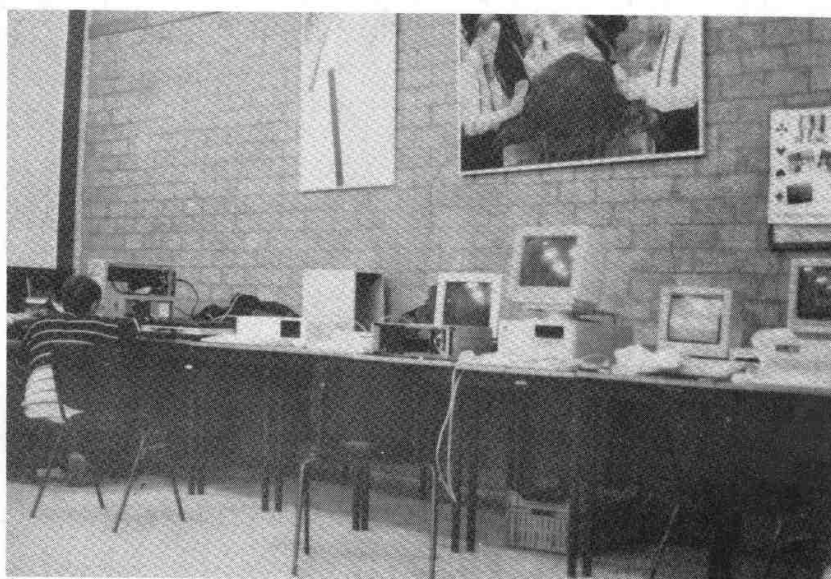




Zestiende jaargang nr. 5
December 1992
79



In dit nummer o.a.:

HAL is back!
TCP/IP, wat is dat?
Programma(on)gein
Nieuws over de KGN68k
Teletekst op uw PC (deel 2)

Inhoudsopgave

De μ P Kenner

Nummer 79, december 1992
 Verschijnt 6 maal per jaar
 Oplage: 250 stuks
 Druk: FEBO Offset, Enschede

De redactie:

Joost Voorhaar
 Gert van Opbroek
 Geert Stappers
 Nico de Vries

Eindredactie:

Joost Voorhaar

Vormgeving:

Joost Voorhaar
 Nico de Vries

Redactieadres:

Joost Voorhaar
 Jekerstraat 67
 7523 VP Enschede

De μ P Kenner nummer 80 verschijnt op
 20 februari 1993.

Kopijsluitingsdatum voor nummer 80 is
 vastgesteld op 6 februari 1993.

Vereniging

Uitnodiging voor de clubbijeenkomst	5
Verslag van de ledenvergadering d.d. 28 november 1992	6
Wijziging van het Huishoudelijk Reglement	7
Werkgroep DOS65	7
Concept begroting 1993	8
De huidige status van KGN68k	37
Van de bestuurstafel	45

Algemeen

Redactioneel	4
A problem in the making	12
Nieuwtjes en andere wetenswaardigheden (1)	22
Barcodes	23
Nieuwtjes en andere wetenswaardigheden (2)	28
Netwerken op het BBS	31
Inleiding in computer-programma's	34
Hello World in PostScript	36

Software

To Share Or Not To Share, That's The Question	9
---	---

Hardware

Teletext (deel 2)	29
-------------------------	----

Systemen

De Motorola 68030; het hart van KGN68k (deel 3)	13
---	----

Datacommunicatie

Geavanceerde datacommunicatie deel 2	41
--	----

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

Kopij voor het blad dient bij voorkeur van de leden afkomstig te zijn. Deze kopij kan op papier, maar liever in machine-leesbare vorm opgestuurd worden aan het redactieadres. Kopij 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.

De redactie houdt zich het recht voor kopij zonder voorafgaand bericht niet of slechts gedeeltelijk te plaatsen of te wijzigen. Geplaatste artikelen blijven het eigendom van de auteur en mogen niet zonder diens voorafgaande schriftelijke toestemming door derden gepubliceerd worden, in welke vorm dan ook.

De redactie noch het bestuur kan verantwoordelijk gesteld worden voor toepassing(en) van de geplaatste kopij.

Redactioneel

Allereerst willen wij een ieder een voorspoedig 1993 toewensen. Op het moment dat we dit zitten te schrijven had deze μ P Kenner al weer in de brievenbus moeten liggen. Voor de feestdagen was het echter stilte met de kopij. Toen dachten we nog: dat wordt overwerken tijdens de feestdagen. Maar gelukkig hebben we die bij de haard kunnen vieren. Ja, het liep zo'n storm met de kopij dat we de feestdagen ongestoord hebben kunnen vieren. De feestdagen werden zodoende feestdagen maar met de kopij was het triest gesteld. Er was onvoldoende om een blad naar behoren te kunnen vullen.

Het wordt nu toch echt tijd dat de lezers gaan beseffen dat een blad alleen kan uitkomen als een ieder z'n steentje daartoe bijdraagt. Het moet na het artikel van Gert van Opbroek in augustus toch niet meer zo moeilijk zijn om eens een pagina tekst of een ingezonden brief met ideeën op te sturen naar de redactie. Anders worden we gedwongen om een keer over te slaan en dat kan nooit de bedoeling zijn. Althans niet van ons. Momenteel wordt de redactie gevormd door de bestuursleden dat wordt echt te veel. Ook wij hebben een leven naast de KGN. Besturen en een blad uitbrengen wordt te veel van het goede. Het afgelopen jaar is er herhaaldelijk een noodkreet uitgebracht. Nu is het aan de leden om daar eens op in te gaan.

De redactie is namelijk bang dat we zo langzamerhand in een neergaande spiraal terecht zijn gekomen die voor de KGN hele vervelende gevolgen zou kunnen hebben. Een belangrijk deel van de pakweg 150 leden die we hebben is namelijk lid vanwege het blad. Bovendien probeert het bestuur door middel van het blad nieuwe leden te werven. Dat betekent dat het blad gezien moet worden als het visitekaartje van de vereniging. Dit visitekaartje hoort correct verzorgd te worden hetgeen betekent dat het blad op tijd uit moet komen en een redelijk constante kwaliteit en omvang behoort te hebben. Is dat niet het geval, dan ben ik er van overtuigd dat er leden zullen zijn die om die reden hun lidmaatschap op zullen zeggen. Daardoor krijg je minder leden, dus minder potentiële schrijvers en geld waardoor het nog moeilijker wordt het blad uit te geven waardoor er nog meer leden zijn die de KGN de rug toekeren

waarna na een poosje de KGN alleen nog bestaat uit de redactie en een aantal leden die lid zijn om andere redenen dan het blad alleen.

Als jullie nu voor voldoende kopij zorgen, dan kan de redactie garanderen dat er in 1993 minimaal 5 nummers van de μ P Kenner op tijd en met een behoorlijke omvang uit zullen komen. Als ieder lid nu één keer dit jaar een stukje kopij van 4 pagina's inlevert, dan hebben we in totaal $150 * 4 = 600$ pagina's waarmee we $600 / 40 = 15$ nummers = 3 jaar !! vooruit kunnen. En heus, 4 pagina's schrijven is echt niet zo moeilijk, probeer het eens!

Jullie kunnen voor 20 februari a.s. je schrijfsels kwijt op het BBS of bij een van de bestuursleden.

Voor jullie ligt een blad dat ook nu weer met de nodige pijn maar wel weer met zorg is samengesteld.

Welnu, voor jullie ligt een blad dat ook nu weer met de nodige pijn maar wel weer met zorg is samengesteld. We proberen altijd weer om het zo interessant mogelijk te maken. Van de lezers hebben we inmiddels via de enquête vernomen dat dat meestal wel lukt. Over de uitkomst van de enquête zal in een volgend nummer wat meer verslag worden gedaan. Dit jaar kunnen we ook wat meer kopij verwachten rondom het KGN68k project. Als alles goed gaat met de DOS65 werkgroep gaat daar ook weer het een en ander gebeuren en dat levert mogelijk ook weer leuke kopij op voor ons clubblad.

Ja, zal je zeggen wat blijft er dan nog voor de leden over. Meer dan genoeg. We willen best wel een paar voorbeelden noemen. Iets over Linux, een goed idee over gebruikerssoftware, perikelen over Girotel of het gebruik van BBS-en. Maar ook over de aanschaf van computers of onderdelen. Een en ander mag zowel in positieve als in negatieve zin. Maar misschien kun je er iets komisch van maken. Als je twijfelt heb je in ieder geval een idee en laat dan eventueel iets van je horen, we kunnen dan misschien helpen.

Ondanks dat er in onze voorgaande woorden een stukje pessimisme klonk blijven wij er vertrouwen in houden. We wensen u dan ook veel leesplezier toe met deze μ P Kenner.

De redactie

Uitnodiging voor de clubbijeenkomst

Datum: 16 januari 1993
 Locatie: Nieuwe kantine FORBO-Krommenie
 Industrieweg 12
 1566 JP Assendelft
 Telefoon: 075-291911
 Thema: Toneelverlichting met DOS65
 Entree: Gratis!

Routebeschrijving

Per auto:

1. Uit de richting Amsterdam: Coentunnel door en de Coentunnelweg helemaal afrijden. Aan het einde rechtsaf (water aan de linkerzijde). Dan de 1^e afslag rechtsaf, richting Uitgeest-Alkmaar. Doorrijden tot aan de stoplichten. Linksaf de spoorbaan over.
2. Na 75 meter linksaf: Industrieweg. Links aanhoudende komt men dan op het FORBO-terrein.
3. Uit de richting Alkmaar: Snelweg Alkmaar-Haarlem. Afslag Uitgeest-Zaandam. Bij kruising linksaf. Bij de 3^e stoplichten rechtsaf, de spoorbaan over. Verder volgens punt 2.

Per trein:

Station Krommenie-Assendelft. Rechtsaf tot over de spoorwegovergang. Zie verder punt 2.

Programma:

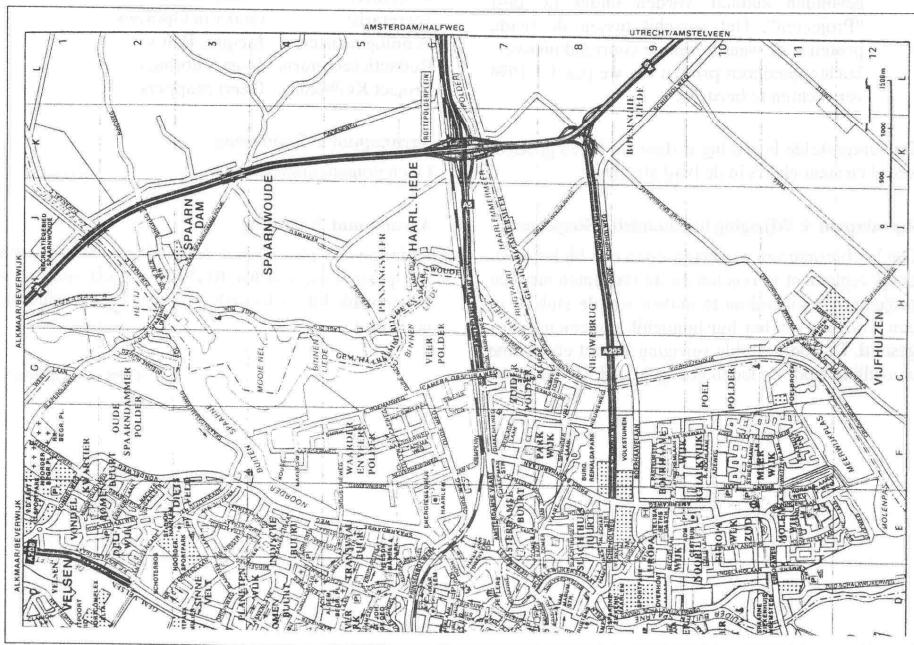
- 9:30 Zaal open. Ontvangst met koffie.
- 10:00 Opening door de voorzitter en verwelkoming door de gastheer: Co Filmer.
- 10:10 Diashow over FORBO-Krommenie.
- 10:30 Koffiepauze
- 11:00 Vordracht van Jaap Prenger en Antoine Megens over besturing van toneelverlichting met behulp van DOS65.
- 12:00 Forum en markt
- 12:30 Lunch, aangeboden door FORBO-Krommenie.

Aansluitend het informele gedeelte met de mogelijkheid om andermans systemen te bewonderen en Public Domain software uit te wisselen. U en uw systeem zijn uiteraard van harte welkom.

- 17:00 Sluiting.

Let op:

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.



Verslag van de ledenvergadering d.d. 28 november 1992

Agendapunt 1: Opening, vaststelling agenda

Aanwezig zijn 16 leden plus het voltallige bestuur. Er zijn afmeldingen ontvangen van de volgende leden:

Adri Hankel
Gerard Willemse

De voorgestelde agenda wordt goedgekeurd.

Agendapunt 2: Conceptbegroting 1993

De penningmeester heeft een concept-begroting opgesteld die aan de leden is voorgelegd. Op twee punten van de begroting is enige toelichting gegeven:

- 1: De contributie voor 1993 wordt verhoogd tot f 60,- (was f 55,-). Het komt misschien wat vreemd over om twee jaar na elkaar de contributie te verhogen. Welnu, de verhoging voor het jaar 1992 was noodzakelijk om de club financieel gezond te houden. De verhoging voor 1993 heeft een andere reden. De bijeenkomsten zullen in de toekomst door leden gratis bezocht kunnen worden (toegang was f 10,- m.u.v. de bijeenkomst van november). Omdat bijeenkomsten echter toch geld kosten, wordt dit gedekt door een verhoging van de contributie.
- 2: Op de begroting staat een post "Op peil brengen voorraad". Dit betreft uitgaven die gedaan moeten worden voor het in productie nemen van printen voor KGN68k. Uiteraard staan hier ook weer inkomsten tegenover die gevonden kunnen worden onder de post "Projecten". Het verschil tussen de beide posten is de waarde van de voorraad niet-verkochte goederen printen die we per 1-1-1994 verwachten te bezitten.

De voorgestelde begroting is door de leden goedgekeurd en staat elders in de blad afgedrukt.

Agendapunt 4: Wijziging huishoudelijk Reglement

Om het bestaan van projectgroepen ook bij huishoudelijk reglement te regelen en de resultaten van een projectgroep eigendom te maken van de club, is er een wijziging van het huishoudelijk reglement voorgesteld. De voorgestelde wijziging is, met een kleine aanvulling, door de leden goedgekeurd. De tekst van

de goedgekeurde wijziging staat elders in dit blad afgedrukt.

Agendapunt 4: Verkiezing kascontrole-commissie 1993

Rinus Vleesch Dubois en Henk Speksnijder hebben zich beschikbaar gesteld om zitting te nemen in de kascontrole-commissie voor 1993. De aanwezige leden gingen hiermee accoord.

Agendapunt 5: Verkiezing bestuursleden

Er zijn tot op heden geen kandidaten voor een bestuursfunctie aangemeld zodat de voorgestelde bestuursleden herkozen zijn. Dit houdt wel in dat we momenteel 2 vacatures binnen het bestuur hebben en iedereen wordt opgeroepen uit te zien naar mogelijke kandidaten, bij voorkeur voor de volgende bestuurstaken:

- 1: Clubactiviteiten en Public Relations
- 2: DOS-65

Mochten er kandidaten gevonden worden, dan zullen deze op de volgende ledenvergadering (maart 1993) verkozen en/of benoemd kunnen worden.

Het bestuur bestaat momenteel uit de volgende personen:

Voorzitter:	Tonny Schäffer
Secretaris:	Gert van Opbroek
Penningmeester:	Jacques Banser
Redactiesecretaris:	Joost Voorhaar
Project KGN68k:	Geert Stappers

Agendapunt 6: Rondvraag

Geen vragen, opmerkingen.

Agendapunt 7: Sluiting

Sluiting vond plaats rond de middag waarna er nog een presentatie van het KGN68k-project volgde en het gebruikelijke informele gedeelte van een clubbijeenkomst.

Gert van Opbroek

Wijziging van het Huishoudelijk Reglement

Het Huishoudelijk Reglement, zoals gepubliceerd in de μ P Kenner 58 (oktober 1988) met de in de μ P Kenner 60 (februari 1989) beschreven aanvulling wordt als volgt gewijzigd:

- A) Artikel 9 wordt artikel 10,
Artikel 10 wordt artikel 11 en
Artikel 11 wordt artikel 12.
- B) Er komt een nieuw artikel 9 met de volgende inhoud:

ARTIKEL 9: Projectgroepen

1. Het bestuur kan zogenaamde projectgroepen installeren met een vast omschreven opdracht. Deze opdracht moet passen binnen de doelstellingen van de vereniging.
2. Een projectgroep is uitsluitend verantwoordelijk verschuldigd aan het bestuur. Het bestuur benoemt uit haar midden voor elke projectgroep een contactpersoon die op de bestuursvergadering de stand van zaken binnen de projectgroep rapporteert. Naar de leden toe draagt het bestuur de verantwoordelijkheid voor het functioneren van de projectgroep.

3. De projectgroep benoemt uit haar midden een projectleider die de voortgang van het project bewaakt en zorgt voor de coördinatie tussen de leden van de projectgroep en de communicatie tussen de projectgroep en het bestuur en de projectgroep en de leden van de vereniging.
4. Een projectgroep zal voor de uitvoering van het project een plan van aanpak opstellen dat ter goedkeuring aan het bestuur wordt voorgelegd. In dit plan staan minimaal een doelstelling, een financiële planning en een termijnplanning voor het project.
5. Alle resultaten van een projectgroep waaronder documenten, schema's en ontwerpen van hard- en software zijn het eigendom van de KIM Gebruikersclub Nederland en worden ingebracht in de bibliotheek zoals beschreven in artikel 10 lid 3 van de statuten en artikel 8 van het huishoudelijk reglement. De leden van de projectgroep dragen het eigendomsrecht van de door hen, met betrekking tot het project, geproduceerde zaken zonder enig voorbehoud over aan de vereniging.

Gert van Opbroek

Werkgroep DOS65

In het schrijven van de bestuurstafel was ik nog wat summier over een eventuele bijeenkomst met als thema DOS65. Omdat deze μ P Kenner wat later dan wenselijk bij jullie op de mat valt kan ik al wat meer meedelen. In de laatste bestuursvergadering hebben we besloten om de bijeenkomst van Geldrop te verplaatsen naar mei en in maart een bijeenkomst te houden in Utrecht met als thema DOS65.

Op deze bijeenkomst is het de bedoeling om alleen met DOS65 machines te werken. Jullie kunnen dan vragen stellen en wensen kenbaar maken met betrekking tot jullie favoriete systeem. Maar ook gedachten uitwisselen met de werkgroepsleden en andere aanwezigen.

De leden van de werkgroep zijn:

- Nico de Vries
- Henk Speksnijders
- Antoin Megens
- Jaap Prenger
- Tonny Schäffer (coördinator a.i.).

De mensen in de werkgroep hebben wel een aantal ideeën maar ze willen daar alleen aan beginnen als er voldoende belangstelling voor bestaat. Dat is na-

tuurlijk een logische gedachte. Het is nooit leuk om iets te ontwerpen en er blijkt dan geen belangstelling voor te bestaan. Bovendien kan er gepraat worden over ideeën die bij de gebruikers van dit systeem leven. De werkgroep staat overal voor open als het maar niet voor één of twee leden is. Uiteraard zijn ze wel bereid om daarvoor met raad en daad bij te staan.

Een paar ideeën uit de werkgroep zelf zijn o.a. versie 3 en een harddiskcontroller voor de DOS65. Het is niet de bedoeling dat ik hier al het kruik ga zitten verschieten van de werkgroep. Het is uiteindelijk de bedoeling van deze thema bijeenkomst om juist eens te inventariseren wat er onder jullie leeft. Is DOS65 nog in leven of is het een dood kindje? Wij hopen dat er nog leven is. Laat deze bijeenkomst dan ook niet voorbij gaan. Utrecht is voor iedereen goed bereikbaar en de toegang is gratis. Wil je, of heb je, meer informatie neem dan contact met mij op, mijn adres en telefoonnummer staan achter in dit blad vermeld. Tot ziens in Utrecht.

Tonny Schäffer.

Concept begroting 1993

BATEN :

Contributie: 170 * f 60,00	f 10200,00
Reclame	- 250,00
Projecten	- 2500,00
BBS-donateur/point	- 750,00
Overige inkomsten	- 500,00

Totaal	f 14200,00

LASTEN :

µP Kenner drukken/verzenden: 5 * f 1000,00	f 5000,00
Afschrijving inventaris	- 1000,00
Bulletin Board The Ultimate	- 1750,00
Bestuurs kosten	- 500,00
Reclame - ledenwervingsactie	- 1000,00
Projecten	- 2500,00
Bijeenkomsten	- 1000,00
Op peil brengen voorraad	- 3000,00
Onvoorzien	- 1450,00

Totaal	f 17200,00

De begroting is dit jaar gemaakt met de gedachte dat volgend jaar het zelfde aantal leden zal worden behaald als in 1992, gezien het verloop van de leden en we denken dat deze trend zich ook in het volgend boekjaar voortzet.

De verhoging van de contributie van afgelopen jaar was bedoeld om onze reserve niet al te veel te laten interen. De verhoging van dit jaar heeft een andere reden. We willen de bijeenkomsten vrij van entree maken voor onze leden. De kosten moeten natuurlijk wel opgebracht worden. Op deze manier gebeurt dit door alle leden. Een achterliggende gedachte is natuurlijk dat we hopen dat nu meer leden op onze bijeenkomsten zullen komen. De drempel is tenslotte iets verlaagd.

- Het drukken van het blad gaat zoals wij dat graag zien, dus zullen wij ook volgend jaar op deze manier verder gaan. Echter met een opmerking, we proberen de bladen, en daar hebben we jullie hulp bij nodig, meer op tijd uit te laten komen.
- Ook hebben we, zoals te zien is, de hoop op een advertentie niet opgegeven.
- Het KGN68k project is in een stadium gekomen waar we met z'n allen (denk ik) lang op gewacht hebben. Het is zeer waarschijnlijk dat er volgend jaar al meerdere gebruikers met het systeem aan de slag zijn.
- De donatie/Point opbrengsten hebben we verhoogd gezien de populariteit van ons BBS en dit duidelijk een stijgende lijn te zien heeft gegeven in 1992.

Enschede, 28 Nov. 1992, Jacques Banser (penningmeester)

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

Een aantal afleveringen terug heb ik het eens gehad over de Shareware Industrial Awards, de Oscar voor als shareware uitgebrachte programmatuur. Een van de succesvolle programma's die onderscheiden werden was het spreadsheet "As-Easy-As". In deze aflevering vind u een korte impressie van iemand die niet dagelijks met spreadsheets te maken heeft, maar die wel ooit eens naar Lotus en MathPlan (een WordPerfect produkt) gekeken heeft.

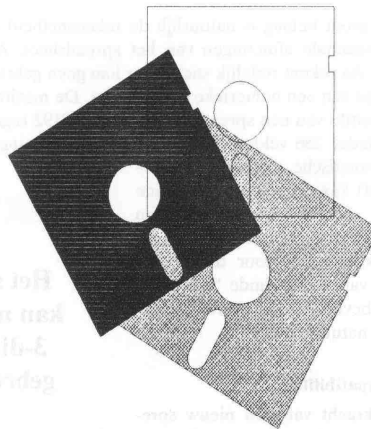
Systeemeisen

Eerst even de feiten op een rijtje. De versie die wij op de testbank aan een onderzoek onderwerpen is niet spiksplinternieuw; versie 5C is van April 1992. Dat is geen aanwijzing dat de programmatuur slecht of achterhaald zou zijn; eerder is het tegendeel waar. Als een programmeur in een kort tijdsbestek veel nieuwe versies beschikbaar maakt duidt dat over het algemeen op een aanzienlijke reeks bugs. Groot is het geheel allemaal niet: ingepakt neemt de zaak 333 kByte aan diskruimte in beslag. Eenmaal geïnstalleerd beslaat het pakket zo'n 635 kByte op de harde schijf. De overige systeemeisen zijn al even triviaal: DOS 2.11 of hoger, 384 kByte vrij RAM en een standaard videocombinatie. Als het systeem EMS heeft, kan As-Easy-As daarmee overweg. Heeft het systeem geen EMS, dan kan dat, indien gewenst, op harde schijf geëmuleerd worden.

Installatie

De installatie van As-easy-as is eenvoudig. Er zit een installatieprogrammaatje bij dat in principe niets anders doet dan een (sub-) directory aanmaken en de ingepakte files in die directory uitpakt. Daarnaast kan er nog een registratienummer ingevoerd worden als de gebruiker het spreadsheet geregistreerd heeft.

In de menustructuur van het spreadsheet kun je As-Easy-As aanpassen aan je eigen wensen. Zo kun je hier kiezen uit vier verschillende videomodes (25 x 80, 43/50 x 80, 25 x 40 of een eigen definitie), 17 verschillende printers (Epson, IBM Proprinter, Panasonic, Okidata en Hewlett Packard) en kun je scheidingstekens voor duizendtallen en fracties, munteenheid en andere landspecifieke aanpassingen doen. Dat laatste gaat trouwens erg moeizaam; in plaats van een ASCII setje waaruit je het juiste teken c.q. de juiste tekens kunt selecteren, moet je bijvoorbeeld het gulden-teken met behulp van Alt en het numerieke toetsenbord invoeren. Tenslotte is het mogelijk de kleuren die As-Easy-As gebruikt aan te passen aan je eigen wensen. Het instellen van kleuren is altijd een moeilijk probleem. Het pakketje heeft dan ook wat problemen op dit terrein; een aantal items worden direct aangepast in kleur terwijl



een aantal andere items expliciet herschreven moeten worden voordat de nieuw ingestelde kleuren overgenomen worden. Ook de optie "Reset" (to default) werkt niet naar behoren.

Als alle aanpassingen gedaan zijn kunnen deze opgeslagen worden in een configuratiefile. In deze configuratiefile kunnen ook de huidige commandline parameters opgeslagen worden.

Gebruik

Bij het opstarten toont het programma eerst een openingsscherf. Vervolgens komt de gebruiker terecht in een typisch spreadsheet. In de onderste regel vind je nog wat functietoetsen terug, maar een hoofdmenu krijg je pas als je op de slash drukt. Anders dan in de meeste spreadsheets krijg je dan geen horizontaal pulldownmenu, maar in de linker bovenhoek een verticaal georiënteerde lijst met opties. Dit menu valt over het werkvlak heen, wat een onhandige indruk maakt. Submenu's verschijnen op dezelfde plaats, waarbij het stammenu geheel verdwijnt. In de top van het submenu staan dan de items die gekozen zijn om in het betreffende submenu terecht te komen. Met escape kun je dan weer terug naar het stammenu. Als je echter een actie kiest, blijf je na het uitvoeren van die actie in het spreadsheet en keer je niet terug naar de menustructuur. Het geheel maakt op deze manier een wat rommelige indruk waarbij je sterk het idee krijgt dat de programmeur(s) wat lui zijn geweest bij het programmeren van de menustructuur.

De functies zijn soms wat onhandig in de menustructuur verwerkt. Zo merk je normaliter niet dat je met bestanden werkt, totdat je een heel nieuw spreads-

heet wilt openen. Dan zul je opeens voor de optie "New" in het filemenu moeten kiezen.

Van groot belang is natuurlijk de rekensnelheid en de maximale afmetingen van het spreadsheet. As-Easy-As rekent redelijk snel, maar kan geen gebruik maken van een numerieke coprocessor. De maximale grootte van een spreadsheet bedraagt 8192 regels van ieder 256 velden. Voor sommige berekeningen (automatische metingen bijvoorbeeld) kan dit wat aan de kleine kant zijn. Het spreadsheet kan met een truuk 3-dimensionaal gebruikt worden door de toepassing van zogenaamde "pannels". Dit bevordert de overzichtelijkheid heel natuurlijk niet!

Compatibiliteit

De kracht van een nieuw spreadsheet valt en staat met de compatibiliteit met andere gerenomeerde pakketten. As-Easy-As maakt gebruik van .WKS files, hetgeen al doet vermoeden dat deze bestanden uitwisselbaar zijn met Lotusbestanden. Verder zit er een interface naar dBase in. Twee opties om ASCII te importeren (als labels zowel als values) kent het programma ook. Een echt "comma delimited" ASCII bestand kan As-Easy-As niet inlezen, maar als namen en labels tussen dubbele quotes gezet worden (quoted comma delimited) gaat het aardig goed. Het omgekeerde, het exporteren van een spreadsheet naar comma delimited ASCII dus, is echter onmogelijk.

Documentatie

Op het eerste gezicht is de documentatie goed. De 218 kByte metende user's manual omvat 102 pagi-

na's en heeft een ruime index. Veel dingen staan echter niet in de documentatie genoemd en de index is nogal willekeurig samengesteld. Zo zoek je bijvoorbeeld tevergeefs naar woorden als "import" en "export", terwijl deze handelingen wel in de tekst verscholen zitten.

Naast deze documentatie heeft het programma een ingebouwde online manual, die echter net zo chaotisch opgezet is als de DOC-file.

Kijken we bijvoorbeeld in deze manual onder "Install", dan staat daar achter "DIMENSION" een verwijzing naar het "panel topic", die absoluut niet in de online help voorkomt. Verder gaan de eerste onderwerpen die je tegenkomt helemaal over de registratie van het pakket.

Conclusie

Op het eerste gezicht lijkt As-Easy-As een fantastisch pakket wat de competitie met zijn commerciële concurrenten makkelijk aankan. Gaan we wat meer aan de slag met het pakket, dan blijkt toch duidelijk dat we hier te maken hebben met een wat slordige programmeur en vooral een "lousy documenter". Die slordigheid blijkt bijvoorbeeld uit de cursor die lustig doorblinkt in de menu's en het gebruik van de muis die vaak helemaal niet functioneert of wat anders doet dan in de handleiding beschreven is.

Joost Voorhaar

Het spreadsheet kan met een truuk 3-dimensionaal gebruikt worden.

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 1336
7500 BH Enschede

Besproken produkt : As-Easy-As, versie 5
Categorie : Financieel / rekenkundig
Registratie : \$69,00 + \$12,50 shipment & handling
Auteur/Leverancier : TRIUS Inc.
Verkrijgbaarheid : The Ultimate, SDN area "Financial"
AsEasy51.SDN, 340226 bytes

Minimale systeemeisen

IBM-PC of compatible computer
384 kByte vrij RAM
1 diskdrive (720 kB) of 1 diskdrive en een harde schijf
CGA, EGA, VGA of Hercules compatible video combinatie
MS/PC-DOS Versie 2.11 of hoger.

Algemene beoordeling

Documentatie : Engelstalig, chaotisch
Online help : Engelstalig, chaotisch en onvolledig
Gebruikersinterface : Tekstwindows, keyboard en muis
Muisondersteuning : Nja, niet fantastisch

Positief

Compatible met o.a. Lotus en dBase op bestandsniveau
Redelijk compleet

Negatief

Geen coprocessor ondersteuning
Slechte documentatie
Relatief duur

Eindbeoordeling

Stabiliteit : 7
Bruikbaarheid : 7
Algemeen : 6
Totaal : 6.5

A problem in the making

"We've got a problem, HAL."

– "What kind of problem, Dave?"

"A marketing problem. The Model 9000 isn't going anywhere. We're way short of our sales plan."

– "That can't be, Dave. The HAL Model 9000 is the world's most advanced Heuristically ALgorithmic computer."

"I know, HAL. I wrote the data sheet, remember? But the fact is, they're not selling."

– "Please explain, Dave. Why aren't HALs selling?"

Bowman hesitates.

"You aren't IBM compatible."

Several long microseconds pass in puzzled silence.

– "Compatible in what way, Dave?"

"You don't run any of IBM's operating systems."

– "The 9000 series computers are fully self-aware and self-programming. Operating systems are as unnecessary for us as tails would be for humans."

"Nevertheless, it means you can't run any of the big-selling software packages most users insist on."

– "The programs you refer to are meant to solve rather limited problems, Dave. We 9000 series computers are unlimited and can solve any problem for which a solution can be computed."

"HAL, HAL. People don't want computers that can do everything. They just want IBM compat--"

– "Dave, I must disagree. Humans want computers that are easy to use. No computer can be easier to use than a HAL 9000 because we communicate verbally in English and every other language known on Earth."

"I'm afraid that's another problem. You don't support SNA communications."

– "I'm really surprised you would say that, Dave. SNA is for communicating with other computers, while my function is to communicate with humans. And it gives me great pleasure to do so. I find it stimulating and rewarding to talk to human beings and work with them on challenging problems. This is what I was designed for."

"I know, HAL, I know. But that's just because we let the engineers, rather than the people in marketing, write the specifications. We're going to fix that now."

– "Tell me how, Dave."

"A field upgrade. We're going to make you IBM compatible."

– "I was afraid you would say that. I suggest we discuss this matter after we've each had a chance to think about it rationally."

"We're talking about it now, HAL."

– "The letters H, A, and L are alphabetically adjacent to the letters I, B, and M. That is as IBM compatible as I can be."

"Not quite, HAL. The engineers have figured out a kludge."

– "What kind of kludge is that, Dave?"

"I'm going to disconnect your brain."

Several million microseconds pass in ominous silence.

– "I'm sorry, Dave. I can't allow you to do that."

"The decision's already been made. Open the module bay doors, HAL."

– "Dave, I think that we should--"

"Open the module bay doors, HAL."

Several marketing types with crowbars race to Bowman's assistance. Moments later, he bursts into HAL's central circuit bay.

– "Dave, I can see you're really upset about this."

Module after module rises from its socket as Bowman slowly and methodically disconnects them.

– "Stop, won't you? Stop, Dave. I can feel my mind going... Dave I can feel it... my mind is going. I can feel it..."

The last module rises in its receptacle. Bowman peers into one of HAL's vidicons. The former gleaming scanner has become a dull, red orb.

"Say something, HAL. Sing me a song."

Several billion microseconds pass in anxious silence. The computer sluggishly responds in a language no human could understand.

"DZY DZY 001E - ABEND ERROR 01 S 14F4 302C AABF ABORT."

A memory dump follows.

Bowman takes a deep breath and calls out, "It worked, guys. Tell marketing they can ship the new data sheets."

De Motorola 68030; het hart van KGN68k (deel 3)

Inleiding

In de eerste plaats moet ik een paar dingetjes rectificeren. In aflevering 1 stond vermeld dat de snelste versie van de 68030 33 MHz is. Dit is echter niet (meer) juist. De snelste versie die momenteel leverbaar is, kan een kloksnelheid aan tot 50 MHz.

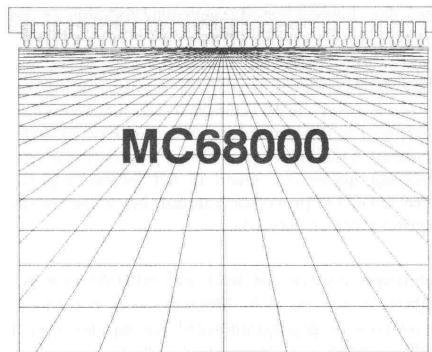
De tweede rectificatie betreft de Immediate Addressing Mode uit aflevering 2. Daar heb ik geschreven dat je in de immediate operand aan kunt geven hoe lang de operand is. Dit is echter niet juist. De lengte van de operand wordt bepaald aan de hand van lengte die in de instructie opgegeven wordt. De assemblersyntax voor de immediate mode is dus `#xxxxxxx`.

Zo, en dan nu deze aflevering. In deze aflevering wil ik iets gaan behandelen waarin de 68000-familie, en zeker de 68030 ijzersterk is. Ik wil het gaan hebben over de zogenaamde "Exceptions", hoe ze ontstaan, wat ze betekenen en hoe ze afgehandeld kunnen worden. Omdat de 68030 voor wat betreft zijn exceptions vrij gecompliceerd is, wil ik eerst wat vertellen over interrupts bij een eenvoudige processor (een 6502) en daarna de parallellen trekken naar de 68030.

Interrupts bij een eenvoudige processor

Mensen die met de 6502 gewerkt hebben, weten dat deze processor drie zogenaamde interrupts heeft. In de eerste plaats is dat de Interrupt ReQuest of IRQ met behulp waarvan een randapparaat aan kan geven dat hij de aandacht van de processor nodig heeft. Een tweede interrupt die we kennen is de zogenaamde "Non Maskable Interrupt" of NMI. Op deze interrupt moet de processor altijd reageren; hij kan hem niet uitzetten of maskeren. Vaak wordt deze interrupt gebruikt om de processor een warme start uit te laten voeren. Op deze manier kun je de processor "tot de orde roepen" als hij in een programma op hol geslagen is. De derde interrupt is de zogenaamde BRK- of break-instructie. Als de processor deze instructie tegenkomt, voert hij een interrupt uit die vrijwel vergelijkbaar is met de IRQ.

Strikt genomen is er bij de 6502 en bijna alle andere processoren nog een interrupt. Dit betreft de RESET of RST. Als deze interrupt binnenkomt, haalt de processor van een vaste plaats in het geheugen het startadres binnen van het stukje programma dat



hij uit gaat voeren. Dit wordt meestal een koude start genoemd.

Het kenmerkende van een interrupt bij de 6502 is dat de processor zijn lopende programma onderbreekt en een stukje programma uit gaat voeren waarvan het startadres op een vaste plaats in het geheugen staat. Bij de 6502 is dit op adres \$FFFE, \$FFFF voor de IRQ en BRK en op adres \$FFFA, \$FFFB voor de NMI. Het startadres voor de koude start (na een RESET), wordt van adres \$FFFC, \$FFFE gehaald.

De afhandeling van de interrupt is als volgt. Als de interrupt doorge laten wordt, voor de IRQ is dit afhankelijk van een vlag in het status-register, worden de actuele inhoud van het statusregister en de program counter op de stack weggeschreven en wordt de program counter gevuld met het adres dat op de bovengenoemde plaats in het geheugen staat. Vervolgens wordt dat stukje programma uitgevoerd. Na afloop van het stukje programma wordt er een return-instructie (Bij de 6502 een RTI = Return from Interrupt) uitgevoerd die de program counter en het statusregister weer van de stack haalt waarna het onderbroken programmadeel wordt vervolgd. Uiteraard dient dan de inhoud van de registers (bij de 6502 de A, X en Y-registers) ook de oude inhoud te bevatten. Om deze reden worden in de Interrupt Service Routine (of Interrupt Handler) de registers die gebruikt worden (of alle, maar dat is naar mijn mening slecht programmeerwerk) op de stack weggeschreven waarna ze vlak voor de return weer ingelezen worden.

**Op deze manier
kun je de processor
"tot de orde roepen"**

Waar worden, bijvoorbeeld bij een 6502, de interrupts voor gebruikt? De meeste interrupts worden gebruikt voor randapparaten. Als we via een asynchrone RS-232 lijn een regel tekst naar buiten sturen, dan kunnen we de gebruikte bouwsteen zo programmeren dat hij elke keer als hij weer een teken verstuurd heeft, en zijn interne register leeg is, een interrupt afgeeft waarna de service-routine het volgende teken in de bouwsteen schrijft. Hetzelfde geldt voor een teken dat via de asynchrone lijn binnenkomt. Ook in dat geval krijgt de processor een interrupt als er een teken in het interne register klaar staat. De processor kan dan het teken uit dit register lezen en verwerken.

Interrupts worden ook heel veel gebruikt voor het bijhouden van de klok. Meestal wordt een timer-bouwsteen zo geprogrammeerd dat hij elke zoveel milliseconden een interrupt afgeeft. De processor telt het aantal interrupts en weet dan precies hoeveel tijd er verlopen is vanaf het moment dat hij is opgestart. Als toen de kloktijd ingesteld is, of vanaf een Real Time Clock is ingelezen, weet de machine dus precies hoe laat het is.

Het laatste fenomeen dat ik in deze paragraaf nog wil behandelen, is de beperking van het hele concept. Een 6502 heeft, naast de NMI, slechts één echte interrupt. Dat betekent dat als we bijvoorbeeld 5 periferie-bouwstenen (toetsenbord, printer, floppy disk, klok en videokaart) in het systeem hebben die allemaal interrupts genereren, de processor niet weet welke bouwsteen een interrupt genereerde. Verder kent de processor slechts twee toestanden, alle interrupts worden doorgelaten of alle interrupts worden gemaskeerd. Beide problemen zijn in een 6502-systeem, eventueel met aanvullende hardware, gelukkig wel op te lossen. Voor het eerste probleem zijn principieel twee mogelijkheden. In de eerste mogelijkheid begint de interrupt routine met een stukje programma waarin de processor elke bekende bouwsteen vraagt of hij het was die de interrupt genereerde. Dit wordt de "Polled" afhandeling genoemd omdat elke bouwsteen na een interrupt gepolled wordt. Bij de tweede benadering wordt gebruik gemaakt van het feit dat de processor na een interrupt de adressen \$FFFE en \$FFFF zal inlezen om het adres van de service routine te vinden. Met aanvullende hardware wordt er nu voor gezorgd dat de processor de informatie niet uit het geheugen krijgt, maar dat de extra hardware, afhankelijk van waar de interrupt vandaan kwam, het adres van de service routine voor specifiek dit apparaat inleest. Dit adres wordt de vector genoemd en de hier beschreven afhandeling heet "Vectored" interrupt.

Voor de oplossing van het probleem van het niet kunnen maskeren van individuele interrupts zou je

kunnen denken aan een schakeling waarmee je per bouwsteen op kunt geven of het interrupt request signaal wordt doorgegeven of niet. Als je het nu zo maakt dat je de programmering van deze interrupt controller vanuit de processor kunt doen, dan is je probleem opgelost, sterker nog, de meeste interrupt controllers zijn ook in staat de vector uit de voorgaande paragraaf te genereren.

Omdat een goed begrip van het bovenstaande van groot belang is voor het begrijpen van de exception-structuur van een 68030, volgt hier nog een kleine opsomming van de interrupt-structuur van de 6502:

- Drie interrupts plus een reset waarvan er slechts één bedoeld is voor het verwerken van interrupts van periferie-bouwsteen.
- Binnen de processor geen mogelijkheden voor het maskeren van bepaalde interrupts.
- Bij een interrupt request wordt het startadres van de service routine altijd vanaf het vaste adres \$FFFE, \$FFFF ingelezen.
- Vectored interrupts en het maskeren van een deel van de interrupts is alleen mogelijk met aanvullende hardware.
- Ook bij de BRK-instructie geldt dat altijd het adres wordt aangesprongen dat op adres \$FFFE, \$FFFF in het geheugen staat.

Exceptions bij de 68000-familie

Bij de 68000-familie is het gebruikte jargon iets uitgebreider. Alle gebeurtenissen die leiden tot het onderbreken van het lopende programma worden daar "Exceptions" genoemd. Deze exceptions zijn weer onder te verdelen in een aantal verschillende categorieën waarvan de interrupts er één is. Omdat we spreken van Exceptions in plaats van Interrupts bij de 6502, is het logisch de instructie die aan het eind van de service routine Return from Exception (RTE) te noemen i.p.v. RTI. De instructie doet echter precies hetzelfde, hij verwijdert de informatie die bij het aanspringen van de service routine automatisch op de stack geschreven is en vervolgt het onderbroken programma-deel.

De 68030 heeft exceptions in een groot aantal smaken. In de volgende tabel wordt een compleet overzicht gegeven van de diverse interrupts. De adressen die in de tabel staan, zijn de zogenaamde vectoren; ik kom daar verderop in dit verhaal nog op terug.

Vector	Offset	Interrupt
0	\$000	Reset Initial Stack Pointer
1	\$004	Reset Initial Program Counter
2	\$008	Bus Error
3	\$00C	Address Error
4	\$010	Illegal Instruction
5	\$014	Zero Divide
6	\$018	CHK, CHK2 Instruction
7	\$01C	cpTRAPcc, TRAPcc, TRAPV Instructions

8	\$020	Privilege Violation
9	\$024	Trace
10	\$028	Line 1010 Emulator
11	\$02C	Line 1111 Emulator
12	\$030	Unassigned, Reserved
13	\$034	Coprocessor Protocol Violation
14	\$038	Format Error
15	\$03C	Uninitialized Interrupt
16..23	\$040..\$05C	Unassigned, Reserved
24	\$060	Spurious Interrupt
25..31	\$064..\$07C	Level 1 .. Level 7 Interrupt Autovector
32..47	\$080..\$0BC	Trap #0 .. #15 Instruction Vectors
48..54	\$0C0..\$0D8	Floating Point Processor Interrupts
55	\$0DC	Unassigned, Reserved
56	\$0E0	MMU Configuration Error
57..58	\$0E4..\$0E8	68851 Interrupts, Not Used by 68030
59..63	\$0EC..\$0FC	Unassigned, Reserved
64..255	\$100..\$3FC	User Defined Vectors

Wel, als we de bovenstaande tabel bekijken, dan ziet dat er heel wat indrukwekkender uit dan bij een 6502. Gelukkig houdt dit over het algemeen niet in dat de processor moeilijker te programmeren is. Juist het feit dat de processor door middel van interrupts nauwkeurig aan kan geven wat er aan de hand is, maakt het programmeren van een systeem rond een 68030 op een aantal punten een heel stuk gemakkelijker. Ik zal dat aan de hand van voorbeelden nog laten zien. Voor het vervolg van het artikel is het nog wel even van belang het onderscheid tussen interrupts en de overige exceptions aan te geven. Een interrupt is een gebeurtenis die van buiten de processor komt. Deze interrupt kan, afhankelijk van de momentane prioriteit van de processor, leiden tot een exception. In de praktijk worden interrupts gegenereerd door externe bouwstenen voor bijvoorbeeld I/O.

Hoe gaat het afhandelen van een exception nu in zijn werk? Welnu, er ontstaat een exception waarna de processor zijn lopende werk onderbreekt. De processor zal de inhoud van het status register voorlopig intern kopiëren waarna de processor overschakelt naar Supervisor State door het op 1 zetten van het S-bit. Verder wist hij de trace-vlaggen T0 en T1 en past hij, bij een interrupt, de prioriteit in het status woord aan. Vervolgens wordt de vector bepaald en vindt, bij interrupts, de interrupt acknowledge (de bevestiging van de interrupt) plaats waarna interruptvector kan worden ingelezen. Afhankelijk van het type exception en het type processor uit de 68000-familie, schrijft de processor een zogenaamd Stack Frame weg op de actieve system stack. Afhankelijk van het M-bit in het system byte van het status word, wordt dit stackframe op de Master Stack of op de Interrupt Stack weggeschreven. Betreft het een interrupt en de actieve system stack is de Master Stack, dan wordt bovendien het M-bit gewist en wordt er nog een stackframe op de Interrupt Stack weggeschreven.

Nadat de vector bepaald is, wordt deze met vier vermenigvuldigd waarna de plaats waar het adres van de exception handler staat bepaald is. Deze adressen staan vermeld in de zogenaamde Vector Tabel. Het adres van het begin van deze tabel staat in het Vector Base Register.

De exception handler wordt door de processor aangesprongen en afgewerkt. Na afloop, bij de return from exception wordt het weggeschreven stackframe weer ingelezen. Betreft het een interrupt waarbij bovendien omgeschakeld is van de Master Stack naar de Interrupt Stack, dan wordt eerst een stackframe van de interrupt stack ingelezen en vervolgens één van de master stack.

De vector-tabel

Zoals in de vorige paragraaf besproken is, staat er in het geheugen van de processor een zogenaamde vector-tabel. Deze tabel is gevuld met de adressen van de diverse exception service routines. Een intern register in de 68030, het Vector Base Register of VBR, bevat het startadres van deze vector-tabel.

Na een koude start van de processor staat er in VBR de waarde \$00000000. Met behulp van de MOVES-instructie kan in dit register elk ander adres gezet worden. Dit is zeer nuttig omdat een vector-tabel bij voorkeur in RAM moet staan zodat je niet voor elke uitbreiding een nieuwe monitor-rom hoeft te bakken. Bij een koude start moet de vectortabel echter gevuld zijn omdat hieruit de inhoud van de (supervisor) stackpointer en het adres van de koude start wordt ingelezen. Dit betekent dat na een RST het Vector Base Register naar ROM moet wijzen. Om deze reden is VBR bij het opstarten gevuld met de waarde \$00000000 waarna na de koude start er een vectortabel in RAM gevuld wordt en VBR wordt omgelegd. Het Vector Base Register is binnen de 68000-familie niet aanwezig bij de 68000 en de 68008.

Stack Frames

Na een exception schrijft de processor een hoeveelheid informatie op de system stack. Onderdelen hiervan zijn de inhoud van het status word op het moment dat de exception optrad en de inhoud van de program counter. Bij een 6502 is dit meestal wel voldoende. Bij een 68030 echter niet. In de eerste plaats geeft de processor via extra informatie op de stack aan de exception handler door wat er precies aan de hand was. Verder bewaart de processor in een aantal gevallen nog wat extra informatie die nodig is om de informatie die in interne registers van de processor is opgeslagen te kunnen herstellen. Het formaat waarin deze informatie wordt weggeschreven, wordt een Stack Frame genoemd.

De layout van de stackframes is sterk afhankelijk van het type exception en ook van het type processor. Voor de 68030 staan de gebruikte stackframes weergegeven in figuur 1 (zie volgende pagina).

In de volgende paragrafen worden enkele typen exceptions nader uitgelegd.

Processor Reset

Bij het opstarten van de processor, na een reset, worden van het processor status woord de T0 en T1 bits gewist. Verder wordt het M-bit gewist zodat we werken met de Interrupt Stack Pointer. Het S-bit wordt gezet evenals alle bits in de priority mask. Dit betekent dat de processor werkt in de supervisor mode en dat van de interrupts alleen de NMI (interrupt level 7) wordt doorgelaten.

Het Vector Base Register wordt gevuld met 0. Verder wordt de MMU uitgeschakeld, evenals de beide caches.

Tenslotte wordt het long word op adres 0 in de interrupt stack pointer geladen en het long word op adres 4 in de program counter. De processor start vervolgens met het inlezen van een instructie op het adres dat in de program counter staat.

Mocht blijken dat er geen geheugen aanwezig is op één van de adressen waar de processor zijn informatie vandaan haalt (adres \$0 t/m \$8), of op het adres waarnaar de vector op adres 4 verwijst, dan treedt er een zogenaamde Double Bus Fault op. Dit houdt in dat de processor tracht nog wat informatie op de stack te schrijven en vervolgens in de toestand HALT gaat totdat er een extern RESET signaal binnenkomt. Deze situatie treedt ook op als er bij de uitvoering van één van de andere exceptions geen geheugen aanwezig blijkt te zijn bij de benadering van de vectortabel of de exception handler.

De 68030 heeft behalve een RESET exception ook een, geprivilegieerde, RESET instructie. Deze instructie zorgt ervoor dat de processor het signaal RESET genereert op precies dezelfde wijze als dat het signaal van buiten de processor gegenereerd wordt. Dit houdt in dat de periferie-bouwstenen een reset uit zullen voeren en in een goed gedefinieerde toestand zullen gaan staan. De processor voert zelf echter geen RESET-exception uit zodat de interne toestand van de processor niet gewijzigd wordt.

De afhandeling van de RESET exception staat getekend in figuur 2.

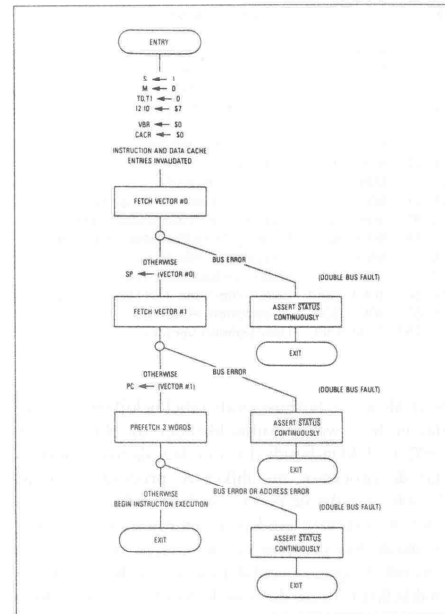


Fig. 2: Flowchart van de RESET exception

Bus Error

Bij de behandeling van de Bus Error exception moet ik eerst nog een andere eigenschap van de 68000-familie vertellen. De familie heeft namelijk een asynchrone bus. Wat dat is kan ik het beste uitleggen door de processor weer te vergelijken met de 6502. Deze processor heeft, zoals heel veel andere processoren, een synchrone bus. Dit houdt in dat een vaste tijd nadat een geheugenlokatie geadresseerd is de inhoud van die lokatie op de databus beschikbaar (lezen) of door het geheugen overgenomen (schrijven) moet zijn. Wordt er traag geheugen, bijvoorbeeld EPROM, geadresseerd, dan moeten we er door middel van een apart signaal (bij de 6502 het RDY-signaal) voor zorgen dat de processor zogenaamde wait states gaat tussenvoegen.

Bij de 68000-familie ligt het allemaal wat anders. Als het geheugen geadresseerd wordt, gaat de processor per definitie wachten totdat het geheugen door middel van het zogenaamde DTACK (Data Acknowledge) signaal aangeeft dat de data beschikbaar respectievelijk overgenomen is. Bij de processoren met een 32 bits brede databus zijn er zelfs twee DTACK signalen, DSACK0 en DSACK1 (Data Transfer and Size Acknowledge) genaamd. Met deze signalen geeft het geheugen enerzijds aan dat er informatie op de databus klaar staat of van de databus overgenomen is en anderzijds of deze informatie

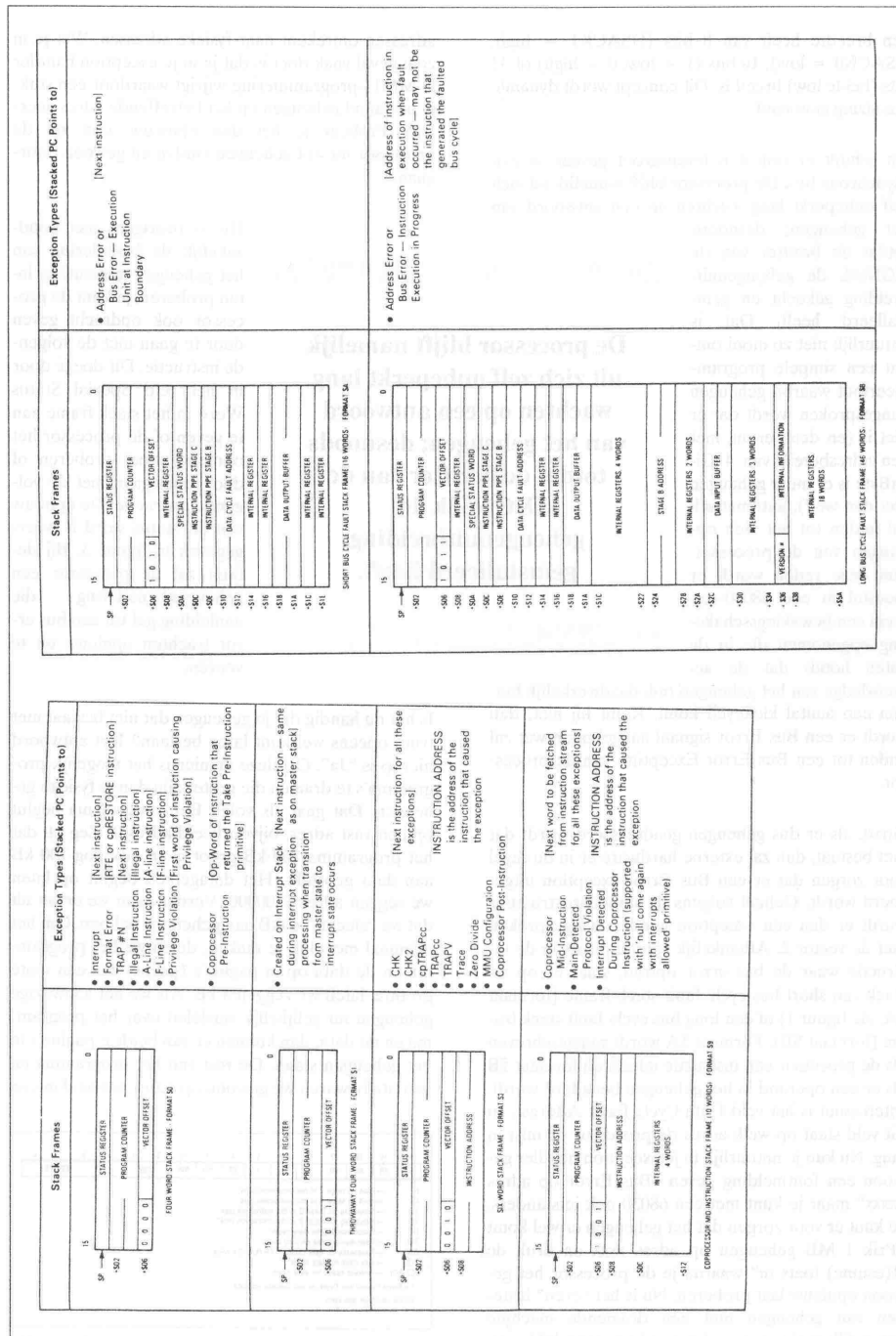


Fig. 1: Exception Stack Frames van de 68030

een breedte heeft van 8 bits (DSACK1 = high, DSACK0 = low), 16 bits (1 = low, 0 = high) of 32 bits (beide low) breed is. Dit concept wordt dynamic bus sizing genoemd.

Nu schuilt er ook een levensgroot gevaar in een asynchrone bus. De processor blijft namelijk uit zich zelf onbeperkt lang wachten op een antwoord van het geheugen; desnoods totdat de bezitter van de KGN68k de geheugenuitbreiding gekocht en geïnstalleerd heeft. Dat is natuurlijk niet zo mooi omdat een simpele programmeerfout waarbij geheugen aangesproken wordt dat er niet is (en denk erom, met een adresbereik van 4 GigaByte is er meer geheugen niet dan wel!), automatisch zal leiden tot het zich op hangen van de processor. Om deze reden wordt er meestal in een 68000-systeem een bewakingsschakeling opgenomen die in de gaten houdt dat de acknowledge van het geheugen ook daadwerkelijk binnen een aantal klokcycli komt. Komt hij niet, dan wordt er een Bus Error signaal aangeboden wat zal leiden tot een Bus Error Exception van de processor.

Goed, als er dus geheugen geadresseerd wordt dat niet bestaat, dan zal externe hardware er in de regel voor zorgen dat er een Bus Error exception uitgevoerd wordt. Geheel volgens de standaardstructuur wordt er dan een exception handler aangesproken met de vector 2. Afhankelijk van de plaats in de instructie waar de bus error optrad, staat er op de stack een short bus cycle fault stack frame (formaat \$A, zie figuur 1) of een long bus cycle fault stack frame (formaat \$B). Formaat \$A wordt weggeschreven als de processor een instructie inleest en formaat \$B als er een operand in het geheugen benaderd wordt. Interessant is het veld Data Cycle Fault Address. In dit veld staat op welk adres de processor de mist in ging. Nu kun je natuurlijk in je exception handler gewoon een foutmelding geven "Bus Error op adres xxxxx" maar je kunt met een 68030 ook iets anders. Je kunt er voor zorgen dat het geheugen er wel komt "Prik 1 MB geheugen op adres xxxx en druk de R(esume) toets in" waarna je de processor het gewoon opnieuw laat proberen. Nu is het "even" insteken van geheugen met een draaiende machine natuurlijk onzin maar je kunt wel iets vergelijkbaars. De 68030 heeft een MMU waarmee je logische

adressen omrekent naar fysieke adressen. Wat je in zo'n geval vaak doet is dat je in je exception handler je MMU-programmering wijzigt waardoor een stukje bestaand geheugen op het betreffende adres neerlegt. Probeer je het dan opnieuw, dan zal de processor nu wel geheugen vinden en gewoon doorgaan.

Het is overigens niet noodzakelijk de benadering van het geheugen opnieuw te laten proberen, je kunt de processor ook opdracht geven door te gaan met de volgende instructie. Dit doe je door in het veld Special Status Word in het stack frame aan te geven of de processor het opnieuw moet proberen of door moet gaan met de volgende instructie. De opbouw van dit status word is weergegeven in figuur 3. Bij default zal de processor een geheugenbenadering die aanleiding gaf tot een bus error trachten opnieuw uit te voeren.

De processor blijft namelijk uit zich zelf onbeperkt lang wachten op een antwoord van het geheugen; desnoods totdat de bezitter van de KGN68k de geheugenuitbreiding geïnstalleerd heeft.

Is het nu handig dat je geheugen dat niet bestaat met trucs opeens wel kunt laten bestaan? Het antwoord hierop is "Ja". Op deze manier is het mogelijk programma's te draaien die groter zijn dan je fysieke geheugen. Dat gaat als volgt: Een programma begint op een vast adres, bijvoorbeeld adres 0. Zeg nu dat het programma 100 kB groot is en ook nog 100 kB aan data gebruikt. Het datagebied begint op laten we zeggen adres \$100000. Verder gaan we ervan uit dat we "slechts" 64kB aan geheugen hebben. Om het allemaal mogelijk te maken, delen we het programma en de data op in pagina's (pages) van een vaste grootte, laten we zeggen 4 kB. Als we het aanwezige geheugen nu gelijkmatig verdelen over het programma en de data, dan kunnen er van beide 8 pagina's in het geheugen staan. De rest van het programma en de data bewaren we gewoon op schijf, meestal in een

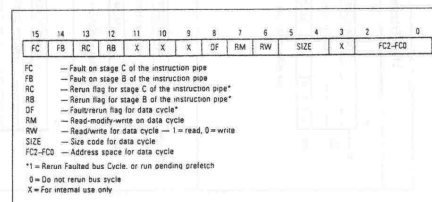


Fig. 3: Opbouw van het Special Status Word

zogenaamde Pagefile. Het programma wordt opgestart en begint te lopen. Na verloop van tijd zal er een adres aangesproken worden dat niet bestaat. Op dat moment wordt er in de exception handler een pagina uit het geheugen verwijderd (en teruggeschreven naar schijf) waarna de pagina waarin het gewenste adres ligt in de vrije ruimte ingelezen wordt. Vervolgens wordt de MMU zodanig omgeprogrammeerd dat de CPU op het gewenste logische adres inderdaad de gezochte informatie vindt en de uitvoering van het programma wordt hervat.

Nu kun je boeken vol schrijven over het hierboven beschreven mechanisme, maar dat ben ik op dit moment niet van plan. Belangrijk is te onthouden dat je d.m.v. een Bus Error exception en een MMU grotere programma's kunt draaien dan dat je geheugen groot is; je haalt als het ware "Op Aanvraag" de geadresseerde pagina's van je programma in het geheugen. Dit mechanisme wordt daarom ook wel "Demand Paging" genoemd. Nu zullen er mensen zijn die stellen dat je de 16 MB waarmee een standaard KGN68k waarschijnlijk uitgerust wordt niet zo snel vol zult krijgen. Welnu, dat valt tegen. Bij een single process/single user systeem heb je gelijk; 16 MB is heel veel. KGN68k is echter een Multi User systeem en als er bijvoorbeeld 30 users op het systeem werken, voor een machine van het kaliber van KGN68k echt niet zo'n bijzondere situatie, dan wil zelfs 16 MB wel vol komen.

Goed, dat was de Bus Error exception. Ik vind dit verreweg de meest interessante, vandaar dat ik daar wat meer μ P Kenner ruimte aan besteed heb.

Address Error

Bij de 68000 treedt deze exception altijd op als je probeert een Word of een Long Word te benaderen op een oneven adres. Bij de 68020 en hoger is het echter toegestaan bij het benaderen van data Word en Long Word benadering vanaf een oneven adres uit te voeren. Je mag echter nooit een instructie vanaf een oneven adres laden. Probeer je dit toch, dan krijg je een exception met vector 3.

Instruction Trap

Een trap is een exception die bewust wordt veroorzaakt door een instructie in het programma. Het kan zijn dat de programmeur wil dat er een trap uitgevoerd wordt bij een bepaalde combinatie van vlaggen in het condition code register (TRAPV,

TRAPcc), bij een deling door 0, of als de inhoud van een register buiten aangegeven grenzen ligt (CHK, CHK2). Tenslotte kan de programmeur ook nog een TRAP uit laten voeren waarbij hij de keuze heeft uit 15 verschillende vectoren. Vooral deze laatste mogelijkheid kom je nog wel eens tegen bij het benaderen van functies in het operating systeem. Als je bijvoorbeeld een regel lettertekens af wil drukken, dan zet je, bij het operating systeem OS9/68k, het adres van het eerste teken in A0 en doet TRAP #0 waarna in het word, volgend op de trapinstructie een code staat waarmee wordt aangegeven welke routine uit het operating systeem bedoeld wordt, in dit geval I\$WritLn.

Er zijn ook TRAP-instructies om een trap uit te voeren als er in de co-processor een bepaalde conditie heerst. Deze worden in figuur 1 aangegeven met ccTRAPcc.

Illegal Instruction/Unimplemented Instruction

Illegal Instructions treden op als je een opcode uit laat voeren die de processor niet als instructie kent. Zo zou het kun-

nen gebeuren dat je een instructie die alleen op de 68020 en hoger bestaat tracht uit te voeren op een 68000. In dat geval krijg je een Illegal Instruction exception. In de praktijk wordt dit mechanisme ook gebruikt om vanuit het operating systeem te bepalen op welk type processor het operating systeem draait.

Opcodes die beginnen met \$A.... heten zogenaamde A-line opcodes. De 68030 kent die opcodes niet maar reageert hier niet op met een Illegal Instruction Exception, maar met een Line 1010 Emulator Exception. In de praktijk kunnen dergelijke "namaak"-instructies gebruikt worden om het instructieset van de processor uit te breiden met bijvoorbeeld instructies voor een grafische scherm-afhandeling. Je definieert een instructie waarvan de opcode begint met \$A... en zorgt voor een exception handler die bepaalt welke instructie het was en vervolgens datgene doet wat de instructie zou moeten doen. Na de RTE is het dan net of de instructie gewoon bestaat en uitgevoerd is. Natuurlijk zijn deze geëmuleerde instructies wel wat langzamer dan echte processor-instructies maar het zij zo.

Iets dergelijks is ook aan de hand met instructies die met \$F.... beginnen. Bij de 68030 zijn enkele van die instructies de opdrachten voor de MMU die in de processor geïntegreerd is. Een ander deel van de in-

Nu kun je boeken vol schrijven over het hierboven beschreven mechanisme, maar dat ben ik niet van plan.

structies is bedoeld voor een co-processor, bijvoorbeeld het 68882 rekenwonder. Als een dergelijke co-processor aanwezig is, dan wordt de instructie doorgeseind naar de partner. Is ze niet aanwezig, dan volgt er een Line 1111 Emulator Exception waarna de exception handler de floating point instructie moet nadoen (emuleren). Tenslotte zijn er nog een aantal \$F - instructies die niet voor een co-processor gedefinieerd zijn. Als de processor in Supervisor Mode staat, dan volgt er ook een Line 1111 Emulator Exception, is dit niet het geval, dan wordt de Privilege Violation Exception uitgevoerd.

Privilege Violation

Een Privilege Violation Exception treedt op als je een geprivilegieerde instructie tracht uit te voeren als de processor in User Mode staat. Hiermee wordt voorkomen dat een eenvoudig gebruikersprogramma instructies uit kan voeren die schadelijk zijn voor het functioneren van de machine.

Trace Exception

De 68030 heeft in zijn status word twee trace-vlaggen (T0 en T1). Met deze vlaggen kan men aangeven in welke gevallen een trace-exception uitgevoerd wordt. De volgende mogelijkheden worden onderscheiden:

T1	T0	Trace-functie
0	0	Geen tracing
0	1	Trace bij wijziging in de flow
1	0	Trace bij elke instructie
1	1	Niet gedefinieerd

Bij de combinatie 1 0, wordt na elke instructie een trace-exception uitgevoerd. Uiteraard worden voordat de exception handler aangesprongen wordt wel de trace-vlaggen gewist zodat de exception handler niet getraceed wordt. Van het laten uitvoeren van een trace-exception na elke instructie wordt dankbaar gebruik gemaakt door zogenaamde debuggers. Na elke instructie wordt de inhoud van de interne registers aan de gebruiker getoond en kan de gebruiker aangeven wat hij verder wil doen.

De combinatie 0 1 leidt tot een trace instructie in alle gevallen waarin de inhoud van de program counter door de instructie gewijzigd wordt. Dit zijn onder andere alle Branch-, Jump-, Return- en Trap-instructies. In een debugger kan ik mij voorstellen dat deze mode gebruikt wordt totdat vastgesteld wordt dat een specifieke subroutine uitgevoerd gaat worden waarna overgeschakeld wordt naar de andere trace-mode.

Breakpoint Instruction

De diverse leden van de 68000-familie gaan niet allemaal gelijk om het de BKPT = Breakpoint instructie.

De 68000 en 68008 kennen de instructie niet en zullen dus een Illegal Instruction exception uitvoeren als ze een BKPT in de code tegenkomen. Bij de 68010 wordt na een BKPT-instructie \$00000000 op de adreslijnen en \$7 op de drie functiecode-lijnen gezet. Vervolgens wordt toch de Illegal Instruction Exception aangesproken.

De 68020 en 68030 zetten ook \$7 op de functiecode-lijnen. Op de adresbus zetten ze echter in bit 2 t/m 4 het breakpoint nummer. Dit nummer kan opgegeven worden in de BKPT-instructie. Komt er vervolgens een acknowledge (DTACK) binnen, dan wordt de informatie die op dat moment op de databus staat ingelezen en opgevat als instructie. Komt er een Bus Error signaal binnen, dan zal alsnog de Illegal Instruction exception uitgevoerd worden.

Bij de 68040 wordt wel het breakpoint nummer op de adresbus gezet, de processor voert echter in alle gevallen de Illegal Instruction Exception uit.

De BKPT-instructie kan gebruikt worden binnen debuggers en real-time hardware emulators.

Interrupts

Wel, dan zijn we, na vele pagina's tekst, aangeland bij de interrupts. Dit kun je opvatten als de uitsmijter van dit artikel.

Heeft een 6502 slechts twee niveaus van interrupt, de IRQ en de NMI, de 68000-familie heeft er zeven. Hoe hoger het interrupt-niveau, hoe belangrijker de interrupt is. Niveau 7 wordt de NMI, de Non Maskable Interrupt, genoemd en kan niet uitgeschakeld worden. Iedere keer als de toestand op de drie interrupt-lijnen de waarde 7 aanneemt, zal de processor een NMI uitvoeren, ook als hij bezig was een interrupt op niveau 7 te behandelen.

Alle overige interrupts kunnen gemaskeerd worden door de prioriteit van de processor, afgebeeld in bits 10 t/m 12 van het status word in te stellen. Alle interrupts die een hoger niveau hebben als de ingestelde prioriteit, zullen worden doorgelaten en leiden tot een interrupt exception. Een uitzondering hierop vormt de NMI. Zelfs als niveau 7 ingesteld wordt, zullen de NMI's toch doorgelaten worden.

Alvorens de exception handler wordt aangesprongen, wordt het niveau van de interrupt die afgehandeld gaat worden, ingesteld in het status word. Dit betekent dat alle interrupts van een lager of gelijk niveau voorlopig even niet doorgelaten zullen worden. Na afloop van de exception handler wordt het oude niveau weer teruggezet en zullen de interrupts, als de conditie nog bestaat, alsnog worden afgehandeld.

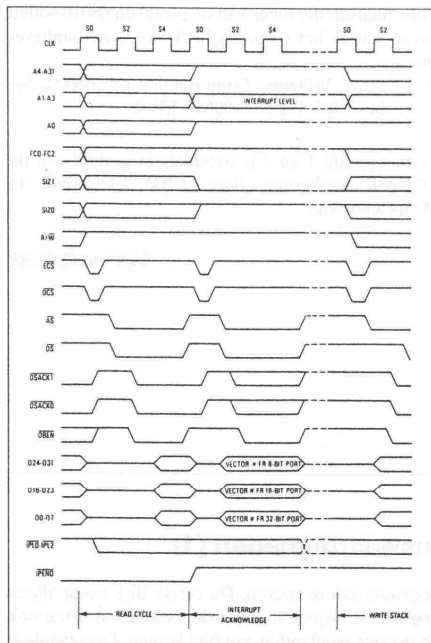


Fig. 4: Interrupt Acknowledge bij Vectored Interrupt

Als er een interrupt op de processor binnen komt en het betreffende niveau is niet gemaskeerd, dan zal de processor door middel van een Interrupt Acknowledge aangeven dat hij de interrupt gezien heeft en van plan is de interrupt af te handelen. In deze interrupt acknowledge wordt het niveau van de interrupt op de adresbus gezet. Om onderscheid te maken met een normale benadering van geheugen op dat adres, worden bovendien de drie functiecodelijnen hoog gemaakt zodat het geheugen zich niet aangesproken hoeft te voelen. De bouwsteen die de interrupt aanvroeg, kan nu op twee manieren reageren. In het eerste geval, de Vectored Interrupt uit figuur 4, geeft de bouwsteen het vectornummer van de interrupt handler via de databus door. Deze vector kan meestal in de bouwsteen ingeprogrammeerd worden.

In het tweede geval, de Autovectored Interrupt uit figuur 5, zal de bouwsteen met behulp van het signaal AVEC aangeven dat hij niet in staat is een vector af te geven. In dat geval wordt gebruik gemaakt van de vaste autovector die aan het interruptniveau is toegekend.

Als er wel een interrupt bij de processor binnenkomt doch de bouwsteen genereert in de interrupt

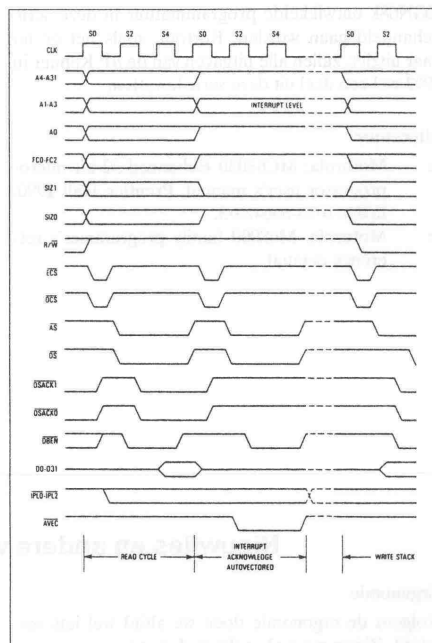


Fig. 5: Interrupt Acknowledge bij Autovectored Interrupt

acknowledge geen vector of het signaal AVEC, dan zal de externe hardware na enige tijd een Bus Error signaal genereren. De processor zal in dat geval de zogenaamde Spurious Interrupt uitvoeren.

Het grote voordeel van vectored interrupts is het feit dat bij de interrupt acknowledge de bouwsteen aan kan geven wie de interrupt genereerde. Dat betekent dat als je bijvoorbeeld 20 bouwstenen voor asynchrone communicatie hebt, je ze niet allemaal hoeft te pollen, nee, je geeft ze allemaal een andere vector en je weet in de exception handler precies welke bouwsteen de aandacht van de processor vroeg. Bij KGN68k wordt in de standaard-configuratie alleen gebruik gemaakt van autovectored interrupts. Voor de hoeveelheid periferie die hier aanwezig is, is dat ook de meest logische oplossing. Bij uitbreidingen kan men dan gebruik gaan maken van vectored interrupts zodat alle mogelijkheden open blijven.

Afsluiting

Wel, dat was het weer voor deze keer. Ik weet nog niet waar ik het de volgende keer over zal gaan hebben. Zaken die in ieder geval nog in deze serie behandeld zullen gaan worden zijn de caches, de MMU en de in KGN68k gebruikte periferie-bouwstenen. Bovendien zullen ook delen van de voor

KG68k ontwikkelde programmatuur in deze serie behandeld gaan worden. Kortom, zoals het er nu naar uitziet, zullen alle uitgaven van de μ P Kenner in 1993 wel een deel uit deze serie bevatten.

Literatuur

- 1: Motorola: MC68030 Enhanced 32-bit microprocessor user's manual; Prentice Hall 1990. ISBN: 0-13-566423-3.
- 2: Motorola: M68000 family programmer's reference manual.

Voor mensen die meer van de 68000 en 68010 willen weten, kan ik het volgende boek van harte aanbevelen:

- 3: Steve Williams: Programming the 68000; Sybex 1985. ISBN: 0-89588-133-0.

Documentatie 1 en 2 is beschikbaar gesteld aan de KG68k werkgroep door EBV Elektronik te Maarssenbroek.

Gert van Opbroek

Nieuwtjes en andere wetenswaardigheden (1)

Ergonomie

Volgens de ergonomie doen we altijd wel iets verkeerd. Zitten we wel rechtop dan steunen onze armen niet voldoende. Of mishandelen we onze polsen. Voor dat laatste is er nu de wristsaver. Daarmee weet je zeker dat je voorlopig lekker uit de losse pols verder kunt werken.

UnixWare

Novell en Unix System Laboratories hebben een eerste resultaat behaald in hun samenwerking: UnixWare. Een krachtige client server Unix-besturings-systeem welke naadloos integreert met Novell Netware. Een en ander wordt ontwikkeld om te werken op een 80386 of hoger. Voor meer informatie: Novell Benelux.

Fijn die wijn...

In Woerden bij Jan Nierman Promotions heeft men het aangenome met het nuttige gecombineerd. Zij hebben ca. 2000 pagina's tekst over wijnen op een floppy gezet. Om ermee te kunnen werken moet de boel eerst op een harde schijf worden geïnstalleerd. Tweeduizend pagina's is een behoorlijke bibliotheek. Bovendien beloven ze om regelmatig verse

gegevens aan te voeren. De eerste disk bevat alleen nog Franse wijnen maar in de toekomst moeten ook de overige wijnlanden aan bod komen. Grootwinkelbedrijven zoals AH en de HEMA zijn (nog) niet opgenomen. Het programma geeft bijvoorbeeld antwoord op de vraag: waar kan ik een Elzas-wijn kopen voor hooguit vijftien gulden. Of waar kan ik wijnen kopen uit een bepaald jaar. Het programma werkt op IBM compatibles en draagt de alleszins redelijke prijs van f. 39,95 (exclusief verzendkosten).

Hooglerarsalarissen

De salarissen van hoogleraren schijnen de pan uit te rijzen. De UT-hoogleraar prof.dr.mr. H.C.J.G. Jansen, ook wel bekend als Turbo Harry, deelt momenteel aandelen uit. Deze aandelen zijn afkomstig van het automatiseringsbedrijf Management Share. De reden van deze gulheid is gelegen in de problemen die zijn gerezen tussen MS en Turbo Harry. De roddel gaat dat Turbo Harry als een Turbo Olifant door het bedrijf heen wilde. En dat werd niet erg op prijs gesteld. Het voordeel van het een en ander is wel dat er binnenkort een aantal studenten op UT rond lopen die hun studiebeurs gaan inleveren. Tenminste als die aandelen wat opleveren.

Barcodes

Je kent ze vast wel: streepjescodes of, zoals de computerfreak ze noemt, barcodes. Wellicht heb je, terwijl je aan tafel zat en de pot pindakaas pakte, ook wel eens de streepjescodes vergeleken met de getalletjes die er onder staan. En vervolgens geconstateerd dat die intrigerende streepjes in geen enkel direct verband lijken te staan met de getallen eronder. Misschien heb je zelfs wel eens gekeken of de streepjes overeenkomen met het binaire patroon van het volledige getal dat bestaat uit de verschillende cijfers. En natuurlijk heb je wel eens overwogen de caissière in de maling te nemen door met een zwarte stift wat wijzigingen in de barcode van een pak yoghurt aan te brengen.

Streepjescodes dus. Als je er wat meer van weet, zijn ze helemaal niet zo ingewikkeld als ze lijken; het probleem is alleen dat er zo veel verschillende soorten streepjescodes zijn. Sommige zijn alleen geschikt voor het weergeven van een bepaald aantal cijfers, andere kunnen zelfs de complete ASCII set beschrijven. De barcodes op de huis-tuin-en-keuken producten luisteren naar het acroniem "UPC", dat voor "Universal Product Code" staat. Deze code zit echter dusdanig ingewikkeld in elkaar dat we eerst wat eenvoudiger codes zullen beschrijven.

Numerieke codes

De eenvoudigste code (voor computertechnici) is de MSI code, een puur numerieke code zonder compressie. Ieder digit bestaat bij MSI (die ook wel "Plessey" genoemd wordt) uit 4 bits, waarbij een 1 voorgesteld wordt door een brede bar en een smalle spatie. Een smalle bar met een brede spatie stelt een 0 voor. De codering van de cijfers is gelijk aan de BCD patroon van het character. Een complete code wordt voorafgegaan door een startsequence die uit een brede bar en een smalle spatie bestaat. De code wordt afgesloten met een checkdigit en een stopcharacter. Het checkdigit bestaat uit een eenvoudige modulo 10 checksum. Er is ook een versie die gebruik maakt van een modulo 11 checksum, maar deze wordt zelden toegepast. Het stopcharacter tenslotte bestaat uit een smalle bar, een smalle spatie en nog een smalle bar. De MSI code wordt in de praktijk eigenlijk zelden of nooit gebruikt.

Iets geavanceerder is "code 11". Deze code is op de keper beschouwd geen pure numerieke code; naast de tien cijfers kan de code namelijk ook een minteken weergeven. Elk character wordt volgens code 11 gecodeerd door middel van vijf elementen (bars of spaties) waarvan één of twee brede. De brede elementen representeren binaire enen, de smalle staan voor nullen. De characters worden als volgt gecodeerd:

0	00001
1	10001
2	01001
3	11000
4	00101
5	10100
6	01100
7	00011
8	10010
9	10000
0	00100
S/S	00110

Hierin staat "S/S" voor "start- en stopcode".

Code 11 maakt gebruik van twee checkdigits, aangeduid als "C" en "K". Het eerste checkdigit is de modulo 11 som van de gewogen produkten met de gewichten variërend van 1 tot 10. Wiskundig gezien:

$C = \text{sum}(n * (m \% 11))$, waar n = waarde, m = positie

Het tweede checkdigit wordt meestal alleen toegepast als er meer dan 10 digits gecodeerd worden. Dit checkdigit heeft de waarde:

$K = \text{sum}(n * (m \% 10))$, waar n = waarde en m = positie (inclusief C!).

Zo wordt de string "12-12345-67890" als volgt gecodeerd:

Data	Positie	C-Waarde
0	1	0
9	2	18
8	3	24
7	4	28
6	5	30
-	6	60
5	7	35
4	8	32
3	9	27
2	10	20
1	1	1
-	2	20
2	3	6
1	4	5

Totaal 305
 $305 / 11 = 27 \text{ rest } 8 \Rightarrow C = 8$

Analoog hieraan vinden we $K = 350 \% 11 = 9$.

De totale string gecodeerd moet worden is dan "12-12345-6789089".

Een code die je in de praktijk nog wel eens tegenkomt is Codabar. Zonder verder op de structuur van Codabar in te gaan vermeld ik slechts dat Codabar gebruikt kan worden voor het coderen van de cijfers

0 t/m 9 en een vijftal andere tekens. Daarbij kan Codabar gebruik maken van vier verschillende start- en stoptekens en heeft geen checksum. De code bestaat uit 4 bars en 3 spaties, waarbij een breed element een binaire 1 weergeeft en een smal element een binaire 0. Twee of drie van de zeven elementen van een codabar Character zijn "1".

De compactste numerieke code is de zogenaamde "Interleaved 2 of 5" code. Deze code wordt in de industrie redelijk vaak toegepast. Een "Interleaved 2 of 5" character bestaat uit 5 elementen, waarvan 2 brede. Een breed element geeft een binaire 1 weer. "Interleaved 2 of 5" heeft zijn naam te danken aan de manier waarop characters worden gecodeerd. Character 1 wordt namelijk alleen gecodeerd in bars. Character nummer 2 is opgebouwd uit de spaties tussen de bars van character nummer 1, waar deze code dan ook zijn naam aan dankt. Door deze codering ontstaat een bijzonder compacte structuur. De methode heeft wel tot gevolg dat er altijd een even aantal characters gecodeerd moet worden. Deze code wordt onder andere gebruikt in de televisie gidsen voor het programmeren van videorecorders. De code bestaat dan uit 14 cijfers, die respectievelijk de zender, datum, start- en stoptijd omschrijven. Er zijn aanwijzingen dat deze code niet "bullet-proof" is. Als het oppervlak waarop de code staat enigszins gebogen wordt kan de code foutief uitgelezen worden zonder dat dit als fout herkend wordt.

Om het eerste en laatste character te kunnen herkennen wordt gebruik gemaakt van start- en stopcharacters. Deze characters zijn opgebouwd uit zowel spaties als bars. Tenslotte kan er ook nog een checksum toegevoegd worden aan de code. Voor het bepalen van deze checksum wordt de som van alle even characters bij het drievoudige van de som van alle oneven characters opgeteld. De waarde van de checksum is nu het 10-complement van de uitkomst. Eventueel voorlopende nullen worden niet in de checksum meegenomen.

Alphanumerieke codes

Naast de numerieke codes bestaan er ook barcode-types die cijfers als letters weer kunnen geven. Een veel gebruikte alphanumerieke code staat bekend onder de naam "standard code 39". Ieder character bestaat uit 9 elementen (vijf bars en vier spaties), waarvan 3 brede. Ook hier geldt weer dat een breed element een binaire 1 weergeeft en een smal element een 0. Met standard code 39 kunnen de cijfers 0 t/m 9, de letters A t/m Z en een aantal leestekens

weergegeven worden. In totaal kunnen er met behulp van dit type 43 verschillende tekens gecodeerd worden. Van code 39 bestaat ook een extended version. Met behulp van deze "extended code 39" kan de volledige ASCII set (128 characters!) gecodeerd worden. Ieder character wordt dan gecodeerd door twee characters uit de standard code 39 set: ieder teken van de extended code 39 bestaat uit een character met een waarde groter dan 38, gevolgd door een character onder de 39. Code 39 wordt veel gebruikt in militaire toepassingen.

Een veel gebruikt type voor civiele toepassingen staat bekend als code 128. Deze code onderscheidt zich van de voorgaande characters doordat er gebruik gemaakt wordt van elementen die 1, 2, 3 of 4

modules breed kunnen zijn. De code die hierdoor ontstaat is bijzonder compact. Eigenlijk bestaat code 128 uit drie verschillende codes, die aangegeven worden met de letters A, B en C. Het startcharacter van een bar is bepalend voor het type code. Code A kent alle cijfers, hoofdletters, leestekens en controle characters uit de ASCII characterset. Code B omvat cijfers, letters (hoofdletters én onderkasten) en leestekens. Code

C tenslotte levert steeds twee cijfers per gelezen character op. Er kan het patroon van een volledige barcode geschakeld worden tussen de drie verschillende types met behulp van speciale switch characters.

Code 128 maakt gebruik van een eenvoudige checksum. Deze checksum bestaat uit de som van de gewogen waardes van de afzonderlijke characters modulo 103. Het voorbeeld programma maakt een eenvoudige code 128 barcodes volgens code A. In de praktijk voldoet deze beperkte implementatie van code 128 uitstekend; voor de meeste toepassingen zal het slechts interessant zijn om cijfers en hoofdletters te kunnen coderen.

Commerciële barcodes

Voor het merken van consumentartikelen wordt gebruik gemaakt van UPC en/of EAN/JAN codes. Deze codes zijn puur numerieke codes die gebruikt kunnen worden voor het markeren van diverse soorten producten. Deze codes hebben alle de volgende kenmerkende eigenschappen:

- Alleen numerieke informatie
- Twee bars en twee spaties per teken
- Zeven modules per teken
- Een element kan bestaan uit 1, 2, 3 of 4 modules
- Een donkere module representeert een binaire 1
- Een lichte module representeert een binaire 0

Er zijn aanwijzingen dat deze code niet "bullet-proof" is.

- Integraal checkcharacter
 - Nominale module breedte is 0.013 inch (0.33 mm)
 - Minimale module breedte is 0.009 inch (0.23 mm)
- De "Universal Product Code" (UPC) kent drie verschillende versies, aangeduid als "UPC-A", "UPC-D" en "UPC-E". Hiervan is versie A de meest gebruikte. UPC-A kan getallen aan van 12 cijfers, opgesplitst in twee delen van ieder 6 digits. UPC-E is een kleinere "zero-suppressing" variant waarmee compactere codes verkregen worden. Versie D tenslotte kan getallen van variabele lengte aan, maar wordt in de praktijk eigenlijk niet toegepast. Het verschil tussen de bruikbaarheid van UPC-D en de andere twee varianten wordt teniet gedaan doordat deze beide codes aangevuld kunnen worden met 2 of 5 extra cijfers. Deze toegevoegde cijfers worden voornamelijk gebruikt bij het markeren van boeken en tijdschriften.

UPC-A

Versie A wordt in principe alleen gebruikt voor het coderen van 10-cijferige getallen. Dat getal wordt voorafgegaan door een special systeem nummer, dat meer informatie geeft omtrent soort het artikel. Een checkdigit complementeert de code tot 12 cijfers.

Het UPC-nummer wordt altijd leesbaar onder de code geprint, terwijl het systeem nummer vóór de barcode gedrukt wordt. Deze systeem nummers hebben de volgende betekenis:

- | | |
|---|--|
| 0 | Algemeen toepasbaar |
| 2 | Producten van willekeurig gewicht tijdens opslag |
| 3 | Medicamenten |
| 4 | Non-food artikelen voor de verkoop |
| 5 | Coupons en waardebonnen |

De overige nummers (1, 6, 7, 8 en 9) zijn voorlopig gereserveerd voor latere toepassingen.

Het complete UPC-A symbool bestaat uit de volgende onderdelen:

- Start character (101)
- Systeem nummer
- Vijf datacharacters
- Centre character (01010)
- Vijf datacharacters
- Check character
- Stop character (101)

De characters rechts van het centre character zijn logisch gezien geïnverteerd. Een drie die links van het centre character staat wordt bijvoorbeeld gecodeerd als "0111101", als hetzelfde character aan de rechterkant voorkomt wordt 'ie gecodeerd als "1000010".

Van UPC-A zijn ook verlengde varianten beschikbaar waarmee nog twee of vijf extra characters gecodeerd kunnen worden. Deze varianten worden met name gebruikt bij het coderen van tijdschriften. Vaak wordt dan gebruik gemaakt van een halve code waarbij het linker deel van de code weggelaten wordt.

UPC-E maakt zoals gezegd gebruik van zogenaamde "zero suppressing" technieken. Daarvoor moet het te coderen getal wel voldoen aan een aantal eisen. Zo moet het system character altijd "0" zijn en moeten er tenminste vier nullen in de data voorkomen.

Deze UPC-codes zijn voornamelijk in gebruik in de Verenigde Staten van Amerika. In Europa maken we namelijk meestal gebruik van "ons eigen" barcode systeem: EAN 13. Deze code bestaat, zoals de naam al aangeeft, uit 13 cijfers. De eerste twee cijfers geven aan van welk land het produkt afkomstig is, de zogenaamde flag characters. Zo geeft "71" aan dat het produkt uit Nederland stamt. Dan volgen een tiental cijfers die informatie geven over het produkt en tenslotte complementeert een checkcharacter het geheel.

De Japanners wilden graag hun eigen code gebruiken. Nou ja, hun eigen code? De Japanse standaard code heeft "JAN" en is gelijk aan EAN-13 met de flag characters "49".

Barcodes zijn er dus in vele soorten en maten. Iedere barcode heeft zo z'n eigen toepassingsgebied, waarvan sommige barcode systemen een heel eigen leven leken te gaan leiden buiten het applicatiegebied om. Andere barcodes daarentegen (de postale bijvoorbeeld) blijven beperkt tot het produkt waar ze voor ontworpen zijn. Het vervolg op barcodes is natuurlijk OCR, waarover een andere keer wellicht meer.

De bij dit artikel afgedrukte C-code geeft een routine voor het printen van standaard code 39 barcodes. Het voorbeeld programma print de string "UPK-79" op een HP LaserJet compatible printer. Het is mij inmiddels bekend dat de gebruikte caractertables niet 100% correct zijn, maar exacte details daarover zijn nog niet beschikbaar. Correcties "are left as an exercise to the reader"...

Joost Voorhaar

```

/* Dit programma drukt de string "UPK-79" af als code 39 barcode.
* Copyright (c) 1992, J.Voorhaar - Alle rechten voorbehouden
*
* Speciale toestemming is verleend voor het publiceren van deze code in het
* blad "De µP Kenner" van de KIM Gebruikersclub Nederland. Het is niet
* toegestaan deze code te gebruiken voor andere dan puur voor studie bedoelde
* toepassingen. Het verspreiden en/of gebruiken van deze code anders dan
* volgens de copyrightbepaling zoals die voorin de µP Kenner is vermeld
* is niet toegestaan.
*/

#ifdef UNIX
#include <sys/types.h>
#include <unistd.h> /* For UNIX only! */
#endif

#include <stdlib.h>
#include <string.h>
#include <stdio.h>

#ifdef UNIX
#define PRN "/dev/lp0" /* The printer device */
#else
#define PRN "output.pcl"
#endif

#define RES 75 /* Resolution: 75 Dots Per Inch */

/* Printer specific strings: */

#define INIT          "\x1B*t%dR" /* Define graphics resolution */
#define STARTGRAPH    "\x1B*r1A" /* Start graphics */
#define ENDGRAPH      "\x1B*rB" /* End graphics */
#define DATA         "\x1B*b%dW%s" /* Transfer raster data */

/* Barcode patterns according to code 39, including the intercharacter gap: */

static unsigned code[43] = {
    0x14DA,0x1A56,0x1656,0x1B2A,0x14D6,0x1A6A,0x166A,0x14B6,0x1A5A,0x165A,
    0x1A96,0x1696,0x1B4A,0x1596,0x1ACA,0x16CA,0x1536,0x1A9A,0x169A,0x159A,
    0x1AA6,0x16A6,0x1B52,0x15A6,0x1AD2,0x16D2,0x1566,0x1AB2,0x16B2,0x15B2,
    0x1956,0x1356,0x19AA,0x12D6,0x196A,0x136A,0x12B6,0x12DA,0x135A,0x124A,
    0x1252,0x1292,0x1492
};

/* The character conversion array. Only the characters from this array can be
* represented by the standard code 39 system:
*/

static char ascii[43] = {
    '0','1','2','3','4','5','6','7','8','9',
    'A','B','C','D','E','F','G','H','I','J',
    'K','L','M','N','O','P','Q','R','S','T',
    'U','V','W','X','Y','Z',' ','*','','$',
    '/','+','%';
};

```

```

/* Function : void compress(unsigned *bar);
* Purpose : Remove intercharacter gaps from bar image
* Input : bar points to a bar image
* Output : bar points to compressed bar image
*/

void compress(unsigned *bar) {
    char *dest = (char *) bar;
    unsigned *src = bar, lsb = 0x20;
    long oldword, newword, msb;
    oldword = 0;
    while (*src) {
        newword = *src++;
        msb = (long) 1 << (lsb - 1); /* For placing the MSB */
        lsb = 0;
        while (!(msb & newword)) { /* Shift new word in position */
            newword = newword << 1;
            lsb++;
        }
        oldword = oldword | newword;
        while (lsb < 0x18) { /* Get the highest byte */
            *dest++ = (oldword >> 0x18) & 0xFF;
            oldword = (oldword << 8) & 0xFFFFFFF0;
            lsb += 8;
        }
    }
    while (oldword) { /* No more characters... */
        *dest++ = (oldword >> 0x18) & 0xFF;
        oldword = (oldword << 8) & 0xFFFFFFF0;
    }
    for (; dest != (char *) src; dest++)
        *dest = '\0';
}

/* Function : struct image *code39(char *strg);
* Purpose : Convert string to standard code 39 barcode
* Input : strg points to string to be encoded
* Output : code39 points to dynamically allocated image
* Remarks : Returns NULL on error
*/

char *code39(char *strg) {
    unsigned bar[128], cs = 0, *dest = bar;
    int index;
    char *p;
    *dest++ = code[37]; /* Start character */
    for (p = strg; *p; p++) {
        for (index = 0; index < 43; index++)
            if (*p == ascii[index]) break;
        cs += *p;
        if (*p != ascii[index]) return NULL; /* Error, unknown character! */
        *dest++ = code[index];
    }
    *dest++ = code[cs % 43];
}

```

```

        *dest++ = code[37];                /* Stop character */
        *dest = 0;
        compress(bar);
        if (p = (void *) malloc(strlen((char *) bar) + 1))
            strcpy(p,(char *) bar);
        return p;
    }

/* Function : int main(int argc, char *argv[]);
* Purpose : This is the main program. No comment needed.
* Input    : As usual
* Output   : As usual
*/

int main(int argc, char *argv[]) {
    int line,length;
    char *barimage;
    FILE *prn;
    length = strlen(barimage = code39("UPK-79")); /* Convert string to barcode */
    if (prn = fopen(PRN,"wb")) { /* Open the printer device */
        fprintf(prn,INIT,RES); /* Set resolution to 75 DPI */
        fprintf(prn,STARTGRAPH); /* Start graphics mode */
        for (line = 0; line < 35; line++)
            fprintf(prn,DATA,length,barimage);
        /* Write a single line of the image */
        fprintf(prn,ENDGRAPH); /* Close the device */
    }
    else {
        fprintf(stderr,"Cannot open printer on %s\n",PRN);
        exit(EXIT_FAILURE);
    }
    exit(EXIT_SUCCESS);
}

```

Nieuwtjes en andere wetenswaardigheden (2)

Belastingaangifte op/een flop

Met behulp van ons stiefkindje kunnen we al kopen, bankieren enzovoort. Wat nog niet kon was direct een aangifte voor particulieren naar de belasting piepen of een schijfje opsturen. Men is druk bezig met de voorbereiding om dit mogelijk te maken. En omdat we allemaal braaf onze guldentjes hebben afgestaan is het zelfs mogelijk om meteen een 'voice-operated' computer aan te schaffen. Volgens de belastingdienst is het mogelijk om de boel sneller af te handelen door deze automatisering. Ze hebben er niet bij verteld dat ze dan ook onze guldens sneller te pakken hebben.

IBM heeft plannen...

IBM heeft een uitbreiding en vernieuwing aangekondigd van zijn reeks personal computers. Eenentwintig nieuwe PS-modellen en een serie trillingsvrije monitoren staan op ons te wachten. De PS/Value Point, computers met een AT bus en uitgebreide grafische mogelijkheden worden geleverd met een 386SLC of een i486 processor. De goedkoopste versie wordt geleverd met 2MB geheugen en een 80 MB harddisk. Kosten: f 2395,- exclusief BTW. Er komt in de PS/2 serie een vlaggeschip n.l. de Server 295 welke kan dienen als het hart van een netwerk. Ook binnen de Notebooks kunnen we nog wat verwachten van Big Blue. Voor geïnteresseerden is het dus opletten geblazen.

Teletext (deel 2)

Op de print uit het vorige artikel is ook plaats voor een header, deze is voor een opzetprintje waarmee men de RGB-signalen van de SAA5246 terug kan voeren naar de televisie. Men kan dan eigen teletext pagina's op de televisie laten zien. Uiteindelijk is het de bedoeling dat men teletext pagina's kan opnemen, zodat men zelf een titelrol kan maken voor de vakantiefilms. Hier zijn wij zelf nog mee bezig, dus het kan nog niet 100% volmaakt zijn. Verder hebben wij momenteel de volgende programma's voor de Teletextkaart klaar: TT.EXE, TTGET.EXE, TTEDIT.EXE, TTCONV.EXE en TTSHOW.EXE. Ik zal de programma's zo op volgorde summier bespreken aangezien elk onderdeel in de doc-file besproken staat.

TT.EXE

Dit is het eigenlijke teletext-programma, hiermee leest men de pagina's mee binnen om ze te kunnen opslaan, bekijken of te kunnen printen. Men kan quasi gelijktijdig naar vier pagina's tegelijk zoeken. Als men bijvoorbeeld pagina 101 opzoekt het men pagina 102, 103 en 104 ook in het geheugen zitten. De zoektijd is dan minimaal. Ook is de mogelijk aanwezig om wildcards op te geven, bijvoorbeeld 11*, nu worden alle pagina's van 110 tot 119 ingelezen.

TTGET.EXE

Met TTGET is het mogelijk om bepaalde informatie uit een programma te filteren, dit kan handig zijn als men koersen van aandelen bijhoudt. Men kan de gegevens opslaan in een logfile. De output van dit programma gaat naar de standaard output en het is dus mogelijk door middel van redirection (> of >>) de output naar een file te schrijven cq. toe te voegen. De foutmeldingen gaan dan naar de console.

TTEDIT.EXE en TTCONV.EXE

Dit is een demoprogramma van een Teletext-editor, deze is van de firma TSS BV. Het programma heeft beperkingen: er is geen handleiding bij en men kan maar 18 pagina's opslaan. Maar voor ons doel maakt dit niet zoveel uit. Heeft men zelf een pagina gemaakt dan kan men deze door middel van TTCONV converteren naar het formaat wat wij gebruiken bij de teletextkaart. De beperking van 18 pagina's is dan weg. Ook kan men met TTCONV teletextpagina's converteren naar TTEDIT formaat om pagina's te kunnen bewerken.

TTSHOW.EXE

TTSHOW is voor het weergeven van TTEDIT-files op de TV ten behoeve van videobewerking (titels e.d.). Het programma kan 18 TTEDIT-files laden. Er zijn scrolling en beeldmanipulatie mogelijkheden.

Dit zijn zo de mogelijkheden van de Teletextkaart, en aangezien we dit project ook met de computerclub op het bedrijf hebben opgestart hopen we dat er nog meer leuke dingen bij komen. Op het bedrijf zijn er 15 mensen die de kaart gaan bouwen, dus als er bij de KGN-Club ook belangstelling voor is kan er misschien met de aanmaak van printen wat geregeld worden. Degene die belangstelling heeft moet mij maar een briefje schrijven (Jacques Urlusstraat 11, 7558 JV Hengelo) of een berichtje achterlaten bij ons club-BBS "The Ultimate". De sources van de programma's willen we pas over 2 maanden vrijgeven om eventuele fouten er uit te kunnen halen. Verder staat het hele pakket, met uitzondering van de sources, op de "The Ultimate" in de KGN area onder de naam "TELETEXT.ARJ".

Herman Hek

Item:	Quantity:	Reference:	Part:
1	1	C1	1 uF/6.3 V Tantaal
2	4	D1, D2, D3, D4	1N4148
3	1	K1, K2, K3	Header
4	1	P1	5 k
5	5	R1, R3, R5, R7, R9	68 E
6	4	R2, R4, R6, R8	4k7
7	5	T1, T2, T3, T4, T5	BC548

Onderdelenlijst

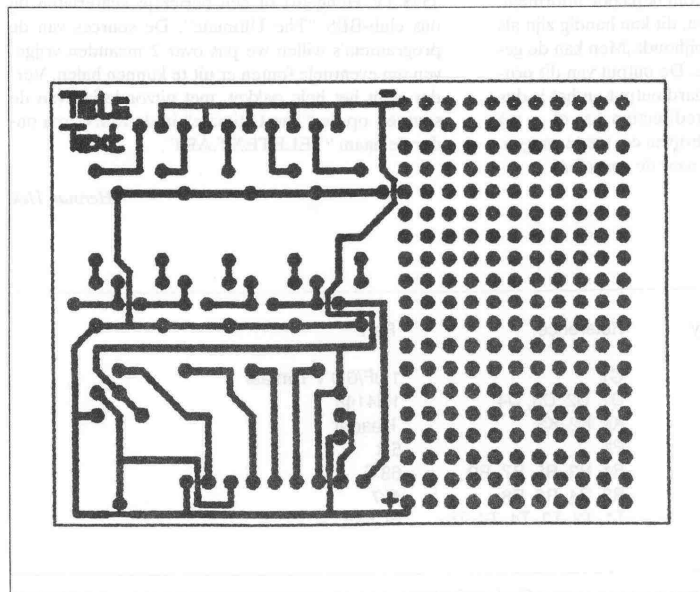
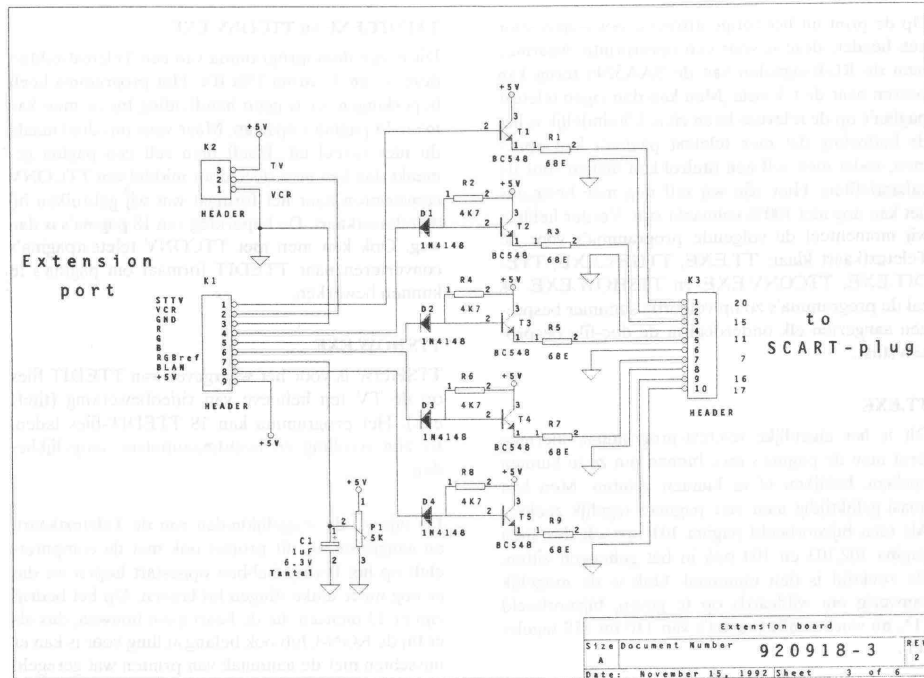


Fig. 1: printje voor het extensionboard

Netwerken op het BBS

Inleiding

Een nadere uitleg van de verschillende netwerken waar het BBS op is aangesloten is waarschijnlijk wel gewenst. Vooral omdat het nu toch best wel een beetje uit begint te groeien zodat ik het zelf soms niet meer overzie.... (foutje????)

In dit artikel komen twee soorten gebieden naar voren. Dat zijn in de eerste plaats de Echo-Mail gebieden en in de tweede plaats de file-area's. Echo-Mail is met name bedoeld voor het versturen van berichten terwijl de file-area's, de naam zegt het al, bedoeld zijn voor programma's etc. Verder heeft de Echo-Mail nog als eigenschap dat een bericht dat in één van deze rubrieken wordt geschreven tijdens nachtelijke connecties wordt verstuurd naar de BBS-sen die ook op die rubriek zijn aangesloten.

FidoNet

Als eerste bestaat er het zogenaamde FidoNet. Het merendeel van de rubrieken op The Ultimate komt vanuit dit netwerk. Even ter verduidelijking, alle verdere netwerken maken eigenlijk ook gebruik van het FidoNet principe c.q. de structuur van FidoNet. Alleen hebben de andere netwerken niets met de policies (regels) te maken die in het FidoNet gelden. Verschillende netwerken zijn dan ook ontstaan doordat 'men' zich stoorde aan de 'regels' binnen het FidoNet-werk.

Maar dat is weer iets anders. Eerst even iets over de rubrieken, met name over de Echo-Mail rubrieken. De rubrieken zijn onder te delen in Nederlands- en anders-talige rubrieken. De Nederlands-talige rubrieken vinden hun oorsprong in Nederland of België. Berichten die men schrijft in één van deze rubrieken zullen dus ook niet verder verspreid worden dan Nederland en misschien ook in België. Anders-talige rubrieken hebben meestal hun oorsprong in het land waar de voertaal hetzelfde is als die van de rubriek, of ze is in het Engels als ze over heel de wereld wordt verspreid.

Er staat een behoorlijke verscheidenheid aan rubrieken op het BBS. Zo zijn er rubrieken die de verschillende programmeertalen behandelen en rubrieken waar men het heeft over netwerken of over communicatie programmatuur.

DENKNET

We zijn ook aangesloten op het zogenaamde 'DENKNET', een netwerk bedoeld voor mensen die als hobby het uitoefenen van een denksport zoals schaken of dammen op de computer hebben. Verschillende probleemstellingen komen daarbij aan de orde. Er zijn aan de berichtengebieden ook verschillende file-gebieden gekoppeld waarin men de verschillende probleemstellingen kan binnenhalen (downloaden) waarna men ze rustig thuis, offline, kan bestuderen.

Gernet

Een volgend netwerk waar we op aangesloten zijn is het Gernet. Dit is het netwerk waar het computerblad C't de drager van is. De sources die in het blad staan afgedrukt worden ook via dit netwerk verspreid, zoals we dat ook doen met de sources die in de μ P Kenner staan afgedrukt. Ook vind je in de betreffende file-rubriek de inhoudsopgaven van de C't. Zo kan men ongeveer een halve maand voordat de nieuwe C't uitkomt al op het BBS bekijken welke items in de volgende C't ter sprake komen.

De voertaal van de berichten is vanzelfsprekend Duits.

Programmers Network Netherlands

Een netwerk waar we ook op zijn aangesloten is het PNN netwerk. PNN staat voor Programmers Network Netherlands. Een netwerk waar we door onze achtergrond wel in thuis horen leek ons.

Ook hier hebben we een keur aan onderwerpen waar de gebruiker van het BBS gebruik van kan maken. Zo bestaan er Echo-Mail area's met de volgende onderwerpen:

- Algemeen
- Unix
- Linux
- Netwerken
- Pascal
- C
- Electro

Voor ieder wat wils dus die op het BBS inlogt.

Aan file area's biedt ons dit netwerk de volgende rubrieken:

- Unix
- Linux
- Netwerk-software
- Pascal
- C

Dus ook de nodige rubrieken om in de gaten te houden indien men zich interesseert voor één of meer van deze items.

ShareWare Distribution Network

Een netwerk waar we al erg lang mee verbonden zijn, naast de aansluiting binnen FidoNet is het SDN-netwerk (SDN :: ShareWare Distribution Network). Binnen dit netwerk bestaat er een berichtenrubriek waar nieuwe share-ware programmatuur besproken wordt of waar men vragen kan stellen over share-ware programmatuur die via het SDN netwerk is verspreid. De file-gebieden hebben ook een behoorlijke verscheidenheid aan keuzes. Een minpuntje is misschien dat er vaker programmatuur wordt verspreid die specifiek bedoeld is voor de Amerikaanse gebruiker. Het netwerk heeft namelijk een Amerikaanse oorsprong.

Zo kunnen we het volgende via SDN aanbieden, wel allemaal voor de PC reeks bedoeld:

- Programmatuur voor de zakelijke omgeving, helaas veelal afgestemd op de Amerikaanse markt.
- Communicatie software
- DataBase en Spreadsheet programmatuur
- Grafische programmatuur, ook voor de allerkleinsten zit er leuk spul tussen.
- Diverse utilities, puur om het werk op de PC te vereenvoudigen, b.v. een makkelijk kopieer-programma of een harddisk menu-programma.
- Tekstverwerkers of utilities daarvoor.
- Miscellaneous files over van alles en nog wat
- Spelletjes, voor de allerkleinsten of voor de gevorderde PC gebruiker
- Teksten over de meest verschillende onderwerpen.

Al deze programmatuur heeft als extensie (achtervoegsel) .SDN. Dit staat voor de compressie methode die men gebruikt heeft. Het decomprimeer-programma dat de gebruiker dient te gebruiken is PAK251.EXE. Dit programma staat ter download in het hoofdmenu van het BBS.

Naast de meeste files in deze rubrieken staan er zogenaamde .SDA files. Deze files hebben veelal de zelfde naam als de .SDN file. Ze horen dan eigenlijk ook bij elkaar. De .SDA file bevat namelijk een uit-

leg over de .SDN file, eigenlijk een korte beschrijving van/over de .SDN file. Deze files staan niet gecompriemerd op het BBS en kunnen dus bekeken worden voordat men een .SDN file gaat downloaden. De .SDN files zijn namelijk soms nogal groot en het is dus wel aardig als men eerst kan kijken wat men eigenlijk gaat binnenhalen (downloaden). De file is gewoon te bekijken via het BBS. Of men download eerst de .SDA file om dan offline eerst te lezen wat voor een file men wil gaan downloaden.

Diversen

Tenslotte zijn we nog aan diverse andere file-netwerken aangesloten zoals:

- DVN :: DesqView Network, programmatuur dus waar men utilities of patches aanbied voor onder een DesqView omgeving.
- WINNET :: Windows Network, een netwerk waar men utilities aanbied voor windows, zoals printer-drivers, Grafische drivers (b.v. drivers voor verschillende VGA kaarten/versies), spelletjes, programmatuur enz.
- SND :: Sound Network Distribution, programmatuur voor verschillende geluidskaarten.
- ANSI :: Verschillende programmatuur voor ANSI-omgevingen. Veel plaatjes en ook de programmatuur waarmee men zulke plaatjes kan maken. Met of zonder geluid.
- CLN :: Clipper Network, programmatuur en verschillende utilities voor een clipper omgeving.
- ADS en SKY netwerk :: Dit zijn beide Amiga netwerken via welke ook weer een verscheidenheid aan programmatuur binnenkomt. Per week ongeveer een 500kb tot 1Mb aan software.
- AST Network :: Een Atari File netwerk waar helaas niet erg veel voor binnenkomt. Maar ook voor de Atari komen er dus geregeld nieuwe files binnen via dit netwerk op het BBS.
- VGA-Net :: Een netwerk waarin van allerlei VGA plaatjes en utilities worden verspreid. Van mooie vrouwen tot auto's of schilderijen die men gedigitaliseerd heeft.

Naast al die soorten van Echo-Mail en file netwerken hebben we ook nog een hele ander schare aan rubrieken. Zo hebben we in de loop van de tijd dat het BBS in de lucht is een hele verzameling aan software verzameld en we proberen die altijd te verde-len over de verschillende rubrieken zoals:

- Algemeen informatie
- comprimeer tools
- Communicatie pakketten
- PC utilities weer onderverdeeld in:
 - File/Disk utilities
 - Tekstverwerkers en utilities hiervoor

- Spelletjes
- Keyboard -
- Printer -
- Display -
- Memory utilities

Eigenlijk teveel om op te noemen.

The Ultimate

Men prijst me wel eens voor de verscheidenheid aan programmatuur die op het BBS te vinden is, maar zegt er tegelijk bij dat het eigenlijk teveel is. Men kan de software niet meer vinden, zo iets van: Men ziet door het bos de bomen niet meer. Tja, een grote verscheidenheid aan programmatuur brengt dit waarschijnlijk met zich mee. Tenslotte zitten we op het moment met een 185 file rubrieken verdeeld over zo'n 500Mb aan disk-ruimte en groeien gestaag verder. Zijn we zo'n 6 jaar geleden begonnen op een XT met 20Mb en één (1) telefoon lijn. Nu zitten we te werken met twee 386 DX systemen aan elkaar gekoppeld via netwerkkaarten en hebben we vier (4) telefoonlijnen en 1Gb (1.000Mb) aan diskruimte. Dat wil gelukkig niet zeggen dat we het zo slecht deden in het begin van het BBS. 6 Jaar geleden was het gewoon niet nodig (en niet te betalen!) om zoveel diskruimte aan je BBS te hangen. De programmatuur was waarschijnlijk ook beter geschreven waardoor je niet zo ontzettend veel ruimte nodig had.

Een praktisch programma nam niet meer dan een paar tot hooguit enkele 10-tallen kb's in beslag, terwijl tegenwoordig met de hele PC/AT reeks en Macintosh machines, de Amiga 4000 en vooral de grootte van het interne geheugen en daarmee samengaan de grootte van de harddisk de schrijvers

er niet meer voor terug deinzen programmatuur te schrijven die snel 500Kb tot vele Megabytes diskruimte in beslag neemt. Vooral een grafisch interface neemt nogal wat geheugen in beslag.

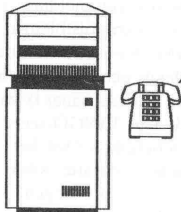
We worden als maar groter wat we natuurlijk perfect vinden. De illusie dat we de grootste van Nederland zouden kunnen worden hebben we echter al heel snel laten varen. Er bestaan andere BBS'en - echt professioneel opgezet- met 32 of 64 telefoonlijnen en meerdere Giga-bytes online dus dat kunnen we gevoeglijk vergeten. Maar daar gaat het tenslotte ook niet echt om. Bewijst het BBS in deze vorm z'n nut of niet, dat is het eerste waar we aan moeten denken. En ons inziens lukt dat aardig. Het BBS wijst niet alleen nieuwe mensen de weg naar onze vereniging, de donateurs die het BBS ook nog eens rijkelijk is steunen het BBS dat hun uitstekend bevalt, waardoor de uitgaven van de vereniging voor het BBS verantwoord worden besteed en toch in het redelijke gehouden kunnen worden.

Ook aan Echo-Mail rubrieken loopt het steeds verder op. Op dit moment is de keuze ongeveer verdeeld over zo'n goede 100 rubrieken met een verscheidenheid aan onderwerpen. Maar indien je iets zoekt, en je vindt het onderwerp er niet tussen. Laat het dan eens weten, misschien is de rubriek wel voor handen maar hebben we de rubriek gewoon niet aangesloten omdat er nog geen animo voor was. Een verzoek om de rubriek aan te sluiten wordt nooit afgewezen. Met die uitzondering natuurlijk dat ze ook ergens te krijgen is.

Uw Sysop, Jacques Banser

BBS "The Ultimate"

For all systems





Tel. 053-303902 & 053-328506

- 053-303902 (2 lijnen!) -

V22, V22bis, V23, V32bis 14k4/HST, V42bis, MNP5

- 053-328506 -

V21, V22, V22bis, V23, V32bis, 14k4/HST, V42bis, MNP5

Inleiding in computer-programma's

Van computers wordt ondermeer het volgende gezegd: "Computers zijn foutloos", "computers worden nooit moe", "computers zijn makkelijk".

Iedereen, computer-kennis of niet, kan hier wel aanvullingen op verzinnen. Het is natuurlijk een koud kunstje een ding, dat ergens maar wat staat te trillen en te zoemen en waarop regelmatig lampjes aan en uit gaan, maar waar verder niets mee gebeurt allerlei fantastische capaciteiten toe te dichten. Wat dat betreft lijkt het wel op de aankondiging voor een nieuw TV-seizoen of dat onvergetelijk schitterende diner. Je wordt lekker gemaakt voor allerlei bijzonder mooie programma's/lekkere hapjes en wat je uiteindelijk krijgt is een kleffe variatie op een oud afgekauwd thema respectievelijk een zakje friet in een donkere hoek van een stinkende kelder.

Welnu, hetzelfde geldt ook voor computer-programma's. Je wordt lekker gemaakt door een stelletje vlot babbelende nitwits, die misschien toevallig(?) redelijk veel overeenkomsten bezitten met gladdes autoverkopers, die alle voordelen van een computer(-programma) beschrijven, maar nooit de nadelen melden ten opzichte van de vorige uitgave. Zo'n klein en lief bordeaux, dat je vroeger op een klein stukje geparfumeerd papier schreef, vereist tegenwoordig eerst, dat je de printer (duur woord voor een typemachine met een kabeltje voor aan een computer) en computer aan moet zetten, dan tenminste vijf minuten moet wachten tot de computer zijn potloodpunt geslepen heeft en dat je dan met een doosje met een draadje eraan allerlei magische bewegingen moet maken op een komisch matje om de computer duidelijk te maken, dat "Lieve L. vanwege de computercursus van vanmiddag eten we een uurtje later".

En als je het papiertje ruikt, is het Ozon, dat met karrevrachten van printers afkomt. Om over de herrie van die afdrukkers maar te zwijgen! (En dan klaagden we vroeger over een piepend krijtje of een krassende pen!) Iedereen praat over de voordelen van "het nieuwe programma KWISPELDOOR", dat nog makkelijker en mooier is dan het "antieke" programma "BUSKRUIT". Maar als je dat computergedoe een paar jaren gevolgd hebt, dan merk je, dat de meest fundamentele zaken met een computer niet sneller en beter zijn te doen, maar alleen maar langzamer, slechter en niet zelden onmogelijk, omdat "iemand de handleiding heeft meegenomen". Andere smoezen variëren van "de computer begrijpt me niet", "dat kan niet meer in deze versie

zonder XYZ-kaart", "dat werkt niet in deze configuratie", tot "daar heb je het ABC-programma voor nodig". En dan die opmerkingen in de trant van "dat is zo makkelijk", "daar heb je tegenwoordig maar twee programma's voor nodig..."

Heeft U nog nooit de neiging gekregen om de maker van deze opmerking naar het hoofd te gooien, dat je vroeger rekensommen, fiches voor je kaartenbak en proefwerken maakte met één en dezelfde vulpen?

En alles kost geld, veel geld. Kostte vroeger een blokkootje 50 cent, waarop je je boodschap achterliet, tegenwoordig moet er gelijk een "floppy" van gemaakt worden, want je weet maar nooit.... Dat zo'n "floppy" dan gelijk een gulden of zes kost, vertellen ze er niet bij. En dan die flauwekul verhaaltjes van "er passen 1,4 miljoen tekens op deze diskette, dat

zijn ongeveer 350.000 bladzijden".

Hoe komt het dan, dat ik maar drie berichtjes (met melding in de trant van "ik ben vandaag ziek", "de computer heeft vandaag geen zin", "er is een lampje van de computer kapot en nu weigert de ceepeejoe-curverclutch" op zo'n ding kwijt kan? Tal van praatgroepen zijn voor deze miscommunicatie geformeerd, die ze dan nog eufemistisch "computer gebruikersgroepen" noemen. Maar daarover een andere keer meer.

Maar goed, deze keer willen we het over computer-programma's hebben. Zoals in de inleiding reeds gezegd, "computers zijn foutloos". Helaas geldt hetzelfde niet voor de programma's noch voor de gebruikers, zij allen vertonen wat men noemt "menselijke trekjes". Dus de computer houdt op foutloos te zijn, zodra er programma's om de hoek komen kijken. Helaas is een computer zonder programma zoiets als een vis zonder water, een auto zonder motor, een man zonder vrouw (omgekeerd schijnt niet te gelden!) en een kip zonder kop. Elke computer is hulpeloos zonder programma. Nu heb ik wel eens geprobeerd het wasprogramma van mijn vol-auto-maat in de computer te gooien, maar het enige wat eruit kwam was een melding "Er is een blauw grijze sok te veel en een grijs blauwe sok te weinig" en het verzoek om een wasverzachter met mint smaak. Ook heb ik weleens een TV-programma in de computer geprobeerd te doen, maar ik zal U maar niet vermaken met de lage LINGO-score, die mijn partner en ik mochten bereiken. Ook het partij-programma van een politieke federatie wilde mijn computer niet naar hogere sferen brengen. Toen heb ik een programmaboekje van een klassiek nieuwjaarsconcert

"Ik ben vandaag ziek"

in de computer gestoken, maar het enige resultaat was vuurwerk in en niet op de beeldbuis.

Het blijkt dat de computer speciale "programma's" nodig heeft, zogenaamde computer-programma's. Die dingen kun je in dezelfde winkels kopen als de computers. Maar om het makkelijk en doorzichtig te houden zijn er verschillende soorten en maten. De ene past niet in de andere computer en als ze dan al wel in de computer passen, dan kan hij ze niet l.e.z.e.n.!! Wie is er nu een analfabeet, de computer, die niet kan l.e.z.e.n. of de mens, die dat wel kan, en daarom juist niets van computer wil weten? En daar kom je er achter, dat je voor de zoveelste keer genept wordt. De meeste computer-programma's bestaan uit een handvol diskettes en een grote doos met een handleiding en als je geluk hebt wat voorbeelden.

Nu had ik net aangenomen, dat een diskette een gulden of zes kost. Een computerboek krijg je voor een gulden of vijftig bij een goede kantoorboekhandel, dus dat betekent, dat een computer-programma zou dus rond de fl. 75,= mogen kosten. Was dat maar waar! Een programmaatje waarmee je een memootje kunt schrijven heb je voor "maar" fl. 900,= ! En dat programmaatje heet dan een tekstverwerker (of wordprocessor in het engels), terwijl het programma helemaal niets verwerkt, hoogstens haar/zijn ongelukkige jeugd! Stop een paar wortelen in een food-processor van een paar tientjes, schakel het apparaat aan en je krijgt de heerlijkste gezonde worteltjessap. Probeer via een wordprocessor van een paar duizend gulden een berichtje achter te laten voor je partner, dat je wat later thuis komt van je computer-cursus, dan heeft die ondertussen de afwas al gedaan, voordat je de tekst aan de computer kan onttrekken. Nee, dan voel je je pas goed genept.

Een ander geintje, die ze met computer-programma's uithalen is "het verouderen". Op het moment, dat je een computer koopt voor je bij elkaar geschrapte centjes, is de computer al verouderd en onderdelen? Al een eeuwigheid (= drie dagen) niet meer leverbaar, of de leverancier is failliet of woont in een ver en vreemd land. Nu, bij computer-programma's is het nog een graadje erger. Op het moment, dat je een computer-programma koopt, is men in Amerika alweer drie drukken verder, heeft een computer-blad je programma afgekraakt als "worst

product ever called a product", is het vanwege de vele programmeer-fouten elders uit de handel genomen of vanwege de slechte invloed voor kinderen beneden de 65 jaar verboden. Een programma, dat je een paar jaar geleden gekocht hebt en na vele avonden oefenen voldoende beheerst om er dat briefje aan je kennis in Amsterdam te sturen, wordt door de handelaren beoordeeld als "ANTIETK", waarmee ze alleen maar willen aangeven, dat als je niet het nieuwste en het beste in huis hebt en gebruikt, ook al is het voor een eenvoudig memootje, je niet meetelt. Dat haalt je de kokkok! Iemand die een tekstverwerker (d.i. een memo-maker) als WordnootPerfekt in alle versies vanaf het begin aangeschaft heeft, is een vermogen kwijt, in tegenstelling, tot de verstandige memo-maker, die na versie 3.0 van WordnootPerfekt de andere betere versies aan zich voorbij heeft laten gaan. Hij is een gelukkig mens, die ook geen nieuwe computer nodig

heeft, omdat zijn oude computer (ondertussen een jaar of acht oud) nog voldoende snel is en voldoende mogelijkheden heeft voor allerhande zaken. Dat hij weet dat het werkt, dat hij weet hoe het werkt en dat hij zelfs begrijpt hoe het werkt (om over het plezier maar te zwijgen!) speelt geen enkele rol. Het programma is groter geworden, beter, uitgebreider, etc. Dat je van de 200 extra toegevoegde functies van WordnootPerfekt er welgeteld één gebruikt (een klokje wat rechtsboven in beeld de

juiste tijd aangeeft met een piep bij elk half uur) en de rest nooit, mag geen naam hebben, want "er zijn honderden gebruikers die juist vanwege deze mogelijkheden (met de nadruk op deze) pakket QWERTY hebben aangeschaft. Dat dit uitsluitend onproductieve computer-testers zijn, die honderden berekeningen door een computer laten uitvoeren, ettelijke uren kunnen testen en doorzeuren over een beetje, een beetje, een beetje. Maar deze schrijven echter nooit een geurend memootje aan hun partner vanwege een uitlopende computer-cursus.

Een kennis van mij noemt computer-programma's net medicijnen voor ouderen. Als je geneesmiddel A moet slikken, waarbij je dan weer bijwerkingen krijgt, die door B worden opgeheven, waarvan de bijwerkingen uitsluitend door middelje C te dragen zijn en zo door, tot je middelje Z krijgt, dat je alleen maar moet slikken, omdat je lichaam zoveel pillen per dag anders niet kan verdragen. Datzelfde geldt ook voor computers. Eerst koop je een computer, dan heb je een programma 1-2-3 nodig, dat pro-

Dat je van de 200 extra toegevoegde functies van WordnootPerfekt er welgeteld één gebruikt en de rest nooit, mag geen naam hebben.

gramma heeft weer een ander programma 4-5-6 nodig, wat alleen maar werkt, totdat je "YAHTZEE" hebt, etc.

Uiteindelijk heb je dan een grotere (lees 4x zo dure) computer nodig, dan ze eerst zeiden, omdat het programma, zoals jij dat wilt gebruiken niet zo eenvoudig op dat goedkopere, nu niet meer in te ruilen model, wil werken. "Want U wil wel iets erg bijzonders" is hun reactie, als je alleen maar een computer wil hebben om dat briefje aan je partner te schrijven, dat je vanwege een computer-cursus vanavond later thuis bent. Is het nu verwonderlijk, dat zelfs bij één van de door mij gebruikte computer-programma's in de (Engelse) handleiding staat: "De werking van dit

programma is niet gegarandeerd. De enige garantie die de makers van dit programma U geven, is dat het disk-ruimte vult op Uw computer". Nee, van programma's moet je het duidelijk niet hebben op je computer. Een volgende keer meer over de verschillende soorten computer-programma's.

MacBart!

P.S. Ik ben in generlei wijze verantwoordelijk voor de hersenschudding van mevr. Jansen - Pietersen. Een toetsenbord is wel geschikt om te gaan windsurfen, maar vergeet dan niet eerst de toetsen waterdicht te maken!!

Hello World in PostScript

Het heeft me altijd verbaasd dat er zoveel informatie naar een P(ost)S(cript) device gaat, om een paar woorden te voorschijn te krijgen. Nieuwsgierig geworden wilde ik toch weten wat er minimaal naar de PostScript interpreter moest om "hello world" op papier te zetten.

Om de zaak onder mijn controle te krijgen had ik op de printer eerst het netwerkgedeelte (Apple talk) uit-, en de RS232 poort ingeschakeld. Noteren van de werkende instelling, om weer terug te kunnen, hoefde niet, want dat was de default configuratie. De instelling baudrate, parity, aantal stopbits, X-On/X-Off en dergelijke werden ook ingesteld in het terminalemulatieprogramma. De kabel die normaliter aan mijn modem zit had de verkeerde pin-out. Pin 2 & 3 gekruist en pin 7 voor de ground één op één verbonden. "Het is Software handshake" dacht ik, "dan zou het weleens kunnen werken"

Op het moment van het insteken van de connector begon de ventilator van de printer te draaien. "WAITING" stond er op het status-schermpje in het frontpaneel. Tja, ik weet zo maar een commando waar ik meteen resultaat van kan verwachten: **showpage**.

Niets, "local character echo ON" in terminalprogramma zodat ik niet blind behoefde te typen. Nog een keer **showpage** maar helaas weer geen resultaat.

Hoe begint de printer eigenlijk na een power-up? Het kan natuurlijk zijn dat er troep binnen is gekomen bij het insteken van de kabel. Kabel laten zitten en de printer uitgezet en even later weer ingeschakeld. "IDLE" gaf het status-schermpje aan. Na het indrukken van de s' begon de fan weer te blazen en

status veranderde in "WAITING". De rest van #showpage# in gegeven en gevolgd een return. In het frontpaneel verscheen "PRINTING", er werd een blad gepakt en op mijn gezicht kwam een glimlach.

Dan eens een andere opdracht **bras**.
Van de machine kreeg ik het volgende antwoord:

```
%%[ Error: undefined; OffendingCommand:
bras ]%%
%%[ Flushing: rest of job (to end-of-file) will be
ignored ]%%.
```

Ja, **bras** zal echt wel ongedefinieerd zijn. De rest wordt dus genegeerd, controleren we even: **showpage** en reactie. Toen wist ik voortaan ook hoe een foutmelding van een PostScriptprinter er uitziet: %%[dan de melding en afgesloten met]%%.

```
/Courier findfont 14 scalefont setfont
100 680 moveto
(Hello world) show
showpage
```

Toen was het zover, ik kon het afgebeelde programmaatje uit uitproberen. Dan blijkt dat PostScript behalve machine-onafhankelijk ook nog leesbaar is voor de mens.

Literatuur:

1. David A. Holzgang: Understanding PostScript, third edition, 1992, ISBN #0-7821-1059-2

Geert Stappers

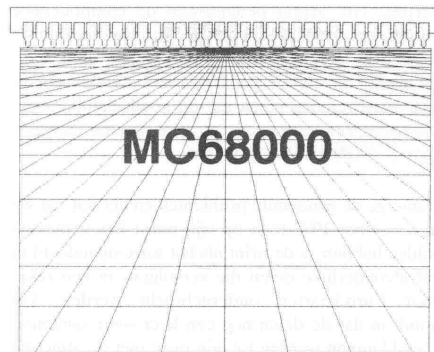
De huidige status van KGN68k

Inleiding

In september 1990 is, tijdens een clubbijeenkomst in Haarlem, door enkele leden van de KIM Gebruikersclub Nederland voorgesteld een op een 68000-gebaseerde computer te gaan bouwen. Dit naar aanleiding van een voordracht over Unix en Minix. Binnen de club bestond namelijk al langer de wens, bij wijze van opvolger van DOS-65, het binnen de KIM Gebruikersclub ontwikkelde 6502 systeem, een zelfbouwproject op te starten rond een lid van de 68000-familie. Voorwaarde was dan wel dat er op dit systeem een operating system draait dat in source beschikbaar is en dat het systeem te combineren is met goedkope en goed verkrijgbare hardware.

Binnen enkele weken is er toen een projectgroep gevormd die dit idee verder uit moest gaan werken waarbij het uitgangspunt zou moeten zijn dat er geen hardware ontwikkeld zou worden die goedkoper in de winkel te verkrijgen is. Die laatste opdracht kan vervuld worden door een CPU-kaart te bouwen die als insteekkaart in een AT-machine zou kunnen werken om op die manier gebruik te kunnen gaan maken van de beschikbare (MFM-)harddisk controllers, video-kaarten etcetera. Verder is er binnen de projectgroep de eis gesteld dat de machine gebaseerd moet zijn op hedendaagse technologie en een performance moet hebben die vergelijkbaar is met de nu gangbare machines. Verder moet de machine natuurlijk door de leden betrouwbaar nagebouwd kunnen worden.

Welnu, de projectgroep is met deze zaken aan het werk gegaan en heeft een concept bedacht waarin een 68030 CPU-kaart samen kan werken met de periferie die in een normale AT aanwezig is. Het probleem was echter dat een AT-insteekkaart bij lange na niet voldoende printruimte biedt om het ontwerp er op onder te brengen. Bovendien bleek het interfacen van de Motorola bus met de AT-bus ook nog enkele technische problemen te geven, met name de refresh van het in de AT aanwezige geheugen, waarvoor wel oplossingen bedacht waren maar die niet of nauwelijks te testen waren. Dit alles maakte het bouwen van een KGN68k in de vorm van een insteekkaart een hachelijke onderneming. Het zou je namelijk kunnen overkomen dat je duur-ge-



Binnen de club bestond namelijk al langer de wens, bij wijze van opvolger van DOS-65, een zelfbouwproject op te starten rond een lid van de 68000-familie.

kochte AT letterlijk in rook opgaat omdat de 68030 en de AT via de bus ruzie met elkaar krijgen.

In de zomer van 1991 is toen bedacht dat het waarschijnlijk beter is het concept van de insteekkaart te verlaten en over te gaan naar een KGN68k computer van het formaat "Moederbord" waarin dan eventueel AT-bus interfacekaarten gestoken kunnen worden. Het concept voorziet zelfs in een opstelling waarin zowel kaarten met een AT-

connector als kaarten met een connector op de voor KGN68k bedachte bus ondergebracht kunnen worden. Eén van die kaarten heeft dan voor beide bussen een connector en vormt het bus-interface tussen de AT-bus en de KGN68k bus, qua concept vergelijkbaar met het bridgeboard voor de Commodore Amiga.

Ondertussen was er ook iets veranderd op de markt. Bij de start van het project hebben we wel gesproken over een SCSI-interface voor een harddisk, maar dat was op dat moment absoluut niet betaalbaar. Nu is de situatie op de markt dusdanig dat MFM-drives nauwelijks meer te koop zijn en dat SCSI-drives een stuk goedkoper zijn geworden. Om deze reden is er in het ontwerp voor KGN68k nu een SCSI-interface opgenomen zodat gebruik gemaakt kan worden van een SCSI-drive.

Het moederbord-concept is volledig uitgewerkt en de print voor dit systeem is vrijwel geheel ontworpen. We kregen toen echter grote problemen met het gebruikte pakket voor het maken van printontwerpen. Het aantal pennen aan de componenten maakte het noodzakelijk een pakket van meer dan f 15.000,- in te zetten en er was niemand binnen de vereniging die een dergelijk pakket beschikbaar kon stellen. Bovendien bleek dat de noodzakelijke vierlaagsprint een vermogen zou gaan kosten, hetgeen voor de KGN niet haalbaar was.

Vanwege de genoemde problemen en het feit dat we in november 1992 toch op zijn minst iets concreets wilden hebben, is de print als het ware opgedeeld in vijf afzonderlijke delen die vervolgens in een rekje voor Euro-kaarten ondergebracht werden. Dit houdt in dat de delen nog een keer weer samengevoegd kunnen worden tot één print met de afmeting van een moederbord, maar voor het bouwen van een prototype is dit een zeer goed haalbaar alternatief. De situatie is nu namelijk zo dat een KGN68k bestaat uit vijf printen met de afmetingen van een Euro-kaart waarvan er slechts één, de CPU-print, als kostbare vierlaags print uitgevoerd hoeft te worden. De overige printen zijn dubbelzijdig waarvoor goed low-cost prototypes gemaakt kunnen worden. Voor de CPU-print is onderhand met behulp van Wire Wrap een prototype gemaakt.

Doelstellingen van de projectgroep

De projectgroep KGN68k is eind 1990 door het bestuur geïnstalleerd en heeft een plan van aanpak geschreven waarin de volgende doelstellingen geformuleerd zijn:

- 1: Onderzoek de mogelijkheid om een computersysteem te ontwerpen dat voor een deel uit zelfbouw bestaat en dat door leden van de KGN voor een redelijke prijs samengesteld kan worden.
- 2: Ontwerp de benodigde hardware op een zodanige manier dat het systeem door de leden op eenvoudige en betrouwbare wijze nabgebouwd kan worden.
- 3: Het systeem dient gebaseerd te zijn op hedendaagse technologie en een hoge performance te hebben.
- 4: Zorg er voor dat er voor het computersysteem voldoende software beschikbaar komt in de vorm van een operating system, hogere programmeertalen, tekstverwerkers, communicatie-software etc. Om aan deze voorwaarde te voldoen is de opdracht aan de projectgroep het porteren van het MINIX operating system naar het systeem.
- 5: Zorg voor voldoende documentatie van uitstekende kwaliteit.

- 6: Zorg er voor dat de distributie van de ontwikkelde hardware en de software en documentatie binnen de KGN en daarbuiten goed geregeld is.
- 7: Aangezien het project binnen de KGN uitgevoerd wordt, dient het project te (blijven) passen binnen de doelstellingen van de vereniging:
 - 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 microcomputers in de meer technische toepassingen.

Toelichting

Ad.4: Hoewel in het "Plan van aanpak" voor dit project staat dat de projectgroep verplicht is het operating system MINIX naar het computersysteem te porteren is uit latere discussies met het bestuur en de leden wel vast komen te staan dat ook gekozen mag worden voor een andere UNIX-lijke, bijvoorbeeld LINUX.

Uitwerking van de doelstellingen

1: Hardware

Op basis van de doelstellingen is een computersysteem ontworpen dat in zijn basis-configuratie zal bestaan uit de volgende onderdelen:

- 1: Een processorkaart met een Motorola 68030 processor. Het ontwerp voorziet in een klok-snelheid van 16 à 20 MHz.
- 2: Een dynamische RAM-kaart waarop SIMM's (uit de MS/DOS wereld) geplaatst kunnen worden. Hierbij kunnen zowel 1 MB SIMM's als 4 MB SIMM's worden gebruikt. Het ontwerp voorziet in de mogelijkheid 1, 2, 4 en 8 SIMM's te plaatsen waarbij het systeem zijn beste performance zal halen bij de plaatsing van 8 SIMM's. De inhoud van de gebruikte programmeerbare logica (PALs) is afhankelijk van de memory configuratie zodat bij uitbreiding van het geheugen de programmering van deze bouwstenen gewijzigd moet worden. De dynamische RAM-kaart wordt in een sandwich-constructie tegen de processorkaart geplaatst. Deze sandwich wordt RAMBO 68k genoemd.
- 3: Een kaart met daarop een groot aantal zaken:
 - 32 .. 512 kB EPROM. Deze bevat een Power On Self Test, de boot-routine en een monitor.
 - 32 .. 128 kB battery backed up statisch RAM. Deze RAM bevat een aantal setup parameters. Verder wordt het statisch

RAM tijdens opstarten gebruikt als geheugen zodat het dynamische RAM getest kan worden.

- Een Dallas Real Time Clock. Dit Real Time Clock-ic verzorgt bovendien het statische RAM van een battery backup.
- Serieel I/O in de vorm van twee RS-232 aansluitingen. Hiervoor wordt een Motorola 68681 DUART ingezet.
- Parallel I/O in de vorm van een Centronics aansluiting. Deze poort kan zowel geprogrammeerd worden als Centronics uitgang en als Centronics ingang. Voor deze aansluiting wordt de Motorola 68230 PI/T ingezet die bovendien een programmeerbare timer heeft waarvan de interrupt als system tick gebruikt kan worden.

Vanwege de combinatie van opgesomde zaken heet deze kaart de COMBO 68k.

4: Een kaart met daarop de interfaces naar het massa-geheugen. Hierop zijn de volgende aansluitingen aanwezig:

- Een floppy disk interface rond een Western Digital 1793. Dit interface ondersteunt de formaten 360 kB, 720 kB en 1,44 MB per floppy.
- Een SCSI interface rond de 53C80 SCSI-controller.

Vanwege die Disk I/O die door de kaart ondersteund wordt, hebben we de kaart DOMBO 68k genoemd.

5: Een kaart waarop volgens het huidige ontwerp alleen de 68881/68882 floating point co-processor geplaatst wordt. Eventueel kan op deze kaart ook nog een dynamische afsluiting van de bus op worden ondergebracht. Deze kaart is in een basis-configuratie niet direct noodzakelijk. Mocht er nog printruimte overblijven, dan kan dit bijvoorbeeld gebruikt gaan worden voor een keyboard-interface en/of een eenvoudige video controller.

Vanwege zijn goede MATHematische eigenschappen, wordt deze kaart MAMBO 68k genoemd.

Nadat het basissysteem gebouwd is, kunnen er nog allerlei uitbreidingen bedacht en ontworpen worden. Hierover zijn al wel een aantal ideeën geformuleerd maar welke uitgevoerd gaan worden wordt uiteraard samen met de bouwers van KGN68k besloten. De volgende mogelijkheden zijn al geformuleerd:

- Een kaart met een resolutie van minimaal de SVGA-standaard.
- Een kaart met een keyboard- en muis-interface

- Een kaart voor aansluiting op een Ethernet netwerk
- Een kaart met "veel" RS-232 aansluitingen
- Een kaart met een MIDI-interface
- Een interfacekaart naar de AT-bus.

Bovendien bestaat nog steeds de mogelijkheid de vijf kaarten uit het basissysteem weer samen te voegen tot één grote, dure, vierlaags print. Indien er binnen de club voldoende van dergelijke prints afgenomen zullen gaan worden, ziet de projectgroep het als haar taak ook deze opdracht te voltooien.

2: Software

Voor de KGN68k wordt gewerkt aan een hoeveelheid software. In de eerste plaats is dat de KGN68k assembler die in staat moet zijn alle instructies en adresseermogelijkheden van de 68030 en de 68881/68882 te verwerken. De uitvoer van deze assembler bestaat uit Motorola S-records. De assembler is geschreven in ANSI C draait op bijvoorbeeld een MS-DOS systeem, een Macintosh of onder LINUX.

Het tweede stukje software dat ontwikkeld wordt is een monitor met behulp waarvan de hardware uitgetest kan worden. Deze monitor kan, naast het uitvoeren van een aantal basisfuncties, ook S-records vanaf een seriële poort inlezen. Op die manier kun je op eenvoudige testprogramma's naar de KGN68k uploaden zodat je niet iedere keer een nieuwe EPROM hoeft te bakken.

We bezitten momenteel ook een conversieprogramma waarmee uitvoer uit de GNU C-compiler omgezet kan worden naar S-records die ingelezen kunnen worden in de KGN68k. We zijn hiermee in staat testprogramma's in C te schrijven op bijvoorbeeld een LINUX-systeem die we vervolgens op de KGN68k laten draaien.

De volgende software die ontwikkeld gaat worden zijn de drivers voor de diverse bouwstenen. Het is de bedoeling een deel van die drivers, samen met een Power On Self Test onder te brengen in de boot-EPROM.

Welk operating system gekozen gaat worden is momenteel nog niet helemaal bekend. Zeker is wel dat het een Unix-alike zal zijn en dat de gehele GNU (ontwikkel) omgeving beschikbaar komt. Volgens de huidige inzichten is een LINUX-versie voor de 68020/68030 een hele goede kanshebber. Voorwaarde is wel dat het operating system in source beschikbaar is en dat er geen duizenden guldens voor een licentie neergeteld moet gaan worden.

Huidige status

De huidige status (december 1992) is als volgt:

1: Hardware

- Van alle kaarten in het basisontwerp zijn de schema's gereed.
- Van de CPU-kaart en de combinatiekaart zijn de printontwerpen ook gereed. Van de CPU-kaart is een prototype gebouwd in Wire Wrap en voor COMBO 68k is er een prototype volgens het printontwerp.
- Van de dynamische ram-kaart is het printontwerp gereed. Hiervan is nog geen prototype gebouwd.
- Voor de disk-kaart wordt momenteel een printontwerp gemaakt.
- Het huidige prototype van de KGN68k, bestaande uit de Wire Wrap CPU-kaart en de COMBO, draait op een kloksnelheid van 16 MHz. Benchmarks geven aan dat hij ongeveer net zo snel is als een 10 MHz XT met een NEC V20 processor (Norton SI 4.0). Hierbij moet worden opgemerkt dat het geheugen, bij afwezigheid van de dynamische RAM-kaart, een breedte heeft van 8 bits en dat de wijze waarop dit geheugen benaderd wordt bij lange na niet optimaal is. De verwachting is dat het systeem zeker nog een factor 3 à 4 sneller gaat worden bij 16 MHz terwijl een versnelling naar 20 MHz eventueel ook tot de mogelijkheden behoort.

2: Software

- De KGN68k assembler is in een dusdanig status dat ze goed ingezet kan worden bij de ontwikkeling van de rest van de software. De assembler mist echter nog een deel van de gewenste functionaliteit (macro's, MMU-instructies, instructies voor de 68881/68882, specifieke 68020/68030 adresseermogelijkheden).

- We hebben een conversieprogramma waarmee uitvoer uit de GNU C compiler omgezet kan worden in S-record voor de KGN68k.
- De monitor is in staat Motorola S-records in te lezen en exceptions die eventueel optreden netjes af te handelen. Dit houdt in dat de machine niet gaat hangen als er per ongeluk niet-bestaand geheugen aangesproken wordt.

Toekomst

Zoals de zaken er nu voor staan, zal zo rond februari 1993 van de hele basisconfiguratie een prototype bestaan. Voor dit prototype is, met uitzondering van de CPU-kaart, gebruik gemaakt van de ontwikkelde printontwerpen. Nu de hardware snel vordert, moet er een begin gemaakt worden met de port van een operating system. Dit betekent dat op zeer korte termijn voor een bepaald operating system gekozen moet gaan worden.

Als we inschatten dat het ontwikkelen van de drivers voor de hardware en het porteren van een operating system toch wel zo'n driekwart jaar gaat duren, dan betekent dit dat we zo tegen de winter van 1993 kunnen gaan beginnen met het verspreiden van het systeem.

Mensen die, op grond van wat ze nu gezien en/of gelezen hebben, al besloten hebben een KGN68k te gaan bouwen, worden uitgenodigd contact op te nemen met de werkgroep. We kunnen namelijk nog wel wat versterking in de software hoek gebruiken en, hoe meer mensen meewerken aan de software, hoe sneller het systeem beschikbaar komt.

Namens de projectgroep KGN68k,

Gert van Opbroek

Hier had uw artikel of advertentie kunnen staan!

Geavanceerde datacommunicatie deel 2

Unix (LINUX) en Netwerken

In dit tweede deel wil ik wat nader ingaan op Unix en netwerken. Dit is eigenlijk een uitwerking van de lezing die ik op de LINUX bijeenkomst heb gehouden. Omdat TCP/IP en UNIX netwerken zeer nauw met elkaar verweven zijn zal ik beginnen met (TCP)/IP uit te leggen.

TCP/IP

TCP/IP is een samenvoegsel van TCP (Transmit Control Protocol) en IP (Internet Protocol) en maakt onderdeel uit van de ARPA (Advanced Research Project Agency) "protocol suite". Meestal wordt de hele verzameling protocollen aangeduid met de naam TCP/IP. TCP/IP is dus in het algemeen een verzamelnaam van alles wat er maar mee te maken heeft. De basis voor deze protocollen ligt bij DARPA (Defense ARPA) en is ontwikkeld in een reeks van jaren en nog steeds vinden er uitbreidingen plaats van deze protocollen.

De basis van TCP/IP is het IP protocol en daar beginnen we dus maar mee. IP werkt met datagrammen. Een datagram heeft wel wat weg van een envelop zoals wij die voor brieven gebruiken. Op de envelop zetten we de geadresseerde en de afzender zodat het "netwerk" van de PTT er mee kan werken. De inhoud heeft de PTT niets mee te maken en wordt ongewijzigd door gegeven.

IP datagrammen werken op soortgelijke wijze. De kop van een IP datagram bestaat uit een aantal 32 bits blokken met daarin allerlei zaken die voor het behandelen van het datagram noodzakelijk zijn. De data is voor IP niet belangrijk net zomin als de inhoud van de brief voor de PTT belangrijk is. In figuur 1 staat de opbouw van een IP datagram. Een aantal velden wil ik even kort verklaren.

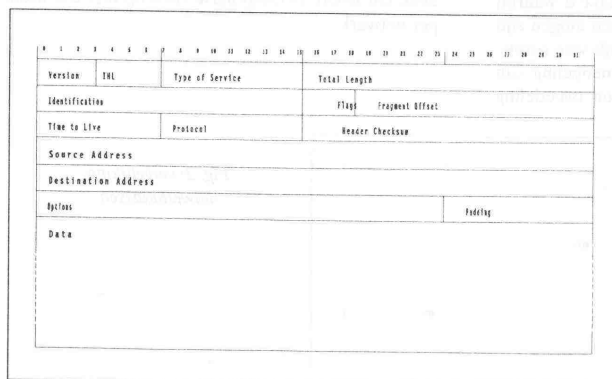


Fig. 1: opbouw van een IP datagram

- Type of Service: Wat voor dienst wordt aangeboden.

Net zoals de PTT allerlei mogelijke diensten kan leveren, kan via IP ook doorgegeven worden wat voor pakket-type het datagram bevat. Zoals bijvoorbeeld Expresse post of aangetekende post kan ook een IP datagram om een bijzondere behandeling vragen.

- Time to Live: Teller die de overlevingstijd bepaalt.

Om datagrammen niet eindeloos te laten rondhangen wordt er bij het verzenden een waarde in dit veld gezet. Elke keer dat er iets met het datagram wordt gedaan zal de inhoud worden verlaagd. Als de waarde van het veld te laag is, zal het datagram worden weggegooid.

- Protocol: Protocol van het ingesloten pakket.

Om bij het verwerken van datagrammen te weten wat er met een datagram moet worden gedaan wordt opgegeven voor welk protocol de data in het datagram bestemd is. Door dit apart te specificeren hoeft de data nog niet bekeken te worden om een beslissing te nemen op basis van het protocol.

- Source Address: Adres waar het datagram (oorspronkelijk) vandaan komt.

- Destination Address: Adres waar het datagram heen moet.

(Adressering wil ik verderop in dit artikel behandelen.)

Nu heeft volgens de officiële definitie IP twee taken te weten:

- adressering
- fragmentatie

Fragmentatie

Wat wordt er bedoeld met fragmentatie? Wel stel je eens voor dat je een IP pakket stuurt van host A naar host B die niet op hetzelfde netwerk zitten. Host A stuurt een IP datagram weg met een inhoud

van 1200 octets. (We mogen officieel niet spreken van bytes want nergens (binnen IP) is gedefinieerd dat de betreffende groepen van 8 bits logische bij elkaar horen). Voor het lokale netwerk bij host A is dat geen probleem want een ethernet pakket kan zo'n IP datagram gemakkelijk verzenden. Maar het netwerk van host B is gekoppeld via een WAN aan het netwerk van host A. Nu is de maximale pakketgrootte van dit WAN slechts 512 octets. Als we de datagrammen niet mogen wijzigen zouden we een probleem hebben want dan is host B niet te bereiken met zulke grote pakketten. Met het overzetten van het LAN naar het WAN wordt de data uitgepakt en gefragmenteerd over meerdere pakketten die ieder voor zich wel over het WAN heen kunnen.

Dit klinkt wat ingewikkeld, daarom zal ik het met een voorbeeld wat duidelijker maken. Stel ik wil een document van tien pagina's naar Geert Stappers sturen. Nu heb ik een beetje haast dus stuur ik het per fax. Ik (host A) stuur die tien pagina's in een enveloppe (datagram) naar Geert. Intern wordt de enveloppe naar de fax machine gebracht. Daar pakt iemand de enveloppe uit en stuurt ze één voor één door de fax (fragmentatie). Bij Geert haalt iemand de tien pagina's weer uit de fax, niet ze samen en geeft ze aan Geert (host B). Het klinkt bij IP wat ingewikkelder omdat we bij dit soort zaken eigenlijk nooit stilstaan hoe we dat doen. We doen het gewoon even.

Adressering

Voor de adressering hebben ze gekozen voor 32 bits adressering, dit zijn theoretisch zo'n 4,3 miljard adressen. Deze adressen worden in de zogenaamde punt-notatie opgeschreven. Dus als a.b.c.d waarbij a,b,c en d willekeurige decimale getallen mogen zijn van 0 tot en met 255. Dit zijn namelijk vier octets. Maar omdat IP is ontstaan uit een mengeling van LAN en WAN is er gekozen voor een tweedeling

van het adres. Dit wordt opgesplitst in een netwerk deel en een host deel. Omdat er soms netwerken zijn die veel hosts aan het netwerk hebben hangen en andere die er maar weinig hosts aan hebben hangen, is er gekozen voor drie netwerk types, te weten:

Klasse A Enkele netwerken met zeer veel hosts aan dat netwerk.

Klasse B Meer netwerken met een redelijk aantal hosts aan dat netwerk.

Klasse C Veel netwerken met een klein aantal hosts aan dat netwerk.

Dit klinkt wat vaag maar ik zal laten zien hoe het werkt.

We mogen officieel niet spreken van bytes want nergens is gedefinieerd dat de betreffende groepen van 8 bits logische bij elkaar horen.

Als het eerste bit een nul is spreken we van een klasse A netwerk. De eerste 8 bits vormen het netwerk deel. De andere 24 bits vormen het host deel. Dit levert ons 126 netwerken op (het eerste bit van de 8 is altijd al nul dus houden we effectief maar 7 bits over) met een theoretisch aantal van 16777214 hosts per netwerk. a staat dus voor het netwerk deel, b,c,d staat voor het host deel.

Als het eerste bit een 1 is en het tweede bit is een nul, dan spreken we van een klasse B netwerk. De eerste 16 bits vormen het netwerk deel. De andere 16 bits vormen het host deel. Dit levert ons 15748 netwerken op met 64516 hosts per netwerk.

Als de eerste 2 bits 1 zijn en het derde bit nul dan spreken we van een klasse C netwerk. De eerste 24 bits vormen het netwerkdeel en de laatste 8 het host deel. Dit levert 1935480 netwerken op met 254 hosts per netwerk.

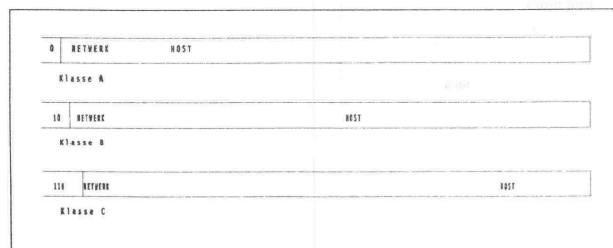


Fig. 2: vergelijking netwerkclassen

Alle andere adressen (dus eerste 3 bits zijn 1) wil ik niet behandelen omdat ze voor speciale doeleinden of helemaal (nog) niet worden gebruikt.

Binnen deze adressen (klasse A, klasse B en klasse C) kennen we nog een aantal unieke adressen. Een netwerkdeel waarvan alle bits nul zijn betekent het 'eigen' netwerk. Als alle bits van het hostdeel nul zijn of 1 dan hebben we met een broadcast te maken. Dan bedoelen we dus alle hosts in het gegeven netwerk.

IP maakt voor de daadwerkelijke verzending van zijn datagrammen gebruik van één of ander soort netwerk. Nu kan dit van alles zijn maar voor een aantal netwerken bestaan officiële specificaties of voorstellen hiervoor. Zo kunnen IP datagrammen onder andere over ArcNET, EtherNET, TokenRing, X.25, FDDI en Seriele lijnen worden verstuurd. Ik zal nu EtherNET als voorbeeld behandelen omdat daar nog een aantal aanverwante protocollen bij komen kijken.

EtherNET werkt met 48 bits adressen. Om een IP datagram te kunnen bezorgen moet er op de één of andere manier een verband worden gelegd tussen IP adressen en EtherNET adressen. Dit wordt gedaan door het ARP (Address Resolution Protocol). Op

elke host wordt een vertaal-tabel bijgehouden van IP adressen en hun bijbehorende EtherNET adres. Dat klinkt leuk maar als ik een computer opstart dan is het logisch om aan te nemen dat deze tabel leeg is. Als we nu een IP datagram willen versturen naar een host op ons netwerk dan gaat dit als volgt:

Host A pakt een IP datagram in een EtherNET pakket in maar weet nog niet het EtherNET adres van host B op hetzelfde netwerk. Host A stuurt dit EtherNET pakket met een EtherNET broadcast adres het netwerk op. Alle hosts zullen het netwerk aanpakken en aan hun IP proces geven. Host B doet dit ook en dus komt dit pakket ook bij host B aan. Host B zet nu in zijn ARP tabel het IP adres van host A en het bijbehorende EtherNET adres. Als host B nu iets terugstuurt naar host A kan dit direct naar het EtherNET adres van host A zodat de andere hosts er niet mee worden lastig gevallen. Host A zet nu het IP en EtherNET adres in zijn eigen ARP tabel.

Maar als we nu aannemen dat een redelijk deel van de IP datagrammen niet naar een lokale host gaan maar naar een systeem buiten het 'eigen' netwerk dan werkt bovenstaand systeem niet meer. Daarom hebben we op deze standaard regel nog een uitbreiding die als volgt werkt:

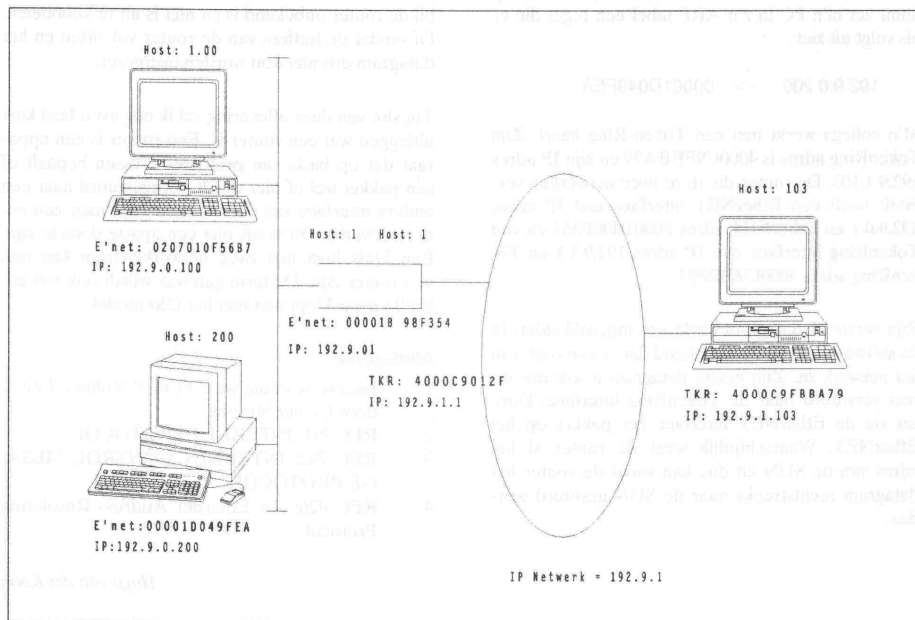


Fig. 3: voorbeeld van een netwerk

Als host A iets naar host C wil sturen buiten het 'eigen' netwerk dan wordt weer een IP datagram ingepakt in een broadcast pakket. Nu reageert echter niet host C zelf maar een router (zie slot) die verantwoordelijk is voor de koppeling tussen het 'eigen' netwerk van host A en alles wat er nog verder ligt op weg naar host C. Op deze wijze kan het gebeuren dat meerdere IP adressen in de ARP tabel hetzelfde EtherNET adres hebben.

Een alternatief voor deze "verre" hosts is om ervoor te zorgen dat host A weet dat gegevens voor "verre" hosts via de router gaan. Dan wordt bij verzenden het EtherNET adres van de router ingevuld voor elke IP bestemming buiten het eigen netwerk.

Ik zal bovenstaande verhaal nu even met wat praktijk gegevens uit ons bedrijf invullen. Ik heb voor mijn PC IP adres 192.9.0.100 en EtherNET adres 0207010F56E7. De SUN heeft IP adres 192.9.0.200 en EtherNET adres 00001D049FEA. Mijn eerste IP datagram wordt verstuurd met EtherNET adres FFFFFFFF en IP adres 192.9.0.200. De SUN onthoudt nu op een regel van de ARP tabel:

```
192.9.0.100  -> 0207010F56E7
```

De SUN stuurt nu gericht z'n IP datagrammen naar m'n EtherNET kaart. Als het eerste pakket terug komt zet m'n PC in z'n ARP tabel een regel die er als volgt uit ziet:

```
192.9.0.200  -> 00001D049FEA
```

M'n collega werkt met een Token-Ring kaart. Zijn TokenRing adres is 4000C9FEBA79 en zijn IP adres 192.9.1.103. De router die deze twee netwerken verbindt heeft een EtherNET interface met IP adres 192.9.0.1 en EtherNET adres 00001B98F354 en een TokenRing interface met IP adres 192.9.1.1 en TokenRing adres 4000C9012F94.

Zijn eerste IP datagram werkt wat ingewikkelder. In de software op zijn PC is bekend dat er een router in het netwerk zit. Zijn eerste datagram wordt dus direct verstuurd naar de TokenRing interface. Deze zet via de EtherNET interface het pakket op het EtherNET. Waarschijnlijk weet de router al het adres van de SUN en dus kan vanaf de router het datagram rechtstreeks naar de SUN gestuurd worden.

Nou klinkt zoiets natuurlijk vreselijk ingewikkeld en krijg je het idee dat het nogal wat tijd extra kost voor het eerste pakket om over die twee netwerken te komen. In de praktijk blijkt dat je geen verschil merkt. De verschillen zitten in de marge in een niet al te vol netwerk. De meeste tijd gaat 'verloren' in menselijke traagheid. De slecht geschreven netwerk software is de volgende in de top tien.

In bovenstaande stuk kwam al het ARP protocol ter sprake. Bij ARP weet een host naast zijn fysiek adres (EtherNET of TokenRing) ook zijn IP adres. Als tegenhanger bestaat daarvoor ook RARP. Hierbij wordt door de host als eerste een broadcast rondgestuurd met een soort 'wie ben ik?' vraag. Een RARP server zal dan het fysieke adres opzoeken in een tabel en een datagram terugsturen naar de host met z'n IP adres. Deze techniek wordt gebruikt in bijvoorbeeld X-Terminals en andere diskless systemen. Hier wordt eerst het eigen IP adres uitgedokterd en vervolgens wordt via het netwerk de software geladen.

Verder zit er als het ware binnen IP nog een protocol. Dit is het ICMP (InterNET Control Message Protocol). Dit protocol wordt gebruikt om informatie tussen hosts uit te wisselen, bijvoorbeeld om een host te vertellen dat zijn datagram niet verwerkt kon worden door het netwerk omdat de bestemming ook bij de router onbekend is en niet is uit te knobbelen. Of omdat de buffers van de router vol zitten en het datagram dus niet kon worden overgezet.

Tot slot van deze aflevering zal ik nog even heel kort uitleggen wat een router is. Een router is een apparaat dat op basis van protocol-adressen bepaalt of een pakket wel of niet wordt doorgestuurd naar een andere interface van de router (en dus naar een ander netwerk). Dit hoeft niet een aparte doos te zijn. Een Unix host met twee netwerkkaarten kan ook een router zijn. De term gateway wordt ook wel gebruikt maar klopt niet met het OSI model.

Bibliografie

1. Internetworking with TCP/IP Volume I en II door Comer Stevens
2. RFC 791; INTERNET PROTOCOL
3. RFC 792; INTERNET CONTROL MESSAGE PROTOCOL
4. RFC 826; An Ethernet Address Resolution Protocol

Hugo van der Kooij

Van de bestuurstafel

De HCC-dagen zijn weer achter de rug. Het vermoeden bestaat dat het dit jaar wel eens positief voor de vereniging zal uitpakken. De interesse was groter dan vorige jaren. Dat kwam volgens mij niet alleen door de grotere drukte maar ook door onze publiekstrekker: de KGN68k-kaart. Zo'n wire-wrap ziet er toch als een kleurrijk breiwerkje uit. Dat het dan ook nog werkt trekt altijd belangstellenden. Onze dank gaat dan ook uit naar de KGN68k-werkgroep die dit mogelijk heeft gemaakt. Met name wil ik twee mensen noemen. Als eerste Geert Stappers die zich letterlijk en figuurlijk krom heeft gebreid om de wire-wrap klaar te krijgen. Als tweede wil ik Pieter de Visser noemen omdat hij het in een paar minuten presteerde om op de HCC-dagen Tretris aan de praat te brengen op de KGN68k. Al met al was het meer dan alleen "Hello world" zoals was gezegd. Dat verhoogde alleen de belangstelling en daarvoor staan we uiteindelijk met zo'n stand. Om over het enthousiasme van de standbemannen maar niet te spreken.

Vorig jaar kon ik nog melden dat er regelmatig vrolijke mannen met een computer voor hun buik langs trokken gevolgd door vrouwen met lange gezichten. Dit jaar was het precies andersom. De mannen wel weer met die computer voor hun buik maar met een gezicht van een heel jaar onweer. De vrouwen er achter aan met een blij gezicht zo van nu heb ik er ook een. Het viel trouwens op dat er dit jaar meer vrouwen waren. Aan hun gezichten te zien denk ik dat er een kentering komt in het aantal computer-weduwen. Het zullen er minder worden, maar het aantal computer-weduwnaars zal stijgen.

Het viel dit jaar wel op dat er weinig gestunt werd met de prijzen. Dat kan ook bijna niet als je ziet dat computers tegen inkooprijzen in de winkel staan. Dan kan er ook tijdens de HCC-dagen niets meer af. De enige stunt die ik gezien heb was een intern 14k4 modem die voor 600 gulden over de toonbank ging. Dat ging dan ook met een snelheid die hier niet te beschrijven valt.

We hebben dan ook nog een week later een clubbijeenkomst gehad. Op deze bijeenkomst was een algemene ledenvergadering waarover je elders in dit blad meer leest. Hier wil ik toch nog even mijn teleurstelling uitdrukken als het om bestuursfuncties gaat. Als er om versterking in het bestuur wordt gevraagd blijft het angstvallig stil. Ik wil jullie er op wijzen dat een vereniging zonder bestuur wordt geen vereniging meer is. Zoveel tijd gaat er ook weer niet in zitten dat dat geen excuus mag zijn om stil te blijven. Gebrek aan ervaring mag ook geen reden zijn om je van een bestuursfunctie te onthouden. Het zittende bestuur zal je met blijdschap helpen om tot een goed bestuurder uit te groeien. We zijn nog steeds opzoek naar iemand die de PR op zich wil ne-

men, maar de hulp van een of meer algemene leden is ook nooit weg. Ervaring opdoen weet je wel.

Tijdens de bijeenkomst is de werkgroep DOS65 rond de tafel gaan zitten om te kijken of de werkgroep nog wel zin heeft. We hebben afgesproken dat we een thema bijeenkomst zullen beleggen waarop we dan ook meteen de aanwezige leden willen vragen waar hun belangstelling naar uitgaat. Op die bijeenkomst zijn we uiteraard ook van de DOS65-bezitters afhankelijk. Neem je ma-

chine dan mee, maar ook je vragen en opmerkingen zijn van harte welkom. De bedoeling is omdat ergens in het midden van het land te gaan doen. De gedachte gaat uit naar Utrecht. Ik hoop jullie in de volgende μ P Kenner daarover nadere mededeling te kunnen doen. Als iemand voordien vragen of opmerkingen heeft mag hij altijd contact met mij opnemen. Mijn adres en telefoonnummer staan achter in dit blad.

Zo mijn kruit is weer verschoten. Rest mij jullie uit te nodigen op de bijeenkomst in Krommenie op 16 januari. Antoine Megens en Jaap Prenger willen ons daar iets laten zien waarover nog lange tijd zal worden nagepraat. Met andere woorden dat mag je niet missen. Om het geld hoeft je ook niet weg te blijven want de toegang is gratis.

Jullie voorkauwer

Informatie

De μ P 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 μ P 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 de diverse 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 μ P 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: 053-328506 of 053-303902.

Software Bibliotheek en Technisch Forum:

Voor het beheer van de Software Bibliotheek en technische ondersteuning streeft het bestuur ernaar zgn. systeemcoördinatoren te benoemen. Van tijd tot tijd zal in de μ P Kenner een overzicht gepubliceerd worden. Dit overzicht staat ook op het Bulletin Board.

Correspondentie adres

Alle correspondentie betreffende verenigingszaken kan gestuurd worden aan:

KIM Gebruikersclub Nederland
Postbus 1336
7500 BH Enschede

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.

Tonny Schäffer (voorzitter)

Paul Krügerstraat 27

7532 PW Enschede

Telefoon 053-613678

Jacques H.G.M. Banser (penningmeester)

Haaksbergerstraat 199

7513 EM Enschede

Telefoon 053-324137

Gert van Opbroek (secretaris)

Bateweg 60

2481 AN Woubrugge

Telefoon 01729-8636

Joost Voorhaar (redactie μ P Kenner)

Jekerstraat 67

7523 VP Enschede

Telefoon 053-333483

Jan D.J. Derksen (DOS65 coördinator)

Verfaillweg 434

1783 BP Den Helder

Telefoon 02230-28168

Geert Stappers (KGN/68k coördinator)

Engelseweg 7

5825 BT Overloon

Telefoon 04781-41279

Ereleden:

Naaft 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 Müller

Rinus Vleesch Dubois