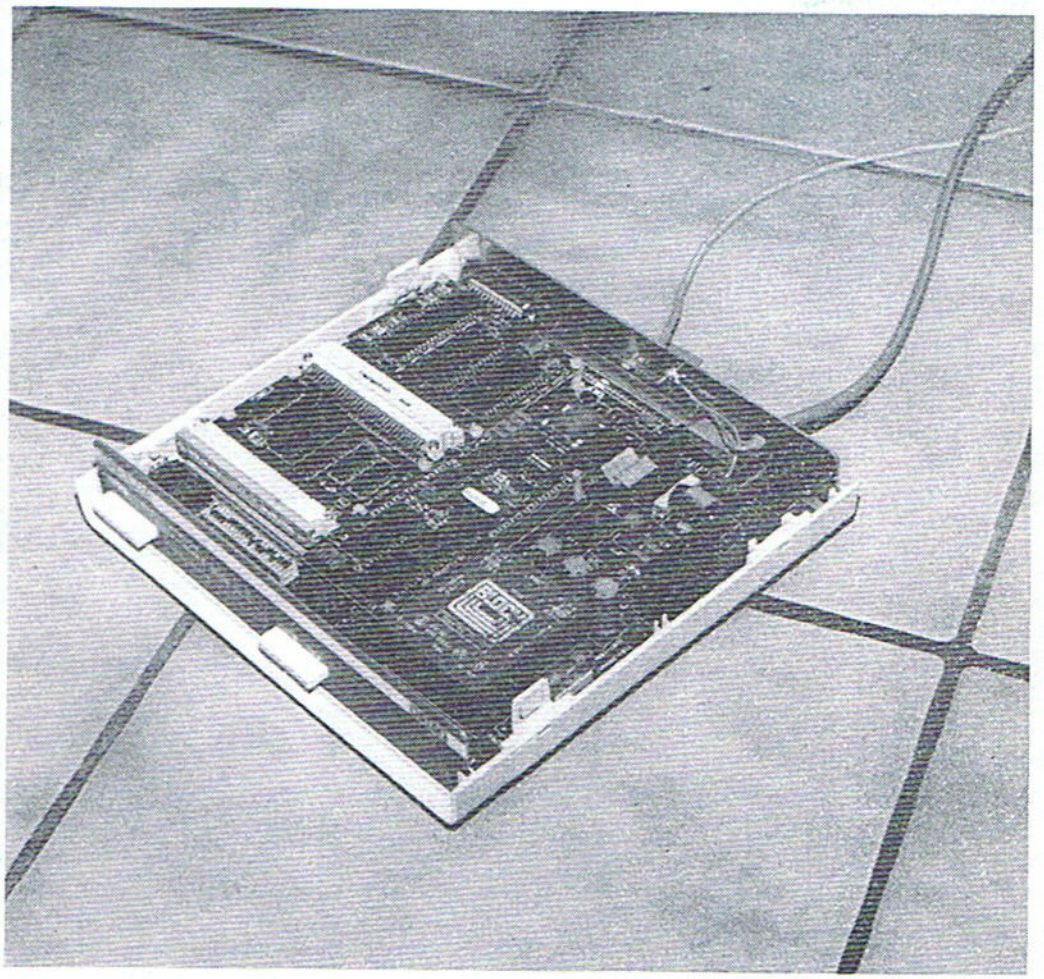
De µP Kenner

Veertiende jaargang nr. 2
April 1990
66



In dit nummer o.a.:

Werking van het openbare telefoonnet
Een harddisk voor DOS-65
Pointers in PASCAL en C
Werking van autorouters
Geluiden uit DOS-65

Inhoudsopgave

De μP Kenner

Nummer 66, april '90
Verschijnt 5 maal per jaar
Oplage: 250 stuks
Druk: FEBO Offset, Enschede

De redactie:

Gert van Opbroek
Bram de Bruine
Antoine Megens
Nico de Vries
Joost Voorhaar

Eindredactie:

Gert van Opbroek

Vormgeving:

Joost Voorhaar

Redactieadres:

Gert van Opbroek,
Bateweg 60
2481 AN Woubrugge

De μP Kenner nummer 67 verschijnt op 18 augustus 1990.

Copijsluitingsdatum voor nummer 67 is vastgesteld op 20 juli 1990.

Vereniging

Algemeen

Fundamenten

Datacommunicatie

Systemen

Talen/software

De μP Kenner is het huisorgaan van de KIM gebruikersclub Nederland en wordt bij verschijnen gratis toegezonden aan alle leden van de club. De μP Kenner verschijnt vijf maal per jaar, in principe op de derde zaterdag van de maanden februari, april, augustus, oktober en december.

Copy voor het blad dient bij voorkeur van de leden afkomstig te zijn. Deze copy kan in papier-, maar liever in machine-leesbare vorm opgestuurd worden aan het redactieadres. Copy kan ook op het Bulletin Board van de vereniging gepost worden in de redactie area.
Nadere informatie kan bij het redactieadres of via het bulletin board opgevraagd worden.

Redactioneel

Excuses

In de eerste plaats wil ik, namens de hele redactie, mijn excuses aanbieden voor het feit dat de μP Kenner de vorige keer véél te laat uitgekomen is. Het was al maart toen nummer 65 van februari bij u in de bus lag; eigenlijk schandalig. U dacht natuurlijk al dat de club ongemerkt terziele gegaan was, wel dat is niet zo, we zijn nog springlevend en zitten vol van energie.

Wat is er allemaal gebeurd? In de eerste plaats was het al ruim aan de late kant voordat de kopij voor het blad bij elkaar was, maar dat was gelukkig nog op het nippertje op tijd klaar. Toen bleek dat het layouten van het blad aanzienlijk meer tijd en inspanning vergde dan ingepland was. Toen dat, rijkelijk laat, gebeurd was, bleek dat onze planning helemaal niet paste bij de planning van de drukker zodat het al met al zo'n twee weken later was dan de datum die we eigenlijk in ons hoofd hadden.

Ik hoop dat het deze keer wat beter gaat, hoewel ik ook nu weer aan de late kant ben met het samenstellen van het blad. Dat komt omdat ik altijd op het laatste moment bedenk wat ik zal gaan schrijven. Dat is eigenlijk een hele nare karaktereigenschap van me; ik doe alles altijd op het laatste nippertje, maar wie weet, leer ik dat nog eens af.

Datacommunicatie

De ervaring leert dat de clubbijeenkomst in mei altijd stevig moet concurreren met het mooie voorjaarsweer. Meestal wordt deze bijeenkomst dan ook matig bezocht (het trieste record van een aantal jaren geleden staat op zes personen). We willen dat eigenlijk dit jaar eens anders hebben en de redactie en het bestuur heeft besloten flink in de aanval te gaan. We gaan de strijd met het mooie weer aan en hebben voor mei een bijeenkomst met als thema datacommunicatie bedacht. We willen, behalve een lezing van Jacques Banser over bulletin boards en van mij over datacommunicatie over een telefoonlijn ook zorgen voor uitgebreide demonstraties. We willen proberen een (bedrijfs-) telefooncentrale te regelen waarop we enkele bulletin boards aan willen sluiten. Via een modem kunnen andere aanwezigen dan contact met deze bulletin boards zoeken. U kunt dus, behalve uw systeem, ook uw modem mee nemen. Verder zorgen we ervoor dat we voor de diverse systemen datacommunicatie-software hebben zodat die, tegen kostprijs, verspreid kan worden. Het gaat hier uitdrukkelijk om Public Domain software, Shareware en vergelijkbare pakketten. Behalve communicatie-software zullen we ook proberen software voor datacompressie en -expansie te bemachtigen.

Verder willen we proberen ook wat interessante aanbiedingen te hebben. We denken hierbij in de eerste plaats aan een leverancier van bijvoorbeeld modems en in de tweede plaats denken we aan het verkopen van IC's waarmee IBM-klonen uitgerust kunnen worden met een tweede COM-poort. De meeste seriële kaarten hebben namelijk één COM-poort en een lege IC-voet. We willen proberen het bijbehorende IC tegen een schappelijke prijs te verkopen.

Tenslotte willen we door middel van het aanschrijven van bijvoorbeeld regionale kranten en huis-aan-huis bladen proberen ook buiten de club wat ruchtbaarheid aan de bijeenkomst te geven. Tenslotte heeft het bestuur van de club zich als doel gesteld aan het eind van het jaar meer leden te hebben dan aan het eind van het vorige jaar. We moeten er dus voor zorgen dat potentiële leden van ons horen.

Ik hoop dan ook dat ik u op de bijeenkomst in mei, in Almere deze keer, mag begroeten en we zouden het prettig vinden als u ook uw buurman of collega over zou kunnen halen mee te komen.

Kopij

Sinds de vorige uitgave is het blad duidelijk veranderd. Ook wat betreft de inhoud zijn we binnen de redactie een iets ander beleid gaan voeren en wat ik van een aantal leden gehoord heb, spreekt het blad de leden goed aan. Zoals ik de vorige keer al aangegeven heb, vinden we dat een echte brievenrubriek in ons blad ontbreekt. Ik zou u op willen roepen om toch vooral te reageren als u iets vragen of te vertellen heeft of anderzijds op het blad of de club wilt reageren. Verder kunnen we natuurlijk kopij altijd gebruiken, maar dat wist u al.

Tenslotte wens ik u veel plezier met uw computerhobby en hopelijk tot ziens in Almere.

Gert van Opbroek

Uitnodiging voor de clubbijeenkomst

Datum: Zaterdag 19 mei 1990
Locatie: Buurtcentrum "De Inloop"
's-Hertogenboschplein 8
1324 WB Almere-Stad
Telefoon 03240-33737

Entree: *f* 10,00

Thema: Datacommunicatie

Routebeschrijving

Per auto:

Vanaf de A6 neemt u de afslag Almere-Stad. Direct daarna afslag Gooise kant (Paralelweg A6). Dan linksaf de havendreef tot het benzinepompstation Esso, dan rechtsaf de weg volgen (Rotterdamweg). Voor de busbaan rechtsaf.

Per openbaar vervoer:

Met de trein rijdt u naar het station Almere-Muziekwijk. Vervolgens kunt u met de stadsbus (lijn 12) tot voor De Inloop vervoerd worden.

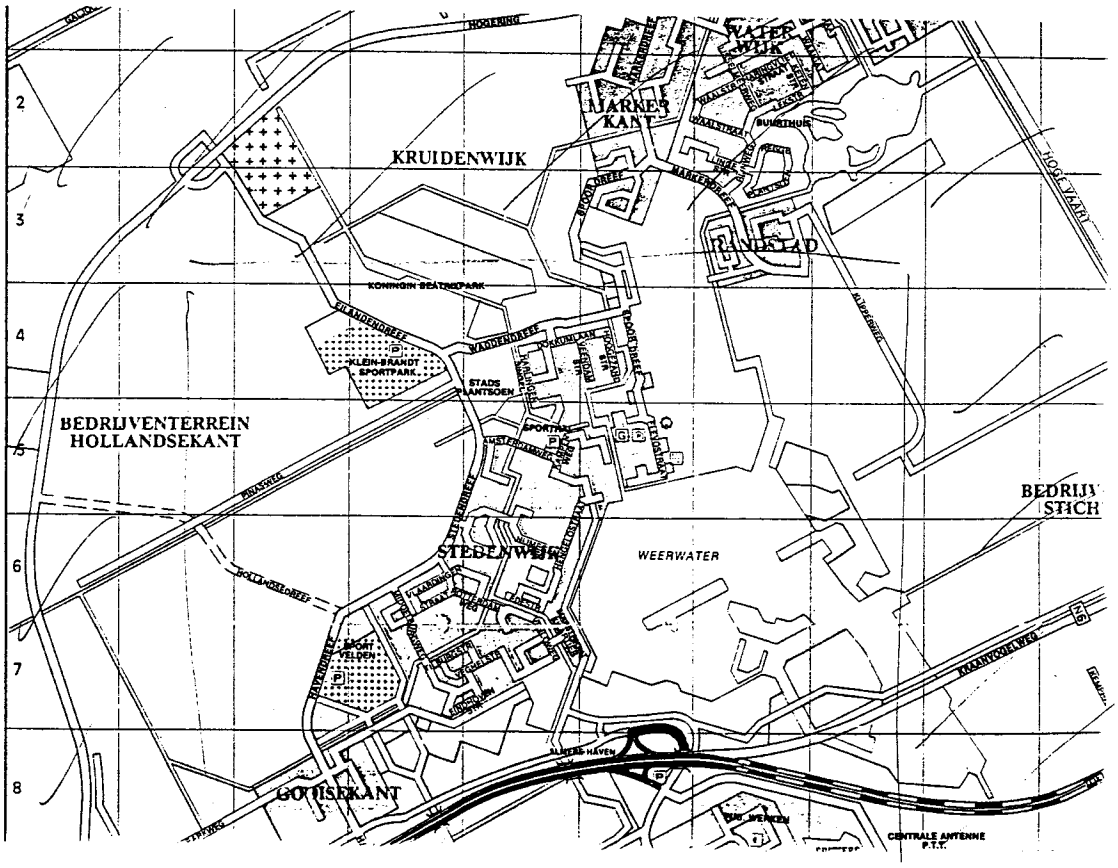
Programma

9:30	Zaal open
10:00	Opening van de clubbijeenkomst
10:10	Voordracht door Gert van Opbroek over datacommunicatie met behulp van telefoonlijnen (zie toelichting)
11:30	Forum en markt
12:00	Lunch; consumpties tegen betaling
14:00	Voordracht van Jacques Banser over The Ultimate; het bulletin board van de KIM Gebruikersclub Nederland
17:00	Sluiting

Verder de hele dag demonstraties van modems in combinatie met een telefooncentrale, bulletin boards en communicatie-software. Bovendien is er uitgebreid gelegenheid met mede clubleden van gedachten te wisselen over hardware software etc. en om Public Domain software uit te wisselen. Breng daarom uw eigen systeem mee (met modem...!?)

Attentie

Het is ten strengste verboden illegale kopieën van software te verspreiden. Aan personen die deze regel overtreden zal de verdere toegang tot de bijeenkomst ontzegd worden. Breng verder alleen software mee die u legaal in uw bezit heeft. Het bestuur aanvaardt geen enkele aansprakelijkheid voor de gevolgen van het in bezit hebben van illegale software.



Toelichting op de aanstaande clubbijeenkomst

De lezing van Gert van Opbroek zal met name gaan over de technische kant van datacommunicatie over een telefoonlijn. De werking van modems zal uitgelegd worden waarbij het vooral zal gaan over de PTT-kant van het modem. Behalve een mondelinge toelichting zullen in de voordracht ook een aantal demonstraties opgenomen worden.

De lezing van Jacques Banser zal voornamelijk gaan over wat er allemaal op een Bulletin Board gebeurt en wat je er zoal mee kunt. Jacques is al enkele jaren de Sysop van ons Bulletin Board en hij zal vanuit zijn kennis en ervaring vertellen wat een Bulletin Board eigenlijk is.

Op de bijeenkomst zal een telefooncentrale aanwezig zijn. Hierop zullen minstens twee Bulletin Boards aangesloten worden. Als u problemen hebt om met uw systeem, modem en communicatie software contacten met een BBS te leggen, breng dan het systeem en het modem mee, dan kunnen ze op de bijeenkomst uitgetest worden.

Voor leden hebben we een bijzondere aanbieding. Een groot aantal PC's heeft de beschikking over één seriële poort terwijl op de uitbreidingskaart nog een paar lege voeten zijn om een tweede poort te bouwen. Op de bijeenkomst bieden we de benodigde IC's (16450, 1488 en 1489) voor *f* 10,00 per pakket. Hiermee kunt u het aantal seriële poorten op uw PC dus uitbreiden van 1 naar 2.

Op de bijeenkomst zal vrij verspreidbare communicatie-software verkregen kunnen worden tegen kostprijs. De organisatoren zullen er voor zorgen dat voor de actuele systemen binnen de club de meest recente versies aanwezig zullen zijn.

Laat u niet door het mooie weer afschrikken en komt in grote getale. Breng bovendien uw systeem en eventueel uw modem mee. Introducés zijn uiteraard van harte welkom.

(Advertentie)

Gevraagd:

Wie heeft de DOS OS-65D V3.1 of V3.3 voor een Junior met de door Elektuur in 1982 gepubliceerde floppy-disk controller.

Reacties graag aan:
René Kemelinckx
Hanenstoot 13
B-3382 Kortenaken (Belgie)
of aan de redactie

Te koop

Wegens ruimtegebrek wordt te koop aangeboden:

Een DOS-65 systeem bestaande uit de volgende componenten:

- Uitgebreide Junior 6502 computer
- Elektuur grote buskaart
- Elektuur VDU-kaart
- Elektuur dynamische 64k RAM kaart
- Elektuur statische 64k RAM kaart (bevat 16k)
- Zelfbouw I/O kaart (2 * 6522, pieper, etc.)
- Elektuur voeding met cassette-deck
- Elektuur toetsenbord
- DOS-65 disk controller kaart
- 2 floppy drives met ingebouwde voeding
- DOS-65 software en documentatie
- Datapoint terminal, te gebruiken als monitor

Vraagprijs: *f* 750,00

Reacties graag aan:
Jeroen Oort
Bentinck straat 85-3
1051 GH Amsterdam
tel. 020-866740 ('s avonds)

Transputers III

Bouwstenen van de toekomst

Dit is een triest moment. Zware klassieke muziek begeleidt deze aflevering in de serie over transputers. Het begin van het einde... Dit is de laatste aflevering van de serie over transputers. Deze keer ga ik het hebben over de process scheduler, de buitenkant van het zwarte doosje waar de gemiddelde transputer zich in pleegt op te houden en tenslotte zal dan een kleine vergelijking met andere, meer gangbare systemen de serie afsluiten.

Over processen en geel-groen gestipte tijdsvretertjes

Een processor kan (helaas) slechts één ding tegelijkertijd. Door de verschillende processen heel snel achter elkaar een klein stukje te laten lopen lijkt het alsof de processen zich tegelijkertijd afspelen. Dit verschijnsel wordt in de computerwereld aangeduid met de term "timesharing".

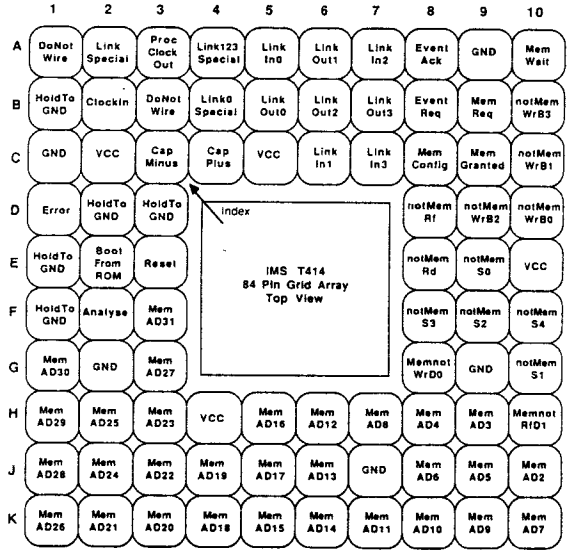
Voor het verdelen van de processortijd en bijhouden van registerwaarden en ander huishoudelijk werk draait er in de meeste multitasking systemen een speciale taak die naar de naam "process scheduler" luistert. De process scheduler is een stuk software dat, als het niet voor de volle honderd procent geoptimaliseerd is, bijzonder veel tijd vraagt van het systeem. In sommige gevallen kan de process scheduler tot wel vijftig procent van de totaal beschikbare processortijd opeten. De transputerontwerpers van INMOS vonden dat het tijd was dat daar wat aan gedaan werd. En dus... bouwden ze een hardwarematige process scheduler.

Een proces binnen de transputer kan actief zijn, stand-by, passief of afgebroken. Is het process actief of standby, dan is het in het normale time-sharing systeem opgenomen. Aan het actieve proces wordt op dat moment gewerkt door de processor. De stand-by status houdt in dat het proces staat te wachten op processortijd. Een passief proces is een proces dat staat te wachten op I/O handelingen en een bepaalde tijd "down" is. Een afgebroken proces kan, bijvoorbeeld door een foutconditie, niet afgemaakt worden.

De workspaces van alle processen zijn opgenomen in één lange geketende lijst (een "linked list"). De process scheduler heeft een zogenaamd "front register" die een pointer bevat die naar het begin van de lijst verwijst. Een andere pointer, het "back-register", wijst naar het einde van de lijst. Een actief process wordt door de process scheduler van de lijst afgekoppeld in behandeling gegeven. Het proces kan stilgelegd worden door een I/O-instructie, een delay instructie, een halt-instructie of simpelweg doordat zijn tijd op is. In het laatste geval krijgt het proces de halt-status en wordt voor eeuwig uit de lijst geknikkerd. Is het proces bezig met I/O, dan heeft de processor er niets meer mee van doen en krijgt het proces de passieve status. In de andere gevallen wordt het process aan het einde van de lijst weer ingevoegd. De werkregisters worden dan in de workspace gesaved en het backregister wordt netjes bijgewerkt. Het volgende process is nu aan de beurt.

In werkelijkheid zijn er zelfs twee verschillende linked lists. Iedere lijst heeft z'n eigen timer en z'n eigen front- en backregister. De lijsten staan in direct verband met de prioriteit van het process. De lijst met prioriteit 0 moet helemaal leeg zijn (i.e.: geen enkel proces met hoge prioriteit is actief en/of stand-by) wil de andere lijst aangesproken worden. Processen met hoge prioriteit kunnen slechts tijdens het uitvoeren van bepaalde instructies onderbroken worden door de timer. Dat zijn alleen de instructies "jump", "loop end" en "end process". Processen met een lagere prioriteit kunnen, behalve door de timer, door processen met een hogere prioriteit afgebroken worden na iedere willekeurige instructie. Er zijn er een aantal gereserveerde geheugenplaatsen waarin de inhoud van de stackregisters gesaved wordt bij een onderbreking door een priority-0 process. Zodra de priority-0 lijst weer leeg is wordt de stack weer hersteld en gaat het tot dan toe actieve proces weer verder.

Dit alles heeft grote gevolgen. Vooral als het gaat om snelheid: het gehele omschakelproces duurt nog geen micro-seconde...



Figuur 1: Pin-layout van de INMOS T414

De harde waren

Er zijn nog een hele hoop andere interessante dingen aan de hand in dat dure zwarte doosje. Misschien reden genoeg de gespecialiseerde vakliteratuur er op na te slaan? In ieder geval genoeg om nog een aardig tijdje mee zoet te zijn...

Veel aardiger is het op dit moment misschien om eens te kijken naar de uiterlijke kenmerken van de T414. Daarom bij deze meteen ook maar de pin-layout van de transputer en een klein overzicht van de verschillende signalen.

Voeding en algemene signalen

Naam	Betekenis	I/O
Vcc	Voeding, +5V	
GND	System ground	
CapMinus	Ontkoppeling voor PLL (negatief)	
CapPlus	Ontkoppeling voor PLL (positief)	
DoNotWire	Njente, nix mee doen!	
HoldToGND	Deze mot aan de GND	

Algemene systeem aansluitingen

Naam	Betekenis	I/O
ClockIn	5 MHz. standaard klok signaal	In
Reset	Reset aansluiting	In
Error	He, er is een foutje opgetreden!	Out
Analyse	zet 'm inhardware analyse mode	In
BootFromROM	H = Boot van ROM L = Boot van Link #0	In
ProcClockOut	Intern kloksignaal	Out

Interrupt aansluitingen

Naam	Betekenis	I/O
EventReq	Aanvraag onderbreking	In
EventAck	Onderbreking is actief	Out

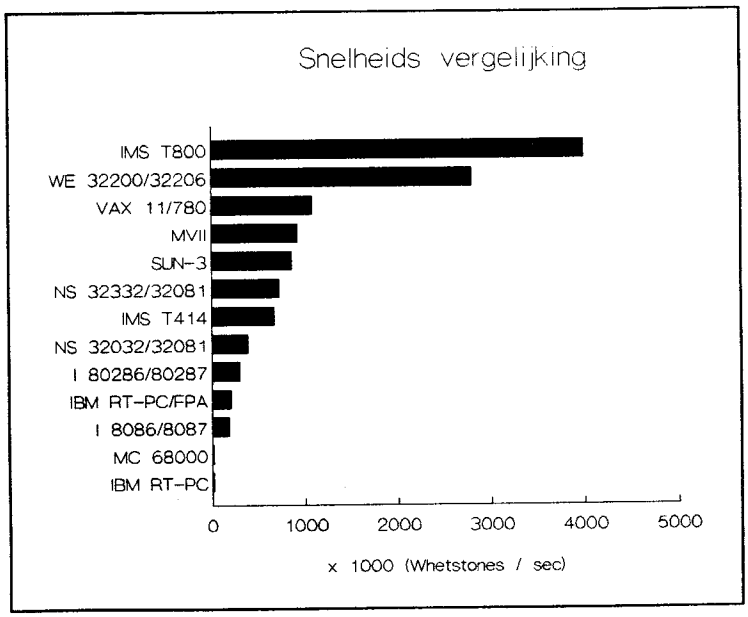
Link aansluitingen

Naam	Betekenis	I/O
LinkIn0-In3	Data in	In
LinkOut0-Out3	Data out	Out
Link123Special	Snelheid links 1 t/m 3	In
Link0special	Snelheid link #0	In
LinkSpecial	Snelheid alle links	In

Geheugen aansluitingen

Naam	Betekenis	I/O
MemNotWrD0	DataBit 0 en schrijf-cycle waarschuwing	I/O
MemNotRefD1	DataBit 1 en refreshcycle waarschuwing	I/O
MemAD02-31	Data- en adres bus	I/O
NotMemRd	Read from memory	Out
NotMemRf	Refresh memory	Out
NotMemS0-S4	Algemene mem signalen	Out
NotMemWrB0-B3	Adressignalen voor byte-writes	Out
MemWait	Expand memory cycle	In
MemGranted	DMA toegestaan	Out
MemReq	Aanvraag voor DMA	In
MemConfig	Geheugen configuratie	In

De meeste signalen zijn dus vrij recht-toe recht-aan. De enige vermeldenswaardige signalen zijn wellicht nog de error- en analyse signalen. Treedt er namelijk ergens in de transputer een fatale fout op, dan gaat het error-signaal hoog en kan eventuele externe hardware de gehele transputer stilleggen. Via de "Analyse" aansluiting kan de transputer dan in een



Figuur 2: Snelheid t.o.v. andere systemen

speciale hardware-analyse mode gezet worden waardoor externe debugging sterk vereenvoudigd wordt.

Snelheidsmaniak

De transputer biedt al met al een goedkoop alternatief voor heel zware rekensystemen. Met name in CAD-toepassingen kan door toepassing van transputers veel snelheidswinst geboekt worden. En dat vaak ook nog tegen lagere kosten dan een gemiddeld CAD-systeem met "standaard" technologie. Dat de transputer inderdaad een snelheidmonster is mag wel blijken uit volgende snelheid vergelijkingsdiagrammetje...

Daar ik zelf (bijna) geen vergelijkingsmateriaal heb, heb ik hiervoor de cijfers van het Duitstalige blad "MC" overgenomen. De vergelijking is gebaseerd op het aantal "Whetstones per second". De preciese omschrijving van de snelheidsvergelijking vallen buiten de doelstellingen van dit artikel en zal ik dientengevolge niet nader toelichten.

Wel, dit besluit de serie over dit bitverslindende monster. Hoewel de transputer nog niet zo oud is, heeft hij in korte tijd al heel wat concurrentie gekregen. Tot nog toe is de prijsstelling van de transputer echter van dien aard dat de concurrentie nog niet serieus te nemen is. Vanuit technisch oogpunt heeft de transputer een aantal eigenschappen die hem een andere kant op stuwen dan zijn concurrenten. De toekomst van de transputer blijft voorlopig dus nog wel even de kleur en geur van (echte) rozen houden...

J.Voorhaar

Methoden en technieken voor datacommunicatie (Deel 2)

Inleiding

In het vorige nummer van de μP Kenner heb ik het gehad over hoe een computer met randapparatuur kan communiceren. Ik heb het gehad over het verschil tussen parallelle overdracht van gegevens en over seriële overdracht. Verder heb ik aangegeven wat de afstanden zijn die men kan overbruggen met behulp van parallelle verbindingen (enkele meters), met een seriële verbinding volgens de RS-232 standaard (enkele tientallen meters) en met een current loop verbinding (enkele honderden meters).

Voor communicatie met bijvoorbeeld met een bulletin board zijn de bovengenoemde afstanden natuurlijk nog veel te klein. Een bulletin board staat meestal minstens enkele kilometers weg en nog beter zou zijn als je met alle bulletin boards in Nederland kunt communiceren. In dat geval voldoen de in de vorige aflevering beschreven technieken in het geheel niet en moeten we gebruik gaan maken van de diensten van een telefoonmaatschappij (in Nederland dus PTT Telecom). In deze aflevering zal ik het gaan hebben over hoe een computer aangesloten wordt op het openbare telefoonnet en hoe het één en ander verder in zijn werk gaat.

Het telefoonnet

Het openbare telefoonnet is hiërarchisch opgebouwd. In deze paragraaf wil ik in grote lijnen aangeven hoe de opbouw van het telefoonnet is. Deze beschrijving is gebaseerd op informatie in ref. 1 waarin het geheel nog veel gedetailleerder beschreven staat.

De telefoonnetten over de hele wereld zijn aangesloten op een internationaal net. Hiervoor staat er in elk land minstens één internationale centrale. Binnen een land, zoals bijvoorbeeld Nederland, hebben we te maken met het interlokale net en diverse lokale netten. Elke abonnee in Nederland is aangesloten op een centrale in zijn eigen lokale net. Ook in de structuur van de telefoonnummers is deze hiërarchie terug te vinden. Iedereen heeft binnen zijn lokale net een abonneenummer. Dit is het nummer van de aansluiting in het lokale net waartoe de aansluiting behoort. Verder heeft elk lokaal net in Nederland een nummer. Dit zijn nummers van twee cijfers (bijvoorbeeld 20 voor Amsterdam) of nummers van vier cijfers (bijvoorbeeld 1720 voor Alphen a.d. Rijn). Om de lokale centrale duidelijk te maken dat er een nummer van een ander lokaal net gekozen gaat worden moet hiervoor een 0 gedraaid worden. De lokale centrale weet dan dat hij contact moet gaan zoeken met het interlokale net. Hetzelfde geldt voor de toegang tot het internationale net. Binnen het internationale net heeft elk land zijn eigen nummer. Voor Nederland is dat nummer 31. Om de interlokale centrale duidelijk te maken dat er toegang gezocht wordt, moet er voor het landnummer en na de 0 een 9 gedraaid worden.

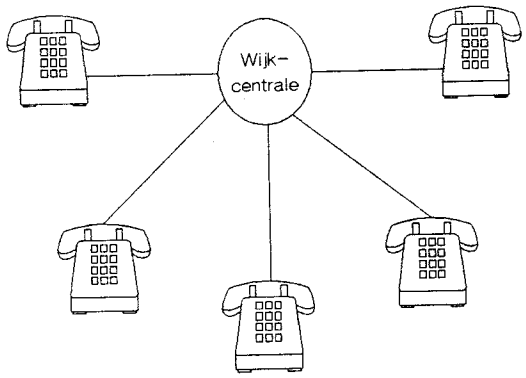


Fig. 1: Telefoonaansluitingen aan een wijk- of eind centrale

In de volgende beschrijving zal de structuur van het lokale en het interlokale net verder uitgediept worden. We beginnen daarbij met het lokale net.

Stel, u woont in een middelgrote plaats en gaat telefoneren met uw buurman. Het telefoontoestel van u en uw buurman zijn beide aangesloten op dezelfde centrale. In figuur 1 is deze situatie getekend. Naar elke telefoonaansluiting in het verzorgingsgebied van een telefooncentrale loopt, vanaf de centrale een lijn. Als u de hoorn van uw toestel opneemt, krijgt u contact met de centrale in uw wijk. Nadat u het nummer van uw buurman gedraaid heeft en uw buurman zijn telefoon opgenomen heeft, worden de lijnen tussen de centrale en u en tussen de centrale en uw buurman met elkaar verbonden. U kunt dan het telefoongesprek voeren. In de figuur wordt gesproken van wijkcentrales in het geval dat er sprake is van slechts één wijk is, wordt gesproken van een eindcentrale. In beide gevallen spreken we van nummercentrales. De telefoonaansluitingen aan een wijk- of eindcentrale moeten binnen een straal van ongeveer 5 km rond de centrale liggen. De situatie met wijkcentrales zal zich dus voordoen bij grotere plaatsen, die van eindcentrales bij de kleinere plaatsen. Zoals uit de figuur wel duidelijk wordt, hebben we in dit geval te maken met een stervormig netwerk. Vanuit de wijkcentrale lopen er stervormig verbindingen naar alle telefoonaansluitingen in de wijk.

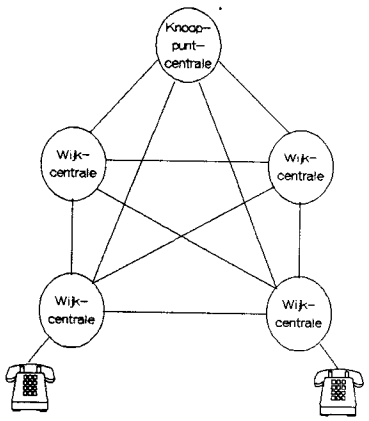


Fig. 2: Wijkcentrales binnen een knooppunt

Wilt u een gesprek voeren met iemand die in een andere wijk van dezelfde plaats woont, dan krijgen we de situatie uit figuur 2. We spreken hier nog steeds over lokaal telefoonverkeer, dus telefoonverkeer waarbij geen netnummer hoeft te worden gekozen. Omdat het gebied te groot is voor één eindcentrale, hebben we te maken met meerdere wijkcentrales. In de figuur is elke wijkcentrale met elke andere wijkcentrale verbonden. Dit is een zogenaamd maasvormig netwerk. In de praktijk hoeft dit niet altijd het geval te zijn. Het is ook mogelijk de wijkcentrales stervormig met een zogenaamde hoofd-wijkcentrale te verbinden waarbij deze dan voor de verbindingen tussen de wijkcentrales zorgt. In de praktijk zal er vaak sprake zijn van een mengvorm. In dat geval liggen er verbindingen tussen een aantal wijkcentrales en lopen een aantal verbindingen via de hoofdwijkcentrale die weer met alle wijkcentrales verbonden is. Figuur 3 is hiervan een voorbeeld.

De bovenstaande beschrijving geldt voor lokaal telefoonverkeer. Om interlokaal verkeer mogelijk te maken, zijn de eind- of wijkcentrales verbonden met de zogenaamde knooppuntcentrales. Een knooppuntcentrale bedient een sector in een district. Binnen een district zijn de lokale netten stervormig met de knooppuntcentrale verbonden. Hierbij kan een knooppuntcentrale maximaal 10 lokale netten in een straal van ongeveer 15 kilometer rond de centrale bedienen. De knooppuntcentrales zijn zelf weer stervormig met de districtscentrales verbonden. Elke districtscentrale kan maximaal 10 knooppuntcentrales in een straal van zo'n 50 kilometer rond de centrale bedienen. Om telefoonverkeer tussen de districten mogelijk te maken zijn de districten in een maasvormig netwerk met elkaar verbonden.

In figuur 4 is het geheel nog eens schematisch weergegeven. We zien hier twee lokale netten, één met een eindcentrale en één met een aantal wijkcentrales die maasvormig met elkaar verbonden zijn. Zowel op de wijkcentrales als op de eindcentrale zijn een aantal aansluitingen aanwezig. De lokale netten zijn gekoppeld aan knooppuntcentrales. Elke knooppuntcentrale bedient een aantal lokale netten. De knooppuntcentrales zijn zelf weer stervormig met districtscentrales verbonden. De districtscentrales zitten in een maasvormig netwerk en zijn stervormig met de internationale centrale verbonden.

Nu is het in de praktijk niet zo dat altijd de structuur zoals in figuur 4 getekend is gebruikt wordt. Er kunnen bijvoorbeeld dwarsverbindingen tussen knooppuntcentrales aanwezig zijn. Voor het begrip van het geheel is dit echter niet van belang.

Om een indruk te geven van de omvang van het geheel, geef ik de situatie zoals die in Nederland ongeveer aanwezig is. Nederland is opgedeeld in 22 districten, elk voorzien van een districtscentrale. Binnen een district kunnen maximaal 10 sectoren aanwezig zijn waarbij elke sector bedient wordt door een knooppuntcentrale. Per knooppuntcentrale kunnen maximaal 10 lokale netten aangesloten worden die dus bestaan uit een eindcentrale of wijkcentrales, al dan niet voorzien van hoofd-wijkcentrales. In Nederland zijn er ongeveer 1100 lokale netten.

In de bovenstaande beschrijving worden de centrales waarop de telefoonaansluitingen aanwezig zijn nummercentrales genoemd. De centrales waarop geen telefoonaansluitingen aanwezig zijn, heten verkeerscentrales, zij regelen het telefoonverkeer tussen de centrales. Verder loopt er steeds één lijn van de nummercentrale naar de abonnee. Tussen de nummercentrale en de centrales onderling lopen er natuurlijk meerdere lijnen. Dit worden bundels genoemd. Met name in de communicatie tussen de

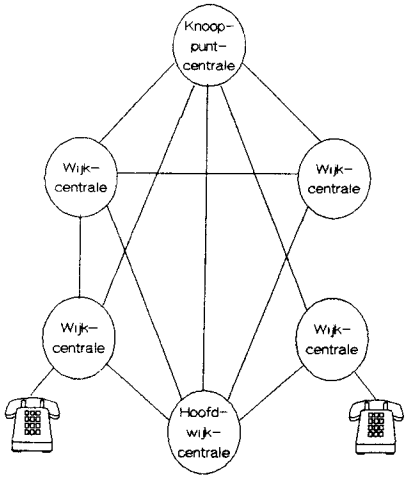


Fig. 3: Mengvorm binnen een lokaal net.

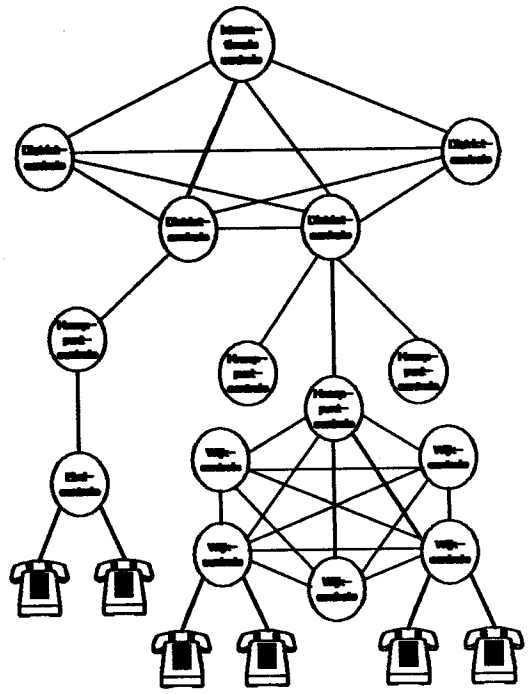


Fig. 4: Overzicht van een nationaal telefoonnet

centrales verandert er de laatste jaren zeer veel. Werd er vroeger voor elke lijn in een bundel een aantal (vier) aparte koperdraadjes gebruikt, tegenwoordig wordt er uitgebreid gebruik gemaakt van multiplexing en glasvezel waarbij door een "draadje" met een dikte van een haar tientallen telefoongesprekken gelijktijdig gevoerd kunnen worden. In het internationale netwerk worden de verbindingen tegenwoordig vaak via satellieten gelegd.

Er zijn misschien mensen die het interessant vinden te weten waar de kosten geregistreerd worden. Welnu voor een lokale verbinding wordt dat door de wijk- of eindcentrale gedaan. Dat is namelijk de enige centrale die bij de telefoonverbinding betrokken is. Voor de interlokale gesprekken wordt dat gedaan door de knooppuntcentrale van degene die de verbinding aanvraagt.

Werking van de telefoon

Bijna iedere abonnee is via twee draadjes aangesloten op de nummercentrale. In de aansluiting zijn dat altijd een rood draadje die "a" genoemd wordt en een blauw draadje die "b" wordt genoemd. Verder is er een blanke aarddraad aangesloten. Om een afscherming tegen storingen van buiten af te krijgen, worden de "a" en "b" draad om elkaar heen gedraaid (twisted pairs).

Als we de telefoon bekijken, dan bestaat die uit een aantal elementen, te weten:

- Een microfoon
- Een luidsprekertje
- Een bel
- Een schakelaar die door de hoorn wordt bediend
- Een draaischijf of een aantal druktoetsen

De microfoon in een telefoon is van een goedkoop soort. Het betreft een zogenaamde koolmicrofoon waarbij de luchttrillingen een membraan in beweging zetten die een hoeveelheid koolstof samendrukt. Op die manier verandert de weerstand van de microfoon en door een spanningsbron op de microfoon aan te sluiten verandert de stroom in het circuit ongeveer gelijk met de trillingen van de lucht. We kunnen hiermee dus een geluidstrilling in de lucht omzetten in een variatie van stroomsterkte. De voor de microfoon benodigde gelijkspanning wordt door de centrale geleverd. Dit is een spanning van tussen de 48 en 60 volt. De stroom die door een telefoon loopt ligt in de orde van 30 milliampere.

In figuur 5 is getekend hoe een telefoon aangesloten is op de centrale. Als de hoorn van de haak ligt, loopt er stroom door de microfoon en de luidspreker van de telefoon. In de centrale wordt het signaal doorgestuurd naar de centrale van de andere partij waar de wisselstroom ook door de microfoon en luidspreker gestuurd wordt. Er loopt dus een wisselstroom door de microfoon en luidspreker van de ene partij en dezelfde wisselstroom loopt door de microfoon en luidspreker van de andere partij. Deze wisselstroom wordt veroorzaakt door variaties in de weerstand van de microfoons die zich in de telefoons bevinden en verder zorgt deze wisselstroom ervoor dat beide luidsprekers het opgevangen geluid weergeven. Als de hoorn op de haak ligt, kan er geen gelijkstroom door de telefoon lopen. Het opnemen van de hoorn wordt door de centrale gedetecteerd aan de hand van het feit dat er stroom door de telefoon en dus de lus gaat lopen. Als er nu een iemand de getekende aansluiting opbelt, dan wordt er

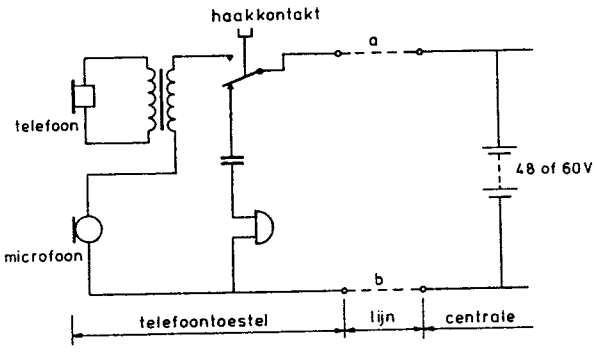


Fig. 5: Telefoon, aangesloten op een centrale

in de centrale periodiek een wisselspanning van ongeveer 75 Volt en een frequentie van 25 Hertz op de telefoon aangesloten. De bel in de telefoon gaat dan bellen. Neemt de abonnee de hoorn van de haak, dan gaat er een gelijkstroom door de telefoon lopen waardoor de centrale weet dat de hoorn opgenomen is zodat de telefoons met elkaar in verbinding gebracht kunnen worden. Als degene die opbelt nu zijn hoorn weer op de haak legt, dan detecteert de centrale dat en wordt de verbinding verbroken.

Voor het kiezen van een nummer worden twee verschillende methoden gebruikt. In de eerste plaats is dat het zogenaamde puls-kiezen. Deze methode wordt toegepast door de oudere kiesschijftelefoons. Alle centrales in Nederland ondersteunen deze manier van kiezen. In de tweede plaats hebben we het zogenaamde toonkiezen. Dit wordt toegepast door de meeste drukknoptelefoons en kan alleen gebruikt worden bij de modernere computergestuurde centrales.

Het pulskiezen gaat als volgt in zijn werk. De kiesschijf bedient een schakelaar waarmee de lijnen a en b kortgesloten worden. Dit kortsluiten gebeurt tien keer per seconde, dus tien pulsen per seconde. Het aantal pulsen dat gegeven wordt is evengroot als het cijfer dat gekozen is waarbij het cijfer 0 tien pulsen geeft. Gedurende het genereren van de pulsen wordt de hoorn ook kortgesloten zodat de pulsen niet in de luidspreker te horen zijn. Uiteraard worden deze pulsen in de centrale gedecodeerd. Bij het toonkiezen wordt er door de telefoon voor elke toets een combinatie van twee toontjes op de lijn gezet. De frequentie van deze toontjes en de toetsen waarmee de frequenties corresponderen zijn internationaal vastgelegd. In tabel 1 is een overzicht van deze frequenties gegeven. Voor het cijfer 5 worden dus de frequenties 770 Hz en 1336 Hz gelijktijdig op de lijn gezet.

Hz	1209	1336	1477	1633
697	1	2	3	
770	4	5	6	
852	7	8	9	
941	*	0	#	

Tabel 1: Toonkeuze tabel

Behalve dat de telefoon signalen (pulsen of tonen) aan de centrale doorgeeft, geeft de centrale ook signalen aan de telefoon door. In de eerste plaats is dat de kiestoon. Dit is een ononderbroken toon die gegenereerd wordt als de hoorn van de haak genomen wordt. In de tweede plaats is er de wektoon. Deze toont hoort men als de bel bij degene die gebeld wordt overgaat. Dit is een toon die met lange tussenpozen onderbroken wordt. In de derde plaats hebben we de bezettoon. Deze toon is hoger als de wektoon en wordt met kortere tussenpozen onderbroken. Tenslotte hebben we ook nog de informatietoon. Deze toon krijgt men als de abonnee (tijdelijk) afgesloten is of als het nummer gewijzigd is.

Een modem aan het telefoonnet

In principe zijn er twee manieren om een modem aan het telefoonnet te koppelen. In de eerste plaats kan dit door middel van een acoustische koppeling. In dat geval wordt de hoorn van de telefoon, na het tot stand komen van de verbinding, op het modem gelegd waarna de signalen via luidsprekers en microfoon (acoustisch dus) uitgewisseld worden. Deze methode wordt in Nederland haast niet meer gebruikt.

In de tweede plaats kennen we de directe koppeling. In dat geval wordt in plaats van de stekker van de telefoon de stekker van het modem in het stopcontact gestoken. De telefoon kan men dan weer bovenop de stekker van het modem steken. Het modem heeft dan dus rechtstreekse toegang tot de draden van het telefoonnet. Dit mag alleen als het modem een goedkeuring van de PTT heeft doch de meeste modems hebben tegenwoordig die goedkeuring. Als het modem niet in bedrijf is, dan is de telefoonlijn op de normale manier verbonden met het telefoontoestel. Is het modem in bedrijf, dan wordt deze verbinding door middel van een relais verbroken en wordt de telefoonlijn aangesloten op het modem. In rust is wel de oproepdetector van het modem op de telefoonlijn aangesloten, parallel aan de bel van de telefoon. Deze oproepdetector detecteert of de wisselspanning van 75 Volt op de lijn aanwezig is (belspanning) en zal, als het modem zo geconfigureerd is, het modem op de bel laten reageren. In figuur 6 is getekend hoe een telefoon en een modem gezamenlijk op het telefoonnet zijn aangesloten.

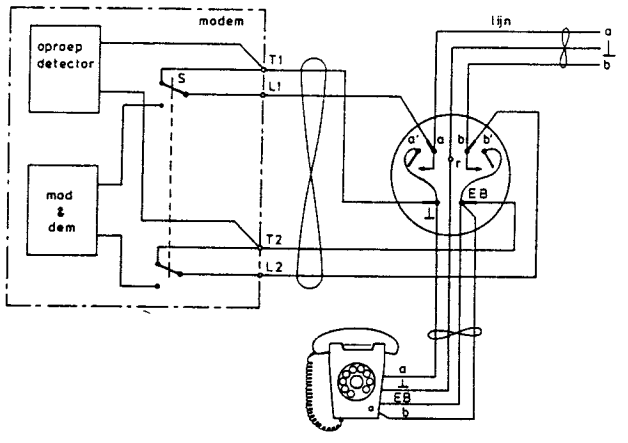


Fig. 6: Telefoon en modem aangesloten op het telefoonnet

Behalve dat een modem aan het telefoonnet gekoppeld moet worden is het natuurlijk zinvol de modem ook met de computer te verbinden. In de meeste gevallen worden zogenaamde externe modems gebruikt. In dat geval wordt de verbinding tussen modem en computer tot stand gebracht door middel van een seriële RS-232 verbinding. Deze staat in de μP Kenner nummer 65 beschreven. Bij interne modems gaat het ongeveer net zo, alleen is het modem in dat geval samengebouwd met een seriële poort en wordt het geheel in een uitbreidingsslot van de computer gestoken.

Signalen

Goed, we hebben het modem met het telefoonnet verbonden en er ligt een verbinding tussen computer en modem (of u hebt een intern modem). Laten we nu eens kijken wat er allemaal gebeurt als we contact zoeken en krijgen met een ander modem. Van de twee modems moet er één zodanig ingesteld staan dat het modem de telefoon opneemt. Dit modem moet in de zogenaamde "ANSWER" -mode staan. Bij de meeste moderne modems kan aangegeven worden hoe vaak de telefoon over moet gaan voordat het modem de telefoon opneemt. De andere modem is de opbeller en die staat dan in de zogenaamde "ORIGINATE" -mode. In de volgende aflevering en op mijn voordracht in Almere zullen we zien dat behalve voor het al dan niet opnemen van de telefoon er ook nog andere verschillen tussen originate en answer zijn. Als de modems in rust zijn, dan staan ze in "SPRAAK" of "TALK" -mode hetgeen wil zeggen dat de telefoon normaal in bedrijf is. Is het modem zelf actief met de lijn bezig, dan staat het modem in "DATA" -mode. Deze laatste mode wordt ook wel "ONLINE" genoemd. Soms wordt door middel van een lampje op het modem aangegeven of het modem in data of talk staat. Vrijwel alle modems geven door middel van een lampje aan dat het modem in aswer-mode, ook wel auto answer genoemd, staat.

Als we contact willen leggen met een modem in answer-mode, dan kan dat op twee manieren. In de eerste plaats kan met behulp van de telefoon het nummer gekozen worden waarna, als het modem opgenomen heeft, we door middel van een knop op het modem (de talk- data-schakelaar) of vanaf de computer met een opdracht aan het modem de verbinding tussen de modems tot stand gebracht wordt. Tegenwoordig kan het echter ook veel gemakkelijker. De meeste modems ondersteunen tegenwoordig namelijk de zogenaamde auto-dial. In dat geval kan de computer het modem de opdracht geven een nummer te kiezen. Als het modem een dergelijke opdracht krijgt, dan schakelt hij eerst zichzelf op de telefoonlijn (data) en doet dan de kiesschijf of de druktoetsen van de telefoon na. Het modem zal zelfs netjes op de kiestoon wachten en vaak ook aangeven als het gekozen nummer in gesprek is. Op deze manier kunnen dus vanuit een communicatiepakket geheel automatisch verbindingen met bulletin boards gelegd worden waarbij de computer bij "in gesprek" het steeds weer opnieuw probeert.

Goed als de opbeller het nummer gekozen heeft, dan zal de antwoorder detecteren dat de bel van de telefoon over gaat. Via pen 22, de Ring Indicator kan dit ook aan de computer doorgegeven worden. Verder is het ook mogelijk dat het modem de kreet "RING" naar de computer doorstuurt bij elke keer als de bel over gaat. Al na gelang de instelling in het modem zal het modem na één of meer keren bellen de telefoon opnemen. Het antwoordende modem schakelt zichzelf daarvoor op de lijn en sluit dus de bijbehorende telefoon af. Vervolgens zet het antwoordende modem een aantal tonen op de lijn en luistert gelijktijdig of de opbeller hierop reageert. De betekenis van deze tonen zal ik in de volgende aflevering uitleggen. Als de modems elkaar begrijpen is de verbinding tot stand gebracht. Dit wordt aangegeven door het signaal Carrier Detect of Data Carrier Detect die is aangesloten op pen 8 in de RS-232 standaard. Verder wordt dit bijna altijd ook door een lampje op het modem aangegeven. Tenslotte laten bij bepaalde modems de modems de computers weten dat ze verbinding hebben door middel van het zenden van het woord CONNECT of CONNECT xxxx naar de computer waarbij xxxx de transmissiesnelheid in bits per seconde is. Nadat dit allemaal gebeurd is, kan met de uitwisseling van gegevens begonnen worden. De informatie wordt via de TxD (pen 2) naar het modem gestuurd die de data omzet in geluid en over de telefoonlijn naar de andere computer stuurt. Via RxD (pen 3) ontvangt een computer informatie die door het modem van de telefoonlijn opgepikt is. Aan beide kanten ontvangt de computer dus over pen 3 en zendt de computer over pen 2. Zolang er verbinding is, is er geluid op de lijn aanwezig. Als dit geluid (de carrier) wegvalt, dan is de verbinding verbroken, de modems leggen de

Als de modems elkaar begrijpen is de verbinding tot stand gebracht. Dit wordt aangegeven door het signaal Carrier Detect of Data Carrier Detect die is aangesloten op pen 8 in de RS-232 standaard.

hoorn op de haak en schakelen terug naar hun normale (talk-) toestand.

Behalve de bovengenoemde signalen, zijn er in de RS-232 standaard nog een paar andere. In de eerste plaats is dat de protective ground (pen 1) die in figuur 4 van de vorige aflevering ten onrechte signal ground heet. Dit is de massa van de apparaten en kan voor afscherming van de kabel tussen modem en computer gebruikt worden. Om storingsredenen mag dit signaal slechts aan één van de twee kanten (modem of computer) aangesloten worden. In de tweede plaats hebben we het signaal request to send (pen 4). Met dit signaal laat de computer het modem weten dat hij iets wil gaan zenden. De computer vraagt aan het modem of zenden toegestaan is en als dat het geval is, dan geeft het modem door middel van het signaal clear to send (pen 5) aan dat de computer mag zenden. Deze twee laatste signalen worden alleen gebruikt bij half-duplex verbindingen. Bij full-duplex verbindingen zijn de signalen bijna altijd hoog. Eventueel kan het clear to send signaal ook gebruikt worden als het modem sneller gegevens van de computer krijgt dan hij over de telefoonlijn kwijt kan. De bitrate van de computer naar het modem is dan hoger dan de bitrate tussen de modems onderling. In dat geval kan het modem dan aan de computer kenbaar maken dat zijn interne buffer vol is en dat de computer zich even koest moet houden. In de praktijk gebeurt dit, geheel automatisch, in de hardware van de computer. Behalve de signal ground (pen 7) die de nul voor alle signalen is, hebben we tenslotte nog de signalen data set ready (pen 6) en data terminal ready (pen 20). Met behulp van data terminal ready laat de computer het modem weten dat hij aan staat en verbinding heeft met het modem, met behulp van data set ready laat het modem weten dat hij aan staat.

Afsluiting

In deze aflevering hebben we het nauwelijks gehad over de computer en maar zeer gedeeltelijk over het modem. Hoewel we het steeds gehad hebben over het telefoonnet heeft dit toch met computers te maken. Niet alleen wordt het telefoonnet vaak gebruikt om informatie tussen computers uit te wisselen, ook computernetten hebben vaak een vergelijkbare structuur. Als voorbeeld noem ik het internationale net waarmee een belangrijk deel van de bulletin boards verbonden zijn. Dit is ook opgesplitst in lokale, interlokale en internationale netten. In plaats van centrales staan er dan op de knooppunten computers die ervoor zorgen dat de informatie op de juiste manier van de afzender naar de geadresseerde komt.

Als we op ons club bulletin board kijken, dan hebben we in het berichtengebied in de eerste plaats de lokale berichten. Deze blijven bij Jacques op het systeem en ieder die ze wil hebben, moet ze bij hem weghalen. In de tweede plaats hebben we de zogenaamde echo-mail. Berichten die in dit gebied gezet worden, worden over het interlokale net (in ons geval het PCC-net) doorgestuurd naar andere bulletin boards. Tenslotte is het mogelijk internationaal berichten uit te wisselen doch in tegenstelling to lokale en echo-mail is dat niet gratis en staat dus niet ter beschikking van iedereen.

In de volgende aflevering wil ik het gaan hebben over de manier waarop de informatie over de telefoonlijn gaat. We gaan dan praten over modulatietechnieken, V21, V22, V23 enzovoort. Dat zal een zeer technisch verhaal worden waarin geprobeerd wordt duidelijk te maken wat er aan de telefoonkant van het modem gebeurt. Op de datacom-dag op onze clubbijeenkomst in Almere (zie uitnodiging) zal ik deze zaken ook gaan behandelen waarbij ik bovendien zal proberen het één en ander te demonstreren en te laten zien op een oscilloscoop.

Literatuur

1: P.C. den Heijer, R. Tolsma: Data-communicatie; Kluwer PC Boeken.

Volgende keer in de µP Kenner

- Intel's nieuwste
- Factorials: breuken zonder verlies van significante cijfers
- Assembler op de PC
- Debug: antiek maar leuk speelgoed
- Weet u nog wat "EDLIN" betekent?

Avonduren

De laatste tijd hoor ik steeds vaker tijdens mijn lunchpauzes verhalen over enge draken, vreemde wapens en heldendaden. Het vreemde is, dat de meeste verhalen zich in de ik-vorm afspelen. Zo ving ik vorige week het volgende gesprek op:

"Goh, ik wordt iedere keer gevangen door die menseneter en dan kom ik nooit op tijd uit die kooi!"

"Och, simpel, dan moet je wachten tot de wachter er aankomt en dan de giftige paddestoel gooien. Dan kun je zo bij de sleutels."

"Wat voor paddestoel?"

"Nou, als je net vrij gekomen bent van die wachters van het ruimteschip, dan kom je bij zo'n doolhof. En daar ligt ook een paddestoel."

"Oh, die heb ik helemaal niet?"

"Nou, en dan moet je je inwrijven met die bessen dus en dan wordt je niet door de krokodillen verslonden."

"Ja, dat goedje stinkt wel hard he."

"Ja. En dan via die boom over het ravijn heen en dan trap je in zo'n val van die menseneter."

"Ja, zover was ik al, maar ik had die paddestoel dus niet."

"Die heb je dus wel nodig. En dan kun je bij die sleutels en dan doe je je hok open en dan loop je zo maar weg. Je komt dan op het strand, maar daar moet je niet blijven want dan wordt je hartstikke doodgeschoten."

Het verhaal ging nog verder, maar inmiddels was Joke, de knappe secretaresse van de directeur, bij ons aan tafel komen zitten en probeerde een gesprek met mij aan te knopen over het concert waar ze de dag tevoren geweest was.

Om vijf uur, toen ik wegging van mijn werk, vroeg ik één van de betreffende collega's of hij nog mee ging naar het Leidseplein om nog snel even een biertje te pakken voor het avondeten. Maar nee, hij moest snel naar huis, vertelde hij, want "hij wilde zijn avonduren met andere avonduren vullen". Pas later begreep ik dat hij zijn avonduren met avonturen wilde vullen.

Diep weggedoken achter zijn PC en met rode oogjes naar het beeldscherm starend vond ik hem 's avonds terug. Op het scherm zag ik een mannetje op een soort ruimtestation druk bezig met bezemen. Allerlei flauwe teksten vlogen over het beeldscherm en voor ik het wist hadden we al 30 punten verzameld, hadden een buil opgelopen door een klap met een stomp voorwerp en waren op weg naar een ander ruimteschip. In de tussentijd werd mij snel uitgelegd dat dit nu de nieuwe rage was. Van "Larry" had ik vast wel eens gehoord, wel, dit was van de zelfde makers. "SpaceQuest" heette 't en het spel was net "herontdekt" door één van mijn collegas.

Vanavond heb ik een aantal uren achter het scherm van mijn PC doorgebracht in de ijdele hoop om hun enthousiasme te begrijpen. Maar helaas, zelfs na een stevige flex wiskey en een bijzonder melige film ("Top Secret", een stevige aanrader voor de avond van een grauwe, regenachtige maandag) kon het mannetje met zijn vreemde gewoontes mij niet boeien. Maar ja, het levert wel geinige discussies op...

Met dank aan Maarten Festen, een inspirerende exhuisgenoot.

Joost Voorhaar

Ik heb interesse in de KGN en wil

☐ Lid worden van de KGN ☐ Meer informatie over de KGN

Naam : _____

Adres : _____

Postcode en Woonplaats : _____

Datum : _____ Handtekening: _____

Dit strookje kunt u ingevuld opsturen aan het secretariaat van:

KIM Gebruikersclub Nederland
Postbus 99650
1000 NA Amsterdam

Winchester drives en de DOS65 computer

Het dos65 operating system houdt rekening met een totaal van vier drives. Er worden vier ram pagina's (256 bytes) gereserveerd voor de system sectors van de vier drives ($ad00-b0ff). Deze drives kunnen diskettedrives zijn, de dos65 diskette controller kaart kan er drie aansturen, maar ook een winchester drive of een ramdisk behoren tot de mogelijkheden. De exacte configuratie moet opgegeven worden bij het assembleren van het operating system (de file 'BOOT'). De dos65 versie zoals die standaard verspreid is, is geassembleerd voor max. 2 diskettedrives (5.25 of 3.5 inch) en een ramdisk. Bij versie 2.02 kan deze ramdisk zelfs 1 Mbyte groot zijn. Voor de ramdisk geldt dat de benodigde drivers al in het operating system ingebakken zitten. Ook wordt er een programma vformat meegeleverd, hiermee kan de gehele ramdisk geformatteerd worden in hetzelfde formaat als een floppy. Als er gebruik gemaakt wordt van dynamische rams moet dit elke keer gebeuren als de computer wordt aangezet, bij statische rams met battery backup hoeft dit maar eenmaal gedaan te worden, net als bij een echte floppy. Een ramdisk heeft natuurlijk veel voordelen, zoals snel laden van commando's, snel assembleren etc. maar daarnaast is een ramdisk ook veel kwetsbaarder. Een kleine spanningsstoring kan veel informatie onbruikbaar maken. Bovendien zijn ramchips nog zo duur dat bijv. een 1 Mbyte ramdisk nog aardig in de papieren loopt, zeker als er statische chips gebruikt worden.

Een alternatief is een winchester drive oftewel harddisk. Ook niet echt goedkoop maar minder kwetsbaar. Verder komen er steeds meer harddisk drives beschikbaar die niet meer voldoen aan de huidige standaard van 20 Mbytes en hoger die in de PC wereld geldt. De kleinere drives zijn echter bij uitstek geschikt voor een computer als de dos65 computer omdat de programma's vaak zo klein zijn dat er heel wat moeite nodig is om bijvoorbeeld 2.5 Mbyte vol te maken.

Drive types

Er zijn een paar standaards voor harddisk interfaces. Een van die standaards is de st506 interface. Deze door Shugart technology (nu Seagate) op de wereld gebrachte norm gaat uit van een 34 pin cable voor de control signalen en een 20 pin cable voor de data. Er zijn drives met zgn. 'edge' connectors, maar ook met standaard flatcable connectors. De power aansluiting is identiek aan die van 5.25 inch floppy drives, deze kan dus ook gewoon gebruikt worden, mits de max. stroom voldoende hoog is (sommige diskdrives gebruiken minder dan een floppydrive). Met de st506 kon niet snel gezocht worden. Om bijv. van track nul naar track 250 te gaan moesten er 250 pulsen van 3msec. gegeven worden. Elke puls veroorzaakte een stap naar een volgend spoor. Een nieuwere versie was de st412 interface standaard. Bij deze standaard wordt er van uit gegaan dat de 'seek' pulsen heel kort zijn (typisch tussen 12 en 200 usec). De pulsen worden geteld en er wordt door de drive in een keer gesprongen naar het juiste spoor. Verder zijn de st506 en st412 uitwisselbaar, de st412 kan ook gebruik maken van de 3msec seek pulsen.

SASI

Een st506/412 drive heeft een nogal ingewikkelde controller nodig. De data die serieel van de drive komt moet gedecodeerd worden, er wordt meestal gebruik gemaakt van MFM (net als bij dual density floppy's). Hiervoor heeft Western Digital bijv. een chip set op de markt gebracht (WD1010 etc.). Elke computer die wil communiceren met een harddisk met st506/412 interface kan hiertoe gebruik maken van die chipset. Elke fabrikant moet dan een controller board ontwerpen. De firma Xebec bracht een eenvoudiger oplossing. Waarom niet een controller die op de harddisk gemonteerd kan worden, met een zeer eenvoudige interface naar de computer toe, iets wat lijkt op een centronics interface? Deze SASI interface (SASI oorspronkelijk van Shugart) is eenvoudig met de computer te verbinden door middel

2	i/o	data0	tijdens een select (sel low)
4	i/o	data1	moet op de datalijnen het
6	i/o	data2	controller address staan.
8	i/o	data3	data 01 = address 1,
10	i/o	data4	data 02 = address 2
12	i/o	data5	data 04 = address 3
14	i/o	data6	data 08 = address 4 etc.
16	i/o	data7	
18-24		spare	----------
26		reserved	
28-34		spare	
36	in	-busy	SASI response to sel and address
38	out	-ack	computer made another byte available
40	out	-rst	reset the SASI controller
42	in	-msg	low if command completed
44	out	-sel	computer selects controller (address)
46	in	-c/d	SASI command (low) or data
48	in	-req	SASI requests another byte
50	in	-i/o	low is data from controller

Alle oneven nummers aan massa. Voedingsplug van Xebec 1410a identiek aan 5.25 inch floppy voedingsplug.

Tabel 1: SASI Cable layout (50-pin flatcable)

van een paar parallel poorten en een 50 polige flatcable.

Er kunnen maximaal 8 Xebec 1410A controllers worden aangesloten op dezelfde flatcable en maximaal twee (dezelfde) drives per controller. Ook zijn er drives die een ingebouwde SASI interface hebben zoals bijv. de 'OWL' drive (zie elektuur computing).

SCSI

De Amerikaanse 'ANSI', het instituut dat de officiële standaards maakt, doopte de SASI interface om naar SCSI en voegde er het een en ander aan toe. Zo kunnen er bijvoorbeeld ook tapedrives gebruik maken van de SCSI interface en werd de snelheid opgevoerd.

Als SASI cable maar met de volgende wijzigingen:

18	dbp data parity bit
26	+5 Volt
32	atn attention
25	reserved

Tabel 2: SCSI Cable layout (50-pin flatcable)

Er worden door diverse fabrikanten SCSI (spreek uit 'skoezie') drives op de markt gebracht. Een voordeel voor de interface ontwikkelaars, bijna al het werk gebeurt in de SCSI drive.

Oudere drives

Voor de hobbyist zijn de goedkoopste drives meestal het meest interessant. Deze drives hebben soms wat andere onhebbelijkheden die nog niet genoemd zijn. Zo is bijvoorbeeld de schrijf dichtheid aan de binnenzijde van de schijf veel hoger dan aan de buitenzijde omdat op de binnenste kleinere tracks net zoveel informatie moet staan als op de veel langere buitenste tracks. Omdat dit moeilijkheden gaf werken de oudere drives met een gereduceerde schrijfstroom vanaf een bepaald spoornummer. Ook is het mogelijk om de timing te wijzigen vanaf een bepaald track, de zgn. write precompensation. Om een voorbeeld te geven, de seagate st225, niet eens zo oud, heeft geen reduced write current nodig (die zetten we dus op 65535, zoveel tracks heeft de drive niet, dus wordt die optie ook niet gebruikt), maar wel write precompensation vanaf track 300 (die moet dus op 300 gezet worden , = 012C hex). Een ouder type, de seagate st412, vraagt om reduced write current vanaf track 128 en write precompensation vanaf track 128, dit wordt hex 0080-0080.

Embedded servo

Dan zijn er nog verschillen in de manier waarop de drive ziet hoe de head gepositioneerd staat. Sommige drives hebben een apart schijfoppervlak met eigen head, alleen maar voor de zgn. servo informatie. Deze informatie is nodig om bij een 'seek' de servo motor te kunnen sturen en daarmee de head op exact de juiste plaats terecht te laten komen. Veel drives hebben deze informatie 'verborgen' zitten tussen de data sectoren (zgn. embedded servo). Dit geeft aanmerkelijk meer ruimte voor data opslag omdat er een heel oppervlak vrijkomt. Voor de Xebec controller kan met een bit gekozen worden voor een van beide opties. Als vuistregel geldt dat een drive met een oneven aantal heads een aparte servohead heeft (daarom blijft er een oneven aantal heads over) en een drive met een even aantal heads embedded servo gebruikt.

ECC burst

Diskdrives maken soms fouten, er kan een bitje 'omgevallen' zijn bijvoorbeeld. Ook is het mogelijk dat een bit eenmaal fout gelezen wordt en de volgende keer weer correct naar buiten komt. Om datafouten te voorkomen zijn er verschillende methoden in gebruik. Ten eerste bevat de datasector een checksum, de data wordt opgeteld via een eenvoudig algoritme en de som wordt achter de data weggeschreven. Als tijdens het lezen blijkt dat de som niet klopt, dan is de data niet juist gelezen of geschreven. Er bevindt zich zowel een checksum achter de data als achter de sector informatie. Bij elke sector staat het sector en track nummer op de schijf geschreven. Als hier wat fout mee gaat, kun je meldingen verwachten als 'sector not found at ...'. Om te voorkomen dat er bij elk fout bit paniek in de elektronische tent is wordt eerst een aantal maal de leesoperatie herhaald (retry). Als de informatie dan nog steeds niet correct tevoorschijn komt, neemt de controller zijn toevlucht tot de zgn. ECC, een error correcting methode. Als de ECC burst lengte op 11 gezet wordt (normale waarde voor Xebec 1410A) dan betekent dit dat een grote fout waarbij 11 bits achterelkaar niet correct gelezen kunnen worden, nog verbeterd kan worden. Dit wordt bereikt door bij het schrijven extra informatie toe te voegen aan de bestaande data.

Verdere opties

Er is een trend om opslagcapaciteit zoveel mogelijk te vergroten zonder daarvoor meer platen in een drive te hoeven monteren. Een voorbeeld hiervan is de RLL encoding. Op de oude manier, met MFM encoding, is een Seagate st225 in staat om ca. 20 Mbyte op te slaan, met RLL kan een st238, praktisch dezelfde drive, tot ca. 32 Mbyte gevuld worden. Dit wordt bereikt door de plaats van de pulsjes op het spoor wat nauwkeuriger te definiëren. Hierdoor kan

de schrijfdichtheid weer wat groter worden waardoor je meer informatie kwijt kunt. Op een RLL controller kun je vaak nog wel een st225 gebruiken, er zijn echter ook al A(dvanced)RLL en E(xtended)RLL controllers die tot 1.9 x zoveel informatie op een drive persen. Dit stelt hogere eisen aan de drive en daarom zijn alleen speciaal daarvoor ontworpen drives geschikt. Ook is er een steeds doorgaande vraag naar hogere snelheden. Ook hier zijn mogelijkheden, zoals de ESDI interface. Drives, werkend met MFM, maken meestal gebruik van 16 of 17 sectoren per track (512 bytes) of 32 sectoren per track (256 bytes). Voor RLL combinaties geldt een aantal van 26 sectoren van 512 bytes en voor ESDI combinaties ben ik 34 sectoren tegengekomen. De Xebec 1410A controller werkt met 17 sectoren van 512 bytes of 32 sectoren van 256 bytes en altijd MFM.

Dos 65

In het operating system wordt rekening gehouden met een harddisk, maar er moet nog wel wat gebeuren voor deze drive ook werkelijk gebruikt kan worden. Ten eerste moet er een verbinding gemaakt worden met de harddisk. Voor een SASI drive is dat eigenlijk geen probleem. In de duitse 'elektuur computing special 6' werd een printplaat beschreven met een SASI interface (en een floppy interface), verkrijgbaar onder nummer EPS 86354-1. Of deze kaart nog verkrijgbaar is weet ik niet maar de interface is eenvoudig en daarom ook best zelf te maken. Een tweede oplossing is de SASI interface van Ad Brouwer, een eenvoudige schakeling met een 6821. Dan de software. Voor de interface van Ad Brouwer zijn de drivers beschikbaar. Deze drivers nemen twee pagina's in beslag (2 x 256 bytes) en moeten bijvoorbeeld door login.com geladen worden. Voor de elektuur interface vind je deze drivers in dit artikel. Door in mijn systeem de I/O een beetje op te schonen (zie artikel Z80 ramdisk) heb ik van $e200 t/m $e3ff precies twee pagina's ram beschikbaar gekregen die door geen enkel programma gebruikt worden. Hier worden mijn drivers geladen. De drivers blijken zelfs in een pagina te passen, bv. van $e500-e5ff. Verder heb je een dos65 versie nodig waarin rekening wordt gehouden met de winchester drivers. De drivers zien er als volgt uit: een schrijfroutine die een sector naar schijf stuurt en een leesroutine die een sector vanaf schijf haalt. Verder moet de SASI controller weten om wat voor drive het gaat (heads/tracks/precomp). Hiervoor is een kleine initialisatie routine nodig die alleen tijdens het opstarten gebruikt wordt. De eerste keer dat een harddisk gebruikt wordt zal deze geformatteerd moeten worden. Voor de elektuur interface is hiervoor door mij een 'wformat' programma geschreven dat met behulp van een configuratie file heel eenvoudig is aan te passen voor een ander disk drive type. Verder kun je met dit programma de drive testen met diverse bitpatronen en een dos65 systeemsector maken. De drive wordt dan geschikt gemaakt voor 2.5 Mbytes. Dit zijn ca 9999 blocks van 256 bytes. Om hoger te gaan moeten eerst wat grenzen verlegd worden in de diverse dos65 programma's. Een volgende keer meer over dit formatteer programma.

Ernst Elderenbosch.

```
; file                    st225.mac
;
; program       winches
;
; function      load winchester disk drivers
;
; usage                   winches
;
; by                      ernst elderenbosch
;
; original by   ad brouwer
;
; date                    04-jun-89       ernst elderenbosch system
;= = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = =

cpuclock equ    2                       ; 2 MHz 6502

worg            equ     $e200           ; drivers memory $e200-e3ff
```

```
wofs        equ     $0000       ; offset DOS65 directory (partition?)
wsiz        equ     160*128     ; size directory

memp        equ     $e8         ; dos65 r/w memory pointer

prtbyt      equ     $d038       ; dos65 print accu hex routine
prtext      equ     $d03b       ; dos65 print text routine

udc         equ     $e080       ; universal disk controller
rwdsasi     equ     udc         ; read write data from/to hd
rssasi      equ     udc+$10     ; read status from sasi
selsasi     equ     udc+$20     ; select sasi controller
resasi      equ     udc+$30     ; reset sasi controller
;
mask        equ     $d9         ; mask for sasi input
wcd         equ     $80
wreq        equ     $40
wmsg        equ     $10
wbsy        equ     $08
wio         equ     $01

werror      equ     $02         error mask

; winchester controller commands
ctstrdy     equ     $00         test drive ready
ctrck00     equ     $01         recalibrate
csense      equ     $03         request sense status
cread       equ     $08         read sector
cwrite      equ     $0a         write sector
cinitdsk equ $0c                initialize drive characteristics

            org     $200

winches     jmp     winit

wcmdbf      fcb     0,0,0,0,1,$07   command buffer (fast step)
;
; Seagate 225 specification (must be after wcmdbf)
wspec       fdb     615         cylinders
            fcb     4           heads
            fdb     $ff,$ff     reduced write current
            fdb     300         write precompensation
            fcb     11          maximum ECC data burst length
;
;
winit       sta     resasi      ; send reset pulse to controller
resdel      lda     #6          ; 450 ms delay
resdel1     jsr     del75       ;  75 ms delay routine
            sec                 ; decrement nr of delay times
            sbc     #1
            bne     resdel1     ; total delay not yet reached
;
testrdy     jsr     selcntr     ; select the controller
            lda     #ctstrdy    ; test drive ready command ($0)
            jsr     taskout     ; send command to controller
            jsr     getstat     ; get the controller status
```

```
                beq     inidisk
                jmp     error           ; error, drive not ready
;
inidisk         jsr     selcntr         ; select the controller
                lda     #cinitdsk       ; initialize drive characteristics cmd ($c)
                jsr     taskout         ; send command to the controller
                ldx     #0              ; init. charact. byte counter
inidsk1         jsr     reqwait         ; wait for cntlr request
                bit     rssasi
                bmi     inidsk2         ; if command request, xfer completed
                lda     wspec,x         ; get data byte
                sta     rwdsasi         ; send it to the controller
                inx                     ; bump the buffer pointer
                bra     inidsk1         ; more to do
inidsk2         jsr     getstat         ; get completion status
                beq     recal           ; test for error completion
                jmp     error           ; error
;
recal           jsr     selcntr         ; select the controller
                lda     #ctrck00        ; recalibrate command code ($1)
                jsr     taskout         ; send command to controller
                jsr     getstat         ; get completion status
                bne     1.f             ; branch on error
                rts
1               jmp     error
;
taskout         cmp     #2              ; check for forbidden code
                beq     9.f
                sta     wcmdb           ; store command in table
                jmp     cmdout          ; standard routine
9               jmp     error           ; error
;
del75                                   ; 75 msec delay routine
                if      cpuclock = = 4
                ldx     #$a5
                ldy     #$ea
                elseif  cpuclock = = 2
                ldx     #$52
                ldy     #$75
                elseif  cpuclock = = 1
                ldx     #$a9
                ldy     #$3a
                endif
del751          dex
                bne     del751          ; lsb zero? = = decrement msb
                dey
                bne     del751          ; msb zero? = = ready
                rts
;
error           jsr     prtext
                fcc     '\rHarddisk error ',0
                jsr     sense
                fcc     '\r\r',0
                rts
;
sense           jsr     selcntr         ; select the controller
```

```
                lda     #csense         ; request sense status command ($3)
                jsr     taskout
                jsr     del75
                lda     rwdsasi         ; get sense byte
                sta     sense0
                and     #$3f            ; extract error code
                jsr     prtbyt
                jsr     prtext
                fcc     ' sense bytes: ',0
                lda     rwdsasi         ; get sense byte
                sta     sense0+1
                jsr     prtbyt
                lda     rwdsasi         ; get sense byte
                sta     sense0+2
                jsr     prtbyt
                lda     rwdsasi         ; get sense byte
                sta     sense0+3
                jsr     prtbyt
;
sense0          res     4,0             ; buffer for sense info
;
;
;+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
;               winchester driver (read write sector)
;+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

                org     worg

winread         jmp     wread
winwrite jmp    wwrite
woff            fdb     wofs            offset address
wmax            fdb     wofs+wsiz       max address

wcmdb           fcb     0,0,0,0,1,$18   command buffer (fast step)

ytemp           fcb     0

;read sector from winchester drive

wread           lda     #cread          ; get read sector command
                jsr     wrwsec          ; start read sector procedure
                bcs     9.f             ; carry set = error
                ldy     #0
read1           jsr     reqwait
                bit     rssasi
                bmi     read2
                lda     rwdsasi
                sta     [memp],y
                iny
                bra     read1
read2           jsr     getstat
                bne     4.f
                rts
;
;write sector to winchester drive
;
wwrite          lda     #cwrite         ; get write sector command
```

```
                jsr     wrwsec      ; start write sector procedure
                bcs     9.f         ; carry set = error
                ldy     #0
write1          jsr     reqwait
                bit     rssasi
                bmi     write2
                lda     [memp],y
                sta     rwdsasi
                iny
                bra     write1
write2          jsr     getstat
                bne     4.f
                rts
;
4               jsr     wrqs
                ldy     rwdsasi     result
                jsr     wrqs
                lda     rwdsasi
                tya                 error?
                beq     9.f
;
werr            sec                 set carry
9               ldy     ytemp
                rts
;
getstat         jsr     reqwait     ; wait for request
                lda     rwdsasi     ; get status
                jsr     reqwait     ; wait for second byte
                ldx     rwdsasi
                and     #2          ; isolate the error bit (%0000 0010)
                rts
;
reqwait         bit     rssasi      ; check request bit (%0100 0000)
                bvc     reqwait     ; loop until request bit set
                rts
;
selcntr         jsr     waitrdy     ; wait for ready
                lda     #1          ; controller select code
                sta     rwdsasi     ; send to transparent latch
                sta     selsasi     ; generate a select strobe
waitbsy         lda     rssasi      ; read sasi status port
                and     #wbsy       ; mask busy status bit (%0000 1000)
                beq     waitbsy     ; loop if ready
                rts
waitrdy         lda     rssasi      ; read sasi status port
                and     #wbsy       ; mask busy status bit (%0000 1000)
                bne     waitrdy     ; loop if busy
                rts
;
;
wrwsec          sty     ytemp
                sta     wcmdb       ; command (r/w)
                dey                 ; calculate address
                cpy     #128        ;  128 sectors/track
                bcs     werr        ;  1 logical track
                tya                 ; equals 4 physical tracks
```

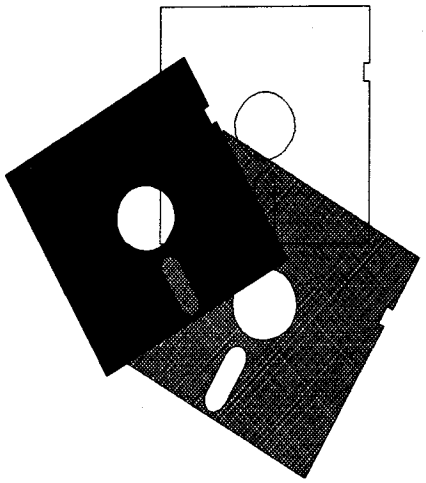
```
                asla
                stx     wcmdb+2         ; rotate right and
                lsr     wcmdb+2         ; shift in right 3 bits
                rora                    ; of track nr (max ff)
                adc     woff            ; offset (this partition)
                sta     wcmdb+3
                tay
                lda     wcmdb+2
                adc     woff+1
                sta     wcmdb+2
                bcs     werr
                cpy     wmax            check maximum address
                sbc     wmax+1
                bcs     werr

; start command

                jsr     selcntr         ; select controller
                jsr     cmdout          ; command to controller
                rts
;
cmdout          ldy     #6
                jsr     reqwait
                ldx     #0
cmd1            lda     wcmdb,x
                sta     rwdsasi
                inx
                dey
                bne     cmd1
                rts
;
wrqs            bit     rssasi          ; check request bit (%0100 0000)
                bvc     wrqs            ; loop until set
                rts
;
                res     fill-*,$ff
;
                org     $e3ff
;
fill
;
                end     winches
```

To Share Or Not To Share, That's The Question

Het aanbod van nieuwe shareware en public domain software werd de afgelopen tijd beheersd door de nieuwe bulletin board programma's. De QuickBBS groep liet de versies 2.62, 2.63 en 2.64 van QuickBBS in minder dan twee weet het levenslicht aanschouwen. Remote Access, het systeem dat inmiddels ook gebruikt wordt op "The Ultimate", het bulletin board van onze vereniging is al weer aan versie 0.04 toe; versie 0.03 is net vers van het modem of zal binnen afzienbare tijd de PTT-kas gaan spekken. De Opus-auteurs lieten na lang wachten versie 1.1 verschijnen van het Opus-systeem. En van FrontDoor, één van de beste mailers van het moment, is versie 1.99c verschenen.

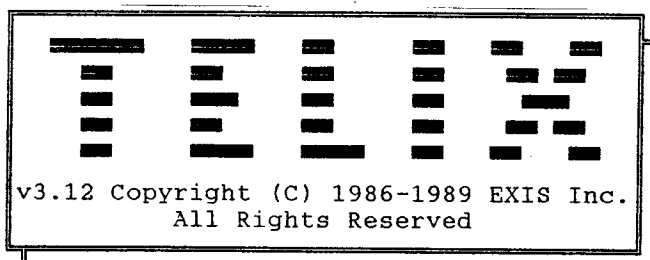


Voor het overige is er weinig nieuws onder de zon. Daarom deze keer een overzicht van een programma dat zijn sporen al ruimschoots verdiend heeft. Het programma heeft (helaas voor diegenen die niet de beschikking hebben over een modem) alweer betrekking op datacommunicatie. Het gaat hier om een meer dan goed werkende kloon van het inmiddels commercieel geworden "Procomm", het terminalprogramma "Telix". De versie die ik op de testbank had draagt het versienummer 3.12.

Sponsoring door de gebruiker

Telix is z.g. "user supported software", een andere naam voor shareware. De gebruiker mag het programma 45 dagen gebruiken ter evaluatie. Na die 45 dagen moeten er negenendertig Amerikaanse dollars betaald worden aan de "Exis Inc.", een groep in Canada. Men krijgt dan een geregistreerde versie en een serienummer toegekend.

Telix staat op de meeste bulletin boards in drie verschillende (zip-) files. Vaak staat er dan nog een script bij waarmee men met behulp van "debug" de reclame tekst uit het programma kan slopen. De eerste file bevat de Telix executable, de terminal definities en een paar voorbeelden van de script-taal van Telix. De tweede file bevat een host-script (een soort mini-BBSje), de script compiler en de documentatie van Telix zelf. De derde file tenslotte bevat de documentatie van de script taal.



```
v3.12 Copyright (C) 1986-1989 EXIS Inc.
          All Rights Reserved

  Telix is a User Supported software product.  It is not and has never
been free software.  You are granted a limited license to use this fully
working version of Telix for an evaluation period of up to 45 days.
If you wish to continue using Telix you must register by sending $39 US
or $44 CDN (Ontario orders please add 8% PST), plus $1/copy for shipping,
to EXIS at the address below.  Please see documentation for charge card
orders and volume pricing.  You will receive a 'registered' version of Telix
and manual on disk plus your own user serial number, and will be legally
entitled to continue using Telix.
  You may give this unregistered copy of Telix to others so that they may
try it out as well, but must follow the terms of the license in the Telix
documentation.  You may not modify or remove ANY of the program or documen-
tation files in the package, and no compensation, except as provided in the
license, may be received for distributing Telix.

          ■ Exis Inc., P.O. Box 130, West Hill, ON  CANADA  M1E 4R4 ■
                ■ Phone (416)-289-4641   FAX (416)-289-4645 ■
 ■ Telix Support BBS - (416)-439-8293 - 2400/1200/300 bps - 24 hours/day ■
```

Fig. 1: Telix helpt je er iedere keer weer aan herinneren dat je het programma nog niet geregistreerd hebt...

Installatie

Bij het opstarten van telix "ziet" het programma zelf dat het nog niet geïnstalleerd is. Er moet dan aangeven worden wat voor soort scherm gebruikt wordt, wat de maximum snelheid van de modem is en waar het communicatie pakket de modem terug kan vinden. De rest van de configuratie vindt geheel in het programma zelf plaats en wordt niet automatisch gedaan. In de praktijk hoeft men eigenlijk alleen maar even de subdirectories te specificeren waar Telix zijn scripts moet zoeken en waar de up- en downloads naar toe moeten.

Telefoonboek

Het pakket kan zelf bellen. Het gebruikt voor het onthouden van lijn-parameters, telefoonnummers en dat soort zaken een soort telefoonboek. Voor de conversie van de BBSlijst zoals die maandelijks samengesteld wordt door Arjen Lentz (SysOp van "Ainex BBS") naar Telix telefoondirectory formaat zijn er een aantal conversieprogramma's beschikbaar. De bespreking van deze programma's vallen buiten het bestek van dit verhaal, maar ze zijn zeker de moeite waard om ze te bekijken.

Naast dit telefoonboek houdt telix ook nog een capture-file bij waarin hele sessie "opgenomen" kunnen worden en een file waarin informatie betreffende de gemaakte verbindingen opgeslagen wordt. Voor het analyseren van deze file zijn ook weer een aantal externe utilities beschikbaar die o.a. een exact overzicht van de telefoonrekening kunnen genereren.

Programmeren

Aan Telix hebben de programmeurs meteen maar even een complete script-taal gekoppeld. Deze taal werd "SALT" gedoopt, de afkorting van "Script Application Languige for Telix". SALT lijkt heel erg sterk op een beperkte C-programmeertaal, maar dan wel met een gigantische library aan routines die speciaal voor modemgebruik ontworpen zijn. Met deze taal is het mogelijk om automatisch in te loggen op verschillende systemen en om een simpel eigen BBS-systeempje op te zetten. Een voorbeeld van een script (mijn inlogscript voor "The Ultimate") is opgenomen in figuur 3.

File transfers

Wat Telix nog sterker maakt dan veel andere communicatie programma's is de grote verscheidenheid aan communicatie protocollen die al ingebouwd zijn in Telix zelf. Daarnaast kunnen op zeer eenvoudige wijze externe protocollen aan het standaardaanbod toegevoegd worden. Telix 3.12 ondersteunt standaard Zmodem, Compuserve Quick B, Xmodem, Xmodem-1k, Xmodem-1k-g, Ymodem (TRUE), Ymodem-g, Kermit, SEAlink, Telink, Modem7 en ASCII. Een hele lijst dus. Men kan echter nog een viertal externe protocollen definiëren.

Documentatie

Hoewel Telix een redelijk help-scherm heeft en de meeste handelingen zo voor de hand liggen dat documentatie bijna overbodig zou zijn is Telix goed gedocumenteerd. De doc-file van Telix zelf vreet maar liefst 172 kB aan diskruimte en beslaat meer dan 60 pagina's A4. Een uitstekende index maakt het zoeken op onderwerp tot een peuleschil.

Naast de documentatie van het programma zelf is er ook nog een hele grote documentatiefile van de SALT programmeerinterface. Deze file omvat maar liefst 213 kB en vult ca. 150 pagina's. De index van dit bestand is kwalitatief van hetzelfde gehalte als die van de telix-documentatie zelf.

Cosmetische probleempjes

Helaas is het niet allemaal goud wat er blinkt. Er zitten een paar trekjes aan het pakket die toch wel wat afbraak doen aan dit verder met kop en schotel (?) boven de grauwe massa uitstekende pakket. Zo kan men niet definiëren dat de capturefile automatisch bij het opstarten van het pakket aan zou moeten staan. Verder zijn er, vergeleken bij het grote aanbod aan interne protocollen, wat weinig mogelijkheden gecreëerd voor het installeren van externe protocollen. Als men bijvoorbeeld JModem, BiModem, Lynx en SuperKey installeert zijn protocol-menus al vol. Toch wel zonde. Wat ook erg jammer is, is de vlo betreffende de disk-controle. In verband met de snelheid houdt Telix namelijk de inhoud van de diskette waar Telix van gestard werd in het geheugen van de PC. Bij het verlaten van Telix wordt dit zonder verdere controle weer op de diskette te-

```
Dialing Directory
          Name                    Number      Line Format     Script
  1  The Ultimate                   303902     2400 N·8·1   KIMLOGIN
  2  Universiteit Twente - BBS      898004     2400 E·7·1   UT-BBS
  3  Universiteit Twente - VAX      898004     2400 E·7·1   VAXLOGIN
  4  SFINX BBS (23:00 - 08:00)     Offline     9600 N·8·1   SFINX
```

Fig. 2: Het telix telefoonboek is heel compleet

ruggezet. Bij mij eindigde een diskette wisseling tijdens een Telix-sessie al enige malen in een catastophe... Sja, ach, en tenslotte is daar nog een persoonlijke wens. De scroll-back buffer kan namelijk niet al te groot gedefinieerd worden in Telix. En... er kan niets uit de scrollback buffer naar de printer gestuurd worden. Toch wel jammer, maar ja, je kunt niet alles willen!

Joost Voorhaar

Besproken produkt	: Telix 3.12
Catogerie	: Datacommunicatie
Registratiekosten	: $39,00
Auteur/Leverancier	: Exis Inc., Canada
Downloadbaar	: The Ultimate, modem area. Telix is verpakt in 3 verschillende files (Tlx312-1.Zip, Tlx312-2.Zip en Tlx312-3.Zip) totaal 316 kByte.

Minimale systeemeisen:

- Standaard PC of PS/2
- Minimaal 180 kB RAM vrij
- HardDisk met 1 diskdrives of 2 diskdrives (360 kB).
- Beeldscherm: CGA, , EGA, VGA, MDA, Hercules etc. etc.
- Modem

Algemene beoordeling

Documentatie	: zeer uitvoerig, Engelstalig
Online help	: matig
Gebruikersinterface	: keyboard, window ondersteuning voor setup doeleinden
Muisondersteuning	: extern

Positief

- Snel
- Uitgebreide script-taal
- Relatief weinig geheugen noodzakelijk
- Snelle diskaccess
- Uitgebreide terminal emulaties beschikbaar
- Terminal emulaties zijn zelf aan te maken en/of te wijzigen
- Functietoetsen zelf definieerbaar

Negatief

- Engelstalig
- Voor host-scripts is een Hayes compatible modem noodzaak
- Zwakke diskette diskchange afhandeling

Eindbeoordeling

Stabiliteit	: 8
Bruikbaarheid	: 9
Totaalresultaat	: 8

```
str user_name[] = "Joost Voorhaar";                    // Replace by your own name!
int loop;

main() {
    if (not _entry_pass) {                             // no password defined!
        prints ("Sorry, I don't know the password for this BBS!");
        return;
    }

    alarm (1);
    for (loop = 1; loop 20; loop = loop + 1) {
        cputc('^M');
        if (waitfor("Frontdoor",1))                    // Wacht tot FD zich meldt
            break;
    }

    if (not waitfor("Esc",60)) {
        prints("BBS cannot be opened.");
        return;
    }

    cputc(27);                                         // Awake BBS
    cputc(27);                                         // De nieuwe FroDo wil 2  Esc's

    if (not waitfor ("PCC-Net",30)) {                  // Wait for logo of "The Ultimate"
        prints("Board not recognized");                // Not found within 20 sec
        return;
    }

    cputc('S');                                        // Skip following text

    if (not waitfor ("name",20)) {                     // Wait for prompt
        prints ("Log-on failed on user prompt");
        return;                                        // ...abort
    }

    cputs (user_name);                                 // send name
    cputc ('^M');

    if (not waitfor ("Password:", 40)) {               // if no prompt for password
        prints ("Log-on failed on password prompt");
        return;                                        // ...abort
    }

    cputs (_entry_pass);                               // send password
    cputc ('^M');
    if (waitfor("Nederland",120)) {                    // Wait for end of welcome screen
        cputc('^M');
    }
}
```

Fig. 3: Een Telix script voor het inloggen op "The Ultimate"

Pointers in Pascal (en C)

Inleiding

In talen zoals Pascal en C hebben we behalve de voor de hand liggende datatypen zoals getallen en letters ook het datatype pointer. Uit gesprekken met anderen heb ik de indruk gekregen dat een hoop mensen niet weten wat ze eigenlijk met die pointers aanmoeten en hoe ze zinvol te gebruiken zijn. Een collega van me (vrijgezel overigens) deed een keer de uitspraak "Pointers, daar moet ik niets van hebben, ik heb liever iets concreets in plaats van een pointer er naar toe. Ik wil geen pointer naar een vrouw, ik wil een echte vrouw". Daarmee heeft hij wel de kern van de pointer aangegeven. Een pointer is niet iets, een pointer is een verwijzing naar iets, en dat iets hoeft er helemaal nog niet te zijn. In het volgende artikeltje en het bijbehorende stukje programmatuur hoop ik iets meer duidelijkheid te scheppen over de pointer en zijn gebruik.

Declaraties en gebruik

In Pascal wordt een pointer bij de declaratie aangegeven met een dakje (^). Achter dat dakje staat naar welk datatype de pointer wijst. Als voorbeeld enkele declaraties uit Pascal:

```
TYPE
      TP_Structuur = ^T_Structuur;
      T_Structuur  =
                RECORD
                   A : INTEGER;
                   B : INTEGER;
                   Volgende : TP_Structuur;
                END;

VAR
      IP       : ^INTEGER;
      Start    : TP_Structuur;
```

In dit voorbeeld hebben we in de eerste plaats een pointer naar een geheel getal, IP. Deze pointer kan het adres van een geheel getal bevatten, dus niet het getal zelf. De pointer wijst dus als het ware naar een geheel getal. Evenzo kan de variabele Start naar een datastructuur van het type Structuur wijzen. Dit is een zogenaamde record-definitie, een aantal variabelen bij elkaar horen en die samen een datastructuur vormen. Deze datastructuur heet in Pascal een RECORD. In dit record zit weer een verwijzing naar eenzelfde datastructuur. Straks zullen we laten zien waarom dat handig kan zijn.

In de taal C kunnen we dezelfde declaraties hebben en die zien er dan als volgt uit:

```
typedef struct T_structuur {
        int         A;
        int         B;
        struct T_Structuur *Volgende;
        } *TP_Structuur;

int                 *IP;
TP_Structuur        Start;
```

Behalve Pascal en C zijn er nog meer talen die gebruik kunnen maken van pointers. Zelfs in assembler kunnen we ze gebruiken. Ik heb al aangegeven dat een pointer niets meer of minder is als een adres van een datastructuur. In assembler kunnen we natuurlijk ook zoiets maken door het adres van de datastructuur ergens op een adres of in een register te schrijven.

Bij pointers is het zo dat er bij de declaratie nog geen ruimte voor de datastructuur waar naartoe de pointer verwijst is gereserveerd. Dat moet expliciet gedaan worden door het aanroepen van een procedure of functie. In Pascal is dit de procedure NEW die aangeroepen wordt met bijvoorbeeld:

```
NEW(Start);
```

In C doen we dit meestal met de functie malloc die echter een iets omslachtiger aanroep heeft:

```
Start = (TP_Structuur)
malloc(sizeof(T_Structuur));
```

Het verschil tussen Pascal en C is het feit dat Pascal "weet" hoeveel geheugen er nodig is. Bij C wordt met malloc een hoeveelheid geheugen gereserveerd en moet expliciet aangegeven worden hoeveel geheugen we nodig hebben. Bovendien moet in C aangegeven worden wat het type van het antwoord is; dat wordt gedaan door een zogenaamde typecast.

Geheugen dat aangevraagd wordt moet ook weer vrijgegeven worden. Na afloop van het programma zorgt het operating systeem hier wel voor doch intern in het programma wordt dit niet automatisch gedaan. Als we dus geheugen aangevraagd hebben en we hebben het niet meer nodig, dan moeten we dit in de programmatuur zelf weer vrijgeven. In Pascal doen we dit met:

```
DISPOSE(Start);
```

en in C met:

```
free(Start);
```

Behalve het verschil in declaratie is er nog een zeer wezenlijk verschil tussen Pascal en C. In C kunnen we een pointer naar een integer zonder meer laten wijzen naar elke gedefinieerde integer. Dat doen we door de operator '&' te gebruiken dus bijvoorbeeld:

```
int A,*B;
{ B = &A;}
```

In standaard Pascal gaat dit niet; daar kunnen pointers alleen wijzen naar variabelen die met NEW gedefinieerd zijn. De meeste implementaties van Pascal kennen echter de ADR of ADRESS-functie waarmee, zoals de '&'-operator in C het adres van de variabele aan de pointer wordt doorgegeven.

Als een pointer nog nergens naar wijst of nergens meer naar mag wijzen, dan moeten we dit aangeven. Zowel in C als in Pascal is hier een speciale waarde (adres) voor gedefinieerd. In C is dit de waarde NULL, in Pascal de waarde NIL. Dit betekent dat de pointer "nergens" naar wijst.

Mogelijkheden

Naar mijn mening worden pointers, vooral bij amateurs, te weinig gebruikt. Met behulp van pointers en bijbehorende datastructuren kan men allerlei gegevens in het programma opslaan zonder dat van te voren bekend is om hoeveel gegevens het gaat. Als onderdeel van dit artikel wordt een programma beschreven waarmee, van een willekeurige niet te grote tekst, onderzocht wordt hoe vaak elk woord in de tekst voorkomt. Na afloop wordt er een overzicht gemaakt waarin elk woord dat in de tekst voorkomt in alfabetische volgorde wordt afgedrukt en hoe vaak het woord in de tekst voorkomt.

Voor een dergelijk probleem kan onder andere voor een lijststructuur gekozen worden. In figuur 1 zijn twee voorbeelden van een dergelijke lijststructuur getekend.

De lijsten bestaan uit records waarin de informatie die we nodig hebben staat. Dit is het woord dat we gevonden hebben, hoe vaak het voorkomt en hoe lang het woord is. Deze gegevens vormen samen een record. Verder is er bij de eerste lijst steeds een pointer die naar het volgende element (record) in de

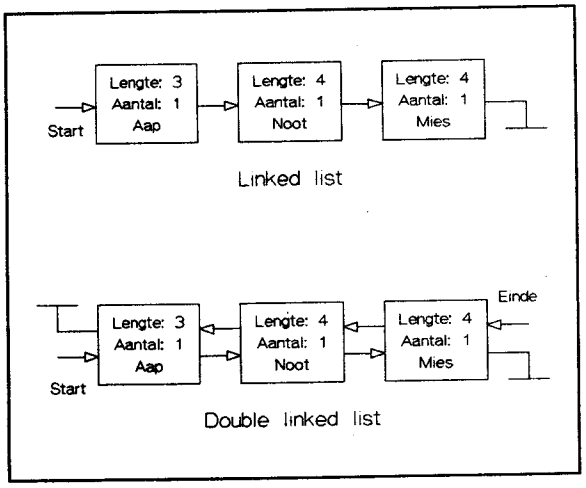


Fig. 1: Voorbeelden van een lijst structuur

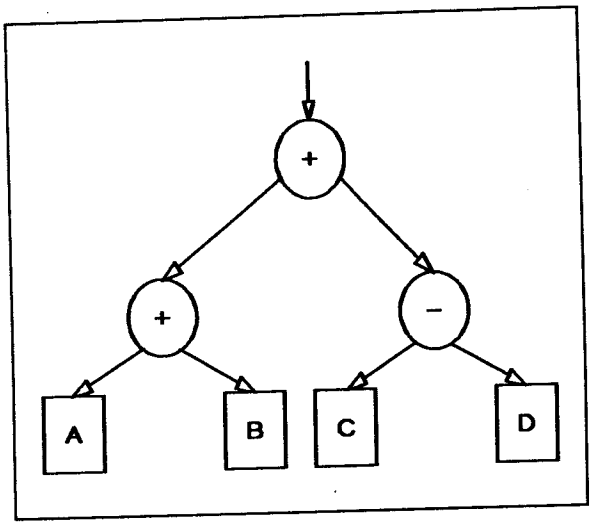


Fig. 2: Boomstructuur

lijst wijst. Naar de kop van de lijst wijst een pointer die we Start genoemd hebben en vanuit een record naar het volgende record wijst de pointer die Volgende heet. Iedere keer als we een nieuw woord in de lijst toe willen voegen, zoeken we de plaats waar het record voor dit woord moet staan. Vervolgens worden de pointers zo veranderd dat het record op de juiste plaats staat. We hoeven dus alleen maar een paar pointers te veranderen en niet iedere keer de hele lijst op te schuiven. In het voorbeeldprogramma, dat gebruikt maakt van de eerste lijststructuur, staat precies hoe dat gedaan moet worden.

De eerste lijst heeft als nadeel dat je maar in één richting door de lijst kunt lopen. Je kunt vanuit een record alleen naar de volgende. Om aan dit bezwaar tegemoet te komen hebben we in de tweede lijst per record een pointer naar de volgende en een pointer naar de vorige. Verder hebben we een pointer naar het begin en naar het einde van de lijst.

Behalve de twee getekende lijsten, zijn er nog meer mogelijkheden. Zo is het bijvoorbeeld ook mogelijk het laatste record in de lijst weer te verbinden met het eerste record. Een dergelijke circulaire lijst wordt vaak gebruikt voor printerbuffers en device drivers. Er is dan een lijst met een vast aantal records gedefinieerd en er wordt gebruik gemaakt van een lees- en een schrijfpointer. Als de lees- en de schrijfpointer beide naar hetzelfde record wijzen, dan is het buffer leeg. Als er iets in het buffer geschreven wordt, gaat de leespointer naar het volgende record, wordt er iets naar de printer gestuurd, dan wordt de schrijfpointer met één verplaatst, net zolang totdat ze weer gelijk zijn.

Behalve lijsten zijn er nog een groot aantal andere mogelijkheden. In figuur 2 is de zogenaamde boomstructuur getekend.

In deze boomstructuur is de rekenkundige expressie:

(A + B) * (C - D)

opgeslagen. Hierbij hebben we twee soorten records, operanden (*,/,-,+) en variabelen (A,B,C,D). In een operand hebben we een pointer naar de linker operator en een pointer naar de rechter operator waarbij de pointer kan wijzen naar een andere operand of naar een variabele. Op deze manier kunnen we de bovengenoemde expressie opslaan. In een volgende uitgave van de μP Kenner zal ik aangeven hoe zoiets in Pascal geprogrammeerd kan worden.

Afsluiting

Ik hoop dat ik in dit artikel en in het bijgevoegde programma-voorbeeld iets meer duidelijkheid over pointers heb kunnen geven. Naar mijn mening is namelijk de pointer het meest krachtige middel om gegevens te structureren, zonder dat je van te voren precies hoeft te weten om hoeveel gegevens het gaat en hoe de structuur precies is. Om deze reden worden pointers ook veelvuldig gebruikt, behalve in de reeds genoemde voorbeelden bijvoorbeeld ook in databases, editors, tekstverwerkers en in data-compressieprogramma's. In een volgende uitgave van de μP Kenner kom ik hier graag nog eens op terug.

Gert van Opbroek

```
PROGRAM Demopointers(Input,Output{,Invoerfile});

{ ************************************************************************

    Versie 1.0: 01-04-1989/G. van Opbroek
    Copyright (C) 1990 KIM Gebruikersclub Nederland.

    Dit programma demonsteert het gebruik van pointers. Het programma legt een zogenaamde "Linked
List"  aan waarin alle woorden die in de tekst in de Invoer_file voorkomen opgslagen worden. Na
afloop wordt er een overzicht afgedrukt waarin alle woorden, netjes gesorteerd worden afgedrukt
met daarbij hoe vaak
    een woord voorkomt.

    Het programma is geschreven in Microsoft Pascal onder PC-DOS doch zou eenvoudig geconverteerd
    moeten kunnen worden naar andere Pascal-versies.

 ************************************************************************ }

CONST  CMaxWoord    = 50 ; { Grootste woordlengte }

TYPE TWoord = PACKED ARRAY [1..CMaxWoord] OF CHAR;
   TWoordpointer = ^TWoordbuffer;
   TWoordbuffer  = RECORD
         Lengte   : INTEGER;
         Aantal   : INTEGER;
         WOORD    : TWoord;
         Volgende : TWoordpointer;
         END;

VAR  Invoerfile   : TEXT;
   Filenaam     : LSTRING(50); { Systeemafhankelijk }
   Woordenlijst : TWoordpointer;
   Aantal       : INTEGER;

{ ************************************************************************

   Functies t.b.v. het omgaan met de lineaire linked list:
```

```
    - MaakWoordbuffer:
        Vraagt van het operating systeem ruimte voor een woordbuffer, initieert dit buffer met nullen en
        spaties en geeft een pointer naar dit buffer als functieresultaat.
    - Voegtussen:
        Deze procedure heeft als parameters een (VALUE) pointer naar een woordbuffer en een
        (REFERENCE) pointer naar een lijst van woordbuffers. Het woordbuffer wordt in de lijst
        gezet voor de door de pointer aangegeven plaats.
    - Verwijder:
        Deze procedure heeft als parameter een (REFERENCE) pointer naar een lijst van woordbuffers.
Uit deze      lijst wordt het door de parameter aangegeven element verwijderd.
    - VerwerkWoordbuffer:
        Deze functie heeft twee (REFERENCE) parameters van het type TWoordpointer. In eerste
        instantie wordt in de lijst de plaats gezocht waar het woord tussengevoegd moet worden.
        Bestaat het woord al, dan wordt de aantal-teller in het woord verhoogd.
        Bestaat het woord nog niet, dan wordt het woord tussengevoegd.
        In het eerste geval wordt bovendien het woordbuffer netjes opgeruimd.

 ********************************************************************** }

 FUNCTION MaakWoordbuffer : TWoordpointer;

 VAR I        : INTEGER;
    Kladwoord: TWoordpointer;

 BEGIN
    NEW(Kladwoord);
    WITH Kladwoord ^
    DO BEGIN
    Lengte  := 0;
    Aantal  := 0;
    FOR I   := 1 TO CMaxWoord DO Woord[I] := ' ';
    Volgende := NIL;
    END;
    MaakWoordBuffer := Kladwoord;
 END;

 PROCEDURE Voegtussen( VAR Lijst  : TWoordpointer;
                       Buffer : TWoordpointer);

 BEGIN
    Buffer ^.Volgende := Lijst;{ Neem de verwijzing over in het buffer }
    Lijst       := Buffer;{ En deze pointer wijst nu naar het buffer }
 END;

 PROCEDURE Verwijder( VAR Lijst: TWoordpointer);

 VAR Kladwoord : TWoordpointer;

 BEGIN
    IF Lijst  NIL
    THEN BEGIN
    Kladwoord := Lijst;
    Lijst    := Kladwoord ^.Volgende;
    DISPOSE(Kladwoord);
    END;
 END;
```

```
PROCEDURE VerwerkWoordbuffer(VAR Lijst,Woord : TWoordpointer);

VAR Kladpointer : TWoordpointer;
    Oudpointer: TWoordpointer;

BEGIN
    IF Lijst = NIL{ De eerste apart behandelen; nadeel van lijsten }
    THEN Lijst := Woord
    ELSE BEGIN
    Kladpointer := Lijst;
    WHILE (Kladpointer^.Volgende  NIL) AND
        (Kladpointer^.Woord  Woord^.Woord)
    DO BEGIN
        Oudpointer  := Kladpointer;
        Kladpointer := Kladpointer^.Volgende;
    END;

    IF Kladpointer^.Woord = Woord^.Woord { We hadden hem al }
    THEN BEGIN
        Kladpointer^.Aantal := Kladpointer^.Aantal + 1;
        Verwijder(Woord);
    END ELSE
    IF Kladpointer^.Woord  Woord^.Woord { Nieuw woord }
    THEN IF Kladpointer = Lijst { Eerste woord}
        THEN Voegtussen(Lijst,Woord)
        ELSE Voegtussen(Oudpointer^.Volgende,Woord)
    ELSE    Voegtussen(Kladpointer^.Volgende,Woord);{ Laatste woord }
    END;
END;

{ ************************************************************************

    De Procedure LeesEnVerwerkWoord leest woorden uit de invoerfile en
    verwerkt deze in de lijst. Spaties en leestekens worden overgeslagen.

    ************************************************************************ }

PROCEDURE LeesEnVerwerkWoord;

VAR C       : CHAR;
    Kladwoord: TWoordpointer;
    Stop    : BOOLEAN;

BEGIN
    Stop := FALSE;
    Kladwoord := NIL;
    C := ' ';
    WHILE NOT Stop
    DO BEGIN

    WHILE NOT (C IN ['0'..'9','A'..'Z','a'..'z']) AND
        NOT  EOF(Invoerfile)
    DO READ(Invoerfile,C);

    IF C IN ['0'..'9','A'..'Z','a'..'z']
    THEN BEGIN
```

```
        IF C IN ['a'..'z'] THEN C := CHR(ORD(C) + ORD('A') - ORD('a'));
        Kladwoord := MaakWoordbuffer;
        Kladwoord^.Aantal := 1;
        REPEAT
      Kladwoord^.Lengte := Kladwoord^.Lengte + 1;
      Kladwoord^.Woord[Kladwoord^.Lengte] := C;
      IF NOT EOF(Invoerfile)
      THEN BEGIN
         READ(Invoerfile,C);
         IF C IN ['A'..'Z'] THEN C := CHR(ORD(C) + ORD('a') - ORD('A'));
      END ELSE C := ' ';
      Stop := NOT (C IN ['0'..'9','A'..'Z','a'..'z'])
         UNTIL Stop;
         IF Kladwoord NIL
         THEN VerwerkWoordbuffer(Woordenlijst,Kladwoord);
      END;
      Stop := Stop OR EOF(Invoerfile);
      END;
  END;

  BEGIN { Hoofdprogramma }
      { Het volgende stukje is systeem-afhankelijk ! }
      WRITE('Geef het pad voor de file die geanalyseerd moet worden : ');
      READLN(Filenaam);
      ASSIGN(Invoerfile,Filenaam);
      { - }
      RESET(Invoerfile);
      Woordenlijst := NIL;
      Aantal := 1;
      WRITELN;
      WRITELN('Inlezen en verwerken van ',Filenaam);
      WHILE NOT EOF(Invoerfile)
      DO BEGIN
      LeesEnVerwerkWoord;
      WRITE('.');
      IF Aantal MOD 75 = 0 THEN WRITELN;
      Aantal := Aantal + 1;
      END;
      WRITELN; WRITELN;
      WRITELN('Resultaten :');
      WRITELN('= = = = = = = = = = = = '); WRITELN;
      WRITELN('Woord                              Lengte  Aantal');
      WHILE Woordenlijst NIL
      DO BEGIN
      WRITELN(Woordenlijst^.Woord,
          Woordenlijst^.Lengte:6,' ',
          Woordenlijst^.Aantal:6);
      Verwijder(Woordenlijst);
      END;
      CLOSE(Invoerfile);  { Systeemafhankelijk }
   END.
```

Fig. 1: Pointers in de praktijk

De IBM-PC en z'n klonen (Deel 8)

De vorige keer hebben we gekeken wat het systeem-BIOS allemaal doet om de PC aan de praat te krijgen, voordat het DOS geboot wordt. Dit maal komt de andere helft van het systeem-BIOS aan bod. Dat is het deel dat het eigenlijke BIOS vormt: het basis invoer- en uitvoer systeem. Dit stukje software kan een groot aantal taken op het allerlaagste niveau in de machine uitvoeren. Zo zijn er functies voor het video, het toetsenbord, de printer- en RS-232-poorten en de floppy disk drives. In de oer-PC wordt zelfs de cassette-interface in het BIOS ondersteund. IBM heeft het BIOS zo in elkaar gezet, dat er nooit naar absolute adressen gesprongen hoeft te worden. En daarbij maakten ze een fout. Lees maar verder.

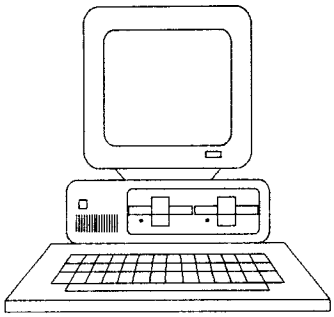
De BIOS-structuur

Ja hoor, alweer: interrupts. Maar nu software interrupts. Want dat is de weg waarmee het BIOS adresonafhankelijk is gemaakt. In 1 van de vorige delen hebben we gezien, dat de processor op twee manieren een interrupt kan krijgen: via de interrupt-request lijn en via een INT-instructie. Over de eerste hebben we het al gehad. De INT-instructie echter nog niet.

Die INT-instructie wordt gevolgd door het interruptnummer als operand. Dit is altijd een constante, die 1 byte lang is. Deze instructie zet het terugkeeradres op de stack, en ook de huidige vlaggen. Uit de eerste 1 kbyte geheugen worden vervolgens offset en segment van de interrupt-handler opgehaald op adres 0000:4*INT-nummer.

Het interruptnummer geeft de soort functie aan in het BIOS. Verderop staat een lijstje. IBM begon te tellen bij 10 (hex). Dat leek wel veilig: Intel had de interrupts 5 t/m 1Fh namelijk als gereserveerd voor toekomstige uitbreidingen in het handboek staan. De 80386 gebruikt echter potentieel een aantal van door het BIOS benutte interrupts. IBM leest dus ook niet altijd de datasheets even goed.....

De via de INT-instructie aangeroepen functie heeft natuurlijk nog wat parameters nodig. Bij de 6502 zouden we die in het geheugen zetten, maar de 8088 en consorten hebben een hele batterij registers, die prachtig als doorgeefluik dienst kunnen doen. IBM heeft ervoor gekozen, in register AH aan te geven wat er precies moet gebeuren. In AL staat dan meestal de meest belangrijke parameter. Parameters die een woord lang zijn, worden meestal door CX en DX overgedragen. Het paar ES:BX is populair om pointers over te brengen.



Een BIOS functie wordt dus meestal zo aangeroepen:

```
MOV   AH,subfunctienummer
MOV   AL,parameter1
MOV   ??,parameterX
MOV   ??,parameterY
INT   YYh
```

Dit mechanisme werkt ook andersom: de functies gebruiken de register ook weer om resultaten aan de aanroeper kenbaar te maken. In dit verband is ook de carry-vlag populair: is de vlag gezet, dan is de aangeroepen functie niet goed uitgevoerd, of niet beschikbaar in een aantal gevallen. Tot zover de structuur van het BIOS. Laten we maar eens kijken welke functies er zijn.

De BIOS functies

Zoals reeds gesteld, zijn vanaf INT 10h de BIOS-functies ter beschikking. Het volledige lijstje in een PC(/XT) ziet er zo uit:

INT	Functie
10h	Video output
11h	Equipment check
12h	Get memory size
13h	Diskette services
14h	RS-232 services
15h	Cassette services
16h	Keyboard services
17h	Parallel printer services
18h	Call ROM-BASIC (alleen IBM)
19h	Boot DOS
1Ah	Time services
1Bh	BIOS break
1Ch	Timer tick interrupt
1Dh	6845 CRTC parameters
1Eh	Diskette parameters
1Fh	Extended CGA character generator

Niet alle INT-functies hierboven zijn ook inderdaad functies. De interrupts 1Dh, 1Eh en 1Fh worden bijvoorbeeld gebruikt om een pointer aan te geven. Van een interrupt staat het adres immers altijd op een vaste plaats. Het DOS kan dat adres dan gebruiken om achter de plek van een tabel te komen. In

het BIOS zijn er dus drie tabellen, die via de vectorentabel gevonden kunnen worden.

Een aantal calls worden vrijwel nooit gebruikt. Zo is de boot DOS call INT 19h eigenlijk een interne BIOS call. Hetzelfde geldt, bij IBM machines althans, voor INT 18h, de aanroep van de BASIC ROMs met de cassette BASIC.

In een PC/XT is de call INT 15h (cassette services) een erfenis van de PC, die immers een cassette interface had. In een XT geeft INT 15h altijd de code 'illegal function' en een gezette carry terug. Het is dan ook de eenvoudigste BIOS-call. Maar we gaan eerst de 'normale' BIOS-calls eens wat nader bekijken.

INT 10h, video services

INT 10h is de interface tussen de gebruiker en de displaykaart in de machine. Zolang de gebruiker met karakters werkt, hoeft hij geen weet te hebben van de soort displaykaart er in de machine zit. Dat blijft voor hem grotendeels onzichtbaar. Omdat er zelfs in de oer-PC twee verschillende displaykaarten konden zitten, waaronder de duizendpoot CGA, is het INT 10h BIOS het grootste van allemaal: ongeveer 2k aan code, de aparte karaktergenerator voor de CGA-kaart in graphics mode niet eens meegerekend (die beslaat ook nog eens een kilobyte). Het INT 10h video-BIOS kent zestien functies, die zoals hierboven al aangegeven met register AH worden aangewezen. Hier zijn ze:

AH	Subfunctie
00	Initialiseer video kaart, wis scherm
01	Stel cursor formaat in
02	Zet cursor op bepaalde positie
03	Lees huidige cursor positie
04	Lees lichtpen positie
05	Kies actieve video pagina
06	Scroll de actieve video pagina op
07	Scroll de actieve video pagina neer
08	Lees attribuut/karakter onder cursor
09	Schrijf attribuut/kar. onder cursor
0A	Schrijf karakter onder cursor
0B	Kies kleuren palet (alleen CGA)
0C	Schrijf een pixel (alleen grafisch)
0D	Lees een pixel (alleen grafisch)
0E	Write TTY
0F	Lees de huidige video status.

Het voert te ver, iedere functie apart in dit verhaal te gaan behandelen: dan wordt deze serie wel dertig delen lang (dat duurt zes jaar!). Ook geven de meeste boeken over machinetaal programmeren wel een volledige lijst. Daarom een korte omschrijving van de zaken die wat minder vanzelfsprekend en/of nieuw zijn.

Een aantal calls worden vrijwel nooit gebruikt. Zo is de boot DOS call INT 19h eigenlijk een interne BIOS call. Hetzelfde geldt, bij IBM machines althans, voor INT 18h, de aanroep van de BASIC ROMs met de cassette BASIC.

Wat in ieder geval nieuw is, is de functie write TTY. Wat deze functie doet, is met een minimum aan parameters een karakter op het scherm zetten. Nu zult u zeggen, dat kan met de functies 09 of 0A ook. Dat is juist. Maar die twee functies doen ook alleen maar dat: het karakter op de plaats van de cursor schrijven. Verder niet. De cursor wordt dus niet eens verplaatst: daarvoor is functie 02 in het leven geroepen. Verder is het zo, dat de ASCII waarden 0 t/m 1F bij IBM ook karakters bevatten, ofschoon bijvoorbeeld 0Dh meestal bedoeld is als carriage return. Dus moet je eigenlijk nog heel wat doen, wil je op de normale manier een karakter op het scherm kalken met de 'normale' video calls. Daarom write TTY. Deze functie beweegt ook vanzelf de cursor, kijkt of we aan het einde van de regel zijn en begint dan automatisch op de volgende, scrollt het scherm zonodig en werkt bovendien per default op de actieve pagina. Verder interpreteert write TTY de karakters ^G (bell), ^J (line feed) en ^M (carriage return) op de gebruikelijke wijze. Tenslotte is ook de keuze van het attribuut automatisch: dat wordt gejat van het vorige karakter.

Met de kennis van write TTY eigenschappen blijkt dat er dus veel meer kan. Zo schrijven de functies 09 en 0A desgewenst ook karakters op de niet actieve display pagina's zodat een scherm onzichtbaar kan worden opgebouwd. Door dat onzichtbare scherm met functie 05h te activeren kan in een oogwenk een heel scherm ververst worden.

Van de functies 09h en 0Ah is nog iets bijzonders te vertellen. Ze lijken bedoeld voor tekstbedrijf, en daar worden zo het meest voor gebruikt. De CGA-kaart kan echter ook in grafische mode bedreven worden, en dan werken deze functies ook. Er is in het BIOS ROM namelijk ook nog een karaktergenerator voor CGA-kaart in grafisch bedrijf opgeslagen. In grafische mode tekenen deze functies het complete gevraagde karakter in pixels op het scherm. Dat schiet niet erg op, maar het werkt wel. De functie 08 doet het omgekeerde: deze leest het karakter onder de cursor, alweer niet alleen in tekst

mode maar ook in grafisch bedrijf. De genoemde karaktergenerator bevat wegens plaatsgebrek overigens slechts de bitpatronen van de karakters 0 t/m 127. Om de tweede helft van de generator te vinden, gebruikt het video BIOS de INT 1Fh vector. Die vector wordt door het BIOS op nul gezet. De gebruiker moet dus eventueel zelf voor de tweede helft van de generator zorgen.

Functie 00h initialiseert de videokaart, en wist ook het scherm. In AL wordt dan het zogenaamde mode-nummer gezet. Omdat dit nummer nogal eens opduikt, even een lijstje:

Mode	Display
00	CGA, 40x25, tekst, zwart/wit
01	CGA, 80x25, tekst, zwart/wit
02	CGA, 40x25, tekst, kleur
03	CGA, 80x25, tekst, kleur
04	CGA, 320x200 graphics, kleur
05	CGA, 320x200 graphics, Z/W
06	CGA, 640x200 graphics, Z/W
07	MDA, 80x25, tekst

Na een mode-reset is de default-kleur altijd normaal wit. De modes 0 en 2 respectievelijk 1 en 3 zien er op een TTL monitor gelijk uit: modes 0 en 1 genereren alleen geen burst in het kleurensignaal waardoor een aangesloten NTSC-monitor overgaat op zwart-wit weergave.

IBM hanteert verder intern de regel dat er alleen een nieuw modenummer wordt gekozen als de essentiële parameters van een mode veranderen. Deze parameters zijn het formaat (resolutie), de indeling van het videoRAM of het adres van het videoRAM. Zo is op een EGAkaart mode 3 in deze opzichten geheel gelijk aan de mode 3 op een CGA kaart, ofschoon er andere karakters (8x14 i.p.v. 8x8 pixels) worden afgebeeld.

In grafische mode tekenen deze functies het complete gevraagde karakter in pixels op het scherm. Dat schiet niet erg op, maar het werkt wel.

Tot zover het video BIOS.

INT 11h, equipment check

De titel lijkt indrukwekkend: er wordt uitgezocht wat er aan spullen aan boord van de PC zit. Dat is niet helemaal waar, want dat heeft de POST al voor ons gedaan. INT 11h geeft alleen het word terug in AX waarin de configuratiegegevens zijn opgeslagen. De onderste helft van dat word komt overigens overeen met de switch-setting op het moederbord, waarbij een ON-switch als nul wordt gelezen. Het bovenste byte wordt als volgt gebruikt:

Bit #:	Betekenis
8:	niet gebruikt
9-11	aantal RS-232 poorten
12:	game adapter aanwezig
13:	niet gebruikt
14-15:	aantal LPT-poorten

Er is geen vlag voor een real time clock.

INT 12h, haal geheugengrootte

Deze call is net zo indrukwekkend als de vorige: de POST heeft deze informatie ook al achtergelaten in een geheugenlocatie. Bij terugkeer bevat AX het aantal kilobytes RAM in de machine.

INT 13h, diskette services

Dit deel van de BIOS is verantwoordelijk voor het verkeer met de floppy disk drives. Het kent zes functies:

AH	Functie
00	Reset/initialiseer disk controller
01	Vraag controller status op
02	Lees 1 of meer sectoren
03	Schrijf 1 of meer sectoren
04	Verifieer 1 of meer sectoren
05	Formatteer een track

Het INT 13h BIOS geeft altijd een carry clear als de operatie geslaagd is, en een carry set bij een fout. Verder is het zo, dat het BIOS gedetailleerdere foutmeldingen afgeeft aan het DOS dan het DOS aan de gebruiker doet. (Dit is 1 van de ernstigste gebreken van MS/PC-DOS).

Functies 00 en 01 spreken wel voor zichzelf. De functies 02 t/m 05 hebben een genormaliseerde interface met betrekking tot de parameter overdracht. DL is altijd het drivenummer, DH het kopnummer. In CH komt het tracknummer, in CL het sectornummer waarmee moet worden begonnen. Tenslotte bevat AL het aantal sectoren en wijst ES:BX een stuk geheugen aan waar de data vandaan komt of heen moet. De enige beperking die geldt, is dat bij gebruik van meerdere sectoren er niet over een track- of kop-grens heen kan worden gewerkt.

Met het diskette BIOS kun je dus rechtstreeks op de schijf kijken en schrijven, langs de MS/PC-DOS file manager heen. Dit is dan ook precies wat bijvoorbeeld programma's als PCTOOLS en de Norton Utilities doen. Ook DEBUG kan het indien gewenst.

Het diskette BIOS maakt gebruik van de parameters die aangewezen worden door de vector voor INT 1Eh. Dit is een tabel voor de initialisatie van de

PD765 floppy disk controller, die onder andere de step-rate bevat. Het standaard PC(/XT) BIOS kent maar 1 drivetype: de 360 kbyte floppy drive met 40 tracks, 1 of 2 koppen, 8 of 9 sectoren per track en 512 bytes per sector. Het KGN-BIOS detecteert daarboven automatisch een 720 kbyte drive, die immers 80 tracks heeft, maar verder dezelfde parameters.

INT 14h, RS-232 services

Met dit stukje BIOS kan er met de RS-232 poorten gewerkt worden. De beschikbare functies zijn:

AH	Functie
00	Initialiseer een poort
01	Zend een karakter
02	Ontvang een karakter
03	Vraag de status op

De verdere interface met deze functies is eenvoudig: AL bevat het karakter of het poortformaat, danwel de status, en DX geeft het poortnummer aan. Dat is alles. Het BIOS ondersteunt de volgende baudrates: 110, 150, 300, 600, 1200, 2400, 4800 en 9600 baud. Er kan met 1 of 2 stop bits, even, oneven of geen pariteit en 7 of 8 databits gewerkt worden. Tenslotte ondersteunt het BIOS ook hardware handshake. XON/XOFF mag de gebruiker zelf doen.

De klok telt gewoon clock-ticks, waarvan er 18,2 in een seconde gaan. De aanroeper moet dus nog even zelf de echte tijd in uren, minuten en seconden uitknobbelen. Een datum kon er in de PC niet vanaf.

INT 15h, Cassette services

Die bestaan dus niet meer. Deze call keert altijd terug met een gezette carry en AH = 86h, hetgeen illegal call betekent.

INT 16h, Toetsenbord services

Met deze call kan het toetsenbord worden afgevraagd. Er zijn in totaal drie functies beschikbaar:

AH	Functie
00	Haal een karakter/scancode
01	Kijk of er een karakter is
02	Haal de shiftstatus

Behalve AH zijn geen verdere registers nodig. Er is immers maar 1 toetsenbord. De functies 00 en 01 zijn sterk verwant. Het verschil is dat functie 00 wacht als er geen toets in de buffer zit. Functie 01 geeft met de Z-vlag aan of er een toets beschikbaar is. Is die er (Z-vlag = 0), dan bevat AL de ASCII waarde, en AH de scan code. Een tweede verschil is, dat bij functie 01 de ASCII-waarde en de scancode in de buffer blijven staan, en bij functie 00 de buffer pointers worden bijgesteld.

De hierboven bedoelde buffer wordt door het BIOS zelf bijgehouden, en kan 16 toetsaanslagen bevatten. De buffer bevat zowel de ASCII-waarden, als de scancodes van de toetsen. Is er geen ASCII code beschikbaar, dan is de ASCII-waarde nul.

INT 17h, Printer services

De printer wordt met deze interrupt bediend. Ook hier zijn weer verschillende functies ten dienste van de gebruiker gesteld:

AH	Functie
00	Druk een karakter af
01	Initialiseer een printerpoort
02	Lees de printerstatus

DX bevat net als bij RS-232 (INT 14h) het poortnummer, en AL het karakter danwel de status. De initialisatie van een poort stuurt ook de lijn INIT een keer aan, waardoor een aantal printers zelf een harde reset binnenkrijgt. Via de printerstatus kun je erachter komen of de printer wel on-line is, en of er papier in zit.

INT 18h, call BASIC ROMs

De bovengenoemde functie is voor echte IBM's. Hij wordt aangeroepen als een DOS boot poging gefaald heeft. Kloon-BIOSsen springen op een aantal manieren met deze call om. Sommige springen inderdaad naar F600:0000, de start van de BASIC-ROMs. Andere printen een tekst in de trant van 'Boot fail, press the any key...', weer andere BIOSsen doen er helemaal niets mee. Het KGN-BIOS valt in zowel de eerste, als de tweede categorie. De eerste, als er IBM BASIC ROMs in de machine zitten, anders in de tweede.

INT 19h, boot DOS

Deze functie is de laatste instructie van de POST. Hij probeert sector 1 van track 0, head 0 van drive 0 (A:) te lezen. Lukt dat, dat wordt die sector geladen op adres 0000:7C00, en wordt naar het eerste geladen byte gesprongen.

Gaat de leesactie echter mis, dan doet IBM een INT 18h, de meeste kloon-BIOSsen beginnen meestal over ontbrekende schijfjes te klagen.

INT 1Ah, Tijd services

Langs deze weg kun je aan de PC vragen hoe laat het is, of de locale klok gelijk zetten. Nu heeft de PC geen batterijklok standaard, dus meestal wordt eerst de klok gelijk gezet, en er pas daarna om gevraagd.

Het is geen mooie klok. Integendeel. De klok telt gewoon clock-ticks, waarvan er 18,2 in een seconde gaan. De teller wordt om middernacht automatisch op nul gezet, compleet met een vlaggetje om aan te geven dat er door 0:00 uur is heengegaan. De aanroeper moet dus nog even zelf de echte tijd in uren, minuten en seconden uitknobbelen. Een datum kon er in de PC niet vanaf.

Met AH = 00 kom je het huidige aantal clock-ticks sinds middernacht te weten in CX:DX. AL is de rol over vlag. Met AH = 01 kun je de klok gelijk zetten. De rol over vlag wordt bij beide aanroepen automatisch gewist.

INT 1Bh, BIOS break

Dit is geen echte BIOS call. De toetsenbord interrupt (INT 09h) voert deze instructie uit, als de toetscombinatie Ctrl-ScrollLock wordt gevonden. Dit is zoals reeds vermeld de normale stop-toets, ofschoon onder DOS de meer gebruikelijke combinatie Ctrl-C ook werkt. INT 1Bh wijst normaliter naar een IRET en doet dus niets.

INT 1Ch, timer tick

Alweer geen echte BIOS call. De timer-interruptroutine voert deze instructie aan het einde uit, nadat de klok-tellers zijn bijgewerkt. De machine voert dus iedere clock-tick een INT 1Ch uit. Ook de INT 1Ch wijst meestal naar een IRET.

INT 1Dh, video parameters

Deze INT wordt alleen gebruikt om een pointer op een vast adres te creëren. Hij wijst naar een tabel in het BIOS, waarin de parameters voor de 6845 op de displayadapter zijn opgeslagen.

INT 1Eh, disk parameters

Voor deze INT geldt hetzelfde als voor INT 1Dh: nu echter voor een tabel met parameters voor de floppy disk controller.

INT 1Fh, CGA extended generator

Alweer een pointer, zie bij INT 10h, het video BIOS. (Wat gaat het zo vlot hè).

De volgende keer...

wordt het echt tijd om het BIOS aan het werk te zetten. Dat zal gebeuren met DEBUG, het standaard ontluisprogramma dat bij ieder DOS wordt meegeleverd. DEBUG kan namelijk misbruikt worden als simpele assembler. Prettige zomervakantie.

Nico de Vries

DOS65 Geluidsprint

De computersystemen die heden ten dage door hobbyisten worden gebruikt zijn voorzien van een mogelijkheid tot het maken van muziek. Voor de DOS65 is nu een geluidsprint ontwikkeld om een "af" systeem te krijgen. Deze geluidsprint kan zonder meer aangesloten worden op de uitbreidingsbus van de DOS65 computer. Het enige dat verder nog nodig is, is een stuk software dat de geluidsprint ondersteunt, b.v. in de vorm van het BASICODE-3 vertaalprogramma.

Het geheel bestaat uit een geluids-IC, de Sound Interface Device (SID) 6581 (wel bekend bij Commodore gebruikers), een tweetal TTL poorten, een Opamp en nog wat componenten.

Even een opsomming van wat de sound controller 6581 in huis heeft:

- 3 programmeerbare, onafhankelijke oscillatoren.
- 4 mengbare golfsoorten per stem.
- 3 mengbare filters (hoog-, laag-, en bandfilter).
- Per stem een envelope generator.
- 2 te koppelen ringmodulatoren.
- Vervormingsmogelijkheid voor externe signaalbronnen.
- 2 8-bits analoog-digitaal omzetters.

M.b.v. de Opamp is het mogelijk om het signaal dat de SID produceert te versterken om het vervolgens aan een luidspreker aan te bieden. Indien de monitor een versterker heeft, is het niet nodig de schakeling rond deze Opamp te bouwen.

De twee TTL poorten zijn nodig voor adresdecodering t.b.v. de SID. Op de NOR poort worden de adreslijnen A12, A11, A10 en A8 aangeboden welke allen laag moeten zijn om een hoog aan de uitgang te krijgen en op de NAND poort worden de adreslijnen A15, A14, A13, A9, A7, A6 en A5 aangeboden welke hoog moeten zijn, waardoor uiteindelijk op de adressen $E2E0-$E22FF het CS signaal actief kan zijn. Via de adreslijnen A0-A4 worden de registers in de SID geadresseerd, mits de SID met CS geselecteerd is.

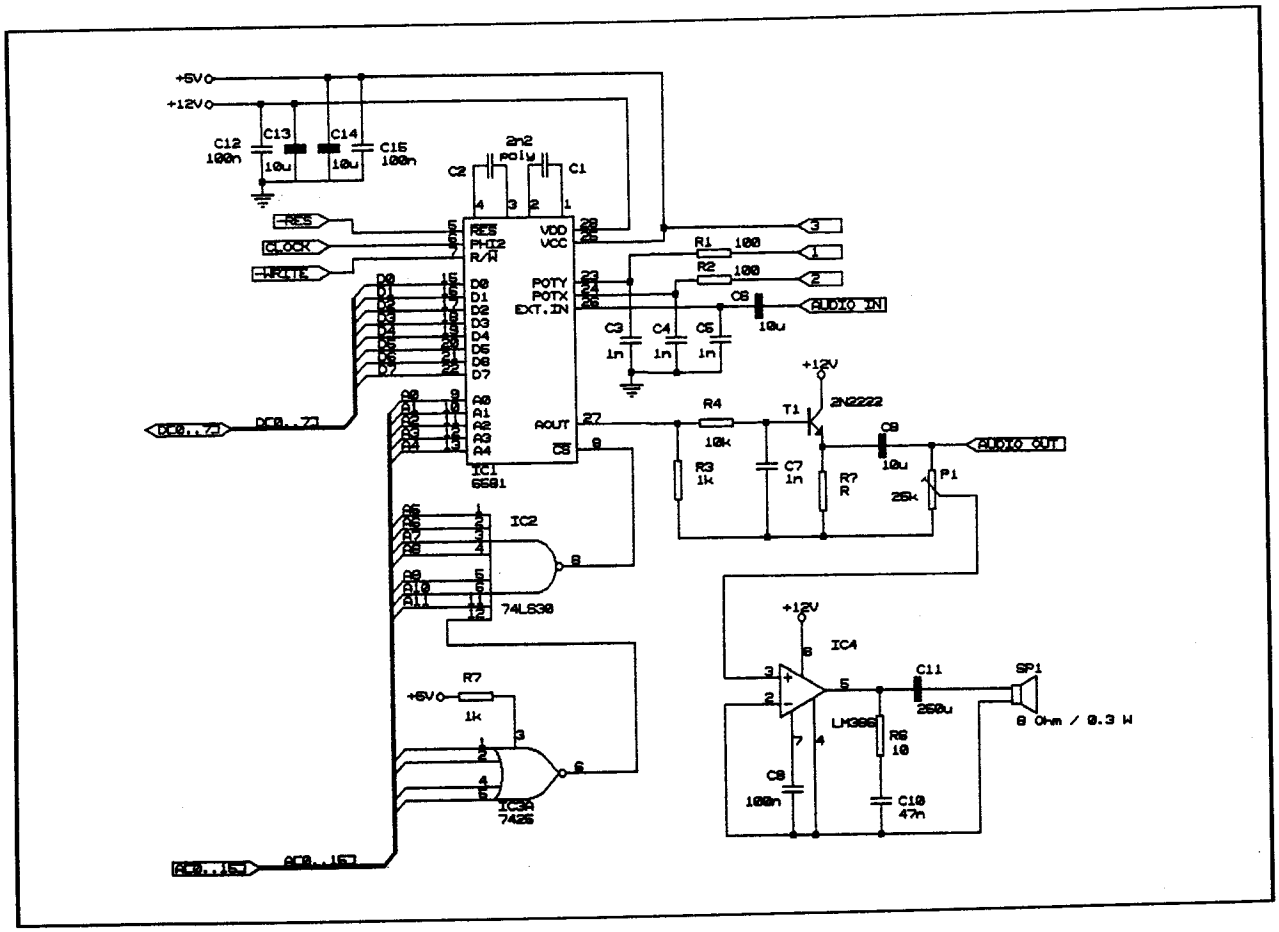


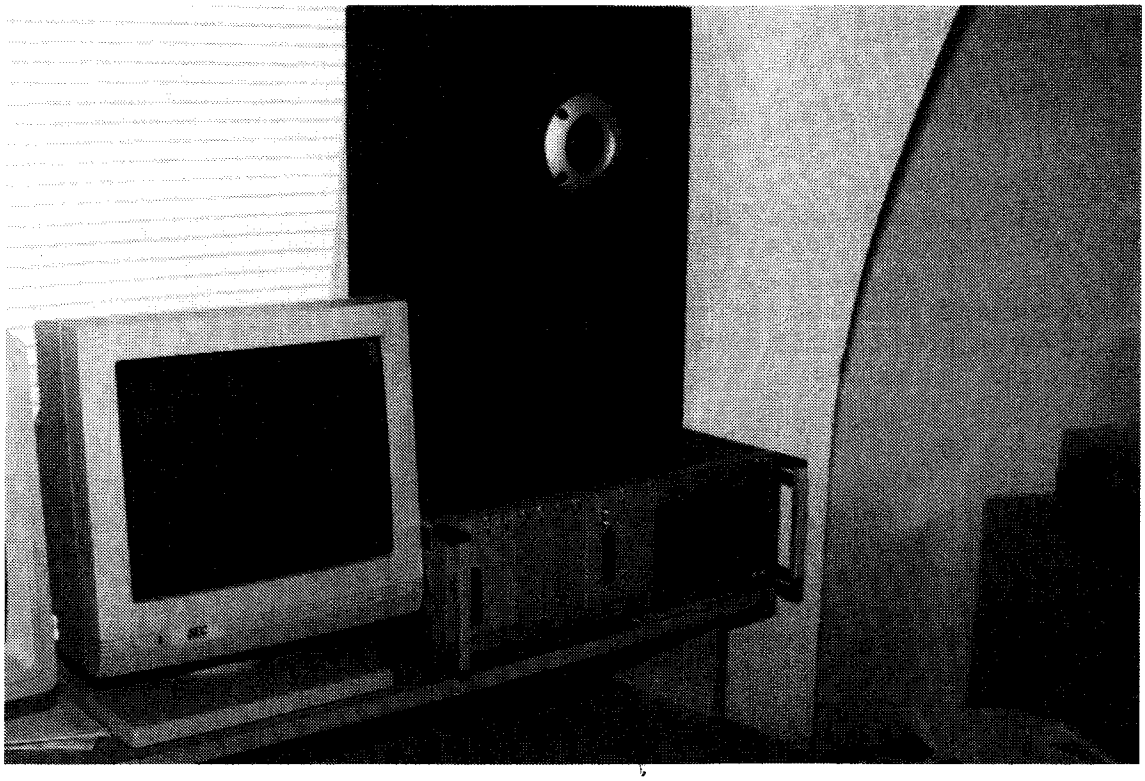
Fig. 1: Schakeling rond de SID-6581 t.b.v. DOS-65

De datalijnen D0-D7 zijn verbonden met de databus van de processor. De condensatoren C1 en C2 worden gebruikt voor de frequentie-filters, die in de SID zijn geïntegreerd. De weerstanden R1 en R2 beveiligen de ingang, anders kan de ontlaadtrap aan de ingang kapot gaan, omdat de spanning bij het ontladen van de condensatoren te groot wordt.

Voor verdere informatie omtrent de SID 6581 wordt verwezen naar o.a. de reference guide van de commodore-64.

Voor de schakeling is een printje ontworpen. Deze print kan eventueel via de club geleverd worden. De print is dan ongeboord. U kunt dit printje bestellen door *f* 15,00 te storten op giro 3757649 ten name van KIM-Gebruikers Club Nederland te Enschede onder vermelding van SID-print. Het BASICODE-3 vertaalprogramma ondersteunt de geluidsprint. Wilt u behalve de geluidsprint ook een schijf met het BASICODE-3 vertaalprogramma, dan moet u *f* 7,50 extra overmaken. Daar de printen aangemaakt worden als er ongeveer 5 stuks besteld zijn, kan het enkele weken duren voordat u de bestelde print ontvangt. Om de geluidsprint op te bouwen, heeft u voor nog ongeveer *f* 75,00 aan onderdelen nodig.

Frank Bens



Route naar overmorgen?

Om een goede printplaat te kunnen vervaardigen is uiteraard een goed ontwerp nodig. Een "goed ontwerp", dat bestaat uit een goed schema en... uit een goede print-layout.

Prints ("Printed Circuit Boards", oftewel "PCB's") layouten is vaak een tijdrovende en complexe slag in het complete ontwerpproces van een electronisch apparaat. Gelukkig maar dat er behoorlijke computerprogramma's zijn die de PCB-layout kunnen ondersteunen of zelfs helemaal voor hun rekening kunnen nemen.

"O ja? Is er wel programmatuur dat volautomatisch een PCB-design kan genereren?", vraagt u nu natuurlijk. Ja, die programmatuur is er! Alleen... de benodigde apparatuur is verschrikkelijk duur en de software idem dito. Meestal wordt de computer wel gebruikt in het ontwerpproces. Soms als vervanger van het oude vertrouwde plakplastic en afwrijfsymbolen, soms ook als volwaardige probleemoplosser. Het enige dat de ontwerper dan nog hoeft te doen, is de onderdelen op de printplaat te definiëren en de computer legt vervolgens volgens een netlist de verbindingen. Hierbij moet de ontwerper dus nog heel wat meer doen dan bijvoorbeeld bij het nog in ontwikkeling zijnde transputersysteem dat schuil gaat achter de illustere naam "Cleopatra". Dat systeem maakt gebruik van één T800 en drie T414 transputers. Verder heeft 't ding een 19 inch high-resolution real-color kleurenbuis, een mammoet harddisk met een capaciteit van enkele GigaBytes, een Megje of wat aan RAM en nog een heleboel leuk ander speelgoed. Je kunt er een standaard schema instoppen, een aantal specificaties opgeven, het ding een nachtje laten pruttelen en klaar is Cleopatra! Een kant en klaar PCB ontwerp ligt op de plotter. Oplossingsgraad: beter dan 98% bij een stukdichtheid van ca. 75% voor schakelingen in conventionele- en een stukdichtheid van ca. 85% voor SMD techniek. In 80% van de gevallen haalt het ding de 100% oplossing...

De gemiddelde oplossingsgraad van goedkope autorouters ligt ergens tussen de 70 en 90%. Een 100% oplossing behoort hier veelal tot de uitzonderingen. Na enige tijd met verschillende autorouters geëxperimenteerd te hebben was mij duidelijk dat er blijkbaar verschillende algoritmen gebruikt werden door de verschillende routing pakketten. Met name het routing algoritme van smArtWork intrigeerde mij. De gemiddelde oplossingsgraad van dit pakketje was bijzonder hoog, zeker als je aanmerking neemt dat het pakket niet in staat is zelf doormetaliseringen of via's te definiëren.

De ANWB onder de autorouters

Enig literatuuronderzoek, gecombineerd met wat eigen experimenten, maakte al snel duidelijk dat smArtWork een uitgebreide versie van "Lee's routing algorythm" gebruikt. Dit algoritme, dat zijn naam te danken heeft aan zijn uitvinder, Dr. L.Y.Lee, is ondanks zijn eenvoud bijzonder doeltreffend. Het gaat uit van een ruitjesvel waarin verschillende elementen het complete ontwerp omschrijven. Alle mogelijke PCB-elementen zijn opgenomen in figuur 1.

Het algoritme gebruikt een tweedimensionaal array waarin het ontwerp bijgehouden wordt, een array waarin de route gezocht wordt. Tevens wordt er een FIFO-queue bijgehouden waarin coördinaten opgeslagen kunnen worden.

Om een lijn tussen twee punten te vinden gaat het algoritme uit van één punt. Dit punt dopen we bij deze maar "source". Het punt waar de verbinding naar toe getrokken moet worden wordt aangeduid als de "target". Allereerst wordt gecontroleerd of de noorderbuurman van de source toevallig de target is. Is dat het geval, dan is de verbinding gevonden en kan hij ook daadwerkelijk getrokken worden. Is dat niet het geval én is de cel ten noorden van de source leeg, dan worden diens coördinaten opgeslagen in de FIFO-queue en wordt in de betreffende cel in het werk-array de code "zuid" gezet. Vervolgens handelt het algoritme dezelfde actie nogmaals af, maar dan met de cel ten oosten van de source. In deze cel wordt dan de waarde "west" gezet. Parallel hieraan worden ook de buurmannen ten zuiden en ten westen van de source behandeld. Het algoritme neemt

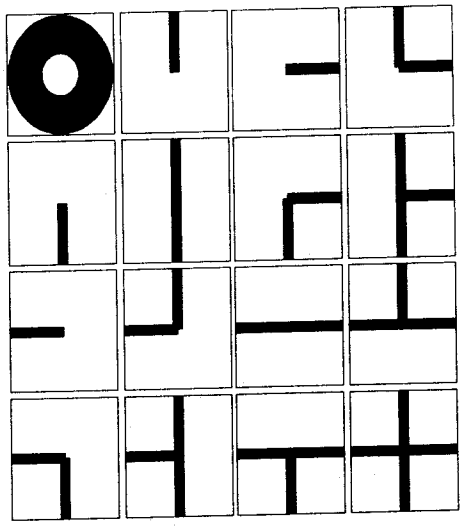


Fig. 1: Basis elementen van Lee's routing algorythm

dan als source de eerste cel in de FIFO-queue en herhaalt het gehele proces. Dit gaat door totdat de queue leeg is (er is dan geen verbinding mogelijk) óf totdat een verbinding gevonden is. Eén en ander wordt nader toegelicht in figuur 2.

Vanuit de target kan nu de weg terug gevonden worden naar de source door simpelweg als een soort kleinduimpje de achtergelaten aanwijzingen terug te volgen en, afhankelijk van de plaats van in- en uittreden van de cel, de cel te vullen met één van de elementen uit figuur 1.

Dit "backtracen" wordt meestal in het PCB-array zelf gedaan in plaats van in het werkarray.

Routes via Manhattan

Lee's routing algorythm garandeert dat er een route gevonden wordt indien er een mogelijke route bestaat. Tevens garandeert het de kortste route volgens de zogenaamde "Manhattan distance". De Manhattan distance is gedefinieerd als de som van alle verticale en alle horizontale afstanden in de totale route.

Lee's algorythm is, mits goed toegepast, redelijk snel. Tevens minimaliseert het algoritme het aantal doormetaliseringen of via's in de printplaat. Logisch, want het algoritme kan ze niet zelf leggen...

Uren, dagen, maanden, jaren...

Nadelen heeft Lee's routing algorythm natuurlijk ook. Allereerst is het opslaan van de PCB-data in een tweedimensionaal array niet de meest handige

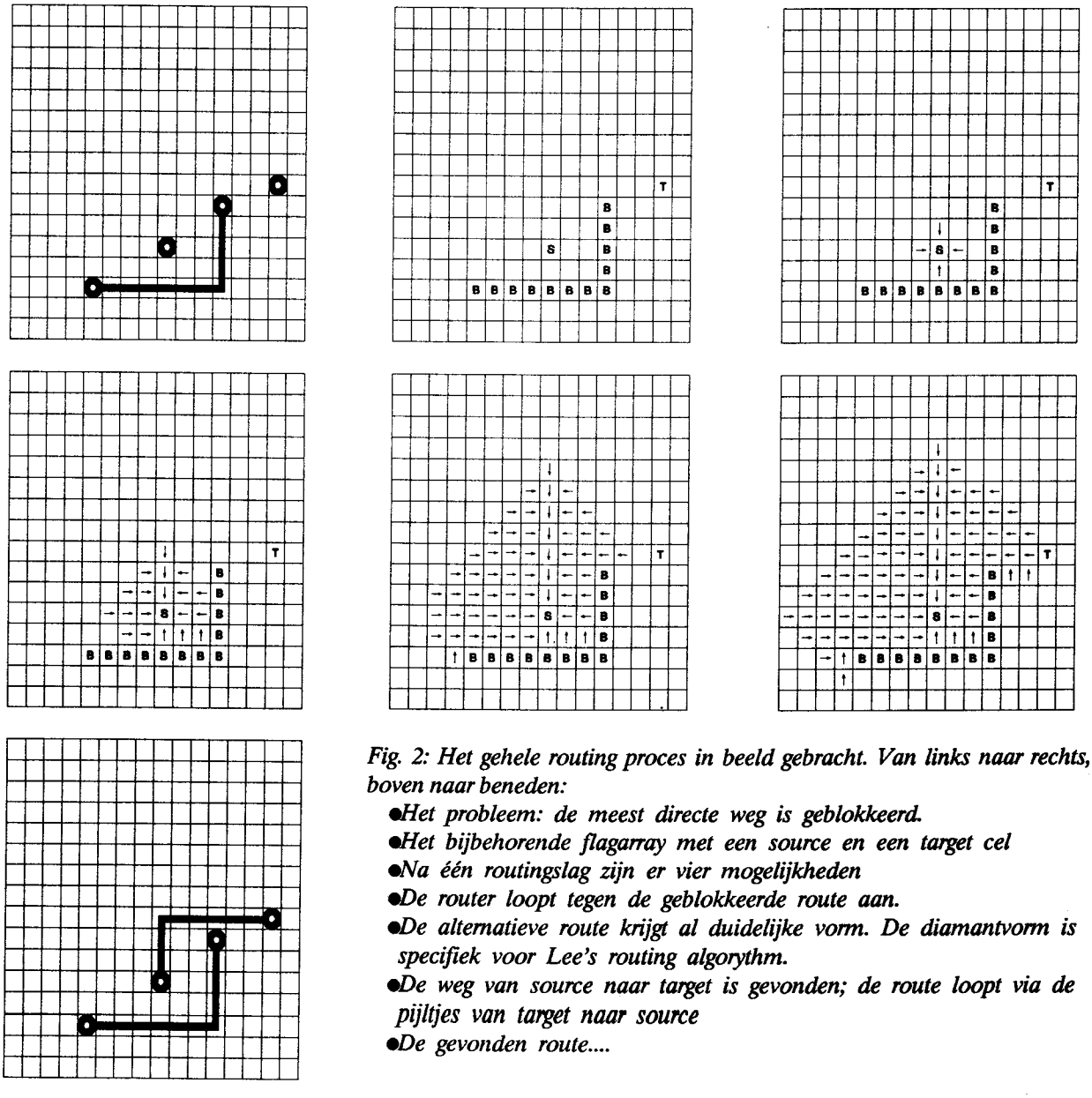


Fig. 2: Het gehele routing proces in beeld gebracht. Van links naar rechts, boven naar beneden:

- *Het probleem: de meest directe weg is geblokkeerd.*
- *Het bijbehorende flagarray met een source en een target cel*
- *Na één routingslag zijn er vier mogelijkheden*
- *De router loopt tegen de geblokkeerde route aan.*
- *De alternatieve route krijgt al duidelijke vorm. De diamantvorm is specifiek voor Lee's routing algorythm.*
- *De weg van source naar target is gevonden; de route loopt via de pijltjes van target naar source*
- *De gevonden route....*

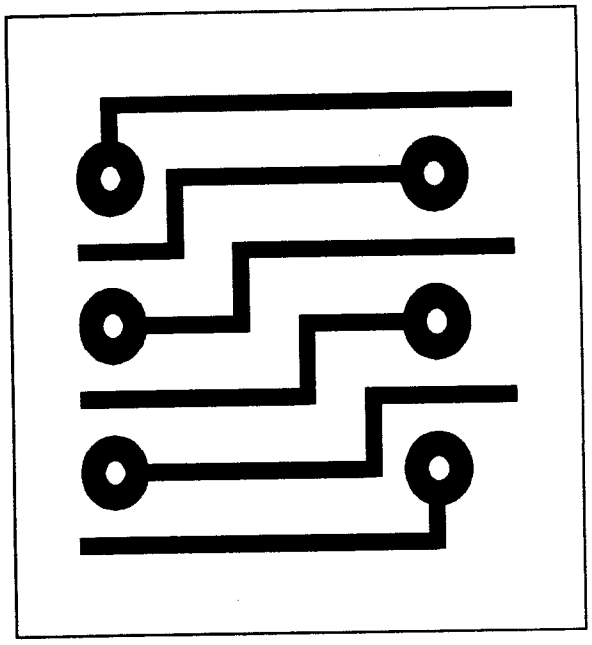


Fig. 3: Het "lopende hoek" probleem

manier bij het verplaatsen van onderdelen. "Rubberbanding" is door deze manier van opslag bijna onmogelijk. Geheugen capaciteit is ook belangrijk voor Lee. In de praktijk wordt er vaak gebruik gemaakt van een zogenaamd "50-mil grid". Dat is een rooster met vakjes van 0.050 inch, oftewel een half E. Dat betekent dan meteen dat de autorouter precies één lijn tussen de pootjes van een standaard TTL-IC door kan wurmen. Voor een printplaat van 160 x 100 mm (een EuroCard formaat) is dat 9,75 kByte per zijde. Op een gemiddelde PCB zijn er 5 verschillende layers nodig (boor- en koper layers van de componenten- en de soldeerzijde plus een stuklayer). Dat is dus voor een eenvoudige EuroCard al 48,75 kByte aan layers. Dan komen er voor het routen nog twee arrays bij. Dat maakt dan in totaal 68.25 kByte. Nog afgezien van de ruimte, benodigd voor de FIFO-queue die behoorlijk groot kan worden. Voor een personal computer valt dat natuurlijk hard mee. Maar, wil men bijvoorbeeld ook SMD-logica aankunnen, dan moet het layout rooster veel fijner zijn. Een 10-mil grid bijvoorbeeld. Voor de EuroCard hebben we dan... 1,7 MByte nodig!

De snelheid loopt ook vrij hard terug bij verfijning van het grid en/of vergroting van de printplaat. Voor een printspoor tussen twee punten die onderling n eenheden volgens de Manhattan distance van elkaar liggen, is de benodigde tijd evenredig aan n^2. Voor korte afstandjes valt dat natuurlijk reuze mee, maar gaan we van het 50-mil grid terug naar een 10-mil rooster... dan kan in plaats van 3 seconden opeens 2 uur nodig zijn om dezelfde route te leggen!

Optimalisatie

Aan het grondalgoritme kan nog lekker veel gesleuteld worden. Om de lijnen meer rechtstreeks te laten open zouden hoeken van 45 graden bijvoorbeeld een welkome verbetering vormen. Daarvoor moet de volgorde van het onderzoeken van de verschillende buurcellen afhankelijk zijn van de richting waarin gezocht moet worden. Daarnaast moet het "L"-vormige element vervangen worden door een schuine streep tussen de twee aansluitpunten.

Een ander probleem bij het standaard algoritme is het probleem van "de lopende hoeken". Stel dat we bijvoorbeeld een memory-array willen routen. Zoals in figuur 3...

We krijgen dan het probleem dat de hoeken, nodig voor de verspringing van een half E, ook ruimte innemen. Een mogelijke oplossing kan gezocht worden in het leggen van twee diagonalen in één cel.

Leuke stof dus om zelf eens mee aan het werk te gaan. Wel, op het bulletinboard van de vereniging heb ik de grond-bouwstenen van mijn onderzoek maar vast voor u klaar gezet. Het geheel bestaat uit twee programma's, geschreven in turbopascal 5.0. Beide programma's zijn beide identiek, op één ding na. Eén van beide programma's maakt gebruik van de techniek van afwijkende volgorde bij het zoeken naar de kortste route. Het andere volgt simpelweg het routing algoritme.

Joost Voorhaar.

Van de voorzitter, of beter: van de bestuurstafel

Want het tweede deel van de titel lijkt me beter. Ondergetekende heeft eigenlijk altijd al gevonden, dat de leden te weinig horen over wat het bestuur allemaal uitspookt op bestuursvergaderingen en wat daar zo al besloten wordt. Vandaar dat dit verhaaltje in het vervolg iets anders gaat heten. Tenslotte is het bestuur er voor de leden en de club, en niet ter ere van zichzelf niet waar?

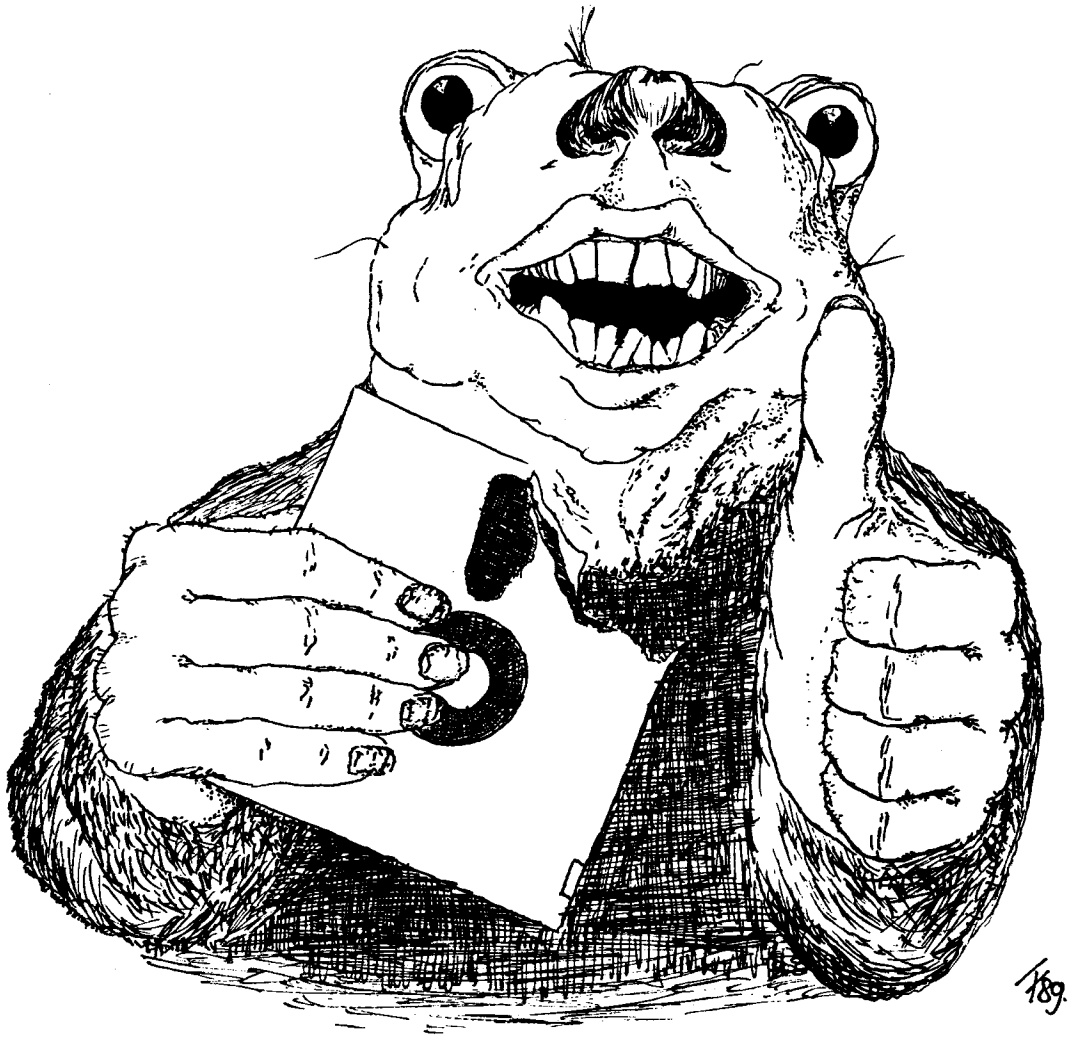
Dus: wat is er laatste maal over de tafel gegaan bij het vergaderen. Van alles. Onder andere een groot aantal ideeën over hoe we het in de toekomst willen gaan aanpakken. Een van de uitvloeiselen daarvan is, dat we voor het eerst een heus persbericht voor de locale pers hebben, waarin de bijeenkomst van onze club wordt aangekondigd. En tweede ding is, dat we ook niet-leden op de bijeenkomst uitnodigen, zij het pas na de lunch. Zo hebben we dan voor de lunch de gebruikelijke lezing en de gebruikelijke privacy.

De bijeenkomst in mei belooft in veel opzichten een mooie te worden. Als eerste hebben een nieuwe, centraal gelegen locatie: Almere. Wat we ook als nieuwtje hebben is een themadag, met wel twee lezingen en compleet met een klein commercieel lokkertje. Kijk maar in de uitnodiging. U komt toch ook, ook als het mooi weer is?

Verder zijn Geert Stappers en Joost Voorhaar zeer actief in het opsporen van potentiële sponsors. De club timmert dus voor zijn doen behoorlijk aan de weg. Dat moet ook, want het ledental is wat aan de bescheiden kant. Over dat ledental gesproken: dat zit weer in de lift, een teken dat de activiteiten van bestuur en redactie hun vruchten afwerpen.

Met dit nummer moet u het even doen, want we gaan zomervakantie houden. Na de zomer gaan we er weer volop tegenaan. De soldeerbouten en de PC's mogen even uit. Prettige vakantie, met naar ik hoop het weer erbij dat u hebben wilt.

Uw voorzitter, Nico de Vries



Informatie.

De μP Kenner (De microprocessor Kenner) is een uitgave van de KIM gebruikersclub Nederland. Deze vereniging is volledig onafhankelijk, is statutair opgericht op 22 juni 1978 en ingeschreven bij de Kamer van Koophandel en Fabrieken voor Hollands Noorderkwartier te Alkmaar, onder nummer 634305. Het gironummer van de vereniging is 3757649.

De doelstellingen van de vereniging zijn sinds 1 januari 1989 als volgt geformuleerd:

- Het vergaren en verspreiden van kennis over componenten van microcomputers, de microcomputers zelf en de bijbehorende systeemsoftware.

- Het stimuleren en ondersteunen van het gebruik van micro-computers in de meer technische toepassingen.

Om deze doelstellingen zo goed mogelijk in te vullen, wordt onder andere 5 maal per jaar de μP Kenner uitgegeven. Verder worden er door het bestuur per jaar 5 landelijke bijeenkomsten georganiseerd, beheert het bestuur een Bulletin Board en wordt er een softwarebibliotheek en een technisch forum voor dediverse systemen in stand gehouden.

Landelijke bijeenkomsten:

Deze worden gehouden op bij voorkeur de derde zaterdag van de maanden januari, maart, mei, september en november. De exacte plaats en datum worden steeds in de μP Kenner bekend gemaakt in de rubriek Uitnodiging.

Bulletin Board:

Voor het uitwisselen van mededelingen, het stellen en beantwoorden van vragen en de verspreiding van software wordt er door de vereniging een Bulletin Board beschikbaar gesteld.

Het telefoonnummer is: 053-303902.

Software Bibliotheek en Technisch Forum:

Voor het beheer van de Software Bibliotheek en technischeondersteuning streeft het bestuur ernaar zgn. systeemcoordinators te benoemen. Van tijd tot tijd zal in de μP Kenner een overzicht gepubliceerd worden. Dit overzicht staat ook op het Bulletin Board.

Het Bestuur:

Het bestuur van de vereniging wordt gevormd door een dagelijks bestuur bestaande uit een voorzitter, een secretaris en een penningmeester en een viertal gewone leden.

Nico de Vries (voorzitter)
Mari Andriessenrade 49
2907 MA Capelle a/d IJssel
Telefoon 010-4517154

Mick Agterberg (secretaris)
Davidvosstraat 29
1063 HV Amterdam
Telefoon 020-131538

Jacques H.G.M. Banser (penningmeester)
Haaksbergerstraat 199
7513 EM Enschede
Telefoon 053-324137

Gert van Opbroek (Redactie μP Kenner)
Bateweg 60
2481 AN Woubrugge
Telefoon 01729-8636

Jan D.J. Derksen
Ed Verkadestraat 9-1
7558 TH Hengelo
Telefoon 074-770970

Geert Stappers
Engelseweg 7
5825 BT Overloon
Telefoon 04788-1279

Ton Smits
De Meren 39
4731 WB Oudebosch

Ereleden:

Naast het bestuur zijn er een aantal ereleden, die zich in het verleden bijzonder verdienstelijk voor de club hebben gemaakt:

Erevoorzitter:
Siep de Vries

Ereleden:
Mevr. H. de Vries van der Winden
Anton Mueller