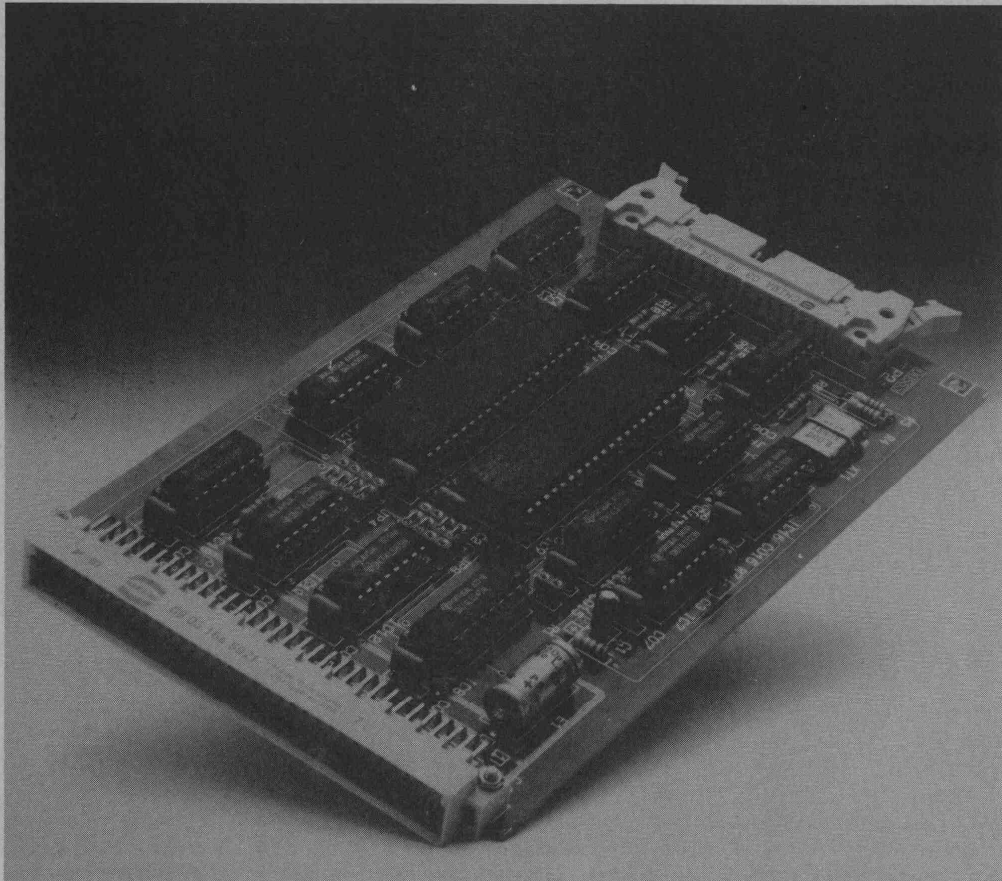




Tiende Jaargang, Nr. 4
Augustus 1966

DE 6502 KENNER

** DE 6502 KENNERS ** -- EEN CLUB VOOR 6XXXX GEBRUIKERS

De vereniging heeft leden in Nederland, België, Duitsland, Frankrijk, Spanje, Portugal, Amerika, Zambia, Denemarken. Het doel van de vereniging is het bevorderen van de kennisuitwisseling tussen gebruikers van 6XXXX-computers, als COMMODORE-64, APPLE, CHE-1, PEARCOM, AIM-65, SYM, PET, BBC ATARI, VIC-20, BASIS 108, PROTON-computers, ITT-2020, OSI, ACC 8000, ACORN ELECTRON, SYSTEM 65, PC-100, PALLAS, MINTA FORMOSA, ORIC-1, STARLIGHT, CV-777, ESTATE III, SBC65/68, KIM, NCS, KEMPAC SYSTEM-4, Elektuur-computers (JUNIOR, en de OCTOPUS), LASER, maar ook 6800, 6809 en 68000-computers. De kennisuitwisseling wordt o.a. gerealiseerd door 6 maal per jaar DE 6502 KENNER te publiceren, door het houden van landelijke clubbijeenkomsten, door het instandhouden van een cassette-bibliotheek en door het verlenen van papierware-service. Regionale bijeenkomsten worden door de leden georganiseerd.

Verschijningsdata
DE 6502 KENNER 1985

derde zaterdag van
februari, april, juni,
augustus, oktober, december.

Inlichtingen over de regio-
bijeenkomsten:

Gerard van Roekel
Van der Palmstraat 11 - C
3135 LK Vlaardingen
Tel.: 010 - 351101

De vereniging is volledig onafhankelijk, is statutair opgericht en ingeschreven bij de Kamer van Koophandel en Fabrieken voor Holland's Noorderkwartier te Alkmaar, onder nummer 634305.

Voorzitter:
Rinus Vleesch-Dubois
Fl. Nightingalestraat 212
2037 NG Haarlem
Tel.: 023 - 330993

Penningmeester:
John F. van Sprang
Tulp 71
2925 EN Kripen/IJssel.
Tel.: 01807 - 20589

Leden:
Adri Hinkel (05490 - 51151) Hardware/software/DOS65
Erwin Visschedijk (05490 - 71416) Hardware/software/DOS65
Bert van Opbroek (01729 - 8636) Hardware/software/PET
Nico de Vries (010 - 502239) Hardware/software/PET
Erevoorzitter: Siep de Vries
Ereleden: Mw. H. de Vries - Van der Winden
Anton Mueller
Lidmaatschap: Hfl. 45,- per kalenderjaar, postrekening 3757649 t.n.v. Penningmeester KIM Gebruikers Club Ned., Kripen/IJssel.

Advertenties: Tarieven op aanvraag bij de redactie.

Bijeenkomsten van de club

derde zaterdag van
januari, maart, mei,
september, november.

Redactie-adres en informatie
over papierware etc.

Willem L. van Pelt
Jacob Jordaensstraat 15
2923 CK Kripen/IJssel.
Tel.: 01807 - 19881

** DE 6502 KENNER ** -- EEN BLAD VOOR 6XXXX GEBRUIKERS

DE 6502 KENNER is een uitgave van de KIM Gebruikers Club Nederland. Het blad wordt verstrekt aan leden van de club. DE 6502 KENNER wordt van kopij voorzien door leden van de club, bij de opmaak van een publikatie bijgestaan door de redactie. De inzendingen van programma's dienen voorzien te zijn van commentaar in de listings en zo mogelijk door een inleiding voorafgegaan. Publikatie van een inzending betekent niet dat de redactie of het bestuur enige aansprakelijkheid aanvaardt voor de toepassing ervan. De inzendingen kunnen geschieden in assembly-source-listings, in Basic, in Basicode, Forth, Focal, Comal, Pascal, Fortran, Cobol, Logo Elan, etc. etc.

De leden schrijven ook artikelen over de door hen ontwikkelde hardware en/of aanpassingen daarop. Zij schrijven tevens artikelen van algemene aard of reageren op publikaties van andere inzenders.

DE 6502 KENNER IS EEN BLAD VAN EN DOOR DE LEDEN

Micro-ADE Assembler/Disassembler/Editor is een produkt van Micro Ware Ltd., geschreven door Peter Jennings en bestemd voor alle 6502-computers. De KIM Gebruikers Club Ned. heeft de copyrights verworven nadat ons lid Sebo Woldringh de 4 K KIM-1 versie uitbreidde tot 8 K KIM-1 versie. Adri Hinkel paste deze aan voor de JUNIOR. Willem L. van Pelt stelde een nieuwe 8 K source-listing voor de JUNIOR samen. De implementatie op andere systemen dan de KIM-1 en JUNIOR kan eenvoudig gebeuren door het aanpassen van de I/O-adressen, welke in de source-listing gemakkelijk te vinden zijn.

FATE Format-lister/cond. Assembler/Tape-utilities/Editor is de door ons lid Rob Banen geschreven source-listing van een 12 K universeel systeem voor de JUNIOR-computer aan de hand van het universele disk operating systeem van de fa. Proton Electronics te Naarden, nu geschikt voor werken met tapes. FATE wordt beschikbaar gesteld met toestemming van Proton.

In de edities van DE 6502 KENNER worden regelmatig mededelingen gedaan over de door de club georganiseerde bijeenkomsten. Ook worden bestuurlijke mededelingen gedaan, naast informatie over hetgeen in de handel te koop is. Leden die iets te koop hebben of iets zoeken kunnen dit in de edities van DE 6502 KENNER bekend maken. Ook worden brieven aan de redactie gepubliceerd, evenals specifieke vragen van leden. De edities worden samengesteld op basis van een groot aantal prioriteiten, welke door een redactievergadering worden gehanteerd. Deze vergadering bestaat uit de vaste medewerkers zoals in de colofon vermeld. Het aantal inzendingen is groter dan in een enkele editie van minimaal 48 pagina's is te verwerken. Hierdoor kan het voorkomen dat een inzending eerst na enige tijd kan worden gepubliceerd.

DE CLUB HEEFT BEHOEFTE AAN MEER LEDEN. WIJ WILLEN MEER AAN KUNNEN BIJEN DAN NU AL HET GEVAL IS. WERF DAAROM EEN LID!

WILT U EEN PRIJSLIJST? STUUR EEN BEFRANKEERDE ENVELOP AAN HET REDAKTIE-ADRES.

Een onafhankelijke jury kent jaarlijks een aantal aanmoedigingspremies toe aan auteurs van gepubliceerde artikelen in DE 6502 KENNER.

DE 6502 KENNER

De 6502 KENNER is een uitgave van de KIM gebruikers Club Nederland.

Adres voor het inzenden van en reacties op artikelen voor DE 6502 KENNER:
Willem L. van Pelt
Jacob Jordaensstraat 15
2923 CK Kripen a/IJssel
Tel.: 01807 - 19881

Vaste medewerkers:

Willem L. van Pelt
Gerard van Roekel
Frans Smeehuijzen
Coen Boltjes

Freelance medewerkers:

Rob Banen
Fred Behringer, (Germany)
Fridus Jonkman
Gert Klein
Roger Langeveld
Marc Lachaert, (Belgium)
Fernando Lopes, (Portugal)
Frank Manshande
Gert van Opbroek
Leif Rasaussen, Sweden
Ruud Uphoff
Frans Verberkt
Herman Zondag

Vertaalwerk:

Fred Behringer (Germany)
Willem van Asperen
Frank Bens
Albert v.d. Beukel
Rene Hettfleisch
Jaap de Hoop
Coen Kleipool (France)
Maarten van Lieshout

Gehele of gedeeltelijke overname van de inhoud van DE 6502 KENNER zonder toestemming van het bestuur is verboden. Toepassing van gepubliceerde programma's, hardware etc. is alleen toegestaan voor persoonlijk gebruik.

DE 6502 KENNER verschijnt 6 x per jaar en heeft een oplage van 500 exemplaren.

Copyright (C) 1986 KIM Gebruikers Club Nederland.

De voorpagina is de DOS65-controllerkaart, ontwikkeld door Ad Brouwer.
CAD/CAM: E. Visschedijk.
I.s.m.: A. Hankel
Fotogr.: Fr. Visschedijk.

I.v.m. auteurswetgeving aanvaardt de redactie geen aansprakelijkheid voor inzendingen. Tenzij anders aangegeven, dient de inzending afkomstig te zijn van de inzender.

INHOUDSOPGAVE DE 6502 KENNER NR. 45 AUGUSTUS 1986

1. Uitnodiging Landelijke Bijeenkomst	20 September 1986	2.
2. Van de redactie		6.
3. Elektor's EC65: Starcatcher; Basic game ... Leif Rasaussen, Sweden		3.
4. ACORN ELECTRON: Disassembler; Basic program ... Simon J. Voortaan		4.
5. The MC68000 Microprocessor; a new processor in our club ... Gert van Opbroek		7.
6. Elektor's JUNIOR computer with VDU, OHIO DOS and bank-switching; NUMBERS; routines to handle the input of numbers from the keyboard REAL TIME CLOCK; routines to set and to display time, based on 146818 IC ... Fernando Lopes, Portugal		12.
7. Elektor VDU-card modifications ... J.C. Rix, England		21.
8. Elektor's JUNIOR or OCTOPUS/EC65 computer: New CENTRONIC routine for the JUNIOR/OCTOPUS 65 ... Coen Boltjes (Transl.: Elya van der Veer)		22.
9. ENQUETE onder de leden van de KIM Gebruikersclub Nederland ENQUIRY among the members of the KIM Users Club The Netherlands		23.
10. JUNIOR: POSVAL SCHAKMONITOR part 2 JUNIOR: POSVAL SCHAKMONITOR for Elektor's JUNIOR with hexdisplay ... Frans Raaijmakers		25.
11. Letter to the editor ... Fred Behringer, Muenchen		42.
12. BASIC: Competitiestanden (Handbal) part 4 ... Gerard Keet		43.
13. MON/DOS65: 6502-Tracer for the MON/DOS65 computer Use of cursor-control-keys ED ... Rene Hettfleisch (Transl.: Maarten van Lieshout)		46.
14. Elektor's OCTOPUS/EC65 computer: OCTOPUS disks. ... Willem van Asperen		

From our member Jaap de Hoop we received information about two new microprocessors: the 6TE 665SC802 and the 6TE 665SC816, 16-bit mups featuring total software compatibility with 8-bit NMOS and CMOS 6500 series mups. The 665SC802 is pin-to-pin compatible with 8-bit 6502 devices currently available, while also providing full 16-bit internal operation. The 665SC816 provides 24 address lines for 16 Mbyte addressing, while providing both 8-bit and 16-bit operation. Both microprocessors are obtainable by Microtronics, Wilgenkade 10, 3992 LL Houten, The Netherlands. Phone: 03403 - 91369. Prices for both microprocessors about Hfl. 75,00.

For Apple II and IIE: Graphics Board usable as 768 Kbyte RAM Disk. Computer Informations Systeme, Am Eisernen Schlag 27, 6000 Frankfurt/Main 50, Germany, offers this product mentioned above. Supporting software utilities like zoom, copy, pattern, text, dump and filling arrays, the colour graphics board features a resolution of 512x512 pixels interlaced or 512x256 non-interlaced. Other features include 48 separate graphics pages for animation, 1 million pixels/s writing speed, 8 basic colours and 28 mixed colours and 8 shades on monochrome monitors. The board can be connected to standard monitors and it features separate BAS and TTL output. Data can be input through light-pen, joy-stick, mouse and digitizer board.

PROGRAMMA

Wout van Dinther en Will Cuijpers zijn onverdroten als het gaat om het beantwoorden van hardware en software problemen aangaande de Octopus. Stuur uw vragen - als het even kan - alvast naar de redactie, dat spaart wat tijd. Bovendien kan er later wellicht een publicatie over volgen. Heeft U bepaalde problemen al opgelost, zet ze dan ook even op papier en stuur het naar de redactie. Het Forum kan worden gevolgd door het tonen van wat nieuwtjes.

11.45 - MARKT. Vraag en Aanbod door de aanwezige leden.
12.00 - Lunch.
12.45 - INFORMEEL GEDRIFT.

Er is volop gelegenheid kennis te maken met andere leden van de club en hun systemen.

4. Heeft U software of hardware ontwikkeld? Breng het mee en geef het af aan de redactie.
Wilt U meedoen aan vertaalwerk?
Maakt U graag leuke illustraties?
Kent U mensen met een 6502, 6809 of 68000?
Geef het op aan de redactie.

17.00 - Sluiting van de bijeenkomst.

[illegible]

NEWS FROM THE PAPERWARE SERVICE

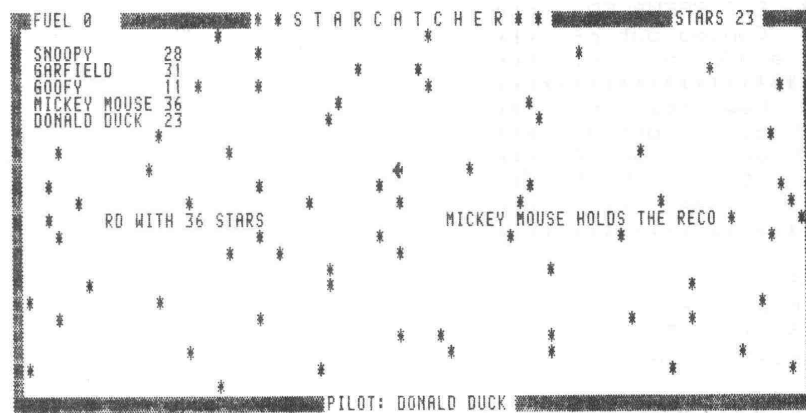
NEW NEW NEW NEW NEW NEW NEW NEW

THE VDU CARD

Article to assure that Elektor's VDU card works well.
Software (6502-machinecode) with introduction.
Many members of the club use this software instead of the
software from Elektor's paperware.
Authors: J.J.A. and J.A.J. Janssen, The Netherlands.
Transl.: Willem van Asperen.
System: Elektor's JUNIOR computer with VDU card.
Publ.: DE 6502 KENNER 31, Apr. 1984, p. 17-26.
Send cheque of Hfl. 19.00 to W.L. van Pelt (eurocheque 9,50)
Price only for European (C.E.P.T.) countries.

THE GRAPHIC DISPLAY

Article with schematic diagram to do graphical work (plotting of graphs etc.) with introduction.
Authors: J. J.A. and J.A.J. Janssen, The Netherlands.
Transl.: Willem van Asperen.
System : Elektor's JUNIOR computer with VDU card.
Publ. : DE 6502 KENNER 36, Febr. 1985, p. 5-11.
Send cheque of Hfl. 14,50 to W.L. van Pelt (eurocheque 5,00)
Price only for European (C.E.P.T.) countries.



```

10 LD=59392 : CR=59371 : IK=57601
11 FOR W=1 TO 6 : CU=7+LO : P=0 : F=500 : R=238
12 POKE CR,64 : DISK!"GO F32F"
20 GOSUB 74
21 PRINT(22,8)W". WHAT IS YOUR NAME "; INPUT N$(W)
22 POKE CR,32 : DISK!"GO F32F"
30 FOR J=0 TO 23 STEP 23 : FOR I=0 TO 79
31 POKE I+J*80+LO,139 : NEXT : NEXT
32 FOR I=0 TO 79 STEP 79 : FOR J=1 TO 22
33 POKE I+J*80+LO,139 : NEXT : NEXT
34 GOSUB 74 : PRINT(3,0)"FUEL "F
35 PRINT(61,0)"STARS "P : Y=12
36 PRINT(31,23)"PILOT: "+N$(W);
37 FOR I=0 TO 99 : GOSUB 72 : PRINT(X,Y)"* " : NEXT
40 IF X=0 OR X=79 OR Y=0 OR Y=23 THEN GOSUB 73
41 POKE CU,32 : CU=X+Y*80+LO
42 IF PEEK(CU)=42 THEN P=P+1 : PRINT(68,0) P
43 POKE CU,R : PRINT(8,0) F : F=F-1 : IF F<0 THEN 50
44 Q=PEEK(IK)-128
45 IF Q= 8 THEN X=X-1 : R=239 : GOTO 40
46 IF Q= 9 THEN X=X+1 : R=237 : GOTO 40
47 IF Q=11 THEN Y=Y-1 : R=236 : GOTO 40
48 IF Q=10 THEN GOSUB 72 : GOTO 40
49 Y=Y+1 : R=238 : GOTO 40
50 P(W)=P : FOR I=1 TO W : PRINT(3,I)N$(I),P(I)
51 IF P(I)>P THEN P=P(I) : B=I
52 NEXT
53 A$=N$(B)+" HOLDS THE RECORD WITH "+STR$(P)+" STARS "
54 FOR I=1 TO 60 : GOSUB 70 : NEXT
55 FOR I=1 TO 60 : GOSUB 71 : GOSUB 70 : NEXT
56 FOR I=1 TO LEN(A$) : GOSUB 71 : NEXT : NEXT
60 DISK!"GO F3E1" : DISK!"GO F32F " : END
70 PRINT(70-I,12) MID$(A$,I,1) : RETURN
71 PRINT(10,12) MID$(A$,I+1,LEN(A$)) : RETURN
72 X=INT(77*RND(1)+1) : Y=INT(22*RND(1)+1) : RETURN
73 P=0 : PRINT(68,0)P : GOSUB 72 : RETURN
74 PRINT(23,0)"* * S T A R C A T C H E R * * " : RETURN

```

```

100 REM*****
101 REM*****
102 REM##      List of Variables      ##
106 REM##      CR.....Cursors registeraddress      ##
107 REM##      IK.....Keyboard data portaddress      ##
108 REM##      CU.....Position on screen      ##
109 REM##      X,Y.....Koordinates for CU      ##
110 REM##      I,J.....Variables for loops      ##
111 REM##      P.....Counts catched stars      ##
112 REM##      F.....Fuel gauge      ##
113 REM##      R.....Airoplane ascii nr.      ##
114 REM##      W.....Game rounds      ##
115 REM##      Q.....Keyboard data      ##
116 REM##      N$.....Name of player      ##
117 REM##      A$.....Record display string      ##
118 REM##
121 REM##      Descriptions      ##
123 REM##      line 10.....Initialise      ##
124 REM##      line 11.....Start of six games-loop      ##
125 REM##      line 12.....Cursor blink, clear screen      ##
126 REM##      line 20-22...Heading, name, cursor off      ##
127 REM##      line 30-37...Make the screen-picture      ##
128 REM##      line 40.....Start of main loop. When      ##
129 REM##      the airoplane drops out of the galaxy,      ##
130 REM##      all collected stars are lost.      ##
131 REM##      line 41.....Del. last plane, calc. new      ##
132 REM##      line 42.....Is it a star inc. star coun.##
133 REM##      line 43.....Place new plane, dec. fuel      ##
134 REM##      line 44.....Get keyboard data (INKEY)      ##
135 REM##      line 45-47...Test for new directions      ##
136 REM##      line 48.....or jump into "hyperspace"      ##
137 REM##      line 49.....If no inkey: go down!      ##
138 REM##      line 50-52...Test for winner      ##
139 REM##      line 53-72..."Rolls" winner-display      ##
140 REM##      Leif Rasmussen Parkvej 1 DK-4534 Hørve      ##
141 REM*****
142 REM*****

```

```
>>>
10REM *****
20REM ***  DISASSEMBLER  ***
30REM ***  Copied out of  ***
40REM ***  a HCC bulletin ***
50REM *****
60REM ***  Rewritten by  ***
70REM ***  S.J. Voortman ***
80REM ***  Beatrixweg 28 ***
90REM ***  3253 BB OUDDORP ***
100REM ***  01878 3113 ***
110REM *****
120
130MODE6:FX5,0
140REM Printer off
150PROCarray:FX200
160REM Capital (upper case) letters
170
180VDU14.3:ON ERROR PROCfout:END
190REM Set paged mode
200PROCinputs
210PRINT"Printer on.":d=GET:IF d<>89 GOTO 280:REM 89 is 'Y'
220VDU15:FX5,1
230REM Printer on
240VDU2,1,27,1,78,1,3
250REM Skip over perforation
260VDU2,1,27,1,87,1,1
270REM Print Enlarged
280PRINT"DISASSEMBLER"
290PRINT"&"+ST$:" - &":EN#""
300VDU2,1,27,1,87,1,0
310REM Reset to normal print style
320
330REPEAT
340OC%=7(S):PRINT:RIGHT$("000"+STR$(S),4):TAB(E):
350MN#=#(OC%)
360T=A(OC%)
370PROCaddress
380PROCwrite
390UNTIL S=EN OR INKEY$(5)="@"
400VDU1,10,2,1,27,1,14:PRINT"End"
410REM One line Enlarged
420VDU15,3:END
430REM Reset paged mode
440
450DEFPROCaddress
460P$="":O$="":AC$="":OP=0
470IF M$(OC%)="" MN#="???":OP=7(S):PROConebyte:S=S+1:ENDPROC
480IFT=0 P$="#&":PROCTwobyte:S=S+2
490IFT=1 P$="&":PROCTwobyte:S=S+2
500IFT=2 P$="&":PROCThreebyte:S=S+3
510IFT=3 P$=" ":PROConebyte:S=S+1
520IFT=4 P$=" ":PROConebyte:S=S+1
530IFT=5 P$="(&":O$=",&":PROCTwobyte:S=S+2
540IFT=6 P$="(&":O$="),&":PROCTwobyte:S=S+2
550IFT=7 P$="&":O$=",&":PROCTwobyte:S=S+2
560IFT=11P$="&":O$=",&":PROCTwobyte:S=S+2
570IFT=8 P$="&":O$=",&":PROCThreebyte:S=S+3
580IFT=12P$="&":O$=",&":PROCThreebyte:S=S+3
590IFT=9 P$="(&":O$=")":PROCThreebyte:S=S+3
600IFT=10P$="&":PROCTwobyte:S=S+2
610ENDPROC
```

DE 6502 KENNER

```
620
630DEFPROCwrite:IF T=10 PROCrel
640PRINTTAB(15);MN$;TAB(19);P$;:IF T<>3:IF T<>10:IF MN$<>"???" THENPRINT:~G$;:
650PRINTTAB(28);AC$
660ENDPROC
670
680DEFPROCarray
690DIM T$(255),M$(255),A(255)
700REPEAT:READ U,V$,W:M$(U)=V$:A(U)=W:UNTIL V$=""
710U=0:REPEAT:T$(U)=STR$(U):U=U+1:UNTIL U=255:ENDPROC
720DATA 8,0,BRK,3,8,1,ORA,5,8,5,ORA,1,8,6,ASL,1,8,8,PHP,3
730DATA 9,ORA,0,10,ASL,4,13,ORA,2,14,ASL,2,16,BPL,10
740DATA 17,ORA,6,21,ORA,7,22,ASL,7,24,CLC,3,25,ORA,12
750DATA 29,ORA,8,30,ASL,8,32,JSR,2,33,AND,5,36,BIT,1
760DATA 37,AND,1,38,ROL,1,40,PLP,3,41,AND,0,42,ROL,4
770DATA 44,BIT,2,45,AND,2,46,ROL,2,48,BMI,10,49,AND,6
780DATA 53,AND,7,54,ROL,7,56,SEC,3,57,AND,12,61,AND,6
790DATA 62,ROL,8,64,RTI,3,65,EOR,5,69,EOR,1,70,LSR,1
800DATA 72,PHA,3,73,EOR,0,74,LSR,4,76,JMP,2,77,EOR,2
810DATA 78,LSR,2,80,BVC,10,81,EOR,6,83,EOR,7,86,LSR,7
820DATA 88,CLI,3,89,EOR,12,93,EOR,8,94,LSR,2,96,RTS,3
830DATA 97,ADC,5,101,ADC,1,103,ROF,1,104,PLA,3,105,ADC,0
840DATA 86A,ROF,4,86C,JMP,9,86D,ADC,2,86E,ROF,2,870,BVS,10
850DATA 871,ADC,6,875,ADC,7,876,ROF,7,878,SEI,3,879,ADC,12
860DATA 87D,ADC,8,87E,ROF,8,881,STA,5,884,STY,1,885,STA,1
870DATA 886,STX,1,888,DEY,3,88A,TXA,3,88C,STY,2,88D,STA,2
880DATA 88E,STX,2,890,BCC,10,891,STA,6,894,STY,7,895,STA,7
890DATA 896,STX,11,898,TYA,3,899,STA,12,89A,TXS,3,89D,STA,8
900DATA 8AC,LDY,0,8A1,LDA,5,8A2,LDX,0,8A4,LDY,1,8A5,LDA,1
910DATA 8A6,LDX,1,8A8,TAY,3,8A9,LDA,0,8AA,TAX,3,8AC,LDY,2
920DATA 8AD,LDA,2,8AE,LDX,2,8B0,BCS,10,8B1,LDA,6,8B4,LDY,7
930DATA 8B5,LDA,7,8B6,LDX,11,8B8,CLV,3,8B9,LDA,12,8BA,TSX,3
940DATA 8BC,LDY,6,8BD,LDA,8,8BE,LDX,12,8C0,CFY,0,8C1,CMP,5
950DATA 8C4,CPY,1,8C5,CMP,1,8C6,DEC,1,8C8,INY,3,8C9,CMP,0
960DATA 8CA,DEX,3,8CC,CPY,2,8CD,CMP,2,8CE,DEC,2,8D0,BNE,10
970DATA 8D1,CMP,6,8D5,CMP,7,8D6,DEC,7,8D8,CLD,3,8D9,CMP,12
980DATA 8DD,CMP,8,8DE,DEC,8,8E0,CPX,0,8E1,SBC,5,8E4,CPX,1
990DATA 8E5,SBC,1,8E6,INC,1,8E8,INX,3,8E9,SBC,0,8EA,NOP,3
1000DATA 8EC,CPX,2,8ED,SBC,2,8EE,INC,2,8F0,BEG,10,8F1,SBC,6
1010DATA 8F5,SBC,7,8F6,INC,7,8F8,SED,3,8F9,SBC,12,8FD,SBC,8
1020DATA 8FE,INC,8,8FF,0
1030
1040DEFPROCinputs
1050INPUT"Start address &"ST$
1060S=EVAL("&"ST$):IF S>&FFFF PROCinputs
1070INPUT"End address &"EN$
1080EN=EVAL("&"EN$):IF EN>&FFFF OR EN<S PROCinputs
1090ENDPROC
1100
1110DEFPROCthreebyte
1120PRINT:RIGHT$("0"+STR$(S),2);TAB(8);RIGHT$("0"+STR$(S71),2);TAB(11);RI
("0"+STR$(S72),2);
1130PROCpeek1:PROCpeek2:PROCpeek3
1140QP=((S72)*256)+(S71)
1150ENDPROC
1160
1170DEFPROCtwobyte
1180PRINT:RIGHT$("0"+STR$(S),2);TAB(8);RIGHT$("0"+STR$(S71),2);
1190PROCpeek1:PROCpeek2
1200QP=(S71)
1210ENDPROC
```

```

1220
1230DEFFPROConebyte;IF MN#="000" PRINTRIGHT$("00"+T$(DCX),2)+":60TO (CE)
1240PRINT;RIGHT$("00"+T$(DCX),2)+": "
1250DEFFPROCone=1
1260ENDPROC
1270
1280DEFFPROC=1
1290IF (7(B-1))>127 P#="A"+STR$(7(B-1)-(BFF-DF))
1300IF (7(B-1))<128 P#="A"+STR$(B+DF)
1310ENDPROC
1320
1330DEFFPROCout
1340PRINT;VDUC,7;IF BFF=17 THEN END
1350REPORT;PRINT" at line "ERL
1360ENDPROC
1370
1380DEFFPROC=1;IF 7(B)>11 AND 7(B)<127 THEN AC=AC+STR$(7(B)) ELSE AC=AC+"
1390ENDPROC
1400DEFFPROC=1;IF (B71)>11 AND (B71)<127 THEN AC=AC+CHR$(B71) ELSE AC=AC+"
1410ENDPROC
1420DEFFPROC=1;IF (B72)>11 AND (B72)<127 THEN AC=AC+CHR$(B72) ELSE AC=AC+"
1430ENDPROC

```

Redactioneel

Op het moment dat ik dit schrijf, half juli '86, zijn er 62 nieuwe leden toegetreden tot onze club, waaronder 39 uit Holland, 14 uit België, 3 uit Engeland, 2 uit Zweden, 2 uit Denemarken, 1 uit Duitsland en 1 uit Spanje. In het gehele jaar 1985 schreven we in totaal 61 nieuwe leden in. De werving van nieuwe leden - we zijn niet zo rijk dat we daar geld in kunnen steken - verloopt dit jaar blijkbaar beter dan in 1985. Dat is hoopgevend. Een opvallend verschil met voorgaande jaren is dat in 1985 Elektuur in haar computer-specials onze naam noemde, en in 1986 deed Elektor Electronics in Engeland dat nog eens heel dik over met een halve pagina gratis "advertisement". Het gevolg van dit laatste was brieven om meer informatie uit diverse Europese landen, maar ook uit Israël, Iran, Irak, India, Zuid-Afrika etc. Het gevolg van beide was een gunstig effect op het nu bestaande resultaat. Veel leden van onze club betrekken hun spullen van Elektuur, en wij van onze kant moedigen dat ook aan, omdat we er ook veel van kunnen leren. Elektuur en onze club hebben een al jaren bestaande onafhankelijke band met elkaar, het is gunstig voor ons beiden.

Eenzelfde onafhankelijke band hebben we al jaren met de Hobby Computer Club, de HCC, tot uiting komend op de jaarlijkse hobbycomputerbeurs in Utrecht, waar ook onze club altijd een gratis plaats kreeg, terwijl o.a. onze club ook in haar Nieuwsbrief werd vermeld in de gele pagina's en wij ook konden aankondigen wanneer onze bijeenkomsten werden gehouden. Enige tijd geleden al werd dit laatste aan ons ontnomen. Dat was een beslissing waarbij de HCC het niet nodig achtte ons daaromtrent te consulteren. Met hun brief van 11 juli 1986 schrijft de HCC ons nu een uitnodiging voor de HCC dagen op 21 en 22 november 1986.

"Voor het eerst sinds jaren vragen wij van u echter een bijdrage in de kosten. U zult niet worden belast voor het 'commerciële' tarief (red: f 900,-), doch voor een bedrag van f 250,- per kraam (excl. btw.), schrijft men verder. Ten opzichte van vorig jaar mogen bezoekers, dus ook onze leden een toegangsprijs betalen die 50% hoger ligt dan vorig jaar: f 7,50.

Afgezien van de vraag of dit soort prijsmaatregelen in Nederland niet onder de prijswetgeving valt, het heeft bij mij de vraag opgeroepen of leden van onze club zich af zullen vragen hoe ernstig de gevolgen zijn. Leden van onze club zien ons natuurlijk graag verschijnen op deze manifestatie. Maar dit soort vercommercialisering in de HCC zou wel eens kunnen betekenen dat wij dat geld liever aan andere dingen besteden. Het bestuur zal hieromtrent, geschrokken als ze is, een ernstige gedachtenwisseling houden.

De HCC-dagen waren voor ons een gelegenheid, die de begroting toch al ernstig aansprakten, maar het uitdragen van onze naam, de presentatie van hetgeen onze club presteert, dat zijn belangrijke overwegingen die de doorslag gaven. Er zijn altijd nog andere wegen als we dat onverhoopt niet meer middels de computerdagen in Utrecht kunnen doen. Maar het doet wel pijn als het niet meer kan. W.L. van Pelt.

Please, send your articles, programs, etc. to the editors office, printed on white paper A-4 format, 8 LPI (lines/inch), 68 lines/page max., a new ribbon on your printer. If you are not yet the owner of a printer, we will help you with typing the text.

New: Complete source-listing of the Micro-Ware Assembler/Disassembler/Editor for Elektor's Octopus/EC65-computer. Documentated by Marc Lachaert, Belgium. Send cheque of Hfl. 74,50 to W.L.v.Pelt (eurocheque 65,00)

THE MC68000 MICROPROCESSOR; A NEW PROCESSOR IN OUR CLUB.

By: Gert van Opbroek
Bateweg 68
2481 AN Houburgge.
Tel 01729-8636

1. Introduction.

As already written by Willem van Pelt in the 6502-Kenner number 41, the executive committee of our club wants to support some new processors. He has written that all processors beginning with a 6 are welcome in this club. Because I am a member of committee and I possess, since January 1986, a 68000 system I was asked to write something about this processor and its differences with our well-known 6502.

First of all, I want to tell why we want to support all processors beginning with 6. Willem van Pelt has written the 6X(C)XXX-processors. That means the processors:

- 1) the 6800-family: that means 6800, 6809
- 2) the 6500-family: 6502, 6510, 65C02
- 3) the 65SC802, 65SC816
- 4) the 68000-family: 68000, 68010

Questions you can ask are: why these processors, what do they have in common and what makes them different from other processor families?

The above mentioned processors all have memory mapped I/O. This means that peripherals, such as I/O chips, FDC's and so on, are part of the memory. Other processors, such as the 8080 and 8086 families do have a separate I/O space and IN and OUT instructions. With memory mapped I/O, you can access the peripherals just like you access memory.

Another thing that these families have in common is the fact that the control bus and timing of all the families is the same. The 68000 family forms, partly, an exception. It has a mode compatible with the 6800 but it also has an asynchronous mode. In a later section, I shall describe this difference. Because the timing and control bus of the families are the same, peripherals from one family can easily be interfaced with processors from the other. So a 6522 VIA can easily cooperate with a 68000 processor.

When we look at the register structure, the instruction set and addressing modes of the 6X(C)XXX families, you will see that although there are smaller or greater differences, there are many similarities. The 65SC802 and 65SC816 even can emulate a 6502.

In the next I shall describe the 68000 processor. A review of the 65C02 can be found in [1], of the 65SC816 in [2] and [3]. In November 1985 Nico de Vries has given a small lecture on this processor on the club meeting in Rijswijk.

Within this article I shall compare the 68000 processor with the good old 6502. People who are working with PDP-11 and VAX computers will find that certain features of the 68000 are comparable with features of those computers but because most members of our club do not have knowledge of these minicomputers, I shall not compare the 68000 with these machines.

2. HARDWARE.

Today the 68000 family has the following members:

- a) The 68000. This processor has all the features of a 68000 but has an 8 bit databus. The address bus of this processor is 24 bits wide.
- b) The 68000. This is the base processor of the 68000 family. It has a full 16 bits databus and a 24 bits address bus.
- c) The 68010. This processor has some differences in the exception scheme and is, because of a more efficient microcode and a higher clock frequency, faster.
- d) The 68020. I do not have information about this processor.

In fig. 1 the pin-out of a 68000 processor is drawn.

D4	1	64	D5
D3	2	63	D6
D2	3	62	D7
D1	4	61	D8
D0	5	60	D9
-AS	6	59	D10
-UDS	7	58	D11
-LDS	8	57	D12
R/-W	9	56	D13
-DTACK	10	55	D14
-BG	11	54	D15
-BGACK	12	53	GND
-BR	13	52	A23
Vcc	14	51	A22
CLK	15	50	A21
GND	16	49	Vcc
-HALT	17	48	A20
-RESET	18	47	A19
-VMA	19	46	A18
E	20	45	A17
-VPA	21	44	A16
-BERR	22	43	A15
-IPL2	23	42	A14
-IPL1	24	41	A13
-IPL0	25	40	A12
FC2	26	39	A11
FC1	27	38	A10
FC0	28	37	A9
A1	29	36	A8
A2	30	35	A7
A3	31	34	A6
A5	32	33	A5

FIG. 1. (note - xxx means NOT(xxx))

In fig. 1, we can see that the 68000 has a full 24-bit address bus. The signal A0 is not available so the 68000 always addresses cells of 16 bit. It also has a 16 bit data bus. The signal -AS (address strobe) indicates a valid memory address on the address bus. The signals -UDS (upper data strobe) and -LDS (lower data strobe) indicates that the processor reads or writes the upper resp. lower databyte. When both signals go down, the processor transfers a full 16-bit dataword.

The CLK input is used by the clock. 68000 processors are available for processor clocks up to 12 MHz. The most commonly used version is the 8 MHz version.

The -BR (bus request), -BG (bus grant) and -BGACK (bus grant acknowledge) are used for DMA devices to become master of the address and data bus.

The signals FC0, FC1 and FC2 indicate the processor status and type of the execution cycle of the processor.

When the -HALT signal is used as input, the processor will stop after completion of the current bus cycle, when the processor stops by itself the -HALT signal is outputted by the processor to indicate that the processor has stopped execution.

The 68000 has an asynchronous bus. That means that after an address is put on the address bus the processor waits until external hardware (memory, devices) has indicated that there are valid data on the data bus (read cycle) or the hardware has taken the data from the data bus. This hardware indicates this situation by giving the -DTACK (data transfer acknowledge) signal. When this signal does not come, the processor waits until "Sint Juttemis". This situation can happen when we address, somewhere in the 16 Mb address space, non-existing memory. For this situation, the processor has a bus error (-BERR) input. When this signal goes down, current bus cycle is terminated. When the -HALT signal also goes down, the bus cycle is rerun when the -HALT signal goes high again. When the -HALT signal does not go down, the processor performs a BUSERROR exception. The -DTACK, -BERR and -HALT signals must be made by external hardware.

The 68000 has 7 interrupt levels. These levels are coded by the -IPL0, -IPL1 and -IPL2 signals. Interrupt level 7 (IPL0, IPL1 and IPL2 go down together) is called NMI (Non Maskable Interrupt).

In a next section, I will write more about exceptions, of which interrupts forms one fall.

As I have already written, peripherals of the 6500 and 6800 family can be interfaced with a 68000. As is well known, these families do not support an asynchronous bus structure. For this reason, the 68000 supports, besides the asynchronous structure, also a synchronous bus structure. This interface is formed by the signals: E (Enable), -VPA (Valid Peripheral Address) and -VMA (Valid Memory Address). The E-signal is used as clock signal for those peripherals (PH1-2). This signal is six CLK periods low and four CLK periods high. This means that in a 8 MHz system this signal has a frequency of 800 kHz. (1/10 of CLK). The -VPA signal indicates that the synchronous mode is chosen (VPA = 0: synchronous, VPA = 1: asynchronous). The -VMA signal indicates that there is a valid address and (for a write operation) valid data on the bus. This signal is synchronized with the E signal. The -DTACK is not used in a synchronous operation.

The timing is as follows:

When synchronous operation is preferred, the system responds after a -AS signal with a -VPA signal. The processor synchronizes then its signals with the E signal and gives a synchronized -VMA signal. This signal has to be used to generate a chip select for the addressed peripheral.

3. PROGRAMMING MODEL.

In this section, I shall describe the register structure of the 68000 processor.

In fig. 2 the structure of the registers is drawn.

As already written, the processor has a 24-bit address bus. To address this amount of memory, the processor has a 32-bit program counter. From this register, the 8 most significant bits are ignored. All addresses can also be 24 bit. An address is stored in a so called long word (4 bytes) and, just like with the program counter, the 8 most significant bits of an address are ignored.

LONG WORD (32 bit)
WORD (16 bit)
BYTE (8 bit)

ADDRESS REGISTERS

A0	
A1	
A2	
A3	
A4	
A5	
A6	
A7	STACK POINTER (USER)
A7	STACK POINTER (SUPERVISOR)

DATA REGISTERS

D0
D1
D2
D3
D4
D5
D6
D7

PROGRAM COUNTER

PC

STATUS REGISTER

T-S---III---XNZVC
SYSTEM USER

FIG. 2 INTERNAL REGISTERS

As can be seen in fig. 2, the 68000 has 16 registers. The eight address registers are meant for address computations but can also be used for other purposes. Address registers can only be used for word and longword operations, data registers can be used for byte, word and longword operations. When we move (store) a word in an address register, the operand is sign extended, this means that the most significant bit of the (word) operand is also stored in the 16 most significant bits of the address register. When we move a byte or word to a data register, the other bits of the register are unaffected. This feature can have very funny results:

```
MOVE.L #000,A0
MOVE.W #0FF,A0
CMPLA.L #0FF,A0
BEQ.B LABEL
```

In the above example, the branch will not be executed, the contents of register A0 = \$FFFF. When we use D0 instead of A0, the branch will be executed because the contents of the register will be \$00FF.

Register A7 is the stackpointer. As you can see in fig. 2, the 68000 has 2 stackpointers. The processor has two operation modes: User mode and Supervisor mode. In Supervisor mode, the processor can execute more instructions. Each mode has its own stackpointer. When the processor is in supervisor mode, there is an instruction to move the user stackpointer to or from an address register. When the processor is in user mode, you can not access the supervisor stack pointer. When an instruction can only be executed in supervisor mode, the instruction is called privileged. There are seven privileged instructions on the 68000.

The status register has a system byte and a user byte. The user byte contains the condition codes as can also be found in the 6802. There is one new condition code: X. This code is a copy of the carry bit, affected only by those instructions which can be used in multi precision operations (i.e. ADD, SUB, ASL,).

The 68000 does not have a decimal (D-) flag. For BCD-operations, the 68000 has some (extra) BCD-instructions.

The system byte contains an interrupt mask (III) for the seven interrupt levels. All interrupts greater than mask value are processed. The S-flag is set when the processor is in supervisor state and when the T-bit (trace) is set, a trace exception will take place after each instruction.

4. ADDRESSING MODES.

An instruction for the 68000 consists of an opcode and zero, one or two operands. The description of these operands is the so-called effective address. The addressing mode is the mode of description of this operand.

The 68000 has 13 addressing modes.

- Data Register Direct: The operand is a data register.
- Address Register Direct: The operand is an address register.
- Address Register Indirect: The address register contains the address of the operand.
- Address Register Indirect with Post-Increment: This is the same as the Address Register Indirect mode but the address in the address register is incremented after the effective address is calculated. For byte instructions

the address is incremented by 1, for word instructions by two and for longword instructions by four.

- Address Register Indirect with Pre-Decrement: This addressing mode is the opposite of the Address Register Indirect mode with Post Increment. These modes are used for stack addressing:

```
MOVE D0,-(A7) ; Push D0
MOVE D0,(A7)+ ; Pull D0
```

There are of course much more applications for these addressing modes. For instance block move:

```
SOURCE: EQU $10000 ; SOURCE ADDRESS
DESTIN: EQU $30000 ; DESTINATION ADDRESS
ENDSRC: EQU $2FFFF ; END SOURCE
START:
MOVEA.L #SOURCE,A0
MOVEA.L #DESTIN,A1
LOOP:
MOVE.W (A0)+,(A1)+
CMPLA.L #ENDSRC,A0
BLS.B LOOP
RTS
```

- Address Register Indirect with Displacement: In this mode, a 16-bit signed constant is added to the indirect address to give the address of the operand.
- Address Register Indirect Addressing with Index: The contents of an address or data register is added to another address register. This gives the address of a memory cell with a pointer. An 8 bit signed displacement is added to this pointer to give the address of the operand.
- Absolute Addressing: This addressing mode has two forms: Absolute Short Addressing and Absolute Long addressing. With Absolute Long addressing, we can address the complete memory. With Absolute Short addressing, we can address from \$000000 to \$007FFF and \$FF8000 to \$FFFFFF. With short addressing, the address is a 16 bit number, with long addressing, the address is a full 24 bit number.
- Program Counter with Displacement: a signed 16 bit quantity is added to the program counter to give the address of the operand. When this mode is used to address variables in memory, the resulting code is position independent; this means that the program can run anywhere in memory without the need to reassemble it.
- Program Counter with Index: This addressing mode is the Address Register Indirect mode with Index with the program counter instead of an address register. The contents of an address or data register is added to the program counter to give the address of a pointer. A signed 8 bit quantity is added to this pointer to give the address of the operand.
- Immediate addressing: The operand is given in the instruction as a constant.
- Status Register Addressing: This mode is used to access the status register of the 68000. We can only access the status register with AND, OR and EOR instructions to set or clear flags in the status register. This mode can only be used in destination operands.

When we compare these addressing modes with the modes of the

6502, we see a lot of differences. The 68000 does not have a zero page although the addresses \$000000 to \$007FFF and \$FF8000 to \$FFFFFF can be addressed with the Absolute Short mode, something like a zero page mode. Absolute addressing on the 6502 is Absolute Long addressing on the 68000 and immediate addressing is the same on both processors with this difference that the 68000 has 8, 16 and 32 bit constants. The 68000 does not have a indexed addressing like the Zero,X and Zero,Y and Absolute,X and Absolute,Y on the 6502. The 68000 supports Indirect addressing, Indexed,Indirect (like X-indexed,Indirect) and Program Counter with Displacement (Relative addressing on the 6502). This last mode is not only used in branches but also in several other instructions. Indirect,Y-Indexed on the 6502 does not have a counterpart on the 68000. This means that when we want to simulate this mode on a 68000 we have to do address calculations in an address register.

5. INSTRUCTION SET.

In general, a computer has four types of instructions:

- 1) Data movement instructions.
- 2) Instructions that combine two operands to produce a result.
- 3) Instructions with one operand to produce a result.
- 4) Instructions that control the program flow.

I shall give a brief comparison of the 68000 and the 6502 for the instructions in these four groups:

5.1 Data movement.

The 6502 has the following instructions in this class:

LDA, LDX, LDY, PHA, PHP, PLA, PLP, STA, STX, STY, TAX, TAY, TSX, TXA, TXS, TYA.

The 68000 has the following instructions in this class:

EXG: Exchange the contents of two registers.
 LEA: Load an address register with an effective address
 MOVE: Move data from the source to the destination; source and destination can be address and data registers, status register, stack, memory and peripherals.
 MOVEQ: Move a small (8 bit) signed constant to the destination.
 MOVEM: Move one or more registers to or from an effective address. (MOVEM D0-D7/A0-A5, -(A7),
 MOVEM (A7)+, D0-D7/A0-A5)
 PEA: Push an effective address on top of the stack.
 SWAP: Swap the low and high order word in a register.

5.2 Two operand instructions:

In this class fall the arithmetical and logical instructions:

6502: ADC, AND, CMP, CPX, CPY, EOR, ORA, SBC.

68000: ADD, AND, CMP, DIV (Divide signed and unsigned), EOR, MUL (Multiply signed and unsigned), OR, SUB.

5.3 One operand instructions.

6502: ASL, BIT, CLC, CLD, CLI, CLV, DEC, DEX, DEY, INC, INX, INY, LSR, ROL, ROR, SEC, SED, SEI.

68000: ASL, ASR, BCHG (Bit change), BCLR (Bit clear), BSET (Bit set), BTST (Bit test), CHK (Check a data register against bounds), CLR, EXT (Sign extend), LSL, LSR,

NEG (Negate 2's complement), NOT (1's complement), ROL, ROR, Scc (Set a byte to FF in an effective address when condition code set), TAS (Test and set), TST.

5.4 Program control.

6502: Bcc (Branch on condition), BRK, JMP, JSR, RTI, RTS.

68000: Bcc, DBcc (Decrement a data register and branch until -1 in this register or cc is false; loop with maximum count), BSR (Branch to subroutine byte and word displacement), JMP, JSR, RTE, RTR, (Load condition codes and program counter from the stack), RTS, TRAP, TRAPV.

5.4 Miscellaneous instructions.

In this class fall instructions that not can be classified in one of the other classes:

6502: NOP.

68000: NOP,
 LINK, UNLINK: The link instruction allocates a stack frame for local variables in high level languages, the unlink instruction releases the stack frame.
 RESET: Reset all external devices.
 STOP: Change exception mask and wait for an exception.

5.5 Remarks

As we all know, the 6502 has a decimal mode. When this mode is set, the arithmetic instructions work in BCD format. The 68000 has separate instructions for BCD calculations: ABCD, NBCD, SBCD.

For multiple precision operations, the 6502 uses the carry bit. With ADC and SBC this carry is used as carry bit. The 68000 does not add or subtract the carry bit in an ADD or SUB operation. It has separate instructions ADDX, NEGX and SUBX instructions. These instructions use the X bit that contains a copy of the C bit in ADD, SUB and NEGX instructions.

For immediate operations with small constants, some instructions have a quick mode, just like MOVEQ. In this case, the instruction is shorter than with a normal immediate operand.

When a data register has to be rolled or shifted, the instruction can contain a constant for the number of bits the register has to be rolled or shifted, or the instruction contains another data register in which this constant is stored. In this last mode, a roll or shift instruction is a two operand instruction.

6. EXCEPTIONS.

One of the most complicated things in the 68000 processor is exception processing. The 6502 has only two interrupts, one break instruction and a reset. The 68000 has much more exceptions.

- A hardware reset. The SSP (supervisor stack pointer) is loaded from address \$000000 (vector 0), the PC from address \$000004 (vector 1).

- A BUSERR exception, used when non-existent memory is addressed.

- Address error: word and longword operations on an odd address result in this exception.
- Illegal instruction.
- Divide by zero.
- CHK instruction. This instruction checks the contents of an address register against bounds. Errors result in this exception.
- TRAPV exception. In the TRAPV instruction exception occurs when the overflow bit is set.
- TRAP instruction. In this instruction a constant 0 to 15 is given. This is the index in the vector table for TRAP instructions. (Address \$000000 to \$0000BF).
- Privilege violation. The 68000 can work in supervisor mode and in user mode. In user mode not all instructions are allowed. When we try to execute such a privileged instruction in user mode, an exception occurs.
- TRACE, when the T-bit is set.
- Line 1010 and line 1111 emulator exceptions on illegal instructions with opcode Axxx and Fxxx respectively. These exceptions can be used to emulate non-existing instructions in software.
- Interrupts. The 68000 has 7 vectored interrupts. With the I-bits in the status register, the priority of the processor is chosen. Each interrupt with higher priority is affected and an exception occurs. During this exception, the mask in the status register is set to the priority of the interrupt. Priority 7 can not be masked out, this is the NonMaskable Interrupt. The 68000 has 7 autovector interrupt vectors, one for each priority. When during the interrupt acknowledge the -VPA signal goes down, the processor takes the address in the corresponding vector to find the interrupt service routine. This method is used for interrupts from devices of the 65xx and 68xx series. The 68000 has also 192 user interrupt vectors (64 to 255). During the interrupt acknowledge, the device puts the vector number on the data bus and the processor finds the address of the service routine in the corresponding entry in the vector table. During this interrupt acknowledge the address on the address bus is \$FFFFn where n is the priority $\times 2$. Although this address is not meant to address ram, it can be used to read the vector from ram on these locations.
- Initialized interrupt vector. This interrupt occurs when a 6xxxx peripheral gives an interrupt and the interrupt vector register in the device is not loaded.
- Spurious interrupt when during a interrupt acknowledge no device responds by putting the interrupt vector on the databus and giving a -DTACK (user interrupt) or giving the -VPA signal (autovector interrupt) a spurious exception occurs.

As you see, the exception capabilities go much further on the 68000 than on the 6502. You have however to pay for these features with a very complicated interrupt acknowledge sequence. I hope, that my description of the exception processing does not contain significant errors, I am not sure I understand the interrupt acknowledge completely.

7. SYSTEMS WITH 68000.

There are already a few complete systems with a 68000 on the market. In this section, I shall describe some of these systems.

- 7.1 Sinclair QL. The Sinclair QL was one of the first computers with a 68000 on the hobby market. It contains a 68000 processor. As far as I know, the computer is not very popular, I think the main reason for this is the fact that there is not many software for this machine.
- 7.2 The Apple Macintosh. This computer is not meant for the amateur. It contains a 68000 processor and has a operating system with so called icons. That means that almost anything is menu driven. When you want to do something you point to the associated icon with the mouse, press the button and the program starts.
- 7.3 The Atari 520 ST. I think that this computer is the IBM PC for the 68000 fans. It has a 68000 processor. It supports very nice graphics and there is already some software for it and the number of programs is growing each month. Beside these facts, the computer is rather cheap.
- 7.4 The Commodore Amiga. Also a 68000 processor, a very new machine. It has more features than the Atari but it also costs a lot of money.
- 7.5 The MC68000, my computer. This computer is published by the german magazine MC. It is a single board computer just like the Junior with a 68000 two 6522 VIA's, a 6845 video controller and 128 or 512 KB ram. On board there are 8 expansion slots. Already available are a floppy controller, 2 MB ram board, 8 line RS232 interface and operating systems CPM-68K (single user) and OS9-68K (multy user). The only problem with software for this computer is the price.
- 7.6 The c't 68000. Just like the MC68000 is this a project of a german magazine: C'T. I do not know any details of this computer.
- 7.7 The EC68000. A new project of Elektor with a 68000 and a 6809 integrated in one computer. It is published in Elektor Computer Special 3.
- 7.8 All kinds of VME-bus machines with operating systems UNIX, OS9-68K, VERSADOS PDOS etc. These machines are meant for professional applications and are rather expensive. These computers are becoming industrial standard for controlling processes.
- 7.9 68000 based computers with UNIX-V operating system like HP Integral and AT&T UNIX PC. These computers are the 68000 versions of the IBM PC.
- 7.10 In the near future, all kinds of upgrade boards with 68000 can be bought for the Apple II and IBM PC.

8. Literature.

- 1) Herman Burgers: 6502 Kenner 31, april 1984.
- 2) Byte August 1984
- 3) Byte September 1984

There are many books for the 68000. Each month there are issued new ones. I can not give a review of books, the best thing to do is, go to your local book shop and look for a good book. Elektor has also issued two books on the 68000, and MC has issued a special on the MC68000.

NUMBER THE JUNIOR PROMOTING CY (W&J) 21:55:19 - TUE 17/JUN/86 PAGE 01

```

00010: *****
00020: *
00030: *          N U M B E R S
00040: *
00050: *****
00060:
00070:
00080: Routines to handle the input of numbers from the key-
00090: board and to convert them between Decimal and Hexade-
00100: cimal notations.
00110:
00120:
00130: D800      NUMBER ORG      $D800
00140:
00150:
00160:      DB 00  DECFLG *      $00DB  decimal flag: 00=dec, FF=hexadecimal
00170:
00180:      ED 00  ONE *        $00ED  buffer for decimal 5-digit number
00190:      EE 00  TWO *        $00EE  +01
00200:      EF 00  TRE *        $00EF  +02
00210:
00220:      F8 00  INL *        $00F8  +0B ($00FB) buffer for hexadecimal 4-digit num
00230:      F9 00  INH *        $00F9  +01
00240:
00250:      FC 00  TEMP *       $00FC  temporary for digit
00260:      FE 00  NIBBLE *     $00FE  number of digits
00270:
00280:      54 EF  DUPLEX *     $EF54
00290:
00300:      *** CLOCK ADDRESSES ***
00310:
00320:      C8 EF  CLKADR *     $EFC8
00330:      C9 EF  CLKDAT *     $EFC9
00340:
00350:      *** EXTERNAL SUBROUTINES ***
00360:
00370:      A1 29  PRINTV *     $29A1  print a string on the VDU
00380:      7A F1  BELL *       $F17A  ring the bell
00390:      5D F3  NPROCHA *    $F35D  print a character
00400:      FE F4  CRLF *       $F4FE  print a CR and a LF
00410:      06 F5  PRBS *       $F506  print a backspace
00420:      09 F5  PRSP *       $F509  print a space
00430:      75 F5  PRBYT *      $F575  print a byte
00440:      7E F5  PRNIBL *     $F57E  print a nibble
00450:      1A F6  ASHETT *     $F61A  ASCII-hexadecimal conversion
00460:      1B FE  RECCHA *     $FE1B  receive 1 character from keyboard
00470:
00480:
00490:
00500:
00510:      *** RECEIVE AN HEX NUMBER (UP TO 4 DIGIT) TO INH,INL ***
00520:      *****
00530:
00540:      Duplex flag must be cleared (00)
00550:      Input ends with a space, or
00560:      A space will be printed after the 4th digit
00570:
00580:      D800 A0 03  RECHEX LDYIM $03  up to 4 digits
00590:      D802 2C    = $2C  (BIT) jump next two bytes
00600:
00610:      D803 A0 01  RECBYT LDYIM $01  2 digits
00620:      D805 A9 FF  LDAIM $FF  hexadecimal!
00630:      D807 D0 04  BNE RECDIG always
00640:
00650:
00660:
00670:
00680:      *** RECEIVE A DECIMAL NUMBER (TO 5 DIGIT) TO TRE,TWO,ONE
00690:      *****
00700:
00710:      Duplex flag must be cleared
00720:      Input ends with a space, or
00730:      A space will be printed after the 5th digit
00740:
00750:      D809 A0 04  RECDEC LDYIM $04  up to 5 digits
00760:      D80B A9 00  LDAIM $00  decimal!
00770:
00780:

```

NUMBER THE JUNIOR PROMOTING CY (W&J) 21:56:35 - TUE 17/JUN/86 PAGE 02

```

00790: *** Receive a digit ***
00800: Error-free numeric input routine:
00810: - check for a digit, decimal or hexadecimal
00820: - do not accept any other character, except
00830: - a SPACE to finish input
00840: - or a BACKSPACE to correct a previous digit.
00850: - check for a "correction" on a digit that doesn't exist
00860: - "beep" the BELL on error.
00870:
00880:
00890: D80D 85 D8 RECDIG STAZ DECFLG set decimal flag
00900: D80F 84 FE STYZ NIBBLE max number of digits
00910: D811 8A TXA save X register
00920: D812 4B PHA
00930:
00940: D813 20 18 FE RECDI JSR RECCHA receive one digit
00950: D816 C9 20 CMPIM SP to finish
00960: D818 F0 4D BEQ SPCEND if it was a SPACE.
00970: D81A C9 08 CMPIM #08 BS to correct
00980: D81C 00 0A BNE RECD if not a BSpace.
00990: D81E C4 FE CPYZ NIBBLE are we at the beginning?
01000: D820 80 13 BCS RECBEL if so, no backspace! (attempt to correct behind
01010: D822 C8 INY the 1st digit)
01020: D823 20 06 F5 JSR PRBS correct the previous digit; V=0
01030: D826 50 EB BVC RECDI always. Receive a new digit
01040:
01050: D828 20 1A F6 RECD JSR ASHETT ASCII to hexadecimal conversion
01060: D82B 80 08 BCS RECBEL if <0 or >F
01070: D82D 24 08 BITZ DECFLG
01080: D82F 70 09 BVS RECVAL if hexadecimal
01090: D831 C9 0A CMPIM #0A
01100: D833 90 05 BCC RECVAL if decimal 0 to 9
01110:
01120: D835 20 20 D9 RECBEL JSR BELLY : error, ring the bell & save Y; C=1
01130: D838 80 D9 BCS RECDI always
01140:
01150: D83A 4B RECVL PHA save nibble
01160: D83B 20 7E F5 JSR PRNIBL print it (DUPLEX is off)
01170: D83E 68 PLA restore it
01180: D83F 85 FC STAZ TEMP
01190: D841 98 TYA get digit count
01200: D842 4A LSR divide by 2; carry=1 if odd; =0 if even
01210: D843 08 PHP save carry status
01220: D844 24 D8 BITZ DECFLG which input buffer?
01230: D846 50 03 BVC RECTAX if decimal.
01240: D848 18 CLC
01250: D849 69 0B ADCIM #0B (INL=ONE+0B) compute hexadecimal-buffer pointer
01260:
01270: D84B AA RECTAX TAX get byte-position (pointer)
01280: D84C 85 ED LDAZX ONE get byte from buffer
01290: D84E 28 PLP restore carry status (even/odd)
01300: D84F 90 0B BCC RCEVEN if even.
01310:
01320: D851 29 0F RECODD ANDIM #0F mask old digit. Y=3 or 1 (odd)
01330: D853 06 FC ASLZ TEMP position correctly the new digit in TEMP
01340: D855 06 FC ASLZ TEMP
01350: D857 06 FC ASLZ TEMP
01360: D859 06 FC ASLZ TEMP
01370: D85B 2C = #2C (BIT) jump next 2 bytes of program code
01380:
01390: D85C 29 F0 RCEVEN ANDIM #F0 mask old digit. Y=2 or 0 (even)
01400: D85E 05 FC ORAZ TEMP include the new one
01410: D860 95 ED STAZX ONE save it (byte)
01420: D862 88 DEY all digits?
01430: D863 10 AE BPL RECDI no; receive more.
01440: D865 30 18 BMI RECD always. YES; stop.
01450:
01460: D867 C8 SPCEND INY
01470: D868 98 TYA number of digits (nibbles) missing.
01480: D869 0A ASLA multiply by 4: compute number of bits.
01490: D86A 0A ASLA
01500: D86B A8 TAY bit count on Y register
01510:
01520: D86C 24 D8 RECROR BITZ DECFLG decimal or hexadecimal? (which buffer?)
01530:
01540: D86E 70 08 RECBVS BVS RECINH if hexadecimal.
01550: D870 46 EF LSRZ TRE rotate nibble (digits) through buffer
01560: D872 66 EE RORZ TWO fill missing digits with zeros
01570: D874 66 ED RORZ ONE
01580: D876 50 04 BVC RECDEY always
01590:

```


NUMBER THE JUNIOR PROMOTING CY (W&J) 21:57:59 - TUE 17/JUN/86 PAGE 03

```

01600: D878 46 F9      RECINH LSRZ INH      idem for decimal
01610: D87A 66 F8      RORZ INL
01620:
01630: D87C 88          RECDEY DEY      all bits moved?
01640: D87D D0 EF      BNE RECBVS
01650:
01660: D87F 68          RECEND PLA      restore X register
01670: D880 AA          TAX
01680: D8C1 4C 09 F5      JMP PRSP      print a SSpace and RTS; V=0
01690:
01700:
01710:
01720:
01730:
01740:
01750:
01760:
01770:
01780: D884 8A          DECHEX TXA
01790: D885 49          PHA
01800: D886 98          TYA
01810: D887 48          PHA
01820: D888 F8          SED
01830: D889 A0 00      LDYIM #00      Y = powers count
01840: D88B 20 AF D8   JSR SUB
01850: D88E 0A          ASLA
01860: D88F 0A          ASLA
01870: D890 0A          ASLA
01880: D891 0A          ASLA
01890: D892 20 AD D8   JSR STOSUB (STAZ INH ...)
01900: D895 05 F9      ORAZ INH
01910: D897 20 AD D8   JSR STOSUB (STAZ INH ...)
01920: D89A 0A          ASLA
01930: D89B 0A          ASLA
01940: D89C 0A          ASLA
01950: D89D 0A          ASLA
01960: D89E 85 F8      STAZ INL
01970: D8A0 20 AF D8   JSR SUB
01980: D8A3 05 F8      ORAZ INL
01990: D8A5 85 F8      STAZ INL
02000:
02010: D8A7 68          DECEND PLA      restore registers
02020: D8A8 A8          TAY
02030: D8A9 68          PLA
02040: D8AA AA          TAX
02050: D8AB D8          CLD
02060: D8AC 60          RTS
02070:
02080: D8AD 85 F9      STOSUB STAZ INH      save nibble
02090: D8AF A2 00      SUB LDXIM #00      X = subtraction times count
02100: D8B1 38          SUBS SEC
02110: D8B2 A5 ED      LDAZ ONE      subtract power from number
02120: D8B4 F9 18 D9   SBCAY POWERS
02130: D8B7 85 ED      STAZ ONE
02140: D8B9 A5 EE      LDAZ TWO
02150: D8BB F9 19 D9   SBCAY POWERS +01
02160: D8BE 90 05      BCC SUBD
02170: D8C0 85 EE      SUBST STAZ TWO
02180: D8C2 E8          INX
02190: D8C3 D0 EC      BNE SUBS
02200: D8C5 C6 EF      DECZ TRE      zero?
02210: D8C7 10 F7      BPL SUBST      if number still positive.
02220: D8C9 E6 EF      INCZ TRE      add power again, if negative.
02230: D8CB A5 ED      LDAZ ONE
02240: D8CD 79 18 D9   ADCAY POWERS
02250: D8D0 85 ED      STAZ ONE
02260: D8D2 8A          TXA      get count
02270:
02280: D8D3 C8          SUBRTN INY
02290: D8D4 C8          INY      increment POWERS pointer
02300: D8D5 60          RTS
02310:
02320:
02330:
02340:
02350:
02360:
02370:

```

*** DECIMAL - HEXADECIMAL CONVERSION ***

Subtract POWERS from the decimal number
until a result negative.

*** HEXADECIMAL - DECIMAL CONVERSION ***

Add powers (hexa)times up to a total decimal number

NUMBER THE JUNIOR PROMOTING CY (W&J) 21:59:11 - TUE 17/JUN/86 PAGE 04

```

02380: D8D6 98      HEXDEC TYA
02390: D8D7 48      PHA
02400: D8D8 8A      TXA
02410: D8D9 48      PHA
02420: D8DA F8      SED
02430: D8DB 20 29 D9 JSR    CLEAR ONE TWO TRE (decimal buffer)
02440: D8DE A0 00 LDYIM $00 Y = powers pointer
02450: D8E0 A5 F9 LDAZ  INH  get high byte
02460: D8E2 F0 03 BEQ    HEXLDY if zero.
02470: D8E4 20 F2 D8 JSR    DECADD add power to number
02480:
02490: D8E7 A0 04      HEXLDY LDYIM $04
02500: D8E9 A5 F8 LDAZ  INL  get low byte
02510: D8EB F0 BA BEQ    DECEND if zero.
02520: D8ED 20 F2 D8 JSR    DECADD
02530: D8F0 D0 B5 BNE    DECEND always & RTS
02540:
02550: D8F2 48      DECADD PHA
02560: D8F3 4A      LSRA      get high byte
02570: D8F4 4A      LSRA
02580: D8F5 4A      LSRA
02590: D8F6 4A      LSRA
02600: D8F7 20 FD D8 JSR    DECAD
02610: D8FA 68      PLA
02620: D8FB 29 0F ANDIM $0F get low nibble
02630:
02640: D8FD AA      DECAD TAX      nibble to X
02650: D8FE F0 D3 BEQ    SUBRTN if zero.
02660:
02670: D900 18      DECC CLC      processor's decimal flag is set!
02680: D901 A5 ED LDAZ  ONE
02690: D903 79 18 D9 ADCAY POWERS
02700: D906 85 ED STAZ  ONE
02710: D908 A5 EE LDAZ  TWO
02720: D90A 79 19 D9 ADCAY POWERS +01
02730: D90D 85 EE STAZ  TWO
02740: D90F 90 02 BCC  DECX
02750: D911 E6 EF INCZ  TRE      it does not matter the Decimal flag, because
02760:                      maximum is 07...
02770: D913 CA      DECX DEX
02780: D914 D0 EA BNE  DECC
02790: D916 F0 BB BEQ  SUBRTN always & RTS
02800:
*****
02810:
02820:
02830:
02840:
02850: D918 96      POWERS = $96 4096, 256, 16, 1
02860: D919 40      = $40
02870: D91A 56      = $56
02880: D91B 02      = $02
02890: D91C 16      = $16
02900: D91D 00      = $00
02910: D91E 01      = $01
02920: D91F 00      = $00
02930:
02940:
02950:
02960: D920 98      BELLY TYA      save Y
02970: D921 48      PHA
02980: D922 20 7A F1 JSR    BELL ring the bell
02990: D925 68      PLA
03000: D926 A8      TAY      restore Y
03010: D927 38      SEC      set carry for later branches (-always)
03020: D928 60      RTS
03030:
03040:
03050:
03060: D929 A9 00      CLEAR LDAIM $00 clear decimal buffer
03070: D92B 85 ED STAZ  ONE
03080: D92D 85 EE STAZ  TWO
03090: D92F 85 EF STAZ  TRE
03100: D931 60      RTS
03110:
03120:
03130:
03140: ***** test to verify NUMBER routines *****
03150:
03160: D932 20 09 D8 decHEX JSR    RECDEC receive decimal
03170: D935 20 84 D8 JSR    DECHEX convert to hexadecimal

```

NUMBER THE JUNIOR PROMOTING CY (W&J) 22:00:17 - TUE 17/JUN/86 PAGE 05

```

03180: D938 A5 F9 LDAZ INH
03190: D93A 20 75 F5 JSR PRBYT print the 4-nibble number
03200: D93D A5 F8 LDAZ INL
03210: D93F 20 75 F5 JSR PRBYT
03220: D942 20 FE F4 JSR CRLF carriage return
03230: D945 50 EB BVC decHEX loop: new conversion ...
03240:
03250: D947 20 00 D8 hexDEC JSR RECHEX receive hexadecimal
03260: D94A 20 D6 D8 JSR HEXDEC convert to decimal
03270: D94D A5 EF LDAZ TRE
03280: D94F 20 7E F5 JSR PRNIBL print the 5-digit number
03290: D952 A5 EE LDAZ TWO
03300: D954 20 75 F5 JSR PRBYT
03310: D957 A5 ED LDAZ ONE
03320: D959 20 75 F5 JSR PRBYT
03330: D95C 20 FE F4 JSR CRLF carriage return
03340: D95F 50 E6 BVC hexDEC loop: receive new number...
03350:
03360:
03370:
03380:
03390:
03400:
03410:
03420:
03430:
03440:
03450:
03460:
03470:
03480:
03490:
03500:
03510:
03520:
03530:
03540:
03550:
03560:
03570:
03580:
03590:
03600:
03610:
03620:
03630:
03640:
03650:
03660:
03670:
03680:
03690:
03700:
03710:
03720:
03730:
03740:
03750:
03760:
03770:
03780:
03790:
03800:
03810:
03820:
03830:
03840:
03850:
03860:
03870:
03880:
03890:
03900:
03910:
03920:
03930:
03940:
03950:
03960:
03970:
03980:
03990:
04000:
04010:
04020:
04030:
04040:
04050:
04060:
04070:
04080:
04090:

```

```

*****
*                                     *
*               REAL TIME CLOCK      *
*                                     *
*****

```

Routines to set and to display time.
Based on the 146818 Real Time Clock IC.
elektor electronics - April 1985.

Format chosen for output:
09:20:59 - SAT 22/FEV/86
(hour:minutes:seconds - day of the week
day of month/month/year)

```

03670: D961 20 A1 29 RTCLK JSR PRINTV
03680: D964 0D = $0D
03690: D965 0A = $0A
03700: D966 4F = $0
03710: D967 50 = 'P'
03720: D968 54 = 'T'
03730: D969 49 = 'I'
03740: D96A 4F = 'D'
03750: D96B 4E = 'N'
03760: D96C 53 = 'S'
03770: D96D 3A = ' '
03780: D96E 0D = $0D
03790: D96F 0A = $0A
03800: D970 53 = 'S'
03810: D971 29 = ')'
03820: D972 45 = 'E'
03830: D973 54 = 'T'
03840: D974 20 = ' '
03850: D975 43 = 'C'
03860: D976 4C = 'L'
03870: D977 4F = 'O'
03880: D978 43 = 'C'
03890: D979 4B = 'K'
03900: D97A 0D = $0D
03910: D97B 0A = $0A
03920: D97C 44 = 'D'
03930: D97D 29 = ')'
03940: D97E 49 = 'I'
03950: D97F 53 = 'S'
03960: D980 50 = 'P'
03970: D981 4C = 'L'
03980: D982 41 = 'A'
03990: D983 59 = 'Y'
04000: D984 20 = ' '
04010: D985 54 = 'T'
04020: D986 49 = 'I'
04030: D987 4D = 'M'
04040: D988 45 = 'E'
04050: D989 0D = $0D
04060: D98A 0A = $0A
04070: D98B BE = ' '
04080: D98C 03 = $03
04090:

```

		Hex		Decimal											
		b6	0	0	0	0	0	0	0	0	0	0	0	0	
		b7	0	0	0	0	0	0	1	1	1	1	1		
		b8	0	0	0	1	1	0	0	0	1	1	1		
		b5	0	1	0	1	0	1	0	1	0	1	1		
b4	b3	b2	b1			0	1	2	3	4	5	6	7		
0	0	0	0	0	NUL	0	0	0	0	@	P	50	86	P	
0	0	0	1	1	SOH	1	0	1	1	A	45	Q	81	a	
0	0	0	1	0	STX	2	2	12	22	B	66	R	82	b	
0	0	1	1	1	ETX	3	3	23	33	C	67	S	83	c	
0	1	0	0	0	EOT	4	4	14	24	D	68	T	84	d	
0	1	0	1	1	ENQ	5	5	15	25	E	69	U	85	e	
0	1	1	0	0	ACK	6	6	16	26	F	70	V	86	f	
0	1	1	1	1	BEL	7	7	17	27	G	71	W	87	g	
1	0	0	0	0	BS	8	8	18	28	H	72	X	88	h	
1	0	0	1	0	HT	9	9	19	29	I	73	Y	89	i	
1	0	1	0	0	LF	A	A	10	2A	J	74	Z	90	j	
1	0	1	1	0	VT	B	B	11	2B	K	75	[	91	k	
1	1	0	0	0	FF	C	C	12	2C	L	76	^	92	l	
1	1	0	1	0	CR	D	D	13	2D	M	77	_	93	m	
1	1	1	0	0	E	14	14	14	2E	N	78	^	94	n	
1	1	1	1	0	F	15	15	15	2F	O	79	^	95	o	

NUMBER THE JUNIOR PROMOTING CY (W&J) 22:01:26 - TUE 17/JUN/86 PAGE 06

```

04100: D98D 20 1B FE      JSR  RECCHA
04110: D990 C9 44      CMPIM 'D
04120: D992 D0 06      BNE  CLOCK
04130: D994 20 FE F4      JSR  CRLF
04140: D997 4C 0C DA      JMP  PRDAT display data
04150:
04160:
04170: D99A A2 00      CLOCK LDXIM $00 *** INPUT TIME AND DATE ***
04180: D99C 8E 54 EF      STX  DUPLEX =====
04190: D99F A0 00      LDYIM $00
04200:
04210: D9A1 B9 E0 DA      ASKFOR LDAAY TEXT
04220: D9A4 30 06      BMI  WAIT
04230: D9A6 20 5D F3      JSR  NPRCHA print text (prompt) for input
04240: D9A9 C8          INY
04250: D9AA D0 F5      BNE  ASKFOR always
04260:
04270: D9AC 29 7F      WAIT  ANDIM $7F
04280: D9AE 20 5D F3      JSR  NPRCHA print last character
04290: D9B1 C8          INY
04300: D9B2 98          TYA
04310: D9B3 48          PHA save pointer
04320:
04330: D9B4 A0 01      RECEIV LDYIM $01
04340: D9B6 20 0B D8      JSR  RECDCE +02 input a number (decimal)
04350: D9B9 A5 ED      LDAZ ONE get it
04360: D9BB DD 88 DA      CMPAX MAX is it correct?
04370: D9BE F0 07      BEQ  SAVE
04380: D9C0 90 05      BCC  SAVE
04390: D9C2 20 7A F1      JSR  BELL error
04400: D9C5 B0 ED      BCS  RECEIV always; receive again
04410:
04420: D9C7 9D 96 DA      SAVE  STAAZ DATA save inputed data for clock
04430: D9CA 20 FE F4      JSR  CRLF
04440: D9CD 68          PLA
04450: D9CE A8          TAY restore index
04460: D9CF E8          INX
04470: D9D0 C0 45      CPYIM $45
04480: D9D2 90 CD      BCC  ASKFOR
04490:
04500:
04510: D9D4 A9 06      SETCLK LDAIM $06 0000.0110 clock mode and SQW settings
04520: D9D6 A2 0A      LDXIM $0A Reg A
04530: D9D8 BE C8 EF      STX  CLKADR prevent time/calendar updates
04540: D9DB 8D C9 EF      STA  CLKDAT
04550:
04560: D9DE A9 B3      LDAIM $B3 1000.0011 going to initialize time! SQW is off.
04570: D9E0 E8          INX 0B Reg : all interrupts are disabled
04580: D9E1 BE C8 EF      STX  CLKADR BCD digits; 24 hour/day;
04590: D9E4 8D C9 EF      STA  CLKDAT daylight savings enabled
04600:
04610: D9E7 A0 00      LDYIM $00
04620: D9E9 B9 96 DA      TAKDAT LDAAY DATA get data for clock
04630: D9EC BE 8F DA      LDXAY ADDRESS get destination register
04640: D9EF BE C8 EF      STX  CLKADR address register
04650: D9F2 8D C9 EF      STA  CLKDAT write it: set time and date
04660: D9F5 C8          INY
04670: D9F6 C0 07      CPYIM $07 all 7 chosen registers?
04680: D9F8 90 EF      BCC  TAKDAT if no
04690:
04700: D9FA A2 0B      LDXIM $0B Reg B
04710: D9FC A9 03      LDAIM $03
04720: D9FE BE C8 EF      STX  CLKADR
04730: DA01 8D C9 EF      STA  CLKDAT time is already initialized.
04740:
04750: DA04 A2 0D      LDXIM $0D Reg D
04760: DA06 BE C8 EF      STX  CLKADR
04770: DA09 AD C9 EF      LDA  CLKDAT read reg D to validate RAM and time
04780:
04790:
04800:
04810: DA0C A0 FF      PRDAT LDYIM $FF *** PRINT TIME AND DATE ***
04820:
04830: DA0E 20 54 DA      JSR  IV get hour 18:33:30 - SAT 22/FEB/86
04840: DA11 20 49 DA      JSR  I print hour; get minutes
04850: DA14 20 49 DA      JSR  I print minutes; get seconds
04860: DA17 20 75 F5      JSR  PRBYT print seconds
04870: DA1A 20 09 F5      JSR  PRSP
04880: DA1D A9 2D      LDAIM '-'
04890: DA1F 20 5D F3      JSR  NPRCHA
04900: DA22 A9 20      LDAIM

```

NUMBER THE JUNIOR PROMOTING CY (W&J) 22:02:44 - TUE 17/JUN/86 PAGE 07

```

04910: DA24 20 51 DA JSR III get number of day of week
04920: DA27 20 71 DA JSR VI print name of day
04930: DA2A A9 20 LDAIM
04940: DA2C 20 51 DA JSR III get number of day of month
04950: DA2F 20 75 F5 JSR PRBYT print day of month
04960: DA32 20 4F DA JSR IIA get number of month
04970: DA35 20 69 DA JSR V print name of month
04980: DA38 20 4F DA JSR IIA get year
04990: DA3B 20 75 F5 JSR PRBYT print year
05000: DA3E 20 FE F4 JSR CRLF
05010: DA41 A9 0B LDAIM $0B
05020: DA43 20 5D F3 JSR NPRCHA
05030:
05040: DA46 4C 0C DA JMP PRTDAT
05050:
05060:
05070:
05080:
*** Auxiliar subroutines ***
05090: DA49 20 75 F5 I JSR PRBYT
05100: DA4C A9 3A II LDAIM
05110: DA4E 2C = $2C BIT
05120: DA4F A9 2F IIA LDAIM
05130: DA51 20 5D F3 III JSR NPRCHA
05140: DA54 C8 IV INY
05150: DA55 A2 0A LDXIM $0A
05160: DA57 8E C8 EF UIP? STX CLKADR
05170: DA5A AD C9 EF LDA CLKDAT Update in progress?
05180: DA5D 30 F8 BMI UIP? yes; wait
05190:
05200: DA5F 8E 9D DA LDXAY REG
05210: DA62 8E C8 EF STX CLKADR
05220: DA65 AD C9 EF LDA CLKDAT read clock registers
05230: DA68 60 RTS
05240:
05250:
05260:
*****
05270: DA69 C9 10 V CMPIM $10 (OCT/NOV/DEC Binary Coded Decimal)
05280: DA6B 90 02 BCC VA
05290: DA6D E9 07 SBCIM $07 convert to hexadecimal + carry
05300:
05310: DA6F 69 07 VA ADCIM $07 add displacement in table + carry
05320:
05330: DA71 85 FC VI STAZ TEMP multiply by three (3 letters by name)
05340: DA73 0A ASLA (x2)
05350: DA74 18 CLC
05360: DA75 65 FC ADCZ TEMP (+1)
05370: DA77 AA TAX (transform pointer)
05380: DA78 BD A4 DA DAYMON LDAAX DAYS
05390: DA7B 30 06 BMI DAY
05400: DA7D 20 5D F3 JSR NPRCHA
05410: DA80 E8 INX
05420: DA81 50 F5 BVC DAYMON always
05430: DA83 29 7F DAY ANDIM $7F
05440: DA85 4C 5D F3 JMP NPRCHA & RTS
05450:
05460:
05470:
05480:
*** Real time clock - auxiliar tables ***
05490: DA88 99 MAX = $99 year - maximum value for input
05500: DA89 12 = $12 month
05510: DA8A 31 = $31 day of month
05520: DA8B 07 = $07 day of week
05530: DA8C 23 = $23 hour
05540: DA8D 59 = $59 minutes
05550: DA8E 59 = $59 seconds
05560:
05570: DA8F 09 ADDRES = $09 address of the clock-registers for SETCLOCK
05580: DA90 08 = $08
05590: DA91 07 = $07
05600: DA92 06 = $06
05610: DA93 04 = $04
05620: DA94 02 = $02
05630: DA95 00 = $00
05640:
05650: DA96 00 DATA = $00 temporaries for inputed clock data
05660: DA97 00 = $00
05670: DA98 00 = $00
05680: DA99 00 = $00
05690: DA9A 00 = $00
05700: DA9B 00 = $00
05710: DA9C 00 = $00
05720:

```

NUMBER THE JUNIOR PROMOTING CY (W&J) 22:03:58 - TUE 17/JUN/86 PAGE 08

05730:	DA9D 04	REG	=	\$04	registers' order for formatted read-clock
05740:	DA9E 02		=	\$02	
05750:	DA9F 00		=	\$00	
05760:	DAA0 06		=	\$06	
05770:	DAA1 07		=	\$07	
05780:	DAA2 08		=	\$08	
05790:	DAA3 09		=	\$09	
05800:					
05810:					
05820:	DAA4 2D	DAYS	=	'-	0 names of days and months
05830:	DAA5 2D		=	'-	
05840:	DAA6 2D		=		Please translate to your own language
05850:					
05860:	DAA7 44		=	'D	1 Sunday
05870:	DAA8 4F		=	'D	
05880:	DAA9 CD		=	'M	
05890:					
05900:	DAAA 53		=	'S	2 Monday
05910:	DAA8 45		=	'E	
05920:	DAAC C7		=	'G	
05930:					
05940:	DAAD 54		=	'T	3 Tuesday
05950:	DAAE 45		=	'E	
05960:	DAAF D2		=	'R	
05970:					
05980:	DAB0 51		=	'Q	4 Wednesday
05990:	DAB1 55		=	'U	
06000:	DAB2 C1		=	'A	
06010:					
06020:	DAB3 51		=	'Q	5 Thursday
06030:	DAB4 55		=	'U	
06040:	DAB5 C9		=	'I	
06050:					
06060:	DAB6 53		=	'S	6 Friday
06070:	DAB7 45		=	'E	
06080:	DAB8 D8		=	'X	
06090:					
06100:	DAB9 53		=	'S	7 Saturday
06110:	DABA 41		=	'A	
06120:	DABB C2		=	'B	
06130:					
06140:					
06150:	DABC 4A		=	'J	1 January
06160:	DABD 41		=	'A	
06170:	DABE CE		=	'N	
06180:					
06190:	DABF 46		=	'F	2 February
06200:	DAC0 45		=	'E	
06210:	DAC1 D6		=	'V	
06220:					
06230:	DAC2 4D		=	'M	3 March
06240:	DAC3 41		=	'A	
06250:	DAC4 D2		=	'R	
06260:					
06270:	DAC5 41		=	'A	4 April
06280:	DAC6 42		=	'B	
06290:	DAC7 D2		=	'R	
06300:					
06310:	DAC8 4D		=	'M	5 May
06320:	DAC9 41		=	'A	
06330:	DACA C9		=	'I	
06340:					
06350:	DACB 4A		=	'J	6 June
06360:	DACC 55		=	'U	
06370:	DACD CE		=	'N	
06380:					
06390:	DACE 4A		=	'J	7 July
06400:	DACF 55		=	'U	
06410:	DAD0 CC		=	'L	
06420:					
06430:	DAD1 41		=	'A	8 August
06440:	DAD2 47		=	'G	
06450:	DAD3 CF		=	'O	
06460:					
06470:	DAD4 53		=	'S	9 September
06480:	DAD5 45		=	'E	
06490:	DAD6 D4		=	'T	
06500:					
06510:	DAD7 4F		=	'O	10 October
06520:	DAD8 55		=	'U	
06530:	DAD9 D4		=	'T	
06540:					

NUMBER THE JUNIOR PROMOTING CY (W&J) 22:05:04 - TUE 17/JUN/86 PAGE 09

06550:	DADA	4E	=	'N	11 November
06560:	DADB	4F	=	'D	
06570:	DADC	D6	=	'V	
06580:					
06590:	DADD	44	=	'D	12 December
06600:	DADE	45	=	'E	
06610:	DADF	DA	=	'Z	
06620:					
06630:					
06640:	DAE0	0D	TEXT =	\$0D	text-prompts for data input
06650:	DAE1	0A	=	\$0A	
06660:	DAE2	41	=	'A	Year:
06670:	DAE3	4E	=	'N	
06680:	DAE4	4F	=	'O	
06690:	DAE5	3A	=	:	
06700:	DAE6	A0	=	:	
06710:	DAE7	4D	=	'M	Month N.
06720:	DAE8	45	=	'E	
06730:	DAE9	53	=	'S	
06740:	DAEA	20	=	:	
06750:	DAEB	4E	=	'N	
06760:	DAEC	2E	=	:	
06770:	DAED	A0	=	:	
06780:					
06790:	DAEE	44	=	'D	Day of Month N.
06800:	DAEF	49	=	'I	
06810:	DAF0	41	=	'A	
06820:	DAF1	20	=	:	
06830:	DAF2	44	=	'D	
06840:	DAF3	4F	=	'O	
06850:	DAF4	20	=	:	
06860:	DAF5	4D	=	'M	
06870:	DAF6	45	=	'E	
06880:	DAF7	53	=	'S	
06890:	DAF8	20	=	:	
06900:	DAF9	4E	=	'N	
06910:	DAFA	2E	=	:	
06920:	DAFB	A0	=	:	
06930:					
06940:	DAFC	44	=	'D	Day of Week N.
06950:	DAFD	49	=	'I	
06960:	DAFE	41	=	'A	
06970:	DAFF	20	=	:	
06980:	DB00	44	=	'D	
06990:	DB01	41	=	'A	
07000:	DB02	20	=	:	
07010:	DB03	53	=	'S	
07020:	DB04	45	=	'E	
07030:	DB05	4D	=	'M	
07040:	DB06	41	=	:	
07050:	DB07	4E	=	'N	
07060:	DB08	41	=	:	
07070:	DB09	20	=	:	
07080:	DB0A	4E	=	'N	
07090:	DB0B	2E	=	:	
07100:	DB0C	A0	=	:	
07110:					
07120:	DB0D	48	=	'H	Hours:
07130:	DB0E	4F	=	'O	
07140:	DB0F	52	=	'R	
07150:	DB10	41	=	'A	
07160:	DB11	53	=	'S	
07170:	DB12	3A	=	:	
07180:	DB13	A0	=	:	
07190:					
07200:	DB14	4D	=	'M	Minutes:
07210:	DB15	49	=	'I	
07220:	DB16	4E	=	'N	
07230:	DB17	55	=	:	
07240:	DB18	54	=	:	
07250:	DB19	4F	=	'O	
07260:	DB1A	53	=	'S	
07270:	DB1B	3A	=	:	
07280:	DB1C	A0	=	:	
07290:					

DE 6502 KENNER

```

07300: DB1D 53      = 'S      Seconds:
07310: DB1E 45      = 'E
07320: DB1F 47      = 'G
07330: DB20 55      = 'U
07340: DB21 4E      = 'N
07350: DB22 44      = 'D
07360: DB23 4F      = 'O
07370: DB24 53      = 'S
07380: DB25 3A      = '
07390: DB26 A0      = '
07400:

```

>Q

```

SOURCE SPACE: START 2A00 END C000      CURR-MAX 5D61 FREE 025247
SYMBOL TABLE: START C000 END CE00 CURR-MAX C2A0 FREE 02912
                STILL PLACE FOR 0364 ENTRIES
XREF TABLE:  START CE00 END D800 CURR-MAX D319 FREE 01255
                STILL PLACE FOR 0251 ENTRIES

```

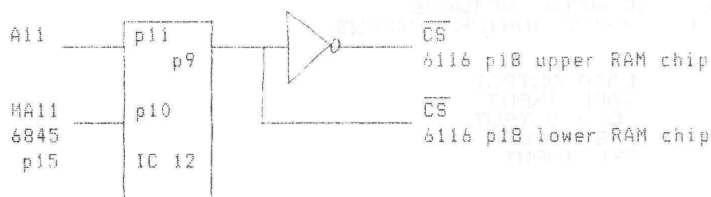
>00

Mr. J.C.Rix
3, Sutton Drive,
SEAFORD,
SUSSEX, BN25 3EU
ENGLAND.

Elektor VDU card modifications

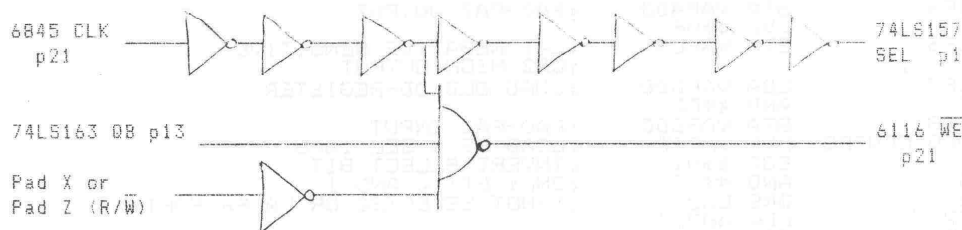
4k RAM

This is a very straightforward alteration involving the addition of an extra 6116 RAM chip mounted 'piggyback' on the existing 6116. On both 6116's pin 18 is splade out separately and connected as diagram below. One of the multiplexers is not used in the original design so I have used this for the additional address line (A11).



Non flicker

This alteration will remove the annoying screen flicker caused by processor access. I used two type 74LS04 chips for the inverters, and re-used N32 (3 input NAND gate).



If any of your members have a better solution to the screen flicker problem I would be pleased to hear about it.

New CENTRONIC routine for the 2 MHz JUNIOR/OCTOPUS 65

Author :Coen Boltjes 015-136812 VIDIBUS 400029830
Translator :Elya van der Veer

After reading the Elektuur Computer Special Issue 2 I stepped up the frequency of the clock of my DOS-Junior to 2 MHz. After testing the computer appeared to function well, without the faster I/O chips models. (I only use a 65C02) Also, it appeared that the waiting loops for the drive did not have to be adjusted. (A TEAC FD55-BV is used) I did have to change the CENTRONIC routine for the printer. The VIA is programmed in such a way that a handshake-pulse is provided automatically when new data are presented. However, the size of the pulse is dependent on the frequency of the clock of the computer. If the frequency is doubled the size of the pulse will be 500 nS. This appeared to be sufficient for my EPSON MX-80 matrix printer, but my BROTHER CE-50 needed more. Therefore, a new programme has been devised generating a larger pulse. After the initiating steps the character is put on the data lines. Then, at CENTRP a waiting loop is executed in order to stabilize the signals. This enables one to use longer connection cables. The delay caused by the loop can be ignored in comparison with the time needed to wait for the printer which usually does not exceed 100 Char/sec. Then, the strobe pulse is generated by writing into the VAPCR \$0C and \$0E successively. The pulse is now 3µS, which can be enlarged by adding some NOP-instructions if necessary. The length of the programme is smaller than the Paperware 4 routine. Thence, it can replace the old one. Furthermore, it is possible to use the new routine somewhere else by adjusting the outputvector #4 (\$2317). In that case the vectoraddress must be one less than the starting address of the routine. The routine can also be applied to the OCTOPUS 65. Then, one should take into account that the communication is different here, so that addresses and masks must be adjusted.

```

10      ; NEW CENTRO-ROUTINE WITH SOFTWARE
20      STOBE OUTPUT FOR 2-MHZ JUNIOR/OCTOPUS
30
40      VIA DEFINITIONS:
50      PA0-PA7      DATA OUTPUT
60      CA1          /ACK INPUT
70      CA2          /STB OUTPUT
80      PB1          /PE INPUT
90      PBO          SEL INPUT
100
110     VIA ADDRESSES
120
130     F800=        VAPBD = $F800
140     F801=        VAPAD = VAPBD+$1
150     F802=        VAPBDD = VAPBD+$2
160     F803=        VAPADD = VAPBD+$3
170     F80C=        VAPCR = VAPBD+$C
180     F80D=        VAIFR = VAPBD+$D
190
200     2363=        AHOLD = $2363
210     F3E2        * = $F3E2
220
230     F3E2 A9FF    INIT      LDA #$FF
240     F3E4 BD03F8  STA VAPADD      ; PA0-PA7 OUTPUT
250     F3E7 A90E    LDA #$0E
260     F3E9 BD0CF8  STA VAPCR       ; CA1 NEGATIVE SENSITIVE
270                                     ; CA2 HIGH OUTPUT
280     F3EC AD02F8  LDA VAPBDD      ; LOAD OLD DD-REGISTER
290     F3EF 29FC    AND #$FC
300     F3F1 BD02F8  STA VAPBDD
310     F3F4 AD00F8  CENTRO      LDA VAPBD
320     F3F7 4901    EOR #$01        ; PA0-PA1 INPUT
330     F3F9 2903    AND #$03        ; LOAD PE & SEL INPUT
340     F3FB D025    BNE END         ; INVERT SELECT BIT
350     F3FD AD6323  LDA AHOLD       ; ONLY BIT 0 AND 1
360     F400 BD01F8  STA VAPAD       ; => NOT SELECTED OR PAPER EMPTY
370     F403 A200    LDX #$00        ; CHAR. ON DATA LINES
380     F405 E8      CENTRP      INX      ; INIT. DATA SET-UP TIME
390     F406 D0FD    BNE CENTRP     ; => SETUP TIME NOT ENDED
400     F408 A90C    LDA #$0C

```

```

410 F40A 8DOCFB STA VAPCR ; CLEAR STROBE OUTPUT
420 F40D A90E LDA #0E
430 F40F 8DOCFB STA VAPCR ; AND SET IT
440 F412 AD00FB CENTRQ LDA VAFBD ; LOAD PE AND SEL
450 F415 4901 EOR #01
460 F417 2903 AND #03
470 F419 D007 BNE END ; =>NOT SELECTED OR PAPER EMPTY
480 F41B AD0DFB LDA VAIFR
490 F41E 2902 AND #02
500 F420 F0F0 BEQ CENTRQ ; =>NO ACKNOWLEDGE
510 F422 AD0DFB END LDA VAIFR ; LOAD FLAGS
520 F425 0903 ORA #03 ; SET BITS 0 AND 1
530 F427 8D0DFB STA VAIFR ; RESET CA1 AND CA2
540 F42A 60 RTS
550 .END

```

PAPERWARE SERVICE (ENGLISH VERSION) LIST OF PRICES

All prices including forwarding charges, etc.
All prices only for European (C.E.P.T.)-countries.

DISASSEMBLER FOR 65(C)02 (Rockwell version)
Nico de Vries, The Netherlands.
Derived from the program originally written for the Apple II by Wozniak and Baum.
This routine in machinecode will run directly on Commodore PET, but easy to adapt to other machines.
Send cheque of Hfl. 25,00 to W.L.v.Pelt (eurocheque 15,50)

DISASSEMBLER FOR 65(C)02 (Synertek/GTE version)
Nico de Vries, The Netherlands.
Refer the announcement above.
Send cheque of Hfl. 25,00 to W.L.v.Pelt (eurocheque 15,50)

WRITE- AND READROUTINE FOR ELEKTOR'S JUNIOR-COMPUTER
Dick Blok, The Netherlands. With thanks to Sebo Woldringh for his advices. Machinecode.
Send cheque of Hfl. 15,00 to W.L.v.Pelt (eurocheque 5,50)

LARGE CHARACTERS WITH THE MICROLINE-80 PRINTER
Frans Bakx, The Netherlands.
A set of subroutines in 6502-machinecode to print large characters: 3 lines high and 4 normal characters width.
Send cheque of Hfl. 15,00 to W.L.v.Pelt (eurocheque 5,50)

DATABAS. A database-program written in Basic for Elektor's JUNIOR-computer with VDU-card and OHIO Scientific OS-650 V3.3 disk operating system.
Jan van Heuven, The Netherlands. Modified by Fernando Lopes, Portugal.
Send cheque of Hfl. 30,00 to W.L.v.Pelt (eurocheque 20,50)

VDU ROTATING NEWSPAPER.
Ivo van Rijssel, The Netherlands. Translated by Frank Bens
Machinecode-program for Elektor's JUNIOR-computer with VDU
Send cheque of Hfl. 15,00 to W.L.v.Pelt (eurocheque 5,50)

TYPE-AHEAD AND PRINTER BUFFER
Marcel Visser, The Netherlands.
Machinecode-program for APPLE II with DOS 3.3 or DIVERSI-DOS.
Send cheque of Hfl. 20,00 to W.L.v.Pelt (eurocheque 10,50)

CENTRONICS PRINTER INTERFACE DEVICE 4 OR 5 ON COMMODORE 64
Ruud Uphoff, The Netherlands. Machinecode.
Send cheque of Hfl. 15,00 to W.L.v.Pelt (eurocheque 5,50)

TOKENIZED Microsoft Basic Keywords and addresses KIM-1 and Elektor's JUNIOR KB-9 Basic.
W.L. van Pelt, The Netherlands.
Send cheque of Hfl. 14,50 to W.L.v.Pelt (eurocheque 5,00)

TOKENIZED Microsoft Basic Keywords and addresses SYM-1 Basic.

W.L. van Pelt, The Netherlands.
Send cheque of Hfl. 14,50 to W.L.v.Pelt (eurocheque 5,00)

TOKENIZED Microsoft Basic Keywords and addresses AIM-65 Basic.
W.L. van Pelt, The Netherlands.
Send cheque of Hfl. 14,50 to W.L.v.Pelt (eurocheque 5,00)

TOKENIZED Microsoft Basic Keywords and addresses APPLEsoft Basic.
W.L. van Pelt, The Netherlands.
Send cheque of Hfl. 14,50 to W.L.v.Pelt (eurocheque 5,00)

TOKENIZED Microsoft Basic Keywords and addresses CBM 40XX and CBM 80XX. Commodore Basic 4.0.
Nico de Vries, The Netherlands.
Send cheque of Hfl. 14,50 to W.L.v.Pelt (eurocheque 5,00)

TOKENIZED Microsoft Basic Keywords and addresses C - 16 Commodore Basic 3.5.
Nico de Vries, The Netherlands.
Send cheque of Hfl. 14,50 to W.L.v.Pelt (eurocheque 5,00)

TOKENIZED Microsoft Basic Keywords and addresses CBM 30XX Commodore asic 2.0.
Nico de Vries, The Netherlands.
Send cheque of Hfl. 14,50 to W.L.v.Pelt (eurocheque 5,00)

TOKENIZED Microsoft Basic Keywords and addresses VIC-20 Commodore Basic V2.
Nico de Vries, The Netherlands.
Send cheque of Hfl. 14,50 to W.L.v.Pelt (eurocheque 5,00)

PATCHES ON KIM-1 MICROSOFT BASIC FOR ELEKTOR'S JUNIOR-comp
Koen van Nieuwenhove, Belgium/W.L. van Pelt, The Netherlands.
With PMODE, TRACE, STEP RESET, PRINTER PG., VIDEO, AUTO, RENUMBER, DATASAVE, EDIT LINE, APPEND.
Send cheque of Hfl. 24,50 to W.L.v.Pelt (eurocheque 15,00)

MICRO-WARE Assembler/Editor with bank-switching
Complete source-listing (Disassembler not included)
Author: Fernando Lopes, Portugal.
System: Elektor's JUNIOR-computer with VDU and OS65-D DOS
With new commands: PUT FILE, PUT TT, LOAD FILE, LOAD TT, toggle centronics flag, return to kernal, and modified error routine, and new subroutines DECBPF, INCBPF, SETPM-SETSC-SETPF, PRDAT to read data and time from the Real Time Clock and to print it in the header lines, SETBK0-SWAPDN-SAVBNK-RSTBNK to manage the bankswitching, LINE NUMBERS.

Refer: DE 6502 KENNER, May 1986, p.44-47
Hardware article "Bank Switching for the JUNIOR-computer" by Fernando Lopes, Portugal.
Send cheque of Hfl. 45,00 to W.L.v.Pelt (eurocheque 35,50)

ENQUETE onder de leden van de KIM Gebruikersclub Nederland.

Het bestuur van de KIM Gebruikersclub Nederland heeft besloten een enquête onder de leden te houden, teneinde in staat te zijn een beleid te voeren dat gebaseerd is op de wensen en ideeën van de leden.

U vindt de enquête op de middenpagina's van dit nummer. De enquête kan in twee delen gesplitst worden. Het eerste deel is bedoeld om het bestuur een idee te geven van het bestand aan systemen binnen de club en wat de drijfveren zijn van diegenen die zich met deze computers bezig houden. Dit betreft de vragen 1 tot en met 5.

Het tweede deel (vraag 6) bevat vragen die het bestuur moeten helpen om een goed beleid voor de toekomst uit te stippelen.

Deelname aan de enquête is geheel vrijwillig, terwijl u ook niet verplicht bent alle vragen te beantwoorden.

Teneinde de deelnemers aan deze enquête enigszins te belonen voor hun moeite heeft het bestuur een aantal presentjes beschikbaar gesteld die verloot zullen worden onder diegenen die niet anoniem aan de enquête deelnemen. Bestuursleden zijn van deze verloting uitgesloten. Beschikbaar zijn de volgende presentjes:

- 10 EPROMs 2764
- 10 paar diskettes, (5.25 inch, leeg)
- 5 SRAMs 4364 (8k x 8)

De verloting zal plaatsvinden op de bijeenkomst te Rijswijk (Zh.) op 15 november 1986. De Voorzitter zal dan uit de inzendingen de winnaars trekken.

Wilt u de ingevulde enquête voor 1 oktober 1986 gefrankeerd sturen naar:

N. de Vries
Mari Andriessenrade 49
2907 MA Capelle aan den IJssel
Nederland

Het bestuur dankt u alvast voor de genomen moeite.

ENQUIRY among the members of the KIM Gebruikersclub Nederland.

The executive committee of the KIM gebruikersclub Nederland has decided to hold an enquiry among the members of the club, in order to be able to develop a policy based on the ideas and wishes of the members themselves.

You will find the Dutch version of the enquiry in the middle of this issue. However, if you prefer an enquiry in English, please apply for one at the address below. You will then receive the same document, written in English.

This enquiry is comprised of two parts. The first part is meant to give the committee an idea of the number of systems our members own and use, and what purposes they are used for. The questions 1 through 5 are meant for this purpose.

The second part (question 6) contains questions to help develop a policy for the future.

Participation in this enquiry is completely voluntary. It is also not mandatory to answer all questions.

To encourage you to participate, the committee has available some little presents. These are to be raffled among those participants that have filled in their names and addresses on the questionnaire. Members of the committee are excluded from the raffle. The following presents are available:

- 10 EPROMs 2764
- 10 sets of two diskettes (5.25 inch, empty)
- 5 SRAMs 4364 (8k x 8)

The raffle will take place on the club meeting in Rijswijk on November 15th 1986. The chairman will then draw the winner from the enquiries received before the closing date.

Please return the completed questionnaire before October 1st 1986, postage paid, to:

N. de Vries
Mari Andriessenrade 49
2907 MA Capelle aan den IJssel
The Netherlands

The committee thanks you in advance for your trouble.

DE 6502 KENNER

1. Persoonsgegevens.

Indien u er de voorkeur aan geeft anoniem te blijven, kunt u verder gaan met de tweede vraag.
(U komt dan NIET in aanmerking voor de verloting).

A. Wat is uw naam?

.....

B. En uw adres?

.....
.....

C. Wat is uw beroep?

.....

D. Geslacht: ☐ Man ☐ Vrouw

E. En uw leeftijd?

- ☐ jonger dan 15 jaar
- ☐ tussen 15 en 25 jaar
- ☐ tussen 25 en 35 jaar
- ☐ tussen 35 en 45 jaar
- ☐ tussen 45 en 55 jaar
- ☐ tussen 55 en 65 jaar
- ☐ ouder dan 65 jaar

F. Spreekt u Engels of kunt u Engels lezen?

☐ Ja ☐ Redelijk ☐ Nee

2. Gegevens betreffende het computersysteem.

A. Welke computer(s) heeft u?

- | | |
|---|---|
| ☐ Junior | ☐ DOS65 |
| ☐ Apple][| ☐ BBC |
| ☐ Acorn Atom | ☐ Acorn Electron |
| ☐ KIM | ☐ Proton |
| ☐ SYM | ☐ AIM |
| ☐ PET | ☐ Commodore 64 |
| ☐ CBM 30XX, 40XX of 8032 | |
| ☐ OCTOPUS/SAMSON65/EC65 | |
| ☐ Geheel zelfgebouwd | |
| ☐ Andere, n.l., | |

.....

B. Hoe lang heeft u al een computer?

- ☐ korter dan een jaar
- ☐ 1 à 2 jaar
- ☐ 3 à 4 jaar
- ☐ langer, nl. jaar

C. Welk operating system gebruikt u?

- ☐ OHIO OS65D
- ☐ DOS65
- ☐ Koen van Nieuwehove's DOS
- ☐ Ik gebruik alleen de software in de (EP)ROMs van de computer
- ☐ Andere, n.l.,

.....

D. Welke zaken maken deel uit van uw systeem?

- ☐ Hex display en keyboard
- ☐ Monitor/ASCII toetsenbord
- ☐ Terminal
- ☐ Teletype
- ☐ Matrixprinter
- ☐ Daisywheelprinter
- ☐ Diskdrive(s), 5.25 inch
- ☐ Diskdrive(s), 8 inch
- ☐ Diskdrive(s), 3 of 3.5 inch
- ☐ Winchesterdisk
- ☐ Cassetterecorder
- ☐ Modem
- ☐ EPROM-programmer
- ☐

E. Waar gebruikt u uw systeem het meest voor?

- ☐ Administratie, bedrijf
- ☐ Administratie, privé
- ☐ Tekstverwerking
- ☐ Bestandsbeheer ('database')
- ☐ Ontwikkelen van software
- ☐ Ontwikkelen van hardware
- ☐ Besturingen
- ☐ Spelen van spelletjes
- ☐ Imponeren van de buurman

3. Software.

A. Op welk niveau bent u actief m.b.t. software?

- ☐ Ik interesseer me uitsluitend voor hardware
- ☐ Ik kopieer alleen software van anderen
- ☐ Ik verbeter programma's van anderen
- ☐ Ik schrijf zelf wel eens een programma
- ☐ Ik schrijf de meeste programma's zelf
- ☐ Ik schrijf alles zelf, operating systems, monitorprogramma's, enz.

B. In welke computertalen kunt u programmeren?

- | | |
|---------------------------------------|------------------------------------|
| ☐ Hex | ☐ Assembler |
| ☐ BASIC | ☐ FORTH |
| ☐ PASCAL | ☐ C |
| ☐ Andere, n.l. | |

C. In welke taal programmeert u voornamelijk?

- | | |
|---------------------------------------|------------------------------------|
| ☐ Hex | ☐ Assembler |
| ☐ BASIC | ☐ FORTH |
| ☐ PASCAL | ☐ C |
| ☐ Andere, n.l. | |

D. wat is uw favoriete computertaal, en waarom?

E. Heeft u een voorkeur voor een bepaalde assembler?

- ☐ Moser (ASM/TED, MAE)
- ☐ Micro-Ade
- ☐ AS uit DOS65
- ☐ Ik ken er maar een, n.l.

☐ Andere,

F. Kunt u behalve de 6502 ook nog andere processoren in machinetaal programmeren?

- | | |
|--------------------------------|------------------------------------|
| ☐ 6800 | ☐ Z80 |
| ☐ 1802 | ☐ 2650 |
| ☐ SC/MP | ☐ 6809 |
| ☐ 68000 | ☐ 8086/8088 |
| ☐ | |

G. Wat vindt u van uw kennis van machinetaal/assembler?

- ☐ Slecht
- ☐ Ik me ermee redden
- ☐ Voldoende
- ☐ Ik ben er goed in

4. Hardware.

A. Op welk niveau bent u actief m.b.t. hardware?

- ☐ Ik heb er geen verstand van, ik koop alles.
- ☐ Ik bouw wel eens een printje (bijv. van Elektuur)
- ☐ Ik heb mijn complete systeem zelf gebouwd.
- ☐ Ik ontwerp zelf hardware
- ☐ Ik heb mijn complete systeem zelf ontworpen.

B. Zou u assemblagewerk voor de club willen doen (bijvoorbeeld DOS65 FDC-kaarten bouwen)?

- ☐ Nee ☐ Ja

C. Kunt u zelf printjes maken?

- ☐ Nee
- ☐ Ja, maar ik vind het teveel werk.
- ☐ Ik plak zelf printjes.
- ☐ Ik ets zelf printjes.
- ☐ Ik plak en ets printjes.
- ☐ Ik werk met CAD/CAM.

D. Kunt u een datasheet van een IC lezen en interpreteren?

- ☐ Ik snap er geen jota van.
- ☐ Ik kijk alleen naar de pin-aansluitingen
- ☐ Ik begrijp het meeste wel.
- ☐ Ja.

5. De club.

A. Geef het clubblad eens een rapport-cijfer (1-10)?

B. Wat vindt u van deze onderdelen/aspecten van het clubblad?

	Goed	Gemiddeld	Slecht
Het redactionele praatje	☐	☐	☐
Uitnodiging voor de bijeenkomsten	☐	☐	☐
Alg. artikelen in verhalende stijl	☐	☐	☐
Assembler listings	☐	☐	☐
BASIC listings	☐	☐	☐
FORTH listings	☐	☐	☐
Hardware ontwerpen	☐	☐	☐
Aanprijzen van clubartikelen	☐	☐	☐
De opmaak	☐	☐	☐

C. Schrijft u wel eens voor de 6502 Kenner?

☐ Nooit ☐ Weinig
☐ Af en toe ☐ Regelmatig

D. In welke taal moet het clubblad zijn?

☐ Alleen maar in het Nederlands
☐ Nederlands en een beetje Engels
☐ Nederlands en Engels door elkaar
☐ Nederlands en een beetje Duits
☐ Nederlands en Duits door elkaar
☐ In het Engels
☐ In het Duits
☐ Allerlei talen door elkaar

E. Welke artikelen moeten er meer in het clubblad verschijnen?

F. Welke artikelen moeten er minder in het clubblad verschijnen?

G. Heeft u nog verdere opmerkingen over het clubblad?

6. Vragen om het toekomstig beleid te bepalen.

A. Wilt u actief in de club zijn, en zo ja, waarin?

- ☐ Nee, ik ben lid voor het blad.
- ☐ Nee, ik heb er te weinig tijd voor
- ☐ Ik zou artikelen willen schrijven
- ☐ Ik zou hardware willen ontwikkelen
- ☐ Ik zou software willen ontwikkelen
- ☐ Ik zou een bijeenkomst willen (helpen) organiseren.
- ☐ Ik zou in het bestuur willen zitten.
- ☐ Ik zou spreekbeurten/lezingen willen houden
- ☐ Ik zou vertaalwerk willen doen

B. De club heet thans KIM Gebruikersclub Nederland. Moet deze naam veranderen?

- ☐ Geen mening
- ☐ Nee
- ☐ Ja, want

C. Wat zou u een goede nieuwe naam voor de club vinden?

- (Tips: 1. Geen CPU-nummer in de naam
2. geen merknamen in de naam
3. belicht het hobby-aspect
4. een Engelse naam mag ook
5. de naam moet gemakkelijk in het geheugen blijven hangen)

- ☐ De 6502 Kenners
- ☐ ENCC (Eerste Nederlandse Computer Club)
- ☐ FDC (First Dutch Computerclub)
- ☐ Club65+
- ☐ Andere, bijv.:

D. Wij houden ons tot dusver alleen met de 6502 bezig. Moet(en) daar nog andere processor(s) bijkomen (bijv. 16-bit)?

- ☐ Ja
- ☐ Nee

E. Moet de club processor-onafhankelijk worden?

- ☐ Ja
- ☐ Nee

F. Wilt u uw antwoord op de vragen D. en E. misschien toelichten?

-
-
-
-

G. Ziet u een logische 'opvolger' voor de 6502?

- ☐ 65C02
- ☐ 6800
- ☐ 6809
- ☐ Andere, n.l.
- ☐ 65816/65802
- ☐ 68000
- ☐ Geen

H. Komt u wel eens op een club-bijeenkomst?

- ☐ Nooit
- ☐ Alleen bij mij in de buurt
- ☐ Af en toe
- ☐ Ik kom op alle bijeenkomsten

I. Wat vindt u van de bijeenkomsten?

- ☐ Slecht
- ☐ Matig
- ☐ Goed

J. Wat vindt u van de lezingen op de bijeenkomsten?

- ☐ Slecht
- ☐ Matig
- ☐ Goed

K. Heeft u nog suggesties voor de bijeenkomsten?

-
-
-
-

Dit was 't.
Bedankt voor uw tijd en moeite.

Het Bestuur.

```

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000
1001
1002
1003
1004
1005
1006
1007
1008
1009
1010
1011
1012
1013
1014
1015
1016
1017
1018
1019
1020
1021
1022
1023
1024
1025
1026
1027
1028
1029
1030
1031
1032
1033
1034
1035
1036
1037
1038
1039
1040

```

```

005D: 01 01 00
0060: 03 05 03
0063: 08
0064: 05 0F A4
0067: 24 00 15
006A: 1F B4
006C: 34 0C 06
006F: 16 22 A2
0072: 26 36
0074: A6 B6 00
0077: 2E 3E AE
007A: BE 66
007C: 76 E6 F6
007F: 00 06 16
0082: 22 A2
0084: 00 04 07
0087: 38 3C 3F
008A: 40 C0 80
008D: 10 30 20

>57 TSW DFB 1, 1, 0, 3, 5, 3, 8
>58 TZET DFB $05, $0F, $A4, $24, $00, $15, $1F, $B4
>59 DFB $34, $00, $06, $16, $22, $A2, $26, $36
>60 DFB $A6, $B6, $00, $2E, $3E, $AE, $BE, $66
>61 DFB $76, $E6, $F6, $00, $06, $16, $22, $A2
>62 RCTI DFB $00, $04, $07, $38, $3C, $3F
>63 RCTII DFB $40, $C0, $80, $10, $30, $20
>64
>65 WIS EQU $90
>66 OZET EQU $91
>67 TEMP2 EQU $92
>68 TEMP3 EQU TEMP2+1
>69 TEMP4 EQU TEMP2+2
>70 TEMP5 EQU TEMP2+3
>71 TEMP6 EQU TEMP2+4
>72 TEMP7 EQU TEMP2+5
>73 TEMP8 EQU TEMP2+6
>74 TEMP9 EQU TEMP2+7
>75 TEMP A EQU TEMP2+8
>76 TEMP B EQU TEMP2+9
>77 TEMP C EQU TEMP2+$0A
>78 TEMP D EQU TEMP2+$0B
>79 TEMP E EQU TEMP2+$0C
>80 TEMP F EQU TEMP2+$0D
>81 KAZ EQU $A0
>82 KWIS EQU $A1
>83 STKWRDKAZEQU $A2
>84 STKWRDKNAZEQU $A3
>85 STKNUMKAZEQU $A4
>86 STKNUMKNAZEQU $A5
>87 STELWRDLOBEQU $A6
>88 STELWRDHOB EQU $A7
>89 STUKWRDLOBEQU $A8
>90 STUKWRDHOB EQU $A9
>91 VELDTELLEREQU $AA
>92 KONKNAZ EQU $AB
>93 STUKIO EQU $AC
>94 POSITIO EQU $AD
>95 SLASTUK EQU $AE
>96 DAMKNAZ EQU $AF
>97 VANVELDHEXEQU $B0
>98 VERTREKHEXEQU $B1
>99 VERTREKREKEQU $B2
>100 EERSTEVELDEQU $B3
>101 BITSTUKIOEQU $B4
>102 BITKAZ EQU $B5
>103 BITKNAZ EQU $B6
>104 BITKAZR EQU $B7
>105 BITKNAZR EQU $B8
>106 BITKAZL EQU $B9
>107 BITKNAZL EQU $BA
>108 RES1 EQU $BB
>109 RES2 EQU $BC
>110 RES3 EQU $BD
>111 POSWRDLOBEQU $BE
>112 POSWRDHOB EQU $BF
>113
>114
>115 ORG $C0

00C0: 00 00 00
00C3: 27 46 66
00C6: 17 00 00
>116 PIONTAB DFB $00, $00, $00, $27, $46, $66, $17, $00, $00

```

```

)C9: 05 05 00
)CC: 10 19 12
)CF: 2D
)D0: 07 FF F7
)D3: F8 F9 01
)D6: 09 08
)D8: 91 90 89
)DB: 99 09 10
)DE: 11 01

)118 STUKTAB DFB $05,$05,$00,$10,$19,$12,$2D
)119 DIRHEX DFB $07,$FF,$F7,$F8,$F9,$01,$09,$08
)120 DIRDEC DFB $91,$90,$89,$99,$09,$10,$11,$01
)121 ORG $E1
)122 KEYBUF EQU $E1 ; TOETSBUFFER
)123
)124 ORG $F9
)125 DISP1 EQU $F9 ; DISPLAYBUFFER
)126 DISP2 EQU DISP1+1 ; IDEM
)127 DISP3 EQU DISP1+2 ; IDEM
)128
)129 ; MONITOR ADRESSEN
)130
)131 SCANDS EQU $1D8E ; TOON INHOUD DISPLAYBUFFER
)132 GETKEY EQU $1DF9 ; HAAL TOETS VAN HEX-PAD
)133
)134 ; PIA ADRESSEN
)135
)136 PAD EQU $1A80 ; POORT A DATA REGISTER
)137 PBD EQU $1A82 ; POORT B DATA REGISTER
)138
)139 ; PIA TIMER ADRES
)140
)141 TIMER EQU $1AF4 ; CLKIT, DISABLE TIMER IRQ
)142
)143 ; VECTOR ADRESSEN
)144
)145 NMIVECTLOBEQU $1A7A
)146 NMIVECTHOB EQU $1A7B
)147 IRQVECTLOBEQU $1A7E
)148 IRQVECTHOB EQU $1A7F
)149
)150 ; SCHAAKMONITOR ADRESSEN
)151
)152 TD/DS EQU $0590 ; TOETSDETEKTIE/DISPLAY
)153 MAINSMON EQU $05A5 ; HOOFDROUTINE SCHAAKMONITOR
)154 CODEMON EQU $05F3 ; CODE-ROUTINE
)155 VELDKODMON EQU $05FC ; VELDKONTROLE EN KODERING
)156 PLUSMON EQU $0629 ; STUK VERPLAATSEN
)157 DAMON EQU $064F ; STUK PLAATSEN/VERWIJDEREN
)158 ADMON EQU $0683 ; BORD VEGEN
)159 PCMON EQU $068D ; NIEUWE PARTIJ
)160
)161 ; POSVAL ADRESSEN
)162
)163 ADD EQU $2000 ; ADD
)164 TOTAL EQU $200E ; TOTAL
)165 MULTIPLY EQU $2022 ; MULTIPLY
)166 DIVIDE EQU $2077 ; DIVIDE
)167 SCAN EQU $20DA ; SCAN
)168 KODEER EQU $20FA ; KODEER
)169 VELDTEST EQU $2111 ; VELDKONTROLE
)170 NAARVELD EQU $212F ; NAARVELD
)171 AFSTAND EQU $2140 ; AFSTAND
)172 EERSTEVELD EQU $2174 ; EERSTE VELD
)173 STATUSI EQU $219C ; STATUS I
)174 STATUSII EQU $21E0 ; STATUS II
)175 PAARD EQU $2227 ; PAARD
)176 LOPER EQU $22A2 ; LOPER
)177 DAME EQU $2318 ; DAME
)178 TOREN EQU $238D ; TOREN
)179 PION EQU $246E ; PION
)180 KONINGI EQU $2582 ; KONING I
)181 KONINGII EQU $266C ; KONING II

```

```

)183
)184
)185
)186
)187
)188
)189
)190
)191
)192
)193
)194
2
)1
)2
2000: 4C 95 2C )3
2003: A5 9B )4
2005: 18 )5
2006: 65 A8 )6
2008: 85 A8 )7
200A: A5 9C )8
200C: 65 A9 )9
200E: 85 A9 )10
2010: 60 )11
2011: A5 A6 )12
2013: 85 9B )13
2015: A5 A7 )14
2017: 85 9C )15
2019: 20 00 20 )16
201C: A5 A8 )17
201E: 85 A6 )18
2020: A5 A9 )19
2022: 85 A7 )20
2024: 60 )21
2025: 84 97 )22
2027: 84 98 )23
2029: 84 98 )24
202B: 84 9C )25
202D: A5 99 )26
202F: F0 48 )27
2031: 10 09 )28
2033: 49 FF )29
2035: 18 )30
2036: 69 01 )31
2038: 85 99 )32
203A: 06 97 )33
203C: A5 9A )34
203E: F0 39 )35
2040: 10 09 )36
2042: 49 FF )37
2044: 18 )38
2045: 69 01 )39
2047: 85 9A )40
2049: E6 97 )41
204B: 46 9A )42
204D: 90 0D )43
204F: A5 99 )44
2051: 18 )45
2052: 65 9B )46
2054: 85 9B )47
2056: A5 98 )48
2058: 65 9C )49
205A: 85 9C )50
205C: 06 99 )51
205E: 26 98 )52
2060: A5 9A )53
2062: D0 E7 )54
2064: A5 97 )55
2066: F0 11 )56

PUT CHESS1.2
ORG $2000
JMP PCMON
LDA TEMPB
CLC
ADC STUKWRDLOB
STA STUKWRDLOB
LDA TEMPC
ADC STUKWRDHOB
STA STUKWRDHOB
RTS
TOTAL LDA STELWRDLOB
STA TEMPB
LDA STELWRDHOB
STA TEMPC
JSR $2000
LDA STUKWRDLOB
STA STELWRDLOB
LDA STUKWRDHOB
STA STELWRDHOB
RTS
MULTIPLY STY TEMP7
STY TEMP8
STY TEMPB
STY TEMPC
M1 LDA TEMP9
BEQ OUTM
BPL M2
EOR #$FF
CLC
ADC #1
STA TEMP9
DEC TEMP7
M2 LDA TEMP4
BEQ OUTM
BPL M3
EOR #$FF
CLC
ADC #1
STA TEMP4
INC TEMP7
M3 LSR TEMP4
BCC M4
LDA TEMP9
CLC
ADC TEMPB
STA TEMPB
LDA TEMP8
ADC TEMPC
STA TEMPC
M4 ASL TEMP9
ROL TEMP8
LDA TEMP4
BNE M3
LDA TEMP7
BEQ OUTM

; INITPOS EQU $2707 ; INITIALISERING
; MAINSPOS EQU $2787 ; STUURROUTINE
; REPRESENTATIE VAN DE STUKKEN
; LEEG VELD = 00
; WITTE PION = 80 ZWARTE PION = C1
; KONING = 82 KONING = C2
; PAARD = 83 PAARD = C3
; TOREN = 84 TOREN = C4
; LOPER = 85 LOPER = C5
; DAME = 86 DAME = C6
; INITIALISEREN
; TEL DE MOMENTELE WAARDE
; BIJ DE STUKWAARDE
; EN BEWAAR RESULTAAT
; TEL STUKWAARDE
; BIJ STELLINGWAARDE
; MET BEHULP VAN ADD
; EN BEWAAR RESULTAAT
; RESET
; NEEM ABSOLUTE WAARDE
; VERMENIGVULDIGER
; JUMP OUT ALS VERM = 0
; SIGN = -1 ALS
; VERM = NEGATIEF
; NEEM ABSOLUTE WAARDE
; VERMENIGVULDIGITAL
; JUMP OUT ALS VERMT = 0
; SIGN +1 ALS
; VERMT = NEGATIEF
; SCHUIF VERMT NAAR RECHTS
; JUMP ALS CARRY = 0
; TEL VERMT BIJ PROD
; SCHUIF VERM NAAR LINKS
; JMP TOTDAT VERMT=0
; NEEM 2 KOMPLEMENT VAN PROD
; ALS SIGN () 0

```



```

2068: A5 9B )58 LDA TEMPB
206A: 49 FF )59 EOR #FF
206C: 18 )60 CLC
206D: 69 01 )61 ADC #1
206F: 85 9B )62 STA TEMPB
2071: A5 9C )63 LDA TEMPC
2073: 49 FF )64 EOR #FF
2075: 69 00 )65 ADC #0
2077: 85 9C )66 STA TEMPC
2079: 60 )67 OUTM
DIVIDE RTS
207A: 84 96 )68 STY TEMP6 ; RESET
207C: 84 9B )69 STY TEMPB
207E: 84 9C )70 STY TEMPC
2080: A5 98 )71 D1 LDA TEMP8 ; TEST DEELTAL
2082: D0 06 )72 BNE D1A
2084: A5 97 )73 LDA TEMP7
2086: F0 54 )74 BEQ OUTD ; JUMP OUT ALS DEELTAL = 0
2088: A5 98 )75 LDA TEMP8 ; SCHUIF DEELTAL ZOVER
208A: 30 08 )76 D1A BMI D2 ; MOGELIJK LINKS
208C: C6 96 )77 D1A1 DEC TEMP6 ; HOUDT AANTAL SHIFTS
208E: 06 97 )78 ASL TEMP7 ; BIJ IN TEMP6
2090: 26 98 )79 ROL TEMP8
2092: 10 F8 )80 BPL D1A1
2094: A5 9A )81 D2 LDA TEMP8 ; TEST DELER
2096: D0 06 )82 BNE D2A ; JUMP OUT ALS DELER = 0
2098: A5 97 )83 LDA TEMP9
209A: F0 40 )84 BEQ OUTD
209C: A5 9A )85 LDA TEMP8
209E: 30 08 )86 D2A BMI D2A1
20A0: E6 96 )87 D2A2 INC TEMP6 ; SCHUIF DELER ZOVER MOGELIJK LINKS
20A2: 06 99 )88 ASL TEMP9 ; HOUDT AANTAL SHIFTS BIJ IN TEMP6
20A4: 26 9A )89 ROL TEMP8
20A6: 10 F8 )90 BPL D2A2
20A8: A5 96 )91 D2A1 LDA TEMP6
20AA: 30 30 )92 BMI OUTD ; JUMP OUT ALS TEMP6 < 0 :
20AC: 38 )93 D3 SEC ; DAN IS DELER > DEELTAL
20AD: A5 97 )94 LDA TEMP7 ; TREK DEELTAL VAN DELER AF
20AF: E5 97 )95 SBC TEMP9
20B1: 85 97 )96 STA TEMP7
20B3: A5 98 )97 LDA TEMP8
20B5: E5 9A )98 SBC TEMP8
20B7: 85 98 )99 STA TEMP8
20B9: 90 08 )100 D4 BCC D4A ; DRAAI CARRY IN QUOTIENT
20BB: 26 9B )101 ROL TEMPB ; TEL DELER BIJ DEELTAL
20BD: 26 9C )102 ROL TEMPC ; ALS CARRY = 0
20BF: F0 13 )103 BEQ D5
20C1: D0 11 )104 BNE D5
20C3: 26 9B )105 D4A ROL TEMPB
20C5: 26 9C )106 ROL TEMPC
20C7: 18 )107 CLC
20C8: A5 97 )108 LDA TEMP7
20CA: 65 99 )109 ADC TEMP9
20CC: 85 97 )110 STA TEMP7
20CE: A5 98 )111 LDA TEMP8
20D0: 65 9A )112 ADC TEMP8
20D2: 85 98 )113 STA TEMP8
20D4: 46 9A )114 D5 LSR TEMP8 ; SCHUIF DELER NAAR RECHTS
20D6: 66 99 )115 ROR TEMP9 ; DECREMENTEER SHIFTELLER
20D8: C6 96 )116 DEC TEMP6
20DA: 10 D0 )117 BPL D3 ; JUMP TOTDAT SHIFTELLER < 0
20DC: 60 )118 OUTD RTS
20DD: A6 B0 )119 SCAN LDX VANVELDHEX ; X = VANVELD
20DF: CA )120 SCAN1 DEX
20E0: 30 18 )121 BMI SCANOUT ; JUMP ALS X < 0
20E2: B5 16 )122 LDA BORD, X ; A = BORD
20E4: F0 F9 )123 BEQ SCAN1 ; JUMP ALS BORD = GEEN S'UK
20E6: 85 94 )124 STA TEMP4
20E8: 29 0F )125 AND #0F
20EA: C5 AC )126 CMP STUKIO
20EC: D0 F1 )127 BNE SCAN1 ; JUMP ALS STUK <> STUKIO
20EE: A5 94 )128 LDA TEMP4
20F0: 29 40 )129 AND #40
20F2: 45 A1 )130 EOR KWIS

```

```

20F4: D0 E9 >132 BNE SCAN1 ; JUMP ALS KLEUR (<) KWIS
20F6: 86 B0 >133 STX VANVELDHEX ; SAVE VANVELD
20F8: F0 02 >134 BEQ SCANOUT2
20FA: A9 FF >135 SCANOUT LDA #FF ; KEER TERUG MET A=FF : ALLE VELDEN GEHAD
20FC: 60 >136 SCANOUT2 RTS
20FD: A5 B1 >137 KODEER LDA VERTREKHEX ; BEREKEN VERTIKALE
20FF: 4A >138 LSRA ; KOORDINAAT
2100: 4A >139 LSRA
2101: 4A >140 LSRA
2102: 85 94 >141 STA TEMP4
2104: A5 B1 >142 LDA VERTREKHEX ; BEREKEN HORIZONTALE
2106: 29 07 >143 AND #7 ; KOORDINAAT
2108: 0A >144 ASLA
2109: 0A >145 ASLA
210A: 0A >146 ASLA
210B: 0A >147 ASLA
210C: 05 94 >148 ORA TEMP4 ; NEEM KOORDINATEN BIJ ELKAAR
210E: 18 >149 CLC ; TEL 11 BIJ RESULTAAT
210F: 69 11 >150 ADC #11
2111: 85 B2 >151 STA VERTREKHEX ; RESULTAAT IN VERTREKHEX
2113: 60 >152 RTS
2114: B5 D8 >153 VELDKONTROLDA DIRDEC, X ; HAAL DECIMALE RICHTINGTABEL
2116: 18 >154 CLC
2117: F8 >155 SED
2118: 65 B2 >156 ADC VERTREKHEX ; TEL OP BIJ VERTREKHEX
211A: D8 >157 CLD ; RESULTAAT IS NAARVELD
211B: 85 94 >158 STA TEMP4 ; NAARVELD IN TEMP4
211D: 29 0F >159 AND #0F
211F: F0 0E >160 BEQ OUTVK1 ; KONTROLEER GELDIGHEID NAARVELD
2121: C9 09 >161 CMP #9
2123: F0 0A >162 BEQ OUTVK1 ; JUMP ALS NAARVELD = ONGELDIG
2125: A5 94 >163 LDA TEMP4
2127: 29 F0 >164 AND #F0
2129: F0 04 >165 BEQ OUTVK1
212B: C9 90 >166 CMP #90
212D: D0 02 >167 BNE OUTVK2 ; JUMP ALS NAARVELD = GELDIG
212F: A9 FF >168 OUTVK1 LDA #FF
2131: 60 >169 OUTVK2 RTS
2132: B5 D0 >170 NAARVELD DIRHEX, X ; HAAL HEXADECIMALE RICHTINGTABEL
2134: 18 >171 CLC
2135: 65 B1 >172 ADC VERTREKHEX ; TEL OP BIJ VERTREKHEX
2137: 85 B1 >173 STA VERTREKHEX
2139: 86 9D >174 STX TEMPD ; NAARVELD IS NIEUW VERTREKVELD
213B: AA >175 TAX
213C: B5 16 >176 LDA BORD, X ; HAAL STUK VAN NAARVELD
213E: 85 AE >177 STA SLASTUK ; EN ZET DAT IN SLASTUK
2140: A6 9D >178 LDX TEMPD
2142: 60 >179 RTS
2143: A5 99 >180 AFSTAND LDA TEMP9 ; HAAL EERSTE VELD
2145: 29 0F >181 AND #0F ; DISTILLEER VERTIKALE KOORDINAAT
2147: 85 9B >182 STA TEMPB
2149: A5 9A >183 LDA TEMP9 ; HAAL TWEEDE VELD
214B: 29 0F >184 AND #0F ; DISTILLEER VERTIKALE KOORDINAAT
214D: 38 >185 SEC
214E: E5 9B >186 SBC TEMPB ; BEREKEN HET VERSCHIL
2150: 10 05 >187 BPL AFST1
2152: 49 FF >188 EOR #FF ; BEREKEN ABSOLUTE WAARDE
2154: 18 >189 CLC
2155: 69 01 >190 ADC #1
2157: 85 9C >191 AFST1 STA TEMPC ; SAVE RESULTAAT
2159: A5 99 >192 LDA TEMP9 ; HAAL EERSTE VELD
215B: 29 F0 >193 AND #F0 ; DISTILLEER HORIZONTALE KOORDINAAT
215D: 85 9B >194 STA TEMPB
215F: A5 9A >195 LDA TEMP9 ; HAAL TWEEDE VELD
2161: 29 F0 >196 AND #F0 ; DISTILLEER HORIZONTALE KOORDINAAT
2163: 38 >197 SEC
2164: E5 9B >198 SBC TEMPB ; BEREKEN HET VERSCHIL
2166: 10 05 >199 BPL AFST2
2168: 49 FF >200 EOR #FF ; BEREKEN ABSOLUTE WAARDE
216A: 18 >201 CLC
216B: 69 01 >202 ADC #1
216D: 4A >203 AFST2 LSRA ; SCHUIF RESULTAAT IN RECHTER NIBBLE
216E: 4A >204 LSRA

```

```

216F: 4A      )206      LSRA
2170: 4A      )207      LSRA
2171: 18      )208      CLC
2172: 65 9C    )209      ADC  TEMPC ; TEL RESULTATEN BIJ ELKAAR
2174: 85 9C    )210      STA  TEMPC ; RESULTAAT IN TEMPC
2176: 60      )211      RTS
2177: A5 A1    )212      FIRSTVELDLDA KWIS ; ZET LOOPRICHTING UIT
2179: F0 04    )213      BEQ  FV1
217B: A2 07    )214      LDX  #7 ; ZWART = 7
217D: D0 02    )215      BNE  FV1A
217F: A2 03    )216      LDX  #3 ; WIT = 3
2181: A5 B0    )217      FV1A LDA VANVELDHEX ; ZET VAN IN VERTREK
2183: 85 B1    )218      STA  VERTREKHEX
2185: 84 B4    )219      STY  BITSTUKIO ; ZET BITSTUKIO OP 01
2187: E6 B4    )220      INC  BITSTUKIO
2189: 20 FD 20 )221      FV2 JSR  KODEER
218C: 20 14 21 )222      JSR  VELDKONTR
218F: C9 FF    )223      CMP  #$FF
2191: F0 07    )224      BEQ  FV3 ; JUMP ALS A=FF : ONGELDIGE
2193: 20 32 21 )225      JSR  NAARVELD
2196: 06 B4    )226      ASL  BITSTUKIO
2198: D0 EF    )227      BNE  FV2 ; JUMP TOTDAT LIJN OP IS
219A: A5 B1    )228      FV3 LDA  VERTREKHEX ; RESULTAAT IN EERSTEVELD
219C: 85 B3    )229      STA  EERSTEVELD
219E: 60      )230      RTS
219F: A5 A1    )231      STATUSI LDA KWIS ; ZET LOOPRICHTING UIT
21A1: F0 04    )232      BEQ  STAT1
21A3: A2 03    )233      LDX  #3 ; ZWART = 3
21A5: D0 02    )234      BNE  STAT2
21A7: A2 07    )235      STAT1 LDX #7 ; WIT = 7
21A9: 84 95    )236      STAT2 STY TEMP5 ; RESET
21AB: 84 96    )237      STY  TEMP6
21AD: 84 97    )238      STY  TEMP7
21AF: E6 95    )239      INC  TEMP5 ; TEMP5 = EERSTEVELD
21B1: 20 FD 20 )240      STAT3 JSR  KODEER
21B4: 20 14 21 )241      JSR  VELDKONTR
21B7: C9 FF    )242      CMP  #$FF
21B9: D0 01    )243      BNE  STAT4 ; JUMP ZOLANG ER GELDIGE VELDEN ZIJN
21BB: 60      )244      RTS
21BC: 20 32 21 )245      STAT4 JSR  NAARVELD
21BF: 06 95    )246      ASL  TEMP5 ; LEFT SHIFT TEMP5 VOOR IEDERE ZET
21C1: A5 AE    )247      LDA  SLASTUK ; HAAL SLASTUK
21C3: F0 EC    )248      BEQ  STAT3 ; JUMP ALS SLASTUK = 0
21C5: 29 07    )249      AND  #7
21C7: C9 02    )250      CMP  #2
21C9: 10 E6    )251      BPL  STAT3 ; JUMP ALS SLASTUK (<) PION
21CB: A5 AE    )252      LDA  SLASTUK
21CD: 29 40    )253      AND  #$40
21CF: 45 A1    )254      EOR  KWIS
21D1: D0 08    )255      BNE  STAT5 ; JUMP NAAR PION KAZ
21D3: A5 95    )256      LDA  TEMP5 ; BEWAAR PIONPOSITIE KNAZ IN TEMP6
21D5: 05 96    )257      ORA  TEMP6
21D7: 85 96    )258      STA  TEMP6
21D9: D0 D6    )259      BNE  STAT3
21DB: A5 95    )260      STAT5 LDA TEMP5 ; BEWAAR PIONPOSITIE KAZ IN TEMP7
21DD: 05 97    )261      ORA  TEMP7
21DF: 85 97    )262      STA  TEMP7
21E1: D0 CE    )263      BNE  STAT3
21E3: A2 05    )264      STATUSII LDX #5 ; ZET LOOPRICHTING NAAR RECHTS
21E5: A5 B3    )265      LDA  EERSTEVELD ; EERSTEVELD IN VERTREKHEX
21E7: 85 B1    )266      STA  VERTREKHEX
21E9: 20 FD 20 )267      JSR  KODEER
21EC: 20 14 21 )268      JSR  VELDKONTR
21EF: C9 FF    )269      CMP  #$FF
21F1: D0 06    )270      BNE  STATIII ; JUMP ALS A = FF : GEEN RIJ
21F3: 84 B7    )271      STY  BITKAZR ; ZET BITCODE AANVALER EN
21F5: 84 B8    )272      STY  BITKNAZR ; VERDEDIGERS RECHTS OP 00
21F7: F0 0D    )273      BEQ  STATIII2
21F9: E6 B1    )274      STATIII1 INC VERTREKHEX ; VERTREKVELD LIET RIJ NAAR RECHTS
21FB: 20 9F 21 )275      JSR  STATUSI
21FE: A5 96    )276      LDA  TEMP6
2200: 85 B7    )277      STA  BITKAZR ; VERDEDIGERS RECHTS IN BITKAZR
2202: A5 97    )278      LDA  TEMP7 ; AANVALLERS RECHTS IN BITKNAZR

```

```

2204: 85 B8 )280 STA BITKNAZR
2206: A2 01 )281 STATI12 LDX #1 ; ZET LOOPRICHTING NAAR LINKS
2208: A5 B3 )282 LDA EERSTEVELD ; EERSTEVELD IN VERTREKHEX
220A: 85 B1 )283 STA VERTREKHEX
220C: 20 FD 20 )284 JSR KODEER
220F: 20 14 21 )285 JSR VELDKONTR
2212: C9 FF )286 CMP #FF
2214: D0 06 )287 BNE STATI13 ; JUMP ALS A = FF : GEEN RIJ
2216: 84 B9 )288 STY BITKAZL ; ZET BITCODE AANVALLERS EN
2218: 84 BA )289 STY BITKNAZL ; VERDEDIGERS LINKS OP 00
221A: F0 0D )290 BEQ STATI1OUT
221C: C6 B1 )291 STATI13 DEC VERTREKHEX ; VERTREKVELD LIGT RIJ NAAR LINKS
221E: 20 9F 21 )292 JSR STATUSI
2221: A5 96 )293 LDA TEMP6
2223: 85 B9 )294 STA BITKAZL ; VERDEDIGERS LINKS IN BITKAZL
2225: A5 97 )295 LDA TEMP7
2227: 85 BA )296 STA BITKNAZL ; AANVALLER LINKS IN BITKNAZL
2229: 60 )297 STATI1OUT RTS
222A: A9 40 )1 PUT CHESS1.3
222C: 85 B0 )2 PAARD LDA #40 ; VANVELD = 40
222E: A9 03 )3 LDA #3 ; STUK = PAARD
2230: 85 AC )4 STA STUKIO
2232: 84 A8 )5 STY STUKWRDLOB ; RESET STUKTELLER
2234: 84 A9 )6 STY STUKWRDHOB
2236: 20 DD 20 )7 STARTP JSR SCAN
2239: C9 FF )8 CMP #FF ; A ( ) FF : STUK GEVONDEN
223B: D0 04 )9 BNE P1 ; VERWERK
223D: 20 11 20 )10 JSR TOTAL
2240: 60 )11 RTS ; RETURN
2241: A5 B0 )12 P1 LDA VANVELDHEX ; P1 KODEERT POSITIE VAN HET STUK
2243: 85 B1 )13 STA VERTREKHEX
2245: 20 FD 20 )14 JSR KODEER
2248: 85 AD )15 P2 STA POSITIO ; AFSTAND PAARD EN KONING KNAZ
224A: 85 99 )16 STA TEMP9
224C: A5 AB )17 LDA KONKNAZ
224E: 85 9A )18 STA TEMP4
2250: 20 43 21 )19 JSR AFSTAND
2253: A9 05 )20 LDA #5
2255: 38 )21 SEC ; 5 - AFSTAND PAARD EN KONING KNAZ
2256: E5 9C )22 SBC TEMP4
2258: 85 99 )23 STA TEMP9
225A: A9 0C )24 LDA #0C ; WEEGFAKTOR IS +12
225C: 85 9A )25 STA TEMP4
225E: 20 25 20 )26 JSR MULTIPLY
2261: 20 03 20 )27 JSR ADD
2264: A5 AD )28 P3 LDA POSITIO ; AFSTAND PAARD - CENTRUM
2266: 85 99 )29 STA TEMP9
2268: A9 44 )30 LDA #44 ; CENTRUM = 44
226A: 85 9A )31 STA TEMP4
226C: 20 43 21 )32 JSR AFSTAND
226F: 85 94 )33 STA TEMP4
2271: A9 55 )34 LDA #55 ; CENTRUM = 55
2273: 85 9A )35 STA TEMP4
2275: 20 43 21 )36 JSR AFSTAND
2278: 18 )37 CLC
2279: 65 94 )38 ADC TEMP4
227B: 85 94 )39 STA TEMP4
227D: A9 06 )40 LDA #6 ; 6 - AFSTAND PAARD - CENTRUM
227F: 38 )41 SEC
2280: E5 94 )42 SBC TEMP4
2282: 85 99 )43 STA TEMP9
2284: A9 10 )44 LDA #10 ; WEEGFAKTOR IS +16
2286: 85 9A )45 STA TEMP4
2288: 20 25 20 )46 JSR MULTIPLY
228B: 20 03 20 )47 JSR ADD
228E: 20 77 21 )48 P4 JSR FIRSTVELD ; ONTWIKKELING
2291: A5 B4 )49 LDA BITSTUKIO
2293: C9 01 )50 CMP #1
2295: D0 9F )51 BNE STARTP ; NIET ONTWIKKELD INDIEN BITVELD = 01

2297: A9 A2 )53 LDA #A2

```

2299:	85	9B	>54	STA	TEMPB	; WEEGFAKTOR IS -94
229B:	A9	FF	>55	LDA	##FF	
229D:	85	9C	>56	STA	TEMPC	
229F:	20	03	>57	JSR	ADD	
22A2:	4C	36	>58	JMP	STARTP	
22A5:	A9	40	>59	LDA	##40	; VANVELD = 40
22A7:	85	B0	>60	STA	VANVELDHEX	; STUK = LOPER
22A9:	A9	05	>61	LDA	#5	
22AB:	85	AC	>62	STA	STUKIO	
22AD:	84	A8	>63	STY	STUKWRDLOB	; RESET STUKTELLER
22AF:	84	A9	>64	STY	STUKWRDHOB	
22B1:	84	AA	>65	STY	VELDTTELLER	; RESET VELDTTELLER
22B3:	20	DD	>66	JSR	SCAN	
22B6:	C9	FF	>67	CMP	##FF	; A (>) FF : STUK GEVONDEN
22B8:	D0	04	>68	BNE	L1	; VERWERK
22BA:	20	11	>69	JSR	TOTAL	
22BD:	60		>70	RTS		
22BE:	A2	08	>71	L1	LDX	#8
22C0:	A5	B0	>72	L2	LDA	VANVELDHEX
22C2:	85	B1	>73	STA	VERTREKHEX	; X = RICHTING
22C4:	CA		>74	DEX		; ZOEK IN VOLGENDE RICHTING
22C5:	CA		>75	DEX		; WERK RICHTING BIJ
22C6:	30	27	>76	BMI	L5	
22C8:	20	FD	>77	JSR	KODEER	; ALLE RICHTINGEN GEHAD
22CB:	20	14	>78	JSR	VELDKONTR	; ZOEK IN DEZELFDE RICHTING
22CE:	C9	FF	>79	CMP	##FF	
22D0:	F0	EE	>80	BEQ	L2	; FF=GEEN GELDIG VELD, VOLGENDE RICHTING
22D2:	20	32	>81	JSR	NAARVELD	
22D5:	A5	AE	>82	LDA	SLASTUK	
22D7:	F0	0E	>83	BEQ	L4	; NAARVELD IS LEEG
22D9:	29	40	>84	AND	##40	
22DB:	45	A1	>85	EOR	KWIS	
22DD:	D0	08	>86	BNE	L4	; STUK KNAZ
22DF:	A5	AE	>87	LDA	SLASTUK	
22E1:	29	07	>88	AND	#7	
22E3:	C9	02	>89	CMP	#2	
22E5:	30	D9	>90	BMI	L2	; PION KNAZ
22E7:	E6	AA	>91	L4	INC	VELDTTELLER
22E9:	A5	AE	>92	LDA	SLASTUK	; L4 : TELT DE VELDEN
22EB:	F0	DB	>93	BEQ	L3	
22ED:	D0	D1	>94	BNE	L2	; NAARVELD=LEEGL: DOORGAAN IN ZELFDE RICHTING
22EF:	A5	AA	>95	L5	LDA	VELDTTELLER
22F1:	38		>96	SEC		; L5 : VERWERKT DE VELDTTELLER
22F2:	E9	07	>97	SBC	#7	
22F4:	85	99	>98	STA	TEMP9	; AANTAL VELDEN - 7
22F6:	A9	0C	>99	LDA	##0C	; WEEGFAKTOR = 12
22F8:	85	9A	>100	STA	TEMPA	
22FA:	20	25	>101	JSR	MULTIPLY	
22FD:	20	03	>102	JSR	ADD	
2300:	A5	B0	>103	L6	LDA	VANVELDHEX
2302:	85	B1	>104	STA	VERTREKHEX	; ONTWIKKELING
2304:	20	77	>105	JSR	FIRSTVELD	
2307:	A5	B4	>106	LDA	BITSTUKIO	
2309:	C9	01	>107	CMP	#1	; NIET ONTWIKKELD INDIEN BITVELD = 01
230B:	D0	A4	>108	BNE	STARTL	
230D:	A9	92	>109	LDA	##92	; WEEGFAKTOR IS -110
230F:	85	9B	>110	STA	TEMPB	
2311:	A9	FF	>111	LDA	##FF	
2313:	85	9C	>112	STA	TEMPC	
2315:	20	03	>113	JSR	ADD	
2318:	4C	B1	>114	JMP	STARTL	
231B:	A9	40	>115	DAME	LDA	##40
231D:	85	B0	>116	STA	VANVELDHEX	; VANVELD = 40
231F:	A9	06	>117	LDA	#6	; STUK = DAME
2321:	85	AC	>118	STA	STUKIO	
2323:	84	A8	>119	STY	STUKWRDLOB	; RESET STUKTELLER
2325:	84	A9	>120	STY	STUKWRDHOB	
2327:	84	AA	>121	STY	VELDTTELLER	; RESET VELDTTELLER
2329:	20	DD	>122	JSR	SCAN	
232C:	C9	FF	>123	CMP	##FF	; A (>) FF : STUK GEVONDEN
232E:	D0	04	>124	BNE	DEEN	; VERWERK

```

2330: 20 11 20 >126 JSR TOTAL
2333: 60 >127 RTS ; RETURN
2334: A5 B0 >128 DEEN LDA VANVELDHEX ; DEEN KODEERT POSITIE VAN HET STUK
2336: 85 B1 >129 STA VERTREKHEX
2338: 20 FD 20 >130 JSR KODEER
233B: 85 AD >131 STA POSITIO
233D: A2 08 >132 DTWEE LDX #8 ; X = RICHTING
233F: A5 B0 >133 DDRIE LDA VANVELDHEX ; ZOEK IN VOLGENDE RICHTING
2341: 85 B1 >134 STA VERTREKHEX
2343: CA >135 DEX ; WERK RICHTING BIJ
2344: 30 15 >136 BMI DZES ; ALLE RICHTINGEN GEHAD
2346: 20 FD 20 >137 DVIER JSR KODEER ; ZOEK IN DEZELFDE RICHTING
2349: 20 14 21 >138 JSR VELDKONTR
234C: C9 FF >139 CMP #FF
234E: F0 EF >140 BEQ DDRIE ; FF = GEEN GELDIG VELD
2350: 20 32 21 >141 JSR NAARVELD
2353: E6 AA >142 DVIJF INC VELDTELLER ; DVIJF TELT DE VELDEN
2355: A5 AE >143 LDA SLASTUK
2357: F0 ED >144 BEQ DVIER ; NAARVELD=LEEG: DUORGAAN IN ZELFDE RICHTING
2359: D0 E4 >145 BNE DDRIE ; NAARVELD HEEFT STUK: VOLGENDE RICHTING
235B: A5 AA >146 DZES LDA VELDTELLER ; DZES VERWERKT DE VELDTELLER
235D: 85 99 >147 STA TEMP9
235F: A5 A5 >148 LDA STKNUMKNAZ
2361: 4A >149 LSRA
2362: 4A >150 LSRA ; WEEGFAKTOR = 9 - AANTAL STUKKEN KNAZ
2363: 4A >151 LSRA
2364: 4A >152 LSRA
2365: 85 98 >153 STA TEMP8
2367: A9 09 >154 LDA #9
2369: 38 >155 SEC
236A: E5 98 >156 SBC TEMP8
236C: 85 9A >157 STA TEMP8
236E: 20 25 20 >158 JSR MULTIPLY
2371: 20 03 20 >159 JSR ADD
2374: A5 AD >160 DZEVEN LDA POSITIO ; AFSTAND DAME - KONING KNAZ
2376: 85 99 >161 STA TEMP9
2378: A5 AB >162 LDA KONKNAZ
237A: 85 9A >163 STA TEMP8
237C: 20 43 21 >164 JSR AFSTAND
237F: A5 9C >165 LDA TEMPC
2381: 85 99 >166 STA TEMP9
2383: A9 F8 >167 LDA #F8 ; WEEGFAKTOR IS -8
2385: 85 9A >168 STA TEMP8
2387: 20 25 20 >169 JSR MULTIPLY
238A: 20 03 20 >170 JSR ADD
238D: 4C 27 23 >171 JMP STARTD
2390: A9 40 >172 TOREN LDA #40 ; VANVELD = 40
2392: 85 B0 >173 STA VANVELDHEX
2394: A9 04 >174 LDA #4 ; STUK = TOREN
2396: 85 AC >175 STA STUKIO
2398: 84 A8 >176 STY STUKWRDLOB ; RESET STUKTELLER
239A: 84 A9 >177 STY STUKWRDHOB
239C: 84 AA >178 STARTT STY VELDTELLER ; RESET VELDTELLER
239E: 20 DD 20 >179 JSR SCAN
23A1: C9 FF >180 CMP #FF ; A (>) FF : STUK GEVONDEN
23A3: D0 04 >181 BNE T1 ; VERWERK
23A5: 20 11 20 >182 JSR TOTAL
23A8: 60 >183 RTS
23A9: A5 B0 >184 T1 LDA VANVELDHEX ; KODEER POSITIE VAN HET STUK
23AB: 85 B1 >185 STA VERTREKHEX
23AD: 20 FD 20 >186 JSR KODEER
23B0: 85 AD >187 STA POSITIO
23B2: A2 09 >188 T2 LDX #9 ; X = RICHTING
23B4: A5 B0 >189 T3 LDA VANVELDHEX ; ZOEK IN VOLGENDE RICHTING
23B6: 85 B1 >190 STA VERTREKHEX
23B8: CA >191 DEX ; WERK RICHTING BIJ
23B9: CA >192 DEX
23BA: 30 34 >193 BMI T7 ; ALLE RICHTINGEN GEHAD
23BC: 20 FD 20 >194 T4 JSR KODEER ; ZOEK IN DEZELFDE RICHTING
23BF: C9 FF >195 JSR VELDKONTR
23C2: C9 FF >196 CMP #FF
23C4: F0 EE >197 BEQ T3 ; FF=GEEN GELDIG VELD: ZELFDE RICHTING

```

```

23C6: 20 32 21 >199 JSR NAARVELD
23C9: A5 AE >200 LDA SLASTUK
23CB: F0 1B >201 BEQ T6 ; NAARVELD = LEEG
23CD: 29 40 >202 AND #40
23CF: 45 A1 >203 EOR KWIS
23D1: D0 15 >204 BNE T6 ; STUK KNAZ
23D3: A5 AE >205 LDA SLASTUK
23D5: 29 07 >206 AND #7
23D7: C9 02 >207 CMP #2
23D9: 30 D9 >208 BMI T3 ; PION KNAZ
23DB: C9 04 >209 CMP #4
23DD: D0 09 >210 BNE T6 ; STUK (>) TOREN KAZ
23DF: A9 50 >211 LDA #50 ; DUBBELE TORENS
23E1: 85 9B >212 STA TEMPB ; WEEGFAKTOR IS +80
23E3: 84 9C >213 STY TEMPC
23E5: 20 03 20 >214 JSR ADD
23E8: E6 AA >215 T6 INC VELDTELLER ; T6 TELT DE VELDEN
23EA: A5 AE >216 LDA SLASTUK
23EC: F0 CE >217 BEQ T4 ; NAARVELD=LEEG: DOORGAAN IN ZELFDE RICHTING
23EE: D0 C4 >218 BNE T3 ; NAARVELD HEEFT STUK: VOLGENDE RICHTING
23F0: A5 AD >219 T7 LDA POSITIO ; T7 VERWERKT DE VELDTELLER
23F2: 85 99 >220 STA TEMP9 ; TEVENS DE AFSTAND TOREN - KONING KNAZ
23F4: A5 AB >221 LDA KONKNAZ
23F6: 85 9A >222 STA TEMP9
23F8: 20 43 21 >223 JSR AFSTAND
23FB: A5 AA >224 LDA VELDTELLER
23FD: 38 >225 SEC ; VELDTELLER -AFSTAND
23FE: E5 9C >226 SBC TEMPC
2400: 85 99 >227 STA TEMP9
2402: A9 10 >228 LDA #10 ; WEEGFAKTOR IS 16
2404: 85 9A >229 STA TEMP9
2406: 20 25 20 >230 JSR MULTIPLY
2409: 20 03 20 >231 JSR ADD
240C: 20 77 21 >232 JSR FIRSTVELD ; T9 : ZEVENDE LIJN
240F: A5 B4 >233 LDA BITSTUKIO
2411: C9 40 >234 CMP #40
2413: D0 09 >235 BNE T10 ; BITCODE (>) 40 : GEEN ZEVENDE LIJN
2415: A9 DC >236 LDA #DC ; WEEGFAKTOR = +220
2417: 85 9B >237 STA TEMPB
2419: 84 9C >238 STY TEMPC
241B: 20 03 20 >239 JSR ADD
241E: A5 B3 >240 T10 LDA EERSTEVELD ; T10 : OPEN LIJN
2420: 85 B1 >241 STA VERTREKHEX
2422: 20 9F 21 >242 JSR STATUSI
2425: A5 96 >243 LDA TEMP6
2427: D0 45 >244 BNE OUTT ; JUMP: LIJN NIET OPEN OF HALF OPEN
2429: A5 97 >245 LDA TEMP7
242B: D0 0C >246 BNE T11 ; JUMP, HALF OPEN LIJN
242D: A9 50 >247 LDA #50
242F: 85 9B >248 STA TEMPB ; WEEGFAKTOR IS +80
2431: 84 9C >249 STY TEMPC
2433: 20 03 20 >250 JSR ADD
2436: 98 >251 TYA
2437: F0 35 >252 BEQ OUTT
2439: A5 97 >253 T11 LDA TEMP7 ; T11 : HALF OPEN LIJN
243B: 84 B4 >254 STY BITSTUKIO ; B4 WORDT BITCODE VERST DOORGESCHOVEN
243D: E6 B4 >255 INC BITSTUKIO ; PION KNAZ
243F: 18 >256 CLC
2440: 4A >257 T11A LSRA
2441: B0 04 >258 BCS T11B
2443: 06 B4 >259 ASL BITSTUKIO
2445: D0 F9 >260 BNE T11A
2447: 20 E3 21 >261 T11B JSR STATUSII
244A: A5 B4 >262 LDA BITSTUKIO ; INDIEN PION KNAZ VERDEDIGD IS,
244C: 0A >263 ASLA ; GEEN HALF OPEN LIJN
244D: AA >264 TAX
244E: 25 B8 >265 AND BITKNAZ ; VERDEDIGING RECHTS ?
2450: D0 1C >266 BNE OUTT
2452: BA >267 TXA
2453: 25 BA >268 AND BITKNAZ ; VERDEDIGING LINKS ?
2455: D0 17 >269 BNE OUTT
2457: A5 B4 >270 LDA BITSTUKIO ; INDIEN PION NIET AANVALBAAR,
2459: 4A >271 LSRA ; GEEN HALF OPEN LIJN

```



```

245A: 4A      )273    LSRA      ; AANVAL RECHTS ?
245B: AA      )274    TAX
245C: 25 B7   )275    AND BITKAZR
245E: D0 05   )276    BNE T12      ; LIJN = HALF OPEN
2460: 8A      )277    TXA
2461: 25 B9   )278    AND BITKAZL ; AANVAL LINKS ?
2463: F0 09   )279    BEQ OUTT
2465: A9 20   )280    LDA ##20    ; WEEGFAKTOR IS +32
2467: 85 9B   )281    STA TEMPB
2469: 84 9C   )282    STY TEMPC
246B: 20 03 20 )283    JSR ADD
246E: 4C 9C 23 )284    JMP STARTT
2471: A9 40   )285    LDA ##40    ; VANVELD = 40
2473: 85 B0   )286    STA VANVELDHEX
2475: A5 A1   )287    LDA KWIS
2477: F0 06   )288    BEQ P10A
2479: A9 01   )289    LDA #1      ; STUK = ZWART PION
247B: 85 AC   )290    STA STUKIO
247D: D0 02   )291    BNE P10B
247F: 84 AC   )292    STY STUKIO ; STUK = WITTE PION
2481: 84 A8   )293    STY STUKWRDLOB ; RESET STUKTELLER
2483: 84 A9   )294    STY STUKWRDHOB
2485: 20 DD 20 )295    STARTPI
2488: C9 FF   )296    CMP ##FF    ; A ( ) FF : STUK GEVONDEN
248A: D0 04   )297    BNE P11      ; VERWERK
248C: 20 11 20 )298    JSR TOTAL
248F: 60      )299    RTS      ; RETURN
2490: A5 B0   )300    PI1 LDA VANVELDHEX ; KODEER POSITIE VAN HET STUK
2492: 85 B1   )301    STA VERTREKHEX
2494: 20 FD 20 )302    JSR KODEER
2497: 85 AD   )303    STA POSITIO
2499: 20 77 21 )304    PI2 JSR FIRSTVELD ; VOORBEREIDING
249C: A5 B3   )305    LDA EERSTEVELD ; B1 = EERSTEVELD
249E: 85 B1   )306    STA VERTREKHEX
24A0: 20 9F 21 )307    JSR STATUSI ; MAAK UITGEBREIDE BITVOORSTELLING
24A3: A5 96   )308    LDA TEMP6    ; VAN PIONNEN FORMATIE
24A5: 85 B5   )309    STA BITKAZ ; PIONNEN KAZ
24A7: A5 97   )310    LDA TEMP7
24A9: 85 B6   )311    STA BITKNAZ ; PIONNEN KNAZ
24AB: 20 E3 21 )312    JSR STATUSII
24AE: A2 00   )313    LDX #0      ; OPMARS
24B0: A5 B4   )314    PI3 LDA BITSTUKIO ; BEREKEN RIJNUMMER
24B2: E8      )315    PI3A INX
24B3: 4A      )316    LSRA
24B4: D0 FC   )317    BNE PI3A
24B6: 86 B2   )318    STX VERTREKREK ; SAVE RIJNUMMER
24B8: CA      )319    DEX
24B9: CA      )320    DEX
24BA: 86 99   )321    STX TEMP9    ; RIJNUMMER - 2
24BC: A5 AD   )322    LDA POSITIO
24BE: 4A      )323    LSRA
24BF: 4A      )324    LSRA
24C0: 4A      )325    LSRA
24C1: 4A      )326    LSRA
24C2: AA      )327    TAX
24C3: B5 C0   )328    LDA $C0,X    ; HAAL OPMARSTABEL
24C5: 85 9A   )329    STA TEMP9
24C7: 20 25 20 )330    JSR MULTIPLY
24CA: 20 03 20 )331    JSR ADD
24CD: A2 FF   )332    LDX ##FF    ; MEERDERHEID UIT 3
24CF: A5 B5   )333    PI4 LDA BITKAZ
24D1: F0 01   )334    BEQ PI4A
24D3: E8      )335    INX
24D4: A5 B6   )336    PI4A LDA BITKNAZ ; X+1 : PION KAZ OP DE LIJN
24D6: F0 01   )337    BEQ PI4B
24D8: CA      )338    DEX
24D9: A5 B7   )339    PI4B LDA BITKAZR ; X-1 : PION KNAZ OP DE LIJN
24DB: F0 01   )340    BEQ PI4C
24DD: E8      )341    INX
24DE: A5 B8   )342    PI4C LDA BITKNAZR ; X+1 : PION KAZ RECHTS
24E0: F0 01   )343    BEQ PI4D
24E2: CA      )344    DEX
24E3: A5 B9   )345    PI4D LDA BITKAZL ; X-1 : PION KNAZ RECHTS

```

24E5:	F0 01	>347	BEQ	PI4E	
24E7:	E8	>348	INX		
24E8:	A5 BA	>349	LDA	BITKNAZL	; X+1 : PION KAZ LINKS
24EA:	F0 01	>350	BEQ	PI4F	
24EC:	CA	>351	DEX		
24EE:	8A	>352	PI4F	TXA	; X-1 : PION KNAZ LINKS
24EF:	30 10	>353	BMI	PI5	; GEEN MEERDERHEID UIT 3 ALS X(0
24F0:	A6 B2	>354	LDX	VERTREKREK	; RIJNUMMER - 2
24F2:	CA	>355	DEX		
24F3:	CA	>356	DEX		
24F4:	86 99	>357	STX	TEMP9	
24F6:	A9 10	>358	LDA	##10	; WEEGFAKTOR IS +16
24F8:	85 9A	>359	STA	TEMPA	
24FA:	20 25 20	>360	JSR	MULTIPLY	
24FD:	20 03 20	>361	JSR	ADD	
2500:	A5 B6	>362	LDA	BITKNAZ	; VRIJPION
2502:	D0 1F	>363	BNE	PI6	; PION KNAZ OP DE LIJN
2504:	A5 B8	>364	LDA	BITKNAZR	
2506:	D0 1B	>365	BNE	PI6	; PION KNAZ RECHTS
2508:	A5 BA	>366	LDA	BITKNAZL	
250A:	D0 17	>367	BNE	PI6	; PION KNAZ LINKS
250C:	A5 B2	>368	LDA	VERTREKREK	
		4	PUT	CHESS1.4	
250E:	85 99	>1	STA	TEMP9	; KWADRATEER RIJNUMMER
2510:	85 9A	>2	STA	TEMPA	
2512:	20 25 20	>3	JSR	MULTIPLY	
2515:	A5 9B	>4	LDA	TEMPB	
2517:	85 99	>5	STA	TEMP9	
2519:	A9 10	>6	LDA	##10	; WEEGFAKTOR IS +16
251B:	85 9A	>7	STA	TEMPA	
251D:	20 25 20	>8	JSR	MULTIPLY	
2520:	20 03 20	>9	JSR	ADD	
2523:	A5 B4	>10	LDA	BITSTUKIO	; DUBBELPION
2525:	45 B5	>11	EOR	BITKAZ	
2527:	F0 0B	>12	BEQ	PI7	
2529:	A9 D8	>13	LDA	##D8	
252B:	85 9B	>14	STA	TEMPB	; WEEGFAKTOR IS -40
252D:	A9 FF	>15	LDA	##FF	
252F:	85 9C	>16	STA	TEMPC	
2531:	20 03 20	>17	JSR	ADD	
2534:	A5 B7	>18	LDA	BITKAZR	; GEISOLEERDE PION
2536:	D0 0F	>19	BNE	PI8	; PION KAZ RECHTS
2538:	A5 B9	>20	LDA	BITKAZL	
253A:	D0 0B	>21	BNE	PI8	; PION KAZ LINKS
253C:	A9 38	>22	LDA	##38	
253E:	85 9B	>23	STA	TEMPB	; WEEGFAKTOR IS -200
2540:	A9 FF	>24	LDA	##FF	
2542:	85 9C	>25	STA	TEMPC	
2544:	20 03 20	>26	JSR	ADD	
2547:	A5 B4	>27	LDA	BITSTUKIO	; ACHTERGEBLEVEN PION
2549:	4A	>28	LSRA		
254A:	AA	>29	TAX		
254B:	25 B7	>30	AND	BITKAZR	
254D:	D0 33	>31	BNE	OUTPI	; JUMP OUT : PION KAZ RECHTS
254F:	8A	>32	TXA		
2550:	25 B9	>33	AND	BITKAZL	
2552:	D0 2E	>34	BNE	OUTPI	; JUMP OUT : PION KAZ LINKS
2554:	8A	>35	TXA		
2555:	4A	>36	LSRA		
2556:	AA	>37	TAX		
2557:	25 B7	>38	AND	BITKAZR	
2559:	D0 27	>39	BNE	OUTPI	; JUMP OUT : RECHTS VERDEDIGBAAR
255B:	8A	>40	TXA		
255C:	25 B9	>41	AND	BITKAZL	
255E:	D0 22	>42	BNE	OUTPI	; JUMP OUT : LINKS VERDEDIGBAAR
2560:	84 95	>43	STY	TEMP5	
2562:	A5 B4	>44	LDA	BITSTUKIO	
2564:	0A	>45	ASLA		
2565:	0A	>46	ASLA		
2566:	AA	>47	TAX		; AANVAL RECHTS
2567:	25 B8	>49	AND	BITKNAZR	

2569:	F0 02	>50	BEQ	PI8A	
256B:	C6 95	>51	DEC	TEMP5	; TELLER -1
256D:	8A	>52	TXA		; AANVAL LINKS ?
256E:	25 BA	>53	AND	BITKNAZL	
2570:	F0 02	>54	BEQ	PI8B	
2572:	C6 95	>55	DEC	TEMP5	; TELLER -1
2574:	A5 95	>56	PI8B	LDA	TEMP5
2576:	85 99	>57		STA	TEMP9
2578:	A9 28	>58		LDA	#28
257A:	85 9A	>59		STA	TEMPA
257C:	20 25	20 >60		JSR	MULTIPLY
257F:	20 03	20 >61		JSR	ADD
2582:	4C 85	24 >62	OUTPI	JMP	STARTPI
2585:	A9 40	>63	KONINGI	LDA	#40
2587:	85 B0	>64		STA	VANVELDHEX ; VANVELD = 40
2589:	A9 02	>65		LDA	#2 ; STUK = KONING
258B:	85 AC	>66		STA	STUKIO
258D:	84 A8	>67		STY	STUKWRDLOB ; RESET STUKTELLER
258F:	84 A9	>68		STY	STUKWRDHOB
2591:	84 95	>69	STARTKI	STY	TEMP5 ; RESET HULPVELDEN
2593:	84 96	>70		STY	TEMP6
2595:	84 AA	>71		STY	VELDTELLER ; RESET VELDTELLER
2597:	20 DD	20 >72		JSR	SCAN
259A:	C9 FF	>73		CMP	#FF
259C:	D0 22	>74		BNE	KI1 ; A (<) FF : STUK GEVONDEN
259E:	A5 A5	>75		LDA	STKNUMKNAZ ; BEREKEN AANTAL STUKKEN KNAZ
25A0:	4A	>76		LSRA	
25A1:	4A	>77		LSRA	
25A2:	4A	>78		LSRA	
25A3:	4A	>79		LSRA	
25A4:	18	>80		CLC	
25A5:	65 AF	>81		ADC	DAMKNAZ ; RESULTAAT + AF
25A7:	38	>82		SEC	
25A8:	E9 02	>83		SBC	#2 ; RESULTAAT - 02
25AA:	F0 13	>84		BEQ	KIRET
25AC:	30 11	>85		BMI	KIRET
25AE:	AA	>86		TAX	
25AF:	18	>87	KIA	CLC	
25B0:	A5 A6	>88		LDA	STELWRDLOB
25B2:	65 A8	>89		ADC	STUKWRDLOB
25B4:	85 A6	>90		STA	STELWRDLOB ; OP DE STELLINGWAARDE
25B6:	A5 A7	>91		LDA	STELWRDHOB
25B8:	65 A9	>92		ADC	STUKWRDHOB
25BA:	85 A7	>93		STA	STELWRDHOB
25BC:	CA	>94		DEX	
25BD:	D0 F0	>95		BNE	KIA
25BF:	60	>96	KIRET	RTS	
25C0:	A5 B0	>97	KI1	LDA	VANVELDHEX ; KODEER POSITIE VAN HET STUK
25C2:	85 B1	>98		STA	VERTREKHEX
25C4:	20 FD	20 >99		JSR	KODEER
25C7:	85 AD	>100		STA	POSITIO
25C9:	A2 08	>101	KI2	LDX	#8 ; X = RICHTING
25CB:	A5 B0	>102	KI3	LDA	VANVELDHEX ; ZOEK IN VOLGENDE RICHTING
25CD:	85 B1	>103		STA	VERTREKHEX
25CF:	CA	>104		DEX	
25D0:	30 1F	>105		BMI	KI4 ; ALLE RICHTINGEN GEHAD
25D2:	20 FD	20 >106		JSR	KODEER
25D5:	20 14	21 >107		JSR	VELDKONTR
25D8:	C9 FF	>108		CMP	#FF
25DA:	F0 EF	>109		BEQ	KI3 ; FF = ONGELDIG VELD
25DC:	20 32	21 >110		JSR	NAARVELD
25DE:	A5 AE	>111		LDA	SLASTUK
25E1:	D0 04	>112		BNE	KI3A
25E3:	E6 95	>113		INC	TEMP5 ; TEMP5 TELT LEBE VELDEN IN LOOPBEREIK
25E5:	D0 E4	>114		BNE	KI3
25E7:	29 40	>115	KI3A	AND	#40
25E9:	45 A1	>116		EOR	KWIS
25EB:	D0 DE	>117		BNE	KI3 ; STUK KNAZ
25ED:	E6 96	>118		INC	TEMP6 ; TEMP6 TELT STUKKEN KAZ IN LOOPBEREIK
25EF:	D0 DA	>119		BNE	KI3
25F1:	A5 95	>120	KI4	LDA	TEMP5

25F3: D0 0B	>122	BNE	KI5	
25F5: A9 F8	>123	LDA	##F8	; WEEGFAKTOR IS -8
25F7: 85 9B	>124	STA	TEMPB	
25F9: A9 FF	>125	LDA	##FF	
25FB: 85 9C	>126	STA	TEMPC	
25FD: 20 03 20	>127	JSR	ADD	
2600: A5 96	>128	LDA	TEMP6	; VERWERK STUKKEN KAZ IN LOOPBEREIK
2602: C9 02	>129	CMP	#2	
2604: 10 15	>130	BPL	KI6	
2606: C9 01	>131	CMP	#1	
2608: D0 06	>132	BNE	KI5A	
260A: A9 F0	>133	LDA	##F0	; WEEGFAKTOR IS -16 VOOR 1 STUK
260C: 85 9B	>134	STA	TEMPB	
260E: D0 04	>135	BNE	KI5B	
2610: A9 E1	>136	LDA	##E1	; WEEGFAKTOR IS -31 VOOR GEEN STUK
2612: 85 9B	>137	STA	TEMPB	
2614: A9 FF	>138	LDA	##FF	
2616: 85 9C	>139	STA	TEMPC	
2618: 20 03 20	>140	JSR	ADD	
261B: 20 77 21	>141	JSR	FIRSTVELD	; VERDEDIGING
261E: A5 B4	>142	LDA	BITSTUKIO	
2620: 29 03	>143	AND	#3	
2622: F0 11	>144	BEQ	KI6B	; KONING TE VEEL NAAR VOREN
2624: A5 AD	>145	LDA	POSITIO	; MAAK BITCODE VAN KONINGSPOSITIE
2626: 4A	>146	LSRA		; IN HORIZONTALE RICHTING
2627: 4A	>147	LSRA		
2628: 4A	>148	LSRA		
2629: 4A	>149	LSRA		
262A: AA	>150	TAX		
262B: 38	>151	SEC		
262C: 98	>152	TYA		
262D: 6A	>153	RORA	KI6A	
262E: CA	>154	DEX		
262F: D0 FC	>155	BNE	KI6A	
2631: 29 E3	>156	AND	##E3	
2633: D0 0B	>157	BNE	KI7	; KONING STAAT VEILIG GENOEG
2635: A9 F0	>158	LDA	##F0	; WEEGFAKTOR IS -31
2637: 85 9B	>159	STA	TEMPB	
2639: A9 FF	>160	LDA	##FF	
263B: 85 9C	>161	STA	TEMPC	
263D: 20 03 20	>162	JSR	ADD	
2640: A5 B3	>163	LDA	EERSTVELD	; PION VOOR DE KONING
2642: 85 B1	>164	STA	VERTREKHEX	
2644: 20 9F 21	>165	JSR	STATUSI	
2647: A5 96	>166	LDA	TEMP6	
2649: D0 0B	>167	BNE	KI8	; PION KAZ AANWEZIG
264B: A9 D7	>168	LDA	##D7	; WEEGFAKTOR IS -41
264D: 85 9B	>169	STA	TEMPB	
264F: A9 FF	>170	LDA	##FF	
2651: 85 9C	>171	STA	TEMPC	
2653: 20 03 20	>172	JSR	ADD	
2656: 20 E3 21	>173	JSR	STATUSII	; PION OP AANGRENZENDE LIJNEN
2659: A5 B7	>174	LDA	BITKAZR	
265B: D0 0F	>175	BNE	OUTKI	; PION KAZ RECHTS
265D: A5 B9	>176	LDA	BITKAZL	
265F: D0 0B	>177	BNE	OUTKI	; PION KAZ LINKS
2661: A9 DC	>178	LDA	##DC	; WEEGFAKTOR IS -36
2663: 85 9B	>179	STA	TEMPB	
2665: A9 FF	>180	LDA	##FF	
2667: 85 9C	>181	STA	TEMPC	
2669: 20 03 20	>182	JSR	ADD	
266C: 4C 91 25	>183	OUTKI	JMP STARTKI	
266F: A9 40	>184	KONINGII	LDA ##40	; VANVELD = 40
2671: 85 B0	>185	STA	VANVELDHEX	
2673: A9 02	>186	LDA	#2	; STUK = KONING
2675: 85 AC	>187	STA	STUKIO	
2677: 84 A8	>188	STY	STUKWRDLOB	; RESET STUKTELLER
2679: 84 A9	>189	STY	STUKWRDHOB	
267B: 84 96	>190	STY	TEMP6	; RESET HULPVELD
267D: 20 DD 20	>191	JSR	SCAN	
2680: C9 FF	>192	CMP	##FF	
2682: D0 04	>193	BNE	KI11	; A <> FF : STUK GEVONDEN
2684: 20 11 20	>194	JSR	TOTAL	

2687:	60	)196		RTS	
2688:	A5 B0	)197	KIII1	LDA	VANVELDHEX ; KODEER POSITIE VAN HET STUK
268A:	85 B1	)198		STA	VERTREKHEX
268C:	20 FD 20	)199		JSR	KODEER
268F:	85 AD	)200		STA	POSITIO
2691:	CA 40	)201	KIII2	LDX	#\$40 ; X = RICHTING
2693:	A2	)202	KIII3	DEX	; SCAN BORD
2694:	30 21	)203		BMI	KIII4 ; FF = ONGELDIG VELD
2696:	B5 16	)204		LDA	BORD,X
2698:	F0 F9	)205		BEQ	KIII3 ; VELD = LEEG
269A:	29 07	)206		AND	#7
269C:	C9 02	)207		CMP	#2
269E:	10 F3	)208		BPL	KIII3 ; STUK = GEEN PION
26A0:	86 B1	)209		STX	VERTREKHEX
26A2:	20 FD 20	)210		JSR	KODEER
26A5:	85 99	)211		STA	TEMP9
26A7:	A5 AD	)212		LDA	POSITIO
26A9:	85 9A	)213		STA	TEMPA
26AB:	20 43 21	)214		JSR	AFSTAND
26AE:	A5 96	)215		LDA	TEMP6 ; TEMP6 TELT AFSTAND KONING - PIONNEN
26B0:	18	)216		CLC	
26B1:	65 9C	)217		ADC	TEMPC
26B3:	85 96	)218		STA	TEMP6
26B5:	D0 DC	)219		BNE	KIII3
26B7:	A5 96	)220	KIII4	LDA	TEMP6 ; VERWERK DE AFSTAND
26B9:	F0 29	)221		BEQ	KIII5 ; GEEN PIONNEN
26BB:	85 97	)222		STA	TEMP7
26BD:	84 98	)223		STY	TEMP8 ; BEREKEN QUOTIENT TOTALE AFSTAND/PIONNEN
26BF:	A5 A3	)224		LDA	STKWRDKNAZ
26C1:	29 0F	)225		AND	#\$0F
26C3:	85 99	)226		STA	TEMP9
26C5:	A5 A5	)227		LDA	STKNUMKNAZ
26C7:	29 0F	)228		AND	#\$0F
26C9:	18	)229		CLC	
26CA:	65 99	)230		ADC	TEMP9
26CC:	85 99	)231		STA	TEMP9
26CE:	84 9A	)232		STY	TEMPA
26D0:	20 7A 20	)233		JSR	DIVIDE
26D3:	A5 9B	)234		LDA	TEMPB
26D5:	38	)235		SEC	
26D6:	E9 06	)236		SBC	#6 ; RESULTAAT - 6
26D8:	85 99	)237		STA	TEMP9
26DA:	A9 B2	)238		LDA	#\$B2 ; WEEGFAKTOR IS -78
26DC:	85 9A	)239		STA	TEMPA
26DE:	20 25 20	)240		JSR	MULTIPLY
26E1:	20 03 20	)241		JSR	ADD
26E4:	A5 AD	)242	KIII5	LDA	POSITIO ; AFSTAND KONING - CENTRUM
26E6:	85 99	)243		STA	TEMP9
26E8:	A9 44	)244		LDA	#\$44 ; CENTRUM = 44
26EA:	85 9A	)245		STA	TEMPA
26EC:	20 43 21	)246		JSR	AFSTAND
26EF:	85 98	)247		STA	TEMP8
26F1:	A9 55	)248		LDA	#\$55 ; CENTRUM = 55
26F3:	85 9A	)249		STA	TEMPA
26F5:	20 43 21	)250		JSR	AFSTAND
26F8:	18	)251		CLC	
26F9:	65 98	)252		ADC	TEMP8
26FB:	85 99	)253		STA	TEMP9
26FD:	A9 F0	)254		LDA	#\$F0 ; WEEGFAKTOR IS -16
26FF:	85 9A	)255		STA	TEMPA
2701:	20 25 20	)256		JSR	MULTIPLY
2704:	20 03 20	)257		JSR	ADD
2707:	4C 7B 26	)258	KIIOUT	JMP	STARTKII
270A:	A5 A0	)259	INIT	LDA	KAZ
270C:	85 A1	)260		STA	KWIS ; KLEURWISSEL = KLEUR
270E:	A0 00	)261		LDY	#0 ; Y=00 GEDURENDE HELE PROGRAMMA
2710:	A2 07	)262		LDX	#7 ; RESET ALLE TELLEN
2712:	94 A2	)263	IIA	STY	STKWRDKAZ,X
2714:	CA	)264		DEX	
2715:	10 FB	)265		BPL	IIA
2717:	84 AF	)266		STY	DAMKNAZ ; RESET AF
2719:	A9 40	)267		LDA	#\$40
271B:	85 B0	)268		STA	VANVELDHEX

271D: A6 B0	>270	I3	LDX	VANVELDHEX ; SCAN BORD
271F: CA	>271		DEX	
2720: 30 67	>272		BMI	OUTINIT ; FF = ONGELDIG VELD
2722: 86 B0	>273		STX	VANVELDHEX ; HAAL VELD
2724: B5 16	>274		LDA	BORD, X
2726: F0 F5	>275		BEQ	I3 ; VEELD = LEEG
2728: 85 AE	>276		STA	SLASTUK
272A: 29 40	>277		AND	##40
272C: 45 A1	>278		EOR	KWIS
272E: D0 1F	>279		BNE	I5 ; STUK KNAZ
2730: A5 AE	>280	I4	LDA	SLASTUK ; STUK KAZ
2732: 29 07	>281		AND	#7
2734: 48	>282		PHA	
2735: AA	>283		TAX	; TEL STUKWAARDE IN A2
2736: B5 C9	>284		LDA	STUKTAB, X ; C9-CF = TABEL
2738: 18	>285		CLC	
2739: 65 A2	>286		ADC	STKWRDKAZ
273B: 85 A2	>287		STA	STKWRDKAZ
273D: 68	>288		PLA	
273E: C9 02	>289		CMP	#2 ; PION ?
2740: 10 04	>290		BPL	I4A
2742: E6 A3	>291		INC	STKWRDKNAZ ; TEL AANTAL PIONNEN IN A3
2744: D0 D7	>292		BNE	I3
2746: A5 A3	>293	I4A	LDA	STKWRDKNAZ ; TEL AANTAL STUKKEN IN A3
2748: 18	>294		CLC	
2749: 69 10	>295		ADC	##10
274B: 85 A3	>296		STA	STKWRDKNAZ
274D: D0 CE	>297		BNE	I3
274F: A5 AE	>298	I5	LDA	SLASTUK ; STUK KNAZ
2751: 29 07	>299		AND	#7
2753: 48	>300		PHA	
2754: AA	>301		TAX	; TEL STUKWAARDE IN A4
2755: B5 C9	>302		LDA	STUKTAB, X ; C9-CF = TABEL
2757: 18	>303		CLC	
2758: 65 A4	>304		ADC	STKNUMKAZ
275A: 85 A4	>305		STA	STKNUMKAZ
275C: 68	>306		PLA	
275D: C9 02	>307		CMP	#2 ; PION ?
275F: 10 04	>308		BPL	I5A
2761: E6 A5	>309		INC	STKNUMKNAZ ; TEL AANTAL PIONNEN IN A5
2763: D0 B8	>310		BNE	I3
2765: A5 A5	>311	I5A	LDA	STKNUMKNAZ ; TEL AANTAL STUKKEN IN A5
2767: 18	>312		CLC	
2768: 69 10	>313		ADC	##10
276A: 85 A5	>314		STA	STKNUMKNAZ
276C: A5 AE	>315	I6	LDA	SLASTUK ; KONING KNAZ
276E: 29 07	>316		AND	#7
2770: C9 02	>317		CMP	#2 ; STUK KONING ?
2772: D0 0B	>318		BNE	I7
2774: A5 B0	>319		LDA	VANVELDHEX ; BEWAAR POSITIE KONING KNAZ
2776: 85 B1	>320		STA	VERTREKHEX ; WANT DIE IS VAAK NODIG
2778: 20 FD	>321	20	JSR	KODEER
277B: 85 AB	>322		STA	KONKNAZ
277D: D0 9E	>323		BNE	I3
277F: C9 06	>324	I7	CMP	#6 ; DAM KNAZ
2781: D0 9A	>325		BNE	I3
2783: A9 02	>326		LDA	#2
2785: 85 AF	>327		STA	DAMKNAZ ; AF=2 : INDIEN DAME AANWEZIG
2787: D0 94	>328		BNE	I3 ; ANDERS 0
2789: 60	>329		RTS	
278A: 48	>330		PHA	; SAVE AKKU
278B: A9 36	>331		LDA	##36
278D: 8D B0	>332	1A	STA	PAD ; STUUR DISPLAYS
2790: A9 12	>333		LDA	##12
2792: 8D B2	>334	1A	STA	PBD
2795: 20 0A	>335	SI	JSR	INIT ; DOE HOOFDSUBS BEHALVE DE KONING
2798: 20 2A	>336	22	JSR	PAARD
279B: 20 A5	>337	22	JSR	LOPER
279E: 20 1B	>338	23	JSR	DAME
27A1: 20 90	>339	23	JSR	TOREN
27A4: 20 71	>340	24	JSR	PION
27A7: A5 AF	>341		LDA	DAMKNAZ ; DOE KONINGII ALS AF (<) 0 EN ALS
27A9: D0 0C	>342		BNE	SIA ; ALS STUKWAARDE KNAZ < 70: ANDERS KI

```

27AB: A5 A4      >344      LDA STKNUMKAZ
27AD: C9 46      >345      CMP ##46
27AF: 10 06      >346      BPL SIA
27B1: 20 6F 26   >347      JSR KONINGII
27B4: 98         >348      TYA
27B5: F0 03      >349      BEQ S2
27B7: 20 85 25   >350      JSR KONINGI
27BA: A5 A6      >351      LDA STELWRDLOB ; ONDERZOEK OF STELLINGWAARDE =
S2                                CMP POSWRDLOB ; DE TOT DAN TOE BESTE ZET
27BC: C5 BE      >352      BNE S3
27BE: D0 0D      >353      LDA STELWRDHOB ; MAAK IN DAT GEVAL PZET
27C0: A5 A7      >354      CMP POSWRDHOB ; GELIJK AAN OZET
27C2: C5 BF      >355      BNE S3
27C4: D0 07      >356      STA PZET
27C6: 85 0F      >357      STA OZET
27C8: 85 91      >358      TYA
27CA: 98         >359      BEQ S6
27CB: F0 2B      >360      LDA STELWRDLOB ; GELIJKE WAARDE
27CD: A5 A6      >361      EOR ##FF ; VERGELIJK STELLINGWAARDE MET DE
S3                                ; TOT DAN TOE BESTE WAARDE
27CF: 49 FF      >362      CLC
27D1: 18         >363      ADC POSWRDLOB ; TEL 1-KOMPLEMENT VAN STELLINGWAARDE
27D2: 65 BE      >364      LDA STELWRDHOB ; BIJ WAARDE TOT DAN TOE BESTE ZET
27D4: A5 A7      >365      EOR ##FF
27D6: 49 FF      >366      ADC POSWRDHOB
27D8: 65 BF      >367      BVS S4A ; BESLIS DE BESTE ZET
27DA: 70 04      >368      BPL S5A
27DC: 10 14      >369      BMI S5
27DE: 30 02      >370      BMI S5A
27E0: 30 10      >371      LDA STELWRDHOB ; ONTHOUD GROOTSTE WAARDE EN
27E2: A5 A7      >372      STA POSWRDHOB ; MANIPULEER N-ZET EN O-ZET ZODANIG
27E4: 85 BF      >373      LDA STELWRDLOB ; DAT SCHAAKPROGRAMMA DE BESTE ZET
27E6: A5 A6      >374      STA POSWRDLOB ; VOLGENS POSVAL DUET
27E8: 85 BE      >375      STY PZET
27EA: 84 0F      >376      LDA ZETII ; PZET=0 : OZET=1
27EC: A5 01      >377      STA OZET ; OF OZET=0 : PZET=1
27EE: 85 91      >378      BNE S6
27F0: D0 06      >379      STY OZET
27F2: 84 91      >380      LDA ZETII
27F4: A5 01      >381      STA PZET
27F6: 85 0F      >382      STY PBD ; SCHAKEL DISPLAYS UIT
27F8: 8C 82 1A   >383      PLA ; SLOTRoutine
27FB: 68         >384      LDX PZET
27FC: A6 0F      >385      CPX OZET
27FE: E4 91      >386      BNE OUTALL
2800: D0 03      >387      CMP TIMER
2802: CD F4 1A   >388      JMP BIF
2805: 4C 2F 2B   >389      OUTALL

```

Letter to the Editor

Dear Willem,

I've read your call for more papers to be written in English in the April edition of DE 6502 KENNER. Although I cannot agree with your remark on Dutch sounding like Chinese to some of us, I'm nevertheless strongly in favour of using English as the preferred means of communication between the members of our club. In Mathematics, German authors are usually requested by the editors of German scientific journals to submit their papers in English. The argument goes as follows: Nothing will be lost (every German scientist has a certain command of the English language) and much can be gained (only think of potential readers from China, Japan, or the Arabic Countries). I cannot see why the same argument should not apply to the activities of our club. I'm sure the English-speaking members of our club will appreciate the double effort the rest of us have to spend with our publications: We first have to find out how to say what we want to say, and then find out how to say it in English. They will certainly forgive us for occasionally maltreating the beloved language of Shakespeare, Wilde, and Shaw.

Fred Behringer, München


```

1 REM "Ø D I N S U P E R T E A M"
2 REM BY GERARD KEET
3 REM RØDENBURG NØ.3
4 REM 1965BL HEEMSKERK
5 REM TEL.02510-39763
7 PRINT CHR$(12);
8 DIM B(35,2)
10 DIM A$(35), A(35,6)
11 INPUT "STANDEN INLEZEN? <J/N>: ";Z$
12 IF Z$<>"J" THEN 100
20 PRINT CHR$(12);PRINT:PRINT
23 PRINT " DE ØDIN-SUPERTEAM-STAND WØRDT NU DØØR HET"
24 PRINT " PRØGRAMMA GEMAAKT ØP BASIS VAN DE"
25 PRINT " KØMPETITIESTANDEN, DIE EERST INGELEZEN"
26 PRINT " WØRDEN."
27 PRINT " AL MET AL DUURT HET ZØ'N 5 MINUTEN"
28 PRINT " VØØRDAT DE STAND ØP HET SCHERM VERSCHIJNT."
29 PRINT " RUSTIG AFWACHTEN DUS MAAR."
32 FØR I=1 TØ 9:PRINT "*";CHR$(21);CHR$(26);:NEXT
33 PRINT "*****";
34 FØR I=1 TØ 9:PRINT CHR$(21);CHR$(10);"*";:NEXT:PRINT
35 PRINT "*****"
100 DATA "DS","DAMESSENIØREN","HS","HERENSENIØREN","DJ"
110 DATA "DAMESJUNIØREN","HJ","HERENJUNIØREN","MA"
120 DATA "MEISJESADSPIRANTEN","JA","JØNGENSADSPIRANTEN"
130 DATA "MP","MEISJESPUILLLEN","JP","JØNGENSPUILLLEN","MW"
140 DATA "MEISJESWELFEN","JW","JØNGENSWELFEN"
145 DATA "DV","DAMESVETERANEN"
200 TT=1:TC=0:TP=0:TS=0:KW=24576
210 IF Z$<>"J" THEN 520
300 PØKE 6777,1
310 X=USR(&"ØB02",0)
320 X=USR(&"14BC",0)
520 IF PEEK(KW)=48 THEN 800
530 CAT$=CHR$(PEEK(KW))+CHR$(PEEK(KW+1))
540 TW=KW+5
550 TEAM$=""
560 FØR CW=TW TØ TW+3
570 TEAM$=TEAM$+CHR$(PEEK(CW)):NEXT CW
580 IF TEAM$="ØDIN" THEN 600
590 TW=TW+22:IF TW<KW+5+12*22 THEN 550
595 KW=KW+5+12*22:GØTØ 520
600 NR$=CHR$(PEEK(TW+5)):IF NR$<>" " THEN 620
610 NR$="1"
620 READ HCAT$
630 IF HCAT$<>CAT$ THEN 620
640 READ A$(TT):A$(TT)=A$(TT)+" "+NR$
650 G$=CHR$(PEEK(TW+12))+CHR$(PEEK(TW+13))
660 P$=CHR$(PEEK(TW+14))+CHR$(PEEK(TW+15))
670 V$=CHR$(PEEK(TW+16))+CHR$(PEEK(TW+17))+CHR$(PEEK(TW+18))
680 T$=CHR$(PEEK(TW+19))+CHR$(PEEK(TW+20))+CHR$(PEEK(TW+21))
690 A$(TT,1)=VAL(G$):A$(TT,2)=VAL(P$):A$(TT,3)=VAL(V$):A$(TT,4)=VAL(T$)
692 IF A$(TT,1)<>0 THEN 700
694 A$(TT,5)=INT(A$(TT,2)*100)/100
696 A$(TT,6)=INT((A$(TT,3)-A$(TT,4))*100)/100
698 GØTØ 720

```

```

700 A(TT,5)=INT((A(TT,2)*100/A(TT,1))/100
710 A(TT,6)=INT((A(TT,3)-A(TT,4))*100/A(TT,1))/100
720 TG=TG+A(TT,1):TP=TP+A(TT,2):TS=TS+A(TT,3)-A(TT,4)
730 TT=TT+1:RESTORE
740 GOTO 590
800 A(TT,5)=INT(TP*100/TG)/100
810 A(TT,6)=INT(TS*100/TG)/100
900 FOR I=1 TO TT-2
910 FOR J=I+1 TO TT-1
920 IF A(I,5)>A(J,5) THEN 1000
930 IF A(I,5)<A(J,5) THEN 970
940 IF A(I,6)>=A(J,6) THEN 1000
970 HAS=A$(I):A$(I)=A$(J):A$(J)=HAS
980 H=A(I,5):A(I,5)=A(J,5):A(J,5)=H
990 H=A(I,6):A(I,6)=A(J,6):A(J,6)=H
1000 NEXT J
1010 NEXT I
1020 GOSUB 1100:REM PRINT
1030 GOSUB 4010:REM VUL AAN
1040 PRINT CHR$(12):CHR$(18);
1050 GOSUB 3010:REM PRINT UITLEG
1060 GOSUB 1100:REM PRINT DEFINITIEF
1090 END
1100 PRINT CHR$(18)
1120 PRINT "PLAATS";TAB(11)"TEAM";TAB(36)"PUNTEN";TAB(56)"SAL DO"
1121 PRINT
1130 FOR I=1 TO TT-1
1132 X$=STR$(A(I,5)):Y$=STR$(A(I,6))
1133 U$=STR$(B(I,0)):V$=STR$(B(I,1)):W$=STR$(B(I,2))
1135 GOSUB 1510
1140 PRINT I;"":U$;"":TAB(11)A$(I);TAB(35)X$;
1141 PRINT TAB(45)"(":RIGHT$(V$,4);"":TAB(54)Y$;TAB(63)"(":W$;"":
1150 NEXT I:PRINT
1170 X$=STR$(A(TT,5)):Y$=STR$(A(TT,6))
1171 V$=STR$(B(TT,1)):W$=STR$(B(TT,2))
1180 GOSUB 1510
1240 PRINT "VERENIGINGSTOTAAL";TAB(35);X$;TAB(45)"(":RIGHT$(V$,4);
1241 PRINT "":TAB(54)Y$;TAB(63)"(":W$;"":
1245 PRINT:PRINT
1246 PRINT SPC(71);
1250 PRINT CHR$(20)
1260 RETURN
1500 REM EDIT GETALLEN NAAR XX.XX
1510 IF I>9 THEN 1530
1520 PRINT " ";
1530 Z$=V$
1531 GOSUB 1610
1532 V$=Z$
1540 Z$=W$
1541 GOSUB 1610
1542 W$=Z$
1550 Z$=X$
1551 GOSUB 1610
1552 X$=Z$
1560 Z$=Y$
1561 GOSUB 1610

```

```

1562 Y$=Z$
1590 RETURN
1600 REM EDIT Z$
1610 IF LEFT$(Z$,1)="-" THEN 1615
1612 Z$=MID$(Z$,2)
1615 IF LEN(Z$)=6 THEN 2000
1620 IF LEN(Z$)=5 THEN 1990
1630 IF LEN(Z$)<4 THEN 1700
1640 IF LEFT$(Z$,1)="-" THEN 1660
1650 Z$=" "+Z$:GOTO 1990
1660 IF MID$(Z$,2,1)="-" THEN 1680
1670 Z$=Z$+"O":GOTO 1990
1680 Z$="-O"+RIGHT$(Z$,3):GOTO 1990
1700 IF LEN(Z$)<3 THEN 1800
1710 IF LEFT$(Z$,1)<>"-" THEN 1730
1720 Z$="-O"+RIGHT$(Z$,2)+"O":GOTO 1990
1730 IF LEFT$(Z$,1)<>"." THEN 1750
1740 Z$="O"+Z$:GOTO 1990
1750 Z$=" "+Z$+"O":GOTO 1990
1800 IF LEN(Z$)=1 THEN 1900
1810 IF LEFT$(Z$,1)="-" THEN 1830
1820 Z$=Z$+".00":GOTO 1990
1830 Z$="O"+Z$+"O":GOTO 1990
1900 Z$=" "+Z$+".00"
1990 Z$=" "+Z$
2000 RETURN
3000 REM PRINTEN UITLEG
3010 PRINT "*****"
3011 PRINT "*"
3012 PRINT "*      Ø D I N   S U P E R T E A M      *"
3013 PRINT "*"
3014 PRINT "*****"
3015 PRINT:PRINT
3020 PRINT "MET DE SUPERTEAMSTAND KUNNEN DE PRESTATIES VAN DE TEAMS"
3030 PRINT "VAN ØNZE VERENIGING ØNDERLING VERGELEKEN WØRDEN."
3040 PRINT "DE CIJFERS KØMEN ALS VØLGT TØT STAND:"
3050 PRINT "HET TØTAAL AANTAL WEDSTRIJDPUNTEN EN HET DØELSALDØ"
3060 PRINT "WØRDEN GEDEELD DØØR HET AANTAL GESPEELDE WEDSTRIJDEN."
3070 PRINT "DEZE GETALLEN ZIE JE IN DE TWEE KØLØMMEN."
3080 PRINT "DE PUNTEN ZIJN BELANGRIJKER DAN HET SALDØ, NET ALS IN DE"
3090 PRINT "KØMPETITIE ZELF."
3100 PRINT "VØØRBEELD: EEN TEAM HEEFT IN ZES WEDSTRIJDEN NEGEN PUNTEN"
3110 PRINT "GEHAALD EN EEN DØELSALDØ VAN 21 PØSITIEF."
3120 PRINT "HET GEMIDDELDE AANTAL WEDSTRIJDPUNTEN (KØLØM 1) IS DAN"
3130 PRINT "9:6=1.50, HET DØELSALDØ (KØLØM 3) IS 21:6=3.50":PRINT
3140 PRINT "TUSSEN HAAKJES DE VØRIGE STAND."
3150 PRINT
3190 RETURN
4000 REM INPUT ØUDE STAND
4010 FOR I=1 TO TT-1
4020 PRINT "TEAM "I":INPUT "<PLTS,PNT,SALDØ>: ";B(I,0),B(I,1),B(I,2)
4030 NEXT I
4090 RETURN

```

DE 6502 KENNER

6502-Tracer for the MON/DOS65 computer.

Author: Rene Hettfleisch, The Netherlands.
System: MON/DOS65 computer

While programming in the assembly-language, a tracer is a handy expedient to trace errors in a program. For this I use the '6502-tracer' of J. Ruppert (Elektor Holland nr. 244, Feb. 1984, page 2-66, 2-67), which I adjusted to MON/DOS65. The working of the tracer will be explained in this article. The original program is given in a hexdump in the article mentioned above. By this the following memory-locations have to be changed:

051C : old \$7E, new \$FE (change of interrupt-vector)
051D : old \$1A, new \$CE
0521 : old \$7F, new \$FF
0522 : old \$1A, new \$CE

06A1 : old \$8F, new \$27 (change of PRBYT)
06A2 : old \$12, new \$E0

06A6 : old \$34, new \$00 (change of PRCHA)
06A7 : old \$13, new \$E0

Furthermore there will be a little program added before and behind the original program; refer the source-listing. The program starts at \$046B and ends at \$072E. The program can be used as a utility by saving it on the system-disk and next you have to give the file the "command-mode" by using the SETMODE-RMDC filename.

After starting the tracer the program will ask the start-address to be traced. Before doing this the program to be traced has to be loaded into the memory. Watch the overlap of the tracerprogram by this. When the startaddress is entered the tracing will be started. It would be wise to send the output to the printer (output-device 2 and 3), because the tracing can't be interrupted in the meantime, after which the tracing can continue. The tracing can be stopped finally by depressing a random key (this causes an interruption). When at the end of the program which had to be traced, a RTS (\$60) will be set, the tracing will be stopped here.

After the tracing has been stopped, a RESET has to be given, after which DOS65 has to be restarted. The reason for this is that the IRQ-pointer will be detoured and I won't be able to get it back normally, so the computer is blocked. However, you can live with this.

Good luck with the tracer!

046B TRACER ORG \$046B

TEMPORARIES AND BUFFERS

ED 00 START * \$00ED
FE CE IRQ * \$CEFE
69 04 IRGSAV * \$0469
0E C1 VAIER * \$C10E

MON65 SUBROUTINES

0F E0 STRING * \$E00F
0C E0 INKEY * \$E00C
09 E0 OUT * \$E000
48 E0 CONVRT * \$E048

046B 20 0F E0 BEGIN JSR STRING
046E 1B = \$1B
046F 69 = 'i
0470 0A = \$0A

0471 20	=	'
0472 36	=	'6
0473 35	=	'5
0474 30	=	'0
0475 32	=	'2
0476 2D	=	'-
0477 74	=	't
0478 72	=	'r
0479 61	=	'a
047A 63	=	'c
047B 65	=	'e
047C 72	=	'r
047D 20	=	'
047E 1B	=	\$1B
047F 6E	=	'n
0480 0D	=	\$0D
0481 0A	=	\$0A
0482 0A	=	\$0A
0483 47	=	'6
0484 69	=	'i
0485 76	=	'v
0486 65	=	'e
0487 20	=	'
0488 73	=	's
0489 74	=	't
048A 61	=	'a
048B 72	=	'r
048C 74	=	't
048D 61	=	'a
048E 64	=	'd
048F 72	=	'r
0490 65	=	'e
0491 73	=	's
0492 73	=	's
0493 3A	=	';
0494 20	=	'
0495 00	=	\$00
0496 A9 00	LDIM	\$00
0498 85 ED	STA	START
049A 85 EE	STA	START +01
049C 20 0C E0	GETCHR JSR	INKEY GET CHARACTER FROM
049F AA	TAX	KEYBOARD
04A0 C9 0D	CMPIM	\$0D IF RETURN
04A2 D0 0E	BNE	BTCHRA
04A4 20 00 E0	JSR	OUT PRINT CR
04A7 A9 0A	LDIM	\$0A
04A9 20 00 E0	JSR	OUT PRINT LF TWO TIMES
04AC 20 00 E0	JSR	OUT
04AF 4C E6 04	JMP	BEGIN
04B2 C9 7F	BTCHRA CMPIM	\$7F IF DELETE
04B4 F0 1C	BEQ	DELETE GO TO DELETE
04B6 20 48 E0	JSR	CONVRT CONVERT ASCII TO
04B9 30 E1	BMI	GETCHR BINARY
04BB A8	TAX	IF VALID CHARACTER
04BC 8A	TXA	
04BD 20 00 E0	JSR	OUT PRINT CHARACTER
04C0 98	TYA	
04C1 0A	ASLA	STORE CHAR IN START
04C2 0A	ASLA	
04C3 0A	ASLA	
04C4 0A	ASLA	
04C5 A0 04	LDYIM	\$04
04C7 0A	LOOPA ASLA	
04C8 26 ED	ROL	START
04CA 26 EE	ROL	START +01
04CC 88	DEY	
04CD D0 F8	BNE	LOOPA
04CF 4C 9C 04	JMP	GETCHR

```
04D2 20 0F EO DELETE JSR STRING REMOVE CHAR FROM
04D5 08 = $08 SCREEN
04D6 20 =
04D7 08 = $08
04D8 00 = $00
04D9 A0 04 LDYIM $04
```

```
04DB 46 EE LOOPB LSR START +01 ERASE CHARACTER
04DD 66 ED ROR START
04DF 6A RORA
04E0 88 DEY
04E1 D0 F8 BNE LOOPB
04E3 4C 9C 04 JMP GETCHR
```

```
04E6 AE FE CE BEGINA LDX IRQ SAVE IRQ-VECTOR
04E9 AC FF CE LDY IRQ +01
04EC 8E 69 04 STX IRGSAV
04EF 8C 6A 04 STY IRGSAV +01
04F2 A9 23 LDAIM END PUSH END ON STACK
04F4 48 PHA
04F5 A9 07 LDAIM END /256
04F7 48 PHA
04F8 AD 0E C1 LDA VAIER STOP CLOCK
04FB 29 7F ANDIM $7F
04FD 8D 0E C1 STA VAIER
```

```
0723 AE 69 04 END LDX IRGSAV GET ORIGINAL IRQ-VECT
0726 AC 6A 04 LDY IRGSAV +01
0729 8E FE CE STX IRQ
072C BC FF CE STY IRQ +01
```

Use of cursor-control-keys ED.

Author: Rene Hettfleisch, The Netherlands
System: MON/DOS65 computer

The Full Screen Editor (ED) for DOS65 is an extended and good editor. I think many MON/DOS65 users will agree with me. In spite of it I think there is one disadvantage by the ED: the cursorcontrolling. My keyboard namely has cursor-control-keys which generates the usually codes:

```
BackSpace : $08
Horizontal Tab : $09
Line Feed : $0A
Vertical Tab : $0B
```

By ED the following keys have to be used for the cursor-control: ^S, ^D, ^X and ^E. I think it is awkward, also because my keyboard doesn't repeat the control-characters.

The solution I found is as follows: the keyboard-interrupt-routine of DOS65 (which loads the characters of the keyboard), will be detoured by a little program which changes the cursor-control-characters \$08, \$09, \$0A and \$0B into ^S, ^D, ^X and ^E resp. ED now recognizes the character of the cursor-control-keys really as cursor-control-keys.

The little program you need for this, is as follows:

```
0E00 AD 11 C1 LDA $C111
0E03 29 FF ANDIM $FF
0E05 C9 08 CMPIM $08
0E07 D0 05 BNE $0E0E
0E09 A9 13 LDAIM $13
0E0B 4C 26 0E JMP $0E26
0E0E C9 09 CMPIM $09
0E10 D0 05 BNE $0E17
```

```
0E12 A9 04 LDAIM $04
0E14 4C 26 0E JMP $0E26
0E17 C9 0A CMPIM $0A
0E19 D0 05 BNE $0E20
0E1B A9 18 LDAIM $18
0E1D 4C 26 0E JMP $0E26
0E20 C9 08 CMPIM $08
0E22 D0 02 BNE $0E26
0E24 A9 05 LDAIM $05
0E26 4C 48 AB JMP $AB48
```

```
0E29 A2 00 LDXIM $00
0E2B A0 0E LDYIM $0E
0E2D 8E 3B AB STX $AB3B
0E30 8C 3C AB STY $AB3C
0E33 60 RTS
```

```
0E34 A2 43 LDXIM $43
0E36 A0 AB LDYIM $AB
0E38 4C 2D 0E JMP $0E2D
```

The conversion-program uses the addresses \$0E00-\$0E28. At \$0E29 starts the program that detoured the pointer in the interrupt-program at \$0E00. On \$0E34 starts the program that sets the pointer in the original position.

To adjust ED the following steps must be proceeded:

1. Rename the editor after itself and in such a way that this only can be called by the command EDITOR and not by the command ED. This goes as follows (refer page 24, 25 DOS65 manual):

```
RENAME EDITOR DUMMY
RENAME DUMMY EDITOR
```

2. Enter the program from \$0E00 and save it as ED1 with startaddress \$0E29. Give the file the command-mode (SETMODE -RWDC ED1).

3. Open a file named as ED in the following way:

```
CREATE ED
ED1
EDITOR
GO 0E34
^D
```

Now the editor can be called with the command ED. The disadvantage of this method is that after the command ED has been given a filename can't be entered at once. The editor first comes in the command-mode, after which a file has to be loaded. The advantage of making use of the cursor-control-keys is much more than the disadvantage.

The English paperware-service contains for instance:

- DATBAS; database for Elektor's JUNIOR with OHIO-DOS. Basic. Jan van Heuven, The Netherlands. Transl: F. Lopes, Portugal.
- VDU ROTATING NEWSPAPER. For JUNIOR with VDU. Machinecode. Ivo van Rijssel, The Netherlands. Transl: F. Bens, Holland.
- Centronics Printer Interface; device 4 or 5 on Commodore 64. Ruud Uphoff, The Netherlands. Machinecode.
- The MC68000 microprocessor; a new processor in our club. Gert van Opbroek, The Netherlands. Article.
- Disassembler for 65C02 (Rockwell version). Machinecode. Nico de Vries, The Netherlands.
- Disassembler for 65C02 (Synertek/GTE version). Nico de Vries, The Netherlands. Machinecode.

DE 6502 KENNER

Octopus disks

=====

Author : W. van Asperen
System : Octopus 65 with OHIO DOS

=====

After installation of a second disk drive on my Octopus 65 (this one now wired as drive A, and the original drive as drive B), it was impossible to boot the system on most disks. I was a novice in the personal computer area, and invoked assistance from more experienced Octopus users. I was advised that the problem could arise from the head step rate and the delay step rate as read from track of a disk on start up of the system. It took me a lot of trials to arrive at the values as now used on all disks (except the source disks #6,7,8,9, and the FORTH disk #10). By exchanging the two drives I could test whether the system could be booted with the new track 0.

The values as found for my drives (Mitsubishi M4853-342 used with double step-rate, so 40 tracks) are

address 26A3 03 step rate
address 26A5 D2 delay step rate

The track-to-track access time specified for this drive is 6 milliseconds. At 1 MHz clock frequency the value on 26A3 of 03 represents a step rate of 6 ms. For a 2 MHz system, this values would have to be 06 for 6 ms.

For other beginning users, please find the applicable instructions below. You will need at least one disk / drive combination that is able to boot your system. This disk must contain the COP/TO file (on track 36), because you need so called Track-zero read/write utility.

- Boot up the system.
- exit from the BEXEC* program by hitting the return key (further called <CR>).
- jump to DOS by entering EXIT and <CR> (system responds with A*)
- enter CA 0200=36,1 <CR> (to load COP/TO i.e. the track 0 read write utility).
- enter GO 0200 <CR> (start the utility)
- enter 2 <CR> (select choice 2 from the menu of the utility)
- enter R6200 <CR> (read track 0 into memory starting at address 6200)
- enter E <CR> (exit the utility)
- enter RE M (jump to the monitor program)
- enter 66A3 and a space (display contents of address 66A3)
- enter the required value and a slash i.e. 03/ (change step rate)
- enter 66A5 and a space (display contents of address 66A5)
- enter the required value and a slash i.e. D2/ (change delay step rate)
- enter .OS65D (go back to DOS)
- enter GO 0200 <CR> (start track 0 R/W utility again)
- enter 2 <CR> (select choice 2 from the menu)
- enter W6200/2200,8 <CR> (write the changed track 0 to disk)

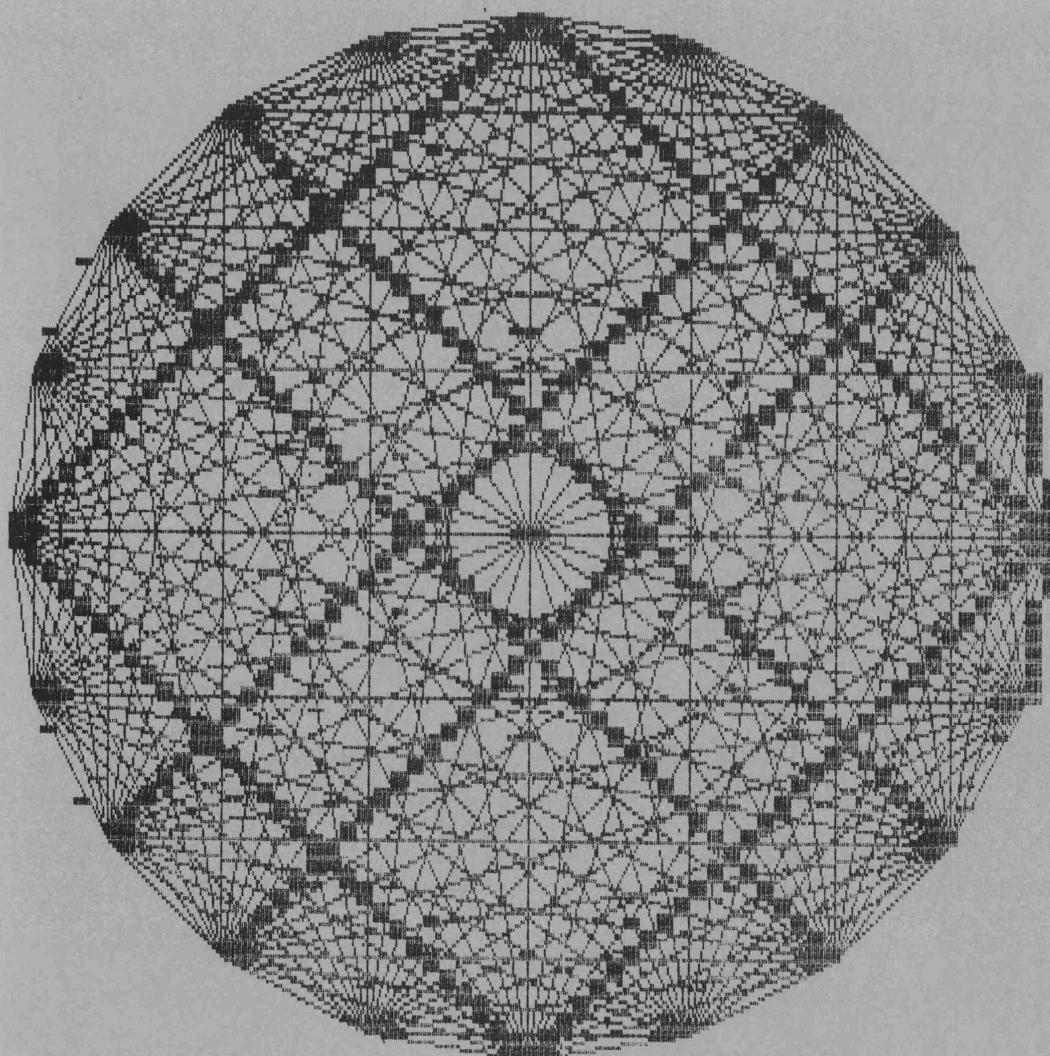
Now you have created a disk that may be able to boot up your system. Please note that the required values for address 26A3 and 26A5 may be different for each drive (refer to the manual of the drive).

It appears to be very usefull first to make a back-up copy of a disk, before you start changing the contents as described before.

I got another hint to solve a printer hang i.e. the printer stops printing before reaching the end of a file. This problem can be solved by changing address 25A9 from 48 to 60 in the same way as described before.

Please note that the MSB of each address is 6 i.s.o 2 when using this method, due to the fact that data loaded into memory from address 2200 at boot up, are now loaded from 6200.

DE 6502 KENNER



Leif Rasmussen
4534 Hørve



Parkvej 1
DK-Danmark