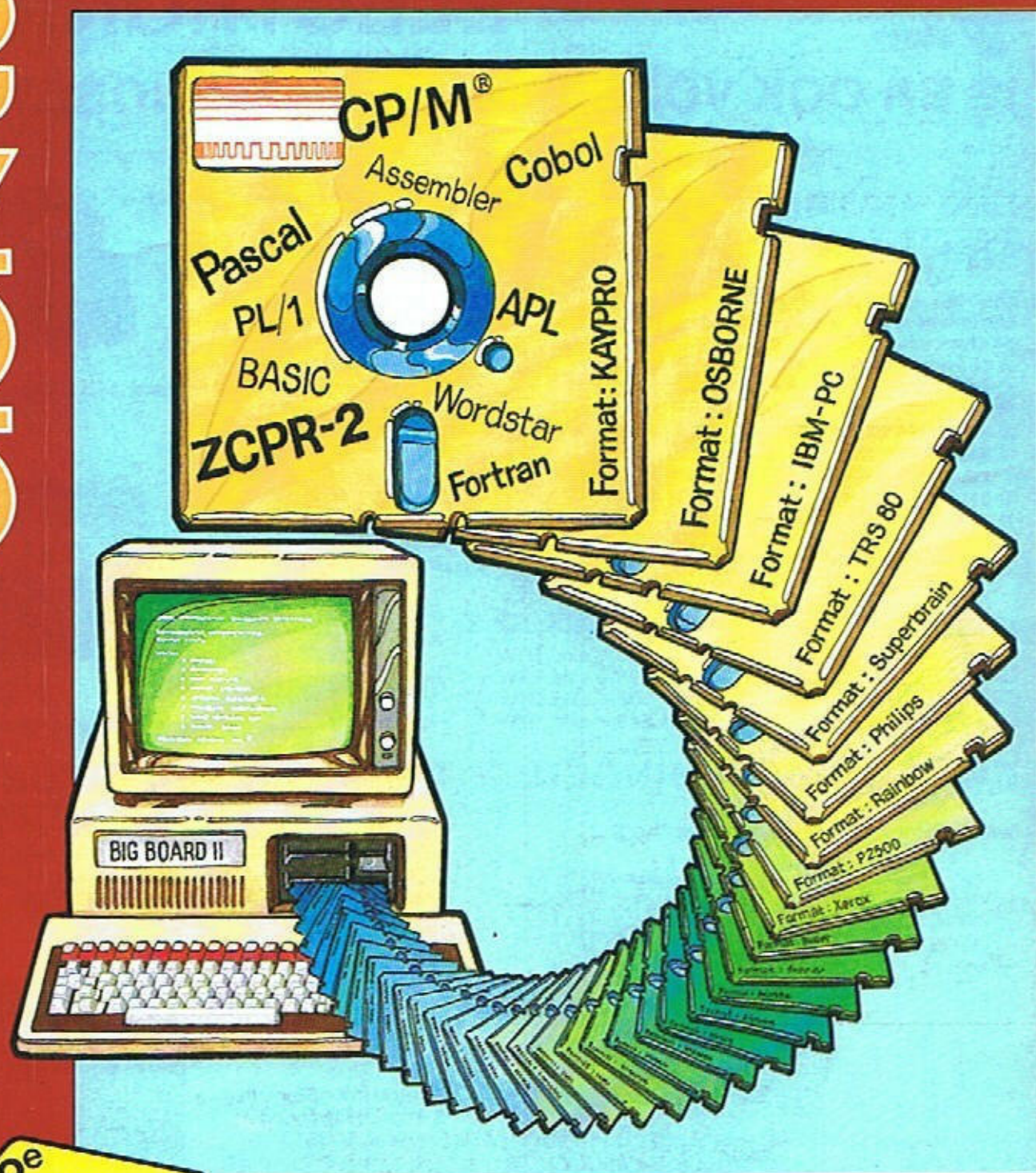


COMPUTING

elektronica **extra**



2e
speciale
computer-uitgave

de Big Board II: een CP/M-kameleon
 BASICODE-2 en kleuren-graphics voor de Octopus 65
 IBM-uitbreidingen • C64-expansion-port • 16-bitters

Perspektieven

"CP/M totaal" is het thema van deze Elektuur-uitgave, waarin het veelgebruikte computer-operating-system CP/M voorgesteld wordt. Ook mensen die reeds langer met een CP/M systeem werken en alle mogelijkheden van dit systeem kennen, vinden hier toch een hoop interessante dingen. In deze tweede computer-special wordt namelijk een bijzonder universeel en doorzichtig Z80-systeem voor zelfbouw beschreven, waarmee men nu en in de toekomst alle kanten uit kan. Deze CP/M-computer, de "Big-Board II", is ook voor de beginnende computerbouwer een geschikt systeem. Hard-disk(Winchester)-systemen hebben net als veel andere apparatuur in de computerbranche de gunstige eigenschap steeds goedkoper te worden, waardoor ze nu ook voor de amateur bereikbaar zijn. Winchester-disks zijn voor Big-Board II helemaal geen probleem door de aanwezigheid van een zogenaamde SASI-bus, zodat zo'n massageheugen zonder veel extra kosten aangesloten kan worden. Behalve veel geheugen biedt de hard-disk ook snelle toegangstijden, zodat de computer bovendien veel snel-

ler kan werken. In de volgende computer-special zullen we dan ook een hard-disk-uitbreiding beschrijven.

Een andere sterke eigenschap van Big-Board II is zijn enorme skala aan input- en output-kanalen. Zowel seriële als parallelle toetsenborden kunnen direkt worden aangesloten. Ook twee printer-aansluitingen (serie en parallel) zijn standaard aanwezig. Een RS232-aansluiting voor terminal en modem is eveneens een geïntegreerd onderdeel van de hardware. Het kiezen van een geschikte floppy-drive kan bijna geen probleem geven, want micro(3 1/2"-), mini(5 1/4"-) en maxi(8")-loopwerken kunnen gelijktijdig aangesloten worden. Het meegeleverde software-pakket VARBIOS maakt namelijk het werken met verschillende typen loopwerken mogelijk. Alsof dit alles nog niet genoeg is, kan VARBIOS ook nog diverse floppy-formaten, onder meer van OSBORNE, IBM-PC en KAYPRO, direkt lezen en beschrijven.

De ene software is de andere nog niet. Veel professionele gebruikers werken graag met CP/M, maar er zijn echter ook velen die een hekel hebben aan CP/M.

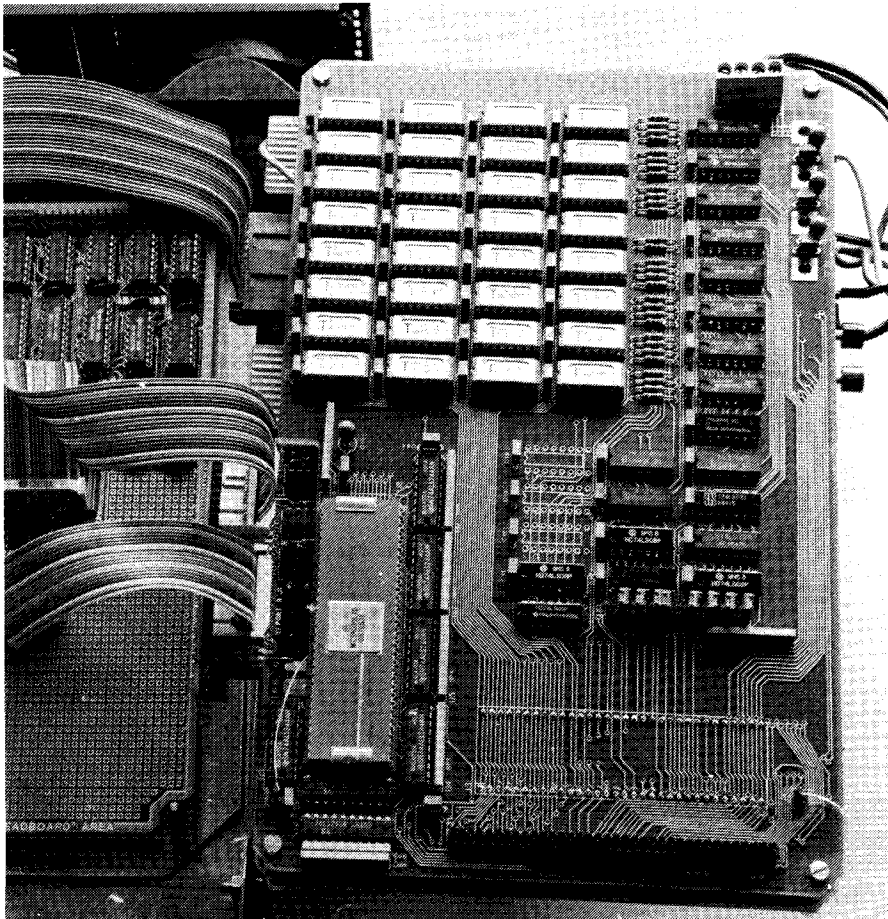
Veel nadelen van CP/M worden hier weg-gewerkt door het nieuwe operating-system "ZCPR-2", dat zonder meer CP/M-kompatibel is en heel wat voordelen biedt t.o.v. CP/M. De source-code van ZCPR-2 is public-domain-software, zodat iedereen deze voor niet-kommerciële toepassingen vrij kan gebruiken.

De in deze uitgave voorgestelde CP/M-machine is zowel hard- als software-matig slechts het topje van een ijsberg. Ten tijde van de redaktionele sluitingsdatum van deze Elektuur-Computing hadden wij in het Elektuur-lab een "Turbo-Big-Board" met een 1-megabyte-RAM-disk, een 50-megabyte-hard-disk en, zoals u op de foto kunt zien, een 68000-kaart, waarop uiteraard CP/M-68K geïnstalleerd is. Daarmee verandert Big-Board II in een twee-processor-systeem, waarbij een 8-bits- en een 16-bits-processor onder een dak samenwerken. De 68000 heeft hierbij zijn eigen 16-bits brede geheugen, wat een flinke werksnelheid garandeert bij een klok-frekwentie van 10 MHz.

En dan nu de slotvraag: welke software is er zoal voor de CP/M-machine? Het antwoord hierop is niet eenvoudig, want dat is een hoop: assembler, BASIC, PASCAL, ALGOL, FORTRAN, COBOL, zelfs de programmeertaal C, al deze talen zijn op de Big-Board II geïmplementeerd.

Natuurlijk zijn we Octopus 65 niet vergeten. Dankzij de 65SC802 wordt de Octopus 65 zonder verdere hardware-wijzigingen een 16-bitter. Met de 65SC816, waarvoor reeds een nieuwe CPU-kaart ontworpen wordt, kan men zelfs 16 Mbyte adresseren. Bovendien is ook een hogere klok-frekwentie mogelijk. Verder vindt u een verzameling nieuwe software voor de "oude" Octopus 65.

We hebben de indruk dat er een grote belangstelling is voor de grafische kaart waarmee in Elektuur in september 1985 werd gestart. De programma's en foto's op de binnenomslag van dit nummer tonen nog eens duidelijk de mogelijkheden van deze kaart. Heeft U gezien wat zo'n klein BASIC-programma allemaal op het scherm kan toveren? De grafische kaart is even goed geschikt voor professionele als voor hobby-toepassingen. We kennen een luidspreker-ontwerper die de Octopus gebruikt voor het sturen van een spectrum-analyzer! De grafische kaart geeft de meetwaarden twee- of driedimensionaal weer op een kleurenmonitor. Tot zover de professionele toepassingen. U kunt de kaart ook gewoon gebruiken om er mooie tekeningen mee te maken, zoals de foto's in deze uitgave laten zien. Grauwe wiskundige formules kunnen op het beeldscherm prachtige grafische beelden creëren.



INHOUD

Elektuur Computing

Deze *Elektuur Computing* is een uitgave van:

Uitgeverij. Elektuur BV,
Peter Treckpoelstraat 2-4, Beek (L)
Telefoon: 04402-89444, Telex 56617
Korrespondentie-adres: Postbus 75,
6190 AB Beek (L)
Kantoor tijden: 8.30-12.00 en 12-16.00 uur
Direkteur: J.W. Ridder,
Bourgognestraat 13a, Beek (L)

Deze uitgave is samengesteld door de redactie van het elektronicamaandblad *Elektuur*.

Medewerkers: A. Nachtmann,
P. vd. Linden, A. Bakkers, W. v. Dinther,
B. Fasel, G. Klein, P. Lavigne, D. Meyer,
A. Sevriens, P. Theunissen

Auteursrecht

De auteursrechtelijke bescherming van deze *Elektuur Special* strekt zich mede uit tot de illustraties met inbegrip van de printed circuits, evenals tot de ontwerpen daarvoor.

In verband met artikel 30 Rijksoktrooiwet mogen de in deze uitgave opgenomen schakelingen slechts voor particuliere of wetenschappelijke doeleinden vervaardigd worden en niet in of voor een bedrijf.

Het toepassen van schakelingen geschiedt buiten de verantwoordelijkheid van de uitgeefster.

Uitgeversmaatschappij *Elektuur* BV - 1986
Printed in the Netherlands. ISSN 0013-5895

De hieronder vermelde printen kunnen worden besteld via de handel en rechtstreeks bij *Elektuur* BV., Beek (L).

(E)PROM's kunt u door *Elektuur* BV. laten programmeren.

Uitsluitel over de bestelwijze en verkrijgbaarheid van onderstaande produkten geeft het overzicht in de laatste uitgave van het elektronicamaandblad *Elektuur* (pag. 6).

PRINT SERVICE

bestelnr.	guldens	Bfrs.	omschrijving
65028	39,60	780	BASICODE-interface

SOFTWARE

CP/M totaal	1
CP/M in 90 minuten	15
BBII: VARBIOS	26
BBII: de ROM-monitor	29
graphics-software	31
BASICODE-2	59
OCTOPUS-uitbreidingen	67
Forth — een inleiding	72

HARDWARE

C64-expansion-port	37
BBII: de hardware	41
IBM-K-FDC	56

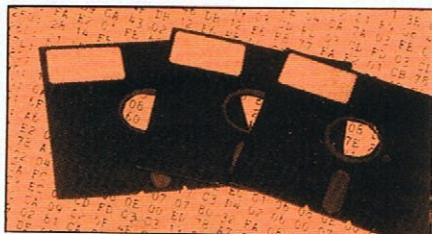
PROGRAMMA'S

block-graphics	74
DEHEBI	77

KNOW-HOW

8 of 16 bits?	79
de 68000 kort en bondig	80
6502-opvolgers	86

SOFTWARE



CP/M totaal

P. v. d. Linden

...Dat is het motto van deze tweede Elektuur-Computing. Deze keuze is niet verwonderlijk, want CP/M is het meest gebruikte operating-system ter wereld voor kleine computersystemen. Met klein bedoelen we hier: groter dan een home-computer (met tenminste één diskette-loopwerk), maar beslist geen "jumbo".

Die "middenmoot" omvat een grote groep machines. Men kan hier beslist niet zeggen dat CP/M in deze groep de alleenheerser is. Integendeel zelfs, sinds kort heeft CP/M een geduchte concurrent, namelijk MS-DOS. In dit artikel wordt niet nader ingegaan op het verschil tussen deze twee systemen. Over dit onderwerp zijn al genoeg publicaties verschenen, helaas maar weinige met een diepere technische achtergrond.

CP/M heeft zijn sterke en zwakke kanten. Het systeem is vaak vrij omslachtig in het gebruik en het is zelfs in de nieuwste versie, 3.0, nog steeds traag. Maar het grote voordeel blijft toch zijn grote bekendheid. Er is erg veel literatuur over CP/M verschenen en de hoeveelheid software die onder CP/M loopt, is enorm.

Ondanks zijn nadelen blijft CP/M een bijzonder aantrekkelijk operating-system, vooral door die ontzettend grote hoeveelheid beschikbare software. En dat zijn beslist niet alleen spelletjes. Want bij sommige populaire computers is het software-aanbod weliswaar heel groot, maar als je een serieuze toepassing zoekt, bijvoorbeeld een goede assembler, dan zit je vast. Dat probleem zal bij CP/M niet snel voorkomen, vooral ook omdat veel programma's afkomstig zijn uit de industrie.

Dat we in dit nummer niet alles over CP/M ter kunnen publiceren, zal duidelijk zijn. Ten eerste is het CP/M-systeem daarvoor veel te omvangrijk en ten tweede moeten we ook rekening houden met degene die in CP/M helemaal niet geïnteresseerd is. Voor die mensen brengen we in deze uitgave o.a. een BASICODE-interface voor de Octopus 65.

Uit het grote aanbod van CP/M-machines hebben wij er een geselecteerd die we van harte bij onze lezers kunnen aanbevelen. Konkreet gezegd, voor de lezers die zich met CP/M bezig willen gaan houden en zelf daarvoor een computer willen bouwen, hebben wij een goede machine

gezocht en gevonden. Die computer is de "Big-Board II". Deze machine is in de Verenigde Staten de meest verkochte single-board-CP/M-zelfbouw-computer. De ontwerper van deze computer, Jim Ferguson, heeft ook een flink aandeel gehad in de ontwikkeling van de Kaypro. Allereerst de belangrijkste technische specificaties van Big-Board II:

- Z80A-processor, 4MHz.
- Z80A-DMA met multiplexer.
- 24 Kbyte EPROM-bereik, 6 sockets waarvan er een voor de monitor-EPROM gereserveerd is. Ook statische RAM's kunnen hier worden geplaatst.
- Een geïntegreerde EPROM-programmer met software. De EPROM wordt op de hoofdprint in een van de EPROM-voetjes geprogrammeerd.
- 64-Kbyte dynamische RAM op de hoofdprint. Met externe RAM-banken kan dit uitgebreid worden tot 256 Kbyte of zelfs tot 1 Mbyte. (Dit laatste is alleen zinvol als men CP/M 3.0 op Big-Board II heeft geïnstalleerd.)
- krachtige monitor met 12 kommando's.
- SASI (Shugart Associates System Interface). Door een XEBEC controller aan te sluiten kunnen Winchester-loopwerken (hard-disks) van alle bekende leveranciers op Big-Board II worden aangesloten.
- floppy-disk-controller met 8877 en 9216D data-separator: 8"-, 5"- en 3 1/2"-loopwerken kunnen alle probleemloos, ook in een gemengde opstelling, gebruikt worden. Zelfs SD en DD zijn gelijktijdig mogelijk dankzij het CONFIG-programma van VARBIOS.
- Z80A-SIO. Twee seriële kanalen met modem-aansluiting en automatische baudrate-selektie.
- 2 Z80A-CTC's. Een daarvan is de baudrate-generator voor de SIO, de andere staat vrij ter beschikking van de gebruiker.
- ASCII-keyboard-aansluiting, naar keuze parallel of serieel.
- 4 parallele aansluitingen met handshake (twee ingangen en twee uitgangen).
- video-controller 6845S met een 8002-attribute-generator, 7 x 11 dot-matrix, en naar keuze een videofrequentie van 15750 Hz of 18600 Hz (VS-norm).

- aansluitbus voor verdere uitbreidingen.
- gaatjesprint-gedeelte voor eigen uitbreidingen van gebruiker.

Het zal wel duidelijk zijn dat het niet onze bedoeling was om met alle geweld een spektakulaire machine aan u voor te stellen, maar om uit het brede scala van CP/M-machines een degelijke en uitbreidbare basismachine te selecteren met een prijskaartje dat de lezer aanspreekt. De beslissing of en hoe men de machine eventueel gaat uitbreiden, ligt geheel bij de gebruiker zelf. Dit zal bijvoorbeeld afhangen van de behoefte aan bepaalde interfaces en de financiële draagkracht.

Een technische beperking van de basismachine is de grootte van het standaardgeheugen, slechts 64 Kbyte. Er wordt momenteel gewerkt aan een uitbreiding tot 256 Kbyte, die hoogstwaarschijnlijk in EC 3 zal worden gepubliceerd. Dan zijn er natuurlijk ook enthousiastelingen die zich beklagen over de klokkrequentie van "slechts" 4 MHz. Onze tip: vergelijk de praktische snelheid van een 4-, 6- en 8-MHz-CP/M-computer eens met hetzelfde programma. U zult er een stopwatch bij moeten pakken. Als men behalve de benchmark-tests ook nog een paar praktijkproeven doet (bijvoorbeeld met Wordstar een brief schrijven, veranderen en uitdraaien), dan wordt snel duidelijk dat alleen een hogere klokkrequentie lang niet zoveel snelheidswinst oplevert dan men zou verwachten. En als de processor ook nog moet werken met vrij langzame RAM's, hoeven we helemaal niets meer te verwachten van die hogere kloksnelheid. Tegenwoordig hoor je steeds vaker de term "CP/M 3.0-machine". In advertenties kom je allerlei merkwaardige verhalen tegen, waarin de indruk wordt gewekt dat het iets bijzonders is als een machine op CP/M 3.0 loopt. Eigenlijk is dat helemaal niet zo bijzonder, want CP/M 3.0 draait op elke CP/M-machine. Natuurlijk moet de machine zich wel aan de standaard-CP/M-konfiguratie houden (geheugenindeling e.d.). Bovendien heeft men een passende, systeemafhankelijke (en voor ieder computer specifieke) BIOS nodig. Dat is bij iedere CP/M-versie al zo geweest en is dus niets bijzonders.

Een heel andere vraag is natuurlijk, of het überhaupt wel zinvol is om CP/M 3.0 op een bestaand systeem te implementeren. Het antwoord hierop kan kort zijn. CP/M 3.0 gebruikt zoveel geheugenruimte dat het volkomen zinloos is op een 64-Kbyte-systeem aan implementeren te denken. Er is dan gewoonweg geen plaats meer voor een programma. Men moet minimaal 128 Kbyte aan geheugen hebben voordat men aan implementeren gaan denken. Pas bij 256 Kbyte is echt comfortabel werken mogelijk. Dit is dan ook de reden dat we over de 256-Kbyte-uitbreidingsprint voor Big-Board II begonnen zijn. Binnen een paar maanden hopen we hiermee klaar te zijn! De CP/M 3.0-versie van VARBIOS is al lang klaar.

Laten we de draad maar weer eens oppakken, we hadden het over het samenspel van hard- en software. Voor Big-Board II zijn in de USA twee zeer interessante hardware-uitbreidingen te koop, namelijk processor-kaarten met de 80186 van Intel en met de 68000 van Motorola. MS-DOS en Unix (Xenix, ...) kunnen we waarschijnlijk ook binnenkort verwachten. We moeten nu dus maar afwachten of deze kaarten op korte termijn ook in Europa verkrijgbaar zullen zijn. De VARBIOS van de Big-Board II kan in elk geval al werken met IBM-PC-diskettes.

Nu we het toch over diskettes hebben, bij het bouwpakket van Big-Board II hoort ook een diskette met software. Hierop staat namelijk een operating-system dat

CP/M-kompatibel is. Naar buiten toe tenminste, want in feite is het een nieuw operating-system. We hebben het over het door Richard Conn (USA) ontwikkelde CCP-alternatief genaamd ZCPR2, de door de Nederlandse firma Twente Digital ontwikkelde BIOS-ervanger VARBIOS en de eveneens door Twente Digital ontwikkelde BDOS-ervanger FDOS. Deze drie software-pakketten zijn "public-domain" en zijn bovendien verbeterde versies van de oorspronkelijke.

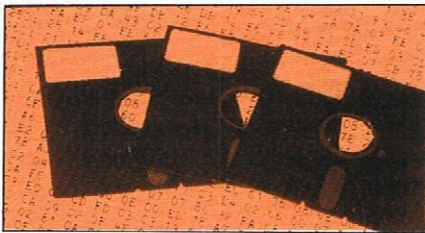
De systeem-diskette zit overigens behoorlijk vol, er staat ongeveer 520 Kbyte software op. Op deze schijf staat ook de complete gedocumenteerde source-code van VARBIOS en ZCPR2. En dat is nog lang niet alles. Die 520 Kbyte bevatten nog meer interessante dingen...

We hebben hier dus een gunstig geprijsde computer met kwaliteits-"public domain"-software. Een ding willen we nog hieraan toevoegen. De software zal door de officiële dealer als bestanddeel van het bouwpakket verkocht worden. Big-Board II is dus alleen compleet met de systeem-diskette te koop. Let wel even op, de software is enkel en alleen voor niet-kommerciële toepassingen public domain, dus bijvoorbeeld voor gebruik in de privé-sektor. Dat de software in het totale pakket meegeleverd wordt, is een voorwaarde geweest van Twente Digital, de ontwikkelaar van de programma's (VARBIOS en FDOS), om ze public domain te verklaren. In de pakketprijs is voor de

systeem-diskette zo'n 200...250 gulden gereserveerd. Daarvoor krijgt men echter wel bij de software een volledig gedocumenteerde source-listing van de systeemsoftware, dus van ZCPR2, VARBIOS en FDOS. Bedenk hierbij dat de source-code van CP/M in de Verenigde Staten in een kluis ligt die ongeveer net zo goed bewaakt wordt als Fort Knox. De Big-Board-II-nabouwer krijgt voor een relatief laag bedrag de compleet gedocumenteerde source-code van een CP/M-kompatibel systeem. Er zijn dus totaal geen geheimen meer voor de gebruiker. Het hele systeem is open. Het complete en uitvoerige manual van ZCPR2 is onder andere verkrijgbaar bij de:

CP/M gebruikersgroep Nederland
Ericalaan 23
2803 BN Gouda

De prijs is ongeveer 45 gulden. Als men dit manual eens goed bestudeert, dan zal snel duidelijk worden dat ZCPR2 niet domweg een vervanger voor de CP/M-CCP is, maar een veel krachtigere versie is van de oorspronkelijke CCP. CCP staat voor Console Command Processor, een interface tussen de computer aan de ene kant en de gebruiker aan de andere kant. Alle ingevoerde commandos worden immers door de CCP behandeld en dan aan de computer doorgegeven.



CP/M in 90 minuten

Loy's Nachtmann

Wil een gebruiker zinvol met CP/M werken, dan zal hij eerst een hoeveelheid boeken moeten bestuderen. Dit zullen door-gaans boeken in de Engelse taal zijn, voor sommigen een probleem. Hoe moet men nu beginnen met CP/M en is er een rode draad te vinden, zodat men er een beetje grip op krijgt? Jazeker, lees dit artikel maar! Neem Big-Board II of een andere CP/M-computer onder de arm, zoek een rustig plekje en werk dan alle oefeningen die in dit artikel staan door. Hebt u dat gedaan, dan kunt u al heel aardig overweg met CP/M.

Voordat men met CP/M kan werken, moet dit operating-system eerst vanaf een diskette in het computergeheugen geladen worden. Dit laden noemt men ook wel "booten", het opstarten van het systeem. Zet nu de computer aan en boot het operating-system. Bij Big-Board II moet

hiertoe de systeem-diskette in drive A aanwezig zijn. Na het aanslaan van een willekeurige toets start het laden van het operating-system. Op het beeldscherm zal na enige tijd "QA>" verschijnen, indien men een systeem-diskette van Twente Digital gebruikt, en "A" bij andere CP/M-systeem-diskettes. Gebruikt u geen Big-Board-II-computer, maar een ander CP/M-systeem, lees dan het handboek na, daar kunt u vinden op welke manier de machine boot. Zodra CP/M zich op het beeldscherm heeft gemeld, kunnen de eerste kommando's ingevoerd worden.

Een kort intermezzo

Alvorens we verder gaan met onze experimenten, is het beter eerst twee kopieën van de systeem-diskette te maken. Hier volgt daarvoor een handleiding:

1. Stop de systeem-diskette in loopwerk A en een lege diskette in loopwerk B. De diskette in loopwerk B moet eerst geformatteerd worden. Bij Big board II heet de formatteringsroutine "DDFORMAT5.COM". Wat de naam van de formatteringsroutine bij andere CP/M-systemen is, kan steeds

in het handboek van het desbetreffende systeem gevonden worden.

2. Het formatteringsprogramma kan nu gestart worden. Type hiertoe in:

```
QA>DDFORMAT5<cr>
```

Er worden nu twee vragen gesteld:

```
"ENTER DRIVE NO: (A - F)" en
"INSERT DISK TO BE FORMATTED INTO
DRIVE AND TYPE: (A,B,D,S)"
```

Wordt de eerste vraag met "B" en de tweede met "D" beantwoord, dan worden de vragen nogmaals gesteld voor de tweede lege diskette.

3. Nu gaan we alle data van de systeem-diskette op de geformatteerde diskette in loopwerk "B" kopiëren.

```
QA>PIP B:=A:**<cr>
```

De loopwerken worden nu geactiveerd en aan het gezoem van de stappenmotor en de eventuele head-load-magneet is te horen dat er gewerkt wordt. Op het beeldscherm is de activiteit van het "PIPCOM" goed te volgen, omdat alle

werkzaamheden worden weergegeven. Herhaal dit voor de tweede diskette.

4. Nadat alle data van de systeemdiskette in loopwerk "A" op de diskette in loopwerk "B" gekopieerd zijn, moeten we het CP/M-systeem nog op de diskette kopiëren. Bij Big-Board II heet het systeemkopieerprogramma "VARSYSG.COM" en bij de andere CP/M-systemen "SYSGEN.COM". Bij "VARSYSG.COM" kunnen 5 1/4"- en 8"- loopwerken gelijktijdig aangestuurd worden. Bij "SYSGEN.COM" kan maar één soort loopwerk worden gebruikt.

```
QA>VARSYSG<cr>
```

Dit systeem-kopieercommando stelt twee vragen:

```
"ENTER SOURCE DRIVE, OR (CR) TO
SKIP" en
"ENTER DESTINATION DRIVE OR (CR)
TO SKIP"
```

Beantwoord de eerste vraag met "A" en de tweede met "B". Herhaal dit ook voor de tweede diskette. We gaan er van uit dat men een computer met twee loopwerken heeft. Einde intermezzo.

De ingebouwde CP/M-kommando's

CP/M kent twee soorten kommando's: namelijk ingebouwde en transient-kommando's. Ingebouwde kommando's zijn, zoals de naam al zegt, in CP/M ingebakken. Deze kommando's zijn:

1. DIR
2. ERA
3. REN
4. TYPE
5. SAVE
6. USER

Ingebouwde kommando's kunnen direkt uitgevoerd worden. Dit wil dus zeggen dat er geen extra programma van de diskette in het geheugen hoeft te worden geladen als zo'n kommando moet worden uitgevoerd.

Bij transient-kommando's is dat anders. Voordat CP/M een transient-kommando kan uitvoeren, moet het eerst het transient-programma van diskette in het geheugen kopiëren. Staat dit eenmaal in het geheugen, dan wordt het automatisch uitgevoerd. Transient-kommando's staan als een file met een naam op de diskette. CP/M leest een transient-programma altijd in het geheugen vanaf het adres 100H en start dit programma door een sprong naar dit adres. Transient-programma's zijn in de inhoudsopgave van de diskette (directory) eenvoudig terug te vinden doordat er achter de naam "COM" staat. Enkele voorbeelden zijn: SYSGEN.COM en LOADCOM.

Door het uitvoeren van deze transient-kommando's kan men de computer eenvoudig veranderen in bijvoorbeeld een BASIC-, Forth- of Pascal-computer. Het is ook mogelijk om zelf transient-kommando's te schrijven. Stel u wilt bijvoorbeeld het scherm schoonmaken. Men kan dan met behulp van een assembler of een hogere programmeertaal

1

```
QA>DIR
A: RI      MAC : STAT      COM : DDT      COM : WS      COM
A: WSMGSGS OVR : WSOVLY1  OVR : MERGPRIN OVR : L80      COM
A: TTT      COM : VARSYSG  MAC : CONFIG  MAC : DDFORMT5 MAC
A: WSYSGEN  MAC : WINFORMT MAC : VARBIOS  REL : VARBIOS  HEX
A: VONESECT LIB : MYDEBLOK LIB : ZMACRO    LIB : WINCONTR LIB
A: SELECT  LIB : MEMDEF   LIB : MXEBEC    LIB : VARBIOS  MAC
A: COLDINIT BAK : CPM59   COM : CPM58    COM : SETUP   BAK
A: NMISYSTM COM : NMI      : CRT      COM : SYSTEM-1 COM
A: NMI      BAK : VARSYSG  COM : NMI      MAC : DDFORMT5 COM
A: ZSID     COM : CONFIG  COM : SYSTEEM  COM : SETUP   LIB
A: NMI      PRN : COLDINIT LIB : NMI      REL : NMI      HEX
A: LOAD     COM : NMI      COM : CRT      BAK : M80      COM
A: CRT      PRN : CRT      REL : CRT      MAC : MBASIC  COM
A: BEMANUAL LST : BM      MAC : VARBIOS  BAK : CPMMAC  MAC
QA)
```

zoals BASIC of PASCAL een programma schrijven dat het scherm schoonmaakt. Het programma gaat nu CLEAR.COM heten. Nadat het programma geassembleerd of gecompileerd is, afhankelijk van de source taal, wordt deze source bewaard. Het transient-source programma is nu altijd terug te vinden, indien men later nog eens wat wil veranderen. Het CLEAR.COM kommando is nu uit te voeren met:

```
QA>CLEAR<cr>
```

Het achtervoegsel COM hoeft men niet op te geven, omdat CP/M automatisch weet dat het om een transient-programma gaat, dat naar 100H geladen moet worden om vervolgens uitgevoerd te worden. Het verschil tussen de beide soorten kommando's zal nu duidelijk zijn. Voor CP/M is er een vrijwel oneindige hoeveelheid transient-kommando's, de navolgende standaard-kommando's zijn op vrijwel iedere CP/M-systeem-diskette terug te vinden:

- * ASM of M80 (een 8080 of Z80-assembler)
- * DDT, SID of ZSID (een debugger)
- * DUMP (drukt een file in hex-formaat af)
- * ED of WS (een tekstverwerker)
- * LOAD of L80 (een loader)
- * PIP (data-transfer-programma)
- * MOVECPM, CPM59 of CPM58 (CP/M-konfiguratieprogramma)
- * STAT (geeft status weer van files, loopwerken en I/O)
- * SUBMIT of XSUB (voert meerdere CP/M-kommando's in één file uit)
- * FORMAT (formateert een diskette, DDFORMT5 bij BB-II)
- * SYSGEN (kopieert het CP/M-operatiesysteem op een diskette, VARSYSG bij BB-II)

Belangrijke tekens bij CP/M

Bij het ingeven van kommando's kan men wel eens een foutje maken. CP/M kent veel kommando's waarmee zo'n fout kan worden gecorrigeerd. Enkele van de gewenste bevelen zijn het starten van dataverzending naar de printer of het onderbreken van een programma. Hier volgt de beschrijving van een aantal kommando's:

Figuur 1. Zo ziet de inhoudsopgave van een CP/M-diskette er uit. Deze inhoud bestaat uit een file-naam en een file-type die door een punt van elkaar gescheiden zijn. De opgave van het file-type is optioneel.

<cr> heeft tot gevolg dat CP/M het kommando in het input-buffer gaat uitvoeren. Er wordt een foutmelding gegeven als CP/M het kommando niet kan uitvoeren. Dit kan gebeuren als men bijvoorbeeld een typefout maakt of als er geen diskette in de drive zit.

<C> onderbreekt een proces dat CP/M aan het uitvoeren is. Daarna wordt een warme start uitgevoerd.

Attentie: Steeds als een diskette in een loopwerk verwisseld wordt, moet ↑C na de CP/M-prompt "A>" of "QA>" gegeven worden.

De directory van de diskettes in de drives wordt na iedere ↑C gelezen. Weigert CP/M op een diskette te schrijven, toets dan <↑C> in.

<IH> verplaatst de cursor een plaats naar links en wist het karakter onder de cursor. <IH> is meestal als een speciale toets ter beschikking, bijvoorbeeld als BS of back-space.

<IP> schakelt de data-uitvoer naar de printer in. De printer-vlag, een software-flipflop, blijft geset tot er voor de tweede keer <IP> gegeven wordt. Dit kommando kan op elk willekeurig moment bij het invoeren van CP/M-kommando's gegeven worden.

<IX> zet de cursor aan het begin van de regel en wist alle tot dan toe ingevoerde kommando's.

De ingebouwde CP/M-kommando's

Met behulp van het DIR-kommando is het mogelijk de hele inhoud van een diskette op beeldscherm of printer te tonen. Figuur 1 geeft zo'n directory van een Big-Board-II-diskette weer. Zoals de directory al laat zien, hebben alle CP/M-files een naam die maximaal acht letters lang is.

Achter de file-naam bevindt zich een type-aanduiding die maximaal drie letters lang is. Aan deze type-aanduiding kan men zien of het om een BASIC-, Pascal- of assembler-file gaat. Bij CP/M zijn de navolgende file-typen gebruikelijk:

ASM is een assembler-source-file. De gebruikte assembler is een 8080-assembler van Digital Research. Voorbeelden: BB2.ASM en ROM.ASM.

MAC is ook een assembler-source-file, maar hier is een M80-assembler van Microsoft gebruikt. M80 kan zowel Z80 als 8080-mnenomics assemblers. Voorbeelden van MAC-files zijn: EDITOR.MAC en MONITOR.MAC.

BAS is een BASIC-source-file. De BASIC-interpretator of -compiler van Microsoft kan met deze files werken. Voorbeelden van BAS-files: LOON.BAS en BELAST.BAS.

BAK is een back-up-file. De CP/M-editor "ED" en de tekstverwerker Wordstar "WS" genereren automatisch zulke file-types. Gaat men met Wordstar een file bewerken, dan maakt CP/M een kopie van deze file, de zogenaamde back-up-file. Men kan dan zoveel veranderen als men wil in de file, terwijl toch steeds nog de oude file beschikbaar is als men niet tevreden is over de ingebrachte veranderingen. Voorbeeld van een BAK-file:

TEXT.LIB wordt TEXT.BAK

COM is een zogenaamde kommando-file. Zoals al eerder vermeld, worden deze files bij het oproepen van een transient-kommando in het geheugen geladen en uitgevoerd. Voorbeelden van COM-files: PIPCOM en LOADCOM.

FOR-files zijn Fortran-files.

HEX-files zijn files in hex-dataformaat. Deze files werken met het Intel-hex-dataformaat en kunnen onder andere door de 8080-assembler van Digital Research of de M80-assembler van Microsoft worden gemaakt.

PAS is de aanduiding voor een Pascal-file.

REL-files zijn relatieve files. De Microsoft macro-assembler M80 zet source-files van het type "MAC" om in relatieve files, dus van het type "REL". Op de M80-assembler zal in een volgende Computing nog aandacht worden besteed.

SUB betekent submit-file. Deze files bestaan meestal uit meerdere CP/M-kommando's. De computer leest deze file in en voert vervolgens de kommando's sekventieel uit. Anders gezegd: de computer krijgt nu geen kommando's van het toetsenbord, maar leest deze uit een file in. SUB-files gebruiken de kommando-files SUBMITCOM en XSUBCOM.

TXT-files zijn text-files. Deze files bevatten enkel en alleen ASCII-karakters.

\$\$\$-files zijn hulp-files waarin kortstondig informatie wordt opgeslagen bij het maken van een tekst-file. De CP/M-editor "ED" en de tekstverwerker "Wordstar"

2

```

0B>DIR *.COM
B: ZSID      COM : ASCII  COM : HEX    COM : CRT    COM
B: VARSYSG   COM : TTT    COM : STAT   COM : WS     COM
0B>

```

gebruiken hulp-files als "kladblok". Als men de tekstverwerker verlaat, worden de hulp-files weer gewist. Doorgaans zal men deze files dus niet in de directory aantreffen.

DIR(ectory)

Met dit kommando is het mogelijk de inhoud van een diskette op beeldscherm of printer te zetten. We geven nu een paar voorbeelden, zodat men de diverse mogelijkheden van het DIR-kommando leert kennen.

```
0A>DIR<cr>
```

De inhoudsopgave wordt, zoals figuur 1 laat zien, op het beeldscherm weergegeven. De inhoudsopgave is van de op dat moment geselecteerde drive "A".

```
0A>DIR B:<cr>
```

De inhoud van loopwerk "B" wordt nu weergegeven. Drive "A" blijft het geselecteerde loopwerk.

```
0A>B:<cr>
0B>DIR<cr>
```

Drive "B" is vanaf nu het geselecteerde loopwerk. De CP/M-melding "0B>" laat dat aan de gebruiker weten. Als men nu <1C> op het toetsenbord intikt, wordt CP/M "warm" gestart. De CCP (Command Console Processor) wordt nu vanaf de diskette in het geheugen gekopieerd. Staat op de diskette in loopwerk "B" geen CP/M-operating-system, dan zal de foutmelding "CAN'T BOOT CP/M" na korte tijd op het beeldscherm verschijnen. Met behulp van het transient-kommando "SYSGEN" (of "VARSYSG" bij BB-II) kan men alsnog het CP/M-operating-system op de diskette kopiëren.

```
0B>A:<cr>
0A>DIR *COM<cr>
```

Drive "A" wordt nu weer het geselecteerde loopwerk. Alle files van het type COM worden op het beeldscherm getoond. Zoals figuur 2 laat zien, is het mogelijk om met behulp van een "*" een hele subgroep van file-namen op te roepen.

```
0A>DIR VAR?????*<cr>
```

Alle files waarvan de naam met "VAR" begint, worden op het beeldscherm zichtbaar gemaakt. De "*" vertegenwoordigt ieder file-type, terwijl elke "?" een willekeurig ASCII-karakter voorstelt. Het zal nu duidelijk zijn dat:

```
DIR<cr>
DIR **
DIR ??????????
dezelfde opdrachten zijn.
```

Figuur 2. Met het "DIR"-kommando kan een deel van de files worden afgedrukt. Hier worden alle "COM"-files van de in loopwerk "B" aanwezige diskette getoond.

```
0A>DIR *COM<1P><cr>
```

Alle files van het type "COM" worden nu op de printer afgedrukt. "1P" schakelt de printer in en kan op een willekeurige plaats in een kommandoregel staan. Pas wanneer voor een tweede keer "1P" ingegeven wordt, zal er geen data meer naar de printer verzonden worden.

ERA(se)

Met het ERA-kommando kan men files van de diskette verwijderen. ERA mag gevolgd worden door dezelfde tekens als bij DIR. ERA wordt voornamelijk gebruikt om oude hulp-files van bijvoorbeeld een assembler of tekstverwerker te wissen. Dank zij dit kommando kunnen dus files die niet meer nodig zijn van de diskette verwijderd worden, zodat er weer plaats vrij komt.

```
0A>ERA CRTCOM<cr>
```

Nu wordt van het momentele loopwerk "A" de file CRTCOM gewist.

```
0A>ERA B:*BAK<cr>
```

Dit kommando zorgt ervoor dat van loopwerk "B" alle files van het type "BAK" gewist worden, loopwerk "A" blijft de geselecteerde drive. We herinneren ons toch nog dat BAK-files worden gebruikt door de CP/M-editor of Wordstar?

```
0A>ERA B:**<cr>
```

Op deze manier kunnen we alle files van de diskette in loopwerk "B" wissen. Voor dat CP/M dit kommando uitvoert, wordt voor de zekerheid toch nog maar even gevraagd of alle files echt mogen verdwijnen. Hiertoe krijgt men de vraag "ALL FILES Y/N?". Wordt deze vraag met Y(es) beantwoord, dan zal CP/M de opdracht uitvoeren.

REN(ame)

Met het REN-kommando kan de naam van een file veranderd worden. Anders gezegd: de naam van een file wordt door een andere naam vervangen.

```
REN          NIEUWE-
NA.TYP=OUDENAAM.TYP<cr>
```

Net zoals bij de 8080/Z80-assembler worden CP/M-kommando's van rechts naar links gelezen. Het gelijk-teken dat tussen beide file-namen staat, kan men zien als

een naar links wijzende pijl. Hierdoor wordt dit kommando voor de 6502-, 6809- en 68000-gebruikers wat overzichtelijker, omdat zij altijd van links naar rechts lezen. Een voorbeeld:

```
QA>REN
CRT.COM=VIDEO.COM<cr>
```

Vanaf nu zal VIDEO.COM CRT.COM gaan heten. Als men een file van het type COM een andere naam gaat geven, moet men zich goed realiseren dat de type-aanduiding COM moet blijven bestaan, wil men deze file als kommando-file blijven gebruiken.

TYPE

Met het kommando TYPE kan men de inhoud van een ASCII-file bekijken via het beeldscherm of de printer. De data worden bloksgewijs vanaf de diskette naar de geselecteerde output-kanalen gezonden. Met "tS" (stopteken) kan men de tekstweergave onderbreken. Geeft men dit kommando nogmaals, dan gaat de computer weer verder met de verzending van de data. Met "tC" kan het TYPE-kommando worden beëindigd. De computer komt nu weer in de kommando-mode. Een voorbeeld:

```
QA>TYPE CRT.MAC<cr>
QA>TYPE B:BB2.TXT*P<cr>
QA>tP
```

De eerste kommandoregel zorgt er voor dat de datafile CRT.MAC van loopwerk "A" op het beeldscherm wordt weergegeven. In de tweede kommandoregel

wordt de datafile BB2.TXT van loopwerk "B" op de printer afgedrukt. Het laatste kommando zorgt ervoor dat de printer weer uitgeschakeld wordt.

SAVE

Met dit kommando is het mogelijk om een machinetaalprogramma of een andere file op de diskette te save onder een zelfgekozen naam. CP/M gaat er van uit dat de hele file in het computergeheugen staat en begint op het adres 100H. De algemene vorm van SAVE is:

```
SAVE p NAAM.COM<cr>
```

In deze vorm geeft p het aantal blokken van 256 bytes aan die gesaved moeten worden. Door het achtervoegsel COM weet CP/M dat het om een kommando-file gaat. Uiteraard zijn ook andere filetypen toegestaan. Het is ook heel eenvoudig om een lege (dummy) file te maken:

```
SAVE 0 PETER.TXT
```

Op deze manier wordt in de directory de file-naam gezet, maar er wordt geen ruimte voor data gereserveerd.

USER

Het USER-kommando is zeer nuttig als men CP/M gebruikt in een multi-user-systeem. MP/M kan namelijk in tegenstelling tot CP/M meerdere terminals gelijktijdig bedienen (timesharing). Iedere gebruiker heeft in dit geval een eigen nummer (tussen 0 en 15), zijn eigen directory en geheugenbereik in de computer.

Vanaf de CP/M-versie 2.2 is het USER-kommando ook in CP/M geïmplementeerd. Hierdoor is het mogelijk de diskette wat overzichtelijker in te delen.

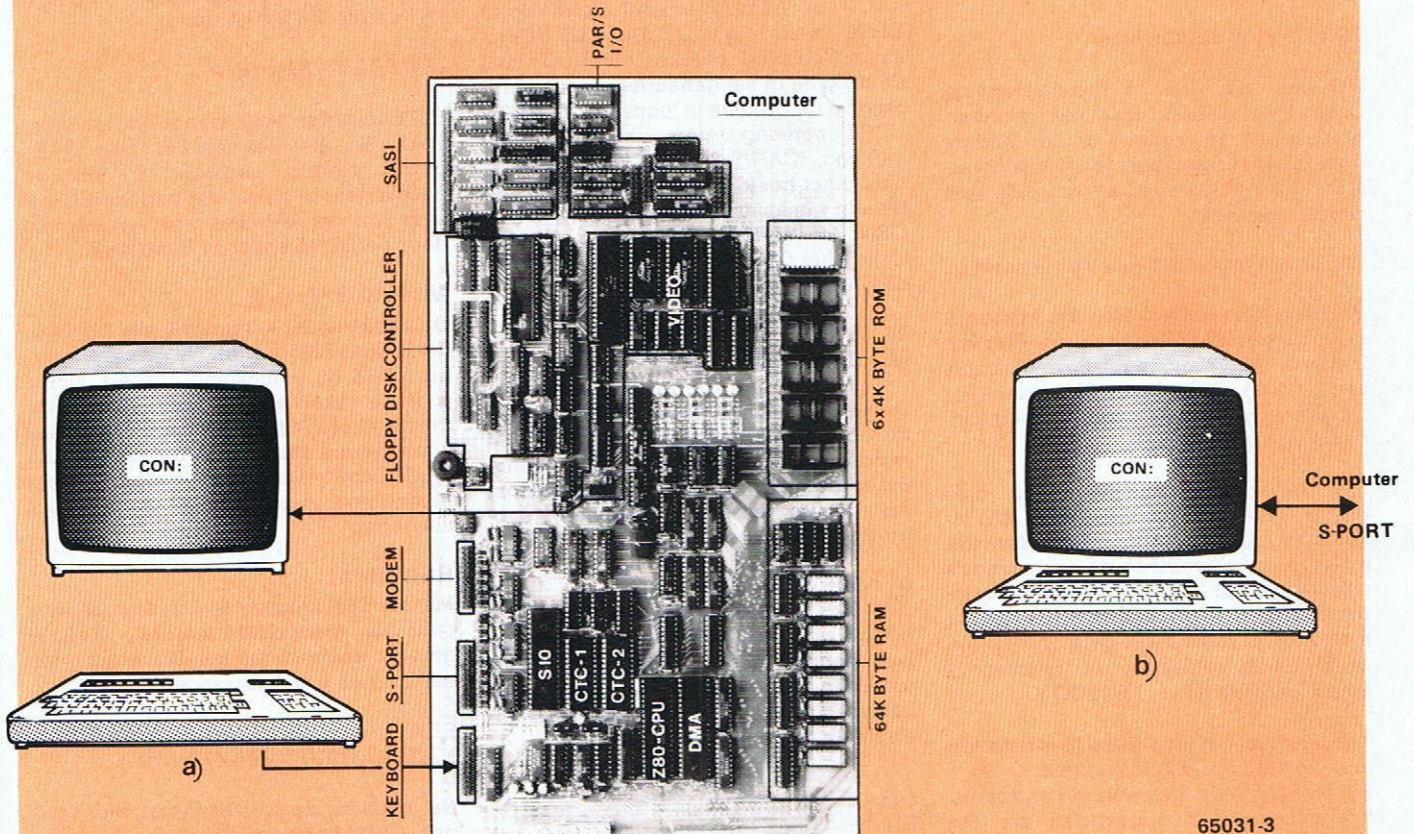
Big Board II kan op twee loopwerken in totaal 1,6 miljoen bytes bewaren. Het is daardoor mogelijk om naast data op de systeem-diskette ook nog een Pascal-compiler, een BASIC-interpretter, een tekstverwerker en een assembler op een schijf te zetten. Met het USER-kommando kan men aan elk user-nummer een eigen transient-processor toekennen. USER-0 is dan bijvoorbeeld een BASIC-systeem, compleet met BASIC-interpretter en BASIC-files. USER-1 kan dan de PASCAL-compiler met PASCAL-files bevatten. De assembler en debugger kunnen in USER-3 bewaard worden. Iedere USER heeft nu zijn eigen directory en is strikt gescheiden van de andere.

In VARI BIOS van Big-Board II is het USER-kommando net zoals bij MP/M geïmplementeerd. Heeft men een gewone CP/M op de computer, dan moet men het transient-kommando "STAT" gebruiken om de momenteel geselecteerde USER-nummers te vinden (zie hiervoor ook het STAT-kommando). Voorbeeld:

```
QA>USER 3<cr>
3A>
```

De computer werkt in het USER-0-gebied. Met het USER-kommando is het mogelijk over te gaan naar het USER-3-werkgebied. Door de terugmelding "3A" geeft de computer aan dat men nu in een ander USER-gebied zit.

3



65031-3

Figuur 3. Bij alle CP/M-computers is een console (terminal) als I/O-device noodzakelijk. Bij Big-Board II is deze console in de computer geïntegreerd (figuur 3a). In figuur 3b zien we een console die door middel van een seriële interface met de computer verbonden is. Welke console door de gebruiker gebruikt wordt, is voor CP/M onbelangrijk, omdat de BIOS de in- en output van de computer regelt.

Figuur 4. "Leest" CP/M het kommando "LST" in een kommandoregel, dan wordt de "LiSting Device", de printer, aangestuurd.

Figuur 5. De file "CRT.MAC" wordt op de console weergegeven. Dat weergegeven wordt door het "PIP"-kommando geregeld. Om te beginnen leest CP/M "PIP.COM" in het geheugen en start dit. Daarna wordt de file "CRT.MAC" in blokken via het geheugen naar het beeldscherm verzonden. Is dit klusje geklaard, dan keert CP/M weer in de kommando-ingave-mode terug. Het kommando "TYPE CRT.MAC" is identiek aan "PIP CON:=CRT.MAC".

Drie belangrijke transient-kommando's:

PIP
STAT
DDFORMT5

Algemeen:

Op een CP/M-computer kunnen meerdere rand-apparaten (I/O-devices) worden aangesloten. Globaal zijn deze in twee verschillende groepen onder te verdelen:

- 1 ASCII-georiënteerde
- 2 file-georiënteerde

ASCII-georiënteerde rand-apparaten verwerken de aangeboden data teken voor teken, voorbeelden hiervan zijn printers, monitors, barcode-lezers en toetsenborden. File-georiënteerde rand-apparaten verwerken de data blokgewijs, dus niet per karakter. Voorbeelden hiervan zijn massageheugens zoals floppy-drives, RAM-disks en bubble-memories.

In de praktijk zijn op een CP/M-computer meestal een monitor en een toetsenbord aangesloten. CP/M kent aan ieder aangesloten apparaat een naam toe. De gebruiker kan zelf de namen van de aangesloten rand-apparatuur veranderen, zie hiervoor het "STAT"-kommando. Normaliter noemt men toetsenborden en monitoren "CONSOLE" (CON:). Een console is een I/O-apparaat dat er bij iedere computer weer anders kan uitzien. Figuur 9 laat zien hoe de console van Big Board II, en de meeste andere CP/M computers, er uit ziet.

Het toetsenbord heeft een parallel-

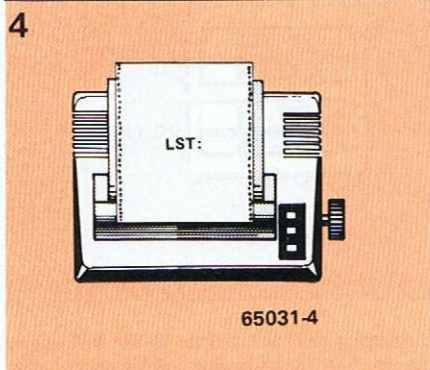
aansluiting op de computer. De ASCII-tekenen worden gegenereerd door de keyboard-encoder, die in de kast van het toetsenbord is ingebouwd. De CRT-controller en de software die nodig is om het beeldscherm aan te sturen, zijn alle op de grote moederprint van de computer ondergebracht. Door middel van een videokabel is de CP/M-computer met de monitor verbonden.

Bij een multi-user-systeem zijn de diverse terminals (consoles) door middel van een seriële RS-232/V24-verbinding aan de computer gekoppeld. De keyboard-encoder, de CRT-controller, het videoschermgeheugen en het schuifregister voor de seriële interface zijn alle in de terminal ondergebracht.

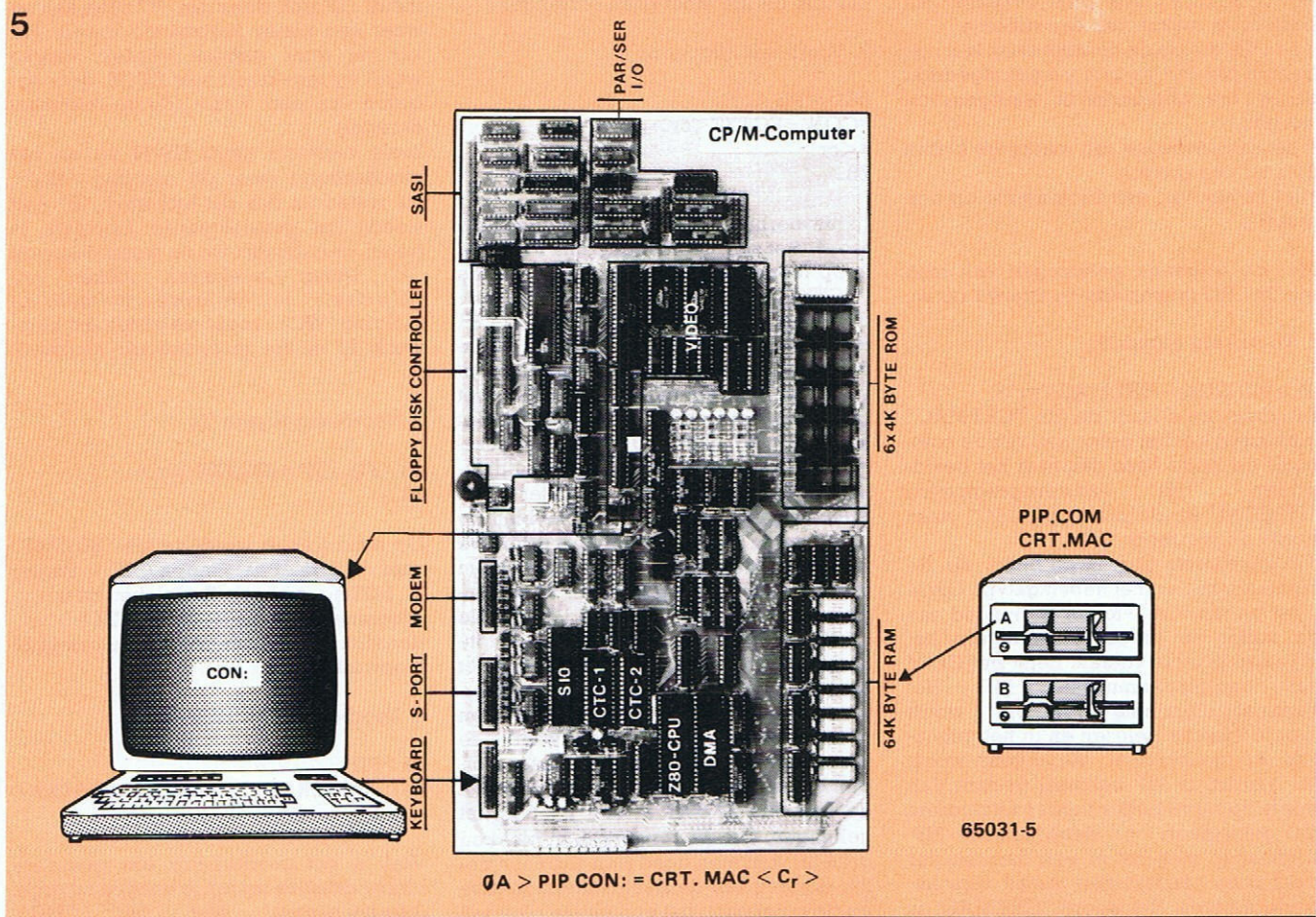
Ook de printer die met de CP/M-computer verbonden is, heeft een naam: "list device" (LST: zie figuur 7). Big-Board II heeft voor de printer een Centronics-interface.

PIP

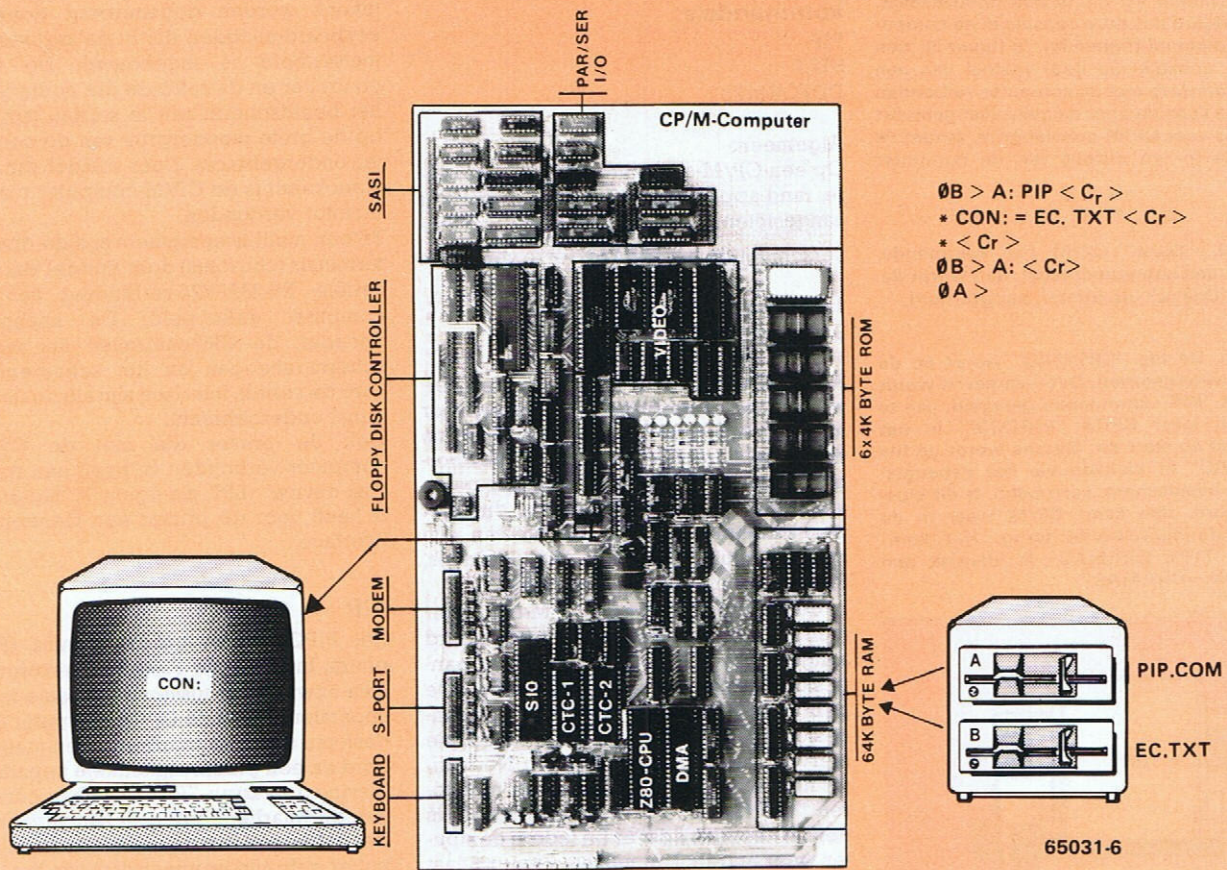
Het PIPCOM-transient-programma (Peripheral Interchange Program) verzorgt de data-uitwisseling tussen de diverse rand-apparaten die op de CP/M-computer aangesloten zijn. Een loopwerk, een toetsenbord en een printer zijn enkele apparaten (devices) waarvan "PIP" data kan ontvangen en waarheen PIP data kan versturen. Het geheugen van de computer wordt als buffer gebruikt wanneer data van het ene naar het andere rand-apparaat verzonden worden. De belangrijkste mogelijkheden die "PIP" aan de gebruiker van een CP/M-computer biedt, zijn:



65031-4



65031-5



- * het kopiëren van een of meerdere files van het ene loopwerk naar de andere. Het is natuurlijk ook altijd mogelijk een back-up te maken van een diskette.
- * een file die op de diskette staat, kan met behulp van PIP op een printer afgedrukt of op het beeldscherm weergegeven worden.
- * het samenvoegen van meerdere kleine files tot een grotere.
- * het maken van een back-up-file op een diskette.

Met de volgende oefeningen kan men met het PIP-kommando vertrouwd raken.

1. Voorbeeld (figuur 5):

ØA>PIP CON:=CRT.MAC<cr>
Dit kommando laadt de file "CRT.MAC" bloksgewijs in het geheugen van de computer en stuurt deze dan naar het beeldscherm. Het transient-programma "PIPCOM" en de file "CRT.MAC" staan beide op loopwerk "A".
Het kommando staat in de buffer van de computer zoals u het hebt ingetypt. CP/M leest na het <cr>-teken de inhoud van de buffer en probeert vervolgens het kommando uit te voeren. Eerst ziet CP/M het transient-kommando PIP. Het transient-programma "PIPCOM" wordt van de diskette gelezen en in het geheugen weggeschreven. Vervolgens wordt het gestart. Is PIP eenmaal gestart, dan wordt de input-buffer verder uitgelezen. Het gelijkteken "=" deelt aan het PIP-kommando mee dat er een data-transfer moet gaan plaatsvinden tussen verschillende devices. Het woord "CRT.MAC" is

de naam van de file die ingelezen moet worden en CON: is de naam van het apparaat dat de file moet gaan ontvangen.

2. Voorbeeld (figuur 6):

```
ØB>A:PIP<cr>
*CON:=ECTXT<cr>
*<cr>
ØB>
```

In dit voorbeeld is loopwerk "B" geselecteerd. Het transient-programma PIPCOM is op loopwerk "A" aanwezig, en de file "ECTXT", die naar een console verzonden moet worden, staat op loopwerk "B". Het kommando in de eerste regel selecteert loopwerk "A" en leest van dit loopwerk het transient-programma PIP in het geheugen, waarna het wordt gestart. Het loopwerk "B" blijft nog steeds het geselecteerde loopwerk. "A" wordt maar tijdelijk gebruikt. Omdat achter het PIP-kommando verder niets staat blijft de computer in het PIP programma zitten. De "*" maakt duidelijk dat PIP klaar is om van de gebruiker opdrachten te ontvangen. In de tweede regel wordt de file "ECTXT" naar het beeldscherm gezonden. Omdat loopwerk "B" nog steeds het geselecteerde loopwerk is, worden hiervan steeds blokken van de file ingelezen in het geheugen opgeslagen, waarna ze vervolgens op de monitor zichtbaar worden gemaakt. De computer leest dus niet de hele file ECTXT in een keer in het geheugen. Dat dit verwerken in blokken gebeurt, kan men goed zien aan het tijdelijk stoppen van de beeldscherm-uitgave op de momenten dat een nieuw blok van

de floppy gelezen wordt. Is het kommando in het input-buffer uitgevoerd, dan meldt PIP zich weer met "*". Men kan nu weer een nieuw kommando geven. PIP kan nu weer verlaten worden, waarna wordt teruggekeerd naar CP/M, door het geven van een <cr> (derde kommandoregel).

Zoals verwacht geeft CP/M nu op het beeldscherm weer de melding "ØB>", we weten nu dus dat loopwerk "B" nog steeds het geselecteerde loopwerk is (regel vier). Omdat we nu weer loopwerk "A" willen selecteren, tikken we "A:=<cr>" in. We krijgen nu weer de melding "ØA>", zodat we weten dat loopwerk "A" nu het geselecteerde loopwerk is.

3. Voorbeeld (figuur 7):

```
ØA>PIP LST:=BB2.TXT<cr>
ØA>
```

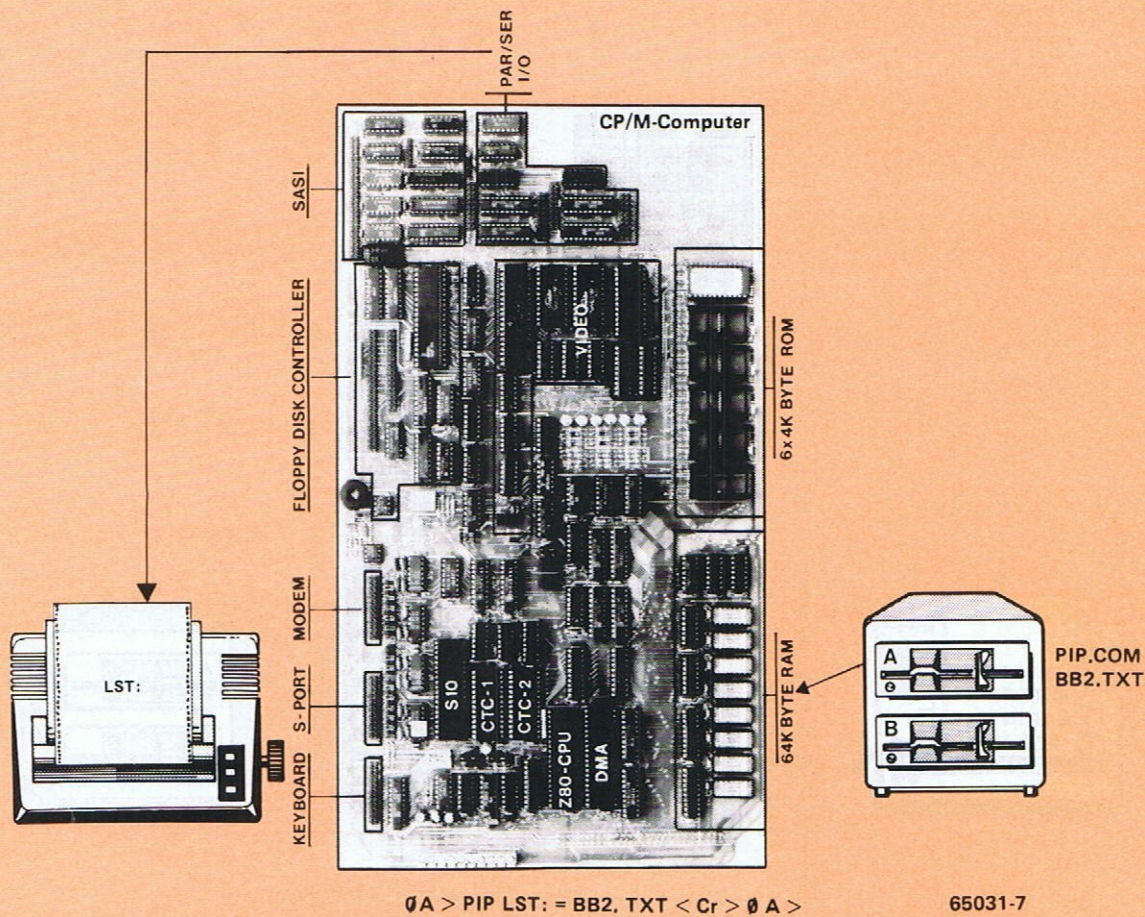
Op deze manier wordt de file "BB2.TXT" naar de printer verzonden (LST:= Printer = LiSTing device). Zowel het transient-programma PIP als de file BB2.TXT bevinden zich in het momenteel geselecteerde loopwerk "A".

4. Voorbeeld (figuren 8 en 11):

```
ØA>PIP
LST:=PETER.TXT,KARIN.TXT,NICOT.TXT
<cr>
```

Vaak is het noodzakelijk om meerdere kleine datafiles samen te voegen tot grote (concatenation). Met het "PIP"-

7



kommando kan men dit op een eenvoudige manier doen. De tekst-files "PETER.TXT", "KARIN.TXT" en "NICO.TXT" staan allemaal op de diskette in loopwerk "B". Deze drie files worden naar de printer gezonden en als een complete file afgedrukt. Het is echter ook mogelijk meerdere kleine files aan elkaar te knopen en als een nieuwe file op de diskette op te slaan. De files mogen op verschillende loopwerken staan:

```
>A>PIP
B:TOTAL.TXT=A:FILE1.TXT;C:FILE2.TXT,
B:FILE3.TXT<cr>
>A>
```

In dit voorbeeld wordt het transient-programma "PIPCOM" van het momenteel geselecteerde loopwerk "A" in het geheugen geladen. De drie data-files "FILE1.TXT...FILE3.TXT" bevinden zich op verschillende loopwerken. "FILE1.TXT" staat op loopwerk "A", "FILE2.TXT" staat op loopwerk "C" en "FILE3.TXT" staat op loopwerk "B". Zij worden nu tot een nieuwe file "TOTAL.TXT" op loopwerk "B" gekombineerd. Na het uitvoeren van dit kommando blijft loopwerk "A" het geselecteerde loopwerk.

5. Voorbeeld (figuur 9):

```
>A>PIP
COPYCRT.MAC=CRT.MAC<cr>1
>A>
```

In dit voorbeeld wordt een kopie gemaakt van een file. Het transient-

programma "PIPCOM" en de file staan beide op het geselecteerde loopwerk "A". De kopie van de file "CRT.MAC" heet "COPYCRT.MAC". Ook deze file komt op loopwerk "A" te staan.

Wil men echter "COPYCRT.MAC" op loopwerk "B" krijgen, dan zal de kommandoregel er iets anders moeten uitzien:

```
>A>PIP
COPYCRT.MAC=CRT.MAC<cr>
>A>PIP B:=A:COPYCRT.MAC<cr>
>A>
of
>A>PIP
B:COPYCRT.MAC=A:CRT.MAC<cr>
of
>A>PIP
B:COPYCRT.MAC=CRT.MAC<cr>
```

In het eerste voorbeeld wordt de file "COPYCRT.MAC" zowel op loopwerk "A" als loopwerk "B" geschreven. In de beide andere voorbeelden wordt de file alleen op loopwerk "B" geschreven.

6. Voorbeeld (figuur 10):

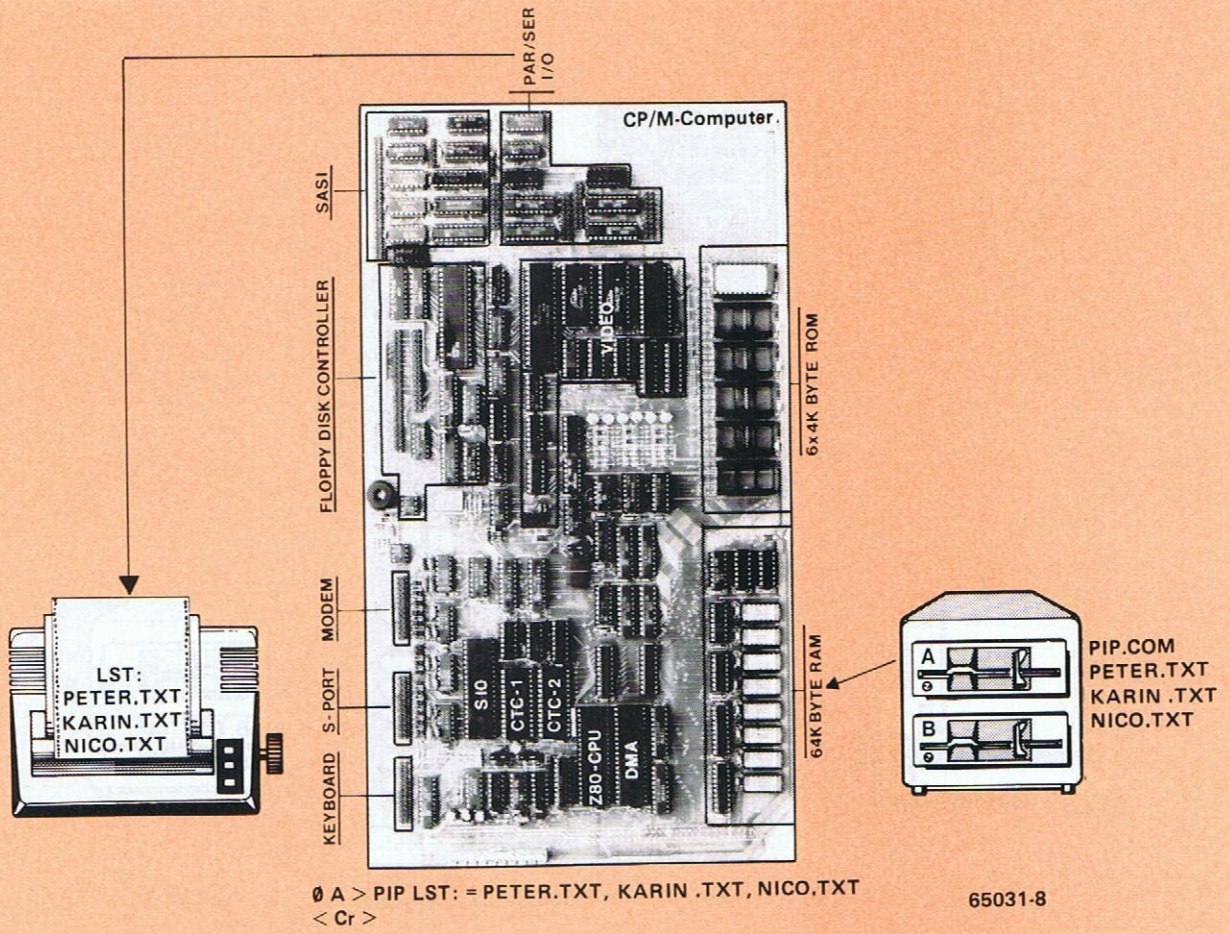
```
>A>PIP B:=*COM[V]<cr>
```

Het "PIP"-kommando kan ook groepen files van de ene diskette naar de andere kopiëren. In het voorbeeld kopiëren we alle files van het "COM"-type van het momenteel geselecteerde loopwerk "A" naar loopwerk "B". Binnen de rechte haakjes staat de parameter "V" in de kommando-buffer (V=VERIFY). Een of meerdere parameters kunnen aan het

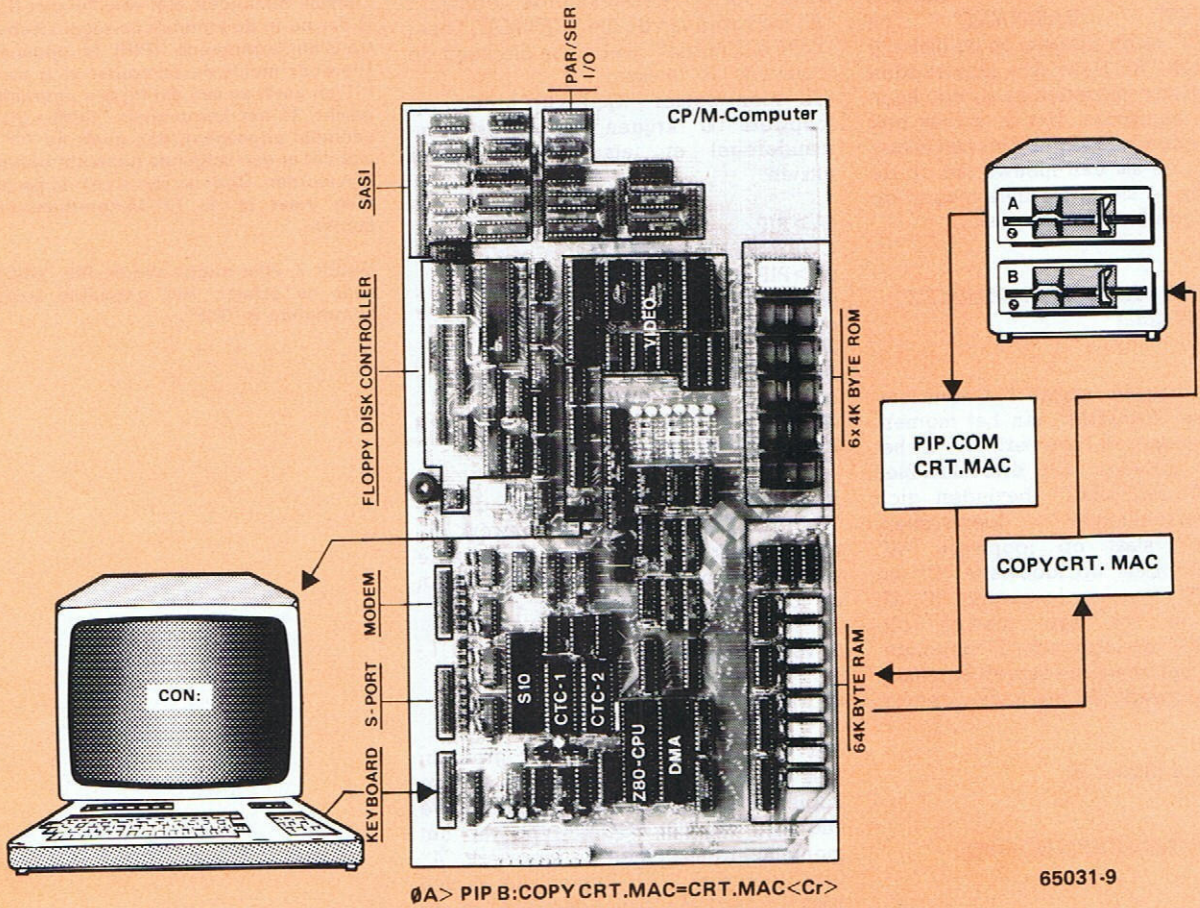
Figuur 6. Hier wordt de file "EC.TXT" naar de console verzonden. Het verschil met figuur 5 is dat nu in de kommando-regel slechts één transient-kommando (PIP) is opgenomen. Hierdoor meldt de computer zich met een "*" en geeft zo aan dat hij een opdracht verwacht. Is het kommando "CON:=EC.TXT" eenmaal uitgevoerd, dan geeft de "*" weer aan dat er een volgende opdracht ontvangen kan worden. Door nu een <cr> te geven kan men weer in de CP/M-kommando-mode komen.

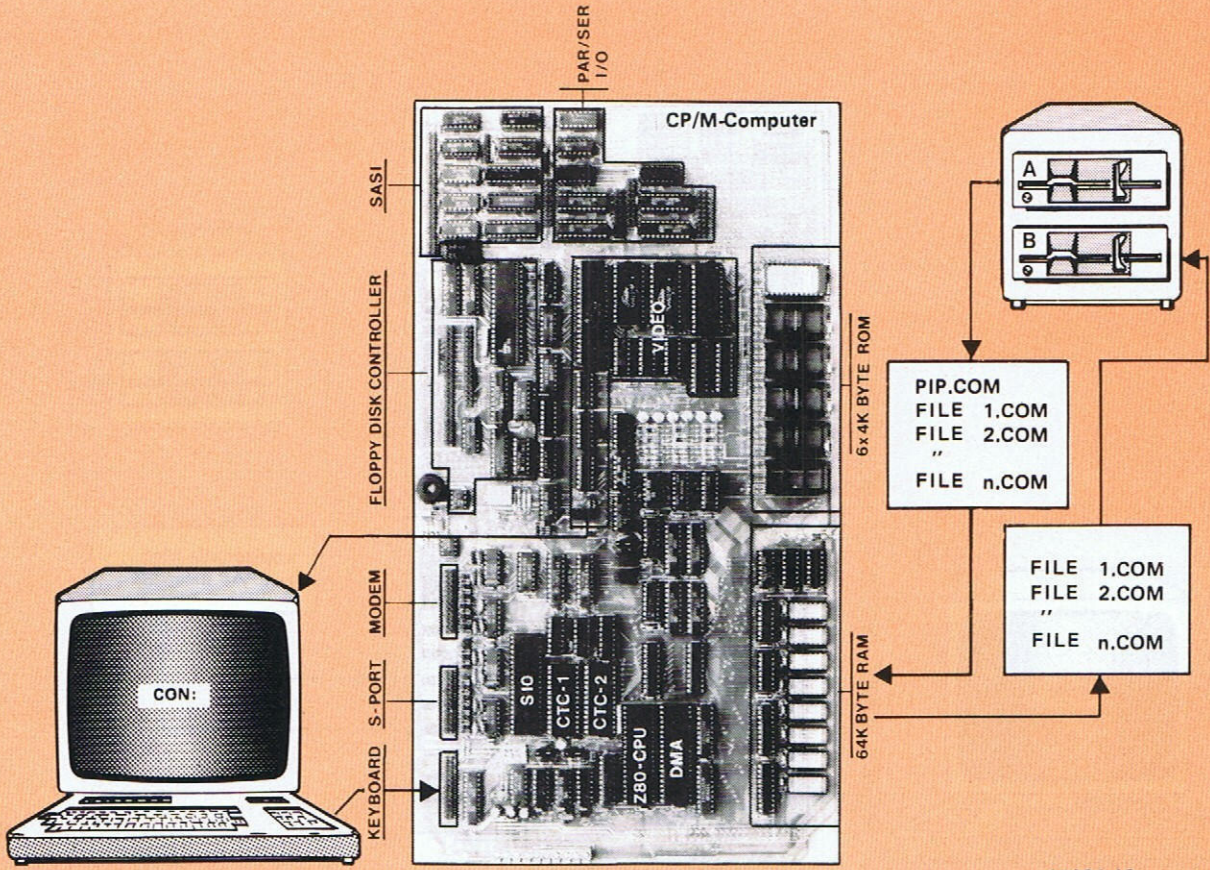
Figuur 7. Hier sturen we de file "BB2.TXT" naar de printer. Het gebruikte transient-kommando is "PIP".

8



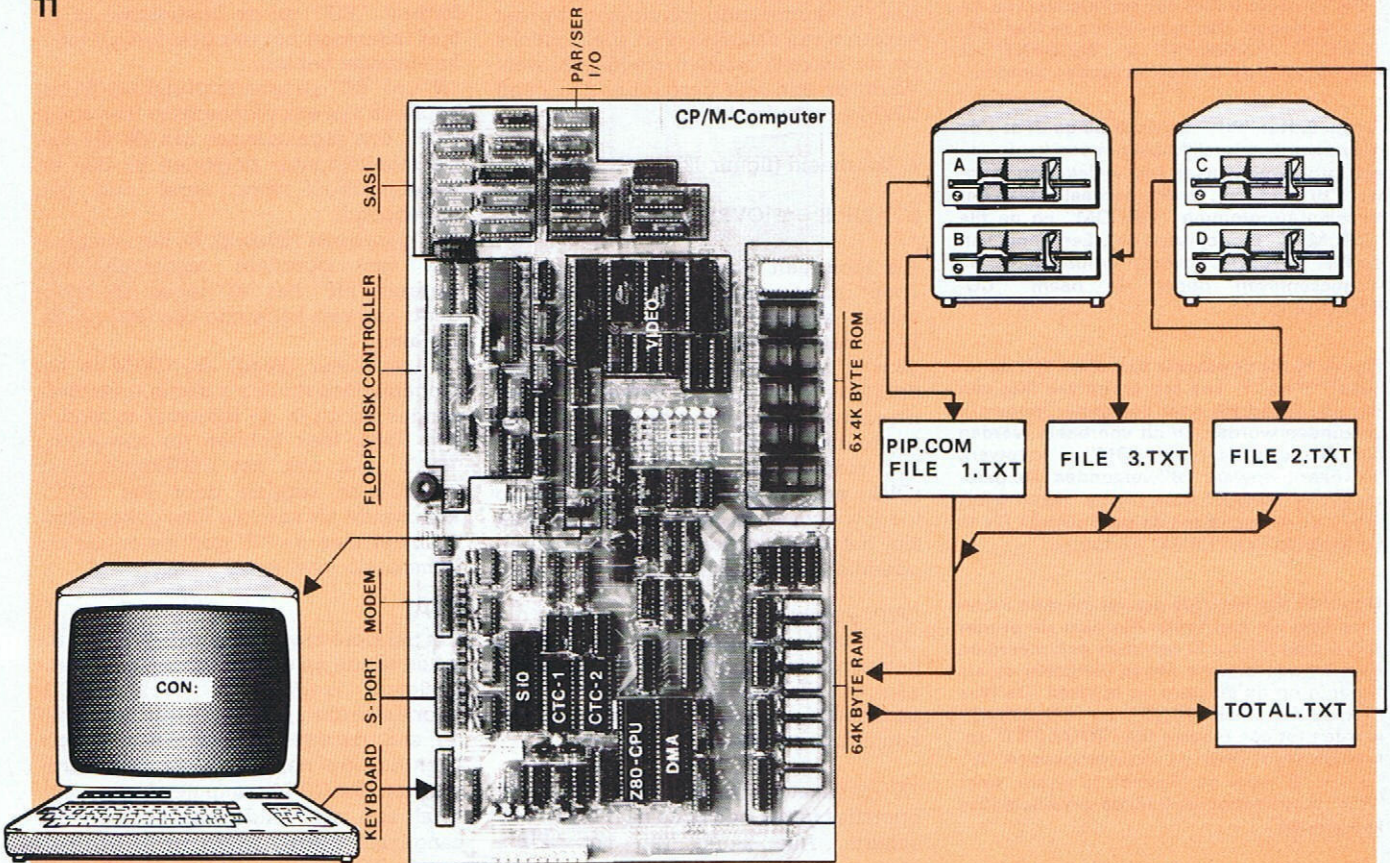
9





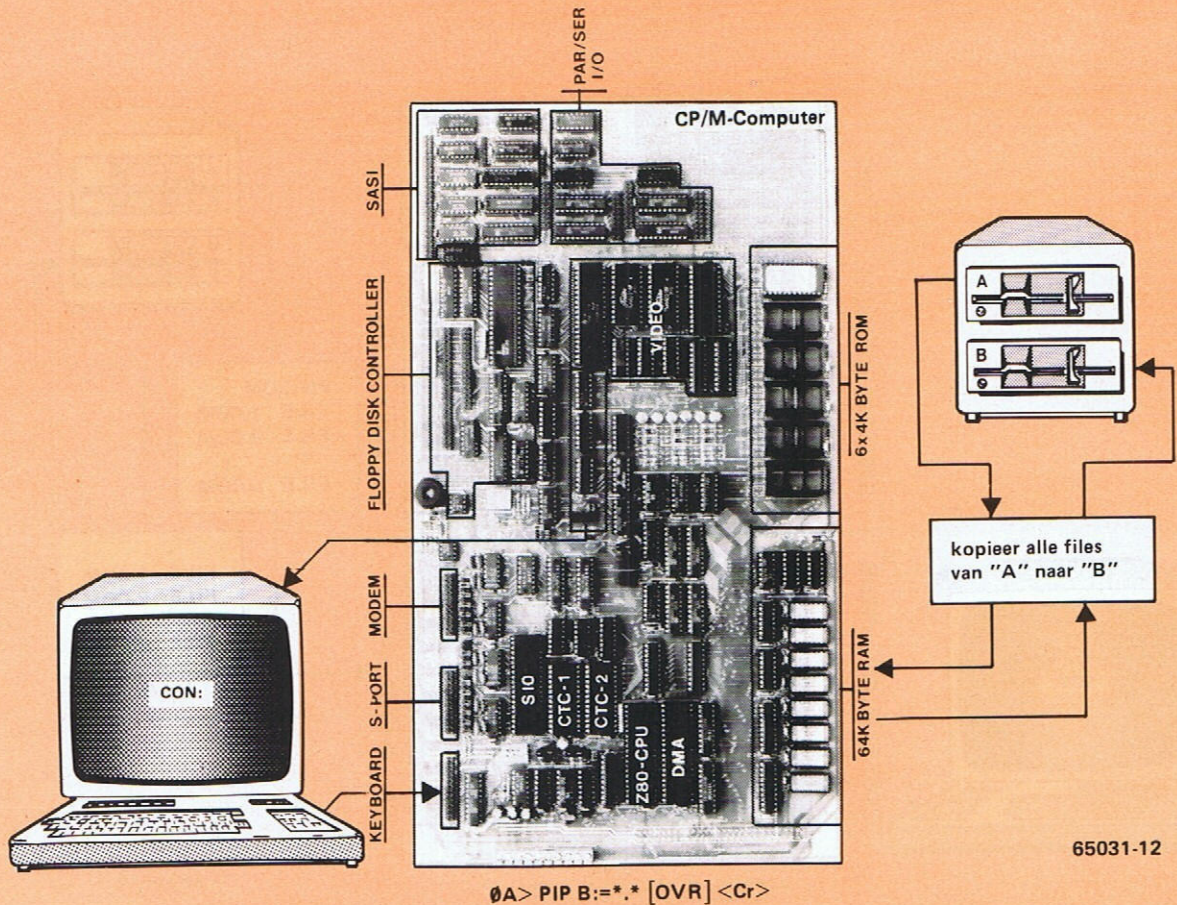
ØA>PIP B:=*.COM [v]<Cr>

65031-10



ØA>PIP B: TOTAL.TXT=A: FILE 1.TXT, C: FILE 2.TXT, B: FILE 3.TXT <Cr>

65031-11



Figuur 8: Het is mogelijk om met behulp van "PIP" meerdere files tot een nieuwe file samen te voegen. Hier zien we hoe de files "PETER.TXT", "KARIN.TXT" en "NICO.TXT" als één file naar de printer verzonden worden.

Figuur 9: Het "PIP"-kommando geeft aan de gebruiker de mogelijkheid om een kopie van een programma naar een willekeurig loopwerk te versturen. In dit voorbeeld staan het transient-programma "PIP.COM" en de file "CRT.MAC" in loopwerk "A". Een kopie van de file "CRT.MAC" wordt op loopwerk "B" weggeschreven onder de naam "COPYCRT.MAC".

Figuur 10: Door gebruik te maken van de tekens "*" en "?" kan een groep van files van het ene loopwerk naar een ander loopwerk verzonden worden. In dit voorbeeld worden alle files van het type "COM" in loopwerk "A" naar loopwerk "B" verzonden. De parameter "V" tussen twee rechte haken geeft "PIP" de opdracht om de gekopieerde file te controleren (V=VERIFY) op fouten.

Figuur 11: Op dezelfde manier als men meerdere files als een totale file naar de printer kan sturen (figuur 8) kan men ook meerdere files tot een nieuwe file te bundelen en als zodanig op de diskette op te slaan. De files "FILE1.TXT", "FILE2.TXT" en "FILE3.TXT" worden tot een nieuwe file "TOTAL.TXT" samengevoegd. Door de loopwerknamen "A", "B" en "C" voor de fileamen te zetten, weet "PIP" in welk loopwerk hij de gewenste files kan vinden.

Figuur 12: Worden alle files van een diskette op een andere gekopieerd, dan moet altijd de parameter "OVR" gebruikt worden. Op deze manier weet men zeker dat de originele file en de kopie daarvan volkomen identiek zijn.

"PIP"-kommando worden toegevoegd. Het "V"-kommando controleert bij het kopiëren van data-files of de files foutloos op de diskette worden weggeschreven. Wordt er een fout gevonden, dan geeft CP/M een foutmelding.

7. Voorbeeld (figuur 12):

```
ØA>PIP B:/**/*.*[OVR]<Cr>
```

Als men een nieuwe systeem-diskette koopt is het zinvol hiervan een kopie te maken, alvorens daarmee te gaan werken. Het behulp van het kommando "B:/*" kan men aan "PIP" mededelen dat alle files van de diskette in het momenteel geselecteerde loopwerk "A" op de diskette in loopwerk "B" gekopieerd moeten worden. Drie parameters (OVR) worden aan het "PIP" kommando toegevoegd, zodat men er zeker van kan zijn dat beide diskettes na de kopieeroperatie identiek zijn.

"PIP"-parameters

"PIP" kent vele parameters. Alleen de belangrijkste zullen we hier kort beschrijven in alfabetische volgorde. "PIP"-parameters staan aan het einde van het kommando tussen rechte haakjes.

"E"-parameter (Echo): Alle kopieeroperaties geven een echo naar het beeldscherm. Alle acties van het "PIP"-kommando zijn nu te volgen.

"F"-parameter (Filter): Alle form-feeds worden bij het kopiëren uit de file gefilterd.

"H"-parameter (Hex-file): Files van het

filetype "HEX" hebben een gedefinieerd formaat. "PIP" zal nu controleren of de files inderdaad het voorgeschreven Intel-hex-formaat hebben.

"N"- of "N2"-parameter (Number): Bij het kopiëren van een of meerdere files voegt "PIP" een regelnummer aan de file toe. De regelnummers beginnen bij één en worden regel voor regel met één opgehoogd.

"O"-parameter (Object): Bij het combineren van meerdere machinetaal-files negeert "PIP" het "Z"-teken. Dit teken geeft normaal het einde van een ASCII-file aan.

"R"-parameter (Read): Bij CP/M is het mogelijk files op een diskette te hebben zonder dat zij in de directory te vinden zijn. Deze files hebben de toevoeging "\$SYS" (zie ook het STAT-kommando). "\$SYS"-files worden door het "PIP"-kommando als normale files gekopieerd, wanneer men de "R"-parameter aan het kommando toevoegt.

STAT

Het transient-programma "STAT" geeft de status van de op een CP/M-computer aangesloten loopwerken. Ook ASCII-georiënteerde apparaten zoals een printer, een cassette-recorder of een modem laten zich met het STAT kommando configureren. De CP/M-computer kan zo optimaal aan zijn randapparatuur worden aangepast. Hierop zullen we in een volgende EC nog terugkomen. Staan er op een diskette zeer belangrijke programma's die onder geen beding overschreven mogen worden, dan kan men ze met het "STAT"-

13

```
OB>A:STAT
A: R/W, Space: 78K
B: R/W, Space: 552K
```

OB>

14

OB>A:STAT DSK:

```
A: Drive Characteristics
6320: 128 Byte Record Capacity
790: Kilobyte Drive Capacity
128: 32 Byte Directory Entries
128: Checked Directory Entries
128: Records/ Extent
16: Records/ Block
40: Sectors/ Track
2: Reserved Tracks
```

```
B: Drive Characteristics
6320: 128 Byte Record Capacity
790: Kilobyte Drive Capacity
128: 32 Byte Directory Entries
128: Checked Directory Entries
128: Records/ Extent
16: Records/ Block
40: Sectors/ Track
2: Reserved Tracks
```

OB>

kommando tot read-only-files omzetten (R/O). De files zijn nu beschermd en kunnen niet meer per ongeluk overschreven worden. Figuur 13 laat de eenvoudigste vorm van een "STAT"-kommando zien. Loopwerk "B" is het momenteel geselecteerde loopwerk. Het "STAT"-kommando staat op de diskette in loopwerk "B". Het is daarom nodig "A:STAT<cr>" in te geven. Op de diskette in loopwerk "A" is van de beschikbare 790 Kbyte nog slechts 78 Kbyte over. Men kan zeggen dat de diskette dus aardig vol is. Op de andere diskette, die in loopwerk "B", is echter 552 Kbyte ter beschikking, dus nog genoeg speelruimte.

Er is nog veel meer informatie over de diskette verkrijgbaar, door middel van: "A:STAT DSK."; zie ook figuur 14. Ook hier is, evenals bij het vorige voorbeeld, het transient-programma te vinden op de diskette in loopwerk "A". Op een diskette staan in totaal 6320 records van ieder 128 bytes. Hiermee is de totale opslagcapaciteit van een diskette dus 790 Kbyte. Aan het einde van de status-opgave is te zien dat per track 40 sectoren aanwezig zijn.

15

A:STAT PIP.COM

```
Recs Bytes Ext Acc
58 8K 1 R/W B:PIP.COM
Bytes Remaining On B: 552K
```

OB>A:STAT PIP.COM \$\$

```
Size Recs Bytes Ext Acc
58 58 8K 1 R/W B:PIP.COM
Bytes Remaining On B: 552K
```

OB>

16

OB>A:STAT X.COM

```
Recs Bytes Ext Acc
7 2K 1 R/W B:ASCII.COM
49 8K 1 R/W B:CONFIG.COM
2 2K 1 R/W B:CRT.COM
8 2K 1 R/W B:HEX.COM
58 8K 1 R/W B:PIP.COM
41 6K 1 R/W B:STAT.COM
9 2K 1 R/W B:TTT.COM
14 2K 1 R/W B:VARSYSG.COM
111 14K 1 R/W B:WS.COM
80 10K 1 R/W B:ZSID.COM
Bytes Remaining On B: 552K
```

OB>

Twee tracks zijn voor CP/M gereserveerd.

Met behulp van het transient-kommando "STAT" kan men ook informatie krijgen over de omvang van een of meerdere files die op een diskette staan. Figuur 15 laat zien dat de file "PIPCOM" in totaal 58 records omvat en circa 8 Kbyte (58*128=7424 byte) lang is. Omdat, zoals figuur 14 laat zien, steeds 16 records als een blok gezien worden, moet dit dus afgerond worden op 8 Kbyte. "EXT" (EXTension) geeft antwoord op de vraag hoeveel 16-Kbyte-blokken door een file gebruikt worden. De file "PIPCOM" is 8 Kbyte lang en past daarom in een "EXT"-blok van 16 Kbyte. Dankzij het attribuut "\$\$" wordt in het voorbeeld van figuur 15 ook nog het "SIZE"-veld afgedrukt.

Figuur 16 geeft nog een ander voorbeeld van het "STAT"-kommando. Hier worden de gegevens van alle "COM"-files in loopwerk "A" gegeven. Ook de "USER"-status kan, zoals figuur 17 laat zien, opgevraagd worden. Big-Board II onderscheidt zich van andere CP/M-systemen in die zin, dat de "USER"-status automatisch

Figuur 13: Dit is de eenvoudigste vorm van het "STAT"-kommando. De diskette in loopwerk "B" is bijna vol. Van de 790 Kbyte is nog slechts 78 Kbyte vrij.

Figuur 14: Het kommando "STAT" geeft de gebruiker informatie over de aangesloten loopwerken. De op Big-Board II aangesloten loopwerken hebben twee koppen (dubbelzijdig) en maken gebruik van double-density (MFM). Op deze manier kan zo'n 790 Kbyte aan informatie op een diskette opgeslagen worden. Maximaal kunnen 128 verschillende file-namen in de directory worden gezet. Een "EXTEND" is een datablok van 16 Kbyte dat 128 records (128 x 128 byte = 16 Kbyte) bevat. Een "BLOCK" bevat 16 records, wat overeenkomt met 2 Kbyte (128 byte x 16 = 2 Kbyte). Op een spoor (TRACK) zitten 40 records. Twee van deze sporen zijn op een diskette gereserveerd voor CP/M.

Figuur 15. Met het "STAT"-kommando kan men ook nadere informatie over een file krijgen. de file "PIP.COM" is 58 records lang en gebruikt daardoor 8 Kbyte op de diskette. "R/W" betekent dat de file niet write-protected is.

Figuur 16. Ook de informatie van een groep files kan men met het "STAT"-kommando opvragen.

Figuur 17: Het kommando "STAT.USER" vertelt de gebruiker welk "USER"-bereik momenteel geselecteerd is. Omdat het meedelen van het "USER"-bereik bij Big-Board II al geïmplementeerd is in de BIOS ("0A>"), hoeft dit kommando niet meer te worden gegeven.

17

0A>STAT USER:

```
Active User : 0
Active Files: 0
0A>
```

wordt weergegeven als op een CP/M-kommando wordt gewacht.

Figuur 18 laat zien dat door het kommando "STAT VAL" de status van alle aangesloten ASCII-randapparaten kan worden weergegeven. CP/M maakt onderscheid tussen logische en fysieke ASCII-apparaten. Een logisch apparaat is een algemene aanduiding van input- en output-randapparaten. Fysieke apparaten zijn die apparaten die ook werkelijk op de computer aangesloten zijn, dus een apparaat dat men zien of aanraken kan. De vier logische apparaten zijn:

CON: Input/output voor console of terminal.

RDR: (ReaDeR) Een ingang waaraan bijvoorbeeld een barcode-lezer, een ponskaartlezer of een cassette-recorder aangesloten kan worden. Op deze ingang kunnen alleen randapparaten aangesloten worden die data aan de computer leveren.

PUN: (PUNcher) Dit is een uitgang waarop bijvoorbeeld een barcode-schrijver, een ponsmachine of een cassette-recorder

18

```

0A>STAT VAL:

Temp R/O Disk: d:=R/O
Set Indicator: d:filename.typ $R/O $R/W $SYS $DIR
Disk Status : DSK: d:DSK:
User Status : USR:
Iobyte Assign:
CON: = TTY: CRT: BAT: UC1:
RDR: = TTY: PTR: UR1: UR2:
PUN: = TTY: PTP: UP1: UP2:
LST: = TTY: CRT: LPT: UL1:
0A>

```

aangesloten kan worden. Op de "PUN"-uitgang kunnen alleen apparaten aangesloten worden die data van de computer ontvangen.

LST: (LiSTing device) Hierop kan men een printer aansluiten.

Aan ieder logisch randapparaat kunnen maximaal vier fysieke apparaten toegekend worden. Dit toekennen kan men met het "STAT"-kommando realiseren. Welk fysiek apparaat aan welk logisch apparaat toegekend is, kan de CP/M-computer in een I/O-byte terugvinden. Dit is een onderwerp waarop we ook nog in een

volgende EC terug zullen komen. Dan volgt nu nog een lijst van mogelijke fysieke apparaten:

TTY: Een schrijf- en leesapparaat (afkorting van TeleTYpe).

CRT: Een lees- en schrijfapparaat met beeldscherm (Cathode Ray Tube)

PTR: Dit is een leesapparaat voor barcode en ponskaarten (Paper Tape Reader).

PTP: Een schrijfapparaat voor barcode en ponskaarten (Paper Tape Puncher).

LPT: Dit is een printer (Line PrinTer).

UC1: Een door de gebruiker te definiëren console (User-defined Console).

Figuur 18. Het kommando "STAT.VAL" geeft de relatie tussen logische en fysieke apparaten. De gebruiker weet welke verdere kommando's bij "STAT" gebruikt mogen worden.

Figuur 19. Het kommando "STAT.DEV" vertelt de gebruiker welke vier fysieke apparaten momenteel aan de vier logische apparaten toegewezen zijn.

19

```

0A>
0A>STAT DEV:
CON: is TTY:
RDR: is TTY:
PUN: is TTY:
LST: is TTY:

0A>

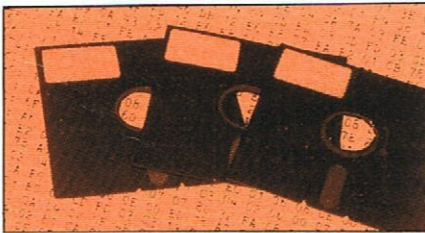
```

URI: en UR2: Twee door de gebruiker te definiëren input-devices (User-defined Reader).

UP1: en UP2: Twee door de gebruiker te definiëren output-devices (User-defined Puncher).

UL1: Een door de gebruiker te definiëren printer (User-defined Listing device).

In figuur 19 is te zien hoe het kommando "STAT.DEV:" werkt. Dit statement laat de gebruiker weten welke fysieke randapparaten momenteel op de computer zijn aangesloten.



VARBIOS

Loys Nachtmann

Wanneer u over een CP/M-machine beschikt, zal dit VARBIOS-programma beslist van groot nut zijn. Door middel van dit programma kan een hele reeks CP/M-software van verschillende afkomst op uw computer draaien. VARBIOS maakt het mogelijk om zo'n kleine veertig gangbare, maar ook "exotische" floppy-formaten te kunnen verwerken. De software van VARBIOS is "public domain"; dit betekent dat de ervaren programmeur dit software-pakket zonder al te veel omwerken op zijn eigen CP/M-machine kan installeren.

CP/M geldt als een klassiek operating-system voor 8-bit-computers en heeft toch al zo'n tien bedrijfsjaren achter de rug. Destijds, toen de eerste versies van Digital Research op de markt kwamen, waren de 8-inch-loopwerken nog standaard. Dit diskette-formaat is tegenwoordig in de Verenigde Staten nog altijd populair; vele

```

1
;XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
;X Switch 1 2 3 X
;X | | | | X
;X 5" | off | off | off |---Single step, 40 tr. dirs=64 X
;X 5" | off | on | off |---Double step, 40 tr. dirs=64 X
;X 5" | on | off | off |---Single step, 80 tr. dirs=64 X
;X 5" | on | on | off |---Single step, 80 tr. dirs=128 X
;X 8" | x | x | on |---Single step, 77 tr. dirs=128 X
;XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

```

programma's zijn uitsluitend op 8"-floppy verkrijgbaar.

Ook de loopwerken zelf hebben gedurende de laatste jaren heel wat verbeteringen ondergaan. Bij de 50"-loopwerken is een opslagcapaciteit van 1 Mbyte al geen uitzondering meer. Daarbij komt dan nog de steeds dalende prijs van mini- en mikro-drives, die het gebruik hiervan steeds interessanter maken. De CP/M-computers gebruiken echter voor het lees/schrijfformaat steeds weer hun eigen formaat. Zodoende zijn er heel wat verschillende formaten in roulatie. Om

deze formaten toch "onder een hoedje te vangen", kan de Big-Board II worden uitgerust met het VARBIOS-programma. Hiermee wordt het doordachte hardware-ontwerp van deze CP/M-computers eigenlijk pas echt compleet, en biedt het de gebruiker de mogelijkheid alle software te kunnen lezen en te kunnen kopiëren.

1) Zowel 50- als 8-inch-loopwerken kunnen gemengd op de computer worden aangesloten, waarbij bijv. ook onderling gekopieerd kan worden.

2) Haast 40 verschillende diskette-formaten (Osborne, IBM, Philips enz.) kunnen met VARIBIOS worden gelezen. Het gewenste formaat wordt d.m.v. een menu in het transiënt-programma CONFIGCOM gekozen.

3) Met VARBIOS kunnen ook loopwerken met verschillende track-density (40 of 80 tracks) tegelijkertijd op de Big-Board II aangesloten worden. Bij 80-track-loopwerken wordt één "step-pulse" per track-wisseling afgegeven, terwijl bij 40-tracks twee pulsen worden afgegeven.

4) Ook loopwerken met een kop en met twee koppen kunnen tegelijkertijd op de computer worden aangesloten.

5) Daar bij 8-inch-drives de data in vergelijking met mini-drives met de dubbele frekwentie worden gelezen en geschreven, is omschakeling van de clock-frekwentie nodig. Bij VARBIOS gebeurt dit automatisch m.b.v. software.

6) Vanzelfsprekend kunnen files van een normale floppy naar hard-disk getransporteerd worden, en omgekeerd. Ook bij een aangesloten hard-disk kunnen de floppy's naar believen gekonfigureerd worden.

7) De nieuwe 3½-inch-loopwerken kunnen ook aangesloten worden, voor zover het 50-compatible versies betreft. Dit is niet altijd het geval; het is dus aan te raden om dit even na te gaan.

Konfiguratie van de Big-Board II

De configuratie van de BB-II gebeurt in twee fasen:

1. Hardware-konfiguratie

Wanneer de computer wordt ingeschakeld, wordt er eerst gevraagd welke loopwerken zijn aangesloten. Na een RESET wordt de stand van de DIP-switches op de printplaat ingelezen. In figuur 1 kunnen we zien hoe deze schakelaars het type drive, van waaruit opgestart wordt, bepalen. Er kan dus opgestart worden met een 50- of een 8-inch-drive. Bij het Big-Board-bouwpakket wordt ook een speciale systeemdiskette geleverd. Het formaat hiervan is: 80 tracks, double sided en double density. De diskette heeft 128 posities voor de directory. Om vanuit

Figuur 1. De stand van de DIP-switches wordt na iedere RESET door het systeem afgevraagd voor de defeniëring van het diskette-formaat van het boot-loopwerk. Om de door Twente Digital geleverde schijf te kunnen gebruiken, moet de voorlaatste combinatie worden gekozen.

Figuur 2. Het CONFIG-programma maakt het mogelijk, om de VARBIOS zodanig te wijzigen dat haast elk CP/M-formaat gelezen kan worden.

Figuur 3. In dit voorbeeld wordt een Osborne-drive gekonfigureerd.

2

0A)config

CONFIG PROGRAM

Warning This program is to be used
In conjunction with a modified monitor and
a modified bios.

DRIVE - A IS CONFIGURED AS "5 1/4 INCH SINGLE STEPS"
DRIVE - B IS CONFIGURED AS "5 1/4 INCH SINGLE STEPS"
DRIVE - C IS CONFIGURED AS "5 1/4 INCH SINGLE STEPS"
DRIVE - D IS CONFIGURED AS "5 1/4 INCH SINGLE STEPS"

WANT TO MAKE ANY CHANGE ? (A-D) OR (CR) TO SKIP -

WANT TO SELECT DOUBLE STEP COUNT FOR 5 1/4" DRIVES ?" (A-D) OR (CR)--

WANT TO CHANGE DRIVE PARAMETERS ? (A-D) OR (CR) --

WANT TO CHANGE SIDE SELECTION ? (A-D) OR (CR) --

END OF CONFIG PROGRAM !!!!

3a

0A)CONFIG

CONFIG PROGRAM

Warning This program is to be used
In conjunction with a modified monitor and
a modified bios.

DRIVE - A IS CONFIGURED AS "5 1/4 INCH SINGLE STEPS"
DRIVE - B IS CONFIGURED AS "5 1/4 INCH SINGLE STEPS"
DRIVE - C IS CONFIGURED AS "5 1/4 INCH SINGLE STEPS"
DRIVE - D IS CONFIGURED AS "5 1/4 INCH SINGLE STEPS"

WANT TO MAKE ANY CHANGE ? (A-D) OR (CR) TO SKIP -

WANT TO SELECT DOUBLE STEP COUNT FOR 5 1/4" DRIVES ?" (A-D) OR (CR)--B

Warning0 This program is to be used
In conjunction with a modified monitor and
a modified bios.

DRIVE - A IS CONFIGURED AS "5 1/4 INCH SINGLE STEPS"
DRIVE - B IS CONFIGURED AS "5 1/4 INCH DOUBLE STEPS"
DRIVE - C IS CONFIGURED AS "5 1/4 INCH SINGLE STEPS"
DRIVE - D IS CONFIGURED AS "5 1/4 INCH SINGLE STEPS"

WANT TO SELECT DOUBLE STEP COUNT FOR 5 1/4" DRIVES ?" (A-D) OR (CR)--

WANT TO CHANGE DRIVE PARAMETERS ? (A-D) OR (CR) --B

deze diskette op te starten, moeten de schakelaars als volgt ingesteld worden:

switch 1: on
switch 2: on
switch 3: off

2) Software-konfiguratie

Is het Disk-Operating-System van drive A geladen, dan kunnen de andere drives via een menu gekonfigureerd worden. Het programma hiervoor heet "CONFIGCOM" en is aanwezig op de schijf die samen met de Big-Board-II-print wordt geleverd. In figuur 2 zien we dat er vier vragen worden gesteld. Allereerst kunnen drive A...D als 5" of 8" worden gekonfigureerd. Na het indrukken van <cr> kan men naar de tweede vraag gaan, als er niets veranderd hoeft te worden.

Nu vraagt het programma of er een enkele of een dubbele step-puls moet worden gegeven (STEP COUNT). Hiermee kunnen de loopwerken op 40 of 80 tracks gekonfigureerd worden. Ook nu kan er weer een <cr> gegeven worden bij default (geen wijziging).

De derde vraag betreft het te gebruiken formaat (DRIVE PARAMETERS). Er kan gekozen worden uit zo'n 40 verschillende formaten, waarbij tevens de drive-parameters worden getoond.

Tot slot vraagt het programma of een enkelzijdig dubbelzijdig loopwerk is aangesloten. Een <cr> sluit het CONFIG-programma af, waarna de drives gekonfigureerd zijn tot aan de volgende RESET.

Een praktijkvoorbeeld

Er is veel CP/M-software voor de Osborne-computer in omloop. Daar deze software relatief goedkoop is, kunnen we deze m.b.v. VARBIOS naar onze Big-Board "vertalen". De B-drive wordt dan met het Osborne-formaat gekonfigureerd. Figuur 3 laat alle mogelijke formaten zien, die met het CONFIG-programma gekozen kunnen worden. Ook het Osborne-formaat maakt hier deel van uit. Wat we verder nog moeten weten, is dat de diskette enkelzijdig wordt gebruikt met double density (SS/DD). Verder telt de Osborne-drive 40 tracks. Meer hoeven we niet te weten!

Stop allereerst de Big-Board-systeemschijf in drive A om CP/M te booten. Start dan het programma "CONFIG". Daar de Osborne met mini-diskettes werkt, kunnen we de eerste vraag met <cr> beantwoorden. Voor het configureren van drive B voor 40 tracks dienen we bij vraag 2 een B in te typen, gevolgd door een <cr>. Dan komt de configuratie van de drive-parameters aan de beurt. We kiezen de letter K en belanden na <cr> bij de vierde en laatste vraag. Hier moeten we tweemaal met B antwoorden om de VARBIOS voor een enkelzijdige drive te configureren. Het programma CONFIG kan nu beëindigd worden door <cr> te drukken. De VARBIOS is nu zodanig aangepast door het CONFIG-programma dat drive B zich nu gedraagt als een Osborne-drive.

Gemengde drives

Het was reeds bekend dat de VARBIOS

3b

ENTER SELECTION FROM FOLLOWING TABLE 0,1,2 ETC.

0 = 5"	SS/SD	BB2	TR=80	BLK=1K	DIRS=64	RES=3	SEC=18	XLT=Y	EXTFL=0014H
1 = 5"	DS/DD	BB2	TR=80	BLK=2K	DIRS=256	RES=2	SEC=XX	XLT=N	EXTFL=0000H
2 = 5"	SS/DD	BB2	TR=80	BLK=2K	DIRS=64	RES=3	SEC=XX	XLT=N	EXTFL=0010H
3 = 5"	DS/DD	BB2	TR=40	BLK=2K	DIRS=64	RES=2	SEC=XX	XLT=N	EXTFL=0000H
4 = 5"	SS/DD	EXIDY1	TR=40	BLK=2K	DIRS=128	RES=2	SEC=40	XLT=Y	EXTFL=0016H
5 = 5"	SS/DD	EXIDY2	TR=40	BLK=2K	DIRS=64	RES=2	SEC=32	XLT=Y	EXTFL=0016H
6 = 5"	SS/DD	KAYPRO	TR=40	BLK=1K	DIRS=64	RES=1	SEC=40	XLT=N	EXTFL=0010H
7 = 5"	SS/SD	OSBORNE	TR=40	BLK=2K	DIRS=64	RES=3	SEC=20	XLT=Y	EXTFL=0014H
8 = 8"	DS/DD	UTS40	TR=77	BLK=2K	DIRS=256	RES=2	SEC=52	XLT=N	EXTFL=0020H
9 = 5"	DS/DD	ITT3030	TR=35	BLK=1K	DIRS=64	RES=4	SEC=32	XLT=N	EXTFL=0004H
A = 5"	DS/DD	ITT3030	TR=70	BLK=2K	DIRS=128	RES=4	SEC=32	XLT=N	EXTFL=0004H
B = 5"	DS/DD	TELCOM	TR=40	BLK=2K	DIRS=64	RES=1	SEC=80	XLT=N	EXTFL=0074H
C = 5"	SS/DD	IBMPC1	TR=40	BLK=1K	DIRS=64	RES=1	SEC=32	XLT=N	EXTFL=0014H
D = 5"	SS/SD	XEROX1	TR=40	BLK=1K	DIRS=32	RES=3	SEC=18	XLT=Y	EXTFL=0014H
E = 5"	SS/DD	XEROX2	TR=40	BLK=1K	DIRS=64	RES=3	SEC=34	XLT=N	EXTFL=0014H
F = 5"	SS/DD	DEC-VT1	TR=40	BLK=1K	DIRS=64	RES=2	SEC=XX	XLT=Y	EXTFL=0010H
G = 5"	DS/DD	TELEVID	TR=40	BLK=2K	DIRS=64	RES=2	SEC=72	XLT=N	EXTFL=0074H
H = 5"	SS/SD	TRS-80	TR=36	BLK=1K	DIRS=64	RES=3	SEC=18	XLT=Y	EXTFL=0014H
I = 8"	DS/DD	HOLBORN	TR=77	BLK=2K	DIRS=256	RES=2	SEC=XX	XLT=N	EXTFL=0020H
J = 5"	SS/DD	SUPERBR	TR=40	BLK=2K	DIRS=64	RES=2	SEC=40	XLT=Y	EXTFL=0094H

Screen break, enter selection or space bar for next screen, or (cr) to skip ? ---)

K = 5"	SS/DD	OSBORNE	TR=40	BLK=1K	DIRS=64	RES=3	SEC=40	XLT=N	EXTFL=0010H
L = 8"	DS/DD	D.UNION	TR=77	BLK=4K	DIRS=256	RES=2	SEC=64	XLT=N	EXTFL=0026H
M = 5"	DS/DD	BULLET	TR=80	BLK=2K	DIRS=128	RES=2	SEC=XX	XLT=N	EXTFL=0020H
N = 5"	DS/DD	PHILIPS	TR=80	BLK=4K	DIRS=128	RES=0	SEC=32	XLT=N	EXTFL=002CH
O = 8"	SS/DD	TRS80M2	TR=77	BLK=2K	DIRS=128	RES=2	SEC=XX	XLT=Y	EXTFL=0012H
P = 5"	DS/DD	IDS	TR=40	BLK=2K	DIRS=128	RES=2	SEC=40	XLT=Y	EXTFL=0024H
Q = 5"	SS/DD	RAINBOW	TR=80	BLK=2K	DIRS=128	RES=2	SEC=40	XLT=Y	EXTFL=0014H
R = 5"	SS/DD	P2500	TR=80	BLK=2K	DIRS=64	RES=3	SEC=32	XLT=Y	EXTFL=0014H
S = 8"	SS/DD	ALDOS	TR=77	BLK=2K	DIRS=128	RES=2	SEC=48	XLT=N	EXTFL=0016H
T = 8"	SS/DD	TRS80	TR=77	BLK=2K	DIRS=128	RES=2	SEC=XX	XLT=Y	EXTFL=0010H
U = 5"	DS/DD	KAYPRO4	TR=40	BLK=2K	DIRS=64	RES=1	SEC=40	XLT=N	EXTFL=0013DH
V = 5"	DS/DD	INTEL	TR=40	BLK=2K	DIRS=64	RES=2	SEC=36	XLT=N	EXTFL=0004H
W = 5"	SS/SD	NEC	TR=40	BLK=1K	DIRS=64	RES=2	SEC=32	XLT=N	EXTFL=0014H
X = 5"	DS/DD	MBC1000	TR=40	BLK=2K	DIRS=64	RES=2	SEC=32	XLT=Y	EXTFL=0024H
Y = 5"	DS/DD	MBC1200	TR=80	BLK=4K	DIRS=128	RES=4	SEC=32	XLT=Y	EXTFL=0024H
Z = 5"	DS/DD	TV8/1602	TR=40	BLK=2K	DIRS=64	RES=4	SEC=XX	XLT=N	EXTFL=0020H

TELEVIDEO 802 & 1602 = "Z", TELEVIDEO 1603 = "M" (BULLET)
? --> K

WANT TO CHANGE DRIVE PARAMETERS ? (A-D) OR (CR) --

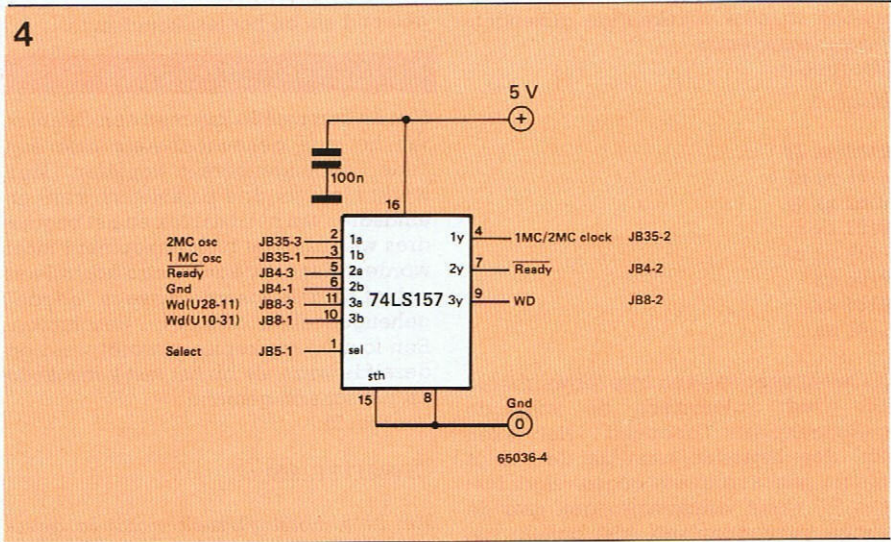
WANT TO CHANGE SIDE SELECTION ? (A-D) OR (CR) --B
DRIVE -- B IS TESTED FOR DOUBLE SIDE ON SELECTION!!!!

WANT TO CHANGE SIDE SELECTION ? (A-D) OR (CR) --B
DRIVE -- B IS SELECTED AS SINGLE SIDED ONLY!!!

WANT TO CHANGE SIDE SELECTION ? (A-D) OR (CR) --

END OF CONFIG PROGRAM !!!!

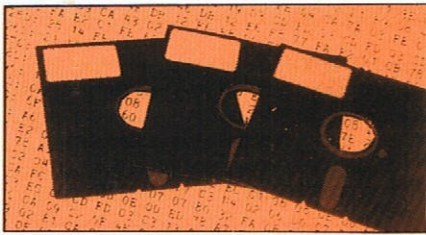
meerdere soorten drives gemengd kan hanteren, zoals 5 $\frac{1}{4}$ "- en 8"-drives, maar bijv. ook een combinatie van 3 $\frac{1}{2}$ "-, 5 $\frac{1}{4}$ " en 8"-drives. De 3 $\frac{1}{2}$ "-drives moeten dan wel compatible zijn met de 5 $\frac{1}{4}$ "-ers. Het is ook mogelijk om data van een hard-disk naar floppy te kopiëren en omgekeerd. Om al deze mogelijkheden te kunnen benutten, moet de in figuur 4 afgebeelde hulpschakeling worden opgebouwd. Hiervoor is plaats vrij op het stukje "gaatjesboard" in de hoek. De 74LS157 is een multiplexer die door software bediend wordt. Hij schakelt om tussen diverse clock-signalen en besturingssignalen. Met stukjes soepele draad kunnen de verbindingen van en naar de konnektoren gelegd worden. "JB35-3" betekent zoveel als "pen 3 van konnektor JB35". Wanneer er twee gelijke drives worden gebruikt en we genoeg nemen met alleen Big-Board-formaat en ook niet denken aan een uitbreiding met een hard-disk, dan hoeft deze schakeling natuurlijk niet opgebouwd te worden. Toch is het aan te bevelen om de schakeling er direkt bij te bouwen, want dan kan de Big-Board II voortaan alle CP/M-formaten lezen.



Public-domain-software

Het source-programma voor de VARBIOS is vrijgegeven als "public-domain-software". Een kopie van dit programma is verkrijgbaar bij de firma Twente Digital.

Figuur 4. Wanneer er verschillende drives tegelijkertijd aan de Big-Board II worden aangesloten, dan moet men deze kleine schakeling opbouwen op het gedeelte gaatjes-board.



BBII: de ROM-monitor

Twente-Digital

Big Board II bezit een monitor die veel nuttige kommando's bevat. Na het inschakelen van de Big Board II wordt de inhoud van de ROM eerst gekopieerd in het RAM-bereik, van F000H tot FFFFH. Na het kopiëren wacht de computer totdat men op een willekeurige toets drukt. Is dat gebeurd, dan probeert BIG Board II of hij CP/M kan laden van drive A. Als in deze drive een CP/M-systeemschijf zit, verschijnt even later op het beeldscherm A voor CP/M en 0A voor ZCPR2. Zit er geen CP/M-schijf in drive A, dan geeft de computer een foutmelding en start hij het monitorprogramma. Het sterretje * geeft aan dat de monitor staat de wachten op een kommando. De monitor heeft twee start-adressen:

koude start: F000H
warme start: F003H

De gebruiker heeft de beschikking over alle monitorkommando's die in figuur 1 staan. Deze kommando's gaan we nu stuk voor stuk bekijken.

Het D(ump)-kommando

Het dump-kommando geeft de geheugeninhoud weer op de monitor. Deze geheugeninhoud wordt zowel hexadecimaal als in ASCII-kode getoond. Elke regel is als volgt opgebouwd:
AAAA DD DD DD...DD DD CCC...C
AAAA is het hexadecimale adres van het eerste getoonde byte DD. C is het ASCII-ekwivalent van elk byte. Op elke regel worden 16 bytes getoond. Voor bytes die

1	d(ump)	D<start>,<end>
	m(emory)	M<address>
	t(est)	T<start>,<end+1>
	f(ill)	F<start>,<end>
	c(opy)	C<source start>,<source end>,<dest start>
	v(erify)	V<source start>,<source end>,<dest start>
	x(change)	X<bank number>
	g(oto)	G<address>
	r(ead)	R<unit>,<track>,<sector>
	b(oot)	B
	i(nput)	I<port>
	o(utput)	O<port>,<data>

2	00 :: 0-32K = Monitor EPROM & CRT RAM; 32-64K = BB2 DRAM.
	01 :: 0-64K = BB2 DRAM.
	02 :: 0-32K = Monitor EPROM & CRT RAM; 32-64K = STD RAM.
	03 :: 0-64K = STD RAM.

geen ASCII-ekwivalent hebben, wordt een "" getoond.
Voorbeelden:

D cr Toon 16 regels vanaf het momentele adres.

D100 cr Toon 16 regels vanaf adres 100H.

D100,300 cr Toon het geheugenbereik van 100H tot 300H.

Het M(emory)-kommando

Met het memory-kommando kan men de

Figuur 1. Twaalf zeer nuttige kommando's staan de gebruiker van de Big Board II ter beschikking.

Figuur 2. Big Board II kan werken met vier geheugenbanken. Met het exchange-kommando kan men een van deze banken selecteren. Kiest men een niet bestaande bank, dan wordt dat met een foutmelding kenbaar gemaakt.

inhoud van elke willekeurige geheugen-plaats veranderen.
Voorbeeld:

M100 cr

```
0100 xx 00
0101 xx 01
0102 xx 1F
0103 xx cr
0104 xx cr
0105 xx -
0104 xx -
0103 xx .
```

Na het intypen van een byte (bijvoorbeeld 1F) wordt automatisch de volgende geheugenplaats "geopend". Het teken "cr" doet hetzelfde, maar laat de inhoud van de geheugenplaats ongewijzigd. Met een "-" wordt teruggesprongen naar de vorige geheugenplaats. Met behulp van een "." kan men het memory-kommando verlaten, waarna de Big Board II wacht op een nieuw kommando.

Het T(est)-kommando

Hiermee kan de gebruiker het RAM-geheugen van de Big Board testen. Het hele geheugen kan worden getest in blokken van 256 bytes, met uitzondering van het monitorbereik F000..F7FFH en FF00..FFFFH.

Voorbeeld:

T80,240 cr

Het test-kommando kijkt alleen naar de meest significante bytes van het ingegeven start- en eindadres. Deze worden door de computer "afgerond". Zo wordt in het voorbeeld het geheugenbereik 0000..01FFH getest. Steeds wanneer een geheugenblok succesvol is getest met een bitpatroon, geeft de computer dat aan met een "+" op het scherm. In totaal moeten 256 +-tekens op het scherm verschijnen als het hele geheugenbereik wordt getest. Is een geheugenplaats niet in orde, dan verschijnt de foutmelding:

AAAA DD should =xx

AAAA is het hexadecimale adres van de defekte geheugenplaats. DD is de data die uit deze geheugenplaats wordt gelezen en XX is de data die er in zou moeten staan.

Het F(ill)-kommando

Met het Fill-kommando kan men een bepaald geheugenbereik vullen met een willekeurig byte. Het fill-kommando kent drie parameters: het start- en het eindadres van het gewenste geheugenbereik en het byte dat in al deze plaatsen moet worden geschreven.

Voorbeeld:

F100,B5FF,1A

Dit kommando vult het hele geheugenbereik voor transient-programma's met het teken CTL-Z. Het fill-kommando controleert ook of het teken daadwerkelijk in elke geheugencel van het bereik staat (verify). Ontdekt het fill-kommando een fout, dan wordt dezelfde foutmelding

getoond als bij het test-kommando.

Het C(opy)-kommando

Met dit kommando kan men een datablok van het ene geheugenbereik naar een ander geheugenbereik kopiëren. Men werkt hier met drie parameters: start- en eindadres van het datablok en het beginadres waarnaar het blok gekopieerd moet worden. Het copy-kommando controleert ook of de gekopieerde bytes in "goede" geheugencellen worden geschreven. Een fout bij het kopiëren wordt weer op dezelfde wijze als bij het test-kommando op het scherm getoond.

Voorbeeld:

CF000,FFFF,1000 cr

Het hele monitor-bereik wordt zo gekopieerd naar de adressen 1000...

Het V(erify)-kommando

Dit kommando vergelijkt twee datablokken in het geheugen met elkaar. Verder werkt dit hetzelfde als het copy-kommando. Ook hier wordt weer een foutmelding (zoals bij test) gegeven als ergens een fout wordt gevonden.

Het (e)X(change)-kommando

Met het exchange-kommando kan men omschakelen naar een andere geheugenbank. Wordt het X-kommando gegeven zonder parameters, dan toont de computer het nummer van de momenteel geselecteerde geheugenbank. De Big Board heeft vier geheugenbanken met de nummers 00..03. Zo'n nummer vormt een 2-bits-kode die de geheugenbankselectie stuurt. Figuur 2 toont alle mogelijke configuraties die men kan selecteren met behulp van het exchange-kommando. Geeft men bijvoorbeeld het kommando X03 cr, dan selecteert Big Board II een RAM-geheugen dat is aangesloten op de STD-bus. Is hier geen geheugen aanwezig, dan geeft de computer een foutmelding en schakelt weer terug naar de laatstgekozen geheugenbank.

Sommige monitorfuncties worden altijd uitgevoerd via de EPROM. Kiest men nu een bank waarin de EPROM is afgeschakeld, dan wordt deze in zo'n geval weer kort ingeschakeld. De gebruiker merkt er niets van als de computer automatisch naar een andere geheugenbank springt. Bij het maken van eigen programma's kan men bijvoorbeeld gebruik maken van de mogelijkheid om te kijken in het beeldherhalingsgeheugen, door middel van het kommando X00 cr of X02 cr. De beeld-buffer begint op adres 6000H en de tabel voor de beeld-attributen begint op adres 7000H.

Het G(oto)-kommando

Met het goto-kommando kan men een machinetaalprogramma starten. Als parameter wordt het startadres van het uit te voeren programma ingegeven. Eindigt het programma met een RET-instructie, dan wordt automatisch teruggesprongen naar de monitor, waarna de Big Board II weer wacht op een nieuw kommando.

Het R(ead)-kommando

Met dit kommando kan een willekeurige sektor van de diskette worden gelezen en in het geheugen gezet. Daarna wordt de sektor-inhoud automatisch op het beeldscherm getoond. Het read-kommando heeft drie parameters: een disk-drive-nummer (0..3), een track-nummer (0..4CH) en een sektor-nummer (1..?). Voor het maximale sektor-nummer kan geen waarde worden gegeven, aangezien deze afhangt van het gebruikte formaat. Bij een te hoog gekozen sektor-nummer verschijnt op de monitor een foutmelding.

Het read-kommando selecteert de gekozen drive, zoekt het gewenste spoor op de diskette, leest vervolgens de bewuste sektor en zet deze in het geheugen.

Voorbeeld:

R0,0,1 cr

Dit kommando leest van drive A sektor 1 van track 0, zet deze in het geheugen en toont hiervan een hexdump op het scherm. Als een fout optreedt bij het lezen van een sektor, geeft de computer de volgende foutmelding:

DISK ERROR XX SD/DD UAA TBB SCC

Daarbij is XX de inhoud van het statusregister van de disk-controller, SD/DD staat voor single density/double density, AA is het drive-unit-nummer, BB is het track-nummer en CC is het sektor-nummer (U = unit, T = track, S = sektor).

Het B(oot)-kommando

Het boot-kommando kopieert CP/M van de diskette in het geheugen en start vervolgens het operating system. Dit kommando begint met het lezen van sektor 1 van track 0 in drive A en slaat de data op in het geheugen vanaf adres 80H. Daarna wordt weer teruggesprongen naar adres 80H, waar dan de informatie staat voor het laden van CP/M in het geheugen.

Het I(nput)-kommando

Met het input-kommando kan men het bitpatroon op de input-poorten van de Big Board II lezen. Dit kommando lijkt erg veel op het M-kommando. Zelfs de bedieningstekens cr, - en . werken hier op dezelfde wijze als bij het memory-kommando.

Voorbeeld:

ID9 cr

Hiermee wordt status-poort D9 gelezen.

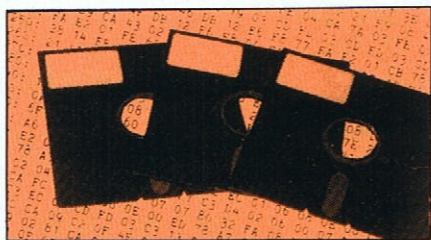
Het O(utput)-kommando

Met dit kommando is het mogelijk om een bitpatroon te zetten op de output-poorten of in periferie-IC's zoals de SIO, de CTC of de 6845-CRTC. Het kommando bevat twee parameters: het poortnummer en het byte dat daar naar toe moet worden gestuurd. Na het uitvoeren van dit kommando wordt weer teruggesprongen naar de monitor.

Voorbeeld:

ODA,AA

Hiermee zet men op output-poort DA het bitpatroon AA.



graphics-software

Paul Lavigne
Denis Meyer

Eigenlijk is dit een overbodige vraag, maar u hebt vast wel de tweede pagina voor in deze special gezien met al die prachtige video-afbeeldingen in kleur. Al deze beelden zijn gemaakt met behulp van de grafische kleurenkaart van Elektuur. We moeten er nog even bij vertellen dat de programma's voor die mooie beelden ernaast zijn afgedrukt, en dat die beelden in werkelijk ook nog voortdurend veranderen op het scherm. Interessant, nietwaar?

De ontwerpers van de grafische kleurenkaart hebben bijzonder veel plezier aan hun werk gehad, wat u duidelijk kunt zien aan de getoonde foto's. In werkelijkheid hebben ze nog veel meer grafische demonstratieprogramma's op schijf staan, teveel om hier allemaal te laten zien.

Degene die de grafische kaart al kent uit de artikelen in Elektuur (september, oktober en november 1985), kunnen we bovendien een nieuwtje vertellen: we hebben nog een intensiteitsbit toegevoegd. Daarmee is het aantal kleuren verdubbeld, dus van 8 naar 16. Hiervoor is natuurlijk wel een RGB-monitor met intensity-ingang noodzakelijk (er zijn wel andere mogelijkheden denkbaar, zoals het mengen van de signalen tot een FBAS-

signaal, maar die geven allemaal een slechtere beeldkwaliteit).

Het opzetten van de hardware voor een grafische kaart is niet zo'n grote klus. Waar de meeste tijd in gaat zitten bij zo'n project is de software. Die software maakt van de grafische kleurenkaart pas een intelligente grafische kleuren-terminal. De publikaties van de grafische kaart in Elektuur hebben daar al heel wat bladzijden in beslag genomen, en dat was ook zeker nodig in verband met de complexe opzet van deze kaart. Toch was er nog ruimte te weinig. Zo was er geen plaats meer voor de demonstratieprogramma's en de assembler-source-listing. Dat doen we echter alsnog in deze special. Nog even ter informatie: voor de ontwikkeling van de hardware, de software en de demonstratieprogramma's is steeds gebruik gemaakt van de Octopus 65, onze computer uit de eerste computer-special. Zoals we u al hebben verteld aan het begin van dit stukje, zijn de listings van de BASIC-demonstratieprogramma's afgedrukt naast de foto's van de videobeelden op de kleurenplaat. Een deel van de assembler-listing vindt u hieronder.

Waarom maar een gedeelte? Wel, de gehele assembler-listing bestaat uit drie

delen: "text", "graphics" en "circle". De hele listing is zo lang dat in deze special weinig plaats zou overblijven voor andere onderwerpen. Daarom drukken we hier eerst het gedeelte "text" af. Weliswaar zijn de twee andere delen voor de geïnteresseerde grafische-kaart-bouwer iets interessanter, maar tenslotte vormt dit gedeelte de basis van het geheel. In volgende specials zullen we de twee overige delen dan publiceren.

Het gedeelte "text" zorgt voor de weergave van alfanumerieke tekens. Bovendien "filtert" het de grafische kommando's uit de binnenkomende tekens en geeft ze verder aan het "graphics"-gedeelte. Dit laatste is verantwoordelijk voor alle grafische acties met uitzondering van de "ronde" zaken. Figuren zoals cirkels, schijven, ringen en delen daarvan worden gegenereerd door het "circle"-gedeelte. De hierbij gebruikte assembler is de ouwe getrouwe MOS-Technology-assembler, die de meeste lezers zeker kennen van de OSI-machines en de DOS-junior-computer. Dit ter informatie voor degene die al gewend was geraakt aan de Microware-assembler.

```

10      :#####
20      :#      VIDEO      #
30      :#      -----      #
40      :#VIDEO UTILITIES WITH EF 9366 #
50      :# WRITTEN BY P.LAVIGNE NOV'83 #
60      :#
70      :# PART 1 : TEXT      #
80      :#
90      :#####
100     :
110 E200      X = $E200      :GDP REGS
120 E200=     STATUS = X
130 E200=     CMD = X
140 E201=     CTRL1 = X+1
150 E202=     CTRL2 = X+2
160 E203=     CSIZE = X+3
170 E205=     DELTAX = X+5
180 E207=     DELTAY = X+7
190 E208=     MSBX = X+8
200 E209=     LSBX = X+9
210 E20A=     MSBY = X+10
220 E20B=     LSBY = X+11
230
240 E218      X = $E218      :HARDWARE REGS
250 E218=     COLOR = X
260 E219=     SCROLL = X+1
270 E21B=     HSCROL = X+3

280 E21A=     PAGE = X+2
290
300 CF80      X = $CF80      :SOFTWARE REGS
310 CF80=     ASAV = X
320 CF81=     XSAV = X+1
330 CF82=     YSAV = X+2
340 CF83=     SAJA = X+3
350 CF84=     SAJX = X+4
360 CF85=     MIRXH = X+5
370 CF86=     MIRXL = X+6
380 CF87=     MIRYH = X+7
390 CF88=     MIRYL = X+8
400 CF89=     CHAR = X+9
410 CF8A=     PIXBUF = X+10
420 CF8B=     CNTR = X+11
430 CF8D=     GRCNT = X+13
440 CF8E=     CURSOR = X+14
450 CF8F=     CRSCNT = X+15
460 CF90=     CRSPNT = X+16
470 CF91=     COLRPT = X+17
480 CF92=     SCRLPT = X+18
490 CF93=     PAGEPT = X+19
500 CF94=     MODE = X+20
510 CF95=     TXTMOD = X+21
520 CF96=     VIDFLG = X+22
530 CF97=     BACKGR = X+23
540 CF98=     VIDPNT = X+24

```


gramma wordt het gewoon met RUN gestart. De routine wordt door het BASIC-programma in de adressen 32768 (\$8000) tot 32816 (\$8030) van de Z-RAM geplaatst. Het startadres van het BASIC-programma in de geheugenruimte van de computer is nu dus veranderd — en dat heeft gevolgen. Bezitters van een BASIC-compiler (die veel sneller werkt dan de ingebouwde BASIC-interpret) zullen tot de onaangename ontdekking komen dat gecompileerde programma's niet meer werken na de trauk met het nieuwe startadres. Dat heeft een eenvoudige reden: met het veranderen van het startadres veranderen ook alle absolute sprongadressen. Dit probleem komt dus alleen maar voor bij een compiler, bij de gewone BASIC-interpreters van de C64 werkt de verplaatsing uitstekend. Diverse BASIC-routines, zoals de beeldscherm-editor, werken ook niet meer als het startadres wordt gewijzigd. De volgende belangrijke kommando's blijven echter altijd goed werken: RUN, STOP, LIST, LOAD en SYS. Dit betekent in de praktijk dat men een programma eerst moet ontwikkelen en testen op de gewone wijze, vervolgens dit programma

opslaan op cassette of floppy, en dan het programma direkt van het opslagmedium op de juiste plaatsen in het pseudo-EPROM-bereik laden. Nadat dit is gebeurd, kan men het programma normaal starten met RUN of SYS32768, of een LIST-kommando geven. Even een opmerking tussendoor: met een reset of een SYS64738 kan men de C64 weer terugzetten in zijn "normale" toestand, wanneer men het startadres heeft gewijzigd met behulp van het programma uit tabel 2.

Handleiding

Tot slot geven we nog een samenvatting voor de bediening:

* Laden van BASIC-programma's in de Z-RAM-module: Module in de expansion-port van de computer steken, daarna C64 inschakelen. Vervolgens de schrijfbeveiliging uitschakelen, het programma uit tabel 2 laden en starten met RUN. Nu kan het gewenste programma worden geladen. Einde programmering.

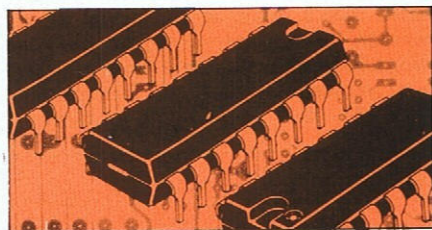
* Starten van BASIC-programma's in de Z-RAM-module: Starten voor de eerste maal

Tabel 2.

```
10 DATA 162,255,120,154,162,0,32
20 DATA 135,255,32,138,255,32,129
30 DATA 255,32,83,228,32,191,227,32
40 DATA 68,166,169,43,133,43,169,128
50 DATA 133,44,32,142,166,32,29,168,88
60 DATA 76,174,167,0,74,128,10,0,128,0
100 FOR I=32768 TO 32816:READ D
110 POKE I,D:S = S + D:NEXT I
120 IF S<>5451 THEN PRINT
    ;;leesfout":END
130 SYS 32768
```

(bijvoorbeeld na het inschakelen) met SYS32768. Daarna kan men starten met RUN of met SYS, en kan het programma worden gelIST.

Het werken met de Z-RAM-module is net zo eenvoudig als het opslaan van programma's op floppy of cassette, alleen gaat het een stuk sneller. Voor bezitters van een C64 is het bovendien interessant om eens in de praktijk te zien hoe met de expansion-port van deze computer kan worden gewerkt. Een van de vele mogelijkheden die er zijn met deze port.



BB II: de hardware

Twente-Digitaal

De BB II, oftewel de Big-Board II, kan met recht een volwassen Z80-computersysteem genoemd worden. Deze CP/M-machine springt er kwa prijs/prestatieverhouding positief uit ten opzichte van andere CP/M-machines, maar levert hiervoor beslist geen "features" in! Om maar wat te noemen: vier seriële poorten, een aantal parallelle I/O-poorten, een uitmuntende video-interface, direkte programmering van EPROM's en EEROM's in de ROM-voetjes, direkte aansluiting van een hard-disk (Winchester) enzovoort. In feite een machine die niets te wensen over laat.

De schema's van de BB II — de hardware — worden op acht aparte pagina's aan het einde van dit artikel gepresenteerd. Laten we maar eens stap voor stap de afzonderlijke delen doornemen.

De klokgenerator

Alle in de computer voorkomende kloksignalen worden door een centrale oscillator verzorgd (schemablad 2). Alleen de dot-klok wordt apart opgewekt. De klok-oscillator "loopt" op 8 MHz. Dit gebeurt

rond U-19, die met de omringende componenten een inverter-kristal-oscillator vormt. Het 8-MHz-sigitaal wordt rechtstreeks door de DRAM-controller, de disk-data-separator en de baudrate-generator gebruikt. Verder wordt de 8 MHz in U-20 door 2, 4, 8 en 16 gedeeld. Het 4-MHz-sigitaal wordt door een inverter en een transistortrapje tot een geschikt kloksigitaal voor de Z80 gemaakt (logische nivo's zijn ook daadwerkelijk 0 V en 5 V). Ook de RESET-schakeling, de write-precompensation voor de floppy en de DRAM-controller worden vanaf de CPU-klok verzorgd. De 2-MHz- en 1-MHz-signalen worden via een jumper naar de floppy-disk-controller geloodst.

A propos, voor de gang van zaken door de 8 schemabladen goed te kunnen volgen, wordt bij de aanduiding van de in- en uitgaande lijnen op een bepaald blad tussen haakjes het bladnummer aangegeven waar de lijnen naar toe lopen. Zo zijn de afzonderlijke schema-delen beter leesbaar.

De RESET-schakeling

Nadat de voedingsspanning ingeschakeld is, blijft flipflop U-18 zolang geset tot dat C21 tot de triggerdrempel is opgeladen (via R5). Op die manier zijn we er zeker van dat de kristal-oscillator "op dreef is". Ook een handbediende RESET is mogelijk. De RESET-schakeling is gesynchroniseerd met de M1-cyclus van

de processor, omdat de Z80 alleen tijdens deze cyclus een RESET aksepteert. Dit gebeurt door de teller U-20 (de bovenste), welke steeds 8 klok-cycli telt en daarna via Q_D de flipflop weer reset.

De I/O-data-buffer

De achtvoudige bus-transceiver N40 verzorgt het dataverkeer tussen de Z80 (via databuffer U-61) en alle input/output-devices die in het I/O-adresbereik liggen. De buffer kan de I/O-databus alleen dan aansturen als het systeem zich niet in een DMA-interrupt-acknowledge-cyclus of in een lees/schrijf-operatie via de I/O-poorten 80...FF bevindt.

1	SIZE	DENS	SMC1	SMC2
	8"	DD	0	0
	8"	SD	1	0
	5.25"	DD	1	0
	5.25"	SD	0	1

65029-1

Figuur 1. Via SMC1 en SMC2 laat de data-separator weten welk type loopwerk met welke lees/schrijfdichtheid op de floppy-interface is aangesloten.

2

```

EQUATES
SYS1 EQU 0C8H ;SYS1 port address
SIDESEL EQU 1 ;SIDESEL bit number
ON EQU 8
OFF EQU 0

```

The following two lines of code set "sidsel":

```

3E 09 LD A,SIDESEL + ON
D3 C8 OUT (SYS1),A

```

The next two lines of code reset "sidsel":

```

3E 01 LD A,SIDESEL + OFF
D3 C8 OUT (SYS1),A

```

65029-2

3

DB2	DB1	DB0	FUNCTION
1	1	1	Reserved for "SASI" DMA
1	1	0	Not Allocated
1	0	1	" "
1	0	0	" "
0	1	1	" "
0	1	0	Data Request (DRQ) from FD controller
0	0	1	SIO Channel B WAIT/READY
0	0	0	SIO Channel A WAIT/READY

65029-3

De buffers voor de adres-, data-, en control-bus

U-63 en U-64 vormen de adresbusbuffers voor de Z80 en de DMA (Direct Memory Access). Beide IC's zijn konstant ge-enabled en verzorgen de interne adresbus en de externe STD-bus. De databus wordt gebufferd door U-61. Alle databus-signalen en de interrupt-vektoren lopen via deze buffer. Interrupts die door de DMA-controller (U-62) gegenereerd worden, verlopen niet over U-61! De databus-buffer is bij DMA-operaties namelijk uitgeschakeld (in "third state"). U-38 buffert een deel van de controlbus (MI, MREQ, IORQ, RD, WR en RFSH) en is, net als de adresbusbuffers, konstant geactiveerd.

De dynamische RAM's

In schemablad 3 vinden we de geheugen-huishouding van de BB II. U-69, U-70 en U-71 zijn (mede) verantwoordelijk voor de besturingssignalen van de dynamische RAM's U-72...U-79, samen 64 Kbyte. Wanneer \overline{BMREQ} actief wordt, dan wordt ook \overline{RAS} actief en wordt de onderste helft van de adresbus in de dynamische RAM's gelezen. Het B/A-sig-naal van de beide multiplexers U-67 en U-68 wordt door drie inverters een aantal nanosekonden vertraagd. Tijdens de volgende cyclus zijn de bovenste adresbits aan de beurt. De opgaande flank van de $2-\phi$ -klok activeert \overline{CAS} , terwijl het signaal \overline{BMREQ} in flipflop U-70 gelezen wordt. De volgende opgaande flank van de ϕ -klok beëindigt de \overline{RAS} -cyclus omdat de \overline{BMI} -cyclus "gemaskeerd" wordt. Dit alles levert een exakte en onkritische timing van de DRAM-besturing.

De ROM's

De ROM-voetjes op de BB II-print zijn niet alleen voorbehouden aan lees-geheugens. Er kan heel wat meer mee gedaan worden:

- 1) plaatsing van ROM's en EPROM's
- 2) plaatsing van statische CMOS-RAM's (2016, 6116 enz.)
- 3) directe programmering van EPROM's

Alvorens de processor zich aan de ROM-bank kan "vergrijpen", moet eerst het D-S-bit van de system-control-port #-1 "0" zijn (deze komen we later nog tegen). De 6 ROM-sockets zijn met het systeem verbonden door buffers en tussengeheugens (latched bus-drivers). Zodoende bestaat de mogelijkheid van directe programmering van EPROM's en EEPROM's. In het adresbereik 0000...5FFF worden bij elke lees/schrijf-cyclus de adresbits in de latches U-59/U-60 gestopt. De adreslijnen RA12...RA14 en de dekodeer U-55 selekteren dan één van de zes ROM-plaatsen U-80...U-85. Het signaal \overline{SMEMOP} (Static MEMORY Operation) klokt de adres-latches U-59 en U-60. De multiplexer U-42 loodst het signaal \overline{SMEMRD} (Static MEMORY Read) naar de OE-aansluiting van de geselecteerde ROM. Ook wordt U-58 door het \overline{SMEMRD} -signaal ge-enabled en doet zo dienst als ROM-databuffer. Bij een schrijfoperatie naar ROM doet U-57 dienst als schrijf-buffer. In tegenstelling tot de lees-buffer U-58 kan deze de data vasthouden Voor het programmeren (of schrijven naar eventuele RAM's) klokt \overline{SMEMWR} (Static MEMORY Write) via multiplexer U-42 de schrijf-data door naar de ROM-socket. Dit

kloksignaal is tevens \overline{WE} voor statische RAM's (pen 21 van het voetje). Het programmeren van EPROMS verloopt via U-41. Deze verzorgt de volgende taken:

- 1) in- en uitschakelen van de CS-dekodeer U-55
- 2) setten en resetten van de PGM-lijn
- 3) in- en uitschakelen van de programmeerspanning V_{PP}
- 4) in- en uitschakelen van het data-tussengeheugen

SYS1: System-control-port 1

In schemablad 7 zien we een aantal I/O-poorten. Op adres 0C8 vinden we system-control-port 1 (U-14). Deze poort is alleen bestemd voor output en bevat de volgende acht signalen:

D-S: Bit 0 is het D-S-bit. Dit geeft de omschakeling tussen DRAM-geheugen en ROM-geheugen aan. Na een RESET is dit bit "0" ten teken dat het onderste geheugenbereik (0000...7FFF) bestemd is voor de ROM-bank. Is dit bit "1", dan is de DRAM-bank of de STD-bus ingeschakeld.

SIDESEL: Bit 1 is het SIDE-SElect-bit. Is dit "0", dan wordt in een dubbelzijdige disk-drive kant 0 geactiveerd, terwijl bij de waarde "1" de andere kop in de drive wordt ingeschakeld. Via U-8 wordt dit signaal rechtstreeks naar de konnektoren voor 5 $\frac{1}{4}$ "- en 8"-loopwerken gevoerd.

4

```

Bit # 7 = SW1:1
Bit # 6 = SW1:2
Bit # 5 = SW1:3
Bit # 4 = SW1:4
Bit # 3 = KBDSTB
Bit # 2 = INDEX
Bit # 1 = RxDB
Bit # 0 = RxDA

```

65029-4

Figuur 2. Een voorbeeld voor een korte assembler-routine voor het setten en resetten van het side-select-bit.

Figuur 3. Via system-port SYS2 worden de DMA-multiplexer-adressen afgegeven. Het bitpatroon op de lijnen DB0...DB2 selekteert tussen SIO, floppy-controller en SASI-interface.

Figuur 4. Op de system-status-port bevinden zich 8 flags, die de CPU informatie geven over de status van de periferie. Ook de stand van de DIP-switches SW1 wordt via deze poort ingelezen.

Figuur 5. Het video-display van de Big-Board II is compatibel met een "Lear Siegler ADM-3A"-terminal. Alle besturingscommando's voor het video-display staan hier afgebeeld.

SMC1, SMC2: Deze bits besturen de data-seperator U-6 van de floppy-disk-interface (schemablad 5). Figuur 1 geeft de kodes van deze bits met de bijbehorende types loopwerken en de lees/schrijf-dichtheid op de schijf.

DDEN: Bit 4 is het control-bit voor de floppy-disc-controller U-10. Een "1" betekent "single density" en een "0" double density.

HLT: Het Head-Load-Timing-bit is eveneens bedoeld voor de floppy-disk-controller. Dit bit moet geset worden ("1" gemaakt worden) nadat de floppy-controller een kommando uitgevoerd heeft waarbij de lees/schrijfkop bediend wordt (head load). De tijd tussen het bedienen van de kop en het zetten van HLT mag beslist niet korter zijn dan de head-load-time!

MOTOR: Bit 6 is het control-bit voor de motoren in de drives. Is dit bit "1", dan worden de motoren ingeschakeld. Wanneer 8"-loopwerken worden gebruikt, moet er een doorverbinding tussen J6, pen 16 en J5, pen 4 worden gelegd. De spindelmotoren van dit soort loopwerken draaien doorgaans op 110 V of 220 V wisselspanning en moeten via een relais geschakeld worden. De aansturing van dit relais kan direkt door het MOTOR-bit geschieden.

BELL: Dit bit bestuurt de piëzo-zoemer. Is het "1", dan wordt de 555-timer (U-86) geactiveerd en deze produceert een 2-kHz-toon.

Ter completering volgt in figuur 2 een korte assembler-routine voor het zetten en resetten van de afzonderlijke bits van SYS1.

SYS2 system-control-port 2

SYS2 (U-13) is de tweede control-poort. Deze bestuurt de vier floppy-disk-selectlijnen, de floppy-disk-test-pen en de adressen voor de DMA-ready-multiplexer (U-12). Evenals SYS1 is ook SYS2 een write-only-port.

DMA-ready-MUX-adressen: De laagste drie bits van SYS2 zijn de DMA-multiplexer-adressen. Figuur 3 laat zien welke periferie-chips naar de RDY-aansluiting van de DMA-controller U-62 (schemablad 2) worden doorgeschakeld.

TEST: Bit 3 is met de TEST-ingang van de floppy-controller U-10 (schemablad 5) verbonden. Is dit bit "0", dan worden alle interne tijdsvertragingen in de floppy-controller op minimum ingesteld. Normaliter is TEST "1". Alleen bij snelle track-wisselingen (track seek) wordt TEST kortstondig "0".

DS0...DS3: Bit 7...4 (let op: volgorde is omgekeerd!) zijn de vier drive-selectlijnen. Met een beetje knutselen in de ROM-monitor en met een multiplexer aan deze lijnen kunnen maximaal 16 drives geselecteerd worden.

System-status-port

De system-status-port kan alleen gelezen worden; deze ligt op I/O-adres 04C. Hierop kan men informatie uitlezen over de toestand van de ontvanger-ingangen van de SIO's (RxD_A, RxD_B), het index-gat van de floppy-interface (INDEX), het strobesignaal van het keyboard (KBDSTB) en de stand van de DIP-switches SW1 (SW1.1...SW1.4). Figuur 4 toont de indeling van de status-port.

Keyboard-port

De keyboard-port bevindt zich op I/O-adres 0D0 en kan alleen gelezen worden. De polariteit van de strobe-puls is hier niet van belang. Na een reset wordt de keyboard-strobe gelezen via bit 3 van de system-status-port. De CTC (Counter/Timer Circuit) U-37 (schemablad 6) wordt zodanig geprogrammeerd dat op de andere keyboard-flank een interrupt gegeven wordt. De keyboard-data worden met het KBDCLK-signaal (afkomstig van de CTC) in U-1 ingelezen.

Opgelet! Het keyboard wordt 8 bits breed gelezen. Beschikt het keyboard maar over 7 bits, dan dient het achste ingangsbij (bit 7) aan massa gelegd te worden!

De EPROM-programmeerpoort

Deze poort is terug te vinden in schema-blad 3 (U-41) en bevindt zich in het I/O-bereik op adres 0C0. Naar deze poort kan alleen geschreven worden.

OUTEN = OUT-ENABLE en WRITE

GATE: Bit 0 van deze poort bedient de multiplexer U-42. Is dit bit "1", dan wordt via uitgang Y2 van U-42 de databusbuffer U-57 geactiveerd en wordt tevens een "1" op de OE-aansluitingen van de ROM-voetjes gezet (te gebruiken bij statische RAM's en bij programmering van EPROM's en EEROM's).

Is bit 0 "0", dan wordt de STD-bus van de Big-Board II ingeschakeld en de signalen SMEMRD en SMEMWR worden vanuit de PAL U-34 (schemablad 7) via U-57 met de OE- en WE-aansluitingen van de ROM-voetjes doorverbonden (zie ook bij BB II: ROM-monitor).

DECODE = ROM-CHIP-ENABLE-dekoder. Is dit bit "0", dan wordt dekoder U-55 (chip-select-dekoder) ge-enabled. Ook dit bit wordt gebruikt bij het programmeren van E(E)PROMS. Programmeert men een 2716, dan wordt de CS-aansluiting "0" gemaakt terwijl pen 21 aan V_{pp} komt te liggen. Voor het programmeren van een 2732 wordt CS juist "1" gemaakt. Via de databus (BD0...BD7) gaan de te programmeren data naar de EPROM (via U-57).

PGM = EEPROM-programmeer-control-bit: Bit 3 dient voor het programmeren van 48016-EEPROM's. Om één byte te programmeren moet PGM 20 ms "1" zijn, terwijl pen 21 aan V_{pp} ligt en pen 20 (CS) "1" is. Voor het wissen van een 48016 dient PGM 200 ms "1" te zijn, moet pen 21 aan V_{pp} liggen en moet pen 20 "0" zijn. V_{pp} is een 25-V-programmeerspanning die via jumpers aan de ROM-sockets aangebo-

5

^G	Bell
^H	Cursor Left (Backspace)
^I	Horizontal Tab
^J	Cursor Down (Linefeed)
^K	Cursor Up
^L	Cursor Right
^M	Carriage Return
^Q	Clear to End of Screen
^X	Clear to End of Line
^Z	Clear Screen and Home
^ä	Escape Sequence Lead-in (see below)
^^	Cursor Home
^_	Display Control Character

ESC = row col	Cursor Addressing
ESC Q	Insert Character
ESC W	Delete Character
ESC E	Insert Line
ESC R	Delete Line
ESC X	Clear Screen and Home Cursor
ESC :	Clear Screen and Home Cursor
ESC T	Clear to End of Line
ESC t	Clear to End of Line
ESC Y	Clear to End of Screen
ESC y	Clear to End of Screen
ESC G attr	Set Attribute
ESC M mode	Set Graphics Mode
ESC . byte	Set Parameter Byte

den wordt. Hoe dit gedaan wordt, staat beschreven in het hoofdstuk "konnektor-indeling".

VPPENA = programmeerspanning-enable-bit. Wanneer dit bit "1" is, dan wordt de programmeerspanning naar de geselecteerde ROM-voet doorgeschaald via U-56 en de transistortrapjes (schemablad 5). Vóórdat VPPENA en OUTEN "1" worden gemaakt, moet het DECODE-bit juist zijn en moeten de data en de adressen in de latches U-57, U-59 en U-60 opgeslagen zijn.

STD-BB = Bus mapping-control-bit: Na een RESET wordt dit bit altijd "0" gemaakt. Geheugens die aangesloten zijn op de STD-bus zijn niet meer toegankelijk voor de processor. Zijn dit bit en het D-S-bit echter "1", dan is de 64-K-geheugenbank via de STD-bus toegankelijk.

Het video-display

Het video-display van de Big-Board II wordt door een output-programma in EPROM bestuurd. Dit programma is het in de USA als standaard gebruikte ADM-3A-terminal-programma. Hierdoor is de Big-Board II compatible met het tekstverwerkingssysteem "WordStar". Dit programma kent alle control- en escape-kommando's die in figuur 5 zijn aangegeven.

De escape-kommando's beginnen met de cursor-adressering ESC = row col. "Row" is hierbij het regel-adres (00...17_{hex}) en "col" het gewenste kolom-adres (00...4F) van de cursor-positie. Voordat het cursor-adresseringskommando uitgevoerd wordt, moet eerst nog 20_{hex} bij het cursor-adres opgeteld worden.

Met het kommando ESC G attr kan men elk gewenst teken op het beeldscherm voorzien van een "attribute" (normal ASCII, reverse video-ASCII, normal graphics, enz.). De variabele "attr" kan de waarde 0...F aannemen, dus er zijn maximaal 16 attributes mogelijk. Deze attributes worden gegeven in figuur 6a.

Wanneer men voor "attr" een waarde kiest tussen 8...F, kunnen m.b.v. ESC M mode een viertal graphic-modes worden gekozen: 0...3. Deze modes zijn te vinden in figuur 6b.

Figuur 6c laat een aantal mogelijkheden zien voor het kommando ESC .byte. "Byte" mag een waarde tussen 0 en 7 hebben.

Opwekking van de kloksignalen

Nadat we eerst de verscheidene control-kodes van het video-display hebben doorgenomen, zullen we ons nu werpen op de video-hardware (schemablad 4). De dot-klok wordt opgewekt door de inverter-oscillator rond U-43. Y2 is een kristal van 16 MHz, een courante waarde. Vanuit deze dot-klok wordt eveneens de character-klok voor de 6845-CRT-Controller en het kloksignaal voor het video-schuifregister in de 8002A-VAC (Video Attribute Controller) afgeleid. Wanneer de karakters in een 7 x 9-matrix worden afgebeeld, moet de dot-klok door 9 gedeeld worden, terwijl bij een 5x7-matrix dit deeltal 7 bedraagt. De deelfactor wordt ingesteld met behulp van jumpers op JB13 (pen 12 en 13 van

6a

	0	Normal ASCII	
	1	Underlined ASCII	
	2	Blinking ASCII	
	3	Underlined, Blinking ASCII	
	4	Reverse Video ASCII	
	5	Reverse Video, Underlined ASCII	
	6	Reverse Video, Blinking ASCII	
attr =	7	Reverse Video, Underlined, Blinking ASCII	
	8	Normal Graphics	
	9	Underlined Graphics	
	A	Blinking Graphics	
	B	Underlined, Blinking Graphics	
	C	Reverse Video Graphics	
	D	Reverse Video, Underlined Graphics	
	E	Reverse Video, Blinking Graphics	
	F	Reverse Video, Underlined, Blinking Graphics	

65029-6a

b

	0	Block Graphics Mode	
mode =	1	Bar Graphics Mode #1	
	2	Bar Graphics Mode #2	
	3	ASCII Character Set Mode	65029-6b

c

	0	No auto-newline, normal scroll, no clock display	
	1	Auto-newline, normal scroll, no clock display	
	2	No Auto-newline, no scroll, no clock display	
	3	Auto-newline, no scroll, no clock display	
	4	No auto-newline, normal scroll, clock is displayed \times	
	5	Auto-newline, normal scroll, clock is displayed \times	
	6	No auto-newline, no scroll, clock is displayed \times	
	7	Auto-newline, no scroll, clock is displayed \times	

XNB: Clock Display is not implemented in Version 1.0 of BB2RM.

65029-6c

U-45). Als geen jumper is aangebracht betekent dat een deeltal van 7.

De 6845-CRT-controller

De 6845 verzorgt de adressen voor de karakters op het beeldscherm, de cursor, de sync-pulsen en diverse andere signalen. Ook is lichtpenbesturing mogelijk. De timing van het beeldscherm wordt vastgelegd d.m.v. 18 programmeerbare registers, die bereikt kunnen worden via een indirect adresregister op poort ODC. Is dit adresregister geladen met een bepaalde waarde, dan wordt dit ge-indexed weergegeven. Op die manier krijgen we een soort "register-file", waarvan de posities in figuur 7 zijn afgebeeld. Wie meer wil weten over de werking en de programmering van de 6845, kan zich verdiepen in de Elektuur-uitgave "Paperware 3".

Het video-geheugen

Het video-geheugen van de Big-Board omvat 4 K, waarvan 2 K het beeldscherm-geheugen vertegenwoordigen (6000...67FF). De andere helft

Figuur 6. Er kunnen zowel ASCII-karakters als graphics weergegeven worden. Elk karakter kan voorzien worden van een "attribute" (figuur 6a). Wanneer een van de acht onderste attributes wordt gekozen, schakelt de video om naar graphics. Er zijn dan 4 verschillende modes mogelijk (figuur 6b). Voor de video-uitgave zijn er 8 parameters (figuur 6c).

Figuur 7. De interne registers van de 6845-CRT-controller. De data in deze registers bepalen de diverse tijd-instellingen van het video-display.

Figuur 8. De "attributes" van een teken dat op het scherm moet worden afgebeeld, worden opgewekt door de 8002-attribute-controller met behulp van de data in de attribute-RAM. Figuur 8a en 8b laten zien hoe een karakter in de twee verschillende graphics-modes wordt weergegeven. In de externe graphics-mode worden de bits van een teken-byte vertaald in verticale streepjes met de breedte van een beeldpunt (8c). Figuur 8d laat zien dat ook in de graphic-mode bit 2...6 in het attribute-byte het attribute van het teken bepalen.

(7000...77FF) dient voor het opslaan van de attribute-gegevens. Het D-S-bit van de control-port behoort "0" te zijn bij access in het video-RAM-gebied. De 3 multiplexers U-47, U-48 en U-49 selecteren de adreslijnen van ofwel de Z80 ofwel de 6845. Normaliter adresseert de 6845 het videogeheugen, maar wanneer de processor of de DMA-controller in het desbetreffende geheugenbereik werken, worden de multiplexers omgeschakeld. Dit gebeurt op kommando van de lijnen CHARCE (CHAracter Chip Enable) en ATTCE (ATTrIBUTE Chip Enable). Is of de ene, of de andere "0", dan heeft de DMA-controller of de CPU toegang tot de video-RAM. Tevens wordt dan flipflop U-32 gereset, zodat het beeldscherm gedoofd wordt. Dit voorkomt hinderlijke beeldruis. De databus wordt over beide RAM's verdeeld door ofwel buffer U-54 te kiezen (met CHARCE) ofwel buffer U-55 (met ATTCE). Het BRD (Bus Read)-signaal bepaalt of er geschreven danwel gelezen wordt.

De attribute-controller

De video-attributes worden verzorgd door U-52, de video-attribute-controller. Dit IC zet elk ASCII-teken dat in het karakter-geheugen staat, om in een puntpatroon. Dit patroon maakt deel uit van een 5 x 7- of een 7 x 9-matrix. Elk byte in de karakter-RAM kan voorzien worden van vijf verschillende attributes: reverse video, blink, underline, strike-through of straight. Bit 0 en bit 1 van het attribute-byte bepalen of we te doen hebben met een ASCII-teken of een graphics-teken. Bit 2...6 bevatten de gekozen attribute-informatie.

Zijn bit 0 en bit 1 in de attribute-RAM "1", dan is het byte in de karakter-RAM een ASCII-teken. Wanneer echter niet beide bits tegelijkertijd geset zijn, dan hebben we te doen met een graphics-karakter. Figuur 8a...8c laten zien, hoe bit 0 en bit 1 de samenstelling van een graphics-teken bepalen. In figuur 8a staat de bitsamenstelling in de zgn. "block-graphics-

8a

D7	D3	
D6	D2	Attribute Data Bit #1 = 0
D5	D1	Attribute Data Bit #0 = 0
D4	D0	

65029-8a

b

:	D	
:	D	Attribute Data Bit #1 = 1
2	:	Attribute Data Bit #0 = 0
:	:	
---	D1	---

65029-8b

c

(7 X 9)	(5 X 7)	
765432107	7654321	
765432107	7654321	
765432107	7654321	Attribute Data Bit #1 = 0
765432107	7654321	Attribute Data Bit #0 = 1
765432107	7654321	
765432107	7654321	

65029-8c

d

Bit 2 = Strike-Through
 Bit 3 = Underline
 Bit 4 = Reverse Video
 Bit 5 = "Character" Blank
 Bit 6 = "Character" Blink

65029-8d

7

- 0 = Horizontal Total -1
- 1 = Number of Characters per Line
- 2 = Horizontal Sync Position -1
- 3 = Bits 4-7 = Vert Sync Width, Bits 0-3 = Horiz Sync Width
- 4 = Number of Rows - 1
- 5 = Vertical Total Adjust
- 6 = Vertical Displayed
- 7 = Vertical Sync Position -1
- 8 = Interlace And Skew
- 9 = Number Of Scans Per Character
- A = Cursor Start-Raster Register
- B = Cursor End-Raster Register
- C = Start Address Register (Low)
- D = Start Address Register (High)
- E = Cursor Register (Low)
- F = Cursor Register (High)
- 10 = Light Pen Register (Low)
- 11 = Light Pen Register (High)

65029-7

mode". Eén blok is dan opgebouwd uit acht afzonderlijke blokjes. In de "fine-graphics-mode" (figuur 8b) worden alleen bit 0, 1 en 2 gebruikt, die elk een lijntje voorstellen. Figuur 8c laat de "external-graphics-mode" zien. Elk bit in de karakter-RAM vertegenwoordigt dan een vertikaal lijntje in een 5 x 7-matrix of in een 9 x 7-matrix. tenslotte worden in figuur 8d de mogelijke attributes met de bijbehorende bits in de attribute-RAM

Seriële I/O

Er is voorzien in twee seriële poorten, SIO-A en SIO-B (schemablad 6). In principe zijn beide poorten identiek, alleen heeft SIO-B een interne baudrate-generator terwijl SIO-A met een externe klok gestuurd moet worden. Deze laatste kan gebruikt worden voor communicatie met modems, die in de regel over een eigen klok beschikken. SIO-B blijft dan over voor andere toepassingen, bijv. verinding met een printer of een terminal. De CTC U-21 funktioneert dan als interne baudrate-generator. De baudrate kan via

software ingesteld worden. In figuur 9 kan men zien, hoe een bepaalde baudrate in het CTC-register kan worden geplaatst.

Floppy-disk-interface

Alle floppy-elektronica is afgebeeld in schemablad 5. Als controller kunnen alle 17XX-typen dienst doen, maar deze geven bij double density af en toe wel eens een lees- of schrijffout. Aan te bevelen is de M8877A van Fujitsu, die pin-compatible is met de 1793- of de 1797-controller. Een aantal taken die normaliter door de controller worden uitgevoerd, worden nu door de CPU verzorgd, zoals SIDE-SELECT en HEAD-LOAD-TIMING. Zodoende kunnen vrijwel alle verkrijgbare loopwerken gebruikt worden. De controller is op de I/O-databus aangesloten. De klok is afkomstig vanuit de 8-MHz-systeemklok. Daar de controller zowel op 1 als 2 MHz kan "draaien", kunnen zowel 50 als 8"-loopwerken toegepast worden. Het programmeren van de controller verloopt via de poorten OD4...OD7. In figuur 10 staan de diverse registers aangegeven. Het resetten van de controller geschiedt via de control-port (U-41, schemablad 3). Het signaal FCRST behoort voor een juiste reset minstens 50 µs laag te zijn.

SASI-interface en overige parallelle poorten

Naast de keyboard-port en de seriële poorten beschikt de Big-Board nog over een achttal parallelle poorten die elk acht bits breed zijn. Alle werken met handshake-signalen, hoewel het ook mogelijk is data zonder handshaking te verwerken. Data-overdracht naar een hard-disk (Winchester) loopt via de SASI-interface. De eerste vier poorten worden opgeëist door deze interface; de resterende poorten staan vrij ter beschikking voor de gebruiker.

SASI-data-output-port 0

Outputport 0 is de SASI-uitgang en deze bevindt zich op adres OD8. De I/O-databus wordt ge-latched in U-93 (schemablad 8). De uitgangen hiervan worden gebufferd m.b.v. de open-kollektor-buffers U-88 en U-89.

SASI-data-input-port 0

Via ingangspoort 0 worden de data van de SASI-bus ingelezen. Deze poort bevindt zich, net als de output-port, op adres OD8 en haalt de data via buffer U-94 naar binnen.

SASI-control-output-port 1

Deze poort dient voor het uitlezen van diverse control-signalen op de SASI-bus en ligt op adres OD9. Via U-96 worden de diverse signalen, waaronder ook SEL en RST, naar buiten uitgevoerd. Het signaal SEL wordt actief als de waarde 0B naar de poort OD9 wordt geschreven. Schrijft men de waarde 03, dan wordt SEL weer inactief gemaakt. Eenzelfde afhandeling geldt ook voor RST met de respectievelijke waarden 0A en 02.

SASI-input-port 1 of SASI-status-port

Deze poort bevindt zich op adres OD9 en is bedoeld voor het inlezen van de status

9 BAUD RATE CONSTANTS

```
01H = 38.2 KBAUD
02H = 19.2 KBAUD
04H = 9600 BAUD
08H = 4800 BAUD
10H = 2400 BAUD
20H = 1200 BAUD
40H = 600 BAUD
80H = 300 BAUD
```

SETTING THE BAUD RATE

```
LD A,47H ;COUNTER MODE, LD TC, RESET
OUT (088H),A ;88H FOR B CHAN, 89H FOR A CHAN
LD A,02H ;19.2 KBAUD CONSTANT (FROM TABLE ABOVE)
OUT (088H),A ;SEND TIME CONSTANT TO CTC
```

BAUD RATES BELOW 300 ARE POSSIBLE BY USING THE CTC IN THE TIMER MODE.

```
104 DECIMAL = 150 BAUD
142 DECIMAL = 110 BAUD
208 DECIMAL = 75 BAUD
```

```
LD A,0000111B ;TIMER MODE, /16, LD TC, RESET
OUT (089H),A ;USE A CHAN THIS TIME
LD A,142 ;142 DECIMAL FOR 110 BAUD
OUT (089H),A ;USE 088H FOR B CHAN
```

65029-9

10

```
0D4H = STATUS (READ) / COMMAND (WRITE)
0D5H = TRACK
0D6H = SECTOR
0D7H = DATA
```

65029-10

11

```
BIT 7 = -ODAV2 (Low when data is available at Output Port 2)
BIT 6 = -IDAV1 (Low when data is available at Input Port 1)
BIT 5 = -C/D ("SASI" control signal "Control/Data")
BIT 4 = -IDAV2 (Low when data is available at Input Port 2)
BIT 3 = -MSG ("SASI" control signal "MESSAGE")
BIT 2 = -ODAV1 (Low when data is available at Output Port 1)
BIT 1 = -BUSY ("SASI" control signal "BUSY")
BIT 0 = -I/O ("SASI" control signal "INPUT/OUTPUT")
```

65029-11

van de SASI-interface. In figuur 11 zijn deze status-flags en de bijbehorende bitposities te zien.

User-output-port 2

Via adresplaats 0DA staat de gebruiker een uitgangspoort ter beschikking. Data die naar deze poort worden geschreven, worden opgeslagen in latch U-103. Omdat hier met handshaking gewerkt wordt, worden bij een schrijfoperatie de volgende zaken afgehandeld:

- 1) De CPU schrijft een byte naar de output-port 2.

Figuur 9. De Big-Board II heeft vier seriële kanalen (SIO's). De baudrate wordt bepaald door de CTC, waarvan de registers vóór elke seriële overdracht moeten worden geladen met parameters. De waarden van de parameters voor diverse baudrates zijn hier gegeven.

Figuur 10. De floppy-controller wordt gestuurd door de I/O-bus van de Big-Board II. Via deze vier I/O-adressen heeft de CPU toegang tot alle registers in de floppy-controller.

Figuur 11. De inhoud van de SASI-statusport, die acht flags bevat. Dit zijn niet alleen de status-bits voor de sturing van de hard-disk-interface, maar ook de "normale" status-flags voor de Centronics-printer.

2) Pen 17 van bus J11 wordt "0" (data-strobe) en laat het aangesloten apparaat (bijv. een printer) weten dat er geldige data beschikbaar is.

3) Bit 7 van de SASI-status-port (ODAV2, zie figuur 11) wordt "0" en geeft de processor hiermee te kennen dat het aangesloten apparaat de data nog niet aangenomen heeft.

4) Het lezen van data door een apparaat gaat via twee signalen:

— OE2 (Output Enable, pen 20 van bus J11) moet laag zijn, zodat de poort zich niet meer in third-state bevindt.

— TAKE2 (pen 19 van bus J11) moet "0" worden nadat het apparaat de data heeft gelezen. TAKE2 is dus gelijk aan de bij een Centronics-interface gebruikte "acknowledge". Verder wordt door TAKE2 ODAV2 (SASI-status-port) geset, waardoor de processor weet dat de data gelezen is en dat nieuwe data naar de poort geschreven kunnen worden. Wanneer de DMA-controller data naar deze poort schrijft, mag TAKE2 niet langer duren dan 1 μ s.

User-input-port 2

Evenals de output-port 2 bevindt zich ook deze poort op I/O-adres 0DA. Er kan met en zonder handshake gewerkt worden. In het laatste geval moet de latch-enable-lijn "0" zijn (pen 19 van bus J10). De CPU leest dan het momentele bitpatroon op de poort in. Met handshake gaat dat als volgt:

1) Het aangesloten apparaat zet een byte op de ingangspoort.

2) Is deze data stabiel, dan zendt het appa-

raat een data-strobe via GIVE2, waarmee de data opgeslagen wordt in latch U-104 op de omlaaggaande flank. Op de omhooggaande flank wordt de data-available-flag (IDAV2) geset ("0" gemaakt). Hiermee wordt de processor te kennen gegeven dat er geldige data in de latch staat.

3) Na het lezen door de processor wordt het IDAV2-sigitaal weer gereset ("1"). De lees-cyclus is dan beëindigd.

User-output-port 3

In feite is deze poort kwa functie gelijk aan output-port 2. De data worden via U-98 naar buiten gestuurd (bus J9), met status-bit ODAV1 voor de handshake-besturing. Poort 3 bevindt zich op I/O-adres 0DB.

User-input-port 3

Ook deze ingangspoort werkt identiek als input-port 2. De handshake-status wordt weergegeven door IDAV1 (bit 6 van de SASI-status-port). De ingangen (bus J8) worden via U-99 naar binnen gevoerd. Het adres van de poort ligt op 0DB.

opgelet:

De SASI-status-flags worden op de uitgangsbuss (JB9) geïnverteerd naar buiten uitgevoerd. In figuur 12 staan deze signalen ter verduidelijking nog eens bij elkaar. Wanneer data-kommunikatie via de SASI-interface gepleegd moet worden (bijv. met een Winchester-drive), dan moet pen 5 van JB9 verbonden worden met pen 5 van JB7.

12

JB9-1 = -IDAV1

JB9-2 = -IDAV2

JB9-3 = -ODAV1

JB9-4 = -ODAV2

JB9-5 = DMAREQ (SASI TRANSFER REQUEST)

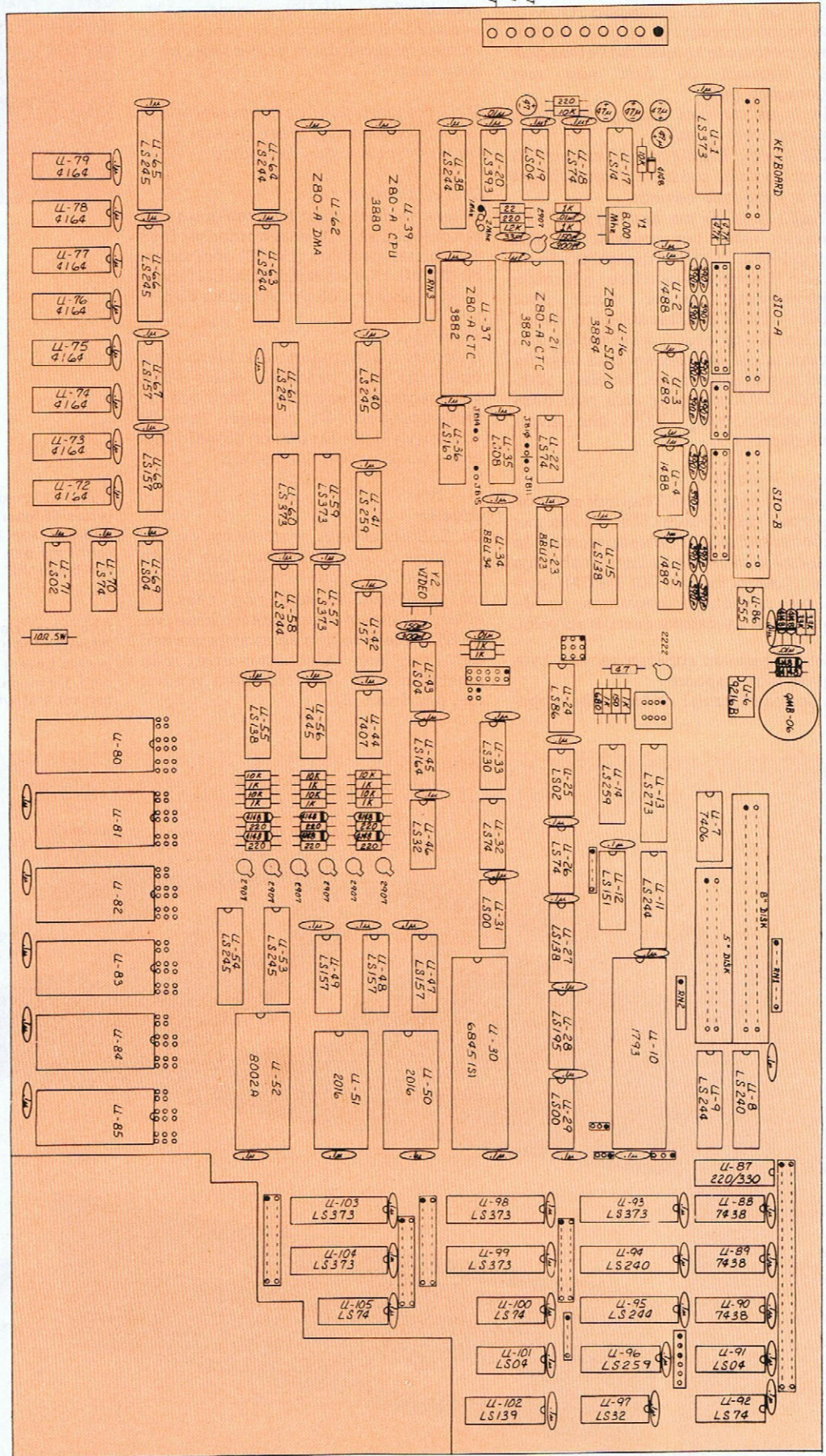
65029-12

Figuur 12. Enkele SASI-statusbits zijn (geïnverteerd) naar konnektor JB9 uitgevoerd. Denk er aan dat de SASI-interface alleen werkt als pen 5 van JB9 is verbonden met pen 5 van JB7.

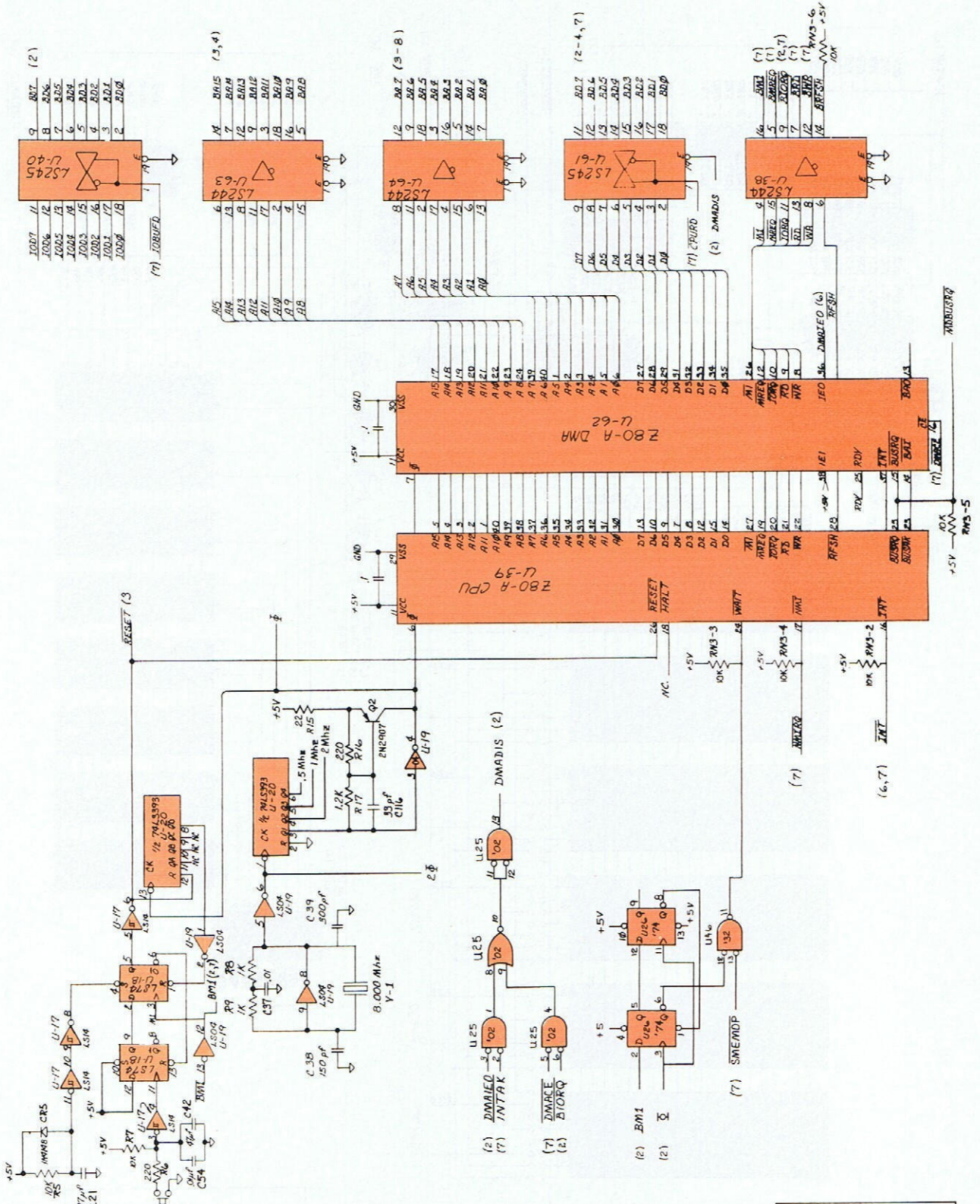
Foto. En zo ziet ie eruit: de Big-Board II met 256-K-DRAM (links boven), XEBEC-controller (rechts boven) en hard-disk BASF 6188 (midden). De mogelijkheid bestaat om het geheugen tot 1 Mbyte DRAM (!) uit te breiden.



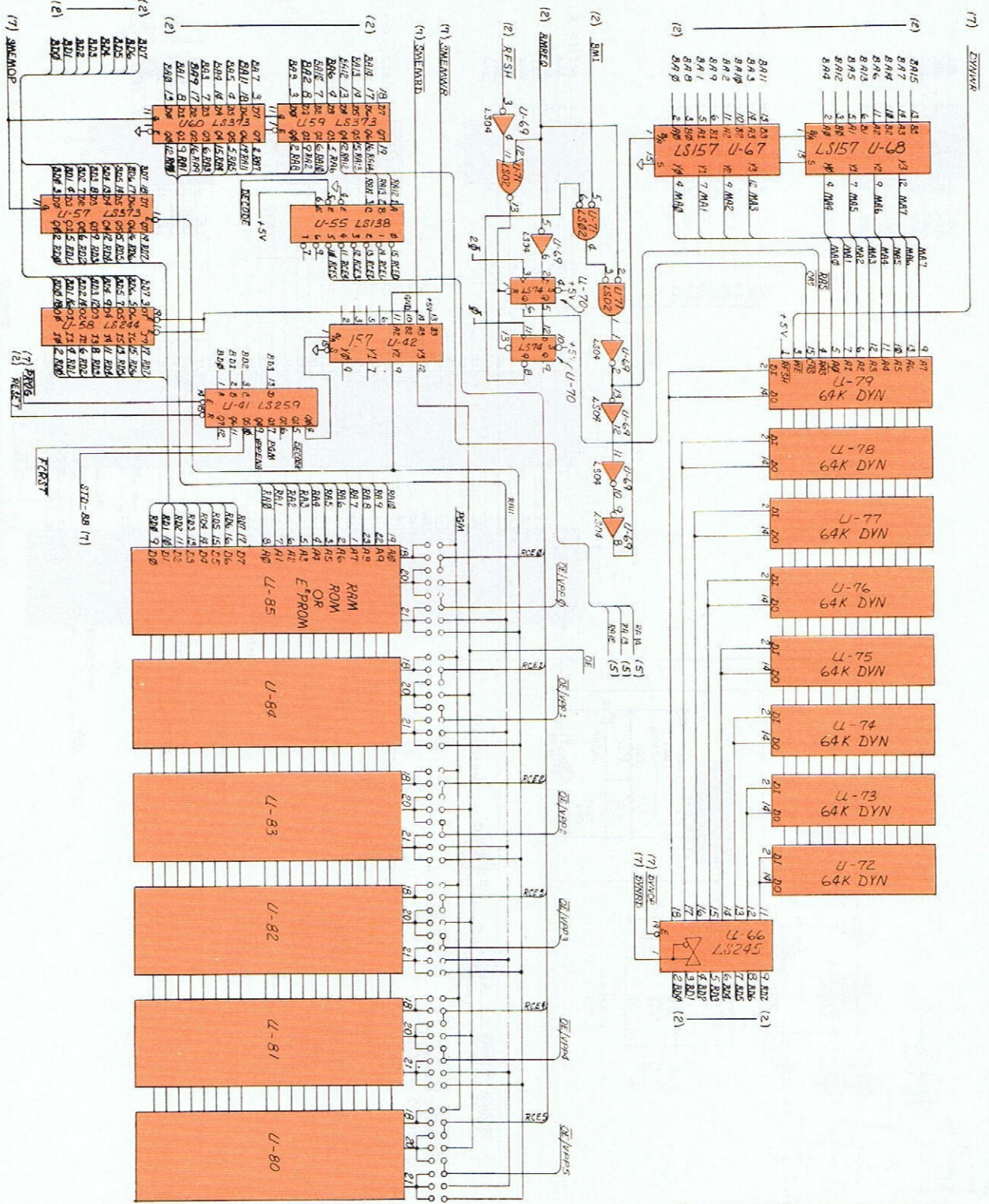
COMPONENT LAYOUT
 JUNE 1983 SHEET 1 OF 8
 J. B. FERGLSON INC.



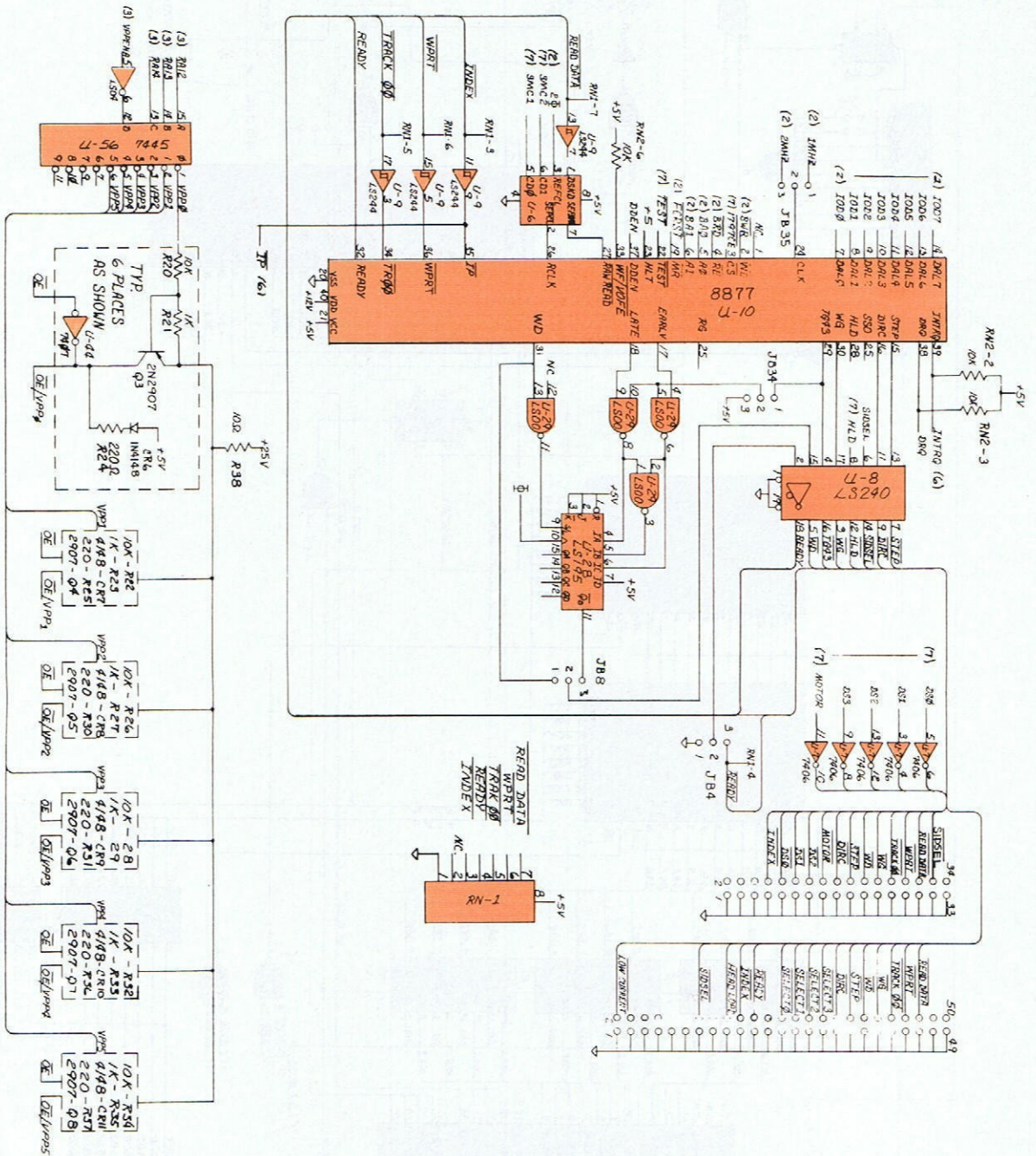
-12V
 GND
 +5V
 +12
 VPP
 SSR
 RST
 RST RET
 LT PEN
 LT RET

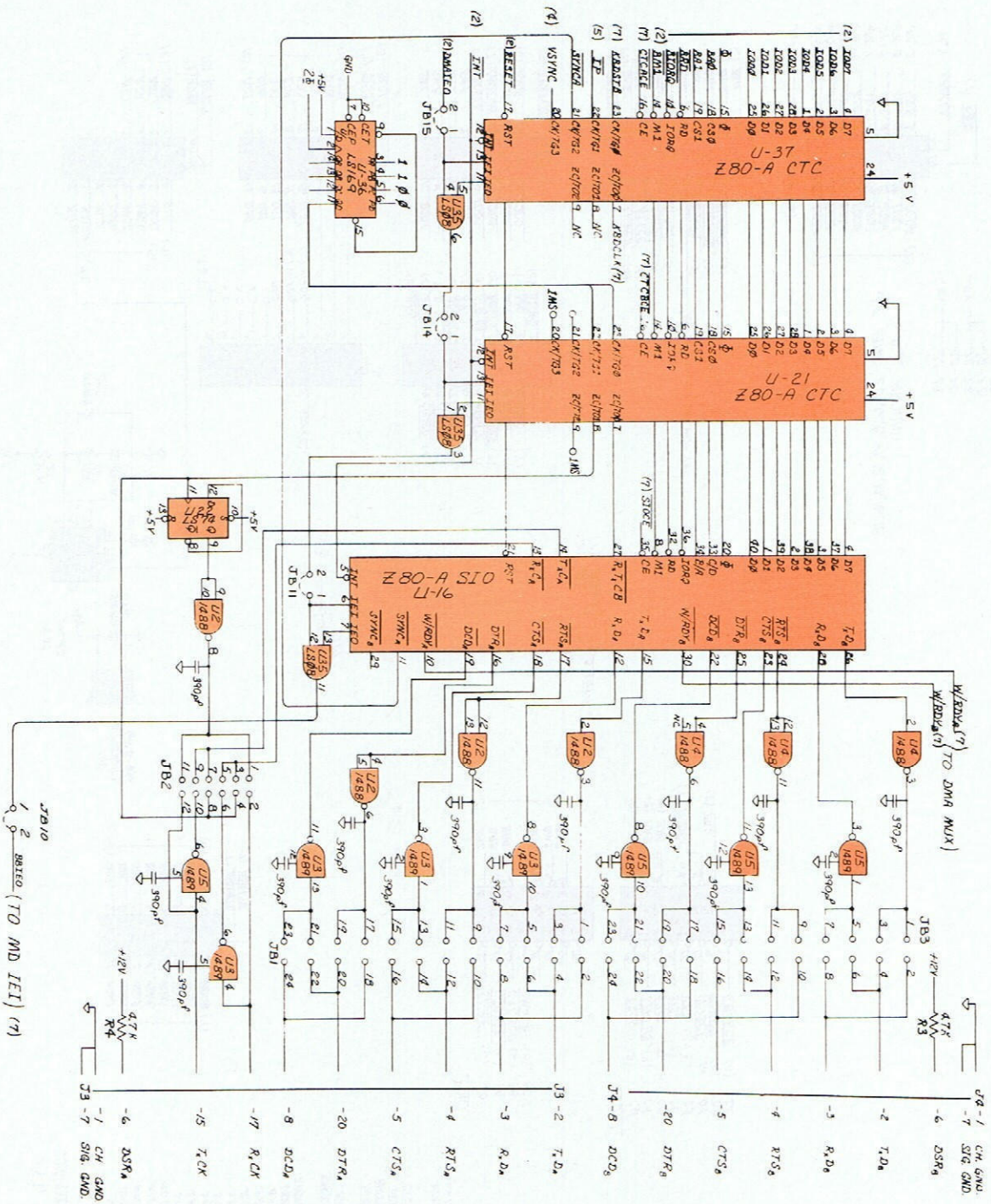


C.P.U.	
JUNE '83	SHEET 2 OF 8
J.B. FERGLISON INC.	



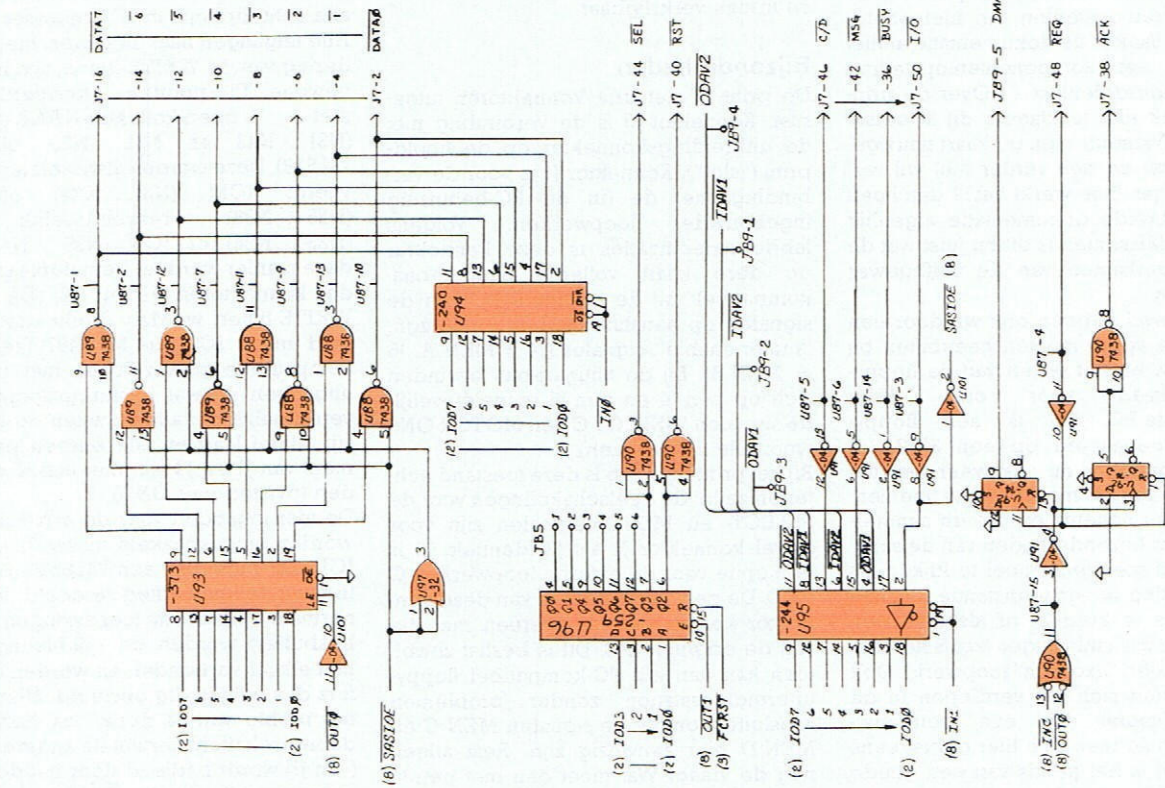
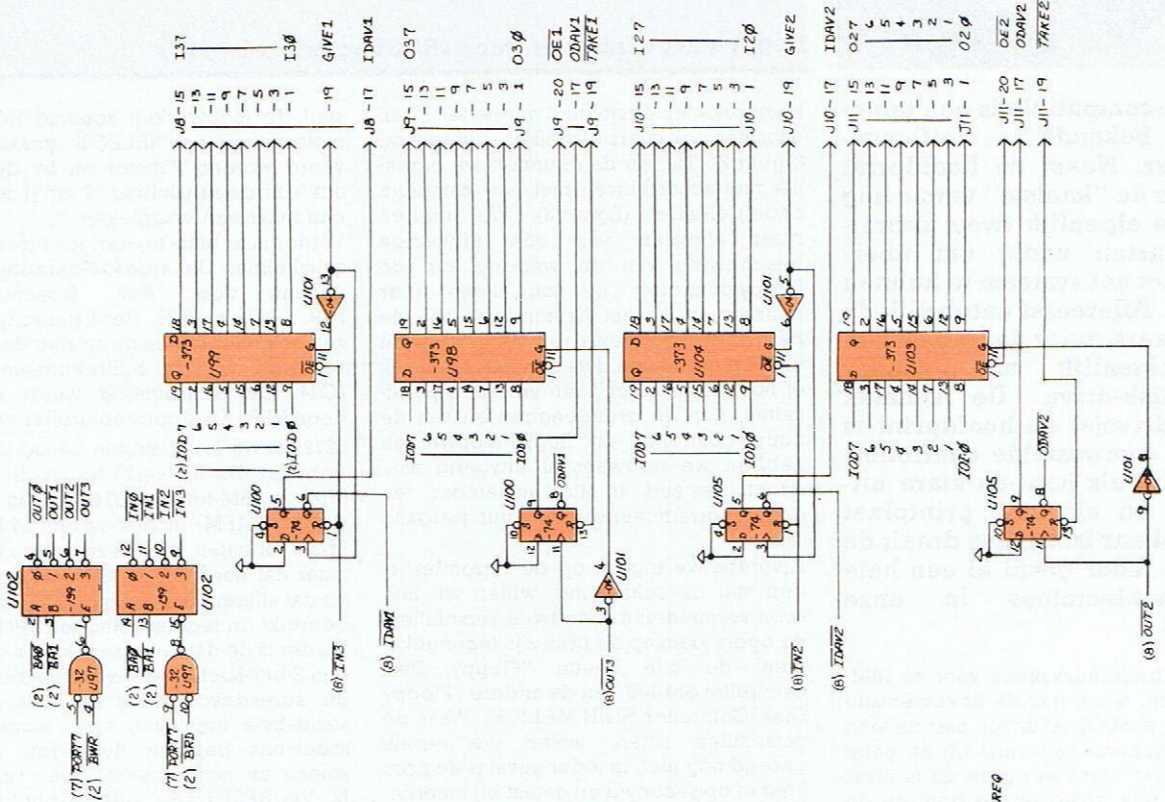
MEMORY	
NOV. 1981	SHEET 3 OF 8
J. B. FERGUSON INC.	





SERIAL I/O

JUNE 1983	SHEET 6 OF 8
J. B. FERGUSON INC.	



SASI AND PARALLEL
INTERFACE
MAY 25, 1982 SHEET 8 OF 8
J. B. FERGUSON INC.