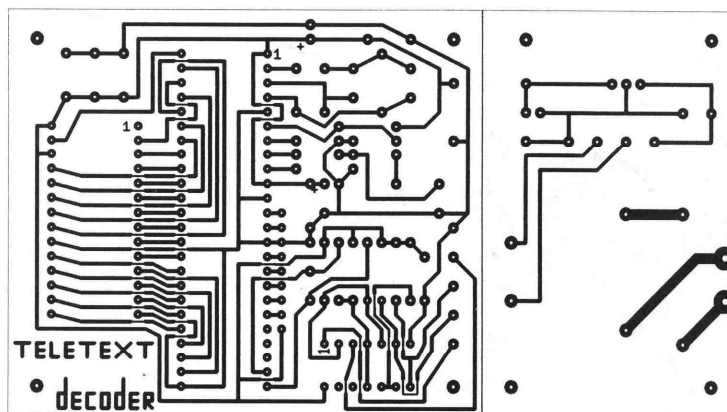




Zestiende jaargang nr. 4
Oktober 1992
78



In dit nummer o.a.:

Teletekst op uw PC
Schaarste in PC-land
Avontuur in Shareware
Datacommunicatie: Netwerken
Verantwoord computergebruik

Inhoudsopgave

De μ P Kenner

Nummer 78, oktober 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 79 verschijnt op
19 december 1992.

Kopijsluitingsdatum voor nummer 79 is
vastgesteld op 5 december 1992.

Vereniging

Uitnodiging voor de clubbijeenkomst	5
Agenda voor de algemene ledenvergadering	6
Oproep	6
De Linux-dag Utrecht	20
Voortgang KGN68k	27
Van de bestuurstafel	33

Algemeen

Redactioneel	4
Connect	19
Inleiding voor verantwoord computergebruik	28
Nieuws en schaarste in computerland	29

Software

To share Or Not To Share, That's The Question	7
Software-ontwikkeling t.b.v. KGN68k	9

Hardware

Teletext	12
----------------	----

Systemen

De Motorola 68030; het hart van KGN68k (deel 2)	21
---	----

Datacommunicatie

Voortgaande Datacommunicatie; NetWare	30
---	----

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

Zo, we hebben deze keer dacht ik wel een redelijk "lekker" blad in elkaar kunnen draaien. Gelukkig met aardig wat "externe" inbreng, zodat we je een gevarieerde inhoud kunnen bieden. Als u dit leest zit de extra bijeenkomst in Utrecht er al weer op. Ik zou graag willen schrijven dat het een perfecte bijeenkomst was met veel leden die we anders nooit zien, maar ja... die bijeenkomst ligt op het moment van dit schrijven nog in het verschiet. Wel zijn we er het in het bestuur over eens dat Utrecht een perfecte plaats is voor het organiseren van dit soort evenementen; het is centraal gelegen en goed bereikbaar met het openbaar vervoer.

Sja, en dan komen de donkere dagen rond sinterklaas, kerst en oud- en nieuw al weer langzaam in zicht. Dat betekent natuurlijk ook weer dat het tijd is voor de HCC-dagen. Gelukkig is de KGN ook dit jaar weer present, mede dankzij de gigantische bult werk die de werkgroep in de laatste maanden heeft verricht. Waar we precies staan dit jaar is nog niet bekend. Wel ongeveer wat we er gaan doen... de leden van de werkgroep zullen er namelijk met gepaste trots een prototype van de KGN68k/30 tonen! Als er geen onverwachte tegenslagen tussen komen zal het prototype ook gedemonstreerd kunnen worden. Verwacht echter geen high-speed uitvoering waarop een compleet operating system draait!

Natuurlijk bent u op de stand van harte welkom om dit nieuwste wonder van techniek te aanschouwen.

Verder hopen we er ook weer iets speciaals te kunnen laten zien; wel eens gezien hoe een moderne laserprinter nou eigenlijk precies werkt? Daar zou je best wel eens een antwoord op kunnen krijgen in de stand! En dan is staat er natuurlijk ook altijd nog een pot hete koffie op het vuur. Nou ja, bij wijze van spreken dan, want vuur maken schijnt in de beurs niet toegestaan te zijn...

Sja, en voor je het dan weet is het weer december. IJs en weder dienende hopen we vóór die tijd nog een μ P Kenner bij u thuis te kunnen laten bezorgen. Of dat lukt is een beetje afhankelijk van de hoeveelheid kopij die we binnenkrijgen; surprise me!

Volgend jaar hopen we actiever dan ooit te worden. Zo zijn er plannen in de richting van een cursus "programmeren in een UNIX environment" en er wordt naarstig gewerkt aan

andere ideeën. Daarvoor hebben we, als altijd, natuurlijk wel hard de input van uwe edele nodig ("Need more input!", wie zei dat ook al weer?). Volgend jaar bestaat de KGN namelijk 15 jaar. Tijd voor een lustrum dus (en als het dit jaar niet lukt... dan proberen we het als echte computerfreaks bij 0x10 jaar!). Op de ideeënlust staan voorlopig slechts wat vage ideeën. Laat hij, die stampvol goeie ideeën zit, opstaan!

Joost Voorhaar.

Uitnodiging voor de clubbijeenkomst

Datum: 28 November 1992
 Locatie: Wijkcentrum "De Ringvaart"
 Floris van Adrichemlaan 98
 2035 VD Haarlem
 Telefoon: 023-363856

Thema: Algemene ledenvergadering
 Entree: gratis

Programma:

9:30 Zaal open met koffie
 10:15 Opening door de voorzitter
 10:30 Algemene ledenvergadering, zie agenda
 12:00 Forum en markt
 12:15 Lunchpauze, consumpties tegen betaling
 13:15 Presentatie KGN68k/30 (onder voorbehoud)

Routebeschrijving

Auto:

Komende uit de richting Utrecht, Amersfoort of Rotterdam: Afslag Haarlem-Zuid; tweede stoplicht links; bij de tweede kruising met stoplichten links; Floris van Adrichemlaan.

Komende uit de richting Alkmaar: afslag Haarlem-Zuid; verder zie boven.

Openbaar vervoer:

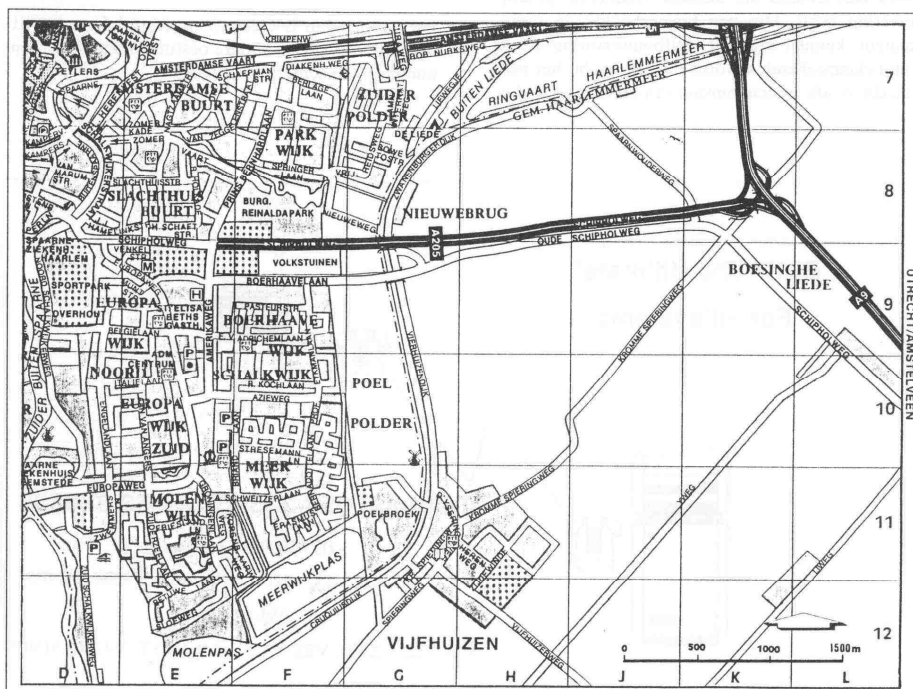
Vanaf het station Haarlem met buslijn 7, 71, 72 of 77; halte Floris van Adrichemlaan.

Aansluitend het informele gedeelte met de mogelijkheid om andermans systeem te bewonderen en niet-commerciële software uit te wisselen. U wordt uiteraard van harte uitgenodigd uw eigen systeem mee te nemen.

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.



Agenda voor de algemene ledenvergadering

Datum: 28 november 1992
Plaats: zie uitnodiging clubbijeenkomst

De volgende agenda wordt voorgesteld:

- 1: Opening ledenvergadering, vastelling agenda.
- 2: Conceptbegroting 1993.
- 3: Wijziging huishoudelijk Reglement. Voorgesteld wordt het bestaan van projectgroepen (zoals KGN68k) in het Huishoudelijk Reglement op te nemen.
- 4: Verkiezing kascontrole-commissie 1993.
- 5: Verkiezing bestuursleden; de volgende bestuursleden treden per 1 januari af en zijn herkiesbaar:
Jacques Banser (Penningmeester),

Gert van Opbroek (Secretaris) en
Geert Stappers (KGN68k)

Aftredend en niet herkiesbaar is:
Jan Dersken (DOS-65)

Met de bestaande vacature in het bestuur betekent dit dat er ruimte is voor 2 nieuwe bestuursleden. Hierbij wordt gedacht aan de bestuurstaken Public Relations en DOS-65.

Kandidaten voor een bestuursfunctie dienen uiterlijk voor aanvang van de vergadering, schriftelijk, bij de voorzitter aangemeld te worden. Deze voordracht dient vergezeld te gaan van de handtekening van tenminste twee leden.

6: Rondvraag.

7: Sluiting.

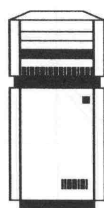
Oproep

De werkgroep KGN/68k is op zoek naar iemand die hen zo nu en dan kan bijstaan met betrekking tot hoog-frequent technieken op printplaten. Gedacht wordt aan iemand die hiermee voldoende praktijkervaring heeft. Het zou bijvoorbeeld een radio-amateur kunnen zijn die zelfbouwervaring heeft. Van belangstellenden wordt niet verwacht, het mag wel, dat ze alle bijeenkomsten van de werkgroep be-

zoeken. Het belangrijkste is dat er een beroep op hun kennis gedaan kan worden wanneer dat nodig is.

Contactpersoon is Geert Stappers telefoon: 04781-41279 of een van de andere bestuursleden. Telefoonnummers zie achterin dit blad.

BBS "The Ultimate" For all systems



Telefoon 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

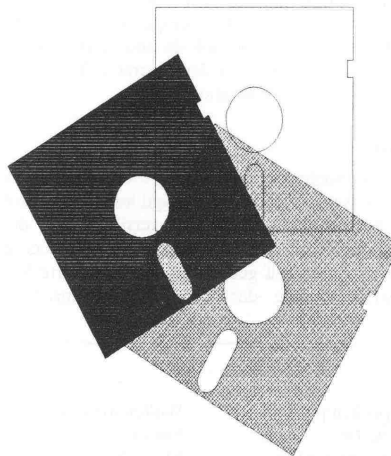
To share Or Not To Share, That's The Question

Heb jij ook zo'n hekel aan verslavende schiet- en actiespelletjes? Dan is dit absoluut niets voor je! Want in deze aflevering van "To Share..." gaan we het hebben over één van de meest verslavende spellen die ik in tijden ben tegengekomen: WolfenStein-3D.

De plot van WolfenStein is even eenvoudig als klassiek: de speler is een geallieerde gevangene in een groot Duits slot en het is zijn taak niet alleen uit het kasteel te ontsnappen, maar als het even kan ook nog de leider van het Duitse legerkamp om zeep te helpen. Daartoe begin je in de cel waarin je opgesloten zat. Je hebt zojuist de soldaat die nu voor je voeten ligt naar de andere wereld geholpen met behulp van een mes en u heeft hem snel ontdaan van zijn revolver. De deur van de cel staat open...

Tijdens je tocht door het kasteel kom je natuurlijk allerlei soldaten tegen die het op je leven gemunt hebben, soms slechts bewapend met een zelfde revolver als die waar je zelf over beschikt; sommige zijn echter veel zwaarder bewapend. In de loop van het spel kan de speler zwaarder geschut, meer munitie en allerlei schatten verzamelen. Daarnaast kom je regelmatig stukken voedsel en verbandtrommels tegen waarmee je eventueel ontstane verwondingen razendsnel weer laat helen. Het lijkt allemaal niet zo moeilijk, maar er schuilt een addertje onder het gras! Niet alleen is het parcours doorspekt met vijandelijke soldaten, honden en andere enge wezens, maar er zijn ook geheime deuren, sleutels om andere deuren mee te openen en last but not least... de uiterst zwaar bewapende monsterlijke reïncarnaties van personen als de führer himself.

Dit alles speelt zich af in een fantastisch 3-dimensionaal decor. Uiteraard stelt dit soort spellen heel wat zwaardere eisen aan de machine waarop het gespeeld wordt dan de gemiddelde editor en/of assembler... voor het spelen van WolfenStein heeft u een 286/16 of hoger nodig (een 386 met 2 MB is aan te raden) en een VGA scherm met kleurenmonitor. Een soundblaster wordt ondersteund, maar is niet noodzakelijk. Wel worden de geluiden die je hoort bij het openen van de zware metalen deuren, en het geschreeuw van soldaten die je ontdekt hebben, dan een stuk realistischer.



In de shareware versie krijg je de mogelijkheid tot het spelen van één episode van negen levels. Of eigenlijk tien, want een dergelijk spel is natuurlijk niet compleet zonder geheim level...

Als je WolfenStein registreert krijg je er nog twee of vijf episodes (afhankelijk van de prijs) bij, wat het totaal op zes episodes van ieder negen levels plus een geheim level per episode is zestig levels brengt. Dat zijn er natuurlijk nogal wat, en WolfenStein is dan ook beslist geen spel dat je even op een vrije zaterdagmiddag uitspeelt.

WolfenStein is dan ook beslist geen spel dat je even op een vrije zaterdagmiddag uitspeelt.

De documentatie lijkt op het eerste gezicht niet aanwezig te zijn. In het hoofdmenu van het spel zelf zit echter een optie "read.me", waar je vijfendertig schermen vol informatie over het spel, de bestelwijze en andere wetenswaardigheden in terug kunt vinden. Natuurlijk is er als je registreert ook een (dun) boekje beschikbaar. Voor de notoire valsspellers onder ons is er ook een hintboek en een geheim wachtwoord voor valsspellen beschikbaar. De eerlijkheid gebied echter te zeggen dat er al enkele fanatiekelingen zo slim zijn geweest om de bestandsstructuur van WolfenStein te doorgronden (die zit inderdaad niet zo verschrikkelijk ingewikkeld in elkaar...) en er zijn dan ook al verschillende bestanden met plattegronden beschikbaar. De eerste nagebouwde levels heb ik ook al gezien, en wellicht dat we binnenkort de eerste WolfenStein gamestudio tegemoet kunnen zien. Registratie is

overigens niet eens zo gek duur voor dit spel. Voor de "kleine versie" (registratie van "Escape from Wolfenstein" en de extra episodes "Eisenfaust" en "Die Führer, Die!") moet \$45,- neergeteld worden; voor 20 dollar meer krijg je ook de andere drie spellen met de illustere titels "A dark secret", "Trail of the madman" en "Confrontation".

Conclusie

Sja, wat moeten we hier nu nog toevoegen? De heren van Apogee zijn er in geslaagd hun toch al aardige spellencollectie aardig te verrijken met deze aankoop. Want deze keer hebben zij het programmeerwerk niet zelf gedaan; voor de geboorte Wolfenstein tekende de firma "ID Software". Zij

achtten dit spel geschikt voor spelers van 13 jaar en ouder. Niet geheel ten onrechte, lijkt me zo. Al is die 13 jaar wellicht nog wat aan de lage kant...

Natuurlijk is niets ooit helemaal perfect. Zo ook Wolfenstein niet. Voor de echte freak wordt het namelijk na een tijdje allemaal wel wat makkelijk. En echt veel variatie zit er niet in; de decors bijvoorbeeld zien er steeds ongeveer hetzelfde uit ondanks de verschillende structuren die gebruikt worden voor de muren en de variatie in tegenstanders. Ondanks dat, toch een leuk spel...

Joost Voorhaar

Besproken produkt	: Wolfenstein 3D, versie 1.1
Categorie	: games
Registratiekosten	: \$45 + \$5 verzendkosten (3 episodes) \$65 + \$5 verzendkosten (6 episodes)
Auteur / leverancier	: ID Software / Apogee software
Verkrijgbaarheid	: The Ultimate, games area File "Wolf3D.Zip", 709574 bytes

Minimale systeemeisen

PC/AT, 386 of 486
PC/MS-DOS
ca. 2 MByte diskruimte
640 kB RAM (met meer gaat 't beter...)
Beeldscherm: VGA color
Ondersteuning van muis, joystick, adlib, soundblaster en mickey soundsource

Algemene beoordeling

Documentatie	: redelijk, engelstalig in programmatuur opgenomen
Bediening	: prima. Via muis, keyboard, etc.
Muisondersteuning	: Yep

Positief

Modern, lekker speelbaar
Goede ondersteuning randapparatuur
Muis en actietoetsen instelbaar
Grafisch fraai uitgevoerd

Negatief

Vrij hoge systeemeisen
Ietwat centonig op de lange duur
Slechte documentatie ("But who needs more?")

Software-ontwikkeling t.b.v. KGN68k

Inleiding

Zoals al enkele keren in de μ P Kenner geschreven is, wordt er ook al hard gewerkt aan de ontwikkeling van software voor het KGN68k-project. Over enkele resultaten hiervan wil ik het in dit artikelje hebben.

In de tweede plaats is dit artikelje ook bedoeld als een presentatie van de gebruikte KGN68k assembler die vanaf nu voor leden van de KGN beschikbaar is.

Ontwikkel-omgevingen etc.

Als de eerste hardware in de vorm van een CPU-kaart en een Statisch RAM + EPROM-kaart, beschikbaar komt is er al vrij snel behoefte aan software om deze hardware uit te gaan testen. Hierbij moet niet direct gedacht worden aan complete operating systems of zelfs maar monitors, nee, één van de eerste programma's zal waarschijnlijk bestaan uit een lusje in de vorm van:

```
lus:    nop          ; Doe niets
        bra.s      lus ; Spring naar het begin
```

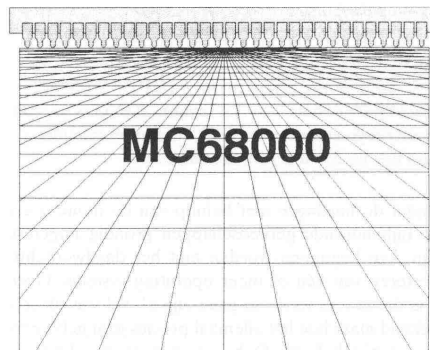
Met behulp van een dergelijk eenvoudig programma en een scoop kun je vrij snel controleren of de CPU-kaart het doet. Tenslotte wordt het geheugen aangesproken en worden er, bij het inlezen van de instructies, steeds twee opeenvolgende woorden ingelezen. Dit patroon is met een scoop zeer goed herkenbaar.

Nu zijn er uiteraard tientallen van dergelijke eenvoudige programmaatjes te bedenken die, om gedraaid te kunnen worden in EPROM gezet zullen gaan worden. Het meest eenvoudige programma kan zelfs zonder een EPROM. Als van de processor alle lijnen van de databus aan massa gelegd worden, dan wordt de volgende instructie uitgevoerd:

```
ori.b    #0,d0
```

Met behulp van een scoop moet je dan op de adresbus kunnen zien dat er steeds een hoger adres benaderd wordt omdat de program counter na elke instructie met 4 wordt opgehoogd.

Nadat je deze eenvoudige test uitgevoerd hebt en de hardware werkt op het eerste gezicht naar tevredenheid, krijg je al vrij snel de behoefte iets uitgebreidere programma's te gaan draaien. Je kunt dan bijvoorbeeld denken aan een programma dat een regel uitvoert, "Hello World, I am KGN68k and running very fast", op het standaard uitvoerapparaat schrijft. In de praktijk ben je dan al aardig op weg een zogenaamde monitor voor het systeem te pro-



grammeren waarmee je een aantal dingen met het systeem kunt doen.

Binnen de werkgroep is de laatste maanden aan een dergelijke monitor gewerkt. Hierbij zijn testversies van deze monitor in EPROMs gebakken en in een Atari ST gestoken. Deze monitor kan onder andere een tekst naar de seriële poort sturen en tekens inlezen vanaf de seriële ingang. Verder kan de monitor een programma dat via de seriële poort aangeboden wordt in het geheugen opslaan en uitvoeren. Het programma moet dan wel in de vorm van zogenaamde S-records worden aangeboden. Dit maakt het mogelijk om op eenvoudige wijze in het stadium dat de monitor loopt, allerlei testprogramma's te ontwikkelen en snel op de hardware uit te testen. Op die manier hoeft je niet voor elk programma een nieuwe EPROM te bakken en heb je ook niet direct de behoefte programma's op floppy of iets dergelijks op te slaan, nee, het programma wordt gewoon vanaf de PC of wat dan ook, in de vorm van S-records naar de KGN68k geupload.

Blijft natuurlijk de vraag hoe je aan een programma in de vorm van S-records komt. Welnu, daar zijn momenteel twee hulpmiddelen voor beschikbaar. In de eerste plaats is dat de KGN68k assembler die binnen de werkgroep ontwikkeld is. Deze assembler is momenteel in staat code voor alle 68000 processoren te genereren tot en met de 68010. Aan de uitbreidingen voor de 68020 en de 68030 wordt heel hard gewerkt. De uitvoer van de assembler bestaat uit een file met S-records die naar de KGN68k geupload kan worden.

Een tweede hulpmiddel waaraan gewerkt wordt, is een conversieprogramma waarmee de uitvoer van de GNU C compiler voor 68000 omgezet wordt naar S-

records voor het KGN68k CPU-board. Met behulp van deze strategie is het programma "Hello World" uit Kernigan en Ritchie al omgezet voor het testsysteem op de Atari.

De KGN68k assembler is geheel geschreven in ANSI C en is reeds getest onder MS/DOS (Turbo C), Macintosh (Think C) en Linux (GNU C). Het conversieprogramma en GNU C zijn momenteel alleen nog getest in een Linux omgeving.

Nadat de hardware met behulp van de monitor en de bijbehorende gereedschappen grondig uitgetest zijn, kan begonnen worden met het daadwerkelijk porteren van één of meer operating systems. Over hoe dat in zijn werk zal gaan zijn al wel wat ideeën bekend maar hoe het allemaal precies gaat gebeuren is nog niet bekend. Ook is er nog geen definitieve keuze voor een bepaalde UNIX-like gemaakt.

Samenvattend kan worden gesteld dat de ontwikkelomgeving de eerste tijd zal bestaan uit een CPU-kaart in combinatie met een kaart met daarop RAM, EPROM en seriële I/O, een ontwikkelcomputer met daarop een editor voor het aanmaken van sources, de KGN68k assembler, een GNU C compiler met conversieprogramma en een monitor in de KGN68k die in staat is een programma in de vorm van S-records in te lezen en op te starten. Als dit allemaal grondig is uitgetest, dan wordt de hardware verder compleet gemaakt (RAM-kaart met een capaciteit van 32 MB ! en disk I/O) en wordt er begonnen met de port van een operating system. Als dat ook allemaal gebeurd is, zal met de productie van bouwpakketten voor de nabouwers begonnen worden.

S-records

S-records zijn een methode om gegevens over de inhoud van geheugen op te slaan of, zoals bij KGN68k, over een seriële communicatielijn uit te wisselen. De naam "S-record" slaat op het eerste teken in een informatie-regel, zoals straks bij de bespreking van het formaat duidelijk zal worden. De volledige naam van het formaat is Motorola S formaat of ook wel Exorcisor. Het formaat is ontwikkeld voor het uitwisselen van informatie tussen een computersysteem en een EPROM programmer.

Elk S record bestaat uit een S als startteken. Vervolgens komt er een cijfer uit de reeks 0 t/m 9 die het type record aangeeft. Daarna komen er twee tekens

waarmee het totaal aantal bytes dat volgt aangegeven wordt. De daarop volgende bytes zijn afhankelijk van het type S record. Daarom eerst een overzicht van de verschillende types:

- S0: Dit is een startrecord waarin informatie over de filenaam of iets dergelijks staat. Een S0 record is niet verplicht.
- S1: Een S1 record bevat een adres met een lengte van 2 byte, gevolgd door een aantal databytes.
- S2: Een S2 record bevat een adres met een lengte van 3 byte.
- S3: Een S3 record bevat een adres met een lengte van 4 byte.
- S7: Een S7 record sluit een blok S3 records af. Eventueel bevat het record een 4 byte startadres.
- S8: Een S8 record sluit een blok S2 records af en bevat eventueel een 3 byte startadres.
- S9: Een S9 record sluit een blok S1 records af en kan een twee byte startadres bevatten.

S-records zijn een methode om gegevens over de inhoud van geheugen op te slaan .

De S4 t/m S6 records worden gebruikt om binnen een bestand over te gaan van bijvoorbeeld S1 naar S4. Deze records worden echter heel weinig gebruikt.

Na de S, het type en de lengte volgen eventueel het adres en een aantal databytes. Het record wordt altijd afgesloten met een checksum met een lengte van 1

byte. Deze checksum wordt berekend door het optellen van alle bytes uit de lengte, het adres en de databytes. Vervolgens wordt het two's complement van deze som genomen en hiervan wordt het minst significante byte als checksum gebruikt.

Opvallend in het Motorola S-formaat is dat alle gegevens afgedrukt worden in hexadecimale notatie. Een adres in het S1 formaat bevat zo vier tekens om twee byte weer te geven. Hetzelfde geldt voor het byte waarmee de lengte wordt aangegeven (2 tekens) en de checksum (ook 2 tekens). Elk databyte wordt in een S-record ook met behulp van twee hexadecimale tekens gecodeerd.

Alle tekens die ontvangen worden tussen de checksum en de volgende S worden genegeerd. Dit betekent dat men een S-records eventueel af mag sluiten met een carriage return en/of een line feed. Vaak stuurt de ontvanger na een correct ontvangen S record een punt terug zodat men het versturen van informatie eventueel kan volgen. Deze laatste zaken zijn echter niet in het protocol vastgelegd.

Binnen KGN68k worden S-records momenteel gebruikt om testprogramma's te uploaden. Uiteraard wordt de informatie geladen op de adressen die in het S-record zijn opgenomen. Nadat de gegevens zijn opgeslagen, wordt het programma gestart op het adres dat in het afsluitrecord staat.

De KGN68k assembler

Aan de KGN68k assembler wordt nog steeds hard gewerkt. Momenteel is ze echter in een toestand dat mensen van buiten de werkgroep best iets aan de assembler kunnen hebben. Voor mensen die bijvoorbeeld wat willen oefenen met het programmeren in assembler voor de 68000 kan het een handig hulpmiddel zijn. De assembler leest de source in vanuit één of meer tekstfiles en maakt twee nieuwe files aan. Uiteraard is dat de listing waarin ook de zogenaamde symbol table is opgenomen en verder een file met S-records die bijvoorbeeld naar de KGN68k geupload kunnen worden.

Voor leden is de KGN68k assembler beschikbaar op The Ultimate in de file area ULTIMATE.DEV. In het pakket dat daar staat zit de assembler in de vorm van een executable voor de PC, een stuurfile met daarin de opcodetabel en de complete beschrijving in WP 4.2 formaat. Dit pakket kan door leden gedownload worden. Zoals al is aangegeven, is de assembler ook geschikt om op andere systemen te draaien. Hiervoor is het noodzakelijk dat er op dit systeem een ANSI compatible C compiler aanwezig is. Mensen die de assembler op een ander systeem willen laten draaien of die de source om andere redenen graag zouden willen hebben moeten even contact met mij opnemen. Ik wil namelijk niet direct afstand doen van het auteursrecht en kan daarom niet toestaan dat er versies van de assembler naar buiten de KGN gaan. Dat betekent dus dat ik wel bereid ben de sources af te staan maar alleen voor studie en het porteren naar andere systemen. Verder wil ik in dat geval ook harde afspraken maken over wat wel en wat niet mag met de sources.

Zoals al is aangegeven, is de assembler nog niet af. Ook zal ze ongetwijfeld nog een aantal fouten bevatten. Als iemand op- of aanmerkingen over de assembler heeft, dan hoor ik dat graag. Bij voorkeur via The Ultimate in de vorm van een bericht aan mij (Gert Van.Opbroek). Hiervoor is op The Ultimate de Message Area ULTIMATE.DEV ingericht. Mocht je fouten ontdekt hebben, geef dan ook de source-regels waarop de assembler een fout maakt erbij. Uiteraard zullen er van tijd tot tijd nieuwe versies van de assembler beschikbaar komen. Deze zullen ook verspreid worden via The Ultimate.

Tenslotte nog een klein overzicht van de huidige en toekomstige mogelijkheden van de assembler:

De assembler heeft de volgende eigenschappen:

- Assembleert naar keuze voor 68000, 68010, 68020 of 68030
- 68040, 68881 en 68882 worden in een later stadium toegevoegd
- Ondersteunt de hele instructieset van de genoemde processoren en alle adresseermogelijkheden
- Genereert code in de vorm van Motorola S-records
- Ondersteunt conditionele assembly, include files en macro's
- Ondersteunt zogenaamde structuur macro's
- Kan optimaliseren voor minimale code

De assembler is beschikbaar voor de bron-systemen:

- MS/DOS
- Apple Macintosh
- Linux

en kan geschikt gemaakt worden voor diverse andere systemen mits voor het betreffende systeem een ANSI C compiler beschikbaar is.

Momenteel zijn van de genoemde opties een aantal zaken nog niet aanwezig. De assembler kent nog geen macro's. Verder kent de assembler nog niet alle adresseer mogelijkheden van de 68020 en hoger.

Afsluiting

Mochten er, naar aanleiding van dit artikel of andere informatie uit de KGN68k werkgroep, mensen zijn die graag mee willen werken aan de verdere ontwikkeling van software voor de KGN68k, dan kunnen die contact met mij of met Geert Stappers opnemen. Vooral mensen die ervaring hebben met het inwendige van een UNIX-achtige kunnen we heel goed gebruiken. Verder ben ik natuurlijk altijd bereid meer informatie over het project, de software-ontwikkeling of de assembler te geven. Hiervoor kun je me via The Ultimate bereiken of 's avonds na 20:30 per telefoon.

Gert Van.Opbroek

Teletekst

Door een artikel in de C'T van november 1991 kwam de gedachte bij mij op om ook een teletekst-decoder te gaan bouwen. Het schema dat zij publiceerden was de eenvoud zelf, alleen dat het een insteekkaart werd sprak mij niet aan. In de zelfde tijd had ik me ook verdiept in de I²C-bus van Philips, zodoende kwam ik op het idee een I²C-interface te maken en die aan te sturen via een printer poort, en via deze interface kon ik dus ook de teletekst-decoder aansturen. Het probleem van insteekkaart was dus verleden tijd. De interface werd een 74LS05 met een paar weerstanden. Men heeft nu ook de mogelijkheid om I²C-ontwerpen die door Elektuur zijn uitgebracht op deze interface aan te sluiten.

Tegelijkertijd was een collega van mij bezig software te ontwerpen voor de teletekstkaart van ELV die besproken is in een uitgave van Elektuur. We hebben deze software aangepast op mijn ontwerp en dat

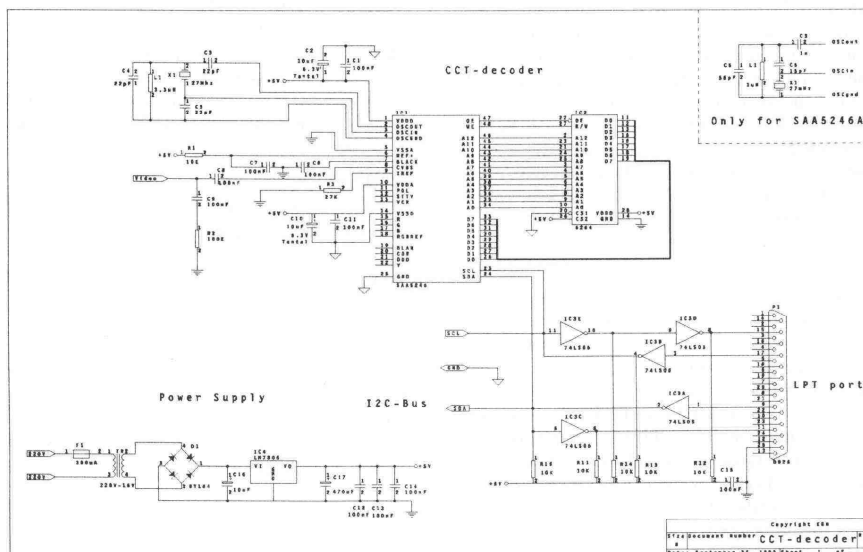
draait momenteel goed. Bij het pakket is ook een programma om printen te tekenen. Dit heb ik van een bulletin board geplukt en voor kleine printen vind ik het erg handig. Met dit programma heb ik ook de print getekend dus iedereen die de teletekst-decoder wil bouwen kan dus zijn eigen print uitprinten.

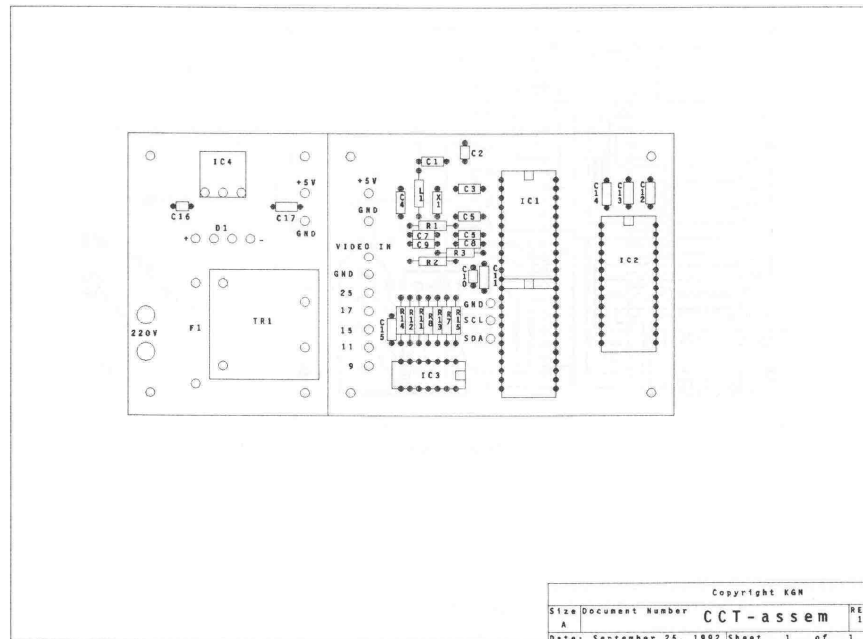
Wie meer wil weten over dit onderwerp raad ik de uitgaven aan van de Duitse C'T uitgave november 1991 en juli 1992. Kan men niet aan de informatie komen dan kan men deze eventueel via mij bekomen. De software is in C geschreven en zal in een volgend nummer volledig beschreven. Bij dit artikel tref je al wel de interface routines aan (in C). Op de komende

bijeenkomsten zal ik de decoder meenemen om te laten zien wat de mogelijkheden zijn.

Herman Hek

Het schema dat zij publiceerden was de eenvoud zelf, alleen dat het een insteekkaart werd sprak mij niet aan.

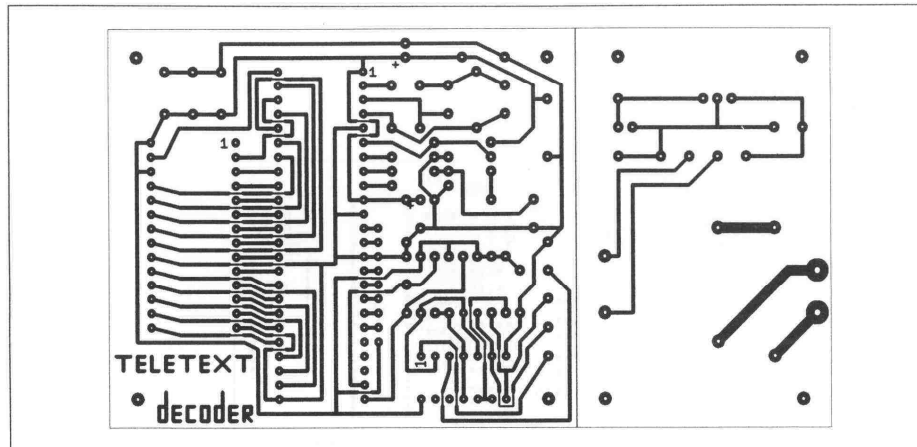




Onderdelen voor Teletextdecoder 920918-1

Aantal	Schemanummer	Omschrijving
10	C1,C6,C7,C8,C9,C11,C12,C13,C14,C15	100nF
2	C2,C10	10uF/6,3V Tantal
1	C16	10uF/16V
3	C3,C4,C5	22pF
1	C17	470uF/16V
1	D1	BY164
1	F1	300mA + Houder
1	IC1	SAA5246
1	IC2	6264
1	IC3	74LS05
1	IC4	LM7805
1	L1	3,3uH
1	R1	10E
1	R2	100E
1	R3	27K
5	R11,R12,R13,R14,R15	10K
1	TR1	220V-16V
1	X1	27MHz

Note: C9 en R2 verwijderen als VIDEO kabel korter is dan 2 meter.



Print, koperzijde

```

/*****
/*          IICMAST (ROUTINES FOR IIC MASTER)          */
/*          1992                                         */
/*          VER 1.0B                                     */
/*          NOT FOR COMERCIAL PURPOSES AUT. T.BIJKERK   */
*****/

int  timeoutc;          /* timeout counter          */
int  outbyt;            /* byte to update/send line */
int  deliic;            /* delay counter for slowerIIC*/

#define TIMEOUTC      3000    /* timeout value clock low */
#define DELIIC        20      /* 15 = 40H/80L SCL (uSec.) */
/* 20 = 50H/75L SCL (uSec.) */
/* 50 = 100H/175L SCL (uSec.) */

#define TRUE          1       /* logical true             */
#define FALSE         0       /* logical false            */
#define PORT          0x378    /* IIC printerport adres    */
#define IICCARD        FALSE   /* select IIC-card at 300H   */
#define IICPRN         TRUE    /* select prn-port for IIC   */

#ifndef SLOW
#define SLOW          TRUE     /* check if already defined */
#endif

#if (SLOW == TRUE)
#define SLOWIIC        for (deliic=0; deliic DELIIC; deliic++);
#else
#define SLOWIIC        ;
#endif

#if (IICCARD == TRUE)

```

```

#define SDAH          SLOWIIC outp (0x300, outbyt = outbyt | 1);
#define SDAL          SLOWIIC outp (0x300, outbyt = outbyt & 0xfe);
#define SCLH          SLOWIIC outp (0x300, outbyt = outbyt | 2); \
                      timeoutc = 0; \
                      while ( (inp (0x300) & 2) == 0 ) \
                      { timeoutc + +; if (timeoutc TIMEOUTC) break; }

#define SCLL          SLOWIIC outp (0x300, outbyt = outbyt & 0xfd);
#define STARTI2C      SDAL SCLL
#define STOPI2C        SDAL SCLH SDAH
#define SDAIN          SLOWIIC if (inp (0x300) & 1)
#define INITLINE       if ((inp (0x300) & 3) == 3) outbyt = outbyt | 3; \
                      else { outbyt = outbyt | 3; SCLL SDAL STOPI2C }

#endif

#if (IICPRN == TRUE)

#define SDAH          SLOWIIC outp (PORT, 0);
#define SDAL          SLOWIIC outp (PORT, 0x80);
#define SCLH          SLOWIIC outp (PORT + 2, 8); \
                      timeoutc = 0; \
                      while ((inp (PORT + 1) & 0x8) == 0) \
                      { timeoutc + +; if (timeoutc TIMEOUTC) break; }

#define SCLL          SLOWIIC outp (PORT + 2, 0);
#define STARTI2C      SDAL SCLL
#define STOPI2C        SDAL SCLH SDAH
#define SDAIN          SLOWIIC if (inp (PORT + 1) & 0x80)
#define INITLINE       if (inp (PORT + 1) & 0x88 == 0x88) { SCLL SDAL STOPI2C }

#endif

/* prototyping */

int  sendbytei2c(char);          /* send one byte */
int  sendi2c(char,int,char*);    /* send number of bytes */
int  receivebytei2c(char*,int);  /* receive one byte */
int  receivei2c(char,int,char*); /* receive number of bytes */

/*****
/*          send one byte 'databyte'
/*          no start and stop I2C, at begin and end SCL=LOW
/*          at return FALSE when acknowlage error else TRUE
*****/

int  sendbytei2c (char databyte)

{
    int  count;

    for(count=7;count = 0;count--) /* 8 bits to do */
    {
        if (((databyte > count) & 1) == 1) /* if bit = 1 high */
        {
            SDAH /* high SDA */

```

```

        }
        else
        {
            SDAL                                /* low SDA */
        }
        SCLH                                    /* pulse SCL */
        SCLL
    }

    SDAH                                        /* release SDA for acq */
    SCLH                                        /* set SCL */

    SDAIN                                        /* read acknowledge */
    {
        SCLL                                    /* reset SCL */
        return (FALSE);                        /* error no ack = 1 */
    }
    else
    {
        SCLL                                    /* reset SCL */
        return (TRUE);                         /* slave acknowledged = 0 */
    }
}

/*****
/*      send 'i2cnumber' bytes from 'i2cdata' to i2c 'adres'
/*      returns FALSE at communication error (acknolage)
/*      it starts and stops the transmission
*****/

int  sendi2c (char adres,int i2cnumber,char *i2cdata)
{
    int  count;
    char ch;

    STARTI2C                                /* start IIC transm. */

    if (!sendbytei2c (adres))                /* send IIC adres */
    {
        STOPI2C                              /* ack error, stop */
        return (FALSE);                      /* signal error */
    }

    for (count=0;count < i2cnumber;count++) /* number of bytes */
    {
        ch = *i2cdata;                       /* get char */
        if (!sendbytei2c(ch))                 /* send char */
        {
            STOPI2C                          /* ack error, stop */
            return (FALSE);                  /* signal error */
        }
        i2cdata = i2cdata + 1;                /* next byte */
    }
}

```

```

        STOPI2C                                /* stop IIC transm. */
        return (TRUE);                          /* signal ok */
    }

    /*******
    /*          receive one byte 'databyte'          */
    /*          no start and stop I2C, at begin and end SCL=LOW          */
    /*          if 'last' (byte) is TRUE it leaves the bus for stop          */
    /*******

    int  receivebytei2c (char *databyte,int last)

    {
    int  count;
    char  ch;

        SDAH                                /* release SDA */
        ch = NULL;                          /* init read buffer */
        SCLL                                /* SCL is low */
        for(count=7;count = 0;count--)      /* 8 bits to do */
        {
            SCLH                                /* set SCL */
            ch = ch < 1;                      /* shift read buffer */
            SDAH                                /* read SDA */
            {
                ch = ch + 1;                  /* if high + 1 */
            }
            SCLL                                /* reset SCL */
        }

        *databyte = ch;                      /* store read data */

        if (last)                          /* check for last */
        {
            SDAH                                /* set SDA = 1 not ack */
        }
        else
        {
            SDAL                                /* set SDA = 0 for ack */
        }

        SCLH                                /* acknowledge puls */
        SCLL
        SDAH                                /* set SDA */
        return (TRUE);                      /* signal ok */
    }

    /*******
    /*          receive 'i2cnumber' bytes to 'i2cdata' to i2c 'adres'          */
    /*          returns FALSE at communication error (acknolage)          */
    /*          it starts and stops the transmission          */
    /*******

    int  receivei2c (char adres,int i2cnumber,char *i2cdata)

    {

```

```
int last;
int count;

STARTI2C /* start IIC transm. */

if (!sendbytei2c (adres)) /* send IIC adres */
{
    STOI2C /* ack error, stop */
    return (FALSE); /* signal error */
}

SDAH /* release SDA */
last = FALSE;
for (count=0;count i2cnumber;count++) /* number of bytes */
{
    if (count + 1 == i2cnumber) /* check for last byte*/
    {
        last = TRUE;
    }
    receivebytei2c (i2cdata,last); /* receive byte */
    i2cdata = i2cdata + 1; /* increment buffer */
}

STOI2C /* stop IIC transm. */
return (TRUE); /* signal ok */
}
```

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 : _____

Connect

De afgelopen weken was het wat stil in de elektronische post. Daarom is deze aflevering van Connect ook wat beperkter dan de bedoeling was. Eigenlijk hebben we maar twee echt aardige berichten gezien; beide in het berichtengebied "assembler.028" en handelen over het vervolg op de i486 van Intel. Dit zwarte blokje dat iedereen in de wandel al "80586" gedoopt had, draagt bij Intel intern nog steeds de naam "P5". Voorheen was het ding al aangekondigd voor het laatste kwartaal van 1992, maar dat zit er niet meer in. Meer waarschijnlijk is het eerste of tweede kwartaal van '93.

Verder was er iemand die de nieuwste opcodes van de 80886 al kende. Met een hele zoutbus quote ik ze hieronder.

Wel serieus bedoeld is de naamsverandering van de P5. Er stond een heel interview in de echomail over de nieuwe naam van de P5, maar dat interview is slechts op twee punten informatief. Ten eerste zal de P5 niet 80586 gaan heten, maar krijgt de naam "Pentium". Eén en ander houdt nauw verband met de rechtszaak van 1991 omtrend de naamgeving van de 386 processoren. Een typenummer is volgens de rechtbank niet beschermd, een naam wel. Vandaar dus geen 80586, maar de Pentium.

Een tweetal andere zaken omtrend die processor werden ook bekend. Ten eerste zal Intel naast de MHz. als maat voor snelheid de iCOMPTM gaan voeren. Dit lijkt mij een interne benchmark van Intel. Wie meer informatie heeft over deze maat mag het zeggen...

Ten tweede werd nog even kort samengevat welke details allemaal al bekend zijn van de Pentium: "Performance in 100 MIPS range; Superscalar design; over 3.1 million transistors; enhanced multiprocessing capabilities; fully compatible with Intel X86 microprocessors; manufactured on BiCMOS .8 micron process; introduction in first quarter 1993".

Tenslotte

Het BBS van de KGN gaat inmiddels steeds meer lijken op een spin in een web. Een web van netwerk-aansluitingen: The Ultimate is inmiddels onderdeel van de volgende netwerken: Fidonet, PCM-net, PNN (Programmer's Network Nederland), Denknets (alles over denksporten) en SIGnet. Daarnaast zijn er dan nog een groot aantal filenetwerken. Wij houden je op de hoogte...

Joost Voorhaar

INTEL 80886 Technical Reference Manual Appendix B: Opcodes

AAO	Add And Overflow	ECO	Electrocute Computer Operator
ADB	Another Damn Bug	EM	Evacuate Memory
AFF	Add Fudge Factor	EPS	Execute Program Sideways
AGB	Add Garbage	EROM	Erase Read Only Memory
AIB	Attack Innocent Bystander	ESL	Exceed Speed of Light
BAC	Branch to Alpha Centauri	EWD	Engage Warp Drive
BAH	Branch And Hang	EZX	Enable ZX81 Mode
BB	Branch on Bug	FFF	Form Feed Forever
BD	Backspace Disk	FM	Forget Memory
BDU	Branch on Dumb User	FPT	Fire Photon Torpedoes
BNR	Branch for No Reason	GA	Guess Answer
BOA	Branch on Operator Absent	GESE	Generate Exciting Sound Effects
BPO	Branch on Power Off	GREM	Generate Random Error Message
BRT	Branch on Tuesdays	GU	Give Up
BTB	Byte The Dust	HELP	Type "No Help Available"
BTJ	Branch & Turn Japanese	IEOF	Ignore End Of File
CAI	Corrupt Accounting Information	IMPG	IMPress Girlfriend
CBA	Compare and Branch Anyway	IOP	Interrupt and Order Piza
CCS	Chinese Character Set	IP	Increment and Pray
CBN	Compare and Branch Never	IU	Ignore User
CMD	CPU Melt Down	JIL	Jump In Lake
CML	Compute Meaning of Life (42)	JRGA	Jump Relative and Get Arrested
CNB	Compute Nervous Breakdown	JU	Jump Operator
CB	Create Bug	LAP	Laugh At Programmer
CRN	Convert to Roman Numerals	LMYB	Logical MaYBe
CWAH	Create Woman and Hold	MAN	Make Animal Noises
DA	Develop Amnesia	MBTD	Mount Beatles on Tape Drive
DBZ	Divide By Zero	MST	Mount Scotch Tape
DD	Destroy Disk	NTGH	Not Tonight, I've Got a Headache
DHTPL	Disk Head Tree Point Landing	OML	Obey Murphy's Law
DLN	Don't Look Now	OTL	Out To Lunch
DOC	Drive Operator Crazy	PIBM	Pretend to be an IBM
DU	Dump User	PPA	Print Paper Aeroplanes
DUD	Do Until Dead	QWA	Quit While Ahead

De Linux-dag Utrecht

In een bijsluiters van de vorige uP-Kenner hebben jullie vernomen dat we op 24 oktober weer vroeg uit de veren moesten om richting Utrecht te vertrekken. De reden was deze keer een extra bijeenkomst speciaal voor Linux. Deze bijeenkomst was niet alleen voor leden maar we wilden proberen ook mensen van buiten de vereniging aan te trekken. De reden zal duidelijk zijn, vers bloed binnen de club vergaren. In dit geval mensen die zich op de een of andere wijze met Linux of een andere Unix bezig houden.

We hadden een drietal ruimtes tot onze beschikking waarin er een was om de inwendige mens te versterken en voor de liefhebbers een rookertje op te steken.

Een ruimte was ingericht met computers waarop Linux stond te draaien. Op een van die machines kon men zien hoe of X-windows werkte. Verder was er in die ruimte gelegenheid om enige hardware aan te schaffen welke was geleverd door de firma Multi Discount uit Enschede. Er was een standje met boeken over Unix en de programmeertaal 'C' die vriendelijk ter beschikking waren gesteld door boekhandel Planeta eveneens uit Enschede. Beide firma's wil ik vanaf deze plaats onze hartelijk dank toezeggen voor hun bereidwillige medewerking.

In een derde ruimte was de mogelijkheid om een lezing te geven. Hugo van der Kooy gaf een inleiding over de mogelijkheid om met Linux een netwerk op te zetten en de aansluiting op bestaande netwerken. Er werd een heldere uiteenzetting gegeven over de alom bekende ethernetkaarten en het TCP/IP protocol. Voor de meeste mensen geen vreemde kretten, maar hoe werkt het? Ook in deze inleiding kon daar niet echt diep op ingegaan worden. Toch was de inleiding duidelijk genoeg om alvast een goede indruk

te krijgen. Bovendien beloofde de inleider dat hij in de toekomst nog uitgebreid op dit onderwerp zal terug komen. Hugo, ook jij onze hartelijke dank.

Vanaf 10.00 uur tot circa 14.00 uur was het een gezellige drukte die zich tussen de drie lokalen afwisselde. Uiteraard ging de meeste belangstelling uit naar de ruimte waar de apparatuur stond. Met uitzondering van het uurtje dat door Hugo in beslag werd genomen.

Was deze bijeenkomst geslaagd? Deze vraag kan met ja en met nee worden beantwoord. De opkomst had wat ons betreft best groter mogen zijn. Maar daarbij steken we ook de hand in eigen boezem. De voorbereidingstijd was eigenlijk wat aan de korte kant. Daardoor zijn er misschien wat mogelijkheden blijven liggen om oproepen, vooral voor de niet leden, wat uitgebreider te plaatsen. Toch kan het ook niet helemaal een flop worden genoemd. De helft van de bezoekers was nog niet lid van de KGN. We verwachten dat er zich in de nabije toekomst zeker een aantal zullen aanmelden.

We hadden meteen de mogelijkheid om het eens in een centrale plaats in het land te doen. Om ook eens te kijken of dat meer leden zou trekken. Dat is ons eigenlijk wel tegengevallen. Alhoewel ook de leden laat op de hoogte zijn gebracht van deze bijeenkomst. Maar goed besturen gaat nou eenmaal niet altijd over rozen. We zijn er van overtuigd dat zo'n dag, mits beter voorbereid, een volgende keer beter zal slagen.

Tonny Schäffer

Deze bijeenkomst was niet alleen voor leden maar we wilden proberen ook mensen van buiten de vereniging aan te trekken.

De Motorola 68030; het hart van KGN68k (deel 2)

Inleiding

In de tweede aflevering van deze serie wil ik een begin maken met de manier waarop je een 68030 programmeert. Mogelijk heb je al elders in deze μ P Kenner gelezen dat de KGN68k assembler beschikbaar is zodat het moment waarop we beginnen met het programmeren van de 68000-familie mooi samenvalt met het beschikbaar komen van de benodigde hulpmiddelen.

Na de vogelvlucht door de processor uit de vorige aflevering, wil ik nu met name de adresseermogelijkheden wat nauwkeuriger gaan behandelen.

Programmeermodel

In deze aflevering gaan we ons vrijwel uitsluitend bezighouden met het programmeren van de processor in User State. Er zullen dus nauwelijks geprivilegieerde instructies voorbij komen. Om een goed houvast te hebben, is in figuur 1 nog een keer het programmeermodel in User State getekend.

De belangrijkste kenmerken van dit programmeermodel zijn:

- 8 dataregisters; D0 t/m D7.
- 8 adresregisters; A0 t/m A7. Hiervan is A7 de zogenaamde stack pointer die ook wel met SP aangeduid wordt. Het betreft in dit geval uiteraard de User Stack Pointer uit het Supervisor Programmeermodel.
- Een program counter die met PC aangeduid wordt.

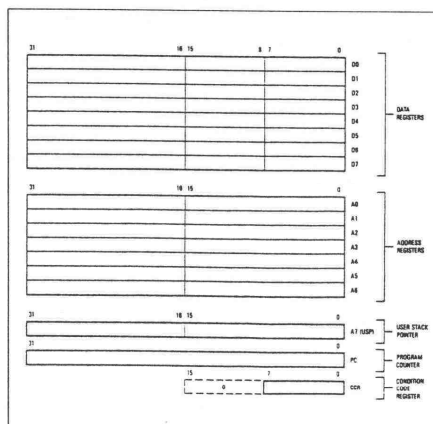
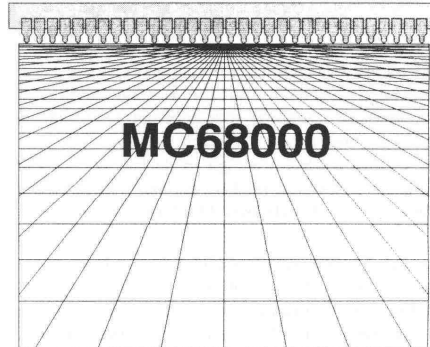


Fig. 1: User Programmeermodel



- Het laagste byte uit het Status Register; ook wel Condition Code Register genoemd. Dit register bevat de Carry, de overflow vlag, de Zero vlag, de Negative vlag en de eXtend vlag.

Verder is het nog van belang te weten dat de adres- en dataregisters een breedte hebben van 32 bits. Bij de dataregisters kan men van deze 32 bits de laagste 8 benaderen (byte benadering), de laagste 16 (word benadering) of alle 32 tegelijk (long word). Benadert men een byte of een word, dan worden de resterende bits in het register niet gewijzigd. Bij de 68020 en hoger kan men ook een zogenaamd bitfield benaderen. Hiervan geeft men de afstand tot het meest significante bit in het register op en de hoeveelheid bits die men wil benaderen. Het meest significante byte (bits 24 t/m 31) geeft men dan aan d.m.v. {0:8}.

Bij een adresregister kan men het laagste word apart benaderen. Dit betreft dan een 16 bits adres. Benadert men het hele register, dan heb je een 32 bits adres. Nu heeft een adresregister ook nog een eigenaardigheid. In tegenstelling tot een dataregister worden bij een adresregister de overige bits wel beïnvloed bij benadering van het minst significante word. Het hoogste bit van het minst significante word wordt gekopieerd naar alle bits van het meest significante word (sign extension). Dit wordt verduidelijkt in het volgende voorbeeld:

```
CLR.L D0           ;Maak register schoon
MOVE.L D0,A0
MOVE.W #$F0F0,D0   ;D0 bevat nu $0000F0F0
MOVE.W #$F0F0,A0   ;A0 bevat nu $FFFFFFF0F0
CMP.L A0,D0         ;Deze registers zijn niet gelijk!
BNE.S Label_1      ;Sprong wordt uitgevoerd.
```

Zo zie je maar, dat je adres- en dataregisters bij een 68030 niet klakkeloos door elkaar kunt halen!

Behalve de genoemde formaten, kent een dataregister ook nog enkele formaten die daar van afgeleid zijn. In de eerste plaats is dit het zogenaamde Quad Word waarbij de informatie in twee dataregisters tezamen een 64 bits breed dataveld vormen. Verder kent men nog twee formaten waarbij men in een half byte alleen de getallen 0 t/m 9 afbeeldt. In dit Binary Coded Decimal formaat worden alleen de bitcombinaties 0000 t/m 1001 gebruikt. De 68000-familie kent hierbij twee varianten; een unpacked variant waarbij slechts één digit in de laagste vier bits aanwezig is en een packed variant waarin in het minst significante byte twee BCD digits zitten.

Uiteraard wordt data niet alleen in een register opgeslagen. Ook in het geheugen kan men gegevens opslaan. Dit gaat in principe op dezelfde manier als in een dataregister. Was het bij de 68000 en 68010 nog noodzakelijk dat een word en een long word altijd op een even adres beginnen, bij de 68030 is dit niet meer het geval. Vanzelfsprekend is het voor de snelheid van het programma het beste als een word op een even en een long word op een viertallig adres begint zodat de processor alle informatie in één keer over zijn 32 bits brede databus kan inlezen. Een instructie (ook een word) moet echter wel altijd op een even adres staan.

Een leuke uitbreiding vinden we tenslotte nog bij de bitfields. Men kan, door de offset in een dataregister te plaatsen, een offset opgeven die loopt van -2 tot de macht 31 tot (2 tot de macht 31) - 1. Dit betekent dat men vanaf het aangegeven adres niet alleen vooruit maar ook achteruit kan en dat men veel verder van het adres weg kan komen dan de vier bytes volgend op dit adres.

Binnen de 68000-familie wordt bij de benadering van een operand gesproken van een effectief adres. Dit is de plaats waar de informatie gevonden kan worden of waar het resultaat moet worden opgeslagen. Effectieve adressen kunnen een register aangeven of een geheugenplaats. Hoe de operand in het effectieve adres precies benadert wordt, staat in de volgende paragraaf die gaat over de diverse adresseermogelijkheden.

Adresseermogelijkheden

De 68000-familie kent een groot aantal adresseermogelijkheden die in de rest van deze paragraaf stuk voor stuk behandeld zullen worden.

1: Data Register Direct Mode

Bij Data Register Direct staat de operand in één van de 8 dataregisters. Hier geeft men dus als operand D0 t/m D7 op.

Assemblersyntax: Dn

2: Address Register Direct Mode

In deze mode geeft men d.m.v. A0 t/m A7 aan welk adresregister bedoeld wordt. In de KGN68k assembler (en veel andere assemblers) mag men in plaats van A7 ook de afkorting SP van Stack Pointer gebruiken.

Assemblersyntax: An

3: Address Register Indirect Mode

Bij de Address Register Direct mode wordt de inhoud van een adresregister bedoeld. Bij de indirecte mode staat in het adresregister het adres van de operand in het geheugen. Deze mode wordt verduidelijkt door figuur 2.

Assemblersyntax: (An)

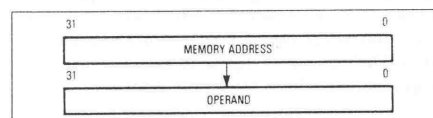


Fig. 2: Address Register Indirect Mode

4: Address Register Indirect with Postincrement Mode

Deze mode lijkt sterk op de Address Register Indirect Mode. Ook in dit geval staat in een adresregister het adres van de plaats waar de operand gezocht moet worden. Het bijzondere van deze mode is echter dat nadat de operand benaderd is, de inhoud van het adres verhoogd wordt met de lengte van de operand. Op deze manier kun je in een lus dus steeds een hoger adres benaderen. Deze methode komt heel goed van pas bij stacks. In het algemeen lopen die op een 68000 van hoog adres naar laag adres. Als je nu een register definieert als stack pointer, dan kun je met Address Register Indirect with Postincrement op eenvoudige wijze iets van deze stack afhalen. Omdat voor de system stack A7 als stack pointer gebruikt wordt, kun je op deze manier dus ook iets van de system stack afhalen:

```
MOVE.B (A7)+,D0 ; Haal een byte van de stack
MOVE.B (A7)+,D1 ; En nog een voor alignment
MOVE.W (A7)+,D2 ; En nu een word
MOVEA.L (A7)+,A0 ; En nu een compleet adres
```

Merk op dat de plus achter het register staat, dit wil zeggen dat eerst de operand opgehaald wordt en daarna de inhoud van het register wordt verhoogd.

Assemblersyntax: (An) +

5: Address Register Indirect with Predecrement Mode

Wordt bij de post increment mode eerst de operand benaderd en vervolgens de inhoud van het register aangepast, bij de predecrement mode wordt eerst de inhoud van het adresregister verlaagd met de lengte van de operand waarna de operand in het geheugen wordt benaderd. Deze mode komt heel goed van pas om iets op de stack te zetten:

```
MOVE.L A0,-(A7) ;Zet een compleet adres op stack
MOVE.W D2,-(A7) ;En een word
MOVE.B D1,-(A7) ;En nu een byte
MOVE.B D0,-(A7) ;En nog één voor de alignment
```

Zoals te verwachten was, staat nu de - voor het register.

Assemblersyntax: `-(An)`

6: Address Register Indirect with Displacement Mode

Zijn de voorgaande modes zeer geschikt om iets van een stack te halen of op de stack te zetten, deze mode is uitermate geschikt om een operand te benaderen die ten opzichte van een adres een vaste afstand heeft. In deze mode wordt namelijk bij de inhoud van een adresregister een vaste constante opgeteld. Deze constante mag een lengte hebben van 16 bits en ligt in de range -32768 t/m $+32767$. Er zijn diverse plaatsen waar deze mode heel handig is. Neem als voorbeeld een I/O chip. Bij de 68000 zijn deze gewoon ergens in het geheugen terug te vinden. Als een dergelijke chip geprogrammeerd wordt, dan zet men het basisadres van de chip vaak in een register en benadert de interne registers, die op een vaste afstand van het basisadres liggen, m.b.v. een displacement. Als voorbeeld een stukje code uit een testprogramma t.b.v. KGN68k:

* Psg

```
Psg      equ $FFFF8800 Parallel I/O & Sound
PsgWr    equ 2          Write register
PsgMfr   equ 7          Multi-Functions Register
PsgPortB equ 15         Centronics Data Lines
PsgPortA equ 14         Floppy, RS232 & Centr.
PsgM      equ %11000000
```

* Init PSG, programmable sound generator

```
lea      Psg,A2
move.b   #PsgMfr,(A2) ;Port A & B are
move.b   #PsgM,PsgWr(A2) ;init. for Output
move.b   #PsgPortA,(A2) ;Select Port A
move.b   #7,PsgWr(A2) ;Deselect Floppy D.Drives
move.b   #PsgPortB,(A2) ;Select Port B
```

Assemblersyntax: `d(An)` of `(d,An)`
waarbij d een 16 bits constante is.

Hierbij is de laatste notatie degene die volgens Motorola (68030 User's Manual) gebruikt moet worden.

Voor een nog beter begrip van deze mode kun je verder nog figuur 3 bestuderen.

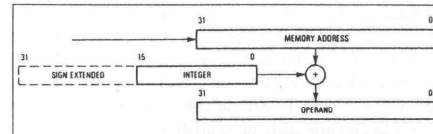


Fig. 3: Address Register Indirect with Displacement Mode

7: Address Register Indirect with Index (8-Bit displacement) Mode

Bij deze mode begint het echt ingewikkeld te worden. Bovendien is dit de eerste mode waarbij er verschillen naar voren komen tussen de 68020 processoren en hoger en de 68000, 68008 en 68010. De Indirect with Index mode is weer een uitbreiding op de Indirect with Displacement. In dit geval is echter de displacement maximaal 8 bits breed zodat de range nu loopt van -128 tot $+127$. Behalve een displacement komt er in deze mode ook nog een register en een schaal-waarde voor. In het register, een adres- of een dataregister, staat een zogenaamde index. Afhankelijk van de opdracht aan de assembler wordt de hele inhoud van het register genomen of alleen het minst significante woord. In dat laatste geval wordt er een zogenaamde sign extension toegepast waarbij het meest significante bit gekopieerd wordt naar het meest significante woord.

Een mogelijkheid die de 68020 en hoger wel hebben en de lagere processoren niet, is de zogenaamde schaal variabele. Bij deze processoren kan men, voordat het adres berekend wordt, de index laten vermenigvuldigen met 1, 2, 4 of 8. Dit maakt het benaderen van het n-de long word in een tabel nog iets gemakkelijker. De operand wordt nu gevonden door de inhoud van het adresregister te nemen, hierbij wordt de displacement opgeteld en de eventueel geschaalde inhoud van het index register.

Blijft natuurlijk de vraag waarvoor dit handig is. Welnu, in hogere programmeertalen komen nogal eens complexe datastructuren voor. Neem bijvoorbeeld in Pascal een array van integers. Je kunt nu het derde byte van de tweede integer benaderen door in het adresregister het startpunt van de array op te nemen, de index (2) in het indexregister, de schaal-waarde op 4 te zetten en voor de displacement het getal 3 in te vullen. Bij een 68000 heb je geen schaal-waarde zodat je daar zelf de inhoud van het indexre-

gister met de lengte van de datastructuur (in dit geval 4 voor een integer) moet vermenigvuldigen.

Assemblersyntax: $d(An, Xn.size*scale)$ of
 $(d, An, Xn.size*scale)$

Hierbij is d een 8 bits constante, Xn geeft een adres- of dataregister aan, $size$ is een 'W' voor 16 bits een 'L' voor de volledige 32 bits van het indexregister en $scale$ bevat de waarde 1, 2, 4 of 8. Bij de KGN68k-assembler kan men 'size' en '*scale' weglaten. In dat geval wordt voor de $size$ de volle breedte van 32 bits gebruikt en voor $scale$ de waarde 1.

Aangezien deze mode en de volgende mode alleen in de lengte van de displacement verschillen, is het rekenschema zoals dat in figuur 4 getekend is ook op deze mode van toepassing.

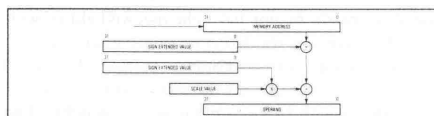


Fig. 4: Address Register Indirect with Displacement Mode

8: Address Register Indirect with Index (Base displacement) Mode

Deze mode komt alleen bij de 68020 en hoger voor. De mode is vrijwel identiek aan de voorgaande met dit verschil dat de displacement nu in plaats van 8 bits ook een lengte van 16 of 32 bits mag hebben. Om deze displacement te onderscheiden van de displacement uit de vorige mode, wordt ze ook wel 'Base Displacement', afgekort met 'bd' genoemd. De mode is getekend in figuur 4.

In het algemeen kan de assembler zelf uitzoeken of hij moet werken met een 8 bits, 16 bits of 32 bits displacement. Kan hij dit niet, dan kan men dit ook aan de assembler opgeven door in de displacement d.m.v. een 'b', een 'w' of een 'l' een lengte op te geven. Je krijgt dan:

Assemblersyntax: $bd.size(An, Xn.size*scale)$ of
 $(bd.size, An, Xn.size*scale)$

Als de lengte niet is opgegeven en de assembler kan de lengte ook niet zelf bepalen, dan is de default uiteraard een lengte van 8 bits zodat je in de voorgaande mode terecht komt. In de Base Displacement mode is het toegestaan één of meer van de gegevens 'adresregister', 'indexregister' en displacement weg te laten. Hiermee kun je dus op eenvoudige wijze bijvoorbeeld een dataregister indirect mode creëren

door het indexregister en de displacement weg te laten.

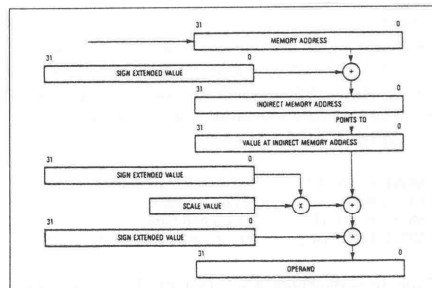


Fig. 5: Memory Indirect Postindexed Mode

9: Memory Indirect Postindexed Mode

Deze mode is niet aanwezig bij de 68000 t/m de 68010. De mode bestaat uit een adresregister, een base displacement, een indexregister met $size$ en $scale$, en een 'Outer Displacement'. De berekening gaat als volgt (zie ook figuur 5):

- Neem de inhoud van het adresregister en tel hierbij de base displacement op.
- Neem de inhoud van het gevonden geheugenadres en tel hierbij de C inhoud van het indexregister en de outer displacement op.
- Het resultaat van a) en b) levert het adres op waar de operand gevonden kan worden.

Evenals bij de vorige modes, heeft deze mode ook een aantal defaults. Wordt de $scale$ weggelaten, dan wordt schaalfactor 1 gebruikt. De default $size$ voor het indexregister is uiteraard weer 32 bits. De beide displacements mogen onafhankelijk van elkaar een breedte van 16 of 32 bits hebben. Verder mag in deze mode mag zo'n beetje alles weggelaten worden; de beide displacements, het adresregister en het indexregister.

Assemblersyntax: $([bd.size, An], Xn, size*scale, od.size)$

10: Memory Indirect Preindexed Mode

Ook deze mode is niet aanwezig bij de 68000 t/m de 68010. De berekening gaat in dit geval als volgt (zie ook figuur 6):

- Neem de inhoud van het adresregister en tel hierbij de C inhoud van het indexregister en de base displacement op.

- b) Neem de inhoud van het gevonden geheugenadres en tel hierbij de waarde van de outer displacement op.
- c) Het resultaat van a) en b) levert het adres op waar de operand gevonden kan worden.

Je ziet, dat in dit geval de inhoud van het indexregister voor de eerste indirectie meegenomen wordt. 6502 programmeurs zullen dit misschien wel herkennen. De Postindexed mode lijkt veel op de Indexed Y mode van de 6502 en de Preindexed mode op de Indexed X mode. (Zou Motorola bij de ontwikkeling van de 68020 in de gaten gehad hebben dat doorgewinterde 6502 programmeurs eigenlijk niet zonder deze modes kunnen)?

Ook in dit geval zijn de beide displacements, het adresregister en het indexregister optioneel.

Assemblersyntax: ([bd.size,An,Xn.size*scale],od.size)

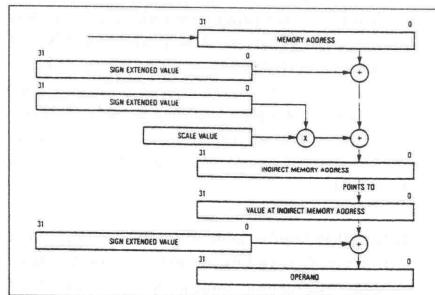


Fig. 6: Memory Indirect Pre-Indexed Mode

11: Program Counter Indirect with Displacement Mode

Door in de Address Register Indirect with Displacement mode het adresregister te vervangen door de program counter, wordt deze mode verkregen. Met behulp van deze mode kan men geheugenlocaties benaderen die op een vaste afstand van de program counter liggen. Hiermee is het mogelijk zogenaamde positie-onafhankelijke code te genereren. Dit wil zeggen dat het programma ook zal draaien als het op een ander adres in het geheugen gezet wordt dan waarvoor het geassembleerd is. Vooral bij Multi Tasking Operating Systems (UNIX, OS9) is dit van groot belang. De gebruikte assembler moet in dat geval de displacement tot het gewenste adres uitrekenen.

Neem de volgende twee voorbeelden:

```
ORG      $1000
```

```
Counter  ds.w      1
;
Start    move.w     Counter,D0; Absolute address
         addi.w     #123,D0
         move.w     D0,Counter; Absolute address
```

Dit voorbeeld kan alleen draaien op adres \$1000.

```
Counter  ORG      $1000
         ds.w      1
;
Start    move.w     Counter(PC),D0; Relative address
         addi.w     #123,D0
         move.w     D0,Counter(PC); Relative address
```

Omdat in dit tweede voorbeeld de geheugenplaats Counter relatief t.o.v. de program counter benaderd wordt, kan dit stukje programma op elk willekeurig adres draaien. Dit betekent dat men de code kan genereren voor adres \$1000 en vervolgens kan draaien op adres \$1F000. Vooral in een multitasking omgeving heeft dit grote voordelen omdat men meestal bij het assembleren niet weet waar de code in het geheugen geplaatst zal gaan worden.

Assemblersyntax: d(PC) of
(d,PC)

12: Program Counter Indirect with Index (8-Bit displacement) Mode en

13: Program Counter Indirect with Index (Base displacement) Mode

Deze modes zijn afgeleid van Address Register Indirect With Index door het adresregister te vervangen door de program counter.

Assemblersyntax: d(PC,Xn.size*scale) of
(d,PC,Xn.size*scale) resp.
bd.size(PC,Xn.size*scale) of
(bd.size,PC,Xn.size*scale)

Evenals bij Address Register Indirect with Index (Base Displacement Mode) mag men ook bij de Program Counter etc. Mode de displacement, het indexregister en de program counter weglaten. Om toch een onderscheid te hebben tussen de modes voor adresregister en program counter, moet men in dat geval de code ZPC (Zero PC) aangeven. Dit heeft dan als resultaat dat men geheugen in de Program Space benadert i.p.v. in de Data Space. Zie hierbij ook de beschrijving van de function codes in deel 1 van deze serie.

14: Program Counter Memory Indirect Postindexed Mode

Deze mode is uiteraard afgeleid van de Memory Indirect Postindexed Mode door het adresregister te vervangen door de program counter.

Assemblersyntax: ([bd.size,PC],Xn.
size*scale,od.size)

Aangezien in deze mode ook weer het nodige mag worden weggelaten, is de beschrijving van ZPC uit de vorige mode ook op deze en de volgende mode van toepassing.

15: Program Counter Memory Indirect Preindexed Mode

Deze mode is uiteraard afgeleid van de Memory Indirect Preindexed Mode door het adresregister te vervangen door de program counter.

Assemblersyntax: ([bd.size,PC,Xn.size*scale],od.size)

16: Absolute Short Addressing Mode

Bij deze mode wordt in de instructie het absolute adres van de operand opgegeven. In dit (short) geval wordt het adres als een 16 bits constante aangegeven waarbij een sign extension plaatsvindt. Met behulp van deze mode kunnen zo de eerste 32768 geheugenplaatsen en de hoogste 32768 geheugenplaatsen in de adresruimte benaderd worden.

Assemblersyntax: xxxx.W of
(xxxx).W

Opmerking: De KGN68k assembler is, onder bepaalde voorwaarden, in staat automatisch te herkennen dat een adres in de 16 bits range ligt. De aanduiding .W kan dan worden weggelaten.

17: Absolute Long Addressing Mode

Hier bevat de instructie het volledige 32 bits adres van de operand.

Assemblersyntax: xxxxxxxx.L of
(xxxxxxx).L

18: Immediate Data

In deze mode staat de operand zelf in de instructie. Het maakt hierbij voor de lengte van de instructie uit of de constante in 1, 2 of 4 bytes gecodeerd moet worden. Mocht de assembler niet zelf in staat zijn de optimale grootte te bepalen, dan kan men dit eventueel d.m.v. een .B, .W of .L achter de constante afdwingen.

Assemblersyntax: #xxxxxxx.S

De bovenstaande 18 adresseermodes zijn de zogenaamde normale modes. Deze modes komen bij een groot aantal instructies voor. Verder kent de 68030 nog een groot aantal speciale modes die alleen bij enkele instructies voorkomen. Enkele hiervan zal ik nog behandelen.

19: Condition Code Register

20: Status Register

In deze modes worden het condition code register (het laagste byte van het status register) en het status register benaderd. In User State kan men bij de 68030 alleen het Condition Code Register benaderen. Alle instructies om het volledige Status Register te benaderen zijn namelijk geprivilegieerd.

Assemblersyntax: CCR resp. SR

21: Program Counter Relative

Deze mode komt voor bij de zogenaamde Branch instructies. Hierbij gaat de processor niet verder met de eerstvolgende instructie maar springt naar een andere plaats in het programma. De nieuwe waarde van de program counter wordt in dat geval niet absoluut opgegeven, alleen het verschil tussen de oude en nieuwe waarde wordt in de instructie aangegeven (spring 14 plaatsen vooruit). Dit heeft weer alles te maken met het position independent programmeren. Zou men in een sprong een absoluut adres opgeven, dan kun je het programma niet ergens anders in het geheugen plaatsen. Zou je dit namelijk wel doen, dan springt de processor bij de eerste de beste sprong naar een adres waar opeens geen programma meer staat. Uiteraard kan de assembler zelf uitrekenen hoe groot de offset is; de programmeur geeft d.m.v. een label aan waar naartoe gesprongen moet worden.

De hele 68000 familie kent relatieve sprongen met een offset van 8 en 16 bits, de 68020 en hoger daarboven nog zogenaamde long jumps met een offset van 32 bits. In een aantal gevallen kan de assembler zelf bepalen hoe groot de sprong zal worden. Wil men afdwingen dat een sprong als 8, 16 of 32 bits gecodeerd wordt, dan kan men dit d.m.v. een .S, .W resp. .L in de instructie aangeven.

Assemblersyntax: Bcc.sexpressie

21: Bitfield

Van een bitfield wordt de offset t.o.v. het hoogste bit in het register of de geadresseerde geheugenlocatie aangegeven en de breedte van het bitfield. Hierbij mag de breedte (width) variëren van 1 tot en met 32 bits en de offset van -2 tot de macht 31 tot (2 tot de macht 31) - 1.

Assemblersyntax: <ea> {offset:width}

Waarbij met <ea> het effectieve adres bedoeld wordt.

22: Register List

De 68000-familie kent een instructie om in één keer een hele lijst registers op de stack te plaatsen of van

de stack te halen. Hierbij kan men de registers opsommen waarbij ze gescheiden worden door een '/' of men geeft een range van registers op door het eerste en laatste register te noemen, gescheiden door een '-'. Verder mag men de twee methoden combineren.

Voorbeelden:

```
MOVEM.L      D0-D7/A0-A6,-(A7)
MOVEM.L      (A7)+,D0-D3/D5-D7/A0-A4
```

Afsluiting

Wel, dat was volgens mij een zwaar stukje. Mensen die de voorgaande paragraaf beheersen kunnen volgens mij goed aan de slag met het programmeren in assembler voor een 68000. Uiteraard heb je dan nog wel een document nodig waarin alle instructies met

hun bijbehorende adresseer modes staan. Dit zou bijvoorbeeld de in de literatuurlijst genoemde Programmers Reference Manual kunnen zijn of een goed boek over programmeren in assembler voor de 68000. Zelf zal ik in de handleiding bij de KGN68k ook een overzicht opnemen.

Literatuur

- 1: Motorola: MC68030 Enhanced 32-bit micro-processor user's manual; Prentice Hall 1990.
- 2: Motorola: M68000 family programmer's reference manual.

Deze documentatie is beschikbaar gesteld aan de KGN68k werkgroep door EBV Elektronik te Maarssenbroek.

Voortgang KGN68k

Er zijn al weer een aantal weken in het nieuwe computerseizoen voorbij. De KGN68k werkgroep heeft ze niet zo maar voorbij laten gaan.

Printed Circuit Board

Eurocard heeft inderdaad voordelen. Voor de print met ROM/DRAM en serial & parallel I/O is de layout af en er wordt al een print van gemaakt. Op de ledenvergadering van november 1992 kun je hem zien. Van de processorprint bestaat ook een layout, maar die heb ik nog niet naar de printfabrikant opgestuurd. Om verschillende redenen: we willen er nog controles op uitvoeren en als voornaamste reden: de print zal te lang onderweg zijn, om hem ook nog te kunnen testen voor de ledenvergadering van november. Als The Ultimate test van het schema wordt nu een prototype van de processorkaart met draadjes opgebouwd. Of daar alle 32 datalijnen naar buiten gebracht worden is nog niet bekend. Waarschijnlijk, maar het is niet nodig voor de eerste testen. Voor "hello, world" heb je aan 8 bits genoeg.

Software

Er zit nu een eenvoudige monitor in een werkende 68000 machine (Atari ST). De code is helemaal ge-

nereerd door de KGN68k assembler. Omdat load, go, memory dump en memory modify wel erg weinig is wordt de monitor uitgebreid met o.a. een disassembler. De vorige keer werd er gesproken over een Gnu 68k compiler waar aan gesleuteld werd. Nu, dat gaat naar wens. Een C-programma wordt gecompileerd en dan gelinkt. Vervolgens wordt die executeble omgezet in een file met S37-records. Met load wordt het dan de target binnengehaald. Zo zijn we dan dicht bij het operating system. De Gnu compiler draait op een LINUX system. Een OS dat door Gnu ondersteunt wordt en de kant van Gnu willen wij ook op, het schijnt de mercedes onder de compilers te schijn. Maar ja, een 68k versie schijnt er niet van te zijn. De werkgroep heeft als taak te zorgen voor een OS waar voldoende software voor is. 68k MINIX porteren naar een MMU of LINUX naar een 68k, that is the question.

Suggesties zijn nog steeds welkom.

Geert Stappers 04781-41279

Inleiding voor verantwoord computergebruik

Omdat tegenwoordig bijna iedereen met computers moet werken, hierbij een eerste inleiding in verantwoord computergebruik. Een moderne computer bestaat uit een beeldscherm, een toetsenbord en een zgn. systeem-kast met wat draadjes en een opening.

Beeldscherm

Het beeldscherm, door de fanatici onder de computergebruikers (herkenbaar aan de vierkante oogjes achter een ronde bril) ook wel "monitor" genoemd is eigenlijk niet meer dan een afgekeurde kleuren-televisie van de bekende fabrikant uit het zuiden des lands. Omdat niemand meer zwart-wit televisies koopt (Je wilt je toch groen en geel ergeren aan de weers-verwachting, de vakantie-man, EK voetballen, etc.) heeft Philips deze afgekeurde kleurentelevisies een passend prijsje gegeven in de verwachting, dat er altijd wel een paar sukkels zijn, die rotzooi kopen, als je er maar het woord "computer" voor plaatst en de prijs verdubbelt.

Om het niet te goed te laten opvallen, dat het eigenlijk om afgekeurde kleuren-televisies gaat, hebben ze alle andere kleuren dan groen of geel (amber in computer commerciële termen) uitgeschakeld. En om te voorkomen, dat mensen naar herhalings van Goede Tijden, Slechte Tijden, Prijzenslag of The Bold & The Beautiful gaan kijken, terwijl ze moeten "werken" wordt er geen afstandsbediening meegeleverd. Om het echt goedkoop te houden hebben ze ook de rest van de knopjes maar in de montage-hal laten liggen. Het enige dat ze niet vergaten was een Aan/Uit-knop, waarvan de functie dan ook door een lampje wordt "verlicht". Voor een weldenkend lezertje zal wel duidelijk zijn, dat de functie van een dergelijk lampje op zijn minst discutabel is. Om het helemaal een "kas"-succes te laten zijn worden - in tegenstelling tot de meest gewaagde kleuren en houtnerven voor de gewone televisie - deze beeldschermen alleen maar in saai grijs geleverd. Zo schep je een markt, als opeens weer houten kasten, mooi gelakt of disco-kleuren in de mode komen. Om de goegemeente helemaal op een dwaalspoor te zetten, hebben ze ook de luidspreker afgekoppeld. Helaas zelden helemaal goed, want na een tijdje aanstaan, gaat de "monitor" vanzelf "piepen", wat eigenlijk alleen maar het bewijs is, dat er wel een luidspreker in zit.

Toetsenbord

Dan is er dat zogenaamde "toetsenbord", dat is een soort bord, met daarop allerlei toetsjes met daarop de diverse lettertjes en cijfertjes. Computer-spelletjes baseren zich op het regelmatig opzoeken van een van de lettertjes of cijfertjes. Om het wat moeilijker te maken hebben ze ook pijltjes en met andere opschriften bedrukte toetsjes toegevoegd. Waarom zo'n ding altijd bij een computer geleverd wordt, zal wel te maken hebben met frustraties van "kinderen, die altijd maar overal aan zaten"!

Alleen al de volgorde van de toetsen slaat nergens op, ongetwijfeld is daar ook heel lang over nagedacht door een staats-commissie, die het uiteindelijk maar overliet aan een overspannen professor, die zijn kleindochtertje inplaats van LEGO maar toetsen om te spelen gaf. Daardoor hebben ze nu deze onbegrijpelijke volgorde. Gelukkig zitten de toetsen een beetje los, zodat je - na een beetje wrikken met een passende schroevendraaier - de volgorde gewoon kunt aanpassen. Hebben ze u ooit uitgelegd, waarom er pijltjes-, HOME, PgDn, PgUp en DELETE toetsen opzitten? Nee, dan heeft U ongetwijfeld niet de nieuwste computerspelletjes gespeeld! De "vering" van die toetsen is trouwens voor het gemak van die zgn. "computer"-hackers (mensen, die met twee vingers

dagen en nachten in opperste extase op toetsjes drukken). Als ze in slaap vallen over hun toetsenbord, zorgen de verende toetsen voor een comfort, waarvoor een donzen kussen zich niet hoeft te schamen!

Waarom er in deze tijd van allerlei draadloze dingen een draadje aan zo'n toetsenbord moet zitten is niet duidelijk. Ook zonder draadjes kun je gewoon op de toetsen drukken! Dus neem een schaar en knip het draadje af. Voor een gebruiker heeft dit dan nog een bijkomend voordeel; je kunt lastige bazen, collega's makkelijker van je af houden! Verder is een toetsenbord uiteraard het bewijs voor het feit, dat de gebruiker kan lezen, schrijven en toetsen. Daarnaast maakt het een bureaublad extra vol, zonder dat je daarvoor aansprakelijk gesteld kan worden.

Om het niet te goed te laten opvallen, dat het eigenlijk om afgekeurde kleuren-televisies gaat, hebben ze alle andere kleuren dan groen of geel uitgeschakeld.

Systeemkast

Dan is er die systeem-kast, dat met allerlei kabeltjes overal aan vast zit. Het nut daarvan is niemand duidelijk. En een hels kabaal maakt het ook. Zou een ventilator inzitten zeggen een paar, er zou een harde schijf in zitten, zeggen weer anderen. Waarvoor die ventilator dient, is natuurlijk geheim, want als je zo'n ding openschroeft, kun je nergens iets zien, waarvoor zo'n ventilator zin heeft anders dan om stof te verzamelen, je vingers ertussen te krijgen of om een hoop herrie te maken.

Wat zo'n harde schijf voorstelt is me evenmin duidelijk. Is het een mislukte Big Mac? Is het een oude discus van Ria Stalman? Is het een diskette, die ze ergens aan hebben vast gelijmd? In ieder geval, het maakt een hoop herrie. Het beste is dan ook, zodra het geluid je teveel irriteert, er een paar goede klappen tegen te geven, dan houdt die herrie vanzelf een keer op. Als de kast op de goede hoogte staat kun je er natuurlijk ook tegen aan schoppen.

Dan zit er vaak een opening aan de voorkant van die kast: voor diskettes zeggen ze, nou dat zal best, maar ik heb er nooit meer dan anderhalve "diskette" in weten te proppen! Waarom hebben ze daar niet direct een soort Jukebox van gemaakt? Wat meer lampjes, meer knopjes, een geinig geluidje erbij en Elvis is Back in Town!

Diskettes

En dan de diskettes. Dat zijn kleine plastic schijfjes, met een metalen schuifje. Vroeger, toen ik nog jong

was en er eigenlijk alleen nog maar ingewijden met computers bezig waren en diskettes alleen maar bij streng geselecteerde winkels voor honderden gulden per stel te koop waren, waren diskettes of "floppies" zoals ze toendertijd door ingewijden genoemd werden nog erg groot. Er zat aan weerskanten een gleuf in, waar je volgens diezelfde ingewijden, nooit met je vingers aan mocht komen. Waarom vraag ik me nog steeds af, Je mocht er ook al niet op schrijven.

Maar goed. Tegenwoordig zijn ze in plastic verpakt en hebben een metalen schuifje. Als je nu bij een beetje "computer-freak" langs komt, heeft die altijd een hele verzameling van die "diskettes" naast zich liggen. Het lijken wel verzamelaars... Waarom snap ik niet, je ziet toch geen verschil tussen de ene en de andere diskette? Of zit het verschil 'm in die stickers, die er zo vaak bijgeleverd worden? Trouwens, waarom heten die dingen opeens geen "floppies" meer? Viel het teveel op, dat automatisering eigenlijk altijd een "flop" is?

Een volgende keer meer: over computer-programma's.

Met vriendelijke groeten, uw MacBart gaat nu een Appletje eten!

Bart Honhoff

Nieuws en schaarste in computerland

Moederborden met duizelingwekkende snelheden stellen op dit moment niets meer voor. Een 80386DX op 40 MHz is de gewoonste zaak van de wereld, en ook een 66 MHz 80486DX met een scheepslading cachegeheugen doet niemand meer verbleken. In het vorige nummer was te lezen, dat Intel er nog wel eens een handje van had om CPU-chips kunstmatig schaars te houden. Dit had tot gevolg, dat de concurrentie versneld in het gat in de markt stapten, met als resultaat dat vooral de 80386DX veel goedkoper werd.

Hogere snelheden

Op dit moment zijn er twee nieuwe processors aan het doorbreken: de AMD 386SX-33 (was maximaal 25 MHz) en de Intel 80486SX-33. Dat laatste heb ik alleen nog maar bij gerucht vernomen. De 386SX-33 blijft bescheiden geprijsd: een moederbord zonder cache met deze CPU komt op ongeveer f 350,-.

Schaarste

Al die goedkope krachtige processoren kunnen alleen hun optimale rekenkracht etaleren als ze door middel van cachegeheugen met het hoofdgeheugen worden gekoppeld. Gevolg: een wereldwijde schaarste aan cachechips, vooral 8kx8. Hierdoor worden een aantal moederborden inmiddels aangeboden met 128k cache, omdat hierin de wat beter verkrijgbare 32kx8 chips zitten.

Tweede schaarste: DRAM (voor de zoveelste keer). De prijzen van 1 Mbyte SIMMs lijken de laatste dagen omhoog te vliegen: er zijn al verdubbelingen geconstateerd. Goed getimed voor de HCC-dagen...

Nico de Vries

Voortgaande Datacommunicatie; NetWare

Dit is het eerste artikel uit een serie die in wil haken op Gert van Opbroek's serie over datacommunicatie. Met deze serie wil ik een aantal zaken uit Gert's serie uitdiepen en ook wat niet genoemde zaken voor het voetlicht brengen.

C.V.tje

Voor dat ik met de serie begin is het wellicht zinvol om mezelf een klein beetje voor te stellen. Ik heb als opleiding MTS elektronica genoten met een persoonlijke voorkeur voor de digitale techniek. Via m'n stage plaatsen en m'n hobby ben ik bij mijn huidige werkgever (Racal-Datacom) terecht gekomen. Zoals de naam al doet vermoeden leggen wij ons toe op datacommunicatie oplossingen. Binnen het bedrijf ben ik nu dik drie jaar bezig met local area netwerken en alles wat er bij onze klanten aan vast zit. Een deel van mijn kennis hierover wil ik in de komende tijd via dit blad spuien.

Ik ben me er van bewust dat ik soms dingen als bekend veronderstel die niet iedereen of wellicht zelfs bijna geen van de lezers weet. Ook zal ik misschien enige fouten maken in één of meer artikelen of naar iemands mening iets of veel te beknopt, of veel te uitgebreid behandelen. Vrees niet, onderaan dit artikel vindt je op welke wijze ik zoal benaderd kan worden voor correcties, aanvullingen of andere opmerkingen over de inhoud of de stijl van mijn verhaal.

Inleiding

Dit eerste deel zal aansluiten op een door bijna niemand bezochte lezing van Jan Derksen in Almere. Ik wil het namelijk hebben over het meeste verspreide netwerk operating system ter wereld: Novell's NetWare.

NetWare is op dit moment eigenlijk in twee varianten verkrijgbaar: NetWare 286 en NetWare 386. Over beide versies zal ik nog wel wat in één van de volgende paragrafen zeggen. Eerst maar iets algemeen over NetWare en de filosofie er achter.

NetWare is een netwerk operating system. Een leuke term die ook goed de taak omschrijft. NetWare is een besturingssysteem (voortaan toch maar O.S. te noemen ;-)) dat het hart vormt van het hele netwerk. NetWare installeer je op een machine die voor jou als fileservers (F.S.) gaat optreden. Hiermee ga je da-

ta (programmatuur & gegevens) op een centrale plaats onderbrengen. De verwerkingskracht blijft nog steeds afhankelijk van je eigen computer maar de opslag van gegevens vindt gecentraliseerd plaats.

Dit klinkt leuk maar nogal theoretisch. Laten we dus even kijken hoe zoiets er uit zou kunnen zien in de praktijk.

Praktijkvoorbeeld

Ik stel een architectenbureau voor met vijf CAD PC's, twee tekstverwerkings PC's en één PC voor de financiële stukken. Dit betekent acht PC's met ook acht harde schijven. Het nadeel is dat bepaalde tekeningen of maar op één van de vijf PC's staat of gekopieerd over alle vijf. Het gevolg hiervan is dat je of iemand achter een PC vandaan moet halen om bij je tekening te kunnen, of je moet bakken geld uitgeven voor vijf grote harde schijven en elke dag alles synchroniseren.

Iedereen voelt natuurlijk op z'n klompen aan dat er iets moet gebeuren om alles wat soepeler te laten lopen. Om de PC's en software niet weg te hoeven doen (Nog lang niet afgeschreven volgens de boekhouder!) gaan we ze koppelen via een netwerk.

We schaffen dus wat netwerk kaarten aan voor de diverse PC's en een negende computer wordt geplaatst (ook met netwerk aansluiting) die dienst zal gaan doen als F.S. (letterlijk: Bestands Dienaar; soort VN vredesmacht). Op deze machine installeren we NetWare. Op de PC's wordt wat workstation (W.S.) software bij geplaatst om de F.S. te kunnen benaderen (via je netwerk kaart).

Deze workstation software zorgt voor een hoop zaken maar ik zal de belangrijkste even noemen. Zo zorgt de W.S. software dat de door de F.S. aangeboden schijf capaciteit naar DOS toe wordt vertaald in extra schijven. Een deel van de F.S. disk is voor jou dus gewoon disk F: naast C: van je eigen harde schijf bijvoorbeeld. Tot zover dit voorbeeld; ik zal nu verder gaan met hoe de zaken intern in de F.S. geregeld wordt.

Interne opbouw NetWare

Ik zal nu de interne opbouw van NetWare twee maal gaan uitleggen. Te weten eerst voor NetWare 286 en vervolgens nogmaals voor NetWare 386. De verschil-

**Nog lang niet
afgeschreven
volgens de
boekhouder!**

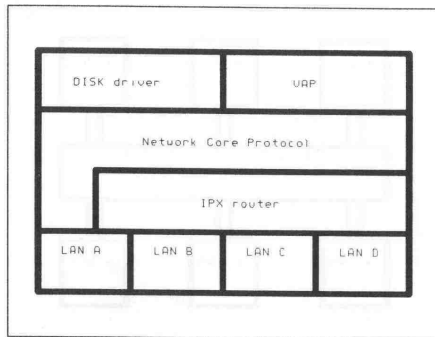


Fig. 1: interne opbouw Netware 286

len tussen deze twee zijn toch zo groot dat ik beide apart wil behandelen.

NetWare 286 is, zoals de naam al aangeeft, gebouwd voor een, op de Intel 80286 gebaseerd, hardware platform. In de praktijk zijn dat achtennegentig procent van alle IBM-AT machines en alle mogelijke klonen die daarvan gemaakt zijn. NetWare 286 draait evenals OS/2 v1.x in de 286 protected mode.

NetWare 286

Figuur 1 geeft een vereenvoudigd overzicht van de interne opbouw van NetWare 286. Onderaan vinden we een viertal LAN drivers namelijk LAN A t/m LAN D. Dit zijn de netwerkkaarten in de fileserver. Nu is het niet zo dat er ook vier kaarten in moeten zitten maar vier is wel het maximum. Als een LAN niet gebruikt wordt, dan wordt bij installatie een NULL driver meegelinkt. De drivers sturen rechtstreeks de hardware aan en bieden een standaard interface aan NetWare intern. Deze drivers moeten door de leverancier van de kaart worden gemaakt volgens specificaties van Novell.

NetWare 286 kent in principe maar één netwerk protocol en dat is IPX/SPX. Vandaar ook de IPX router binnen de fileserver. Over het waarom van IPX routing wil ik het straks nog hebben. Boven de IPX router vinden we het "Network Core Protocol" (NCP voor vrienden). Dit is eigenlijk de kern waar het om draait. Bijna alle zaken, die er in een verbinding tussen fileservers onderling en tussen fileserver en werkstation gebeuren, kunnen worden samengevat als verzoeken om deze of gene file te lezen of dergelijk acties. Hier zien we opeens de software op het werkstation weer verschijnen. Deze vertaalt namelijk allerlei verzoeken van het werkstation naar de fileserver in zogenaamde NCP requests. Wat echter ook opvalt is dat het NCP alleen maar is verbonden met LAN A. Dit betekent dat alles van LAN B t/m LAN D via LAN A moet gaan. Dit wordt echter al-

leen maar intern verwerkt en verschijnt dus niet op het netwerk dat aan LAN A hangt. De keuze voor deze constructie is historisch gegroeid maar heeft wel een paar vervelende gevolgen. Als LAN A namelijk problemen krijgt kan via LAN B t/m D ook niet meer gewerkt worden. In de praktijk betekent dit een afweging extra. Het beste kan men voor het LAN het type netwerk nemen dat de minste storingen ondervindt. Dit probleem is geheel opgelost met NetWare 386 maar dat komt nog. De NCP gaat vervolgens de verzoeken van de werkstations verwerken en gaat dan bijvoorbeeld een diskdriver aanspreken. Deze diskdrivers werken net zoals de LAN drivers. Dat wil zeggen, ze sturen zelf de hardware aan en bieden een standaard interface aan NetWare.

Nu kan een fileserver meer dan alleen maar bestanden bewaren. Zo kennen we nog iets wat core printing wordt genoemd. Dit betekend dat de lokale poorten van de fileserver (parallel en serieel) gebruikt kunnen worden om lokaal te printen. Ook deze aanvragen gaan via de NCP. Hier komt echter nog wel een heel verhaal met queing (wachtrijen) bij kijken maar daar wil ik nu niet dieper op in gaan. In beperkte mate kan een fileserver meer aanvullende diensten bieden. Deze worden bij het opstarten geladen als VAP (Value Added Process; waarde toevoegend process). Ook deze VAP's communiceren via het NCP.

Dit is een heel summiere beschrijving van hoe NetWare 286 intern in elkaar steekt. Ik had dit misschien wel uitgebreider kunnen doen maar het is niet de bedoeling dat ik een boek ga maken van de uP Kenner. Laten we dus maar verder gaan met NetWare 386.

NetWare 386

NetWare 386 is een volledig opnieuw geschreven versie die alleen maar draait op 80386 en 80486 processoren. De interne opbouw is heel anders en dat is goed te zien in figuur 2.

De kern van NetWare 386 bestaat alleen maar uit een kaal operating system met daarin het NetWare filesystem. Alle taken die een fileserver moet uitvoeren moeten echter apart geladen worden. Deze taken worden NLM's (NetWare Loadable Modules) genoemd. Dit zijn gewoon bestanden met een vaste extensie. Zo kennen we 4 groepen van bestandsnamen voor deze modules volgens onderstaande tabel.

*.LAN Driver voor een netwerk kaart. Deze kan meerdere malen worden geladen als dat nodig is. (Bijvoorbeeld drie 3COM kaarten voor drie stukken EtherNET.

- *.DSK Driver voor de harddisk (controller). Ook deze kan zonodig meerdere malen worden geladen.
- *.NAM Driver voor ander filesystemen dan DOS. Een vertaling van DOS bestandsnamen naar Novell zit al in het basis operating systeem. Andere filesystemen (Op dit moment NFS, MACintosh en OS/2's HPFS) moeten apart geladen worden.
- *.NLM Alle andere modules. Dit zijn bijvoorbeeld modules voor het printen, het ondersteunen van andere netwerk protocollen als IPX enzovoorts.

Standaard moet dus minimaal één DSK driver en één LAN driver worden geladen wil NetWare 386 lijken op wat NetWare 286 kon. Dit lijkt misschien omslachtig maar het biedt een haast ongekend aantal combinatie mogelijkheden. Zo kan je onmiddellijk een netwerk deactiveren zonder de server opnieuw te linken. Of ondersteuning voor de macintosh installeren zonder dat één van de bestaande DOS gebruikers dat merkt.

De over Netware 286 gemelde problemen met de interne router komen in principe onder NetWare 386 niet meer voor. De interne IPX/SPX router (ook deel van het basis operating system) werkt niet meer via LAN A naar de NCP maar via een Intern IPX netwerk. Dit is dus alleen een logisch netwerk en is niet verbonden aan enige hardware. Problemen met één netwerk hebben normaal geen enkele invloed op de ander netwerken.

Ook over NetWare 386 is een boek te schrijven maar dat is niet de bedoeling van dit artikel. Ik wil slechts een kleine aanzet geven. Misschien kan hier op een interactieve manier mee verder gegaan worden op een club bijeenkomst (lieft wel één die wat drukker wordt bezocht als die in Almere).

Tenslotte wil ik nog heel even ingaan op IPX/SPX omdat dit protocol zo nauw verbonden is met NetWare.

IPX/SPX

IPX is een afkorting van Internet Packet eXchange en is oorspronkelijk uitgedacht door Xerox. Het protocol is in principe hardwareonafhankelijk maar heeft vaak wel een heel directe koppeling met de hardware. Een IPX pakket bestaat uit een stuk adressering en de user data. De opbouw op bit niveau laat ik achterwege maar de adressering zal ik nog wel doornemen.

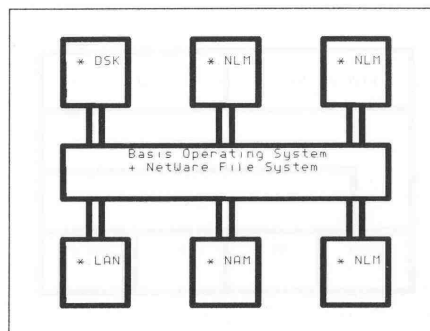


Fig. 2: interne opbouw Netware 386

IPX adressen bestaan uit 10 bytes. Namelijk 4 bytes voor het netwerk adres en 6 bytes voor het node adres. Deze keuze komt duidelijk uit het Xerox lab omdat dat toen ook met EtherNET bezig was. EtherNET kent aan elke node een 6 bytes groot nummer toe dat wereldwijd uniek is. Door het node nummer van IPX even groot te houden als het EtherNET adres is het heel simpel om het hardware adres als node adres gebruiken en de keuze voor 4 netwerk bytes zal zelfs voor hele grote netwerken voorlopig nog voldoende blijken. Routen met IPX pakketten is dus simpelweg even kijken naar het netwerk adres.

SPX is een protocol boven op IPX. IPX kent geen zaken zoals volgordebepaling en foutafhandeling. Dit zijn zaken die de beide betrokken nodes maar moeten uitvechten. IPX is gewoon een snel en vies (Quick 'n Dirty) protocol dat erg weinig overhead heeft. Om toch betrouwbaar te kunnen werken bestaat er ook SPX (Sequenced Packet eXchange). Hierin zit wel een foutafhandeling en ook de volgorde is nu bepaald. Deze protocollen worden meestal in één adem genoemd vandaar mijn aanduiding IPX/SPX.

Slotopmerkingen

Wie, naar aanleiding van dit artikel, contact met mij wil opnemen kan dat op de volgende manieren doen:

- 1 Via NETMAIL: 2:512/32.23
Hugo van.der.Kooij
- 2 Via "The Ultimate".
- 3 Via de telefoon: 01899-28194
(ma t/m vr, 19:30-22:00)
- 4 Via de post: Hugo van der Kooij
Rust-Hoff
Broekpolderweg 3
3155 EP Maasland

Van de bestuurstafel

Na de relatieve stilte van de vakanties breekt nu de drukte weer aan. Het begon met de Linux-dag in Utrecht en straks moeten we naar de HCC-dagen. Het is ons gelukt om daar een gratis stand te bemachtigen. Gelukkig maar. We hadden niet geweten waar we het toch onaanzienlijke bedrag vandaan weg moesten halen.

We werden gered door de bel. Met behulp van de KGN68k werkgroep die ons kon beloven een eerste blik in de machine te kunnen werpen. Dat was voor de HCC interessant genoeg om ons een gratis stekkie te gunnen.

Deze besparing maakte het ons mogelijk om de Linux-dag te organiseren en deze gratis toegankelijk te maken. Dat gold ook voor niet leden. We hopen hierdoor de volgende keer te kunnen melden dat dat een toestroom van nieuwe leden heeft opgeleverd. Als dat een succes is lijkt zo'n dag voor herhaling vatbaar. Mogelijk met een ander onderwerp en misschien samen met een zustervereniging.

Het zijn zomaar wat ideeën die binnen het bestuur spelen. Andere ideeën zijn een cursus/workshop in de taal 'C', kleine projectjes die op de clubbijeenkomsten kunnen worden gehouden en dergelijke. We willen daarmee een grotere betrokkenheid van de leden bereiken bij onze vereniging. Betrokkenheid die jullie overigens ook zelf kunnen tonen door het bestuur eens op de hoogte te brengen van ideeën die jullie zelf hebben. Ik wil hier nu niet zeggen dat we niets afwijzen. Maar het zal zeker onze aandacht krijgen en als het mogelijkheden heeft ook tot uitvoering brengen.

LAAT EENS WAT VAN JE HOREN

De telefoonnummers en adressen van het bestuur staan nog steeds op de laatste pagina van dit blad. Je

kan je dan ook nooit excuseren met: die bestuurders zijn nooit te vinden. Bovendien zijn we ook aanspreekbaar op de clubbijeenkomsten.

Nu ik het toch over clubbijeenkomsten heb, we waren 19 september in Almelo. Geert Stappers hield daar een voordracht over de softwareontwikkeling voor de KGN/68k. Hij ging daarbij wat dieper in op hetgeen hij in de vorige μ P Kenner reeds had beschreven. Het voordeel van zo'n voordracht is dat je bij onduidelijkheden meteen vragen kunt stellen. De geschreven en gesproken uitleg samen, maken dat alles nog begrijpelijker wordt. Nadat we de inwendige mens versterkt hadden

gingen we in de middag over tot het informele gedeelte. Nou, wat heet informeel. Adri Hankel had zijn stereo D/A omzetter voor Modplay waarover hij in het april nummer van ons lijfblad had geschreven bij zich. Hij liet een demonstratie horen waarbij de liefhebbers begonnen te watertanden. Voor de twijfelaars onder jullie kan ik zeggen dat het zeer de moeite waard is om Adri's geesteskind na te bouwen. Als er mensen zijn die denken dat niet te kunnen maar het wel willen is het

misschien mogelijk om dat eens op een clubbijeenkomst te doen. Laat het ons even weten dan gaan we zeker actie ondernemen.

Zo dat was het dan voor deze keer. Rest me nog om jullie er op te wijzen dat de HCC-dagen op 20 en 21 november zijn. Dat een week later op 28 november de clubbijeenkomst in Haarlem zal zijn. Voor de leden die een keer hun ogen uit willen kijken zou ik willen zeggen noteer maar vast de datum voor Krommenie op 16 januari a.s.

Tonny Schäffer

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 microcomputers 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 dediverse systemen in stand gehouden.

Landelijke bijeenkomsten:

Deze worden gehouden op bij voorkeur de derde zaterdag van de maanden januari, maart, mei, september en november. De exacte plaats en datum worden steeds in de 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:

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

Erevoorzitter:
Siep de Vries

Ereleden:
Mevr. H. de Vries-van der Winden
Anton Müller
Rinus Vleesch Dubois