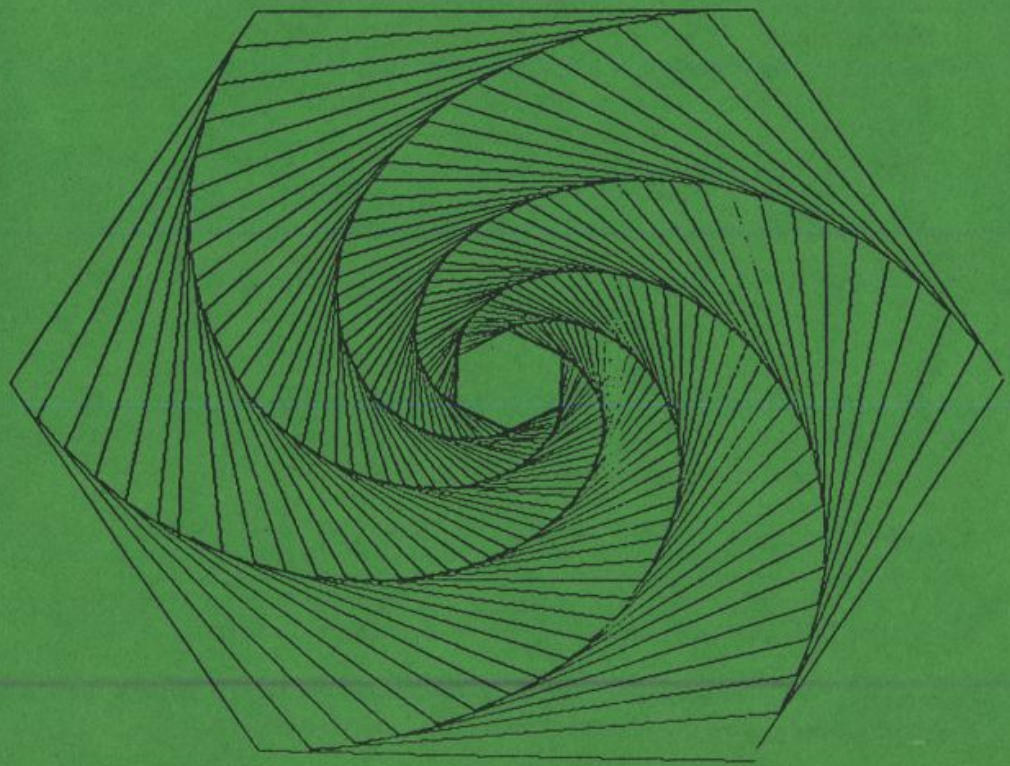
BULLETIN

SINCLAIR GEBRUIKERSGROEP GRONINGEN/ASSEN



7e jaargang nummer 5 januari '90

BULLETIN SGG



COLOFON

VOORZITTER:
Jan Dirk Burggraaf
Kluivingskampenweg 30
9761 BP Eelde
☎ 05907-1697

SEKRETARIS:
Martin den Hollander
Numero Dertien 8
9644 TV Veendam
☎ 05978-45474

PENNINGMEESTER/
VERHUUR:
Jan Arends
Heiligelaan 66
9636 CP Zuidbroek
☎ 05985-2247
Giro 5965342 t.n.v.
rekening SGG

VICE VOORZITTER/
MATERIAALCOMM
PR:
J. van Alteren
De Grouw 6
9351 LP Leek
☎ 05945-15678

REPARATIE:
C. van Krimpen
Koldakker 34
9407 BM Assen
☎ 05920-70093

ALGEMEEN:
Roelof Koning
Selwerderstraat 26
9717 GK Groningen
☎ 050-124298

REDAKTIE:
Mevr. F. Elstrodt
Kam. Onnesstraat 172
9727 HS Groningen
☎ 050-263930

Rudy Biesma
Betuwe 18
9405 JJ Assen
☎ 05920-50643

Het SGG-bulletin is een uitgave van de Sinclair Gebruikersgroep Groningen. Het bulletin verschijnt 10 keer per jaar.

Artikelen, listings of andere inzendingen zijn voor verantwoording van de inzender.

De sluitingsdatum voor kopij wordt in elk bulletin vermeld.

Overname van artikelen, illustraties en andere publikaties uitsluitend toegestaan met toestemming van de redaktie.

Het lidmaatschap van onze gebruikersgroep bedraagt f 17,50 per kalenderjaar voor personen tot en met 17 jaar voor oudere personen is dit f 25,00 per kalenderjaar. Bij deze prijs is het abonnement op het bulletin inbegrepen.

U kunt lid worden van de SGG door U op te geven bij de penningmeester.



VAN DE REDAKTIE



HALLO ALLEMAAL

Voor u ligt alweer een nieuw bulletin, dat de voorplaat weer hetzelfde is, komt niet doordat wij geen nieuwe ideeën hebben.
Als u op tijd de acceptgiro betaalt kunnen we volgende keer weer een andere voorplaat hebben. Bij de PTT moeten n.l. alle te verzenden bulletins gelijk zijn. Bij niet genoeg leden om in aanmerking te komen voor het goedkope tarief, moeten wij dus een oplossing zoeken. Vandaar dus.

In Duo screen (dec. bul) is een foutje gezien.
Op blz 13 staat POINT EQU #2E, dit moet zijn POINT EQU #22CE.

Verleden maand was het thema SCREENS, deze maand gaat alles sneller dan snel.
Frans geeft u de controle over een pijl, die u dus over het scherm, dus ook over screens kan sturen.
Roelof laat alles nog sneller van de Opus drive laden, snel dus en natuurlijk verpakt in een mooi menuutje.
Verder kan het nog wat bijzonders, gauw lezen dus.
En met de aanpassingen en veranderingen van de heer Grunefeld is de redaktie in haar nopjes, zo werkt het veel sneller en hou ik tijd over om iets te tekenen. En geen geplak meer.
Binnenkort willen we wat aandacht besteden aan allerlei leuke routine's die je in de acht K Ram van de Multiface kan laden.
Hierbij een oproep wie helpt mij aan "Your Sinclair" het augustus nummer van 1989.

```
**********************************************************************
*                                                                    *
*                    Wat kunt u verwachten:                          *
*          Op de a.s gebruikersavond twee demonstraties.             *
*                  Het DTP van de heer F. grunefeld.                 *
*  en Rudy laat voor ons zijn Margrietwielprinter draaien.           *
*                                                                    *
**********************************************************************
```

In dit nummer:

-Gebruikersbijéénkomsten	: redaktie
-Van onze voorzitter	: Jan Dirk
-DTP	: F.Grunefeld
-Een snelle pijl	: Frans Postma
-Opus Speedloader	: Roelof Koning
-Carrier Command	: Frans Postma
-Machine Code stap voor stap deel 7	: J.v.Alteren
-Snel en precies rekenen deel 27A	: H.v.Abbe

SLUITINGSDATA KOPIJ 31 JANUARI

GEBRUIKERSBIJEENKOMSTEN



In het: **DENKSPORTCENTRUM**
OLIEMULDERSWEG 43
GRONINGEN

Telefoon: 050-126937

DATA GRONINGEN:

20 jan. zater van 14.00 -17.30
13 feb. dins van 19.30 -22.30
17 mrt. zater van 14.00 -17.30
10 apr. dins van 19.30 -22.30
12 mei. zater van 14.00 -17.30
12 jun. dins van 19.30 -22.30

JANUARI 1990

ZO	-	7	14	21	28
MA	1	8	15	22	29
DI	2	9	16	23	30
WO	3	10	17	24	31
DO	4	11	18	25	-
VR	5	12	19	26	-
ZA	6	13	20*	27	-

FEBRUARI 1990

ZO	-	4	11	18	25
MA	-	5	12	19	26
DI	-	6	13*	20	27
WO	-	7	14	21	28
DO	1	8	15	22	-
VR	2	9	16	23	-
ZA	3	10	17	24	-

DATA HOUTEN:

13 JAN. 10 MRT. 12 MEI.
23 JUNI. 8 SEPT. 27 OKT.
In het HCC-kantoor Standermolen 8
te Houten. tel:03403-78788.

DATA HOOGEVEEN:

8 JAN. maandag vanaf 19.30
In het Wijkcentrum "DE MAGNEET" ORION 2 HOOGEVEEN.

DATA EEMSMOND:

21 DEC. donderdag van 19.00 tot 22.00
23 DEC. zaterdag van 13.00 tot 17.00 (kerstinstuif)
In de L.O.M. School "De Wenakker" Pastorielaan 2 Appingedam.

****

VERHUUR VERHUUR VERHUUR

U kunt bij de club het volgende huren:

A.M.X. muis met interface
CURRAH microspeech
VTX 5000 modem

Inlichtingen bij de heer Jan Arends.

****

CASSETTE SERVICE

Van bijna alle programma's die in ons bulletin stonden hebben wij een cassette bandje gemaakt.
U kunt dit bandje op een gebruikersbijeenkomst lenen, meestal ligt deze bij de deurwacht, zoniet vraag het dan aan Flora.
De programma's waar Copyright op zitten, kunnen natuurlijk niet op deze cassette voorkomen.

****

VAN ONZE VOORZITTER



Een nieuw jaar dus dan ook een nieuw begin.
Als bestuur hebben we al weer de nodige vergaderingen achter de rug. Harde noten kraken, jawel.
Tot de conclusie komen dat we hopen dat iedereen zo snel mogelijk zijn contributiegeld overmaakt opdat we de balans op kunnen maken hoeveel leden we in het jaar 1990 hebben.
Totdusver gaat alles nog lekker, maar we voelen de hete adem van de peecees in onze nek. Niet zo verwondelijk natuurlijk als je ziet op wat voor agressieve manier men ze aan de man probeert te brengen. Maar laten we wel zijn, de echte speccie gebruiker blijft natuurlijk zijn vereniging trouw.
De laatste bijeenkomst werd weer goed bezocht en er waren weer vele intressante zaken te zien. Herman Vesper met zijn satelliet ontvanger en de daarbij behorende software op de spectrum gaven mooie beelden te zien. Ook aan andere tafels werd druk gewerkt aan prachtige routine's in programma's. Zo zie je maar weer dat men nog lang niet uit geprogrammeerd is op de spectrum en dat doet mij deugd moet ik zeggen.
Indien er vragen zijn op hard- en software gebied, schroom dan niet om te vragen. We zijn er voor. En dan bedoel ik alle leden.

J.D. Burggraaf.



DESKTOP PUBLISHING



DTP

Toen ik met de Spectrum begon dacht ik, als ik dat ding goed kan bedienen ben ik al een hele Piet. En nu word ik aangekondigd als de auteur van een Desk Top Publishing programma. Dit stukje zal uit de doeken doen wat er precies aan de hand is. De ingredienten voor mijn programma zijn: TASWORD 3 en een screendump programma. Daarvoor koos ik Tascopy, net zoiets als copycode maar heeft als voordeel dat vooraf de kantlijn bepaald kan worden. Een tweede voordeel is een minimum verlies aan tekstgeheugen want tascopy is relocatable. Met deze combinatie is het mogelijk om afwisselend een stukje tekst en dan weer een plaatje te printen.
Het run-programma moest voor deze extra activiteiten nogal uitgebreid worden. Omdat daarvoor niet genoeg ruimte is, heb ik het in tweeën gedeeld. De volgende regels zijn aan het eerste deel toegevoegd.

```
 15 CLEAR VAL "64767": LOAD *VAL "1";"tascopybw3"CODE VAL "
64768"
 70 LOAD *1;"tw3tascopy"
 110 SAVE *1;"run" LINE 15
```

Het deel dat in regel 70 wordt geladen ziet er als volgt uit:

```
 1   INPUT "Scherm? ";a$,"5 =A5 6 =A6 ";a: IF a<5 OR a>6
 THEN  GO TO 1
 2  INPUT "L.mar. ";m: IF m>40 OR a=5 AND m>8 THEN GO TO 2
 3  LOAD *VAL "1";a$SCREEN$ : GO TO a
 5  POKE VAL "64941",VAL "m": RANDOMIZE USR VAL "64768":
 GO TO VAL "10"
 6  POKE VAL "65197",VAL "m": RANDOMIZE USR VAL "65024"
 10 LPRINT CHR$ (27);"2";: RANDOMIZE USR VAL "25000"
 110 SAVE *1;"tw3tascopy" LINE 10: REM alleen basic
```

Het eerste deel van regel 10 zet de regelsprong weer op normaal want die was door tascopy ingesteld op 8 puntjes per regel en dat is voor een tekst niet genoeg.
Regel 110 staat in verband met de waarschuwing onderaan.

HET GEBRUIK:

Het is handig om vooraf de regelnummers te noteren van de tekst blokken die boven of onder een plaatje moeten worden afgedrukt. Eventueel kan ook alvast door middel van lege regels de plek van een plaatje worden vast gelegd.
Klaar? Dan naar het hoofdmenu. kies de B van terug naar BASIC en vervolgens: CLEAR->ENTER, RUN->ENTER.
Dan 3 vragen nl. de naam van het plaatje, of het een A5 of een A6 moet worden en de linker marge. De printer moet nu klaar staan want het printen begint direkt na het laden. Laat je het hele plaatje uit printen dan kom je terug in het hoofdmenu. Stop je eerder, dan moet dat via GOTO 10 en de cyclus is rond.

WAARSCHUWING: Als je TASWORD 3 gaat SAVEN of copieeren via de eigen SAVE-routine, dan krijgt het tweede deel basic de naam run en dat is fout. Het moet (in dit voorbeeld) "tw3tascopy" zijn. maar er kunnen maar 3 letters staan, kies dus b.v. "twc" o.i.d. Hoe je dat kunt doen staat beschreven in SINCLAIR-IMPULS No.54 blz:29. Ik heb de naamswijziging aangebracht met het programma LINKED. De naam:"run" staat in TASCODE2, in het 24e of 25e blok.

Op zaterdag 20 januari zal ik mijn programma demonstreren.

F.Grunefeld, Ommelanderdrift 27, Bedum. Tel. 05900-13505

****

VAN DE PENNINGMEESTER

Berichtje

Tijdens de laatst gehouden bestuursvergadering is in verband met de steeds stijgende kosten en het daardoor dalend saldo op de giro besloten de entree voor de bijéénkomsten te verhogen met één gulden.
Voor de leden is dit nu f2,50, voor introducee's f4,50 zonder bulletin en f5,50 inclusief bulletin.

De contributie betaling verloopt vrij redelijk.
Heeft U dit nog niet voldaan en wilt U toch zeker zijn van toezending van het volgend bulletin, maak dan voor 1 Februari a.s even die acceptgiro over. Bedankt.

J.H.Arends
penningmeester S.G.G.

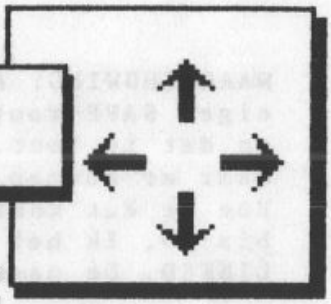
****

ESGEEGEETJE

TE KOOP AANGEBODEN:
Een nieuwe 3,5 -80 tracks enkelzijdige diskdrive f65,00.

H.Vesper tel:050-734307

EEN SNELLE PIJL



POINTER V2.3

Deze routine print een pijl op het scherm, die kan worden bewogen met de toetsen:Q,A,U,I en P. Q,A,U en I bewegen de pijl , en P is 'select'. Wordt er op 'P' gedrukt, dan stopt de routine, en keert terug naar BASIC.
De laatste positie van de pijl staat dan in 61000 en 61001, dit is de PRINT AT positie.
De toetsen kunnen verandert worden door de regels aan te passen. B.v. CP "Q" naar CP "W" als u de "W" wilt gebruiken als omhoogtoets. Denk er wel aan dat de capslock altijd aan staat, en dat de besturingstoetsen dus nooit kleine letters mogen zijn!

De pijl-UDG is makkelijk te veranderen, gewoon bij het 'DEFINE POINTER GRAPHIC' gedeelte de waardes achter de % veranderen. Dit zijn de binaire waarden, zoals je ze ook vanuit BASIC invoert.

Het HL' register moet veilig weggezet worden, omdat je anders problemen krijgt bij terugkeren naar BASIC.

De routine zet aan het begin het scherm weg op 50000. Iedere keer als de pijl bewogen wordt, wordt dat scherm weer teruggekopieerd (+kleuren). Dat geeft een wat schokkerig effect, maar je kunt nu ook over screen$ bewegen.

De beschrijvingen achter de instructies, geven niet aan wat die instructie doet, maar wat het doel ervan is. Bij de beschrijvingen, ga ik er van uit dat u wel al wat machinecode kent. Als u de machine-code kursus in het bulletin gevolgd hebt, zult u het meeste wel begrijpen.

Even iets over alternatieve registers. Heel simpel gezegd, zijn dit extra registers die je kunt op roepen met EXX. Dit 'EXX' wisselt de normale registers en de extra registers. Op deze manier heb je dus de beschikking over 16 registers (als je niet naar de index-registers kijkt). Deze alternatieve registers worden aan geduid met een accent-teken. B.v. HL', dat is dus het alternatieve HL register.
Frans Postma. (11/12/1989)

```
*L-                          ;GENS commando
*D+                          ;GENS commando
*C-                          ;GENS commando
;FRP 1989                    ;maker + jaar
;Pointer V2.3                ;naam + versie nummer
;stores whole screen         ;korte omschrijving

        ORG  60000           ;start adres
        EXX                  ;gebruik ' registers
        PUSH HL              ;zet HL' veilig weg
        EXX                  ;gebruik normale registers
```

Vervolg POINTER V2.3

```
        LD    HL,23612          ;zorg dat je op onderste  deel v/h
        RES   0,(HL)            ;scherm print. RES zet byte op 0
                                ;hier dus byte 0 van 23612
        LD    B,10              ;zet start-waarden
        LD    C,10

;SET CAPSLOCK
        LD    A,8
        LD    (23658),A         ;zet caps-lock aan

;DEFINE POINTER-GRAPHIC
        LD    HL,(23675)        ;adress van eerste UDG
        LD    (HL),%00001111    ;laad HL met binair 00001111
        INC   HL
        LD    (HL),%00000011
        INC   HL
        LD    (HL),%00000101
        INC   HL
        LD    (HL),%00001001
        INC   HL
        LD    (HL),%00010000
        INC   HL
        LD    (HL),%00100000
        INC   HL
        LD    (HL),0            ;is gelijk aan %00000000, maar
        INC   HL                ;dat is zo omslachtig
        LD    (HL),0

;COPY SCREEN                    ;copieer scherm naar 50000
        EXX                     ;gebruik de alternatieve registers
        LD    HL,16384
        LD    DE,50000          ;LDIR copieert een blok code,
        LD    BC,6912           ;dat start op HL en lengte BC
        LDIR                    ;heeft naar DE toe.
        EXX                     ;gebruik weer normale registers


;NOW RE-PRINT SCREEN
LOOP    EXX                     ;copieer scherm nu weer naar
        LD    HL,50000          ;16384 toe. zodat steeds het
        LD    DE,16384          ;'start' scherm op 16384 staat
        LD    BC,6912           ;gebruikt alternatieve registers
        LDIR
        EXX

;NOW PRINT ARROW
        LD    A,17              ;print pijl met INK 0 en PAPER 7
        RST   #10               ;RST #10 print een karakter op
        LD    A,7               ;het scherm
        RST   #10               ;chr$16 = INK controle
        LD    A,16              ;chr$17 = PAPER controle
        RST   #10
        LD    A,0
        RST   #10
```

Vervolg POINTER V2.3

```
        LD   A,22              ;chr$22 = PRINT AT controle
        RST  #10
        LD   A,B               ;print at B,C
        RST  #10
        LD   A,C
        RST  #10

;GRAPHIC "A" IS CHR$(144)

        LD   A,144
        RST  #10               ;print GRAPHIC A

;NOW FORMER POS=CUR.POS

        PUSH BC                ;DE=BC
        POP  DE


;KEY-LOOP WAITS FOR            ;in de onderstaande CP "x" -
;VALID KEYPRESS                ;instructies, moet u de x ver-
;IF VALID >> H=1               ;veranderen om een andere
                               ;besturings toets te kiezen

KLOOP   LD   A,(23560)         ;laad A met laatst ingedrukte
        LD   H,0               ;toets (LASTK)

        CP   "Q"               ;test voor geldige toetsen
        JR   NZ,N1             ;indien geldig H=1 en de
        DEC  B                 ;bijbehorende actie word uitgevoerd
        LD   H,1               ;toetsen:Q =omhoog
N1      CP   "A"                        A =omlaag
        JR   NZ,N2                      U =links
        INC  B                          I =rechts
        LD   H,1                        P =select
N2      CP   "U"
        JR   NZ,N3
        DEC  C
        LD   H,1
N3      CP   "I"
        JR   NZ,N4
        INC  C
        LD   H,1
N4      CP   "P"               ;test voor select
        JR   Z,PEND            ;zo ja, beeindig dan routine

        LD   A,H
        CP   1                 ;test voor geldige toets
        JR   NZ,KLOOP          ;nee, dan naar toetsen scan

        LD   HL,23560          ;maak LASTK leeg
        LD   (HL),0

;CHECK IF VALUES ARE
;VALID. ELSE LOOP AROUND
;THE SCREEN.
```

Vervolg POINTER V2.3

```
        LD   A,B
        INC  A                  ;verhoog A want testen op
        CP   0                  ;-1 is niet mogelijk
        JR   NZ,OK1
        LD   B,21               ;if B=-1 then B=21
OK1     CP   23
        JR   NZ,OK2
        LD   B,0                ;if B=22 then B=0
OK2     LD   A,C
        INC  A
        CP   0
        JR   NZ,OK3
        LD   C,31               ;if C=-1 then C=31
OK3     CP   33
        JR   NZ,OK4
        LD   C,0                ;if C=32 then C=0

OK4     JP   LOOP               ;spring naar loop

;END PROGRAM RESTORE
;POSITION BC
;PUT BC IN 61000/1

PEND    EXX                     ;copieer 'start' scherm
        LD   HL,50000           ;weer naar 16384
        LD   DE,16384           ;gebruikt alternatieve registers
        LD   BC,6912
        LDIR
        EXX

;PUT AWAY BC AND EXIT

        LD   A,B
        LD   (61000),A          ;POKE 61000,B
        LD   A,C
        LD   (61001),A          ;en POKE 61001,C

        EXX                     ;haal HL' op
        POP  HL
        EXX
*L+                             ;GENS commando
;END OF M.C. ROUTINE
;SHOULD BE UNDER 61000

        RET                     ;keer terug naar basic
*L-                             ;GENS commando
```

Frans Postma.

OPUS SPEEDLOADER



Door Roelof Koning. Deel1,
volgende maand deel 2.

Na een gesprek met Koen Willems van S.G.Hoogeveen, waarin hij naar bepaalde informatie m.b.t. de Discovery vroeg, ben ik een aantal M.C.routines die ik een poos geleden al in mijn OPUS-ROM ingebouwd heb, eens opnieuw gaan bekijken.

Een van deze extra routines betreft een soort 'load-p' commando zoals de DISCIPLE die ook kent, en dit werkt al een poos tot tevredenheid bij mij. Opeens bedacht ik hoe weinig moeite het zou kosten om er een 'zelfstandig codeblok' van te maken dat dan plaats kon vinden in de RAM 6116 die velen al in hun machine ingebouwd hebben.

Het aardige van deze 'load-p' routine is dat deze werkt volgens een snelload-principe (waarvoor ik dankbaar gebruik heb gemaakt van door Edwin Blink aangedragen informatie), en dat er dus uitstekend een RUN-rogramma mee gemaakt kan worden.

Zo gedacht, zo gedaan, waarbij ik en passant de screen-layout van een door Koen Willems gemaakt runprogramma overgenomen heb. (Het klinkt nu net alsof ik zelf niets meer bedenk!)

Voordat ik verder ga met een listing, eerst even een stukje theorie over het 'disk-format' m.b.t. snelloaden. Zie ook blz. 10 van het OPUS operating-manual.

Een disk draait met 300 toeren per minuut, dus elke omwenteling duurt 0.2 seconden. In die tijd draaien er (bij een normale OPUS schijf) 18 sectoren van 256 bytes onder de opname/weergave-kop door. Per sector is er dus ca. 1/100 sec. beschikbaar om deze in te lezen en te verwerken.

De Z-80 chip kan zijn deel gemakkelijk aan, maar het systeem van de OPUS kan dit niet. (Buffers, tabellen, enz.). Wanneer de inhoud van de bijv. eerste sector is overgedragen en de OPUS gereed is om de tweede sector te lezen, dan is deze allang onder de kop van de drive doorgedraaid, en moet er gewacht worden totdat deze sector weer 'langskomt'.

Om deze wachttijden te voorkomen wordt de sector met rugnummer twee gezet op de dertiende 'sector-plaats' op een track van 18. Sector 3 komt dan op plaats acht, enz.. OPUS schijven worden geformatteerd met een interleave-factor 13. De tijd die nodig is om de kop van track naar track te bewegen word opgevangen door een soortgelijke factor, de 'skew'. Deze is standaard 5.

Wanneer we de 'diskcontroller-chip' in de OPUS rechtreeks gaan aanspreken, buiten het OPUS-systeem om, dan wordt het tempo van data-lezen en verwerken dusdanig hoger dat na sector 1 meteen sector 2 'gepakt' kan worden, maar helaas komt deze op een normale schijf pas langs nadat er eerst 12 andere sectors onder de kop gepasseerd zijn. Dus moet de snelloader staan te wachten!

Het is voor werkelijk snelloaden nodig om de schijf met interleave-factor 1 te formatteren. Gelukkig kan dit vrij eenvoudig met de OPUS. Voor SAVEn of LOADen m.b.v. het normale OPUSsysteem is een dusdanig aangemaakte schijf weer erg traag.

Omdat alleen in het 'load p' commando is voorzien, en het daarnaast noodzakelijk is dat de div. programma-onderdelen op een vaste plaats in de CAT staan, is het runprogramma vooral handig voor schijven die 'af' zijn, bijv. gevuld met games. Het aanmaken van zo'n schijf vereist enige zorg, immers in de Basicloader moet ook 'load-p' gebruikt worden. Zelf bracht ik van een spel eerst de CODEdelen naar schijf, SAVEde daarna de aangepaste BASIC. Als laatste werd het RUNprog. aangepast en ge-SAVEd.

Dan nu over het RUN-programma zelf, eindelijk.
Een MachineCode-deel bevindt zich in REMregel 0, de M.C. hiervoor moet helaas wachten tot volgende maand. Daarna wordt het keuzemenu via BASIC aangemaakt, dit is dus gemakkelijk aan te passen.

Wanneer e.e.a. geRUNd is, dan kent de OPUS het commando 'LOAD x', waarbij x een getal voorstelt. Na het geven van dit commando (direct of via een prog.regel) zal het 'x-ste' file uit de CATalogue worden geladen. Dit werkt voor de categorieen PROG en CODE (geen DATA dus). Dit nieuwe commando gebruikt de snelload-methode.

Een 'extraatje' wil ik nog belichten. Wanneer d.m.v. BREAK uit het runprog. gesprongen wordt, dan blijft het LOAD x commando bestaan. Dit zit immers in de 6116 RAM. Van normaal geformatteerde schijven LOADt dit met normale snelheid.

Ook is er dan het commando 'LOAD @ x,y,z' beschikbaar, dat de inhoud van track x, sector y, overbrengt naar het geheugen op adres z. Hiermee kunnen bijna alle schijven die in de gebruikte drive 'passen', (qua tracks, single of double sided e.d.), gelezen worden. Op mijn 80 tracks drive lees en schrijf ik bijv. probleemloos de sectoren van PC-, Atari-, en DISCIPLE-disks.

In het commando 'SAVE @ x,y,z' is ook voorzien, dit brengt data ter lengte van een sector van het aangegeven adres naar de schijf.

Deze commando's zijn ook erg handig om de boot-up-sector (het 'ankerblok' = 0,0) van een schijf te bekijken en/of te veranderen. De nieuwe commando's werken overigens altijd met drive 1.

Let er nog even op dat in de OPUS-catalogue de sectoren weliswaar 'doorgenummerd' zijn, (1-720), maar dat de diskcontrollerchip hier geen boodschap aan heeft.

Dan nu een deel van het intypewerk:

Hulpprogramma: Formatteer een disk met int.l./skew = 1/17

```
10 CLOSE #10: OPEN #10;"CODE "
20 POINT #10;8192
30 POINT #10;10+CODE INKEY$#10+256*CODE INKEY$#10
40 POINT #10;CODE INKEY$#10+256*CODE INKEY$#10
50 IF CODE INKEY$#10<>1 THEN PRINT "Use drive 1": STOP
60 LET tabel=CODE INKEY$#10+256*CODE INKEY$#10
100 POINT #10;tabel+11
110 PRINT #10;CHR$ 1;CHR$ 17;:REM let op ';'
120 CLOSE #10
200 PRINT "Interleave/skew nu: 1/17"
300 PRINT '" Insert new disk in drive 1 "
310 INPUT "DISK name? ";i$
320 PRINT ''FLASH 1;" FORMATTING NOW ";i$
400 FORMAT 1;i$
500 RANDOMIZE USR 14070: STOP: REM restore table
```

De variable 'tabel' in r.60 wijst naar de disk-info-tabel van drive 1. In deze tabel staat ook de overige info m.b.t. de disk. Let op dat de interleave en skew niet als zodanig op de disk vastgelegd zijn, en daarom ook niet in de tabel ge-updated worden. Het prog. herstelt de tabel naar z'n standaard waarden.

MENU-programma: (remreg.0 volgt later)

```
10 CLEAR #: DIM q$(13)
20 REM: hier komt een USR wanneer reg.0 gemaakt is
30 RESTORE : PRINT AT 7,2; INVERSE 1;"    MENU     ": READ h:
   FOR l=8 TO 8+h: READ q$(2 TO ): PRINT AT l,2;q$: NEXT l
40 PLOT 16,120: DRAW 103,0: DRAW 0,-(h+2)*8: DRAW -103,0:
   DRAW 0,(h+2)*8: LET l=9
50 GOSUB 150
60 IF INKEY$<>"" THEN GOTO 60: REM kan evt. vervallen
70 LET i$=INKEY$: IF i$="" THEN GOTO 70
80 IF CODE i$=13 THEN GO SUB 150: FLASH 1: GOSUB 150:
   GO TO 1000+(l-9)*100
90 IF i$<>" " THEN GO TO 60
100 GOSUB 150: LET l=l+1: IF l-8>=h THEN LET l=9
110 GOTO 50
150 PRINT AT l,3;OVER 1;INVERSE 1;q$(3 TO ): RETURN
200 DATA 8,""
210 DATA "naam1","naam2","n3","n4","n5","n6","n7",""
```

De volgende regels kunnen pas ingevoerd worden wanneer de REMregel aangemaakt is, nog even geduld dus.

```
 999 REM  loaders naar behoefte aanmaken
1000 LOAD 3
1100 FORMAT "j";1: LOAD 6
1200 OPEN #3;"t" RND 64: BORDER 0: LOAD 9
1300 LOAD 12
1400 LOAD 15
1500 REM enz.
2000 STOP
```

Het 'menukeuze' deel van dit programma is ook voor andere doeleinden erg geschikt, het loont volgens mij echt de moeite om hiermee (ook) eens te stoeien.

Volgende maand gaan we een REMregel van 740 bytes lang samenstellen, plus een programma om hier data in te poken. Deze regel moet dan geMERGEd worden met bovenstaand menuprogramma, samen met een nieuwe reg. 20.

Tot zover voor deze keer. Wil je niet wachten, en/of heb je geen zin in intypen, neem dan een schijfje mee naar de gebruikersmiddag.

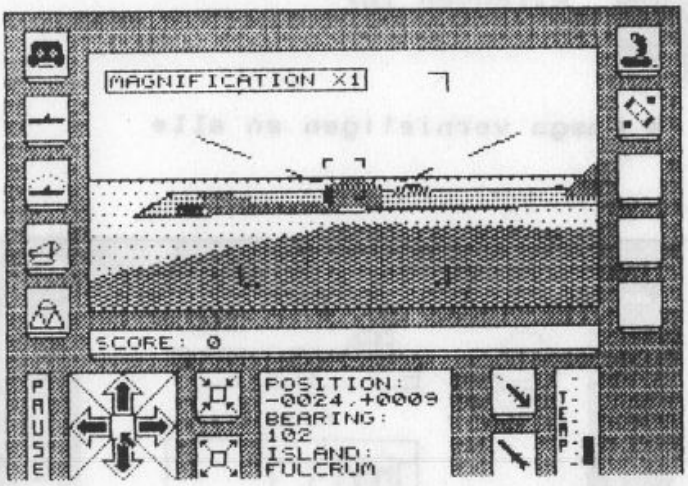
Roelof.

CARRIER COMMAND



Spelreview "CARRIER COMMAND"
van Frans Postma.

Dit spel is zowel een stategie als aktie-spel, je kan namelijk uit twee opties kiezen voor je begint. Action of Strategy game.
In dit spel bestuur je een vliegdekschip en heb je de beschikking over 3 vliegtuigen en 3 amfibivoertuigen met bijbehorende bewapening e.d.
Je doel is de vijand, die ook een vliegdekschip en een heleboel vliegtuigen heeft, te vernietigen en alle eilanden (ongeveer 23 !) terug te veroveren.



aanval van een vijandelijk eiland.

Bij Strategy start je met alle eilanden neutraal.Omdat je voorraden niet oneindig zijn, moet je eilanden veroveren en daar fabrieken opzetten. De goederen die deze fabrieken maken, worden naar een door jouw aangewezen eiland vervoerd (in het begin is dat VULCAN). Als je meer eilanden verovert kun je niet steeds terugvaren naar dat eiland en moet je dus een nieuw 'stockpile' eiland benoemen.

Bij Action start je met de helft van de eilanden in jouw handen , en de andere helft is van de vijand.

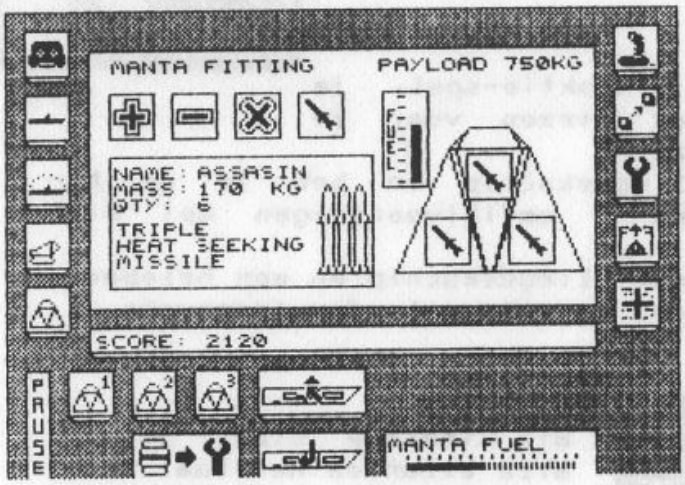
Hoe meer eilanden je verovert, hoe sneller er geproduceerd zal worden. Als je natuurlijk wel hebt ingesteld, wat, en hoeveel er waarvan geproduceerd moet worden. In het begin staat namelijk alles op 0, en wordt er dus niks geproduceerd !!

Een eiland verover je door eerst de vijand weg te schieten, dit kan je doen met Manta's (vliegtuigen) , Walrussen (amfibivoertuigen), of met de Epsilon zelf (je vliegdekschip).

Voor een aanval zijn Manta's het beste, omdat die meer raketten kunnen meenemen. Als een eiland eenmaal neutraal is (kleur is dan groen), moet je er een Walrus heensturen met een fabriek-opbouw-eenheid (dit heet een ACCB unit). Zodra de eenheid op het eiland is neergezet, is het eiland in jouw handen.
Als een eiland van jouw is wordt het blauw (Vijandelijk is rood). Vergeet niet je stock-pile eiland te verplaatsen als je eenmaal een eiland ingenomen hebt, omdat je anders een heel eind moet varen voordat je de voorraden op kunt halen !

vervolg Carrier Command.

De Omega (het vijandelijke vliegdekschip) zal echter steeds proberen jouw eilanden in te nemen.

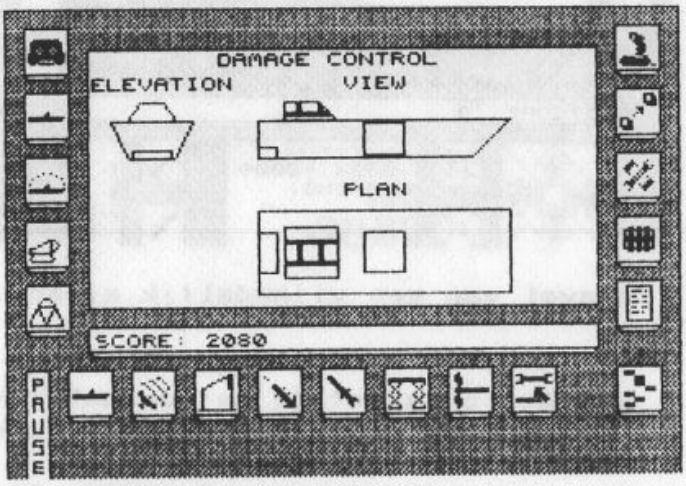


Ook kun je de Omega zelf tegenkomen, dan ben je wel even bezig, voor je die de grond in hebt geschoten (hier heb je ongeveer 2 Walrussen, 3 Manta's en 23 oppervlakte raketten voor nodig). Heb je de Omega eenmaal kapot geschoten dan wordt het aardig makkelijk (want niemand neemt jouw eilanden in)

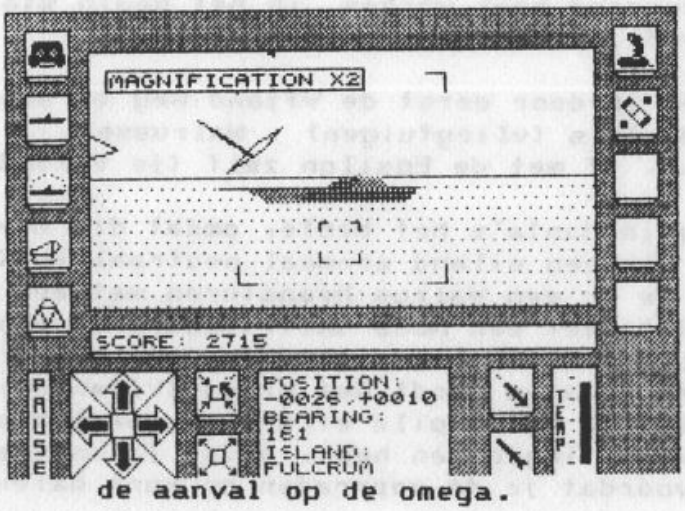
Het hoofddoel in dit spel is: de Omega vernietigen en alle eilanden weer in te nemen.

Conclusie:

het spel op zich is leuk, als je maar éénmaal weet hoe je alles moet bedienen. Want het werkt namelijk met allemaal icons, dit zal wel komen doordat het spel van de Amiga overgezet is.

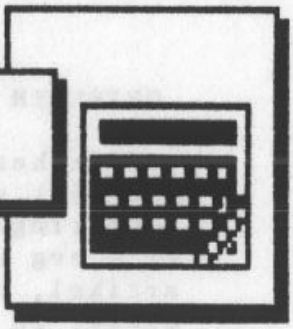


overzicht schade.



de aanval op de omega.

MACHINECODE STAP VOOR STAP DEEL 7



(c) J. v. Alteren, SGG; 020190.

Uit ZX-Computing afl.: apr/mei '85, blz. 88 ev.
door David Nowotnik.

7e aflevering:

DEEL 3: Werken met Vlaggen [Flags]: (vervolg)

2. VERGELIJKINGEN.

Een eenvoudig voorbeeld van het gebruik van 'Flags' is in samenhang met de machinetaal-opdracht CP (compare). [to compare = vergelijken] Dit geeft de mogelijkheid om de inhoud van het A-register te vergelijken met een getal of met de inhoud van elk ander register of met de inhoud van een byte in het geheugen, aangewezen door HL. De CPU vergelijkt door de inhoud van het register waarmee wordt vergeleken af te trekken van het A-register. Het resultaat wordt echter niet opgeslagen, het beinvloedt alleen de waarde van de Flags. Hier volgt een typerend voorbeeld van het gebruik van de CP-instruktie:

```
CP E
JR NZ, next
```

Na de vergelijking van de waarde in het A-register met die in het E-register, stelt de CPU de Z-Flag in, afhankelijk van het resultaat .
Andere flags worden ook door de vergelijking beinvloed.
Als A = E dan wordt de Z-flag op gezet, als A ongelijk is aan E wordt de Z-flag 'gereset'.

De opdracht 'JR NZ, next' test de waarde van de Z-flag.
Als de Flag 'gereset' is (NZ -> Not Zero), dan zal een relatieve sprong naar de positie in het programma, die met 'next' wordt aangegeven, plaatsvinden. Is de Flag gezet dan zal het programma direct doorgaan met de opdracht onmiddelijk volgend op 'JR NZ, next'.
worden vastgesteld of opnieuw bepaald aan de hand van de relatieve waarden van de twee registers die werden vergeleken.
Het is ook mogelijk om vast te stellen of de ene waarde groter is dan de andere.
Liever dan U dat allemaal te vertellen, had ik een programma gegeven dat het effect van bepaalde vergelijkingen laat zien: dat wordt daarom het eerste voorbeeld dat verderop in het artikel wordt gegeven. (niet in deze afl.)
Door verschillende waarden in registers te zetten, ben je in staat voor jezelf de 'Flag-toestand' te bekijken voor: 'kleiner dan' en 'groter dan' door het programmaatje toe te passen. Dus t.z.t. uitproberen !!

OPTELLEN EN AFTREKKEN

Of je het leuk vindt of niet, computerprogramma's bergen vaak wat gereken in zich en machinecode is daarop geen uitzondering. De rekenkundige mogelijkheden op de Z80-processor zijn erg eenvoudig: optellen, aftrekken en in een later artikel, de instructies/opdrachten om twee getallen met elkaar te vermenigvuldigen of op elkaar te delen. Alle andere wiskundige bewerkingen worden van deze basisopdrachten afgeleid. De vorige keer beschreef ik de simpele onderwerpen als de waarde van een register of registerpaar met '1' verhogen (INC) of verlagen (DEC).
Er zijn een aantal mc-opdrachten die ingewikkelder optellingen of aftrekkingen mogelijk maken. De keuze in registers is voor deze mogelijkheden wat beperkter dan via INC en DEC.
Het optellen of aftrekken van een enkelvoudig register gebeurd altijd bij of van het A-register.

Optellen houdt in:
Het bijtellen van een getal, van de inhoud van een register of een byte in het geheugen (geadresseerd door het HL-paar), bij de waarde in het A-register en het terugplaatsen van het resultaat in het A-register.

Evenzo: aftrekken houdt in:
De waarde van een getal, een register of geheugen-byte van de waarde in 'A' afhalen en het terugplaatsen van het resultaat in hetzelfde A-register.

TWEE-BYTES-BEWERKINGEN ZIJN BEPERKTER.

Optellen houdt in: bij de waarde van het HL-register-paar de inhoud van een ander register optellen en het resultaat terugplaatsen in het HL-register.

Aftrekken houdt in: de waarde van een registerpaar aftrekken van de waarde in HL, en het antwoord in HL terugplaatsen. (dit gaat altijd inclusief de Carry-flag)

Bij rekenen met een-bytes getallen ligt het bereik van de getallen tussen 0 en 255, terwijl het bereik bij de twee-bytes berekeningen tussen 0 en 65535 ligt.

Een tweede moeilijkheid is de Carry-flag.
Het is mogelijk twee getallen bijelkaar te tellen, dan de waarde van de 'Carry' (0 of 1) daaraan toe te voegen juist voordat de uiteindelijke waarde wordt opgeslagen.
Bij aftrekken is het mogelijk om van het resultaat van de aftrekking van twee waarden ook nog eens de waarde van de Carry-flag er af te trekken, voor het nieuwe resultaat wordt opgeborgen.
Dit geeft twee 'manieren' van optellen; t.w. ADD en ADC, en ook twee van aftrekken: SUB en SBC, met en zonder Carry-flag. Bij 2-bytes aftrekkingen is het echter alleen mogelijk om de Carry-flag mee te nemen. Hopeloos in de bonen?

Ik zal proberen het uit te leggen:

Bij eenvoudig rekenwerk is het gemakkeklijk om daarbij de vormen ADD en SUB te gebruiken, die geen 'carry'-flag nodig hebben.
Dus: als je twee registers of twee register-paren optelt of aftrekt gebruik dan ADD of SUB. Op die manier is er weinig kans om fouten te maken door een verkeerde waarde van de 'Carry'-flag.
Uiteraard zal de 'Carry-flag' wel worden beinvloed door het antwoord van het optellen of aftrekken. Voor meer ingewikkelde berekeningen, met getallen die in diverse bytes zijn opgeslagen, zal je ADC en SBC moeten gebruiken.

De bedoeling van het gebruik van de Carry-flag is, dat je daardoor te weten kan komen, of er al dan niet een optelling of aftrekking heeft plaats gehad, met als uitkomst een grotere dan wel kleinere waarde dan is toegestaan, afhankelijk van de beperkingen van de register-grootte.

Met behulp van de Carry-flag kun je de één onthouden / één lenen regels toepassen bij optelling / aftrekking.

Voor gebruik van ADC of SBC moet je de Carry-flag de juiste waarde geven, voordat de berekening wordt gedaan.
Er bestaat geen speciale opdracht om de carry-flag op 0 te zetten, maar dit kan bereikt worden door eerst de carry-flag op 1 te zetten met SCF (Set Carry Flag), gevolgd door de opdracht CCF, (Complement Carry Flag). CCF maakt de waarde van de C-flag tegengesteld: 0 wordt 1, en 1 wordt 0.

De twee opdrachten zijn elk 1 byte groot (opcodes zijn 37 (SCF) en 3F (CCF) hexadecimaal). De carry-flag kan ook veranderd dmv 1 instructie van 1 byte. De volgende keer laat ik zien, hoe dat gaat.

Alle opcodes voor optellen en aftrekken in tabel 2 (volgende aflevering). Merk daarbij op dat voor 2-bytes berekeningen alleen SBC beschikbaar is, wees er dus zeker van dat de Carryflag wordt ingesteld, zoals de bedoeling is.
Fig. 2 (dus niet tabel 2) verderop in de voorbeelden, geeft een BASIC-programma dat je helpt om alle twijfels over de opdrachten over optellen en aftrekken te laten verdwijnen.

Zo, dit was de nieuwe tekst voor deze keer.

Als U ook in het nieuwe jaar zoveel van deze pagina's door zult komen, wordt deze cursus een succes. Er zijn nogal wat afleveringen te gaan.

Sterkte voor heel 1990 !

J. v. Alteren.

---------- WORDT VERVOLGD ---------

SNEL EN PRECIES REKENEN MET DE ZX 81 DEEL 27A



In de delen 27A en 27B behandel ik het Probleem van de Maand juni 1989 uit de MCC Nieuwsbrief nr. 117. Dat probleem heb ik de naam

"T A L S T E L S E L S"

gegeven en de opgave luidde als volgt:

"Als ik zonder meer beweer: 12 + 12 = 12, dan spreek ik wartaal. Maar schrijf ik:

12(4) + 12(6) = 12(12)

en bedoel ik dan met 12(4) het getal 12 in 't 4-tallig stelsel enzovoorts, dan klopt de bewering. Dan immers komt dit neer op 6 + 8 = 14.
Gevraagd wordt nu de positieve gehele getallen G van 6 cijfers te zoeken, die voldoen aan:

G(14) = G(13) + G(10) + G(9)

Het getal G mag niet met een nul beginnen."

BASIC-PROGRAMMA BS-1

Als we het gezochte getal G door ABCDEF in het decimale stelsel voorstellen is de cofactor van A in de op nul herleide vergelijking:

14**5-(13**5+10**5+9**5) = 7482

en die van B:

14**4-(13**4+10**4+9**4) =-6706.

De 4 overige cofactoren zijn te berekenen als:

van C -1182, van D -154, van E -18 en van F -2.

Na deze omzetting van de vergelijking met de talstelsels in een met decimale grootheden volgt het BASIC-programma (BS-1) van figuur 1 als logische vertaling van de opgave. Hierin bevat de regel 90 de "konditie" voor een juiste oplossing.
En inderdaad levert dit PG de 14 goede antwoorden, die staan in figuur 2. Alleen duurt dat 3 uur en 40 min !
Voor goed begrip: de variabelen behoeven maar tot 6 te lopen, omdat in het 9-tallige stelsel de 9 niet kan voorkomen

MACHINECODE PROGRAMMA MC-1

Het maken van een Machinecode routine MC-1 konform BS-1 is simpel. In figuur 3A wordt deze gegeven en in figuur 3B de Hexdump daarvan.
Die RTN is qua LOOPs afgeleid van "REKENEN" (23B) en gebruikt de CALCULATOR voor het berekenen van de op nul herleide konditie van regel 90 uit BS-1. In wezen wordt berekend:

H+H +18*L +154*D +1182*E +6706*B -7482*C

waarbij de overeenkomst van deze registers met de in het BASIC-PG gebruikte variabelen duidelijk zal zijn.

De 6 LOOP-variabelen worden in volgorde op de CALCULATOR-STACK (STK) geplaatst in het RTN-deel AD 4091/B4. Daarna in het deel AD 40B5/DB succesievelijk vermenigvuldigd en opgeteld als uit de bijschriften volgt, waarbij de instruktie 01 (xch = exchange) een zeer nuttige rol vervult bij het beschikbaar maken van steeds de juiste grootheid als "last value" op de STK.
Als het resultaat van de zoeven vermelde berekening "niet-nul" is, wordt met 'jump-true' over de PRINT-RTN naar AD 40FC, label NO-PRINT, gesprongen. Dat adres is 22 hex bytes verder, vandaar de instruktie 00 22.
Als de konditie "= 0" wel wordt vervuld, worden de zes registers in de juiste volgorde gePRINT in het RTN gedeelte PRINT G, zie de figuur 4.

De benodigde tijd is met deze RTN teleurstellend groot: bijna 1 uur en 59 min ! De oorzaak is dat alleen door de MC-LOOPs tijdwinst wordt geboekt. Het gebruiken van de CALCULATOR is -zoals vaker door mij gesignaleerd- tijdrovend. In wezen voert regel 90 van het BASIC-PG tot hetzelfde gebruik van de CALCULATOR als in het MC-deel COMPUTE CONDITION wordt gedaan.

De bedoeling van dit artikel is te laten zien dat zowel in BASIC als in Machinecode een enorme tijdsbesparing mogelijk is. Met als gevolg een BS-2 programma dat er 11 min en 20 s over doet, ruim 19 keer zo snel als BS-1. En een MC-2 routine die slechts 2½ s nodig heeft. Dat is 2850 * zo snel als MC-1 en 5280-maal zo snel als BS-1.
In "REKENEN" (27B) bespreek ik die programma's.

H A N V A N A B B E

P. S.

De FLOATING POINT hex digits die nodig zijn om de gewenste decimale getallen op de CALCULATOR-STACK te krijgen, bepaal ik met een simpele truc. Daarover en over andere nuttige aanwijzingen die ik kan geven voor het gebruik van de CALCULATOR ben ik best bereid eens een artikel te schrijven.
Indien daar interesse voor is laat me dat dan via de REDAKTIE van het BULLETIN weten.

FIG. 3A "TALSTELSELS" PG: MC-1

```
 16514 4082--0E08      LD    C,08      begin-
C-LOOP 4084--0608      LD    B,08        value
B-LOOP 4086--1E08      LD    E,08          of
E-LOOP 4088--1608      LD    D,08            the
D-LOOP 408A--2E08      LD    L,08              six
L-LOOP 408C--2608      LD    H,08                LOOPs
H-LOOP 408E--C5        PUSH  BC        stack
       408F--D5        PUSH  DE          the six
       4090--E5        PUSH  HL            variables

VARIABLES TO CALCULATOR-STACK

       4091--78        LD    A,B
       4092--08        EX    AF,AF"    store A shortly
       4093--79        LD    A,C
       4094--CD1D15    CALL  151D      STK-A' = C
       4097--08        EX    AF,AF"    retrieve A
       4098--CD1D15    CALL  151D      STK-A = B
       4098--E1        POP   HL        make
       409C--D1        POP   DE          D and E
       409D--D5        PUSH  DE            available
       409E--E5        PUSH  HL              from stack
       409F--7A        LD    A,D
       40A0--08        EX    AF,AF"    store A shortly
       40A1--7B        LD    A,E
       40A2--CD1D15    CALL  151D      STK-A' = E
       40A5--08        EX    AF,AF"    retrieve A
       40A6--CD1D15    CALL  151D      STK-A = D
       40A9--E1        POP   HL        make H and L
       40AA--E5        PUSH  HL          available
       40AB--7C        LD    A,H
       40AC--08        EX    AF,AF"    store A shortly
       40AD--7D        LD    A,L
       40AE--CD1D15    CALL  151D      STK-A' = L
       40B1--08        EX    AF,AF"    retrieve A
       40B2--CD1D15    CALL  151D      STK-A = H

COMPUTE CONDITION

       40B5--EF        RST   28;FPA:
       40B6--2D/0F/01/30 35 10/        dupl/+/xch/stk 18/
       40BC--04/0F/01/30 38 1A/04/     */+/xch/stk 154/*/
       40C3--0F/01/30 7B 13 C0/04/     +/xch/stk 1182/*/
       40CA--0F/01/30 7D 51 90/04/     +/xch/stk 6706/*/
       40D1--0F/01/30 7D 69 D0/04/     +/xch/stk 7482/*/
       40D8--03/00 22/34               -/jump-true    /end
                                           if <> 0 to
                                             NO-PRINT

PRINT G

       40DC--E1        POP   HL        retrieve
       40DD--D1        POP   DE          the six
       40DE--C1        POP   BC            variables
       40DF--79        LD    A,C       A = value of C
       40E0--C61C      ADD   A,1C      make CHR of value
       40E2--D7        RST   10        PRINT:   C-value
       40E3--78        LD    A,B
       40E4--C61C      ADD   A,1C
       40E6--D7        RST   10                 B-value
       40E7--7B        LD    A,E
       40E8--C61C      ADD   A,1C
       40EA--D7        RST   10                 E-value
       40EB--7A        LD    A,D
       40EC--C61C      ADD   A,1C
       40EE--D7        RST   10                 D-value
       40EF--7D        LD    A,L
       40F0--C61C      ADD   A,1C
       40F2--D7        RST   10                 L-value
       40F3--7C        LD    A,H
       40F4--C61C      ADD   A,1C
       40F6--D7        RST   10                 H-value
       40F7--AF        XOR   A
       40F8--D7        RST   10        PRINT
       40F9--D7        RST   10          2 spaces
       40FA--1804      JR    4100      NEXT-H

NO-PRINT 40FC--34                      end-calc
       40FD--E1        POP   HL        retrieve
       40FE--D1        POP   DE          the six
       40FF--C1        POP   BC            variables
```

```
NEXT-H  4100--25        DEC  H
        4101--F28E40    JP   P,408E  H-LOOP if H >= 00
NEXT-L  4104--2D        DEC  L
        4105--F28C40    JP   P,408C  L-LOOP if L >= 00
NEXT-D  4108--15        DEC  D
        4109--F28A40    JP   P,408A  D-LOOP if D >= 00
NEXT-E  410C--1D        DEC  E
        410D--F28840    JP   P,4088  E-LOOP if E >= 00
NEXT-B  4110--05        DEC  B
        4111--F28640    JP   P,4086  B-LOOP if B >= 00
NEXT-C  4114--0D        DEC  C
        4115--C28440    JP   NZ,4084 C-LOOP if C >= 01
        4118--C9        RET          RETurn to BASIC
```

FIG. 3B HEXDUMP ROUTINE MC-1

```
4082--0E08 0608 1E08 1608--068
408A--2E08 2608 C5D5 E578--36B
4092--0879 CD1D 1508 CD1D--272
409A--15E1 D1D5 E57A 087B--47E
40A2--CD1D 1508 CD1D 15E1--2E7
40AA--E57C 087D CD1D 1508--2ED
40B2--CD1D 15EF 2D0F 0130--258
40BA--3510 040F 0130 381A--0DB
40C2--040F 0130 7B13 C004--196
40CA--0F01 307D 5190 040F--1B1
40D2--0130 7D69 D004 0300--1EE
40DA--2234 E1D1 C179 C61C--424
40E2--D778 C61C D77B C61C--465
40EA--D77A C61C D77D C61C--469
40F2--D77C C61C D7AF D7D7--569
40FA--1804 34E1 D1C1 25F2--3DA
4102--8E40 2DF2 8C40 15F2--3C0
410A--8A40 1DF2 8840 05F2--398
4112--8640 0DC2 8440 C9   --322
```

FIG. 1 "TALSTELSELS" PG: BS-1

```
  10 FOR A=1 TO 8
  20 FOR B=0 TO 8
  30 FOR C=0 TO 8
  40 FOR D=0 TO 8
  50 FOR E=0 TO 8
  60 FOR F=0 TO 8
  90 IF 7482*A=6706*B+1182*C+154
*D+18*E+F+F THEN PRINT A;B;C;D;E
;F
 110 NEXT F
 120 NEXT E
 130 NEXT D
 140 NEXT C
 150 NEXT B
 160 NEXT A
```

FIG. 2 PRINTOUT PG BS-1 EN BS-2

```
106245
110488
110503
216748
221234
327480
331737
438225
442468
548728
553214
663717
774448
885180
```

FIG. 4 PRINTOUT PG MC-1 EN MC-2

```
885180  774448  663717  553214
548728  442468  438225  331737
327480  221234  216748  110503
110488  106245
```

ESGEEGEETJES



Te koop aangeboden:
QL (AH-ROM) f 180,00 * t.v.-splitter f24,00
QL (JS-ROM) f 220,00 * Transformboxen f35,-
QL (JM-ROM) f 200,00 * EPROM-Toolkit II+handl f35,-
Letterwielprinter f325,- * Sincl GP1000 printer f450,00
RAM-ROM-module + handl f100,- * Lint Brother/Centron f7,50
Philips 80 monitor groen f150,- * Teac 3,5-dr 720 KB nw f230,-
RAM-ROM-module +handl f100,- * EPROM-ICE +handl f35,-
EPROM-Basicode3+hl startk f75,- * Multi-EPROM-kaart QL f110,-
Mitsubishi 5 1/4-dr van 720 KB naar 1,22 MB omschakelaar f235,-
Mitsubishi 3 1/2-dr 360 KB f100,-
Canon 3 1/2-dr van 720 KB naar 1.22 MB omschakelaar f240,-
Cartridges: nieuw f7,00 - gebruikt f4,50.
Schakeling om 2 computers op 1 monitor aan te sluiten f50,-
Tafel voor 2 QL's, monitor en printer f50,-
Orig. Spectrum-programma's, en boeken o.a. Micro-Prolog, Primer
Orig. MS-Dos-programma's en handleidingen
Spectrum Schadow Rom DisAssembly, The Complete Spectrum ROM Dis. D.Jones: Delving Deeper into your ZX Spectrum, Tony Baker: Mastering Mach code on your ZX Spec. P. Holmes: Spec. Machcode Made Easy, Vol II.
Originele QL-programma's en boeken: Psion-Pakket, Archiver, Liberator 3.2, Eye-Q, ICE, Matchpoint, Cavern, Karate, PCB-Designer, Prof. Astrologer, ICE, Micro-Bridge, Prolog, Cartridge-Dokter, GST QL-Macro Assembler enz., met handleidingen, Jrgngn QL-World, Quasars. B.Allan: QL Companion, T Bridge & R Williams: Sinclair QL. Adventures, A.Dickens: QL Advanced User Guide, M.Gandolf: Advanced Programming with the Sinclair QL, C.Opie: QL Assembly Language Programming, S.Papert: Programmeertaal LOGO, A. Pennel: Assembly Language Programming on the QL, 68008-microprocessor, The Sinclair QDOS Companion, T.Tebbe: QL-Technical Guide. P.Jaulent e.a: 68020/68030 Microprocessors and their Coprocessors, K.Mathews: Assembly Language Primer for the Macintosh, C.L.Morgan: Hidden Powers of the Macintosh.
Fred Vink tel: 02230-34250

****

Te koop gevraagd:
QL-UNIX en ROM-APL van Micro-APL + handleiding.
Fred Vink tel: 02230-34250 Den Helder.

****

Te koop aangeboden:
Spectrum 48K in Lo Profile toetsenbord, Interface I + 2 microdrives + Sanyo recorder + Timex 2040 printer.
Martin den Hollander tel: 05978-45474.

****

Te koop gevraagd:
Een origineel SEIKOSHA LINT met inktkussentje.
Bel H. van Abbe tel: 01751-14216

****

PORT BETAALD
GRONINGEN

DRUKWERK

AFZ:

REDAKTIE SGG:
Mevr. F. Elstrodt
Kam. Onnesstraat 172
9727 HS Groningen

AAN: