

Een professioneel systeem voor de prijs van een home-computer

De EC-68 is een klein maar krachtig computersysteem, dat ook voldoet aan professionele eisen. Dit systeem biedt een bijzonder gunstige prijs/kwaliteitsverhouding als we kijken naar de beschikbare hoeveelheid software en de daaruit voortvloeiende gebruiksmogelijkheden. Een klein systeem met zeer veel mogelijkheden dus. Dat zult u na het lezen van de diverse Flex-artikelen in deze EC beslist met ons eens zijn.

Zoals de titel al laat zien, werkt de EC-68 met het operating system Flex. Deze naam zal de meeste van onze lezers waarschijnlijk weinig of niets zeggen. Daarom besteden we in het software-gedeelte van deze Computing-uitgave (en in de komende nummers) de nodige aandacht aan dit interessante systeem. Op de computermarkt is, zoals we reeds hebben opgemerkt, erg veel software voor Flex verkrijgbaar, waaronder diverse bekende programma's. Maar alvorens nog meer tekst wordt gewijd aan die software, moet eerst de computer zelf worden beschreven.

De systeem-hardware

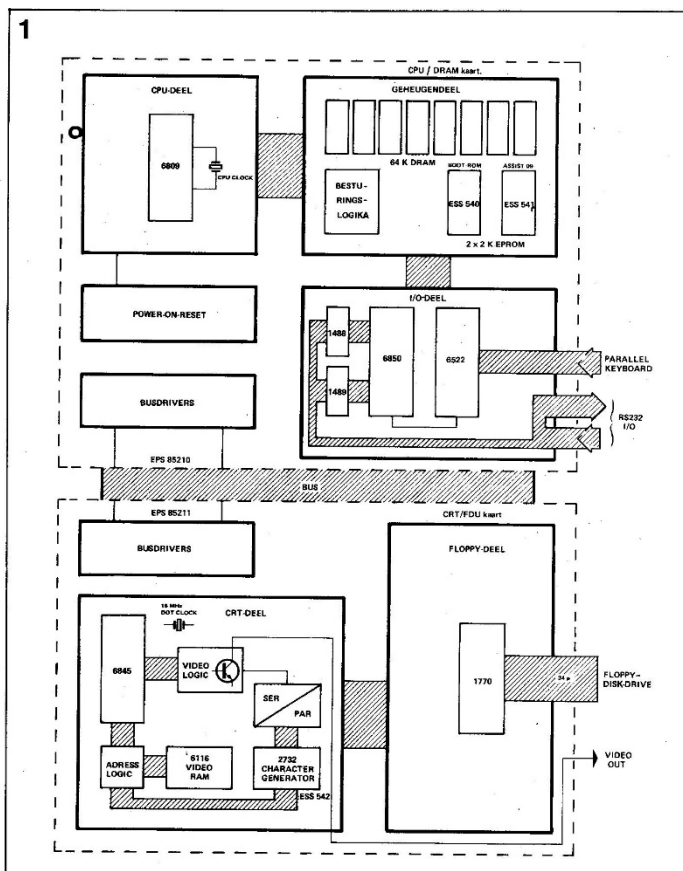
We zullen in dit gedeelte niemand vermoeden met uitgebreide verhandelingen waarbij de functie van elk weerstandje uit de doeken wordt gedaan. We volstaan met de bespreking van het uitgebreide blokschema.

Een computersysteem voor huishoudelijk of semi-professioneel gebruik bestaat altijd uit een vaste set functionele bouwstenen. Voor een Flex-computer zijn de volgende delen onontbeerlijk: een CPU-deel, een geheugendeel, een videodeel en een koppeling naar een snel magnetisch opslagmedium (een floppy-disk-interface dus). Met deze opsomming zijn ook de hoofdbestanddelen van het blokschema gegeven. Lezers die al vaker een bouwbeschrijving van een computersysteem hebben gelezen, komen, na het maken van een eenvoudig rekensommetje, tot de konklusie dat het systeem naar alle waarschijnlijkheid zal bestaan uit een viertal eurokaarten. Fout! Door een uitgekend ontwerp zijn voor de bouw van deze computer slechts twee printen van eurokaartformaat nodig. Op één print zit het CPU- en geheugendeel, op de tweede de floppy-disk-interface en het videogedeelte.

Het blokschema van de Flex-computer is gegeven in figuur 1.

Natuurlijk bestaat een computer niet alleen uit twee eurokaarten met elektronica. Verder hebben we nog nodig:

EC-68 — de Elektuur-Flex-computer



- * Een parallel-keyboard (ASCII).
- * Een monitor met 75-Ω-ingang en een minimale bandbreedte van 16 MHz.
- * Minimaal een floppy-drive met 40 of 80 tracks en een Shugart-bus. Het loopwerk moet dubbelzijdig werken, aangezien de systeemdiskette dubbelzijdig is.
- * Een netvoeding die de volgende spanningen levert: 5 V/3 A, +12 V/2 A (voor de floppy-drives), +12 V/250 mA en -12 V/250 mA (voor de RS-232-interface). Een voeding die wat meer stroom kan leveren, kan natuurlijk nooit kwaad.
- * Een bus-print met konnektors, bijvoorbeeld de Omnibus (EPS 83102).

Aangezien we hier te maken hebben met een eurokaarten-computer, gelden voor de opzet dezelfde regels als we in EC1 hebben beschreven voor de Octopus 65. Dat geldt overigens ook voor de kast waarin de EC-68 wordt gebouwd. Het

Figuur 1. Dit blokschema geeft een overzicht van de diverse functie-delen in de EC-68 en de verdeling hiervan over twee eurokaarten.

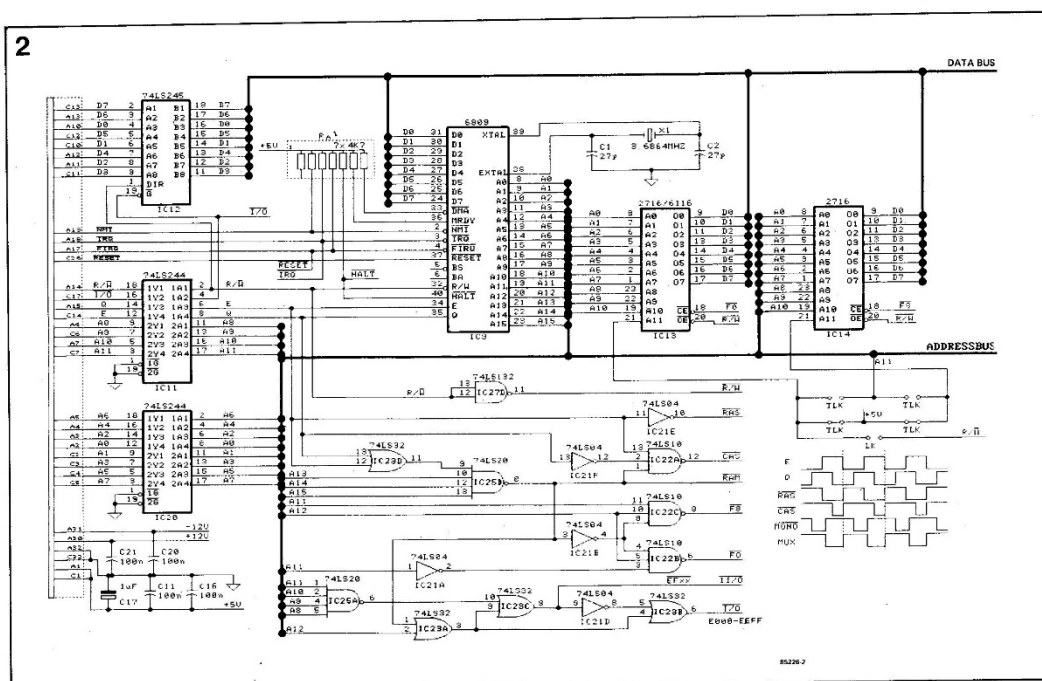
Figuur 2...4. Drie door de computer getekende schema's zijn nodig om de hele schakeling van het CPU-DRAM-gedeelte weer te geven. Het is wel duidelijk dat op de print flink met de beschikbare ruimte moest worden gewoerd.

enige verschil met de Octopus, kwa opbouw, is het gebruik van dubbelzijdige drives.

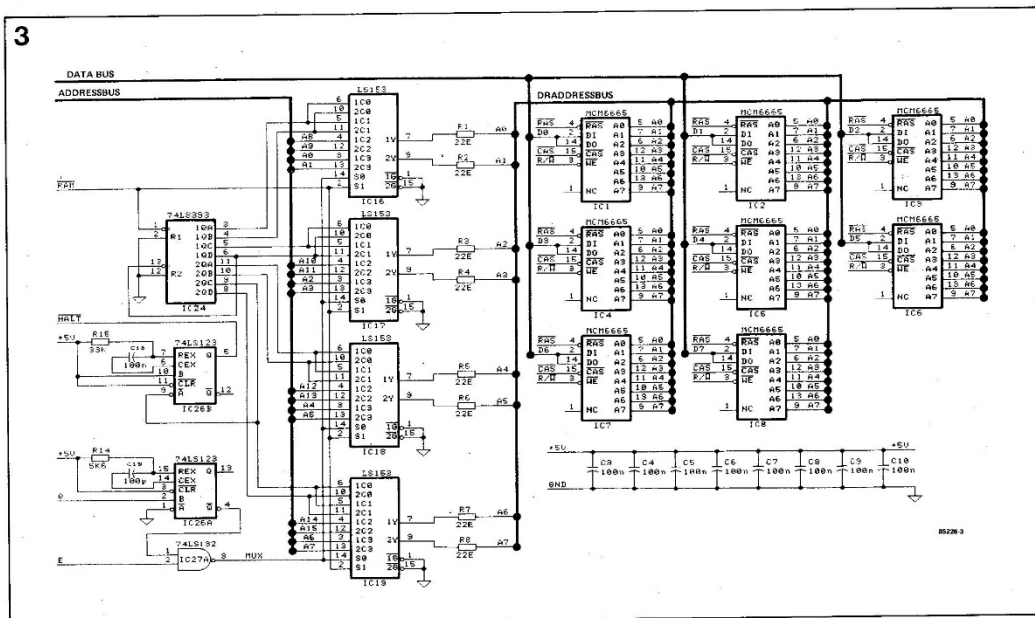
Het CPU-gedeelte

Het hart van elke computer is de microprocessor. Flex is een operating system dat is geschreven voor de 6800 en 6809.

2



3



De CPU in de Elektuur Flex-computer is de 6809 (IC9). De Flex-versie hiervoor heet dan ook toepasselijk "Flex-9". Figuur 2 toont het schema van het gedeelte rond de 6809. Zonder software kan ook de 6809 niet werken. De firmware (Assist-09 en Bootrom) bevindt zich in twee EPROM's van het type 2716 (IC13 en IC14). Standaard hebben we te maken met een systeem dat draait op een klokfrequentie van 3,6864 MHz. Men heeft hier gekozen voor deze

frequentie, omdat hiervan ook de baudrates van de ACIA worden afgeleid. Door de frequentie van de systeemklok te verdubbelen (de schakeling rond XI), het toepassen van B-types voor de 68xx-IC's, het gebruik van snellere dynamische RAM's en het veranderen van de timing van de RAM's kan men pogen over te schakelen op een 2-MHz-systeem. We sluiten de mogelijkheid van omschakeling naar een hogere systeemklok niet uit,

maar raden het op dit moment echter af vanwege de te verwachten problemen met de timing. In het begin van deze hardwarebeschrijving werd vermeld dat het hele systeem is opgebouwd rond een tweetal printen van eurokaartformaat. Het gebruik van bus-georiënteerde printen maakt het aansluiten van uitbreidingen (bijvoorbeeld RAM-disk, PROM-programmer) bijzonder eenvoudig. IC11, IC12 en IC20 zijn



Met de draadbruggen kan men kiezen tussen een 2716 of 6116 op de plaats van IC13. Het is namelijk mogelijk deze CPU/DRAM-kaart voor andere toepassingen in te zetten dan als CPU/RAM voor de Elektru Flex-computer. In onze toepassing wordt voor IC 13 een 2716 gebruikt. In het geval dat een 6116 wordt gebruikt, of een ander type EPROM, moet de adressering (en mogelijk het R/W-sigitaal) met behulp van draadbruggen aangepast worden. De in figuur 2 aangegeven draadbruggen geven de situatie weer bij gebruik van twee EPROM's van het type 2716. Over de inhoud van de twee 2716's volgt in het software-gedeelte van deze EC uitgebreidere informatie. De serie poorten dient voor het maken van een aantal chip-select-signalen en de RAS- en CAS-signalen voor de dynamische RAM's.

Zoals zojuist werd verteld, worden op de CPU/RAM-print dynamische RAM's (4164) ingezet. In figuur 3 is het DRAM-deel in tekening gebracht.

Verder was er op de eerste print nog plaats voor I/O-chips. In figuur 4 is aangegeven hoe die I/O in elkaar zit. De kom-

Kijken we nog even naar de klokkingen van de 6880 (pen 3 en pen 4 van IC15), dan zien we ook waarom er gekozen is voor zo'n merkwaardige kristalfrequentie. De VIA-poort CB1 (pen 18 van IC10) wordt door de firmware als klokdeeler geprogrammeerd, waardoor de systeemklok wordt gedeeld tot ACIA-klok. Dat kan alleen maar als de systeem-

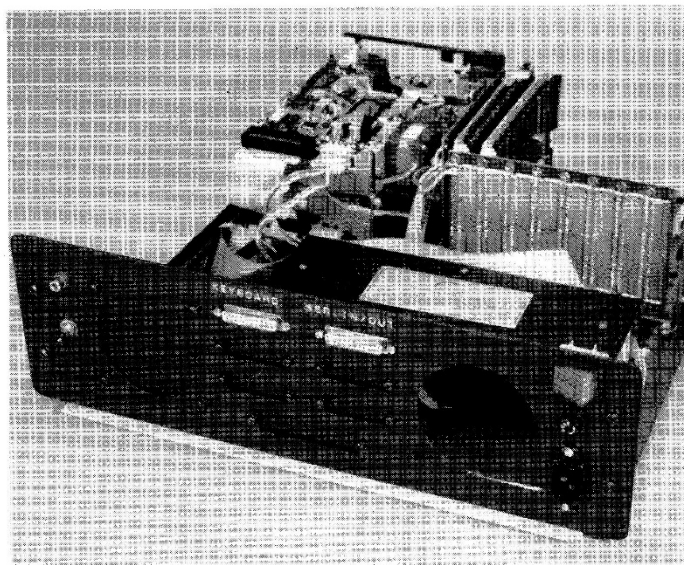
De aansluitgegevens van de 40-polige konnektor met de parallel-poorten voor keyboard en printer zijn afgebeeld in figuur 5. Figuur 6 geeft de aansluitingen van de 16-polige konnektor voor de seriële interface. In het figuurbijschrift vindt u de aansluitgegevens voor de 25-polige D-konnektor die gewoonlijk bij RS-232-interfaces wordt toegepast.

Omdat we te maken hebben met een busgeoriënteerd systeem is het niet overbodig de signalen en spanningen op de bus te geven. Zie hiervoor figuur 7. Ter afsluiting merken we nog op dat de computer bij het opstarten een systeem-reset genereert. Daarvoor is de schakeling rond IC27 (poort B en C) verantwoordelijk (zie figuur 4). U hoeft dus niet te beginnen met het drukken van de reset-knop na het inschakelen van de voedingsspanning. Feitelijk staat ons niets meer in de weg om de CPU/DRAM-print van componenten te voorzien. Hierbij moet, in verband met de grote pakkingdichtheid, wel nauwkeurig gewerkt en gesoldeerd worden. Door die grote dichtheid kon ook niet worden vastgehouden aan de typische Elektuur-layout voor de printen. (zie de print-layout met componentenopstelling in figuur 8).

Vergeet vooral de vijf draadbruggen niet! Het spreekt voor zich dat u de met gelijke letters aangegeven punten met elkaar moet verbinden (dus A met A, B met B, etc). Bij IC13 worden de draden direct aan de aansluitpennen van het EPROM-voetje gesoldeerd, waarna de overgebleven uiteinden van de draden tussen IC20 en IC21 worden vastgesoldeerd. Let hierbij goed op! Als u de print met de componentenzijde naar u toe houdt, met de buskonektor naar links, dan zijn de onderste vijf soldeerplaatsen tussen IC20 en IC21 de aansluitingen A...E. Het daarboven liggende soldeerpunt is alleen een doorgemetalliseerd gat en mag verder niet gebruikt worden! Het is verder raadzaam IC-voeten te gebruiken, alsmede de in de onderdelenlijst aangegeven konektoren. De twee EPROM's kunnen in de Elektuur Software Service geprogrammeerd worden, waarbij het linker IC (IC13 dus) moet worden voorzien van het programma Bootrom en het rechtse (IC14) van Assist-09. De ESS-nummers vindt u op de inhoudspagina onder de kolofon.

Video- en floppy-interface

Op de tweede print bevindt zich het resterende stuk elektronica dat nodig is voor het basissysteem, het videogedeelte en de floppy-disk-interface. Een onontbeerlijk IC voor het verwerken van digitale signalen tot video-beelden is de CRTC (Cathode Ray Tube Controller).



Hiervoor is de bekende 6845 genomen. Helaas zijn er echter nog wel wat meer IC's nodig om alles op de juiste manier te kunnen verwerken (zie figuur 9 en 10). Het beeldscherm is opgebouwd uit 24

regels van elk 80 tekens. Wel geteld gaan er dus op een scherm $80 \times 24 = 1920$ tekens. Ergens in de computer moet de beeldscherm inhoud opgeslagen worden. Daarvoor is een 2-Kbyte-RAM nodig. Het ligt voor de hand als 2-K-RAM een 6116 te kiezen. In figuur 9 is dat IC 20. In dit IC worden de tekens opgeslagen in de vorm van ASCII-kodes. Het opslaan geschiedt via IC1 en IC10, terwijl de adressering gebeurt door IC17, IC18 en IC19.

5 CA-1	0 - 1	2 - 0	- 12 volt
CA-2	0 - 3	4 - 0	- 12 volt
PA-0	0 - 5	6 - 0	+ 12 volt
PA-1	0 - 7	8 - 0	
PA-2	0 - 9	10 - 0	
PA-3	0 - 11	12 - 0	
PA-4	0 - 13	14 - 0	
PA-5	0 - 15	16 - 0	
PA-6	0 - 17	18 - 0	
PA-7	0 - 19	20 - 0	
PB-0	0 - 21	22 - 0	
PB-1	0 - 23	24 - 0	
PB-2	0 - 25	26 - 0	+ 5 volt
PB-3	0 - 27	28 - 0	+ 5 volt
PB-4	0 - 29	30 - 0	+ 5 volt
PB-5	0 - 31	32 - 0	
PB-6	0 - 33	34 - 0	
PB-7	0 - 35	36 - 0	Gnd
CB-1	0 - 37	38 - 0	Gnd
CB-2	0 - 39	40 - 0	Gnd

uit

Figuur 5. De parallel-interface. De parallel-port PA is de keyboard-ingang. Op PA-0 staat bit 0, op PA-1 bit 1, enz. Het strobe-sigitaal staat op CA-1 (data worden inlezen bij de eerste flank van de strobe-puls). Via CB-1 krijgt de ACIA (6850) zijn kloksigitaal. Deze lijn is dan ook verbonden met pen 3 en 4 van de 6850. Port PB kan als parallel-uitgang voor een printer worden gebruikt (Centronics-interface).

Figuur 6. Dit zijn de spanningen en signalen zoals ze op konektor 2 van de CPU/DRAM-kaart voorkomen. Deze aansluiting wordt gebruikt voor datacommunicatie volgens de RS-232-standaard. De standaard-RS-232-konektor (die op de kast van de computer wordt gemonteerd) moet als volgt worden aangesloten:

- 01 = Gnd (ground)
- 02 = TxD (transmit data)
- 03 = RxD (receive data)
- 04 = RTS (ready to send)
- 05 = CTS (clear to send)
- 06 = niet gebruikt
- 07 = Gnd (ground)
- 08 = DCD (data carrier detect)
- 09
- ...
- 15 = niet gebruikt
- 16 = TxClk (transmit clock)
- 17 = niet gebruikt
- 18 = niet gebruikt
- 19 = +5 volt
- 20 = DTR (data terminal ready)
- 21
- ...
- 25 = niet gebruikt

6 GND	0 - 1	2 - 0	
TxD	0 - 3	4 - 0	
RxD	0 - 5	6 - 0	TxCik
RTS	0 - 7	8 - 0	- 12 volt
CTS	0 - 9	10 - 0	+ 12 volt
	0 - 11	12 - 0	+ 5 volt
GND	0 - 13	14 - 0	DTR
DCD	0 - 15	16 - 0	

7

	A	C
+ 5 volt	0 - 01 - 0	+ 5 volt
A-0	0 - 02 - 0	A-1
A-2	0 - 03 - 0	A-3
A-4	0 - 04 - 0	A-5
A-6	0 - 05 - 0	A-7
A-8	0 - 06 - 0	A-9
A-10	0 - 07 - 0	A-11
	0 - 08 - 0	
	0 - 09 - 0	
D-0	0 - 10 - 0	D-1
D-2	0 - 11 - 0	D-3
D-4	0 - 12 - 0	D-5
D-6	0 - 13 - 0	D-7
R/W	0 - 14 - 0	E
Q	0 - 15 - 0	
	0 - 16 - 0	
	0 - 17 - 0	/O select
FIRQ	0 - 18 - 0	
IRQ	0 - 19 - 0	
NMI	0 - 20 - 0	
	0 - 21 - 0	
	0 - 22 - 0	
	0 - 23 - 0	
	0 - 24 - 0	
	0 - 25 - 0	
	0 - 26 - 0	Reset
	0 - 27 - 0	
	0 - 28 - 0	
	0 - 29 - 0	
+ 12 volt	0 - 30 - 0	
- 12 volt	0 - 31 - 0	
Gnd	0 - 32 - 0	Gnd

Met het inlezen van de RAM zijn we slechts halverwege de klus, omdat de inhoud van de RAM nog in voor ons leesbare vorm op het beeldscherm moet worden omgezet. In IC 21 (een EPROM van het type 2732) ligt voor elk ASCII-karakter de punt-matrix opgeslagen. IC12 (74LS377) dient als tussengeheugen. De ASCII-kode wordt gebruikt als adres voor de 2732, zodat het bijbehorende puntpatroon aan de uitgangen van de EPROM verschijnt.

De output van de 2732 wordt vervolgens omgezet in een serieel signaal (IC 22), gesommeerd met een aantal andere signalen en vervolgens versterkt tot een voor een monitor geschikt videosignaal (T1). De zogenaamde dot-clock-frekventie (de frekwentie voor de klok-ingang van IC 22) is 16 MHz en wordt gegenereerd door een oscillator die bestaat uit een aantal inverters (IC13). De CRTC is IC2; dit IC coördineert de diverse besturingssignalen voor het video-proces. Verder maakt

Onderdelenlijst CPU/DRAM-kaart

Weerstanden (allemaal 1/8 W):

R1...R8 = 22 Ω
 R9 = 390 Ω
 R10...R12, R15 = 33 k
 R14 = 5k6
 Ra1 = 7 x 4k7 (SIL-array of 7 aparte weerstanden)

Kondensatoren:

C1, C2 = 27 p
 C3...C11, C16, C18, C20, C21 = 100 n ker.
 C12...C14 = 330 p ker.
 C15 = 47 μ /16 V
 C17 = 1 μ /16 V
 C19 = 100 p ker.

Halfgeleiders:

Z1 = 3 V/400 mW zenerdiode
 IC1...IC8 = 4164 (200 ns voor basisversie)
 IC9 = 6809
 IC10 = 6522
 IC11, IC20 = 74LS244

IC12 = 74LS245

IC13 = 2716 (ESS 540)

IC14 = 2716 (ESS 541)

IC15 = 6850

IC16...IC19 = 74LS153

IC21 = 74LS04

IC22 = 74LS10

IC23 = 74LS32

IC24 = 74LS393

IC25 = 74LS20

IC26 = 74LS123

IC27 = 74LS132

IC28 = 1489 (75189)

IC29 = 1488 (75188)

Diversen:

X1 = kristal 3,6864 MHz (klein model)

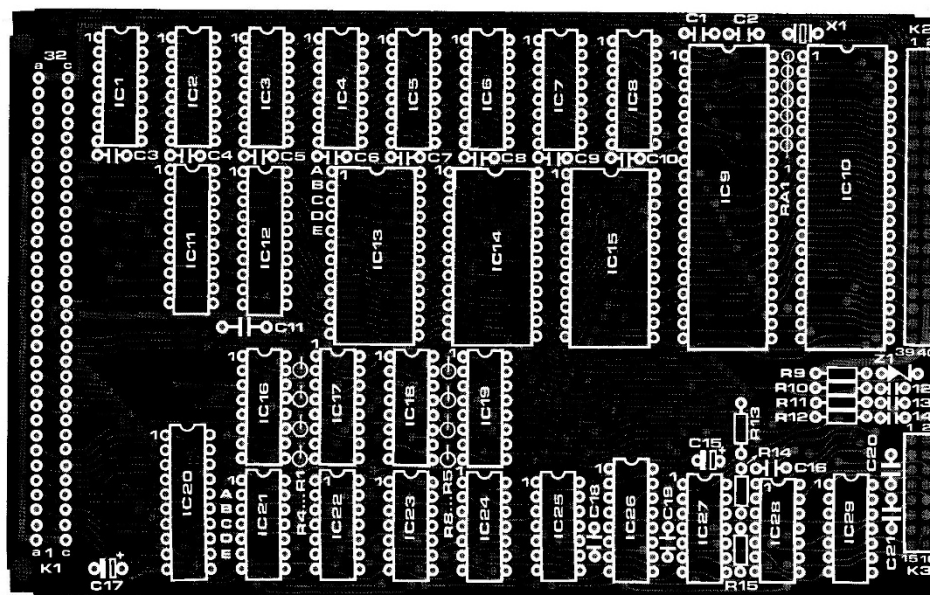
K1 = 64-polige buskonnektor, male, volgens DIN 41612

K2 = 40-polige male konnektor voor flat-cable en bijbehorende stekker, bijv. Molex nr.

5342-40-GS1 en 5320-40-BGS1

K3 = 16-polige male konnektor voor flat-cable en bijbehorende stekker, bijv. Molex nr. 5342-16-GS1 en 5320-16-BGS1

8



9

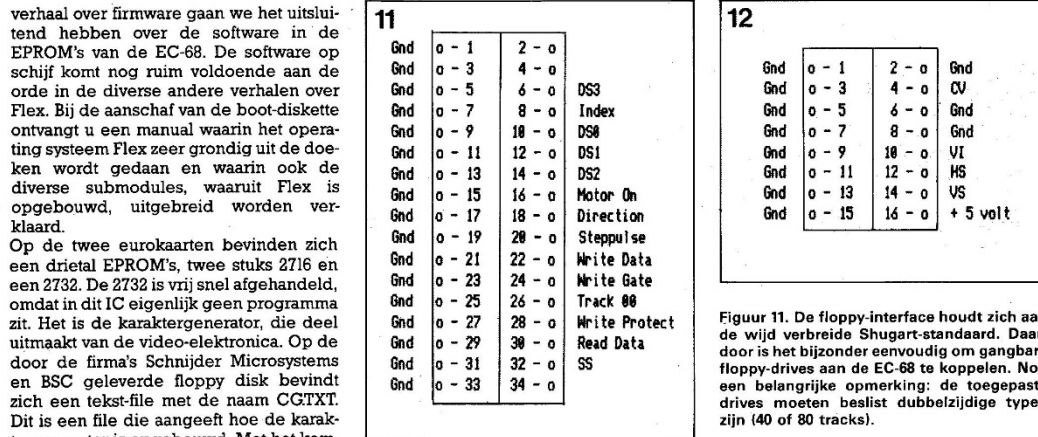
Figuur 9 en 10. Ook de schema's van de FDC/CRT-kaart zijn door de computer getekend. Er zitten op deze print iets minder IC's dan op de CPU/DRAM-kaart. Dit is vooral te danken aan de toepassing van een sterk geïntegreerde floppy-controller.

De CPU adresseert de CRTC net zoals de andere periferie-chips (de VIA en ACIA). Voor de adresdekodering zijn IC9 en IC3 (linksboven in figuur 9) verantwoordelijk. De floppy-disk-controller staat getekend in figuur 10. Ook hier is de eigenlijke controller slechts één IC (de WD 1770 van Western Digital); IC nummer 7 dus. De besturingselectronica om IC 7 is een stuk eenvoudiger van opzet dan die van het videogeedeel. Dit komt omdat de WD 1770 een zeer gecompliceerd IC is. Door

de verregaande integratie zijn bijvoorbeeld de instelpotentiometers (zoals die soms nog in oudere ontwerpen met FDC's voorkomen) achterwege gebleven, daar de zogenaamde data-separator ook in de FDC is geïntegreerd. Derhalve is geen enkele afregeling nodig. De 1770 is verantwoordelijk voor het schrijven en lezen van data van en naar de floppy-disk. Ook hier wordt met seriële communicatie gewerkt. De diverse besturingsignalen worden gebufferd door IC8, IC9 en IC16. Tevens beveiligen ze de cure WD 1770 tegen calamiteiten, wanneer in- en uitgangen zouden worden kortgesloten door foutief aanbrengen van de konektor die de floppy-disk-drive met de floppy-disk-interface verbindt. Dit gevaar bestaat omdat alle oneven genummerde aders van de bandkabel die floppy-disk drive en CRT/FDC-print verbindt, aan massaliggen. Dit is gedaan ter voorkoming van storing tussen de signalen onderling. Om diezelfde reden is het ook af te raden de bandkabel onnodig lang te maken. Uit figuur 10 blijkt dat de Elektuur Flex-computer EC-68 met maximaal 4 floppy-disk-drives kan werken (DS0 t/m DS3 betekent drive select 0 t/m 3). Bij het op de juiste manier inklemmen van de bandkabel in de bandkabel-konektor (zit ader nr. 1 ook op pen 1 van de bandkabel-konektor?) kan er met de diverse verbindingen eigenlijk niets meer misgaan, omdat de door ons aangegeven Molex-konektoren allemaal polarisatieknokken hebben en dus niet fout in het "male" konektorgedeelte kunnen worden gestoken. Het is trouwens raad-

De firmware

Zoals reeds eerder geschreven, kan hardware niet werken zonder software (omgekeerd trouwens ook niet!). De software voor de EC-68 kan men kwa plaatsing onderverdelen in een tweetal groepen: de software op de printen in de vorm van EPROM's en de software op floppy-disks. Typisch voor het operating system Flex is, dat nagenoeg alle software op floppy-disk staat (dit in tegenstelling tot de vele homecomputers die diverse EPROM's bevatten met BASIC-interpreters, tekstverwerkers, etc.). Bij Flex is het aantal EPROM's tot het noodzakelijke minimum beperkt; de rest staat op schijf. Hierin ligt de kracht van Flex. De software is zeer flexibel, omdat ze zeer makkelijk kan worden aangepast en veranderd, en er wordt niet meer ruimte dan nodig van het totale geheugenblok in beslag genomen door EPROM's. We geven graag toe dat hierdoor het systeem trager werkt dan een systeem met veel EPROM's aan boord. De access tot een EPROM is immers stukken sneller dan de access tot een floppy-disk-drive. In het



Figuur 12. De konnektor voor de monitor. Gewoonlijk gebruikt men hiervan alleen het composite-video-sig-naal en de massa-aansluiting. Via een 75- Ω -koax-kabel wordt het sig-naal dan naar de monitor gevoerd. Op de konnektor staan verder nog het geïnverteerde video-sig-naal en de twee synchronisatiesig-nalen.

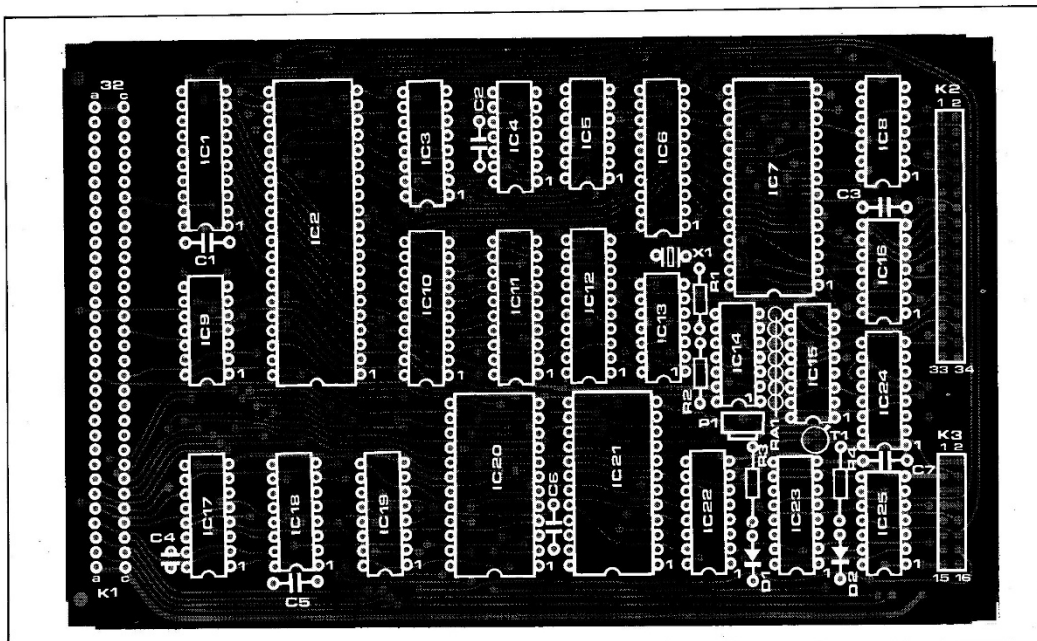
Tabel 1. De belangrijkste systeem-adressen en de geheugen-indeling van de EC-68.

Op de twee eurokaarten bevinden zich een drietal EPROM's, twee stuks 2716 en een 2732. De 2732 is vrij snel afgehandeld, omdat in dit IC eigenlijk geen programma zit. Het is de karaktergenerator, die de uitmaakt van de video-elektronica. Op de door de firma's Schnijder Microsystems en BSC geleverde floppy disk bevindt zich een tekst-file met de naam CGTXT. Dit is een file die aangeeft hoe de karaktergenerator is opgebouwd. Met het kommando LIST CGTXT kan de file bekeken worden. Dit ter informatie voor degenen die vinden dat het nodig is de presentatie van bepaalde karakters op het beeldscherm te veranderen. Er blijven vervolgens nog een tweetal EPROM's over. Daarin moeten dus wel de programma's zitten. Dit zijn Bootrom en Assist-09. We zullen de inhoud van de EPROM's vluchtig doornemen en wel in dezelfde volgorde als ze door de microprocessor gebruikt zullen worden.

Assist-09 is een monitorprogramma dat van origine is geschreven door de firma Motorola. Inmiddels is Assist-09 vrij voor gebruik door iedereen. Dat maakt het ontwikkelen van een monitorprogramma een stuk makkelijker. De ontwerper van de EC-68 is ook uitgegaan van Assist-09 en

heeft daarin een paar wijzigingen aangebracht om het programma aan te passen aan de EC-68. Bij het inschakelen van de EC-68 begint de 6809 met programma-executie vanaf adres FFFE (dat is de koude-start-vektor die in elke 68090-chip vast is ingebakken). De 6809 springt naar het begin van de EPROM op F800 (die namelijk de bovenste twee KByte van de memormap bezet).

Het belangrijkste doel van de Assist-09-monitor is het "booten" van het operating system. Na het inschakelen van de EC-68 verschijnt op het beeldscherm de tekst ASSIST 09, gevolgd door de monitor-prompt (voorgesteld door het teken >). Door achter dit teken de letter F (goed geraden, de F van Flex) te tikken,



Onderdelenlijst CRT/FDC-kaart

Weerstanden (allemaal 1/8 W):

R1, R2 = 560 Ω

R3 = 1 k

R4 = 2k7

Ra1 = 7 $\times$ 220 Ω (SIL-array of 7 aparte weerstanden)

Kondensatoren:

C1...C7 = 100 n ker.

Halfgeleiders:

D1, D2 = 1N4148

T1 = BFY 90

IC1, IC10 = 74LS245

IC2 = 6845

IC3 = 74LS139

IC4 = 74LS32

IC5 = 74LS00

IC6, IC12 = 74LS377

IC7 = 1770 (Western Digital)

IC8, IC16 = 74LS06

IC9 = 74LS138

IC11 = 74LS244

IC13 = 74LS04

IC14 = 74LS74

IC15 = 4050

IC17...IC19 = 74LS157

IC20 = 6116

IC21 = 2732 (ESS 542)

IC22 = 74LS165

IC23 = 74LS163

IC24 = 74LS175

IC25 = 74LS02

Diversen:

X1 = kristal 16 MHz (klein model)

K1 = 64-polige buskonnektor, male, volgens DIN 41612

K2 = 34-polige male konnektor en bijbehorende flat-cable-steker, bijv. Molex nr. 5342-34-GS1 en 5320-34-BGS1

K3 = 16-polige male konnektor en bijbehorende flat-cable-konnektor, bijv. Molex nr. 5342-16-GS1 en 5320-16-BGS1

Tabel 1.

:F800...FFFF:Assist-09 (EPROM);	:EC06:Floppy Disk Controller (sektor-register);	:CC00...D3FF:DOS;
:F000...F7FF:Bootrom (EPROM);	:EC05:Floppy Disk Controller (track-register);	:C980...CBFF:System Files Area;
:E800...EFFF:I/O;	:EC04:Floppy Disk Controller (status- en kommandoregister);	:C840...C97F:System FCB (File Control Block);
:ED00...EFFF>User-I/O;	:EC00:CRTC;	:C700...C83F:Scheduler & Printer Spooler (niet geïmplementeerd);
:EF00...EFFF:ACIA, VIA;	:E800...EBFF>User-I/O;	:C100...C6FF:Utility Command Area;
:EF80:VIA (parallele I/O);	:E000...E7FF:Videomemory;	C080...C0FF:Input Buffer;
:EF60:ACIA (seriële I/O);	:C000...DFFF:FLEX;	C000...C07F:Stack Area (SP wordt op C07F geïnitieerd);
:EC00...ECFF:DISK, VIDEO;	:DE00...DFFF:Disk Drivers;	:C000:Flex-operating-system-laadadres;
:EC07:Floppy Disk Controller (data-register);	:D400...DFFF:FMS (File Management System);	:0000...BFFF>User-RAM;

gevolgd door een carriage return, zal Assist-09 routines uit Bootrom oproepen en wordt het operating system Flex in het geheugen van de computer geladen. Basisfuncties in Assist-09 zijn onder ande-

re een kommando waarmee men de inhoud van de verschillende registers kan bekijken en modificeren (het zogenaamde "Registers"-commando), een kommando waarmee men de inhoud van een

geheugenblok kan bekijken en veranderen (de "Memory"- en "Display"-kommando's). Deze en nog diverse andere kommando's liggen op machinetaalniveau. Dit soort kommando's kan ook worden

uitgevoerd door Flex-utilities op schijf. Het zou hier te ver voeren de diverse routines van Assist-09 en Bootrom die hiervoor nodig zijn uit te spitten. Voor degenen die alles over Assist-09 en Bootrom willen weten, verwijzen we weer naar de bootdisk, deze keer naar de tekst-files ASSIST09.TXT en BOOTROM.TXT.

Assist-09 bevat een aantal hulpprogramma's voor manipulaties op machinenivo. Omdat bij het communiceren met de computer een aantal machine-specifieke (lees: hardware-afhankelijke) programma's dienen te worden uitgevoerd, is er nog gebruik gemaakt van een tweede EPROM: Bootrom. Hierin staan de video-, floppy- en I/O-drivers die specifiek zijn voor de EC-68. Specifiek, omdat ze gebruik maken van informatie en kommando's die ze op bepaalde adressen wegschrijven of inlezen. Uit de vijf schema's heeft de oplettende lezer al een aantal adressen kunnen opmaken, omdat ze bij de diverse IC's (waarop iets dergelijks van toepassing is) vermeld zijn. Zo wordt keyboard-input bijvoorbeeld ingelezen via de PIA op adres \$EF80, seriële input en output gaat via de ACIA op adres \$EF60, de floppy-disk-controller zit op adres \$EC04, etc.

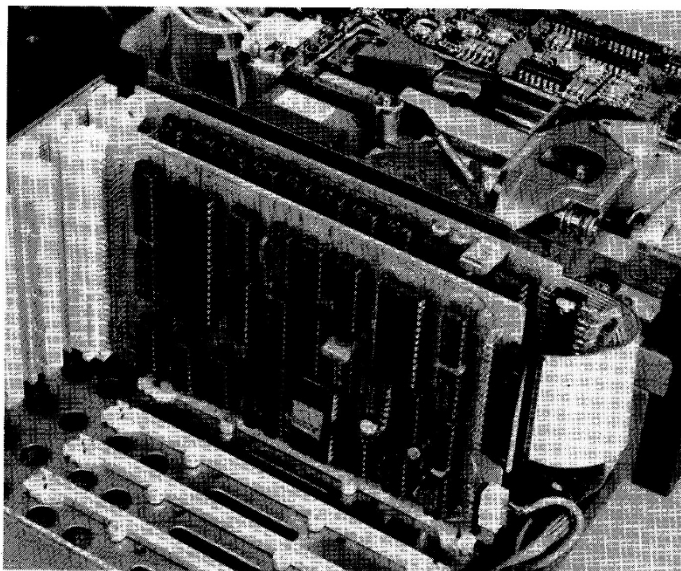
De diverse drivers zijn uitgebreid beschreven in de diverse op schijf staande files. Als u echt tot op bitnivo wilt nagaan wat er gebeurt bij het opstarten van de EC-68, moet u de diverse .TXT-files maar eens listen.

Tot slot vindt u een zeer globale memory-map (met de interessantere adressen) in tabel 1.

Verkrijgbaarheid

Hieronder volgt een lijstje van firma's die moeilijk verkrijgbare delen voor de Flex-computer leveren.

*De boot-diskette met het operating system Flex en de diverse info-files is verkrijgbaar bij:



Schnijder Microsystems
Schooteindseweg 8a
5756 BD Vlierden
tel. 04930-13666
en
BSC B.V.
Lochemseweg 28
7244 RS Barchem
tel. 05734-570

*De floppy-disk-controller WD1770 wordt geïmporteerd door:

BV. Diode
Hollantlaan 22
3526 AM Utrecht
tel. 030-884214

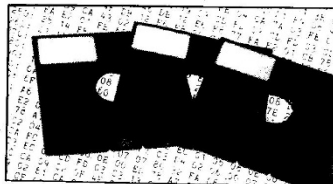
*De Molex-konnektoren worden geleverd door:

Molex B.V.
Visserstraat 13
5612 BS Eindhoven
tel. 040-450365

We danken de HCC-68XX-gebruikersgroep België voor het ter beschikking stellen van de video-driver voor dit Flex-systeem.

Binnenkort in Elektuur Computing nummer 4:

- de EC-65K — een nieuwe zelfbouwcomputer met 65816-CPU op 4 MHz
- Flex-software-overzicht en diagnostics
- Pascal-tekstverwerker
- Astrologie-programma
- Harddisk voor IBM-PC's



In dit nummer van *Elektuur Computing* presenteren we een 6809-systeem op twee eurokaarten: de EC-68. En daar hoort natuurlijk ook software bij. Wij hebben hier gekozen voor het in de V.S. wijd verbreide operating system Flex-9. Het software-aanbod voor Flex-9 is bijzonder groot. In dit artikel vertellen we wat meer over Flex-9, de software-basis van de EC-68.

Het operating system Flex is een diskette-gebaseerd systeem voor CPU's van het type 68xx (bijvoorbeeld 6800, 6802, 6809). Dit operating system is onder de computerhobbyisten nauwelijks verspreid. In de industrie is Flex vrij bekend. Momenteel wordt Flex daar echter verdrongen door zijn opvolger UNI-FLEX. De prijs van UNI-FLEX is echter dermate hoog dat een hobbyist zich dat niet kan permitteren. Ook voor kleine firma's vormt die prijs al een struikelblok. UNI-FLEX schijnt echter nog krachtiger te zijn dan UNIX. Maar nu terug naar Flex. Flex-9 is waarschijnlijk het sterkste operating system dat verkrijgbaar is voor de 6809. Het is in vele opzichten beter dan CP/M.

Wie zich intensief wil bezighouden met Flex, kunnen we het Amerikaanse tijdschrift "68 Micro Journal" aanbevelen. Wie geen moeite heeft met de Engelse computer-vaktaal, vindt hierin veel interessante hard- en software-informatie. Zoals al een paar keer is opgemerkt, is er veel software leverbaar voor Flex. En mocht men in ons kleine kikkerlandje voor een bepaalde toepassing geen programma vinden, dan kan nog altijd een programma in de Verenigde Staten besteld worden. Want dat is het grote pluspunt van Flex: elke bezitter van een Flex-systeemschijf kan *alle* Flex-programma's op zijn systeem laten draaien.

Het operating system Flex wordt op de markt gebracht door de Amerikaanse firma Technical Systems Consultants (TSC). Daarnaast leveren diverse softwarehuizen gebruikers-software voor Flex. Hieronder volgt een globaal overzicht van het Flex-operating-system, onderverdeeld in functionele groepen. Dit overzicht is zeker niet compleet, maar geeft een goede indruk van de mogelijkheden van het systeem.

De systeem-software

De diverse files, die gebruikt worden bij het "booten" (dat is een mooi Engels woord voor het opstarten van de computer), staan op de systeem-diskette. Hier volgt een opsomming van een aantal files op deze schijf. We beginnen uiteraard met Flex zelf (op disk FLEX.SYS

het operating system Flex

1

De door de gebruiker ingegeven tekst is onderstreept.

```
+++ LIST STARTUP.TXT
0.VERIFY IS ON
0.HECHO 00
0.HECHO 1B 2E 33
0.HECHO 0D 0A
0.DATE
0.HECHO 0D 0A
0.ECHO Good day FLEX software at your command
0.ECHO ### E L E K T O R .6809 Flex computer ###
0.ECHO My system information is:
0.ECHO [1] Memory capacity is 64 Kbytes
0.ECHO [2] Microprocessor is 6809
0.ECHO [3] Disc format according to FD 1771
0.ECHO [4] If drive is 40 tracks and we have double side
0.ECHO      and double density with 18 sectors per track.
0.ECHO      Then we have a disc capacity of 1404 sectors
0.ECHO      of 256 bytes each. Thus we have 376 Kbytes of
0.ECHO      disc storage.
0.HECHO 0D 0A
0.HECHO 0D 0A
0.ASN W=0 S=0
0.TTYSET WD=30 DP=25 PS=Y
0.TTYSET TB=50 EJ=00 ES=20 BE=00
0.LIST 0.START1.TXT
+++
```

genaamd). Voor het gebruikersgemak is aan het DOS een systeem-file verbonden waarin de diverse foutmeldingen zijn opgenomen (filenaam: ERROR.SYS). Na het inlezen van het DOS zal Flex zoeken naar een command-file (herkenbaar aan het achtervoegsel CMD) die EXEC (van execute) heet, EXEC.CMD. Flex voert namelijk de kommandoregel uit die staat in een tekst-file met de naam STARTUP.TXT. Meestal is dat: EXEC 0.START.TXT. Er moet dus ook een file START.TXT op disk staan. Deze tekst-file wordt door de gebruiker zelf geschreven met behulp van een tekstverwerker en bevat een aantal kommandoregels die

Figuur 1. Zo kan de inhoud van een STARTUP.TXT-file er bijvoorbeeld uit zien.

achter elkaar worden uitgevoerd (EXEC 0.STARTUP.TXT betekent dus gewoon: voer een aantal kommando's uit in de volgorde zoals deze zijn opgesomd in de tekst-file STARTUP.TXT). Figuur 1 geeft een typisch voorbeeld van de mogelijke inhoud van zo'n STARTUP-file. Elk kommando neemt een regel in beslag, omdat Flex elke regel als één kommando beschouwt. Een uitleg van de functie van de verschillende kommandoregels is wel

fpinit	formatteringsbuffer openen	fpinit <ea-formule>, ak
fpreal	gebroken getal naar formatteringsbuffer sturen	fpreal <ea-REAL>, ak
fpheX	hex-getal naar formatteringsbuffer sturen	fpheX <ea-HEX>, ak
fpint	geheel getal naar formatteringsbuffer sturen	fpint <ea-INTEGER>, ak
fpalfa	string naar formatteringsbuffer sturen	
fpend	formatteringsbuffer sluiten	fpend ak
fpchar	een teken naar formatteringsbuffer sturen	fpchar <ea-CHAR>, ak
chkfmt	floating-point-formaat testen	chkfmt <ea1>, <ea2>
floppy	naar floppy-systeem een kommando sturen	floppy <ea-buffer>
inchr.x	lees een teken van het toetsenbord	inchr.x <ea>
inhex.x	lees hexadecimaal van het toetsenbord	inhex.x <ea>
input	lees een string vanaf het toetsenbord in een buffer	input <ea>
list	stuur een assembler-instructie naar het beeldscherm	
msg	zet een foutcode om in karakters	
nocode	genereer geen object-code	
nolist	stuur geen listing naar het beeldscherm	
offset	definieer een offset voor de object-code	
org	definieer een begin van een programma	
outchr.x	stuur een karakter naar het beeldscherm	outchr.x <ea>
outhex.x	hex-ASCII-konversie naar het beeldscherm	outhex.x <ea>
page	nieuwe pagina	
print	stuur een string naar de printer	print <ea>
printer	stuur een listing naar de printer	
run	start een programma	
setdate.x	stel datum in voor real-time-klok	setdate.x <ea>
settime.x	stel de tijd in voor de real-time-klok	settime.x <ea>
time.x	lees tijd uit real-time-klok	time.x <ea>

tekenverklaring bij tabel 4

.x	.b, .w, .l
ak	een adresregister als format-pointer
<ea1>	effektief bron-adres
<ea2>	effektief bestemmingsadres
formule	zie hieronder

voorbeeld van een formule:

fm: format c'decimaal = 'xs4d.2dx.6a'.cl

fm:	label
c	carriage return
l	line feed
a	plaats voor alfanumerieke tekens
d	plaats voor numerieke tekens (digits)
s	plaats voor plus of min (sign)
h	plaats voor hexadecimaal tekens
x	plaats voor spaties
z	nulonderdrukking (leading zero suppression)
e	plaats voor exponent
.	plaats voor tekent
:	scheidingsteken voor volgende operanden
()	prioriteitshaken
@...9	herhalingsteller (repeat counter)

moment van dit schrijven op de EC-68K geïmplementeerd zijn. In het al eerder gememoreerde artikel "de 68000 kort en bondig" maken we ons verder vertrouwd met deze instructies. Een aantal praktische programmaproblemen laten zien hoe gemakkelijk deze assembler werkt. De meeste floating-point-instructies zijn net zoals de meeste andere 68000-instructies opgebouwd. Daarbij staat "ea" voor effectief adres:

mnemonic <ea1>, <ea2>

Een voorbeeld: geeft u de instructie fadd d0,d1

dan wordt het floating-point-getal van dataregister 0 opgeteld bij het floating-point-getal in register d1. Het resultaat van deze optelling staat in d1.

Een ander voorbeeld toont hoe groot de overeenkomst is tussen de 68000-pseudo-instructies en de normale 68000-instructies:

divs d1,d1 en

fddiv d2,d3

De eerste instructie, een normale 68000-instructie, levert een deling op tussen twee gehele getallen (integers). Het getal in d1 wordt gedeeld door het getal in d0. Het resultaat van de deling staat in d1 op bit 15...0. Bit 31...16 bevatten de rest van de deling. Op soortgelijke wijze verloopt de deling tussen twee floating-point-getallen. Het getal in d3 wordt gedeeld door het getal in d2, waarbij het resultaat, ook weer een floating-point-getal, in d3 terecht komt. Hoe de opbouw van een floating-point-getal is, laat figuur 2 zien.

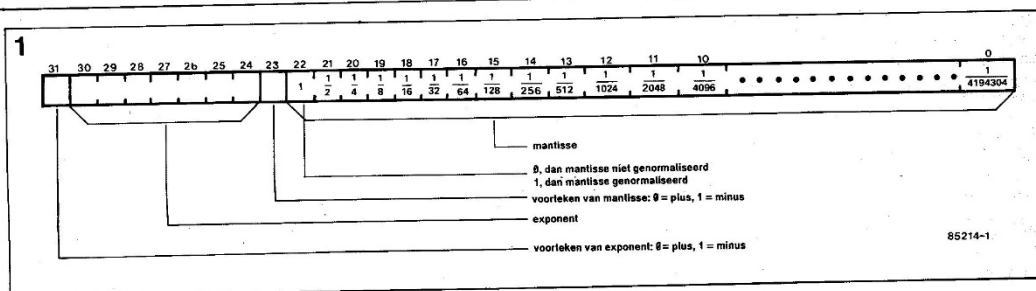
Ook de goniometrische functies hebben de gebruikelijke schrijfwijze in het effectieve adresveld:

fsin d0,d1.

De EC-68K berekent in dit voorbeeld de sinus van een bepaalde hoek. De hoek, in graden, bevindt zich als een floating-point-getal in dataregister d0. De berekende sinus komt als floating-point-getal te staan in d1. Het verhaal voor de exponentiële functies is eigenlijk analoog:

fexp d4,d5.

De exponent van de te berekenen e-macht vindt de processor in d4. De EC-68K berekent voor u ed 4 (waarbij e het grondtal is van de natuurlijke logaritme) en zet het resultaat in d5. Verdere details over de pseudo-instructies zijn te vinden in tabel 5 en 6. Zoals gezegd is de opbouw van een floating-point-getal te vinden in figuur 2 en hoe u daarmee moet werken, wordt getoond in de demoprogramma's in het artikel "de 68000 kort en bondig" (deel 2)



DATE.CMD drukt de bij het opbooten ingetypte datum op het beeldscherm af.

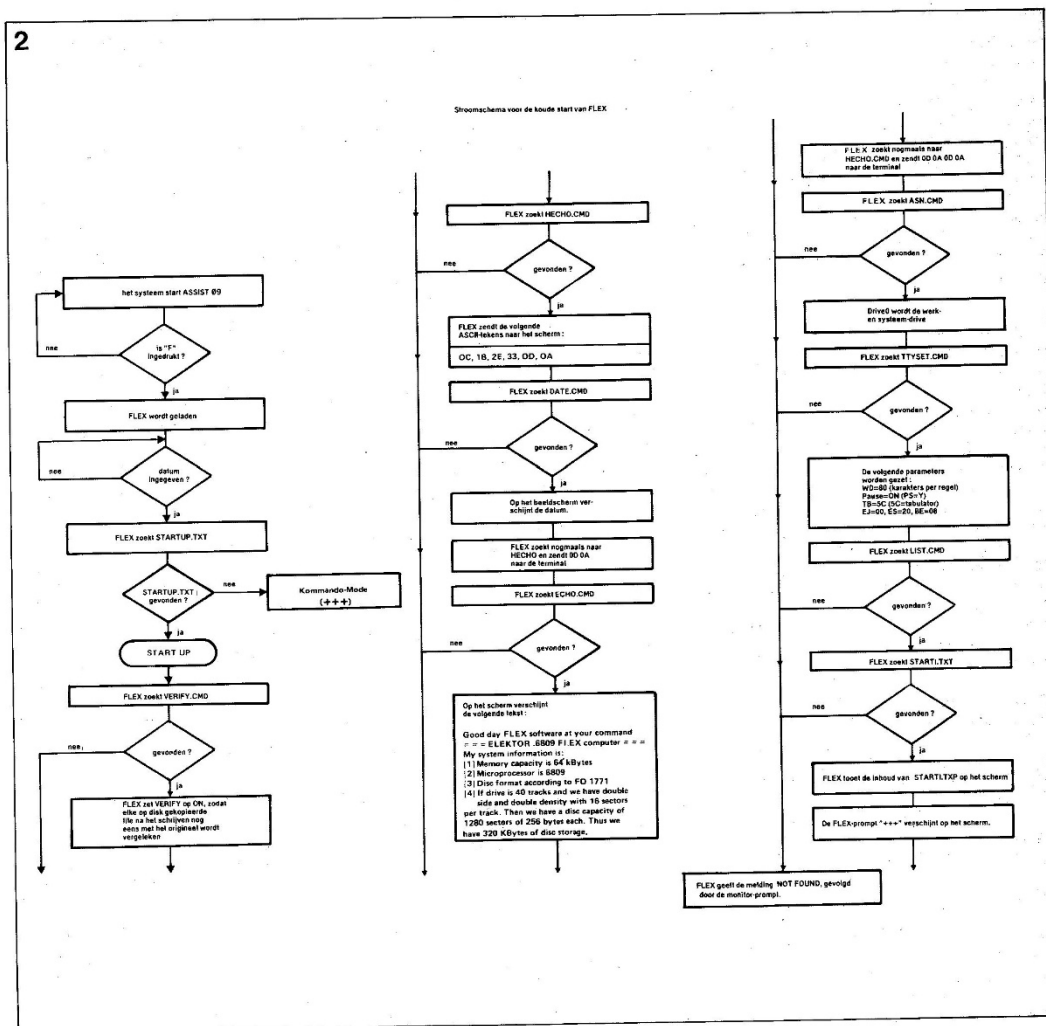
ASN.CMD (afkorting van assign) is de volgende command-file die we tegenkomen. Assign wordt gebruikt door het DOS voor het toekennen van work-drive en system-drive. "Arme" computergebruikers met een enkele disk-drive zullen in deze regel schrijven: `ASN W=0 S=0` wat tot gevolg heeft dat drive 0 zowel work-als

TTYSETCMD is een heel belangrijke file, omdat deze dient te worden gebruikt voor het initialiseren van een aantal terminal-parameters. Met terminal-parameters wordt onder andere bedoeld: het aantal karakters per regel (WD = 80: 80 tekens per regel), het aantal regels per beeldscherm (DP = 25: 25 regels per beeldscherm), welk ASCII-teken gebruikt wordt voor de TAB (tabulator); in ons voorbeeld SC.

LISTCMD sluit de rij. De naam spreekt voor zich. **List** geeft de inhoud van de daarna genoemde file op het beeldscherm. **@LIST @START1.TXT** betekent dus: druk de inhoud van de tekst-file genaamd **START1.TXT** op het beeldscherm af. Omdat het mogelijk is vanuit de **START1.TXT**-file naar bijvoorbeeld een tekstverwerker te gaan (de regel **@SCR DEMO.TXT** is daarvoor voldoende) komt de **START1.TXT** goed van pas om de

Software

Elektuur Computing 49



```

3
+++ newdisk
ARE YOU SHURE? Y
SCRATCH DISK IN DRIVE 0? Y
80 TRACKS (N=40 TRACKS)? Y
DOUBLE SIDED DISK? Y
DOUBLE DENSITY DISK? Y
18 SECTORS (N=16 SECTORS)? Y
VOLUME NAME? ELEKTOR
VOLUME NUMBER? 1

FORMATTING COMPLETE
TOTAL SECTORS = 2844
+++

```

gebruiker enige huishoudelijke informatie te geven over het programma waarmee Flex opstart (bijvoorbeeld enkele beschrijvingen van de belangrijkste kommando's). Op een andere disk, die gebruikt wordt voor het testen en ontwikkelen van BASIC-programma's, staat de file START2.TXT waarin de gebruiker op de hoogte wordt gebracht van het feit dat er na het inladen van bijvoorbeeld Extended BASIC (XBASIC) onmiddellijk kan worden begonnen met het intikken van BASIC-programmaregels.

Het spreekt voor zich dat de diverse in START.TXT genoemde kommando-files ook op disk aanwezig moeten zijn (anders reageert Flex met "NOT FOUND"). Een file die even belangrijk is als FLEX.SYS is de file LINKCMD. Hij is nodig voor het verbinden (linken) van Flex met de bootstrap-loader (syntax: LINK FLEX <CR>). Eindigen we het stuk systeemsoftware met het bespreken van een aantal files die voor het fatsoenlijk werken met Flex eigenlijk onontbeerlijk zijn.

PCMD en **PRINT.SYS** worden gebruikt wanneer de output niet naar het beeldscherm, maar naar een parallel-printer moet worden gestuurd. In plaats van LIST file-naam.extension wordt het nu: P LIST file-naam.extension.

CATCMD en **DIRCMD** drukken een catalog of directory van een disk af op het beeldscherm. P DIR 1 zet dus de directory van de schijf in drive nummer 1 op papier. DIR geeft uitgebreidere informatie dan CAT.

APPENDCMD wordt gebruikt voor het aan elkaar knopen van files (voorbeeld: APPEND file-naam1.txt, file-naam2.txt, etc...).

BUILDCMD is een heel eenvoudig programma voor het maken van tekst zonder enige tekstverwerkingsfaciliteiten (syntax: BUILD file-naam). Het is niet nodig een extension mee te geven, omdat de

```

4 +++ DIR

DIRECTORY OF DRIVE NUMBER 0

DISK: ELEKTOR #1   CREATED: 28-MAY-85

FILE#  NAME      TYPE      BEGIN  END    SIZE   DATE    PRT
-----
1  FLEX      .SYS      01-01  01-17   23    28-MAY-85
2  ERRORS    .SYS      01-18  01-18    4    28-MAY-85
3  EXEC      .CMD      01-1C  01-1C    1    28-MAY-85
4  STARTUP   .TXT      01-1D  01-1D    1    28-MAY-85
5  VERIFY    .CMD      01-1E  01-1E    1    28-MAY-85
6  ECHO      .CMD      01-1F  01-1F    1    28-MAY-85
7  HECHO     .CMD      01-20  01-20    1    28-MAY-85
8  DATE      .CMD      01-21  01-22    2    28-MAY-85
9  ASN       .CMD      01-23  01-23    1    28-MAY-85
10 TTYSET    .CMD      01-24  02-01    2    28-MAY-85
11 LIST     .CMD      02-02  02-04    3    28-MAY-85
12 LINK     .CMD      02-05  02-05    1    28-MAY-85
13 P        .CMD      02-06  02-06    1    28-MAY-85
14 PRINT    .SYS      02-07  02-07    1    28-MAY-85
15 CAT      .CMD      02-08  02-0E    7    28-MAY-85
16 DIR      .CMD      02-0F  02-13    5    28-MAY-85
17 APPEND   .CMD      02-14  02-16    3    28-MAY-85
18 BUILD    .CMD      02-17  02-17    1    28-MAY-85
19 COPY     .CMD      02-18  02-1C    5    28-MAY-85
20 DELETE   .CMD      02-1D  02-1E    2    28-MAY-85
21 NEWDISC  .CMD      02-1F  03-01    7    28-MAY-85
22 0        .CMD      03-02  03-03    2    28-MAY-85
23 RENAME   .CMD      03-04  03-04    1    28-MAY-85

FILES=23. SECTORS=76. LARGEST=23. FREE=1328

+++

```

aangemaakte file automatisch de extension TXT krijgt.

COPYCMD wordt gebruikt voor het dupliceren van files naar een andere schijf.

DETELCMD wist files op de schijf. Voor de zekerheid vraagt DELETE altijd een bevestiging van het voornemen om een file te verwijderen (ARE YOU SURE?). Het intikken van Y (yes) volstaat dan voor het verwijderen van de track- en sektorinformatie uit de directory (of catalog). De informatie zelf wordt dus niet verwijderd, maar de gebruikte sectoren worden vrijgegeven voor beschrijving met nieuwe files.

NEWDISCCMD is waarschijnlijk een van de meest gebruikte files, omdat deze nodig is voor het formatteren van nieuwe

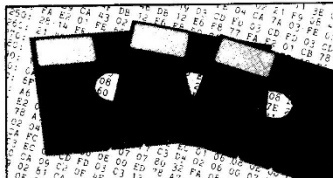
Figuur 2. Dit stroomschema toont het verloop van de startup-procedure. Als Flex is geladen, wordt gezocht naar de file STARTUP.TXT. In die file staan alle kommando's die de computer nodig heeft om de door de gebruiker ingegeven configuratie aan te nemen. De kommando's worden door de computer ingelezen alsof ze van het toetsenbord komen.

Figuur 3. Zo gaat het formatteren van een 80-track-ds-dd-diskette. Voor het begin van de procedure moet de te formatteren diskette in drive 0 zijn gestopt. Een reeds geformatteerde diskette wordt gewoon opnieuw geformatteerd. Daarbij gaat alle reeds aanwezige informatie op de schijf verloren!

Figuur 4. Een typisch voorbeeld van een directory van een diskette, dat verschijnt als een DIR-kommando wordt gegeven.

schrijven. NEWDISC stelt een aantal vragen aan de gebruiker om zo aantal tracks, single of double density, single of double side, volume-naam, volume-nummer vast te kunnen stellen. Na het formatteren controleert NEWDISC tevens of alle sectoren bruikbaar zijn en meldt eventueel slechte sectoren (BAD SECTORS).

OCMD is het output-kommando, waarmee het resultaat van een andere kommandoregel (bijvoorbeeld het opvragen van een directory) niet naar het beeldscherm gaat, maar naar de schijf. Syntax-voorbeeld: **ODIR**. De directory verschijnt niet op het beeldscherm, maar gaat als **DIR.OUT** naar de disk. Het resultaat van **O...** is dus altijd een output-file.



Technical Systems Consultants biedt voor de uitbreiding en ondersteuning van het disk-operating-system een breed programma "utilities" aan. Daartoe behoren enkele kleine programma's die voor de Flex-gebruiker eigenlijk onmisbaar zijn. In dit artikel bespreken we de interessantste programma's uit de zesdelige utility-set. In de documentatie die bij de diskette geleverd wordt, staan alle op deze schijf aanwezige utilities natuurlijk uitgebreid beschreven. In dit artikel introduceren we enkele nieuwe termen. We zullen bij elke file aangeven: de syntax, de default-waarden, een voorbeeld, het resultaat van de bewerking en eventueel hoe de output er uit zal zien. Met syntax wordt bedoeld: de exacte formulering van de kommandoregel (de grammatica van de computer dus); default staat voor een waarde of een tekst die automatisch door de computer wordt aangenomen indien de gebruiker zelf niets heeft ingetypt. Een tekstverwerker zal bijvoorbeeld automatisch naar een file met de extension **TEXT** (tekst) zoeken indien de gebruiker in de kommandoregel geen extension heeft aangegeven. Indien er een tekstfile in een voorbeeld wordt gebruikt, dan is dat een willekeurige file met de naam **"PARAGRI.TEXT"**. Maar nu terzake.

FIND

FINDCMD zoekt strings in tekstfiles en toont de regels waarin de aangegeven strings staan op het beeldscherm (of op papier). Eigenlijk is **FIND** typisch een kommando uit een tekstverwerker (het komt daarin ook voor!). Hierin ligt ook de grote kracht van deze command-file. Het is namelijk niet nodig een tekstverwerker te gebruiken om in een tekstfile (of enige andere file) een bepaalde letter, een woord, een getal of zelfs een complete zin te zoeken. Het laden van een tekstverwer-

RENAME.CMD sluit de rij. Met **RENAME** kunnen namen en extensions van files worden veranderd. **DEMOTXT** kan bijvoorbeeld een **BASIC**-demonstratieprogramma zijn dat met de tekstverwerker is gemaakt. **XBASIC** kan echter geen **TEXT**-file laden of runnen. Het kommando **RENAME DEMOTXT DEMO.BAS** is dan voldoende om de **TEXT**-extension in een **.BAS**(**BASIC** source code)-extension te veranderen.

De zojuist vermelde files (en nog enkele niet genoemde) maken deel uit van het software-pakket dat standaard wordt geleverd bij **FLEX**. Een uitvoerige beschrijving van deze files vindt men in het manual: *The Flex Disk Operating System* dat (evenals de *Advanced Programmers Guide*) meegeleverd wordt met de soft-

ware. Tenslotte willen we als voorbeeld een directory geven van een disk waarop de tot nu toe besproken files staan opgeslagen (figuur 4). De door de gebruiker ingegeven tekst is hierbij onderstreept.

En dan nog een allerlaatste opmerking. Met **Flex** is het mogelijk met meerdere disk-drives tegelijk te werken. Bij gebruik van één drive is drive 0 automatisch zowel system- als workdrive (zie de bespreking van de **ASN**-file). Bij meerdere drives typt men het drive-nummer voor de naam van de kommando-file. **2 DIR** geeft bijvoorbeeld de directory van de disk in drive nummer 2. Kommando's kunnen bij **Flex** overigens altijd worden ingetikt na het verschijnen van de system-prompt **"+++"**.

Flex-utilities

ker van gemiddelde lengte kost meer tijd dan het laden van **FINDCMD**!

* syntax: **FIND**,file-naam,te zoeken string

* defaults: file-extension is **TEXT**

* voorbeeld:
+++ **FIND**,PARAGRI,disk

* resultaat: Op het scherm verschijnen alle tekstregels uit de file **"PARAGRI"** waarin de tekst "disk" staat. Voorts worden de regelnummers gegeven.

* output:
+++ **FIND**,PARAGRI,Disk
(regelnummer)(eerste regel waarin "disk" staat)
(regelnummer)(tweede regel waarin "disk" staat)
...
...
TOTAL STRING OCCURENCE IS XX

Zoals we in het voorbeeld kunnen zien, wordt na alle uitgegeven regels ook nog het aantal gevonden plaatsen getoond.

WORDS

WORDSCMD is nog een programma dat goed van pas kan komen bij het gebruik van teksten. Met **WORDS** wordt het aantal woorden en regels in een tekst-file geteld. Een handig hulpje bij het maken van teksten die gebonden zijn aan een maximum of minimum aantal woorden en/of regels (verslagen bijvoorbeeld).

* syntax: **WORDS**,file-naam

* defaults: extension is **TEXT**

* voorbeeld:
+++ **WORDS**,PARAGRI

* Resultaat: Op het scherm verschijnen

het aantal woorden en regels die voorkomen in de tekst-file **PARAGRI**. Als markering tussen de woorden worden de spaties en het teken nieuwe regel/cursor-naar-rechts herkend.

* output:
+++ **WORDS**,PARAGRI
TOTAL WORD COUNT IS 1662
TOTAL LINE COUNT IS 388

TYPOS

TYPOSCMD is een luxe versie van **WORDS**. **TYPOS** telt niet alleen de woorden, maar vermeldt er ook nog bij hoeveel keer ze in de tekst voor komen. Dit programma is uitermate praktisch als men wil controleren of een bepaald woord niet al te vaak voorkomt. Omdat sommige woorden wel erg vaak worden gebruikt (zoals de, het en een) kan men ook een maximaal aantal opgeven. Doet men dit niet, dan kiest het systeem automatisch het aantal drie.

* syntax: **TYPOS**,file-naam,max. aantal

* defaults: extension is **TEXT**, aantal is 3 (als er geen aantal wordt ingetypt, worden alle woorden die drie maal en minder voorkomen, afgedrukt).

voorbeeld:
+++ **TYPOS**,PARAGRI,5

* resultaat: een lijst van alle woorden die in de tekst-file **PARAGRI** worden alle woorden opgesomd die 1, 2, 3, 4 en 5 maal voorkomen (eerst het aantal en dan het woord zelf), te beginnen met het grootste aantal (hier dus 5).

* output: spreekt voor zich en zou alleen maar bladvulling zijn (wordt derhalve achterwege gelaten).

+++ DUMP.SCRHELP.SYS

```

03 09
03 0A 00 01 43 6F 6D 6D 61 6E 64 73 20 69 6E 20 0000 Commands in
74 68 65 20 63 6F 6D 6D 61 6E 64 20 6D 6F 64 65 the command mode
3A 0D 0D 50 41 47 45 3D 4E 09 08 53 65 74 20 6E :PAGE=N Set n
65 77 20 70 61 67 65 6C 65 6E 67 74 68 2E 0D 4C ew pagelength.L
49 4E 45 3D 4E 09 08 53 65 74 20 6E 65 77 20 6C INE=N Set new l
69 6E 65 6C 65 6E 67 74 68 2E 0D 53 54 41 54 55 inelenth..STATU
53 3D 54 52 55 45 09 06 45 6E 61 62 6C 65 20 73 S=TRUE Enable s
74 61 74 75 73 20 6C 69 6E 65 20 69 6E 20 74 65 tatus line in te
78 74 20 6D 6F 64 65 2E 0D 53 54 41 54 55 53 3D xt mode..STATUS=
46 41 4C 53 45 09 05 44 69 73 61 62 6C 65 20 73 FALSE Disable s
74 61 74 75 73 20 6C 69 6E 65 2E 0D 49 4E 53 45 tatus line..INSE
52 54 3D 54 52 55 45 09 06 45 6E 61 62 6C 65 20 RT=TRUE Enable
61 75 74 6F 20 66 69 65 6C 64 69 6E 67 2E 0D 49 auto fielding..I
4E 53 45 52 54 3D 46 41 4C 53 45 09 05 44 69 73 NSERT=FALSE Dis
61 62 6C 65 20 61 75 74 6F 20 66 69 65 6C 64 69 able auto fieldi
6E 67 2E 0D 46 49 45 4C 44 3D 4E 09 0A 44 65 66 no..FIELD=N Def

03 0A
03 08 00 02 69 6E 65 20 66 69 65 6C 64 20 63 68 0000 line field ch
61 72 61 63 74 65 72 20 61 73 20 64 65 63 69 6D aracter as decim
61 6C 20 6E 75 6D 62 65 72 2E 0D 46 49 45 4C 44 al number..FIELD
3D 27 43 09 09 44 65 66 69 6E 65 66 69 65 6C 'C Define
64 20 69 61 72 61 63 74 65 73 20 41 d charac
53 70 63 68 61 72 7D 4D 45 SCII
08 44 65 4F 6D NT=

```

SPLIT

SPLITCMD komt goed van pas bij het werken met (te) grote teksten. Vaak blijken te grote teksten onoverzichtelijk te worden en is het wenselijk met een gedeelte van de originele tekst verder te werken. Met SPLIT kan een tekst worden gesplitst in twee nieuwe teksten. Het afbreekpunt is een regelnummer.

* syntax: SPLIT,input file-naam (naam van de bestaande file),output file-naam nummer 1 (naam van de file die loopt van regel 1 tot aan het afbreekpunt),output file-naam nummer 2 (naam van de file die de tekst bevat vanaf het breekpunt tot aan het einde van de originele file),nummer van de regel waar de scheiding ligt tussen de twee nieuwe files.

* defaults: extension is TXT

* voorbeeld:

+++ SPLIT,PARAGR1,DEEL1,DEEL2,100

* resultaat: de tekst-file paragr1 wordt gesplitst in een tekst-file deel 1, die regel 1 tot en met regel 99 bevat, en een tekst-file deel 2 die regel 100 tot het einde van de tekst in PARAGR1 bevat. Bij deze operatie blijft de originele tekst PARAGR1.TXT ongeschonden.

LOW-UP, UP-LOW

LOW-UPCMD en UP-LOWCMD zijn command-files die zeer sterk op elkaar lijken. Ze worden gebruikt voor het konver-

teren van kleine letters naar hoofdletters en omgekeerd. Een omzetting van kleine letters naar hoofdletters kan bijvoorbeeld van pas komen bij BASIC-programma's die geschreven zijn voor een BASIC-interpreter die beide soorten letters kan verwerken, en vervolgens "gerund" moeten worden met een BASIC-interpreter die uitsluitend hoofdletters verwerken kan.

* syntax: LOW-UP,originele file-naam met grote en kleine letters,naam van de nieuwe file waarin alle kleine letters zijn vervangen door hoofdletters. Het is vrij duidelijk hoe UP-LOW er uit zal zien.

* defaults: extension is TXT

* voorbeeld:

+++ LOW-UP,KLEIN,GROOT

* resultaat: de tekst-file KLEIN (kleine en hoofdletters) wordt omgezet in de file HOOFD (die geen kleine letters meer bevat, maar alleen hoofdletters).

DUMP

DUMPCMD wordt gebruikt voor het "dumpen" (op het beeldscherm of op papier) van de inhoud van een of meerdere disk-sektoren. De output geeft links boven het track- en sektor-adres van de sektor (of bij langere files van de diverse sektoren) en vervolgens op elke regel telkens 16 bytes in hex-vorm met helemaal

rechts de daarbij behorende ASCII-waarden. Dit programma komt goed van pas als men bij beschadigde disks wil nagaan wat er nog op de schijf staat (na een crash).

* syntax: DUMP,file-naam.extension

* defaults: extension is BIN (binary)

* voorbeeld:

+++ DUMP.SCRHELPSYS

* resultaat: op het beeldscherm verschijnt het start-adres van de file SCRHELPSYS (track-nummer en sektor-nummer uitgedrukt in hex) met daaronder 16 regels met op elke regel 16 bytes en daarnaast de eventuele ASCII-kode. Zo worden alle sektoren (elke sektor bevat 256 bytes) achter elkaar afgebeeld.

* output: zie figuur 1

CHECK

CHECKCMD wordt gebruikt om de inhoud van twee files met elkaar te verge-

lijken. Het resultaat van de vergelijking (dat kan zijn: THE FILES CHECKED DO NOT MATCH OF THE FILES CHECKED ARE IDENTICAL) wordt op het beeldscherm afgedrukt. Dit is een handig hulpmiddel wanneer de verdenking bestaat dat men te maken heeft met twee files met verschillende naam, maar met dezelfde inhoud.

* syntax: CHECK,file-naam1,file-naam2

* defaults: extension is TXT

* voorbeeld:

```
+++
CHECK,PARAGR1.TXT,PARAGR1.BAK
```

* resultaat: de inhoud van de tekst-file PARAGR1 en de backup-file worden met elkaar vergeleken.

* output:

```
+++
CHECK,PARAGR1.TXT,PARAGR1.BAK
THE FILES CHECKED ARE IDENTICAL
```

MAP

MAPCMD is een programma ter bepaling van de laad- en executie-adressen van command-files. Het geeft dus aan waar in het geheugen de "gemapte" command-file wordt weggeschreven en op welk adres executie van dat programma plaats zal vinden (dit adres wordt ook het transfer-adres genoemd). Dit programma komt goed van pas in combinatie met het SAVE-kommando (waarbij de inhoud van aangegeven geheugenblokken onder een gekozen naam naar schijf wordt weggeschreven).

* syntax: MAP,file-naam

* defaults: extension is BIN

* voorbeeld:

```
+++ MAP,MAPCMD
```

* resultaat: Op het beeldscherm worden start- en eind-adressen afgebeeld van de geheugenblokken waarin MAPCMD wordt ingelezen, gevolgd door het transfer-adres.

* output:

```
+++ MAP,MAPCMD
C100-C126
C100-C1E8
C100
```

FREE

FREECMD geeft aan hoeveel plaats er nog over is op de disk. Dit wordt uitgedrukt in het aantal sektoren en de hoeveelheid kilobytes.

* syntax: FREE X

* defaults: drive 0

* voorbeeld: FREE

* resultaat: op het beeldscherm verschijnt de tekst:
SECTORS REMAINING = X
APPROXIMATE KILOBYTES = YY

* output:

```
+++ FREE SECTORS REMAINING = 7
APPROXIMATE KILOBYTES = 17
```

TEST

TESTCMD kan gebruikers van een slechtere kwaliteit floppies een hoop ellende besparen. Alle sektoren op de disk worden getest en de eventuele slechte sektoren (bad sectors) worden gerapporteerd (met hun hex-adres).

* syntax: TEST,drive-nummer

* defaults: work-drive (zoals gekozen met ASN)

* voorbeeld: TEST,0

* resultaat: de diskette in drive nummer 0 wordt getest op bad sectors (de foutmelding zegt: BAD SECTOR AT ...). Uiteraard zal er ook een foutmelding plaatsvinden indien er een drive-nummer wordt gekozen van een niet aanwezige drive (melding: ILLEGAL DRIVE NUMBER).

* output:

```
+++ TEST,0
BAD SECTOR AT NN-MM
BAD SECTOR AT XX-YY
enz.
```

...

TEST COMPLETED

of:

```
+++ TEST,0
TEST COMPLETED
```

De laatste melding vindt plaats indien er geen bad sectors aangetroffen werden.

RPT

RPTCMD kan veel werk besparen bij het "runner" van programma's die meerdere malen dienen te worden uitgevoerd ("gerund"). Normaliter zal de gebruiker geneigd zijn de bewuste kommandoregel telkens opnieuw in te tikken. Dat is mogelijk maar niet erg efficiënt. Bij gebruik van RPT (repeat) hoeft alleen maar te worden ingetikt hoeveel keer de aangegeven kommandoregel dient te worden herhaald.

* syntax: RPT,aantal keer dat de kommandoregel dient te worden herhaald,de kommandoregel zelf.

* defaults: zijn niet van toepassing

* voorbeeld: RPT,3,P LIST PARAGR1.TXT

* resultaat: de inhoud van de tekst-file PARAGR1 wordt drie maal op papier afgedrukt.

PDEL

PDELCMD betekent: prompting delete. Bij de gewone delete geeft men de naam aan van de file of files die moeten worden gewist. PDEL vraagt niet om de namen van de files die gewist moeten worden, maar geeft de namen op het beeldscherm, waarna de gebruiker bij elke file

aan kan geven of deze moet worden gewist of niet. De volgorde waarin men dat doet, komt overeen met de volgorde van de files in de directory. Bij elke naam op het beeldscherm moet achter het vraagteken (verschijnt achter de file-naam) een Y (delete ? antwoord: YES) of N (No) of een carriage-return-teken (dat betekent: breek uitvoering van PDEL af) worden ingetikt. Na het afbreken komt het systeem terug in de command-mode (+++). Pas op: PDEL vraagt niet om verifikatie! Bij het gebruik van DELETCMD wordt er altijd nog gevraagd: ARE YOU SURE? PDEL doet dat niet. Y intikken betekent dus: weg file. In de kommandoregel kan nog een extra selectie worden gemaakt. Het opnemen van .TXT heeft zo tot gevolg dat alleen de files met de extensie .TXT worden afgedrukt. Het is ook mogelijk alle files te nemen die beginnen met de letter A (of een combinatie van letters), etc.

* syntax: PDEL,drive-nummer(opties-lijst)

* defaults: zijn niet van toepassing.

* voorbeeld: PDEL,0,TXT

* resultaat: Alle tekst-files die voorkomen op de diskette in drive 0, worden op het scherm getoond, waarbij telkens een Y, N of Carriage Return moet worden ingetoetst. Daarna verschijnt pas de volgende naam verschijnt. Dat gaat zo door totdat de laatste file is verschenen of een carriage return is ingetikt.

MEMDUMP

MEMDUMPCMD maakt het mogelijk de inhoud van het geheugen op het beeldscherm weer te geven. Dat gebeurt in blokken van 256 bytes. Na het dumpen van het eerste blok kan de gebruiker kiezen uit een drietal kommando's: F (forward), B (backward) of carriage return. Na het intikken van F of f wordt het volgende geheugenblok van 256 bytes afgedrukt. Bij B of b wordt het vorige geheugenblok getoond. De carriage return heeft ook hier weer de betekenis van afbreken van het programma en teruggaan naar de command-mode.

* syntax: MEMDUMP,startadres in hex

* defaults: indien geen startadres wordt ingetikt, begint MEMDUMP automatisch op adres 0000.

* voorbeeld: MEMDUMP,F000

* resultaat: op het beeldscherm verschijnt de inhoud van geheugenblok F000 tot F0FF.

MEMTEST

MEMTESTCMD spreekt voor zich. Dit programma test alle RAM-geheugenplaatsen.

* syntax: MEMTEST,startadres,eindadres

* defaults: zijn niet van toepassing.

* voorbeeld: MEMTEST,0,01FF

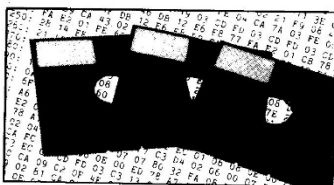
* resultaat: de geheugenlokatie van 0000

tot 01FF worden met toevalsnummers gevuld, waarna die nummers weer worden teruggelezen en vergeleken met de originele. Dit wordt een tijdje herhaald (het testen van een blok van 4 K duurt ongeveer 60 minuten). Er mag natuurlijk niet getest worden in het geheugengebied waar het programma MEMTEST zelf staat (de oplettende lezer weet al lang dat met behulp van het programma MAP, dat al eerder werd beschreven, makkelijk kan worden vastgesteld op welke geheugenlocaties MEMTESTCMD werd weggeschreven). Het met succes afwerken van een testcyclus wordt aangegeven door

het afdrukken van een "I" op het beeldscherm. Het onderbreken van MEMTEST gaat op een vrij brute manier: door het indrukken van de reset-knop.

Bij de aanschaf van de Utility-Command-set van TSC zal blijken dat er nog veel meer utilities in die set voorkomen. Voor de duidelijkheid volgt ter afsluiting van dit artikel nog een lijstje (in alfabetische volgorde) van de besproken utilities:

CHECK
DUMP
FIND
FREE
LOW-UP
MAP
MEMDUMP
MEMTEST
PDEL
RPT
SPLIT
TEST
TYPOS
UP-LOW
WORDS



graphics-software

Zoals in onze tweede Elektuur Computing beloofd, gaan we hier verder met het tweede en derde deel over de stuursoftware voor de grafische kleurenkaart. De hardware van deze kaart hebben we, zoals u weet, reeds in diverse Elektuur-nummers besproken.

Hier dus nu deel 2 ("graphics") en deel 3 ("circle") van de stuursoftware voor de 6502-processor. Bij de Octopus 65 ligt de stuursoftware in het bereik \$C000...CFFF en kan heel eenvoudig op een systeem-schijf worden "geïmplementeerd". Zo:

Op een 80-tracks-diskette is voor het programma nog plaats genoeg beschikbaar. Bij een 40-tracks-exemplaar is dat helaas niet het geval en moet er een utility-programma met een lengte van 2 tracks van de kopie van de systeem-schijf worden verwijderd (bijvoorbeeld eentje die men toch niet vaak gebruikt). Met de

DOS-bevelen "CALL" en "SAVE" worden de twee tracks dan op de dienovereenkomstige sporen van de graphics-systeem-schijf gezet. Daarna moet in het programma BEXEC* op een geschikte plaats een extra kommandoregel worden ingebouwd. Bijvoorbeeld als volgt:

1905 DISK!"CA C000=60,1";DISK!"CA C800=61,1";DISK!" GO C000"

Uiteraard is dit slechts een voorbeeld. Het regelnummer en de track-nummers hangen af van de BEXEC*-versie die men heeft en de voor de grafische software gebruikte tracks. Ga dus eerst na hoe het bij uw versie zit! De grafische software wordt, als alles goed is gedaan, automatisch bij het booten meegeladen. Men kan het grafische gedeelte dus direkt na het laden met BASIC-regels aanspreken (ook in BASIC-immediate-mode).

Tot slot nog even dit: op de binnenkant

van de omslag ziet u enkele fraaie kleurenfoto's die tonen waartoe de diverse demoprogramma's in staat zijn. Bij het aanschouwen van de beelden zult u het met ons eens zijn dat de grafische kaart behoorlijk wat in zijn mars heeft. En dit zijn nog maar "stilstaande" plaatjes! In werkelijkheid verandert de afbeelding namelijk voortdurend van vorm, zodat met recht over "kunst uit de computer" kan worden gesproken. Maar dat zult u zelf wel zien wanneer u de diverse demo's op uw eigen Octopus laat draaien.

Een allerlaatste opmerking nog: Omdat de bij de kleurenfoto's behorende programma's deels te lang waren om ze ernaast af te drukken, hebben we de demoprogramma's opgenomen in het hoofdstukje "programma's", dat u verderop in deze Elektuur Computing kunt vinden.

.EX
97 TRACK(S)
AXLO SAAMG. - - M.L. GRA

AXre. _ RE AS.

.A

```

10      :*****
20      :# VIDEO #
30      :#-----#
40      :#VIDEO-UTILITIES WITH GDP9366 #
50      :# WRITTEN BY P.LAVIGNE NOV'83 #
60      :#-----#
70      :# PART II : GRAPHICS #
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 E21A=     PAGE = X+2
280 E21B=     HSCROL = X+3
290      :
300 CF80      X = $CF80 :SOFTWARE REGS
310 CF85=     MIRXH = X+5
320 CF86=     MIRXL = X+6
330 CF87=     MIRYH = X+7
340 CF88=     MIRYL = X+8
350 CF89=     CHAR = X+9
360 CF8A=     PIXBUF = X+10
370 CF8B=     GRONT = X+13
380 CF91=     COLRPT = X+17

```

SERVICE

printen zelf maken

■ U hebt hiervoor nodig: een spuitbus transparant-spray, een layout-pagina, een UV-lamp, natronloog en positief foto-gevoelig printmateriaal (evt.

zelf maken met positieve fotokopieerlak en printmateriaal).

■ De foto-gevoelige koperzijde van het printmateriaal wordt met de transparant-spray goed nat gespoten.

■ De uit de layout-pagina geknipte koper-layout (in spiegelbeeld) legt u met de gedrukte zijde op het natte printmateriaal. Druk het papier licht aan en verwijder eventuele opgesloten lucht-

belletjes door voorzichtig met een prop papier over de layout te strijken.

■ Het geheel kan nu met een UV-lamp belicht worden. De belichtingstijd is afhankelijk van de gebruikte UV-lamp, de afstand hiervan tot het printmateriaal en het foto-gevoelige materiaal.

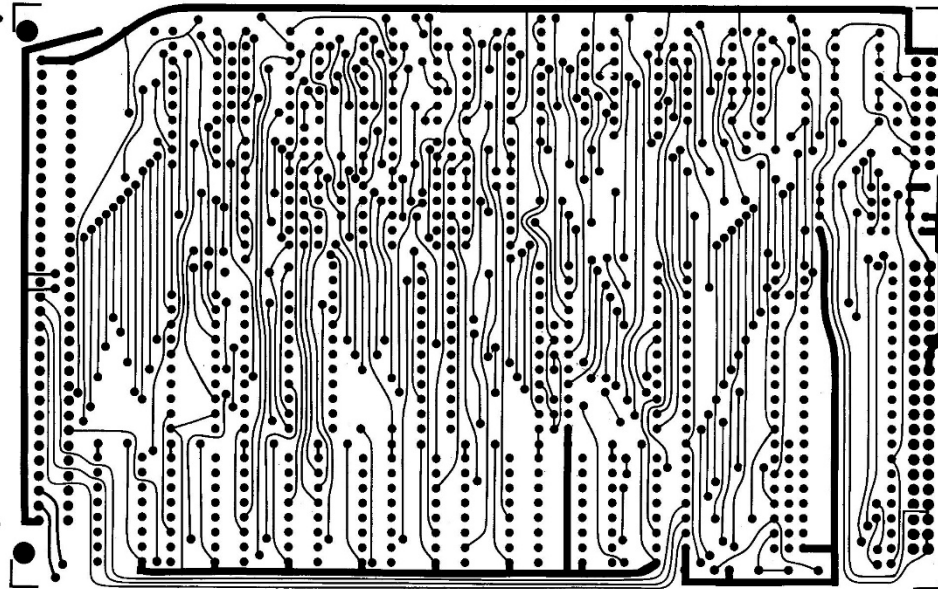
■ Na het belichten verwijderd u het layoutvel (nog meerdere malen bruikbaar) en spoelt u het printmateriaal onder stromend water schoon.

■ Na het ontwikkelen van de foto-gevoelige laag in natronloog (ongeveer 9 gram

in 1 liter water oplossen) kan de print in ijzer-3-chloride (500 gram FeCl_3 in 1 liter water) geëts worden. Spoel daarna de print grondig schoon (en ook uw handen!), verwijder met wat staalwol het foto-gevoelige laagje van de kopersporen en boor de gaatjes.

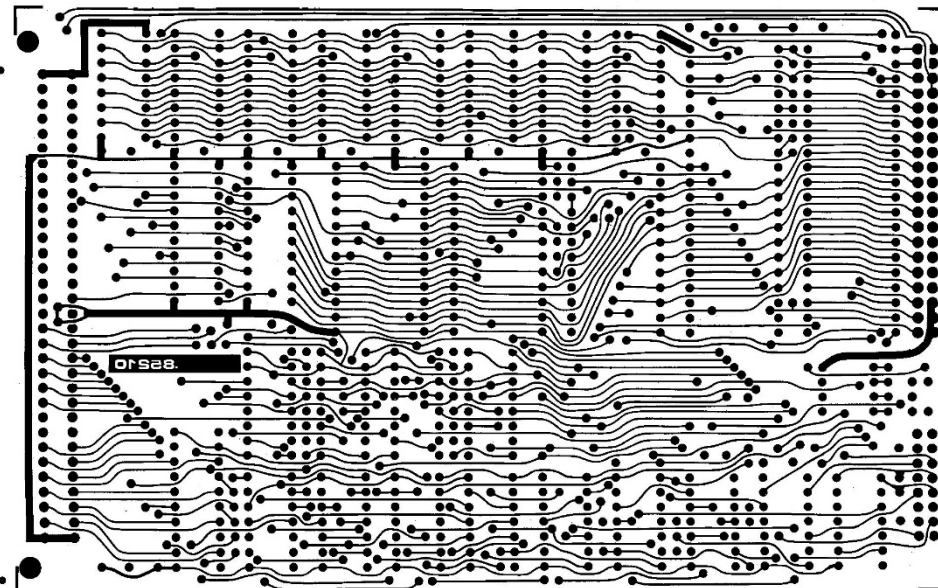
■ Spaar ons milieu en gooi geen uitgewerkte chemicaliën of resten ervan achteloos in de gootsteen, maar informeer in uw gemeente naar een hiervoor bestemd depot!

komponentenzijde



EC68: CPU/DRAM-kaart

koperzijde



SERVICE

EC68: CRT/FDC-kaart

