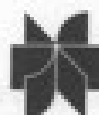


Praktijkboek voor de
ZX SPECTRUM

Luc Smeesters

Praktijkboek voor de ZX Spectrum



KLUWER TECHNISCHE BOEKEN B.V. - DEVENTER - ANTWERPEN

Vertaling: H.E. Leydens - Bartray
Lay-out: W. Wijnolts

ISBN 90 201 17041
D/1983/0108/221
Oorspronkelijke titel: Découvrir le Spectrum

© Edition DIDECAR, Didecar Marketing SPRL
Rue du Planiau, 1-B 1301 Wavre - Belgique
© 1983/1984 van de Nederlandse vertaling bij Kluwer Technische Boeken B.V. Deventer

1e druk 1983
2e oplage 1984

Niets uit deze uitgave mag worden verveelvoudigd en/of openbaar gemaakt door middel van druk, fotokopie, microfilm of op welke andere wijze ook, zonder voorafgaande schriftelijke toestemming van de uitgever.

No part of this book may be reproduced in any form, by print, photoprint, microfilm or any other means without written permission from the publisher.

Ondanks alle aan de samenstelling van de tekst bestede zorg, kan noch de redactie noch de uitgever aansprakelijkheid aanvaarden voor eventuele schade, die zou kunnen voortvloeien uit enige fout, die in deze uitgave zou kunnen voorkomen.

Woord vooraf

De Club Micro Europe heet de ZX Spectrum van harte welkom op het vaste land van Europa. Als eerste computer met kleur en met geluid, opgebouwd met zeer grootschalig geïntegreerde schakeling (VLSI) en daarmee een grotere betrouwbaarheid, kan de ZX Spectrum van nu af aan worden geleverd met een groot aantal randapparaten en software-pakketten. In de loop van de zes maanden na de introductie van de computer in Engeland werden een twintigtal boeken uitgebracht, jammer genoeg voor het merendeel in het Engels. Gezien de belangstelling die er ook aan deze zijde van Het Kanaal voor de ZX Spectrum bestaat is het niet meer dan logisch dat er nu ook Nederlandstalige publikaties gaan verschijnen.

Dit boek, dat kunnen we u verzekeren, verschilt heel sterk van alle andere bekende, soortgelijke uitgaven. Het wil zowel beschrijvend als praktisch zijn. De overzichten van instructies en functies zijn zowel van toepassing op de ZX81 als op de ZX Spectrum. Ten slotte wordt nog het algemene toepassingsgebied van computers behandeld. Hierbij worden suggesties gedaan voor een rationeel gebruik van bestandsmateriaal, sorteermethoden en tekstverwerking. Wij zijn er dan ook van overtuigd dat dit boekje een onmiskenbare aanvulling is op de gebruikershandleidingen zoals die met de ZX Spectrum worden meegeleverd.

Raymond Betz
Voorzitter van Club Micro Europe

Iets over de schrijver

Wim van der Meer

Luc Smeesters is in het dagelijks leven technisch-commercieel medewerker van een multinationale onderneming op het gebied van de informatica. Van huis uit elektronicus is hij ook een verstokt radio-amateur. Als free-lance redacteur werkt hij mee aan verscheidene radio-amateurbladen evenals aan een Belgisch weekblad, speciaal ten behoeve van de micro-informatica.

Wim van der Meer
redacteur van CIB-Micro-Bureaus

Inhoud

Inleiding	9
1. De toepassingsgebieden van de Spectrum	11
2. Een vergelijking	13
2.1 ZX80.	13
2.2 ZX81.	14
2.3 ZX Spectrum	16
2.4 Functie-overzicht	18
2.5 Instructie-overzicht	20
3. Overeenkomsten	24
3.1 Analyse van de software	24
3.1.1 Berekenings- en verwerkingsinstructies	25
3.1.2 READ, DATA en RESTORE	25
3.1.3 DEF (FN); door de gebruiker gedefinieerde functie	27
3.1.4 VAL\$.	28
3.1.5 CLEAR	28
3.1.6 Aanpassen van Spectrum-BASIC aan andere BASIC-versies	28
3.1.7 Machinetaal; een inleiding	40
3.1.8 De werking van de microprocessor Z80A.	41
3.2 Analyse van de hardware	44
3.2.1 Algemeen	44
3.2.2 TV-ontvanger.	44
3.2.3 Schrijven op het beeldscherm.	45
3.2.4 Invoeren van kleur.	47
3.2.5 De grafische taal van de Spectrum.	48
3.2.6 De Spectrum muzikaal	50
3.2.7 Cassetterecorder als randapparaat	51
3.2.8 Op de uitbreidingsconnector aan te sluiten randapparatuur	53
3.2.9 Databeheer toegankelijk via de uitbreidingsconnector	54
3.2.10 ZX printer	58
4. De nabije toekomst	59
4.1 Algemeen	59
4.2 Telecommunicatie-adapter	59
4.3 Het hardware-aspect van de RS232C-standaard	62
4.4 Het software-aspect; ASCII-code	63
4.5 Microdrives	64

5. Werken voor het gebruik	66
5.1 Sinclair-computers	66
5.1.1 ZX81	66
5.1.2 ZX Spectrum	66
5.2 Het concept van een 'commerciële onderneming'	67
5.3 Soorten bestanden	68
5.3.1 Bestandstabellen	69
5.3.2 Bestandsreeksen	69
5.4 Sorteermethoden	70
5.4.1 Sorteren volgens de 'bubble-up'-methode	70
5.4.2 Sorteren door tussenvoegen/verschuiven	71
5.4.3 Binair sorteren	72
5.4.4 Methode Shell	72
5.4.5 Methode Shell-Metzner	73
5.5 Tekstverwerking	78
6. Enkele slotopmerkingen	90
7. Aanhangsels	92
7.1 ASCII-code	92
7.2 Verklarende woordenlijst	94

Inleiding

Voor wat betreft de microcomputer voor privégebruik zijn de afgelopen jaren rijk aan noviteiten geweest. Tussen de vele produkten die de amateur aangeboden krijgt heeft de firma Sinclair, dank zij het interessante aanbod van haar produkten, onmiskenbaar de aandacht op zich weten te vestigen. De prijs/prestatie-verhouding is zeker een rechtvaardiging voor het immense commerciële succes van die apparatuur. Het gevolg van de verkoop van meer dan 600.000 systemen is echter wel dat er een popularisatie en democratisering van het gebruik van microcomputers in de familiekring optreedt.

De voorloper van deze reeks, de ZX80 die in 1980 werd uitgebracht, verbaasde en verraste door zijn geringe omvang en vele mogelijkheden. De ZX81, die een jaar later werd geïntroduceerd, heeft met zijn prijs/prestatie-verhouding op heel wat amateurs en professionals een bijzonder goede indruk gemaakt. De originaliteit, eenvoud en de zwierigheid van de geboden oplossingen hebben de voorliefde ervoor gewekt. Hoewel er ook wel wat terechte kritieken werden gespuid hebben die het grote succes van de ZX81 nooit nadelig kunnen beïnvloeden. Het in de handel brengen van een nieuw model, de ZX Spectrum, heeft wel de besluitvorming bemoeilijkt, maar niet of er een gezinscomputer aangeschaft zou worden.

De schrijver heeft zich tot doel gesteld te laten zien dat goedkope microcomputers zeker aan de verlangens van de gebruiker kunnen voldoen, mits die zijn bedoelingen met kennis van zaken weet te formuleren. Teneinde het specifieke woordgebruik in de informatica van zijn mysteries te ontdoen is achterin het boek een lijst van technische woorden, met vertalingen van de belangrijkste Engelstalige woorden, opgenomen. Enkele praktische toepassingsvoorbeelden illustreren de beste manier om het gestelde doel te bereiken. Een vergelijking van de computers van Sinclair, aan de hand van vergelijkende tabellen, maakt het de lezer mogelijk om al naar gelang de bedoelde toepassing zijn keus te bepalen.

Een diepgaande analyse van de karakteristieke eigenschappen, zowel vanuit het oogpunt van de software als van de hardware, zonder daarbij de mogelijkheden tot uitbreiding met behulp van randapparatuur uit het oog te verliezen, completeren de beschrijving van deze laatstgeborene uit de Sinclair-laboratoria. Teneinde de compatibiliteit te vergroten en aanpassing mogelijk te maken van voor andere microcomputers ontwikkelde programma's, worden in een meer gedetailleerd hoofdstuk de middelen toegelicht om met betrekking tot deze systemen de leemtes en verschillen in de syntax van het Spectrum-BASIC te omzeilen. De voornaamste moge-

lijkheden, zoals die worden geboden door uitbreidingen met randapparatuur als diskteststations en telecommunicatie-interfaces, worden tot in detail behandeld.

Ten slotte analyseert en behandelt de schrijver twee echt technische, maar veelvuldig toegepaste onderwerpen, te weten: het opzetten van bestanden en sorteermethoden. Het boek besluit met een paragraaf over de aan de Spectrum aangepaste ASCII-code, die zowel als bron van informatie en als naslagwerk kan dienen.

De auteur heeft de inhoud van het boek zo opgesteld dat de lezer niet alleen de theorie kan volgen, maar ook de praktijk kan toepassen. Het boek is geschreven in een eenvoudige, heldere stijl, die ook voor de beginner begrijpelijk is. De auteur heeft de inhoud van het boek zo opgesteld dat de lezer niet alleen de theorie kan volgen, maar ook de praktijk kan toepassen. Het boek is geschreven in een eenvoudige, heldere stijl, die ook voor de beginner begrijpelijk is.

De auteur heeft de inhoud van het boek zo opgesteld dat de lezer niet alleen de theorie kan volgen, maar ook de praktijk kan toepassen. Het boek is geschreven in een eenvoudige, heldere stijl, die ook voor de beginner begrijpelijk is. De auteur heeft de inhoud van het boek zo opgesteld dat de lezer niet alleen de theorie kan volgen, maar ook de praktijk kan toepassen. Het boek is geschreven in een eenvoudige, heldere stijl, die ook voor de beginner begrijpelijk is.

De auteur heeft de inhoud van het boek zo opgesteld dat de lezer niet alleen de theorie kan volgen, maar ook de praktijk kan toepassen. Het boek is geschreven in een eenvoudige, heldere stijl, die ook voor de beginner begrijpelijk is. De auteur heeft de inhoud van het boek zo opgesteld dat de lezer niet alleen de theorie kan volgen, maar ook de praktijk kan toepassen. Het boek is geschreven in een eenvoudige, heldere stijl, die ook voor de beginner begrijpelijk is.

De auteur heeft de inhoud van het boek zo opgesteld dat de lezer niet alleen de theorie kan volgen, maar ook de praktijk kan toepassen. Het boek is geschreven in een eenvoudige, heldere stijl, die ook voor de beginner begrijpelijk is. De auteur heeft de inhoud van het boek zo opgesteld dat de lezer niet alleen de theorie kan volgen, maar ook de praktijk kan toepassen. Het boek is geschreven in een eenvoudige, heldere stijl, die ook voor de beginner begrijpelijk is.

1. De toepassingsgebieden van de Spectrum

De computer is een kampioen in het ordenen, het op volgorde plaatsen. Alle informatie, zoals die door de computer wordt verwerkt, wordt eerst in code omgezet. De computer verwerkt op die manier zonder onderscheid letters, cijfers en begrippen. De informatieverwerking staat onder besturing van een reeks van instructies die hem door de gebruiker worden opgedragen. Deze instructiereeks, het programma, beïnvloedt de werking van het centrale deel van de machine, de rekenkundig-logische eenheid. De werking van het hart van het systeem berust op rekenkunde en logica. Om de door het programma voorgeschreven bewerkingen uit te voeren bedient men zich van de eenvoudigste, fundamentele bewerkingen. Het is de eenvoud van die bewerkingen die de grote snelheid mogelijk maken. En, het is de veelheid van eenvoudige dingen die het mogelijk maakt complexe taken uit te voeren.

De computer is een hulpmiddel. Van hulpmiddelen bedienen we ons om onze prestaties op te voeren. Hulpmiddelen assisteren ons bij afzonderlijke taken. Ze vergemakkelijken, verbeteren en versnellen onze arbeid.

De computer is een geestelijk hulpmiddel. De computer vertienvoudigt onze intellectuele prestaties. De computer is programmeerbaar. De instructies uit het computerprogramma worden naar de letter uitgevoerd zonder dat de inhoud ervan wordt gewijzigd. De instructies die de computer accepteert stellen hem bijvoorbeeld in staat om:

1. wiskundige berekeningen uit te voeren,
2. grote hoeveelheden informatie op te slaan en een deel ervan ten behoeve van een specifiek doel te selecteren, en
3. een aantal belangrijke alternatieven te beoordelen om tot de beste oplossing voor een gegeven probleem te komen, zonder dat het onderwerp daarvan een wiskundig karakter hoeft te hebben.

De computer is een besturend hulpmiddel. De computer helpt ons toezicht te houden op andere, krachtiger hulpmiddelen of machines.

De computer is een precisie-hulpmiddel. Een goed geprogrammeerde computer voert zijn taak uit met maar een geringe kans op fouten. In bepaalde gevallen kan hij zichzelf op fouten controleren en die corrigeren.

De computer is een snel hulpmiddel. Een computer als de Spectrum kan 350.000 machine-instructies per seconde uitvoeren. Zijn, om een bepaalde taak door de computer te laten uitvoeren een miljoen elementaire bewerkingen nodig en levert dat binnen een tijdsbestek van enkele seconden een resultaat op, dan is dat vaak veel sneller dan de mens aan welk alternatief dan ook kan afschatten.

De computer is een betrouwbaar hulpmiddel. De hem opgedragen reeksen van bewerkingen worden onvermoeid steeds weer opnieuw herhaald.

Uitgaande van de gegeven beschrijving staat het buiten elke discussie dat de computer is voorbestemd om de mens bij zijn ondernemingen te assisteren. De toepassingsgebieden laten zich in vier categorieën verdelen. Naarmate de bezigheden thuis meer op het beheer van een onderneming lijken, zijn de toepassingsvoorbeelden voor een belangrijk deel ook op de familiekring van toepassing.

Binnen de familiekring, om te beginnen, beperkt het gebruik van de computer zich niet tot spellen alleen. Exacte kennis en studie van hoe bij het programmeren te werk te gaan maakt het mogelijk huishoudelijke vraagstukken op te lossen. Uitbreiden van de computer met communicatiemiddelen naar de buitenwereld (de zogenaamde interfaces) maakt besturing van fysische grootheden mogelijk en staat het toe een automatische bewakingsfunctie uit te oefenen. Ten slotte kan een perfecte beheersing van de computer zijn bezitter er toe aanzetten hem als een doeltreffend partner voor zijn professionele werkkring te gaan beschouwen. De exacte en snelle besturing van informatie brengt een betere besluitvorming met zich mee en vergroot de doeltreffendheid en de resultaten van de onderneming.

De vier categorieën zijn:

- de ludieke computer,
- de huishoudelijke computer,
- de persoonlijke of individuele computer, en
- de professionele computer.

Men treft de computer aan in:

- de industrie: waar men hem gevaarlijk en monotoon werk toevertrouwt, kwaliteitscontrole, exacte procesbesturing, het meten van fysische grootheden in vijandige milieus, simulaties enz.;
- de dienstverlenende sector: bij het bestandsbeheer - klanten, leveranciers, voorraden enz., de comptabiliteit, fakturering, onbetaalde rekeningen enz. - tekstverwerking van brieven, steeds terugkerende correspondentie, contracten enz.;
- de research: bij het verzamelen en verwerken van gegevens, grafische weergave daarvan, routinematige berekeningen, modelvorming enz.;
- het onderwijs: bij de administratie, lesroosters, bibliotheek enz., geprogrammeerde cursussen, onvermoeibaar repetitor, praktische oefeningen enz.;
- de vrije tijd: als spelpartner, zelfontwikkeling, regelen van de verwarming, HiFi, alarm, planten begieten.

De computer verhoogt dan ook de capaciteiten van de mens. Hij draagt bij aan de ontwikkeling van de kenmerkende bekwaamheden van de individu, te weten: logica, organisatievermogen en creativiteit. De mogelijkheden van de automaat, zoals die zich uit de computer ontwikkelden, ontlasten de mens van kleurloze industriële activiteiten ten gunste van een ontwikkeling van de menselijke aanleg.

2. Een vergelijking

Strikt beperkt tot de door de firma Sinclair vervaardigde computerprodukten heeft deze vergelijking tot doel de karakteristieke eigenschappen te analyseren van de drie aan het publiek aangeboden microcomputers: de ZX80, de ZX81 en de ZX Spectrum. Deze drie produkten, van de hand van dezelfde constructeur, zijn het resultaat van een ontwikkelingsgang die wel heel duidelijk de belangstelling van de producent illustreert om tegemoet te willen komen aan de behoefte van de gebruiker. Het is dan ook interessant deze ontwikkelingsgang te beschrijven en even stil te staan bij die verschillen die de ZX Spectrum maken tot de microcomputer die de meeste troeven in zich verenigt.

2.1 ZX80



Vandaag de dag wordt de ZX80 niet meer in de handel gebracht. In deze studie is hij dan ook uitsluitend van historisch belang en alleen de meest opvallende kenmerken worden besproken. Het eerste belangrijke verschil tussen de ZX80 en zijn opvolgers schuilt in de uitvoering van de hardware. In feite telt de ZX80 21 geïntegreerde schakelingen. Het tweede verschil heeft betrekking op de software. De BASIC-interpreter is ondergebracht in 4 Kbyte leesgeheugen (ROM). De derde bijzonderheid heeft betrekking op het geringere aantal mogelijkheden van het instructierepertoire en de grafische tekens. De calculator werkt uitsluitend met gehele getallen en wiskundige berekeningen (logaritmen, trigonometrie, exponenten) zijn niet mogelijk.

Door de ROM van de ZX80 door een herziene versie te vervangen konden gebruikers van de ZX80 profiteren van de voordelen van de software en de mogelijkheden van de calculator van het type 'ZX81'. Uitbreiden van het lees/schrijfgeheugen (RAM) tot 16 Kbytes behoort eveneens tot de mogelijkheden.

2.2 ZX81

Vanuit het oogpunt van de apparatuur bezien is de ZX81 minder omvangrijk en telt hij niet meer dan vier geïntegreerde schakelingen. Dit staaltje van techniek verdient alle bewondering. Bij een nader onderzoek blijken de prestaties van de zwarte doos over het geheel genomen bijzonder positief. De BASIC-interpreter is ondergebracht in een 8 Kbytes ROM. De machine werkt met twee snelheden. Langzaam, waarbij het beeldscherm voortdurend wordt ververs en verwerkingen en berekeningen uitsluitend worden uitgevoerd gedurende de beeldsynchronisatie-interval. Snel, waarbij verwerkingen en berekeningen naar prioriteit en ten koste van het beeldscherm worden uitgevoerd. De werksnelheid is dan vier maal zo hoog.

De computer verwerkt getallen met zwevende komma en geeft de acht meest significante cijfers weer. Hij werkt met wetenschappelijke notatie die verschil maakt tussen mantisse en exponent. Het rekenbereik loopt van 10^{38} tot $4 \cdot 10^{-39}$. De veertig toetsen van het in niveaus onderverdeelde toetsenbord van de ZX81 bevat niet alleen hoofdletters, cijfers, speciale tekens, twintig grafische tekens met hun inversen, maar ook sleutelwoorden (instructies, commando's), die via het toetsenbord worden ingevoerd en die door het systeem net als een enkel karakter worden geïnterpreteerd. Om de beschikbare ruimte beter te benutten zijn aan bepaalde toetsen wel tot vijf verschillende functies toegekend. Deze worden verkregen door gebruik te maken van de cursor-functies in combinatie met de hoofd-lettertoets (SHIFT).

De cursorfuncties worden aangegeven door een witte letter op een zwarte achtergrond:

sleutelwoord	K	(Keyword)
alfanumeriek karakter	L	(Letter)
variabelen bewerken	F	(Functie)
grafisch symbool	G	(Grafiek)

Het beeldscherm is onderverdeeld in 24 regels van elk 32 karakters. De ruimte is opgedeeld in twee velden. Enerzijds de bovenste 22 regels waarop het programma of het resultaat wordt weergegeven. Anderzijds de beide onderste regels waarop commando's of instructies kunnen worden afgebeeld. Met behulp van dat deel van het beeldscherm voert de gebruiker zijn dialoog met de ZX81: uitvoeringscodes, foutmeldingen enz.

Om te worden ingevoerd moet elke instructie worden beëindigd door op het toetsenbord de toets 'FUNCTION/NEWLINE' in te drukken. Daarop wordt de instructie onmiddellijk uitgevoerd. In het geval van een syntaxfout of van een logische fout geeft de cursor 'S' de plaats van die fout aan. Correctie ervan is mogelijk door de cursor naast de fout te zetten en die te wissen door de toets 'RUBOUT/0' in te drukken. Wordt nu aan het eind van de regel het commando 'FUNCTION/NEWLINE' gegeven, dan blijkt de bevestiging daarvan uit het feit dat de tekst vanuit het onderste veld van het beeldscherm naar het bovenste wordt overgebracht. Wijzigen van een bestaand programma is mogelijk met de opdracht 'EDIT'. De cursor op het bovenste veld van het beeldscherm geeft de lopende regel aan en kan verticaal worden verplaatst. Nadat de opdracht is gegeven wordt de lopende regel van het bovenste veld van het beeldscherm overgebracht naar het onderste. Wijzigen gebeurt op dezelfde manier als een correctie.

De BASIC-software van de Sinclair ZX81 wordt gekenmerkt door:

- Een enkele instructie per programmaregel. Opgemerkt dient te worden dat de regellengte meer dan 32 tekens mag bedragen, zonder beperking op het aantal.
- Zeer krachtige instructies waarmee substrings kunnen worden gekozen.

```
10 LET a$="rekenkunde"  
20 PRINT a$(2 TO 7)  
30 REM  
40 REM 1 2 3 4 5 6 7 8 9 10  
50 REM r e k e n k u n d e  
60 REM * * * * *
```

ekenku

Met een dergelijke instructie kan intern worden afgekort wat in andere BASIC-versies met behulp van de instructies LEFT\$, MID\$ en RIGHT\$ gebeurt. Deze kunnen weliswaar veel gemakkelijker te gebruiken zijn, maar zijn ronduit minder elegant.

- Voorwaardelijke instructies waarmee een logische interpretatie van de voorwaarden mogelijk is. Deze nemen dan, afhankelijk of de conditie 'waar' of 'niet waar' is, de waarde '1' of '0' aan. Invoeren van logische vergelijkingen is eveneens mogelijk. Deze afwijkingen ten opzichte van andere BASIC-versies maken het mogelijk om binnen een kleinere geheugenruimte zeer bewerkelijke wissels te definiëren.

Na een grondige studie van de interne werking van het Sinclair ZX81-BASIC mag men verwachten interessante BASIC-programma's te kunnen schrijven die niet meer dan 1Kbyte RAM-geheugenruimte in beslag nemen.

2.3 ZX Spectrum



Gemakshalve zullen we de ZX Spectrum verder met de afkorting 'Spectrum' aanduiden.

Vanuit het oogpunt van de hardware omvat de Spectrum de vier geïntegreerde schakelingen van de ZX81, 16 Kbyte RAM-geheugen en de elektronische componenten die voor het kleurenbeeld en het geluid zorgen. Met behulp van interne geheugenuitbreidingen kunnen in totaal 48 Kbyte lees/schrijf-geheugen voor de gebruiker toegankelijk worden gemaakt. Afgezien van de aanvullende elektronische componenten zijn wij op bekend terrein. Als resultaat van de ontwikkelingsgang van de ZX81 biedt de Spectrum naast alle voordelen van de ZX81 ook een oplossing voor de gebreken en tekortkomingen van de ZX81.

De Spectrum is een volwassen informatica-product dat de vergelijking met andere in de handel verkrijgbare computers gemakkelijk kan doorstaan. Het is een meesterstukje dat deze 'kleine Europese producent' de uitdaging van producenten aan de andere kant van de Atlantische Oceaan en uit het land van de Rijzende Zon durft te aanvaarden. In vergelijking met de ZX81 is de zwarte doos groter. Het toetsenbord met de 40 toetsen is vergroot en is uitgerust met druktoetsen. Toetsaanslagen worden akoestisch bevestigd met een 'klik'. Omdat het aantal functies werd vergroot zijn er nu maar liefst zes functies per toets. De functies worden geactiveerd door de cursor-MODE. Behalve de cursors **K**, **L**, **F** en **G** van de ZX81 wordt het werken ermee nog eens vergemakkelijkt door drie andere cursors. **D** geeft de lopende regel aan, **C** zet de hoofdletters vast en **E** zorgt voor uitbreiding van de functies (Extended).

Gezien de ingewikkelde bediening vergt de eerste gebruikstijd wel wat meer aandacht. Maar het invoeren van instructies met behulp van een enkele toets, zoals bij de ZX81, versnelt het redigeren van programma's aanzienlijk. Om het comfort nog verder te vergroten zijn nog verscheidene andere gebruiksmogelijkheden opgenomen: akoestisch bevestigingssignaal, vastzetten en automatisch herhalen van de hoofdletters.

De BASIC-interpreter neemt nu 16 Kbyte ROM-geheugen in beslag. De calculator is vergelijkbaar met die van de ZX81 in de FAST-functie, maar

dan wel vier maal zo snel. De beeldschermbesturing werkt onafhankelijk van de calculator. De beeldstabiliteit en het rekenen met gelijktijdige presentatie van een beeld zijn onveranderd gebleven. Deze functionaliteit is vergelijkbaar met die van andere systemen en corrigeert de lastige werkwijze van de ZX81.

Het besturingssysteem van de Spectrum is zeer duidelijk op de buitenwereld gericht. De verfijnde besturing van het kleurenbeeld, de elegante wijze waarop uitgaande van de taal klanken worden geproduceerd, de invoer/uitvoer-instructies, het grote aantal functies voor de besturing van de micro-drives en ten slotte de algehele compatibiliteit van de karakterset met standaard-ASCII. In vergelijking met de ZX81 zijn de systeemcommando's sterk verfijnd. Behalve het bewaren en teruglezen van programmabestanden herkent het systeem onafhankelijk variabelen, bytes en beelden. Verifiëren en samenvoegen van een bestand in het geheugen met een bestand op cassette behoort eveneens tot de mogelijkheden. Bedieningsfouten worden in tekstvorm (in het Engels, dat is waar) weergegeven naast een technisch verslag in cijfers.

Het beeldscherm van de Spectrum laat evenals dat van de ZX81 24 regels van elk 32 karakters zien. Het beeldscherm is onderverdeeld in twee velden: één van 22 regels voor het lopende programma, het andere van twee regels is bedoeld voor de dialoog met het systeem. Bij het inschakelen is het beeldscherm van een door de Spectrum gestuurde kleuren-TV wit. De karakters worden daarop in zwart geschreven.

De karakterset van de Spectrum bestaat uit: alfanumerieke karakters in hoofd- en kleine letters, de gebruikelijke speciale tekens (zoals het teken %) en uit bijzondere tekens (bijvoorbeeld © : copyright), 16 grafische karakters en 21 grafische karakters waarvoor de gebruiker de 64 samenstellende elementen kan definiëren. Deze creatieve mogelijkheid maakt het bijvoorbeeld mogelijk de voor een bepaalde wetenschappelijke richting specifieke karakters te tekenen, bijvoorbeeld Griekse letters. Ook dient het over elkaar heen drukken te worden genoemd. Dat is een door de leverancier geopperde mogelijkheid om letters te accentueren. Voorts kan aan elk karakter een kenmerk worden meegegeven dat het knippen en extra helderheid definieert. In de grafische functie is het beeldscherm opgedeeld in twee velden. Een achtergrond of rand (BORDER) en de voorgrond of wel het papier. Deze nemen onafhankelijk één van de acht mogelijke kleuren aan. Wordt met een zwart/wit-TV gewerkt, dan worden, in plaats van de verschillende kleuren, grijs tinten weergegeven. Het grafische veld is opgebouwd uit 192 lijnen en 256 onafhankelijk toegankelijke beeldpunten. Met de grafische functies kunnen punten, lijnen, bogen en cirkels worden getekend.

Geluid wordt gegenereerd met behulp van de instructie 'BEEP'. Met deze instructie kan de hoogte en de duur van een toon worden vastgelegd. De grondtoon is do 3. Door programmeren kunnen een gematigde toonladder, een natuurlijke toonladder of glijdende tonen worden verkregen. Het akoestische signaal kan van de EAR-plug worden afgetapt en kan zonodig extern worden versterkt.

Het manipuleren met externe bestanden op cassettes heeft eveneens merkbare verbeteringen ondergaan. Wegschrijven, lezen en verifiëren gaan vijf maal sneller en veel betrouwbaarder dan met de ZX81. Ook biedt de Spectrum de mogelijkheid om het weggeschrevene op het beeldscherm te controleren. Buiten randapparatuur als cassetterecorder en beeldscherm

beschikt de Spectrum ook nog over andere uitbreidingsmogelijkheden. Evenals bij de ZX81 drukt de Spectrum de informatie ook op een Sinclair ZX printer af. In de nabije toekomst zal ook een telecommunicatie-module volgens de EIS-norm RS232C kunnen worden geleverd, wat verbindingen tussen computers onderling mogelijk maakt. Op die manier kunnen dan bestanden over telefoonlijnen worden uitgewisseld. Dezelfde interface zal het dankzij de ASCII-standaard mogelijk maken om met de Spectrum en met behulp van de traditionele printers gedrukte documenten te produceren.

Van alle voor de Spectrum aangekondigde uitbreidingsmogelijkheden is microdrive wel het belangrijkste. Deze vrij toegankelijke magnetische geheugeneenheden kunnen per station 100Kbytes bevatten. In totaal kunnen acht stations worden aangesloten. De toegangstijd tot de opgeslagen informatie bedraagt gemiddeld 3,5 seconden. De informatie-overdrachtsnelheid is 16Kbyte per seconde. Door hun omvang, de toegangstijd en overdrachtsnelheid maken deze stations de Spectrum tot een computer die heel goed in staat is om in een zeer korte tijd een grote hoeveelheid informatie te verwerken.

2.4 Functie-overzicht

ZX80	ZX81	Spectrum	functie	
x	x	x	ABS	Absolute waarde.
	x	x	ACS	Berekent de arccosinus in radialen.
x	x	x	AND	Logische functie, EN-functie (prioriteit 3).
	x	x	ASN	Berekent de arcsinus in radialen.
	x	x	ATN	Berekent de arctangens in radialen.
		x	ATTR	Definieert voor een punt X,Y op het beeldscherm de kenmerken: knipperen, extra helderheid, papierkleur (PAPER) en inktkleur (INK).
		x	BIN	Maakt het mogelijk om met behulp van '0' en '1' een getal in binaire schrijfwijze voor te stellen.
x	x	x	CHR\$	Geeft het karakter waarvan de code overeenkomt met het gehele getal, aangegeven door de variabele.
x	x	x	CODE	De waarde van de code van het eerste karakter van een string.
	x	x	COS	Berekent de cosinus van een hoek uitgedrukt in radialen.

x x EXP	Berekent de macht van 'e' (exponentieel).
x FN	Roept een door de gebruiker gedefinieerde functie op (zie DEF).
x IN	Haalt de aan de invoer/uitvoerpoort aangeboden waarde binnen en slaat die op in de accumulator van de microprocessor.
x x INKEY\$	Leest het toetsenbord uit, haalt het karakter of de lege string binnen; functie van de cursor L of C.
x x INT	Presenteert de waarde van het gehele getal na afronding naar beneden.
x x LEN	Meet en presenteert de lengte van een string.
x x LN	Berekent de neperse logaritme (grondtal 'e').
x x x NOT	Logische inverterende functie (prioriteit 4).
x x x OR	Logische functie, OF-functie (prioriteit 2).
x x x PEEK	Adresseert en leest de byte op het aangegeven adres.
x x PI	3,14159265
x POINT	Beschrijft voor het in coördinaten aangegeven punt de kleur van het papier (PAPER) of van de inkt (INK).
x x x RND	Genereert een willekeurig getal.
x SCREEN\$	Presenteert het karakter van de genoemde coördinaten.
x x SGN x	Geeft het teken van 'x' ('+', '0', '-').
x x SIN	Berekent de sinus van een hoek uitgedrukt in radialen.
x x SQR	Berekent de vierkantswortel.
x x STR\$	Zet een getal om in een string.
x x TAN	Berekent de tangens van een hoek uitgedrukt in radialen.
x TL\$	Maakt het mogelijk het linker deel van een string af te kappen.
x x x USR	Roept de subroutine in machinetaal op.
x x VAL	Geeft de waarde van de in de string geschreven numerieke expressie.

x VAL\$	Geeft de waarde van de in de string geschreven expressie.
x x -	Keert het teken van een getal om (prioriteit 9).
x :	Scheidt meerdere instructies op een regel van elkaar.

2.5 Instructie-overzicht

ZX80 ZX81 ZX Spectrum instructies en/of commando's		
x BEEP		Activeert de luidspreker gedurende x seconden en met y halve tonen (+ of -) ten opzichte van de gemiddelde do.
x BORDER		Bepaalt de kleur van de rand van het beeldscherm.
x BRIGHT		Bepaalt de helderheid van de weer te geven karakters.
x CAT		Is van toepassing op microdrives, stelt de index op.
x CIRCLE		Trekt een cirkel rond het in coördinaten aangegeven middelpunt en met de aangegeven straal.
x x x CLEAR		Wist variabelen, maakt bezette geheugenruimte vrij, initialiseert het systeem.
x x x CLEAR,n		Vergelijkbaar met CLEAR, maar probeert zo mogelijk n geheugenplaatsen bovenin het RAM-geheugen te reserveren.
x CLOSE		Is van toepassing op microdrives, sluit de toegang tot een bestand af.
x x CLS		Initialiseert het beeldschermgeheugen, wist het beeldscherm.
x x x CONT		Vervolgt de uitvoering van een programma na een onderbreking.
x x COPY		Is van toepassing op de Sinclair-printer, kopieert de 22 regels van het beeldscherm.
x DATA		Opgenomen in een programma legt deze instructie een lijst van numerieke waarden en/of strings aan.

x	DEF FN ()	Door de gebruiker gedefinieerde functie.
x	DELETE	Is van toepassing op microdrives, heft een bestand op.
x	DIM x ()	Gereserveerde geheugenruimte voor numerieke waarden, een dimensie.
x x	DIM x\$ (-,-)	Gereserveerde geheugenruimte voor numerieke waarden, meerdere dimensies.
x	DRAW	Tekent binnen de ruimte van het beeldscherm een vector vanuit de momentele positie naar de genoemde coördinaten en onder de voorgeschreven hoek.
x	ERASE	Is van toepassing op microdrives, wist een bestand.
x	FAST	Voert een berekening uit ten koste van de beeldschermpresentatie.
x	FLASH	Bepaalt het knipperen van karakters op het beeldscherm.
x x x	FOR-TO	Instructie voor een lus (stapgrootte één).
x x	FOR-TO-STEP	Instructie voor een lus (stapgrootte anders dan één).
x	FORMAT	Is van toepassing op microdrives, initialiseert het medium.
x x x	GOSUB	Sprong naar de subroutine.
x x x	GOTO	Onvoorwaardelijke sprong.
x x x	IF-THEN	Voorwaardelijke uitvoering.
x	INK	Bepaalt de kleur waarmee de karakters op het beeldscherm worden geschreven.
x x x	INPUT	Instructie om op invoer van het toetsenbord te wachten.
x	INVERSE	Keert het video van het daaropvolgende karakter om.
x x x	LET	Kent een waarde toe aan een variabele.
x x x	LIST	Presenteert, uitgaande van de gegeven referentie, het programma op het beeldscherm.
x x	LLIST	Drukt, uitgaande van de gegeven referentie, het programma af.
x x x	LOAD	Laadt een programma en de variabelen.

x	LOAD-DATA	Laadt de numerieke waarden of de aangegeven string.
x	LOAD-CODE	Laadt een aangegeven aantal bytes.
x	LOAD-SCREEN\$	Laadt vanuit de cassette het gespecificeerde bestand dat de karakteristieken van het beeldscherm bevat.
x x	LPRINT	Drukt de resultaten af op de printer.
x	MERGE	Voegt een in het geheugen opgeslagen programma samen met een programma dat eerder op cassette of microdrive werd geladen.
x	MOVE	Is van toepassing op microdrives, verplaatst een bestand.
x x x	NEW	Initialiseert het systeem.
x x x	NEXT	Instructie tot het beëindigen van de lus (FOR-TO-STEP).
x	OPEN	Is van toepassing op microdrives, opent een bestand.
x	OUT	Brengt de inhoud van de registers B en C over naar de uitvoer-interface.
x	OVER	Laat de karakters op het beeldscherm over elkaar heen verschijnen.
x	PAPER	Legt de kleur van het papier (achtergrond) vast.
x x	PAUSE	Stopt de uitvoering van de verwerking, maar laat de beeldscherminhoud ongemoeid.
x x	PLOT	Tekent op de in coördinaten aangegeven positie en in de kleur 'INK', legt de nieuwe aanvangscoördinaten vast.
x x x	POKE	Wijzigt de aangegeven byte in de aangegeven waarde.
x x x	PRINT	Drukt op het punt van uitgang af.
x x	PRINT AT	Drukt op het aangegeven punt af.
x x	PRINT TAB	Drukt in de aangegeven kolom af.
x x x	RAND	(Randomize), initialiseert de variabele SEED van de willekeurige getallengenerator.
x	READ	Leest de waarden opgegeven in DATA.

x x x	REM	Instructie om commentaar op te nemen.
x	RESTORE	Initialiseert de leeswijzer van de instructie READ.
x x x	RETURN	Instructie om de subroutine te beëindigen en terug te keren naar het hoofdprogramma.
x x x	RUN,n	Start het programma op de aangegeven regel (voert uit: CLEAR GOTO n).
x	SAVE-LINE	Schrijft programma en variabelen weg en bepaalt de regel waarmee het programma na het inlezen zal worden gestart.
x	SAVE-DATA	Schrijft de variabelen weg (numeriek of string).
x	SAVE-CODE	Schrijft een gedefinieerd aantal bytes weg vanaf het genoemde adres.
x	SAVE-SCREEN\$	Schrijft het programma weg op het equivalent van SAVE CODE 16 384,6912.
x	SCROLL	Schuift het beeldscherm een regel omhoog.
x	SLOW	Laat de processor zijn bewerkingen gedurende de beeldsynchronisatie-intervallen uitvoeren; de beeldinhoud blijft tijdens de bewerkingen bewaard; vertraagt de berekeningen.
x	UNPLOT	Wist het met PLOT getekende punt, werkt als een vlakgom.
x x x	STOP	Stopt verder uitvoering van een programma.
x	VERIFY	Vergelijkt de van een extern bestand gelezen inhoud met de inhoud van het geheugen.

3. Overeenkomsten

Hoewel blijkt dat de ZX81 is voortgekomen uit de ZX80 en dat er bindingen tussen beide machines bestaan, blijkt dat niet het geval als men de ZX81 met de Spectrum vergelijkt. Er is alleen compatibiliteit in de opzet en het schrijven van de programma's in BASIC. Voor zover de instructies van de taal voor alle drie de machines identiek zijn, geldt dat ook voor de machines.

Als gevolg van verschil in omvang en in 'operating system' beletten de organisatiestructuren van de geheugens (ZX80: 4 Kbyte ROM; ZX81: 8 Kbyte ROM; Spectrum: 16 Kbyte ROM) de rechtstreekse overdracht van programma's waarbij van systeemvariabelen gebruik wordt gemaakt. De codering van de karakters, specifiek voor de ZX80/81, is voor de Spectrum in ASCII-code; dit is een nieuwe hindernis. Ten slotte verschilt de snelheid waarmee informatie door de ZX80/81 wordt weggeschreven met die van de Spectrum wat elke vorm van informatie-overdracht door andere media uitsluit.

Uit dit vergelijkend onderzoek volgt dat de ZX81 en de Spectrum twee specifieke produkten zijn die zich ieder op specifieke gebruikers richten. Gezien zijn prijs is de ZX81 een uitstekend hulpmiddel om met de computer vertrouwd te raken. Het grote aantal gebruikers en de vele uitbreidingsmogelijkheden zijn daarvan het onmiskenbare bewijs. Spectrum-gebruikers zullen een aanzienlijk groter bedrag moeten investeren maar komen daardoor op een nieuw, veel hoger prestatieniveau te verkeren, vergelijkbaar met dat van veel duurdere informatica-produkten.

3.1 Analyse van de software

Het is niet de bedoeling om in dit boekje een cursus BASIC te geven. Daarvoor zijn tal van doeltreffende en krachtige lesboeken verkrijgbaar, om nog maar niet te spreken van de handboeken die met de Spectrum worden meegeleverd. Wij gaan van de veronderstelling uit dat de lezer een zekere voorkennis van ZX81-BASIC bezit en beperken ons tot het beschrijven van die instructies in de Spectrum-taal die geen deel uitmaken van de software voor de ZX81.

Als men met Spectrum-BASIC begint, zal men ongetwijfeld verrast zijn door de, in vergelijking met de ZX81, uitgebreidere woordenschat. Een eenvoudig overzicht van de functies en instructies zoals die in de vergelij-

kende tabel in paragraaf 2.5 werden opgesomd, geeft het volgende beeld:

ZX80 : 30

ZX81 : 60

Spectrum : 100

Aan de hand hiervan zullen we de toename van het aantal functies laten zien dat sedert de komst van de Spectrum met deze nieuwe instructies mogelijk is geworden. Deze nieuwe instructies laten zich in drie categorieën opdelen:

- berekenings- en verwerkingsinstructies,
- besturingsinstructies voor interfaces en randapparatuur,
- grafische instructies.

Begonnen wordt met een beknopt overzicht van een aantal specifieke Spectrum-BASIC-instructies. Voor zover in het BASIC-dialect van de Spectrum niet alle, in andere versies van BASIC beschikbare instructies zijn overgenomen, en teneinde een gemakkelijke aanpassing van voor andere machines geschreven programma's mogelijk te maken, wordt dit hoofdstuk besloten met een verwijzing naar mogelijke vervangingsmethoden.

3.1.1 Berekenings- en verwerkingsinstructies

De eerste kenmerkende instructie is het teken ':'. Dank zij dit teken is het nu mogelijk om op eenzelfde programmaregel meerdere instructies achter elkaar te schrijven.

```
1 REM w=waarde
2 REM dit programma berekent
  de machten 1 t/m 5 van
  de waarde w.
3 REM
10 LET w=2: FOR i=1 TO 5: PRIN
T w^i: NEXT i
```

```
2
4
8
16
32
```

3.1.2 READ, DATA en RESTORE

Deze drie instructies werden opzettelijk bijeengevoegd. Ze werken eensgezind in hetzelfde programma. Omdat het ontbreken ervan zich in het ZX81-BASIC ernstig liet voelen begroeten we ze hier met genoeg. Het voornaamste doel ervan is het mogelijk te maken programmavariabelen die binnen dit zelfde programma worden opgesteld, te definiëren en achtereenvolgens te gebruiken. De instructie INPUT is heel bruikbaar om er

tijdens de uitvoering van een programma waarden mee aan een variabele toe te kennen. In het geval van een lange lijst van cijfers of termen echter maakt het feit dat de uitvoering van het programma voor het invoeren van elke waarde moet worden onderbroken, deze methode omslachtig. De instructies READ en DATA blijken efficiënte hulpmiddelen om een lijst van waarden toe te kennen, vergelijkbaar met de instructie LET, maar sneller en soepeler. In tegenstelling tot INPUT werken READ en DATA binnenin het programma en niet interactief via het toetsenbord. READ en DATA zijn twee verschillende instructies maar worden paarsgewijs gebruikt. De instructie READ dient om waarden toe te kennen en de instructie DATA om de te gebruiken waarde te beschrijven. De instructie READ roept de variabelen op. De instructie DATA laat een lijst van waarden zien. Het aantal variabelen en de lijst van waarden moeten hetzelfde aantal elementen bevatten.

```

10 READ a,b,c,d,e
20 PRINT a'b'c'd'e'
30 PRINT "het rekenkundig gemi
ddelde is: "
40 PRINT (a+b+c+d+e)/5
50 DATA 98,76,83,92,87

98
76
83
92
87
het rekenkundig gemiddelde is:
87.2

```

De instructie RESTORE werkt samen met de instructies READ en DATA in die zin dat het hiermee mogelijk is om dezelfde set waarden, die met de instructie DATA werd vastgelegd, opnieuw te gebruiken voor verschillende definities van de variabelen opgegeven met de instructie READ.

```

10 READ a,b,c,d
20 PRINT "sin 30 ,45 ,60 ,90 "
30 PRINT SIN a'SIN b'SIN c'SIN
d
40 DATA PI/6,PI/4,PI/3,PI/2
50 REM 30 45 60 90
60 RESTORE
70 PRINT "cos 30 ,45 ,60 ,90 "
80 READ w,x,y,z
90 PRINT COS w'COS x'COS y'COS
z

sin 30 ,45 ,60 ,90
0.5
0.70710678
0.8660254
1

```

```

cos 30 ,45 ,60 ,90
0.8660254
0.70710678
0.5
0

```

In de praktijk verplaatst zich tijdens de uitvoering van het programma een interne wijzer, die de volgende waarde aangeeft die aan de variabele moet worden toegekend. De instructie RESTORE zet de wijzer terug naar het begin van de reeks waarden die met de instructie DATA werd vastgelegd. Met de instructies READ en DATA kunnen zowel getal-variabelen als string-variabelen worden verwerkt.

3.1.3 DEF (FN); door de gebruiker gedefinieerde functie

De gebruikersfuncties kunnen worden toegepast wanneer de functies uit de systeembibliotheek niet voldoen. Kan een steeds terugkerende bewerking van het programma niet door een geschikte systeemfunctie worden uitgevoerd, dan is het met Spectrum-BASIC mogelijk een functie naar maat te maken. De aldus gedefinieerde functie kan niet buiten dat programma worden gebruikt, maar spaart ongetwijfeld geheugenruimte en interpretatietijd.

De instructie DEF (definition) wordt gebruikt om een functie te benoemen en te formuleren. Geadviseerd wordt om de gebruikersfunctie reeds aan het begin van het programma te definiëren. De instructie FN (function) wordt gebruikt om aan de Spectrum de kenmerken van de te definiëren of te gebruiken functie mee te delen. De functie krijgt een naam, voorgesteld door een alfabetisch karakter (kleine letters of hoofdletters door elkaar). De variabelen worden gedefinieerd tussen haakjes. Het 'is gelijk'-teken sluit de functie af en begint de beschrijving van de functie. Er wordt gebruik gemaakt van de BASIC-syntax om de gewenste mathematische functies te beschrijven.

```

10 REM omzetten van graden
11 REM CELSIUS naar FAHRENHEIT
20 DEF FN t(x)=(9/5)*x+32
30 INPUT "graden CELSIUS? ";c
40 PRINT c;" graden CELSIUS =
",FN t(c);" graden FAHRENHEIT"

```

```

21 graden CELSIUS =
69.8 graden FAHRENHEIT

```

De gedefinieerde functie kan zowel numerieke- als string-variabelen verwerken. Voor bewerkingen met string-variabelen dient het \$-teken te worden toegevoegd.

```

10 REM Vervangende functie
11 REM voor de instructie:
12 REM "LEFT$"
20 DEF FN l$(a$,n)=a$( TO n)

```

```

30 LET a$="Spectrum"
40 PRINT FN 1$(a$,2)
50 REM
60 REM 1 2 3 4 5 6 7 8
70 REM S p e c t r u m
80 REM * *

```

Sp

3.1.4 VAL\$

VAL\$ lijkt in syntax en in uitvoering op de instructie VAL. In het geval van VAL\$ wordt een string-variabele verwerkt, het resultaat daarvan is een letterwaarde. Bij weggeschreven geïndiceerde bestanden bereikt men een selectievereenvoudiging door string-variabelen op te roepen.

3.1.5 CLEAR

De laatste instructie van de groep is CLEAR. Deze is bedoeld voor gebruikers die met de geheugenadressering willen manipuleren en een geheugen-gebied willen vastleggen dat de BASIC-interpreter niet zal adresseren. Met CLEAR kan een voor de interpreter transparante geheugenruimte worden gedefinieerd door de systeemvariabelen te wijzigen die het adres van het hoogste geheugenbyte aangeven. Deze variabele heet RAMTOP. Ook kan men er meer ruimte voor BASIC mee reserveren door de grafische ruimte van de gebruiker te overschrijven. Het commando CLEAR voert de volgende acties uit:

- wist alle variabelen,
- wist, net als de instructie CLS, de inhoud van het beeldschermgeheugen,
- voert een RESTORE-instructie uit, en
- initialiseert de GOSUB-stack en verzorgt de verplaatsing van die stack naar het nieuwe adres van RAMTOP.

3.1.6 Aanpassen van Spectrum-BASIC aan andere BASIC-versies

De vetgedrukte termen komen voor in BASIC-programma's die ten behoeve van andere systemen dan de Spectrum werden geschreven. Hieronder volgt een overzicht van vergelijkingen, vertalingen en aanpassingen van deze BASIC-dialecten, naar het Spectrum BASIC-dialect.

ABORT	Onderbreekt de uitvoering van een programma, twee
STOP	overeenkomstige instructies zijn STOP en BREAK.
BREAK	
APPEND	Voegt een programma in het geheugen samen met een
MERGE	programma van externe media, is verwant aan MERGE.

%
INT Teken dat wordt gebruikt om uitsluitend gehele getallen aan te geven, waardoor minder geheugenruimte nodig is. Vanuit het oogpunt van gemakkelijk gebruik zou men, zonder dat dit winst aan geheugenruimte oplevert, de functie INT kunnen gebruiken waarmee eveneens numerieke waarden in gehele getallen worden omgezet.

```
10 PRINT INT (5/3)
```

ASC
CODE (-) De functie ASC geeft de numerieke ASCII-code die overeenkomt met het argument van het unieke karakter. Is te vervangen door CODE.

CALL
USR De instructie CALL wordt gebruikt voor het oproepen van een subroutine in assembleertaal, de Spectrum voert deze functie uit met behulp van de instructie USR.

```
10 PRINT USR 32500
```

CHAIN
LOAD
MERGE Het commando CHAIN zorgt voor het aan elkaar rijgen van programma's. Is verwant aan bepaalde versies van de instructie LOAD in het geval het éne programma door een ander wordt vervangen, of aan de instructie MERGE als men twee programma's wil samenvoegen.

COMMON
DIM
SAVE/LOAD-
CODE
SAVE/LOAD-
DATA De instructie COMMON zorgt ervoor dat de waarden van de variabelen en tabellen in het werkgeheugen bij gebruik van het volgende segment van een serie programma's bewaard blijven. Deze instructie heeft in Spectrum-BASIC geen equivalent. Wel kan een vervangende procedure worden toegepast voor zover de Spectrum de variabelen in de vorm van bytes of geheugengebieden kan veilig stellen respectievelijk laden met behulp van de instructies SAVE/LOAD-CODE en SAVE/LOAD-DATA. De bruikbare variabelen van het programma moeten daarbij in een gereserveerde zône worden geladen met behulp van een DIM-instructie, die de elementen bevat voor voortzetting van het segment van het volgende programma.

```
10 DIM a$(50)
```

```
20 SAVE "common" DATA a$()
```

DEGREES De Spectrum werkt met hoeken uitgedrukt in radialen. Omdat de omtrek van een cirkel gelijk is aan 2π x de straal, is de omtrek van een cirkel met een straal gelijk aan één gelijk aan $2\pi \times 1 = 2(3,14159)$, wat overeenkomt met:

$$360^\circ = 2\pi$$

$$180^\circ = \pi = \text{PI}$$

Omrekenen gebeurt met de vergelijking:

$$\text{radialen} = (\text{PI} * \text{graden})/180$$

$$\text{graden} = (180 * \text{radialen})/\text{PI}$$

DEL

DELETE
LIST =
NEWLINE

De opdracht DEL wist de instructies van het lopende programma vanaf het nummer van de individueel opgegeven instructies. Deze bewerking moet regel voor regel worden uitgevoerd door het regelnummer in te voeren en over het toetsenbord het commando NEWLINE te geven. Daardoor wordt de programmaregel vervangen door een lege regel, wat dus gelijk staat met het wissen ervan.

DIM

Bij bepaalde vormen van BASIC mogen na het sleutelwoord DIM meerdere variabelen worden ingevoerd. De Spectrum eist wel dat alle variabelen worden voorafgegaan door het sleutelwoord. De Spectrum accepteert daarentegen tabellen van meer dan twee dimensies.

```
10 DIM a(2): DIM b(3,4)
20 DIM c(5,6,7)
```

GET

INPUT
INKEY\$

Dit sleutelwoord kan twee betekenissen hebben:

1. Als commando roept het een op schijf opgeslagen bestand op, de alternatieve instructie van de Spectrum is INPUT.
2. Als instructie zorgt GET voor bewaking van het toetsenbord teneinde het volgende in te voeren karakter binnen te halen. Deze functie wordt uitgevoerd door de functie INKEY\$ van de Spectrum.

```
1 REM simulatie van de
2 REM instructie "GET"
10 LET a$=INKEY$
20 IF a$="" THEN GO TO 10
30 LET a=CODE a$
40 PRINT "de ingedrukte toets
is: ";a$
50 PRINT "de bijbehorende code
is: ";a
```

```
de ingedrukte toets is: A
de bijbehorende code is: 65
```

IF-THEN-ELSE

Voor deze variant van de selectie-instructie heeft de Spectrum geen direct equivalent. Afhankelijk van de toepassing kan men hem door andere oplossingen vervangen. Zie de listing op de volgende pagina.

```

10 PRINT "om te beginnen is y=
10      als x=1 dan zal y=y-
2 worden"

```

```

20 PRINT "x =: "; INPUT x: PR
INT x: LET y=10: PRINT "y =: 10"
: PRINT

```

```

30 IF x=1 THEN LET y=y-2: IF
x=1 THEN GO TO 60

```

```

40 PRINT "x <> 1": STOP

```

```

50 REM Deze techniek vervangt

```

```

51 REM de IF...THEN...ELSE...

```

```

52 REM structuur.

```

```

60 PRINT "x = 1": PRINT "y = "
;y

```

```

om te beginnen is y=10

```

```

als x=1 dan zal y=y-2 worden

```

```

x =: 0

```

```

y =: 10

```

```

x <> 1

```

```

om te beginnen is y=10

```

```

als x=1 dan zal y=y-2 worden

```

```

x =: 1

```

```

y =: 10

```

```

x = 1

```

```

y = 8

```

IF-THEN : Spectrum-BASIC.

LET

```

1 REM Aanpassing van de

```

```

2 REM IF...THEN...instructie

```

```

10 PRINT "In het begin is y=0"

```

```

20 PRINT "Indien x=1 dan wordt

```

```

y=2,      indien x<>1 dan word
t y=3."

```

```

30 LET y=0: PRINT "y = ";y

```

```

40 PRINT "x = "; INPUT x: PRI

```

```

NT x: PRINT

```

```

50 LET y=(2 AND x=1)+(3 AND x<
>1)

```

```

60 PRINT "y = ";y

```

```

In het begin is y=0

```

```

Indien x=1 dan wordt y=2,

```

```

indien x<>1 dan wordt y=3.

```

```

y = 0

```

```

10 x = 0
20 y = 3

In het begin is y=0
Indien x=1 dan wordt y=2,
indien x<>1 dan wordt y=3.
30 y = 0
40 x = 1
50 y = 2

```

LEFT\$ STR\$(TO n)

Instructie om het linker deel van een string-variabele te bewerken, de veelzijdiger instructie STR\$() van de Spectrum biedt een soortgelijke oplossing.

```

1 REM een andere manier om
2 REM de instructie LEFT$
3 REM te simuleren.
10 LET a$="ZX Spectrum"
20 LET b$=a$( TO 4)
30 PRINT b$

```

ZX S

LOG LN

Omdat de Spectrum uitsluitend met de natuurlijke logaritme (grondtal e=2,71828182) werkt kan het nuttig zijn van grondtal te veranderen:

$\log x \text{ (getal)} = \text{LN (getal)} / \text{LN (x)} = \text{nieuw grondtal}$

MAT

Met de vele verwerkingsvormen van deze zeer krachtige instructie kunnen matrices worden bewerkt. Van deze instructie bestaat geen enkel equivalent. De enige oplossing is de elementen van de matrix individueel te bewerken.

```

1 REM *****
2 REM *bewerken van matrices*
3 REM *****
4 REM
5 REM
100 REM initieren van matrix 1
101 REM
105 PRINT "matrix a"
110 PRINT "aantal rijen   : ";
INPUT r: PRINT r
120 PRINT "aantal kolommen: ";
INPUT k: PRINT k
130 DIM a(r,k)

```



```

140 FOR i=1 TO r: PRINT "waarde
van rij ";i
150 FOR j=1 TO k: PRINT "
kolom ";j;
160 INPUT a(i,j): PRINT " ";a
(i,j)
170 NEXT j
180 NEXT i
200 REM initieren van matrix 2
201 REM
205 PRINT "matrix b"
210 PRINT "aantal rijen : ";:
INPUT r: PRINT r
220 PRINT "aantal kolommen: ";:
INPUT k: PRINT k
230 DIM b(r,k)
240 FOR i=1 TO r: PRINT "waarde
van rij ";i
250 FOR j=1 TO k: PRINT "
kolom ";j;
260 INPUT b(i,j): PRINT " ";b
(i,j)
270 NEXT j
280 NEXT i
290 REM
300 REM controleren van de
301 REM matrixwaarden
310 PRINT "wilt u de waarde(n)
van de matrices controleren
? (ja/nee)": INPUT a$
320 IF a$<>"ja" THEN GO TO 400
330 PRINT "welke matrix? (a/b)
: ";: INPUT m$: PRINT m$
340 PRINT "welke rij? : ";: I
NPUT r: PRINT r
350 PRINT "welke kolom? : ";: I
NPUT k: PRINT k
360 IF m$="a" THEN LET m=a(r,k)
)
365 IF m$="b" THEN LET m=b(r,k)
)
367 PRINT "de waarde is ";m
370 PRINT "waarde wijzigen? (ja
/nee): ";: INPUT a$: PRINT a$
380 IF a$<>"ja" THEN GO TO 310
390 PRINT "nieuwe waarde? : ";:
INPUT w: PRINT w
393 IF m$="a" THEN LET a(r,k)=
W
395 IF m$="b" THEN LET b(r,k)=
W
399 GO TO 310

```

```

400 REM *****
401 REM * matrixbewerkingen *
402 REM *****
403 REM
410 PRINT "kies een bewerking"
"1 - optellen" "2 - aftrekken"
"3 - scalair vermenigvuldigen"
4 - wijzigen" "5 - afdrukken":
INPUT b: PRINT b
420 IF b<=0 OR b>5 THEN GO TO
410
430 GO TO 100*b+400
500 REM optellen
510 PRINT "optellen"
520 PRINT "beide matrices moete
n evenveel elementen bevatten":
PRINT
525 PRINT "MAT a = MAT a + MAT
b": PRINT
530 FOR i=1 TO r
540 FOR j=1 TO k
550 LET a(i,j)=a(i,j)+b(i,j)
560 NEXT j
570 NEXT i
599 GO TO 410
600 REM aftrekken
610 PRINT "aftrekken"
620 PRINT "beide matrices moete
n evenveel elementen bevatten":
PRINT
625 PRINT "MAT a = MAT a - MAT
b": PRINT
630 FOR i=1 TO r
640 FOR j=1 TO k
650 LET a(i,j)=a(i,j)-b(i,j)
660 NEXT j
670 NEXT i
699 GO TO 410
700 REM vermenigvuldig scalair
710 PRINT "scalaire vermenigvul
diging"
720 PRINT "factor? : ";: INPUT
f: PRINT f: PRINT
725 PRINT "MAT a = MAT a * fact
or": PRINT
730 FOR i=1 TO r
740 FOR j=1 TO k
750 LET a(i,j)=a(i,j)*f
760 NEXT j
770 NEXT i
799 GO TO 410
800 REM wijzigingen

```

```

810 GO TO 300
900 REM afdrukken resultaten
910 PRINT "resultaat van de bew
erking:": PRINT
920 FOR i=1 TO r
925 PRINT "rij   : ";i: PRINT
930 FOR j=1 TO k
940 PRINT a(i,j)
950 NEXT j
960 NEXT i
970 GO TO 300

```

```

matrix a
aantal rijen   : 2
aantal kolommen: 2
waarde van rij 1
           kolom 1   10
           kolom 2   20
waarde van rij 2
           kolom 1   30
           kolom 2   40

```

```

matrix b
aantal rijen   : 2
aantal kolommen: 2
waarde van rij 1
           kolom 1    5
           kolom 2   15
waarde van rij 2
           kolom 1   25
           kolom 2   35

```

```

wilt u de waarde(n) van de
matrices controleren? (ja/nee)

```

```

kies een bewerking
1 - optellen
2 - aftrekken
3 - scalair vermenigvuldigen
4 - wijzigen
5 - afdrukken
3
scalaire vermenigvuldiging
factor? : 5

```

```

MAT a = MAT a * factor

```

```

kies een bewerking
1 - optellen
2 - aftrekken
3 - scalair vermenigvuldigen
4 - wijzigen
5 - afdrukken

```

```

5
resultaat van de bewerking:
rij : 1
50
100
rij : 2
150
200
wilt u de waarde(n) van de
matrices controleren? (ja/nee)

```

MID\$ Instructie om het middendeel van een string-variabele
STR\$(m TO n) te bewerken, de veelzijdiger instructie **STR\$** van
de Spectrum biedt een soortgelijke oplossing.

```

1 REM simulatie van de
2 REM instructie: MID$
3 REM
10 LET a$="Spectrum"
20 LET b$=a$(4 TO 6)
30 PRINT b$
100 REM
101 REM 1 2 3 4 5 6 7 8
102 REM S p e c t r u m
103 REM * * *

```

ctr

MOD Deze instructie berekent de restwaarde van een
deling. Omdat de Spectrum niet over een soortgelijke
instructie beschikt moet die worden vervangen door een
algoritme of een gebruikersfunctie.

```

1 REM simulatie van de
2 REM instructie: MOD
3 REM
10 PRINT "deeltal? "; INPUT d
: PRINT d
20 PRINT "deler? "; INPUT r
: PRINT r
30 DEF FN m(d,r)=d-(INT (d/r)*
r)
35 PRINT : PRINT "resultaat :
";INT (d/r)
40 PRINT "de rest is: ";FN m(d
,r)
deeltal? 263
deler? 40

```

```
resultaat : 6  
de rest is: 23
```

ON GOTO ON GOSUB GOTO GOSUB

Dit zijn zeer krachtige switch-instructies omdat ze, bepaald door een variabele, meerdere vertakkingen kunnen kiezen. In Spectrum-BASIC zijn ze niet beschikbaar. Een doelmatig alternatief bestaat dankzij een wijziging van de door Spectrum-BASIC boven andere BASIC-versies geboden mogelijkheden. Zo kunnen regelnum-meraanduidingen het resultaat zijn van de berekening van een expressie. Door één en ander zo in te richten dat de nummers van de lussen gehele veelvouden zijn kan de volgende algoritme worden toegepast:

```
10 GOSUB 100 * X + 100
```

Kan X de waarden van 0 tot 5 aannemen, dan levert dit op:

```
10 ON X GOSUB 100, 200, 300, 400, 500
```

```
1 REM simulatie van de  
2 REM instructie ON...GOSUB  
3 REM  
10 PRINT "naar welke subroutin  
e zal wor- den gesprongen hangt  
af van de waarde van X, die hi  
er tussen 0 en 5 mag variëren.  
": PRINT  
20 PRINT "geef de waarde van X  
: "; INPUT X: PRINT X  
30 GO SUB 100*X+100  
40 GO TO 20  
100 REM subroutine voor X=0  
110 REM vervanging van ON.GOSUB  
200 REM subroutine voor X=1  
210 PRINT "de waarde van X is 1  
": RETURN  
300 REM subroutine voor X=2  
310 PRINT "de waarde van X is 2  
": RETURN  
400 REM subroutine voor X=3  
410 PRINT "de waarde van X is 3  
": RETURN  
500 REM subroutine voor X=4  
510 PRINT "de waarde van X is 4  
": RETURN  
600 REM subroutine voor X=5  
610 PRINT "de waarde van X is 5  
": RETURN
```

naar welke subroutine zal wor-

den gesprongen hangt af van de waarde van X, die hier tussen 0 en 5 mag variëren.

```
geef de waarde van X: 0
de waarde van X is 1
geef de waarde van X: 1
de waarde van X is 1
geef de waarde van X: 2
de waarde van X is 2
geef de waarde van X: 3
de waarde van X is 3
geef de waarde van X: 4
de waarde van X is 4
geef de waarde van X: 5
de waarde van X is 5
geef de waarde van X: 6
```

REPEAT-UNTIL Hoewel het Spectrum-BASIC niet over deze instructie beschikt, waarmee een reeks net zo lang kan worden herhaald tot aan een bepaalde gedefinieerde conditie wordt voldaan, is er wel een alternatief denkbaar om een conditie te toetsen en een soortgelijke techniek te gebruiken als die welke beschreven werd voor de instructies ON-GOTO, ON-GOSUB.

```
1 REM simulatie van de
2 REM instructie REPEAT.UNTIL
3 REM
5 LET a=1: PRINT a
10 REM begin van de
20 REM verwerkingslus
30 LET a=a+1
40 PRINT a
50 IF a<>5 THEN GO TO (a=5)*6
+10
60 PRINT "einde van de verwerk
ing"

1
2
3
4
5
einde van de verwerking
```

GOTO (cond)
(exp) Daar de hoogste prioriteit wordt toegekend aan de bewerking van de expressie is een functie GOTO (exp) geldig als de uitkomst van de bewerking een geheel getal is.

PRINT USING PRINT

Spectrum-BASIC kent het toevoegsel 'USING' niet. Hieronder volgt een alternatief om variabele uitkomsten volgens aangegeven formaat af te drukken.

```
10 REM *** PRINT USING ***
20 PRINT "formaat: ";
30 INPUT f$
40 PRINT f$
50 IF LEN f$>10 THEN GO TO 10
60 REM ** positie van de punt
70 REM ** formaatlengte
80 FOR i=1 TO LEN f$
90 IF f$(i)="." THEN LET p=i
100 IF f$(i)="." THEN LET d=LE
N f$-p
110 NEXT i
120 REM ** te formateren waarde
130 PRINT "waarde = ";
140 INPUT r
150 PRINT r
160 REM ** formateren
170 LET v$=STR$ r
180 FOR i=1 TO LEN v$
190 IF v$(i)="." THEN LET v=i
200 NEXT i
210 IF v>p THEN PRINT "past ni
et in formaat": GO TO 20
220 REM ** het gehele deel
230 LET r$=v$( 1 TO v-1)
240 FOR n=1 TO p-v
250 LET r$=" "+r$
260 NEXT n
270 REM ** het decimale deel
280 FOR i=1 TO LEN f$-LEN v$
290 LET v$=v$+"0"
300 NEXT i
310 LET r$=r$+v$(v TO v+d)
320 REM ** resultaat
330 PRINT "resultaat = ";r$
340 PRINT
350 GO TO 20
```

```
formaat: ##.####
waarde = 12.34
resultaat = 12.3400
```

```
formaat: ####.##
waarde = 12.34
resultaat = 12.34
```

```
formaat: ###.###
waarde = 1234.5
```

past niet in formaat

formaat: ###.###

waarde = 2.2

resultaat = 2.200

RIGHT\$

STR\$(m TO)

Instructie om het rechter deel van een string-variabele te bewerken. De meer flexibele instructie STR(m TO) van de Spectrum biedt een soortgelijke oplossing.

```
10 REM *** simulatie RIGHT$  
20 LET a$="ZX Spectrum"  
30 LET b$=a$(8 TO )  
40 PRINT b$
```

trum

SCR

NEW

De functie SCRATCH, ofwel het wissen van de geheugeninhoud, wordt in Spectrum-BASIC uitgevoerd met de instructie NEW.

SEG\$

STR\$

De instructie SEG\$ komt overeen met de instructie STR\$. Beiden berekenen ze een deel van de string-variabele.

WAIT

PAUSE

Komt overeen met de instructie PAUSE van Spectrum-BASIC.

3.1.7 Machinetaal; een inleiding

Uit het oogpunt van volledigheid en om de mogelijkheden van de Spectrum te kunnen beoordelen dienen eerst de mogelijkheden van de assembleertaal te worden toegelicht. Immers, evenals bij de ZX80 en de ZX81 is het binnen het kader van een BASIC-programma mogelijk om met de Spectrum instructies in machinetaal in de vorm van een hexadecimale code in te voeren. Voor een rationeel gebruik van deze zeer efficiënte maar bewerkelijke programmeermethode is in elk geval een grondige kennis vereist van het Spectrum-systeem, van de geheugenorganisatie en bovenal van de logica en van de machinetaal van de Z80A-processor waarmee het systeem is uitgerust.

Ongetwijfeld zullen gebruikers van een ZX80 of van een ZX81, die met deze werkwijze begonnen zijn, weinig moeite hebben om zich op de Spectrum in te werken. Ze zullen niettemin het geheugenbeheer moeten bestuderen en bepaalde invoerroutines van hun assembleertaal moeten herzien. De prestaties van de Spectrum werden voor wat betreft het BASIC duidelijk verbeterd, dit boekje getuigt daarvan. De mogelijkheden van de computer werden gewijzigd en de opgedane ervaring op de juiste wijze gestructureerd. Het leek ons niet juist om binnen het kader van dit boekje de machinetaal van de Spectrum te behandelen, maar niettemin opent paragraaf 3.1.8 de deur naar het programmeren in machinetaal.

3.1.8 De werking van de microprocessor Z80A

Het hart van de Spectrum wordt gevormd door een microprocessor van het type Z80A. Daarbij horen twee types geïntegreerde schakelingen, en wel: een ROM (Read Only Memory) of leesgeheugen en een RAM (Random Access Memory) of willekeurig toegankelijk geheugen. Bij de microprocessor wordt gebruik gemaakt van micro-instructies of machinetaal. Met behulp van deze taal wordt vanuit de ROM een reeks van gegevens opgehaald. Deze bevatten de grammatica-regels van het BASIC-systeem. Aan de hand van de door de ROM afgegeven richtlijnen haalt de microprocessor de variabele elementen op die in de RAM liggen opgeslagen, bijvoorbeeld: de instructies van het BASIC-programma en de bijbehorende variabelen.

Uitvoeren van een BASIC-programma betekent vertalen van de BASIC-commando's en instructies in machinetaal. Deze worden vervolgens met grote snelheid in een zeer groot aantal micro-instructies uitgevoerd. De machinetaal bestaat, evenals BASIC, uit instructies en commando's. De flexibiliteit daarvan is beperkt. De registers van het systeem worden aangeduid met A, B, C, D, E, H en L. Deze kunnen gehele getallen van 0 tot en met 255 bevatten (het grootste getal dat met een binair getal van 8 bits kan worden voorgesteld). Om wille van de eenvoud geeft men de voorkeur aan de hexadecimale code (0 tot 16 decimaal). Hierbij gebruikt men de tekens 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E en F om de cijfers 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14 en 15 voor te stellen. Samengevat: een groep van twee hexadecimale cijfers stelt voor: voor de tientallen een veelvoud van zestien maal tien, en voor de eenheden een veelvoud van zestien maal één. Het is niet nodig om symbolische conversies uit te voeren, de aanduiding van de grootte-orde is voldoende. Een variabele in machinecode kan dus een waarde tussen 00 en FF bezitten. De microprocessor houdt niet altijd rekening met fouten in zijn registers. Optellen van twee getallen waarbij de uitkomst de waarde FF (hex.) overschrijdt brengt dan ook een overschrijding van de capaciteit teweeg. Alleen de beide laatste cijfers blijven behouden.

Bekijken we nu eens een stukje machinetaal:

LD A,9A	deze instructie komt overeen met de BASIC-code LET
---------	--

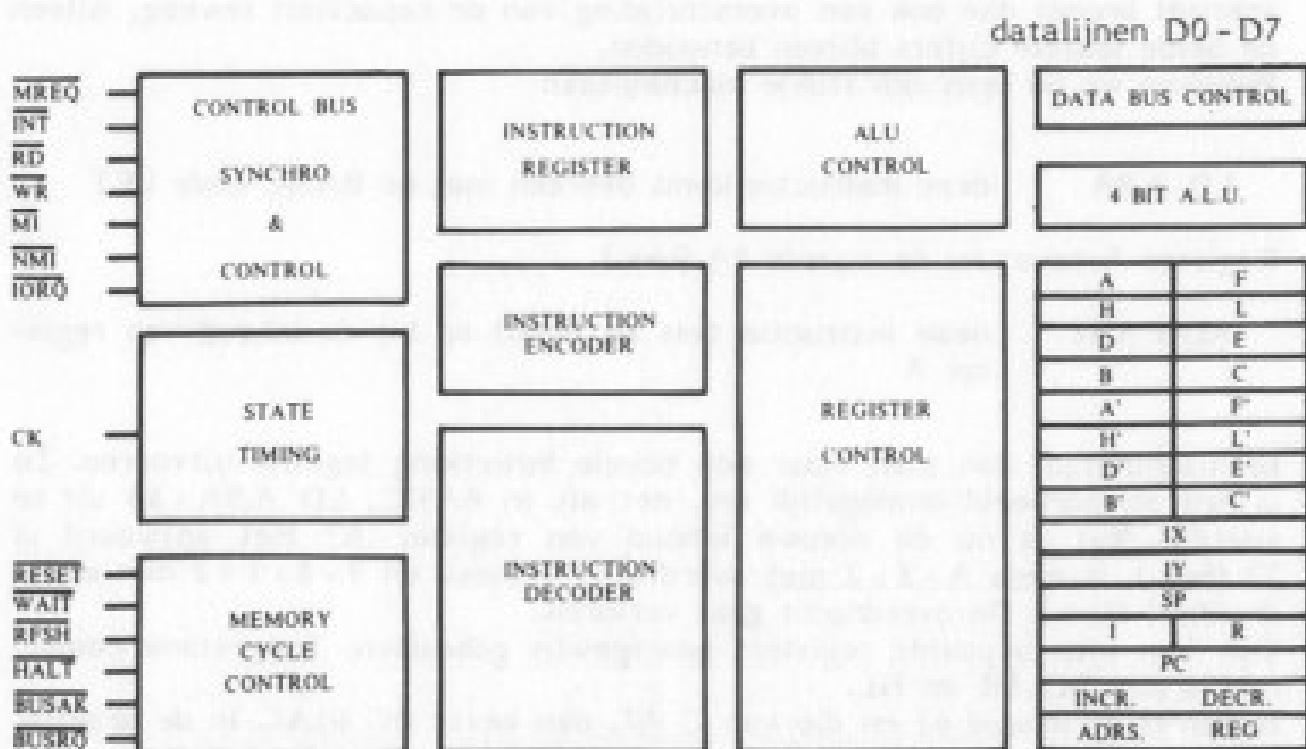
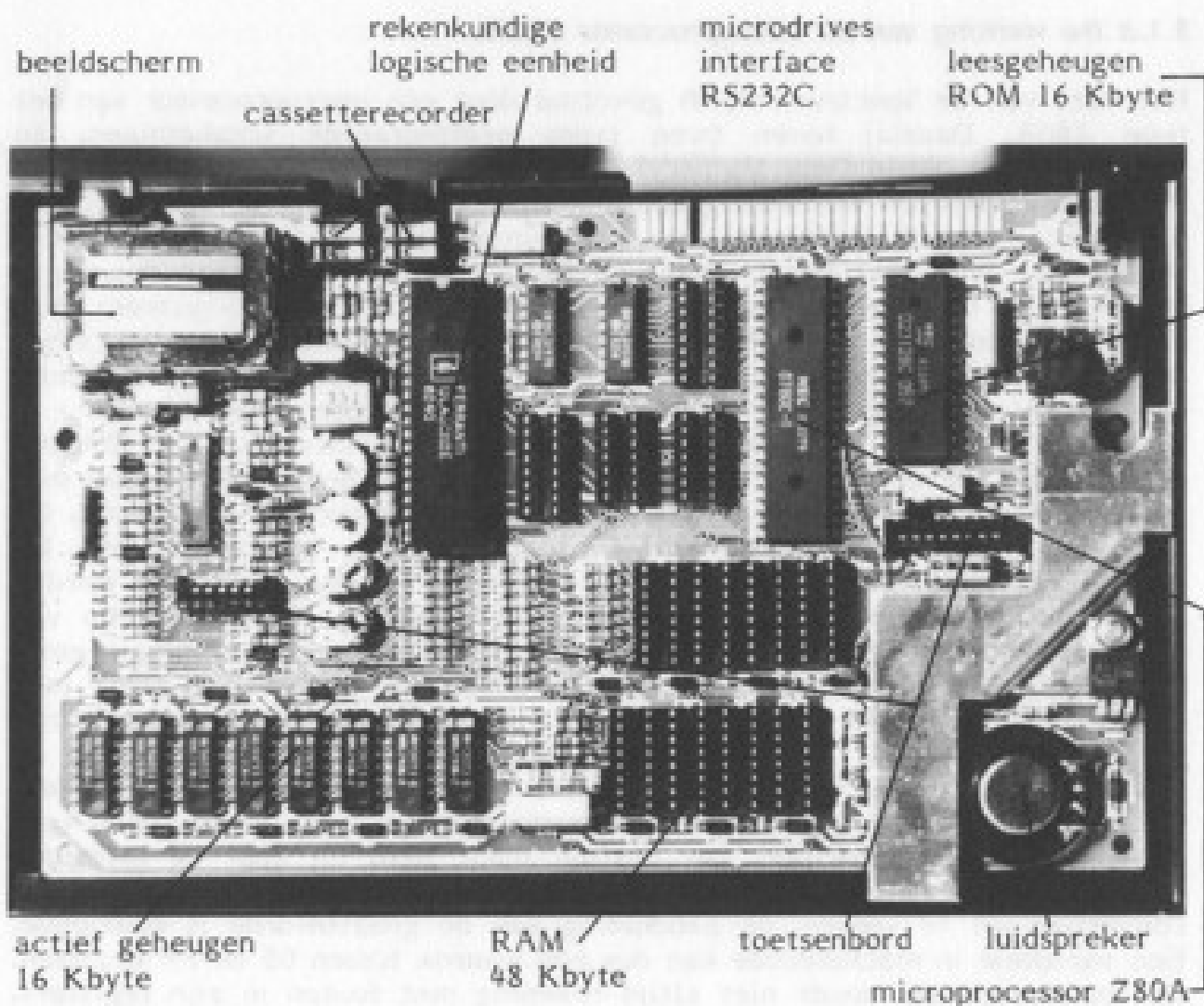
Register A bevat nu de waarde 9A (hex.).

ADD A,88	deze instructie telt 88 (hex.) op bij de inhoud van register A
----------	--

In machinetaal kan men maar een enkele bewerking tegelijk uitvoeren. Zo is het bijvoorbeeld onmogelijk om, net als in BASIC, LD A,9A+88 uit te voeren. Wat is nu de nieuwe inhoud van register A? Het antwoord is 22 (hex.), immers $A + 8 = 2$ met overdracht 1 (hex.) en $9 + 8 + 1 = 2$ met overdracht 1 (hex.). De overdracht gaat verloren.

Ook kan men bepaalde registers paarsgewijs gebruiken. Toegestane combinaties zijn BC, DE en HL.

Is van B de inhoud 61 en die van C A7, dan bevat BC 61A7. In de praktijk wordt elke machine-instructie voorgesteld door een informatiecode, ook wel mnemonic genoemd, die tevens de operationele code (OPCODE) is.



Wat de Z80A dus nodig heeft zijn de codes (hex. codes) dan wel de hexadecimale vertaling van de mnemonics.

Voorbeeld: - Het laden van het registerpaar BC laat zich uitdrukken in de mnemonic LD BC. De hexadecimale code daarvan luidt 01.
 - Het terugkeren uit de subroutine, ook wel gebruikt om naar BASIC terug te keren laat zich uitdrukken in de mnemonic RET. De hexadecimale code daarvan luidt C9.

We zijn nu zover dat met een toepassingsvoorbeeld begonnen kan worden. Is de Spectrum ingeschakeld, dan wacht hij op een BASIC-instructie. Om ons in de interne besturing van de processor te mengen geven we de instructie USR waarvan de variabele het beginadres van het programma in machinetaal aangeeft.

Elk geheugenadres bestaat uit een paar hexadecimale codes, zoals het voorbeeld laat zien:

ADRES	INHOUD	BETEKENIS VAN DE MNEMONIC	VERTALING
24567	01 (H)	LD BC	OPCODE
24577	00 (H)	0 0	waarde B
24578	00 (H)	0 0	waarde C
24579	C9 (H)	RET	OPCODE

Stel nu dat we de Spectrum de instructie USR 24567 geven. De Spectrum voert daarop de reeks instructies uit die hij aantreft om ten slotte, nadat de instructie C9 (RET) is uitgevoerd, terug te keren naar BASIC. De inhoud van registerpaar BC is nu gelijk aan nul.

Teneinde de amateur iets van de machinetaal te laten proeven volgt hieronder de opbouw van het geheugen zoals de constructeur die aan ons verstrekt.

0	4000 (H) 16384 (D)	8000 (H) 32768 (D)	FFFF (H) 65535 (D)
ROM	RAM	RAM	

BEELDSCHERM- GEHEUGEN	ATTRIBUTEN- GEHEUGEN	PRINTER- BUFFER	SYSTEEM- VARIABELEN	MICRODRIVE- MAPS	CHANNEL- INFORMATIE	80 (H)
16384	22528	23296	23552	23734	CHAN5	

PROGRAMMA- GEHEUGEN	VARIABELEN- GEHEUGEN	80 (H)	EDIT- GEHEUGEN	NL	800 (H)	WERK- GEHEUGEN
PROG	VARS		ELINE			WORKSP

REKEN- STACK	VRIJE GEHEUGENRUIMTE	MACHINE- STACK	GO SUB- STACK	?	3E (H)	USER DEFINED GRAPHICS
STKBOT	STKEND	sp			RANTOP UDG	PRAHT

Een dergelijke beschrijving heeft zeker zijn nut, bijvoorbeeld om met behulp van de instructie POKE op daartoe geschikte plaatsen de geheugeninhoud te kunnen wijzigen. Het lijkt geen enkele twijfel dat de gebruiker dankzij deze instructie over een krachtig gereedschap beschikt om in het systeem in te grijpen. Doelmatig gebruik ervan verhoogt de prestaties. Een onjuist gebruik echter kan helaas vele uren werk teniet doen, gelukkig zonder gevolgen voor het systeem.

Zoals uit de afbeelding van de geheugenindeling blijkt, is het geheugen onderverdeeld in secties waarvan de bestemming wordt bepaald door de toepassing. 'BEELDSCHERMGEHEUGEN' bevat het TV-beeld. De daarvoor gebruikte complexe structuur om de beeldinhoud te kunnen beschrijven vergt voordat er iets kan worden gewijzigd een perfecte beheersing van het operating systeem.

'PRINTER BUFFER' bevat de voor de Sinclair-printer bestemde karakters. Het samenstel 'SYSTEEMVARIABLEN' zoals bijvoorbeeld 'PIP' of de lengte van de 'KLIK' van het toetsenbord, staat beschreven in het handboek van de Spectrum en u doet er goed aan dat te raadplegen.

In 'MICRO DRIVE MAPS' worden gegevens met betrekking tot de microdrives opgeslagen.

3.2 Analyse van de hardware

3.2.1 Algemeen

Het nu volgende hoofdstuk behandelt de besturing door de Spectrum van extern aangesloten apparatuur. Daarin komt alles aan de orde wat betrekking heeft op het beeldscherm, de cassetterecorder, de printer en in beknopte vorm ook de door Sinclair aangekondigde randapparaten zoals die in de nabije toekomst leverbaar zullen zijn; de telecommunicatie-interface en de microdrives. In weerwil van de in de titel genoemde hardware worden ook software-elementen besproken die daarmee onverbrekkelijk zijn verbonden. Aan deze presentatievorm is de voorkeur gegeven teneinde een nuttig gebruik ervan te vergemakkelijken.

3.2.2 TV-ontvanger

Net als zijn voorgangers kan ook de Spectrum op een huiskamer-TV worden aangesloten. Gebruikers van het PAL-systeem zullen geen enkele moeilijkheid hebben om de uit Engeland afkomstige Spectrum te gebruiken. Het verschil in standaardisatie van TV-normen dwingt de gebruikers in Frankrijk er toe hetzij een aan het SECAM-systeem aangepaste Spectrum aan te schaffen dan wel zich te voorzien van een voor PAL-standaard of multi-standaard geschikte TV-ontvanger. Afgezien van dit verschil bestaan er geen verdere karakteristieke verschillen, noch in hardware noch in software.

De Spectrum wordt aangesloten op de antenne-ingang. Goed aangesloten converseert hij met ons via UHF-kanaal 36. Met het geleverde videosignaal kunnen acht kleuren worden weergegeven: blauw, rood, magenta, groen, cyaan (lichtblauw), geel, zwart en wit. Wordt een zwart/wit TV-ontvanger gebruikt, dan zal de Spectrum een grijsschaal van acht grijstinten te zien geven.

3.2.3 Schrijven op het beeldscherm

Evenals de ZX81 adresseert de Spectrum 22 regels van elk 32 karakters.

PRINT	schrijft op de momentele positie van de cursor.
PRINT TAB	schrijft op de getabuleerde positie en op de door de cursor aangegeven regel.
PRINT AT	schrijft op de in coördinaten aangegeven regel/kolom.
PRINT a'b'	schrijft de inhoud tussen de accenten steeds op een nieuwe regel.

```
10 REM *** PRINT op de vol-
20 REM *** gende regel.
30 FOR i=1 TO 5
40 PRINT i;"oppervlakte ";i^2'
   "inhoud      ";i^3: PRINT
50 NEXT i
```

```
1
oppervlakte 1
inhoud      1

2
oppervlakte 4
inhoud      8

3
oppervlakte 9
inhoud     27

4
oppervlakte 16
inhoud     64

5
oppervlakte 25
inhoud    125
```



Het eerste verschil is de mogelijkheid om de nominale schrijffunctie - zwarte letters op een witte ondergrond - door omkeren te wijzigen in witte letters op een zwarte ondergrond. De toets 'CAPS SHIFT' zorgt in combinatie met toets 4 'INVERSE VIDEO' voor de zwarte ondergrond, en in combinatie met toets 3 'TRUE VIDEO' voor een witte ondergrond. Dezelfde toets 'CAPS SHIFT' zorgt in combinatie met toets 2 'CAPS LOCK' voor uitsluitend hoofdletters. Dit is een bistabiele functie: een tweede keer indrukken van deze combinatie keert de functie om. Het laatste verschil betreft het functioneren van de instructie INPUT. Immers hiermee kan een karakter-string worden afgedrukt, waarvan de rol boven alles is, een toelichting op de in te voeren variabele te geven.

De tekst die de instructie INPUT begeleidt wordt op het beeldscherm in het communicatieveld afgebeeld, dus op de regels 23 en 24.

Een tweede variant van de instructie maakt het mogelijk om string-vari-

belen zonder aanhalingstekens in te voeren (het systeem handelt dan alsof ze er wel zijn). De syntax vereist dan wel dat de string-variabele wordt voorafgegaan door het sleutelwoord LINE.

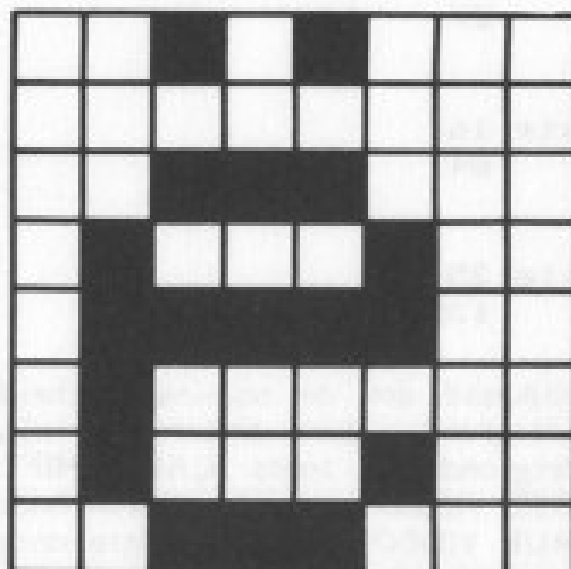
```
10 REM variatie op INPUT #1
20 PRINT "Geef tekst ": INPUT
a$
"L"
```

```
10 REM variatie op INPUT #2
20 PRINT "Geef tekst ": INPUT
LINE a$
```

L

Om geleidelijk aan tot de grafische wereld door te dringen beginnen we met de functie BIN van de Spectrum waarmee 21 verschillende karakters kunnen worden gedefinieerd. Een toepassingsvoorbeeld dat zich vooral heel goed voor de Franse taal leent is het schrijven van kleine letters met accenten. In de van huis uit Engelse karakterset van de Spectrum is daarin niet voorzien.

Met de instructie BIN kunnen 64 elementen worden gedefinieerd waarmee de karakters uit acht lijnen van elk acht punten worden samengesteld. Een karakter wordt opgebouwd met acht BIN-instructies, die elk één van de acht samenstellende lijnen van het karakter definiëren.



```
BIN 0 0 1 0 1 0 0 0
BIN 0 0 0 0 0 0 0 0
BIN 0 0 1 1 1 0 0 0
BIN 0 1 0 0 0 1 0 0
BIN 0 1 1 1 1 1 0 0
BIN 0 1 0 0 0 0 0 0
BIN 0 1 0 0 0 1 0 0
BIN 0 0 1 1 1 0 0 0
```

```
10 REM programmeerbare
20 REM graphics
30 REM
40 REM eerste methode
50 FOR n=0 TO 7
60 INPUT bytewaarde
70 PRINT n,bytewaarde
```

```

80 POKE USR "E"+n,bytewaarde
90 NEXT n
100 PRINT "E"

```

```

10 REM programmeerbare
20 REM graphics
30 REM
40 REM tweede methode
50 FOR n=0 TO 7
60 READ r
70 POKE USR "E"+n,r
80 NEXT n
90 DATA BIN 00101000,BIN 00000
000,BIN 00111000,BIN 01000100,BI
N 01111100,BIN 01000000,BIN 0100
0100,BIN 00111000
100 PRINT "E"

```

```

10 REM programmeerbare
20 REM graphics
30 REM
40 REM derde methode
50 FOR n=0 TO 7
60 READ r
70 POKE USR "E"+n,r
80 NEXT n
90 DATA 40,0,56,60,124,64,60,5

```

6

```
100 PRINT "E"
```

Hierboven het voorbeeld van het karakter *ë* en het voor de samenstelling ervan benodigde programma. De instructie POKE USR geeft tussen aanhaalingstekens de 'naam' van de letter van het alfabet die als referentie dient voor het samen te stellen karakter.

Terloops zullen we de stuurinstructie SCREEN\$(l,k) noemen die het op de genoemde coördinaten lijn-kolom (l,k) aanwezige karakter aangeeft. Zie ook het conversieprogramma 'van decimaal naar binair' in hoofdstuk 3.2.9.

3.2.4 Invoeren van kleur

Het oppervlak van het beeldscherm is onderverdeeld in twee velden: de voorgrond waarop wordt geschreven, het papier (Engels: PAPER), en de achtergrond of kantlijn (Engels: BORDER) die het blad omgeeft. Het hoeft geen betoog dat voor het schrijven van karakters net als op het palet van een schilder een stel kleuren (Engels: INK) beschikbaar is. De kantlijn (BORDER), het beeldvlak (PAPER) en de inkt (INK) kunnen onafhankelijk van elkaar een van de acht mogelijk kleuren aannemen. Bij het inschakelen bepalen de initialiseringscondities de kleur van de kantlijn, ongeacht de kleur van het papier, dat wil zeggen WIT. De kleur van de

inkt is ZWART. Om nog even te herhalen, de acht mogelijke kleuren zijn:

- 0 = zwart
- 1 = blauw
- 2 = rood
- 3 = lichtpaars of magenta
- 4 = groen
- 5 = lichtblauw of cyaan
- 6 = geel
- 7 = wit

Aan elk karakter kunnen kenmerken worden meegegeven. Hiermee is het mogelijk om de helderheid (BRIGHT) te veranderen. Ook is mogelijk: knipperen (FLASH), dit is het met regelmatige tussenpozen laten omkeren van de voor inkt (INK) en papier (PAPER) gekozen kleuren. Een laatste opmerkelijk kenmerk is de instructie OVER waarmee twee karakters over elkaar heen tot een samengesteld karakter kunnen worden afgedrukt. Een toepassingsvoorbeeld daarvan zou heel goed het schrijven van letters met accenten kunnen zijn. Ten slotte geeft de instructie ATTR verkorting van ATTRIBUTE in gecodeerde vorm de karakteristieken en kenmerken van een met lijn/kolom op het beeldscherm aangegeven positie.

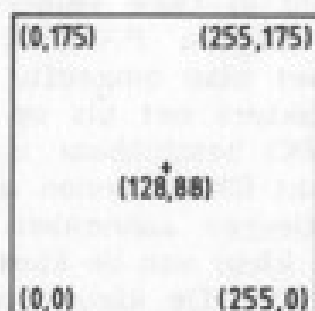
Samengevat:

Acht kleuren.

- BRIGHT - twee helderheden voor de kleur.
- BORDER - bepaalt de kleur van de kantlijn (achtergrond).
- PAPER - bepaalt de kleur van het papier (voorgrond).
- INK - bepaalt de kleur waarin de karakters worden geschreven.
- FLASH - knipperen door omkeren van de kleuren van PAPER en INK.
- OVER - over elkaar heen schrijven van twee karakters.
- ATTR - inbrengen van de kenmerken van een gegeven karakter.

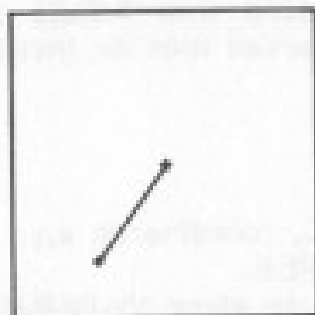
3.2.5 De grafische taal van de Spectrum

Wordt, door toets 'CAPS SHIFT' gecombineerd met toets '9' te gebruiken, de grafische functie ingeschakeld, dan wijzigt de Spectrum de definitie van het voor de gebruiker toegankelijke beeldveld van (32 x 8) bij (22 x 8) in (256 x 192) = 704 beeldpunten (Engels: PIXELS = Picture ELEMENTS). Elk PIXEL (beeldelement) wordt afzonderlijk opgeslagen en is toegankelijk via een stelsel van coördinaten waarvan de oorsprong zich in de linker benedenhoek van het beeldscherm bevindt.



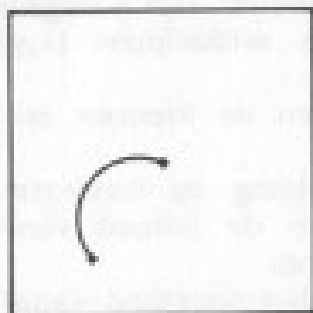
- 10 REM een punt in het midden
- 20 REM van het beeldscherm
- 30 REM
- 40 PLOT 128,88

Een punt op het beeldscherm wordt gedefinieerd met behulp van de instructie PLOT en de coördinaten (x,y) van het betreffende punt.



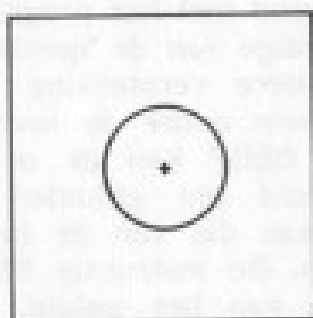
```
10 REM een vector vanuit het  
20 REM midden van het scherm  
30 REM naar linksonder  
40 REM  
50 PLOT 128,88  
60 DRAW -64,-44
```

De Spectrum beschikt tevens over een instructie waarmee tussen twee PIXELS een rechte lijn kan worden getrokken. Met behulp van het sleutelwoord DRAW legt men de relatieve coördinaten van het tweede PIXEL vast.



```
10 REM een cirkelboog vanuit  
20 REM het midden van het  
30 REM beeldscherm naar  
40 REM linksonder  
50 PLOT 128,88  
60 DRAW -64,-44,PI/3
```

Met een derde parameter van de instructie DRAW kan een cirkelboog worden vastgelegd waarvan de lengte wordt uitgedrukt in radialen. De richtingszin van de cirkelboog wordt bepaald door het teken van de parameter.



```
10 REM een cirkel in het  
20 REM midden van het scherm  
30 REM met een straal van 44  
40 REM  
50 PLOT 128,88  
60 CIRCLE 128,88,44
```

Ook het tekenen van cirkels is één van die krachtige instructies van de grafische taal van de Spectrum. De parameters van de instructie CIRCLE leggen de coördinaten van het middelpunt en de lengte van de straal vast. Opgemerkt dient te worden dat de instructies om de attributen te definiëren ook bij de grafische functies horen:

PAPER, INK, FLASH, BRIGHT, INVERSE, OVER

Met de instructie POINT kan de kleur van een PIXEL worden opgevraagd. Deze informeert voor een bepaalde coördinaat (x,y) omtrent de kleur van

INK of PAPER op het betreffende punt. Hoewel niet tot deze groep behorend kan met het commando 'SAVE naam SCREEN\$' of 'SAVE naam CODE 16384,6912' een grafische afbeelding naar cassette worden weggeschreven. De opname daarvan op band kan worden gecontroleerd met behulp van de instructie 'VERIFY naam SCREEN\$' en het laden ervan met de instructie 'LOAD naam SCREEN\$'.

Samenvatting:

PLOT x,y	tekent een punt, kleur INK, coördinaten x,y.
PLOT INVERSE I;	tekent een punt, kleur PAPER.
PLOT OVER I;	tekent een punt en keert de kleur PAPER/INK om.
DRAW $\pm x, \pm y$	tekent een cirkelboog vanuit de laatste positie PLOT naar het door de coördinaten gegeven punt.
DRAW $\pm x, \pm y, \pm z$	tekent een cirkelboog waarvan de hoek en de richting in de derde parameter zijn vastgelegd.
CIRCLE x, y, r	trekt een cirkel met als middelpunt (x,y) en als straal (r).
POINT (x,y)	bepaalt het verband tussen de kleuren en een PIXEL (PAPER/INK).
SAVE - SCREEN\$	bergt de grafische afbeelding op cassette op.
VERIFY - SCREEN\$	verifieert door vergelijken de inhoud van een grafisch beeldschermbestand.
LOAD - SCREEN\$	laadt een grafisch beeldschermbestand vanaf de cassette.

3.2.6 De Spectrum muzikaal

Van alle nieuwe en voor de Spectrum specifieke instructies levert deze problemen op! Deze houden namelijk niet precies verband met het programmeren van een algoritme, hoewel ze wel in het inwendige van de Spectrum worden gegenereerd. Omdat het resultaat voor verdere versterking aan de aansluitbus EAR beschikbaar komt, is besloten hem onder de invoer/uitvoerinstructies op te nemen. Met de instructie BEEP kan de in de Spectrum ingebouwde luidspreker worden geactiveerd om geluiden te produceren waarvan de toonhoogten zijn aangepast aan die van de tonen van het 'Wohltemperierte Klavier' van mijnheer Bach. De instructie BEEP heeft twee parameters: de duur en de toonhoogte van het geluid. De duur wordt bepaald in seconden of in decimale breuken daarvan en de hoogte door vergelijking met de noot DO van het derde octaaf van een piano. Toonhoogte-wijzigingen komen tot stand door met een halve toon te schuiven. Ter herinnering, de sprongen tussen de noten zijn:

0	1	2	3	4	5	6	7	8	9	10	11	12
DO	DO#	RE	RE#	MI	FA	FA#	SOL	SOL#	LA	LA#	SI	DO

Opgemerkt dient te worden dat er tussen MI en FA en tussen SI en DO een halve toon ligt. Tussen alle andere tonen ligt een sprong van een hele toon. De toonladder van DO-majeur, dat wil zeggen uitsluitend de witte toetsen

van de piano, is met het volgende programma ten gehore te brengen:

```
10 REM toonladder van C
20 BEEP .5,0: BEEP .5,2: BEEP
.5,4: BEEP .5,5: BEEP .5,7: BEEP
.5,9: BEEP .5,11: BEEP .5,12: S
TOP
```

De natuurlijke toonladder, die van de viool, kan met het volgende programma ten gehore worden gebracht:

```
10 REM de natuurlijke schaal
20 REM zoals de violist die
30 REM speelt.
40 BEEP .5,0: BEEP .5,2.039: B
EEP .5,3.86: BEEP .5,4.98: BEEP
.5,7.02: BEEP .5,8.84: BEEP .5,1
0.88: BEEP .5,12: STOP
```

Met behulp van een lus wordt de toonhoogte glijdend gemaakt zodat die aan de toonaard van science-fiction-muziek doet denken:

```
10 REM snelle toonwisselingen
20 REM
30 FOR n=10 TO 50
40 BEEP .02,n
50 NEXT n
```

3.2.7 Cassetterecorder als randapparaat

Van alle goedkope, maar belangrijke randapparaten voor de Spectrum behoort de cassetterecorder zeker tot de voornaamste. In afwachting van de microdrives is de cassette in feite het medium bij uitstek om gegevens en programma's die men door de Spectrum wil laten verwerken op te bergen. Evenals bij zijn voorgangers de ZX80 en de ZX81 kan ook bij de Spectrum met een audio-cassetterecorder van middelmatige kwaliteit worden volstaan. Het is opvallend dat die middelmatige apparaten voor onze applicatie in aanmerking komen.

Het op te nemen audiosignaal, dat door de Spectrum voor opname wordt afgegeven, is een combinatie van twee tonen die overeenkomen met de beide door de Spectrum afgegeven binaire tekens. Om enkele technische redenen, zoals de bandbreedte, het signaalniveau en faseverschil, geeft men er de voorkeur aan een eenvoudige, robuuste mono-recorder te gebruiken.

Onder de tegen de werking van de ZX81 ingebrachte bezwaren zijn er twee die betrekking hebben op het opnemen van programma's. De eerste betreft de uitzonderlijk gevoelige signaalniveau-instelling tijdens opname/weergave, de tweede de werksnelheid. Het blijkt dat de research- en ontwikkelingsafdelingen van Sinclair de lucht van deze klachten hebben gekregen. De Spectrum doet dan ook zijn voordeel met geheel herziene elektronische datatransmissieschakelingen, zowel vanuit het oogpunt van

techniek als voor wat betreft de snelheid. De transmissiesnelheid is vijf maal groter dan die van zijn voorgangers. De signaalgevoeligheid is gewijzigd en bij diezelfde gelegenheid werden ook de mogelijkheden tot verificatie en informatie tijdens de uitvoering verbeterd.

De Spectrum maakt voor de vier voor dataverwerking belangrijkste bewerkingen gebruik van de cassetterecorder. Wegschrijven naar cassette, verifiëren van de weggeschreven gegevens door die te vergelijken met de inhoud van het geheugen, laden van het geheugen vanaf de cassette, en samenvoegen, waarbij in het geheugen van de cassette afkomstige gegevens worden samengevoegd met de reeds in het geheugen opgeslagen gegevens. Deze vier bewerkingen worden geïnitieerd door instructies, de bijbehorende parameters hebben betrekking op het soort gemanipuleerde informatie. Tijdens de uitvoering wordt informatie omtrent de lopende activiteiten verstrekt door signalen die ontstaan uit de kleurveranderingen van de kantlijn van het beeldscherm en de frequentie van die veranderingen. Op de voorgrond van het beeldscherm wordt in klare taal het soort gegeven en de eventuele naam van het lopende gegeven beschreven.

Zo wordt bijvoorbeeld het laden van het geheugen opgedragen met de instructie LOAD, eventueel gevolgd door de naam van het programma tussen aanhalingstekens. In afwachting van de opdracht is de kantlijn CYAAN van kleur. Bij ontvangst van het synchronisatietoontje licht de beeldrand ROOD op en die twee kleuren wisselen elkaar af tot de gegevens binnenkomen die met de naam van het programma overeenkomen. Op dat moment verdubbelt de toonhoogte, het ritme van het geluid komt tot uiting in een snelle wisseling van de kleuren GEEL en BLAUW van de kantlijn. Na het lezen van de titel herneemt de kantlijn de afwisselende kleuren CYAAN/ROOD en de voorgrond geeft de naam van het gevonden programma. Komt de naam van het programma overeen met de opgedragen naam, dan zorgt het systeem voor het laden van dat programma en licht de beeldrand opnieuw in de kleuren BLAUW/GEEL op, tot de handeling is voltooid.

De kleuren van het beeldscherm zijn afhankelijk van de initialisering en van de condities van de uitvoering zoals die zijn vastgelegd in het programma.

De vier soorten commando's zijn:

SAVE	voor het laden op cassette,
VERIFY	voor het verifiëren door vergelijken met de in het geheugen vastgelegde informatie,
LOAD	voor het laden van het geheugen na initialisatie ervan, en
MERGE	voor het laden van het geheugen met veiligstelling van de inhoud ervan en samenvoegen van nieuwe variabelen en programmaregels met de voorgaande.

De intrinsieke mogelijkheden maken gelijktijdig manipuleren met programma's en variabelen mogelijk, zowel met getal- en string-variabelen als het laden van bytes.

Omdat niet met alle commando's alle manipulaties mogelijk zijn, zijn

de mogelijkheden in de vorm van een tabel samengevat:

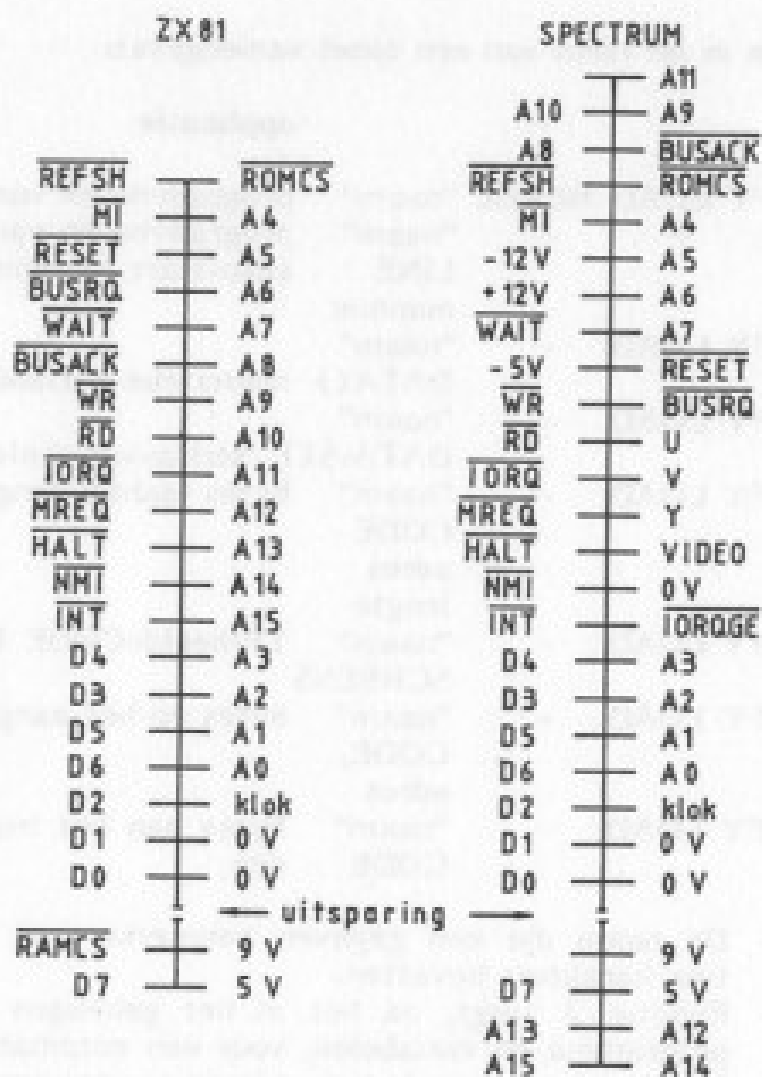
commando's				applicatie		
1	SAVE	VERIFY	LOAD	MERGE	"naam"	programma en variabelen
2	SAVE	-	-	-	"naam"	programma en variabelen
					LINE	auto-start regelnummer
					nummer	
3	SAVE	VERIFY	LOAD	-	"naam"	DATA() numerieke variabelen
4	SAVE	VERIFY	LOAD	-	"naam"	DATA-\$() string-variabelen
5	SAVE	VERIFY	LOAD	-	"naam"	bytes (aantal=lengte)
					CODE	
					adres	
					lengte	
6	SAVE	VERIFY	LOAD	-	"naam"	TV-beeld=CODE 16384,6912
					SCREEN\$	
7	-	VERIFY	LOAD	-	"naam"	bytes op het aangegeven adres
					CODE,	
					adres	
8	-	VERIFY	LOAD	-	"naam"	bytes aan het begin van het laden
					CODE	

Opmerkingen: - De naam die een gegeven kenmerkt mag niet meer dan tien karakters bevatten.
 - Functie 2 zorgt, na het in het geheugen laden van het programma en variabelen, voor een automatische start van het programma op het genoemde regelnummer.

3.2.8 Op de uitbreidingsconnector aan te sluiten randapparatuur

De eerste opmerking die zich opdringt als men de uitbreidingsconnectors van de ZX81 en van de Spectrum met elkaar vergelijkt is dat ze onderling niet gelijk zijn. De ZX81 heeft een connector met twee rijen van 23 contacten en een sleutel (zie afbeelding) op de plaats van het derde contact. De Spectrum heeft een connector met twee rijen van 28 contacten waarbij de sleutel zich ter hoogte van het vijfde contact bevindt. Voor zover de aansluiting wordt uitgevoerd met behulp van een stekker met aan de uiteinden open aansluitpennen, zijn deze laatsten fysisch compatibel. Bij een nauwkeuriger onderzoek blijkt dat de signalen niet op dezelfde wijze over de connector zijn verdeeld. Dit beperkt de compatibiliteit. De configuratie van de signalen maakt het onmogelijk om het 16 Kbyte uitbreidingsgeheugen van de ZX81 op de Spectrum aan te sluiten. De compatibele signalen maken het in elk geval wel mogelijk om de Sinclair-printer te gebruiken.

Onder de signalen waarmee de connector van de ZX81 is aangevuld treft men meerdere voedingsspanningen aan evenals een of ander mysterieus VIDEO-sigitaal. In de wetenschap van het project van de miniatuur TV-ontvanger met plat beeldscherm van Clive Sinclair, kan men zich afvragen of er in de toekomst een aan het Spectrum-concept aangepast video-



←
bovenkant

randapparaat verkrijgbaar zal zijn, waarmee het mogelijk zal zijn een compleet systeem in een zeer beperkte ruimte onder te brengen; in een diplomatenkoffertje bijvoorbeeld. Bedenkt men verder dat men die TV-ontvanger met plat beeldscherm tegen de prijs van een diskettestation denkt aan te bieden, dan zal dat voor concurrenten die integrale systemen aanbieden een reden te meer zijn om voor wat betreft de prijs/prestatie-verhouding zeker met de Spectrum rekening te houden.

3.2.9 Databeheer toegankelijk via de uitbreidingsconnector

Bekijkt men de op de uitbreidingsconnector aanwezige signalen nader, dan kan men daarin vier soorten onderscheiden:

1. Voedingsspanningen: 0 V, +5 V, -5 V, +12 V, -12 V.
2. Stuursignalen: TORQ, HALT, ROMCS enz.
3. Adressignalen: A0, A1, ... A15.
4. Datasignalen: D0, D1, ... D7.

We hebben al gezien dat de microprocessor Z80A de toegang tot het ge-

heugen (RAM en ROM) bestuurt door de beide geheugens te adresseren. Er bestaan $2^{16} = 65536$ adressen (A0-A15). Behalve dat adressen toegang tot het geheugen geven, maken ze het ook mogelijk externe apparaten te gebruiken en over de datalijnen (D0-D7) binaire informatie door te geven. De Z80A van de Spectrum werkt met de adresbits A0, A1, A2, A3 en A4 voor het aansturen van externe apparaten. De adreslijnen A5, A6 en A7 worden niet gebruikt en zijn dus beschikbaar voor de gebruiker. Om de signalen geldig te gebruiken moeten de bits A0 tot en met A4 een logisch '1'-niveau voeren.

Om het gebruik van data te vergemakkelijken beschikt het Spectrum-BASIC over twee instructies voor invoer/uitvoerbesturing, die lijken op de instructies PEEK en POKE waarmee toegang tot het geheugen wordt verkregen.

IN-adres: leest het aan de geadresseerde poort aangeboden gegeven.

OUT-adres: voert het door de waarde gedefinieerde gegeven uit via de geadresseerde poort.

In de praktijk gebruikt de Spectrum die functies zelf. Verificatie ervan is gemakkelijk, deze is beschreven in het gebruikershandboek en beschrijft het gebruik van het toetsenbord. De besturing van het toetsenbord is opgebouwd rond het adres 254, in de leesfunctie om te bepalen welke van de 40 toetsen werd ingedrukt. Het toetsenbord is verdeeld in acht zones van vijf toetsen.

Het adres wordt afgeleid uit de vergelijking:

$$ADRS = 254 + 256 * (255 - 2^Z) \quad Z = 0 \text{ t/m } 7$$

ADRS	Z	D0	D1	D2	D3	D4	D4	D3	D2	D1	D0	Z	ADRS
63486	3	1	2	3	4	5	6	7	8	9	0	4	61438
64510	2	Q	W	E	R	T	Y	U	I	O	P	5	57342
65022	1	A	S	D	F	G	H	J	K	L	(E)	6	49150
65278	0	(C)	Z	X	C	V	B	N	M	(S)	(B)	7	32766

Voert men het volgende programma uit:

```

10 REM gebruik van I/O
20 REM
30 REM stoppen met BREAK
40 REM
50 INPUT "nummer van de toetse
rij ";z
60 LET data=254+256*(255-2^z)
70 PRINT AT 0,0;"rij ";z;TAB 8
;"adres ";data;AT 0,20;"data ";I
N data
80 GO TO 60

```

```
rij 1   adres 65022 data 247
```

dan verkrijgt men voor de variabele data één van de volgende waarden, die omgezet in binaire code overeenstemmen met:

DATA	D7	D6	D5	D4	D3	D2	D1	D0
255	1	1	1	1	1	1	1	1
254	1	1	1	1	1	1	1	0
253	1	1	1	1	1	1	0	1
251	1	1	1	1	1	0	1	1
247	1	1	1	1	0	1	1	1
239	1	1	1	0	1	1	1	1

Men kan hieruit afleiden dat de datalijn die overeenkomt met de ingedrukte toets bij het indrukken ervan naar het '0'-niveau omslaat.

Voor hen die dit resultaat willen verifiëren volgt hieronder een kort conversie-programma aan de hand waarvan men zich kan overtuigen. Wordt het achter het voorgaande programma uitgevoerd, dan kan bij regel 50 worden begonnen.

```

10 REM conversie van decimaal
20 REM naar binair
30 REM
40 INPUT "decimale waarde? ";d
ata
50 LET a$="": LET deci=data
60 FOR n=0 TO 7
70 LET bin=deci-INT (deci/2)*2
80 IF bin=0 THEN LET a$="0"+a
$
90 IF bin<>0 THEN LET a$="1"+
a$
100 LET deci=INT (deci/2)
110 NEXT n
120 PRINT AT 0,11;"bin"
130 PRINT AT 1,9;"d 76543210"
140 PRINT AT 2,0;"data ";data;A
T 2,11;a$
150 GO TO 20

```

```

          bin
d 76543210
data 1    00000001

```

```

          bin
d 76543210
data 10   00001010

```

bin


```

      d 76543210
data 100  01100100

```

```

      bin
      d 76543210
data 255  11111111

```

Voor hen die de mogelijkheden van dit programma wensen uit te breiden zorgt de volgende uitbreiding voor de hexadecimale conversie.

```

1 REM aanvullende routine
2 REM om ook hexadecimale
3 REM waarden te verkrijgen
4 REM
5 REM
150 PRINT AT 0,20;"hex"
160 LET a=48: LET h1=INT (data/
16)
170 IF h1>9 THEN LET a=55
180 LET h$=CHR$ (h1+a): LET a=4
8
190 LET h2=data-(INT (data/16)*
16): IF h2>9 THEN LET a=55
200 LET e$=CHR$ (h2+a): LET d$=
h$+e$
210 PRINT AT 2,20;d$
220 GO TO 40

```

```

1 REM conversie van getallen
2 REM van decimaal naar
3 REM binair en hexadecimaal
4 REM
5 REM
40 INPUT "decimale waarde? ";d
ata
50 LET a$="": LET deci=data
60 FOR n=0 TO 7
70 LET bin=deci-INT (deci/2)*2
80 IF bin=0 THEN LET a$="0"+a
$
90 IF bin<>0 THEN LET a$="1"+
a$
100 LET deci=INT (deci/2)
110 NEXT n
120 PRINT AT 0,11;"bin"
130 PRINT AT 1,9;"d 76543210"
140 PRINT AT 2,0;"data ";data;A
T 2,11;a$
150 PRINT AT 0,20;"hex"
160 LET a=48: LET h1=INT (data/
16)

```

```

170 IF h1>9 THEN LET a=55
180 LET h#=CHR# (h1+a): LET a=4
8
190 LET h2=data-(INT (data/16)*
16): IF h2>9 THEN LET a=55
200 LET e#=CHR# (h2+a): LET d#=
h#+e#
210 PRINT AT 2,20;d#
220 GO TO 40

```

```

          bin      hex
d 76543210
data 200  11001000 C8

```

Enkele goede adressen om te onthouden:

ADRES	TYPE	BIT	APPLICATIE
254	UIT	D4	luidspreker
254	UIT	D3	microfoonaansluiting
254	UIT	D2	kleur van de rand
254	UIT	D1	
254	UIT	D0	
251	IN		
251	UIT		printerbesturing beschikbaarheidstest
254			uitvoer van de waarde
247			zijn beschikbaar voor
239			nieuwe randapparatuur

3.2.10 ZX printer

De voor de gebruiker van de ZX81 reeds beschikbare Sinclair-printer blijkt volledig compatibel met de Spectrum. De lage prijs ervan dringt de kritieken van tegenstanders, als de benodigde gealuminiseerde papiersoort, geringe breedte en de zeer trage werking, naar de achtergrond. De aan de Spectrum aangepaste Sinclair-printer heeft onbetwist voordelen. Men kan er zonder beperkingen de beeldscherm inhoud mee afdrukken met de hele karakterset van hoofdletters, kleine letters, grafische of door de gebruiker gedefinieerde tekens evenals van punt-naar-punt geconstrueerde grafische afbeeldingen. Waar het eigenlijk alleen maar aan ontbreekt is kleur. Maar Clive Sinclair denkt ook daaraan...

De Spectrum-commando's voor de printer zijn dezelfde als die voor de ZX81 en wel:

COPY	levert een afdruk van de beeldscherm inhoud.
LPRINT	de instructie PRINT, aangepast aan de regeldrukker. De kenmerken van TAB, AT en ' zijn eveneens beschikbaar.
LLIST	de instructie LIST, aangepast aan de regeldrukker.
BREAK	stop tijdens de uitvoering van een van de drie hiervoor genoemde instructies.

4. De nabije toekomst

4.1 Algemeen

In het kader van deze analyse van het Spectrum-systeem van Sinclair dienen ook die produkten te worden genoemd die reeds door de constructeur werden aangekondigd en die binnenkort leverbaar zullen zijn. Hoewel het onmogelijk is om de karakteristieke eigenschappen van deze randapparaten tot in detail aan te geven, is het niettemin mogelijk het technische en praktische milieu van deze aanvullende produkten te beschrijven. De beide aangeboden randapparaten zijn: een module voor aanpassing aan de telecommunicatie-standaard RS232C en een microdrive.

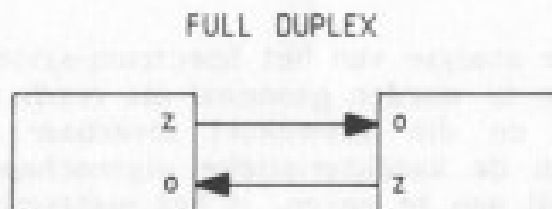
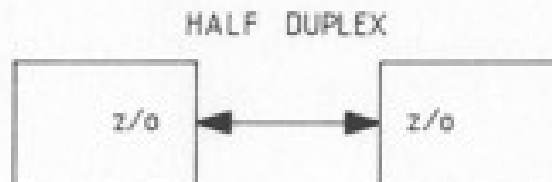
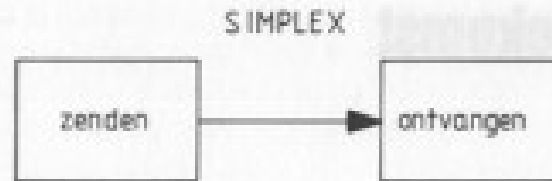
4.2 Telecommunicatie-adapter

In het volgende worden de gebruikte standaards in grote lijnen geschetst. Daarbij blijven de praktische aspecten van die gegevens buiten beschouwing en wordt de essentie ervan slechts oppervlakkig aangegeven.

Bij elke vorm van communicatie is er sprake van een bron, een drager en een ontvanger van een bericht. Om doeltreffend te kunnen zijn moet de informatie volledig zijn. Dit houdt in dat de inhoud ervan geen fouten mag bevatten en dat 'bron' en 'ontvanger' in dezelfde taal met elkaar converseren.

De ZX80 en de ZX81 beschikken door de voor hen specifieke codering over geen enkele mogelijkheid om gegevens met andere systemen uit te wisselen.

Bij de Spectrum daarentegen heeft men zich gehouden aan de internationale ASCII-code (American Standard Code for Information Interchange = Amerikaanse standaardcode voor informatie-uitwisseling). De door de meerderheid van de constructeurs van informatica-apparatuur internationaal erkende en gebruikte ASCII-code maakt deze informatie-uitwisseling wel mogelijk. In deze code zijn met 7 bits 128 codes vastgelegd waarvan er 96 afdrukbare en 32 stuurcodes zijn. De 96 afdrukbare codes bestaan uit de 26 letters van het alfabet, hoofdletters zowel als kleine letters, de cijfers en 34 speciale tekens. De gevolgde standaardisatie is vastgelegd volgens de standaard RS232C van de EIA-norm. De Europese equivalent is de CCITT-norm V24. Deze standaard bepaalt de datatransmissietechniek. Hiermee zijn er drie verbindingsvormen mogelijk.



SIMPLEX verbindingwijze in één specifieke richting.

HALF DUPLEX tweezijdige verbinding; elk op zijn beurt; één paar draden.

FULL DUPLEX tweezijdige verbinding; gelijktijdig uitzenden en ontvangen; twee paar draden.

De transmissiesnelheid is gestandaardiseerd. Deze wordt uitgedrukt in een aantal karakters per seconde (CPS) dan wel in BAUD (aantal eenheidsimpulsen per seconde). Het formaat van de gegevens is een seriële bit-voor-bit transmissie over een tweedraadslijn. Vindt de transmissie asynchroon plaats, (er is geen dirigent die het tempo bepaalt), dan bepaalt de zender het tempo. Asynchroon bedrijf is vergelijkbaar met een typiste die aan het werk is, de werksnelheid is niet constant. Om het de ontvanger mogelijk te maken het begin en het eind van elke transmissie te kunnen vaststellen wordt ook een begin- en een eindcode meegegeven.

Teneinde elk uitgezonden karakter te kunnen detecteren en eventueel te corrigeren voert het bericht een test- of pariteitsbit mee. De pariteit kan nul, transparant, even of oneven zijn.

nul	de pariteit is niet geteld.
transparant	pariteit bestaat, maar men houdt er geen rekening mee.
even	men verifieert of het aantal 1-bits even is.
oneven	men verifieert of het aantal 1-bits oneven is.

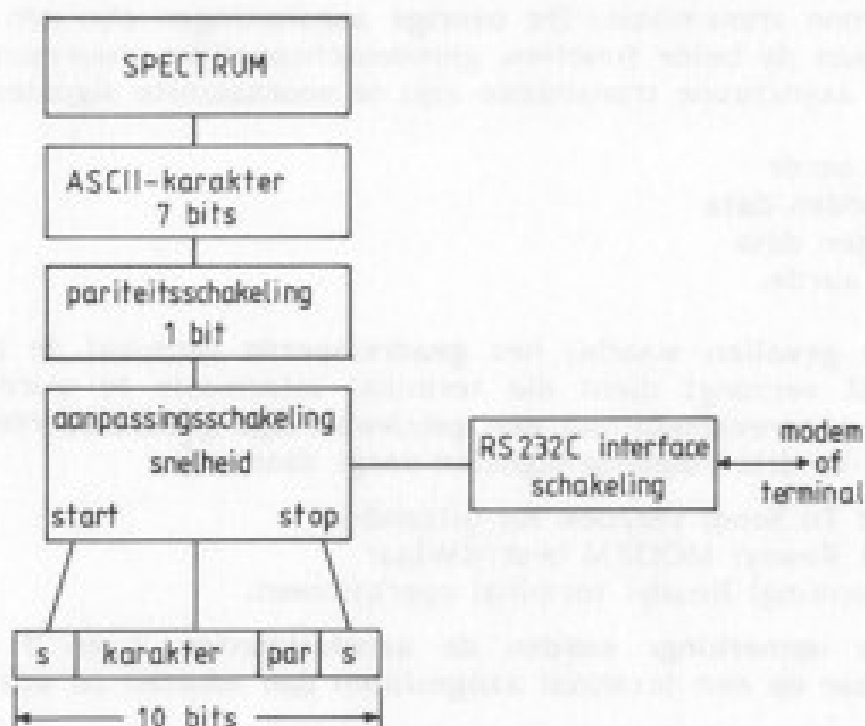
In de beide laatste gevallen kan aanvullende informatie nodig zijn. Het genereren van de pariteit is voorbehouden aan een elektronische schakeling.

Deze analyseert elk in ASCII-code uit te zenden karakter. Het resultaat van deze analyse van de zeven bits is een achtste bit die in het geval van even pariteit de 1-bits telt en er eventueel het zijne aan toevoegt om het totaal even te maken. In het geval van oneven pariteit moet het totaal aantal 1-bits oneven zijn.

Deze informatie, in binaire vorm, moet nog een laatste aanpassing ondergaan. De transmissie vindt niet in logische vorm (met twee spanningsniveaus) plaats, maar in analoge vorm.

De transmissie komt tot stand door een apparaat dat MODEM heet. MODEM is een samentrekking van de afkortingen van MODulator en DEModulator. Dit apparaat is in feite zender en ontvanger. MODEM's staan overigens onder direct toezicht van de directie van de PTT en zijn altijd onderhevig aan de goedkeuring daarvan, omdat in het geval van een telefonische verbinding gebruik wordt gemaakt van huurlijnen.

Een minder ingrijpend alternatief, maar even goed aan regels gebonden, is het gebruik van een akoestisch MODEM. Met dit apparaat wordt de interface niet op de telefoonlijn aangesloten maar wordt de telefoonhoorn gebruikt en binaire informatie in zodanige akoestische vorm uitgezonden dat die door de microfoon van het telefoonapparaat kan worden verwerkt.



Als men nu de verschillende elementen op een rijtje zet dan vindt men: De Spectrum die volgens ASCII-code zeven bits pariteit levert. De pariteitsschakeling die er één bit aan toevoegt. De aanpassingsschakeling die het begin en het eind aangeeft door er een START-bit en een STOP-bit aan toe te voegen. Dit samenstel wordt in serie en met de juiste, door de analoge interface-schakeling gekozen snelheid uitgezonden naar een MODEM of naar een met de EIA-standaard compatibele terminal.

De RS232C-interface wordt niet altijd op de telefoonlijn aangesloten. Voor zover de talrijke computerterminals aan dezelfde standaard voldoen is het gebruik van snelle printer-terminals onder gebruikmaking van kwaliteits- en/of normaal papier, voldoende motief om dit te rechtvaardigen.

Immers door dit medium heeft tekstverwerking zijn ware betekenis gekregen. Opzetten, opmaken en wijzigen gebeuren op het beeldscherm. Het afdrukken gebeurt op afdrukapparatuur, aangepast aan de kwaliteitsbehoefte en de door de toepassing vereiste snelheid.

Na de telecommunicatie-interface te hebben behandeld beginnen we met de betreffende hardware- en software-aspecten.

4.3 Het hardware-aspect van de RS232C-standaard

Zoals bij de meeste systemen komt de telecommunicatieverbinding volgens standaard RS232C tot stand met een connector met trapeziumvormige doorsnede en 25 mannelijke of vrouwelijke contacten. Het merendeel van de daarop aanwezige signalen is bestemd voor de besturing van de signaaluitwisseling vanuit het MODEM en voor het bewaken van fluctuaties in het elektrische signaalniveau op de telefoonlijnen. Hoewel het weinig waarschijnlijk is dat u in dit niveau zult kunnen ingrijpen, volgt hieronder wel een overzicht van de verschillende aanwezige signalen. Uit een nader onderzoek blijkt dat er twee kanalen zijn. De eerste serie (pen 1 tot 8) is bestemd voor asynchrone transmissie. De tweede serie (pen 11 tot 19) voor synchrone transmissie. De overige aansluitingen zijn een aantal, voor de sturing van de beide functies, gemeenschappelijke stuurspanningen. Voor wat betreft asynchrone transmissie zijn de voornaamste signalen:

1. Chassis aarde
2. Uitgezonden data
3. Ontvangen data
7. Signaal aarde.

In bepaalde gevallen waarbij het geadresseerde terminal de besturing van het MODEM verzorgt dient die terminal informatie te worden verschaft omtrent de aanwezigheid van een gelijkwaardige geadresseerde. De samenhang tussen de drie volgende signalen zorgt daarvoor:

4. Request To Send: verzoek tot uitzenden
6. Data Set Ready: MODEM bedrijfsklaar
20. Data Terminal Ready: terminal operationeel.

Een laatste opmerking: worden de aansluitpunten 2 en 3 niet op een MODEM maar op een terminal aangesloten dan moeten ze worden gekruist, voorbeeld:

Spectrum (2)		terminal (3) en Spectrum (3)		terminal (2)
PIN	BESCHRIJVING			TOEPASSING
1 FG	frame ground	chassis aarde		aansluiting
2 TD	transmitted data	uitgezonden data		uitgang
3 RD	received data	ontvangen data		ingang
4 RTS	request to send	verzoek tot uitzenden		uitgang
5 CTS	clear to send	toestemming tot zenden		ingang

PIN		BESCHRIJVING		TOEPASSING
6	DSR	data set ready	MODEM bedrijfs- klaar	ingang
7	SG	signal ground	signaal aarde	aansluiting
8	DCD	data carrier detect	draaggolf gedetec- teerd	ingang
9	+	tension test (+)		
10	-	tension test (-)		
11	QM	equalizer mode		
12	(S)CTS	secondary CTS		
13	(S)CTS	secondary CTS		
14	(S)TD	secondary TD		
15	TC	transmitter clock		
16	(S)RD	secondary RD		
17	RC	receiver clock		
18	DCR	divided clock RX		
19	(S)RTS	secondary RTS		
20	DTR	data terminal ready	terminal bedrijfs- klaar	uitgang
21	SQ	signal qual.		
22	RI	ring indicator	bel indicator	ingang
23		data rate select		
24	(TC)	ext tx clock		
25		busy		



Samenvattend: bij telecommunicatie volgens Standaard RS232C, met of zonder MODEM, dient te worden gelet op:

1. De transmissiesnelheid. Deze wordt uitgedrukt in BAUD (110, 150, 300, 600, 1200, 2400, 4800 ...) en moet aan weerszijden van de lijn dezelfde zijn.
2. De transmissiefunctie - Simplex, Half Duplex of Full Duplex. Beide apparaten moeten op dezelfde wijze met elkaar communiceren.
3. Op het uitgezonden signaalformaat. Hoewel het merendeel van de apparaten in feite met een startbit en een stopbit werkt, werken bepaalde apparaten nog met twee stopbits.
4. Op de pariteit - nul, transparant, even of oneven. Beide apparaten moeten om een uitzending te verifiëren dezelfde algoritme hanteren.
5. Op de elektrische verbinding. In het geval van een MODEM zal dit geen moeilijkheden opleveren. In het geval van een terminal dient ervoor te worden gezorgd dat de verbindinglijnen voor datatransmissie en -ontvangst worden gekruist.

4.4 Het software-aspect; ASCII-code

Het software-aspect hangt ten nauwste samen met het feit dat de Spectrum de ASCII-standaard volgt. Zoals hiervoor al werd aangegeven telt de ASCII-code 32 stuurcodes. Verscheidene van die stuursignalen hebben

betrekking op de terminal of op de telecommunicatielij. Hieronder volgt een overzicht van de standaard en de stuurcodes, en de afwijkingen daarvan.

CODE	SPECTRUM	ASCII	BETEKENIS	TOEPASSING
0	-	NUL	null	
1	-	SOH	start of heading	
2	-	STX	start of text	
3	-	ETX	end of text	
4	-	EOT	end of transmission	
5	-	ENQ	enquiry	
6	PRINT command	ACK	acknowledge	
7	EDIT	BEL	bell	bel
8	cursor naar links	BS	back space	spatie naar links
9	cursor naar rechts	HT	horizontal tab	tabulator horizontaal
10	cursor omlaag	LF	line feed	regelopschuif
11	cursor omhoog	VT	vertical tab	tabulator verticaal
12	DELETE	FF	form feed	pagina-opschuif
13	ENTER	CR	carriage return	wagenterugloop
14	number	S0	shift out	kleine letters
15	-	S1	shift in	hoofdletters
16	control INK	DLE	data link escape	
17	control PAPER	DC1	device control 1	externe sturing 1
18	control FLASH	DC2	device control 2	externe sturing 2
19	control BRIGHT	DC3	device control 3	externe sturing 3
20	control INVERSE	DC4	device control 4	externe sturing 4
21	control OVER	NAK	negative acknowledge	
22	control AT	SYN	synchronous idle	
23	control TAB	ETB	end transmission block	
24	-	CAN	cancel	opheffen
25	-	EM	end of medium	
26	-	SUB	substitute	
27	-	ESC	escape	ontsnappingscode
28	-	FS	file separator	
29	-	GS	group separator	
30	-	RS	record separator	
31	-	US	unit separator	
127	©	DEL	delete	wegnemen

Voor wat betreft het karakter-repertoire volgt de Spectrum exact de ASCII-code. De Spectrum bedient zich zoals de tabel laat zien, van bepaalde stuurcodes. Voor die codes is een eigen functie gedefinieerd, waardoor de compatibiliteit met externe apparatuur verloren gaat en de werking ervan niet kan worden gereproduceerd. Deze laatste opmerking is overigens niet beperkend. In feite is de sturing van een extern aangesloten printer heel goed mogelijk met behulp van de instructie CHR\$. Het is zaak hiermee rekening te houden.

4.5 Microdrives

Het tweede door Sinclair aangekondigde apparaat, en verreweg van de

grootste betekenis, is een microdrive met de volgende karakteristieke eigenschappen:

capaciteit 100 Kbyte,
overdrachtssnelheid 16 Kbyte/s,
gemiddelde toegangstijd 3,5 seconde,
mogelijkheid om 8 microdrives aan te sluiten.

Hoewel tot op de dag van vandaag geen verdere mededelingen zijn gedaan, is het door vergelijking met andere systemen heel goed mogelijk de prestaties van de software te extrapoleren.

Voor zover de microdrive willekeurig toegankelijke magnetische bestanden verwerkt, beschikt het over een instructie-repertoire dat door de cassette-recorder in combinatie met het geheugen wordt gebruikt.

Uitgaande van de capaciteit van één eenheid, in vergelijking met de maximale omvang van het geheugen, is het al zeker dat per station meer dan één bestand zal kunnen worden opgeborgen. Redenen genoeg om een CATalogus op te stellen.

Omdat het aantal eenheden tot acht kan worden uitgebreid, dient de herkomst van de informatie vastgesteld te kunnen worden. Dit is de bestaansreden van instructies voor het openen en het sluiten van de toegang tot de geheugens: OPEN # (openen nr.) en CLOSE # (sluiten nr.).

Gaat men een microdrive-medium voor de eerste maal gebruiken dan moeten bepaalde initialiseringstaken worden uitgevoerd. Initialiseren betekent geheugenruimte toekennen, verifiëren van de toegankelijkheid, kwaliteitscontrole maar ook reserveren van geheugenruimte om er lijsten van bestandsnamen en hun lengte op vast te leggen. Deze bewerking wordt uitgevoerd met de instructie FORMAT.

Het bewerken van bestanden maakt ook het bijwerken daarvan noodzakelijk, zoals verplaatsen, sorteren en wissen van bij te werken bestanden. Deze handelingen zijn mogelijk dankzij de instructies MOVE (verplaatsen), ERASE (wissen), LIST (uitlijsten), SAVE (wegschrijven), VERIFY (verifiëren), LOAD (laden) en MERGE (samenvoegen).

Hoewel deze commentaren hypothetisch zijn, zijn de veranderingen in prestaties wel waarschijnlijk. Rest ons slechts geduld te oefenen en bij voorbaat onze bestanden zo op te zetten dat de maximale prestaties van die toekomstige microdrives snel te benutten zullen zijn. Dit alles in de hoop binnenkort nadere informatie te zullen ontvangen...

Het beschikbaar komen van precieze karakteristieke eigenschappen zal het mogelijk maken om het beheer van databestanden op microdrives tot in detail uit te werken, te beginnen met de in het volgende hoofdstuk opgesomde toepassingen.

5. Wenken voor het gebruik

5.1 Sinclair-computers

5.1.1 ZX81

Snijdt men het onderwerp informatica aan dan is er altijd wel iemand die iets meer van de praktische mogelijkheden van de computer wil weten dan alleen de zuiver ludieke aspecten ervan.

En dat is nu juist wat de ZX81 heeft: praktische toepassingsmogelijkheden. Wel dient men er zich van bewust te zijn dat de beperkingen opgelegd door de omvang van de geheugens, de snelheid van de computer, de omvang van het toetsenbord, het gebruik van sequentiële bestanden op een trage cassetterecorder (150 baud), en het isolement afgedwongen door de onmogelijkheid van de standaard tot communicatie, hinderpalen zijn die alleen door de prijs en een hoge mate van motivatie kunnen worden weggenomen. Men moet in elk geval toegeven dat de ZX81 op dit moment één van de meest aantrekkelijke informatica-hulpmiddelen is. De lage kosten, het grote gebruikscomfort, maar bovenal de grote verbreiding ervan, hebben het mogelijk gemaakt om amateur en leek elkaar te laten treffen; elkaar te helpen en het terrein van de informatica te ontdekken. De ontwikkelingsinspanningen van meer dan 600 000 gemotiveerde en actieve gebruikers laten zich moeilijk ontkennen.

5.1.2 ZX Spectrum

Met de komst van de Spectrum en de bijbehorende randapparatuur is het merendeel van de bezwaren tegen de ZX81 komen te vervallen. Het geheugen heeft een respectabele omvang, de computer is krachtig, het toetsenbord is goed te bedienen, de microdrives maken vrij toegankelijke bestanden mogelijk en het systeem volgt de internationale standaardisatie. Maar, wees niet overdreven optimistisch!

Met de Spectrum zijn niet alle problemen van de informatica opgelost. Maar hij is in elk geval in staat zich tot de problemen van een unieke gebruiker te richten die er reeksen van onderwerpen mee kan bewerken en die wat geduld heeft om zijn resultaten te bereiken. Aan de hand van deze commentaren kan de lezer nu een exacte conclusie trekken: de toepassing van een informaticasysteem moet, om echt opgenomen te worden en de beoogde diensten op doelmatige wijze te verrichten, zorgvul-

dig worden onderzocht. Teneinde de doelmatigheid te bepalen moeten bovenal eisen worden gekwantificeerd. Doet men dit niet dan is een vergelijking van enige betekenis onmogelijk.

Omdat de Spectrum vele voordelen biedt die soortgelijke, vaak duurdere apparaten niet bieden, is het interessant om mogelijke applicaties tegen elkaar af te wegen. De computer is een hulpmiddel dat uitmunt in het vergelijken en klassificeren. De toepassing ervan dient men dan ook te zoeken in kleine ondernemingen teneinde de bedrijfsleiding te ontlasten. Alvorens een conclusie te kunnen trekken dient ook de omgeving te worden onderzocht. We willen ons hier niet in de plaats stellen van een hoogleraar in de economie of in die van een specialist op dit gebied, maar behandelen hier een ongepubliceerde studie over het onderwerp 'de commerciële onderneming'. Het doel van deze bewust algemeen gehouden studie is globaal de toepassingskaders van de Spectrum vast te stellen, maar bovenal twee onderwerpen op te voeren waar we wat nader op zouden willen ingaan. De studie van de opbouw van rationele bestanden of databanken (van het formaat van de Spectrum) en een voor de verwerking van bestanden belangrijk onderdeel: sorteertechnieken.

5.2 Het concept van een 'commerciële onderneming'

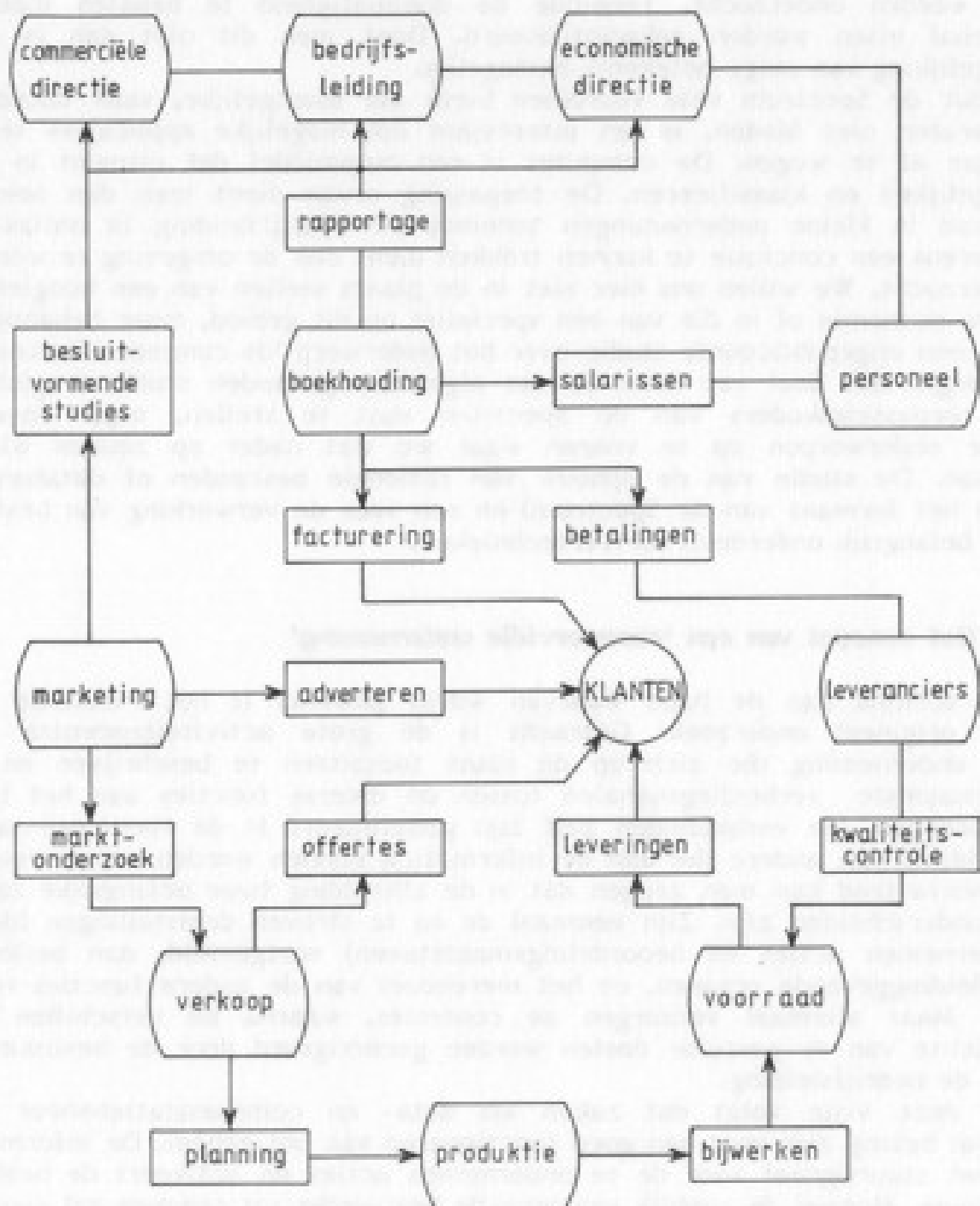
Het schema aan de hand waarvan wordt gewerkt is het resultaat van een origineel onderzoek. Getracht is de grote activiteitencentra van die onderneming die zich op de klant toespitsen te beschrijven en de voornaamste verbindingskanalen tussen de diverse functies aan het licht te brengen. De verbindingen zelf zijn gedefinieerd in de vorm van taken temidden van andere die aan de informatica kunnen worden toevertrouwd. Samenvattend kan men zeggen dat in de afbeelding twee belangrijke zaken te onderscheiden zijn. Zijn eenmaal de na te streven doelstellingen (de te ondernemen acties en beoordelingsmaatstaven) vastgesteld, dan beslissen de leidinggevende organen, en het merendeel van de andere functies voert uit. Maar allemaal verzorgen ze controles, waarna de verschillen ten opzichte van de gestelde doelen worden gecorrigeerd door de beslissingen van de bedrijfsleiding.

Uit deze visie volgt dat zaken als data- en communicatiebeheer van vitaal belang zijn voor een goed functioneren van het geheel. De informatie is het stuursignaal voor de te ondernemen acties en activeert de besluitvorming. Hoewel de aanblik van vuur de bestuurder tot remmen zal aanzetten, zal hij als men hem het uitzicht belemmert zeker te laat remmen.

In een commerciële onderneming draait alle informatie om de klant, hij is het doel. Verkoop, produktie en dienstverlening documenteren allemaal hun activiteiten en bezigheden. Elke klant heeft een eigen archiefkaart en groeit de onderneming voorspoedig, dan centraliseert men de archiefkaarten tot een bestand. In de informatica heet dat een databank.

Hoewel elke afdeling waarschijnlijk aan de naam van de klant moet refereren is de te bewerken informatie specifiek. Het belang van de ene klant is niet hetzelfde als dat van een andere, maar wel treft men vaak raakpunten. Vult het ene bestand het andere aan dan levert het geheel nuttige informatie en zal dus gesorteerd moeten worden om er de essentiële punten aan te kunnen onttrekken.

Het nemen van beslissingen heeft niet altijd betrekking op een specifiek



geval. De verkoopstrategie van de onderneming en de algemene leiding doen een beroep op algemene onderzoeken die op het klantenbestand zijn gebaseerd of uit algemene informatie zijn afgeleid. Wiskunde en statistiek zijn de hulpmiddelen voor het opstellen van de algoritme voor de besluitvorming.

Hieruit blijkt wel het belang van bestandsstructuren en de bestaansredenen van sorteermethoden.

5.3 Soorten bestanden

Te oordelen naar zijn prestaties is de Spectrum aangepast aan het beheren van individuele bestanden zoals die worden toegepast in de algemene dage-

lijkse praktijk en doorgetrokken zouden kunnen worden tot binnen het kader van de kleine onderneming. De krachtige instructies maken het mogelijk om, na een zorgvuldige analyse van de bedoelde applicatie, tot een aangepaste snelle oplossing voor toegang en dimensie te komen. Wij adviseren de gebruiker dan ook zijn eigen bestanden op te bouwen teneinde aan zijn specifieke behoeften tegemoet te kunnen komen zonder van algemene oplossingen afhankelijk te hoeven zijn.

Wil een bestand werkbaar zijn, dan moet het mogelijk zijn om de samenstellende elementen ervan te kunnen invoeren, opzoeken, verifiëren, corrigeren en wegnemen. Het nu volgende verklaart het verloop van de gedachtengang en verduidelijkt de voordelen en beperkingen van verschillende oplossingen.

5.3.1 Bestandstabellen

Met behulp van de instructie DIM is het mogelijk om geheugenruimte te reserveren en er zowel numerieke als alfabetische gegevens in op te bergen. Een ander voordeel is dat de informatie gemakkelijk toegankelijk is en wederzijdse betrekkingen tussen de tabellen gemakkelijk vast te stellen zijn. Zowel toegang als correlatie is een kwestie van indiceren.

Een bezwaar is dat de dimensie van alle weggeschreven informatie vast ligt. Dit betekent dat, wanneer de grootste variabele als maatstaf wordt gehanteerd, een verkwisting van geheugenruimte voor de talloze variabelen met kleinere afmetingen ontstaat en dat wanneer de kleinste variabele als maatstaf genomen zou worden er een systematisch afkappen met verlies van informatie optreedt.

Deze oplossing is bij uitstek geschikt voor variabelen met vaste lengte.

5.3.2 Bestandsreeksen

De werkelijke kracht van de instructie voor het afkappen schuilt in de gemakkelijke keuze van een datablok uit een reeks van achtereenvolgens weggeschreven datablokken. Een toegangsmethode tot de informatie zou kunnen zijn dat aan het begin van elk datablok de lengte ervan wordt vermeld. Bedraagt de lengte van 1 tot 255 dan zal er niet meer dan één byte in beslag worden genomen. Het zoekprogramma springt dan van blok naar blok.



Een andere methode zou kunnen bestaan uit het opstellen van een tabel waarin op numerieke volgorde alle wijzers, de lengte van het datablok en het adres van het eerste karakter van de variabele wordt vastgelegd. Een dergelijk bestand heet een geïndiceerd bestand.

0	4	10	10	12	S	L	V
					1	4	0
					2	6	4
					3	2	10
S = serienummer L = lengte V = verplaatsing							

Met de bestanden waarvan hier sprake is wordt natuurlijk in het geheugen gemanipuleerd en het gebruik van een cassette recorder is daarbij te omslachtig en de toegangssnelheid te pover.

Dat is niet het geval met een microdrive. Capaciteit en toegangstijd laten het toe ze als werkbare uitbreidingen van het geheugen te beschouwen. De toegangstijd ligt in de orde van grootte van 3,5 seconden en de overdrachtssnelheid van 16 Kbyte per seconde; voor het vernieuwen van een sector van 16 Kbyte rekent men 5 seconden, wat heel acceptabel is.

5.4 Sorteermethoden

Meer dan op enig ander gebied is voor een efficiënte dataverwerking tevoren een zorgvuldige analyse vereist. Voor de problemen bestaat er geen wondermiddeltje. De computer maakt als prestatieversterker geen enkel verschil tussen toenemen en afnemen. Beiden worden met eenzelfde factor versterkt. Een onjuiste analyse bewerkstelligt een slechte keus die op een fiasco voor het verwachte resultaat kan uitdraaien.

In deze uiteenzetting is er sprake van selectietechnieken en wel in het bijzonder voor in het geheugen opgeslagen informatie. De juiste methoden om door externe media afgegeven informatie te selecteren vallen buiten het kader van deze beknopte beschrijving en zouden weinig nut hebben omdat de besturing van de randapparatuur van de Spectrum blokken gegevens overbrengt en geen afzonderlijke elementen. Overdracht van een binair bestand met unieke elementen is uitgesloten gezien de overbelasting van de besturing die een dergelijke praktijk met zich mee zou brengen.

De kwaliteit van een sorteermethode hangt af van het soort bestand: sequentieel of geïndiceerd. Van zijn inhoud: letters of cijfers. Van het aantal elementen waaruit het is samengesteld, maar bovenal van de toestand van de rangschikking. De keus van een methode hangt af van het relatieve belang dat aan de specifieke prestaties van de een of andere methode wordt toegekend, en waarbij rekening wordt gehouden met de hiervoor genoemde parameters.

5.4.1 Sorteren volgens de 'bubble-up'-methode

Bij het sorteren volgens de 'bubble-up'-methode worden twee opeenvolgende elementen met elkaar vergeleken en in het geval van een verkeerde rangorde de plaats van het ene element met die van het andere verwisseld. Hierbij dient men vooral het domme karakter van de computer in het oog te houden. De door de mens intuïtief gedane simpele vaststelling dat het eind van een bestand is bereikt vergt van de computer dat die een specifieke vraag stelt. De algoritme ziet er als volgt uit:

Rij = aantal elementen (6): Begin = 0

Als Rij <= Begin - Eind van de rangschikking

Voor begin + 1 (1) tot en met Rij (6) - rangschikking

 Voor Rij (6) tot Begin + 1 (1) tel elementen Rij (Q) - 1

 Als element (Q) >= element (Q-1) - VOLGENDE TELLING

 Zet element Q in wachtrij

 Vervang element (Q) door element (Q-1) } VERWISSELEN

 Vervang element (Q-1) door wachtrij

 Tellen van volgende element Q

Volgende rangschikking

ele- menten	begin	T=0 ①	T=1 ②	T=2 ③	T=3 ④	T=4 ⑤	T=5 ⑥
1		20	20	0	0	0	0
2		30	0	20	10	10	10
3		0	30	30	10	20	20
4		40	10	10	30	30	30
5		10	40	40	40	40	40
6		50	50	50	50	50	50

Het voornaamste bezwaar van deze algoritme is dat er niet de werkelijke toestand van de rangschikking mee wordt vastgesteld. Uit het voorbeeld blijkt immers dat uitgaande van doorgang no. 3 het bestand al gesorteerd is. Een eerste wijziging zou kunnen zijn de waarde van een wijzer op het niveau van de laatste sortering bij te houden. De algoritme ziet er als volgt uit:

```

Rij = aantal elementen : Begin = 0
Als Rij (Q) = Begin (T) Einde rangschikking
Wijzer = Rij
  Voor element Rij tot Element Begin + 1 Element Rij - 1 (Q)
    Als E (Q) >= E (Q-1) Volgende telling
    VERWISSELEN
    Wijzer (I) = Element Rij (Q)
  Volgende telling
Begin = Wijzer

```

Het voorbeeld laat zien dat er als het bestand al gerangschikt is, er geen wijziging in de wijzer (Lus (1)) optreedt, zodat begin- en rij-waarde beiden overeenkomen met de variabele Wijzer.

ele- ment	T=0		Q=6		T=1		Q=6		T=2		Q=6		T=6
1	20	I=1	0		0				0				Q=6
2	30	I=2	0	20	20	I=2	10		10				
3	0		30		30	I=3	10	20	20				
4	40	I=4	10		10		30		30				
5	10		40		40				40				
6	50				50				50				
	I=T=1				I=T=2				I=T=6				

5.4.2 Sorteren door tussenvoegen/verschuiven

Bij deze methode kent men aan het eerste element een oneindig kleine waarde toe, vergelijkt die met elke volgende waarde en voegt de waarde er tussen door een andere te verschuiven.

A (0)	-	T	T	T	T	T	T	T	T	T	T	T
A (1)	20	20	20	20	20	20	0	0	0	0	0	0
A (2)	30	30	30	30	30	20	20	20	20	20	20	10
A (3)	0	0	0	0	30	30	30	30	30	30	20	20
A (4)	40	40	40	40	40	40	40	40	40	30	30	30
A (5)	10	10	10	10	10	10	10	10	40	40	40	40
A (6)	50	50	50	50	50	50	50	50	50	50	50	50
Buffer (T)-IE30		20	30	0	0	0	40	10	10	10	10	10
Doorgang 1		2	3	4	5	6	7	8	9	10	11	12

5.4.3 Binair sorteren

Kenmerkend voor het binair sorteren is dat het bestand in twee gelijke delen wordt gesplitst en dat de waarde van het in te voegen element wordt vergeleken met de waarde van de beide variabelen links en rechts ervan teneinde de plaats van het in te voegen element te bepalen. Het sorteren wordt beëindigd als de waarde links + 1 gelijk is aan de waarde rechts.

element	A(1)	A(2)	A(3)	A(4)	A(5)	A(6)	buffer
waarde	10	20	40	50	60	70	30
links	-1	-1	0				
rechts	5	2	2	2			
midden	2	0	1				

Het bezwaar van deze programma's is dat de uitvoeringstijd ervan toeneemt met het aantal elementen en wel evenredig met het kwadraat van het aantal.

Sorteren door tussenvoegen, sorteren door verwisselen en daarvan afgeleide methoden hebben gemeenschappelijk dat naburige elementen verplaatst worden. Gezien het feit dat het aantal verplaatsingen toeneemt met de omvang van het bestand zou men kunnen overwegen de snelheid te vergroten door alleen de voornaamste verplaatsingen uit te voeren.

5.4.4 Methode Shell

Evenals bij binair sorteren kiest men om te beginnen een punt halverwege de lengte van de tabel en vergelijkt de elementen aan weerszijden van dat punt. Klopt de volgorde niet, dan verwisselt men ze. Zo werkt men element voor element vanuit het midden tot de aanvankelijk grote sprongen tot één zijn afgenomen.

De algoritme ziet er als volgt uit:

Voor het midden = INT (aantal/2) tot 0

kiezer = 1

Voor teller van 1 tot aantal elementen - midden


```

sorteren = teller + midden
Als element (teller) > element (sorteer) dan INVERSIE
kiezer = 0
Volgende teller
Als kiezer = 1 dan midden = INT (midden/2)
Volgende midden

```

Uit een meer gedetailleerde analyse en een oordeelkundige vergelijking blijkt dat het sorteren met tussenniveaus sterk lijkt op het sorteren door verwisselen en niet erg doelmatig is.

5.4.5 Methode Shell-Metzner

De methode Shell-Metzner is bedoeld om het sorteren door verwisselen van de tussenniveaus te vervangen door sorteren door tussenvoegen, een duidelijk elegantere maar vooral krachtiger methode.

De algoritme ziet er als volgt uit:

```

FOR midden = INT (aantal/2) TO 0
  FOR I = midden + 1 TO aantal
    X = element (I)
    FOR J = I - midden TO I stap - midden: REM INSERTION
      IF X < element (J) THEN element (J + midden) = X : NEXT I
      element (J + midden) = element (J)
    NEXT J
    element (J + midden) = X
  NEXT I
  midden = INT (midden/2): REM INCREMENT
NEXT midden

```

Uit de voortgang in de zich steeds verder ontwikkelende informatietheorie (onder andere dankzij de duizenden Sinclair-gebruikers) volgt dat de verandering in de loop van de procedure, toenemen door er opeenvolgend lagere waarden aan toe te kennen, de uitvoeringstijd met 5 tot 20% verbetert. Zolang het speurwerk wordt voortgezet wachten we geduldig de ontdekking van een wondersorteermethode af!

We besluiten deze introductie tot sorteermethoden in de hoop duidelijk te hebben gemaakt dat de beste methode altijd die is die, na serieus de vraag beoordeeld te hebben waarop men een antwoord wenst, het verdienstelijkst blijkt.

Uit de praktijk blijkt dat als functie van de omvang van een tabel de prestaties dezelfde orde van grootte aannemen als de prestatie-coëfficiënten:

sorteermethode	aantal elementen			
	50	100	200	500
tussenvoegen	20	70	250	
selecteren	25	75	275	
verwisselen	50	200	750	
Shell	30	90	225	
Shell-Metzner	15	30	90	250

Het volgende programma illustreert deze sorteertechnieken. Het bestaat uit vier onderdelen, een real-time klok om de uitvoeringstijd te bepalen, een generator voor willekeurige string-variabelen waarvan de lengte van 2 tot 10 willekeurige karakters beslaat, een sorteertabel volgens de methode met verwisselen gevolgd door een sortering van dezelfde tabel volgens de methode Shell-Metzner.

De afgebeelde resultaten geven voor de Spectrum en voor een gegenereerd bestand (de waarden daarvan veranderen bij elke nieuwe programma-run) de produktietijd van de 100 ongeordende variabelen, het aantal verwisselingen, vergelijkingen en de duur van het rangschikken.

Matrix A\$ bevat het ongeordende bestand, en matrix B\$ hetzelfde maar nu gerangschikte bestand.

Het blijkt dat de methode Shell-Metzner vijf maal sneller is.

Men verkrijgt een beter leesbare indruk van de duur door regel 10 te wijzigen in:

```
10 DEF FN t()=INT(65536*PEEK ...)/50
```

```

1 REM *****
2 REM * test de verwerkings-*
3 REM * tijden van verschil-*
4 REM * lende sorteermetho-*
5 REM * den. Er worden 100 *
6 REM * random strings gege-*
7 REM * nereerd, gesorteerd *
8 REM * en eventueel geprint*
9 REM *****
10 DEF FN t()=(65536*PEEK 2367
4+256*PEEK 23673+PEEK 23672)/50
20 LET z1=FN t()
30 DIM a$(100,15)
40 DIM b$(100,15)
50 FOR i=1 TO 100
60 LET l=INT (RND*10+2): LET t
$=""
70 FOR j=1 TO l
80 LET t$=t$+CHR$ (INT (RND*26
+65)): NEXT j
90 LET a$(i)=t$: NEXT i
100 LET z2=FN t()
110 FOR i=1 TO 100: LET b$(i)=a
$(i): NEXT i
120 PRINT "generatie duurde ";z
2-z1
130 PRINT "5', ik sorteer!!!!"
140 REM *****
150 REM * bubble up methode *
160 REM *****
170 LET n=100: LET a=0: LET b=0
180 LET c=0: LET e=0
190 REM *** start sorteren
```

```

200 LET t1=FN t()
210 LET a=a+1
220 IF a=n THEN GO TO 340
230 LET b=a+1
240 LET c=c+1
250 IF b$(a)>b$(b) THEN GO TO
290
260 LET b=b+1
270 IF b>n THEN GO TO 210
280 GO TO 240
290 LET e=e+1
300 LET t$=b$(a)
310 LET b$(a)=b$(b)
320 LET b$(b)=t$
330 GO TO 260
340 LET t2=FN t()
350 REM *****
360 REM *methode shell-metzner*
370 REM *****
380 FOR i=1 TO 100: LET b$(i)=a
$(i): NEXT i
390 LET z1=FN t()
400 LET c1=0: LET s1=0: LET a1=
1: LET n=100
410 IF 2^a1>n THEN GO TO 440
420 LET a1=a1+1
430 GO TO 410
440 LET f1=2^a1-1
450 LET f1=INT (f1/2)
460 IF f1=0 THEN GO TO 630
470 LET d1=n-f1
480 LET b1=1
490 LET a1=b1
500 LET e1=a1+f1
510 LET c1=c1+1
520 IF b$(a1)>b$(e1) THEN GO T
O 560
530 LET b1=b1+1
540 IF b1>d1 THEN GO TO 450
550 GO TO 490
560 LET s1=s1+1
570 LET t$=b$(a1)
580 LET b$(a1)=b$(e1)
590 LET b$(e1)=t$
600 LET a1=a1-f1
610 IF a1<1 THEN GO TO 530
620 GO TO 500
630 LET z2=FN t()
640 REM *****
650 REM *      resultat     *
660 REM *****
670 CLS

```

```

680 PRINT AT 0,0;"sort.";AT 0,8
;"wissel.";AT 0,17;"vergel.";AT
0,25;"tydsd."
690 PRINT AT 2,0;"bubble";AT 2,
10;e;AT 2,19;c;AT 2,26;t2-t1
700 PRINT AT 4,0;"metzner";AT 4
,10;s1;AT 4,19;c1;AT 4,27;z2-z1
710 STOP
720 REM *****
730 REM * afdrukken strings *
740 REM *****
750 CLS
760 PRINT "a$","b$"
770 PRINT
780 FOR i=1 TO 100
790 PRINT a$(i),b$(i)
800 NEXT i

```

```

sort.      wissel.  vergel.  tydsd.
bubble      2175      4950      227.04
metzner      375      809      57.96
0001

```

a\$	b\$
VYLIMZJEGIV	ABKOUYY5
DWXA	AKRIPIWNQF
LZD	ARVYNQ
BGPXLGXL	ASYTPCVOZ
KLTY	AWWKQDJRSL
EPCE	BGPXLGXL
INP	BGSSKAWO
GWAVUREW	CIJDBEP
KPLNLBW	CKQI
EM	CLRWLLAQTI
BGSSKAWO	CNEW
HLKVFDVTV	CVCOZAQIDU
UREXTPYCU	CXBVEWFS
UEQVH	DIIJBWZGDV
XJAI	DIORLROVCXQ
DIORLROVCXQ	DWXA
VLUGBBMYIWO	EM
FEDOHSEQMHO	EPCE
IUBVYRNE	EUVBOXTT
KHAQAHH	EXTSQJHW
WBROXD	FEDOHSEQMHO
CLRWLLAQTI	FHENTBFQU
FHENTBFQU	FMHJEEASGL
WE	FTORJZ

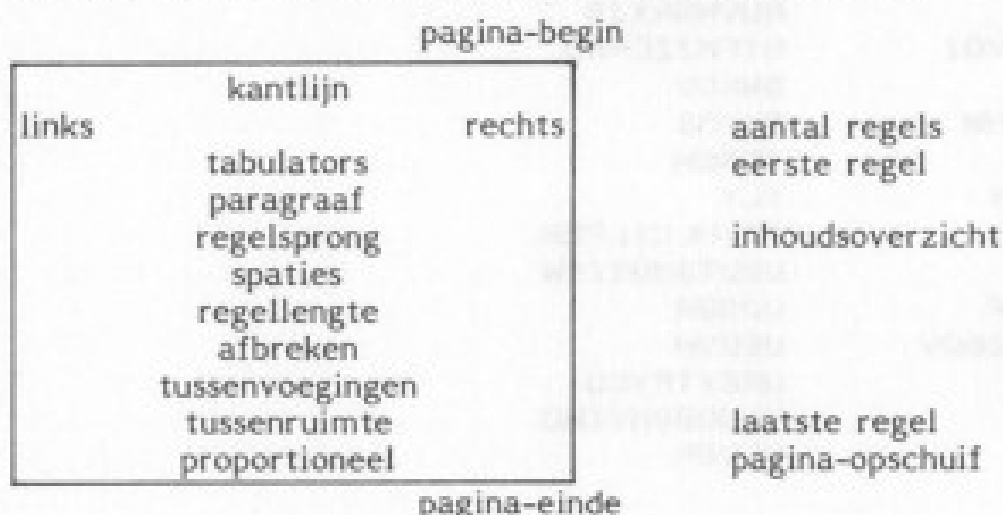
TLY	GVHDBZHKVSN
QYD	GWAVUREW
KZPBL	HHZWZU
TEASH	HLKVFDVTV
QMIWCO	HOOJHSUY
MN	ICILXPPWGB
PMMR	IEJZBG
NFY	IEVHRWYVHA
LOWRGSNN	INP
IEJZBG	IUBVYRNE
IYSFUN	IYSFUN
ICILXPPWGB	JBXBWKGTS
MHXDBDQJ	JUKWNVJBEG
MUDRRLVB	KEB
CNEW	KHADAHH
RE	KLTY
WTPVNFE	KPLNLBW
AKRIFIWNQF	KYGB
SIFMZZEPAU	KZPBL
VT	LEVRNGWNHS
JBXBWKGTS	LOWRGSNN
VVITJXH	LVIWABCA
MAUSAKLGPB	LZD
VLVSR	MHXDBDQJ
XGDSQEKHX	MN
HOOJHSUY	MUDRRLVB
SYYPJ	NCLMWF
LVIWABCA	NFY
LEVRNGWNHS	OCGUBLV
OCICD	OCICD
NCLMWF	OXTXMY
EUVBOXTT	PMMR
ZBHTVEZQ	PWTOPE
WTIKMDC	PXHM
YMKJDIMYC	QFQFNRTC
ABKOUYY5	QMIWCO
TTYHLCILPBW	QYD
RNZSAYA	RE
AMWKQJRSL	RNZSAYA
KYGB	RUVNSRXJB
ASYTPCVOZ	SIFMZZEPAU
PWTOPE	SWKCV
EXTSQJHW	SYYPJ
ARVYNQ	TEASH
ZBXTWY	TLY
FTORJZ	TTYHLCILPBW
VNDZ	UBUTJMGTYW
ZNXXGEF	UDDBR
DIIBWZGOV	UEQVH
XARDIN	UREXTPYCU
PXHM	VLQBBMYIWO
WSY	VLVSR

WRYY	VNDZ
YPJUG	VT
CVCOZADIDU	VVITJXH
IEVHRWYVHA	VYLIMZJEGIV
CXBVEWFS	WAUSAKLGPB
OCGUBLV	WBROXD
YQVNI	WE
YINQXROFC	WRYY
QFDFNRTC	WSY
CIJDBEP	WTIKMDC
CKQI	WTPVNFE
UDDBR	XARDIN
XBG	XBG
SNKCV	XGDJSQEKHX
UBUTJMBTZYW	XJAI
JUKWNVJBEG	YINQXROFC
YIXNTAV	YIXNTAV
HHZWZU	YMKJDIMYC
ZTGMKH	YPJUG
FMHJEEASGL	YQVNI
KEB	ZBHTVEZQ
RUVNSRXJB	ZBXTWY
OXTXMY	ZNXXGEF
GVHBZHKVSN	ZTGMKH

5.5 Tekstverwerking

Slaat men enkele bladzijden terug en bekijkt men het organisatieschema van het 'concept van een onderneming' dan blijkt dat de talloze verbindingslijnen gedocumenteerd kunnen worden. Hoewel de tendens naar een zuiver gecomputeerd documentenbeheer bij onze burens aan de andere kant van de Atlantische Oceaan al wel operationeel zal zijn, is het dat in onze samenleving nog niet. Het zal nog wel enkele jaren duren, of maanden (laten we maar optimistisch blijven!) voordat we het overmatig papiergebruik zullen zien verdwijnen en dat er zaken gedaan zullen worden zonder correspondentie.

De bedoeling is hier door te trachten nauwkeurig het bestek op te stellen waaraan een dergelijk applicatieprogramma zou moeten voldoen, een aantal aspecten van de tekstverwerking nader te belichten.



Natuurlijk moeten eerst de toepassingsmogelijkheden worden belicht. Bijvoorbeeld: begeleidend schrijven, afdrukken van etiketten, herinnering tot betaling, rondschrijven, opstellen van een persoonlijke brief, opstellen van verslagen.

Rangschikt men de verschillende mogelijkheden in volgorde van toenemende complexiteit, dan verkrijgt men:

Afdrukken van etiketten - hiervoor kan worden volstaan met de referentie-elementen uit het klantenbestand.

Rondschrijven - tekst zonder belangrijke wijzigingen, altijd gelijk, afhankelijk van de geadresseerde kan de aanhef verschillen (Mijnheer, Mevrouw).

Herinnering tot betaling - standaardtekst waarvoor de variabelen worden opgeroepen uit de boekhouding.

Opstellen van persoonlijke brieven - specifieke tekst waarvan de inhoud vaak aan wijzigingen onderhevig is.

Opstellen van verslagen - deze laatste soort levert natuurlijk de grootste moeilijkheden op omdat er alle hiervoor genoemde soorten in zijn verenigd en er onder andere ook grafieken in opgenomen moeten kunnen worden.

Op deze uiteenlopende aspecten van tekstverwerking is ongetwijfeld het klassieke spreekwoord 'Komt men over de hond, dan komt men ook over de staart' van toepassing, maar zeker niet met een computer waarvan de geheugencapaciteit niet het uitvoeringsprogramma en de variabelen zou kunnen bevatten.

De eerste de beste opmerking hierbij heeft betrekking op het systeem van afdrukken. In weerwil van zijn, gezien de prijs, zeer aanvaardbare prestaties kan niet met de Sinclair-printer worden volstaan omdat het formaat waarop wordt afgedrukt onvoldoende is. Men dient dus te beschikken over een printer die is ingericht voor een gestandaardiseerd papierformaat, bijvoorbeeld DIN A4 (210 x 297 mm). In Engelse maten zou dit overeenkomen met ongeveer 8 1/2".

Een tweede opmerking betreft de kwaliteit van het afgedrukte. Het merendeel van de afdrukkende terminals is uitgerust met een afdrukkop met naalden volgens het principe van de Sinclair-printer - de matrix. Is de geadresseerde niet te kritisch dan zal hij uw post wel accepteren. Niettemin zal het merendeel van hen ernstige kritiek laten horen die in feite een gebrek aan aandacht jegens de geadresseerde verbergt nu hij dan toch door een automaat bediend wordt. De oplossing daarvoor schuilt in een inslag-printer.

Printers met margrietwiel zijn populair door hun prestaties mits de prijs niet een criterium voor uitsluiting is. In vergelijking met de Spectrum is hij tien keer zo duur.

Bepaalde matrix-printers leveren een betere aanslagkwaliteit dankzij een tweede, enigszins verschoven doorgang van de afdrukkop over dezelfde regel. Hiermee kunnen zelfs grafische afbeeldingen worden gemaakt. Ten slotte zijn er voor de knutselaars nog tweedehands printers met

kogelkop te koop. Hoewel deze apparaten mechanisch kwetsbaar zijn is deze oplossing voor de economen onder ons zeker acceptabel.

Een laatste opmerking met betrekking tot de printer betreft het papier-transportstelsel met perforatie. De zelfklevende etiketten zoals die gebruikt worden voor het adresseren van enveloppen worden ook in de handel gebracht op een drager met perforatierand. Dit biedt het voordeel van een stabiel en nauwkeurig papiertransport.

Bepaalde drukkerijen kunnen zig-zag gevouwen voorgedrukt papier leveren waarvan het transport door een perforatieband wordt verzorgd.

Die transportmethode maakt verdere automatisering mogelijk zonder tussenkomst van de bediener. Voorts zijn voor bepaalde printers nog accessoires leverbaar waarmee automatisch losse bladen kunnen worden ingevoerd.

Natuurlijk biedt de moderne techniek de gebruiker tal van alternatieven. Uit deze bevindingen volgt dan ook dat alles van de toepassing afhangt en dat ook hier een nader onderzoek vruchten zal afwerpen. Doet men voor het Sinclair-systeem een beroep op een extern, afdrukkend systeem, dan zal dat in elk geval over een RS232C-interface moeten beschikken.

De besturing van de printer komt tot stand met behulp van de stuurcodes van de ASCII-code. Om uw kennis hiervan op te halen kunt u één en ander naslaan in het voorafgaande hoofdstuk, waarin de aspecten ervan meer in detail werden behandeld.

In het geval van eenvoudige en steeds weer terugkerende documenten is de beste aanpak de applicatie tot in detail te bestuderen. Een duidelijk organisatieschema zal het mogelijk maken een duidelijk en doelmatig programma op te stellen. Daartoe hoort ook het drukken van etiketten en van brochures. Men zou zelfs het drukken van etiketten als een subprogramma van een omvangrijker tekstverwerkingsprogramma kunnen beschouwen.

Het organisatieschema stelt de uitvoering van het geheel voor door te veronderstellen dat alle activiteiten vanuit hetzelfde programma bestuurd kunnen worden.

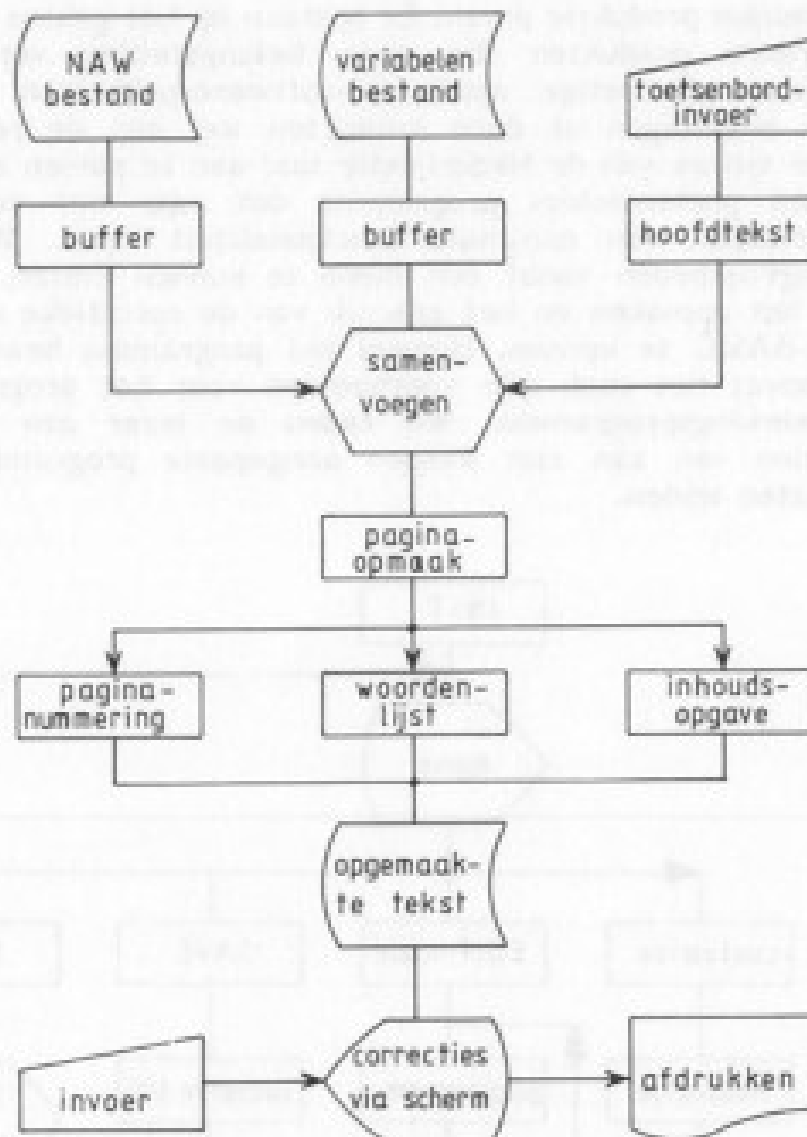
Zonder iets te willen uitsluiten kunnen we stellen dat de produktie van een boek zeker veel meer geheugenruimte vergt dan het genereren van een getypte brief.

Het sorteren van bestanden en het oproepen van in de buffergeheugens (gereserveerde geheugenruimte waarop een MERGE-commando geen invloed heeft) opgeslagen variabelen kan mogelijk voor de rest.

Het op de juiste plaats in de hoofdtekst samenvoegen, vanaf het toetsenbord of opgeroepen uit een bestand, is mogelijk door herkenning in de hoofdtekst van 'wildcards' (eigenlijk 'bladwijzers'). Deze wildcards zijn samengesteld uit combinaties van weinig gebruikte karakters, bijvoorbeeld ©, #, £, \$ enz.

Wenst men op een bepaalde plaats in de tekst een uit een hoofdbestand van variabelen opgehaald bedrag op te nemen, dan gebruikt men bijvoorbeeld de combinatie #f die op het moment van tussenvoegen de inhoud van het buffergeheugen zal oproepen. Voor zover het op papier zetten betreft, adviseren wij u het voorgaande te raadplegen, waarin het merendeel van de punten wordt opgesomd waarmee bij het verwerken van tekst rekening moet worden gehouden.

Voorts vestigen we de aandacht van de lezer op het afbreken van woorden. Deze praktijk waarbij een woord aan het eind van een lettergreep met een



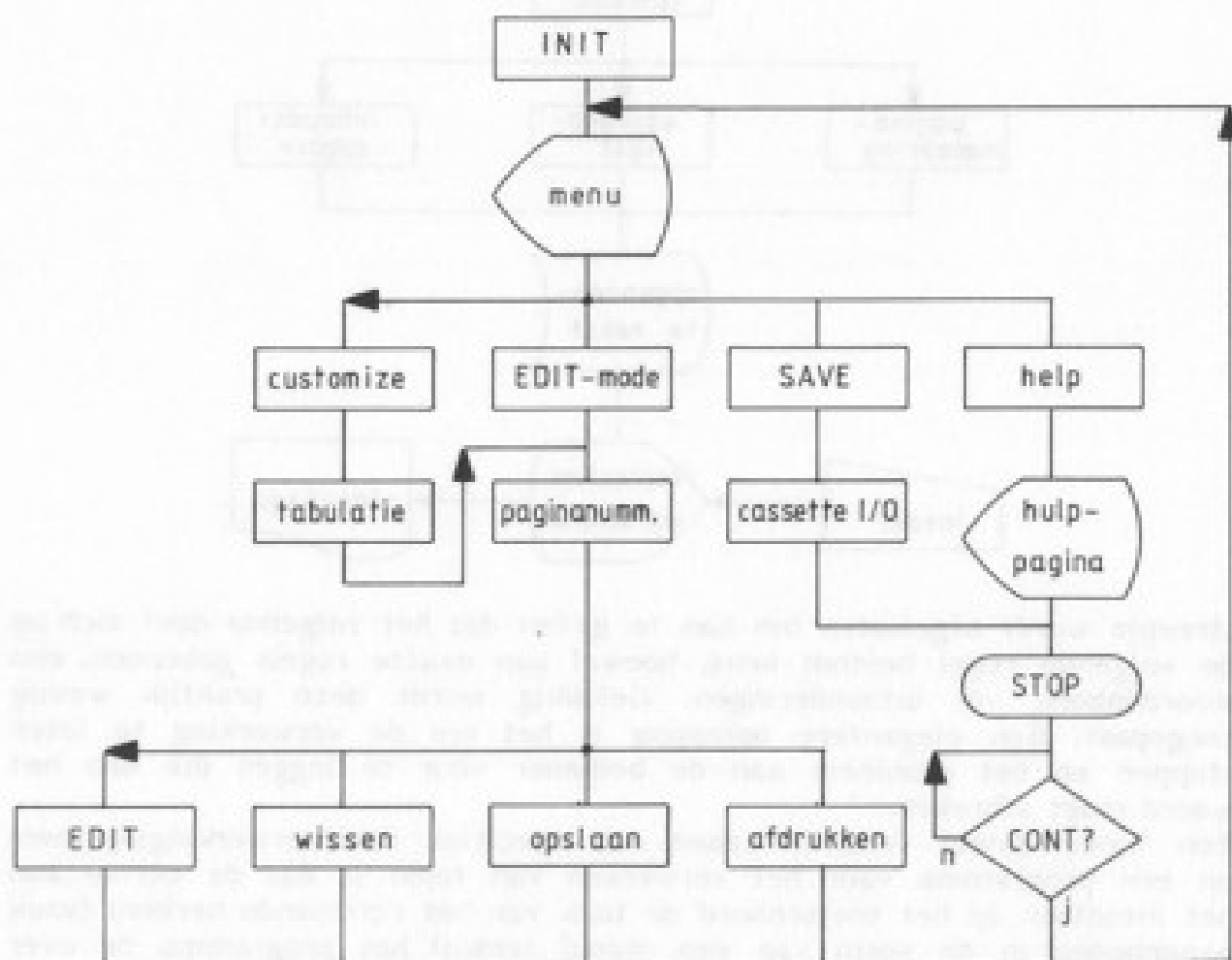
streepje wordt afgesloten om aan te geven dat het volgende deel zich op de volgende regel bevindt kent, hoewel aan exacte regels gebonden, een woordenboek vol uitzonderingen. Gelukkig wordt deze praktijk weinig toegepast. Een elegantere oplossing is het om de verwerking te laten stoppen en het probleem aan de bediener voor te leggen die dan het woord moet afbreken.

Een fundamenteel verschil tussen een specifiek tekstverwerkingssysteem en een programma voor het verwerken van tekst is dat de eerste aan het intoetsen op het toetsenbord de taak van het commando herkent (vaak aangeboden in de vorm van een menu) terwijl het programma de over het toetsenbord en in de tekst opgenomen commando's moet interpreteren. In het geval van het programma bepaalt men dat de commando's aan het begin van elke regel moeten staan. Om ze te scheiden in het geval meerdere commando's op een regel beginnen kan een herkenningskarakter worden gebruikt, bijvoorbeeld © (weinig gebruikt). Op deze wijze zou het besturingsprogramma op doelmatige wijze commando's, tekst en wildcards voor het tussenvoegen van variabelen kunnen verwerken.

Het uiteindelijke document kan zichtbaar worden gemaakt op het beeldscherm dat dan een exacte kopie van het eindresultaat laat zien, of worden afgedrukt voor beoordeling en eventuele correcties voorafgaand

aan de uiteindelijke produktie ervan. Er bestaan op het gebied van tekstverwerking meerdere produkten die onze belangstelling verdienen. Voor de uit Engeland afkomstige applicatie-software-pakketten is het goed zich ervan te overtuigen of deze produkten wel aan de regels van het schrijven en de syntax van de Nederlandse taal aan te passen zijn.

Hier volgt een pretentieloos programma dat voor het verwerken van tekst de kenmerken van minimale functionaliteit bezit. Men dient om de gebruiksmogelijkheden vanaf een menu te kunnen kiezen, enkele comando's voor het opmaken en het gebruik van de specifieke eigenschappen van Spectrum-BASIC te kennen. Hoewel het programma bewust eenvoudig gehouden is bevat het toch alle voorbeelden voor het programmeren van een tekstverwerkingsprogramma. Wij raden de lezer aan zich bij het zelf ontwikkelen van aan zijn wensen aangepaste programma's door dit voorbeeld te laten leiden.



```

1 REM *****
2 REM *   tekstverwerking   *
3 REM *****
4 REM
5 REM
10 DIM a$(5,704)
15 REM
16 REM ***** titel + menu

```

```

17 REM
19 CLS
20 PRINT AT 3,6;"**** * * *"
* * *
30 PRINT AT 4,8;" * * * * *"
*
40 PRINT AT 5,7;"** * *"
* *
50 PRINT AT 6,6;"* * *"
* *
60 PRINT AT 7,6;"**** * *"
* * *
70 PRINT AT 12,6;"1","customiz
e"
80 PRINT AT 14,6;"2","EDIT-mod
e"
90 PRINT AT 16,6;"3","SAVE"
100 PRINT AT 18,6;"4","help"
110 INPUT "selecteer functie :
";x
120 IF x<1 OR x>4 THEN GO TO 1
10
130 GO TO 2000*x
2000 REM *****
2001 REM * customize *
2002 REM *****
2003 REM
2004 REM
2010 PRINT AT 12,6; FLASH 1;"1"
2020 PAUSE 100: CLS
2030 PRINT AT 21,0;"Wilt u tabul
aties specifiseren?"
2040 INPUT "ja=1 : nee=2 ";b
2050 IF b=2 THEN GO TO 19
2060 IF b<>1 THEN GO TO 2030
2065 CLS
2070 INPUT "aantal tabulaties? "
;g
2090 DIM a(g)
2100 FOR f=1 TO g
2110 PRINT "tab ";f;" = ";
2120 INPUT a(f)
2130 LET a(f)=a(f)-1
2140 IF a(f)>31 THEN GO TO 2120
2150 PRINT a(f)+1
2160 NEXT f
2170 LET z=1
2180 PAUSE 100
2190 GO TO 19
4000 REM *****
4001 REM * EDIT-mode *
4002 REM *****

```

```

4003 REM
4004 REM
4010 PRINT AT 14,6; FLASH 1;"2"
4015 PAUSE 100: CLS
4020 INPUT "geef paginanummer ";
P
4024 IF p>5 OR p<1 THEN GO TO 4
020
4026 CLS
4028 PRINT a$(p)
4030 PRINT AT 0,0; FLASH 1;"_";
4040 LET l=0: LET c=-1
4050 IF INKEY#<>" " THEN GO TO 4
050
4055 IF INKEY#="" THEN GO TO 40
55
4060 LET b#=INKEY#
4070 IF c>30 THEN GO TO 4300
4080 IF b#=CHR$ 13 THEN GO TO 4
350: REM ENTER-toets gedrukt
4090 IF b#=CHR$ 9 THEN GO TO 42
50: REM cursor right gedrukt
4100 IF b#=CHR$ 8 THEN GO TO 44
00: REM cursor left gedrukt
4110 IF b#=CHR$ 10 THEN GO TO 4
500: REM cursor down gedrukt
4120 IF b#=CHR$ 11 THEN GO TO 4
600: REM cursor up gedrukt
4130 IF b#=CHR$ 12 THEN GO TO 4
800: REM DELETE-toets gedrukt
4140 IF b#=CHR$ 205 THEN GO TO
4020: REM STEP-toets gedrukt
4150 IF b#=CHR$ 204 THEN GO TO
4700: REM TO-toets gedrukt
4160 IF b#=CHR$ 195 THEN GO TO
5000: REM NOT-toets gedrukt
4170 IF b#=CHR$ 203 THEN GO TO
6070: REM THEN-toets gedrukt
4180 IF b#=CHR$ 7 THEN GO TO 49
00: REM EDIT-toets gedrukt
4190 IF l=21 AND c=31 THEN GO T
O 4045
4200 IF c=30 THEN GO TO 4350
4210 LET c=c+1
4220 PRINT AT 1,c;b#;
4230 PRINT AT 1,c+1; FLASH 1;"_ "
;
4240 GO TO 4050
4241 REM *****
4242 REM * cursor right *
4243 REM *****
4244 REM

```

```

4245 REM
4250 IF c=30 THEN GO TO 4270
4260 LET c=c+1
4270 PRINT AT 1,c;" ";
4280 PRINT AT 1,c+1; FLASH 1;"_"
;
4290 GO TO 4050
4300 REM *****
4301 REM *      regel vol      *
4302 REM *****
4303 REM
4304 REM
4305 PRINT AT 1,31;" ";
4310 LET c=0: LET l=1+1: GO TO 4
090
4345 REM *****
4346 REM *      ENTER      *
4347 REM *****
4348 REM
4349 REM
4350 IF l=21 THEN GO TO 4050
4360 IF b#=CHR$ 13 THEN PRINT A
T 1,c+1;" ";
4365 IF b#<>CHR$ 13 THEN PRINT
AT 1,c+1;b#;
4370 LET c=-1: LET l=1+1
4380 PRINT AT 1,0; FLASH 1;"_";
4390 GO TO 4050
4395 REM *****
4396 REM *      cursor left  *
4397 REM *****
4398 REM
4399 REM
4400 IF c=-1 THEN GO TO 4440
4410 PRINT AT 1,c;" ";
4420 PRINT AT 1,c+1;" ";
4430 LET c=c-1
4440 PRINT AT 1,c+1; FLASH 1;"_"
;
4450 GO TO 4050
4495 REM *****
4496 REM *      cursor down  *
4497 REM *****
4498 REM
4499 REM
4500 PRINT AT 1,c+1;" ";
4510 IF l=21 THEN GO TO 4540
4520 LET l=1+1
4530 LET c#=SCREEN$ (1,c)
4540 PRINT AT 1,c+1; FLASH 1;"_"
;
4550 IF INKEY#="" THEN GO TO 4550

```

```

4560 LET b$=INKEY$
4570 IF b$<>CHR$ 10 THEN GO TO 4050
4580 PRINT AT 1,c+1;c$;
4590 GO TO 4510
4595 REM *****
4596 REM *      cursor up      *
4597 REM *****
4598 REM
4599 REM
4600 PRINT AT 1,c+1;" "
4610 IF l=0 THEN GO TO 4640
4620 LET l=l-1
4630 LET c$=SCREEN$ (1,c)
4640 PRINT AT 1,c+1; FLASH 1;"_"
;
4650 IF INKEY$="" THEN GO TO 4650
4660 LET b$=INKEY$
4670 IF b$<>CHR$ 11 THEN GO TO 4050
4680 PRINT AT 1,c+1;c$;
4690 GO TO 4610
4695 REM *****
4696 REM *      TO      *
4697 REM *****
4698 REM
4699 REM
4700 PRINT AT 1,c+1;" "
4710 IF z>g THEN LET z=1
4720 LET c=a(z)
4730 PRINT AT 1,c+1; FLASH 1;"_"
;
4740 LET z=z+1
4750 GO TO 4050
4795 REM *****
4796 REM *      DELETE      *
4797 REM *****
4798 REM
4799 REM
4800 INPUT "Welke regel? ";n
4810 IF n>21 THEN GO TO 4800
4820 PRINT AT 1,c+1;" ";
4830 FOR k=0 TO 31
4840 PRINT AT n,k;" ";
4850 NEXT k
4860 PRINT AT 1,c+1; FLASH 1;"_"
;
4870 LET l=n: LET c=c-1
4880 GO TO 4050
4895 REM *****
4896 REM *      EDIT      *

```

```

4897 REM *****
4898 REM
4899 REM
4900 INPUT "Welke pagina? ";p
4940 FOR i=0 TO 21
4950 FOR j=0 TO 31
4960 LET a$(p,1+(j+32*i))=SCREEN
$ (i,j)
4970 NEXT j
4980 NEXT i
4990 GO TO 19
4995 REM *****
4996 REM *          NOT          *
4997 REM *****
4998 REM
4999 REM
5000 COPY
5010 GO TO 4050
6000 REM *****
6001 REM *          SAVE          *
6002 REM *****
6003 REM
6004 REM
6010 PRINT AT 16,6; FLASH 1;"3"
6020 PAUSE 100
6030 CLS
6040 PRINT "Save het programma o
p cassette."
6050 PRINT AT 4,0;"Zet de record
er op opname."
6055 PRINT AT 6,0;"Druk daarna o
p ENTER."
6060 SAVE "ztext"
6070 GO TO 19
8000 REM *****
8001 REM *          help          *
8002 REM *****
8003 REM
8004 REM
8010 PRINT AT 18,6; FLASH 1;"4"
8020 PAUSE 100
8030 CLS
8040 PRINT "Met dit programma ku
nt u tot"
8050 PRINT "maximaal 5 pagina's
tekst ver-"
8060 PRINT "werken. Er zijn vers
cheidene"
8070 PRINT "pagina-opmaakfunctie
s beschik-"
8080 PRINT "baar en in de EDIT-m
ode kunt u"

```

```

8090 PRINT "de tekst bewerken."
8100 PRINT
8110 PRINT "Omschrijving van de
functies:"
8120 PRINT
8130 PRINT "ENTER      : naar volge
nde regel"
8140 PRINT "CURS.>    : spatie"
8150 PRINT "CURS.<    : backspace"
8160 PRINT "CURS.^    : regel omho
og"
8170 PRINT "CURS.!    : regel omla
ag"
8180 PRINT "DELETE    : verwijder
regel"
8190 PRINT "STEP      : EDIT pagin
a"
8200 PRINT "TO        : naar volge
nde TAB-stop"
8210 PRINT "EDIT      : bewaar pag
ina in a$"
8220 PRINT "THEN      : verwijder
pagina"
8230 PRINT "NOT       : COPY pagin
a op printer"
8235 PRINT AT 21,0; FLASH 1;"Dru
k op CONT om verder te gaan."
8240 STOP
8250 CLS
8260 GO TO 15

```

```

**** * * * * *
  * * * * *
** * * * *
* * * * *
**** * * * *

```

```

1      customize
2      EDIT-mode
3      SAVE
4      help

```

Met dit programma kunt u tot maximaal 5 pagina's tekst verwerken. Er zijn verscheidene

pagina-opmaakfuncties beschikbaar en in de EDIT-mode kunt u de tekst bewerken.

Omschrijving van de functies:

ENTER : naar volgende regel
CURS.> : spatie
CURS.< : backspace
CURS.^ : regel omhoog
CURS.↓ : regel omlaag
DELETE : verwijder regel
STEP : EDIT pagina
TO : naar volgende TAB-stop
EDIT : bewaar pagina in a\$
THEN : verwijder pagina
NOT : COPY pagina op printer

Druk op CONT om verder te gaan.

Na de functie "1" uit het hoofdmenu ben ik nu in de EDIT mode gegaan.

De vorige regel was gestart met het eerst tabulatie teken, deze regel is begonnen met de tweede tabulatie.

De eerste tabulatie is op 5 gezet, de tweede op 10.

Nu volgt enig tabulatie-gebruik:

1	2
32	41
14	312

-

6. Enkele slotopmerkingen

Aan het eind van onze bespiegelingen gekomen is het onze vurige wens dat we de lezer hebben kunnen overtuigen van het grote belang van het hulpmiddel computer.

Net zoals de stoommachine de industriële revolutie inleidde, brengt de computer in feite een stille omwenteling op gang waarvan de weerslag alle lagen van onze beschaving treft. Een opmerkelijke eigenschap onderscheidt deze beide revoluties. De stoommachine heeft de kwantiteit van de produktie opgevoerd. De professionele computer dient tevens de kwaliteit van die produktie. De stoommachine beïnvloedde slechts het professionele milieu, de computer raakt ook aan het gezinsleven en het onderwijs. Hij neemt steeds terugkerende, inspannende en/of oninteressante taken op zich en daarvan bevrijd kan de mens zijn inventieve en creatieve mogelijkheden ontwikkelen.

Ons tijdperk van crises streeft naar een verdeling van de arbeid en verruiming van de vrije tijd. Wij twijfelen er niet aan of de microcomputer zal een actief bestanddeel van onze vrije tijd worden. In dit stadium menen wij de eerste schreden van de gebruiker in goede banen te hebben geleid door hem te wijzen op het belang van een zorgvuldige analyse van zijn doelstellingen en van elke te maken keuze.

Bewust van de belangstelling van het publiek voor de produkten van Sinclair, komt het ons voor dat de ZX Spectrum een doelmatige oplossing biedt waarvan de prijs/prestatieverhouding meer dan gunstig is. Wij hopen te hebben bijgedragen aan een voorspoedige integratie van dit nieuwe medium. Dit boek is voor alles bedoeld om kennis te spreiden en enthousiasme te wekken.

Als het juist is dat informatica-activiteit intellectueel en vaak individueel is dan vloeit er kennis voort uit deze spreiding. Alvorens afscheid te nemen is ons laatste advies aan de lezer zich aan te sluiten bij een microcomputer-club. Vaak lokaal of regionaal, zijn deze verenigingen van 'gegrepenen' vaak de springplank naar een doelmatiger gebruik van uw computer. De microcomputer-club kan u wel vertellen welke in de buurt van uw woonplaats de plaatselijke groepen zijn.

Voor nadere informatie over lidmaatschap en activiteiten van computerclubs kunt u zich wenden tot:

De Hobbycomputerclub
Ledenadministratie
Postbus 240
2640 AE Pijnacker
Nederland

Deze club heeft ook een afdeling in België; of u kunt zich wenden tot:

Micro Europe
Chemin du Moulin 38
B-1328 Ohain
Belgique

Voor opbouwend commentaar houden we ons gaarne aanbevolen. Met uw suggesties kunt u zich wenden tot de uitgever, Kluwer Technische Boeken te Deventer of tot Edition Didécar, BP 1006, B-1300 Wavre.

7. Aanhangsels

7.1 ASCII-code

Om tot een zo doelmatig mogelijke omzettingstabel te komen zijn in elk vakje meerdere betekenissen voor eenzelfde code ondergebracht. Onderstaande tabel geeft de verschillende verklaringen.

		©	Spectrum-code
	DEL		ASCII-code
decimaal	127	7F	hexadecimaal

7.2 Verklarende woordenlijst

Gezien het belang van het Engels als informatiebron is van bepaalde belangrijke termen het Engelse woord tussen haakjes vermeld. Cursief gedrukte woorden verwijzen naar een definitie elders in deze lijst.

accumulator	Hoofdregister van de microprocessor. Wordt speciaal door de rekenkundig/logische eenheid (ALU) gebruikt en fungeert als tussengeheugen bij informatie-overdracht tussen geheugen en randapparatuur.
adres	Een getal dat naar een byte in het geheugen verwijst.
afbreken	Grammaticale procedure die de plaats aangeeft waar een woord aan het eind van een regel moet worden afgebroken.
afkappen	Bij het manipuleren met strings om een bepaald deel van een reeks karakters weg te nemen.
alfanumeriek	Bedoeld worden alle afdrukbare karakters, hoofdletters en kleine letters, cijfers en speciale schriftekens.
algoritme	Een gedefinieerde opeenvolging van regels of bewerkingen om een probleem met een eindig aantal instructies op te lossen.
ALU	Afkorting van Arithmetic Logic Unit; rekenkundig/logische eenheid. Onderdeel van de microprocessor dat aan de getallen opgeslagen in de accumulator uitsluitend rekenkundige bewerkingen uitvoert als optellen, aftrekken (door optellen van het complement), vermenigvuldigen (door herhaald optellen) en delen (door herhaald aftrekken). De ALU zorgt tevens voor de logische bewerkingen als EN, OF, NIET en VERGELIJKEN.
AND	Logische EN-poort waarvan het uitgangssignaal 'waar' is als alle ingangssignalen 'waar' zijn en het uitgangssignaal 'niet waar' is als een enkel uitgangssignaal 'niet waar' is.
apparatuur (hardware)	Het fysische samenstel van computerapparatuur en -randapparatuur.
argument	1. Een variabele die op afroep van een subroutine wordt verstrekt. 2. Een variabele waarvan de waarde van de functie afhangt. 3. De referentie nodig om toegang te krijgen tot de waarde in een tabel, ook wel <i>index</i> genoemd.

ASCII	Afkorting van American Standard Code for Information Interchange. Binaire 7-bits code waarmee 128 codes kunnen worden samengesteld: 96 afdrubbare codes en 32 stuurcodes.
assembleerprogramma (assembler)	Computerprogramma dat een programma geschreven in symbolische taal (symbolen en mnemonics om er instructies en adressen mee aan te geven) omzet in binaire machinetaal.
assembleren	Het omzetten van symbolische programma's (geschreven met mnemonics) naar programma's in machinetaal (geschreven in bijvoorbeeld hexadecimale codes).
asynchroon	Heeft betrekking op de telecommunicatie; het karakter-voor-karakter overbrengen van data zonder systematische chronologische koppeling tussen de karakters.
BAUD	Eenheid om transmissiesnelheid in uit te drukken, gelijk aan het aantal signaalveranderingen per seconde. Het aantal BAUD's is gelijk aan het aantal bits per seconde als, en alleen als, een verandering precies overeenkomt met een bit.
BCD	Afkorting van Binary Coded Decimal. Dit soort code gebruikt vier binaire informatie-bits om een decimaal getal van 0 t/m 9 weer te geven. Een getal van meerdere decimalen wordt derhalve weergegeven door evenveel groepen van vier bits als er decimalen zijn.
bestand	Een samenstel van gegevens van allerlei aard met betrekking tot een bepaald onderwerp. Deze gegevens kunnen op verschillende manieren georganiseerd zijn: Sequentieel: waar bij de records elkaar opvolgen. Willekeurig: waarbij de plaats van een record niet door een algoritme wordt aangegeven. Geïndiceerd: waarbij de plaats van een record in verkorte vorm wordt vastgelegd, de index in een tabel.
bibliotheek	Een samenstel van programma's en gegevens opgeslagen in het geheugen (systeemgeheugen en randapparatuur) dat voor het programma toegankelijk is.
bi-directioneel (duplex)	Geeft in de telecommunicatietechniek het soort transmissie aan en wel een waarbij gelijktijdig en onafhankelijk kan worden uitgezonden en ontvangen.
binair (binary)	Heeft betrekking op alle informatie die twee stabiele toestanden kan aannemen. Rekenmethode met de beide tekens 0 en 1.
bit	De naam van het binaire cijfer dat twee waarden - 0 en 1 - kan hebben.

bron	Specifiek voor het programmeren in machinetaal. Het bronprogramma is de eerste stap; wordt door het systeem vertaald in een DOEL-programma.
buffer	Geheugengebied waarin informatie in afwachting van verdere verwerking wordt opgeslagen.
bug	Programmafout die de juiste werking van een programma verstoort.
byte	Elementaire eenheid van informatie bestaande uit een aantal bits dat door de hardware bepaald wordt. In het geval van de Spectrum werkt de microprocessor Z80A met 8-bit bytes.
capaciteit	Grootte die de omvang van een rekenregister aangeeft; geeft het grootste aantal bytes aan dat het kan bevatten.
carry	Zie: overdracht.
CCITT	Afkorting van Consultative Committee on International Telephone and Telegraph. Internationale organisatie, onderdeel van de International Télégraph Union, die normen en technische voorschriften opstelt met betrekking tot de telecommunicatie en PTT-inrichtingen in West-Europa.
code	Samenstel van tekens samengevat in een lijst die de betekenis ervan aangeeft. Voorbeeld: <i>ASCII-code</i> , <i>BCD-code</i> , <i>Operationele code</i> .
commando	Soort instructie met betrekking tot functies die specifiek door het <i>operating system</i> bestuurd worden. Voorbeeld: <i>redigeren</i> , <i>initialiseren</i> van het geheugen, <i>prepareren</i> van diskettes enz.
commentaar	Heeft betrekking op een programma: alle informatie die de werking van dat programma kan verklaren.
compiler	Zet een hogere programmeertaal om in een machinetaal. De compiler voert de vertaling in één slag uit. Na invoeren van het programma verzamelt hij alle voor de werking benodigde elementen en vertaalt die in één slag. Hoewel meestal sneller vergt de compiler voordat hij gebruikt kan worden veel voorbereidend werk en testen. Om een BASIC-programma uit te voeren doet de Spectrum een beroep op een <i>interpreter</i> .
cursor	Aanwijzer voor de operator. Zo kan de cursor tijdens het opstarten van een programma de momentele functie aangeven.

decrement	In afnemende volgorde verminderen met één. Vergelijkbaar met een teller waarvan de inhoud met één wordt verlaagd.
doel-code	Heeft betrekking op de machinetaal; in de opeenvolgende stadia van de ontwikkeling van een programma wordt, uitgaande van het bronprogramma, opgesteld in symbolische taal, gebruik gemaakt van mnemonics. Door te assembleren wordt die opeenvolging van codes omgezet in een overeenkomstige reeks van binaire waarden. Dit type programma heet doelprogramma.
drukker	Trage randapparatuur die de informatie op papier afdruckt. Al naar gelang de wijze waarop de karakters tot stand komen onderscheidt men drukkers met: Letterwiel, waarbij de karakters langs de omtrek van een ronddraaiende schijf zijn aangebracht; Letterkogel, waarbij de karakters op het oppervlak van een bol zijn aangebracht; Matrix, waarbij de karakters tot stand komen met behulp van één of meer naalden die door de volgorde waarin de punten langs evenwijdige lijnen worden afgedrukt, de karakters bepalen. De Sinclair-printer behoort tot deze laatste categorie.
dubbele precisie	Verbeteren van de rekennauwkeurigheid van een computer onder gebruikmaking van een hogere programmeertaal. Geeft aan dat om een getal te kunnen weergeven meerdere bytes nodig zijn. Verdeelt het getal in twee stukken: de mantisse en de exponent. De Spectrum verwerkt alle getallen met dubbele precisie.
duplex	Zie: bidirectioneel.
exponent	Geeft bij verwerken van een getal met dubbele precisie de plaats van de komma aan met behulp van een positieve of negatieve macht van 10 (+38 in het geval van de Spectrum). De exponent geeft het aantal decimalen aan.
gedrukte bedrading	Isolatieplaat met daarop gedrukte kopersporen die de diverse onderdelen van een elektronisch apparaat onderling verbinden.
geheugen	Vitaal onderdeel van een computer. Men onderscheidt twee soorten geheugens: Het leesgeheugen of ROM, afkorting van Read Only Memory, dat het startprogramma en de interpreter van het systeem bevat. De inhoud ervan is niet te wijzigen en blijft ook zonder netspanning bewaard. Het lees/schrijf-geheugen of RAM, afkorting van Random

Access Memory, wat een geheel willekeurig toegankelijk geheugen is. Hierin worden de variabelen, de gebruikersprogramma's en de resultaten weggeschreven. De inhoud ervan gaat bij uitschakelen van de voedingsspanning verloren.

geïntegreerde schakeling (chip) Zeer sterk geminiaturiseerde elektronische schakeling waarin halfgeleiders worden gebruikt om een samenstel van eenvoudige functies te realiseren die samen een complexe schakeling vormen.

half duplex In de telecommunicatietechniek een vorm van communicatie waarbij de informatie-uitwisseling tussen afzender en ontvanger of omgekeerd, afwisselend, ieder op zijn beurt, gebeurt over dezelfde lijn.

hardware Engels woord waarmee de apparatuur wordt bedoeld.

hexadecimaal Kenmerkt een telwijze met 16 toestanden omdat met vier bits *binair* 16 toestanden kunnen worden voorgesteld. Om de cijfers hoger dan 9 (decimaal) te kunnen weergeven gebruikt men de letters A, B, C, D, E en F, en wel als volgt:

decimaal:

0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16

hexadecimaal:

0 1 2 3 4 5 6 7 8 9 10 A B C D E F

increment In stijgende volgorde, verhogen met één. Vergelijkbaar met een teller waarvan de inhoud met één wordt verhoogd.

index Wijzer die informatie omtrent de plaats geeft. Is van toepassing op bestanden, op lussen, op subroutines en wel in het bijzonder op het programmeren in machinetaal waarbij er één of meer registers voor gereserveerd worden.

initialiseren De eerste procedure die na het inschakelen wordt uitgevoerd. Aan de hardware wordt een bepaalde aanvangswaarde toegekend. Aan de software een samenstel van instructies en commando's om met de referentie-elementen een verwerking op gang te brengen.

instructie Eenheid van informatie die normaal deel uitmaakt van een reeks die de computer de te volgen werkwijze opgeeft.

interface Apparatuur waarmee de verschillende technieken van twee soorten apparatuur op elkaar worden aangepast

	en de dialoog mogelijk wordt. In de informatica een elektronische schakeling die het de computer mogelijk maakt om te communiceren en zijn <i>randapparatuur</i> te besturen.
interpreter	In de informatica een systeem om hogere programmeertalen in machinetaal om te zetten. De interpreter analyseert en voert een <i>instructie</i> tegelijk uit. Deze methode vergt minder geheugenruimte dan een <i>compiler</i> , maar in vergelijking daarmee heeft de interpreter door het herhalen van lussen, veel meer tijd nodig. De interpreter heeft tevens het voordeel van gemakkelijk redigeren en een snel en gemakkelijk op gang brengen van programma's. Ook de Spectrum werkt voor het vertalen van BASIC in machinetaal met een interpreter.
interrupt	In machinetaal een detectiesignaal dat de uitvoering van een programma bepaalt. De interrupt kan uit de hardware of uit de software afkomstig zijn. Bijvoorbeeld: een programma dat wacht op het indrukken van een toets op het toetsenbord (het indrukken veroorzaakt een interrupt) waardoor de opeenvolgende bewerkingen (herkennen, interpretatie, beeldpresentatie enz.) worden onderbroken of vertraagd.
K(ilo)-byte	In de informatica heeft het voorvoegsel 'K' de betekenis 2^{10} of wel 1024. Wordt gebruikt om de capaciteit van het systeemgeheugen of van externe geheugens aan te geven.
label	In assembleertaal de eerste van vier delen van een regel, waarmee men door een soortnaam een actief gedeelte of een subroutine van het programma aangeeft.
logica	Vorm van wiskunde volgens de grondbeginselen van de Boolese algebra. Is van toepassing op de <i>binäre</i> notatie en vormt door uitbreiding naar elektronische schakelingen de basis voor de computer en de programma's die hem besturen.
LSB	Afkorting van Least Significant Bit. <i>Bit</i> met de laagste waarde. Tegenovergestelde van <i>MSB</i> .
LSI	Afkorting van Large Scale Integration. Grootschalige integratie, kenmerkt de technologie van de geïntegreerde schakelingen waarmee de Spectrum is uitgerust.
lus (loop)	Reeks van instructies waarvan de laatste weer de eerste als volgende te verwerken instructie aangeeft.

lustabel	Door het <i>operating system</i> in een <i>RAM</i> gereserveerde geheugenruimte om er de adressen in onder te brengen waarnaar het programma na het beëindigen van een lus moet terugkeren.
macro-instructie	In assembleertaal een <i>instructie</i> die de uitvoering van meerdere opeenvolgende micro-instructies omvat.
mantisse	Volgens de definitie van de <i>dubbele precisie</i> wordt een te verwerken getal door de computer in twee delen gesplitst. Het eerste deel, de mantisse, geeft het getal. Het tweede, de exponent, geeft met een macht van tien de plaats van de komma voor dat getal aan.
masker	<i>Binair woord</i> waarmee door logische manipulatie één of meer <i>bits</i> van een ander binair woord naar keus, kunnen worden geanalyseerd.
microdrive	Exclusief voor Sinclair: willekeurig toegankelijk magnetisch geheugen-randapparaat. Capaciteit per stuk 100 Kbyte, overdrachtssnelheid 16 Kbyte per seconde, gemiddelde toegangstijd 3,5 seconden.
micro-instructie	In machinetaal, een <i>instructie</i> .
mnemonic	In machinetaal, het samenstel van alfabetische codes dat gebruikt wordt om er in afgekorte vorm een <i>instructie</i> mee weer te geven.
monitor	Het samenstel van programma's, opgeslagen in <i>ROM</i> , voor het opstarten van het systeem, het besturen van de <i>randapparatuur</i> , het interpreteren van de gebruikers-commando's en het uitvoeren van de programma's.
MSB	Afkorting van Most Significant Bit. Bit met de hoogste waarde. Wordt in machinetaal gebruikt om er de positie van een getal in een register mee aan te geven. Zie LSB.
NOT	Logische, inverterende functie waarvan het uitgangssignaal 'niet waar' is als het ingangssignaal 'waar' is.
opcode	Van het Engelse 'Operation Code', zie de definitie van ' <i>operationele code</i> '. In het kort een alfabetische code die een <i>instructie</i> in machinetaal definieert.
operand	Bij een programma in machinetaal de derde van de vier delen van een regel van elke <i>instructie</i> . Dit deel geeft in symbolische taal of code (binair of hexadecimaal) de met een instructie gebruikte <i>variabele</i> aan.

operating system	In ROM opgeslagen programma's voor het initialiseren van de computer, het vertalen van instructies vanuit hogere programmeertalen in voor de microprocessor begrijpelijke machinetaal en het besturen van gegevens-ontvangende of -afgeevende randapparatuur.
operationele code (opcode)	In assembleertaal de tweede van de vier delen van de regel, die het soort uit te voeren bewerking aangeeft.
operator	Individuele gebruiker van een computersysteem.
OR	Logische OF-functie waarvan het uitgangssignaal 'waar' is als minstens één ingangssignaal 'waar' is.
overdracht (carry)	Overschrijden bij een optelling de bij elkaar getelde waarden van een decimaal het getal 9, dan spreekt men bij de overgang naar de eerstvolgende hogere decimaal van 'overdracht'. Op het niveau van de microprocessor wordt, in machinetaal, de overdracht door een speciale 'vlag' aangegeven.
overflow	Afkomstig uit het Engels; betekent overschrijden van de capaciteit. In machinetaal als aan een accumulator een waarde wordt toegevoerd die hoger is dan met de bytes ervan kan worden voorgesteld. Deze capaciteits-overschrijding wordt door een speciaal detectiecircuit aangegeven met een vlag.
parameter	In de wiskunde een variabel element dat de waarde van een functie beïnvloedt.
pariteit	In de telecommunicatie besturende informatie waarmee fouten als gevolg van de transmissie kunnen worden vastgesteld. De pariteit kan nul zijn, transparant, even of oneven.
patch	Van het Engelse 'oplappen'. In de informatica een opeenvolgende reeks van instructies die in een programma worden opgenomen om er voor een gewijzigde toepassing een functie aan toe te voegen.
programmatuur (software)	Het samenstel van informatie, instructies en commando's die, samengevoegd tot een programma, het de computer mogelijk maken zijn elektronische schakelingen te activeren en de informatie te verwerken.
RAM	Afkorting van Random Access Memory, wat willekeurig toegankelijk geheugen betekent.

randapparatuur	Apparatuur voor het invoeren/uitvoeren van door de computer te verwerken dan wel verwerkte informatie.
real time	Engels voor 'werkelijke tijd', wat wil zeggen dat de gegevens verwerkt worden op het moment dat er een verandering optreedt. Zo moeten bijvoorbeeld temperatuurmetingen 'real time' verwerkt worden omdat de betekenis ervan anders verloren gaat.
redigeren (editing)	In de informatica het wijzigen van programma's of van bestaande bestanden.
refresh	Engels voor 'verversen'. Deze handeling wordt door de computer automatisch uitgevoerd aan dynamische geheugenschakelingen (RAM). Technisch houdt dit in dat met regelmatige intervallen de geheugenschakelingen elektronisch moeten worden verversd om de informatie in het geheugen te kunnen behouden.
register	Geheugenzone die op het niveau van de microprocessor als tussenopslag fungeert. Deze zones zijn met de instructie-set van de machinetaal direct toegankelijk. Het zijn de eigenlijke werkzones van het systeem. Een speciaal register is de accumulator. Alle door de rekenkundig/logische eenheid bewerkte gegevens passeren de accumulator.
ROM	Afkorting van Read Only Memory. Letterlijk een geheugen dat alleen kan worden uitgelezen. In de praktijk een type geheugen waarvan de inhoud vastligt en niet door manipuleren met elektrische signalen kan worden gewijzigd.
RS232C	Nummer van een door de EIA (Electronic Industry Association) opgestelde technische standaard, waarin specificaties ten behoeve van de telecommunicatie worden gegeven.
sequentieel	Kenmerk van een bestand waarin de gegevens de één na de ander op elkaar volgen; zoals bijvoorbeeld de informatie die op een magneetband is vastgelegd.
shift	In machinetaal het manipuleren met de inhoud van een register waardoor de informatie rechtsom of linksom draait. Voor het toetsenbord verstaat men onder SHIFT het overgaan op hoofdletters of kleine letters.
simplex	Geeft in de telecommunicatie een enkelvoudige verbinding tussen zender en ontvanger aan.
sleutel	Mechanisch hulpmiddel om te verhinderen dat een symmetrische connector op verkeerde wijze wordt aangesloten.

software	Engelse term voor <i>programmatuur</i> .
sprong (jump)	In machinetaal komt dit overeen met een vertakkings-instructie.
stapel (stack)	Het samenstel van registers dat deel uitmaakt van de microprocessor. Een stapel vergemakkelijkt het manipuleren met indices of wijzers.
string	Heeft betrekking op een reeks van lettervariabelen; een opeenvolging van alfabetische karakters.
subroutine	Reeks van <i>instructies</i> die met gegevens manipuleren en waarvan de functie in één en hetzelfde programma verscheidene malen worden gebruikt. De subroutine wordt geselecteerd door een <i>wijzer</i> op het niveau van een <i>switch</i> (wissel).
syntax	Het samenstel van regels en conventies voor het schrijven van een programmeertaal.
systeem	Het voor een constructeur specifieke samenstel van apparatuur en programmatuur.
taal	Het samenstel van <i>instructies</i> en grammaticaregels aan de hand waarvan programma's kunnen worden samengesteld. Men onderscheidt: Assembleertaal en machinetaal waarbij de uit te voeren instructies worden uitgedrukt in eenvoudige aanwijzingen met weinig draagkracht zoals <i>mnemonics</i> of cijfercodes. Hogere programmeertalen, zoals BASIC, waarbij de communicatie met de computer plaats vindt door middel van instructies die de menselijke taal zo dicht mogelijk benaderen en waarbij getracht wordt eenzelfde <i>syntax</i> te volgen.
teken (sign)	Geeft een reëel getal aan en of dit positief of negatief is. Wordt op het niveau van de microprocessor aangegeven met een speciale vlag.
terminal	Andere naam voor een <i>randapparaat</i> voor het uitvoeren van (toetsenbord) of ontvangen van informatie (bijvoorbeeld op het beeldscherm).
toegangstijd (access time)	Kenmerkt het verwerken van informatie. Voor zover het randapparatuur als disktestations betreft wordt er de tijd onder verstaan die de apparatuur nodig heeft om toegang te krijgen tot en het overbrengen van specifieke gegevens.

underflow	Engelse uitdrukking voor negatieve overdracht (ook wel 'Borrow' genoemd); tegenovergestelde van overschrijden van de capaciteit (<i>overflow</i>).
variabele	<p>Algemene term voor alle symbolen van het programma die verschillende waarden kunnen aannemen. Men onderscheidt:</p> <ul style="list-style-type: none"> - Systeemvariabelen afkomstig van het operating system. Deze geven of het adres van bepaalde specifieke geheugenplaatsen aan of een grootte die door het programma kan worden geconditioneerd (tellen van regels, duur van het geluid, kleur van het beeldscherm). - Numerieke variabelen, programmaparameters waaraan alleen in cijfers uit te drukken waarden kunnen worden toegekend. - String-variabelen, programmaparameters waaraan alleen alfanumerieke karakters (inclusief cijfers) als waarden kunnen worden toegekend.
vertakking (branch)	<p>Plaats in een programma waar afhankelijk van een variabele één of meer verschillende keuzes worden gemaakt. Er bestaan twee soorten:</p> <p>de voorwaardelijke vertakking, die tot stand komt als functie van een expliciete voorwaarde en dan ook wel wissel wordt genoemd, en</p> <p>de onvoorwaardelijke vertakking als instructie voor het uitvoeren van een lus.</p>
vlag (flag)	In machinetaal de specifieke informatie-bit die de toestand rond bepaalde acties van de <i>accumulator</i> aangeeft, bijvoorbeeld: overflow, carry, zero, sign.
V-24	Nummer van een door de CCITT (Consultative Committee on International Telephone and Telegraph) opgestelde technische standaard, waarin specificaties ten behoeve van de telecommunicatie zijn vastgelegd; op soortgelijke wijze als bij de RS232C.
wetenschappelijke notatie	Schrijfwijze voor getallen zoals die ook door de computer bij het rekenen met <i>dubbele precisie</i> wordt gebruikt en waarbij het getal in twee delen wordt splitst. De <i>mantisse</i> en de <i>exponent</i> die de plaats van de komma met een macht van tien aangeeft.
wissel (switch)	Een wijze van programmeren waarbij afhankelijk van de functie van een <i>variabele</i> in het programma, een keus wordt gemaakt.
woord	Het samenstel van bits dat door de computer als een-

heid van informatie wordt verwerkt. Het aantal bits wordt door de *apparatuur* bepaald. In het geval van de Spectrum zijn dat 8 bits.

wijzer (index) Variabele waarvan de waarde de keus van een wissel bepaalt.

zwevende komma Zoals bij de wetenschappelijke notatie waarbij een grootte in cijfers in twee delen (mantissee en exponent) wordt geschreven. De Engelse uitdrukking hiervoor is 'Floating point'.

