

# BULLETIN

## SINCLAIR GEBRUIKERSGROEP GRONINGEN/ASSEN





# COLOFON

**VOORZITTER:**

Jan Dirk Burggraaf  
Kluivingskampenweg 30  
9761 BP Eelde  
tel.05907-1697

**SEKRETARIS:**

Martin den Hollander  
Numero Dertien 8  
9644 TV Veendam  
Tel. 05978-45474

**PENNINGMEESTER:**

S.E. Kroon  
Oosterhoutstraat 96  
9401 NK Assen  
Tel. 05920-15912  
Giro 5212298 t.n.v.  
rekening SGG

**VICE VOORZITTER/****PENNINGMEESTER:**

J. van Alteren  
De Grouw 6  
9351 LP Leek  
Tel. 05945-15678

**VERHUUR:**

C. van Krimpen  
Koldakker 34  
9407 BM Assen  
Tel. 05920-70093

**REDAKTIE BULLETIN:**

F. Elstrodt  
Kam. Onnesstraat 172  
9727 HS Groningen  
Tel: 050-263930

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

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

De sluitingsdatum voor kopij wordt in elk bulletin vermeld.

Opgave voor lidmaatschap + abonnement op het bulletin per jaar bij de penningmeester (zie boven).

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

**H**et lidmaatschap van onze gebruikersgroep bedraagt f17,50 per kalenderjaar voor mensen tot en met 17jaar, en voor mensen ouder dan 17 jaar bedraagt dit f25,00 per kalenderjaar. Bij deze prijs inbegrepen is ook het bulletin dat U thuisgestuurd krijgt.

## VAN DE REDAKTIE



Hallo allemaal.

**U** heeft vast wel door het late verschijnen van ons bulletin vernomen dat er iets aan de hand was. Nu dat was ook zo, wij werden overspoeld met PECH. Inmiddels zijn we de nodige pech te boven gekomen, en gelukkig verkocht ik de afgelopen jaarmarkt genoeg om bij Bert een grote printer te kunnen kopen. Ik kon niet echt zeggen dat het storm liep, maar er zijn toch leuke deeltjes gemaakt. Ik kocht een Smith-Carona fastext 80. En daar stoei ik dus aardig wat mee, want het werkt wel even anders dan de seikosha. Toen ons krantje van de drukker kwam, zagen we de nodige 1ste maal fouten, helaas, maar tegenover die fouten stond een gezellig dik nummer, dus oordeel zelf, maar mild. In dit nummer treft U erg leuke artikelen aan zoals:



- \_\_Het netjes afronden van getallen\_\_
  - \_\_Geheugen uitbreiding 256K\_\_
  - \_\_Geluid voor de spectrum\_\_
  - \_\_Een deuntje voor de 128K\_\_
  - \_\_Tip voor de 128K gebruikers\_\_
- de red.

## GRONINGEN

GEBRUIKERSAVOND 25 FEBRUARI

**D**e eerst volgende avond in Groningen is op dinsdag 26 januari, we hopen dat U in grote getallen aanwezig zult zijn. Voor februari wordt het donderdag de 25ste, dus kom allen naar :  
School "de Weijert"  
van Schendelstraat 1  
Groningen.



## HET AFRONDEN VAN GETALLEN DOOR Bert v.d. Zaag.

```
10 let NK=3: LET VK=3: LET GE=123.4567
20 LET G1=10^NK
30 LET G2=INT ((GE*G1) +.5)
40 LET G$=STR$ G2
50 IF LEN G$<VK+NK THEN LET G$= " " +G$: GOTO 50
60 LET G$ = G$ (1 TO VK) + "." + G$ (VK+1 TO )
70 PRINT GE, TAB 32-LEN G$ ; G$
```

HIERIN IS NK: HET AANTAL CIJFERS ACHTER DE KOMMA.  
VK: HET AANTAL CIJFERS VOOR DE KOMMA.  
GE: EEN WILLEKEURIG GETAL.

VK IS NODIG OM ( ALS U GETALLEN GEBRUIKT TUSSEN BIJV. 0-1000 )  
VOOR HET GETAL EEN AANTAL SPATIES TE PLAATSEN, WAARDOOR DE  
GETALLEN RECHT ONDER ELKAAR KOMEN TE STAAN.

```
BIJV. NK=3, VK=3, GE=123.4567
RUN....
123.4567                123.457
```

```
OF BIJV. (NK=3, VK=3,) GE=3.4567
RUN....
3.4567                  3.457
```

```
BEIDE GETALLEN TER VERGELIJKING:
123.456                123.457
3.456                  3.457
```

NK GEBRUIKT DE COMPUTER OM ( ALS HET AANTAL GETALLEN ACHTER DE  
KOMMA MINDER ZIJN ALS NK ) EEN AANTAL NULLEN ACHTER HET GETAL  
TE PLAATSEN.

```
BIJV.: NK=3, VK=3, GE= 23.4567    RUN...
23.457                            23.457
```

```
EN... NK=3, VK=3, GE=123.4        RUN...
123.4                            123.400
```

```
NU ALLE GETALLEN EVEN ONDER ELKAAR:
123.4567                123.457
3.4567                  3.457
23.4567                23.457
123.4                   123.400
```

---

## ASSEN

### GEBRUIKERSAVOND 11 FEBRUARI.

**D**e eerst volgende avond in Assen is op donderdag 14  
januari. De gebruikersavonden zijn elke tweede donderdag  
van de maand en dus voor de tweede maal dit jaar op  
donderdag 11 februari.

Allemaal komen en je apparatuur meenemen! Tot ziens, in:

Het Markehuus  
Walakker 11  
Assen (Peelo = Wijk 7 )

## VAN DE VOORZITTER



Vanaf deze plaats wil ik een ieder een gelukkig nieuwjaar toewensen.

Ik hoop dat we een vruchtbaar verenigingsjaar zullen beleven, waarin we een aantal zaken die we ons tot doel hebben gesteld, kunnen gaan realiseren.

Zal de vereniging nog verder groeien of zal er een verzadigingspunt optreden, of in het andere geval is het ook mogelijk dat er leden zijn die afhaken om wat voor reden dan ook.

Dit laatste is natuurlijk niet te hopen.

Als ik mijn oor te luisteren leg blijken er nog heel veel mensen te zijn die een Sinclair machine in huis hebben en er nog zo veel vragen over hebben, dat het mijnsinziens met de medeclubleden mogelijk kan zijn om de geheimen van de Sinclair machines op te lossen.

Veel vragen zijn er over Tasword, Masterfile en de printeraansturing met deze programma's.

Gelukkig hebben we wel leden die een passend antwoord hebben op deze vragen.

Ik hoop dat jullie met ons mee willen denken over het wel en wee van onze vereniging, opdat we ook dit jaar weer tot een goed computerjaar kunnen rekenen.

J.D. Burggraaf

---

gezien 128K voor f199,00

BIJ KWANTUM

---

KOPIJ VOOR FEBRUARI GRAAG VOOR  
26 JANUARI OF OP DE GEBRUIKERSAVOND

---

### KOPIJ

**A**lles wat U graag kwijt wil in het bulletin, zoals Artikelen, listings, recensies of screens graag voor het volgende nummer opsturen voor 26 januari (1988) of inleveren op de volgende gebruikersavond in Groningen of Assen. Artikelen het liefst getikt in Taswoord 2 of 3, en het kan aangeleverd worden op cassette, cartridge, Opus disk. We zien het met veel belangstelling tegemoet.

## 128K GEBRUIKERS

**H**et blad de Sinclair gids heeft een cassette en boeken-servive. Op de cassette SG-04 staat o.a. een programma waar U Taswoord 2 mee kunt omzetten naar Taswoord 128. Tevens zo'n program voor Masterfile 128.

Voor Basis Abonnees is de prijs f12,50. Voor de latere abonnees f17,50.

---

## GELUID VOOR DE SPECTRUM

```
1  REM WATER
10  PRINT " WATER"
20  FOR K = 0 TO 7 STEP 1 : BORDER K : BEEP .01 , RND * 60 :
    NEXT K : BORDER 3 : PAUSE 50
30  FOR K = 60 TO 0 STEP - 1 : BEEP . 01,K : PAPER RND * 6 :
    CLS : RUN
40  SAVE " WATER " LINE 1
```

---



---

## MUZIEK VOOR DE 128K

```
1  REM EEN AARDIG DEUNTJE
10  PLAY " M 2 1 U W 0 X 1 0 0 0 N 3 c C C c e E E e g G 1 g f e
    c C c e E ( c C ) ) " , " U 3 a & & & ) " , " 5 $ B A
    G E & & & F A G C ( ( & & & ) ) H "
20  SAVE " DEUNTJE " LINE 1
```

---

## S P E C T A C U L A I R

```
1  REM SPECTACULAIR
10  FOR F = 30 TO 110 STEP 7 : CIRCLE F,145,F/5 : NEXT F
20  PRINT AT 3,7 ; " SPECTACULAIR !"
30  SAVE ! " CONE " CODE 16384,2048 : SAVE ! "blank " CODE 18432
    ,2048
40  FOR G = 1 TO 0 STEP -1 : FOR F = 0 TO 32
50  PAUSE 1 + G * (33 - F )
60  LOAD ! " CONE " CODE 16384 + F : NEXT F : NEXT G
70  FOR I = 0 TO 10
80  FOR F = 0 TO 2 : LET G = F + 1 : IF G = 3 THEN LET G =0
90  LOAD ! " blank " CODE 16384 + ( 2048 * F ) :
    LOAD ! " CONE " CODE 16384 + ( G*2048 )
100 NEXT F : NEXT I
110 FOR F = 0 TO 4096 STEP 32
120 LOAD ! " CONE " CODE 16384 + F : NEXT F
```

---



## **Spectrum geheugenuitbreiding 256K ram +96K rom**

Door Roelof Koning.

Twee vragen: Is dit mogelijk? En zo ja, wat kun je ermee? De eerste vraag is snel beantwoord: ja het kan, maar er is een tamelijk technische ingreep in de SPECTRUM nodig. Het antwoord op de tweede vraag is moeilijker te geven, de gebruiksmogelijkheden hangen voornamelijk af van je eigen inventiviteit en vaardigheid in MachineCode. Zelf heb ik Ramdisk\*6 van de Discovery versie 2.2 "omgebouwd", ik beschik nu over een ramdisk van 223K (!) op mijn 48K SPECTRUM. Verder valt er te denken aan zaken zoals Basic-kommando uitbreidingen in een ROMbank, of het "implanteren" van een extra computertaal.

In dit artikel wil ik me overigens beperken tot de technische kant. Het hier beschreven ontwerp is bedoeld om als hulpprint op de Spectrum-printplaat te worden gestoken, in een IC-voet die op de plaats van de ROMchip moet komen. Deze wijze van monteren bergt een aantal nadelen in zich, zoals moeilijk soldeerwerk en een kast die niet meer past. Aan de andere kant zijn daar de voordelen: een eenvoudige hulpprint, korte leidingen, minder kontaktovergangen en een vrije edge-connector. Doorslaggevend bij de keuze is uiteindelijk de overweging geweest dat er ook bij externe montage een portie soldeerwerk op de Spectrumprint nodig zou blijven. Voor het uitgebreide geheugen is namelijk gebruik gemaakt van dynamische 256K Ramchips die op de Spec-print moeten worden gemonteerd op de plaats van de normale 32K-chips, en er moeten een paar aansluitingen gemaakt worden op signalen die niet op de edgeconnector aanwezig zijn.

Een probleem dat bij elke uitbreiding voor de Spectrum weer opduikt is het vinden van vrije IN/OUT adressen. Ik heb afgezien van compatibiliteit (kwa besturing) met de SPECTRUM-128, omdat de 8 Rambanken elk 32K groot zijn en het alternatief zou zijn geweest 16 banken van 16K en dit zou nogal wat extra chips gevergd hebben. Gekozen is om de Rambanken te schakelen dmv de kommando's IN 0 t/m 7, en de Rombanken dmv IN 8 t/m 10 en IN 12 t/m 14.

Een ander probleem was het refreshen van zoveel geheugen, de architectuur van de Spectrum is daar niet op ingericht. Een ontwerp-idee uit Happy Computer bracht uitkomst, adreslijn 7 van de RAMchips wordt tijdens de refreshpuls aangesloten op een oscillator die deze lijn beurtelings hoog en laag maakt.

Voor de ROM's ben ik uitgegaan van 32K EPROMs waarvan voor elke bank de helft gebruikt wordt. Overigens kunnen ook 16K- en 8K EPROMs gemonteerd worden, deze bevinden zich dan onder de hoge bankno's. Uiteraard is het ook mogelijk de originele SPECTRUM-ROM te plaatsen, die zich dan (vanwege z'n extra ROMSELEKT aansluiting) onder de lage bankno's laat aanspreken.

Hier volgt de beschrijving van het schema.

IC1 (een comparator) decodeert IN-opdrachten waarbij de waarde op de ADRES-lijnen kleiner is dan 16. IC2a decodeert verder voor ADRESsen groter of kleiner dan 8. Afhankelijk van dit laatste onderscheid wordt IC3 of IC4 (latch) geactiveerd, die dan de waarden van ADRES-lijnen 0-2 opslaat.

Decoder IC6a kiest dan de juiste (EP)ROM mbv de door A0 en A1 vastgelegde bits, en het door A3 vastgelegde bit bepaalt of de hoge of de lage helft werkzaam zal zijn. Een extra voorziening is nodig om de gekozen ROMbank te kunnen uitschakelen als Interface 1 of (in mijn geval) Opus Discovery in actie komt. Dit wordt verzorgd door IC6a die hier als NANDpoort fungeert.

Welke RAMbank actief is wordt bepaald door de in IC4 vastgelegde waarde, deze wordt gebruikt door de multiplexer IC5 die als het ware een verlengstuk vormt van de multiplex-adressering van de SPECTRUM zelf.

IC7 is geschakeld als oscillator (frequentie ergens rond de 500 Herz) en deze verzorgt samen met IC2b, D1, en D2 het "knippen" van A7 tijdens het refreshen. Om dit mogelijk te maken moet er wel een voorrangskwestie opgelost worden, dit kan met behulp van een weertandje a la Clive (R2). Deze moet (op de SPEC-print) opgenomen worden in de A7-lijn van de nieuwe RAMchips.

Verder is er voorzien in een automatische reset van de latches bij het inschakelen. De timing (C1 en R1) hiervan is tamelijk kritisch wanneer de Opus Discovery aangesloten is.

Wenken mbt het nabouwen.

De hierbij afgebeelde layout van de benodigde hulp-printplaat spreekt (hoop ik) voor zichzelf, de IC's worden rechtstreeks op de print gesoldeerd en voor de (EP)ROM voeten moeten exemplaren met gedraaide bussen gebruikt worden, ivm het solderen aan de bovenzijde van de print.

Dan nu puntsgewijs de werkzaamheden die op de SPEC-printplaat uitgevoerd moeten worden.

a. De 8 chips van het 32K geheugenblok uitsolderen, IC-voetjes insolderen, hier 8 chips van het type 41256 plaatsen nadat eerst pen 1 van deze chips omhoog gebogen is. Deze pennen komen dus niet in de voetjes, maar worden met elkaar verbonden en op aansluiting A8 vd hulpprint aangesloten.

b. De draadbrug tbv de H-L bankselekt verwijderen, de TI-OKI draadbrug moet over OKI. De vrije TI aansluiting (het dichtst bij de MIC-plug) verbinden met aansluiting BANK.

c. Pen 1 van IC26 (R/C selekt) verbinden met aansluiting SELEKT.

d. De verbinding van IC26 pen 9 met de SPECprint onderbreken, en deze onderbreking overbruggen met een weerstand van 470 ohm. (Dit is R2 in het schema.) De printplaatzijde van deze weerstand verbinden met de aansluiting A7bis.

e. Ioreq, Rd, Refresh, en eventueel Wr, (te vinden in



de buurt van de edgeconnector) verbinden met de betreffende aansluitingen op de hulpprint.

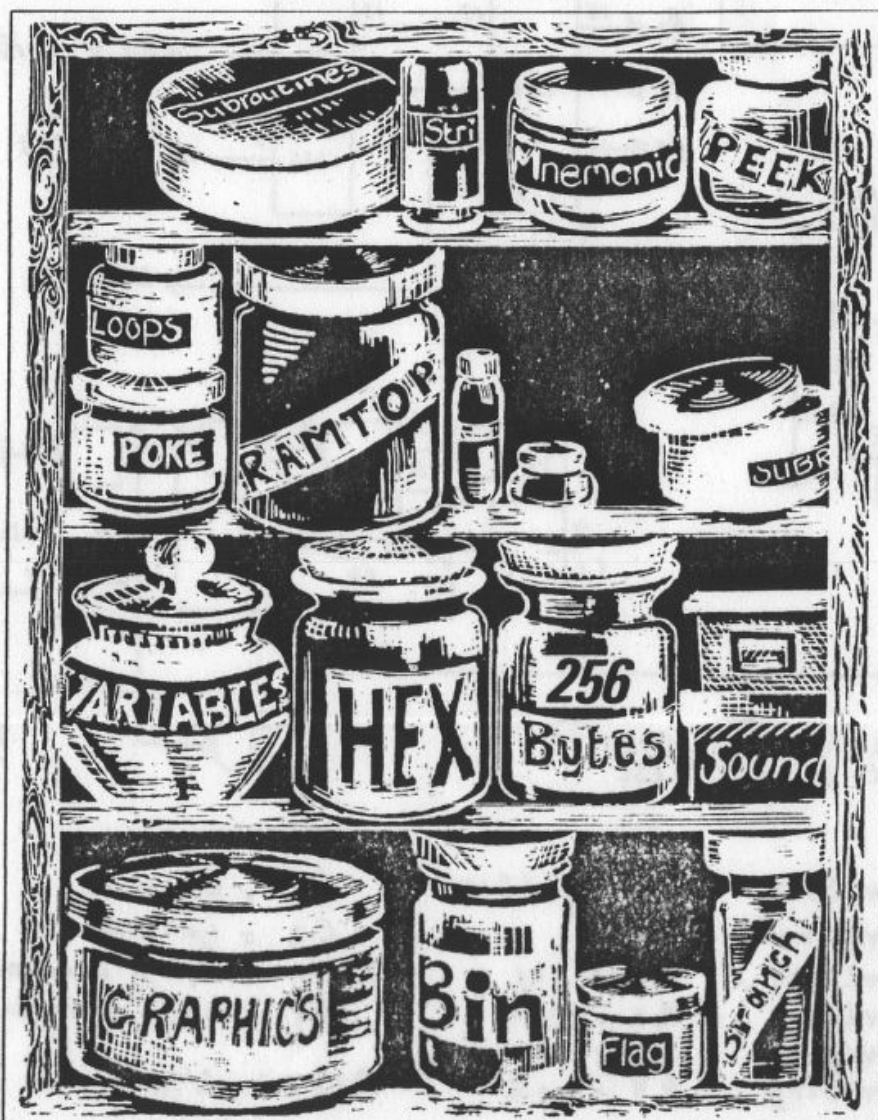
f. De SPEC-ROM (voorzichtig!) uitsolderen, hier een Wirewrap IC-voet zodanig insolderen dat deze ca. 1cm. boven de print uitsteekt. (Een normale voet kan ook, deze dan ophogen door er een extra voet op te steken.)

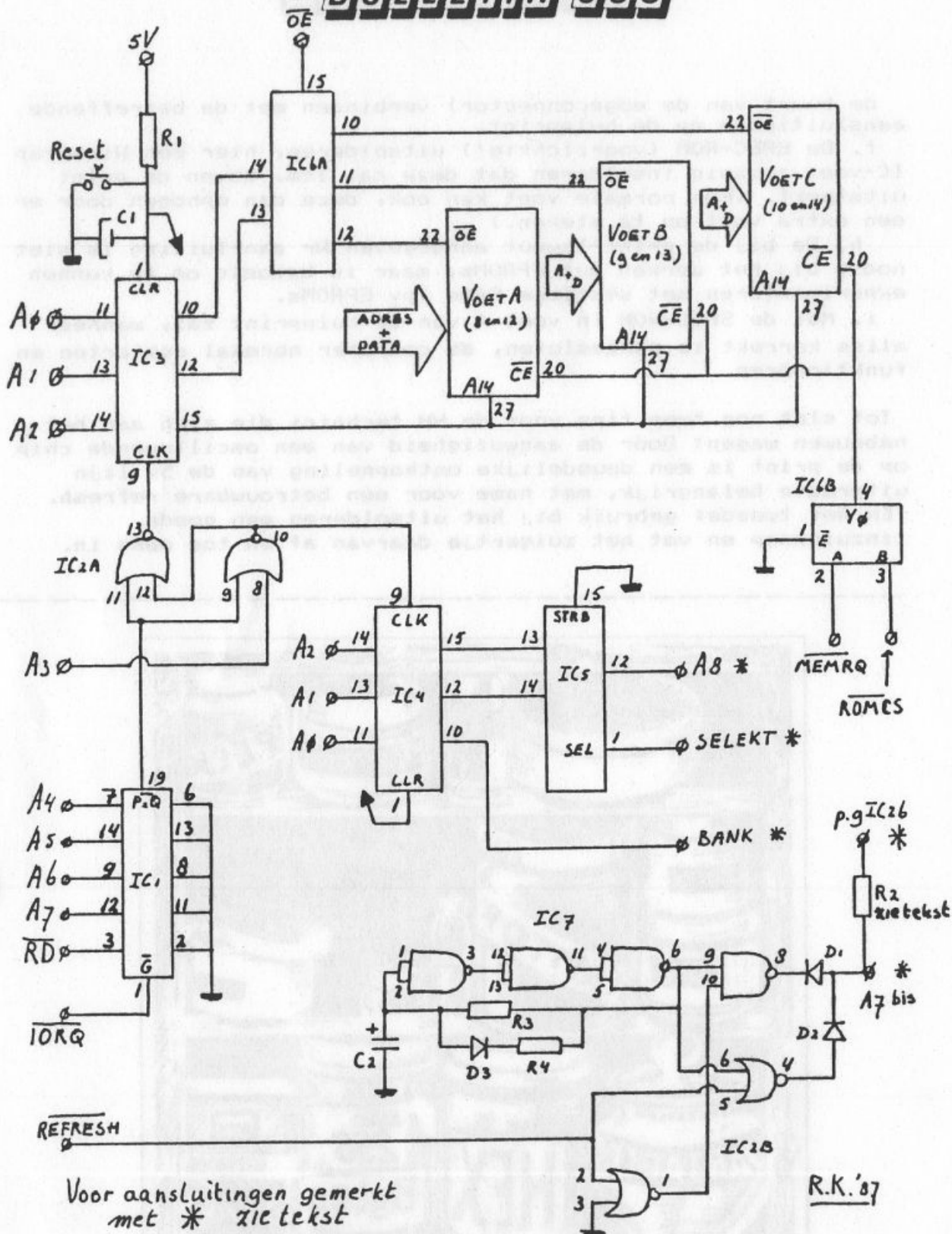
h. De bij de print-layout aangegeven Wr aanrluiting is niet nodig bij het werken met EPROMs, maar is bedoeld om te kunnen experimenteren met statiese RAMs ipv EPROMs.

i. Met de SPEC-ROM in voet A van de hulpprint zal, wanneer alles korrekt is aangesloten, de computer normaal opstarten en functioneren.

Tot slot nog twee tips voor de HH technici die zich aan het nabouwen wagen: Door de aanwezigheid van een oscillerende chip op de print is een deugdelijke ontkoppeling van de 5V lijn uitermate belangrijk, met name voor een betrouwbare refresh.

En het tweede: gebruik bij het uitsolderen een goede tinzuigpomp en vet het zuigertje daarvan af en toe eens in.



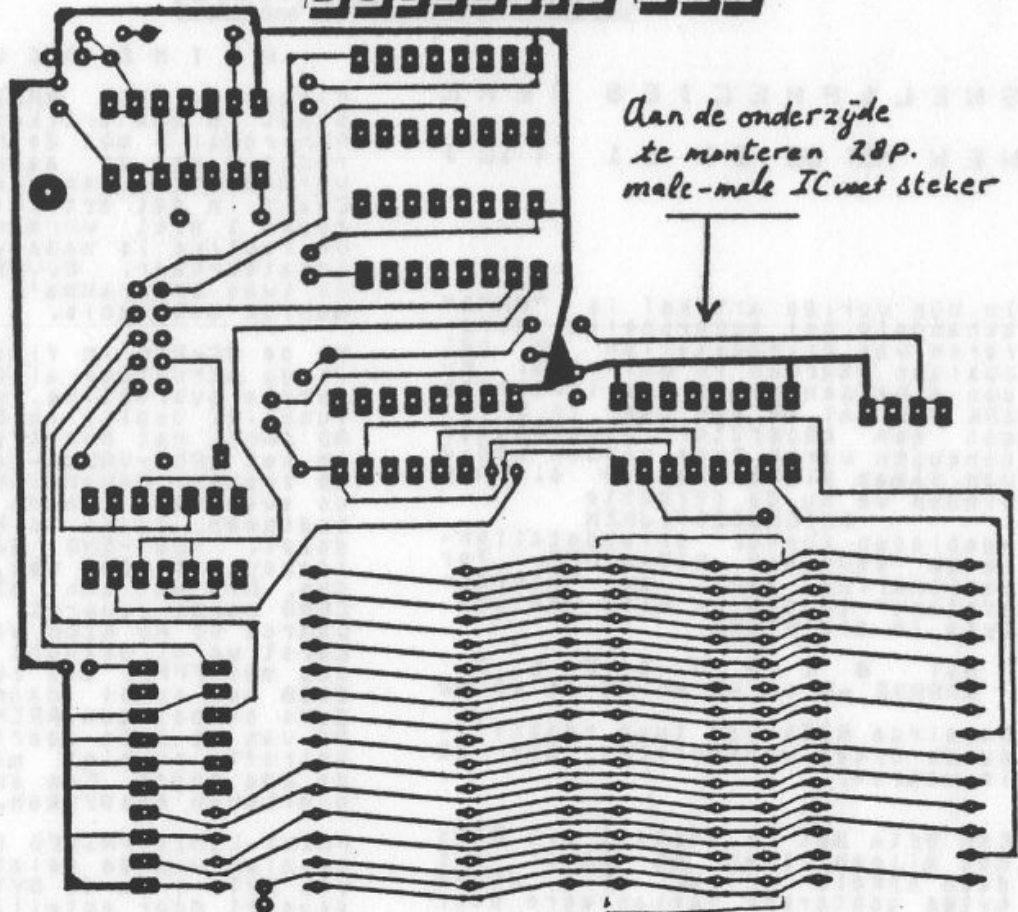
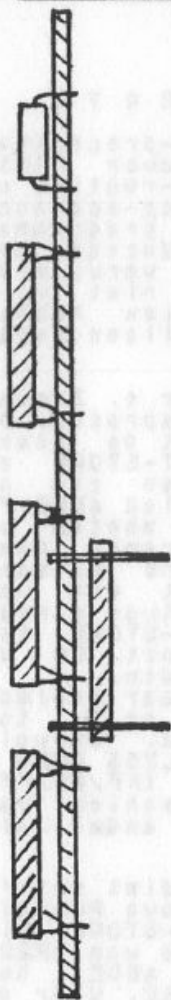


IC1 = 74LS68  
 IC2 = 74LS02  
 IC3 = 74LS174  
 IC4 = 74LS174  
 IC5 = 74LS157  
 IC6 = 74LS139

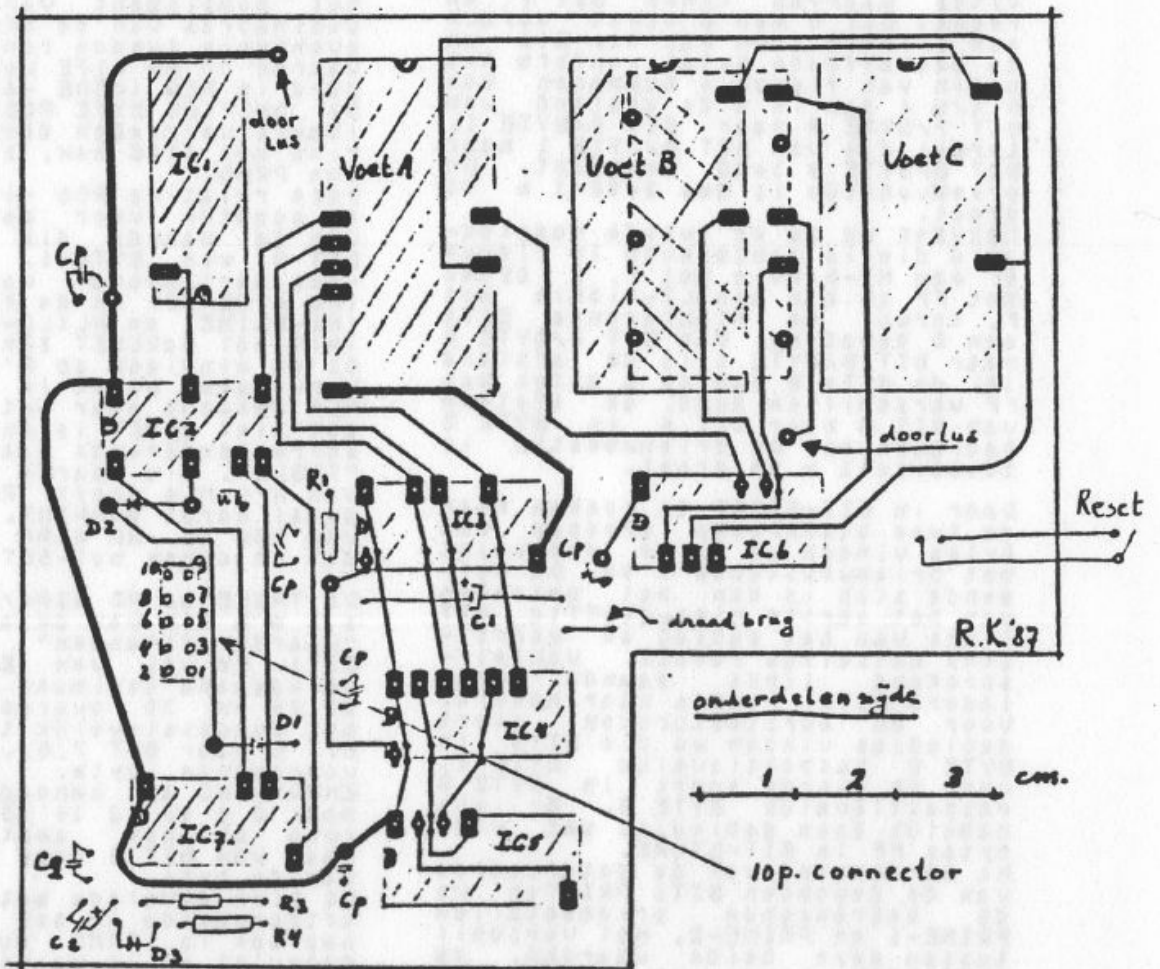
IC7 = 74LS132  
 R1 = 2k $\Omega$   
 R2 = 470 $\Omega$   
 R3 = 1k  
 R4 = 100 $\Omega$

C1 = 0,5 $\mu$   
 C2 = 1 $\mu$  tantaal  
 5x Cp = 0,01 $\mu$  keramiek zie print  
 Cg = 0,5 $\mu$  keramiek " "  
 D1,2,3 = 1N4148





- 1 - A8
- 2 - SELEKT
- 3 - BANK
- 4 - IORQ
- 5 - n.c.
- 6 - RD
- 7 - WR
- 8 - RESET
- 9 - n.c.
- 10 - A7 bis



### SNEL & PRECIES REKENEN MET DE ZX 81 (12)

In mijn vorige artikel is "PRMS" behandeld het supersnelle genereren van priemgetallen en het opslaan daarvan in BIT-vorm. Bij een geheugenruimte van tenminste 35K kon dat in een keer in 5 s. met een beperkter beschikbare geheugen waren twee ronden nodig van samen minder dan 8 s. Hoe vinden we nu de gezochte PRIEMWOESTIJNEN -gebieden zonder priemgetallen- in de "gevolde" BIT-STORE. Ter herinnering eerst de "waarde-afstand" tussen de BITS van een byte in die STORE:

BIT	0	1	2	3	4	5	6	7
WAARDE	+8	+4	+2	+4	+2	+4	+6	+2

Dezelfde BITS van twee naastliggende bytes verschillen dus 30 in waarde.

Een byte met FF bestaat uit BITS met alleen 1-en en bevat dus geen enkele priem. Drie zulke bytes achtereenvolgend geflankeerd door bytes waarvan links BIT 7 en rechts BIT 0 een 0 bevat vormen een priemwoestijn van 91. Als we de betreffende bytes konform het begin van figuur 4 nummeren van 0 t/m 4 dan is 2 de afstand van BIT 7/BYTE 0 naar BIT 0/BYTE 1, terwijl die van BIT 0/BYTE 1 naar BIT 0/BYTE 4 3\*30 bedraagt. De priemwoestijn is dus  $2+90-1 = 91$  groot. Bekijken we nu de tweede mogelijkheid die is aangeduid in figuur 4: een HI-nibble met F, 2 bytes met FF en dan een LO-nibble met F, terwijl de flankerende BITS een 0 bevatten. Van BIT 3/BYTE 0 naar BIT 0/BYTE 1 is de afstand 18; de BITS 0 van de 2 bytes met FF verschillen  $2*30$ ; de afstand van BIT 0 naar BIT 4 in BYTE 3 bedraagt 16. De priemwoestijn is  $18+60+16-1 = 93$  groot.

Door in BIT-STORE te zoeken naar de twee beschreven groepen van bytes vinden we alle gebiedjes met priemwoestijnen > 90. De volgende stap is dan het opzoeken van het eerste niet-geSETte BIT links van het gebied en vervolgens hetzelfde rechts; vanzelfsprekend links gaande naar lagere BITS, rechts naar hogere. Voor de eerstbesproken soort gebiedjes vinden we die BITS in BYTE 0 respectievelijk BYTE 4, voor de tweede soort in BYTE 0 respectievelijk BYTE 3. Er zijn namelijk geen gebiedjes met vier bytes FF in BIT-STORE. Na berekening van de getalwaarde van de gevonden BITS PRINTEN we de bedrenzende priemgetallen PRIME-1 en PRIME-2. Het verschil tussen deze beide waarden, de AREA-SIZE, wordt ook bepaald en gePRINT.

Figuur 1, het BASIC-programma, staat in mijn artikel over PRMS. REM-regel 3 met de MC-routine en regel 7 met de aanroep-opdracht vormen het eigenlijke programma. Zoals in dat artikel gezegd mag regel 1 niet worden verwijderd. De routine is namelijk niet vrijlokateerbaar. Bovendien hebben de twee programma's alleen gezamenlijk betekenis.

Nu de MC-RTN in figuur 4. Zoveen is de structuur al besproken. De eerste subroutine, met de zoek-functie, begint in BIT-STORE op AD 4600. Het heeft geen zin om in het BASE-VALUE-gebied 45DE/FF te zoeken. Bovendien moeten we de eventuele tweede ronde daar beginnen, zoals in PRMS uiteengezet. TEST-END gaat uit van zoeken tot 6800 hex, twee rondes dus. Als met een BIT-STORE tot C900 wordt gewerkt, moet C9 de waarde op AD 415B worden. Eerst wordt gezocht naar gebiedjes met FFFF; dat zijn er 21 tot 6800 hex en 54 daarna. Vervolgens of het een PRIMELESS AREA > 90 van de 1ste soort (FF/FF/FF) betreft, zo niet misschien van de 2de soort. Een en ander als hierboven besproken.

PRINT-LIMIT-PRIMES begint met 't bepalen van de relatieve positie van BYTE-1 in de BYTE-STORE. Dit gebeurt door optelling van BA22, het komplement van 45DE, het beginadres van de STORE. Voor de eventuele tweede ronde moet de waarde op AD 417E worden veranderd in NEG (45DE - 4100) = FB22. De SHIFTED BYTE POSITION wordt immers verkregen door vermindering met 4100 hex, zie figuur 2 van PRMS.

Deze relatieve POS -in BC- vormt de counter voor de berekening van de waarde die behoort bij BIT 0 van BYTE-1. Die waarde -een dertigvoud- wordt gevormd (AD 4185/98) in de registers A' (HI-PRIME) en HL (LO-PRIME), die initieel geRESET zijn op 00 resp. 01 en eindigen op A'0 resp. HLO. Vervolgens wordt in BYTE-0 met RLA gezocht naar het eerste BIT dat niet geSET is en de daarmee korresponderende waarde (DE) in "TABLE". Die waarde afgetrokken van A'0, HLO geeft PRIME-1. Dat getal wordt gePRINT. Ten behoeve van SBC op AD 41AA is gezorgd dat de carry not-SET is.

De TABLE op AD 4204/13 bevat de 30 m m e n van de al besproken "waarde-afstanden" tussen de BITS. En wel van ENTRY-DEC af teruggaand decimaal 2 8 12 14 18 20 24 en 30 overeenkomend met het successievelijk teruggaan van BIT 0 naar BIT 7, 6, ..., 0 van de voorgaande byte. Evenzo van ENTRY-INC af omhooggaand decimaal 0 6 10 12 16 18 22 28 behorend bij het achtereenvolgens gaan van BIT 0 naar BIT 0, 1, ..., 7 van de byte. De hier gevolgde methode met een aftrekwaarde of een bijtelwaarde had ook in PRMS kunnen worden gebruikt en is waarschijnlijk korter en sneller, maar dat heb ik (nog) niet onderzocht.



Voor de bepaling van PRIME-2 wordt opnieuw uitgegaan van de waarde A'0, HLO. Gezoocht wordt nu de eerste byte hoger dan BYTE-1 die niet FF is. Per byte met waarde FF wordt 30d bijgeteld. Vervolgens wordt met RRA gezocht naar het eerste BIT dat niet ge-SET is. De daarmee korresponderende waarde (DE) gevonden via ENTRY-INC in de TABLE wordt ook bijgeteld en dat geeft PRIME-2. Dit getal wordt wederom gePRINT.

AREA-SIZE. Dankzij de instructie duplicate in de subroutine PRINT-VAL staan de waarden PRIME-1 en PRIME-2 nog op de CALCULATOR STACK. Na omwisseling worden ze van elkaar afgetrokken verminderd met 1 en gePRINT. Na New Line gaan we verder met het zoeken van FFFF-gebiedjes, beginnend met de byte < > FF gestacked op AD 41C7. Omdat de sprong naar AD 4159 te groot is, maken we gebruik van de instructie op AD 417B en het feit dat RST 10 altijd NZ eindigt.

PRINT VALUE 6-HEXDIGITS NUMBERS. Deze subroutine berekent met behulp van de CALCULATOR:  $65536 * HI-PRIME + LO-PRIME$ , de waarde van PRIME-1 of PRIME-2. Die waarden worden niet alleen gePRINT maar ook bewaard op de STACK. Zoals bij de voorgaande subroutine behandeld kan de grootte van de PRIEMWOESTYNNEN daarmee worden bepaald.

## R E S U L T A T E N

In figuur 5 staat de PRINTOUT van de 6 PRIME DESERTS > 90 met daarbij de twee oplossingen van het HCC PROBLEEM. Tevens in dezelfde volgorde de korresponderende bytes in BIT-STORE voor zover relevant. Merkwaaardig is dat de drie grootste woestijnen alle in het "eerste ronde"-deel voorkomen! Ook dit programma is zeer snel. Vrijwel alle tijd gaat zitten in de ADD-30 loop (AD 418E/98) n.l. gemiddeld 0.3 s/woestijn en het gebruiken van de FPA-RTNs. De PRINTOUT van de eerste 3 woestijnresultaten kost 2.0 s, die van de tweede 3 -eventueel in de 2de ronde- 2.7 s.

De volgende keer behandel ik het zeer snel vinden en uitPRINTen van een reeks priemgetallen beginnend bij een van tevoren gekozen RANGNUMMER.

H A N V A N A B B E

FIG. 4 MC-ROUTINE PRIME DESERTS

```

FIND PRIMELESS AREA >90 :
.. /FF/FF/FF/.. OR F./FF/FF/.F
BYT 0 1 2 3 4 0 1 2 3

16726
FIND-FF/FF
TEST-END
4155--210045 LD HL,4600 ENTRY BIT-STORE
4159--7C LD A,H
415A--FE88 CP 88 END BIT-STORE?
415C--D0 RET NC TO BASIC if YES
415D--56 LD D,(HL) HL = BYTE-1
415E--23 INC HL
415F--5E LD E,(HL) HL = BYTE-2
4160--13 INC DE DE if DE was FFFF
4161--7A LD A,D DE = DE?
4162--B3 OR E
4163--20F4 JR NZ,4159 FIND-FF/FF if not
4165--54 LD D,H DE = HL = BYTE-2
4166--5D LD E,L
4167--13 INC DE
4168--1A LD A,(DE) DE = BYTE-3
4169--FEFF CP FF
416B--2810 JR Z,417D PRIME-1 if A=FF
416D--E60F AND 0F
416F--FE0F CP 0F 10-NIBBLE A = F?
4171--20E6 JR NZ,4159 FIND-FF/FF if not
4173--1B DEC DE
4174--1B DEC DE
4175--1B DEC DE
4176--1A LD A,(DE) DE = BYTE-0
4177--E6F0 AND F0
4179--FEF0 CP F0 11-NIBBLE A = F?
TO-FIND-FF/FF 417B--20DC JR NZ,4159 FIND-FF/FF if not
    
```

# BULLETIN SGG

## PRINT LIMIT-PRIMES + SIZE AREA

<b>PRIME-1</b>	417D--2B	DEC HL	HL = BYTE-1
	417E--1122BA	LD DE,BA22	NEG 45DE
	4181--E5	PUSH HL	stack BYTE-1
	4182--19	ADD HL,DE	
	4183--44	LD B,H	BC = relative position
	4184--4D	LD C,L	in BYTE-STORE
	4185--210100	LD HL,0001	RES LO-PRIME
	4188--111E00	LD DE,001E	30d
	4188--08	EX AF,AF"	
	418C--AF	XOR A	RES HI-PRIME
	418D--08	EX AF,AF"	carry F' not SET
<b>ADD-30</b>	418E--19	ADD HL,DE	LO-PRIME 30d up
	418F--3003	JR NC,4194	NO-INC-HI if NC
	4191--08	EX AF,AF"	
	4192--3C	INC A	INC HI-PRIME
	4193--08	EX AF,AF"	carry F' not SET
<b>NO-INC-HI</b>	4194--08	DEC BC	DEC
	4195--78	LD A,B	until
	4196--B1	OR C	zero
	4197--20F5	JR NZ,418E	ADD-30 if > zero
	4199--E3	EX (SP),HL	stack LO-PRIME
	419A--2B	DEC HL	HL = BYTE-1
	419B--7E	LD A,(HL)	BIT-INFO
	419C--23	INC HL	HL = BYTE-1
	419D--E3	EX (SP),HL	stack BYTE-1
	419E--E8	PUSH HL	HL = LO-PRIME+BIT/8
	419F--110C42	LD DE,420C	ENTRY-DEC
<b>FIND-BIT-NOT-SET</b>	41A2--17	RLA	shift until NC
	41A3--1D	DEC E	E = POS in TABLE
	41A4--38FC	JR C,41A2	FIND-BIT-NOT-SET
	41A5--1A	LD A,(DE)	A = value in TABLE
	41A7--4F	LD C,A	B = 00 : BC = A
	41A8--08	EX AF,AF"	carry F' not SET
	41A9--F5	PUSH AF	A' = HI-PRIME+BIT/8
	41AA--ED42	SBC HL,BC	HL = LO-PRIME
	41AC--3001	JR NC,41AF	NO-DEC-HI if NC
	41AE--3D	DEC A	DEC HI-PRIME
<b>NO-DEC-HI</b>	41AF--08	EX AF,AF"	
	41B0--CDEA41	CALL 41EA	PRINT-VAL
<b>PRIME-2</b>	41B3--08	EX AF,AF"	
	41B4--F1	POP AF	retrieve A'
	41B5--08	EX AF,AF"	
	41B6--E1	POP HL	retrieve HL
	41B7--C1	POP BC	retrieve BYTE-1
<b>FIND-BYTE ≠ FF</b>	41B8--03	INC BC	BC = BYTE-2 or higher
	41B9--0A	LD A,(BC)	BIT-INFO
	41BA--FEFF	CP FF	Z if A = FF
	41BC--111E00	LD DE,001E	30d
	41BF--19	ADD HL,DE	LO-PRIME 30d up
	41C0--3003	JR NC,41C5	NOT-INC-HI if NC
	41C2--08	EX AF,AF"	
	41C3--3C	INC A	INC HI-PRIME
	41C4--08	EX AF,AF"	Z-flag SET by CP FF?
<b>NOT-INC-HI</b>	41C5--28F1	JR Z,41B8	FIND-BYTE ≠ FF if yes
	41C7--C5	PUSH BC	(BC) ≠ FF
	41C8--42	LD B,D	B = D = 00
	41C9--110B42	LD DE,420B	ENTRY-INC
<b>FIND-NOT-SET-BIT</b>	41CC--1F	RRR	shift until NC
	41CD--1C	INC E	E = POS in TABLE
	41CE--38FC	JR C,41CC	FIND-NOT-SET-BIT
	41D0--1A	LD A,(DE)	A = value in TABLE
	41D1--4F	LD C,A	BC = A
	41D2--09	ADD HL,BC	HL = LO-PRIME
	41D3--3003	JR NC,41D8	INC-NOT-HI if NC
	41D5--08	EX AF,AF"	
	41D6--3C	INC A	INC HI-PRIME
	41D7--08	EX AF,AF"	
<b>INC-NOT-HI</b>	41D8--CDEA41	CALL 41EA	PRINT-VAL
<b>AREA-SIZE</b>	41DB--EF	RST 28;FPA:	
	41DC--01/03/A1/03/34	CALL 15DB	exch-/0/-/end
	41E1--C0DB15	LD A,76	PRINT-FF
	41E4--3E76	RST 10	N/L; RST ends NZ
	41E6--D7	POP HL	(HL) ≠ FF
	41E7--E1	JR 417B	TO-FIND-FF/FF
	41E8--1891		



# BULLETIN SGG

## PRINT VALUE 6-HEXDIGITS NUMBERS

PRINT-VAL	41EA--44	LD	B,M	
	41EB--4D	LD	C,L	
	41EC--CD2015	CALL	1520	STK-B
	41EF--08	EX	AF,AF"	
	41F0--CD1D15	CALL	1510	STK-A'
	41F3--08	EX	AF,AF"	
	41F4--EF	RST	28:FPA:	
	41F5--30 00 41	00/04/0F/2D/34	b555/44/dupl/end	
	41FD--CDD815	CALL	15DB	PR/NT-ZP
	4200--AF	XOR	A	
	4201--D7	RST	10	PRINT
	4202--D7	RST	10	2 spaces
	4203--C9	RET		

## TABLE

ENTRY-DEC	4204--1E	18	14	12	0E	0C	08	02	ENTRY-INC
	420C--00	06	0A	0C	10	12	16	1C	

FIG. 4A HEXDUMP PRIME DESERTS

4156--2100	467C	FE88	D856--38F
415E--235E	137A	B320	F454--329
4166--5D13	1AFA	FF28	10E8--3A5
416E--0FFFE	0F20	E61B	1B1B--827
4176--1AE6	F0FE	F020	DC28--1285
417E--1122	BAE5	1944	4D21--869
4186--0100	111E	0008	AF08--239
418E--1930	0308	3C08	0B78--283
4196--B120	F5E3	2B7E	23E3--1112
419E--E511	0C42	171D	38FC--584
41A6--1A4F	08F8	ED42	3801--710
41AE--3D08	CDEA	4108	F108--830
41B6--E1C1	030A	FEFF	111E--987
41BE--0019	3003	083C	0828--192
41C6--F1C5	4211	0842	1F1C--857
41CE--38FC	1A4F	0938	0308--481
41D6--3C08	CDEA	41EF	0103--815
41DE--A103	34CD	D815	3E76--841
41E6--D7E1	1891	444D	CD20--991
41EE--1508	CD1D	1508	EF30--579
41F6--0041	0004	0F2D	34CD--385
41FE--DB15	AFD7	D7C9	--1048

4204--1E18	1412	0E0C	0802--080
420C--0006	0A0C	1012	161C--078

16726-2100	467C	FE88	D856- 911
16734-235E	137A	B320	F454- 809
16742-5D13	1AFA	FF28	10E8- 933
16750-0FFFE	0F20	E61B	1B1B- 827
16758-1AE6	F0FE	F020	DC28-1285
16766-1122	BAE5	1944	4D21- 869
16774-0100	111E	0008	AF08- 239
16782-1930	0308	3C08	0B78- 283
16790-B120	F5E3	2B7E	23E3-1112
16798-E511	0C42	171D	38FC- 584
16806-1A4F	08F8	ED42	3801- 710
16814-3D08	CDEA	4108	F108- 830
16822-E1C1	030A	FEFF	111E- 987
16830-0019	3003	083C	0828- 192
16838-F1C5	4211	0842	1F1C- 857
16846-38FC	1A4F	0938	0308- 481
16854-3C08	CDEA	41EF	0103- 815
16862-A103	34CD	D815	3E76- 841
16870-D7E1	1891	444D	CD20- 991
16878-1508	CD1D	1508	EF30- 579
16886-0041	0004	0F2D	34CD- 385
16894-DB15	AFD7	D7C9	-1048

16900-1E18	1412	0E0C	0802- 128
16908-0006	0A0C	1012	161C- 112

FIG. 5 PRIME DESERTS > 90

## PRINTOUT

370261	370373	111
396733	396833	99
492113	492227	113
604073	604171	97
860143	860239	95
927869	927961	91

## CORRESPONDING BYTES IN BIT-STORE

AD 7614	FE/FF/FF/BF
7986	F7/FF/FF/BF
85F1	B7/FF/FF/FF/EF
9485/5365	95/FF/FF/FF/
85DD/74DD	F7/FF/FF/DF/
BEAE/7DAE	/FF/FF/FF/

## REKTIKATIES REKENEN ZX 81 (11)

Uit de 10de regel tekst onder de tabel met het begin van de BIT-STORE moet het woord "niet" worden weggelaten.

In de laatste subroutine van figuur 2

BIT SETTING AT BYTE POSITION heb ik een foutje gemaakt in het middelste deel. Dat gedeelte is alleen nodig bij een beperkt beschikbaar geheugen, zoals ik heb uiteengezet; de fout treedt dan op bij de tweede ronde. De foute routine gaat na het BIT-SETTEN van een BASE-VALUE onmiddellijk de BYTE-POSITION verlagen met 4100 hex en werkt dan verder met deze "deshifte" waarde. Vanzelfsprekend moet dat pas gebeuren als  $H \geq 46h$  is. Door deze fout worden van 7 deelbare getallen de korresponderende BITS in het BASE-VAL gebied niet geSET. Overigens zonder desastreuze gevolgen voor deel (12) van de serie. Wel achter voor deel (13), vandaar deze rektifikatie. De verbeterde figuur 1, de korrekte subroutine van figuur 2 en het betreffende deel van de hexdump van figuur 2A volgen hieronder.

Een derde foutje betreft de laatste regel van figuur 3. Bij mijn beperkte geheugenruimte van 18K staat daar hetzelfde als in de regel beginnend met 8000 hex. De verbeterde figuur 3 volgt hieronder.

H A N V A N A B B E

FIG. 1 PRMS: GENERATION AND BIT-STORAGE PRIMES <1E6 + FINDING PRIME DESERTS

```
1 REM-REGEL 206 BYTES
3 REM-REGEL 174 + 16 BYTES
5 RAND USR 16514
7 RAND USR 16726
```

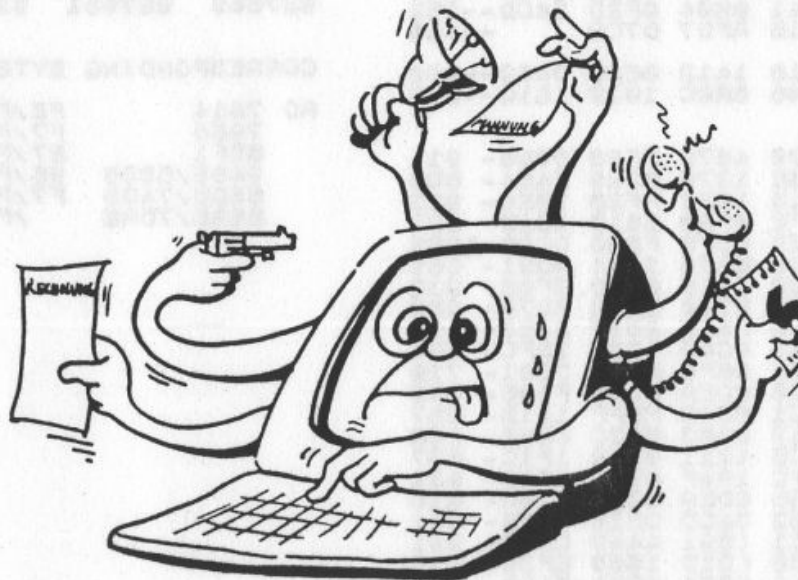




Fig. 2

## BIT SETTING AT BYTE POSITION

<b>SETTING</b>	4119--78	LD	A,B	A=B= BYTE-COUNT VAL
	411A--D9	EXX		
	411B--5F	LD	E,A	BYTE-COUNT VALUE
	411C--1600	LD	D,00	
	411E--19	ADD	HL,DE	H'L= BYTE-POS
	411F--E5	PUSH	HL	stack
	4120--D9	EXX		
	4121--E1	POP	HL	retrieve BYTE-POS
	4122--D1	POP	DE	retrieve BASE-VAL
<b>SET-BASE-VAL</b>	4123--1814	JR	4139	NOT - SHIFTED
	4125--7C	LD	A,H	
	4126--FE46	CP	46	carry if H < 46 h
	4128--3006	JR	NC,4130	NO - BASE-VAL
	412A--7E	LD	A,(HL)	
	412B--81	OR	C	SET-BIT
	412C--77	LD	(HL),A	
	412D--19	ADD	HL,DE	
	412E--18F5	JR	4125	SET-BASE-VAL
<b>NO-BASE-VAL</b>	4130--D641	SUB	41	shift BYTE-POS
	4132--67	LD	H,A	with 4100 hex
	4133--3E45	LD	A,45	
<b>INC-POSITION</b>	4135--19	ADD	HL,DE	add BASE-VAL
	4136--BC	CP	H	until H > 45 h
	4137--30FC	JR	NC,4135	INC-POSITION
<b>NOT-SHIFTED</b>	4139--41	LD	B,C	B=C, BIT to be SET
<b>MAKE SET-CODE</b>	413A--3EBE	LD	A,BE	= C6-08
	413C--C608	ADD	A,08	
	413E--CB08	RRC	B	
	4140--30FA	JR	NC,413C	MAKE SET-CODE
	4142--324841	LD	(4148),A	POKE SET-CODE
<b>CONT-SETTING</b>	4145--3E97	LD	A,97	
	4147--CBC6	SET	0,(HL)	C6...FE: BIT 0...7
	4149--19	ADD	HL,DE	add BASE-VAL
	414A--BC	CP	H	until H > 87 h
	414B--30FA	JR	NC,4147	CONT-SETTING
	414D--62	LD	H,D	
	414E--6B	LD	L,E	HL=DE=BASE-VAL
	414F--C9	RET		

Fig. 2A

4120--D116	147C	FE46	3006--2F3
412A--7EB1	7719	18F5	0641--3E3
4130--673E	4519	8BC30	FC41--32C
413A--3EBE	C608	CB08	30FA--3C7
4140--3248	413E	87CB	0619--32A
414A--8C30	FA62	8BC9	--37C
16674-D116	147C	FE46	3006--755
16682-7EB1	7719	18F5	0641--995
16690-673E	4519	8BC30	FC41--812
16698-3EBE	C608	CB08	30FA--957
16706-3248	413E	87CB	0619--810
16714-8C30	FA62	8BC9	--802

FIG. 3 ENKELE REGELS FIT-STOFF

45DE--00	20	10	81	49	24	C2	06
45E8--2A	B0	E1	0C	15	59	12	61
45EE--19	F3	2C	2C	C4	22	A6	5A
760C--F6	E9	BA	BB	5D	3F	E5	EB
7614--FE	FF	FF	BF	DD	62	BA	64
761C--7B	ED	FA	15	FB	33	F6	F7
85E9--AC	DF	F7	7B	F5	65	2F	AC
85F1--B7	FF	FF	FF	EF	E3	3B	DF
85F9--FF	6F	95	FA	E7	F9	F6	BB
87F0--FA	FB	8B	F7	EF	1E	FD	FA
87F8--F5	3C	BF	7D	2E	6B	FF	F7
8800--CC	BD	DA	5F	E6	BF	A9	DF
8808--DF	EE	37	75	F3	F9	79	A7

LAATSTE 2 REGELS OP AD 4700 4708  
ALS SHIFTED SETTING IS TOEGEPAST

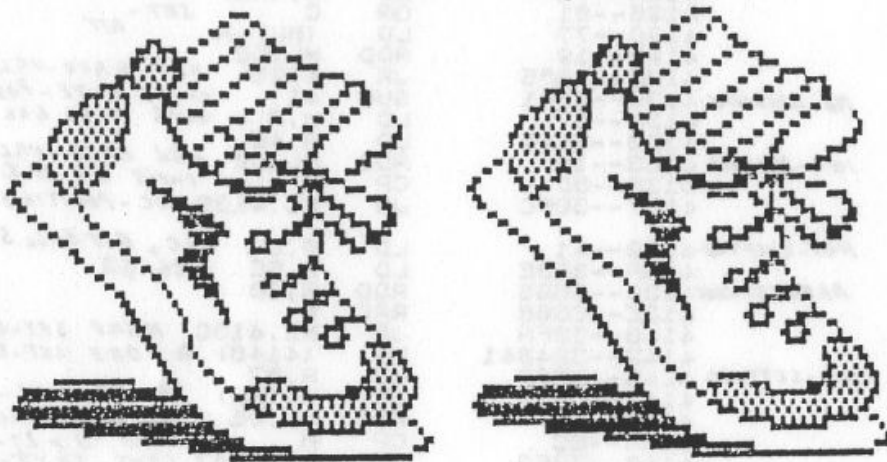
## DEMONSTRATIE VAN DE 256K RAM

Op donderdag 25 februari in Groningen.

en

op donderdag 11 februari in Assen.

Trek uw stoute schoenen aan en kom dit zien.



## KAPOT ???? OF DEFECT

U kunt op de gebruikersavonden aan de tafel bij de deur Uw kapotte of defecte computerspullen aanbieden ter reparatie. Dit wordt dan zo mogelijk gerepareerd tegen kostprijs cq kleine vergoeding.

MOCHT U IN DE DECEMBER MAAND GEEN BULLETIN ONTVANGEN HEBBEN WILT U DAN EVEN CONTACT OPNEMEN MET EEN VAN DE BESTUURSLEDEN. ER IS DAN IETS MIS GEGAAN WAARVOOR WIJ ONS VERONTSCHULDIGEN. SORRY VOOR HET ONGEMAK.

DE RED.

## contributie

Een kleine herinnering voor onze leden die in december vergeten zijn de contributie te storten op giro nr 5212298 t.n.v. SGG.

stort dus dat geeltje

zodat we plannen kunnen maken in dit nieuwe jaar.



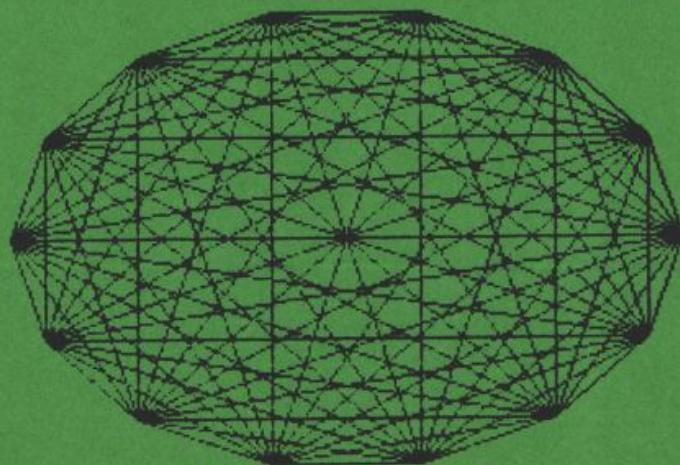
# ESGEEGEETJES



-----  
Een snelle reactie van een lezer van ons bulletin.  
Hier het adres van een zaak waar men een 8 polige din steker  
kan kopen:

YPMA'S  
RADIO ONDERDELEN  
BOVEN OOSTERDIEP 61  
VEENDAM.  
TEL. 05987 - 17458.

-----  
Te koop aangeboden:  
Tractorfeed voor een commodore printer, één pootje is gebroken.  
Wie kan mij helpen aan een tractorfeed voor de Smith-Carona.  
tel: 050 - 263930.





**DRUKWERK**

**PORT BETAALD**  
Groningen

Alz.:

SGG  
redaktieladres

F. Elstrodt  
Kam. Onnesstraat 172  
9727 HS Groningen

AAN: