

Dit is het logische vervolg op het eerder gepubliceerde verhaal over de Phunsky. Voor degenen die niet in het bezit zijn van het clubblad met voornoemde beschrijvingen en schema's, kunnen deze alsnog krijgen bij ondergetekenden. Toen was er al sprake van een klein en een groter systeem. Het hart van beide systemen vormt de C.P.U. kaart. Deze wordt in het kleine systeem al gebruikt maar nog niet ten volle benut. Dingen die toen niet persé nodig waren, zijn toen ook niet behandeld en komen nu aan de beurt.

De 2 k RAM op de C.P.U. kaart was toen niet direct noodzakelijk, maar kon zonder meer gebruikt worden. Maar nu moet de 2 k RAM aanwezig zijn voor b.v. de scratchpad van de monitor. Deze 2 k RAM kan uitgeschakeld worden door buslijn A21 laag te maken. Als op dezelfde geheugenlocatie een ander geheugen geplaatst zou worden (b.v. een PROM) dan zal de eerste uitgeschakeld moeten kunnen worden. Verwijderen van de print is hier niet mogelijk, vandaar deze aanpak. De RAM is gesitueerd van adres 0800 t/m 0FFF.

De monitor zit verpakt in een 2 k EPROM van het type 2716 (2516). De beschrijving van de monitor volgt nog later. De monitor is evenals de RAM uitschakelbaar maar dan d.m.v. buslijn A20. We zagen dit al in het kleine systeem. De stand van buslijn A20 bepaalde met welke monitor gewerkt moest worden. Als A20 laag is (hoofdmonitor uitgeschakeld) en A22 ook laag is, dan zijn beide monitoren uitgeschakeld.

Op de C.P.U. kaart wordt de C-poort "gelatched" aangeboden aan de busprint. Bit 0 t/m 3 van de C-poort wordt gebruikt om het geheugen tussen adres 4000 en 7FFF (pagina 3 en 4) uit te breiden tot 16 maal deze capaciteit. Dit zelfde gebeurt met bit 4 t/m 7 maar dan voor de geheugenlocaties 1800 t/m 1FFF en deze laatste ook voor de Extended poort uitbreiding tussen 80 en FF. Op de locaties 1800 t/m 1FFF komt in dit systeem een PROM kaart te zitten waar gekozen kan worden tussen 2 k RAM of 3 verschillende (E)PROM's met daarin b.v. het M.D.C.R. programma en een soort disassembler.

De interface kaart heeft nauwe banden met de C.P.U. kaart. Er is geen communicatie mogelijk tussen de gebruiker en de uP zonder de interface. Hierop worden aangesloten: een ASCII-keyboard, een in- en uitgang voor seriële dataverwerking volgens de RS-232 norm en eventueel een M(ini) D(igital) C(assette) R(ecorder). Alle signalen lopen via de D-poort. De D-poort is speciaal hiervoor gereserveerd en wordt in het hele systeem verder niet gebruikt. De D-poort heeft hier 2 input poorten en 1 output poorten. De keuze tussen de input poorten wordt gemaakt m.b.v. de output poort bit 1. Is deze bit laag, dan kan de bovenste inputpoort (74LS244) gebruikt worden, maar is dit bit hoog dan komt de keuze op de onderste input poort.

Eerst bespreken we de output poort.

Aan bit 0 verschijnt het T.T.Y. signaal. Dit wordt omgezet van een 0 - 5 volt logica naar + en - 12 volt, d.m.v. de op-amp 741. Bit 1 is al behandeld. Bit 2 reset de keyboard flip-flop's, maar hierover komen we verderop nog terug. Bit 3 stuurt eventueel via een "LoFi" trapje een luidsprekertje. Dit luidsprekertje kan b.v. ondergebracht worden in het keyboard. De monitor geeft een pulstrein af op bit 3 als de video routine van de monitor een ASCII BELL karakter krijgt. Bit 4 t/m 7 sturen de M.D.C.R. Van deze bits is bit 4 voor de REV (reverse), dit is bedoeld om de band achteruit te laten draaien. Bit 5 FWD (forward) laat de band vooruit draaien. Als de aansluitingen REV en FWD beide hoog zijn staat de band stil. Bit 6 WCD (write command) is het "opname" commando, hierdoor wordt de wiskop in werking gesteld en wordt het WDA (write data) signaal, bit 7, doorgelaten. Het WDA signaal bevat de seriële data die opgenomen moet worden.

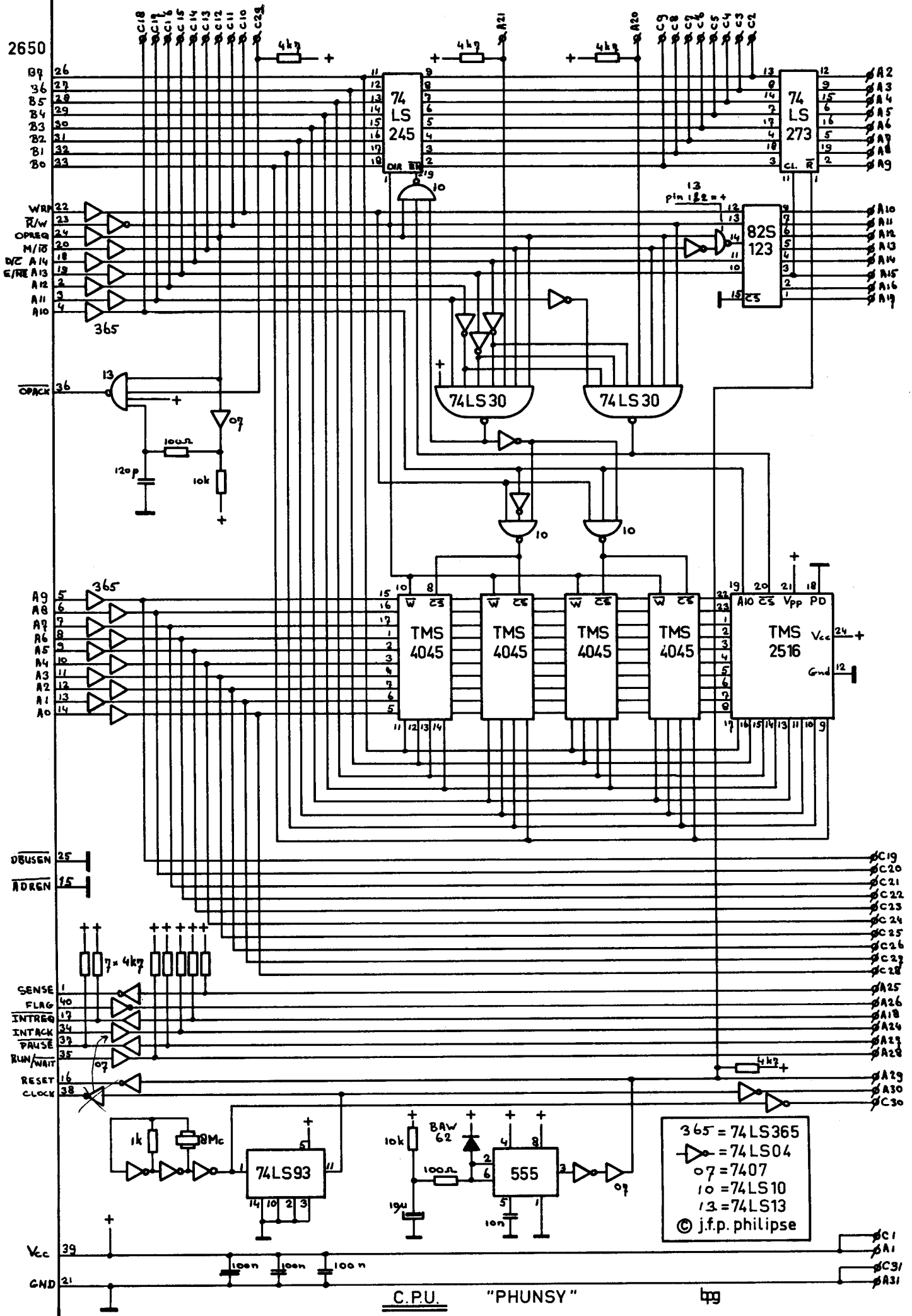
We vervolgen nu met de bovenste input poort.

Hiervan zijn bit 3 t/m 7 ook weer voor de M.D.C.R. gebruikt. Bit 7 RDC (read clock) dient voor de synchronisatie van het uit de recorder komende signaal. Bit 6 RDA (read data) bevat het seriële signaal van de tape naar de computer. Bit 5 BET (begin/end of tape) wordt laag als de tape aan het begin of het einde van de band is. Bit 4 CIP (cassette in position) is laag als er een cassette aanwezig is in de M.D.C.R. Bit 3 WEN (write enable) is laag als de opnamebeveiliging van de cassette geplaatst is en de cassette in de recorder geplaatst is. Bit 1 en 2 zijn verbonden met een schakelaar en kunnen zowel hoog als laag gemaakt worden. Deze schakelaarstanden worden "afgescanned" door middel van het monitorprogramma. Met Sk 1 (bit 1) dicht (laag) verloopt het hele dataverwerkingsproces serieel via de T.T.Y. aansluiting. Dit voor wat betreft de input van een keyboard op een teletype zo wel als de inhoud van een ponsbandlezer en de uitsturing naar een schrijver of een ponsbandschrijver. Het ASCII-keyboard staat nu buiten werking. Met schakelaar Sk 1 open wordt de eventuele data van het keyboard verwerkt en wordt de T.T.Y. aansluiting uitsluitend gebruikt om een ponsbandlezer te lezen dan wel te schrijven. Sk 2 schakelt de baudrate van de T.T.Y. in- en uitgang om tussen 110 en 300 baud. De baudrate van de M.D.C.R. is vast ingesteld op 6000. Bit \emptyset is de ingang voor de RS-232 signalen die aangeboden worden op de T.T.Y. aansluiting. Deze signalen worden eerst geclipped en daarna geïnverteerd omgezet in een 0 - 5 volt niveau. Op de T.T.Y. aansluiting kan ook een cassetterecorder aangesloten worden, dit dient wel te geschieden via een omzetkastje wat op zich al een apart apparaatje vormt. In dit kastje wordt het seriële computersignaal omgezet in twee frequenties en op een voor de cassetterecorder aanvaardbaar niveau gebracht. Tevens zit er in het kastje een schakeling om de twee frequenties om te zetten in enen en nullen. Dit kastje heeft z'n eigen netvoeding en kan ook gebruikt worden om informatie van cassette rechtstreeks op een T.T.Y. te zetten, hetgeen zeer handig blijkt te zijn. Dit kastje wordt later nog gepubliceerd. De onderste input poort is uitsluitend bestemd voor het ASCII-keyboard. Op het moment dat er een toets wordt ingedrukt op het keyboard wordt er ook een strobe puls opgewekt die positief gaand is. Deze puls zorgt ervoor dat de uitgangen van de flip-flop's, gevormd met NAND poorten, hoog worden. Dit wordt aangeboden aan bit 7 van de input poort. Als de computer hier een hoog signaal waarneemt gaat hij eerst de flip-flop's resetten d.m.v. output poort bit 2. Daarna gaat hij kijken welke ASCII-code is aangeboden aan de keyboard plug. Dit komt via de inputpoort bit \emptyset t/m 6 op de databus en wordt verder verwerkt. Om te voorkomen dat er een willekeurig teken naar binnen wordt gehaald bij het aanzetten van de voeding en het toevallig hoog zijn van de uitgang van de flip-flop's worden de flip-flop's gereset d.m.v. de power-on reset. Daar niet alle keyboards zijn of worden uitgerust met een reset knop is op de interface kaart hiertoe een schakelingetje geplaatst. In het kleine systeem zat de resetknop op het hex-keyboardkastje. Bij het grote systeem is dit kastje niet noodzakelijk waardoor de mogelijkheid om te resetten zou vervallen. Tot zover de C.P.U. kaart en de interface kaart. Theoretisch kan met deze twee kaarten een busprint en een voeding een systeem worden opgezet dat zou werken d.m.v. een T.T.Y. Maar beperkt is het wel, daarom zullen er nog in komende afleveringen de volgende uitbreidingen komen:

- video-interface
- 8 k RAM kaart (statisch)
- 16 in- en 16 out interface
- RAM/ 3 x EPROM kaart (2 k elk)
- 8/16 k RAM kaarten (dynamisch)

De volgende aflevering zal de monitor behandelen.

2650



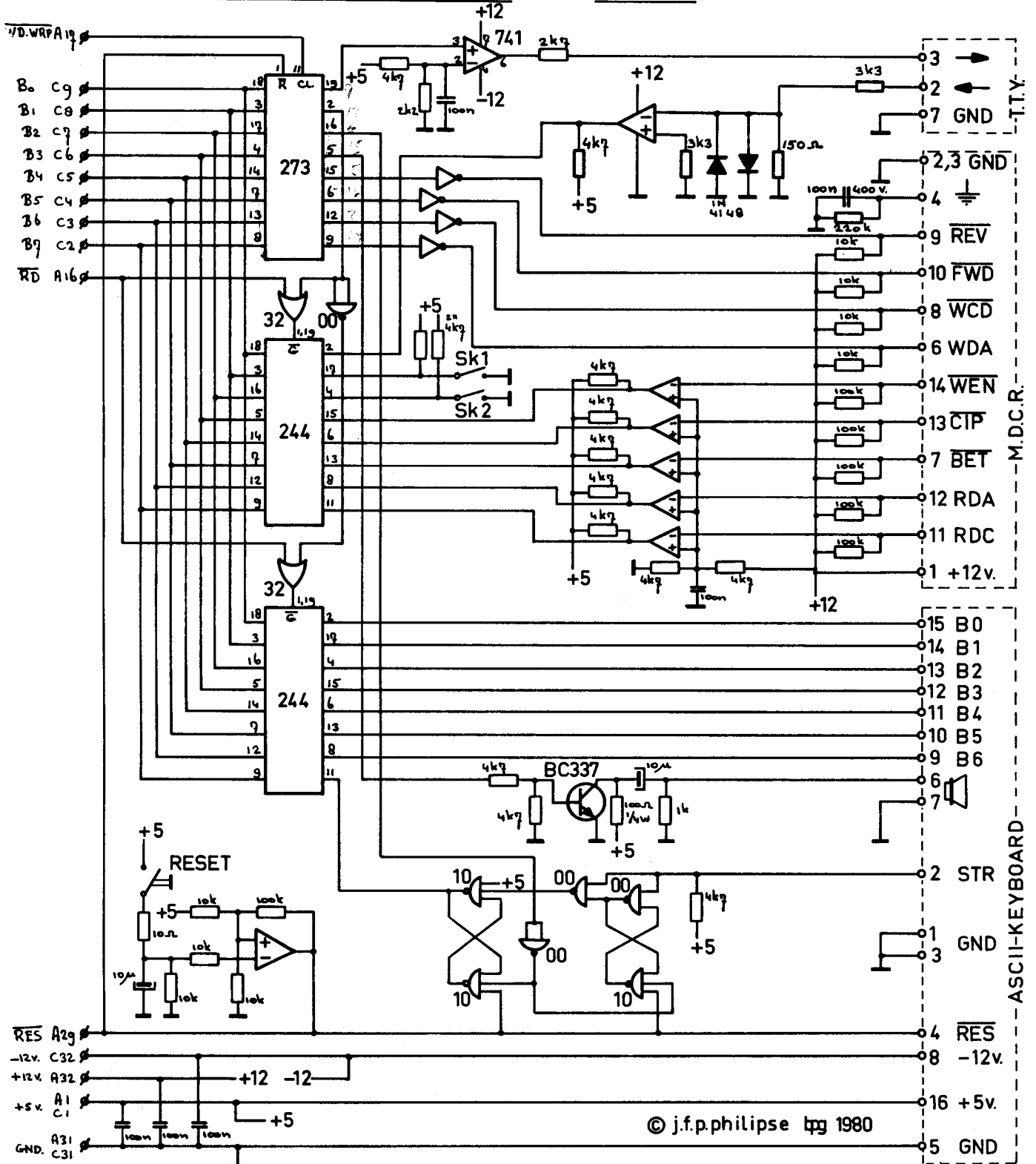
365 = 74LS365
 07 = 74LS04
 07 = 7407
 10 = 74LS10
 13 = 74LS13
 © j.f.p. philipse

C.P.U. "PHUNSY"

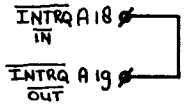
lpg

INTERFACE t.b.v. ASCIIKEYBOARD-T.T.Y.-M.D.C.R.

"PHUNSY"



© j.f.philipse tpg 1980



	dicht	open	
Sk1	ser.	par.	
Sk2	110	300	baud

00	= 74LS00	244	= 74LS244
7406	= 7406	273	= 74LS273
10	= 74LS10		
32	= 74LS32		

Phunsy bus:

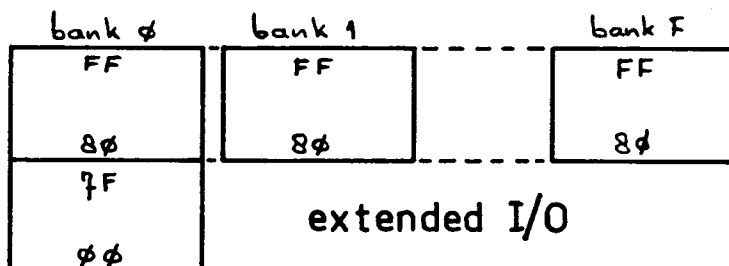
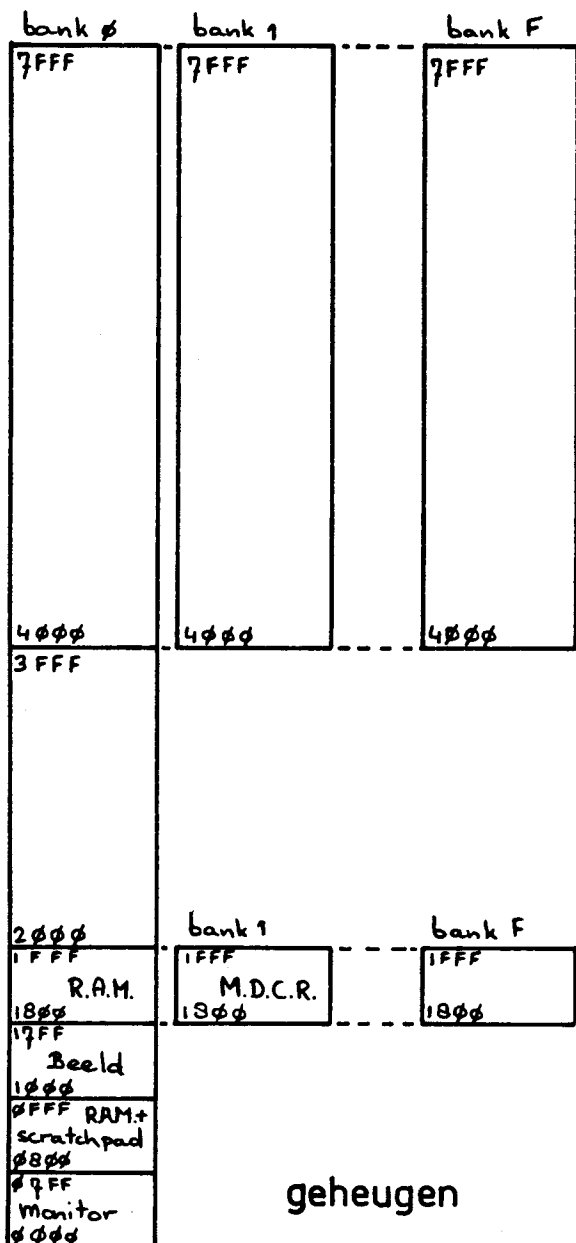
A 1	+5 Volt	C 1	+5 Volt
A 2	Co7	C 2	B7
A 3	Co6	C 3	B6
A 4	Co5	C 4	B5
A 5	Co4	C 5	B4
A 6	Co3	C 6	B3
A 7	Co2	C 7	B2
A 8	Co1	C 8	B1
A 9	Co0	C 9	B0
A10	not used	C10	WRP
A11	not used	C11	R/W°
A12	not used	C12	OPREQ
A13	not used	C13	M/IO°
A14	RC°	C14	A14-D/C°
A15	WC.WRP°	C15	A13-E/NE°
A16	RD°	C16	A12
A17	WD.WRP°	C17	A11
A18	INTRQ IN°	C18	A10
A19	INTRQ OUT°	C19	A 9
A20	CPU-ROM DIS° (000-7FF)	C20	A 8
A21	CPU-RAM DIS° (800-FFF)	C21	A 7
A22	DMAREQ°	C22	A 6
A23	DMAACK°	C23	A 5
A24	INTACK°	C24	A 4
A25	SENSE°	C25	A 3
A26	FLAG°	C26	A 2
A27	PAUSE°	C27	A 1
A28	RUN/WAIT°	C28	A 0
A29	RESET°	C29	OPACK
A30	1 Mc	C30	8 Mc
A31	GND	C31	GND
A32	+12 Volt	C32	-12 Volt

°=inverted.

Phunsy Monitor

In de hier volgende aflevering wordt, zoals aangekondigd, de monitor beschreven, strikt software dus. Het is ons gebleken dat het een onmogelijke opgave was om de monitor in z'n geheel te ontleden en bij wijze van spreken per instructie te behandelen. Dit zou een Peyton Place-achtig geheel worden, waarbij de redactie van het E.C.A.-bulletin het eerstkomende jaar niet hoefde te klagen over kopij. Daarom hebben wij besloten om de listing niet te publiceren, voor de beschrijving is dit niet relevant. De beschrijving is zoveel mogelijk gegoten in de vorm van een gebruiksaanwijzing. Dit geeft de geachte lezer enig idee in het werken met deze monitor. Er zijn naast de hier genoemde mogelijkheden nog een aantal zaken die bij het praktische gebruik erg comfortabel kunnen zijn, maar niet aangeroerd worden omwille van de overzichtelijkheid. Degenen, die het naadje van de monitorkous wensen te weten, kunnen aankloppen bij de ondergetekenden voor een kopie van de listing met eventuele uitleg van specifieke gevallen. Van deze monitor is ook een versie die aangepast is op de 2650 clubprint. Deze aanpassing bestaat hoofdzakelijk uit een anders opgezette beeldroutine. Hiervoor is echter wel een hardware interface gedeelte nodig, zoals beschreven in het vorige Phunsy deel. Eerst een overzicht van de geheugen opbouw met bankmogelijkheden.

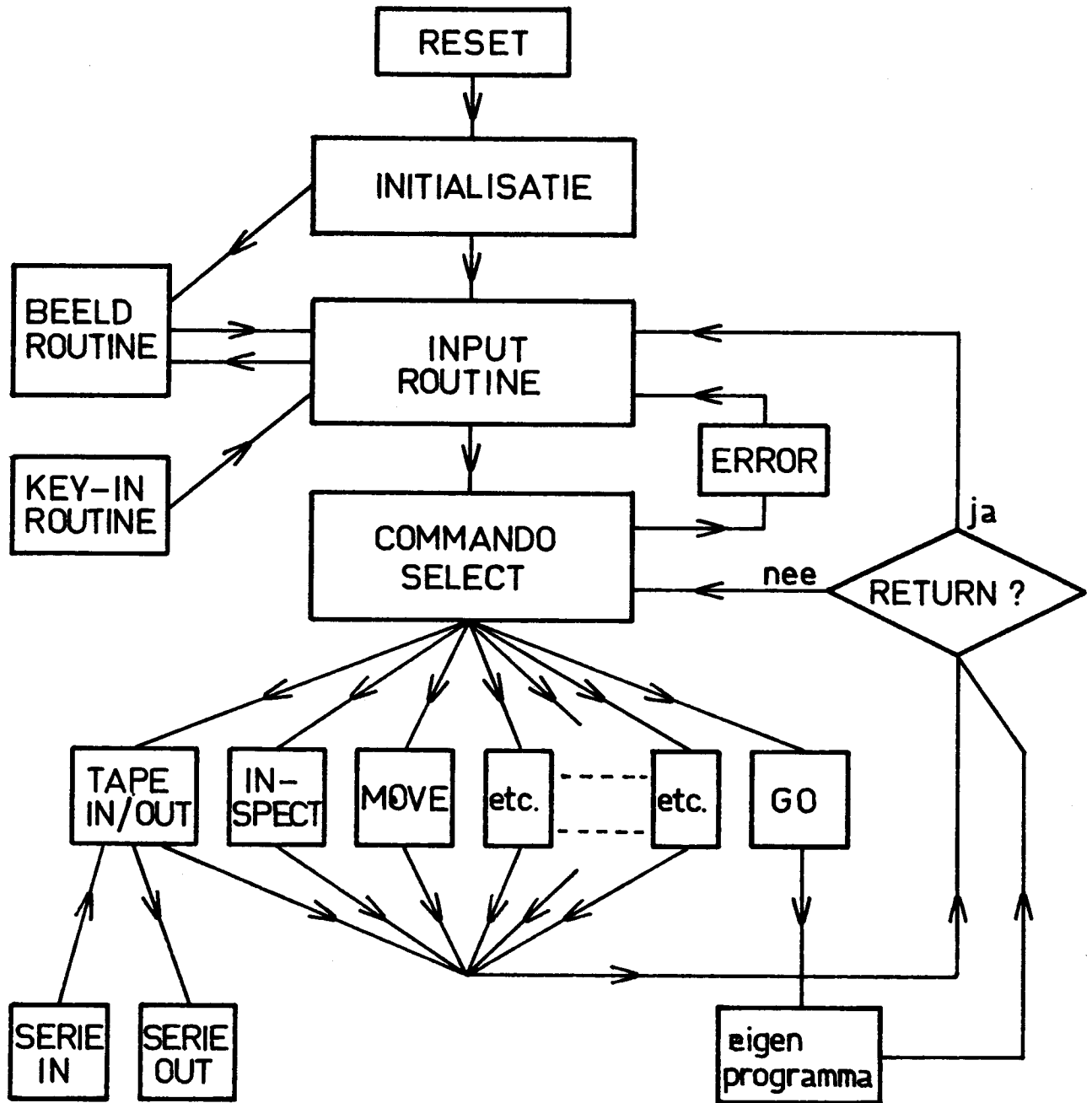
Die bankmogelijkheden zijn bedoeld om het geheugenbestand drastisch te kunnen uitbreiden (max. 302 k bytes). We onderscheiden twee bankselecties voor het geheugen en één bankselectie t.b.v. de extended I/O poort uitbreiding. Bankselectie wordt toegepast op de adressen 1800 t/m 1FFF. Dit zijn blokken van 2 k elk en uitstekend geschikt voor dis-assembler, line-assembler, printeroutine, M.D.C.R. programma, modem programma, PROM programmer, etc. We geven deze blokken aan met de letter U. Het originele blok wordt bank 0 genoemd en de uitbreidingen 1 t/m F. Deze worden geselecteerd door de C-output poort bit 4 t/m 7. Op de adressen 4000 t/m 7FFF wordt ook bankselectie toegepast, maar nu met blokken van 16 k elk. Deze blokken geven we aan met de letter Q. Ze worden geselecteerd d.m.v. C-output poort bit 0 t/m 3. De extended poorten kunnen ook worden uitgebreid en worden dan omgeschakeld d.m.v. C-output poort bit 4 t/m 7. Dit geldt alleen voor de poorten 80 t/m FF. Dit houdt verband met het "banken" van de adressen 1800 t/m 1FFF. Elke bank heeft nu een stuk geheugen en een aantal eigen input- en output poorten.



geheugen

extended I/O

De opbouw is hieronder blokschematisch weergegeven. Een aantal verbindingen zijn weggelaten om het geheel overzichtelijk te houden.



Beschrijving van de opbouw.

De reset is zuiver hard-ware opgezet door de power-on reset en hand-reset en spreekt voor zich.

Het blok initialisatie is een soort software preset. Deze wist het scherm, zet de cursor linksboven (home), maakt de output poorten C en D nul, kijkt naar de standen van de schakelaars op de interfaceprint en voorziet de scratch-pad van de benodigde RAM informatie.

De input routine werkt nauw samen met de inputbuffer van een 1/4 k bytes. In het begin van deze routine wordt eerst de inputbufferpointer op nul gezet. Alle ingetypte commando's komen in de inputbuffer te staan, terwijl ze ook op het scherm te bewonderen zijn. Bij het intypen van adressen en data kan men de voorafgaande nullen weglaten. Uittreding uit de input routine geschiedt uitsluitend na het return commando (hier verder aangeduid met CR).

Het commando-select blok begint ook weer met de inputbufferpointer op nul te zetten. Het tast de inputbuffer af totdat er genoeg gegevens voor handen zijn om een bepaald commando daadwerkelijk uit te voeren. Een commando bestaat uit een ASCII karakter, eventueel voorafgegaan door een of meerdere adressen. Voorbeeld: de adressen 800 t/m BFF moeten op tape worden gezet. Ingetypt wordt:

800 - BFF W CR (spaties worden niet ingetypt!)

Als een commando onjuist is, wordt er een ERROR-melding gegeven en gaat het programma terug naar de input routine.

Hierna volgen de uiteindelijke routines. Dit zijn alle subroutines waardoor ze ook separaat gebruikt kunnen worden.

Hieronder volgt per routine een korte omschrijving en het beginadres.

1) Inspect Memory (I) "0449"

Met deze routine is het mogelijk de inhoud van geheugenplaatsen op het scherm te zetten in rijen van 16 bytes.

Voorbeeld: 0 - FF I CR (geheugen blok)
" : EE9 I CR (enkele geheugenplaats)
" : EE9 III CR (3 opeenvolgende geheugenplaatsen)

2) Change Memory (:) "048A"

Hiermee kan men een geheugenplaats wijzigen of een n-aantal geheugenplaatsen inlezen. Tussen elke byte moet een spatie worden opgenomen.

Voorbeeld: 800 : 4 20 CC E 80 17 CR
" : 801 : 41 CR (1 adres verandert)

3) Move Memory (M) "04D4"

Hiermee kan men een blok geheugen verplaatsen. Het start- en eindadres van het blok wordt ingetypt alsmede het eerste adres van de bestemming. Hiertussen moet een > teken worden opgenomen.

Voorbeeld: 800 - 8FF > C00 M CR

4) Verify Memory (V) "04F3"

Hiermee kan een bepaald blok vergeleken worden met een ander blok. De intype wijze is hetzelfde als met "Move Memory". Ongelijkheden worden aangegeven op het scherm.

Voorbeeld: 800 - 8FF > C00 V CR

Bij ongelijkheden verschijnt op het scherm b.v. :

81A : 04 C1A : 08
81B : 00 C1B : 0B

5) Write Tape (W) "0540"

Met deze routine kan een stuk geheugen op band worden weggeschreven. Dit gebeurt via de D-poort bit 0 en de RS-232 schakeling. De snelheid waarmee dit gebeurt wordt bij de initialisatie bepaald, afhankelijk van de stand van de baudrateschakelaar Sk 2. Wil men hiervan afwijken dan moet 1 byte in de scratch-pad gewijzigd worden, nl. "0EE9".

FE = 110 baud, 5A = 300 baud. Na wijziging mag niet meer gereset worden. Het wegschrijven wordt voorafgegaan door de benodigde informatie om bij het teruglezen weer op het zelfde adres uit te komen. Elke 1/4 k bytes data wordt gevolgd door een checksum.

Voorbeeld: 800 - CFF W CR

6) Read Tape (R) "05E8"

Nu kan data van tape naar geheugen worden gezet. Dit loopt via de D-inputpoort bit 0 en de RS-232 schakeling. V.w.b. de baudrate zie bij Write Tape.

Voorbeeld: R CR

- 7) Set Breakpoint (S) "0631"
 Deze routine kan een programma onderbreken in RAM met de bedoeling om een momenteel inzicht te krijgen in de registers, P.S.U. en P.S.L. in de processor. Op de gewenste plaats vervangt de monitor een instructie door een branch instructie naar de monitor. Wordt het breekpunt bereikt, dan wordt de oude instructie teruggezet en worden alle registers op het beeld gezet. Hierna heeft men de keuze om door te gaan met het programma of terug naar de monitor te gaan. Wil men doorgaan, dan moet C (continue) worden ingedrukt. Wenst men niet door te gaan dan is elke toets daarvoor geschikt (behalve de C dus).
 Voorbeeld: 820 S CR
 800 G CR (stel dat hier het programma start)
 als het breekpunt is bereikt verschijnt op het scherm:
 : 01 R1 : 40 PSL : 40
- 8) Go to Memory (G) "0717"
 Deze routine start een programma op het ingetypte adres. Hiermee wordt uit de monitor gesprongen alsof het programma een subroutine is. Dit geeft de mogelijkheid om na een programma een return te zetten (RETC/RETE). Maar in het programma mag dan niet de Program Stack Pointer op nul worden gezet.
 Voorbeeld: 800 G CR
- 9) Calculate (=) "0726"
 Hiermee is het mogelijk om twee bytes op te tellen of af te trekken.
 Voorbeeld: 5 + 3 = CR
 op scherm: 08
 Voorbeeld: D3 - 5 = CR
 op scherm: CE
 Voorbeeld: 35 - 1A = CR
 op scherm: 1B
- 10) U is een commando met twee functies.
 1^e Bankselect
 Wordt eerst een HEX getal ingetypt en daarna de U, dan verschijnt dat getal op de C-poort bit 4-7. Hiermee wordt een bank geselecteerd tussen 1800 en 1FFF. Zie voor de banken de geheugenopbouw.
 2^e Go to address 1800
 Wordt de U ingetypt zonder dat er een HEX getal aan voorafgaat, dan springt de monitor naar adres 1800. Dit is dus hetzelfde als 1800 G CR. Ook is het mogelijk om een bank te selecteren en er tegelijkertijd naar toe te springen door b.v. 5UU CR in te typen. 5U selecteert en de tweede U verzorgt de sprong ernaar toe.
 De extended input-output poorten 80-FF worden mee geselecteerd met U.
- 11) Voor Q geldt hetzelfde als voor U, behalve:
 1^e De bank wordt geselecteerd door de C-poort bit 0-3.
 2^e De bankselectie gaat nu van 4000 t/m 7FFF.
- 12) Tape (T) "0770"
 Dit veroorzaakt een bank 1 select voor 1800-1FFF en een sprong er naar toe. In bank 1 zit namelijk het M.D.C.R. programma (dit wordt nu niet behandeld).
 Het commando T is identiek aan 1UU CR.

Beeldroutine

De beeldroutine is gebaseerd op een beeld van 64 karakters per regel en 32 regels. Totale beeldgeheugen is dus 2 k bytes. Dit geheugen bevindt zich op de adressen 1000 t/m 17FF. De beeld(sub)routine start op adres "00EB". Als een ASCII karakter in R0 staat en er wordt gesprongen naar de beeldroutine, verschijnt dit karakter op het beeld

of wordt de cursorplaats veranderd, waarna weer wordt teruggesprongen naar waar hij mee bezig was.

Een aantal commando's zijn uitsluitend bestemd voor de beeldroutine. Het zijn de volgende:

NUL (ctr. \emptyset) " $\emptyset\emptyset$ " : scherm wordt schoongemaakt en de cursor wordt linksboven geplaatst
ENQ (ctr. E) " $\emptyset5$ " : de lijn waar de cursor op staat wordt schoongemaakt vanaf de cursor
CAN (ctr. X) " 18 " : de lijn waar de cursor op staat wordt schoongemaakt
ACK (ctr. F) " $\emptyset6$ " : het scherm wordt schoongemaakt vanaf de plaats waar de cursor zich bevindt
FS " $1C$ " : cursor naar rechts
GS " $1D$ " : cursor naar beneden
RS " $1E$ " : cursor naar links
US " $1F$ " : cursor naar boven

BS (ctr. H) " $\emptyset8$ " : cursor naar links, maar nu wordt ook de input-bufferpointer verlaagd (in tegenstelling tot RS)
NAK (ctr. U) " 15 " : cursor naar rechts, maar nu wordt ook de input-bufferpointer verhoogd en de karakterplaatsen, waar de cursor overheen gaat, worden overgenomen in de input buffer (in tegenstelling tot FS)

Software matig kunnen de kantlijnen van het beeld bepaald worden. Dit kan zeer handig zijn als het niet gewenst is dat het hele beeld "gescrolled" wordt, maar b.v. alleen het onderste deel. Na een reset staan de kantlijn-bytes zodanig gepreset dat het hele beeld gebruikt wordt. De volgende scratch-pad geheugenplaatsen zijn hiervoor bestemd:

" $\emptysetEF\emptyset$ " : na reset $\emptyset\emptyset$ bovenste kantlijn

" $\emptysetEF1$ " : na reset $\emptyset\emptyset$ linker kantlijn

" $\emptysetEF2$ " : na reset $1F$ onderste kantlijn

" $\emptysetEF3$ " : na reset $3F$ rechter kantlijn

Bij het veranderen moet er op gelet worden dat de linkerkantlijn in HEX niet groter wordt dan de rechterkantlijn of andersom. Ditzelfde geldt ook voor de boven- en onderkantlijnen.

Serieel gebruik

Voor wat betreft de seriële in- en output nog het volgende. Op de interfaceprint zit een schakelaar Sk 1 waarmee gekozen kan worden tussen seriële en parallelle communicatie. Bij de initialisatie worden een paar bytes in de scratch-pad gepreset, deze dragen er zorg voor dat gesprongen wordt naar de gekozen voorkeur op het moment dat het nodig is om in- of output te bedrijven. Door de bytes te veranderen is het mogelijk om serieel in en parallel uit, of andersom, te bedrijven. Ook kan d.m.v. een heel klein programmaatje zowel serieel als parallel uit gebruikt worden. De in- en output routines starten op de volgende adressen:

parallel in : " $\emptyset\emptyset54$ " (ASCII keyboard)
parallel uit : " $\emptyset\emptysetEB$ " (video-interface)
serieel in : " $\emptyset5C8$ " (RS-232 aansluiting)
serieel uit : " $\emptyset588$ " (RS-232 aansluiting)

Frank Philipse

Ben Postema

© 1981

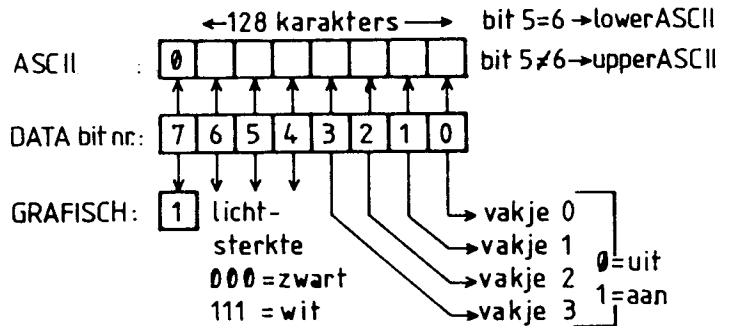
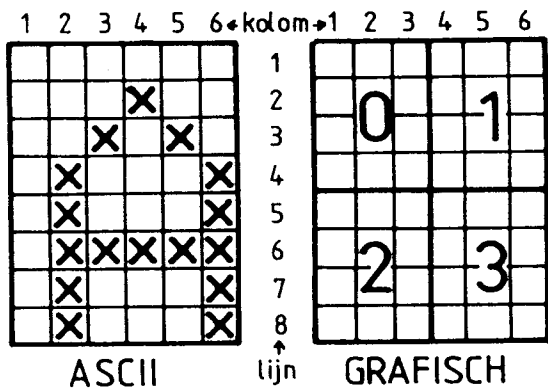
Publicatie van de DRAMprint was een mooie aanleiding om de gewijzigde busprint aansluitingen het licht te laten zien. Dit geeft degenen die niet bekend zijn met de Phunsys een houvast over het hoe en waar van de busaansluitingen. De wijzigingen t.o.v. de eerste publicatie (E.C.A.bulletin, nr. 5, 1980) zijn niet dramatisch. De controllijnen E, RE, WE en WE.WRP (A10 t/m A13) zijn komen te vervallen. Dit was geen bezwaar, daar ze nog niet gebruikt werden. Het doel was tweeledig. Enerzijds spaarde dit op de C.P.U. print een PROM (82S123) uit en anderzijds is de decodering van de vervallen signalen zeer eenvoudig te verwezenlijken. De vrijgekomen lijnen kunnen nu voor eigen doeleinden gebruikt worden. Tevens is de lijn RAMDIS (1800 - 1FFF) (A22) komen te vervallen. Hiervoor in de plaats is samen met A23 twee Direct Memory Access controllijnen geschapen, één om de processor te vragen om DMA en één antwoordlijn van de processor ten teken dat er DMA bedreven kan worden. Het DMA circuit hoort thuis op de C.P.U. print en brengt de data-, adres- en controllijnen afkomstig uit de processor in de tri-state positie evenals de lijnen Co0 t/m Co7. Het schema van de aangepaste C.P.U. print is nog niet gepubliceerd.

Voor de rest zijn de aansluitingen hetzelfde gebleven.

Phunsys bus:

A 1	+ 5 volt	C 1	+ 5 volt
A 2	Co7	C 2	B7
A 3	Co6	C 3	B6
A 4	Co5	C 4	B5
A 5	Co4	C 5	B4
A 6	Co3	C 6	B3
A 7	Co2	C 7	B2
A 8	Co1	C 8	B1
A 9	Co0	C 9	B0
A10	not used	C10	WRP
A11	not used	C11	R/ <u>W</u>
A12	not used	C12	OPREQ
A13	not used	C13	M/ <u>I/O</u>
A14	<u>RC</u>	C14	A14 - D/ <u>C</u>
A15	<u>WC.WRP</u>	C15	A13 - E/ <u>NE</u>
A16	<u>RD</u>	C16	A12
A17	<u>WD.WRP</u>	C17	A11
A18	<u>INTRQ IN</u>	C18	A10
A19	<u>INTRQ OUT</u>	C19	A 9
A20	<u>CPU-ROM DIS</u> (0000-07FF)	C20	A 8
A21	<u>CPU-RAM DIS</u> (0800-0FFF)	C21	A 7
A22	<u>DMAREQ</u>	C22	A 6
A23	<u>DMAACK</u>	C23	A 5
A24	<u>INTACK</u>	C24	A 4
A25	<u>SENSE</u>	C25	A 3
A26	<u>FLAG</u>	C26	A 2
A27	<u>PAUSE</u>	C27	A 1
A28	<u>RUN/WAIT</u>	C28	A 0
A29	<u>RESET</u>	C29	OPACK
A30	1 Mc	C30	8 Mc
A31	GND	C31	GND
A32	+ 12 volt	C32	- 12 volt

Alvorens we beginnen met de beschrijving van het schema gaan we eerst kijken hoe de karakters gevormd worden. Elk karakter bestaat uit een veld van 6 kolommen bij 8 lijnen, dus 48 punten. Twee PROM's staan borg voor de karaktervorming. Men kan hiermee maximaal 128 karakters vormen: hoofdletters, kleine letters, cijfers, grieks alfabet en diverse tekens. Bij deze karakters wordt geen gebruik gemaakt van de bovenste lijn en de linker kolom in het veld. Deze dienen als spatie tussen de regels c.q. de karakters onderling. Alle veldjes liggen n.l. tegen elkaar in verband met de grafische tekens. Voor grafische tekens wordt een karakterveldje opgesplitst in 4 vakken. Er zijn dus 16 mogelijkheden om een veld te vullen. Ook kan van een heel veld (dus voor 4 vakjes) de grijstint worden ingesteld en wel in 8 stappen. De keuze tussen grafische of ASCII karakters wordt bepaald door 1 bit in de datainformatie, bit 7. Totaal dus 128 ASCII karakters en 128 grafische mogelijkheden.



Het beeld heeft 32 regels van elk 64 karakters, hetgeen precies overeenkomt met 2k geheugen. Dit geheugen loopt van 1000 t/m 17FF en wordt zo op het scherm gezet dat geheugenplaats 1000 linksboven is en 17FF rechts-onder. Gewoon oplopend dus, van links naar rechts en van boven naar beneden.

Het schema:

Het hart van de schakeling wordt gevormd door de tijdbasis. Deze wordt gevoed met een 8 Mc klok, afkomstig uit de C.P.U.-print. De tijdbasis dirigeert alle signalen van en naar de juiste plaats en op het juiste moment. Het geheugenblok van 2k bytes, nodig voor de beeldinhoud, kan als een normaal geheugen worden gebruikt, terwijl het "tegelijktijd" ook op het scherm wordt gezet, zonder dat dit op het scherm storende effecten geeft. Hoe dat in z'n werk gaat zullen we hieronder proberen uit te leggen. We gaan er hier wat uitvoerig op in, gedeeltelijk om de geïnteresseerden de gebruikelijke opbouw van een videosignaal te tonen, gedeeltelijk om het wezenlijke verschil bij deze interface duidelijk te maken.

De tijdbasis:

Deze schakeling bestaat uit 5 counters met wat logica er omheen. Deze tellers tellen zo dat de gwenste signalen aan de diverse uitgangen beschikbaar zijn. K is een van de belangrijkste. Dit signaal heeft een frequentie van 15625 hertz, oftewel de lijnfrequentie. K is 16 µS laag en 48 µS hoog en bepaalt, voor wat betreft één lijn, wanneer er wel en geen video doorgelaten mag worden. De carry-out van de eerste teller is de strobe, welke afhankelijk van K al of niet een lijn van een karakter in het schuifregister van de video-eindschakeling zet. Is K laag dan is de strobe gesperd en blijft het schuifregister nullen uitschuiven. Het BL signaal zorgt voor de rasterblanking, dit signaal reset het schuifregister, waardoor de uitgang ook een nul is. Gedurende het schrijven van een lijn bepalen de signalen E t/m J welke karakters achtereenvolgens op

een regel worden gezet. De signalen O t/m S tellen de regels. Deze signalen E t/m J en O t/m S worden, afhankelijk van signaal D uit de eerste teller, afgewisseld met de adreslijnen aangeboden op de adresingangen van de RAM. Verder zijn de signalen L t/m N ervoor om bij te houden welke lijn van een karakter aan de beurt is. Deze gaan naar de ingangen A₀ t/m A₂ van de karakter PROM's. Het videosyncsignaal wordt gedecodeerd uit diverse telleruitgangen.

Het videosignaal:

Het videosignaal (zie bovenste grafiek) bestaat uit 313 lijnen, waarvan er 256 kunnen worden voorzien van beeldinformatie. Na het geschreven beeld volgen eerst 13 lege lijnen. Daarna zijn er 4 lijnen die de rastersynchronisatie vormen. Deze worden zo gegenereerd dat de tijd tussen de neergaande flanken van de lijn-sync constant blijft. Hierna volgen nog 40 lege lijnen, alvorens het beeld weer informatie mag bevatten. De verhouding van deze lege lijnen is zo gekozen, dat, bij een normaal afgestelde TV, het beeld in het midden tussen de boven en onderkant staat. De lijn-sync is een signaal dat 60 μ S hoog is en 4 μ S laag is. De tijd dat er video op een lijn mag zijn wordt bepaald door signaal K. De informatie wordt echter 750 nS vertraagd, doordat het er in 6 stappen van 125 nS door het schuifregister wordt uitgeschoven. De tijd tussen video en de lijn-sync is 3,25 μ S en na de sync is deze tijd 8,75 μ S. De sync zelf is 4 μ S, samen dus 16 μ S. De tijd dat er videosignaal aanwezig kan zijn is 48 μ S. Deze tijd valt binnen de genormaliseerde tijd van 52 μ S, waardoor het beeld dus niet alleen wat korter is maar ook wat smaller. Ook hier zijn de tijden zo gekozen dat het beeld weer in het midden valt. Aangezien er bij het ontwerp een poort met 5 ingangen over was, is ook de colour-burst tijd uitgedecodeerd. Het is n.l. mogelijk in plaats van grijs-tinten kleuren te maken door achter deze schakeling nog een kleurengenerator te hangen.

De geheugenschakeling: (zie de onderste grafiek)

Het probleem van de geheugenschakeling was, dat het niet geheel vast ligt wanneer de 2650 iets in het geheugen wil veranderen en wanneer niet. De OPREQ is er wel altijd even lang, maar hoe laat vertelt hij niet. Daarom is de I/O van deze schakeling asynchroon. Het signaal D bepaalt botweg of de μ P of de tellers aan de beurt zijn en kiest door middel van 3 vier-bit selectors. Tussen het op het beeld zetten van 2 karakters is 250 nS tijd over om I/O te doen. (De RAM's moeten dus wel vrij snel zijn. Voor het lezen van het geheugen wordt, of het interessant is of niet, steeds de aanwezige informatie na de 250 nS in de outputlatch geklokt. Tegen de tijd dat de OPREQ wegvalt is de informatie altijd aanwezig. Van de μ P naar het geheugen toe is echter lastiger. Er is een schakeling met twee flipflops die er voor zorgt dat na het begin van de OPREQ de eerste HELE tijd voor I/O (250 nS) gebruikt wordt om iets in de RAM's te zetten. Hoe dit gaat is het beste even te bekijken op de grafiek daar anders dit verhaal wat al te lang zou worden. Hetzelfde geldt voor de tel-sequence.

De karaktervorming:

Deze schakeling heeft twee PROM's die de ASCII tekens bevatten. Voor grafische tekens is er een schakeling die bepaalt welk vakje aan of uit is en een schakeling die de gradatie bepaalt. De informatie wordt door het schuifregister uitgeschoven naar een soort D/A omzetter. Deze bestaat uit een paar weerstanden die het zwart-niveau en de gradatie bepalen. Bij ASCII tekens is het gradatieniveau altijd het hoogste. Een transistor haalt tenslotte de impedantie omlaag.

Samenvatting:

De gehele schakeling past op een Eurokaart. Er is echter geen echte print-lay-out, dus was het wel een beetje soldeerwerk. De RAM's moeten echt een toegangstijd hebben van maximaal 200 nS, anders verschijnen er vreemde toestanden op het scherm, vooral bij het doorschuiven van het beeld.

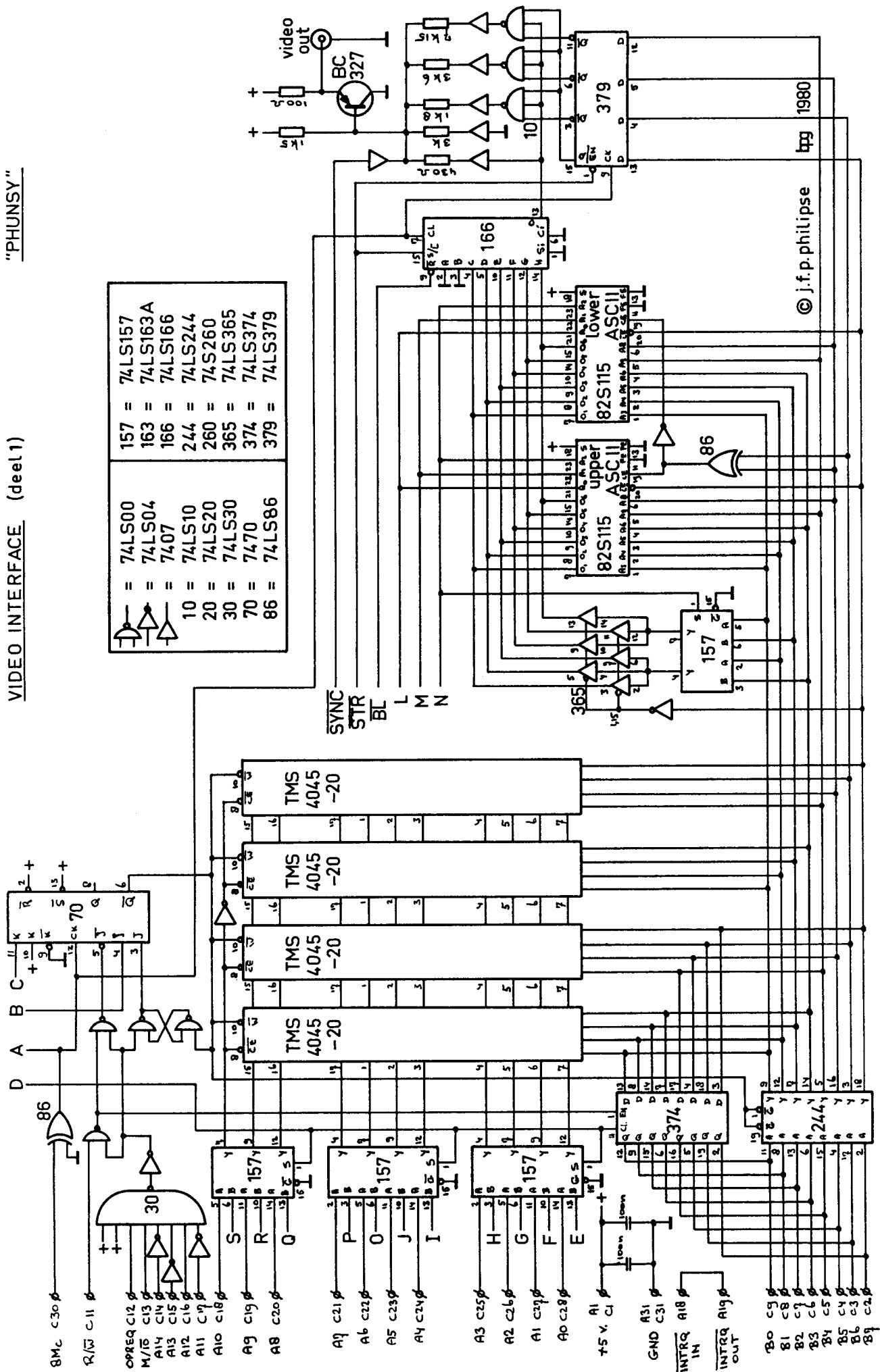
Frank Philipse

Ben Postema

© 1981

VIDEO INTERFACE (deel 1)

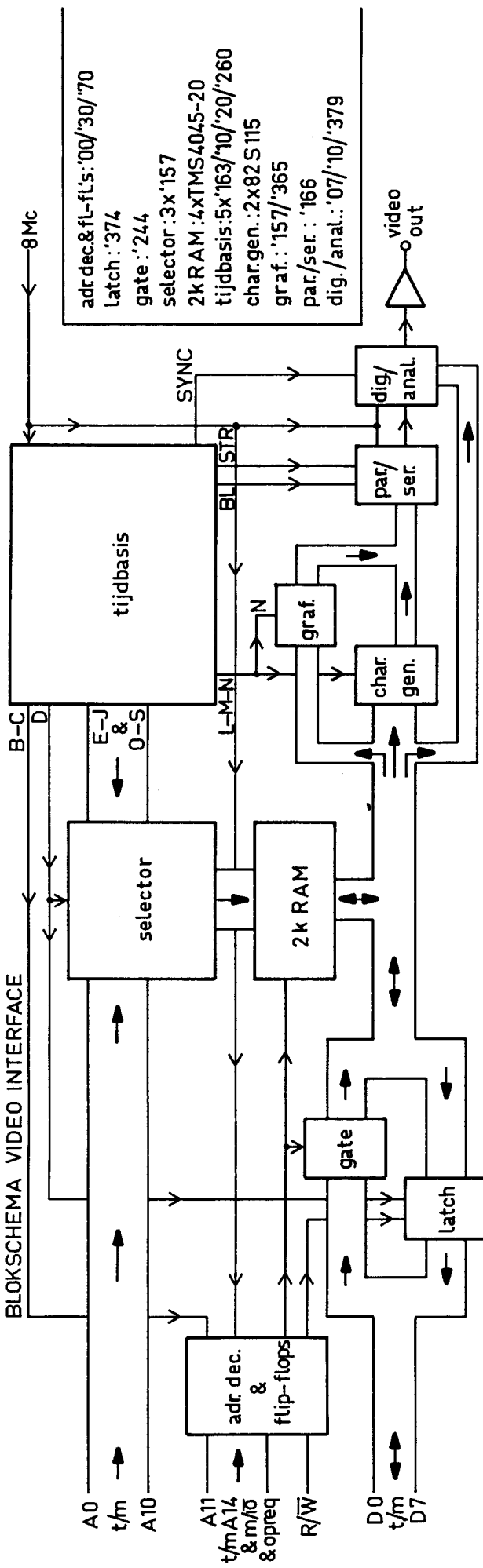
"PHUNSY"



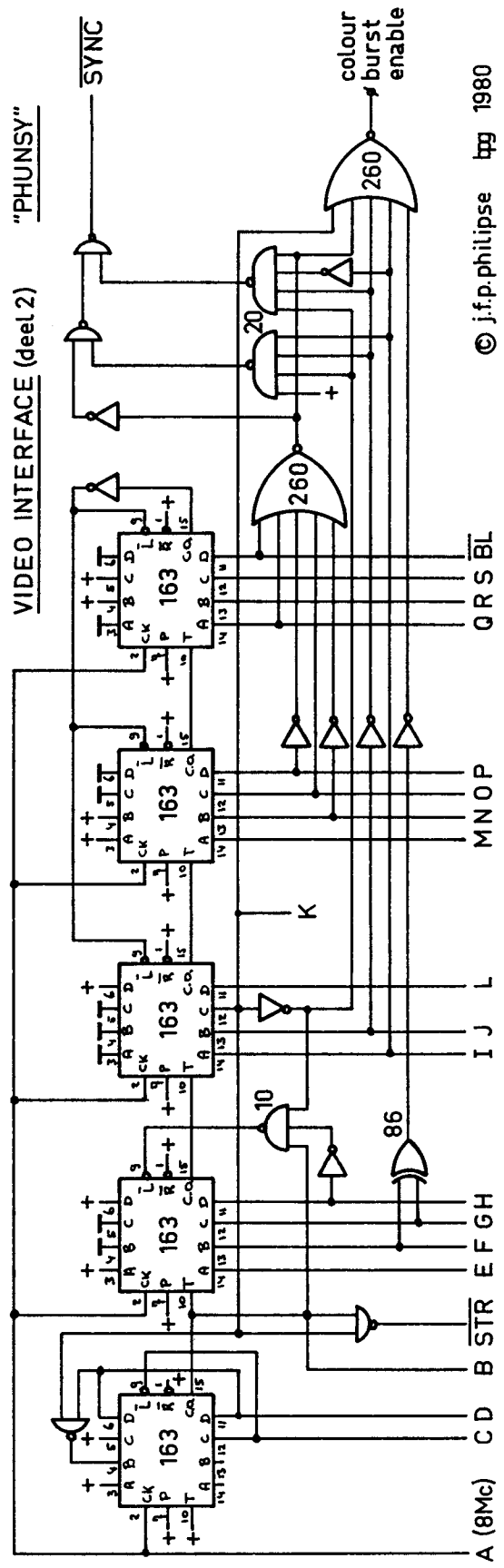
	=	74LS00	157 =	74LS157
	=	74LS04	163 =	74LS163A
	=	7407	166 =	74LS166
	10 =	74LS10	244 =	74LS244
	20 =	74LS20	260 =	74S260
	30 =	74LS30	365 =	74LS365
	70 =	7470	374 =	74LS374
	86 =	74LS86	379 =	74LS379

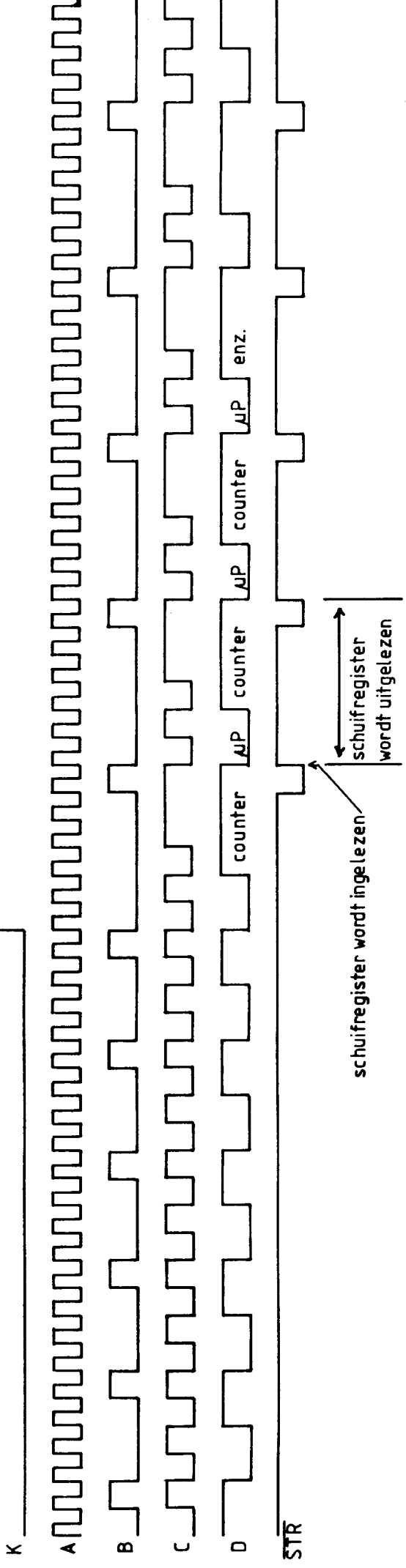
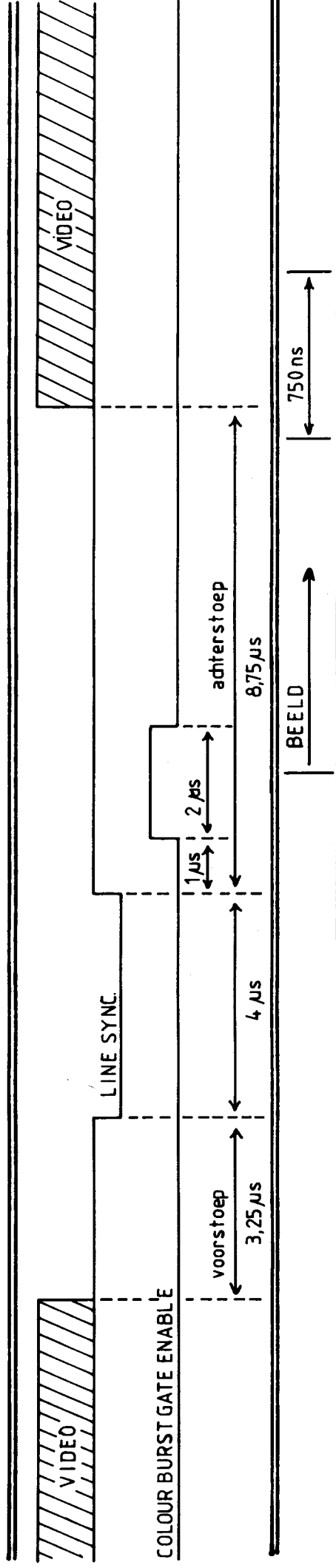
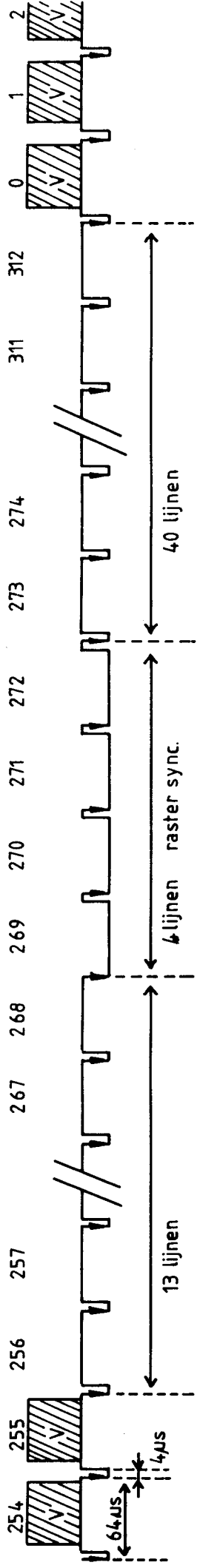
© j.f.p.philipse bpg 1980

BLOKSCHEMA VIDEO INTERFACE



"PHUNSY"





Deze geheugenprint is speciaal ontworpen voor "2650" geörienteerde systemen, terwijl de praktische uitvoering geheel ingepast is in het Phunsy kader. Dit wil echter niet zeggen dat de print niet voor andere 2650 systemen of zelfs andere processors te gebruiken is. Er zal dan wel het nodige veranderd moeten worden, daar komen we later op terug. Dit ontwerp gaat er vanuit dat de 2650 processorklok ca. 1 Mc. is, terwijl voor de refreshing ca. 8 Mc. moet worden aangeboden. Een probleem bij de opzet was van ongeveer dezelfde aard als in de video-interface; van de OPREQ, die verantwoordelijk is voor het beginnen van de I/O (memory) cycle, is wel bekend hoe lang hij duurt (1,5 μ sec.), maar niet wanneer deze komt.

Gemakshalve delen we het schema in de volgende blokken:

- A) adres (en bank) decoding
- B) geheugen en refreshing

A) De 32 K geheugen bestaat in feite uit 2 aparte blokken van 16 K elk. Beide blokken zijn geadresseerd van H'4000' t/m H'7FFF'. Elk blok kan echter "gebankt" worden. Dit is totaal 16 x mogelijk. Dit banken geschiedt m.b.v. de buslijnen Co \emptyset t/m Co3. Voor het fijne van de "bankzaken" verwijzen wij naar het hoofdstuk MONITOR, E.C.A.bulletin nr. 8, 1981. De adresdecoding ligt vast, de bankdecoding gebeurt met 4 schakelaars i.s.m. de lijnen Co \emptyset t/m Co3, die allen gebufferd zijn.

B) Zie voor refreshing ook het timing diagram.

Refreshing is onontbeerlijk voor dynamische RAM's, zonder dit zou de data slechts ongeveer 2 msec. worden vastgehouden. Maar door nu voortdurend een adres in te kloppen op de adresingangen kan voorkomen worden dat onze RAM's aan geheugenverlies gaan lijden. Elke RAM is opgebouwd uit een matrix die bestaat uit 7 rijen en 7 colommen. Deze rijen en colommen hebben naar buiten toe dezelfde aansluitingen. Met twee andere pinnen wordt geselecteerd of de rijen danwel de colommen aan de beurt zijn: RAS (Row Address Select) en CAS (Colom Address Select). Voor het refreshen wordt alleen de RAS gebruikt. Het hart van het refreshgebeuren is de programmeerbare 16-teller "74LS163". Deze wordt gevoed door de 8 Mc. klok. Tijdens het refreshen verschijnt aan de uitgangen A,B,C en D (pinnen 14, 13, 12 en 11) achtereenvolgens de tellerstanden 2,3,4,5,2,3,4,..... enz. Bij de overgang van 5 naar 2 wordt er een negatiefgaande puls aangeboden aan een 128-teller, de "74LS393". De uitgangen van deze teller worden via een gate-buffer aangeboden aan de adresingangen van de RAM's, terwijl bij het laagworden van het RAS signaal deze tellerstand wordt ingeklokt in de RAM's. Maar nu moet er gelezen of geschreven worden. Het begin van de I/O cycle kondigt zich aan met het hoog worden van de OPREQ. Dit schakelt een flip-flop, gevormd met twee NOR's (IC 9) om. De 16-teller maakt nu eerst z'n cyclus af t/m stand 5, daarna verandert de telcyclus, deze wordt nu A,B,C,D,E en F. Dit gebeurt door het hoogworden van uitgang D. Dit signaal zet nu de hele boel op z'n kop. De 128-teller gate-buffer wordt gedisabled. Daarvoor in de plaats worden de adreslijnen van de adresbus aangeboden. Dit gebeurt in twee keer. De laagste 7 adressen worden doorgelaten tijdens de tellerstanden B en C ($\overline{AL} = \emptyset$), de adreslijnen A7 t/m A13 gedurende de tellerstanden D en E ($\overline{AH} = \emptyset$). Bij de overgang van B naar C wordt de \overline{RAS} laag, waardoor de aangeboden adressen ingeklokt worden op de rijen in de RAM's. Bij de overgang van D naar E wordt één van de \overline{CAS} 's laag. Deze zijn n.l. apart opgesplitst in twee \overline{CAS} 'sen, voor elk blok één. Ze worden geselecteerd bij de bankselectie. Wordt een \overline{CAS} laag, dan wordt als $\overline{R/W}$ laag is (write) data van de databus in het geheugen gezet. Is de $\overline{R/W}$

hoog (read) dan wordt de data uit het geheugen bij het laagworden van uitgang D (16-teller) ingeklokt in de D-latch flip-flops (IC 4). Dit verdwijnt in de databus op het moment dat hier om gevraagd wordt. Het laag worden van D heeft ook tot gevolg dat de (NOR) flip-flop omgezet wordt en de 16-teller gepreset wordt in de stand 2. Hiermee is de I/O cycle afgewerkt en begint de refreshing weer. De teller telt weer 2, 3, 4, 5, 2, 3, 4, enz. en de 128-teller krijgt weer pulsen en gaat weer door waar hij gebleven was, terwijl de adresbus-gates weer gedisabled zijn.

B.K.D.R.A.M.

De hiervoor beschreven print kan ook, na een ingreep, dienen als 8 K dynamische RAM print. Als geheugen IC's moeten dan de typen "4096" gebruikt worden. Bij deze IC's dienen de pinnen 13 als chip-select. Deze gaan gebruikt worden om de set van 8 IC's (totaal 4 K bytes) te selecteren. Dit gebeurt niet meer met het $\overline{\text{CAS}}$ signaal, zoals in de 32 K versie. De $\overline{\text{CS}}$ van de IC's 18 t/m 25 gaan nu naar pin 8 van IC 8, terwijl de $\overline{\text{CS}}$ van de IC's 26 t/m 33 naar pin 8 van IC 12 gaan. Alle $\overline{\text{CAS}}$ 'sen van de geheugen IC's staan nu parallel.

De bankselectie vervalft. De DIL schakelaars hiervan worden nu gebruikt als adresdecodering, voor elk blok van 8 IC's vier schakelaars. Elk blok kan nu in het hele adresbestand van 32 K gezet worden, op elke plaats. Uitgekeken moet worden, dat niet parallel aan ROM, RAM geschakelt wordt. Sk 1/5 schakelt adreslijn 12, Sk 2/6 adreslijn 13, Sk 3/7 adreslijn 14 en met Sk 4/8 kan het betreffende blok uitgeschakeld worden.

De praktische veranderingen staan vermeld in de tabel, punten op de print die onderbroken moeten worden staan op het schema aangegeven als volgt: ~~—X—~~. Totaal 12 stuks. De diverse doorverbindingen kunnen het best met draad gelegd worden aan de onderkant van de print.

Andere systemen/processors

In andere systemen, waarin 16 adreslijnen gebruikt worden en uiteraard geen bankmogelijkheid aanwezig is, kunnen de bankselectieingangen gebruikt worden om b.v. adreslijn 15 te selecteren of eventuele andere selectielijnen. Om deze print te gebruiken op adressen lager dan adres H'4000' moet de adreslijn 14 ge-inverteerd worden, dit kan gedaan worden met een nog ongebruikte NOR-poort in het IC 16.

Ook moet men rekening houden met het feit dat een I/O cycle 10 refresh-klokpulsen duurt.

Verder moet er nog op het volgende worden gelet: zou de 8 Mc. refresh-ingklok betrokken worden van een aparte oscillator dan kunnen er problemen optreden doordat de refreshing niet synchroon loopt met de μP klok. Praktijk heeft uitgewezen dat een condensator van ca. 2 nF tussen pin 1 IC 9 en massa het probleem verhelpt.

De print is ontworpen voor gebruik van MKM 100 nF/100 volt condensatoren, de 250 volt typen zijn eigenlijk mechanisch te groot. Er kunnen sluitingen ontstaan tussen de zijkant van de condensatoren en eronderlopende printsporen, zeker bij gebruik van 250 volt typen. Ook moeten de IC-voeten met zorg bekeken worden of ze aan de onderkant geen sluitingen kunnen veroorzaken met printsporen die tussen twee pinnen doorlopen.

De ontkoppelaar's moeten het liefst tantaaltypen zijn van minstens 16 volt werkspanning.

Frank Philipse

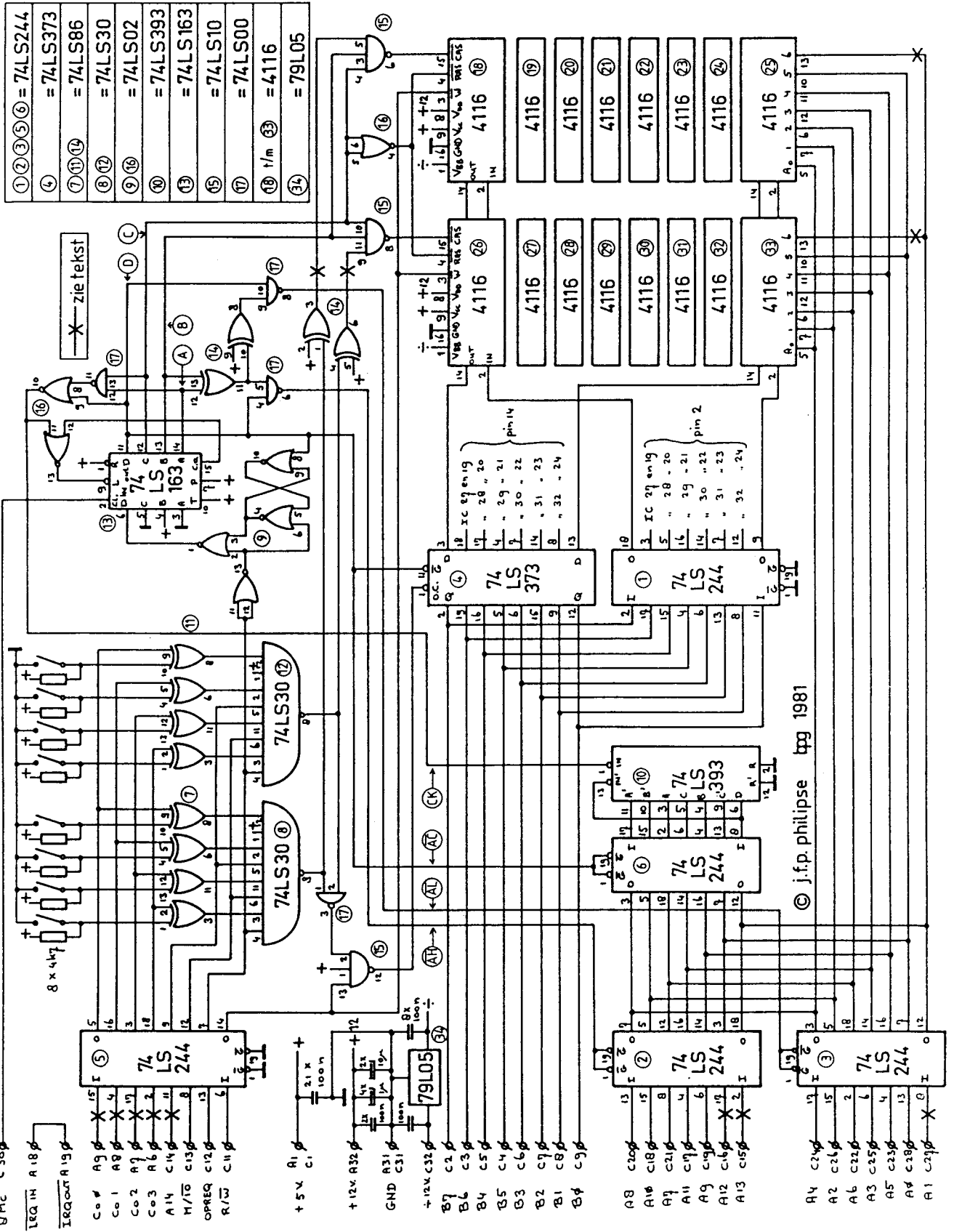
Ben Postema

c 1981

32 k D.R.A.M. print

SK1 2 3 4 5 6 7 8

8 Mc c30d

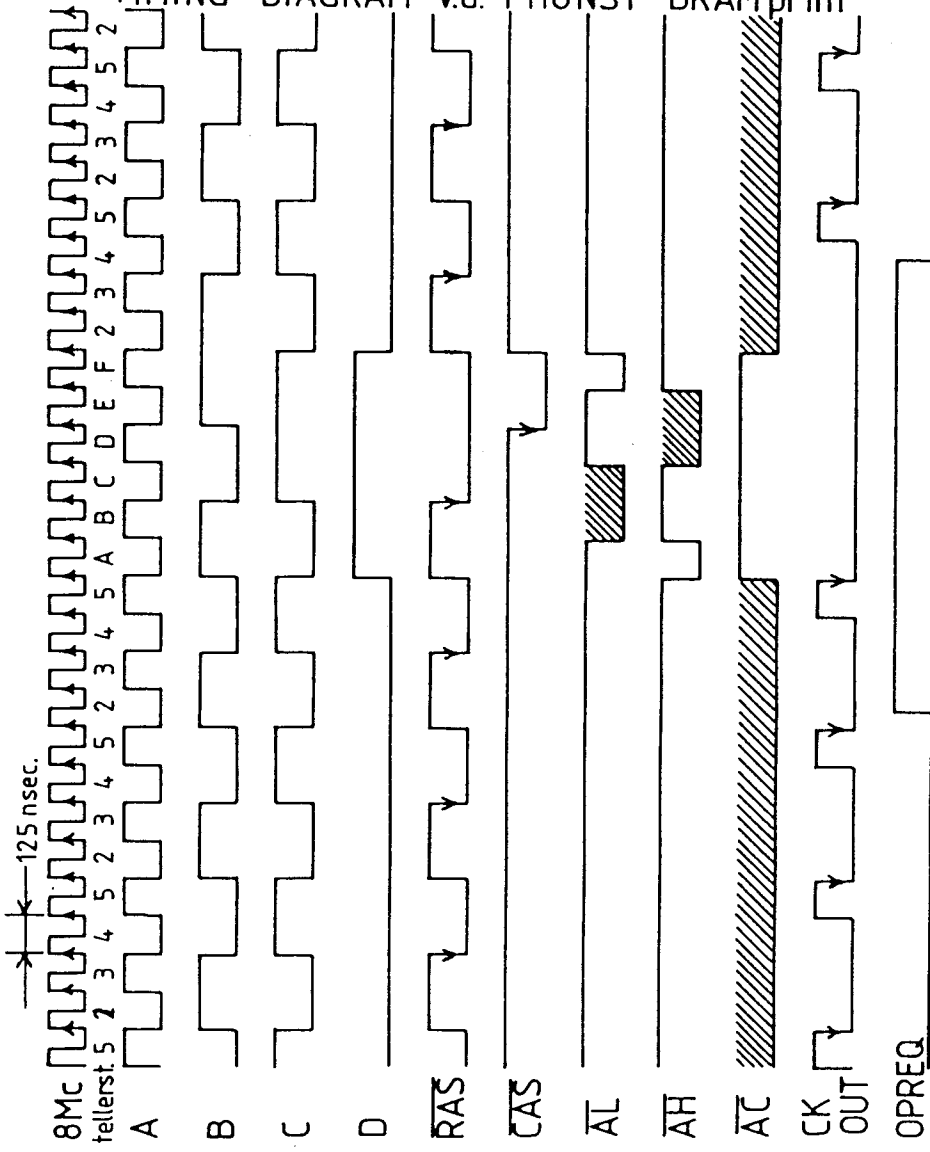


"PHUNSY"

- ① ② ③ ④ ⑤ ⑥ = 74LS244
- ⑦ = 74LS373
- ⑧ ⑨ ⑩ ⑪ ⑫ ⑬ = 74LS86
- ⑭ = 74LS30
- ⑮ = 74LS02
- ⑯ = 74LS393
- ⑰ = 74LS163
- ⑱ = 74LS10
- ⑲ = 74LS00
- ⑳ ㉑ ㉒ ㉓ = 4116
- ㉔ = 79L05

© j.f.p. philipse tpg 1981

TIMING-DIAGRAM v.d. PHUNSY DRAM print

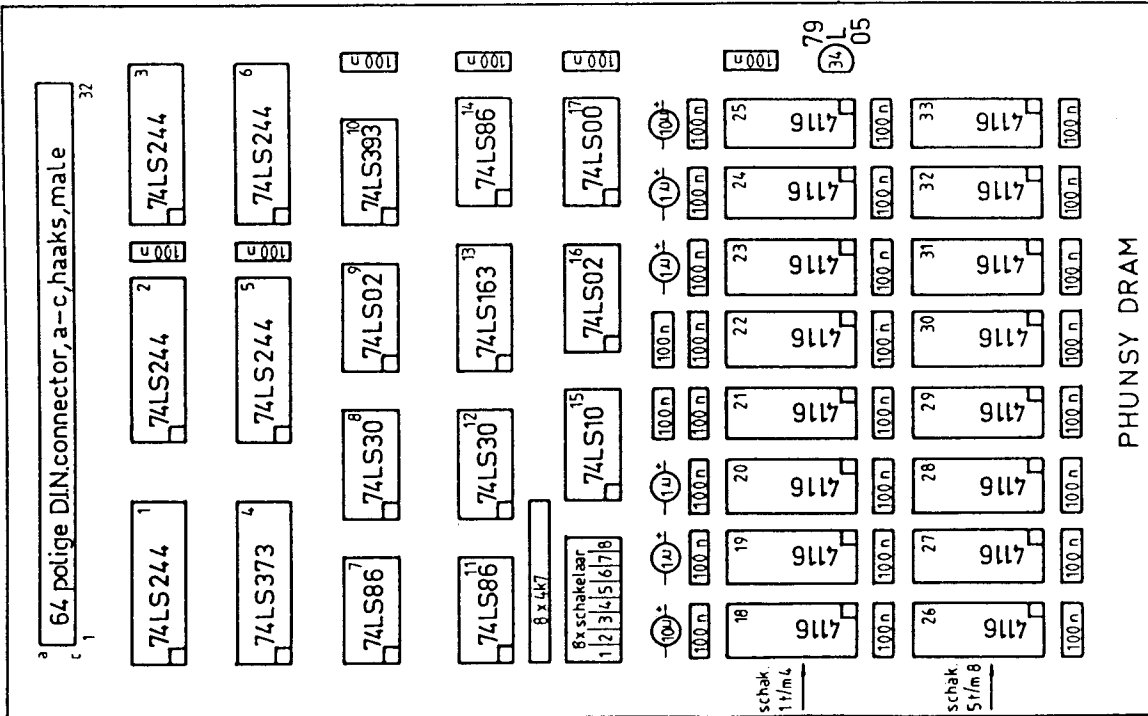


adres:	32 K versie				8 K versie			
	Sk 1/5	Sk 2/6	Sk 3/7	Sk 4/8	Sk 1/5 (A12)	Sk 2/6 (A13)	Sk 3/7 (A14)	Sk 4/8 (EN)
4000-7FFF	o	o	o	o	o	o	o	o
bank 0	o	o	o	o	o	o	o	o
bank 1	o	o	o	o	o	o	o	o
bank E	d	d	d	d	d	d	d	d
bank F	d	d	d	d	d	d	d	d
0000-0FFF	o	o	o	o	o	o	o	o
1000-1FFF	d	d	d	d	d	d	d	d
2000-2FFF	o	d	d	o	o	o	o	o
3000-3FFF	d	d	d	d	d	d	d	d
4000-4FFF	o	o	o	o	o	o	o	o
5000-5FFF	d	o	d	o	d	d	d	o
6000-6FFF	o	d	d	o	o	o	o	o
7000-7FFF	d	d	d	d	d	d	d	d
-----	x	x	x	x	x	x	x	x

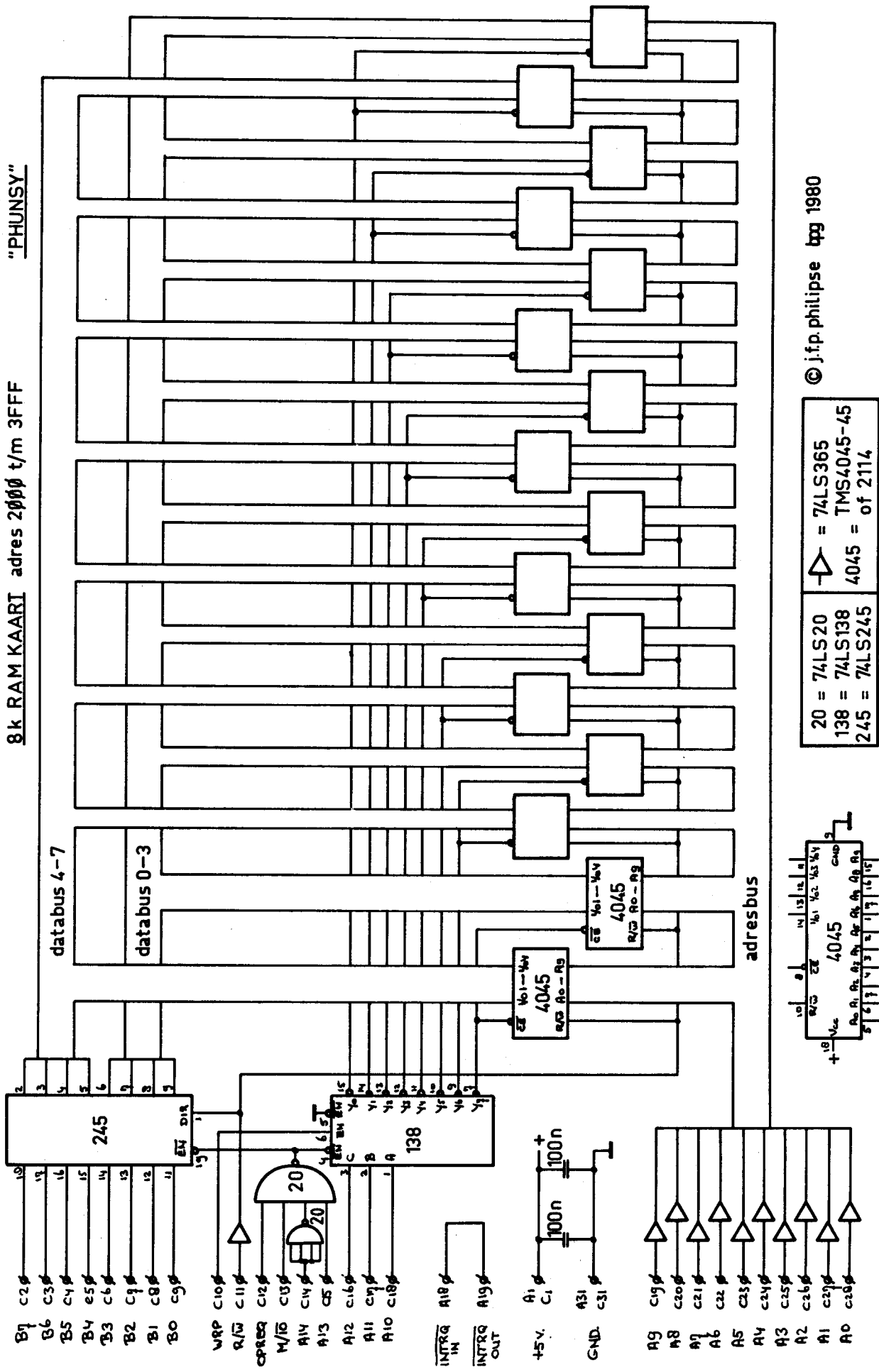
o = open d = dicht x = niet van belang

Wijzigingen, nodig voor
 ombouw naar 8 K DRAM :

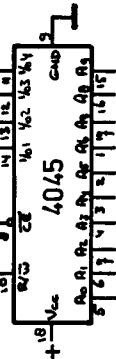
pin 13	IC 25	→	pin 8	IC 8
pin 13	IC 33	→	pin 8	IC 12
pin 5	IC 15	→	+ 5 volt	
pin 9	IC 15	→	+ 5 volt	
pin 11	IC 5	→	+ 5 volt	
pin 15	IC 5	→	ground	
pin 4	IC 5	→	connector C 14	
pin 17	IC 5	→	connector C 15	
pin 2	IC 5	→	connector C 16	
pin 17	IC 2	→	connector C 27	



8 k RAM KAARI adres 2000 t/m 3FFF "PHUNSY"



20 = 74LS20	↘	= 74LS365
138 = 74LS138		TMS4045-45
245 = 74LS245		4045 = of 2114



© j.f.p.philipse bpg 1980

- B7 c20
- B6 c30
- B5 c40
- B4 e50
- B3 c60
- B2 c70
- B1 c80
- B0 c90

databus 4-7

databus 0-3

adresbus

- WRP c100
- R/W c110
- CPREQ c120
- M/IO c130
- A14 c140
- A13 c150
- A12 c160
- A11 c170
- A10 c180

245

20

20

138

4045

4045

4045

4045

4045

4045

4045

4045

4045

4045

4045

4045

4045

4045

4045

4045

4045

4045

4045

4045

4045

4045

- INTRQ IN
- INTRQ OUT
- +5V
- GND
- A9
- A8
- A7
- A6
- A5
- A4
- A3
- A2
- A1
- A0

A18

A19

A1

A2

A3

A4

A5

A6

A7

A8

A9

A10

A11

A12

A13

A14

A15

A16

A17

A18

A19

A20

A21

A22

A23

A24

- C1
- C31
- C19
- C20
- C21
- C22
- C23
- C24
- C25
- C26
- C27
- C28

100n

100n

100n

100n

100n

100n

100n

100n

100n

100n

100n

100n

100n

100n

100n

100n

100n

100n

100n

100n

100n

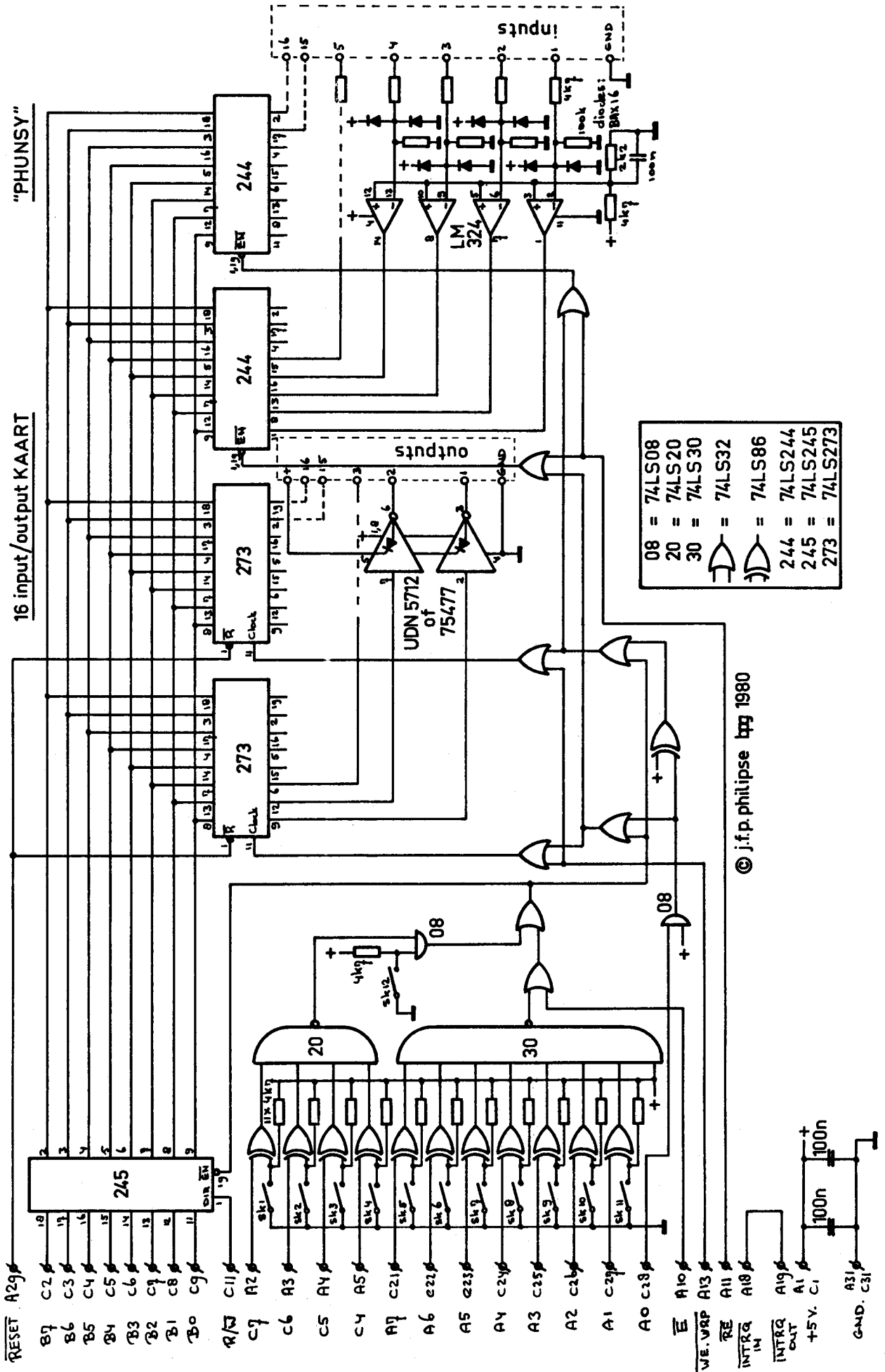
100n

100n

100n

100n

100n

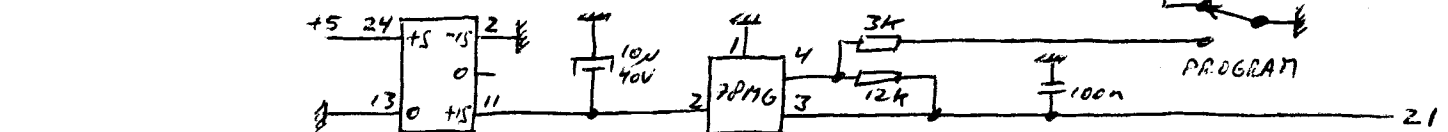
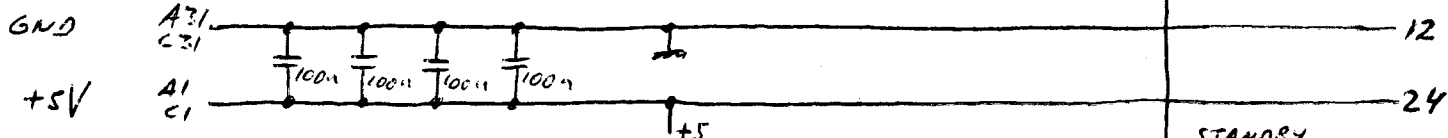
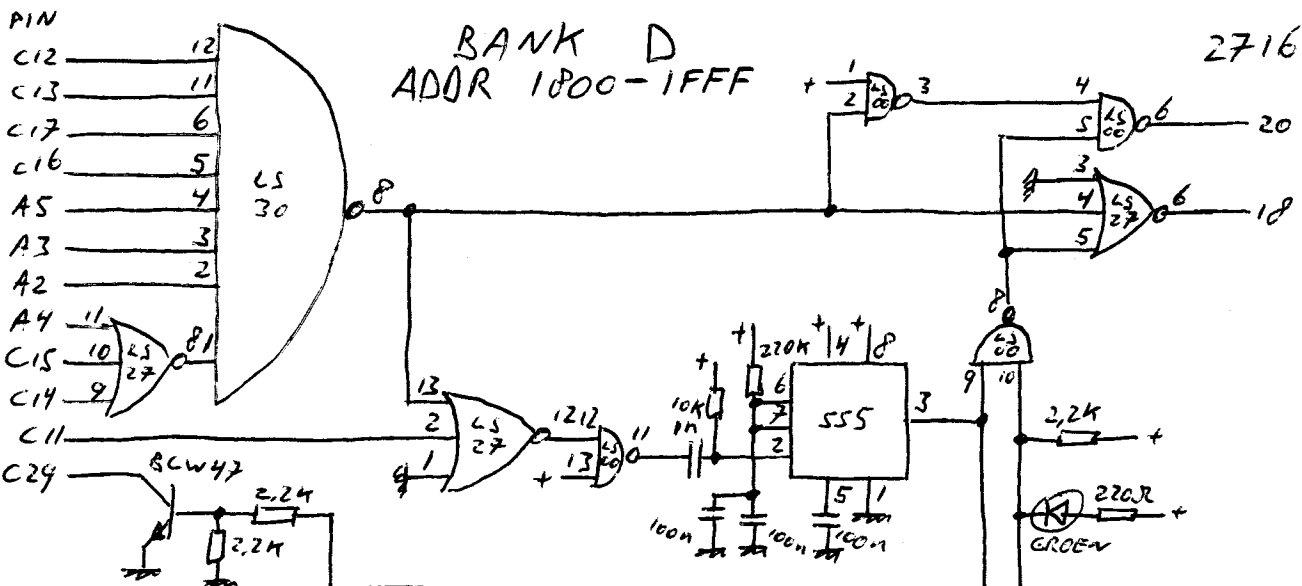


© j.f.p. philipse bpg 1980

PHUNSY 2716 PROGRAMMER

PHUNSY BUS

- OPREQ C12
- M/I0 C13
- A11 C17
- A12 C16
- C4 A5
- C6 A3
- C7 A2
- C5 A4
- A13 C15
- A14 C14
- R/W C11
- OPACK C29



A10	C18	19
A9	C19	22
A8	C20	23
A7	C21	1
A6	C22	2
A5	C23	3
A4	C24	4
A3	C25	5
A2	C26	6
A1	C27	7
A0	C28	8
B7	C2	17
B6	C3	16
B5	C4	15
B4	C5	14
B3	C6	13
B2	C7	11
B1	C8	10
B0	C9	9

De MDCR (Mini Digitale Cassette Recorder).

De MDCR is een recorder voor het opslaan van digitale informatie op een cassette. De informatie wordt serieel op de cassette gezet met een snelheid van 6000 BAUD. Dat wil zeggen: 6000 bits per seconde ofwel 750 bytes per seconde. De informatie wordt in blokken van 256 bytes op de cassette gezet en er kunnen 128 blokken op elke kant van een cassette. De spoeltijd van de cassette is gelijk aan de leestijd (afspeeltijd) en is ruim 90 seconde. Tussen de blokken op de tape is een ruimte open gelaten om de recorder te kunnen stoppen voor elk blok. Elk blok wordt voorafgegaan door een bloknummer, zodat het MDCR programma weet waar hij is. Bij het schrijven (opnemen) stopt de tape steeds voor een blok omdat het bloknummer blijft staan. Zou het bloknummer steeds opnieuw worden geschreven dan zou het in de loop der tijd te veel kunnen verschuiven ten opzichte van de oorspronkelijke positie. De tape begint met 1 S leeg, dan het eerste bloknummer (00), dan 75 mS leeg, dan het eerste blok, dan 275 mS leeg, dan het tweede bloknummer (01), dan weer 75 mS leeg, dan het tweede blok, etc.. Het MDCR programma werkt uitsluitend met hexadecimale getallen, dus de blokken zijn ook zo genummerd. Van 00 t/m 7F. (0 t/m 127.) De tijd tussen de bloknummers is 700 mS. Dit maal 128 plus aan het begin van de cassette 1 S leeg duurt samen 90,6 seconde. Door een dopje uit de cassette te halen, voorkomt men dat een tape gewist wordt.

Gebruiksaanwijzing van de MDCR vanuit de monitor.

Het commando T van de monitor is identiek aan het commando IUU. (zie monitor beschrijving.) Dat houdt in: wanneer de monitor een T tegenkomt gaat hij naar het MDCR programma. Voor elk commando voor de MDCR moet dus een T staan.

De commandos zijn:

CAT	catalog
RUN	run
REW	rewind
SAVE	save
LOAD	load
LOCK	lock
DELETE	delete
RENAME	rename
INIT	initialize

INIT:

Dit commando is ervoor om een nieuwe cassette te initialiseren. Wanneer dit wordt gedaan, wordt de tape voorzien van bloknummers en de blokken worden volgezet met nullen. De bloknummers worden om de 700 mS op de tape gezet tot het eind van de tape bereikt is. Er worden echter altijd 80 (128) blokken op de tape gezet. Dus als de tape langer duurt dan 90 S (normaal ongeveer 96 S) wordt na bloknummer 80 (128) tussen de bloknummers geen blok meer geschreven. Dit commando nooit uitvoeren met een cassette waar nog waardevolle informatie op staat! Natuurlijk kan INIT wel gebruikt worden om alle gegevens van een tape te wissen.

CAT:

Hiermee wordt een overzicht van wat er op een cassette staat op het beeld gezet. Dit gebeurt in 8 kolommen. De 1e geeft de programmanummers aan (1 t/m 16 decimaal). De 2e de namen van de programma's, de 3e het geheugenadres waar het programma hoort te staan, de 4e de lengte van het programma, de 5e geeft het bloknummer aan van de tape waar het programma staat (dit wordt met 80 verhoogd als het programma gelocked is), de 6e is voor de taalcode, de 7e geeft de geheugenbank aan waar het programma hoort en de 8e is voor diverse toepassingen te gebruiken.

SAVE:

Dit wordt gebruikt om een programma dat in het geheugen staat op de cassette te schrijven. Vanuit de monitor b.v.:

```
TSAVE VERHAAL 4000 800 L2 Q1
```

De T vertelt de monitor dat er iets met de MDCR moet gebeuren. SAVE vertelt het MDCR programma dat er iets op tape gezet moet worden. Daarna moet de naam ingetypt worden van hetgeen op tape gezet moet worden. Deze naam mag maximaal 8 aaneengeschreven karakters bevatten. Na de naam moet een spatie volgen om aan te geven waar de naam ophoudt. Daarna komt het beginadres van het op tape te zetten programma gevolgd door de lengte. Deze mogen beide variëren tussen 0 en 7FFF. Nu kunnen er nog drie dingen toe gevoegd worden. Gebeurt dit niet, dan neemt het MDCR programma aan dat ze 00 zijn. Deze drie dingen zijn een L voor de taal, een Q voor de bank en een S voor eventueel gebruik, elk gevolgd door een getal tussen 1 en FF en een spatie. Bevind het programma zich in Q1 zoals in het voorbeeld, dan moet dit vermeld worden, anders wordt totaal andere informatie op tape gezet. (voor details zie monitor en editor beschrijving.) De L (language) geeft aan wat voor soort informatie op tape wordt gezet. Voorlopig is vastgesteld:

00 diversen

01 2650 machinetaal met startadres is executeadres

02 gewone text

03 basic zonder speciale keywordcodes

04 lotel programma's

Het voorbeeld is dus een stuk text met de naam VERHAAL dat begint op adres 4000, 800 bytes lang is en in bank 1 staat. (let wel: alles hexadecimaal.) Er kan niet in een keer een stuk text op tape gezet worden dat in 2 of meer banken staat. Dit moet ook in 2 of meer keren op de cassette gezet worden.

Wordt een naam opgegeven die reeds op de cassette staat, dan wordt er naar de overige gegevens niet meer gekeken en de gegevens die op de tape staan worden gebruikt.

Is dat echter toch gewenst dan moet een andere naam gebruikt worden of het oude programma moet eerst van de cassette gewist worden. (zie DELETE.)

Het commando SAVE werkt in het laatste geval niet als het programma dat eigenlijk al op tape staat gelocked is. (zie LOCK.)

LOAD:

Dit commando is ervoor om iets van de cassette in het geheugen te zetten. b.v.:

TLOAD VERHAAL

verdere gegevens zijn niet nodig, omdat alle informatie op de cassette aanwezig is om de informatie exact weer in het geheugen te zetten zoals het er stond.

RUN:

Dit kan alleen gebruikt worden wanneer een programma in 2650 machinetaal is geschreven en bij save L1 is opgegeven. b.v.:

TRUN PROGRAM

Nu wordt eerst het programma van de tape in het geheugen gezet, waarna het programma meteen gerund wordt.

DELETE:

Dit is om een programma wat niet meer gewenst is van de tape te wissen. b.v.:

TDELETE VERHAAL

Dit commando werkt niet als deze text gelocked is. (zie LOCK.)

RENAME:

Wanneer een naam van een programma dat op de cassette staat veranderd moet worden kan dat met RENAME. b.v.:

TRENAME VERHAAL STORY

In het eerste blok op de cassette wordt hiermee alleen de naam veranderd. De rest van de gegevens blijven gewoon gehandhaafd. Dit commando werkt niet als de tape gelocked is. (zie LOCK.)

LOCK:

Dit commando is om te voorkomen dat aan een programma op tape niets veranderd kan worden. b.v.:

TLOCK VERHAAL

Nu wordt in het eerste blok op de cassette in de catalogus bij het betreffende programma een bit gezet. (het 7e bit van de byte die het eerste bloknummer aangeeft.) Hierdoor zijn een paar commandos die iets aan een programma kunnen veranderen niet meer te gebruiken voor dat programma. Dit zijn: SAVE, DELETE en RENAME. Omdat het een software lock is is het nog wel mogelijk, dat als bij experimenten iets fout gaat, de MDCR toch op wissen wordt gezet. Om dit te voorkomen is het zeer aan te bevelen een cassette met een dopje erin niet in de MDCR te laten wanneer hiermee niet gewerkt wordt.

UNLOCK:

Dit is een commando dat een lock van een programma op tape opheft. b.v.:

TUNLOCK VERHAAL

REW:

Dit staat voor rewind ofwel terugspoelen. b.v.:

TREW

Het is aan te bevelen om een cassette niet op te bergen wanneer de tape niet is teruggespoeld. Dit omdat er dan door diverse oorzaken (mechanische vooral) belangrijke informatie verloren kan gaan. Er zijn een paar commandos die na uitvoering automatisch terugspoelen. Dit zijn: CAT, LOCK, UNLOCK, DELETE en RENAME.

De monitor heeft een input buffer van 255 karakters. Mede hierdoor is het mogelijk om een aantal commandos achterelkaar in te typen, waarna ze ook respectievelijk uitgevoerd worden. b.v.:

TSAVE VERHAAL 4000 800 L2 Q1 TLOCK VERHAAL TCAT

Na een return wordt eerst het stuk text op tape gezet dat staat in bank 1 van 4000 t/m 47FF. Daarna wordt het gelocked en daarna verschijnt het aangepaste inhouds-overzicht op het scherm. Na CAT wordt te tape automatisch teruggespoeld.

ERRORS:

READ ERROR:

Verschijnt wanneer er iets niet goed op de cassette staat. B.v. als een bit verkeerd gelezen wordt (drop-out) dan klopt een checksum niet en de tape stopt met lezen.

CASS ERROR:

Als de cassette om een of andere reden stopt of vast zou lopen voor het einde van de tape is bereikt, verschijnt deze foutmelding. Ook als bij spoelen de cassette uit de recorder wordt gehaald.

INPUT ERROR:

Verschijnt wanneer het MDCR programma de ingetikte gegevens niet begrijpt.

TOO LONG ERROR:

Op een cassette kan niet meer dan 31 3/4 Kbytes worden gezet. Wanneer men tracht dit te overschrijden verschijnt deze foutmelding.

NOT ON TAPE ERROR:

Wanneer na een load, lock, unlock, delete of rename een naam wordt opgegeven die niet op de cassette voorkomt, kan dit programma uiteraard ook niet behandeld worden.

CAT FULL ERROR:

Er kunnen 16 verschillende programma's op een cassette worden opgeslagen. Wil men dit overschrijden, dan volgt deze foutmelding.

LANGUAGE ERROR:

Als een RUN commando is gegeven en de taalcode behorende bij dat programma is niet 01, zal het programma niet gerund worden omdat aangenomen wordt dat het dan geen 2650 machinetaal betreft.

LOCKED ERROR:

Wanneer een programma gelocked is kan het door het MDCR programma niet verwijderd of veranderd worden. (zie LOCK). Wordt dit echter toch geprobeerd dan verschijnt deze foutmelding.

WRITE PROTECTED ERROR:

Dit is een melding die verschijnt indien er iets op de cassette veranderd of bijgevoegd moet worden, terwijl de cassette niet voorzien is van een dopje (links boven in de cassette).

Er dient nog vermeld te worden, dat op het moment dat er door het MDCR programma gelezen wordt en de cassette wordt uit de recorder genomen, dan volgt geen error. Het MDCR programma blijft wachten op het volgende bit. Gebeurt dit, dan resetten of een cassette (zonder dopje) in de recorder plaatsen waarna een read error volgt.

Ook is het noodzakelijk dat wanneer een load of run commando wordt opgegeven, eerst een cat wordt gedaan indien dit het eerste is wat gedaan wordt na het inzetten van de cassette. De gegevens over de programma's worden nl. door load en run niet eerst van de cassette gehaald en het kan dus zijn dat de gegevens van een andere cassette nog in het geheugen staan. Als dat zo is zal er normaalgesproken een not on tape error verschijnen, tenzij dezelfde naam op de vorige tape ook voorkwam en is het zeer waarschijnlijk dat verkeerde informatie van de cassette in het geheugen wordt gezet.

Wanneer er programma's van een cassette worden gehaald met delete wordt de rest van de programma's niet automatisch teruggeschoven, zodat alles weer netjes achterelkaar op de tape staat. Er komt dus een gat van een bepaalde hoeveelheid blokken. Bij het saven van een programma wordt gezocht naar het eerste gat dat groot genoeg is voor de hoeveelheid informatie. Wil men toch alles aaneensluitend op de cassette hebben dan kan dat door van voor naar achter, lettend op de bloknummers, de programmas te deleten en weer te saven.

Mini Digital Cassette Recorder

- - - -

Phunsy Format (1981)

De cassette is zo ingedeeld dat er 128 blokken van elk 256 bytes per kant op kunnen. De Speeltijd is ruim 90 S per kant en dat komt neer op 700 mS per blok. Aan het begin van de cassette is 1 S ongebruikt gelaten. Daarna volgen om de 700 mS een bloknummer. Het eerste bloknummer is 00 en er wordt doorgenummerd tot aan het eind van de tape. De blokken 00 t/m 7F worden gebruikt en de blokken die daarna nog op de tape staan niet. Dit zijn er meestal nog ongeveer tien. Bij initialisatie worden de bloknummers op de tape gezet en achter elk bloknummer van 00 t/m 7F wordt een blok gezet volgeschreven met nullen. Achter de resterende bloknummers wordt niets geschreven. Voor en na elk bloknummer staat AA (hex). Een datablok begint ook met AA, daarna een checksum, daarna de data (256 bytes) en het eindigt weer met AA.

Schematisch:

```

bot                                                                 eot
----N-BK---N-BK---N-BK---N-BK//--N-BK---N-BK---N-----N-----
!  !      !      !      !      !      !      !      !      !      !
!1S.!  00 !  01 !  02 !  03 !  7E !  7F !  80 !  81

```

N bloknummer
BK blok
bot begin of tape
eot end of tape

```

/--AAbnAA---AAcsxxxxxxxxxxxxxxxxxxAA-----AAbnAA--/
!          !                                     !
!-75 mS- !                                     !
!          !                                     !
!          -700 mS-                             !

```

- leeg
AA hex byte AA
bn bloknummer
cs checksum
xx data

Na elk blok is ongeveer ²⁷⁵~~350~~ mS ruimte om te kunnen stoppen. Bij het schrijven op tape worden n.l. de bloknummers niet opnieuw geschreven omdat de bloknummers anders zouden gaan verschuiven.

De bytes worden op tape geschreven beginnend met bit 0. De checksum is de som van het volgende datablok van 256 bytes, zonder iets met de carry te doen.

Het eerste blok (00) wordt gebruikt voor de catalog. Hierin kunnen de gegevens voor 16 verschillende programma's worden opgeslagen. De eerste helft van het blok heeft twee functies. Van byte 00 t/m 7F wordt bit 7 van elke byte gebruikt om aan te geven in welk blok informatie staat en in welk blok niet. Byte 00 bit 7 voor blok 00 etc. t/m byte 7F bit 7 voor blok 7F. Bit 0 t/m 6 van deze bytes worden gebruikt om de programma namen in op te slaan. Elke naam mag 8 bytes lang zijn, dus kunnen er 16 namen in die er gewoon met 7-bits ascii instaan. Byte 00 t/m 07 voor de eerste naam, 08 t/m 0F voor de tweede etc..

De tweede helft van het blok bevat de rest van de informatie. Byte 80 t/m 87 is voor programma 1, 88 t/m 8F voor programma 2, etc. en als laatste F8 t/m FF voor programma 16. Voor elk programma 8 bytes dus.

Het 1e byte is voor het startadres, lower order.

Het 2e byte is voor het startadres, higher order.

Het 3e byte is voor de lengte, lower order.

Het 4e byte is voor de lengte, higher order.

Het 5e byte heeft twee functies. In bit 0 t/m 6 staat aangegeven welk blok het eerste is van een bepaald programma. Een programma wordt altijd in opeenvolgende blokken gezet. Het 7e bit van het 5e byte wordt gebruikt als software lock. Staat er in dit bit een 1 dan kan het programma niet veranderd of overschreven worden. Is dit een 0 dan kan dat wel.

Het 6e byte geeft de taal aan van het programma (of text of wat ook). Wat dit betreft ligt nog niet geheel vast hoe, maar voorlopig is het als volgt. Bit 4 t/m 7 geven aan voor welke processor. 0 voor de 2650, 1 voor de 6502, 2 voor de 6800, etc.. Dit is maar een voorbeeld, echter voor de 2650 is het zo al in gebruik. Bit 0 t/m 3 is voor welk soort programma. 0 voor diversen. 1 voor machine taal, 2 voor gewoon text, 3 voor basic, 5 voor source code assembler, etc..

Het 7e byte is een extra adres byte. Het kan gebruikt worden voor banknummers, zoals dit bij de phunsy gedaan wordt, maar kan ook gebruikt worden als extra byte bij het startadres, wanneer er gewerkt wordt met een micro die meer geheugen kan adresseren.

Het 8e byte tenslotte, wordt gebruikt om aan te geven het hoeveelste deel van een programma dit is en uit hoeveel delen het bestaat. Er kunnen n.l. best programma's zijn die veel langer zijn dan 32 K, of programma's die om een andere reden niet aan een stuk weggeschreven kunnen worden. Bit 0 t/m 3 geeft aan het hoeveelste deel en bit 4 t/m 7 geeft aan uit hoeveel delen het programma bestaat. Zo kan een programma dus uit 16 delen bestaan. Staat er 00 in dit byte, dan is er geen sprake van meerdere delen. Dit komt dan overeen met 11. Staat er b.v 53 in dan is het het derde deel van 5 delen.

