72 8E 40 C1       STX  AR
2300: FB75 8D 41 C1       STA  RFILE
2305: FB78 60              RTS
2310:
2315:                      *FIRST LINE TO CRT*
2320:
2325: FB79 A2 0C          FLNCRT LDXIM DSPSTH
2330: FB7B A5 05          LDA  FLN      +01
2335: FB7D 8E 40 C1       STX  AR
2340: FB80 8D 41 C1       STA  RFILE
2345: FB83 E8              INX
2350: FB84 A5 04          LDA  FLN
2355: FB86 8E 40 C1       STX  AR
2360: FB89 8D 41 C1       STA  RFILE
2365: FB8C 60              RTS
2370:
2375:
2380:                      *COMPUTE THE CURRENT RAM POINTER*
2385:
2390: FB8D 18              CRAMPT CLC
2395: FB8E A5 0A          LDA  SCRPTR
2400: FB90 65 12          ADC  RAMBEG  RAMPTR=SCRPTR+RAMBEG
2405: FB92 85 00          STA  RAMPTR
2410: FB94 A5 0B          LDA  SCRPTR +01
2415: FB96 65 13          ADC  RAMBEG +01
2420: FB98 29 D7          ANDIM $D7  MAX IS $D7FF
2425: FB9A 85 01          STA  RAMPTR +01
2430: FB9C 18              CLC
2435: FB9D A5 00          LDA  RAMPTR  RAMPTR=RAMBEG+SCRPTR+TEMCOL
2440: FB9F 65 11          ADC  TEMCOL
2445: FBA1 85 00          STA  RAMPTR
2450: FBA3 90 08          BCC  CRAMP
2455: FBA5 E6 01          INC  RAMPTR +01
2460: FBA7 A5 01          LDA  RAMPTR +01
2465: FBA9 29 D7          ANDIM $D7  MAX IS $D7FF
2470: FBAB 85 01          STA  RAMPTR +01
2475:
2480: FBAD 60              CRAMP  RTS
2485:
2490:                      *ERASE TO END OF LINE*
2495:
2500: FBAE A6 06          ERTEOL LDX  CLN
2505: FBB0 A4 07          LDY  CLN      +01
2510: FBB2 86 0A          STX  SCRPTR  SCRPTR=CLN
2515: FBB4 84 0B          STY  SCRPTR +01
2520: FBB6 A4 0F          LDY  COL      GET THE CURR. COLUMN

```

```

2525: FBB8 84 11          STY  TEMCOL  TEMCOL=COL
2530: FBBA A2 20          LDXIM '      SPACE TO X
2535: FBBC A0 00          LDYIM $00    INDEX=0
2540:
2545: FBBE 20 8D FB      EREOL  JSR    CRAMPT  RAMPTR=RAMBEG+SCRPTR+TEMCOL
2550:
2555: FBC1 8A              EROLX  TXA      SPACE--->RAM
2560: FBC2 91 00          STAIY  RAMPTR
2565: FBC4 E6 11          INC   TEMCOL  TEMCOL=TEMCOL+1
2570: FBC6 E6 00          INC   RAMPTR
2575: FBC8 D0 08          BNE   EROXX
2580: FBCA E6 01          INC   RAMPTR  +01  RAMPTR=RAMPTR+1
2585: FBCC A5 01          LDA   RAMPTR  +01
2590: FBCE 29 D7          ANDIM $D7    MAX IS $D7FF
2595: FBD0 85 01          STA   RAMPTR  +01
2600:
2605: FBD2 A5 11          EROXX  LDA   TEMCOL  IS TEMCOL>=CHAPLN?
2610: FBD4 C5 14          CMP   CHAPLN
2615: FBD6 90 E9          BCC   EROLX  BRANCH ON NO
2620: FBD8 60              RTS
2625:
2630:                      *ERASE TO END OF SCREEN*
2635:
2640: FBD9 A6 0E          ERTEOS LDX   INLINE
2645: FBDB E8              INX                      LAST LINE?
2650: FBDC E4 15          CPX   LPSCR
2655: FBDE F0 CE          BEQ   ERTEOL  IF YES DO ONLY "EOL"
2660: FBE0 20 AE FB      JSR   ERTEOL  DO "EOL" FOR THIS LINE
2665:
2670: FBE3 18              EREOS  CLC
2675: FBE4 A5 0A          LDA   SCRPTX
2680: FBE6 65 14          ADC   CHAPLN  SCRPTX=SCRPTX+CHAPLN
2685: FBE8 85 0A          STA   SCRPTX  OR SCREEN POINTER DOWN
2690: FBEA A5 0B          LDA   SCRPTX  +01
2695: FBEC 69 00          ADCIM $00
2700: FBEE 29 07          ANDIM $07    MAX IS $7FF
2705: FBF0 85 0B          STA   SCRPTX  +01
2710: FBF2 C5 09          CMP   LLN    +01 IS SCRPTX=LLN?
2715: FBF4 D0 06          BNE   EEOL
2720: FBF6 A5 0A          LDA   SCRPTX
2725: FBF8 C5 08          CMP   LLN
2730: FBFA F0 07          BEQ   EEOLB  ERASE LAST LINE AND STOP
2735:
2740: FBFC 84 11          EEOL   STY   TEMCOL  TEMCOL=0
2745: FBFE 20 BE FB      JSR   EREOL
2750: FC01 B0 E0          BCS   EREOS
2755:
2760:
2765: FC03 84 11          EEOLB  STY   TEMCOL  TEMCOL=0
2770: FC05 F0 B7          BEQ   EREOL
2775:
2780:                      *ERASE THE FIRST LINE*
2785:
2790: FC07 A0 00          ERAFLN LDYIM $00
2795: FC09 84 11          STY   TEMCOL
2800: FC0B A2 20          LDXIM '

```

```

2805: FC0D A5 04          LDA    FLN
2810: FC0F 85 0A          STA    SCRPTR  SCRPTR=FLN
2815: FC11 A5 05          LDA    FLN      +01
2820: FC13 85 0B          STA    SCRPTR  +01
2825: FC15 4C BE FB      JMP    EREOL   NOW ERASE THE FIRST LINE
2830:
2835:                    *ERASE THE LAST LINEX
2840:
2845: FC18 A0 00          ERALLN LDYIM $00
2850: FC1A 84 11          STY    TEMCOL  TEMCOL=0
2855: FC1C A2 20          LDXIM  '
2860: FC1E A5 08          LDA    LLN
2865: FC20 85 0A          STA    SCRPTR  SCRPTR=LLN
2870: FC22 A5 09          LDA    LLN      +01
2875: FC24 85 0B          STA    SCRPTR  +01
2880: FC26 4C BE FB      JMP    EREOL   NOW ERASE THE LAST LINE
2885:
2890:                    *PUT A CHARACTER ON THE SCREENX
2895:
2900: FC29 18              TVPUT  CLC
2905: FC2A A5 06          LDA    CLN
2910: FC2C 65 12          ADC    RAMBEG
2915: FC2E 85 00          STA    RAMPTR  RAMPTR=CLN+RAMBEG
2920: FC30 A5 07          LDA    CLN      +01
2925: FC32 65 13          ADC    RAMBEG  +01
2930: FC34 85 01          STA    RAMPTR  +01
2935: FC36 A0 00          LDYIM $00
2940: FC38 18              CLC
2945: FC39 A5 0F          LDA    COL      RAMPTR=CLN+RAMBEG+COL
2950: FC3B 65 00          ADC    RAMPTR
2955: FC3D 85 00          STA    RAMPTR
2960: FC3F 90 08          BCC    TPX
2965: FC41 E6 01          INC    RAMPTR  +01
2970: FC43 A5 01          LDA    RAMPTR  +01
2975: FC45 29 D7          ANDIM $D7     MAX IS $D7FF
2980: FC47 85 01          STA    RAMPTR  +01
2985:
2990: FC49 A5 02          TPX    LDA    AHOLD
2995: FC4B 91 00          STAIY RAMPTR
3000: FC4D E6 0F          INC    COL      COL=COL+01
3005: FC4F 4C EE FA      JMP    ADJUST
3010:
3015:                    *CURSOR DOWNX
3020:
3025: FC52 E6 0E          CURDN  INC    INLINE  INLINE=INLINE+1
3030: FC54 20 CE FA      JSR    CLNDN   CURRENT LINE DOWN
3035: FC57 A4 0E          LDY    INLINE
3040: FC59 C4 15          CPY    LPSCR   IS INLINE>=LPSCR?
3045: FC5B 90 11          BCC    CURDNX  BRANCH ON NO
3050: FC5D 20 BE FA      JSR    FLNDN   FIRST LINE DOWN
3055: FC60 20 79 FB      JSR    FLNCRT  FIRST LINE--->GRT
3060: FC63 20 DE FA      JSR    LLNDN   LAST LINE DOWN
3065: FC66 20 18 FC      JSR    ERALLN  ERASE LAST LINE
3070: FC69 A4 15          LDY    LPSCR
3075: FC6B 88              DEY
3080: FC6C 84 0E          STY    INLINE  INLINE=LPSCR-1

```

```

3085:
3090: FC6E 4C 35 FB CURDNX JMP ADJCUR ADJUST THE CURSOR AND RETURN
3095:
3100: *CURSOR UP*
3105:
3110: FC71 C6 0E CURUP DEC INLINE INLINE=INLINE-1
3115: FC73 20 9E FA JSR CLNUP CURRENT LINE UP
3120: FC76 A4 0E LDY INLINE IS INLINE=NEGATIVE?
3125: FC78 10 10 BPL CURUPX BRANCH ON NO
3130: FC7A 20 8E FA JSR FLNUP FIRST LINE UP
3135: FC7D 20 79 FB JSR FLNCRT FLN--->CRT
3140: FC80 20 AE FA JSR LLNUP LAST LINE UP
3145: FC83 20 07 FC JSR ERAFLN ERASE THE FIRST LINE
3150: FC86 A0 00 LDYIM $00
3155: FC88 84 0E STY INLINE INLINE=0
3160:
3165: FC8A 4C 35 FB CURUPX JMP ADJCUR ADJUST THE CURSOR AND RETURN
3170:
3175: *CURSOR LEFT*
3180:
3185: FC8D C6 0F CURLFT DEC COL COL=COL-1
3190: FC8F 4C EE FA JMP ADJUST ADJUST THE CURSOR AND RETURN
3195:
3200: *CURSOR RIGHT*
3205:
3210: FC92 E6 0F CURRGT INC COL COL=COL+1
3215: FC94 4C EE FA JMP ADJUST ADJUST THE CURSOR AND RETURN
3220:
3225: *HOME CURSOR*
3230:
3235: FC97 A2 00 HOME LDXIM $00
3240: FC99 86 0E STX INLINE INLINE=0
3245: FC9B 86 0F STX COL COL=0
3250: FC9D A6 04 LDX FLN
3255: FC9F 86 06 STX CLN CLN=FLN
3260: FCA1 A6 05 LDX FLN +01
3265: FCA3 86 07 STX CLN +01
3270: FCA5 4C EE FA JMP ADJUST ADJUST THE CURSOR AND RETURN
3275:
3280: *MASTER RESET OF THE MEMORY MAPPED VDU*
3285:
3290: FCA8 A9 00 RESET LDAIM $00
3295: FCAA 85 04 STA FLN FLN=$0000
3300: FCAC 85 05 STA FLN +01
3305: FCAE 85 06 STA CLN CLN=$0000
3310: FCB0 85 07 STA CLN +01
3315: FCB2 85 08 STA LLN RESET LAST LINE
3320: FCB4 85 09 STA LLN +01
3325: FCB6 85 0F STA COL COL=0
3330: FCB8 85 0E STA INLINE INLINE=0
3335: FCBA A6 15 LDX LPSCR
3340: FCBC CA DEX X=LPSCR-1
3345:
3350: FCBD 18 RSA CLC
3355: FCBE A5 14 LDA CHAPLN
3360: FCC0 65 08 ADC LLN

```



```

3365: FCC2 85 08          STA   LLN      LLN=(LPSCR-1)*CHAPLN
3370: FCC4 90 02          BCC   RSB
3375: FCC6 E6 09          INC   LLN      +01
3380:
3385: FCC8 CA              RSB   DEX
3390: FCC9 D0 F2          BNE   RSA
3395: FCCB 20 D7 FC      JSR   CRTINT  SET THE CRT TIMING REGISTERS
3400: FCCE 20 97 FC      JSR   HOME    HOME CURSOR
3405: FCD1 20 D9 FB      JSR   ERTEOS  CLEAR THE SCREEN
3410: FCD4 4C 79 FB      JMP   FLNCR   FLN--->CRT AND RETURN
3415:
3420:                      *INITIALIZE THE CRT CONTROLLER*
3425:
3430: FCD7 A2 00          CRTINT LDXIM $00
3435:
3440: FCD9 8E 40 C1      CIA   STX   AR      SET THE FILE INDEX
3445: FCDC B5 20          LDAX  TABLE
3450: FCDE 8D 41 C1      STA   RFILE  VALUE--->FILE
3455: FCE1 E8              INX
3460: FCE2 E0 10          CPXIM $10    SET ONLY THE TIMING REGISTERS
3465: FCE4 D0 F3          BNE   CIA
3470: FCE6 60             RTS
3475:
3480:
3485:
3490:                      *COMPUTE THE COMMAND ADDRESS INDEX*
3495:
3500: FCE7 A2 00          COMCOM LDXIM $00
3505: FCE9 A0 01          LDYIM $01
3510:
3515: FCEB BD 09 FD      COMCOA LDAX  COMTAB
3520: FCEE C5 10          CMP   ESCFLG  FIND THE COMMAND
3525: FCF0 D0 0A          BNE   COMCOB
3530: FCF2 B9 09 FD      LDAY  COMTAB
3535: FCF5 C5 02          CMP   AHOLD
3540: FCF7 D0 03          BNE   COMCOB
3545: FCF9 18             CLC
3550: FCFA 90 08          BCC   COMCOC  C=0 --> X=ADDRESS INDEX
3555:                      AND RESET THE ESC-FLAG
3560: FCFC E8             COMCOB INX
3565: FCFD E8             INX
3570: FCFE C8             INY
3575: FCFF C8             INY
3580: FD00 E0 36          CPXIM $36
3585: FD02 90 E7          BCC   COMCOA  C=1 --> NO COMMAND
3590:
3595: FD04 A9 00          COMCOC LDAIM $00
3600: FD06 85 10          STA   ESCFLG  RESET THE ESCAPE FLAG
3605: FD08 60             RTS
3610:
3615:
3620:                      *COMMAND TABLE*
3625:
3630: FD09 00          COMTAB =      $00
3635: FD0A 0D          =      $0D    <CR>
3640: FD0B 00          =      $00

```

3645: FD0C 0A	=	\$0A	<LF>
3650: FD0D 00	=	\$00	
3655: FD0E 08	=	\$08	<BS>,<CTL-H> BACK SPACE
3660: FD0F 1B	=	\$1B	
3665: FD10 48	=	^H	<ESC H> CURSOR HOME
3670: FD11 1B	=	\$1B	
3675: FD12 41	=	^A	<ESC A> CURSOR UP
3680: FD13 1B	=	\$1B	
3685: FD14 42	=	^B	<ESC B> CURSOR DOWN
3690: FD15 1B	=	\$1B	
3695: FD16 43	=	^C	<ESC C> CURSOR RIGHT
3700: FD17 1B	=	\$1B	
3705: FD18 44	=	^D	<ESC D> CURSOR LEFT
3710: FD19 1B	=	\$1B	
3715: FD1A 4B	=	^K	<ESC K> ERASE TO EOL
3720: FD1B 1B	=	\$1B	
3725: FD1C 4A	=	^J	<ESC J> ERASE TO EOS
3730:			
3735: FD1D 00	COTABA =	\$00	
3740: FD1E 0B	=	\$0B	<CTL-K> CURSOR UP
3745: FD1F 00	=	\$00	
3750: FD20 0C	=	\$0C	<CTL-L> CURSOR RIGHT
3755: FD21 00	=	\$00	
3760: FD22 11	=	\$11	<CTL-Q> ERASE TO EOS
3765: FD23 00	=	\$00	
3770: FD24 18	=	\$18	<CTL-X> ERASE TO EOL
3775: FD25 00	=	\$00	
3780: FD26 1A	=	\$1A	<CTL-Z> CLEAR SCREEN & HOME
3785: FD27 00	=	\$00	
3790: FD28 1E	=	\$1E	<CTL-^> CURSOR HOME
3795: FD29 00	=	\$00	
3800: FD2A 0A	=	\$0A	<CTL-J> CURSOR DOWN
3805:			
3810: FD2B 00	COTABB =	\$00	
3815: FD2C 10	=	\$10	<CTL-P> SELECT/DESELECT CENTRONICS
3820: FD2D 1B	=	\$1B	
3825: FD2E 52	=	^R	<ESC><R> DELETE LINE
3830: FD2F 1B	=	\$1B	
3835: FD30 2A	=	^X	<ESC><X> CLEAR SCREEN & HOME
3840: FD31 1B	=	\$1B	
3845: FD32 3A	=	^:	<ESC><:> CLEAR SCREEN & HOME
3850: FD33 1B	=	\$1B	
3855: FD34 54	=	^T	<ESC><T> ERASE TO EOL
3860: FD35 1B	=	\$1B	
3865: FD36 74	=	^t	<ESC><t> ERASE TO EOL
3870: FD37 1B	=	\$1B	
3875: FD38 59	=	^Y	<ESC><Y> ERASE TO EOS
3880: FD39 1B	=	\$1B	
3885: FD3A 79	=	^y	<ESC><y> ERASE TO EOS
3890:			
3895: FD3B 00	COTABX =	\$00	
3900: FD3C 06	=	\$06	<CTL-F> SELECT/DESELECT AUTOLF
3905: FD3D 00	=	\$00	
3910: FD3E 02	=	\$02	<CTL-B> SEL/DESEL HALF DUPLEX
3915:			
3920: FD3F FF	=	\$FF	

3925:	FD40	FF	=	\$FF
3930:	FD41	FF	=	\$FF
3935:	FD42	FF	=	\$FF
3940:	FD43	FF	=	\$FF
3945:	FD44	FF	=	\$FF
3950:	FD45	FF	=	\$FF
3955:	FD46	FF	=	\$FF
3960:	FD47	FF	=	\$FF
3965:	FD48	FF	=	\$FF
3970:	FD49	FF	=	\$FF
3975:	FD4A	FF	=	\$FF
3980:	FD4B	FF	=	\$FF
3985:	FD4C	FF	=	\$FF
3990:	FD4D	FF	=	\$FF
3995:	FD4E	FF	=	\$FF

4000:

4005:

COMMAND ADDRESS TABLE

4010:

4015:	FD4F	14	COMADR =	\$14	CR
4020:	FD50	FA	=	\$FA	
4025:	FD51	1E	=	\$1E	LF
4030:	FD52	FA	=	\$FA	
4035:	FD53	2D	=	\$2D	CURSOR LEFT
4040:	FD54	FA	=	\$FA	
4045:	FD55	33	=	\$33	CURSOR HOME
4050:	FD56	FA	=	\$FA	
4055:	FD57	39	=	\$39	CURSOR UP
4060:	FD58	FA	=	\$FA	
4065:	FD59	3F	=	\$3F	CURSOR DOWN
4070:	FD5A	FA	=	\$FA	
4075:	FD5B	45	=	\$45	CURSOR RIGHT
4080:	FD5C	FA	=	\$FA	
4085:	FD5D	2D	=	\$2D	CURSOR LEFT
4090:	FD5E	FA	=	\$FA	
4095:	FD5F	4B	=	\$4B	ERASE TO EOL
4100:	FD60	FA	=	\$FA	
4105:	FD61	51	=	\$51	ERASE TO EOS
4110:	FD62	FA	=	\$FA	

4115:

4120:

NORMAL VIDEO COMMANDS

4125:

4130:

4135:	FD63	39	=	\$39	CURSOR UP
4140:	FD64	FA	=	\$FA	
4145:	FD65	45	=	\$45	CURSOR RIGHT
4150:	FD66	FA	=	\$FA	
4155:	FD67	51	=	\$51	ERASE TO EOS
4160:	FD68	FA	=	\$FA	
4165:	FD69	4B	=	\$4B	ERASE TO EOL
4170:	FD6A	FA	=	\$FA	
4175:	FD6B	24	=	\$24	CLEAR & HOME
4180:	FD6C	FA	=	\$FA	
4185:	FD6D	33	=	\$33	CURSOR HOME
4190:	FD6E	FA	=	\$FA	
4195:	FD6F	3F	=	\$3F	CURSOR DOWN
4200:	FD70	FA	=	\$FA	

4205:	FD71	64	=	\$64	SELECT/DESELECT CENTRONICS	
4210:	FD72	FA	=	\$FA		
4215:	FD73	57	=	\$57	DELETE LINE	
4220:	FD74	FA	=	\$FA		
4225:	FD75	24	=	\$24	CLEAR & HOME	
4230:	FD76	FA	=	\$FA		
4235:	FD77	24	=	\$24	"	
4240:	FD78	FA	=	\$FA		
4245:	FD79	4B	=	\$4B	ERASE TO EOL	
4250:	FD7A	FA	=	\$FA		
4255:	FD7B	4B	=	\$4B	"	
4260:	FD7C	FA	=	\$FA		
4265:	FD7D	51	=	\$51	ERASE TO EOS	
4270:	FD7E	FA	=	\$FA		
4275:	FD7F	51	=	\$51	"	
4280:	FD80	FA	=	\$FA		
4285:	FD81	6D	=	\$6D	SELECT/DESELECT AUTOLF	
4290:	FD82	FA	=	\$FA		
4295:	FD83	76	=	\$76	SEL/DESEL HALF DUPLEX	
4300:	FD84	FA	=	\$FA		
4305:						
4310:	FD85	FF	=	\$FF		
4315:	FD86	FF	=	\$FF		
4320:	FD87	FF	=	\$FF		
4325:	FD88	FF	=	\$FF		
4330:	FD89	FF	=	\$FF		
4335:	FD8A	FF	=	\$FF		
4340:	FD8B	FF	=	\$FF		
4345:	FD8C	FF	=	\$FF		
4350:	FD8D	FF	=	\$FF		
4355:	FD8E	FF	=	\$FF		
4360:	FD8F	FF	=	\$FF		
4365:	FD90	FF	=	\$FF		
4370:	FD91	FF	=	\$FF		
4375:	FD92	FF	=	\$FF		
4380:	FD93	FF	=	\$FF		
4385:	FD94	FF	=	\$FF		
4390:						
4395:						
4400:						
4405:					*CENTRONICS OUTPUT*	
4410:						
4415:	FD95	AD 10 C1	CENTRO	LDA	VBPBD	
4420:	FD98	29 08		ANDIM	\$08	IS THE PRINTER SELECTED?
4425:	FD9A	F0 29		BEQ	CTROB	PB3 = SEL INPUT
4430:	FD9C	AD 10 C1		LDA	VBPBD	
4435:	FD9F	29 10		ANDIM	\$10	PAPER EMPTY?
4440:	FDA1	D0 22		BNE	CTROB	PB4 = PE INPUT
4445:						
4450:	FDA3	A5 02	CTROA	LDA	AHOLD	
4455:	FDA5	8D 00 C1		STA	VAPBD	OUTPUT THE CHARACTER AND
4460:						
4465:	FDA8	AD 10 C1	WAIT	LDA	VBPBD	
4470:	FDAB	29 08		ANDIM	\$08	SEL ?
4475:	FDAD	F0 16		BEQ	CTROB	
4480:	FDAF	AD 10 C1		LDA	VBPBD	

```

4485: FDB2 29 10      ANDIM $10      FE ?
4490: FDB4 D0 0F      BNE      CTROB
4495: FDB6 AD 0D C1    LDA      VAIFR  WAIT FOR ACKNOWLEDGE
4500: FDB9 29 10      ANDIM $10      SAMPLE THE CB1 FLAG
4505: FDBB F0 EB      BEQ      WAIT
4510: FDBD AD 0D C1    LDA      VAIFR  RESET CB1 & CB2 FLAGS
4515: FDC0 09 18      ORAIM $18
4520: FDC2 8D 0D C1    STA      VAIFR
4525:
4530: FDC5 60          CTROB  RTS
4535:
4540:
4545:
4550:                *INITIALIZE CENTRONICS*
4555:
4560: FDC6 A9 FF      INICEN LDAIM $FF      PB0...PB7 = OUTPUT
4565: FDC8 8D 02 C1    STA      VAPBDD
4570: FDCB A9 A0      LDAIM $A0      CB2 = WRITE HANDSHAKE PULSE OUTPUT
4575: FDCD 8D 0C C1    STA      VAPCR  CB1 = NEG. EDGE SENSITIVE
4580: FDD0 AD 12 C1    LDA      VBPBDD
4585: FDD3 29 E7      ANDIM $E7      PB3,PB4 = INPUT
4590: FDD5 8D 12 C1    STA      VBPBDD
4595: FDD8 A9 00      LDAIM $00      RESET THE PRINTER FLAG
4600: FDDA 85 34      STA      CENFLG
4605: FDDC 60          RTS
4610:
4615:                *MOVE THE CRT FILE FROM ROM TO RAM*
4620:
4625: FDDD A9 00      MOVCRT LDAIM $00      REFRESH RAM STARTS AT $D000
4630: FDDF A2 D0      LDXIM $D0
4635: FDE1 85 12      STA      RAMBEG
4640: FDE3 86 13      STX      RAMBEG +01
4645: FDE5 A9 00      LDAIM $00
4650: FDE7 A8          TAY
4655: FDE8 A6 16      LDX      FORMAT  GET THE CURR. FORMAT
4660: FDEA F0 06      BEQ      MCRTB
4665:
4670: FDEC 18          MCRTA  CLC
4675: FDED 69 12      ADCIM $12      COMPUTE THE INDEX
4680: FDEF CA          DEX
4685: FDF0 D0 FA      BNE      MCRTA
4690:
4695: FDF2 AA          MCRTB  TAX
4700:
4705: FDF3 BD 12 FE    MCRTC  LDAX  CRTINA
4710: FDF6 99 20 00    STAY  TABLE  MOVE TABLE
4715: FDF9 E8          INX
4720: FDFA C8          INY
4725: FDFB C0 12      CPYIM $12
4730: FDFD D0 F4      BNE      MCRTC
4735: FDFF 88          DEY
4740: FE00 B9 20 00    LDAY  TABLE  SET SCREEN PARAMETERS
4745: FE03 85 15      STA  LPSCR
4750: FE05 88          DEY
4755: FE06 B9 20 00    LDAY  TABLE
4760: FE09 85 14      STA  CHAPLN

```

4765: FE0B 60

RTS

4770:

4775:

INDIRECT NMI & IRQ VECTORS

4780:

4785: FE0C 6C 37 00 NMIJMP JMI NMIVVEC

4790: FE0F 6C 39 00 IRQJMP JMI IRQVEC

4795:

4800:

4805:

CRT TIMING TABLES

4810:

15MHz

- 80x24 -

16MHz



4820:

4825: FE12 80 78 CRTINA = 0 \$80 HORIZONTAL TOTAL-1 = 129-1 CHAR.

4830: FE13 50 = 1 \$50 HORIZONTAL DISPLAYED = 80 CHAR.

4835: FE14 60 60 = 2 \$60 HORIZ. SYNC. POSITION = 96 CHAR.

4840: FE15 08 = 3 \$08 VERT./HORIZ. SYNC WIDTH = 16/8

4845: FE16 22 = 4 \$22 VERTICAL TOTAL-1 = 34 CHAR. LINE

4850: FE17 00 = 5 \$00 VERT. TOTAL ADJ. = 0x64 MICRO SE

4855: FE18 18 = 6 \$18 VERTICAL DISPLAYED = 24 LINES

4860: FE19 1C = 7 \$1C VERT. SYNC. POSITION = 29 CHAR. L.

4865: FE1A 00 = 8 \$00 MODE CONTROL

4870: FE1B 08 = 9 \$08 SCAN LINES-1 = 9-1

4875: FE1C 00 = 10 \$00 CURSOR START

4880: FE1D 09 = 11 \$09 CURSOR END

4885: FE1E 00 = 12 \$00 DISPLAY START (NOT NEEDED)

4890: FE1F 00 = 13 \$00

4895: FE20 00 = 14 \$00 CURSOR POSITION (NOT NEEDED)

4900: FE21 00 = 15 \$00

4905: FE22 50 = 16 \$50 CHARACTERS/LINE

4910: FE23 18 = 17 \$18 LINES/SCREEN

4915:

4920:

- 80x25 -

4930:

4935: FE24 80 = \$80

4940: FE25 50 = \$50

4945: FE26 60 = \$60

4950: FE27 08 = \$08

4955: FE28 22 = \$22

4960: FE29 00 = \$00

4965: FE2A 19 = \$19

4970: FE2B 1C = \$1C

4975: FE2C 00 = \$00

4980: FE2D 08 = \$08

4985: FE2E 00 = \$00

4990: FE2F 09 = \$09

4995: FE30 00 = \$00

5000: FE31 00 = \$00

5005: FE32 00 = \$00

5010: FE33 00 = \$00

5015: FE34 50 = \$50

5020: FE35 19 = \$19

5025:

5030:

- 64x16 -

5035:

5040: FE36 64 = \$64

5045:	FE37	40	=	\$40
5050:	FE38	49	=	\$49
5055:	FE39	05	=	\$05
5060:	FE3A	16	=	\$16
5065:	FE3B	0E	=	\$0E
5070:	FE3C	10	=	\$10
5075:	FE3D	12	=	\$12
5080:	FE3E	00	=	\$00
5085:	FE3F	0C	=	\$0C
5090:	FE40	00	=	\$00
5095:	FE41	09	=	\$09
5100:	FE42	00	=	\$00
5105:	FE43	00	=	\$00
5110:	FE44	00	=	\$00
5115:	FE45	00	=	\$00
5120:	FE46	40	=	\$40
5125:	FE47	10	=	\$10
5130:				
5135:			- 64x24 -	
5140:				
5145:	FE48	64	=	\$64
5150:	FE49	40	=	\$40
5155:	FE4A	52	=	\$52
5160:	FE4B	05	=	\$05
5165:	FE4C	22	=	\$22
5170:	FE4D	00	=	\$00
5175:	FE4E	18	=	\$18
5180:	FE4F	1C	=	\$1C
5185:	FE50	00	=	\$00
5190:	FE51	08	=	\$08
5195:	FE52	00	=	\$00
5200:	FE53	09	=	\$09
5205:	FE54	00	=	\$00
5210:	FE55	00	=	\$00
5215:	FE56	00	=	\$00
5220:	FE57	00	=	\$00
5225:	FE58	40	=	\$40
5230:	FE59	18	=	\$18
5235:				
5240:			- 90x22 -	
5245:				
5250:	FE5A	87	=	\$87
5255:	FE5B	5A	=	\$5A
5260:	FE5C	6A	=	\$6A
5265:	FE5D	08	=	\$08
5270:	FE5E	22	=	\$22
5275:	FE5F	00	=	\$00
5280:	FE60	16	=	\$16
5285:	FE61	1C	=	\$1C
5290:	FE62	00	=	\$00
5295:	FE63	08	=	\$08
5300:	FE64	00	=	\$00
5305:	FE65	09	=	\$09
5310:	FE66	00	=	\$00
5315:	FE67	00	=	\$00
5320:	FE68	00	=	\$00

5325:	FE69 00	=	\$00
5330:	FE6A 5A	=	\$5A
5335:	FE6B 16	=	\$16
5340:			
5345:	- 48x12 -		
5350:			
5355:	FE6C 46	=	\$46
5360:	FE6D 30	=	\$30
5365:	FE6E 3A	=	\$3A
5370:	FE6F 05	=	\$05
5375:	FE70 16	=	\$16
5380:	FE71 0E	=	\$0E
5385:	FE72 0C	=	\$0C
5390:	FE73 12	=	\$12
5395:	FE74 00	=	\$00
5400:	FE75 0C	=	\$0C
5405:	FE76 00	=	\$00
5410:	FE77 09	=	\$09
5415:	FE78 00	=	\$00
5420:	FE79 00	=	\$00
5425:	FE7A 00	=	\$00
5430:	FE7B 00	=	\$00
5435:	FE7C 30	=	\$30
5440:	FE7D 0C	=	\$0C
5445:			
5450:	- 24x24 -		
5455:			
5460:	FE7E 38	=	\$38
5465:	FE7F 18	=	\$18
5470:	FE80 26	=	\$26
5475:	FE81 05	=	\$05
5480:	FE82 22	=	\$22
5485:	FE83 00	=	\$00
5490:	FE84 18	=	\$18
5495:	FE85 1C	=	\$1C
5500:	FE86 00	=	\$00
5505:	FE87 08	=	\$08
5510:	FE88 00	=	\$00
5515:	FE89 09	=	\$09
5520:	FE8A 00	=	\$00
5525:	FE8B 00	=	\$00
5530:	FE8C 00	=	\$00
5535:	FE8D 00	=	\$00
5540:	FE8E 18	=	\$18
5545:	FE8F 18	=	\$18
5550:			
5555:	- 80x24 (USA) -		
5560:			
5565:	FE90 00	=	\$00
5570:	FE91 00	=	\$00
5575:	FE92 00	=	\$00
5580:	FE93 00	=	\$00
5585:	FE94 00	=	\$00
5590:	FE95 00	=	\$00
5595:	FE96 00	=	\$00
5600:	FE97 00	=	\$00


```

5605: FE98 00          =      $00
5610: FE99 00          =      $00
5615: FE9A 00          =      $00
5620: FE9B 00          =      $00
5625: FE9C 00          =      $00
5630: FE9D 00          =      $00
5635: FE9E 00          =      $00
5640: FE9F 00          =      $00
5645: FEA0 50          =      $50
5650: FEA1 18          =      $18
5655:
5660:                *VT 52 DISPLAY*
5665:
5670: FEA2 D8          RESVTA CLD
5675: FEA3 78          SEI
5680: FEA4 A2 FF          LDXIM $FF      RESET THE STACK POINTER
5685: FEA6 9A          TXS
5690: FEA7 20 63 FF          JSR  INIKBD  INIT. PAR. KEYBOARD
5695: FEA8 20 85 FF          JSR  CTLCMD  READ ACIA CONTROL/COMMAND REG.
5700: FEAD 20 AE FF          JSR  GETFOT  GET THE SCREEN FORMAT
5705: FEB0 20 DD FD          JSR  MOVCRT  MOVE THE CRT TABLE & SCREEN FORMAT
5710: FEB3 20 A8 FC          JSR  RESET   INIT. CRT
5715: FEB4 AD 10 C1          LDA  VBPBD  SET READY LOW
5720: FEB9 29 FE          ANDIM $FE
5725: FEBB 8D 10 C1          STA  VBPBD
5730: FEBE AD 12 C1          LDA  VBPBDD
5735: FEC1 09 01          ORAIM $01    PB0=READY-OUTPUT
5740: FEC3 8D 12 C1          STA  VBPBDD
5745: FEC6 A9 00          LDAIM $00
5750: FEC8 85 17          STA  INDEX  NO CHARACTERS IN BUFFER
5755: FECA 85 18          STA  BUFFER
5760: FECC A9 44          LDAIM RECIRQ SET IRQ VECTOR
5765: FECE A2 FF          LDXIM RECIRQ /
5770: FED0 85 39          STA  IRQVEC
5775: FED2 86 3A          STX  IRQVEC +01
5780: FED4 20 C6 FD          JSR  INICEN INITIALIZE CENTRONICS I/O
5785: FED7 A9 00          LDAIM $00
5790: FED9 A2 00          LDXIM $00
5795: FEDB 85 35          STA  DUPLEX SELECT FULL DUPLEX
5800: FEDD 86 36          STX  AUTOLF OUTPUT NO AUTO LF TO CENTRONICS
5805: FEDF 58          CLI
5810:
5815: FEE0 20 76 FF          VTA  JSR  GETKBD
5820: FEE3 D0 2C          BNE  VTAT  ACTIVE KEY?
5825: FEE5 A5 18          LDA  BUFFER
5830: FEE7 F0 F7          BEQ  VTA   ANY CHARACTERS IN BUFFER?
5835: FEE9 A2 00          LDXIM $00
5840:
5845: FEEB E8          VTAS  INX
5850: FEED B5 18          LDAX BUFFER
5855: FEEE 85 02          STA  AHOLD
5860: FEF0 20 DE F9          JSR  VIDEO
5865: FEF3 A5 34          LDA  CENFLG
5870: FEF5 F0 03          BEQ  VTACEN
5875: FEF7 20 25 FF          JSR  SPECEN OUTPUT THE CHAR. TO THE PRINTER
5880:

```

```

5885: FEFA E4 17      VTACEN CPX   INDEX
5890: FEFC D0 ED      BNE   VTAS
5895: FEFE 78         SEI
5900: FEFF A2 00      LDXIM $00
5905: FF01 86 18      STX   BUFFER
5910: FF03 86 17      STX   INDEX
5915: FF05 58         CLI
5920: FF06 AD 10 C1   LDA   VBPBD
5925: FF09 29 FE      ANDIM $FE
5930: FF0B 8D 10 C1   STA   VBPBD   VIA2,PB0 = OUTPUT
5935: FF0E 4C E0 FE    JMP   VTA
5940:
5945: FF11 20 D1 FF    VTAT   JSR   ACIOUT
5950: FF14 A5 35      LDA   DUPLEX
5955: FF16 F0 03      BEQ   VTATXY 0=FULL DUPLEX
5960: FF18 20 DE F9    JSR   VIDEO  ECHO TO SCREEN
5965:
5970: FF1B A5 34      VTATXY LDA   CENFLG
5975: FF1D F0 C1      BEQ   VTA   OUTPUT TO CENTRONICS?
5980: FF1F 20 25 FF    JSR   SPECEN
5985: FF22 4C E0 FE    JMP   VTA
5990:
5995:
6000:
6005:
6010:                *SPECIAL CENTRONICS OUTPUT*
6015:
6020:
6025: FF25 A5 36      SPECEN LDA   AUTOLF
6030: FF27 F0 17      BEQ   VTATZX ADD AN AUTOLF?
6035: FF29 A5 02      LDA   AHOLD
6040: FF2B C9 0D      CMPIM $0D   CHECK FOR CR
6045: FF2D D0 11      BNE   VTATZX
6050: FF2F 20 95 FD    JSR   CENTRO OUTPUT A CR
6055: FF32 A5 02      LDA   AHOLD
6060: FF34 48         PHA
6065: FF35 A9 0A      LDAIM $0A
6070: FF37 85 02      STA   AHOLD
6075: FF39 20 95 FD    JSR   CENTRO OUTPUT A LF
6080: FF3C 68         PLA
6085: FF3D 85 02      STA   AHOLD
6090: FF3F 60         RTS
6095:
6100:
6105: FF40 20 95 FD    VTATZX JSR   CENTRO OUTPUT THE CURR. CHAR.
6110: FF43 60         RTS
6115:
6120: FF44 48         RECIRQ PHA
6125: FF45 AD 10 C1   LDA   VBPBD
6130: FF48 09 01      ORAIM $01   READY=HIGH
6135: FF4A 8D 10 C1   STA   VBPBD
6140: FF4D 8A         TXA
6145: FF4E 48         PHA
6150: FF4F AD 31 C1   LDA   ACIASR RESET THE IRQ LINE
6155: FF52 E6 17      INC   INDEX
6160: FF54 A6 17      LDX   INDEX

```

```

6165: FF56 AD 30 C1      LDA   RECREG GET THE CHARACTER
6170: FF59 95 18        STAX  BUFFER AND STORE IT IN THE FIFO
6175: FF5B A9 01        LDAIM $01
6180: FF5D 85 18        STA   BUFFER THERE ARE CHAR. IN THE FIFO
6185: FF5F 68          PLA
6190: FF60 AA          TAX
6195: FF61 68          PLA
6200: FF62 40          RTI
6205:
6210:
6215:
6220:
6225:
6230:
6235: FF63 A9 01      INIKBD LDAIM $01      ENABLE INPUT LATCHING
6240: FF65 8D 0B C1      STA   VAACR
6245: FF68 AD 0C C1      LDA   VAPCR
6250: FF6B 09 01      ORAIM $01      CA1 = ACTIVE POSITIVE EDGE
6255: FF6D 8D 0C C1      STA   VAPCR
6260: FF70 A9 00      LDAIM $00      PA0...PA7=INPUT
6265: FF72 8D 03 C1      STA   VAPADD
6270: FF75 60          RTS
6275:
6280:
6285:
6290:
6295: FF76 AD 0D C1      GETKBD LDA   VAIFR
6300: FF79 29 02      ANDIM $02      CA1 FLAG = 1?
6305: FF7B F0 07      BEQ   GETKA
6310: FF7D AD 01 C1      LDA   VAPAD READ KEYBOARD DATA AND
6315: FF80 29 7F      ANDIM $7F
6320: FF82 85 02      STA   AHOLD RESET CA1 FLAG
6325:
6330: FF84 60          GETKA RTS
6335:
6340:
6345:
6350:
6355:
6360: FF85 A9 02      CTLCMD LDAIM $02      VBPB1 = OUTPUT
6365: FF87 8D 12 C1      STA   VBPBDD
6370: FF8A A2 00      LDXIM $00      VBPB1 = LOW
6375: FF8C 8E 10 C1      STX   VBPBD
6380: FF8F AD 11 C1      LDA   VBPAD GET ACICMD
6385: FF92 49 FF      EORIM $FF
6390: FF94 8D 32 C1      STA   ACICMD
6395: FF97 A9 04      LDAIM $04      VBPB2 = OUTPUT
6400: FF99 8D 12 C1      STA   VBPBDD
6405: FF9C 8E 10 C1      STX   VBPBD VBPB2 = LOW
6410: FF9F AD 11 C1      LDA   VBPAD GET ACICTL
6415: FFA2 49 FF      EORIM $FF
6420: FFA4 8D 33 C1      STA   ACICTL
6425: FFA7 8E 13 C1      STX   VBPADD VBPAD = INPUT
6430: FFAA 8E 12 C1      STX   VBPBDD VBPBD = INPUT
6435: FFAD 60          RTS
6440:

```

```

6445:
6450:
6455:          *GET FORMAT*
6460:
6465: FFAE AD 12 C1  GETFOT LDA   VBPBDD
6470: FFB1 29 1F          ANDIM $1F   PB7...PB5 = INPUT
6475: FFB3 8D 12 C1          STA   VBPBDD
6480: FFB6 AD 10 C1          LDA   VBPBD
6485: FFB9 49 FF          EORIM $FF
6490: FFBB A2 05          LDXIM $05
6495:
6500: FFBD 4A          GETFA  LSRA
6505: FFBE CA          DEX
6510: FFBF D0 FC          BNE   GETFA
6515: FFC1 85 16          STA   FORMAT
6520: FFC3 60          RTS
6525:
6530:
6535:
6540:          *ACIA RECEIVER*
6545:
6550: FFC4 A9 08          ACIIN  LDAIM $08   RECEIVER MASK
6555: FFC6 2D 31 C1          AND   ACIASR RECEIVER FULL?
6560: FFC9 F0 F9          BEQ   ACIIN
6565: FFCB AD 30 C1          LDA   RECREG
6570: FFCE 85 02          STA   AHOLD   READ & SAVE THE CHAR.
6575: FFD0 60          RTS
6580:
6585:
6590:          ACIA TRANSMITTER*
6595:
6600: FFD1 A9 10          ACIOUT LDAIM $10   TRANSMITTER MASK
6605: FFD3 2D 31 C1          AND   ACIASR TRANSMITTER EMPTY?
6610: FFD6 F0 F9          BEQ   ACIOUT
6615: FFD8 A5 02          LDA   AHOLD
6620: FFDA 8D 30 C1          STA   TRAREG GET AND TRANSMIT THE CHAR.
6625: FFDD 60          RTS
6630:
6635:
6640:          *CHECK TERMINAL ROUTINE*
6645:
6650: FFDE D8          CHE   CLD
6655: FFDF 78          SEI
6660: FFE0 A2 FF          LDXIM $FF
6665: FFE2 9A          TXS
6670: FFE3 20 63 FF          JSR   INIKBD
6675: FFE6 20 AE FF          JSR   GETFOT
6680: FFE9 20 DD FD          JSR   MOVCR
6685: FFEC 20 A8 FC          JSR   RESET
6690:
6695: FFEF 20 76 FF          CHECKA JSR   GETKBD
6700: FFF2 F0 FB          BEQ   CHECKA
6705: FFF4 20 DE F9          JSR   VIDEO
6710: FFF7 4C EF FF          JMP   CHECKA
6715:
6720:
-T

```

SYMBOL TABLE 3400 386E

ACIASR	C131	ACICMD	C132	ACICTL	C133	ACIIN	FFC4
ACIOUT	FFD1	ACURA	FB3F	ACURB	FB4A	ACURC	FB65
ACURX	FB4D	ADJCUR	FB35	ADJU	FB06	ADJUST	FAEE
ADJV	FB32	ADSA	FB11	AHOLD	0002	AR	C140
AUTOLF	0036	BRKTST	F9F6	BUFFER	0018	CENFLG	0034
CENTRO	FD95	CHAPLN	0014	CHECK	F9FE	CHECKA	FFEF
CHE	FFDE	CIA	FCD9	CLNDN	FACE	CLNUP	FA9E
CLN	0006	CLRHM	FA24	COL	000F	COMADR	FD4F
COMCOA	FCEB	COMCOB	FCFC	COMCOC	FD04	COMCOM	FCE7
COMTAB	FD09	COTABA	FD1D	COTABB	FD2B	COTABX	FD3B
CRAMP	FBAD	CRAMPT	FB8D	CRTINA	FE12	CRTINT	FCD7
CTLCMD	FF85	CTROA	FDA3	CTROB	FDC5	CURDN	FC52
CURDNX	FC6E	CUREND	000B	CURLFT	FC8D	CURPOH	000E
CURPOL	000F	CURRGT	FC92	CURSOR	000C	CURSTA	000A
CURUP	FC71	CURUPX	FC8A	DELLIN	FA57	DOWN	FA3F
DSPSTH	000C	DSPSTL	000D	DUMMY	001F	DUPLEX	0035
EEOL	FBFC	EEOLB	FC03	ERAFLN	FC07	ERALLN	FC18
EREOL	FBBE	EREOS	FBE3	ERLNK	FA4B	EROLX	FBC1
EROXX	FBD2	ERSCRX	FA51	ERTEOL	FBAE	ERTEOS	FBD9
ESCFLG	0010	ESC	001B	FEED	FA1E	FLNCRT	FB79
FLNDN	FABE	FLNUP	FA8E	FLN	0004	FORMAT	0016
GETFA	FFBD	GETFOT	FFAE	GETKA	FF84	GETKBD	FF76
HOCU	FA33	HOME	FC97	HORDIS	0001	HORTOT	0000
HSYPOS	0002	INDEX	0017	INICEN	FDC6	INIKBD	FF63
INLINE	000E	IRQJMP	FE0F	IRQVEC	0039	JMPVEC	0032
LEFT	FA2D	LIPENH	0010	LIPENL	0011	LLNDN	FADE
LLNUP	FAAE	LLN	0008	LPSCR	0015	MCRTA	FDEC
MCRTB	FDF2	MCRTC	FDF3	MODE	0008	MOVCR	FDD0
NMIJMP	FE0C	NMIVEC	0037	RAMBEG	0012	RAMPTR	0000
RECIRQ	FF44	RECREG	C130	RESET	FCA8	RESVTA	FEA2
RETURN	FA14	RFILE	C141	RIGHT	FA45	RSA	FCBD
RSB	FCC8	SCANLN	0009	SCRPTR	000A	SPACE	0020
SPECEN	FF25	TABLE	0020	TEMCOL	0011	TOGDUP	FA76
TOGGLE	FA64	TOGLF	FA6D	TOSCR	FA88	TPX	FC49
TRAREG	C130	TVPUT	FC29	UP	FA39	UPDATH	0012
UPDATL	0013	VAACR	C10B	VAIER	C10E	VAIFR	C10D
VALID	FA7F	VALVEC	FA11	VAPAD	C101	VAPADD	C103
VAPADN	C10F	VAPBD	C100	VAPBDD	C102	VAPCR	C10C
VASR	C10A	VATACH	C105	VATACL	C104	VATALH	C107
VATALL	C106	VATBCH	C109	VATBCL	C108	VBACR	C11B
VBIER	C11E	VBIFR	C11D	VBPAD	C111	VBPADD	C113
VBPADN	C11F	VBPBD	C110	VBPBDD	C112	VBPCR	C11C
VBSR	C11A	VBTACH	C115	VBTACL	C114	VBTALH	C117
VBTALL	C116	VBTBCH	C119	VBTBCL	C118	VERDIS	0006
VERTOT	0004	VHSYWI	0003	VIDEND	F9EB	VIDEO	F9DE
VSYPOS	0007	VTACEN	FEFA	VTAS	FEEB	VTAT	FF11
VTATXY	FF1B	VTATZX	FF40	VTA	FEE0	VTOTAJ	0005
WAIT	FDA8						

SYMBOL TABLE 3400 386E

HORTOT 0000	RAMPTR 0000	HORDIS 0001	AHOLD 0002
HSYPOS 0002	VHSYWI 0003	FLN 0004	VERTOT 0004
VTOTAJ 0005	CLN 0006	VERDIS 0006	VSYPOS 0007
LLN 0008	MODE 0008	SCANLN 0009	CURSTA 000A
SCRPTR 000A	CUREND 000B	CURSOR 000C	DSPSTH 000C
DSPSTL 000D	CURPOH 000E	INLINE 000E	COL 000F
CURPOL 000F	ESCFLG 0010	LIPENH 0010	LIPENL 0011
TEMCOL 0011	RAMBEG 0012	UPDATH 0012	UPDATL 0013
CHAPLN 0014	LPSCR 0015	FORMAT 0016	INDEX 0017
BUFFER 0018	ESC 001B	DUMMY 001F	SPACE 0020
TABLE 0020	JMPVEC 0032	CENFLG 0034	DUPLEX 0035
AUTOLF 0036	NMIVEC 0037	IRQVEC 0039	VAPBD C100
VAPAD C101	VAPBDD C102	VAPADD C103	VATACL C104
VATACH C105	VATALL C106	VATALH C107	VATBCL C108
VATBCH C109	VASR C10A	VAACR C10B	VAPCR C10C
VAIFR C10D	VAIER C10E	VAPADN C10F	VBPBD C110
VBPAD C111	VBPBDD C112	VBPADD C113	VBTACL C114
VBTACH C115	VBTALL C116	VBTALH C117	VBTBCL C118
VBTBCH C119	VBSR C11A	VBACR C11B	VBPCR C11C
VBIFR C11D	VBIER C11E	VBPADN C11F	RECREG C130
TRAREG C130	ACIASR C131	ACICMD C132	ACICTL C133
AR C140	RFILE C141	VIDEO F9DE	VIDEND F9EB
BRKTST F9F6	CHECK F9FE	VALVEC FA11	RETURN FA14
FEED FA1E	CLRHOH FA24	LEFT FA2D	HOCU FA33
UP FA39	DOWN FA3F	RIGHT FA45	ERLNX FA4B
ERSCRX FA51	DELLIN FA57	TOGGLE FA64	TOGLF FA6D
TOGDUP FA76	VALID FA7F	TOSCR FA88	FLNUP FA8E
CLNUP FA9E	LLNUP FAAE	FLNDN FABE	CLNDN FACE
LLNDN FADE	ADJUST FAEE	ADJU FB06	ADSA FB11
ADJV FB32	ADJCUR FB35	ACURA FB3F	ACURB FB4A
ACURX FB4D	ACURC FB65	FLNCRT FB79	CRAMPT FB8D
CRAMP FBAD	ERTEOL FBAE	EREOL FBBE	EROLX FBC1
EROXX FBD2	ERTEOS FBD9	EREOS FBE3	EEOL FBFC
EEOLB FC03	ERAFLN FC07	ERALLN FC18	TVPUT FC29
TPX FC49	CURDN FC52	CURDNX FC6E	CURUP FC71
CURUPX FC8A	CURLFT FC8D	CURRGT FC92	HOME FC97
RESET FCA8	RSA FCBD	RSB FCC8	CRTINT FCD7
CIA FCD9	COMCOM FCE7	COMCOA FCEB	COMCOB FCFC
COMCOC FD04	COMTAB FD09	COTABA FD1D	COTABB FD2B
COTABX FD3B	COMADR FD4F	CENTRO FD95	CTROA FDA3
WAIT FDA8	CTROB FDC5	INICEN FDC6	MOVCRT FDDD
MCRTA FDEC	MCRTB FDF2	MCRTC FDF3	NMIJMP FE0C
IRQJMP FE0F	CRTINA FE12	RESVTA FEA2	VTA FEE0
VTAS FEEB	VTACEN FEFA	VTAT FF11	VTATXY FF1B
SPECEN FF25	VTATZX FF40	RECIRQ FF44	INIKBD FF63
GETKBD FF76	GETKA FF84	CTLCMD FF85	GETFOT FFAE
GETFA FFBD	ACIIN FFC4	ACIOUT FFD1	CHE FFDE
CHECKA FFEF			

