

een volwassen computer
voor beginners

JUNIOR COMPUTER

boek 3

uitgeversmij.
elektuur b.v.

Junior-computer 3

een
volwassen
computer
voor
beginners

A. Nachtmann
G.H. Nachbar

Uitgave:

Uitgeversmaatschappij Elekfuur B.V.

Postbus 75 6190 AB Beek (L)

De printen uit dit boek zijn opgenomen in de EPS (Elektuur-printservice). Voor nadere informatie wordt verwezen naar de laatstverschenen uitgave van Elektuur. Daar treft men ook nadere informatie aan over de Elektuur-Software-service.

© UITGEVERSM AATSCHAPPIJ ELEKTUUR B.V. 1981
Overname van de inhoud van dit boek of een deel daarvan zonder schriftelijke toestemming van de uitgeefster is verboden.

AUTEURSRECHT

De auteursrechtelijke bescherming van de inhoud strekt zich mede uit tot de illustraties met inbegrip van de printed circuits en tot de ontwerpen daarvoor.

In verband met Artikel 30 Rijksoktrooiwet mogen de opgenomen schakelingen slechts voor partikuliere of wetenschappelijke doeleinden worden vervaardigd en niet in of voor een bedrijf.

Het toepassen van schakelingen geschiedt buiten verantwoordelijkheid van de uitgeefster.

NADRUKECHT:

Voor Duitsland: Elektor Verlag GmbH, 5133 Gangelt 1.

Voor Groot Britannië: Elektor Publishers Ltd. Canterbury.

Voor Frankrijk: Elektor sarl LeDoulieu, 59940 Estaires.

Printed in the Netherlands.

ISBN 90 70160 18 8

We gaan uitbreiden!

Er had óók boven kunnen staan: "van junior-junior-computer naar senior-junior-computer". Van een volwassen computer voor beginners naar een nóg volwassener computer voor beginners.

Wat bedoelen we daarmee?

Kijk, het gaat hierom: voor de één betekent de standaard-junior-computer van de boeken 1 en 2 het einde, voor de ander het begin. "Het einde" in de zin van: "zoiets zocht ik nou nèt om kennis te maken met de mikro-computer en er mee te leren omgaan". En "het begin" in de zin van: "aardig, maar we willen méér". Het staat vast dat velen van de houden-zo-groep gaandeweg zijn bekeerd tot de wij-willen-meer-groep. En — sprekend van "bekeerd" — het risico lopen om te ontaarden in de "godsdienst-waanzin" van: "het is nooit genoeg"; men is pas verzadigd als na een denkbeeldig deel 37 de junior-computer is uitgroeid tot een soort IBM-computer.

Moraal: Uitbreiden? Prima! Maar dan beheerste groei en geen wildgroei. Dus overzichtelijke, optimaal gekozen uitbreidingen aan hardware en huishoudelijke software. Dus een goede keuze van nieuwe soldeerpunten en nieuwe bytes.

En wát dan wel?

Punt één. Het telkens weer opnieuw intoetsen van een programma is te vergelijken met een boek dat men, alvorens het te lezen, eerst zelf moet schrijven, op basis van aantekeningen. Het maakt daarbij niet uit of het boek door u of door anderen is bedacht. En een boek hoeft toch eigenlijk maar één keer te worden geschreven, nietwaar? Voor computerprogramma's en andere data is de cassette-band heel goed als boek te gebruiken. Het eenmalige schrijven en het willekeurig vaak lezen van het boek zijn mogelijk met de hardware van de cassette-interface en de software van het programma Tape Management. Werkelijk een heel bijzonder boek, die cassette-band, want je kunt het als het moet nog herschrijven ook!

Punt twee. Het mag dan bij mensen zo zijn dat de wens om "meer hersens" te bezitten het gevolg is van zelfonderschatting — bij computers ligt dat anders. Geheugenuitbreiding is nodig als men grotere programma's moet kunnen maken en als men meer huishoudelijke software in huis wil hebben (in EPROM) of halen (van de cassette-band in RAM).

Punt drie. Graag méér dan de 23 mogelijkheden tot onze aktie via het standaard-toetsenbord en graag ruimere mogelijkheden voor een reactie van de computer dan via zes displays. Maak hem dus geschikt voor de aansluiting van een randapparatuur. Kies er één met een groot toetsenbord, die een aantal of alle regels "geschiedenis" van de opgave of uitvoering van een programma op het tv-scherm laat zien of op papier afdrukt. Hierdoor is het mogelijk om een programma onder supervisie van de Printer Monitor direkt of via de cassette-band op te geven en uit te voeren. En dat met een aanzienlijk groter bedieningscomfort dan met de standaardmonitor. Maar je kunt het ook anders bekijken. Om bijvoorbeeld "in BASIC" met de junior-computer te kunnen omgaan is die uitbreiding met extra drukwerk nodig en moet er een "junior-journaal" zijn, op de tv of op papier.

Punt vier. De bestaande PIA wordt gebruikt voor oude en nieuwe systeem-

programma's. Aangezien één van de belangrijke toepassingen van een mikrocomputer de direkte kommunikatie met de buitenwereld is (applikaties via I/O), is het geen gek idee om een I/O-bouwsteen toe te voegen die de gebruiker voor de volle 100% ten dienste staat, dus zonder huishoudelijke verplichtingen.

Er zijn drie hoofdstukken. In hoofdstuk 10 gaan we het hebben over de theorie en de praktijk van de uitgebreide junior-computer. Dan, in hoofdstuk 11, volgt het cassette-gebeuren, inclusief het systeemprogramma Tape Management, dat werkt op basis van het standaard-toetsenbord en van de zes displays. Afgezien van de cassette-recorder hoeft er dus niets op de uitgebreide junior-computer te worden aangesloten. In hoofdstuk 12 wél: we gaan uit van de Elekterminal en het ASCII-toetsenbord, dat zijn eerder door Elektuur gepubliceerde bouwprojecten. In dit hoofdstuk leren we ook om te gaan met de Printer Monitor. Met praktijkvoorbeelden, zoals gebruikelijk.

De junior-computer is vanaf boek 1 door ons "een volwassen computer voor beginners" genoemd. Volwassen betekent: volgroeid. Dit boek gaat over uitbreiden, dus over groei. Iedere beginner kan zelf bepalen in welke fase de junior-computer volwassen is. En is dat pas het geval na dit boek 3, dan is men bepaald geen beginner meer.

A. Nachtmann

G.H. Nachbar

P.S. Afgezien van een paar "hard copies" is dit boek niet door de computer gemaakt maar door mensen. De auteurs stellen schriftelijke op- of aanmerkingen van de lezers over de inhoud van dit boek op prijs.

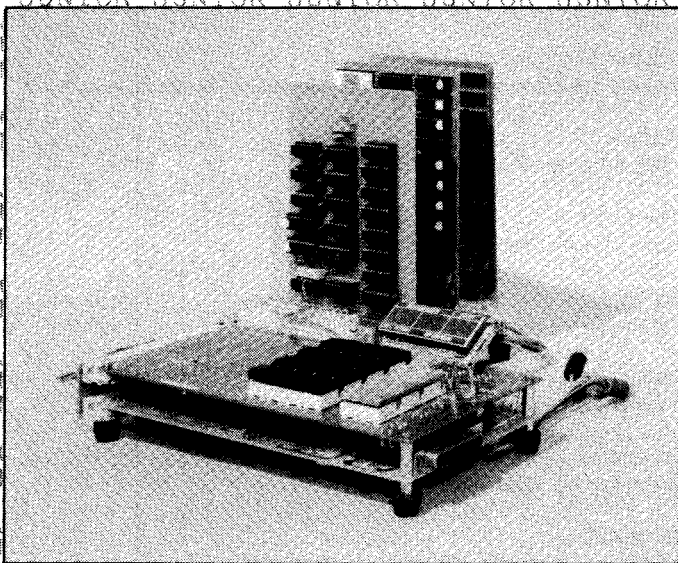
Er is nog meer!

Junior Computer 4 is de tweede helft van het derde boek van de "Junior-trilogie". Het is ondermeer gewijd aan:

- Een nieuw systeemprogramma: de PM-editor (PME)
- De PM-software
- De PME-software
- De TM-software
- Uitgebreide listings van alle systeemprogramma's

Inhoud

Hoofdstuk 10	7
<i>Junior groeit – Van “alles op één kaart” naar “nog meer op meerdere kaarten”</i>	
Hoofdstuk 11	83
<i>Magnetische geheugenuitbreiding – Data van, naar en aan de lopende band</i>	
Hoofdstuk 12	125
<i>Junior op tv! – Méér toetsen en méér te zien</i>	
Aanhangsel 1	191
<i>Basiskaart plus één RAM/EPROM-kaart</i>	
Aanhangsel 2	195
<i>PIA-RAM-uitbreiding van de standaard-monitor</i>	
Aanhangsel 3	200
<i>Vektoren en buskaartgeheugen</i>	
Aanhangsel 4	204
<i>Hex dump van het programma Tape Monitor (TM)</i>	
Aanhangsel 5	206
<i>Hex dump van het programma Printer Monitor (PM)</i>	
Aanhangsel 6	208
<i>RAM-testprogramma</i>	
Aanhangsel 7	215
<i>ASCII-kodetabel</i>	



Junior groeit

van
 "alles op één kaart"
 naar
 "nog meer op meerdere kaarten"

Verscheidene kaarten (= printen) spelen een rol bij het "groter groeien" van de junior-computer. In dit hoofdstuk zullen alle aspecten van dit kaartenspel worden behandeld, inclusief de spelregels. Als alle kaarten op tafel liggen ziet de gegroeide junior-computer er zó uit:

- Extra "digitale grijze cellen" (RAM en/of EPROM) tot maximaal 64 kilobytes.
- Alles wat nodig is om data door te geven van of naar magnetische geluidsband (cassette-interface).
- Een bredere kijk op de buitenwereld door toevoeging van een I/O-bouwsteen (VIA) en van een koppeling (RS232-interface) met randapparatuur, zoals een groot (ASCII-)toetsenbord en een video-terminal of een geschikte printer.

De interface-kaart speelt hierbij een sleutelrol, als troefkaart. In het eerste deel van dit hoofdstuk zal de elektronica die erop zit worden besproken. Dan volgt een uitgebreide bouwbeschrijving. De uitgebreide junior-computer is voor een deel gebaseerd op bestaande, al eerder door Elektuur gepubliceerde kaarten. Hiervan krijgt de RAM/EPROM-geheugenkaart een korte junior-behandeling. Men is niet verplicht om alle voorgestelde uitbreidingen (meteen) te gaan bouwen; een minder vèrgaande uitbreiding past bij een minder diepgaand tasten in de beurs en

is met name mogelijk als het om geheugenuitbreiding gaat. Ook dáárover gaan we het in dit hoofdstuk hebben.

Junior groeit. Naar volwassenheid, het volgroeid zijn — als dat al niet het geval was.

Groei is natuur-lijk. De natuur zorgt voor groei. De natuur van junior-computergebruikers veroorzaakt groeiwensen en groeiprikkels, een vruchtbaar Elektuur-klimaat veroorzaakt de realisatie van de groei, zij het dat dat tijd vergt. En dus geduld. Er is ooit de vergelijking gemaakt tussen mens en computer: de hardware is het vlees en bloed, de software de geest. Vanaf een zekere leeftijd is de lichamelijke groei van de mens voltooid; de geestelijke groei gaat gewoon door. Anders gezegd: de hardware, nodig voor de groei naar de uitgebreide junior-computer is beschikbaar.

De uitgebreide junior maakt het gebruik van bepaalde randapparatuur (cassette-recorder, groot toetsenbord plus video terminal of printer) mogelijk. Daardoor is een bepaalde "geestelijke", software-groei noodzakelijk respectievelijk mogelijk geworden. Junior vergroot zijn woordenschat in machinetaal; denk aan elders in dit boek besproken systeemprogramma's. Maar er is meer: hij kan nu ook "vreemde" talen leren; hij "denkt" dan weliswaar nog steeds in zijn moedertaal, dus in machinetaal, maar hij drukt zich uit, op printerpapier of op het tv-scherm, in een hogere taal, zoals bijvoorbeeld BASIC. Waar het om gaat is dat de software-groei (meer systeemprogramma's, één of meerdere hogere programmeertalen) door kan gaan zonder dat daarvoor, op basis van de uitgebreide junior-computer, nog wezenlijke hardware-uitbreidingen nodig zijn.

Hoezo groeien!?

Bus, kaart en buskaart

We zeggen nou wel dat groei natuurlijk is, maar in ons geval klopt er één ding niet, en dat is maar goed ook. Want normaal gesproken gaat de groei in de natuur vanzelf, het gebeurt hoe dan ook. Het al dan niet groeien naar de uitgebreide junior-computer daarentegen kan men zelf beïnvloeden, bijvoorbeeld door te besluiten tot helemaal geen uitbreiding, of een uitbreiding in stappen. Dat brengt ons op vragen zoals: is het allemaal wel nodig? Zo ja: welke uitbreidingen? Vragen die elke junior-konsument voor zichzelf moet beantwoorden, en dan het liefst los van argumenten zoals: "de burens een nieuwe auto? Dan wij ook". U krijgt hulp bij uw overwegingen bij het nu volgende "bustochtje à la carte".

Eén of meer kaarten?

Geen groei of een beheerste groei, maar geen wildgroei

Voor al amateurgebruik bedoelde computersystemen zijn er in twee soorten: bussystemen en "single board"-systemen. Bij een **bussysteem** is

elk onderdeel op een kaart (= print) ondergebracht. De kaarten zijn onderling even groot (bijvoorbeeld eurokaart: 100 x 160 mm) en doorverbonden via een "bus", dat is een raamwerk van elkaar niet kruisende, parallelle leidingen waarbij punten met gelijke positie (bijvoorbeeld konnektorpennen) onderling zijn doorverbonden. De praktische uitvoering is soms via draadverbindingen, soms via een print: de buskaart. De lengte van de bus is **in principe** onbeperkt. "The sky is the limit", zeggen de Engelsen. Het bussysteem is universeel; het kan (gaandeweg) zeer uitgebreid worden met als gevolg een computer met vele gebruiksmogelijkheden.

Bij een **single board**-systeem is alles op één kaart gezet. Het hangt er maar helemaal van af wat er op zit (en dus hoe groot de kaart is) als het gaat om de vraag wat de gebruikersmogelijkheden ervan zijn. Bij de behoefte aan uitbreidingen is dat "alles op één kaart" vaak een blok aan het been. Men heeft verkeerd gegokt bij de aanschaffing: dat krijg je als je alles op één kaart zet.

"Single board" wordt vooral toegepast bij oefen/leersystemen ("digitale home trainers") en bij betrekkelijk eenvoudige procesbesturingen (waarbij de computer permanent is ingeschakeld). In zijn standaardvorm is de junior-computer een single board-computer. En bedoeld voor de al genoemde toepassingen. Duizenden gebruiken hem. Er is kennelijk sprake van een "Campari-effekt": je hoort en leest zo veel over "chips" en "computers" dat het interessant wordt om er zelf eens iets mee te gaan doen, al is het maar één keer in je leven. Vele applicaties (toepassingen; gebruik van de poorten) zijn ons inmiddels bekend, variërend van A/D-omzetters tot de sturing van een opdamproces bij de halfgeleiderfabrikage. Er is niets op tegen om het, voorlopig of definitief, te houden bij de (legio) mogelijkheden van de standaard-junior-computer en gewoon te vergeten dat er een uitbreidingskonnektor is. Of kan zijn, want dan hoeft je die konnektor zelfs niet te monteren.

Welke groeimogelijkheden?

De meest voor de hand liggende uitbreidingsmogelijkheden

Toch is die uitbreidingskonnektor er niet zomaar. Door dat ding verliest het "alles op één kaart" zijn gokbetekenis. Voor de één is de junior-computer al volwassen, volgroeid zonder er gebruik van te maken, voor de ander is dat pas het geval als via die uitbreidingskonnektor ten minste het volgende kan worden gerealiseerd:

- **Meer geheugen.** Extra RAM voor gebruikersprogramma's en als kladblok (temporaries) voor de variabele output-data van systeemprogramma's, en voor systeem- of vertaalprogramma's die op de band staan. Extra EPROM waarin aanvullende systeemprogramma's kunnen worden gezet. Met concrete voorbeelden van wat men erin kan zetten. Dus niet alleen het huis, maar ook alvast een paar bewoners, zoals "TM" van hoofdstuk 11 en "PM" van hoofdstuk 12.
- **Extra I/O.** De bestaande PIA krijgt een kollega, zodat het verkeer van en naar de buitenwereld over het dubbele aantal "rijstroken" kan beschikken, met als gevolg een grotere capaciteit.
- **Hardware** ten behoeve van het gebruik van de magnetische geluidsband

als **data-archief** (achtergrondgeheugen) en als **data-bibliotheek**. Bij een archief denkt men aan het al dan niet tijdelijk opslaan van al dan niet volttooide programma's, bij een bibliotheek aan geleende, gekopieerde of gekochte programma's: hetzij gebruikersprogramma's, hetzij systeemprogramma's (die hoeven niet perse "resident", in EPROM aanwezig te zijn), hetzij bijvoorbeeld een 8 K-BASIC-vertaalprogramma ("interpreter"). Deze hardware is beter bekend onder de kreet **cassette-interface**. Het gaat om tweerichtingsverkeer; de aanwezigheid van lees/schrijfgeheugen (RAM) is absoluut noodzakelijk, hoewel de inhoud van een EPROM op de band kan worden gezet. Bij de cassette-hardware hoort cassette-software: het systeemprogramma Tape Monitor (TM) dat in hoofdstuk 11 zal worden besproken.

- Hardware ten behoeve van het verkeer van en naar bepaalde randapparatuur, waarbij dat verkeer is gebonden aan afgesproken verkeersregels: de **RS-232-interface**. Welke randapparatuur? Een groot (ASCII-)toetsenbord biedt meer mogelijkheden (toetsfuncties) dan het al aanwezige hexadecimale toetsenbord. En in plaats van een uitvoer (output) op zes 7-segmentdisplays valt er te denken aan een uitvoer op papier (printer) of op het tv-scherm (video terminal). Konkreet denken we aan de Elekterminal plus ASCII-toetsenbord, al eerder door Elektuur gepubliceerde ontwerpen. Zodra er meer toetsen, respectievelijk papier of tv-scherm beschikbaar zijn gaat een flink aantal nieuwe systeemprogramma's tot de mogelijkheden behoren, zoals een tekst-editor, assembler, disassembler (van hexdump naar listing) enzovoorts. Een kant en klaar systeemprogramma (Printer Monitor, PM) op basis van de genoemde randapparatuur komt aan de orde in hoofdstuk 12. En nu is meteen ook alle hardware aanwezig om bijvoorbeeld BASIC op de junior-computer te kunnen draaien.

Tot zover, heel in het kort, de uitbreidingen. Zodra de uitbreidingskonnector wordt gebruikt is de junior-computer dus niet meer "single-board". Is het daarmee zo'n eindeloos (in meerdere betekenissen) bussysteem geworden? Nee! Het gaat hier om afrondende hardware en dat is iets anders dan "eindeloos". Het enige dat er na dit boek 3 nog bijkomt is een 16 K-dynamische RAM-kaart. Geen wezenlijke hardware-uitbreiding maar een goedkoop alternatief voor twee RAM/EPROM-geheugenkaarten met elk 8 K-RAM.

Verder lijkt de in de praktijk gerealiseerde opzet van de uitgebreide junior-computer helemáál niet op een zuiver bussysteem. Alle genoemde uitbreidingen, los van de randapparatuur en een deel van het extra geheugen — dat laatste (buskaartgeheugen) uitsluitend indien men dat wenst — zitten op de interface-kaart, die net zo groot is als de basiskaart (hoofdprint) van de junior-computer, en, zoals zal blijken, daaronder wordt geplaatst.

Aan de interface-kaart kan als men dat wil een bestaande SC/MP-buskaart worden gekoppeld; die dient als drager voor één of meer geheugenkaarten. Ondanks de mogelijke aanwezigheid van een buskaart is de uitgebreide junior-computer geen bussysteem. Maar wat dan wél? Gezien de sandwich-konstruktie van interface-kaart en basiskaart (die wij ons voorstellen; en waarom u niet?) kun je misschien nog het beste spreken van "double single board". Of "dubbeldekker". Nog belangrijker dan de vraag welke

naam het beestje (nou ja ... "tje" ...) moet krijgen is de beschrijving ervan. Wat let ons om dat nu te doen?

De interface-kaart

De kaart waar alles om draait

Afgezien van de randapparatuur (die hoort er wel bij, maar rekenen we niet tot de uitgebreide junior-computer) zitten alle of niet alle (dat hangt van uw keuze af) uitbreidingen op de interface-kaart. Of er behalve die kaart nog een buskaart + geheugenkaart(en) bijkomen hangt helemaal af van de grootte van uw bytes-honger. We houden in ieder geval rekening met een flinke eetlust.

Het woord "interface" betekent koppeling, overgang, aanpassing. De longen bijvoorbeeld vormen de interface van "lucht" en "bloedsomloop". Men kan hier denken aan de koppeling van de standaard-junior-computer met de uitbreidingen en aan de koppeling met de buitenwereld via de extra I/O, cassette- of RS232-interface. Een wezenlijk kenmerk van interface is het tweerichtingsverkeer. Nu zijn we aan de bespreking toe van de elektronica op de interface-kaart; dat gebeurt zonder blokschema-"ouverture".

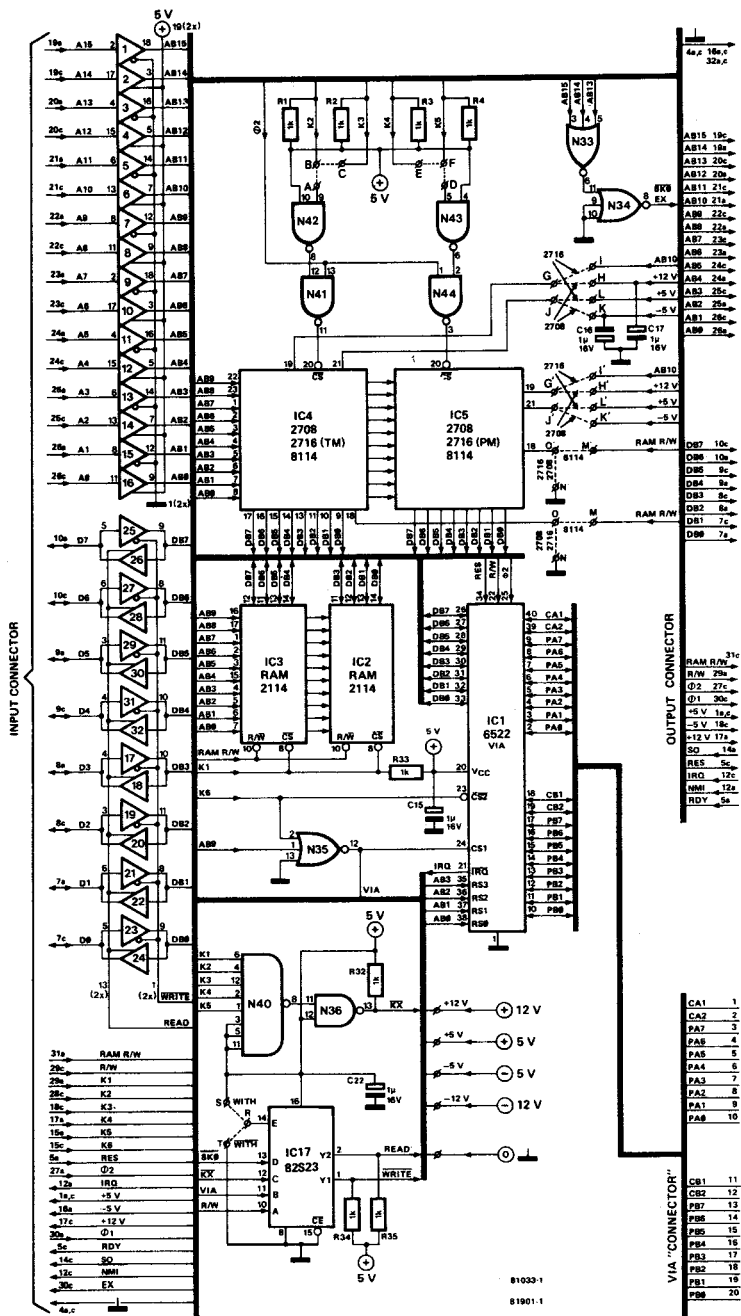
Alle elektronica van en op de interface-kaart is terug te vinden in de figuren 1 en 2. We beginnen met figuur 1.

Buffering

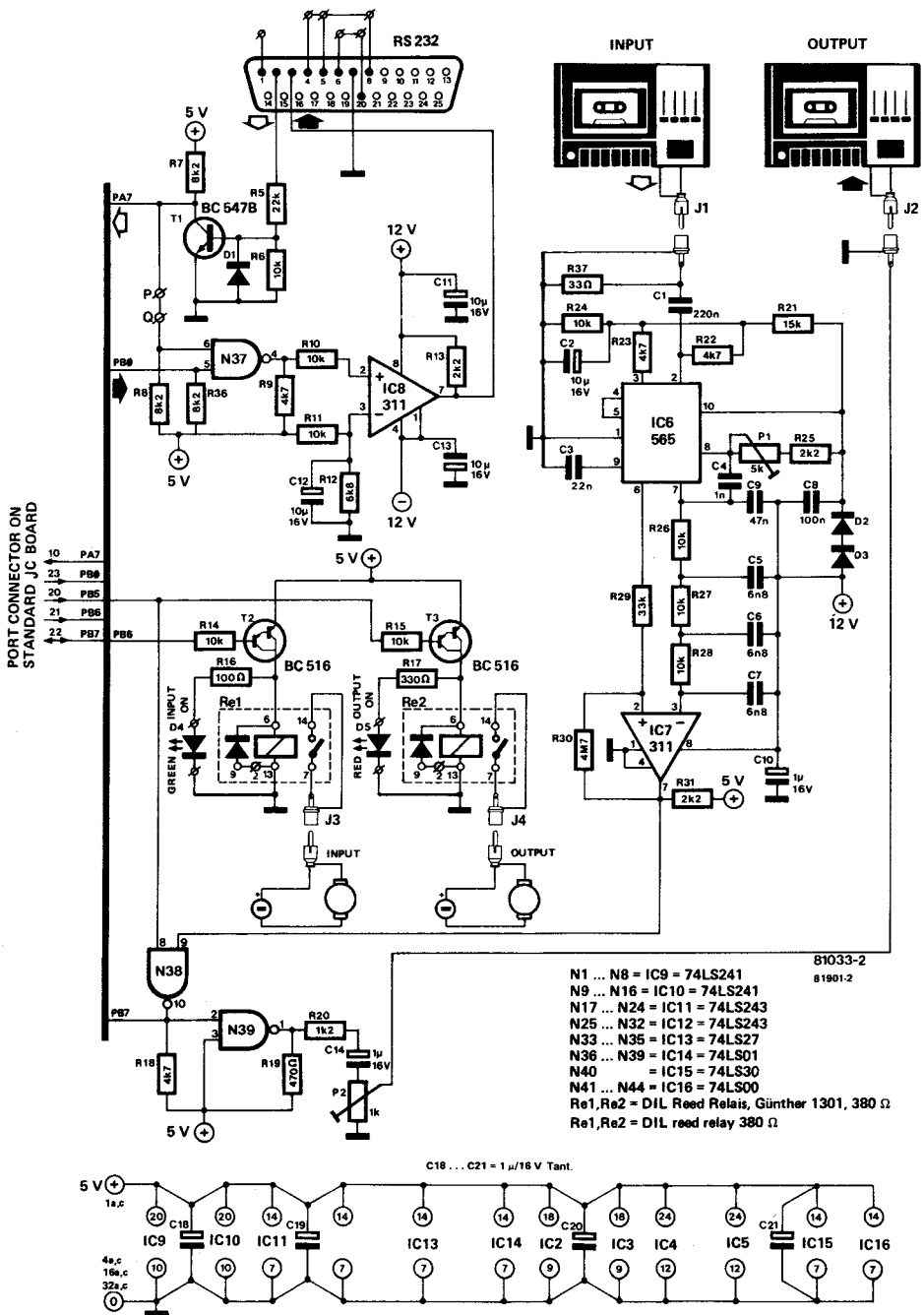
Tegen een stootje kunnen

De uitbreidingen beginnen aan de ingangskonnektor (INPUT CONNECTOR), links in figuur 1. Deze moet zijn verbonden met de uitbreidingskonnektor (EXPANSION CONNECTOR) van de standaard-junior-computer. Een heel wat dunnere "navelstreng" tussen de "moeder-jc" en de uitbreidingen wordt gevormd door vijf verbindingen naar "moeders" poortkonnektor (PORT CONNECTOR); zie figuur 2. De diverse signalen aan de ingangskonnektor ziet men, met uitzondering van EX en K1 .. K6, want die zijn alleen op de interface-kaart nodig, terug bij de uitgangskonnektor (OUTPUT CONNECTOR), rechtsboven in figuur 1. Deze dient voor de koppeling met de buskaart, die op zijn beurt plaats biedt aan één of meer geheugenkaarten. Let erop dat een signaal met een pen a aan de ingang een pen c heeft aan de uitgang, en omgekeerd. Het "waarom" hiervan komt aan de orde bij de bouwbeschrijving, verderop in dit hoofdstuk.

Adreslijnen A zien we terug als AB en datalijnen D als DB. De B staat voor "gebufferd". Buffering, wat is dat eigenlijk? De huiselijke betekenis van de reden tot buffering is misschien het beste samen te vatten met: "anders kan Bruin het niet trekken". Geen buffering is als een vrachtauto met een bromfietsmotor: als de zaak al in beweging komt moeten we ons niet al te veel voorstellen van de snelheid. Zonder buffering raken de adres- en data-lijnen elektrisch overbelast; de vereiste logische niveaus, dus grofstoffelijke elektrische spanningen, komen helemaal niet of veel te laat tot stand. En



Figuur 1. Het gedeelte van de elektronica op de interfacekaart dat te maken heeft met de drie bussen: adresbus, databus en stuurbus. Dus extra geheugen I/O, buffering, adres-dekodering en databuffersturing.



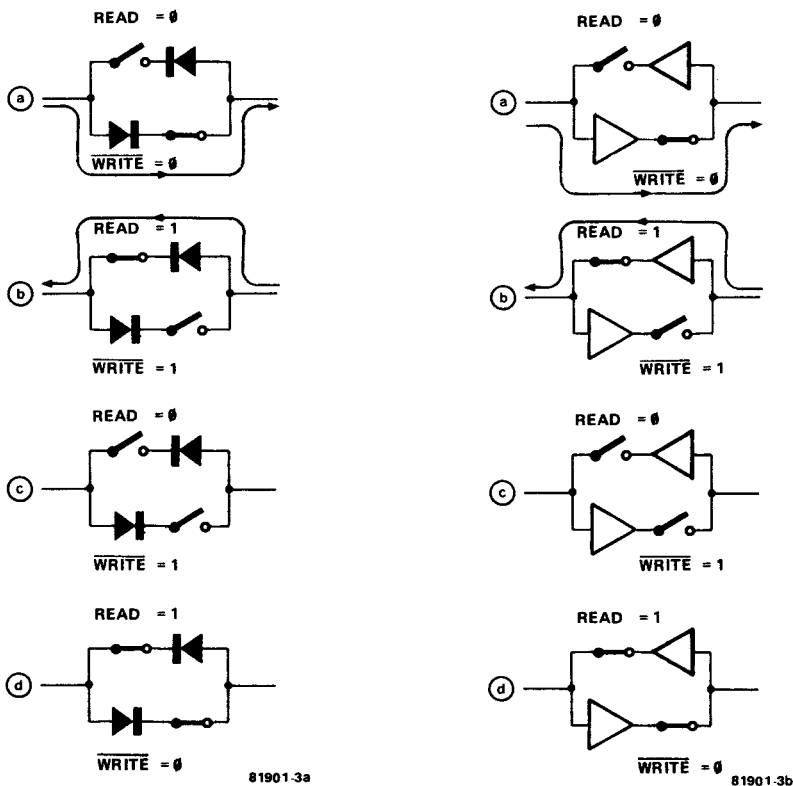
Figuur 2. Het gedeelte van de elektronica op de interfacekaart dat te maken heeft met datatransport van en naar RS 232-genormeerde randapparatuur en één of twee cassette-recorders. De aansluiting op de junior-computer gaat via vijf verbindingen met de bestaande (PIA-)poortkonnektor.

tijd (denk aan de 1 MHz-klok; elke miljoenste seconde een "hartslag") is belangrijk. Een nodeloze tijdsvertraging van 10 nanosekonden (tien miljardste) kan al teveel van het goede zijn.

Een andere reden voor buffering is dat het in het geval van de datalijnen nu mogelijk is om de datastroom te blokkeren in een bepaalde richting of in beide richtingen. Data kent een leesrichting, naar de μP toe, en een schrijfrichting, vanuit de μP . Adressering gebeurt altijd op initiatief van de microprocessor, gaat dus van de μP uit.

Stellen we ons een buffer voor als een driehoek met een verticale zijde: de ingang, en een uitgang, te weten de punt van de driehoek. In figuur 1 zien we 32 van zulke driehoeken: N1 ... N32. De adresbuffers zijn N1 ... N16. Omdat adressering altijd vanuit de 6502- μP op de standaardkaart plaatsvindt staan de buffers met de ingang links en de uitgang rechts.

De databuffers zijn N17 ... N32. Hier is sprake van een even genummerde



Figuur 3. Zeg het met dioden: De werking van de databuffers N17 ... N32 van figuur 1 voorgesteld met "elektrische datastromen" en doorlatende of sperrende "diode-poorten".

groep buffers met ingang rechts, uitgang links, en van een groep oneven genummerde buffers met ingang links, uitgang rechts. Zoals al opgemerkt hebben de databuffers tevens een politieagent-functie. Dit aspect is geïllustreerd in figuur 3. Vergelijk een datastroom met een elektrische stroom en het verkeersproces met de stand van twee schakelaars, elk in serie met een diode. De μP zit aan de linker kant in figuur 3, de uitbreidingen rechts. De stand van de schakelaars is een vertaling van de logische nivo's van de signalen WRITE en READ. Er zijn vier situaties mogelijk, waarvan er één ongewenst is:

1. De oneven genummerde poorten N17 ... N32 zijn actief, de even genummerde inactief. Dat is het geval als WRITE en READ allebei logisch nul zijn. Er is een datastroom mogelijk van de standaardkaart naar de uitbreidingen. Dat is het geval bij het schrijven van data in niet-standaardgeheugen of I/O. Plaatje @ van figuur 3.
2. De oneven genummerde poorten N17 ... N32 zijn inactief (geïsoleerd, "tri-state"), dat wil zeggen hoogohmig aan ingang en uitgang; de even genummerde zijn actief. Dat is zo als WRITE en READ logisch één zijn. Er is een datastroom mogelijk van de uitbreidingen naar de standaardkaart. Dat is het geval bij het lezen van data uit niet-standaardgeheugen of I/O. Plaatje @ van figuur 3.
3. Alle poorten N17 ... N32 zijn inactief als WRITE logisch één is en READ logisch nul. De datastroom van en naar de uitbreidingen is dan geblokkeerd. Dat moet als er gelezen en mag als er geschreven wordt op de standaardkaart. Plaatje @ van figuur 3.
4. De vierde mogelijkheid, namelijk WRITE logisch nul en READ logisch één zou inhouden dat alle 16 poorten N17 ... N32 tegelijk actief zijn. Dit mag nóóit gebeuren. Het is: òf lezen, òf schrijven, maar nooit beide tegelijk.

N.B. Let op de overeenkomst in positie van de "diodepunten" (katoden) van figuur 3 en de punten van de driehoeken in figuur 1!

De VIA

Extra I/O

De VIA IC1, van het type 6522, is een hoofdstuk apart. En wel een hoofdstuk van boek 4. Hier zij opgemerkt dat deze "Versatile (= veelzijdige) Interface Adapter" nog meer mogelijkheden in huis heeft dan de standaard-I/O-bouwsteen (PIA) van het type 6532. De voor de gebruiker van belang zijnde verbindingen met de buitenwereld zijn te vinden bij de VIA "CONNECTOR" (is geen echte konnektor, althans niet op de print, vandaar de aanhalingstekens), rechtsonder in figuur 1. De 6522 wordt gestuurd vanuit de adreslijnen AB0 ... AB3 en met diverse signalen van de stuurbus. De aanwezigheid van acht datalijnen zal ook niemand verbazen, evenals het uitgangssignaal IRQ (strikt genomen IRQ), dat te maken heeft met een eventuele interrupt na een time out van de timer.

Evenals de 6532 heeft de 6522 twee selektiesignalen. De chip-select-ingang CS2 is net als de CS2 van de 6532 verbonden met het uitgangssignaal K6 van de adresdekoder IC6 op de standaardkaart. Zijn kollega, CS1, is verbonden met de uitgang van poort N35, die alles aan de weet komt via

K6 en AB9. Voor de selectie, aktivering van de VIA moet $\overline{CS2}$ logisch nul zijn en CS1 logisch één. Nu is N35 een NOR-poortje; de uitgang ervan gaat alleen de hoogte in als zowel AB9 als K6 het lagerop zoeken (voor de 6532 moet A9 nou juist één zijn).

Als we vaststellen dat A8 of AB8 niet is aangesloten op de PIA en ook niet op de VIA, dan geldt voor de VIA 6522 het volgende adresbereik:

1800 ... 18FF

(ook te bereiken op 1900 ... 19FF)

(AB8 = X; AB9 = 0; K6 = 0)

Voor de PIA 6532 geldt het adresbereik:

1A00 ... 1AFF

(ook te bereiken op 1B00 ... 1BFF)

(A8 = X; A9 = 1; K6 = 0)

Sluiten we de dubbeladressering vanwege A8 = AB8 = X uit, dan zijn er zowel voor de PIA als voor de VIA 256 adressen beschikbaar. Net als bij de 6532-PIA (19 verschillende "geheugenplaatsen" die meervoudig zijn geadresseerd) zijn bij de VIA niet op al die 256 adressen 256 verschillende geheugenplaatsen aangesloten. Zoals men ziet in figuur 1 zijn de adreslijnen AB4 ... AB7 niet aangesloten. Er zijn dus 16 verschillende VIA-geheugenplaatsen.

Waarom eigenlijk een extra I/O-bouwsteen? Wel, de PIA is al belast met huishoudelijke taken zoals de bediening van het zesdelig 7-segmentdisplay en van het hexadecimale toetsenbord. Daar komen nog taken bij, in het kader van de cassette- en RS232-interface. Er zijn daardoor beperkingen verbonden aan programma's met de PIA, in combinatie met het gebruik van de monitor. Wil men bijvoorbeeld een programma à la hoofdstuk 6 van boek 2 stap voor stap doorlopen, dan kan het goed fout gaan (even los van het feit dat men dan kritische toepassingen van de timer wel kan vergeten). Bij stap voor stap hoort namelijk de sprong, na elke instructie, naar de SAVE-ingang van de monitor. Sla hoofdstuk 7 van boek 2 er maar op na. Indien de toestand van de poorten geheel of gedeeltelijk afwijkt van de voor de juiste werking van de monitor vereiste toestand, kan de monitor niet helemaal of helemaal niet functioneren.

En zo zijn er meer ongewenste situaties denkbaar. Dat wordt allemaal voorkomen als men de PIA reserveert voor de interne stofwisseling van de junior-computer (monitor-subroutines blijven uiteraard beschikbaar voor gebruikersprogramma's), en de VIA voor 100% bestemt voor eigen programma's en applicaties.

Intermezzo

Adresdekodering

Van adres naar "postkode"

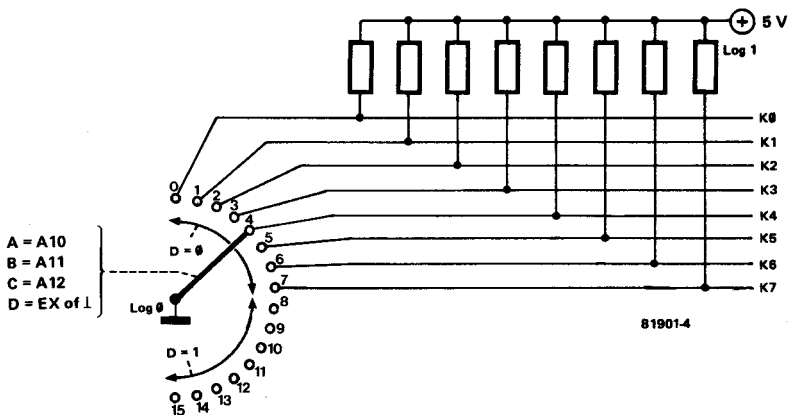
Zoals in elke stad of dorp groepen adressen (woningen etc.) iets gemeenschappelijk hebben, namelijk de postkode, zo hebben bij de junior-computer bepaalde groepen adressen (geheugenplaatsen) een gemeenschappelijk kenmerk, namelijk een bepaald selectiesignaal. Voor de te selektieren en te adresseren zaken op de standaardkaart en — zoals zal blijken — op de interface-kaart zijn dat de signalen K0 ... K7, afkomstig uit de adres-

dekoder IC6 van de standaardkaart. En voor, via de buskaart aangesloten geheugen zijn óók van die selektiesignalen nodig; die worden op de geheugenkaart(en) zèlf geproduceerd (zie de bespreking van de RAM/EPROM-kaart, aan het eind van dit hoofdstuk).

Omdat dat voor de bespreking hier en nu zinvol is halen we nog even die IC6 op uit hoofdstuk 1. In figuur 4 is de werking van die adresdekoder voorgesteld in de vorm van een keuzeschakelaar (hardware) en in tabel 1 in de vorm van een zogenaamde "waarheidstabel" (de software-wereld van enen en nullen). Als een bepaald selektiesignaal $K_0 \dots K_7$ wordt gekozen is de "gelukkige" K logisch nul; de andere zeven worden elk via een weerstand omhoog getrokken, naar de +5 V-voedingsspanning, en zijn logisch één. Een nul leidt dus tot uitverkiezing. Vandaar dat de meeste geheugenchips een $\overline{CS}$ hebben; het streepje duidt op: actief indien logisch nul.

Nu heeft de schakelaar van figuur 4 geen acht, maar 16 standen. De andere acht kontakten zijn niet aangesloten. Dat brengt ons op de vraag wat er van invloed is op de stand van die schakelaar. Dat zijn drie adreslijnen. En verder het punt D van IC6, dat óf aan massa ligt (dus logisch nul is), óf verbonden is met aansluiting EX op de uitbreidingskonnektor. De keuze: massa of EX wordt bepaald door die ene draadbrug op de standaardkaart.

Ligt D aan massa, en is dus D logisch nul, dan is één van de kontakten $0 \dots 7$ aan bod. In het geval $D = EX$ hangt het van het logisch nivo van EX af of één van de kontakten $0 \dots 7$ (EX logisch nul), danwel één van de loze kontakten $8 \dots 15$ (EX logisch één) is uitverkoren. Bekijken we nu de rechter kolom van tabel 1. Wat zien we? Als D nul is wordt er iets gedekodeerd op de standaardkaart of op de interface-kaart. Als D één is is dat niet het geval en moet het dus gaan om iets dat niet op de beide kaarten zit, dus om geheugen dat via de buskaart is aangesloten (buskaart-



Figuur 4. Een grofstoffelijke voorstelling van zaken van de adresdekoder IC6 op de standaardkaart: met een keuzeschakelaar. De acht aangesloten kontakten horen bij de selektie van iets adresseerbaars op de basis- of interface-kaart.

Tabel 1. Het enen-en-nullenverhaal (waarheidstabel) van de adresdekoder IC6 op de standaardkaart.

D = 1 (ϕ) of D = EX = 8K $\bar{\phi}$	C = A12	B = A11	A = A10	K7	K6	K5	K4	K3	K2	K1	K $\bar{\phi}$	s = standaardkaart i = interface-kaart b = buskaartgeheugen
0	0	0	0	1	1	1	1	1	1	1	0	s
0	0	0	1	1	1	1	1	1	1	0	1	i
0	0	1	1	1	1	1	1	1	0	1	1	i
0	1	0	0	1	1	1	0	1	1	1	1	i
0	1	0	1	1	1	0	1	1	1	1	1	i
0	1	1	1	1	0	1	1	1	1	1	1	s + i
1	0	0	0	1	1	1	1	1	1	1	1	s
1	0	0	1	1	1	1	1	1	1	1	1	b
1	0	1	0	1	1	1	1	1	1	1	1	b
1	0	1	1	1	1	1	1	1	1	1	1	b
1	1	0	0	1	1	1	1	1	1	1	1	b
1	1	0	1	1	1	1	1	1	1	1	1	b
1	1	1	0	1	1	1	1	1	1	1	1	b
1	1	1	1	1	1	1	1	1	1	1	1	b

geheugen). We zien dat dan D met EX moet zijn verbonden, omdat D anders niet één kan worden. We zien ook dat zonder buskaartgeheugen de draadbrug rond D kan blijven zitten zoals hij was: naar massa. Dat komt omdat zonder buskaartgeheugen alle, tot nu toe nog niet gebruikte K-aansluitingen van IC6 worden benut voor alles wat er aan uitbreidingen kan worden geadresseerd, en dus moet kunnen worden geselecteerd.

Geheugenuitbreiding

Extra RAM- en/of EPROM-geheugensteuntjes

De geheugenuitbreiding omvat een aantal IC's op de interface-kaart, desgewenst aangevuld met buskaartgeheugen. De tot de laatste behorende RAM/EPROM-kaart komt aan het eind van dit hoofdstuk aan de orde, de extra bytes op de interface-kaart nu. De door K $\bar{\phi}$ en K7 geselecteerde geheugenbereiken zijn volledig benut op de standaardkaart, K6 is voor I/O en de door K1 . . . K5 geselecteerde bereiken kunnen volledig worden benut voor extra geheugen. Op de interface-kaart. Elke K is goed voor 1 K, dus maximaal 5 K. En daarmee is de volledige 8 K benut, die standaard is gedecodeerd (zie tabel 1).

Eerst maar de RAM. Extra RAM is nooit weg. De standaard-RAM ligt op de pagina's 00 ... 03. Als we 1 K RAM erbij nemen (vier pagina's dus) zou het heel fijn zijn als dat de pagina's 04 ... 07 zijn. Want dan zijn grotere gebruikersprogramma's "aan één stuk" mogelijk. Wel, het is fijn, want deze RAM (IC2 plus IC3; elk IC 1024 halve bytes, nibbles) wordt geselecteerd (op de $\overline{CS}$ -ingang, die dan logisch nul moet zijn) met K1. Dat is dus dik in orde.

Voor zowel IC4 als IC5 kan men naar keuze 1K-RAM (8114), 1K-EPROM (2708) of 2K-EPROM (2716) nemen. De selectie gebeurt met één of twee signalen K2 ... K5 per IC en met behulp van het kloksignaal $\Phi 2$ (poorten N41 ... N44). Dat laatste vanwege een goede timing van het lezen en (indien van toepassing) schrijven.

De adressering ziet er als volgt uit:

K2 = 0: adressen 0800 ... 0BFF

K3 = 0: adressen 0C00 ... 0FFF

K4 = 0: adressen 1000 ... 13FF

K5 = 0: adressen 1400 ... 17FF

In het geval: 2708 of 8114 heeft men per IC meerdere mogelijkheden, omdat dan niet de maximale 4K van IC4 plus IC5 wordt benut. Bij de keuze voor twee stuks 2716 is dat wel het geval: voor IC4 worden K2 en K3 gekombineerd en voor IC5 K4 en K5, met als gevolg een adresbereik 0800 ... 0FFF voor IC4 en 1000 ... 17FF voor IC5.

Ten behoeve van de adresselektie zijn de soldeerpenen A ... F aanwezig; deze moeten afhankelijk van de keuze voor IC4 en IC5 op een bepaalde manier met elkaar worden doorverbonden. Hetzelfde geldt voor de penen G ... O en G' ... O'. Verbindingen tussen deze punten hangen samen met de keuze van de voedingsspanningen en met de vraag of een extra adreslijn AB10 of het signaal RAM-R/W nodig is of niet.

N.B. Zie ook tabel 7.

Volledige of onvolledige adresdekodering?

Zestien of dertien adreslijnen?

De drie hoogste adreslijnen, A15 ... A13, zijn niet gebruikt op de standaardkaart. Drie van de zestien adreslijnen spelen dan geen enkele rol; de overige dertien wel. Met dertien van die adreslijnen kun je 2 tot de macht 13 oftewel 8K (1K = 1024) verschillende geheugenplaatsen adresseren. En dat is precies de 8K die met de acht K-signalen van de adresdekoder (zie figuur 4) kan worden geselecteerd. Er is al verteld dat die 8K volledig kan worden opgesoupeerd op de standaardkaart en de interface-kaart. Zolang we ons tot die twee kaarten beperken, **dus zonder buskaartgeheugen**, hebben we nog steeds genoeg aan een systeem met dertien adreslijnen.

Maar nu willen we ook nog buskaartgeheugen gebruiken. De genoemde 8K is bezet en we moeten dus een beroep doen op de resterende 64 min 8 is 56K. Nu zullen de drie tot nu toe "ongebruikte" adreslijnen wel een rol gaan spelen. Er moeten nu twee dingen gebeuren:

1. Er moet voor worden gezorgd dat, als er buskaartgeheugen moet worden geadresseerd, en dus geselecteerd, er niets wordt geselecteerd dat op de andere twee kaarten zit; dus niets van de al vele malen ge-

noemde 8K (zie ook tabel 1). Je kunt namelijk niet op twee adressen tegelijk schrijven of op twee adressen tegelijk lezen, want dat wordt een puinhoop. Bekijk figuur 4 en tabel 1 en de oplossing is duidelijk: verbind punt D met EX (draadbrug omleggen) en zorg ervoor dat bij de selectie van buskaartgeheugen die EX logisch één is. Immers, dan staat de schakelaar in een loze positie en is geen van de signalen $K0 \dots K7$ nul.

2. Zorg voor aanvullende adresdekodering op de van het buskaartgeheugen deel uitmakende RAM/EPROM-kaart(en). Dát bespreken we dan aan het eind van dit hoofdstuk.

Hoe krijg je die EX nou één als dat moet? Voer de adreslijnen A13, A14 en A15 toe aan een OR-poort. In figuur 1 (rechtsboven) is dat gebeurd via de NOR-poort N33 en de als inverter gebruikte N34, welke laatste EX levert. Verbind EX met punt D van IC6 op de standaardkaart. Zodra één van de drie adreslijnen ook maar logisch één is wordt EX logisch één; alleen als ze alle drie aan de grond zitten (logisch nul) is EX logisch nul. Dit betekent dat het adresplaatje er nu zó uitziet:

Al het adresseerbare (geheugen, PIA, VIA) op standaard- en interface-kaart is bereikbaar op de adressen:

$0000 \dots 1FFF$ (pagina's $00 \dots 1F$)

Dus 32 pagina's. Dat is samen 8K. Klopt!

Afhankelijk van de hoeveelheid (maximaal 56K) legt het buskaartgeheugen geheel of gedeeltelijk beslag op het adresbereik:

$2000 \dots FFF$ (pagina's $20 \dots FF$)

Dus maximaal $256 - 32 = 224$ pagina's van elk een kwart K.

Zolang er met drie adreslijnen minder kan worden gewerkt (dus zonder buskaartgeheugen) is er sowieso geen sprake van volledige adresdekodering. Daar hebben we het over gehad. Maar er is nog iets meer aan de hand: er is dan eveneens sprake van **meervoudige adressering**. Dát bespreken we nu.

Een adres geeft men weer met vier hexadecimale cijfers (= nibbles). Tot het linkse nibble van dat adres, en dus tot het linker nibble van de pagina waar dat adres bij hoort, horen de "overbodige" adreslijnen A13, A14 en A15. Omdat elk van die drie hoog of laag mag springen zonder dat je er iets van merkt, komt elk adres dan acht keer voor. Pagina's $0X$, met $X = 0 \dots F$, bereik je ook op de pagina's $2X, 4X, 6X, 8X, AX, CX$ en EX . Pagina's $1X$, met $X = 0 \dots F$, kom je ook tegen op de pagina's $3X, 5X, 7X, 9X, BX, DX$ en FX . Let wel: uitsluitend bij drie niet gebruikte adreslijnen.

Zodra het signaal EX wordt toegevoerd aan punt D van de adresdekoder IC6 op de standaardkaart is:

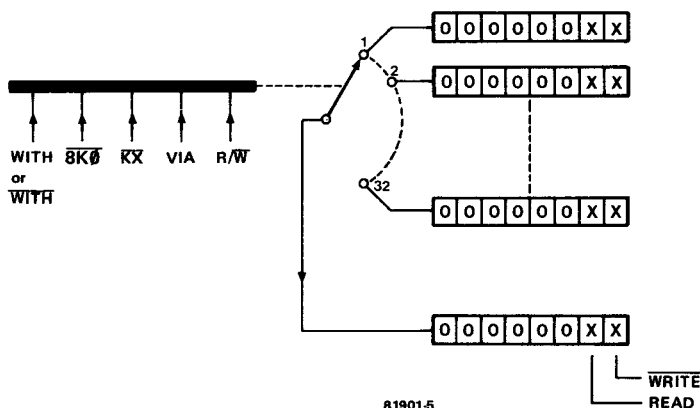
1. aan een noodzakelijke voorwaarde voor volledige adresdekodering voldaan; echt volledige adresdekodering ontstaat pas als de volledige 56K buskaartgeheugen is aangesloten;
2. er geen sprake meer van meervoudige, maar enkelvoudige adressering. Eén adres per geheugenplaats is meer dan voldoende, en bij gebruik van buskaartgeheugen zelfs noodzakelijk. Zonder buskaartgeheugen blijft er sprake van achtevoudige adressering.

Databuffers manipuleren

Data-verkeersregeling

Het uitgangssignaal EX van daarnet gaat, verkleed als $\overline{8K\emptyset}$, eveneens naar een adresingang van de PROM IC17. Deze PROM bevat 32 bytes van elk 8 bits; van die acht zijn alleen de bits Y1 (= WRITE) en Y2 (=READ) van belang. Deze bits sturen de databuffers N17 ... N32. Voor 32 bytes zijn de vijf adreslijnen E ... A genoeg; die adreslijnen omvatten vijf, direct nader te omschrijven signalen.

De werking van de PROM is in figuur 5 in beeld gebracht. Een keuzeschakelaar heeft 32 posities. De feitelijke positie hangt af van de logische nivo's van 5 adreslijnen. Afhankelijk van de positie ontstaat op de 8 data-uitgangen een bepaald bitpatroon van 8 bits, één byte. De 32 bytes moeten van te voren worden "ingebakken" (geprogrammeerd) in de PROM.



Figuur 5. De PROM IC17, voorgesteld als een "byte-omschakelaar". Er zijn vijf adreslijnen van invloed op de stand van de schakelaar, die 32 verschillende posities kent, maar geen 32, onderling verschillende bytes. Niet alle 32 posities worden daadwerkelijk benut.

Tabel 2 geeft het baksel weer in de vorm van enen en nullen. Zolang de adreslijnen alle een gedefinieerd logisch nivo hebben is een bepaald bitpatroon op de uitgang van de PROM aanwezig. De PROM kan continu worden gelezen; dat komt omdat de stuuringang $\overline{CE}$ aan massa ligt.

Waarom doen we zo moeilijk met die PROM? Tenslotte moet je die programmeren en uit wat er in het hoofdstukje over buffering is gezegd zou je kunnen konkluderen dat kan worden volstaan met het aan elkaar gelijk maken van die signalen R/W, READ en $\overline{WRITE}$. Waarom moeilijk doen als het makkelijk kan?

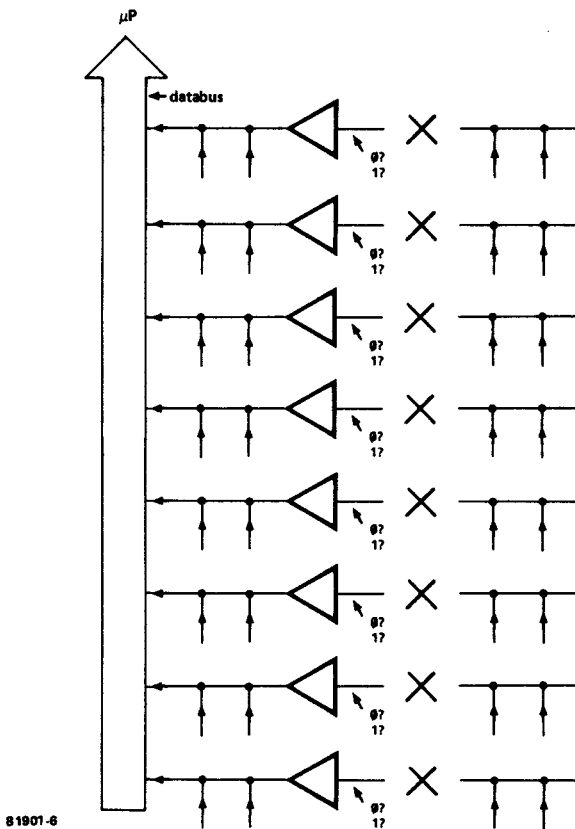
Om twee redenen.

Ten eerste: stel we lezen data op de standaardkaart. Als R/W dienst zou doen als READ en als $\overline{WRITE}$, staan er tijdens dat lezen acht databuffers met de ingang gericht op de standaard-databus. Hun ingangen hangen niet

Tabel 2. De inhoud van de PROM IC17. De programmeergegevens zijn kompakt weergegeven als hexadecimale adressen en data.

PROM-adres (hex)	E = WITH of WITH	D = 8K0	C = KX	B = VIA	A = RW	Y3 ... Y8	Y2 = READ	Y1 = WRITE	PROM-data (hex)	
00	0	0	0	0	0	0	0	0	00	①
01	0	0	0	0	1	0	1	1	03	
02	0	0	0	1	0	0	0	0	00	
03	0	0	0	1	1	0	0	0	00	
04	0	0	1	0	0	0	0	1	01	②
05	0	0	1	0	1	0	0	1	01	
06	0	0	1	1	0	0	0	0	00	
07	0	0	1	1	1	0	1	1	03	
08	0	1	0	0	0	0	0	0	00	③
09	0	1	0	0	1	0	0	0	00	
0A	0	1	0	1	0	0	0	0	00	
0B	0	1	0	1	1	0	0	0	00	
0C	0	1	1	0	0	0	0	1	01	④
0D	0	1	1	0	1	0	0	1	01	
0E	0	1	1	1	0	0	0	0	00	
0F	0	1	1	1	1	0	0	0	00	
10	1	0	0	0	0	0	0	0	00	⑤
11	1	0	0	0	1	0	1	1	03	
12	1	0	0	1	0	0	0	0	00	
13	1	0	0	1	1	0	0	0	00	
14	1	0	1	0	0	0	0	1	01	⑥
15	1	0	1	0	1	0	0	1	01	
16	1	0	1	1	0	0	0	0	00	
17	1	0	1	1	1	0	1	1	03	
18	1	1	0	0	0	0	0	0	00	⑦
19	1	1	0	0	1	0	0	0	00	
1A	1	1	0	1	0	0	0	0	00	
1B	1	1	0	1	1	0	0	0	00	
1C	1	1	1	0	0	0	0	0	00	⑧
1D	1	1	1	0	1	0	1	1	03	
1E	1	1	1	1	0	0	0	0	00	
1F	1	1	1	1	1	0	0	0	00	

effektief aan geheugen (zijn daarvan geïsoleerd), omdat dat geheugen niet is geadresseerd. De ingangen leiden dus een leven als vrije jongen: elke ingang kan één of nul zijn, elke uitgang dus ook. Die bufferuitgangen zijn laagohmig — daar zijn het nou juist buffers voor — en ze kunnen dus de databus "hun wil opleggen", met als gevolg mogelijke fout gelezen data. In wezen is het een soort poging om twee adressen tegelijk parallel te lezen. Alles gaat goed zolang het in beide gevallen om een nul of een één gaat. Maar bij ongelijke logische nivo's wint de bufferuitgang het. Zie ook figuur 6. Remedie: blokkeer die buffers (plaatje © van figuur 3). In het



Figuur 6. De ongewenste situatie die optreedt als er door de 6502 wordt gelezen uit iets adresseerbaars op de standaardkaart en als de acht databuffers niet zijn geblokkeerd. Er is sprake van acht keer "twee data parallel", waarbij de data op de buffer-uitgangen overheerst. Er zijn geen problemen als alle acht te lezen databits in logisch nivo gelijk zijn aan het uitgangsnivo van de bijbehorende buffer. De enige, gegarandeerd goede oplossing is echter om in dergelijke gevallen de databuffers te blokkeren.

geval van lezen op de standaardkaart heeft de 6502 immers achter de buffers (rechts in figuur 3) niets te zoeken, laat staan vinden! Overigens: bij het schrijven van data op de basiskaart heeft de toestand van de databuffers geen enkele schadelijke invloed en zijn blokkeermaatregelen niet strikt noodzakelijk.

Ten tweede: de vektoren NMI, RES en IRQ zitten eigenlijk in pagina FF. Dat weten we nog van hoofdstuk 3. Pagina FF maakt, indien er geheugen op is aangesloten, deel uit van buskaartgeheugen. Indien gekozen wordt voor buskaartgeheugen moet op pagina FF EPROM zijn aangesloten, met op de adressen FFFA . . . FFFF door de 6502 op te halen vektor-data. Doet men dat niet dan komt de junior-computer, via de opstart-vektor RES, nooit tot leven.

Kiest men niet voor buskaartgeheugen, dan moeten de vektoren zoals vanouds uit de standaard-EPROM worden opgehaald. Dat gebeurt net als vroeger door de zaak te beduvelen: isoleer de databuffers en de drie hoogste adreslijnen A13, A14 en A15 hebben totaal geen invloed op de gebeurtenissen. Omdat zonder buskaartgeheugen punt D van de adresdekoder op de standaardkaart aan massa ligt en niet aan EX, wordt tijdens het ophalen van één van de drie vektoren de standaard-EPROM geselecteerd, via K7.

Het is nu misschien ook duidelijk waarom de vektoren uit buskaartgeheugen moeten komen (dus uit EPROM op pagina FF) als er buskaartgeheugen is. Zou men de vektoren dan namelijk uit de standaard-EPROM willen halen, dan zou de al beschreven beduveltruuk betekenen dat, vanwege de isolatie van de databuffers, het buskaartgeheugen geïsoleerd, dus onbereikbaar, dus zinloos wordt!

De PROM

Een PROMenade van vijf adreslijnen

De vijf adreslijnen die iets te zeggen hebben over de PROM zijn:

- A. Het signaal R/W of eigenlijk R/ $\overline{W}$ (het streepje betekent: actief als logisch nul, één is lezen, nul schrijven). Hangt samen met de vraag of er sprake is van lezen of schrijven. Vrij logisch dat R/W een vinger in de PROM-pap heeft.
- B. Het signaal VIA. Komt uit N35 en is tevens de CS1 van de VIA IC1. Is logisch één als K6 en AB9 nul zijn. Dus als de VIA is geselecteerd. Van het 1K-adresblok met K6 = 0 zit de PIA vóór de buffers en de VIA erachter.
- C. Het signaal $\overline{KX}$. Is afkomstig uit N36, die de uitgang van N40 aktiveert. Is nul als één der leidingen K1 . . . K5 nul is, dus als er geheugen op de interface-kaart aan bod is.
- D. Het signaal $8K\overline{0}$ is gelijk aan EX en is nul als de eerste 8K wordt geadresseerd (geheugen en I/O op de standaard- en interface-kaart) en één als het gaat om buskaartgeheugen.
- E. Punt E van IC17 is met +5 V verbonden (draadbrug RS), of met massa (draadbrug RT). Punt E moet nul zijn als er geen buskaartgeheugen is aangesloten (vandaar het bijschrift $\overline{WITH}$, de engelse en logische versie van "niet-met" = zonder). Punt E moet logisch één zijn als er wel buskaartgeheugen is aangesloten (bijschrift WITH, "met"). De draadbrug-oplossing is gekozen omdat deze samenhangt met een lezerskeuze, namelijk: buskaartgeheugen of niet.

In tabel 2 is de inhoud van de PROM overzichtelijk samengevat. Alle programmeergegevens staan erin. Uw PROM kan eventueel door ons, Elektuur dus, worden geprogrammeerd (ESS 508). De bits Y3 . . . Y8 zijn nul en spelen verder geen enkele rol. In principe zijn er 32 verschillende situaties mogelijk die leiden tot een keuze uit één van de drie toegestane combinaties van READ en WRITE. In werkelijkheid zijn het er veel minder. Afgezien van de opdeling in lezen en schrijven blijven er acht situaties over:

1. Schrijven of lezen van geheugen op de interface-kaart (IC2 . . . IC5).

- Acht databuffers staan in de door lezen of schrijven bepaalde richting ("enabled", databuffers vrijgegeven) (PROM-adressen 00 en 01).
2. Lezen van EPROM en lezen of schrijven van RAM of PIA op de basis-kaart. Databuffers geblokkeerd ("disabled") (PROM-adressen 04 en 05).
 3. Schrijven of lezen van de VIA. Zit op de interface-kaart, dus achter de buffers. De databuffers werken mee, in een richting die samenhangt met de vraag of het gaat om lezen of om schrijven (PROM-adressen 06 en 07).
 4. Schrijven of lezen van buskaartgeheugen (PROM-adressen 0C en 0D). Omdat, evenals in de gevallen 1, 2 en 3, E logisch nul is (WITH), gaat het om de situatie zonder buskaartgeheugen. De databuffers moeten in dit geval zijn geblokkeerd. De vektoren NMI, RES en IRQ worden automatisch uit de standaard-EPROM van de junior-computer gehaald.
 5. Zie 1 (PROM-adressen 10 en 11)
 6. Zie 2 (PROM-adressen 14 en 15)
 7. Zie 3 (PROM-adressen 16 en 17)
 8. Voor een deel zie 4 (PROM-adressen 1C en 1D). Omdat, evenals in de gevallen 5, 6 en 7, E logisch één is (WITH), gaat het om de situatie waarbij buskaartgeheugen is aangesloten. Ook het buskaartgeheugen zit achter de buffers, die dan ook open moeten staan voor dataverkeer in een bepaalde richting, inclusief het dataverkeer tijdens het ophalen van één van de drie vektoren op pagina FF.

Dat waren ze, alle acht. We zien dat de helft van de 32 bytes van de PROM IC17 echt nodig is (acht situaties maal lezen + schrijven). De andere 16 toestanden zijn onbelangrijk, omdat de overeenkomende combinaties van $D = 8K0$, $C = KX$ en $B = VIA$ domweg niet voorkomen. Uit tabel 2 blijkt dat in die 16 gevallen Y1 en Y2 logisch nul zijn. De databuffers staan dan op schrijven ingesteld en dat kan werkelijk nooit kwaad.

We zeggen nou wel dat er afgezien van lezen en schrijven acht situaties zijn, maar het zijn er eigenlijk maar vijf. In twee van de vijf gevallen zijn de databuffers geblokkeerd:

- a. Er wordt iets op de standaardkaart geadresseerd;
 - b. Er worden vektoren opgezocht in pagina FF, maar omdat er geen buskaartgeheugen is wordt alsnog verwezen naar de standaard-EPROM.
- In de andere drie gevallen is dataverkeer mogelijk via de databuffers;
- c. Er wordt geheugen op de interface-kaart geadresseerd;
 - d. De VIA, op de interface-kaart, wordt geadresseerd;
 - e. Er wordt buskaartgeheugen geadresseerd, bijvoorbeeld voor het opzoeken, door de 6502, van één van de drie vektoren.

Waarmee figuur 1 helemaal is besproken. Het venijn: dekodering en buffersturing, zat in de staart. Nu iets heel anders: figuur 2.

Cassette-interface

Bits op de band en bits van de band

Het grootste gedeelte van figuur 2 betreft elektronica ten behoeve van de cassette-interface: alles wat, afgezien van de software (TM; zie hoofdstuk 11) nodig is voor het zenden van data naar de cassette-recorder

(schrijven) of het lezen van data van de cassette-recorder. Tijdens het lezen van data wordt data van de band weergegeven, tijdens het schrijven van data op de band opgenomen.

De data gaat naar of komt van de 6502- μ P via poortlijn PB7 van de poort-konnektor. Tijdens het schrijven van data, dus tijdens de subroutine DUMP/DUMPT van het systeemprogramma TAPE MANAGEMENT, is PB7 uitgang, net zoals PB5 en PB6. Poort PB5 is dan logisch nul en PB6 logisch één. Dat heeft tot gevolg dat de ingang van N38 met pennummer 8 ook laag is en de uitgang van N38 dus hoog (N38 heeft één open-kollektor-uitgang, die hoogohmig is als-ie logisch één is). Omdat PB5 logisch nul is wordt via R15 de PNP-darlington T3 opengestuurd, met als gevolg dat de rode LED D5 (OUTPUT ON) oplicht en dat relais Re2 is bekrachtigd; de twee kontakten van uitgang J4 zijn met elkaar doorverbonden. Nemen we deze kontakten op in serie met de motor van de opnamerecorder OUTPUT (afstandsbediening), dan kan deze worden gestart via de software (de recorder moet dan wel gereed staan voor opname).

Omdat het tijdens het schrijven PB6 "1" is, is T2 gesperd, de groene LED D4 gedoofd, relais Re1 niet bekrachtigd en zijn de kontakten van J3 geopend. De uitgang van N39 voert ons via R20 en C14 naar P2, waarmee het maximale opnamenivo moet worden ingesteld. De data-uitgang is J2.

Tijdens het lezen van data van de band, dus tijdens de subroutine RDTAPE van TAPE MANAGEMENT, is PB7 als ingang geschakeld; ook nu zijn PB5 en PB6 uitgang, Poort PB5 is nu "1" en PB6 "0". Met als gevolg dat N38 het uitgangssignaal van IC7 geïnterverteerd aan PB7 kan doorgeven. Poort N39 blijft gewoon doorverbonden. Dat betekent dat er ook signalen terechtkomen op de data-uitgang J2, maar dat kan helemaal geen kwaad.

Omdat PB5 "1" is, is de rode LED D5 gedoofd en zijn de kontakten van J4 geopend. Daarentegen is PB6 nu nul, en dat heeft tot gevolg dat T2 geleidt, de groene LED D4 (INOUT ON) oplicht, relais Re1 bekrachtigd is en dat de kontakten van J3 (afstandsbesturing van de weergave-recorder INPUT) gesloten zijn.

Men is niet verplicht om gescheiden recorders te nemen voor het opbergen en ophalen van data. Bij het gebruik van één recorder voor beide taken vervalt het nut van de beide relais, omdat afstandsbesturing dat niet zo nodig moet. Het kan natuurlijk wel: een kwestie van doorverbinden van de kontakten van J3 en J4 (beide relais moeten aanwezig zijn!).

De sturing van T2 en T3 levert in ieder geval nuttige visuele informatie op over wat men eigenlijk aan het doen is. De groene LED licht op tijdens het lezen van data en de rode LED doet dat tijdens het schrijven van data. De kleurkeuze is logisch; denk aan de rode lampen in opname-studio's en elders, en aan groene verkeerslichten.

De PLL

De frekwentievolger als bitmaker

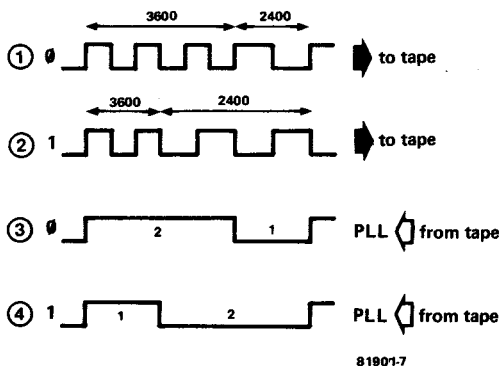
Het gedeelte van figuur 2 tussen de data-ingang J1 en de ingang van N38 met pennummer 9 is nu aan bespreking toe. De schakeling rond IC6 en IC7 dus. De KIM-kenners onder u zullen ongetwijfeld bij het zien van dit gedeelte van figuur 2 zeggen: "Hè, waar heb ik dat meer gezien". Klopt.

Omdat we van mening zijn dat het geen zin heeft om het wiel opnieuw uit te vinden. Wel zijn de hoekige kanten van het wiel afgerond. Anders gezegd: het gaat om verbeterde KIM-hardware (zie ook schakeling 31 uit de halfgeleidergids 1980).

Voor de bespreking van IC6, IC7 en omstreken is het voldoende om te weten dat de data op de band staat in de vorm van een reeks van een toontje met een frekwentie van ca. 3600 Hz, afgewisseld door eentje met een frekwentie van ca. 2400 Hz. Dus een tijdje 3600 Hz, daarna een tijdje 2400 Hz, daarna weer 3600 Hz, en daarna weer 2400 Hz, enzovoorts. Bij een bit hoort een 3600 Hz-tijdje en een dito 2400 Hz. Zie ook de plaatjes ① en ② van figuur 7.

Uit de 3600 Hz-tijdsduur ten opzichte van de 2400 Hz-tijdsduur blijkt dat het bit "1" dan wel "0" is. De schakeling rond IC6 en IC7 zorgt ervoor dat de uitgang van IC7 "1" is als er 3600 Hz op de ingang J1 staat en "0" als er sprake is van 2400 Hz. De software van RDTAPE zorgt ervoor dat uit de twee tijdsduren van een 3600 Hz/2400 Hz-cyclus een bit "0" of "1" wordt gedestilleerd.

De schakeling van IC6 vormt samen met de omringende componenten een zogenaamde PLL (Phase Locked Loop: fasevergrendelde regellus). Dat klinkt erg "duur" en het is mogelijk om een heel boek te vullen met formules over dat ding. Neem ons niet kwalijk als we dat niet doen en in plaats daarvan de werking uitleggen in een paar zinnen. Zie de PLL, met name in de toepassing op de interface-kaart als een "frekwentievolver". Net zoals bij een emittervolger de uitgangsspanning de ingangsspanning volgt past een interne oscillator zich in de frekwentie aan aan de frekwentie van het ingangssignaal, binnen een zeker frekwentiebereik en boven een minimum-nivo van het ingangssignaal. Die interne oscillator heeft een frekwentie die afhangt van een stuurspanning (VCO). Zonder ingangs-



81901-7

Figuur 7. Elk naar de cassette-band te zenden bit bestaat uit een stukje hoge frekwentie, gevolgd door een stukje lage frekwentie. Hoe lang elk stukje duurt hangt ervan af of het bit logisch nul of één is. Door de laagdoorlaatfilterwerking van de opname/weergaveketen van de band zullen de teruggespeelde spanningen er niet meer zo rechthoekig uitzien. De positie van de nuldoorgangen blijft echter gehandhaafd en is voor de PLL geschikt grondstof voor het halffabrikaat (plaatje ③ of ④), waaruit de software van het programma TM het eindprodukt 0 of 1 destilleert.

signaal is de frekventie ca. 3000 Hz, dus precies tussen 2400 Hz en 3600 Hz. Staat er 3600 Hz op de ingang, dan gaat de frekventie met 600 Hz omhoog; bij 2400 Hz op de ingang met 600 Hz omlaag. Voor een verlaging van de frekventie is een verhoging van de stuurspanning nodig en voor een frekventieverhoging een verlaging van de stuurspanning. De hoogte van de stuurspanning hangt dus samen met de frekventie van het ingangssignaal en biedt na vergelijking met een vaste spanning de mogelijkheid om logisch onderscheid te maken tussen frekquenties: "1" voor de ene frekventie, "0" voor de andere, geheel in overeenstemming met het principe van FSK (Frequency Shift Keying; zeg maar: frekventieverschuivingslogica).

De PLL wordt uit +12 V gevoed via de parallelschakeling van D2 + D3 en C8. Dit wat merkwaardige netwerk zorgt ervoor dat de PLL via pen 10 behoorlijk verschoond blijft van allerlei stoorspanningen op de voeding, die de werking van de PLL zouden kunnen verstoren. De "schone 12 V" (ca. 11 V vanwege de dioden) wordt eveneens gebruikt voor de gelijkspanningsinstelling van de ingangen (pennen 2 en 3 van IC6) via R21 . . . R24. Bij de KIM wordt daarvoor een 5 V-voedingsspanning gebruikt die vergeven is van stoorspanningen die regelrecht de (weliswaar gebalanceerde) ingangen ingaan. En dat terwijl er daar óók zo'n "schone 12 V" voorhanden is! Verder is er hier, anders dan bij de KIM, geen sprake van een ca. tienvoudige verzwakking van het ingangssignaal voordat pen 2 is bereikt.

Alle genoemde verbeteringen hebben tot gevolg dat men van goeden huize moet komen om nu ingangsdata de mist in te laten gaan (bijvoorbeeld door "dropouts" op de band, een vuile of niet goed afgestelde kop, enzovoorts). Het is ons al een paar keer overkomen dat *achteraf* bleek dat er data werd ingelezen, terwijl een massaverbinding met de recorder ontbrak!

Van de ingangen 2 en 3 (een soort verschilversterking) wordt alleen ingang 2 gebruikt (asymmetrische sturing). De koppeling met J1 gaat via C1, die een veel kleinere waarde heeft dan de KIM teneinde veel meer storing beneden 2400 Hz buiten de deur te houden. Weerstand R37 is nodig in het geval dat gebruik wordt gemaakt van de luidsprekeruitgang (of koptelefoonuitgang) van een cassette-recorder, waarbij de luidspreker automatisch wordt afgekoppeld.

Van invloed op de VCO-frekventie zonder ingangssignaal zijn C3, R25 en P1. De methode, volgens welke P1 wordt ingesteld bepaalt in grote mate de kwaliteit en betrouwbaarheid waarmee data wordt gelezen van de band. Er zullen verschillende methoden worden gegeven voor de juiste instelling/afregeling van P1.

Punt 7 van IC6 is de uitgang van de PLL. Deze voert de al eerder behandelde regelspanning die ervoor zorgt dat de VCO-frekventie in de pas loopt met het ingangssignaal. Dit punt is via C9 verbonden met +12 V. Deze condensator vormt samen met een weerstand van 3K6 in het IC het zogenaamde loopfilter (lusfilter). Dit filter is van groot belang voor het goed, niet al te "emotioneel" (overshoot) reageren van de PLL op frekventieveranderingen aan de ingang. De uitgang van de PLL gaat via het ladderfilter R26 . . . R28 & C5 . . . C7 naar de min-ingang van de komparator (= vergelijker) IC7. De plus-ingang hiervan ligt via R29 aan een door IC6 geleverde vaste gelijkspanning (op pen 6).

De dimensionering van dat ladderfilter hangt samen met de snelheid

waarmee de 3,6 kHz- en 2,4 kHz-frekwenties elkaar opvolgen. En dat hangt op zijn beurt weer samen met de snelheid waarmee de bits op de band zijn geschreven, en van de band moeten worden gelezen. Dat is de zogenaamde baudrate, het aantal verzonden of ontvangen bits per seconde: 800 (baud) voor de junior-cassette-hardware *en* -software.

We herinneren ons nog wel dat een verhoging van de VCO-frekwentie tot 3600 Hz een verlaging van de uitgangsspanning op punt 7 tot gevolg had (in een gesloten regellus zijn oorzaak en gevolg moeilijk van elkaar te onderscheiden), en een verlaging naar 2400 Hz een verhoging van die spanning. Er van uitgaand dat de gefilterde uitgangsspanning op de min-ingang van IC7 van lager resp. hoger is dan de gelijkspanning op de plus-ingang van IC7, krijgen we de situatie dat de uitgang van IC7 "hoog" is (+5 V, via R31) als de ingangsfrekwentie 3600 Hz is, en "laag" (praktisch op aardpotentiaal) als de ingangsfrekwentie 2400 Hz is. Dat is precies wat we nodig hebben. De uitgangsspanning van IC7 wordt via N38 geïnverteerd (hoog wordt laag, en omgekeerd) en op PB7 gezet. Zie ook de plaatjes ③ en ④ van figuur 7.

N.B. De PLL reageert niet oneindig snel op veranderingen van de ingangsfrekwentie. Dat heeft tot gevolg dat de uitgang van IC7 niet in één keer omklapt van "1" naar "0" of van "0" naar "1" tijdens de overgang van 3600 Hz naar 2400 Hz respectievelijk van 2400 Hz naar 3600 Hz. Nee, de uitgang fietst een aantal keren heen en weer tussen het ene en het andere nivo alvorens definitief het eindnivo aan te nemen: de zogenaamde PLL-jitter, die erg veel lijkt op kontaktdender van (bijvoorbeeld) toetsen. De software van de subroutine RDTAPE (zie hoofdstuk 11 en boek 4) weet echter wel raad met dat tijdelijke wispelturige gedrag.

Randapparatuur aansluiten

RS 232 - interface

De ter bespreking resterende schakelingen van figuur 2 (linksboven) stellen weinig voor in vergelijking tot de enorme hoeveelheid "moeilijke" elektronica, inclusief science fiction-achtige kretten als "UART", die zit in de randapparatuur, die via de voornoemde simpele schakelingen op de junior-computer kan worden aangesloten.

Het gaat om een data-zender en een data-ontvanger. Poort PA7 doet dienst als ingang, PB0 als uitgang. De ontvanger is T1 met wat spul eromheen. Het is belangrijk om vast te stellen dat T1 enen op de ingang (bovenkant R5) omdraait in nullen, en omgekeerd . . .

Nemen we aan dat ingang 6 van N37 aan +5 V ligt via R8 en vergeten we even wat er via de draadbrug P-Q gebeurt, dan inverteert N37 het ingangsnivo PB0. De uitgang van N37 stuurt via R10 de vergelijk IC8. Er wordt vergeleken met de spanning op het knooppunt R11/R12/C12. Is de uitgang van N37 "1", dan is de uitgang van IC8 ca. +12 V; in het geval "0" ca. -12 V. Ook nu stellen we vast dat er tussen PB0 en de uitgang van IC8 — los van nivo-aanpassing — sprake is van omkering van logische nivo's.

Data-ingang (via R5) en -uitgang (IC8) zijn aangesloten op een genormaliseerde 25-polige D-konnektor, ook wel RS232-konnektor genoemd. Dat "RS232" is niet zo maar een kreet, maar een norm, afspraak voor seriële

datacommunicatie: bit voor bit. Elk byte van acht bits wordt voorafgegaan door een startbit, ten teken dat begonnen is met het verzenden van een byte. Bytes zijn gekodeerd volgens de ASCII-kode.

De RS232-norm legt de twee mogelijke logische nivo's vast van een datacommunicatieleiding, inclusief de bijbehorende spanningsnivo's. Later is daar de pinning van de D-konnektor bijgekomen. Voor een logische nul geldt een lijnnivo tussen +5 V en +15 V (versie RS232C), voor een logische één een lijnnivo tussen -5 V en -15 V. Bij "ons" is het circa +12 V respectievelijk -12 V. U ziet dat bij een "1" een lage spanning hoort en bij een "0" een hoge; dat noemen ze negatieve logica. Door nu zowel na het ontvangen als voor het zenden data (bits) te inverteren hoeft er in de computer niet met "gespiegelde" data te worden gewerkt.

De D-konnektor heeft een aantal pennen die onderling zijn doorverbonden. Andere doorverbindingen zijn mogelijk, afhankelijk van de soort aan te sluiten randapparatuur. Wat is er zo al concreet aan te sluiten op de junior-computer via de RS232-konnektor? We denken met name aan de Elekterminal, een video-terminal, inclusief ASCII-toetsenbord (met de toetsen van een tiepmachine *plus* een aantal speciale toetsen), die is ontworpen in het kader van de SC/MP-computer, maar eveneens uitermate geschikt is voor de junior-computer.

Het systeemprogramma PRINTER MONITOR is gebaseerd op het ASCII-toetsenbord als invoer (diverse toetskommando's) en de eigenlijke Elekterminal of een geschikte printer (*niet*, althans voorlopig niet, de eerder door Elektuur gepubliceerde metaalfolie-printer) als uitvoer (display).

Met de bespreking van figuur 2 is de bespreking van alles wat op de interface-kaart zit afgerond. Voordat we de uitgebreide junior-computer gaan bouwen moeten we eerst nog wat vertellen over de basiskaart en de voeding. Want ook de voeding wordt uitgebreid. En, afhankelijk van hoe men gaat uitbreiden, zijn er wijzigingen nodig op de basiskaart: andere weerstandswaarden, misschien een andere draadbrug en misschien een stukje elektronica op postzegelformaat erbij.

Basiskaart en voeding aangepast

Konstruktief ingrijpen in en op bestaande kaarten

Zoals er bij een menselijke operatie bloed vloeit in het kader van het beter worden, zo vloeit er soldeertin op de basiskaart en op de bestaande voedingsprint in het kader van de operatie: groter worden van de junior-computer. Je kunt natuurlijk stellen dat het toch bekend was dat er uitbreidingen komen en dat wij daar rekening mee hadden moeten houden, maar het is helemaal geen nakaartkwestie à la: gebeurd is gebeurd. Het betreft namelijk wijzigingen die uitsluitend nodig zijn bij uitbreidingen en niet als men het op en bij de standaard-junior-computer houdt (zoals in het voedingsgeval). Verder zijn bepaalde wijzigingen afhankelijk van de keuze van gebruiker: wát voor uitbreidingen? Vergelijk het met de situatie dat men een huis koopt en plannen heeft voor de bouw van een dakkapel —

ooit. Het heeft dan weinig zin om dan maar alvast een gat in het dak te maken. Afijn, u weet wel wat we bedoelen.

De **wijzigingen op de basisprint** bestaan zoals zo veel goede zaken uit drie (onderdelen):

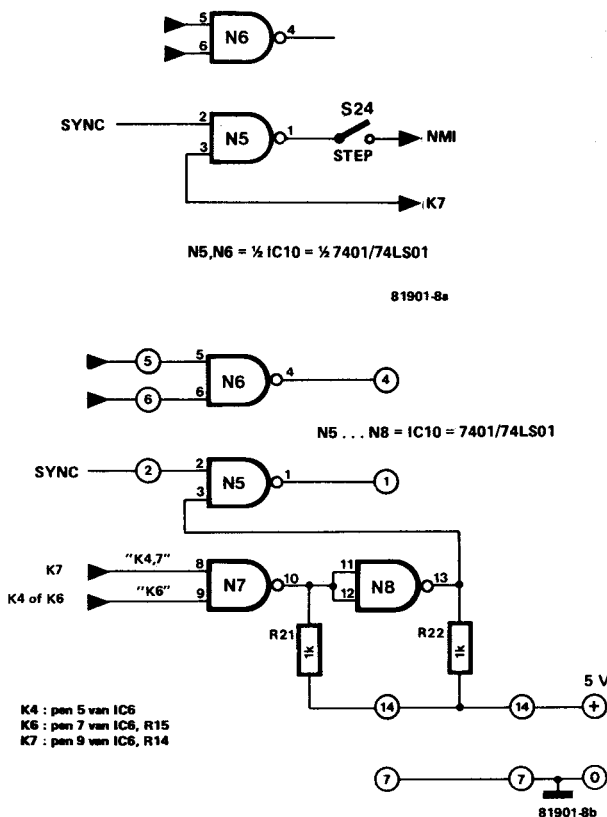
1. **De wijziging van een paar weerstanden.** We hebben gezien dat de adreslijnen en de datalijnen gebufferd zijn alvorens ze de uitbreidingen dienen. Alle andere leidingen (stuurbus) zijn dat niet. Met uitzondering van het signaal RAM-R/W is dat ook niet nodig. Dit signaal kan alternatief worden gebufferd door verlaging van de pullupweerstand R5 op de basiskaart tot 470Q. Door de lagere weerstand reageert dit signaal sneller op, hem opgelegde wijzigingen in logisch nivo. Overigens: de pullupweerstand R14, R15 en R16 varen eveneens wel bij een dergelijke weerstandsoperatie (snellere selektiesignalen K0, K6 en K7).
2. **Het omleggen van de draadbrug rond de adresdekodeer IC6 van massa naar EX.** Zoals we hebben kunnen zien is dat alleen noodzakelijk als er buskaartgeheugen wordt aangesloten. Dus D van IC6 blijft aan massa zitten als uitsluitend de interface-kaart erbij komt.
3. **Uitbreiding van de elektronica rond poort N5, die nu in meer dan één geval het stap voor stap doorlopen van een programma blokkeert.** Figuur 8a geeft de oude situatie, figuur 8b de nieuwe en figuur 9 een paar van die "logische heen-en-weer-plaatjes" (pulsdiagrammen), die te maken hebben met de stofwisseling binnen de 6502.

Zoals men weet uit het instructie-overzicht achterin boek 1 vergt de uitvoering van elke instructie een bepaalde hoeveelheid tijd in de vorm van een geheel aantal klokperiodes. Gedurende één zo'n klokperiode wordt de opcode van een nieuwe uit te voeren instructie uit het geheugen opgehaald, in de laatste uitvoeringsfase van de vorige instructie. Dit data ophalen gaat gepaard met het logisch één worden van het signaal R/W (plaatje ② van figuur 9; er moet immers een opcode worden gelezen), en met de opwekking van een SYNC-puls (plaatje ③). In de standaardsituatie (figuur 8a) zorgt zo'n SYNC-puls via poort N5 voor een interrupt (NMI) **tenzij het signaal K7 logisch nul wordt**. Dus er ontstaat een NMI zolang de standaard-EPROM niet is geadresseerd. Treedt een NMI op, dan wordt de lopende instructie afgehandeld (dus die waarvan de opcode werd opgehaald).

Gaan we ervan uit dat de NMI-sprongvektor in het geval stap voor stap op de monitor is gericht (1C00), dan is het stap voor stap, dus instructie voor instructie doorlopen van een programma geblokkeerd, zodra het om monitor-instructies gaat. Dat is logisch, want de werking van de monitor staat en valt met het **ononderbroken** doorlopen van hele reeksen instructies. Denk aan het wachten op een nieuwe ingedrukte toets en aan het multiplexen van de zes displays.

Welnu, voor het programma Printer-Monitor van hoofdstuk 12 is een soortgelijke blokkade nodig, evenals voor met de hand opgegeven programma's in PIA-RAM; dat laatste zowel in combinatie met de standaard-monitor (zie Aanhangsel 2) als met PM. Het programma PM wordt geselecteerd via K4 (die dan met K5 is doorverbonden), de PIA-RAM via K6. Vandaar de schakeling van figuur 8b, waarin geen NMI optreedt tijdens een SYNC-puls indien K7 nul is en indien K4 of K6 nul is, afhankelijk van de vraag welke van de twee is aangesloten.

Op de twee weerstanden na is de elektronica van figuur 8b in feite al

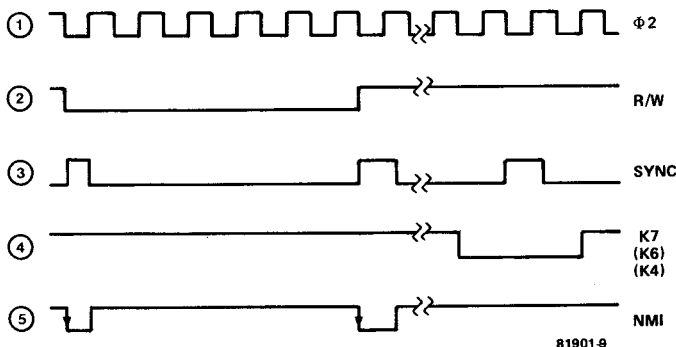


Figuur 8. Zo was het (8a) en zo wordt het (8b): Aanvullende elektronica op de basiskaart, nodig om in meer dan één geval de mogelijkheid tot het stap voor stap, dus instructie voor instructie doorlopen van een programma te blokkeren. Dat is van toepassing bij het gebruik van het programma PM of van de standaard-monitor (zie Aanhangsel 2).

aanwezig op de standaardkaart; slechts twee van de vier poorten van IC6 waren tot nu toe in gebruik. Dat wordt nu wel even anders. Voor deze uitbreiding is een piepklein printje nodig.

Voor alle zekerheid zeggen we het nog maar eens: deze uitbreiding is uitsluitend nodig indien men het programma PM gaat gebruiken, en dan nog alleen indien men prijs stelt op de stap-voor-stap-optie van PM. Een andere reden voor deze uitbreiding is het decimaal rekenen met de standaard-monitor (zie Aanhangsel 2).

Opgelet (1)! Er is één situatie waarbij er drie K-signalen moeten worden aangesloten op de schakeling van figuur 8b: K7, K4 en K6. Dat is het geval indien met stap voor stap een gebruikersprogramma wil doorlopen onder supervisie van PM, waarbij er in het gebruikersprogramma decimaal wordt



Figuur 9. Tijdens de fase van de uitvoering van een instructie waarin de opcode wordt opgehaald uit het geheugen, ontstaat een SYNC-puls. Deze kan in sommige gevallen een interrupt (NMI) uitlokken. Als dat gebeurt wordt altijd de lopende instructie (dus die waarvan de opcode werd opgehaald) volledig uitgevoerd.

gerekend. Voor nadere bijzonderheden verwijzen we naar figuur 17 van hoofdstuk 12 en de bespreking daarvan in de tekst.

Opgelet (2)! Het signaal K7 moet **altijd** op de schakeling van figuur 8b zijn aangesloten, óók als men van de diensten van PM gebruik maakt in plaats van de standaard-monitor, en óók als de vektoren IRQ, RES en NMI uit buskaartgeheugen worden opgehaald! Waarom? Omdat de afwikkeling van een IRQ (vaak via de instructie BRK) of een NMI (stap voor stap!) niet mag worden geblokkeerd! De stap van IRQ-vektor naar IRQ-sprongvektor en de stap van NMI-vektor naar NMI-sprongvektor vereisen namelijk de uitvoering van een instructie JMP-IND. Zie ook figuur 32 van hoofdstuk 3 van boek 1.

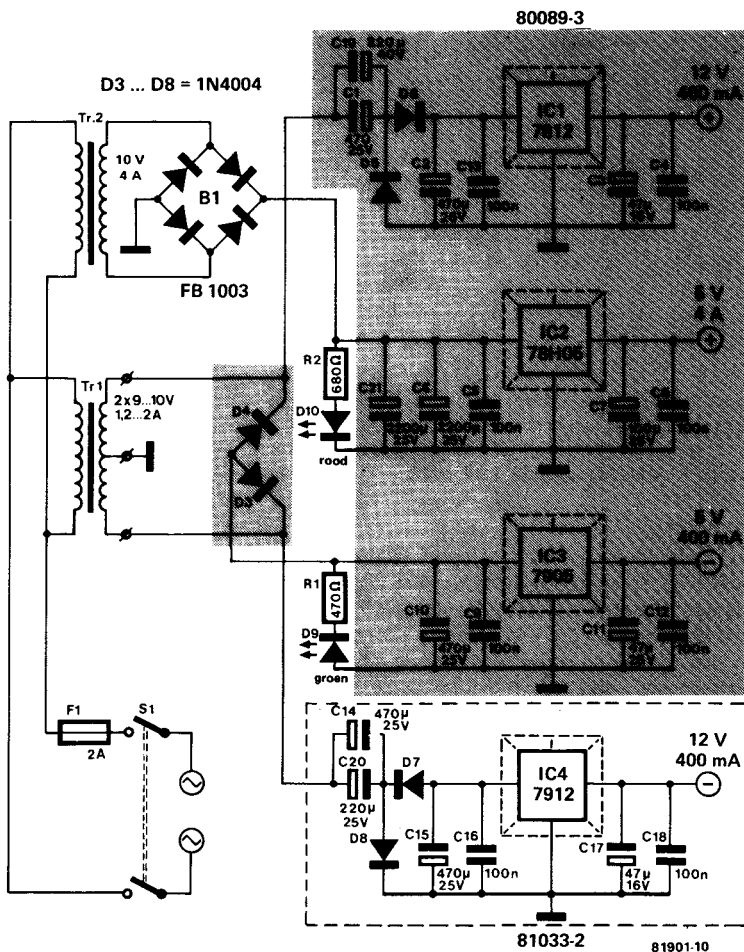
Uitbreiding voeding

Meer en grotere monden voeden

Het zal duidelijk zijn dat, naarmate er meer monden, kaarten dus, te voeden zijn, er méér moet worden voorgeschoteld. Vooral EPROMs zijn dorstige lieden en doen een stevig beroep op de elektronenkraan. Daarnaast is er nu ook min 12 volt nodig voor de RS232-interface (zie figuur 2). In figuur 10 is het schema te zien van de nieuwe voeding. Het gaat om een mengsel van "oud", "nieuw" en "gewijzigd". Deze voeding levert:

- +5 volt, maximaal 4A (was 1A)
- −5 volt, maximaal 400 mA (was 100 mA)
- +12 volt, maximaal 400 mA (was 100 mA)
- −12 volt, maximaal 400 mA (nieuw)

Dat is allemaal ruim voldoende voor de combinatie van basiskaart, interface-kaart en vijf geheugenkaarten. In principe krijgt of heeft alle, op de uitgebreide junior-computer aangesloten randapparatuur, zoals de Elekterminal, een eigen voeding. De Elekterminal kan echter eventueel evenéens worden gevoed uit de uitgebreide voeding (zie hoofdstuk 12).



Figuur 10. Het schema van de uitgebreide voeding voor de uitgebreide junior-computer.

De papieren uitbreidingen van de junior-computer zijn nu allemaal besproken. Dat wil zeggen: op de tot het buskaartgeheugen behorende geheugenkaarten na. De RAM/EPROM-geheugenkaart komt in het derde en laatste deel van dit hoofdstuk aan de orde. Maar nu eerst iets anders: de praktische realisatie van de uitgebreide junior-computer. De bouw dus! Oude en nieuwe printen vol- of verbouwen, de bedrading, de konnektors, de (nieuwe?) kast, een cursus "ligt hem aan hem of aan mij" (foutzoeken) – u krijgt het allemaal voorgeschoteld. Nu!

De bouw van de uitgebreide junior-computer

Nu niet alleen de hersens, maar ook de handen erbij!

Nu breekt een fase aan, waarin er niet alleen "voor u is gedacht" (bijvoorbeeld het lezen van een beschrijving van hoe iets werkt), maar waarin door u ook moet worden meegedacht en meegedaan. Tenminste: als u besluit tot de bouw van de uitbreidingen. In andere gevallen betekent dit gedeelte van dit hoofdstuk voor u wellicht een oriëntatie-tocht, die best wel eens in een ontdekkingsreis kon uitmonden. De ontdekking namelijk dat het ook wat de te bouwen uitbreidingen betreft wel meevalt, en dat dat in elk geval geen reden is om niet tot uitbreiding te besluiten.

Een groot aantal bouwtips van hoofdstuk 1 van boek 1 is ook nu van toepassing. Lees het betreffende gedeelte nog maar eens opnieuw, dat kan nooit kwaad. In aanvulling daarop nog wat algemene opmerkingen.

Het gebruik van een klein soldeerboutje betekent eveneens dat dun soldeertin moet worden gebruikt ($\frac{1}{2}$ à 1 mm)! Het gebruik van tin voor de reparatie van dakgoten e.d. is zinloos. De ervaring wijst uit dat dit advies nodig is.

Ga uiterst voorzichtig om met de print. De printsporen kunnen kapot gaan door oververhitting, maar ook door vormen van schijnbaar ongevaarlijke mechanische belasting, zoals het onder druk verschuiven over een tafel of werkblad, bezaaid met stukjes draad, tin enzovoorts. Behoed uw kostbare printen hiervoor! Beschikt u niet over een bankschroefje voor de montage van een print, monteer dan tijdelijk vier boutjes M3X40 in de hoeken, zodat afstand bewaard wordt met het werkblad dat men zo veel mogelijk schoon dient te houden.

Let er, ook zonder gebruik van IC-voeten, bij de montage van de IC's op dat *alle* 14, 16 of hoeveel het er ook mogen zijn pennen zijn aangesloten! Als men niet goed oplet buigt een IC-pen onder het IC-lichaam en op het eerste gezicht lijkt alles in orde. Op het tweede gezicht niet, en dat is uiterst vervelend.

Er is inmiddels een buitengewoon elegant alternatief voorhanden voor het "doorfluiten" van verbindingen met een beltrafo (figuur 14 van hoofdstuk 1). Dat is de "verbindingstester", gepubliceerd in de juni-uitgave 1981 van het tijdschrift *Elektuur*.

Gebruik **soepel** draad voor alle draadverbindingen en draadbruggen. Soepel draad bestaat uit een groot aantal dunnere koperdraadjes. Een draaddoorsnede van ca 0,3 mm² is voldoende voor de meeste verbindingen. **Voor de verbindingen 0 (massa) en +5 V naar de voeding moet dik soepel draad worden gebruikt: minstens 1 mm²!**

Nogal vaak, zo blijkt uit ondermeer technische vragen, wordt als iets niet (meteen) werkt overgegaan tot de aanschaf van één of meerdere nieuwe IC's. In 99 van de 100 gevallen is dat volkomen overbodig! Op basis van nieuwe IC's, afkomstig uit de bonafide handel kan men vaststellen dat die IC's gewoon goed zijn. Beter is het om zijn energie te steken in de controle op juiste en onjuiste (kortsluitingen) verbindingen. Een reservevoorraad IC's is prachtig voor de handel, hoe "fide" ook, maar is voor het overige volkomen overbodig.

Het menu van de bouw van de uitbreidingen: Achtereenvolgens komen aan de orde:

1. de aanpassing van de basiskaart;
2. de bouw van de uitgebreide en gewijzigde voeding;
3. het volbouwen van de interface-kaart;
4. het met elkaar verbinden van de basiskaart, de interface-kaart en eventueel de buskaart;
5. de behuizing van de uitgebreide junior-computer;
6. enige tips voor het foutzoeken.

De punten komen in de genoemde volgorde aan de orde omdat je nu eenmaal niet alles tegelijk kunt behandelen. Het zal bijvoorbeeld duidelijk zijn dat punt 5 invloed heeft op de punten 4 en, voor een deel, 2.

Aktiepunt één: de basiskaart

Kleinere weerstanden en een klein printje

Neem de basiskaart (= hoofdprint), haal de voedingsaansluitingen los (dat moet toch gebeuren) en bekijk tabel 3. Er worden één of vier weerstanden in waarde gewijzigd en, indien nodig, een draadbrug verlegd. U kunt de oude weerstand(en) vervangen door (een) nieuwe of er een 560 ohm-weerstand aan parallel solderen. Dat laatste kan op de, vanuit de bestaande

Tabel 3.

Wijzigingen hoofdprint JC:

R5 = 470 Ω (of 560 Ω parallel aan "oude R5) (absoluut nodig)

R14, R15, R16 = 470 Ω (of 560 Ω parallel aan "oude" R) (mag)

1 draadbrug D-EX (uitsluitend indien buskaartgeheugen is aangesloten; dus zonder buskaartgeheugen blijft D met massa doorverbonden)

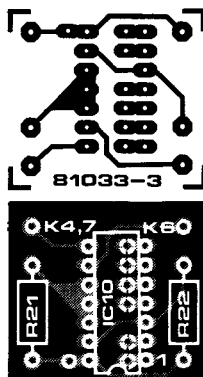
weerstand beredeneerde, andere kant van de print gebeuren en heeft als voordeel dat het gemakkelijker gaat. In het andere geval moet namelijk eerst de oude weerstand worden verwijderd (aan beide kanten doorknippen en met een tang de draadeindjes verwijderen; wél even de soldeerpunten verhitten met de soldeerbout!). Voor de "junior-junioren" onder u even wat kleurcodes:

470 ohm: geel-violet-bruin-(goud)

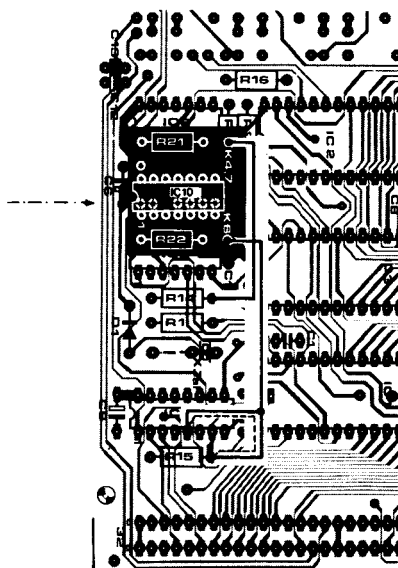
560 ohm: groen-blauw-bruin-(goud)

Gaat u het programma Printer Monitor niet gebruiken, of ziet u af van PM zijn stap-voor-stap-mogelijkheden (wat jammer zou zijn) of wilt u niet decimaal kunnen werken met de standaard-monitor (zie Aanhangsel 2), dan kunt u nu doorgaan naar het volgende aktiepunt. Met de naar we aan-nemen nauwelijks gedunde gelederen lezen we nu verder over de **opsteek-print**:

- **schema** → figuur 8b; figuur 17 van hoofdstuk 12
- **print** → figuur 11
- **bouwtekening** → figuur 12
- **onderdelen** → tabel 4



Figuur 11. De opsteekprint met daarop de schakeling van figuur 8b. De met de basisprint door te verbinden pennen van IC10, zeven stuks, zijn elk aangegeven met een in vieren gedeeld cirkeltje. Deze verbindingen gaan uit van de koperzijde van de opsteekprint. De aansluitingen "K4,7" en "K6" op de print dient men zó op te vatten: K7 wordt altijd aangesloten. De tweede aansluiting is K4 of K6. Voor een eventuele derde aansluiting wordt verwezen naar figuur 17 van hoofdstuk 12.



81901-12

Figuur 12. De aansluiting van de opsteekprint op de basiskaart.

Tabel 4.

Opsteekprint EPS 81033-3

R21, R22 = 1 k (doorlopende nummering
hoofdprint JC): bruin-zwart-rood-(goud)
IC10 = 7401, 74LS01 (uitsluitend indien het
oorspronkelijke exemplaar moet worden
uitgesoldeerd)
1 print EPS 81033-3
1 IC-voet 14-pens (tenzij IC10 al in voet zat)
1 verbindings-IC-voet (zie tekst)
2 soldeerpennen

Er wordt gebruik gemaakt van twee, tot nu toe niet gebruikte poorten van IC10. Twee deden al dienst, als N5 en N6; de andere twee zijn nu ook de klos en gaan voortaan als N7 en N8 door het leven. Om nu met maximaal gemak de extra elektronica op de basiskaart te herbergen (de noodzakelijke signalen SYNC en K7 zitten niet op de uitbreidingskonnektor, dus de interface-kaart is hiervoor onbruikbaar) is gekozen voor een opsteekprint met nieuwe IC10, in de plaats van de oude IC10. Eigenlijk is "imperial-print" beter op zijn plaats omdat de opsteekprint evenwijdig staat aan de basiskaart.

Als IC10 al in een voet zat is de operatie vrij simpel; anders moet eerst een voet worden aangebracht, onder opoffering van het oude exemplaar: pen voor pen doorknippen en elke pen met een tang onder verhitting met de soldeerbout uittrekken. Men kan vervolgens de soldeergaten weer open maken met behulp van desoldeer-litze (zuigt soldeertin op) of van een desoldeerbout. Vervolgens een 14-pens IC-voet monteren.

De opsteekprint biedt plaats aan IC10, twee weerstanden en twee soldeerpennen voor de K-aansluitingen. Verder aan de koperzijde de punten ①, ②, ④ . . . ⑦ en 14. Deze dienen als aanknopingspunt voor de verbindingen met de IC10-voet op de basis-kaart en zijn op de componenten-opdruk van de opsteekprint aangegeven met zeven "doorkruiste" cirkels. Neem voor deze verbindingen zeven stukjes draad of een open 14-pens IC-voet, waarvan de niet-gebruikte aansluitingen 3 en 8 . . . 13 zijn afgeknipt (dát zijn namelijk uitsluitend op de opsteekprint gebruikte IC-pennen).

Het is natuurlijk óók mogelijk om de opsteekprint via zeven draadjes direkt aan te sluiten op de basiskaart, dus zonder tussenkomst van IC-voet(en). Keuze genoeg dus.

Ja, en dan nog de twee K-verbindingen. De aansluiting K7 moet altijd worden gemaakt. In sommige gevallen biedt een weerstand een aanknopingspunt (figuur 12). In het geval PM is K4 de tweede aansluiting. Zonder PM kan de opsteekprint weggelaten worden, of worden gebruikt om stap voor stap decimaal te kunnen rekenen met de standaard-monitor; zie Aanhangsel 2. In dat geval is K6 de tweede aansluiting. Dan is de stap-voor-stap-mogelijkheid niet alleen voor de monitor maar ook voor programma's in PIA-RAM geblokkeerd.

P.S. Met enige handigheid (en voorzichtigheid! denk aan kortsluiting) is er

een alternatief aanknopingspunt voor K4: pen 17a van de uitbreidingskonnektor. Is vooral handiger als IC6 op de basiskaart in een voet zit.

Let op! De opschriften "K4,7" en "K6" op de opsteekprint (figuren 11 en 12) en in figuur 8b kunnen wellicht zorgen voor enige verwarring. Het is niet: Soms K4, soms K7 en altijd K6, maar: Altijd K7 en, afhankelijk van wat u wilt, K4 of K6 als tweede aansluiting.

En verder nog dit: In sommige gevallen is een derde K-aansluiting op de opsteekprint nodig. Namelijk indien men met PM een gebruikersprogramma, waarin decimaal wordt gerekend, stap voor stap wil doorlopen, danwel gebruik wil maken van één of meer BRK-instructies. Dit is besproken in hoofdstuk 12. Nu u toch aan het bouwen bent is het geen gek idee om niet alleen de werkzaamheden volgens figuur 8b van dit hoofdstuk uit te voeren, maar ook die volgens figuur 17 van hoofdstuk 12. U kunt dan stap voor stap werken met de standaard-monitor en met PM, en dat zowel binair als decimaal. Kortom: u heeft dan alle mogelijkheden in huis.

Aktiepunt twee: voeding uitbreiden

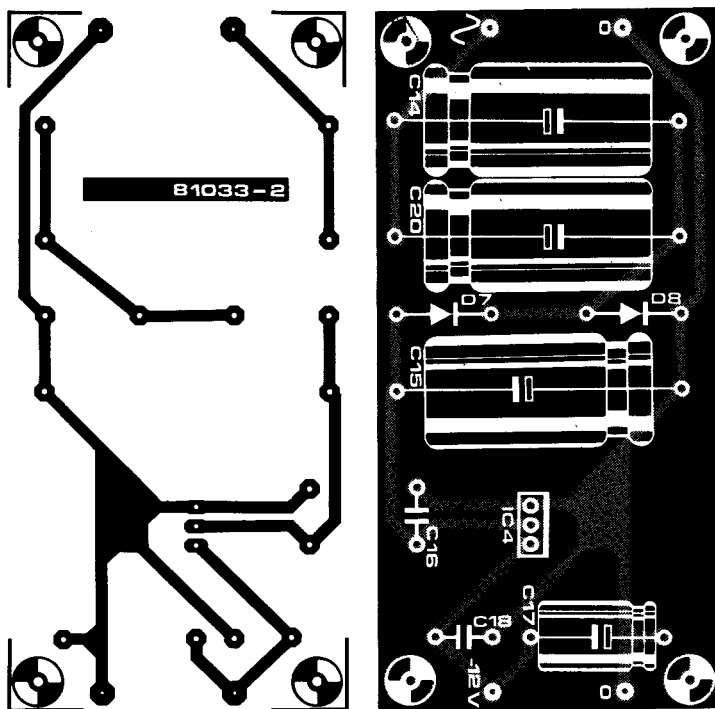
Meer (milli)amperes en min 12 volt

- schema → figuur 10
- print -12 V → figuur 13
- bestaande voedingsprint → *Junior Computer 1*, p. 30 ... 32
- onderdelen → tabel 5
- bouwtekening → figuur 14

Beginnen we maar eens met de bestaande voedingsprint. Dioden D1 en D2 worden verwijderd en kunnen nu dienst doen als D7 en D8. Verder moeten de bestaande IC1, IC2 en IC3 en de vingerkoelplaat eraf. Na het afbrekend nu het opbouwend werk. Parallel aan en bovenop C1 monteert men C19. Let op de juiste polariteit. Het is ook mogelijk om C1 te vervangen door een elko van $680 \mu/40 \text{ V}$; C19 vervalt dan. Parallel aan en bovenop C6 komt C21. Ook hier mag $C6 = 4700 \mu$ worden gekozen, waarbij C21 vervalt.

De nieuwe IC1 en IC3 zien er niet zo transistor-achtig meer uit als de oude; ze hebben een koelplaatje, dat in beide gevallen aan de kant van C2 komt te zitten! Elk koelplaatje wordt gemonteerd op een, vertikaal op te stellen losse koelplaat (zie figuur 14). Het metaal van IC1 en IC3 is verbonden met de middelste pen. Bekijkt men de aansluitingen van de IC's, dan ziet men dat het om verschillende spanningen gaat. Indien IC1 en IC3 zonder elektrische isolatie op de bijbehorende koelplaat zijn gemonteerd **mogen de koelplaten van IC1 en IC3 elkaar niet raken!** Verbuig daarom de aansluitdraden van één van de IC's of desnoods beide zodanig dat, onder handhaving van de verticale stand (schoorsteeneffekt: een goede warmteafvoer), de koelplaten elkaar niet raken. Monteer de koelplaten met de koelvinnen in onderling tegengestelde richtingen.

De nieuwe IC2 ziet er net zo uit als de oude en heeft dezelfde aansluitingen. Omdat er als gevolg van de sterk vergrote stroomafname een grotere warmteontwikkeling in IC2 kan ontstaan moet de bestaande vingerkoelplaat worden vervangen door een TO3-voorgeboorde koelplaat die met de vinnen vertikaal moet worden opgesteld, bijvoorbeeld tegen de achterkant van de kast. Laat enige ruimte tussen de onderkant van de



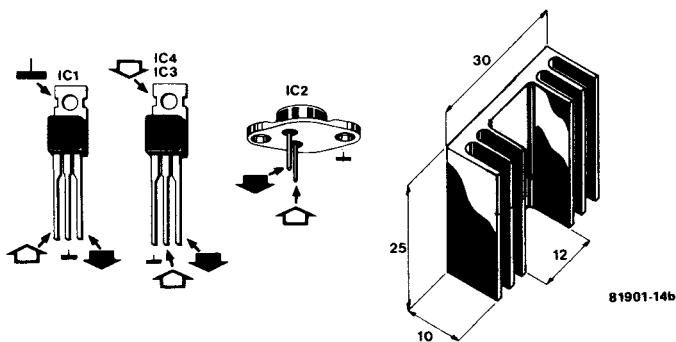
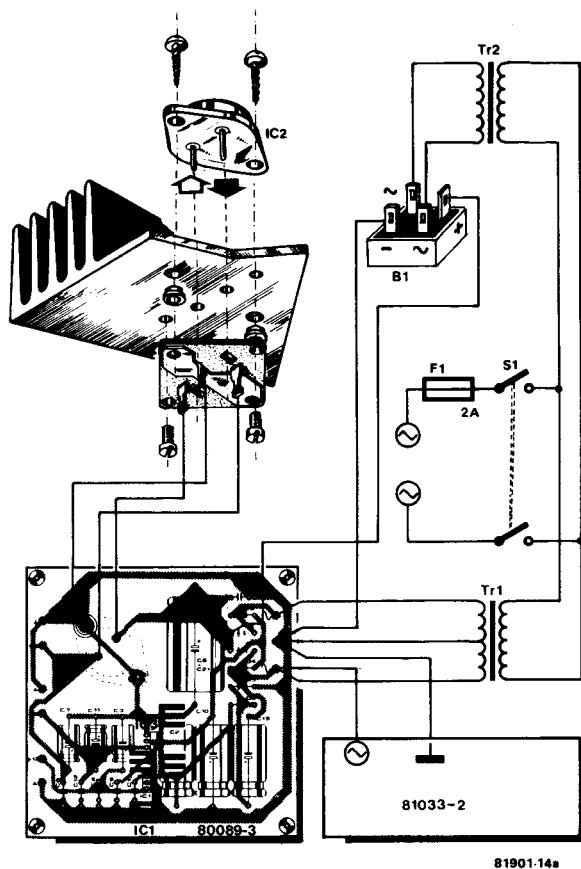
Figuur 13. De print van de min-12-V-voeding.

koelplaat en de kastbodem, in verband met een goede luchtcirkulatie. Of boor anders ter plaatse wat gaten in de kastbodem (en zet de kast dan op voetjes!). Die IC2 kan zonder isolatieplaatje op de koelplaat worden gemonteerd. Dat betekent dan wel dat het metaal van de kast officieel aan massa ligt, zodat sluitingen via metalen afstandsbusen die in contact komen met printsporen uit den boze zijn.

Wel, als u de bouw van de standaard-junior-computer tot een goed einde hebt gebracht moet het volbouwen van de min 12 volt-print een peuleschilletje voor u zijn. Vergeet niet om IC4 te voorzien van een zelfde koelplaat als gebruikt voor IC1 en IC3. De positie van de koelplaat staat duidelijk aangegeven in figuur 13.

De bouwtekening van figuur 14 helpt u een eind op weg als het gaat om de vraag hoe de diverse dingen aan elkaar te hangen. Bijvoorbeeld de bedrading van de nieuwe bruggelijkrichter B1 voor het zware +5 V-werk, die zonder isolatie op de kastbodem kan worden gemonteerd. Of op de achterkant, als er maar contact is met koelend metaal. Gebruik voor de aansluitingen van B1 van die handige kabelschoentjes, die je ook tegenkomt in het elektrische boordnet van auto's.

En over de bedrading nog dit. De voedingsaansluitingen voor de uitgebrei-



Figuur 14. De belangrijkste details van de montage en bedrading van de uitgebreide voeding (4a) en enige IC- en koelplaat-details (14b).

Tabel 5.

Onderdelenlijst uitgebreide voeding

(extra print: EPS 81033-2: voeding -12 V)

N.B. "%" betekent: gewijzigd

"&" betekent: nieuw (doorlopende nummering)

Kondensatoren:

C1,C2,C10,C14(&),C15(&) = 470 μ /25 V

C3,C11,C17(&) = 47 μ /16 . . . 25 V

C4,C5,C8,C9,C12,C13,C16(&),

C18(&) = 100 n MKH

C6,C21(&) = 2200 μ /25 V (C21//C6)

(of C6(%) = 4700 μ /25 V; C21 vervalt)

C7 = 100 μ /25 V

C19(&),C20(&) = 220 μ /40 V

(C19//C1; C20//C14) (C19 en C20 vervallen

bij C1(%),C14(%) = 680 μ /40 V)

Halfgeleiders:

IC1(%) = 7812 (TO-220)

IC2(%) = 78H05 (TO-3)

IC3(%) = 7905 (TO-220)

IC4(&) = 7912 (TO-220)

D1,D2 = vervallen; zie D7 en D8

D3,D4,D5,D6,D7(&),D8(&) = 1N4004

B1(&) = FB 1003

Tr1 = bestaande trafo

Tr2(&) = 1 x 10 V/4 A

S1 = bestaande netschakelaar

F1(%) = 2-A-zekering

(&): koelmateriaal voor IC1,IC2,IC3,IC4

optie (indikatie)

D9 = LED groen; R1 = 470 Ω : geel-violet-
bruin-(goud)

D10 = LED rood; R2 = 680 Ω : blauw-grijs-
bruin-(goud)

de junior-computer vinden plaats via vijf leidingen (massa = nul inbegrepen) naar vijf daarvoor bestemde aansluitpennen op de interface-kaart. Die leidingen komen van de min-12 V-print en van de ietwat gewijzigde bestaande voedingsprint. En denk erom: soepel draad gebruiken (in meerdere kleuren), waarbij de nul- en de +5 V-draden minstens 1 mm² dik zijn! Vergeet overigens niet de nieuwe, zwaardere zekering in de zekeringhouder te plaatsen. De oude zou met die twee trafo's die we nu hebben bij het inschakelen meteen de geest geven en dan krijg je allemaal maar paniek om niets. Men kan als men wil een LED-indikatie aanbrengen. De rode LED D10 verradt bij oplichten het onder stroom zijn van Tr2 (+5 volt), de groene idem dito van de bestaande Tr1 (overige voedingsspanningen).

De LEDs krijgen dan een plaatsje ergens in het zicht (gaatjes boren met een diameter zodanig dat de LEDs net hun kop opsteken, zoals de asperges bij ons; met een geschikte lijm vastlijmen; eventueel gebruik maken van een rubber tule). De serieweerstand (R1, R2) moet zwevend worden bedraad. Voor "verse" LEDs geldt overigens een ezelsbruggetje om de aansluitingen uit elkaar te houden: K - K, oftewel *katode* = *korte aansluitdraad*.

Het oplichten van de LEDs geeft overigens geen 100% uitsluitel over de korrekte hoogten van de voedingsspanningen.

Niets weerhoudt u ervan om de voeding alvast te testen, zonder dat de uitgebreide junior-computer is aangesloten. U kunt, als de spanningen in orde zijn (opgegeven waarde $\pm 5\%$) testen onder extreme bedrijfscondities door een dummy-belasting (de extreme werkelijkheid nabootsen) aan te sluiten in de vorm van een weerstand (30 ohm/5 W voor ± 12 V, $12\frac{1}{2}$ ohm/2 W voor -5 V en $1\frac{1}{4}$ ohm/20 W voor $+5$ V). Let wel, dat **hoeft** niet, maar geeft wel een indruk van het koelend vermogen van de koelplaten voor IC1... IC4. Blijkt de koeling te krap te zijn (vele trafo's geven te hoge sekondaire spanningen ter compensatie van verliezen die er niet blijken te zijn, met als gevolg te hoge "primaire" gelijkspanningen voor de stabilisator-IC's en daardoor een hogere warmte-ontwikkeling), dan kan men nog altijd zijn maatregelen nemen. Deze belastingtest is eigenlijk pas zinvol als de voeding in de (of een) kast is ingebouwd. Dit in verband met het effect van al dan niet voldoende ventilatie, dus koeling.

Aktiepunt drie: de interface-kaart volbouwen

Een interface-je bouwen

- **schema's** → figuur 1
→ figuur 2
- **onderdelen** → tabel 6
- **print** → figuur 15 (opdrukzijde inclusief koper-layout) (verkleind)
→ figuur 16 (koper-layout opdrukzijde) (verkleind)
→ figuur 17 (koper-layout "koperzijde") (verkleind)
- **draadbruggen IC4/IC5** → tabel 7
- **konnektors** → figuur 19
- **IC-plattegronden** → figuur 18

(Voor een versie van figuur 15 op ware grootte, maar met de koper-layout van de "koperzijde": zie figuur 7 op pagina 5-60 van *Elektuur*, mei 1981)

Nou, dat is er weer net zo eentje als de hoofdprint van de standaard-junior-computer: aan twee kanten een koperbanen-patroon en geleidende gaten (doormetallisering). Omdat *de meeste, maar niet alle* onderdelen op één bepaalde kant komen te zitten (die we voortaan de "opdrukzijde" zullen noemen) is er hier afgezien van een dubbelzijdige componentenopdruk. En het is goed om alvast in de gaten te houden dat de opdrukzijde zich weliswaar van nature het beste leent om als bovenkant te worden betiteld, maar dat dit gezien de voorkeursopstelling van de kaart (komt later) nou net de verkeerde betiteling is. Dus: opdrukzijde onder.

Dus figuur 15 toont ons één kant van de EPS-print, figuur 16 diezelfde kant, maar dan zonder witopdruk en figuur 17 de andere kant: de "koperzijde".

Tabel 6.

Onderdelenlijst voor de interface-kaart

Weerstanden:

R1, R2, R3, R4, R32, R33, R34,
R35 = 1 k: bruin-zwart-rood-(goud)
R5 = 22 k: rood-rood-oranje-(goud)
R6, R10, R11, R14, R15, R24, R26, R27,
R28 = 10 k: bruin-zwart-oranje-(goud)
R7, R8, R36 = 8k2: grijs-rood-rood-(goud)
R9, R18, R22, R23 = 4k7: geel-violet-
rood-(goud)
R12 = 6k8: blauw-grijs-rood-(goud)
R13, R25, R31 = 2k2: rood-rood-rood-(goud)
R16 = 100 Ω : bruin-zwart-bruin-(goud)
R17 = 330 Ω : oranje-oranje-bruin-(goud)
R19 = 470 Ω : geel-violet-bruin-(goud)
R20 = 1k2: bruin-rood-rood-(goud)
R21 = 15 k: bruin-groen-oranje-(goud)
R29 = 33 k: oranje-oranje-oranje-(goud)
R30 = 4M7: geel-violet-groen-(goud)
R37 = 33 Ω (zie tekst)
P1 = 5 k (4k7) meerslagen-instelpotmeter
P2 = 1 k instelpotmeter

Kondensatoren:

C1 = 220 n MKH
C2, C11, C12, C13 = 10 μ /16 V tantaal
C3 = 22 n MKH
C4 = 1 n MKH
C5, C6, C7 = 6n8 MKH
C8 = 100 n MKH
C9 = 47 n MKH
C10, C14 . . . C22 = 1 μ /16 V tantaal
(totaal 10 stuks)

Halfgeleiders:

T1 = BC 547B en ekwivalenten
T2, T3 = BC 516
D1, D2, D3 = 1N4148
D4 = LED groen
D5 = LED rood
IC1 = 6522 (Rockwell, Synertek)
IC2, IC3 = 2114

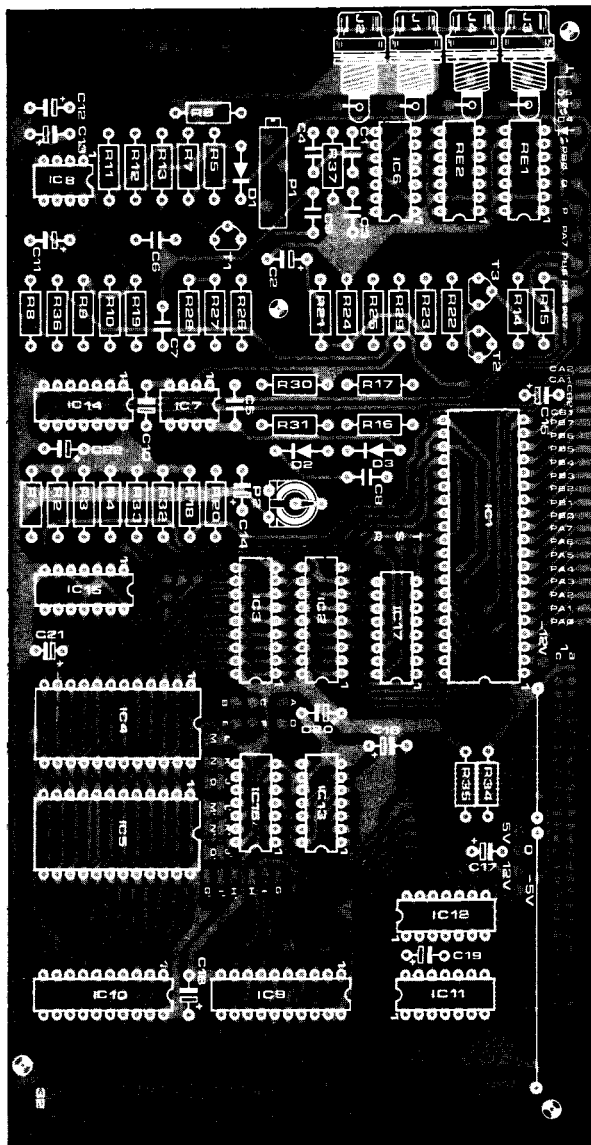
IC4 = 2716 (TM; ESS 506), 2708, 8114
IC5 = 2716 (PM; ESS 507), 2708, 8114
IC6 = 565
IC7, IC8 = 311
IC9, IC10 = 74LS241
IC11, IC12 = 74LS243
IC13 = 74LS27, 7427
IC14 = 74LS01, 7401
IC15 = 74LS30, 7430
IC16 = 74LS00, 7400
IC17 = 82S23 (ESS 508), 74188
(niet ESS 508!!)

Diversen:

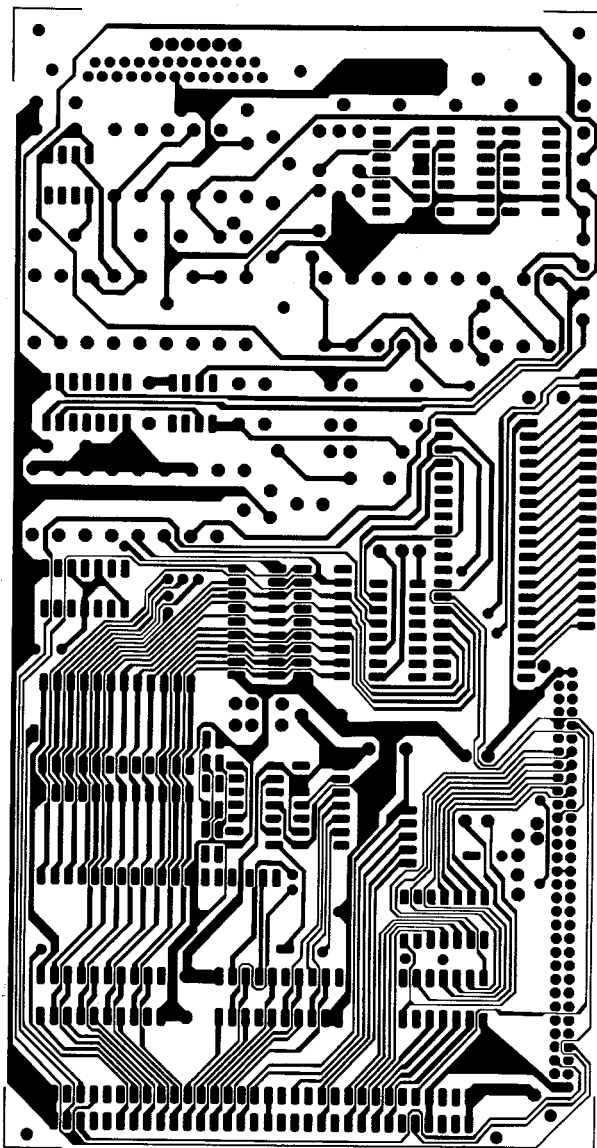
Re1, Re2 = DIL reed relais
Günther 1301, 380 Ω
2 IC-voeten 8-pens
9 IC-voeten 14-pens
1 IC-voet 16 pens
2 IC-voeten 18-pens
2 IC-voeten 20-pens
2 IC-voeten 24-pens (zie ook tekst)
1 IC-voet 40-pens
5 draadbruggen op de print (afgezien van de
geletterde)
J1, J2, J3, J4 = "cinch"-chassisdeel
1 25 polige female D-konnektor (RS 232) haaks
(printmontage) (zie ook figuur 19e)
20 soldeerpenen (VIA-"konnektor")
29 soldeerpenen (van letters voorziene
aansluitpunten)
1 ingangskonnektor 64-polig haaks;
DIN 41612; male! (gelijk aan de
uitbreidingskonnektor van de standaard-
junior-computer) (zie ook figuur 19a)
5 soldeerpenen (verbindingen met de
poortkonnektor)
5 soldeerpenen (voedingsaansluitingen)
3 soldeerpenen (aansluitingen van de LED's)
1 uitgangskonnektor 64-polig haaks; female
(zie ook figuur 19c) (zie ook tekst!!)

Het volbouwen van de interface-kaart, punt voor punt.

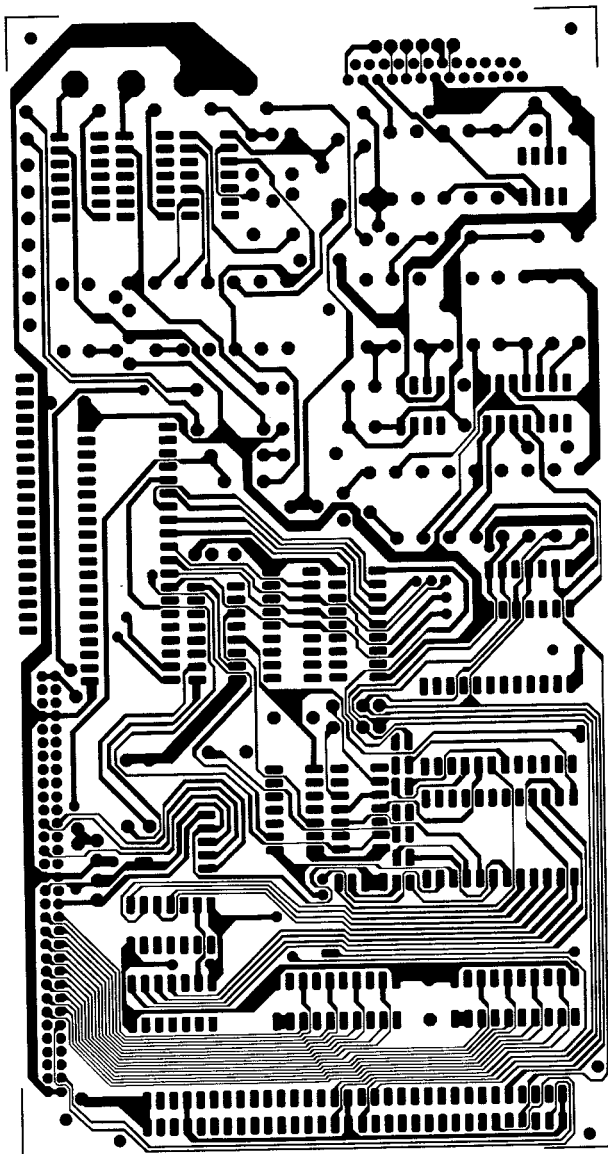
1. Eerst maar de *weerstanden*. De afgeknipte draden kunnen we misschien later nog ergens voor gebruiken. De kleurcodes staan in tabel 6. Het zijn er 36 of 37, afhankelijk van de vraag of R37 kan vervallen of niet. Deze weerstand is alleen nodig als gebruik wordt gemaakt van de luidspreker- of hoofdtelefoonuitgang van de (INPUT-) cassette-recorder (automatische afkoppeling van de ingebouwde luidspreker). Voor de meeste, wat simpelere cassette-recorders is men aangewezen op een dergelijke uitgang. Heeft men een "echte" recorder met een lijn-uitgang, dan vervalt R37. Anders zou een buitengewoon sterke signaalverzwakking optreden en dat is nergens goed voor: waarom zou je de PLL-goden verzoeken?
2. De *instelpotmeters*. P2 is van het gewone doordeweekse type, P1 is een precisietrimmer voor de PLL-afregeling (die in hoofdstuk 11 wordt besproken).



Figuur 15. De onderdelenzijde van de interface-kaart, inclusief het bijbehorende koperbanenpatroon (verkleind weergegeven!).



Figuur 16. Het koperbanenpatroon van de onderdelenzijde (= opdrukszijde) van de interface-kaart (verkleind weergegeven!).



Figuur 17. Het koperbanenpatroon van de "koperzijde" van de interface-kaart (verkleind weergegeven!).

Tabel 7

Draadbruggen rond IC4 en IC5

IC	geheugen	type	G . . . O G' . . . O'	A . . . F	geheugenbereik
IC4	1K-RAM	8114	O - M	A - B	0800 . . . 0BFF
	1K-EPROM	2708	O - N G - H J - K	A - B	0800 . . . 0BFF
	2K-EPROM	2716	O - N G - I J - L	A - B - C ¹	0800 . . . 0FFF ¹
IC5	1K-RAM	8114	O' - M'	D - C ² D - E ⁴	0C00 . . . 0FFF ² 1000 . . . 13FF ⁴
	1K-EPROM	2708	O' - N' G' - H' J' - K'	D - C ³ D - E ⁴	0C00 . . . 0FFF ³ 1000 . . . 13FF ⁴
	2K-EPROM	2716	O' - N' G' - I' J' - L'	D - E - F ⁵	1000 . . . 17FF ⁵

¹ Bestemd voor systeemprogramma TAPE MONITOR (TM); ESS 506

² Voorkeur indien IC4 = 8114 (aaneengesloten RAM-bereik)

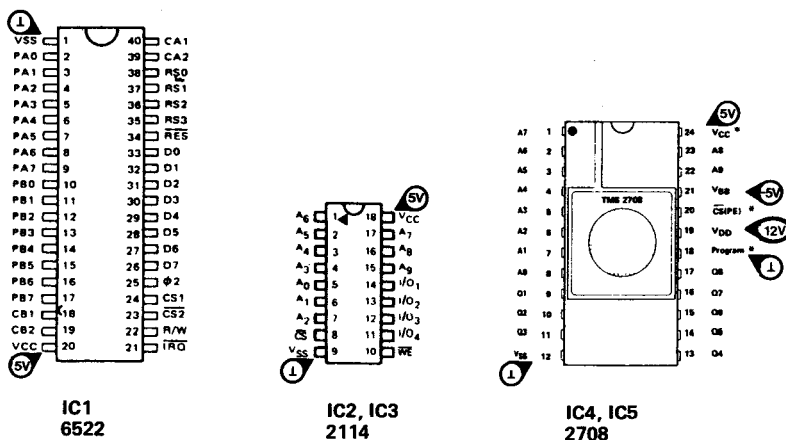
³ Voorkeur indien IC4 = 2708 (aaneengesloten EPROM-bereik) of indien IC4 = 8114 (aaneengesloten geheugenbereik)

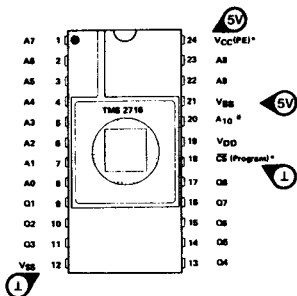
⁴ Bij IC4 = 2716

⁵ Bestemd voor systeemprogramma PRINTER MONITOR (PM); ESS 507

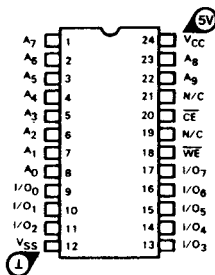
N.B. Er zijn nog meer K-aansluitingen mogelijk. Alleen de meest logische keuzen zijn in de tabel opgegeven.

Figuur 18. Plattegronden van alle IC's van de interface-kaart, inclusief ge- of te meten gelijkspanningen, waar dat zinvol is.

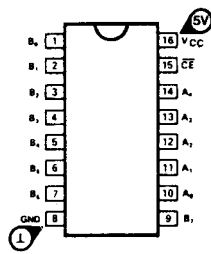




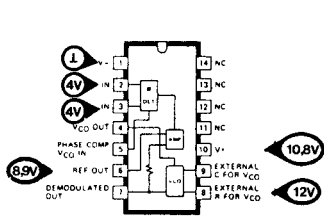
IC4, IC5
2716



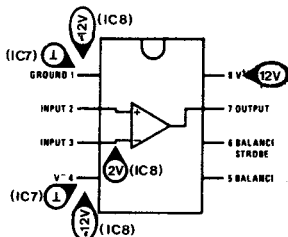
IC4, IC5
8114



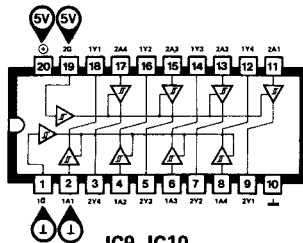
IC17
82S23



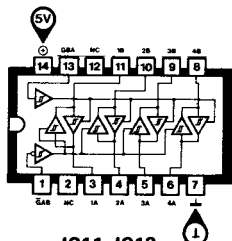
IC6
565



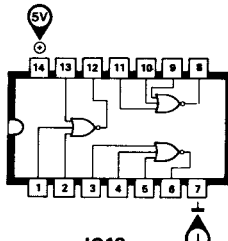
IC7, IC8
311



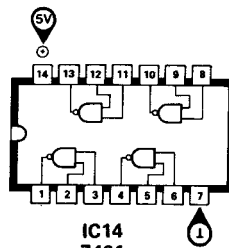
IC9, IC10
74LS241



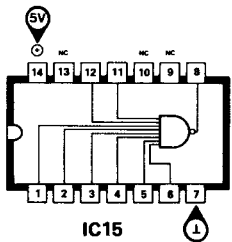
IC11, IC12
74LS243



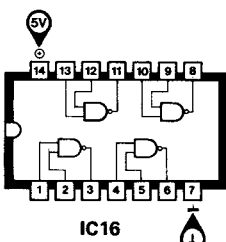
IC13
7427
74LS27



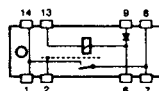
IC14
7401
74LS01
open kollektor



IC15
7430
74LS30



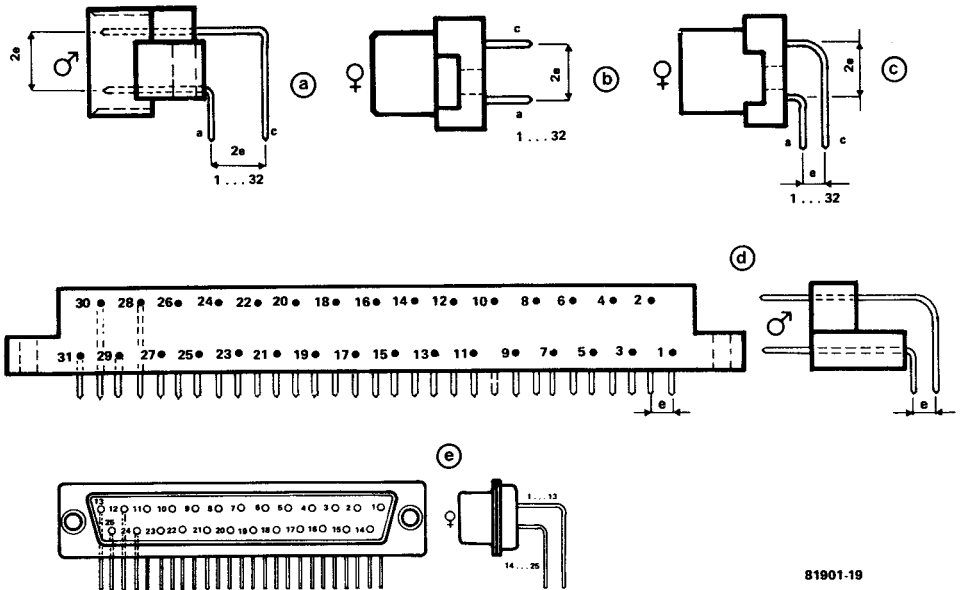
IC16
7400
74LS00



Re1, Re2
1301/380

81901-18

Figuur 19. Een overzicht van de diverse konnektors die bij de bouw van de uitgebreide junior-computer een rol (kunnen) spelen.



81901-19

3. De *MKH-kondensatoren*. Let erop dat de korte zijden, met daarop de aansluitingen, elektrisch geleidend zijn en dat dus geen contact mag optreden met belendende, mogelijk wat uit de koers geraakte metalen delen van andere componenten (helaas blijkt dit soort tips niet overbodig te zijn).
4. De *tantaalelko's*. Let op de juiste positie van de plus-aansluiting en van (wat net zo belangrijk is) de minaansluiting.
5. *Draadbruggen/soldeerpennen*. Als we goed hebben geteld zijn er op de interface-kaart 62 punten betrokken bij een aansluiting of draadbrug. Ofschoon in tabel 6 uitsluitend van soldeerpennen sprake is is dat geen "must". Draadbruggen en draadaansluitingen kunnen ook direkt worden gesoldeerd. Temeer daar om technische redenen de betrokken gaten net zo groot zijn als de gewone gaten; dus zijn alleen dunne soldeerpennen geschikt. Voor een deel hangt de hoeveelheid ijzerwerk samen met keuzen gebaseerd op de actiepunten 4 en 5.
Bij gebruik van een Elekterminal en de meeste printers blijft de draadbrug P-Q achterwege. En dan een heel belangrijke draadbrug: wél buskaartgeheugen → draadbrug R-S (& D-EX op standaardkaart!) geen buskaartgeheugen → draadbrug R-T (& D-1 op standaardkaart!) De overige verbindingen tussen "geletterde" punten volgen uit de keuze van IC4 en IC5. Zie daartoe tabel 7.

Vergeet de twee "gewone" draadbruggen niet, bij de uitgangskonnektor (opdrukzijde) en ook niet de drie draadbruggen bij de RS232-konnektor (doorverbindingen 4 & 5, 5 & 8 en 6 & 20). Deze laatste draadbruggen zijn niet strikt nodig bij Elekterminal-gebruik, maar nu kan men er nog goed bij.

6. *IC-voeten*. Gezien de verbindingsskwaliteit en het gemak waarmee IC's kunnen worden verwisseld is er eigenlijk helemaal niets op tegen om IC-voeten te gebruiken, óók voor de DIL-relais. Dan moet echter niet alleen de positie van de IC's in orde zijn, maar ook de positie van *alle* pennen! Anders is het middel erger dan de kwaal. De meeste IC-voetjes hebben een uitsparing op een korte zijde of een speciale aanduiding voor pen 1. Hoewel voor *de voetjes* twee posities mogelijk zijn is het handig om de positie op de print van elk voetje in overeenstemming te brengen met de IC-positie. Voor de laatste: zie de componentenopdruk van figuur 15.

Bij de montage van de voeten drukt men deze goed tegen de print aan. Vervolgens soldeert men voorlopig twee diagonaal tegenover elkaar liggende pennen, bijvoorbeeld 1 en 8 voor een 14-pens voet. Nu heeft men twee handen vrij. Dan soldeert men definitief de overige pennen vast en dan pas definitief de twee diagonale pennen. Mechanisch en elektrisch kontakt verzekerd!

Als men kiest voor EPROM voor IC4 en of IC5 kan men met voordeel gebruik maken van een speciale 24-pens IC-voet waarbij, onder gebruikmaking van een ontkoppelbare klemverbinding de EPROM zonder mechanische overredingskracht op het IC kan worden ingezet en verwijderd. Dat verhoogt het uitwisselgemak en de levensduur van de 24 pennen (type Burndy DIL BQ24P-101).

7. *Transistoren en gewone dioden*. Let op de positie van het "buikje" van de transistoren en van de katoden (en dus ook van de anoden) van de dioden.
8. *LEDs*. De cassette-LEDs D4 en D5 komen niet op de print maar moeten zichtbaar zijn op het bedieningspaneel van de kast. Aansluitpunten voor de verbindingen zijn op de print aanwezig. Denk aan de Katode-Kortregel voor nieuwe LEDs.
9. *IC's*. Let op de juiste positie! Alle pennen echt aangesloten? Grondig controleren voorkomt veel nodeloze ellende achteraf.

Minstens één IC moet geprogrammeerd zijn en dat is IC17, de PROM (zie tabel 2). Dat kan via de ESS (voor het type 82S23, niet voor de op zich bruikbare 74188!). Kiest men voor IC4 een EPROM, dan moet ook deze zijn geprogrammeerd. Gaat het om een 2716 met daarin het programma TM, dan kan het programmeren door ons via ESS 506 plaatsvinden. Of u doet het zelf, aan de hand van de hex dump achterin dit boek. Zonder TM kan men overigens net zo goed alle cassette-hardware weglaten, tenzij men zelf alternatieve cassette-software heeft.

Soortgelijke overwegingen gelden voor IC5. Bij de keuze voor EPROM hoort programmeren. Bij de keuze voor 2716, met daarin PM, kan een keuze voor ESS 507 horen. Zelf doen kan ook, ook nu op basis van een hex dump achterin dit boek. Moet je natuurlijk wel over 2716-programmeerspullen (kunnen) beschikken. Bij de keuze voor PM

hoort de keuze voor TM, in verband met de toetsfuncties G en S van PM.

10. De *relais* Re1 en Re2 zijn eigenlijk alleen zinvol in het geval van twee cassette-recorders, eentje voor het opbergen van programma's (OUTPUT) en één voor het ophalen van data (INPUT). En verder moet op beide cassette-recorders een schakelkontakt voor afstandsbediening (remote control) voorhanden zijn. In alle andere gevallen mag u die relais rustig weglaten, inclusief de twee voeten, J3 en J4.
11. Dan de *chassisdelen J1 . . . J4*, voor de verbindingen met de cassette-recorder(s). Ze kunnen direkt, op de koperzijde worden gemonteerd, in combinatie met — naar zal blijken — gaten in de rechter zijkant van de kast. Een andere aanpak is de montage op bijvoorbeeld de achterkant van de kast; J3 en J4 moeten geïsoleerd, vrij van massa worden gemonteerd — dat wil zeggen: als ze nodig zijn (zie punt 10). Maak in dat geval de verbindingen naar J1 en J2 met behulp van afgeschermd draad.
12. De *ingangskonnektor* wordt indien nodig (zie aktiepunt 4) gemonteerd op de daarvoor bestemde korte zijde van de interface-kaart en met het konnektorlichaam aan de opdrukszijde en dus het soldeerwerk aan de "koperzijde". Het montagevoorschrift is in wezen hetzelfde als dat voor de IC-voeten (punt 6). Zie verder ook figuur 19a.
13. De *RS232-konnektor*. Indien de 25-polige D-konnektor op de print wordt gemonteerd moet gebruik worden gemaakt van het in tabel 6 omschreven en in figuur 19e getekende type: haakse pennen voor printmontage. De montage geschiedt op de daarvoor bestemde korte zijde van de interface-kaart *op de "koperzijde"*. Montage op de opdrukszijde leidt tot gespiegelde en dus verkeerde verbindingen. De ontwerpkeuze voor montage op de "koperzijde" leidt tot een effectief, economisch gebruik van de ruimte tussen interface- en basiskaart, zodat deze kaartsandwich niet nodeloos dik wordt. Montage op de print van de RS232-konnektor houdt in dat randapparatuur wordt aangesloten via de rechter zijkant van de kast. Het is ook mogelijk om deze randapparatuurkonnektor op de achterkant van de kast te monteren (neem dan een type dat geschikt is voor draadmontage) en de verbindingen met de interface-kaart te maken via "flat cable", montage draad met meerdere rijstroken.
14. De *VIA-"konnektor"* omvat 20 soldeerpennen of draadaansluitingen. De VIA is met name bedoeld voor applicaties en het ligt voor de hand om hiervoor een konnektor te bestemmen op de achterkant van de kast of op het bedieningspaneel van de kast. Te denken valt aan een 31-polige konnektor (female) van hetzelfde type als de PIA-poort-konnektor van de standaard-junior-computer. Verbindingen met de interface-kaart gaan dan weer via flat cable.
15. Tenslotte de *uitgangskonnektor*. Deze is in ieder geval niet nodig als geen gebruik wordt gemaakt van buskaartgeheugen. Maar ook met buskaartgeheugen is ie niet altijd nodig. Er valt namelijk ook iets te zeggen voor direkte draadverbindingen met de buskaart. Kiest men voor een konnektorverbinding met de buskaart, dan is het konnektortype volgens figuur 19c een mogelijkheid (montage op de "koperzijde"!). Echter: er zijn daarbij problemen te verwachten —

die overigens oplosbaar zijn, lees maar verder. De afstand tussen a-pennerij van de uitgangskonnektor en de rand van de print is zo groot dat de uitgangskonnektor niet tegen de rand van de print kan worden gemonteerd. Dat komt omdat aan beide zijden printsporen lopen waarvoor elders geen plaats is. Montage op de print betekent dat de pennen niet ver genoeg door de print steken om betrouwbare soldeerverbindingen te garanderen. Een oplossing is het wegbuigen, met een tang, van de rechte, haakse hoeken zodat lengte gewonnen wordt. Of men neemt een konnektor van het type figuur 19b in de versie met pennen van minstens 13 mm, bijvoorbeeld een wire-wrap-type, en buigt de pennen op maat. Montage tegen de rand van de print is dan mogelijk. Nog een andere mogelijkheid is de keuze voor een konnektor van het type figuur 19b in de versie voor draadmontage, en de verbindingen met de print te maken via 64 draadjes. Wordt de afstand groter (bijvoorbeeld bij de aansluiting van buskaartgeheugen via een konnektor op de achterkant van de kast) dan nemen we natuurlijk flat cable in plaats van 64 draadjes.

Zie voor deze kwestie alvast de figuren 20b en 20c en lees hierover meer in aktiepunt vier — waar we nu aan toe zijn.

Trouwens: er is nog een alternatief. We hebben het gehad over een a-pennenrij en een c-pennenrij. Bestaat er zoiets als een tussenliggende b-pennenrij? Jazeker! Namelijk bij een 96-polige konnektor! Dus als uitgangskonnektor kan óók een 96-polige konnektor worden gebruikt, waarvan de a-pennenrij wordt afgeknipt en waarvan de b-pennenrij wordt verbonden met de a-pennenrij van de interface-kaart. Uiteraard moet de aansluitende buskaartkonnektor dan óók 96-polig zijn en moet de afstand tussen de haaks omgebogen b- en c-pennenrij zodanig zijn dat de pennen passen in de buskaart.

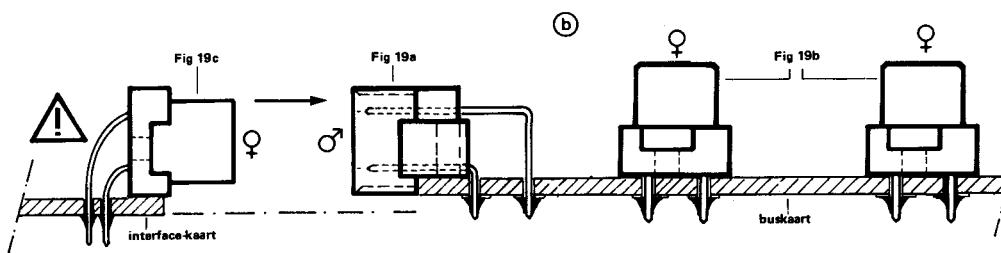
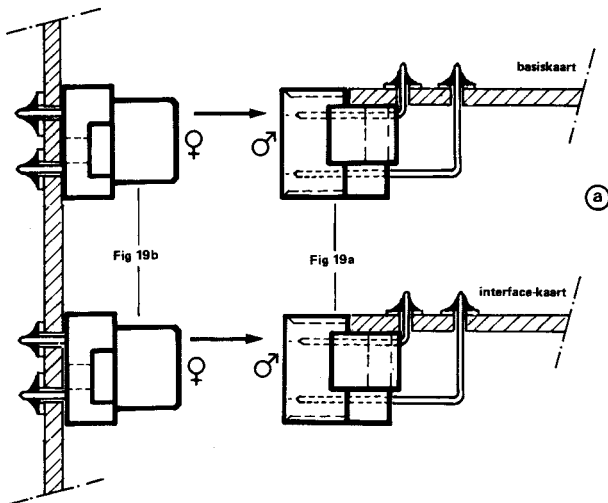
Aktiepunt vier: Kaarten doorverbinden

Kaarten op tafel!

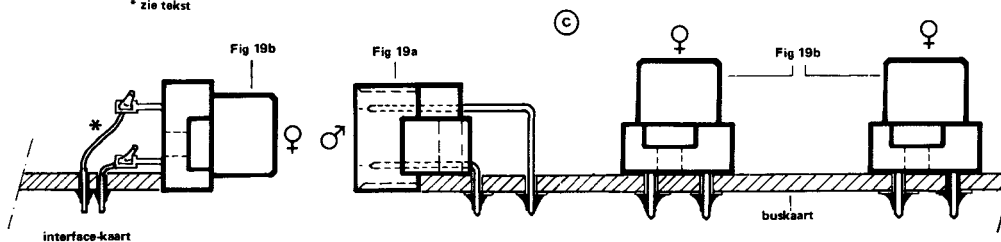
- **bouwtekening** → figuur 20 (details)
- **onderdelen** → tabel 8
- **konnektors** → figuur 19
- **buskaart** → figuur 21 (verkleind)

Nu dreigen we een beetje in de problemen te raken omdat er vele raakvlakken zijn tussen dit aktiepunt en het volgende, dat gaat over de kast of kasten. Maar voordat u met bouwen begint heeft u vast wel beide onderdelen gelezen.

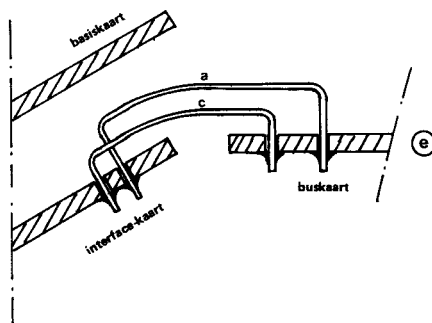
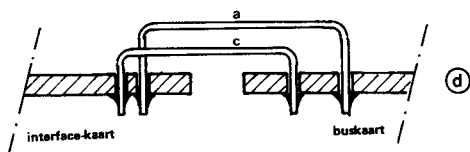
De interface-kaart is net zo groot als de standaardkaart en heeft op overeenkomstige plaatsen bevestigingsgaten: vijf stuks. Dat is handig. Want de toevoeging van de interface-kaart hoeft dan immers geen oppervlakteverdobbeling te betekenen, omdat hij in principe onder de basiskaart kan worden gemonteerd. De keuze: standaardkaart boven ligt voor de hand omdat zich daarop de hexadecimale toetsen en de twee schakelaars bevinden. En die moeten toegankelijk zijn voor de gebruiker. Als we dus denken aan een sandwichkonstruktie met de basiskaart boven en in combinatie met montage tegen het schuin oplopende bedieningspaneel van de

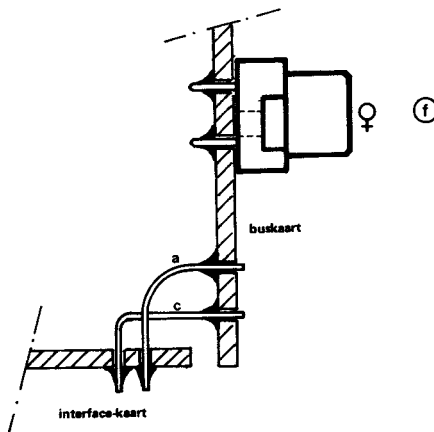


* zie tekst



81901-20





Figuur 20. Een aantal konstruktie-details (verbindingen tussen kaarten) die bij de bouw van de uitgebreide junior-computer een rol (kunnen) spelen.

Tabel 8.

Elektrische verbindingen totaal-systeem

a) basiskaart ↔ interface-kaart

2 konnektoren 64-polig recht; female
(zie ook figuur 19b)

1 print EPS 80024 (deels gebruikt) of
1 plaatje montaprint (met 2,54 mm-
gaatjespatroon)

1 konnektor 31-polig male (zie ook
figuur 19d)

b) interface-kaart ↔ buskaart

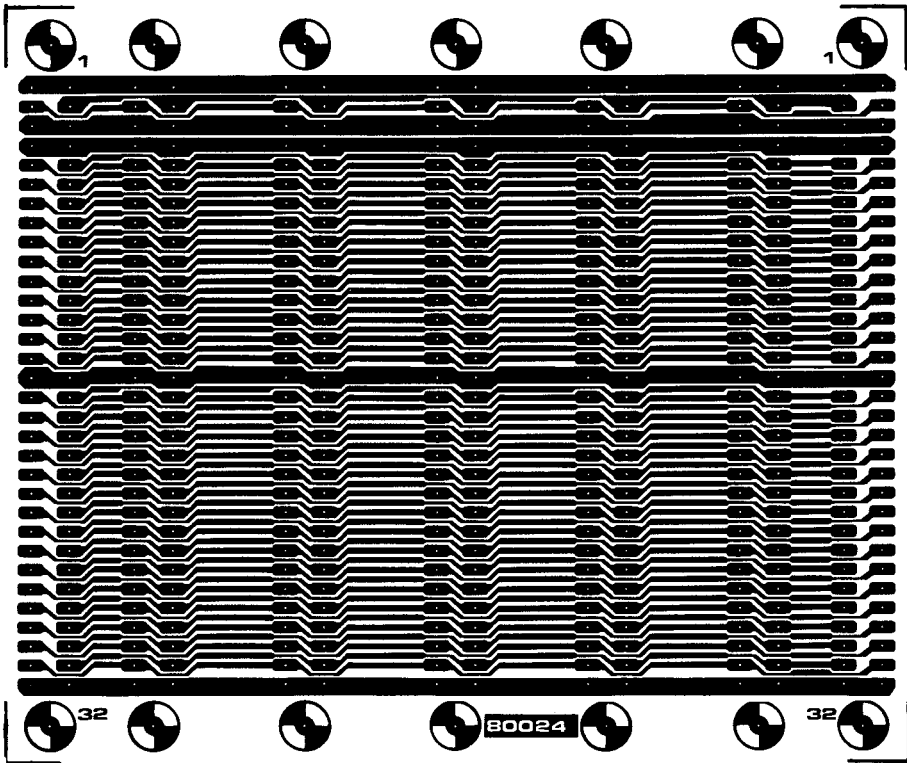
1 konnektor 64-polig haaks; male (zie ook
figuur 19a)

1 à 5 konnektoren 64-polig recht; female
(zie ook figuur 19b)

1 print EPS 80024 of 1 plaatje montaprint
(met 2,54 mm-raster)

kast, dan is een belangrijke eis dat de dikte van de sandwich niet te groot is, omdat anders de toetsen in ergonomische zin (dus kwa bedienings-gemak) niet binnen handbereik liggen.

Gelukkig valt die dikte wel mee. Hij wordt gedikt-teerd door de grootte van de schakelaars S24 en S25. De afstand tussen beide kaarten is hooguit een centimeter of drie. Als het in aktiepunt drie al niet duidelijk is geworden, dan vertellen we het nu nog maar eens een keer: de interfacekaart neemt de onderliggende positie in van de sandwich met de "koperzijde" boven en dus de opdrukzijde, met vrijwel alle onderdelen, onder; verder gaan we uit van een positie met de ingangskonnektor links en onderop en de RS232-konnektor, indien gemonteerd, rechts en bovenop. Deze uitgangs-



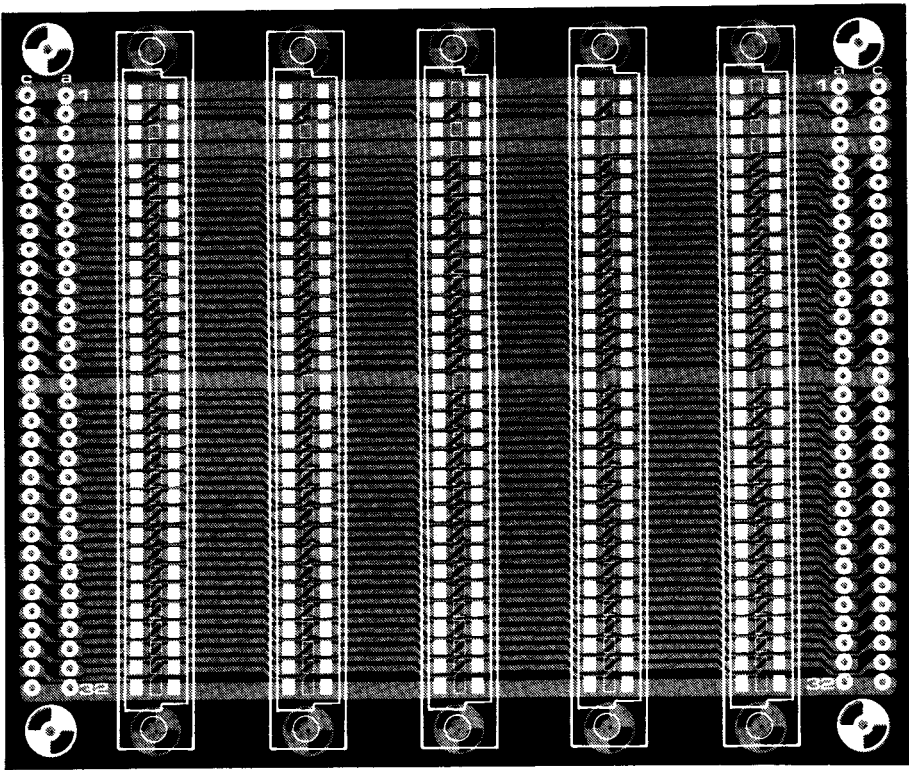
Figuur 21a. Printgegevens van de buskaart.

positie van de interface-kaart is logisch als men uitgaat van de daarmee samenhangende positie van de basiskaart, één verdieping hoger: toetsen boven en in een stand die, vanuit gebruikersstandpunt bezien voor de hand ligt. Dan zit dus de uitbreidingskonektor van de standaardkaart links en onderop.

Met andere woorden: de uitbreidingskonektor van de standaardkaart en de ingangskonektor van de interface-kaart zitten aan dezelfde kant: links. Ze onderling doorverbinden is een kwestie van vertikaal en recht oversteken.

Dan de buskaart. De uitgangskonektor zit op de "koperzijde" en op die lange zijde van de interface-kaart die het verst is verwijderd van de gebruiker/waarnemer; verder zit ie het dichtst bij de ingangskonektor. Kortom: linksachter. Van daaruit "vertrekt" de buskaart in een richting, van de gebruiker weg. Vanuit die buskaart steken 1 à 5 geheugenkaarten de lucht in, met printvlak(ken) evenwijdig aan de lange zijden van de kaarten-sandwich.

Het zojuist geschetste panorama is het uitgangspunt voor beschouwingen over de diverse verbindingen tussen de kaarten. Er zijn echter varianten



mogelijk. De invloed van de kastkeuze op deze verbindingen wordt eveneens behandeld, hoewel daar in aktiepunt vijf pas het laatste woord over zal zijn gezegd.

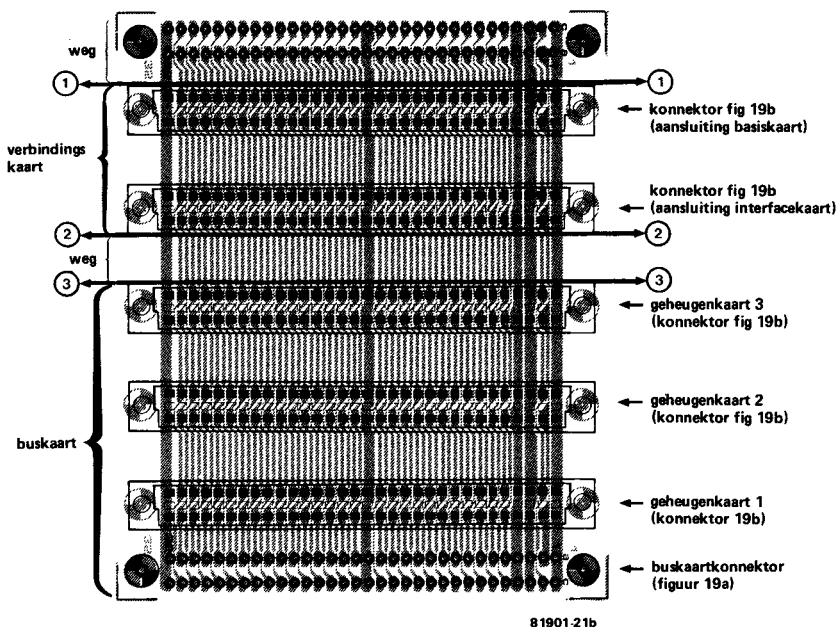
Van basiskaart naar interface-kaart

64 keer oversteken

Er zijn verschillende manieren om de verbindingen tussen uitbreidings- en ingangskonnekter te leggen:

1. Maak gebruik van een *gedeelte van de buskaart*. Er is tot nog toe veel gepraat over "buskaart" (en "buskaartgeheugen"), maar wat is dat eigenlijk konkeert voor een ding? We bedoelen de SC/MP-buskaart EPS 80024, die is besproken in het novembernummer 1979 van het tijdschrift *Elektuur* en waarvan figuur 21a alles laat zien.

De gedachte is om van deze kaart een middenste deel te nemen dat drie geheugenkaart-konnektoren omvat waarvan de middelste wordt overgeslagen en waarvan de beide onbruikbare uiteinden worden afgezaagd ... Protest!!! Zo'n mooie print met van die fragiele printsporen twee



Figuur 21b. Indien men niet meer dan drie geheugenkaarten als buskaartgeheugen gebruikt, kan een deel van de buskaart EPS 80024 worden gebruikt als verticale verbindingsprint tussen de standaardkaart en de interface-kaart. In deze figuur is aangegeven welke delen men voor wat gebruikt, inclusief twee niet meer gebruikte delen. Het gedeelte boven de lijn ①-① is niet meer nodig, evenals het gedeelte tussen de lijnen ③-③ en ②-②. Het gedeelte tussen de scheidslijnen ②-② en ①-① is de verbindingskaart en wat er overblijft is de ingekorte buskaart. Het gaat hier om de componentenzijde (= opdrukszijde) van de buskaart, dus wat de buskaart betreft datgene dat men ziet als men van boven af op de uitgebreide junior-computer neerziet, en wat de verbindingskaart betreft datgene dat men ziet als men vanuit een rechtse, of zo men wil oostelijke positie (zie de foto's 1 . . . 3) in de ruimte tussen de standaardkaart en de interface-kaart kijkt. Let op de positie van de drie plus twee figuur-19b-konnektoren; de aansluitende mannetjeskonnektor (op een geheugenkaart, op de standaardkaart en op de interface-kaart) kan maar in één positie in de doorverbindingskonnektor worden geplaatst. Relaterend aan de positie van figuur 21b vallen de loodrecht op figuur 21 staande kaarten ongeveer samen met de bovenste witte rand van de desbetreffende doorverbindingskonnektor op de ingekorte buskaart respectievelijk de verbindingskaart.

Laat het uiteenvallen van figuur 21b in twee bruikbare en twee niet meer gebruikte delen bij voorkeur niet plaats vinden via (figuur)zagen (en als men dat toch doet: eerst de diverse konnektors monteren — dit in verband met het losraken van printbanen), maar (laten) knippen met een schaar die geschikt is voor het knippen van dun metaal (1 . . . 2 mm). Er is vast wel een goede ijzerwareenzaak of machinefabriekje bij u in de buurt waar u dit voorelkaar krijgt, tegen geringe kosten en met weinig ergernis.

keer doorzagen!? Bent u nou helemaal... En wordt de sandwich dan niet nodeloos dik? Wel, niets weerhoudt u van deze methode indien de genoemde bezwaren in uw geval niet van toepassing zijn. Zie ook figuur 20a (zij-aanzicht). Let er goed op dat de koperbanen op de buskaart asymmetrisch zijn. De punten 3 en 4 zijn verbonden met dikke koperleidingen de, bij omgedraaid gebruik van de kaart als punt 3 en 4 dienende leidingen niet! Kijkt men zodanig tegen de sandwich aan dat het zij-aanzicht volgens figuur 20a van toepassing is, dan zitten de punten 32 het dichtst bij; de punten 1 zijn het verst verwijderd.

2. *Maak zelf een verbindingssprintje* van experimenteerprint met een gaatjesraster van 2,54 mm. De afstand tussen de konnektors (figuur 20b) kan men afstemmen op de individuele situatie. Een afstand tussen de figuur 19b-konnektors op dat "printje" van minimaal anderhalf keer dezelfde afstand op de buskaart is mogelijk. Eventueel de schakelaars S24 en S25 van de basiskaart halen en apart monteren. Tussen overeenkomstige punten van beide konnektors moeten verbindingen worden gelegd. Men kan eventueel de verbindingen tussen loze, niet gebruikte konnektorpennen laten vervallen.

3. Men verbindt twee figuur 19b-konnektors door via flat cable. Soepeler kan het niet!

MAAR DIT IS TEVENS DE BESTE METHODE OM IN DE PROBLEMEN TE RAKEN WEGENS FOUTIEVE VERBINDINGEN, ZOALS ONDERLINGE KORTSLUITINGEN!!

4. Men ziet helemaal af van konnektors, dus óók van de uitbreidings- en ingangskonnektor, en legt de verbindingen direkt tussen de diverse c & a-soldeerpunten. We zijn eerlijk gezegd niet kapot van deze methode. Kosten besparen is leuk, maar dit is alleen iets voor buitengewoon ervaren soldeerders, en nou juist die zien het grote nut in van konnektors!

Een andere groep verbindingen tussen de beide sandwich-kaarten betreft vijf aansluitingen op de poortkonnektor. Aan de kant van de basiskaart soldeert men de vijf draden op de juiste punten van een 31-polige male-konnektor (figuur 19d) die in de bestaande poortkonnektor wordt gestoken. Men kan natuurlijk ook vijf losse soldeerpennen nemen in plaats van dat ding van figuur 19d. Of je soldeert de draden op de soldeerpunten van de poortkonnektor. Of je laat de PIA-poortkonnektor vervallen en gebruikt hem als VIA-applikatiekonnektor achterop de kast (omdat de VIA geheel vrij is van huishoudelijke taken; dit in tegenstelling tot de PIA, die steeds meer te doen krijgt, maar daar hebben we het al over gehad). Komt de poortkonnektor te vervallen, dan worden de vijf draden aangesloten op de betreffende poortkonnektor-soldeerpunten van de basiskaart.

Van interface-kaart naar buskaart

Openbaar vervoer naar hogere adressen

Dit stukje is alleen voor hen bestemd die de bus nemen. Besluit men (voorlopig) niet tot buskaartgeheugen, dan kan men het overslaan of ter oriëntatie en ter bezinning lezen.

De busverbinding is geschetst in de figuren 20b en 20c. Als buskaart kan

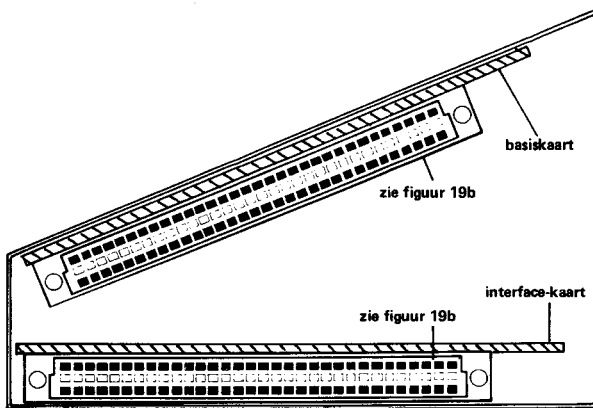
men de al genoemde print EPS 80024 (figuur 21) nemen (met één konnektor figuur 19a en één à vijf konnektoren figuur 19b). Of men maakt zelf een buskaart van experimenteerprint. Men kiest dan zelf het aantal geheugenkaartkonnektors en men legt zelf de verbindingen. Let ook nu bij gebruik van de "echte" buskaart EPS 80024 op het asymmetrische koperbanenpatroon. In de al eerder omschreven sandwich-positie zitten de punten 32 links en de punten 1 rechts.

Voor alle 64-polige konnektors van de sandwich geldt dat de a-pennerij het dichtst bij de rand van de print zit. Dat is in overeenstemming met wat er op de konnektors staat gedrukt. Bij de buskaart is het net andersom: de c-pennerij zit het dichtst bij de rand. Het is hierom dat een leiding met een a-aansluiting op de ingangskonnektor een c-aansluiting op de uitgangskonnektor heeft, en omgekeerd (zie figuur 1).

N.B. Alle aanduidingen a en c in figuur 19 horen bij het opschrift op de konnektor en dus in het geval van de buskaart niet bij de op die print vermelde a en c!

Verbindt men interface- en buskaart via 64 draadjes, dus zonder konnektors, dan zijn de a-draadjes even lang als de c-draadjes (figuur 20d). Gebruikt men een konnektorverbinding, dan zijn er twee mogelijkheden (figuren 20b en 20c), die samenhangen met de al besproken keuze voor de uitgangskonnektor. "Op de kaart" versus "tegen de kaart". Let op het hoogteverschil: in het geval: figuur 20b liggen de kaarten niet in elkaars verlengde, in het geval: figuur 20c wél.

Stel dat in het kader van een bepaalde kastkeuze de basiskaart schuin oploopt. De interface-kaart is dan eigenlijk wel gedwongen om ook schuin te gaan (of je moet een verticale verbindingsprint maken waarbij de beide



Figuur 22. Bij gebruik van een lessenaarkast, bijvoorbeeld de voor de standaard-junior-computer gebruikte kast, kan men overwegen om gebruik te maken van een verbindingsprint basiskaart-interface-kaart, waarbij de beide konnektors niet parallel aan elkaar zijn gemonteerd. Let wel, dit is geen klusje dat iedereen zo maar even doet. In verband met de schuine konnektor kan men geen gebruik maken van experimenteerprint met een 2,54 mm-gaatjesraster.

vrouwjes niet evenwijdig aan elkaar staan, zie figuur 22). De buskaart houden we, mede gezien de bevestiging graag horizontaal (of vertikaal, maar dat komt direkt). Een busverbinding volgens figuur 20e is dan de enige oplossing; met konnektors lukt het niet, tenzij men een "verleng-snoer" maakt van twee konnektors en flat cable.

Naast een horizontale is een vertikale stand van de buskaart mogelijk. Onder handhaving van de positie van de pennen 1 en 32 moet de buskaart met de andere kant worden aangesloten. Zie figuur 20f. Deze variant is interessant als men de voorkeur geeft aan een lagere maar diepere kast. Dit omdat een buskaart korter is dan een geheugenkaart. Een schuine variant à la figuur 20e is ook hier mogelijk. Al met al zijn er nogal wat situaties waarbij dat wellicht wat moeilijke gedoe met die uitgangskonnektor komt te vervallen. Bijvoorbeeld omdat men is aangewezen op draadverbindingen.

Aktiepunt vijf: Onderdak

De behuizing van de uitgebreide junior-computer

Net zoals je huizen hebt met een losse schuur en garage zijn er woningen waar alles onder één dak zit. Iets dergelijks kan optreden bij de uitgebreide, of misschien zelfs al bij de standaard-junior-computer. De voeding bijvoorbeeld neemt relatief veel plaats in. Waarom dan niet de voeding in een aparte kast zetten? De eigenlijke kast blijft dan lekker klein. Dat is één mogelijkheid.

Andere mensen zweren met even veel recht van spreken bij één kast met alles erin. Dat die kast dan wat groter moet zijn neemt men op de koop toe. Kompaktheid voor alles, en geen gedoe met verbindingsdraden die je kunt verwisselen.

Met de uitbreidingen neemt het aantal "woningen" naar alle waarschijnlijkheid sterk toe: cassette-recorder(s), groot toetsenbord, video-terminal plus tv-toestel. Het is natuurlijk mogelijk om een monsterverbond te sluiten en werkelijk alle onderdelen van het systeem in één grote kast onder te brengen. Waarschijnlijker is het dat er sprake is van een paar kasten. En als er toch al een paar zijn maakt het ook niet zoveel meer uit als je de eigenlijke uitgebreide junior-computer verdeelt over meerdere kasten, waarbij de voeding en of het buskaartgeheugen apart onderdak krijgen. Dat laatste is met name aantrekkelijk als daardoor de bestaande kast voor de standaard-junior-computer kan blijven gehandhaafd!

Goed, er zijn dus twee hoofdmogelijkheden, met voor beide mogelijkheden allerlei varianten. We beperken ons hier tot de eigenlijke uitgebreide junior-computer, dus de randapparatuur blijft buiten beschouwing. Er zijn in overeenstemming met voornoemde hoofdmogelijkheden twee kasttypen denkbaar: het lessenaarmodel en het versterkerkastmodel. Het **versterkerkastmodel** (figuur 23b) is zo genoemd omdat het er net zo uitziet als de indertijd populaire behuizingen voor audioversterkers, uitgerust met elektronenbuizen ("vertel nog eens van vroeger, pappa"). De kasten bestaan nog steeds. Is uitstekend geschikt voor de alles-in-één-kast-aanpak. Onder het schuine bedieningspaneel (nog beter is een vlak bedieningspaneel; dan kunnen de interface-kaart en de buskaart worden doorverbonden via konnektors) komt de kaarten-sandwich en in het hoge deel het buskaart-

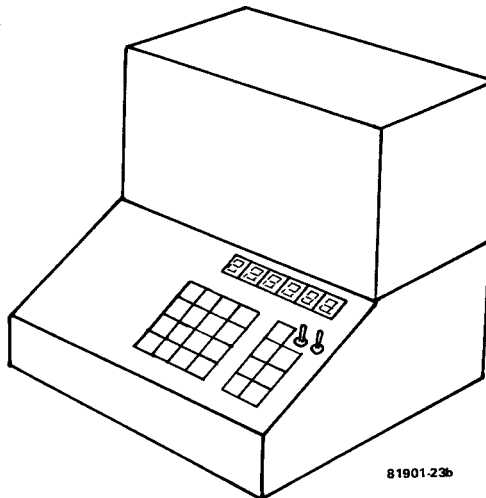
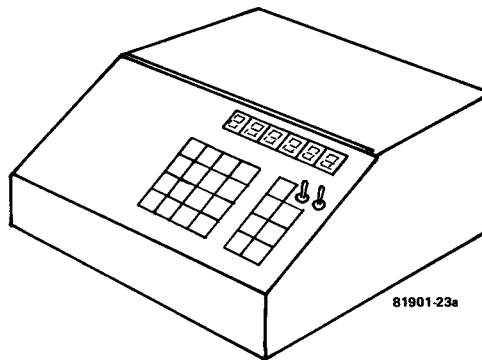
geheugen (links) en de uitgebreide voeding (met twee trafo's; rechts). Het **lessenaarmodel** (figuur 23a) zou in de meeste gevallen wel eens al toegepast kunnen zijn voor de standaard-junior-computer, en dan wellicht niet precies zoals in figuur 23a. Dit kasttype is in het algemeen wat slanker en kompakter van lichaamsvorm dan de versterkerkast en leent zich het beste voor de aparte-kasten-aanpak, hetzij omdat men een overtuigd aanhanger is van die "school", hetzij omdat men de bestaande standaardkast in gebruik wil houden en omdat de uitgebreide voeding er toch niet meer in past. En dat allemaal nog los van de vraag of men buskaartgeheugen neemt of niet.

Een paar overwegingen bij uw huisvestingskeuze:

1. De al eerder beschreven sandwich van interface- en basiskaart verdient sterk de voorkeur boven andere oplossingen, zoals bijvoorbeeld beide kaarten naast elkaar.
2. Het is technisch mogelijk en dus verantwoord om verbindingen tussen kaarten (basiskaart ↔ interface-kaart; interface-kaart ↔ buskaart) te maken via flat cable met een lengte van maximaal 50 cm.

MAAR WÉÉT WAARAAN U BEGINT! NIET VOOR BEGINNENDE SOLDEERARTIESTEN!

3. Kiest men voor een lessenaarkast (eventueel de bestaande standaardkast) én voor buskaartgeheugen, dan kan men de uitgangskonnektor op de achterkant van de kast monteren en daarop direkt of via een "verlengsnoer" van konnektors en flat cable de buskaart aansluiten (die in het flat cable-geval wellicht in een apart kastje is ingebouwd).
4. Zorg voor een stevige mechanische bevestiging van de diverse kaarten, ook onderling, zoals bij de kaartensandwich. Laat de kaarten nooit uitsluitend via konnektors aan elkaar hangen. Als bijvoorbeeld de buskaart in de achterkant van de kast wordt gestoken (zie punt 3), zorg dan voor pootjes met de juiste lengte. Pas bij het gebruik van afstandsbusen en dergelijke op voor kortsluitingen van printsporen tegen het metaal, waardoor nog los van de vraag of dat metaal een gedefinieerde spanning voert (bijvoorbeeld nul volt via IC2 van de voeding; zie eerder), er onderlinge kortsluitingen zouden kunnen optreden. Is de situatie van figuur 20b van toepassing, vang dan het hoogteverschil tussen de beide kaarten op met op maat gezaagde afstandsbusen, opvullingen of iets anders in die geest.
5. Tot slot een soort "checklist" van allerlei dingen die op een kastwand zitten of erdoorheen steken, en die een rol spelen bij de keuze van uw huisvestingsaanpak.
 - a. Hexadecimaal toetsenbord.
 - b. Zesdelig 7-segmentdisplay. Blijft net zoals a gewoon nodig, ook als men veel gaat doen met ASCII-toetsenbord en een video-terminal of printer.
 - c. STEP-schakelaar S24.
 - d. Display-schakelaar S25. Deze schakelaar en die van c kan men eventueel van de basiskaart afhalen en elders plaatsen. Bijvoorbeeld om de kaartensandwich dunner te maken; zo dun dat hij in de bestaande kast past.
 - e. Bestaande 31-polige poortkonnektor (PIA). In verband met de komst van de VIA-konnektor kan men deze laten vervallen (hoeft niet perse).
 - f. Uitbreidingskonnektor. Misschien had u die al ergens op de standaardkast gemonteerd en misschien zijn er in uw geval mogelijkheden om ook

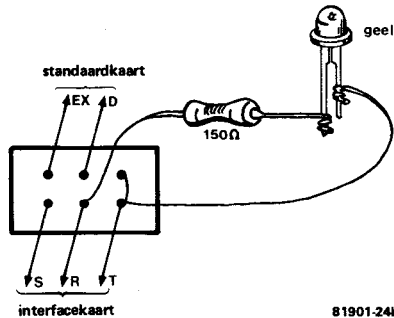
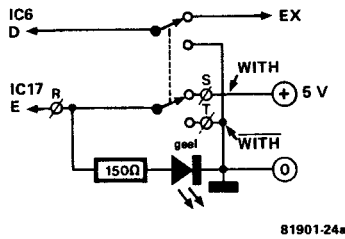


Figuur 23. Twee hoofdtypen kast voor de behuizing van (een deel van) de uitgebreide junior-computer. Het hangt ervan af of u kiest voor "alles in één kast" of voor in meerdere kasten welk type voor u geschikt is. Er zijn allerlei varianten van beide typen mogelijk/denkbaar/te maken. Vooral een versie van de versterkerkast met vlak verlopend bedieningspaneel biedt voordelen.

de interface-kaart daarop extern aan te sluiten (aparte kast, met of zonder buskaartgeheugen). Dat zou wél betekenen dat het principe van de kaartensandwich, waar onze voorkeur naar uitgaat (zie punt 1), zou moeten worden verlaten.

- g. Doorvoer netsnoer (of euro-entree), netschakelaar en zekeringhouder. Op de voedingskast of op de "kast". En denk erom: 2A-zekering! (dat waren de "standaarddingen". Nu de nieuwe gevallen)
- h. RS 232-konnektor. Op de zijkant en direkt op de print gemonteerd of

- op de achterkant, en met de print verbonden via flat cable. Let dan goed op de juiste aansluitingen; de pinning 1...25 is gebaseerd op montage van de 25-polige D-konnektor op de koperzijde. Bij gebruik van de Elekterminal kunnen de doorverbindingen (draadbruggen) tussen pennen (zie bijvoorbeeld figuur 1) achterwege blijven.
- i. Chassisdelen voor de verbindingen met de cassette-recorder. Achterop lijkt ons het handigste. Een DIN-chassisdeel in plaats van J1 en J2 mag ook; J3 en J4 zijn alleen nodig bij afstandsbediening. Men kan eventueel ook gebruik maken van onderbreekbare draadverbindingen door J1...J4 te vervangen door vrouwtjes-cinch-stekers.
 - j. Cassette-LEDs. Op een in het oog lopende plaats, want daar zijn ze voor. Monteer ze gedeeltelijk in een gat dat is voorzien van een kleine rubber-ring (tule).
 - k. Twee net-indikatie-LEDs. Op de voedingskast of op de junior-computerkast. Zie ook j.
 - l. Vijf bussen voor de aansluiting (via stekers) van de voedingsspanningen, inclusief massa. Alleen bij een voeding in een aparte kast. Bezuinig niet op de dikte van de verbindingsdraden, vooral niet voor de nul (massa) en de +5 volt. Soepel draad 1 mm² (één millimeter-kwadraat) is prima. Gebruik verschillende kleuren draad in verband met het gevaar voor verwisseling. Het kan geen kwaad om alle verbindingpunten ter plaatse van de stekerbussen te ontkoppelen naar het nulpunt met behulp van tantaal-elko's van 1 µ/16 V; let op de juiste polariteit!
- Er zijn vijf centrale aansluitpunten op de interface-kaart voor alle voedingsspanningen, inclusief massa. De basiskaart en de geheugenkaart(en) (via de buskaart) worden gevoed via de diverse konnektors.
- N.B. In wezen komt dit punt 1, indien van toepassing, twee keer voor: één keer op de voedingskast en één keer op de juniorkast.
- m. VIA-konnektor. Komt in de plaats van de oude PIA-poort-konnektor of bij die konnektor. Afhankelijk van uw applicatieplannen kan hij achterop de kast worden gezet of ergens in het zicht, op het bedieningspaneel.
 - n. Uitgangskonnektor. Uitsluitend indien buskaartgeheugen extern wordt aangesloten; hetzij direkt (buskaart insteken), hetzij via flat cable + geschikte mannetjeskonnektor.
 - o. Schakelaar WITH/WITH. Niets anders dan een dubbelpolige schakelaar waarmee men kan kiezen tussen de draadbruggen R-S & D-EX en R-T & D-L. Zie figuur 24. Dit is zinvol als buskaartgeheugen extern wordt aangesloten en als men in sommige gevallen van het gebruik ervan afziet. Dus in het geval van een keuze: buskaartgeheugen, ja of nee, bij voorhanden zijnd buskaartgeheugen. In het WITH-geval licht een gele LED op: LED in het zicht s.v.p. Neem eventueel hiervoor S25 (display-schakelaar; zie d en vervang de laatste door een permanente draadverbinding.



Figuur 24. Een schakelaar waarmee kan worden gekozen tussen met of zonder buskaartgeheugen is handig in het geval dat buskaartgeheugen voorhanden is, maar eventueel kan worden afgekoppeld, dus met name in het geval dit geheugen op de achterkant van de kast wordt aangesloten: direct of via flat cable. Eventueel kan men hiervoor de displayschakelaar S25 nemen — welke laatste dan wordt vervangen door een permanente doorverbinding.

Dat was dan de bouw van de uitgebreide junior-computer. We kunnen niet de pretentie hebben dat het om een super-volledige "fool proof" bouwbeschrijving gaat. Dat komt ondermeer omdat u als bouwer een groot aantal keuzemogelijkheden hebt. Er zijn waarschijnlijk net zo veel verschillende bouwplannen als bouwers. Laten we nu eens gaan kijken of het allemaal werkt en of u na uw scheppende arbeid kunt zeggen dat het goed is, en in overeenstemming met de vele Elektuur-prototypes en met wat erover op papier is gezegd in dit hoofdstuk. Dus of het in overeenstemming is met "onze" scheppende arbeid.

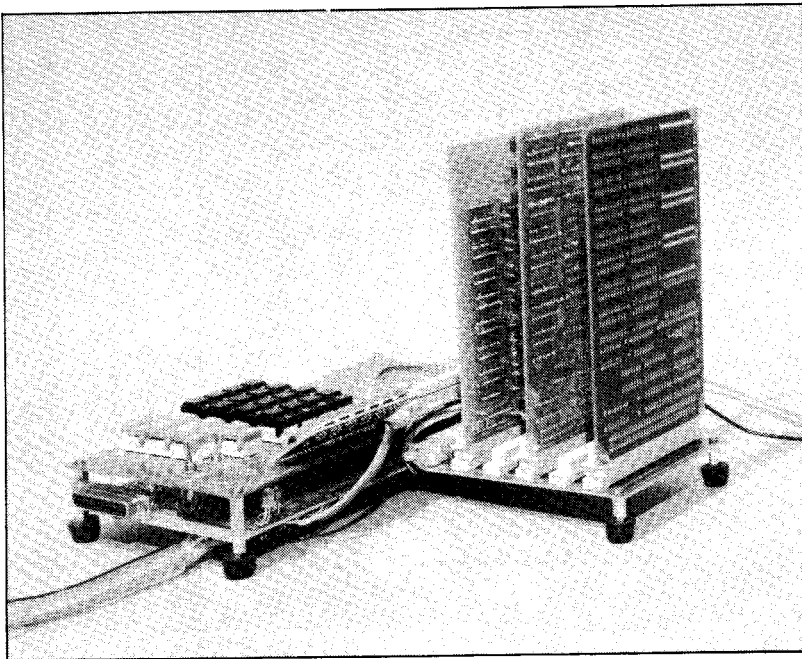


Foto 1. Een uitgebreide junior-computer, gezien vanuit "noordoostelijke" richting (oriëntatie op basis van een logisch gebruikersstandpunt; de gebruiker heeft tijdens de bediening van de toetsen het gezicht naar het noorden gericht). De RS232-konnektor is direkt op de interface-kaart gemonteerd, evenals de INPUT- en OUTPUT-LED.

Aktiepunt zes: Ingebruikname — het liefst vrijwel direkt en, als dat jammer genoeg niet het geval is, fouten opsporen en verhelpen

Oftewel: hoe krijgt men de vergrote junior zo snel mogelijk aan de praat, en dan het liefst zonder spraakles.

We hebben de titel en ondertitel van dit (laatste) aktiepunt lekker lang gemaakt als compensatie voor uw hopelijk korte "verblijf" in dit tekst-gedeelte. De controle vooraf omvat twee aspecten: hardware-georiënteerde controle en software-georiënteerde controle. En denk daarbij maar weer aan de mens-computervergelijking in de vorm van "hardware ↔ vlees en bloed; software ↔ geest". Is het bij de mens zo dat ziek/gezond zijn niet automatisch betekent dat hij geestelijk ziek/gezond is, bij de wat minder gekompliceerd in elkaar zittende computer zijn er wel bepaalde konklusies te trekken. Als bijvoorbeeld na een software-test blijkt dat nieuw RAM-geheugen kan worden beschreven en gelezen, dus dat de software-kant goed zit, dan mag je daaruit konkluderen dat de verwante hardware in orde is. Door een slimme keuze en volgorde van software-tests kan men via deductie een eventuele fout op een steeds kleiner wordend gebied lokaliseren: het fout(en)-indammen.

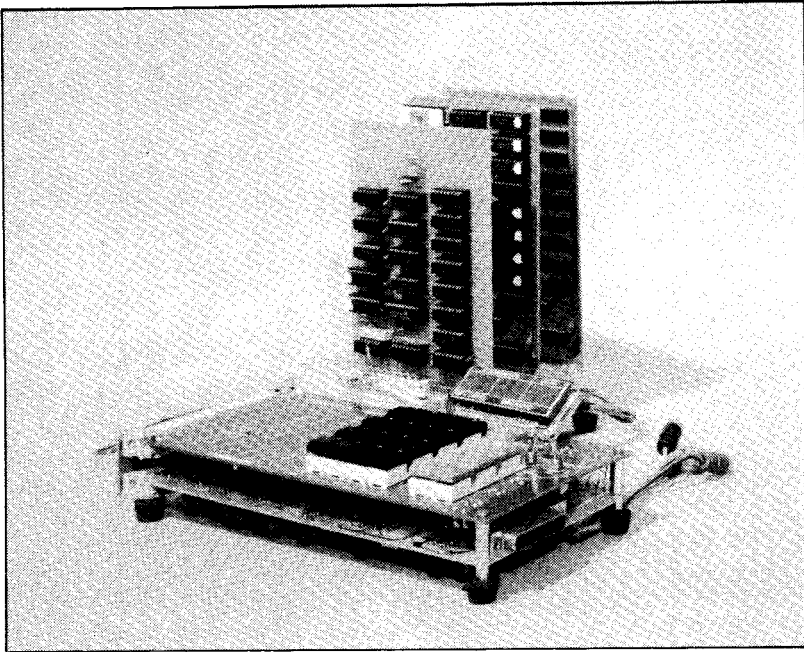


Foto 2. Als foto 1, maar nu vanuit zuidoostelijke richting. Merk op dat de "koper-zijde" van de interface-kaart boven ligt. Voor de aansluitingen op de cassette-recorder(s) is gebruik gemaakt van vrouwelijke cinch-stekers. De dichtstbij staande geheugenkaart is een prototype (op "gaatjesprint") van de 16K-dynamische RAM-kaart.

Maar laten we eerst maar eens de hardware-georiënteerde controle behandelen. Dat betreft allerlei controles voordat de voedingsspanningen (korrekt!) zijn aangesloten en dus lang voordat junior wordt ingeschakeld. Indien men de junior-computer tevens met buskaartgeheugen heeft uitgebreid verdient het aanbeveling om de tekst te beginnen zonder buskaartgeheugen. Men moet dan tijdelijk de situatie *WITH* realiseren, bijvoorbeeld met een schakelaar (zie figuur 24). Als e.e.a. in orde is kan men het systeem vervolgens met buskaartgeheugen (*WITH*) testen.

Een aantal algemene controlepunten, genoemd op de pagina's 34 en 35 van boek 1, is hier en nu onverminderd van toepassing. Er van uitgaand dat u eerder al de basis-junior-computer heeft gebouwd kunt u met voordeel gebruik maken van uw indertijd opgedane bouwkennis en andere ervaringen ("laat ik voortaan daar en daar goed op letten"). Nu wat algemene controlepunten:

1. **Verbindingen.** Het kon best wel eens zo zijn dat de kans op foute of ontbrekende verbindingen toeneemt volgens het kwadraat of misschien wel de derde macht van het aantal verbindingen, soldeerpunten, konnektorpennen, enzovoorts. Omdat de uitgebreide junior-computer

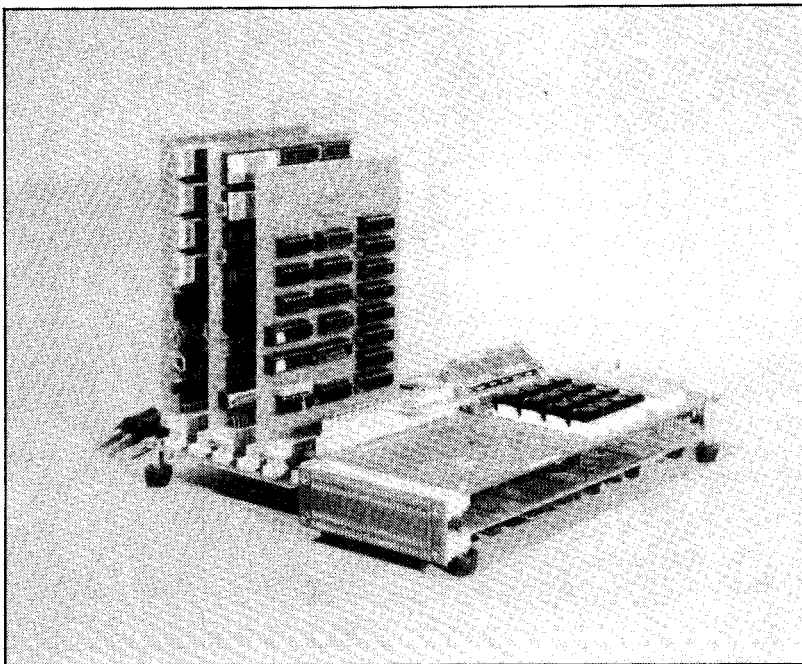


Foto 3. Als foto 1, maar nu vanuit zuidwestelijke richting. De verbindingen tussen de uitbreidingskonnektor van de standaardkaart en de ingangskonnektor van de interface-kaart lopen via een stukje buskaart EPS 80024 (zie figuur 21b). Tevens ziet men de aansluitingen op de poortkonnektor van de standaardkaart, via een eigenlijk voor printmontage bestemde 31-polige konnektor.

een sterk vergroot aantal verbindingen kent staat de controle op ontbrekende of foute (kortsluiting, aansluitdraden verwisseld) verbindingen nummer één op de controle-hitparade. Test de verbindingen met een universeelmeter die op ohmmeting is ingesteld, of met de "verbindingstester" die is gepubliceerd in de juni 1981-uitgave van *Elektuur*. Zijn van alle IC's alle pennen goed aangesloten? Geen aansluitdraden verwisseld of met verkeerde punten verbonden? Voeren de mannetjes-plus vrouwtjeskonnektors hun verbindingswerk wel goed uit? Alle programmeerbare draadbruggen juist gelegd?

2. Let op de juiste positie van al die onderdelen die weliswaar op twee manieren kunnen, maar slechts op één manier geplaatst mogen worden. Zoals: IC's, elko's, dioden, transistoren, de 25-polige D-konnektor (RS 232). Liever een keer te veel gecontroleerd dan te weinig.

U heeft werkelijk alles gecontroleerd wat er vooraf te controleren viel. U heeft misschien een kollega-bouwer (bijvoorbeeld van de gebruikers-club) er, onafhankelijk van u, naar laten kijken. Het wordt tijd om de voedingsspanningen aan te sluiten, nadat de voeding apart is getest (zie het bouwverhaal onder "voeding"). U sluit de voeding aan op de interface-

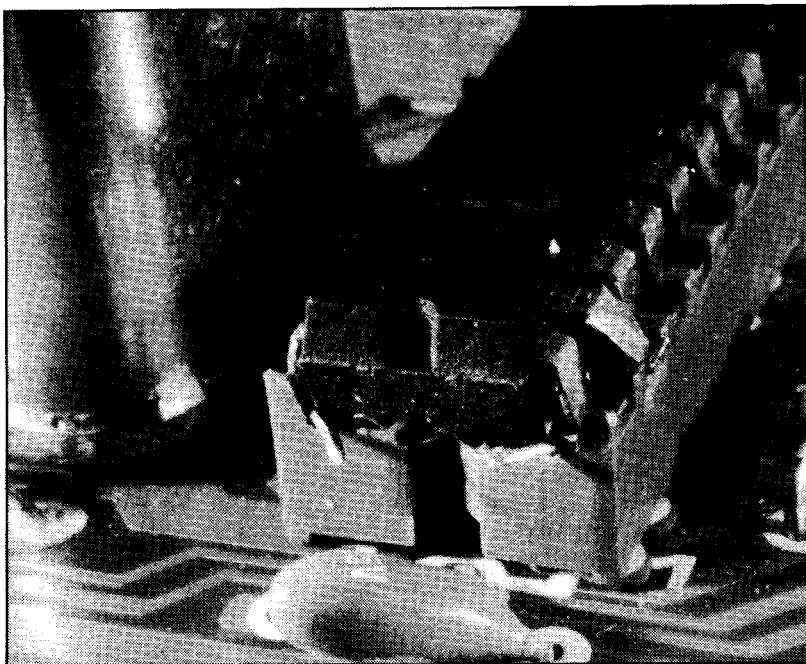


Foto 4. Dit bedoelen we nou met: "niet alleen de positie van het IC goed, maar ook de positie van alle aansluitpennen". Verder commentaar lijkt ons overbodig.

kaart met vijf verschillend gekleurde soepele draden. De massa- en +5 V-leidingen zijn minstens één millimeter-kwadraat dik. Doet men dat niet, dan kan men buitengewoon gekke dingen verwachten; niet alleen gek maar ook vervelend, laat daarover geen misverstand bestaan. **De vijf aansluitingen zijn grondig gecontroleerd!**

Voordat u de junior-computer inschakelt nog een laatste moment van persoonlijke bezinning: is **werkelijk alles grondig gecontroleerd**? Zeker weten? Okay, dan moet u het zelf weten. De junior-computer wordt ingeschakeld. Men kan vervolgens twee dingen doen: U bent ongeduldig en u drukt de RST-toets in. Als de displays oplichten betekent dat dat het standaardgedeelte van de uitgebreide junior-computer werkt. Of u bent geduldig en u meet allerlei spanningen op IC-pennen aan de hand van de informatie van figuur 18. **Meet die spanningen op de IC-pennen en niet op de soldeerpunten aan de koperzijde!** Waarom? Omdat dan meteen een eventueel falend IC-kontakt wordt ontdekt. Pas op dat er geen kortsluiting met aangrenzende IC-pennen ontstaat via de meetpen. Dan kan het middel erger zijn dan de kwaal. Na het indrukken van RST moet dus het display oplichten. Is dat niet zo en heeft u de standaardkaart al eerder opgebouwd

en goed bevonden, dan rijzen vragen als: Krijgt de standaardkaart wel voedingsspanningen? Hoe zit het met de draadbrug rond punt D van IC6? Naar massa zonder buskaart en naar EX met buskaart. Is in het laatste geval EPROM aangesloten op pagina FF en is de RESET-vektor (adressen FFFC en FFFD) wel gespecificeerd (FFFC:1D; FFFD:1C)? Draadbrug R-S aangebracht? Doet de standaard-ic het wél als tijdelijk de draadbruggen D-massa en R-T worden gelegd? Of als de kaart wordt losgenomen en als vanouds via tijdelijke voedingsverbindingen wordt beproefd? Controleer nog eens met name de interface-hardware rond de adres-dekodering en de databussturing, inclusief de PROM IC17. Is die trouwens wel geprogrammeerd?

Zodra de werking van de standaard-monitor in orde bevonden is hebben we een goed stuk gereedschap in handen om verdere tests te verrichten: de monitor! Bijvoorbeeld voor het testen op goede werking van de nieuwe RAM: in ieder geval IC2 & IC3 en eveneens in het geval van een 8114 voor IC4 en/of IC5. Wat doet men namelijk? Men kiest een paar adressen die binnen het bijbehorende RAM-adresblok zitten. Toetst één zo'n adres in via AD . Er verschijnt data op de rechter twee displays. Men geeft nu via DA data op die anders is dan de zichtbare data. Verschijnt vervolgens de nieuwe data in beeld, dan is met zekerheid vastgesteld dat de betreffende RAM-geheugenplaats goed wordt beschreven en gelezen (zie hoofdstuk 7 van boek 2). Herhaal dit voor een paar andere adressen binnen het adresblok (bijvoorbeeld via +), en voor alle andere RAM-adresblokken.

Bij EPROMS is het zo dat ze alleen maar goed hoeven te worden gelezen. Het schrijven komt overeen met het programmeren ervan. Zit er TM in IC4 en PM in IC5, dan kan men via de monitor kijken of er goed wordt gelezen. De hex dumps van TM en PM vindt men achterin dit boek.

Ja, en dan moeten nog gecontroleerd worden: de VIA, de cassette- en de RS 232-interface. Voordat de cassette-hardware bedrijfsklaar is moet eerst nog de PLL worden afgeregeld, met de precisietimmer P1. Daarover meer in hoofdstuk 11. De VIA komt pas in boek 4 aan bod en over de juiste werking van de RS 232-hardware kan pas iets zinnigs worden gezegd als de bijbehorende randapparatuur is gebouwd en aangesloten. En dat is pas na hoofdstuk 12 het geval.

De uitgebreide junior-computer is nu gebouwd en getest. Voor bepaalde aspecten van de ingebruikname is men aangewezen op hoofdstuk 11 of 12. Gebruikt men buskaartgeheugen, dan wil men wel graag weten hoe het zit met die RAM/EPROM-geheugenkaart. Hoe maak je die kaart geschikt voor gebruik als buskaartgeheugen? Ziedaar de inhoud van het nu volgende derde en laatste deel van dit hoofdstuk.

De RAM/EPROM-geheugenkaart

Bytes-buitenwijken die via de bus bereikbaar zijn

De RAM/EPROM-(geheugen)kaart is evenals de 16K-dynamische RAM-kaart — die niet in dit boek zal worden besproken — een bouwsteen waarmee maximaal 56K buskaartgeheugen (pagina's 20... FF) kan

worden aangesloten op de uitgebreide junior-computer. Daarnaast kan men één zo'n geheugenkaart direkt op de standaard-kaart aansluiten: een geheugen-buitenwijk dus waarvoor men niet de bus hoeft (sterker: mag) te pakken en waarvoor een kort ritje via de uitbreidingsbus volstaat. Voor deze laatste mogelijkheid: zie Aanhangsel 1.

Deze geheugenkaart is als bouwproject met print (EPS 80120) besproken in een artikel in het tijdschrift *Elektuur*, september 1980. Het is niet de bedoeling om dat artikel in dit boek te gaan herkauwen. Wél is het zinvol om een aantal aanvullende en speciaal voor de junior-situatie van belang zijnde gebruikersaspecten te bespreken. En wat de EPROMs betreft: op de kaart is plaats voor óf 1 à 4 2708's, óf 1 à 4 2716's, óf 1 à 4 2732's. Dus geen typen doorelkaar gebruiken! Met de prijs van de 2732 (4K) is het nog steeds buitengewoon treurig gesteld. In deze "junior-RAM/EPROM-special" zal er geen aandacht aan worden besteed. De 2716 daarentegen gaat het helemaal maken. Hij heeft twee keer zoveel in huis als de 2708 (1K), is minder dan twee keer zo duur als die 2708, is veel eleganter en makkelijker te programmeren en heeft geen -5 V en +12 V nodig.

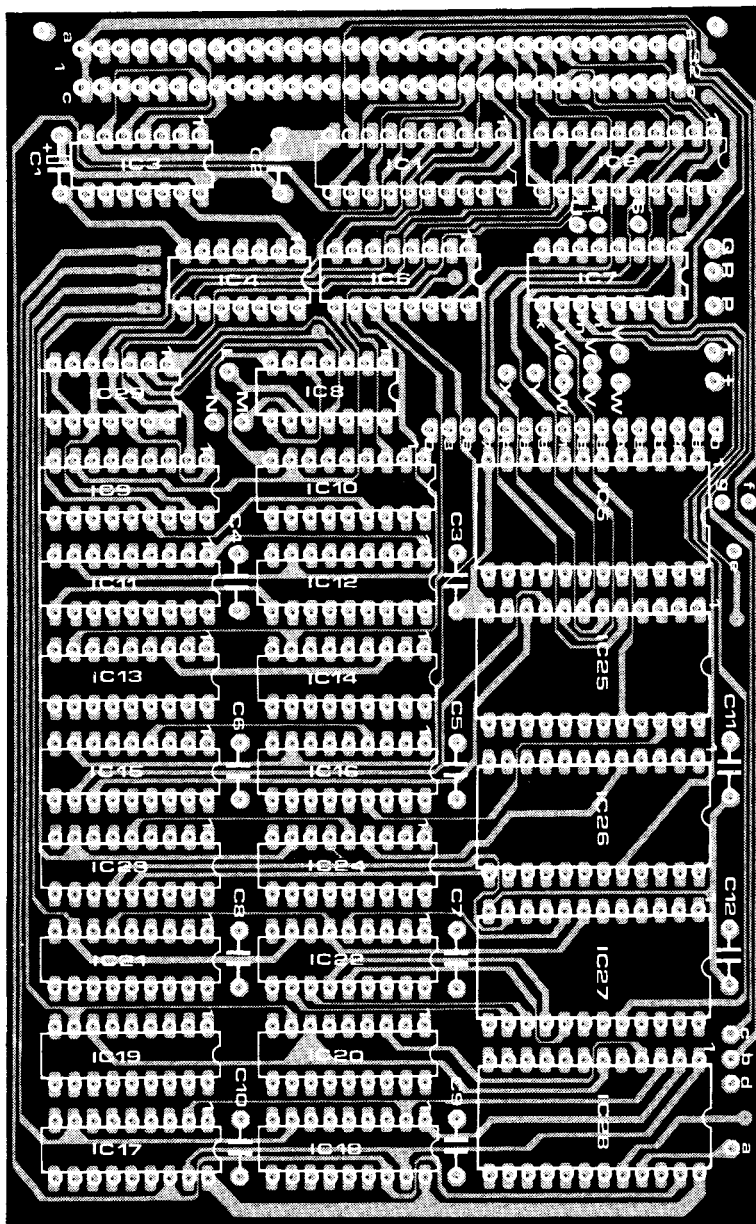
Het schema van de RAM/EPROM-kaart staat in figuur 25. In figuur 26 staat de print EPS 80120 (eurokaartformaat: 100 x 160 mm), en dan alleen de componentenzijde met bijbehorende opdruk plus het koperbanenpatroon van de koperzijde. Alle andere printgegevens staan in het *Elektuur*-artikel van september 1980. In figuur 27a staan de IC-plattegronden en de interne schema's van de dekodeer-IC's IC5, IC6 en IC7. Figuur 27b toont de waarheidstabellen van de hoofddekodeer IC5 en van de secundaire dekoders IC6 en IC7.

Aanvullende gegevens van de RAM/EPROM-kaart

Puntjes op de "i" en extra "i's"

De databuffers van de RAM/EPROM-kaart zijn in de schrijfrichting actief als er geen geheugen op de kaart is geadresseerd en, uiteraard, als er data moet worden geschreven in RAM. De databuffers zijn alleen dan in de leesrichting actief als op de kaart aanwezige (dat wil zeggen: gedekodeerde!) RAM of EPROM wordt gelezen. Er treden bij juist gebruik dus geen gevaarlijke toestanden à la figuur 6 op.

In tabel 9 is de werking van de hoofdadresdekodeer, IC5, samengevat (zie ook figuur 27b). Het gaat daarbij om de opdeling in zestien blokken van elk 4K. Die blokken worden in het september-artikel "pagina's" genoemd. Verwarrend, want bij "ons" is een pagina FF oftewel 256 bytes groot. Bij IC5 moet men er goed op letten dat er IC-pennummers zijn, en printaansluitpunten (Ø . . . F) en dan ook nog IC-gerichte funktionele aanduidingen (zie de plattegrond van IC5 in figuur 27a). Een bijdrage aan de mogelijke verwarring wordt verder geleverd door het feit dat de adreslijnen A12 . . . A15 niet op de meest logische wijze zijn aangesloten op IC5. Uit tabel 9 blijkt verder dat voor een 4K-blok alleen het meest linkse nibble van het adres vastligt; (ØXXX . . . FXXX); de andere drie (X) kunnen elk variëren van Ø tot en met F. De sekundaire adresdekodering voor de RAM gaat via IC6 die wordt gestuurd uit drie lagere adreslijnen en uit N2. Het gaat om maximaal 8K-RAM. Vandaar de acht uitgangen van IC6 (zie ook



Figuur 26. De print van de RAM/EPROM-kaart, voor zover het gaat om de componentenopdruk plus het koperbanenpatroon van de koperzijde.

Tabel 9. Hoofddekodering RAM/EPROM-kaart (IC5).

A15 (A)	A14 (B)	A13 (C)	A12 (D)	adres (hex)	welke uitgang laag?	
					print- aansluit punt	IC5-pen
0	0	0	0	0XXX	0	1
0	0	0	1	1XXX	1	9
0	0	1	0	2XXX	2	5
0	0	1	1	3XXX	3	14
0	1	0	0	4XXX	4	3
0	1	0	1	5XXX	5	11
0	1	1	0	6XXX	6	7
0	1	1	1	7XXX	7	16
1	0	0	0	8XXX	8	2
1	0	0	1	9XXX	9	10
1	0	1	0	AXXX	A	6
1	0	1	1	BXXX	B	15
1	1	0	0	CXXX	C	4
1	1	0	1	DXXX	D	13
1	1	1	0	EXXX	E	8
1	1	1	1	FXXX	F	17

figuur 27b) en de twee ingangen van N2, welke laatste selectie-informatie ontvangen uit de hoofddekoder IC5 (aansluitingen X en Y). Tenminste: als het om meer dan 4K-RAM gaat, want anders volstaat één doorverbinding. De sekundaite adresdekodering voor de EPROM vindt plaats via IC7 (zie ook figuur 27b). Ook deze wordt type-afhankelijk gestuurd uit twee lagere adreslijnen en uit N1, die in verbinding staat met IC5, via de aansluitingen V en/of W.

Men heeft ontelbare keuzemogelijkheden bij het kiezen van bepaalde plaatsen op de geheugenkaart (hier is de "memory map" bedoeld) voor RAM en voor EPROM. Is daar nou niet een beetje systeem, dus ordening in te brengen? Jazeker! Uitgaand van maximaal 56K buskaartgeheugen, namelijk de adressen 2000 ... FFFF (dus de pagina's 20 ... FF) is de nu volgende keuze logisch, handig en overzichtelijk:

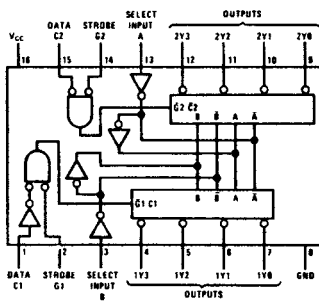
RAM: Begin bij adres 2000 en ga al naar gelang de behoefte min of meer ver in de richting van toenemende adressen

(dus lager op de geheugenkaart);

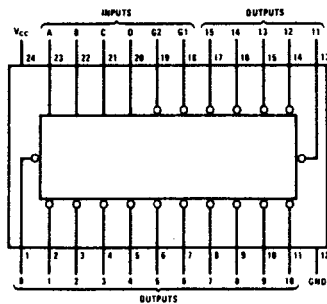
EPROM: Begin bij adres FFFF en ga al naar gelang de behoefte min of meer ver in de richting van afnemende adressen

(dus hoger op de geheugenkaart).

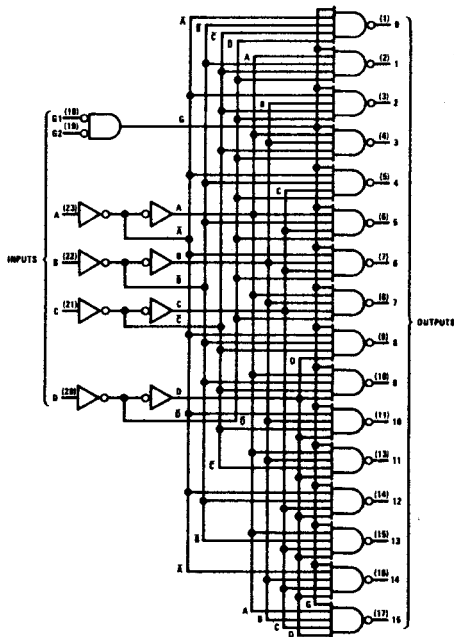
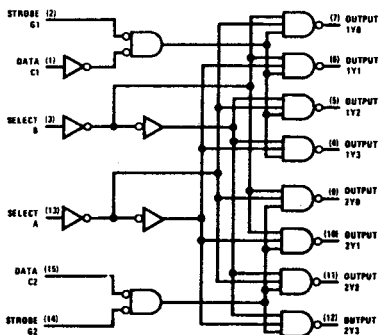
Let wel: dit concept geldt onverminderd voor het gebruik van meerdere geheugenkaarten. De keuze is ondermeer gebaseerd op het feit dat op pagina FF EPROM aanwezig moet zijn met vektor-data. Laten we op basis van deze twee uitgangspunten maar eens gaan kijken hoe we de RAM en de EPROM moeten dekoderen (dus welke plaats op de "memory map"), afhankelijk van de hoeveelheid RAM/EPROM-kaarten en, wat de RAM betreft, voorlopig afgezien van de mogelijkheden van de 16K-dynamische RAM-kaart.



IC6, IC7
74155



IC5
74154



81901-27a

Figuur 27a. Plattegronden en interne schema's van de dekodeur-IC's van de RAM/EPROM-kaart.

IC5

INPUTS					OUTPUTS																
G1	G2	D	C	B	A	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
L	L	L	L	L	L	L	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H
L	L	L	L	L	H	H	L	H	H	H	H	H	H	H	H	H	H	H	H	H	H
L	L	L	L	H	L	H	H	L	H	H	H	H	H	H	H	H	H	H	H	H	H
L	L	L	L	H	H	H	H	L	H	H	H	H	H	H	H	H	H	H	H	H	H
L	L	L	H	L	L	H	H	H	H	L	H	H	H	H	H	H	H	H	H	H	H
L	L	L	H	L	H	H	H	H	H	L	H	H	H	H	H	H	H	H	H	H	H
L	L	L	H	H	L	H	H	H	H	L	H	H	H	H	H	H	H	H	H	H	H
L	L	L	H	H	H	H	H	H	H	L	H	H	H	H	H	H	H	H	H	H	H
L	L	H	L	L	L	H	H	H	H	H	H	L	H	H	H	H	H	H	H	H	H
L	L	H	L	L	H	H	H	H	H	H	H	L	H	H	H	H	H	H	H	H	H
L	L	H	L	H	L	H	H	H	H	H	H	L	H	H	H	H	H	H	H	H	H
L	L	H	L	H	L	H	H	H	H	H	H	L	H	H	H	H	H	H	H	H	H
L	L	H	L	H	H	H	H	H	H	H	H	L	H	H	H	H	H	H	H	H	H
L	L	H	H	L	L	H	H	H	H	H	H	L	H	H	H	H	H	H	H	H	H
L	L	H	H	L	H	H	H	H	H	H	H	L	H	H	H	H	H	H	H	H	H
L	L	H	H	H	L	H	H	H	H	H	H	L	H	H	H	H	H	H	H	H	H
L	L	H	H	H	H	H	H	H	H	H	H	L	H	H	H	H	H	H	H	H	H
L	H	X	X	X	X	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H
H	L	X	X	X	X	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H
H	H	X	X	X	X	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H

H = High Level, L = Low Level, X = Don't Care

IC6

3-LINE-TO-8-LINE DECODER OR 1-LINE-TO-8-LINE DEMULTIPLEXER

INPUTS				OUTPUTS							
SELECT			STROBE OR DATA	(0)	(1)	(2)	(3)	(4)	(5)	(6)	(7)
C [†]	B	A	G [‡]	2Y0	2Y1	2Y2	2Y3	1Y0	1Y1	1Y2	1Y3
X	X	X	H	H	H	H	H	H	H	H	H
L	L	L	L	L	H	H	H	H	H	H	H
L	L	H	L	H	L	H	H	H	H	H	H
L	H	L	L	H	H	L	H	H	H	H	H
L	H	H	L	H	H	H	L	H	H	H	H
H	L	L	L	H	H	H	H	L	H	H	H
H	L	H	L	H	H	H	H	H	L	H	H
H	H	L	L	H	H	H	H	H	H	L	H
H	H	H	L	H	H	H	H	H	H	H	L

[†]C = inputs C1 and C2 connected together

[‡]G = inputs G1 and G2 connected together

H = high level, L = low level, X = don't care

IC7

INPUTS				OUTPUTS			
SELECT		STROBE	DATA				
B	A	G2	C2	2Y0	2Y1	2Y2	2Y3
X	X	H	X	H	H	H	H
L	L	L	L	L	H	H	H
L	H	L	L	H	L	H	H
H	L	L	L	H	H	L	H
H	H	L	L	H	H	H	L
X	X	X	H	H	H	H	H

81901-27b

Figuur 27b. De waarheidstabellen van de hoofdadresdekker IC5 en van de sekundaire adresdekoders IC6 en IC7.

RAM

Er moeten worden gewerkt met even aantallen 2114's: één 2114 voor elke 1024 halve bytes (nibbles). In tabel 10 is de voorkeursadressering en -dekodering opgegeven voor hoeveelheden RAM die variëren van 1K tot maximaal 8K. Daarbij gelden de draadverbindingen:

X-2 en Y-+5 V voor 1K ... 4K (gedekodeerde adressen 2000 ... 2FFF) (Y-X mag ook);

X-2 en Y-3 voor 5K ... 8K (gedekodeerde adressen 2000 ... 3FFF).

U bent niet tevreden met 8K-RAM en u wilt meer. Dat kan. Neem een tweede kaart en de totale hoeveelheid RAM varieert van 9K tot en met 16K. Zie tabel 11. Voor de tweede RAM/EPROM-kaart gelden de draadverbindingen:

X-4 en Y-+5 V voor 1K ... 4K extra (gedekodeerde adressen 4000 ... 4FFF) (Y-X mag ook);

X-4 en Y-5 voor 5K ... 8K extra (gedekodeerde adressen 4000 ... 5FFF).

Voor nog meer RAM trekt u er nog een RAM/EPROM-geheugenkaart bij. De printaansluitingen van de hoofddekoder die vervolgens aan de beurt zijn voor doorverbinding, al naar gelang de hoeveelheid RAM, zijn:

6 met X } derde kaart
7 met Y }

8 met X } vierde kaart
9 met Y }

enzovoorts.

Tabel 10. RAM op de eerste kaart (adresbereik: pagina's).

hoeveelheid	IC9/10	IC11/12	IC13/14	IC15/16	IC17/18	IC19/20	IC21/22	IC23/24
1K	20...23	nee	nee	nee	nee	nee	nee	nee
2K	20...23	24...27	nee	nee	nee	nee	nee	nee
3K	20...23	24...27	28...2B	nee	nee	nee	nee	nee
4K	20...23	24...27	28...2B	2C...2F	nee	nee	nee	nee
5K	20...23	24...27	28...2B	2C...2F	30...33	nee	nee	nee
6K	20...23	24...27	28...2B	2C...2F	30...33	34...37	nee	nee
7K	20...23	24...27	28...2B	2C...2F	30...33	34...37	38...3B	nee
8K	20...23	24...27	28...2B	2C...2F	30...33	34...37	38...3B	3C...3F

Tabel 11. RAM op de tweede kaart (adresbereik: pagina's).

hoeveelheid	IC9/10	IC11/12	IC13/14	IC15/16	IC17/18	IC19/20	IC21/22	IC23/24	
9K	40...43	nee	nee	nee	nee	nee	nee	nee	
10K	40...43	44...47	nee	nee	nee	nee	nee	nee	
11K	40...43	44...47	48...4B	nee	nee	nee	nee	nee	
12K	40...43	44...47	48...4B	4C...4F	nee	nee	nee	nee	
13K	40...43	44...47	48...4B	4C...4F	50...53	nee	nee	nee	
14K	40...43	44...47	48...4B	4C...4F	50...53	54...57	nee	nee	
15K	40...43	44...47	48...4B	4C...4F	50...53	54...57	58...5B	nee	
16K	40...43	44...47	48...4B	4C...4F	50...53	54...57	58...5B	5C...5F	

Het zij nogmaals vermeld: er is voorlopig geen rekening gehouden met de 16K-dynamische RAM-kaart. Het is mogelijk om vanaf adres 2000 te beginnen met dynamische RAM of om gewone RAM af te wisselen met dynamische RAM. Zorg in dat laatste geval wel voor volledig bezette blokken van 8K-RAM. Anders krijg je "gaten". Een van de band gelezen programma zou in zo'n gat kunnen vallen.

EPROM: 2716

In het geval 2716 is er plaats voor maximaal $4 \times 2 = 8\text{K}$ EPROM op elke RAM/EPROM-kaart. Er moet dus per kaart een beroep worden gedaan op één of twee te selekteren 4K-blokken uit de hoofddekoder IC5. Daartoe worden één of twee draadverbindingen gelegd van V en/of W naar één of twee aansluitpunten 2 . . . F. Bij twee verbindingen (6K of 8K) moet de ene verbinding op een oneven aansluitpunt zitten en de andere op een even aansluitpunt.

In de richting van toenemende adressen is de selektievolgorde niet logisch in het geval dat de draadbruggen P-R en S-T worden aangebracht, zoals ook in figuur 25 is aangegeven. Dit is geïllustreerd in tabel 12a, waarin is aangegeven welk IC wanneer aan bod is (selektiesignaal logisch nul; zie ook figuur 27b). Deze tabel is van toepassing als de uitgang van N1 logisch nul is, dus als het, onder invloed van IC5, de bedoeling is dat er een selektie wordt gemaakt uit IC25 . . . IC28.

Indien men de draadbruggen P-T en R-S legt in plaats van P-R en S-T, dan blijkt uit tabel 12b dat de selektievolgorde nu wel logisch is. Dit is handig als men grotere programma's heeft die meerdere EPROMS omvatten.

Tabel 12a. Selektie IC25 . . . IC28 bij draadbruggen P-R en S-T (2716).

A12 A	A11 B	IC25 k	IC26 l	IC27 m	IC28 n
0	0	0	1	1	1
0	1	1	1	0	1
1	0	1	0	1	1
1	1	1	1	1	0

Tabel 12b. Selektie IC25 . . . IC28 bij draadbruggen P-T en S-R (2716).

A12 B	A11 A	IC25 k	IC26 l	IC27 m	IC28 n
0	0	0	1	1	1
0	1	1	0	1	1
1	0	1	1	0	1
1	1	1	1	1	0

De voorkeursadressering en -dekodering voor de enige of eerste kaart op basis van het beginnen vanaf pagina FF is gegeven in tabel 13. Daarbij horen de volgende draadverbindingen:

e-g en a-d; P-T en R-S;

V-F en W-V voor 2K of 4K;

V-F en W-E voor 6K of 8K.

Is er meer dan 8K-EPROM nodig en ziet men af van het gebruik van 2732's, dan moet een tweede kaart worden genomen. Voor de EPROM op die tweede kaart geldt tabel 14, plus de nu volgende draadverbindingen:

Tabel 13. 2716-EPROM op de eerste kaart (adresbereik: pagina's).

hoeveelheid EPROM	IC25	IC26	IC27	IC28
2K	nee	nee	nee	F8...FF
4K	nee	nee	F0...F7	F8...FF
6K	nee	E8...EF	F0...F7	F8...FF
8K	E0...E7	E8...EF	F0...F7	F8...FF

Tabel 14. 2716-EPROM op de tweede kaart (adresbereik: pagina's).

hoeveelheid EPROM	IC25	IC26	IC27	IC28	
10K	nee	nee	nee	D8...DF	inklusief 8K van tabel 13 (eerste RAM/EPROM-kaart)
12K	nee	nee	D0...D7	D8...DF	
14K	nee	C8...CF	D0...D7	D8...DF	
16K	C0...C7	C8...CF	D0...D7	D8...DF	

e-g en a-d; P-T en R-S;

V-D en W-V voor 2K of 4K **extra** EPROM (totaal 10K of 12K);

V-D en W-C voor 4K of 6K **extra** EPROM (totaal 14K of 16K).

Is er nog meer 2716-EPROM nodig, dan zal het nu wel duidelijk zijn wat er moet gebeuren: per toegevoegde kaart verspringen naar de eerstvolgende één of twee, tot nu toe nog niet gebruikte aansluitpunten van IC5, met afnemende grootte van het getal waarmee elk aansluitpunt is aangegeven.

EPROM: 2708

Vervangt men in tabel 12a A11 door A10 en A12 door A11, dan krijgt men het 2708-selektieplaatje voor IC25 . . . IC28. Voor de eerste of enige RAM/EPROM-kaart met 2708-EPROM ziet de voorkeursindeling er uit als in tabel 15. Daarbij gelden de volgende draadverbindingen:

e-f en a-c; P-Q en S-T; V-F en W-V.

Voor meer dan 4K-EPROM met 2708 is men aangewezen op een tweede RAM/EPROM-kaart. Voor die tweede kaart geldt tabel 16 en de draadbruggen:

Tabel 15. 2708-EPROM op de eerste kaart (adresbereik: pagina's).

hoeveelheid EPROM (2708)	IC25	IC26	IC27	IC28
1K	nee	nee	nee	FC...FF
2K	nee	nee	F8...FB	FC...FF
3K	nee	F4...F7	F8...FB	FC...FF
4K	F0...F3	F4...F7	F8...FB	FC...FF

Tabel 16. 2708-EPROM op de tweede kaart (adresbereik: pagina's).

hoeveelheid EPROM (2708)	IC25	IC26	IC27	IC28	
5K	nee	nee	nee	EC...EF	inklusief 4K op de eerste kaart
6K	nee	nee	E8...EB	EC...EF	
7K	nee	E4...E7	E8...EB	EC...EF	
8K	E0...E3	E4...E7	E8...EB	EC...EF	

e-f en a-c; P-Q en S-T; V-E en W-V.

Voor nóg meer EPROM met 2708 trekt men er nog een kaart of kaarten bij en neemt achtereenvolgens en naar behoefte de IC5-aansluitingen D, C, B enzovoorts in gebruik.

En dan nog dit:

Nog wat losse maar niet onbelangrijke opmerkingen

1. Vergeet de draadbrug L-M niet!
2. De RAM/EPROM-kaart moet worden uitgerust met een mannetjes-konnektor volgens figuur 19a (op de komponentenzijde). Op de buskaart komen, afhankelijk van de hoeveelheid geheugenkaarten, 1 à 5 vrouwtjeskonnektors volgens figuur 19b. De geheugenkaarten staan dus haaks (loodrecht) op de buskaart.
3. Let erop dat de IC5-aansluitpunten op de print niet in de volgorde 0 . . . F zitten, maar in de volgorde van de bijbehorende IC5-pennen. Dit in verband met printontwerpoverwegingen. Hier dus goed oppassen!
4. Verbind niet gebruikte aansluitingen V, W, X en Y met een op dezelfde poort aangesloten "kollega" of met de +5 V.
5. De voorkeursopstelling van EPROM op de geheugenkaart, namelijk: EPROM opbouwen vanaf FFFF (dus in een richting met afnemend adres; omhoog op de geheugenkaart) houdt verband met het feit dat **op pagina FF EPROM moet zijn aangesloten** (dat wil zeggen: in ieder geval op de adressen FFFA . . . FFFF, maar 1K is nu eenmaal de kleinste maat, en dat zijn vier pagina's), **met op de adressen FFFA . . . FFFF vektor-data!** Deze kwestie is uitgebreid aan de orde gekomen bij de

bespreking van de databussturing van de interface-kaart.

Op de basis van handhaving van de standaard-EPROM (versie D), op de basiskaart van de junior-computer, moeten de volgende adressen van de EPROM IC28 als volgt zijn geprogrammeerd:

NMIL; adres FFFA data 2F;

NMIH; adres FFFB data 1F;

RESL; adres FFFC data 1D;

RESH; adres FFFD data 1C;

IRQL; adres FFFE data 32;

IROH; adres FFFF data 1F.

Na het inschakelen van de junior-computer wordt gesprongen naar de opstartroutine RESET van de standaardmonitor. De sprongvectoren IRQ en NMI kunnen op de bekende wijze worden gespecificeerd op PIA-RAM (pagina 1A). Uiteraard kan de individuele junior-gebruiker kiezen voor andere oplossingen, die dan samenhangen met zijn systeem-programma('s). Bedenk in ieder geval dat zonder gespecificeerde reset-vektordata de junior-computer na het inschakelen zo dood als een pier zal blijven! Specificeer de NMI- en IRQ-vektordata zodanig dat, via een eveneens in EPROM gebakken JMP-indirekt, naar RAM wordt gesprongen, waar men dan de sprongvectoren afhankelijk van de situatie kan vastleggen. Zie figuur 32 en de bijbehorende tekst van hoofdstuk 3 van boek 1.

6. Indien u denkt om op grond van uitbreidingen in het aantal systeem-programma's zo maar eventjes een EPROM te kunnen verhuizen van de ene naar de andere IC-voet, dan komt u bedrogen uit: niet alleen het startadres verandert dan, maar ook alle operand-data (in ieder geval ADH) die betrekking heeft op absolute adressen binnen de EPROM. Als men zo'n operatie wil uitvoeren moet de EPROM opnieuw worden geprogrammeerd, onder aanpassing van de voornoemde operand-data. Hiervoor is het absoluut noodzakelijk dat men over een uitgebreide, goed gedocumenteerde programmalisting beschikt van de te verplaatsen ("relocaten") EPROM. Een hex dump is dus niet voldoende, tenzij men over een dissassembler-systeemprogramma beschikt of dat "met de hand" uitzoekt, met een behoorlijk grote kans op fouten.
7. Het is uiteraard mogelijk om in het geval van meer dan één kaart de ene kaart met 2716-EPROMs te vullen en een andere met 2708's, waarbij de "van-onder-af"-regel kan blijven gehandhaafd. Men moet dan zelf de bijbehorende draadverbindingen uitzoeken. Bij EPROMs is een eventueel optredend "gat" overigens niet zo erg als bij RAM, vooral niet als de beschikbare 56K buskaartgeheugen bij lange na niet wordt benut.
8. Zoals uit het voorgaande blijkt is het toegestaan om een door IC5 gencodeerd 4K-blok op een RAM/EPROM-kaart niet volledig te benutten voor RAM of EPROM. Het is echter niet toegestaan om het onbenutte gedeelte op (een) andere kaart(en) wél te benutten! En waarom dan wel niet? Wel, stel dat van een 4K-blok een deel op de ene kaart zit en een ander deel op een andere kaart of op andere kaarten. De 4K hoeft dan nog helemaal niet volledig te zijn benut. Herinneren we ons dat zodra een adres, behorend tot die 4K, aan bod

is, de selectie van het gehele 4K-blok tot gevolg heeft. Gaat het om een leesoperatie, dus is het de bedoeling dat data wordt opgehaald van één kaart, dan zijn de databuffers in de leesrichting actief op die kaarten waarop de bewuste 4K is geselecteerd. Dus de ongewenste data-parallelsituatie van figuur 6 treedt op.

Konklusie: je mag best een 4K-blok op een bepaalde kaart niet helemaal benutten, maar maak geen gebruik van de rest op (een) andere kaart(en).

Praktischer uitgedrukt: in het geval van meerdere RAM/EPROM-kaarten wordt elke doorverbinding met een bepaald IC5-aansluitpunt niet of hooguit één keer gebruikt.

9. De doorverbinding tussen de diverse massa-aansluitingen (4a/c&16a/c&32a/c) vindt plaats op de interface-kaart, bij de uitgangskonnek-tor.
10. Men kan één RAM/EPROM-kaart direkt op de basiskaart aansluiten. Voor een gebruiksaanwijzing kan men terecht in Aanhangel 1.

Hoofdstuk 10 is uit. U gaat nu òf naar hoofdstuk 11 om na de aansluiting van uw cassetterecorder(s) en de afregeling van de PLL te gaan spelen met het programma Tape Monitor, òf u gaat eerst naar hoofdstuk 12, om u te oriënteren over eventueel nog te bouwen of te wijzigen randapparatuur, en om vervolgens het systeemprogramma Printer Monitor te gaan bestuderen en het geleerde in de praktijk te brengen. U gaat in ieder geval verder op uw ontdekkingsreis door dit boek 3. Want dit hoofdstuk 10 was daarvan nog maar het begin. En over ontdekkingsreis gesproken: misschien is er sprake van enige ontberingen: wat is het moeilijk of wat doen ze moeilijk, of beide. Maar bedenkt u zich wèl dat Columbus een hele brede oceaan moest oversteken alvorens Amerika te ontdekken. En dat terwijl hij dat van te voren niet wist. Maar wij weten wat er voor u te ontdekken valt. Dus lees nou maar door!

Magnetische geheugenuitbreiding

Data van, naar en áan de lopende band

De hardware van de cassette-interface en de software van het systeemprogramma Tape Management (TM) maken het samen mogelijk om data te transporteren van de junior-computer naar de magnetische geluidsband, en omgekeerd. In wezen wordt óf een soort instructie LDA, óf een soort STA uitgevoerd. En dan niet voor "één data", maar voor een heel datablok, bijvoorbeeld een programma. En dan niet in of uit gewone RAM — die aan geheugenverlies lijdt na het uitschakelen van de netspanning — maar in "magnetische RAM": de geluidsband van een cassette-recorder.

Men is nu in staat om een programma-archief aan te leggen, of een programma-bibliotheek op te zetten, waarbij het kopiëren in het geheugen van de junior-computer nauwelijks nog gepaard gaat met dat vervelende toetswerk. En wat er nog aan toetswerk overblijft gaat heel makkelijk met behulp van vijf nieuwe toetsfuncties voor de vijf "oude" toetsen AD . . . PC.

Wat komt er allemaal kijken bij het doorgeven van data van of naar de band? Hoe gaat men in de praktijk te werk? Hoe richten we de PLL goed af, zodat de junior-computer leest wat er (op de band) staat? Van de antwoorden op al deze vragen biedt dit hoofdstuk een uitgebreide weergave; uw opname daarvan verstevigt de band met de junior-computer.

Toch wel handig dat je computer-data net als geluid kunt vastleggen op de band, en op ieder moment dat dat nodig is weer tevoorschijn kan toveren en het de junior-computer weer in het hoofd (= het geheugen) prenten kan. Enen en nullen op dezelfde manier bevriezen en ontdooien als de "output" van J.S. Bach, E. Presley, of de eerste hulppartij van J. Junior jr. Waarom eigenlijk? Er zijn nogal wat redenen denkbaar voor de noodzaak om data op de band te schrijven of van de band te lezen. U heeft bijvoorbeeld zelf een gebruikersprogramma ontwikkeld en wilt dat bewaren voor later gebruik. Zet het even in dat notitieboek: de cassette-band, en u heeft al uw RAM weer beschikbaar voor andere activiteiten. Of u kunt met een gerust hart de junior-computer uitschakelen: wat de junior-computer vergeet heeft de band onthouden. Later kunt u zo'n programma weer van de band lezen en in RAM-geheugen van de junior-computer schrijven. En dat allemaal zonder al te veel van dat vervelende toetswerk.

Of neem het geval dat u een groot programma aan het editen bent. U kunt het programma in ge-edite vorm (dus nog niet geassembleerd) op de band zetten. U schrijft zo'n programma op de band, gaat het vervolgens assembleren en het blijkt nog niet helemaal of helemaal (nog) niet te werken, omdat de computer weliswaar dommer maar eveneens slimmer is dan u als programmeur. Moet nu het programma helemaal opnieuw door de editor-molen lopen (na het assembleren zijn immers de labels weg)? Nee! U leest het ge-edite programma weer in, start de editor warm en gaat aan de slag om het programma alsnog in orde te krijgen. Wellicht komen er labels bij. U kunt nu, alvorens tot assembleren over te gaan, de gewijzigde situatie op de band zetten. En u mag dat herhalen zo vaak als nodig is. Een typisch voorbeeld dus van de cassette-band als databank: niet uw geld maar uw data zet u tijdelijk weg.

Maar er zijn nog meer toepassingen. Namelijk het lezen van de band van programma's die uzelf nooit eerder heeft ontwikkeld en ingetoetst. En dan hebben we het over het gebruik van de cassette-band als **programmabibliotheek**. De kollektieve bibliotheek van de gebruikersclub of de uitwisseling van gebruikersprogramma's tussen twee junior-gebruikers. Of programma's die compleet op band te koop zijn (wellicht een toekomstige ESS-activiteit van Elektuur). Men kan natuurlijk in dat soort situaties de programma's byte voor byte of instructie voor instructie intoetsen op basis van de voorhanden zijnde papieren dokumentatie: een kwestie van ophoesten via AD, DA en +, of via de editortoetsen. Bij langere programma's is dat echter buitengewoon vervelend werk. Daar komt nog bij dat de kans op fouten groot is; vele malen groter dan de kans op verminking van data na het schrijven op en weer teruglezen van de band. Nee, het gebruik van de magnetische geluidsband als een soort achtergrondgeheugen is pure noodzaak.

De cassette-band

Magnetische RAM

Gaan we uit van een simpele maar degelijke cassette-recorder (mono) met een programmateller, van vijftig à honderd gulden. Gaan we verder uit van C-60-cassettes (2 x 30 minuten) van een gulden of vijf (u ziet: dit boek

gaat tot ver in de tachtiger jaren mee!). U bent dan werkelijk de data-koning te rijk. Want ga maar na. Verderop zal blijken dat de datasnelheid vijftig data-bytes per seconde bedraagt. Bij een bandsnelheid van 4,76 cm/sec is dat iets minder dan een millimeter per data-byte. Als we uitgaan van 25 minuten "netto" voor één kant, dan kunnen er $25 \times 60 \times 50 =$ ruim 73 kilo's van elk 1024 bytes op de band. Dat is méér dan de 64K die de 6502 zonder kunstgrepen met zestien adreslijnen kan adresseren. En dat is dan nog maar één spoor. Goedkoper kan het gewoon niet. Tenzij u voor die 73K een notitieblok met 100 vel A4 neemt. Maar dan moet u zelf lezen en schrijven!

Hoe staat het op de band?

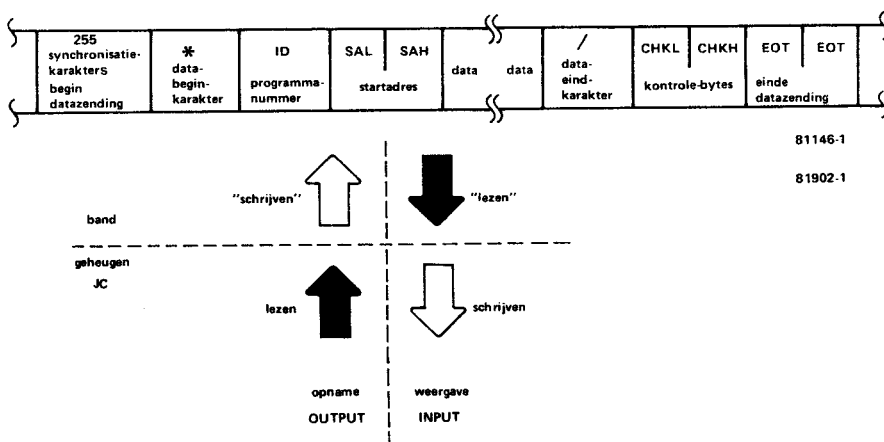
De band als datablokkendoos

Laten we ons eens bezig houden met de vraag wat er nodig is om een bepaalde hoeveelheid data die de band op gaat te dokumenteren. Wat is er naast de te verzenden data voor data nodig? Eerst nog iets over de te verzenden data. Er is sprake van:

- een datazending, en van
- een datablok.

Een datazending omvat een datablok, voorafgegaan door een "inleiding" en gevolgd door een "slot". Dus naast het hoofdgerecht is er soep en een dessert. In figuur 1 is geschetst hoe de onderdelen van een datazending op de band elkaar opvolgen, in de maaltijdtermen van zonet: het menu.

Een maaltijd in negen gangen.



Figuur 1. De diverse onderdelen van een op de band geschreven datazending.

① Synchronisatie-karakters

Waar begint het?

Men kan meerdere datablokken op één en dezelfde band zetten; denk maar even aan die 73K van daarnet. Er zijn dus meerdere datazendingen, met pauzes daartussen. Een pauze kan ook bestaan uit ingesproken commentaar bij een programma, bijvoorbeeld: "Mijn eerste 24-uursklok; ID=24 (geen gek idee — Schr); startadres zus en zo; geheugenplaatsen voor gelijk-zetten die en die". Bij het teruglezen van de band moet de junior-computer dit soort informatie kunnen negeren; zijn hardware zal weliswaar braaf enen en nullen maken uit de ingesproken tekst, maar dat is niet de data die in zijn geheugen moet worden gezet. Een datazending onderscheidt zich ondubbelzinnig van andere informatie op de band (inklusief brom en ruis) door dat bijbehorende begin van 255 synchronisatie-karakters. Net zoals bij een karnavalsoptocht of wielervedstrijd de komst ondubbelzinnig wordt aangekondigd door de reclamekaravaan.

Er is nu iets gezegd over synchronisatie: het in de pas lopen van de junior-computer bij het teruglezen van een datazending. Maar wat is een karakter voor een ding? Of een teken, want onder die naam kom je ze ook wel tegen. In wezen bedoelen we: **ASCII-karakter**. De ASCII-kode is een acht-bitskode, waarbij in zeven van de acht bits cijfers, letters, leestekens en een aantal andere zaken zijn verstopt; het achtste bit is bedoeld voor speciale taken; in ons geval is het nul. Een ASCII-tabel vindt u achterin dit boek, in Aanhangel 7. Alle onderdelen van een datazending gaan als bits van één of twee ASCII-bytes naar de band. De verzending gaat bit voor bit. Dat heet seriële dataoverdracht en zal in hoofdstuk 12 nader worden besproken. De hexadecimale weergave van de ASCII-kode van een synchronisatie-karakter is 16. Dus een datazending begint op de band met 00010110, en dat 254 keer herhaald.

② Data-beginkarakter*

Begin datazending is bekend

Ter afsluiting van de 255 synchronisatie-karakters en als aankondiging dat nu de echte data komt (waaronder het datablok) is het data-beginkarakter "*" (sterretje; ASCII-kode hexadecimaal: 2A) bedoeld.

③ Het programmanummer: ID

Om welke datazending gaat het?

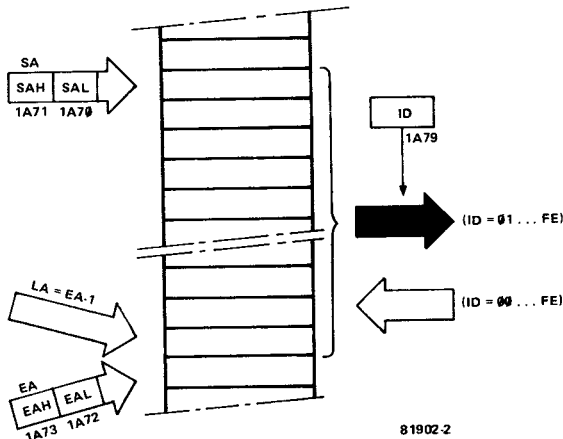
Prachtig natuurlijk dat je zo veel datablokken op één kant van een band kwijt kunt, maar je moet ze wél uit elkaar kunnen houden. Een hexadecimaal programmanummer ligt voor de hand. Bij het schrijven op de band zijn er 254 verschillende programmanummers mogelijk, namelijk van 01 tot en met FE. Bij het lezen van de band kunnen de programmanummers 00 en FF een speciale rol spelen. Maar dat komt nog wel. Het is zaak om niet dezelfde nummers te nemen voor twee datazendingen, zeker niet

als die twee op dezelfde band zitten. Het ID moet vóór het schrijven op de band worden opgegeven. Bij het teruglezen moet een ID 00 ... FF worden opgegeven.

④ Rechter startadresbyte SAL

⑤ Linker startadresbyte SAH en verder het laatste adres LA en het eindadres EA

Al die adressen horen bij het plaatje van figuur 2. Het startadres SA hoort bij het eerste adres van het op de band te zetten of van de band te halen datablok. Dit moet vóór het feitelijke datatransport naar de band aan de junior-computer worden opgegeven. Het laatste adres LA is zoals de naam al zegt het laatste adres van het naar de band te zenden of van de band te halen datablok. Niet dit adres moet vóór het feitelijke datatransport naar



Figuur 2. Een datablok met de bijbehorende grootheden (parameters), die een rol spelen bij de bijbehorende datazending.

de band worden opgegeven, maar het eindadres EA, dat één hoger is dan LA. Uit figuur 1 maken we op dat EA niet op de band staat. De EA-informatie zit verpakt in SA plus de lengte van het datablok, dus het aantal data-bytes, tevens het aantal adressen. Bij het teruglezen van de band hoeft uitsluitend in het geval ID=FF een startadres SA te worden opgegeven.

Waarom die komische EA, in plaats van de wat logischer keuze voor LA? Nu kunnen vergissingen ontstaan omdat een adres, één hoger moet worden opgegeven. Dit is gebeurd in verband met compatibiliteit met de gang van zaken bij de KIM-computer. Misschien is de opmerking overbodig, maar zorg ervoor dat SA altijd kleiner is dan EA. Data gaat in de volgorde met toenemend adres de band op.

Waarom eigenlijk SA en EA? Omdat de computer moet weten op welk adres hij moet beginnen met het lezen (schrijven op de band) of schrijven (lezen van de band) in het RAM-geheugen.

Okay, maar waarom die SA op de band? Het is toch fijn om bij het terug-lezen van de band een startadres te kiezen? Het kan (ID=FF), maar is in het algemeen helemaal niet zo fijn. **Bedenk dat een programma (= datablok) absolute adresoperanden bevat, die geheel of gedeeltelijk** (bij een pagina-verschuiving alleen de ADL's) **moeten worden gewijzigd bij de keuze voor een ander startadres**, dus bij de keuze voor een andere positie in het geheugen.

⑥ Het datablok

De eigenlijke data

Dáár is het allemaal om begonnen: een datablok naar de band wegschrijven om het ooit een keer terug te lezen van de band. Elk data-byte bestaat uit twee data-nibbles. Elk data-nibble gaat in de vorm van een ASCII-byte de band op. Per data-byte worden dus achtereenvolgens 16 bits naar de band verzonden. Het datablok omvat alle data van de adressen SA t/m LA, dus SA tot EA (zie figuur 2). Er zijn in principe geen beperkingen voor de grootte van een datablok. Het op de band schrijven (dus lezen uit geheugen) kan zowel op basis van RAM-data als EPROM-data gebeuren. Bij het teruglezen van de band is de aanwezigheid van RAM-geheugen essentieel; EPROM kan uitsluitend via programmeren worden beschreven.

⑦ Het data-eindkarakter "/"

Alle data verzonden of ontvangen

Dit karakter (ASCII : 2F) vertelt de junior-computer bij het lezen van de band dat alle data van het datablok is afgehandeld (bij het schrijven van data op de band is dat bekend uit het samenvallen van de adreswijzer POINT (POINTH, POINTL) met EA). Nu kan worden begonnen met het opschrijven respectievelijk vergelijken van controle-bytes. Daarover nu meer.

⑧ De controle-bytes CHKL en CHKH

*Groepsgewijze datacontrole aan de hand van individuele databijdragen
oftewel: is de data naar alle waarschijnlijkheid goed of niet goed
teruggelezen van de band?*

Nu volgt een moeilijk verhaal, dat misschien niet helemaal goed overkomt, maar als onderwerp heeft dat je behoorlijk goed kunt controleren dat data al dan niet goed overkomt na verzenden (schrijven op de band) en ontvangen (teruglezen van de band). Er kan onderweg iets mis gaan: data wordt verminkt of gaat totaal de mist in. De cassette-band, samen met de cassette-recorder, is en blijft de relatief gevoeligste kandidaat voor data-verminking; de beste PLL van de wereld helpt niet als niet alle te lezen data de opname/weergavekop van de cassette-recorder passeert, of voldoende sterk is vertegenwoordigd.

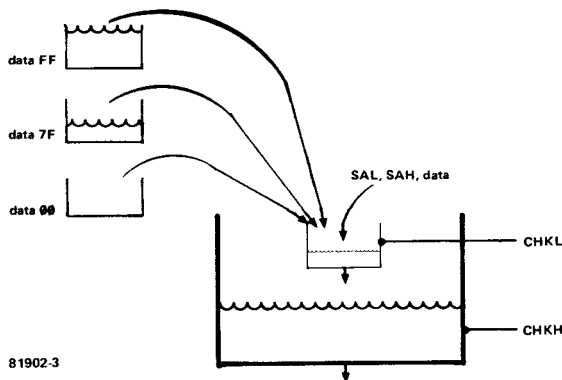
We kunnen controleren of wat er aan data van de band wordt gelezen

overeenstemt met wat er indertijd op is geschreven. Niet data voor data, maar voor het datablok inclusief SAL en SAH, op basis van individuele data-bijdragen. Dat gaat zo:

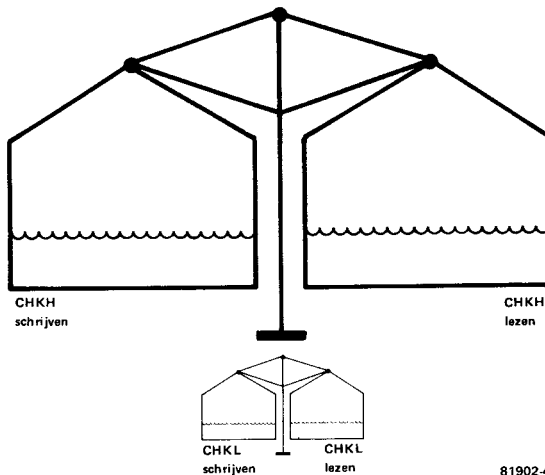
Aan het begin van een datazending naar de band hebben de geheugen-plaatsen CHKL en CHKH beide een inhoud 00. Vanaf SAL worden achter-eenvolgens de waarden van SAL, SAH en alle data van het datablok opgeteld bij CHKL. Het gaat om de waarden vóór de omzetting in de ASCII-verzendkode. Het programmanummer ID doet dus niet mee. Het aantal keren dat de inhoud van CHKL de waarde FF overschrijdt wordt bijgehouden in CHKH. Is ook CHKH vol (FF), dan begint alles weer opnieuw. De eindstanden van CHKL en CHKH zijn bereikt na het verzenden van de laatste geheugeninhoud van het datablok en worden op de band genoteerd.

Bij het lezen van data van de band gaan we precies hetzelfde doen, dus eveneens op basis van een beginwaarde van 00 voor CHKL en CHKH. Als alle data van het datablok is gelezen en uit ASCII is terugvertaald wordt de balans opgemaakt. De leeswaarde van de beide controle-bytes wordt vergeleken met de op de band genoteerde schrijfw waarde. Stemmen de beide CHKH- en/of CHKL-waarden niet overeen, dan kun je er bijna donder op zeggen dat er een verminkte dataoverdracht heeft plaatsgevonden. Stemmen beide waarden in beide gevallen overeen, dan is de kans buitengewoon groot dat alles is goed gegaan.

Een plaatje. Figuur 3. Neem een klein vat, CHKL en een groot vat, CHKH, dat 256 keer (FF) zo groot is. Beschouw de datastroom in het kleine vat als een vloeistofstroom. Is de data FF, dan komt de maximale hoeveelheid vloeistof in het kleine vat terecht: precies gelijk aan de inhoud van dat vat; is de data 00, dan komt er niets bij. Zodra het kleine vat dreigt over te lopen opent zich de bodem om een hoeveelheid vloeistof door te laten, gelijk aan de inhoud van het kleine vat; niet meer en niet minder. Hetzelfde gebeurt als het grote vat dreigt over te lopen; dan verdwijnt een hoeveelheid vloeistof, gelijk aan de inhoud van het grote vat. Als alle data is binnengestroomd is er sprake van twee vloeistofnivo's: de eindstanden, c.q. -waarden.



Figuur 3. Het lekke-vaten-model van de berekening van de controle-bytes CHKL en CHKH.



81902-4

Figuur 4. Het balans-model van de vergelijking van de twee CHKL- en CHKH-waarden.

Bij het lezen van de band gebeurt alles weer net zo als in figuur 3. Het "de balans opmaken" mag men letterlijk opvatten: zie figuur 4. Zijn de twee grote en de twee kleine vaten even zwaar, dan is waarschijnlijk alles in orde; slaat minstens één van de balansen door in een bepaalde richting, dan is er waarschijnlijk iets misgegaan.

Nu kunnen we u over het "waarom?" doodgooien met statistiek, informatietheorie en meer van die democratische verworvenheden van de twintigste eeuw. Nee dus. Wel zij vermeld dat de schijnbaar nutteloze omzetting van data-bytes in twee ASCII-bytes bij het schrijven op de band er iets mee te maken heeft.

Tóch een poging tot een verklaring. Neem een bank. Men wil goed op andermans geld passen en voorkomen dat iemand zich laat insluiten, teneinde 's nachts met de buit uit te breken. Dus wat doet men? Men telt elke dag het aantal binnenkomende klanten en het aantal vertrekkende klanten. Bij sluitingstijd worden beide aantallen met elkaar vergeleken. Of er bij overeenstemming reden is voor tevredenheid respectievelijk reden tot paniek bij het niet kloppen van de aantallen hangt helemaal af van de betrouwbaarheid van de telling. Er kunnen twee mensen tegelijk, naast elkaar de fotocel passeren; iemand heeft de lotto gewonnen en verlaat het gebouw na een beleggingsgesprek met de directie via de dienstuitgang. Waar het om gaat is dat er fouten kunnen ontstaan omdat de controle niet optimaal is. Twee fouten in de controle kunnen elkaar opheffen, zodat alles schijnbaar in orde is.

Het gedoe met CHKL en CHKH is te vergelijken met die bankcheck. Echter: de mensen worden niet alleen geteld (bij binnenkomst en bij vertrek) maar ook elk gewogen. Elke keer dat de 100 kg wordt overschreden wordt er 100 kg toegevoegd aan een ton (1000 kg). Van de totale hoeveelheid tonnen wordt het maximale gehele aantal tonnen afgetrokken. De beschreven controlemethode met CHKL en CHKH is, gegeven de acht-

bits middelen, nagenoeg waterdicht. En denk bij dat "waterdicht" niet aan die plotseling lekkende vaten van figuur 3!

⑨ Twee einde-zendingstekens EOT

Hekkesluiters

Zijn bedoeld als ondubbelzinnige vaststelling van het einde van een data-zending; EOT betekent End Of Transmission. De ASCII-kode van de tekens is 04. Hiermee zijn alle onderdelen van figuur 1 besproken. U ziet, er komt nogal wat bij kijken. Op het aantal synchronisatiekarakters na (255, in plaats van 100) is de gang van zaken dezelfde als die bij de KIM-computer.

Hoe gaat het op de band?

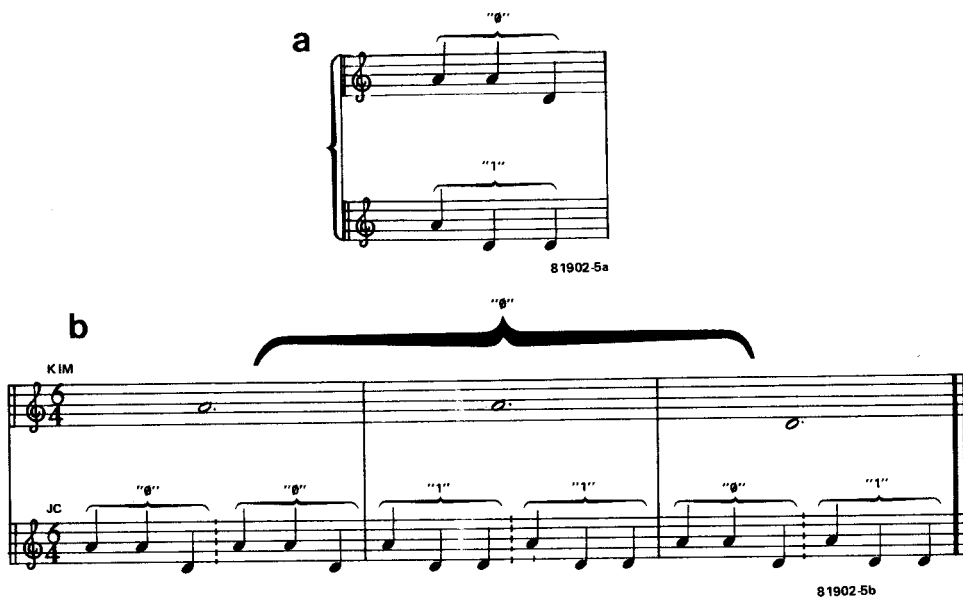
Digitale muziek met drie noten per bit

We weten inmiddels dat data serieel, bit voor bit de band op gaat. De vraag is: hoe staan al die bits op de band? Nu wordt een logische één ook wel vereenzelvigd met "hoog" en een logische nul met "laag". Je zou dus een bit één kunnen vertalen in een hoge frekwentie en een bit nul in een lage. De weergave op de band van een bitstroom bestaat dan uit een aaneenschakeling van hoge en lage frekwenties. Dit is óók het geval bij de junior-computer, maar dan niet op basis van een koppeling hoog-één en laag-nul. Maar op welke basis dan wèl?

Zowel een bit één als een bit nul wordt vertaald in een stukje hoge frekwentie, gevolgd door een stukje lage frekwentie. Het verschil tussen een één en een nul zit hem in de tijdsduur van de hoge ten opzichte van de lage frekwentie. In beide gevallen is de totale tijdsduur gelijk. Bij een logische nul duurt de hoge frekwentie twee keer zo lang als de lage en bij een logische één is het precies omgekeerd: de lage frekwentie duurt twee keer zo lang als de hoge.

Het gaat dus om tijdsverhoudingen van twee op een of een op twee. En aangezien twee plus een drie is kun je een bit muzikaal weergeven met drie noten. In figuur 5a is dat te zien. De bit-nul-melodie bestaat uit twee hoge noten, gevolgd door een lage noot, en de bit-één-melodie uit één hoge noot, gevolgd door twee toontjes lager.

Nu heeft de junior-computer geen piano nodig voor het schrijven van bits op de band. Het zou trouwens een kamerbrede piano moeten zijn, want in het echt is de hoge frekwentie 3600 Hz en de lage frekwentie 2400 Hz. In figuur 6a zien we geen noten maar pulstreintjes, die elk bestaan uit een stukje hoge frekwentie ("korte wagons" van de pulstrein) en een stukje lage frekwentie (de "lange" wagons). Een bit 1 is vertaald als drie halve perioden 3600 Hz plus vier halve perioden 2400 Hz (plaatje ② van figuur 6a). Een bit 0 krijgt een pulstrein van zes halve perioden 3600 Hz plus twee halve perioden 2400 Hz ① van figuur 6a). De treintjes zijn even lang, namelijk 9T, waarbij T gelijk is aan de halve periodetijd van 3600 Hz. Altijd begint het met de hoge frekwentie en de tijdsuren van de hoge en



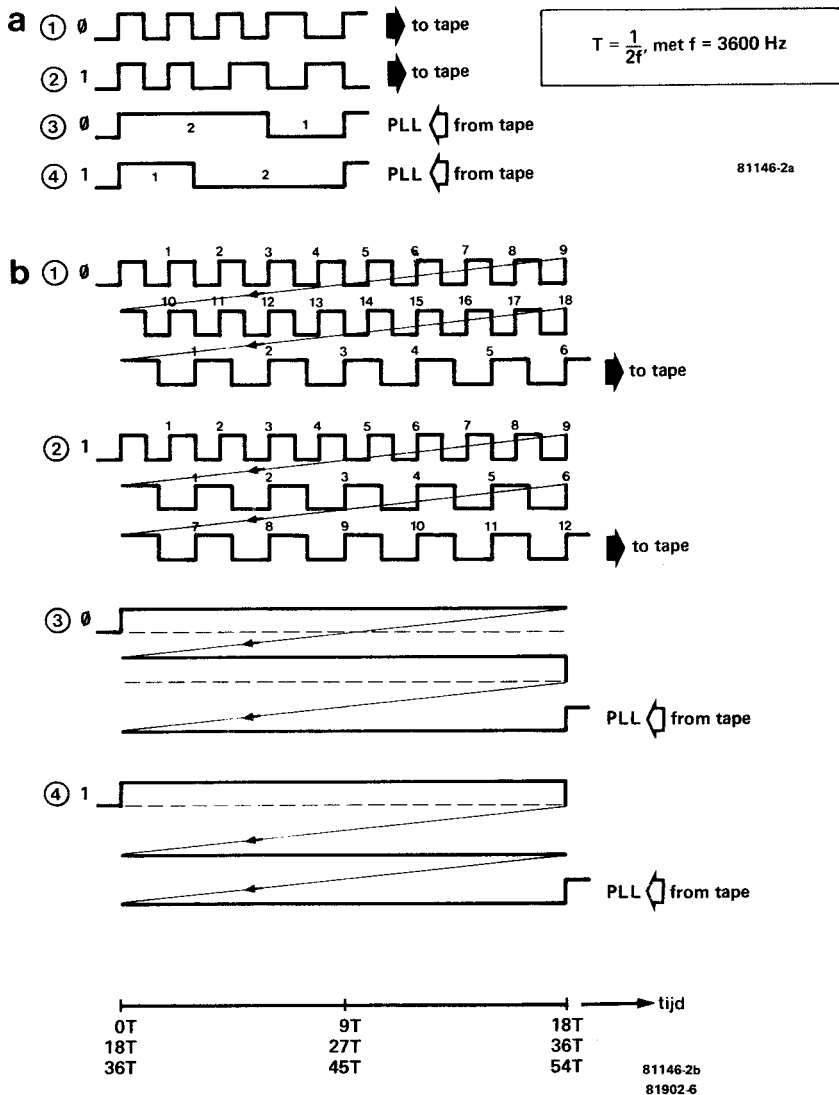
Figuur 5. Zeg het met noten: een bit bestaat uit twee hoge noten plus één lage of één hoge plus twee lage noten. Een noot geeft een periode met een hoge frekwentie of een periode met een lage frekwentie weer (figuur 5a). De "bit-muziek" wordt bij de junior-computer (bijna) zes keer zo snel opgenomen en weergegeven als bij de standaard-KIM-computer (figuur 5b).

lage frekwenties verhouden zich als twee op een of een op twee. In het muziekplaatje van figuur 5a duurt elke noot 3T.

Ter vergelijking is in figuur 6b de gang van zaken bij de standaard-KIM-computer getekend. De tijdas is twee keer teruggebogen om de tekening niet naar rechts uit de hand te laten lopen. Een bit één bestaat hier namelijk uit negen hele perioden 3700 Hz (in figuur 6b afgerond op 3600 Hz) plus twaalf hele perioden 2400 Hz. Een bit nul omvat achttien hele perioden "3600 Hz" plus zes hele perioden 2400 Hz.

Bij de KIM duurt het allemaal dus nogal wat langer dan bij de junior-computer. De muziek van figuur 5b illustreert dit. In de tijd die de KIM nodig heeft voor het "spelen" van één bepaalde bit-melodie heeft de junior-computer er al (bijna) zes bits opzitten: Jan daar ligt een kip in 't water versus de minutenwals van Chopin.

Terug naar de digitale kant van de zaak. De tijdsduur van een pulstrein (dat is de zogenaamde *bittijd*) is bij de KIM bijna zes keer zo lang als bij de junior-computer. Er wordt dus ook bijna zes keer trager data op de band geschreven en van de band gelezen. Er zijn overigens allerlei snelle varianten voor de KIM ontwikkeld, met inbegrip van software die op zijn beurt weer sneller is dan die van de junior-computer. Met name J. Butterfield heeft de nodige steentjes bijgedragen met zijn "Hypertape". In boek 4, bij de bespreking van de cassette-software gaan we dieper in op aspecten van



Figuur 6. De tijdens het schrijven op de band uitgezonden signalen, horend bij een nul (①) en een bit één (②) voor de junior-computer (a) en de KIM-computer (b). Teneinde de plaatjes b niet al te breed te laten worden is de gemeenschappelijke tijdas, onderaan in de figuur, twee keer "teruggezet". De tijdas is verdeeld in eenheden van een halve periode van 3600 Hz (T). Plaatjes ③ en ④ zijn de overeenkomstige signalen bij het lezen van de band (PLL-uitgangsspanning, met afgeregelde PLL). Deze signalen worden, na inversie, door de computer gebruikt om de van de band gelezen data te rekonstrueren en in het geheugen te schrijven. Vergelijking met figuur 5 leert dat een junior-noot 3T sekonden duurt en een KIM-noot 18T sekonden.

snelheid, nauwkeurigheid en betrouwbaarheid, die stuk voor stuk een belangrijke rol spelen.

Nu even wat rekenwerk. De bittijd van de junior-cassette-software bedraagt negen halve perioden van 3600 Hz, dus $9 \times 139 = 1250$ mikrosekunden. Dat komt neer op 800 bits per seconde, dus 100 ASCII-bytes per seconde, dus 50 data-bytes per seconde. In verband met die 800 bits per seconde spreekt men ook wel van een **Baudrate** van 800.

Het schrijven van een datablok, of juister: een datazending op de band gebeurt met de subroutine DUMP/DUMPT, die wordt aangeroepen uit het hoofdprogramma Tape Management. Gedurende het schrijven zijn de zes displays gedoofd. Daar staat tegenover dat de rode opname-LED oplicht. Vooraf moeten een ID 01 ... FE, het startadres SA en het eindadres EA (dat één hoger is dan het laatste adres LA van het te verzenden datablok) worden opgegeven. Vanaf het eerste adres SA tot en met het laatste adres LA wordt de data van het datablok uit het geheugen gelezen. Dat gebeurt met behulp van de welbekende adreswijzer POINT, die ook een rol speelt bij het displayen van een adres plus data. Direkt na afloop van de data-verzendroutine DUMP/DUMPT is de inhoud van POINT gelijk aan EA. Dit kan men controleren door het indrukken van de toets RST. Het display toont ons dan het eindadres EA, met data.

Hoe komt het van de band?

Van bits op de band naar bytes in het geheugen

We hebben gezien dat de bits op de band staan als twee hoge noten plus één lage (0) of één hoge noot plus twee lage (1; zie figuur 5a). Bij het lezen van de band moeten de notentrio's in bits worden terugvertaald, de bits tot ASCII-bytes worden gegroepeerd en de ASCII-bytes omgezet in data-bytes; die laatste moeten in het geheugen worden geschreven vanaf het op de band genoteerde (ID = 00 ... FE) of vooraf opgegeven (ID=FF) startadres.

Eerst maar eens de bits rekonstrueren. Bij het lezen van de band zijn de opgenomen signalen ① en ② van figuur 6a mooi rechthoekig van vorm (zie alvast figuur 17). De opname/weergaveketen van de cassette-recorder zorgt ervoor dat de rechthoekige vorm verloren gaat. Door de laagdoorlaat-filterwerking zal er iets sinusvormigs zijn te zien op de weergave-ingang INPUT van de junior-computer (zie alvast de figuren 18 en 19). Waar het echter voor de PLL om gaat is dat de nuldoorgangsinformatie niet verloren gaat. Of anders gezegd: de frekwentie-informatie (hoge of lage noten) blijft behouden. Dáár valt voor de PLL mee te werken, met als gevolg dat plaatje ① (in de "sinusversie") van figuur 6 wordt omgetoverd in ③, en ② in ④. In die plaatjes ③ en ④ ziet u dat notenverhaal van figuur 5a trouwens zeer goed weerspiegeld. Bij de bespreking van de PLL in hoofdstuk 10 is de overgang van ① naar ③ en van ② naar ④ uitgebreid besproken.

Het uitgangssignaal van de PLL (③ of ④) wordt geïnverteerd (hoog wordt laag, en omgekeerd) toegevoerd aan de eigenlijke junior-computer via een poortlijn. De lees-software destilleert uit het PLL-uitgangssignaal een bit 1 of 0 op basis van een vergelijking van de tijdsduren van het hoog respek-

tievelijk laag zijn van het omgekeerde PLL-uitgangssignaal. Er wordt dus gekeken of er van een notentrio meer hoge dan lage noten zijn of juist omgekeerd. Er wordt niet gevraagd: is de 3600 Hz-tijdsduur 6T lang en de 2400 Hz-tijdsduur 3T lang en is het dus een nul? Nee, er worden geen absolute tijdsduren gemeten, maar met elkaar vergeleken. Is de ene tijdsduur beduidend langer dan de andere, dan ligt de waarde van het bit vast. Omdat de absolute tijdsduren binnen een ruime marge geen enkele rol spelen is het zonder enig probleem mogelijk om via de KIM op de band geschreven programma's terug te lezen op de junior-computer, ondanks de bijna zes keer tragere gang van zaken. Immers, de signalen ③ en ④ van figuur 6b zijn gelijkvormig aan de signalen ③ en ④ van figuur 6a.

Een voordeel van de relatieve tijdsmethode, toegepast bij zowel KIM als junior, is dat de invloed van variaties in de bandsnelheid (4% cm/seconde voor de meeste cassette-recorders) nihil is (mits de PLL goed is afgeregeld). Dat is gunstig voor de verwisselbaarheid van cassette-recorders, die absolute bandsnelheidsverschillen kunnen vertonen. Verder is de negatieve invloed van kortstondige bandsnelheidsvariaties ("flutter" en zo) nihil. En dat allemaal vanwege de enorme marge tussen de verhouding twee op één voor de ene bitwaarde en één op twee voor de andere bitwaarde. Een (tijdelijk door storingen ontstane) verhouding 4 : 3 (i.p.v. 2 : 1) respectievelijk 3 : 4 (i.p.v. 1 : 2) levert voor de lees-software geen enkel probleem op bij het vaststellen van de juiste bitwaarde. En bedenkt u zich maar eens aan de hand van figuur 5a wat dat betekent: het plaatje met twee hoge noten en één lage noot verschilt sterk van het plaatje met één hoge noot en twee lage noten; hoe minder in het oog lopend is het verschil tussen een plaatje met vier hoge en drie lage noten en een plaatje met drie hoge en vier lage noten!

Bij de bespreking van de PLL in hoofdstuk 10 is de PLL-jitter aan de orde gekomen. Het ging om ongeoorloofde, vroegtijdige en nog niet definitieve veranderingen in logisch nivo van het PLL-uitgangssignaal. Bij de tijdsmetingen ten behoeve van het vaststellen van de bitwaarde wordt elke tijdsperiode waarbinnen jitter optreedt niet meegerekend. In wezen wordt zowel van de hoge-noten-tijd als van de lage-noten-tijd een tijdje afgetrokken. Dit betekent dat de genoemde verhoudingen 1 : 2 en 2 : 1 al iets minder extreem worden. Maar gezien de al eerder besproken marge is dat geen enkel probleem. De niet mee te rekenen jittertijden gaan wel een rol spelen als men probeert om nog hogere lees- en schrijfsnelheden te realiseren. Want dat gaat gepaard met een kortere bittijd (= tijdsduur hoge + lage noten); bij een gelijkblijvende jittertijd gaat de laatste meer en meer overheersen. Dus vergéét al die super-snelle cassette-interfaces maar, die ooit zijn gepubliceerd, als u óók nog betrouwbaar die snelle data van de band wil lezen: haastige spoed is zelden goed!

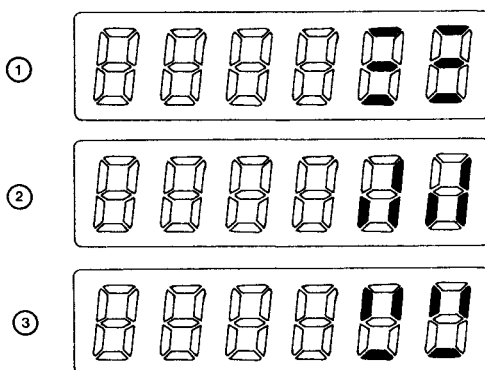
De subroutine RDTAPE

Van de band lezen en ook nog wat te zien krijgen

Het lezen van de band gebeurt tijdens de subroutine RDTAPE, waarop een beroep kan worden gedaan door Tape Management en trouwens ook door het programma Printer Monitor van hoofdstuk 12. Nu wordt niet zomaar

klakkeloos de eerste de beste data ingelezen die de revu (= opname/weergavekop) passeert. Nee, een bepaalde, door de gebruiker gekozen data-zending moet door de junior-computer worden opgenomen ter verdere verwerking (waaronder het in het geheugen schrijven van het van de ASCII-verpakking ontdane datablok). De gebruikerskeuze omvat de opgave vooraf van een programmanummer (ID) en, in één bepaald geval, een startadres. Nu kunnen er vele datazendingen staan op één en dezelfde cassetteband. Het is dus handig om over visuele informatie te beschikken over de stand van zaken. Wat gebeurt er of wat gebeurt er juist niet als de lopende band via de weergavekop de PLL-ogen van de junior-computer iets te zien geeft?

Behalve de groene INPUT-LED — die oplicht tijdens het verblijf in RDTAPE — worden de rechter twee (ex-data-)displays gebruikt om gedurende het lezen van de band (in de zin van het sturen van de PLL) te laten zien wat er aan de hand is. Want zoals gezegd: niet alles wat wordt gelezen is bestemd voor het geheugen. De rechter twee displays kunnen op drie verschillende manieren oplichten (zie de figuren 7 en 8):



81146 3
81902-7

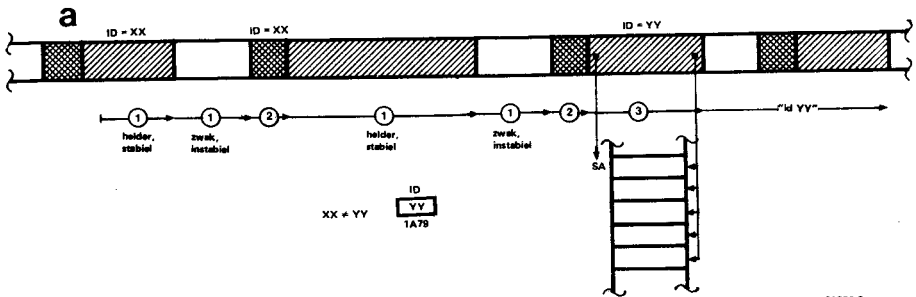
Figuur 7. Tijdens het zoeken, vinden en teruglezen van een datazending op de band is er iets te zien op de rechter twee displays van de junior-computer. Dit is het geval na het indrukken van de toets GET van TM.

Plaatje ①

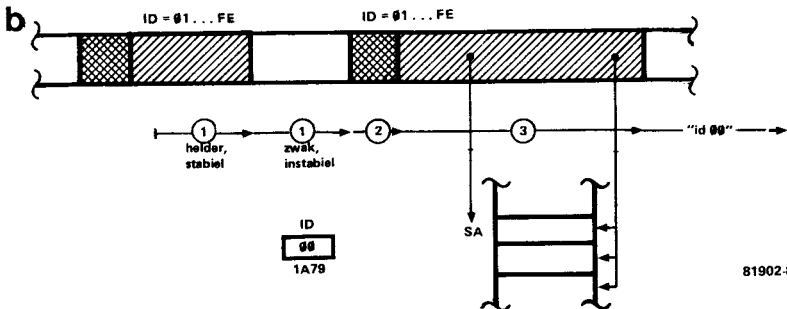
"U zit er naast"

Er doen zich verschillende situaties voor waarbij ① is te zien:

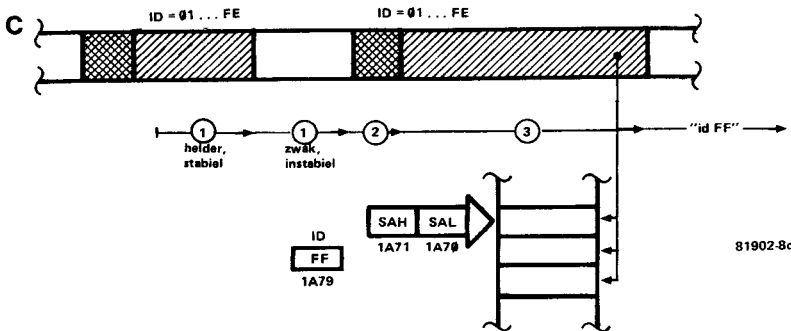
- De langs de opname/weergavekop lopende band (cassette-recorder in de stand weergave) geeft een periode tussen twee datazendingen weer, of een lege band, of commentaar. De displays lichten niet helder op; duidelijk ziet men Di5 en Di6 elkaar afwisselen.
- Er passeert een datazending de revue, waarvan we het begin hebben gemist, of een (vanaf het begin ingelezen) datazending met een van het gewenste, vooraf opgegeven programmanummer afwijkende waarde. De displays geven een helder en stabiel beeld te zien; ze lichten schijnbaar tegelijk op.



81902-8a



81902-8b



81902-8c

Figuur 8. Op zoek naar de datazending die we moeten hebben, in het geval ID=01...FE (figuur 8a), ID=00 (figuur 8b) en ID=FF (figuur 8c). Na het indrukken van toets GET starten we midden in een datazending. Dubbel gearceerde gebieden bevatten synchronisatiekarakters, enkelvoudig gearceerde gebieden data en blanco gebieden bevatten niets of "data" die voor de computer niet interessant is.

Plaatje ②

"U heeft een begin te pakken"

Dit plaatje is zichtbaar als er synchronisatie-karakters aan de orde zijn. Dus als het gaat om het begin van een (nieuwe) datazending. Bij de eerste

karakters kunnen onregelmatigheden optreden; dat is te zien aan een onrustig beeld. Het kan voorkomen dat kortstondig wordt teruggeschakeld op plaatje ①. Gaandeweg (pakweg een seconde later) is het beeld helder en stabiel geworden. Er staan 255 synchronisatie-karakters op de band; die 255 ASCII-bytes nemen een tijd in beslag van ca. 2½ seconde. De junior-computer gaat uit van een ondubbelzinnig begin van een datazending als er tien synchronisatie-karakters achter elkaar en ononderbroken zijn vastgesteld. Lukt dat niet meteen, dan zijn er vanwege die 255 stuks nog een dikke twintig herkansingsmogelijkheden. De KIM-datazendingen beginnen met 100 synchronisatie-karakters. Dus minder herkansingsmogelijkheden, met een groter gevaar dat een datazending zo maar wordt genegeerd. Okay, er is een begin vastgesteld van een datazending. Is het nu óók nog de datazending die we zoeken: Dàt zou fijn zijn.

Plaatje ③

"Hebbes!"

Als direkt na ② plaatje ③ te zien is dan is de gewenste datazending gevonden; het bijbehorende datablok wordt in het geheugen van de junior-computer geschreven. Onder "gewenste datazending" verstaan we de datazending met het vooraf opgegeven ID 01 ... FE of de eerste de beste datazending die volledig, van het prille begin tot het bittere einde, de opname/weergavekop passeert (ID 00 of FF).

Als direkt na ② plaatje ① terugkeert (met heldere, stabiele displays) betekent dit dat het niet om de gewenste datazending gaat. Deze situatie treedt nooit op als vooraf een programmanummer 00 of FF is opgegeven. Maar misschien is de volgende datazending wèl de gewenste. Dat geduldig afwachten kan natuurlijk aanzienlijk worden beperkt als behalve het programmanummer ook de positie op de band van de datazending wordt genoteerd. Een bandteller is dus reuze handig.

N.B. Voor een illustratie bij de gang van zaken: zie figuur 8a.

Spelen met datablokken

ID 00 en ID FF: eigen ideeën

Bij het schrijven op de band moet vooraf een programmanummer 01 ... FE worden opgegeven en bij het teruglezen van de band kan men niet alleen een keuze doen uit die 254 mogelijkheden; men kan ook kiezen voor een programmanummer 00 of FF.

Hoe zit het met die bijzondere gevallen 00 en FF?

Programmanummer 00. Er gebeurt het volgende:

- het datablok van de eerste de beste (volledig gelezen) datazending wordt in het geheugen van de junior-computer geschreven;
- het startadres SA in het geheugen is gelijk aan het op de band genoteerde startadres;
- het op de band genoteerde programmanummer wordt dus genegeerd;
- zie ook figuur 8b.

Programmanummer FF. Er gebeurt het volgende:

- het datablok van de eerste de beste (volledig gelezen) datazending wordt in het geheugen van de junior-computer geschreven;
- het startadres SA in het geheugen is gelijk aan een vooraf opgegeven startadres SA;
- het op de band genoteerde programmanummer wordt dus genegeerd;
- het op de band genoteerde startadres wordt ook genegeerd;
- zie ook figuur 8c.

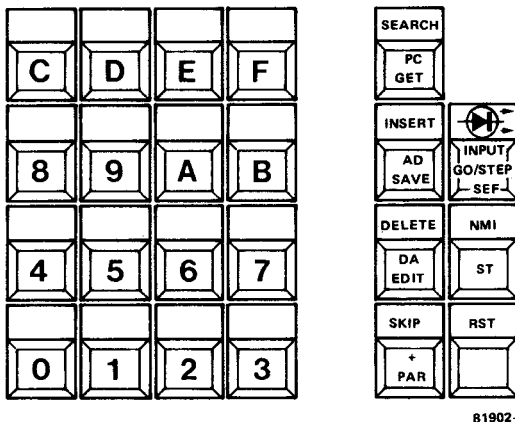
Het is dus mogelijk om te manipuleren met datablokken. Men kan een willekeurig programma met behoud van startadres in het geheugen zetten. Of men kan een willekeurig programma op een andere plaats in het geheugen zetten, dus verplaatsen ("relocaten"). We zeggen nou wel "willekeurig", maar het zou best wel eens om een bewuste keuze kunnen gaan. Het is dus vaak nodig om met behulp van de bandteller (snelspoelen van de cassette-recorder) een datazending klaar te zetten alvorens met het inlezen te beginnen. Denk bij het verplaatsen van een datablok aan de noodzaak tot wijziging, geheel of gedeeltelijk, van absolute adresoperanden. Die dingen bestaan echter nog niet in ge-edite programma's c.q. datazendingen en met name daar ligt de toepassing van het programmanummer FF.

TAPE MANAGEMENT (= TAPE MONITOR)

Lezen en schrijven met de band en met de junior-computer

Het systeemprogramma Tape Management (TM) — ook wel Tape Monitor genoemd — verzorgt al uw opdrachten in het kader van het schrijven op en lezen van de band van bepaalde data, in de vorm van datazendingen. Het programma zit in EPROM (IC4, een 2716) en staat u dus direkt na het inschakelen van de junior-computer ter beschikking. Onder het nummer ESS 506 kunt u uw EPROM door Elektuur laten vullen met alle TM-bytes. U kunt dat ook zelf doen, aan de hand van de hex dump in Aanhangsel 4, achterin dit boek. Het programma is langer dan 1024 maar korter dan 2048 bytes; in de 2716 is dus nog plaats voor toekomstige, aanvullende huishoudelijke bytes. Het programma omvat de geheugenplaatsen 0800...0C7F. Het startadres is 0810, niet 0800! In sommige gevallen verlaat men TM via de editor (zie boek 2), in andere gevallen neemt men na verrichte diensten afscheid door het indrukken van de toets RST; dan staan de diensten van de (standaard-)monitor u weer ter beschikking.

Het programma TM bestaat voor het overgrote deel uit de eigenlijke schrijf-subroutine DUMP/DUMPT en de eigenlijke lees-subroutine RDTAPE. De rest omvat hulpsoftware die het, op basis van nieuwe functies voor de bestaande toetsen AD...PC, mogelijk maakt om buitengewoon gemakkelijk en comfortabel allerlei gegevens (**parameters**) op te geven alvorens tot schrijven op of lezen van de band over te gaan. Het "nieuwe" toetsenbord is te zien in figuur 9. We laten nu de vijf nieuwe toetsfuncties voor de vijf oude, bestaande toetsen stuk voor stuk de revue passeren.



Figuur 9. Nieuwe TM-taken voor het oude, bestaande toetsenbord van de junior-computer.

① Toets PAR (ex-plus; ex-SKIP)

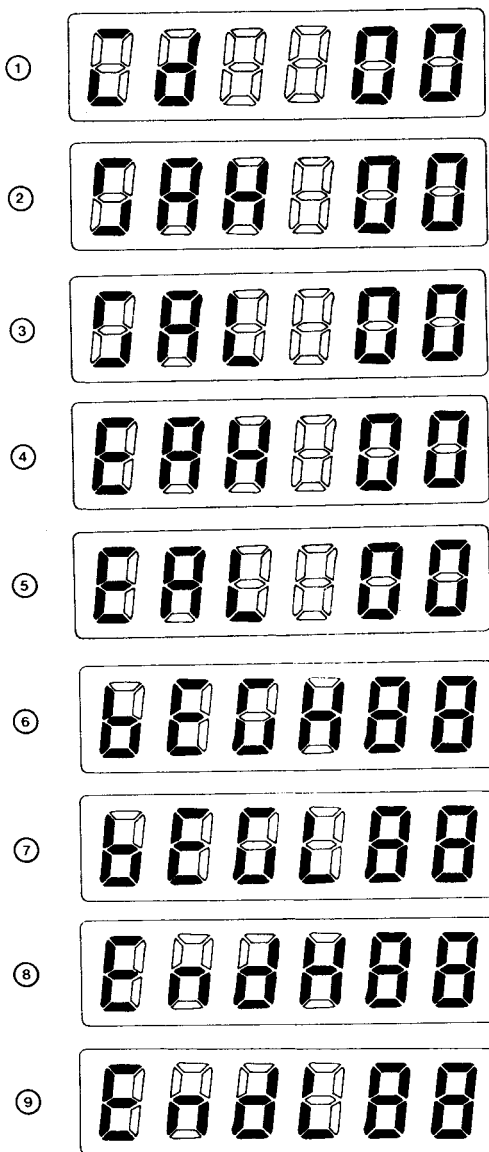
Parameters vooraf opgeven

Beginnen we gewoon maar eens met een toetsenverhaal:

Wat indrukken?	Wat is er te zien?
RST 0810 GO	id 00 (plaatje ① van figuur 10)
2 C	id 2C
PAR	SAH 00 (plaatje ② van figuur 10)
02	SAH 02
PAR	SAL 00 (plaatje ③ van figuur 10)
PAR	EAH 00 (plaatje ④ van figuur 10)
03	EAH 03
PAR	EAL 00 (plaatje ⑤ van figuur 10)
F F	EAL FF
PAR	bEGHXX (plaatje ⑥ van figuur 10)
PAR	bEGLXX (plaatje ⑦ van figuur 10)
PAR	EndHXX (plaatje ⑧ van figuur 10)
PAR	EndLXX (plaatje ⑨ van figuur 10)
PAR	id 2C
PAR	SAH 02
PAR	SAL 00
PAR	EAH 03
0 4	EAH 04
PAR	EAL FF
0 0	EAL 00
PAR	bEGHXX

enzovoorts.

Wat is er gebeurd? Het ging kennelijk om de voorbereidingen voor het schrijven op de band van een datazending (via toets SAVE, waarover later



81146 4
81902-10

Figuur 10. De negen parameters die bij het datatransport van of naar de band een rol kunnen spelen, kan men stuk voor stuk opgeven met de toets PAR, tijdens het verblijf verblijf in TM.

meer), want er is een programmanummer (ID=id) opgegeven en ook een startadres SA en een eindadres EA. Het was de bedoeling dat het datablok van 0000 tot en met 03FF als datazending nummer 2C naar de band ging. Nog net op tijd ging er ons een lichtje op: die lastige kwestie van EA=LA+1. Door net zo lang PAR in te drukken dat EAH en vervolgens EAL aan bod zijn kon de fout worden hersteld.

Bestudering van de toetsenakties plus displayplaatjes levert een aantal interessante konklusies:

- Alle negen parameters die een voorbereidende rol (kunnen) spelen bij de overige vier toetsfuncties van TM zijn toegankelijk en kunnen elk worden gespecificeerd door het indrukken van twee numerieke toetsen.
- Bij het opstarten van TM zijn de parameters ID, SAH, SAL, EAH en EAL nul. De overige (editor-)parameters hebben een waarde die willekeurig is (je vlak voor aanroep TM ingeschakeld) of samenhangt met de voorgeschiedenis (verblijf in de editor). De plaatjes ⑥ . . . ⑩ van figuur 10 laten de data "8" zien, ter illustratie van de dontcare-situatie (alle segmenten kunnen oplichten).
- Er is sprake van een kringloop: alle parameters verschijnen periodiek in beeld via het herhaald achter elkaar indrukken van PAR.
- Zodra een numerieke toets is ingedrukt is de gedisplayde parameter gewijzigd. De toetswaarden (nibbles, halve data-bytes) ziet men van rechts uit de twee datadisplays inschuiven, net als in de datamode van de standaardmonitor.
- We zien dat, als het om een bepaald adres gaat, eerst het linker adresbyte ADH aan bod is en dan het rechter adresbyte ADL. Dat is handiger dan de L-H-volgorde omdat je een adres opschrijft, of "in je hoofd zit", in de volgorde H-L.

Eigenlijk is die toets PAR, en daarmee de overige vier toetsfuncties van TM, een luxe. Maar geen overbodige luxe. Dat de lees- en schrijf-subroutines essentieel bij TM horen zal duidelijk zijn, maar al die poespas met PAR en figuur 10? Natuurlijk is het zo dat alle negen parameters zijn gehuisvest in een RAM-geheugenplaats in pagina 00 of 1A (zie tabel 1). Natuúrlijk kun je de parameter-data ook gewoon opgeven via de standaard-monitor, dus via de toetsen AD, DA en +. Maar vergissingen zijn niet uitgesloten: was 00E2 nou voor BEGADL of voor BEGADH? Die vraag zoeken we op . . . Wat is er nou handiger dan al die negen parameters, die je trouwens niet altijd allemaal nodig hebt, op zo'n handig PAR-rijtje te zetten!?

② Toets SAVE (ex-AD; ex-INSERT)

Datazendingsopdracht

Dat SAVE riekt naar redden, bewaren. Klopt. Er wordt namelijk een datablok als datazending naar de band gestuurd. Dat gaat zo:

- geef een programmanummer ID op (01 . . . FE, dus **niet 00 en FF**);
- geef een startadres SA op;
- geef een eindadres EA op; EAH en EAL zijn gebaseerd op een adres dat één hoger is dan het laatste adres LA van het te verzenden datablok;
- zet een (direkt aansluitend) leeg, onbespeeld bandgedeelte klaar. Pas op

Tabel 1. Geheugenplaatsen in pagina 1A van TM. Dit soort "temporaries" (óók die van de standaard-monitor, de editor, de assembler en van PM) nooit overschrijven tijdens het teruglezen van een datazending!

TEMPORARY DATA BUFFERS

SY	*	\$1A69	SYN COUNTER
BYTE	*	\$1A6A	BYTE FROM TAPE
CHAR	*	\$1A6B	CHARACTER FROM TAPE
HIGHER	*	\$1A6C	3600 HZ HALF PERIODE DELAY
LOWER	*	\$1A6D	2400 HZ HALF PERIODE DELAY
CHKL	*	\$1A6E	CHECK SUM LOW
CHKH	*	\$1A6F	CHECK SUM HIGH
SAL	*	\$1A70	START ADDRESS LOW
SAH	*	\$1A71	START ADDRESS HIGH
EAL	*	\$1A72	END ADDRESS LOW
EAH	*	\$1A73	END ADDRESS HIGH
SYNCNT	*	\$1A74	
BITS	*	\$1A75	AMOUNT OF BITS
FIRST	*	\$1A76	HALF PERIODE AMOUNT OF 3600 HZ
SECOND	*	\$1A77	HALF PERIODE AMOUNT OF 2400 HZ
GANG	*	\$1A78	TEMP OF PBD-BITS
ID	*	\$1A79	ID OF THE DUMPT PROGRAM
NMI	*	\$1A7A	NMI VECTOR

- voor het "overspelen" van een oude datazending, die moet worden bewaard (desnoods via de ingebouwde luidspreker van de cassette-recorder kontroleren of er geen 2400 en 3600 Hz is te horen);
- noteer de stand van de programmateller, ID, SA en EA. Deze gegevens kunnen eventueel ook deel uitmaken van het vooraf op de band ingesproken commentaar;
 - zet de cassette-recorder in de opnamestand. Zorg voor een voldoende hoog opnameniveau. Start de band;
 - spreek eerst een eventueel commentaar in of druk meteen SAVE in. Er wordt nu via de subroutine DUMP/DUMPT een datazending op de band gezet. De rode OUTPUT-LED (D5) licht op. De zes displays zijn gedoofd. Wacht totdat de junior-computer zich terugmeldt met "id XX", waarbij XX het opgegeven programmanummer is;
 - stop de cassette-recorder.

Dat is dus de praktische gang van zaken bij het in elkaar en op de band zetten van een datazending. Indien men beschikt over een cassette-recorder met afstandsbediening start de band automatisch na het indrukken van SAVE (wél vooraf in de opnamestand zetten!). Het inspreken van commentaar moet dan gebeuren bij uitgeschakelde afstandsbediening; eventueel een doorverbindingsplug in de uitgang voor afstandsbediening steken. Direkt nadat de datazending op de band is gezet verschijnt na het indrukken van RST het eindadres EA in beeld, met data. Dit is het eerste adres (SA) van een eventueel later te verzenden aansluitend datablok.

③ Toets GET (ex-PC; ex-SEARCH)

Ophalen wat we zoeken

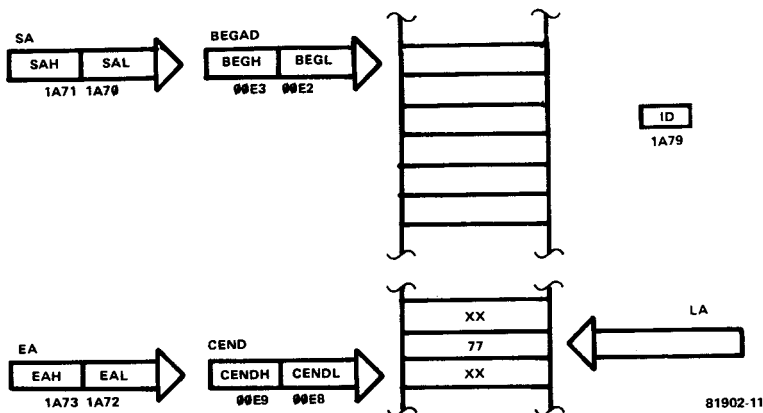
Het indrukken van deze toets zorgt ervoor dat, met behulp van de subroutine RDTAPE een tot een bepaalde datazending behorend datablok van de band wordt gelezen en in het geheugen van de junior-computer wordt geschreven. We gaan als volgt te werk:

- geef een programmanummer ID op. De mogelijkheden variëren van 00 tot en met FF, hoewel er op de band uitsluitend programmanummers 01 . . . FE voorkomen. Doet men dit niet, dan volgt ID uit de oude inhoud van geheugenplaats ID (plaatje ① van figuur 10);
- geef in het geval ID=FF een startadres SA op. Laat men dit achterwege, dan volgt SA uit de oude inhoud van SAH en SAL (plaatjes ② en ③ van figuur 10). Direkt na het opstarten van TM zijn SAH en SAL nul, zodat er sprake is van een startadres 0000;
- plaats de cassette waarop de gewenste datazending zit in de cassette-recorder. In het geval ID=01 . . . FE is het niet zo belangrijk dat men de exakte plaats kent van de datazending; als men maar vroeg genoeg op de band begint. In de gevallen ID=00 of FF moet men de band vlak voor de gewenste, door de junior-computer op te nemen datazending klaar zetten. Immers, in deze gevallen wordt het eerste de beste datablok ingelezen en ingeschreven;
- druk de toets GET in. De groene INPUT-LED licht op;
- zet de cassette-recorder in de weergavestand. Start de band.
(Nu verschijnen de plaatjes van figuur 7 op het display. Op een gegeven moment is plaatje ③ van figuur 7 te zien: het gewenste datablok gaat het geheugen in.)
Indien het datablok volledig is ingelezen en ingeschreven meldt de junior-computer zich op het display terug met de boodschap: "id XX", waarbij XX het vooraf opgegeven programmanummer is — mits de data foutloos is ingelezen. Zie het einde van dit hoofdstuk;
- stop de cassette-recorder.

Zó gaat het als alles goed is, dus als er bij het CHKH/CHKL-gebeuren geen fouten zijn ontdekt. Is dat wél het geval, dan volgt geen terugkeer uit de subroutine RDTAPE en volgt er geen terugmelding "id XX" op het display.

Nadat een datablok is ingelezen en nadat men met de PAR-toets allerlei parameters bekijkt is alleen de inhoud van ID (①) en — in het geval ID=FF — de inhoud van SAH en SAL (② en ③) relevant voor het zojuist ingelezen datablok. De inhoud van EAH en EAL (④ en ⑤ van figuur 10) heeft waarschijnlijk geen enkele gelijkenis met de EA van het zojuist ingelezen datablok; ze spelen alléén bij schrijven op de band (toets SAVE) een rol. N.B. Indien men direkt na het inlezen van een datablok de toets RST indrukt (terug naar standaardmonitor) verschijnt het bij de ingelezen datazending horende eindadres EA (volgt indirekt uit SA en de grootte van het datablok) in beeld, met data. Dat wil zeggen: zolang het niet om een ID FF ging.

Is er gekozen voor een programmanummer FF, dan gebeurt er na het inlezen nog iets meer. De adreswijzer POINT, die is gebruikt bij het



Figuur 11. De situatie van figuur 2, maar nu in het geval van een editor-datazending, die na het indrukken van toets SEF van TM op de band wordt geschreven en ha het indrukken van GET kan worden teruggelezen van de band.

schrijven van het datablok in het geheugen, wordt met één verlaagd; vervolgens wordt het startadres (SAH, SAL) gelijk gemaakt aan de met één verlaagde POINT (zou men dus RST indrukken dan verschijnt het laatste adres LA van het zojuist ingelezen datablok in beeld, met data).

Waarom die verlaging met één en waarom het gelijk maken van het startadres aan het laatste adres LA van het zojuist ingelezen datablok? Antwoord: om uit meerdere datazendingen bestaande **ge-edite programma's** (die nog niet zijn geassembleerd en dus nog labels bevatten) **naadloos aan elkaar te plakken**. **Let wel: wil men gewone programma's aan elkaar plakken, dan moet men het nieuwe startadres met één verhogen omdat anders de inhoud van het laatste adres van het oude datablok wordt overschreven door de inhoud van het eerste adres van het nieuwe datablok!**

Een verklaring hiervoor geven we aan de hand van figuur 11. Op basis van een startadres SA, gelijk aan BEGAD en een eindadres EA, gelijk aan CEND gaat een ge-edit programma als datazending de band op (zie de bespreking van toets SEF). De variabele eindadreswijzer CEND staat gericht op de eerstvolgende vrije geheugenplaats. Eén plaatsje hoger (dus één adres lager) staat het EOF-teken 77. Als nu meerdere ge-edite blokken aan elkaar moeten worden geplakt moeten de tussenliggende EOF-tekenen worden verwijderd zonder dat er gaten vallen. Vandaar die overlapping van één geheugenplaats via het verlagen van POINT en het daaraan gelijk maken van het nieuwe startadres.

④ Toets EDIT (ex-DA; ex-DELETE)

Koude start editor

Indrukken van deze toets levert hetzelfde op als AD 1 C B 5 GO, oftewel de **koude** start van de editor, voor het "van de grond af" editen van een

programma. De opgave vooraf van BEGAD en ENDAD doet men nu met die handige PAR-toets (plaatjes ⑥ . . . ⑩ van figuur 10). Na het indrukken van EDIT verschijnt het EOF-teken 77 op het display. Al met al niets nieuws, alleen gaat het koud opstarten van de editor nu iets sneller en comfortabeler in zijn werk.

⑤ Toets SEF (ex-GO; ex-INPUT)

Deze toetsfunctie heeft vele eigenschappen gemeen met SAVE. Het gaat namelijk om: Save Edited File, schrijf een ge-edit programma op de band. Dus alle data van BEGAD (=SA) tot CEND (=EA), dus tot en met EOF=77 (zie figuur 11).

De toetsfunctie SEF gaat er van uit dat men zojuist met de editor heeft gewerkt en dat de variabele eindadreswijzer CEND in overeenstemming is met de laatste stand van zaken. **De junior-computer is dus niet tussentijds uitgeschakeld.** Hoe gaan we te werk?

- ga via de standaardmonitor naar TM:
RST Ø 8 1 Ø GO XX
Na GO is TM opgestart. Er verschijnt plaatje ① van figuur 10 op het display; XX is het ingetoetste gekozen programmanummer. Dat is het enige dat men vooraf moet opgeven, omdat SA en EA automatisch vastliggen via BEGAD, respectievelijk CEND;
- Zet een (direkt aansluitend) leeg, onbespeeld bandgedeelte voor. Pas op voor het "overspelen" van een al op de band aanwezige datazending, die moet worden bewaard;
- noteer de stand van de programmateller en de gekozen ID; **SA en EA worden pas na het voltooien van de datazending genoteerd!** Dit omdat pas na het indrukken van SEF SAH, SAL, EAH en EAL worden geladen met respectievelijk BEGADH, BEGADL, CENDH en CENDL;
- zet de cassette-recorder in de opnamestand. Zorg voor een voldoende hoog opnamenivo. Start de band;
- druk SEF in. Er wordt nu via de subroutine DUMP/DUMPT een datazending op de band gezet. De rode LED OUTPUT (D5) licht op. De zes displays zijn gedoofd. Wacht totdat de junior-computer zich terugmeldt met de eerste instructie van het ge-edite programma dat zojuist de band is opgegaan. Dit omdat direkt na het zenden de displaywijzer CURAD gelijk wordt gemaakt aan BEGAD en vervolgens gesprongen wordt naar de editor (**warne start**).
- stop de cassette-recorder;
- noteer SA en EA van het zojuist verzonden editor-datablok. Er geldt: SAH=BEG(AD)H; SAL=BEG(AD)L; EAH=END(AD)H en EAL=END(AD)L. **Men heeft deze gegevens alleen nodig indien men bij het teruglezen van een editor-datablok nog gebruik moet maken van de editor (via een warme start), in plaats van het teruggelezen editor-datablok direkt na terugkeer te assembleren.** De gegevens over SA en EA zijn als volgt te pakken te krijgen:

RST	van editor terug naar monitor
Ø 8 1 Ø GO	(ID)
PAR	(SAH)

PAR (SAL)
 PAR (EAH)
 PAR (EAL)
 PAR PAR (BEGH) (BEGL); zie SAH en SAL
 PAR PAR (ENDH) (ENDL); eventueel eveneens noteren
 Wil men direkt na het noteren terug naar de editor **via een warme start**
 (dat is het geval na SEF als men geen uitstapje maakt voor het noteren)
 dan kan worden volstaan met:
 RST 1 C C A GO

Er hoeft geen BEGAD en geen ENDAAD te worden opgegeven. De
 variabele eindadreswijzer CEND staat netjes gericht op de lege plaats
 onder de geheugenplaats met de EOF 77 en de displaywijzer CURAD
 valt samen met BEGAD: de eerste instructie van het editor-datablok is
 op het display te zien. Let wel: er is van uitgegaan dat de
 junior-computer niet is uitgeschakeld geweest!
 Waarom is het eigenlijk verstandig om in bepaalde gevallen SA en EA te
 noteren? Daarover nu meer.

Een belangrijke SEF-toepassing

U bent bezig met de ontwikkeling van een gebruikersprogramma. U zet dat
 programma in het (RAM-)geheugen via de editor. Op een gegeven moment
 denkt u zo ver te zijn dat het programma klaar is en, na te zijn geassem-
 bleerd, kan worden getest. Dit soort momenten komt ons bekend voor; de
 ervaring leert dat er heel vaak te vroeg gejuicht is. Er blijkt nog van alles
 niet te kloppen. Na het assembleren zijn de labels uit het programma
 verdwenen. In die vorm is het totaal ongeschikt voor editorgebruik. Er zit
 niets anders op dan het programma opnieuw te editen, van de grond af aan
 (en dan maar hopen dat alles op papier is vastgelegd!). Tenzij . . . Tenzij
 men zo verstandig is geweest om **voordat tot assembleren is overgegaan het
 programma in ge-edite vorm op de band te zetten.**

Hoe gaat dat in zijn werk?

- Als het zover is dat het ge-edite programma aan assembleren toe is, en
 daarna aan het testen, dan gaat men te werk zoals in het hoofdstukje
 over de SEF-toets is verteld. Dus via SEF het editor-datablok op de
 band zetten **en de ID, SA en EA noteren. Plus de bandtellerstand!**
- Na SEF en het noteren (tijdens het verblijf in TM) moet men via de
 standaard-monitor naar de assembler:
 RST 1 F 5 1 GO
- Het programma wordt gestart (AD (startadres) GO). Er zijn twee
 uitslagen mogelijk van de test van het programma: of de computer doet
 wat u wilt, of de computer wil iets wat u niet doet. Dus in orde of niet
 in orde.
- Als het programma niet in orde is moet u eerst maar eens diep na-
 denken over wat er is fout gegaan. Dat kan wel even duren, dus schakel
 rustig de junior-computer uit, dat is goed voor de energiebesparing. U
 heeft immers toch alles wat nodig is genoteerd.
- Op een gegeven moment bent u zo ver dat het programma kan worden
 verbeterd/aangepast, op basis van een voorhanden zijnde ge-edite versie.
 Dus schakel de junior-computer in, druk RST 0 8 1 0 GO X Y (XY is de

ID) en ga verder te werk zoals is beschreven in het hoofdstukje over de GET-toets. Druk vervolgens de toetsen:

RST Ø Ø E 2

DA	P P	BEGADL = PP =genoteerde SAL
+	Q Q	BEGADH = QQ =genoteerde SAH
+	R R	ENDADL = RR =genoteerde ENDADL
+	S S	ENDADH = SS =genoteerde ENDADH
+	P P	CURADL = BEGADL
+	Q Q	CURADH = BEGADH
+	T T	CENDL = genoteerde EAL
+	U U	CENDH = genoteerde EAH
AD	1 C C A GO	warne start editor

Omdat CURAD gelijk is gemaakt aan BEGAD verschijnt na de warme start van de editor de eerste instructie (of label) van het teruggelezen ge-edite programma. Breng via de editor de door u uitgebroede programma-verbeteringen aan en herhaal de beschreven procedure zo vaak als nodig is.

- Indien het programma wél in orde is hebben we de ge-edite versie ervan niet meer nodig, tenzij het programma later nog wordt uitgebreid. In ieder geval kan het goede programma via een datazending op de band worden gezet (dus nadat het geassembleerd is). Verder is programma-dokumentatie op papier geen slechte zaak. U kunt een hex dump maken (zie hoofdstuk 12) of u kunt het programma in assembler-vorm dokumenteren. Dus instructie voor instructie, met een kolom voor de labels, een kolom voor de bytes, een kolom voor de mnemonics en een kolom voor kommentaar (toelichting). In het laatste geval kunt u nog veel gemak hebben van de editor, namelijk om een geassembleerd programma instructie voor instructie te doorlopen met behulp van de SKIP-toets. Dit is in hoofdstuk 5 van boek 2 behandeld, maar we geven hier en nu nog even aan hoe dat gaat:
- Maak CURAD gelijk aan BEGAD, dat is het startadres van het programma.
- Maak CEND gelijk aan ENDAD. Het kan zijn dat ENDAD naar aanleiding van het verbeteren van het programma is verplaatst. Overtuig u ervan dat CEND=ENDAD gericht is op een adres dat minstens twee hoger is dan de laatste instructie van het programma. Anders kan men het programma niet helemaal instructie voor instructie door SKIPpen (er volgt dan de foutmelding EEEEE).
- Start de editor **warm**. Met de SKIP-toets worden achtereenvolgens alle instructies doorlopen.

Waarschuwing. In alle gevallen waar het gaat om een voorhanden zijnd editor-datablok moet men nooit en te nimmer de editor koud starten, bijvoorbeeld via de toets EDIT! Een koude start is alleen nodig als van de grond af aan wordt gestart met editen. Bij een koude start van een al ge-edit blok worden één of meerdere geheugenplaatsen overgeschreven door de EOF 77. Bij een warme start staat het EOF-teken helemaal aan het eind en wordt er geen nieuwe 77 vanaf het begin doorgeschoven.

Nog een opmerking. Als er via ID=FF meerdere ge-edite datablokken worden ingelezen (met 77 aan het eind van elk datablok) geldt als CEND het eindadres van het laatst ingelezen datablok! Denk ook om

een aanpassing van ENDAD. Men kan onder gebruikmaking van het programmanummer FF bij het teruglezen van één of meerdere ge-edite datablokken het (de) datablok(ken) verplaatsen. Het spreekt vanzelf dat de van belang zijnde parameters aan de hand daarvan moeten worden gewijzigd.

De PLL-afregeling

Lezen wat er (op de band) staat

Bij het lezen van de band vormt de PLL de ogen van de junior-computer. Het is dan ook buitengewoon belangrijk dat die PLL zodanig wordt ingesteld dat hij niet verziende is en ook niet bijziende, maar goedziende. Dat betekent dat elk "bandbit" na verwerking door de PLL en door de software van RDTAPE de juiste, oorspronkelijke waarde nul of één moet krijgen. Samenvattend: er moet worden gelezen wat er staat.

De afregeling van de PLL vindt plaats met de precisie-instelpotmeter P1, op de interface-kaart (zie figuur 2 van hoofdstuk 10). De PLL levert een uitgangssignaal dat afwisselend hoog en laag is. Tenminste: zolang P1 niet al te ver in de richting van een uiteinde is verdraaid. Want wat is het geval? Het ingangssignaal bestaat uit afwisselend stukjes 3600 Hz en stukjes 2400 Hz. De PLL-frekwentie zonder ingangssignaal (dat is de zogenaamde "free running" VCO-frekwentie) is afhankelijk van de stand van P1. Wil er sprake zijn van een goed funktionerende PLL, dan moet die "vrije PLL-frekwentie" ergens halverwege tussen 2400 Hz en 3600 Hz liggen, dus pakweg in de buurt van de 3000 Hz. Is namelijk de PLL-frekwentie hoger dan 3600 Hz of lager dan 2400 Hz, dan werkt de PLL niet (hij is dan niet "gelocked") en zal het uitgangssignaal continu hoog of laag zijn.

Dus: het is raadzaam om vóór de PLL-afregeling P1 ongeveer in de middenstand te plaatsen! Dat komt overeen met het vijf keer ronddraaien van de schroef, vanuit een uiterste stand.

De PLL-afregeling houdt in dat op basis van de aanwezigheid van een PLL-uitgangssignaal dat afwisselend hoog en laag is (PLL funktioneert) de tijdsverhouding hoog:laag of laag:hoog twee op één is. Er zijn twee methoden om deze afregeling te realiseren:

- Regel P1 af op basis van de band teruggelezen testdata, waarbij de testdata via visuele informatie op het display op korrekt teruglezen kan worden gecontroleerd. Plaats bovendien P1 in het midden van het regelbereik waarvoor sprake is van korrekt teruggelezen data. Dit is de indirecte methode om te komen tot de genoemde tijdsverhoudingen één op twee of twee op één.
- Regel P1 af op basis van de band teruggelezen testdata, waarbij het PLL-uitgangssignaal met behulp van de oscilloscoop wordt afgeregeld op tijdsverhoudingen één op twee of twee op één van een hoog-periode en een aansluitende laag-periode of van een laag-periode en een aansluitende hoog-periode.

Er is eerder het een en ander verteld over de grote marge die er bestaat tussen de verhouding 2 : 1 voor de ene bitwaarde en 1 : 2 voor de andere bitwaarde. Hieruit mag niet de konklusie worden getrokken dat

P1 ergens in de middenstand kan worden geplaatst en dat de feitelijke afregeling van P1 achterwege zou kunnen blijven. Gewoon doen! Trouwens, het is zo geiept. Nu zullen de beide PLL-afregelmetoden in detail worden besproken.

1. PLL-afregeling via het display

Bij deze afregelmethode is het absoluut noodzakelijk dat het programma TM via de EPROM IC4 is "aangesloten". Trouwens: zonder TM is de PLL-afregeling zinloos, tenzij u zelf cassette-software hebt ontwikkeld. We gaan er van uit dat TM werkt: na RST 0 8 1 0 GO verschijnt er "id 00" op het display. De afregeling, punt voor punt:

- 1). Toets de twee hulpprogramma's van tabel 2 in. Het programma 0200 ... 0250 maakt gebruik van een subroutine van TM. Het zorgt ervoor dat gedurende 4 minuten uitsluitend synchronisatie-karakters op de band worden geschreven (is dus geen volledige datazending). Als dat gebeurd is gaat het terug naar de monitor. Het programma 0251 ... 0283 zorgt voor het teruglezen van de synchronisatie-karakters van de band en heeft als resultaat plaatje ① of ② van figuur 7 op het display; er worden vier TM-subroutines gebruikt.
- 2). Stop een cassette in de cassette-recorder en start deze in de opname-stand. Zet instelpotmeter P2 (op de interfacekaart) in de middenstand en de opname-nivoregelaar — als zo'n ding voorhanden is; bij cassette-recorders met automatische opname-nivoregeling niet — in de middenstand.
- 3). Toets AD 0 2 0 0 GO. De rode opname-LED licht op. Er worden synchronisatie-karakters op de band geschreven.

Tabel 2. Twee hulpprogramma's voor het schrijven op respectievelijk lezen van de band van testdata, ten behoeve van de PLL-afregeling volgens de display-methode.

M

HEXDUMP: 200, 250

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0200:	A9	7D	8D	6C	1A	A9	C3	8D	6D	1A	A9	03	8D	76	1A	A9
0210:	02	8D	77	1A	A9	47	A2	FF	8D	82	1A	8D	78	1A	8E	83
0220:	1A	A9	00	A2	7F	8D	80	1A	8E	81	1A	A9	DD	8D	00	1A
0230:	8D	01	1A	18	A9	01	6D	00	1A	8D	00	1A	A9	00	6D	01
0240:	1A	8D	01	1A	B0	08	A9	16	20	A3	0A	4C	33	02	4C	1D
0250:	1C															

JUNIOR

M

HEXDUMP: 251, 283

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0251:	A9	32	8D	82	1A	8D	78	1A	A9	7E	8D	83	1A	A9	7F	8D
0261:	81	1A	A9	FF	8D	6B	1A	20	C2	0B	6E	6B	1A	AD	6B	1A
0271:	20	E8	0B	C9	16	D0	F0	20	36	0C	20	5D	0C	C9	16	F0
0281:	F6	D0	DF													

JUNIOR

- 4). Na ca. 4 minuten stopt de schrijverij. De rode LED dooft. Het display laat "0200A9" zien. Stop de recorder; spoel ca 4 minuten terug (noteer de stand van de bandteller bij het starten van het schrijfprogramma).
- 5). Start de recorder in de weergavestand (bij gebruik van een luidspreker-uitgang volumeregelaar in de middenstand) en toets AD 0 2 5 1 GO. De groene weergave-LED licht op. Zitten we op een bandgedeelte vóór de synchronisatie-karakters, dan is met zekerheid plaatje ① van figuur 7 te zien (Let wel: P1 moet ongeveer in de middenstand staan!). Weet men met zekerheid dat er synchronisatie-karakters aan bod zijn, dan is óf ② (onwaarschijnlijk) óf ① van figuur 7 te zien. Nu kan worden begonnen met de afregeling van P1.
- 6). Verdraai P1 met behulp van een schroevendraaiertje zodanig dat uitsluitend nog plaatje ② van figuur 7 is te zien. Geen paniek als dat niet lukt; het zou dan best wel eens zó kunnen zijn dat de vier minuten synchronisatie-karakters om zijn. Een kwestie van terugspoelen en opnieuw beginnen (het leesprogramma kan daarbij gewoon ingeschakeld blijven). Bij een juiste instelling van P1 is **gedurende de weergave van synchronisatie-karakters** permanent ② te zien en nooit, al is het maar even, ①.
- 7). Er is een bepaald regelgebied van P1 waarbinnen data (in ons geval synchronisatie-karakters) korrekt van de band wordt gelezen. Zowel door linksom als door rechtsom verdraaien van P1 kan men een positie bereiken waar er leesfouten worden gemaakt (plaatje ① in plaats van plaatje ②). De ideale stand van P1 is die waarbij men evenveel rechtsom als linksom moet draaien om dat te bereiken, dus in het midden van een P1-instelgebied waarvoor korrekt wordt gelezen. Tot zover de eerste methode voor het afregelen van de PLL. De afregelprocedure kan eventueel meerdere keren worden herhaald. Men kan voor de zekerheid een programma in RAM waarvan de data bekend is (dokumentatie op papier!) op de band schrijven (als een echte datazending; zie eerder in dit hoofdstuk), de junior-computer een tijdlang uitschakelen (zodat de RAM het programma verliest) en weer inschakelen en vervolgens het programma teruglezen van de band, en byte voor byte of instructie voor instructie kijken of het overeenstemt met het oorspronkelijke programma.

2. PLL-afregeling via de oscilloscoop

Indien men (eventueel tijdelijk) de beschikking heeft over een oscilloscoop (een dubbelstraalstype is helemaal prachtig) en daar ook nog een beetje mee om kan gaan is er een alternatief waarbij testdata niet uitsluitend via software (methode 1) maar via "hardware" (grofstoffelijke spanningen) op korrekte lezing ervan wordt afgeregeld met de precisie-instelpotmeter P1. Daartoe wordt eerst met een hulpprogramma gedurende een zelf te kiezen tijd een bitstroom op de band geschreven die bestaat uit afwisselend nullen en enen: 0 - 1 - 0 - 1 - 0 - 1, enzovoorts.

De bits van de band worden teruggelezen zonder de hulp van leessoftware. Een oscilloscoop wordt aangesloten op de uitgang van de PLL, dat is pen 7 van IC7 (knooppunt R30/R31) in het schema volgens figuur 2 van

Tabel 3. Een hulpprogramma, waarmee afwisselend nullen en enen op de band worden geschreven (zo lang als men maar wil; afbreken door RST-toets in te drukken), ten behoeve van de PLL-afregeling via de oscilloscoop.

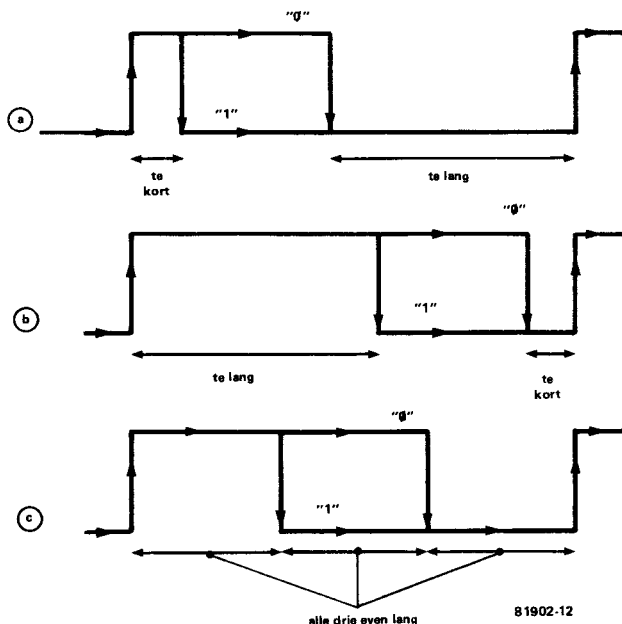
M

HEXDUMP: 200,23F

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0200:	A9	7D	8D	6C	1A	A9	C3	8D	6D	1A	A9	03	8D	76	1A	A9
0210:	02	8D	77	1A	A9	47	A2	FF	8D	82	1A	8D	78	1A	8E	83
0220:	1A	A9	00	A2	7F	8D	80	1A	8E	81	1A	20	C8	0A	20	E5
0230:	0A	20	E5	0A	20	C8	0A	20	C8	0A	20	E5	0A	4C	2B	02
0240:																

JUNIOR

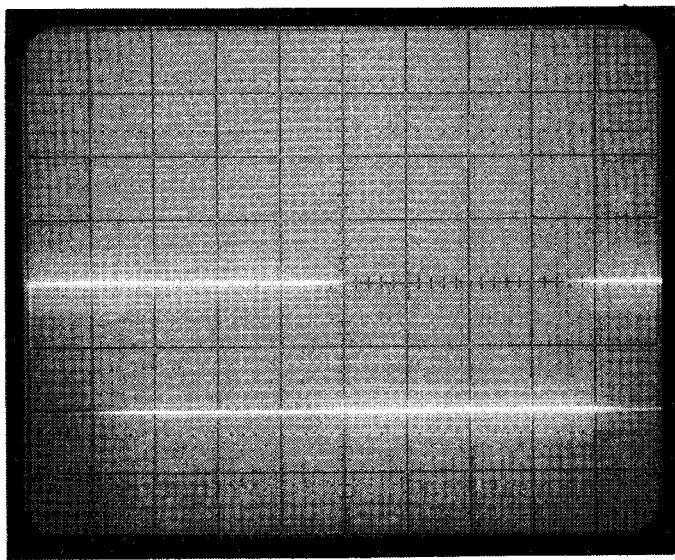
hoofdstuk 10. Met P1 goed afgeregeld moet het skoopplaatje er uitzien als in figuur 12c en de foto van figuur 15. Typische "misstanden" van P1 leveren figuur 12a en de foto van figuur 13, respectievelijk figuur 12b en de foto van figuur 14 op. In alle gevallen gaat het om een goed funktionerende ("gelockte") PLL; daartoe moet P1 vóór de afregeling ongeveer in de middenstand worden gezet.



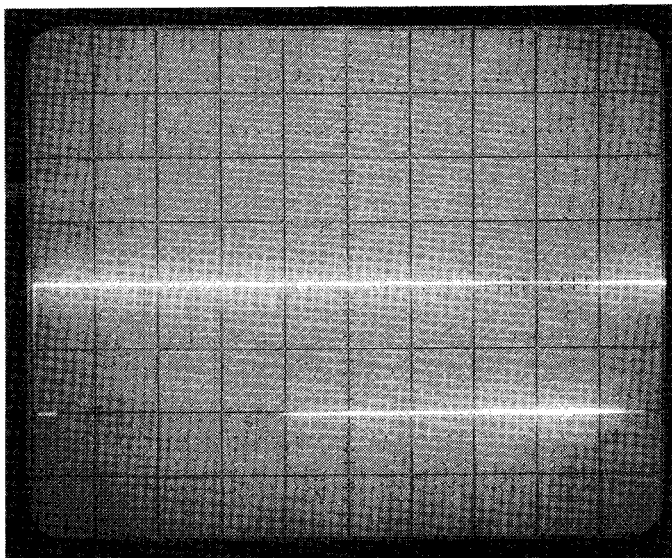
Figuur 12. Drie keer twee bit-signalen, betrekking hebbend op de uitgang van de PLL. Plaatjes (a) en (b) betreffen de niet-afgeregelde, plaatje (c) de afgeregelde PLL. In alle drie gevallen functioneert de PLL (is "gelocked"); dat is het geval als P1 vóór de afregeling ongeveer in de middenstand staat.

De signalen van figuur 12 komen dus overeen met de foto's van figuur 13 ... 15. Figuur 12 is naast de foto's in dit hoofdstuk opgenomen omdat we liever niet in foto's zitten te kalken met pijltjes en zo, en bovendien omdat de foto's een overlapping te zien geven op die tijdstippen waar sprake is van een verandering van logisch nivo. Dat komt door het gezamenlijke effect van de PLL-jitter en de niet oneindig korte belichtings-tijd. In alle gevallen gaat het om skoopplaatjes. Indien er achter elkaar afwisselend enen en nullen van de band worden gelezen zijn er schijnbaar tegelijkertijd en onderling even helder een bit-één-sig-naal én een bit-nul-sig-naal zichtbaar. Op de foto's beslaat een bittijd negen hokjes; een kwestie van de variabele tijdbasis gebruiken. In alle drie gevallen van figuur 12 is het bit-nul-sig-naal langer hoog en derhalve korter laag dan het bit-één-sig-naal. Dit is in overeenstemming met het feit dat een logische nul een langere hoge-frekwentiebijdrage heeft dan een logische één; zie de plaatjes ① en ② van figuur 6.

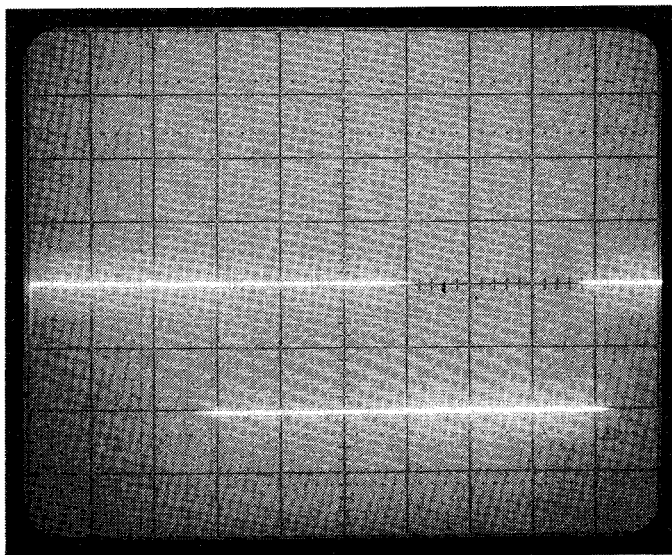
Maar naast overeenkomsten zijn er óók verschillen tussen de drie plaatjes van figuur 12. In figuur 12a duurt de hoog-tijd te kort en dus de laag-tijd te lang voor zowel een bit-één-sig-naal als voor een bit-nul-sig-naal. In figuur 12b duurt de hoog-tijd te lang en derhalve de laag-tijd te kort voor zowel een bit één als een bit nul. De twee genoemde situaties komen overeen met een foute instelling van P1; in het ene geval een instelling, (te ver) linksom verdraaid vanuit de juiste instelling van P1 en in het andere geval een instelling, (te ver) rechtsom verdraaid vanuit de korrekt afgeregelde positie van P1. De situatie bij korrekte afregeling van P1 is weergegeven in figuur 12c en de bijbehorende foto van figuur 15. De tijd waarin het sig-naal altijd hoog is (dus zowel bij een bit nul als een bit één) is gelijk aan de tijd



Figuur 13. Vergelijk figuur 12a: niet (korrekt) afgeregelde PLL.



Figuur 14. Vergelijk figuur 12b: niet (korrekt) afgeregelde PLL.

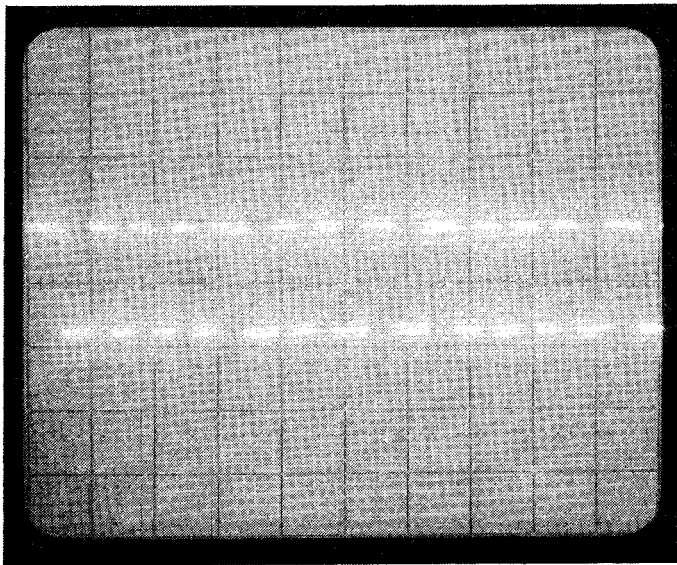


Figuur 15. Vergelijk figuur 12c: korrekt afgeregelde PLL.

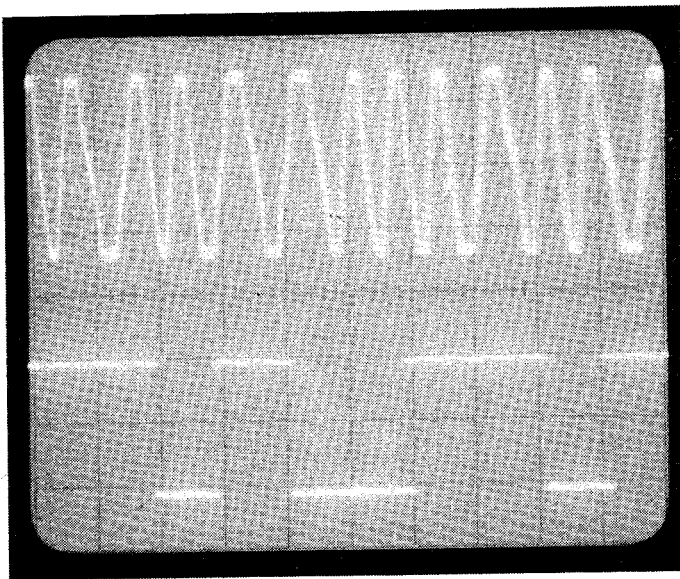
waarin het signaal altijd laag is, en verder gelijk aan de tijd waarin het signaal hoog (bit nul) of laag (bit één) is; die laatste tijd is trouwens nagenoeg konstant voor alle drie situaties van figuur 12.

Tot zover de eigenlijke PLL-afregeling volgens de tweede methode, dus met de oscilloscoop. Nu nog wat aanvullende gegevens. In figuur 16 staan zeven tekeningetjes bij elkaar, die elk drie pijltjes hebben. Eén pijl stelt de VCO-frekwentie van de PLL voor, de andere twee de lage en de hoge frekwentie waarmee de bits worden gekodeerd (zie figuur 6a), dus de twee PLL-piano-noten van figuur 5. In figuur 16a gaat het om de situatie bij korrekt afgeregelde PLL. De figuren 16b en 16c horen bij een korrekt funktionerende maar niet korrekt afgeregelde PLL. De figuren 16d en 16e horen bij een niet korrekt funktionerende PLL. Dit soort toestanden krijg je als P1 te ver in een uiterste stand staat.

Figuur 16f lijkt op 16b en treedt op als data wordt opgenomen op de cassette-recorder A en weergegeven via cassette-recorder B, waarbij de bandsnelheid van B 10% hoger is dan die van A. De nominale snelheid is 4,76 cm/sec, maar er kunnen verschillen optreden, hoewel die 10% van het voorbeeld lichtelijk overdreven is. Figuur 16g toont de omgekeerde situatie, met B 10% trager dan A. Dit lijkt erg veel op figuur 16c. Bij verschillende bandsnelheden moet eigenlijk de PLL opnieuw worden afgeregeld: zie de gestippelde VCO-pijlen in de figuren 16f en 16g. Dat kan natuurlijk maar is niet strikt nodig. En wel omdat de snelheidsverschillen wel loslopen. Verder is er de al besproken grote tolerantie van de leessoftware van TM voor van de nominale verhouding afwijkende hoog/laagverhoudingen. En



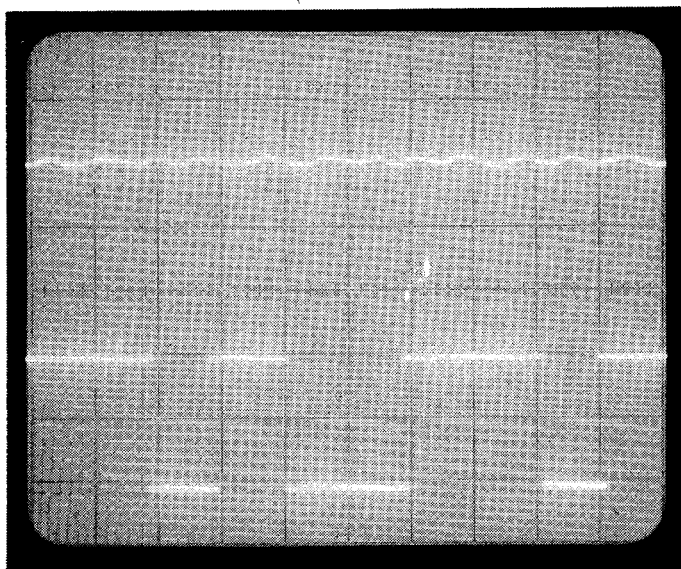
Figuur 17. Uitgangssignaal op de OUTPUT-aansluiting J2 van de junior-computer tijdens het naar de band zenden van afwisselend enen en nullen (testprogramma van tabel 3). De nivoregelaar P2 is helemaal opengedraaid (vertikaal: 2 V per hokje).



Figuur 18. Ingangssignaal op de INPUT-ingang J1 van de junior-computer tijdens het van de band lezen van afwisselend enen en nullen (bovenste spoor); PLL-uitgangsspanning (onderste spoor). De volumeregelaar van de cassette-recorder is op maximale weergavesterkte ingesteld (beide sporen: vertikaal 2 V per hokje).

ten derde: de afregeling van de PLL volgens het criterium van figuur 12c zorgt ervoor dat P1 zich precies midden in een instellingsgebied bevindt waarvoor het korrekt lezen van data van de band gewaarborgd is (bij een gelijke bandsnelheid voor opname en weergave; heeft u een aparte OUTOUT- en INPUT-recorder, dan is de PLL optimaal afgeregeld voor die combinatie, met eventuele verschillen in bandsnelheid). Omdat we precies midden in een veilig gebied zitten is er een marge voor zowel lagere als hogere absolute bandsnelheden. Er is, statistisch gezien de gulden middenweg gekozen. N.B. Dit is de belangrijkste reden om bij afregelmethode 1 niet zo maar een instelling van P1 te nemen waarbij alles goed gaat, maar een positie van P1 precies midden in een afregelgebied waarbij alles goed gaat.

Tot slot van dit PLL-verhaal nog wat foto's. Figuur 17 laat ons het uitgangssignaal zien (op de loper van de helemaal opengedraaide potmeter P2), dat ontstaat tijdens het afwisselend nullen en enen schrijven op de band, met het programma van tabel 3. Men ziet heel mooi de stukjes hoge frekwentie (ongeveer drie halve perioden in een hokje) en de stukjes lage frekwentie (ongeveer één hele periode in een hokje). Men ziet ook het onderscheid tussen een bit nul en een bit één. Figuur 18 laat op het bovenste spoor het uitgangssignaal van de cassette-recorder zien (op J1); het onderste spoor laat ons de uitgangsspanning van de PLL zien. U ziet dat de cassette-recorder een soort sinussen maakt van die mooie blokken



Figuur 19. Als figuur 18, maar nu met een tot minimale proporties teruggebracht ingangssignaal op J1 (beide sporen: 2 V per hokje vertikaal). De PLL funktioneert uitstekend binnen een enorm groot ingangsspanningsbereik. In verband met mogelijke storingen rond de werking van de cassette-recorder (drop-outs en dergelijke) is het verstandig om altijd te zorgen voor een voldoende groot ingangsnivo op J1: liever figuur 18 dan deze figuur, want je moet nooit de PLL-goden verzoeken.

van figuur 17. De PLL werkt echter op basis van frekwentie-informatie en dat is prima in orde.

Is er in figuur 18 sprake van een dijk van een ingangssignaal (volumeregeelaar van de cassette-recorder helemaal opgedraaid), in figuur 19 is het ingangssignaal drastisch verkleind, hetgeen de PLL buitengewoon weinig doet. Hetgeen prima is.

Wat er nog meer bij komt kijken

Een aantal praktische wenken

1. Welke cassette-recorder(s)?

Koop niet de goedkoopste recorder, maar neem óók geen te dure. Mechanische duurzaamheid is belangrijker dan een veelheid aan allerlei oxideknoppies. Neem gerust een mono-recorder. Een bandteller is reuze handig en dus een "must". Verder is een 8-ohm-luidsprekeruitgang, waarbij de ingebouwde luidspreker wordt uitgeschakeld, geen overbodige luxe. Er is dan een, met de volumeregeelaar instelbare, flink grote uitgangsspanning beschikbaar. Dat is alleen maar voordelig. Zo'n luidsprekeruitgang wordt

meestal doorverbonden via een telefoonstekker $\varnothing = 3,5$ mm. Een ingang voor afstandsbediening (remote control; starten van de band) is eigenlijk wel handig, maar uitsluitend bij gebruik van gescheiden recorders OUTPUT en INPUT voor opname en weergave. Deze ingang wordt meestal doorverbonden via een telefoonstekker $\varnothing = 2,5$ mm.

2. Hoe één recorder aansluiten?

Verbind de opname-ingang van de cassette-recorder (meestal DIN; pin 3 is "signaal", pin 2 massa) met J2 (Cinch-stekker). Verbind de luidspreker-uitgang (telefoonstekker $\varnothing = 3,5$ mm) met J1 (Cinch-stekker). De aansluitingen voor afstandsbesturing J3 en J4 worden niet gebruikt.

N.B. Het is mogelijk om met één recorder (met een stuuringang voor remote control) toch gebruik te maken van afstandsbediening. De twee kontakten van J3 en J4 moeten dan doorverbonden zijn; tevens moeten beide relais, Re1 en Re2, op de interface-kaart zijn aangesloten, via 14-pens IC-voeten. Doorverbinding via twee draden plus Cinch-stekker plus telefoonstekker $\varnothing = 2,5$ mm.

3. Twee recorders

Men kan gebruik maken van twee recorders: een aparte recorder OUTPUT voor het schrijven op de band van datazendingen, en een aparte recorder INPUT voor het teruglezen van de band van datazendingen. Tussen de twee recorders zal een levendig verkeer plaatsvinden van cassettes. Met name bij het gebruik van twee recorders is afstandsbediening reuze handig. Van de junior-computer gaan vier verbindingen naar de twee recorders: a) luidsprekeruitgang INPUT naar J1; b) opname-ingang OUTPUT naar J2; c) remote control ingang van INPUT naar J3; d) remote control ingang van OUTPUT naar J4. Voor bijzonderheden over de verbindingen: zie de punten 1 en 2.

De INPUT-recorder moet permanent in de stand PLAY (weergave) staan, de OUTPUT-recorder permanent in de stand RECORD (opname). Een doorverbinding tussen de twee kontakten van J3 of J4 zorgt ervoor dat de band van de betreffende cassette-recorder gaat lopen.

Bij afregeling van de PLL in de situatie met twee cassette-recorders schrijft men de testdata via het bijbehorende hulpprogramma (tabel 2, eerste helft; of tabel 3) op de cassette-band die wordt afgedraaid op de OUTPUT-recorder. Nadat dit is gebeurd haalt men de cassette uit de OUTPUT-recorder en zet hem in de INPUT-recorder. Na terugspoelen kan dan worden overgegaan tot één van de eerder beschreven PLL-afregelmethoden. De PLL wordt optimaal afgeregeld: er wordt volledig rekening gehouden met eventuele verschillen in absolute bandsnelheid (nominaal 4,76 cm/sec) tussen de INPUT- en OUTPUT-recorder.

Direkt na het indrukken van SAVE, GET of SEF start de band van één van de twee recorders. Vooral bij SAVE en SEF moet de band snel op snelheid zijn, omdat anders het begin van een datazending vanwege het nog niet op snelheid zijn met een schijnbaar hogere bitfrequentie wordt opgetekend. Een voorbeeld. Stel we zitten op halve snelheid: ca 2,4 cm/sec. Frequenties van 2,4 kHz en 3,6 kHz uiten zich dan bij volle snelheid (weergave) als 4,8 kHz en 7,2 kHz, en daar bakt de beste PLL van de wereld niets van.

Gelukkig begint elke datazending met 255 synchronisatie-karakters (tijdsduur ca. 2½ sec) en daar kunnen we er best 50 à 100 van missen: een halve à een hele sekonde voor het starten van de lopende band.

4. Werken op nivo

Goed opnemen en goed weergeven

Het is erg belangrijk dat elke op de band geschreven datazending voldoende sterk is opgenomen, en dat elke van de band teruggelezen datazending voldoende sterk wordt weergegeven. Waarom? Omdat er allerlei kapers op de kust zijn die ervoor zorgen dat het uitgangssignaal van de cassette-recorder — dat als ingangssignaal voor de PLL dient — tijdelijk verzwakt wordt. En nu is de PLL weliswaar met weinig tevreden (zie bijvoorbeeld figuur 19, en vergelijk dat met figuur 18), maar als er onder het minimum-ingangsnivo wordt gedoken, al is het maar heel even, dan gaan er bits van de datazending de mist in en wordt er van de band teruggelezen met spellingsfouten. Dat kan nooit de bedoeling zijn.

Er zijn twee dingen waarvoor men moet oppassen: verlies van hoge tonen en zogenaamde "drop-outs". Bij een verlies aan hoge tonen moeten we vooral denken aan een sterke verzwakking van de 3600 Hz-toontjes ten opzichte van de 2400 Hz-toontjes. De oorzaken hiervan kunnen zijn een vuile en/of scheef staande opname/weergavekop, of een versleten kop; óf het feit dat de datazending te zwak op de band is opgenomen: bedenk dat de gekombineerde opname/weergave-frekwentiekarakteristiek van cassette-band een verzwakking voor hoge tonen oplevert die groter is naarmate er zwakker wordt opgenomen!

Dan de drop-outs. Het gaat hierbij om een bij herhaling optredende terugval of uitval van het weergavesignaal van de recorder. Vaak gebeurt dat maar "heel eventjes": nauwelijks te horen, maar ondertussen zijn waarschijnlijk enige tientallen bits de revue gepasseerd. Gelukkig treedt een totale signaaluitval hoogst zelden op; meestal is de terugval een kwestie van een signaalverzwakking van 10 à 20 keer (in vaktaal: 20 à 26 dB). En hieruit trekken we een belangrijke konklusie: **zorg ervoor dat het aan de PLL toe te voeren uitgangssignaal van de recorder enige tientallen malen sterker is dan absoluut nodig is!** De kans dat een drop-out dan fatale gevolgen heeft in die zin dat er bits de mist ingaan is dan uiterst miniem, om niet te zeggen nihil. Neem gerust een uitgangssignaal à la figuur 18, bovenste spoor, met een 5 tot 6 volt (top-topwaarde) signaalsterkte. Heus, de PLL gaat er niet van kapot! En nu weet u ook waarom de voorkeur moet uitgaan naar een cassette-recorder met een luidsprekeruitgang. Er is veel meer spanning aanwezig dan op een lijnuitgang (gewoon maar de volumeregelaar wijddopen zetten). Verder is een lijnuitgang weliswaar laagohmig, maar door de lage ingangsimpedantie van de PLL (een paar kilo-ohm; de prijs die we moeten betalen voor een hoge gevoeligheid van de PLL, dat wil zeggen: nauwelijks signaalverzwakking tussen J1 en de eigenlijke PLL-ingang — dit in tegenstelling tot de KIM) treedt toch nog de nodige verzwakking op.

N.B. Weerstand R37 vervalt bij gebruik van een lijnuitgang!

Is er een hoge-tonenregelaar ("treble") op de cassette-recorder aanwezig en wordt de luidsprekeruitgang gebruikt, zet deze dan gerust helemaal open. Okay, het weergavenivo, daar hebben we onze maatregelen getroffen. En het opnamenivo? Verwaarloos ook dat niet. Nog los van een eventueel hoge-tonenverlies is het niet verstandig om zwak op te nemen en er maar van uit te gaan dat het bij de weergave allemaal wel weer in orde komt. Zet maar rustig de potmeter P2 op de interface-kaart helemaal open (zie figuur 17, met zijn dik $4\frac{1}{2}$ volt top-top), met name bij cassette-recorders met automatische opname-nivoregeling (waarbij een sterkteregelaar dus ontbreekt). Veel van dat soort recorders is uitgerust met een ingebouwd mikrofoontje. We hebben het bij de diverse prototypes van de uitgebreide junior-computer al vaak gepresteerd om een datazending op te nemen en allerlei sterke signalen door te geven aan de mikrofoon, waarbij het teruglezen van de datazending geen centje pijn opleverde! Ook zijn wel eens massaverbindingen vergeten (brom!); dit bleek dan achteraf. Uiteraard zijn dit allerlei geprovoceerde situaties en het is niet de bedoeling om u aan te raden om hetzelfde te doen: nóóit de goden verzoeken! Het is maar om aan te tonen dat datazendingen bij versturing en ontvangst tegen een stootje kunnen — mits uw opname- en weergavewerk van hoog nivo is. Zorg er verder voor dat de opname/weergavekop niet vuil wordt of blijft, en in positie blijft. Gebruik een bepaalde cassette niet tientallen malen achter elkaar en neem eens een keer een nieuwe; die dingen hebben ook niet het eeuwige leven en zo duur zijn ze nu ook weer niet. Neem een programma op twee cassettes op (reservekopie) of maak na verloop van tijd een nieuwe datazending van dat programma op een verse cassette-band. En als u zich na het in acht nemen van alle raadgevingen van dit punt 4 toch nog onzeker voelt (dus als u lijkt op dat jonge vogeltje dat vlieglessen neemt met een parachute) dan kunt u altijd nog het volgende doen: schrijf van een programma meerdere datazendingen achter elkaar op de band, eventueel op verschillende cassettes. Dat gaat door het herhaald indrukken van de toets SAVE of SEF. Het teruglezen (toets GET) moet u dan natuurlijk ook herhaalde malen doen.

U weet nu zo'n beetje alles van wat er komt kijken bij het gebruik van de cassette-band als notitieblok voor grote hoeveelheden bytes. U weet hoe datazendingen in elkaar zitten. U weet van hoofdstuk 10 hoe de PLL werkt. U heeft de gebruiksaanwijzing van TM leren kennen en weet hoe u de PLL de kunst kunt leren om data te lezen zonder letters of woorden over te slaan of zonder van een a een e te maken. U heeft al een aantal praktische raadgevingen gekregen. U hebt echter nog niet kunnen lezen hoe u de cassette-hardware en -software in gebruik kunt nemen — nadat de uitgebreide junior-computer is gebouwd en nadat eventuele troebelen zijn geschoten. Daarover kunt u nu het een an ander lezen.

5. Alles okay? Anders eerst okay maken!

Het lezen en schrijven aan de praat krijgen

We gaan ervan uit dat de gebouwde uitgebreide junior-computer globaal is gecontroleerd volgens de richtlijnen die in hoofdstuk 10 zijn besproken. We beginnen maar eens met de software.

Doet TM het? Deze vraag kan al voor een groot deel worden beantwoord zonder dat daarvoor de aansluiting van een cassette-recorder nodig is. We gaan er van uit dat een 2716-EPROM is geprogrammeerd met TM en dat die 2716 is gestoken in IC-voet nummer 4 (niet IC5!), en verder dat de positie van het IC en van alle 24 pennen in orde is. We toetsen nu AD 0 8 1 0 GO (het startadres is 0810, niet 0800!) Er moet nu "id 00" in beeld verschijnen. Drukt men EDIT in dan verschijnt er 77 in beeld (wél ENDAD hoger maken dan BEGAD). Drukt men SAVE in dan licht de rode OUTPUT-LED D5 op; de kontakten van J4 zijn doorverbonden (kontrolleren met ohm-meter of verbindingstester) (mits Re2 is aangesloten!). Men kan zonder aangesloten cassette-recorder(s) een datazending wegschrijven ("droogschrijven"); zorg ervoor dat EA groter is dan SA en druk SAVE in. Er moet de terugmelding "id XX" volgen. Op dezelfde manier kan men de werking van toets SEF controleren. Zet met behulp van de editor een fantasieprogramma in elkaar, in RAM. Denk daarbij aan de spelregels voor BEGAD en ENDAD. Druk vervolgens SEF in. Er moet na enige tijd de eerste instructie van dat fantasie-programma te zien zijn. Ook nu moet na het indrukken van SEF de rode OUTPUT-LED oplichten en moeten de kontakten van J4 zijn doorverbonden.

En als dit allemaal niet in orde is? Controleer de inhoud van de TM-EPROM aan de hand van de hex dump van Aanhangel 4 en met behulp van de toetsen AD en + (terugkeer naar monitor via RST). Controleer de hardware rond T3. Zijn de verbindingen (vijf stuks) tussen de interfacekaart en de poortkonnektor van de standaardkaart wel allemaal, en goed gemaakt?

We kunnen zonder aangesloten cassette-recorders niet alleen "droogschrijven", maar ook "drooglezen". Na het indrukken van toets GET moet de groene INPUT-LED oplichten; mits Re1 niet is aangesloten moeten nu tevens de kontakten van J3 zijn doorverbonden. Na het indrukken van GET is plaatje ① van figuur 7 te zien, met name als men zorgt voor wat stoorspanningen in de buurt van J1 (TL-buizen o.i.d.). Het display zal hoogst waarschijnlijk zwak zijn en men kan duidelijk Di5 en Di6 elkaar zien afwisselen.

Gaan we ervan uit dat bij dit droog oefenen alles in orde is bevonden, dan kunnen we nu voorbereidingen treffen voor de afregeling van de PLL. De of beide cassette-recorders worden aangesloten. Zie de eerder gegeven gebruiksaanwijzingen. Er wordt nu, afhankelijk van de te kiezen afregelmethode, of de twee hulpprogramma's van tabel 2, of het schrijfprogramma van tabel 3 ingetoetst. Vervolgens wordt de betreffende afregelprocedure (zie eerder in dit hoofdstuk) uitgevoerd. Als dat niet lukt — wat dan?

- Controleer of het (de) testprogramma's goed is (zijn) ingetoetst.
- Kijk na of instelpotmeter soms niet helemaal dichtgedraaid is. Want dan komt er geen testdata op de band!
- Controleer de verbindingen met de cassette-recorder(s).
- Staat P1 vóór de PLL-afregeling wel ongeveer in de middenstand? Want anders funktioneert de PLL niet (is niet "gelocked"; zie de figuren 16d en 16e).
- Is er wel met voldoende nivo opgenomen en weergegeven? Lees anders de eerder gegeven gebruiksaanwijzingen nog eens goed na.
- Controleer de PLL-hardware, dat is IC6, IC7 en omgeving. Te controle-

- ren spanningen aan IC6 en IC7 vindt men in figuur 18 van hoofdstuk 10.
- Is voor de weergave een lijnuitgang van de cassette-recorder gebruikt, en is toch weerstand R37 aangebracht? Meteen verwijderen! Bij gebruik van een lijnuitgang treedt met die R37 een knots van een verzwakking op.
- Beschikt u over een oscilloskoop, probeer dan na te gaan in hoeverre skooopplaatjes te maken zijn zoals de figuren 13 . . . 15 en 17 . . . 19. Het is dan in ieder geval mogelijk om de fout te lokaliseren.

Tot zover enige aanwijzingen voor het foutzoeken. Ze kunnen misschien niet voor de volle honderd procent direkt aanleiding geven tot de oplossing van gerezen problemen, maar mocht u er echt niet uitkomen dan kan Elektuur u technische antwoorden geven op uw schriftelijke of telefonische vragen. Hoe dat precies moet en gaat kunt u in elke uitgave van het tijdschrift Elektuur lezen.

Laatste controle

Nadat alles in orde is bevonden en de PLL is afgeregeld kunt u aan de hand van een test-datablok nagaan of het verzenden en weer ontvangen van een datazending geheel volgens de regels, namelijk korrekt gebeurt. Aan het einde van het hoofdstukje over de display-PLL-afregelmethode is daarover al iets gezegd. Het komt er op neer dat een byte voor byte bekend programma op de band wordt gezet, dat vervolgens de junior-computer wordt uitgeschakeld en dat na het inschakelen het programma weer wordt teruggelezen, ingeschreven in RAM, en gecontroleerd. Omdat die RAM tussentijds bewusteloos is geweest en na het wakker worden aan geheugenverlies blijkt te lijden, is er een absolute garantie dat er goed is teruggelezen als het testprogramma byte voor byte klopt met wat er stond . . . zegt men. Want het is ons al een paar keer overkomen dat data in RAM na het een weekend lang uitgeschakeld zijn van de junior-computer nauwelijks gewijzigd was bij het opnieuw inschakelen!

Datazending fout ingelezen?

Er is het een en ander verteld over de functie van de controle-bytes CHKL en CHKH (zie ook de figuren 3 en 4). De vraag rijst wat er gebeurt als de waarde van CHKL en/of die van CHKH aan het eind van het teruglezen en inschrijven van het datablok in het geheugen niet overeenstemmen met de op de band genoteerde waarde van CHKH en CHKL, die bij het schrijven op de band tot stand zijn gekomen. Overigens: de controle wordt alleen uitgevoerd voor de gezochte en gevonden datazending.

Stel u heeft GET ingedrukt. De cassette-band loopt, met de recorder in de weergavestand. Wordt de gezochte datazending weergegeven, dan krijgt u eerst plaatje ② van figuur 7 te zien (synchronisatie-karakters) en daarna plaatje ③: het gezochte datablok wordt in het geheugen van de junior-computer geschreven. Zodra het hele datablok is ingeschreven en blijkt dat zowel CHKH als CHKL kloppen zorgt TM voor de terugmelding "id XX", waarbij XX of het programmanummer is van de gezochte datazending, of 00 of FF. Die terugmelding houdt in dat de subroutine RDTAPE wordt verlaten en dat naar het hoofdprogramma van TM wordt teruggesprongen. Klopt of CHKH niet, of CHKL niet, of kloppen beide niet, dan volgt geen

terugkeer uit de subroutine RDTAPE, en blijft dus de terugmelding "id XX" achterwege. Het display schakelt terug van plaatje ③ van figuur 7 naar plaatje ① ("tussenteken") en als men de band blijft laten draaien zal wellicht na verloop van tijd plaatje ② van figuur 7 te zien zijn: de synchronisatie-karakters van een volgende datazending (die in het geval ID=FF in het geheugen zal worden geschreven).

De overschakeling van plaatje ③ naar ①, in plaats van van ③ naar "id XX" betekent dat er hoogstwaarschijnlijk bij het teruglezen van de datazending iets fout is gegaan. Hoogstwaarschijnlijk, omdat de controlemetode, zoals eerder beschreven, niet absoluut waterdicht is.

Mocht dit verschijnsel zich voordoen, dan kan men het datablok opnieuw inlezen (recorder terugspoelen, GET hoeft niet opnieuw te worden ingedrukt). Blijft er sprake van fout inlezen (dus geen terugmelding "id XX"), dan kunnen de oorzaken zijn: a) te laag opname/weergavenivo; b) korrekt opname/weergavenivo, maar te veel drop-outs, waarbij er een aantal tussen zitten die kennelijk te sterk zijn (deze situatie is buitengewoon onwaarschijnlijk, maar je zult het maar net treffen); c) de datazending is opgenomen op een cassette-recorder waarvan de bandsnelheid kennelijk nogal fors afwijkt van de bandsnelheid van de recorder waarmee de datazending is weergegeven. Een "vreemde" cassette-recorder dus. Potmeter P1 (PLL-afregeling) kan best op korrekt datalezen zijn ingesteld, maar staat kennelijk niet in het midden van het instelgebied waarvoor dat geldt (statistisch gunstige marge voor het opvangen van snelheidsverschillen).

Let wel, het gaat hier niet om een verschil in bandsnelheid tussen de INPUT- en de OUTPUT-recorder in de situatie met twee aparte recorders. In dat geval zijn snelheidsverschillen verdiskonteerd in de PLL-afregeling.

Nee, het gaat om snelheidsverschillen bij het schrijven van een datazending op de band — met verschillende cassette-recorders van verschillende eigenaars. We hebben het dus over de situatie die op kan treden als men cassette-band en van iemand anders kopieert, in het kader van een programma-uitwisseling in individueel of gebruikersklubverband.

Er valt iets voor te zeggen om cassette-band en die voor uitwisseling zijn bestemd, of laten we zeggen: die op een andere cassette-recorder moeten kunnen worden afgespeeld, te voorzien van testdata voor het bijregelen van de PLL met P1. Bijvoorbeeld de eerste helft van tabel 2 (vier minuten lang synchronisatie-karakters op de band). Men kan dan na het intoetsen van het leesprogramma (tweede helft van tabel 2) de PLL bijregelen ten behoeve van de opvang van bandsnelheidsverschillen met "vreemde" cassette-recorders. Potmeter P1 zit vlak bij de rechter korte kant van de interfacekaart en is eventueel bereikbaar via een gaatje (ca. 5 mm, eventueel met rubber tule) in de rechter zijkant van de kast.

Dit kan men allemaal doen, maar of het in de praktijk allemaal écht nodig is . . . Als men ervoor zorgt dat de PLL goed is afgeregeld moet men van goeden huize komen om het fout te laten gaan (er zit niet voor niets zo'n fraaie precisietrimmer P1 op de interface-kaart, in plaats van dat zevenmijls-stappen-geval op de KIM). Die testdata op elke cassette-band, waar we het net over hadden, is dan zo'n soort voorzorgsmaatregel die te vergelijken is met Nederlandse of Belgische defensiemaatregelen tegen een mogelijke invasie van de Luxemburgers . . .

Junior op tv!

Méér toetsen en méér te zien

Het dashboard van de "junior-junior-computer" bevat de nodige toetsen. Nodige toetsen — nodig voor een niet al te groot aantal verschillende toetshandelingen. Verder kunnen we via een zesdelig zeven-segmentdisplay zien wat we doen of wat het programma doet. Dit standaard-bedieningspaneel biedt verrassend veel bedieningsmogelijkheden. Maar niet álle wenselijke mogelijkheden. Het oog wil méér en de handen eveneens: een groter toetsenrepertoire. Vandaar de aansluiting, op de uitgebreide junior-computer, van randapparatuur die een dashboard-uitbreiding mogelijk maakt welke is te vergelijken met de overgang van een "deusjevo" naar een DC-9 (bij wijze van spreken dan, want niemand die het met de junior-computer hogerop zoekt hoeft van ons piloot te worden).

Waar gaat het om? De Elekterminal plus ASCII-toetsenbord wordt verbonden met de junior-computer en met een tv-toestel. Het tv-scherm biedt plaats aan meer dan die ene "regel" van zes karakters, die op het standaard-display wordt geschreven. En of het nu om numerieke of om funktietoetsen gaat, of om grafische funkties — het ASCII-toetsenbord biedt een enorm repertoire aan toetsfunkties.

Nu deze randapparatuur eenmaal voorhanden is zou het natuurlijk heel fijn zijn om de nieuwe mogelijkheden te benutten in de vorm van een systeemprogramma. Daar is voor gezorgd: het systeemprogramma Printer Monitor is een uitgebreide versie van de standaard-monitor. Het heeft tien toetskommando's en "uit"

zich op het tv-scherm, in de vorm van een boodschap waarvan de inhoud ondermeer afhangt van de ingedrukte toets(en).

Al met al is het nu mogelijk om een écht junior-journaal op tv te brengen, met een veelheid aan gebeurtenissen en niet uitsluitend op bepaalde tijdstippen s'avonds. En in het geval van dit hoofdstuk is de "nieuwslezer" het programma PM, maar in de toekomst is het niet uitgesloten dat hij in een hogere programmeertaal, bijvoorbeeld BASIC, tot u spreekt.

Van kinderpiano tot concertvleugel en van kijkdoos tot echte tv — zó zou je de in dit hoofdstuk beschreven overgang van standaard- naar ASCII-toetsenbord en van zesdelig zeven-segmentdisplay naar tv-scherm kunnen samenvatten. Daarmee is niets ten nadele gezegd van de standaard situatie. Integendeel, met name in de beginperiode is een overzichtelijk aantal toetsen en kijkmogelijkheden alleen maar voordelig om snel met het verschijnsel computer in het algemeen en met de verschijning junior-computer in het bijzonder vertrouwd te raken. Daar komt nog bij dat, op basis van het bestaande toetsenbord en display, er sprake is van meervoudig gebruik van alle vijf software-funktietoetsen: we hebben de standaard-monitor, de editor en, sinds hoofdstuk 11, TM. Technisch is het geen probleem om dat nog een aantal keren te herhalen. Daar zijn echter nogal wat bezwaren aan verbonden. Hoe krijg je al die verschillende opschriften nog op de toetsen? Maar op de rand zetten, net als bij de gulden? En is een aantal van vijf verschillende kommando-toetsfuncties niet een beetje weinig? Okay, je kunt één toets opofferen als "hoofdlettertoets" (net als bij een typemachine) en de overige vier toetsen elk een dubbele funktie geven. Toegegeven: het is mogelijk om nieuwe kommando's te maken door uit te gaan van combinaties van twee of meer, achter elkaar en in de juiste volgorde ingedrukte toetsen. Maar nog afgezien van andere bezwaren zijn we eerlijk gezegd niet zo kapot van "PC + F B A" in plaats van "G O S U B" (hé, was dat niet een BASIC-kommando? Er wordt daar decimaal gerekend en de toetsen A . . . F zijn dan óók beschikbaar). Dus: meer toetsen en toetsfuncties, maar dan wél graag met zo'n groot ASCII-toetsenbord, dat trouwens kwa opbouw helemaal is ingericht voor kommunikatie (via de videoterminal) met de junior-computer en met het tv-scherm.

Dan het kijk-aspekt. Op het standaard-display is plaats voor één adres met data of voor een 3-byte-instructie, of voor zes datanibbles die afhangen van een (uw) gebruikersprogramma. Er zijn nu drie soorten uitbreidingen denkbaar, en wenselijk. Vat datgene dat op het display is te zien op als de "tekst" van een regel. Op één regel is plaats voor zes "tekstelementen".

1. **Méer tekstelementen.** Met een zeven-segmentdisplay is maar een uiterst beperkt aantal verschillende tekstelementen (kleine en hoofdletters, cijfers, leestekens) op natuurlijke wijze, dus herkenbaar weer te geven. Het zou fijn zijn als men kon beschikken over de mogelijkheden van een normale typemachine, met uitzondering van de keuze: hoofd- of kleine

letters, en van leestekens die bij het afdrucken geen positieverandering van de wagen (carriage) tot gevolg hebben (zoals ' , ' " en ^).

2. **Meer op een regel.** Zes tekstelementen (karakters) op een regel is toch wel al te weinig. Denk aan de situatie bij het afdrucken van een hex dump. Bij gebruik van de Elekterminal is er per regel plaats voor 64 karakters. Dat is in feite dus te vergelijken met een 64-delig zeven-segmentdisplay.
3. **Meer regels.** Telkens wordt op het standaard-display één regel weergegeven. De tekst van de regel past zich aan aan de nieuwe situatie, als gevolg van een actie van de programmeur of van het programma. De oude regel wordt overschreven. De voorgeschiedenis gaat verloren. Dat is vaak erg jammer. Het is ook niet mogelijk om op meerdere regels onder elkaar bepaalde resultaten af te drukken die het gevolg zijn van het indrukken van een bepaalde kommando-toets. Dus heel graag meerdere regels tegelijkertijd zichtbaar! Bij gebruik van de Elekterminal is er plaats voor maximaal 16 regels op het tv-scherm. Maakt men in plaats van een video-display (Elekterminal) gebruik van een geschikte (en dure!) papierprinter (mèt groot toetsenbord), dan is het aantal regels onbeperkt. Dat wil zeggen: zo lang de papiervoorraad strekt.

Alle genoemde wensen op het punt van "meer omhanden hebben" en "meer te zien krijgen" gaan in vervulling indien men de Elekterminal en het bijbehorende toetsenbord via de RS 232-interface aansluit op de junior-computer en via de UHF/VHF-modulator aansluit op een (zwart/wit-)tv-toestel. Lezers van dit boek die tevens minstens sinds de herfst van 1978 het tijdschrift *Elektuur* lezen zal het zijn opgevallen dat Elekterminal, toetsenbord en modulator geen nieuwe, speciaal voor de junior-computer ontworpen apparaten zijn, maar indertijd gepubliceerde projecten ten behoeve van een andere Elektuur-hobby-computer, de SC/MP.

Het is niet de bedoeling om de betreffende tijdschriftartikelen in dit hoofdstuk te herhalen, laat staan er een "junioruitleg" bij te geven: alleen al de bespreking van de werking van de Elekterminal zou een (overigens best interessant) uit de kluiten gewassen hoofdstuk vergen. Wél wordt, verderop in dit hoofdstuk, aandacht gewijd aan specifieke bouw- en gebruiksaspecten in verband met gebruik met de junior-computer. Uiteraard dient u te beschikken over de oorspronkelijke tijdschriftartikelen teneinde de apparaten te kunnen bouwen en het een en ander aan de weet te komen over de werking. Indien u nog niet over die artikelen beschikt kunt u deze verkrijgen via de EKS (kopieën-service). Hoe dat precies moet kunt u lezen in de nieuwste uitgave van het tijdschrift *Elektuur*. Daarin ontdekt u hoogstwaarschijnlijk een hoogst interessant alternatief voor dat relatief dure kopiën-gedoe: het boek *SC/MP-computer voor zelfbouw, deel 2*, dat sinds de zomer van 1981 verkrijgbaar is en waarin alle genoemde artikelen overzichtelijk op een rijtje staan, inclusief andere artikelen over de Elekterminal, andere randapparatuur en meer SC/MP-gerichte onderwerpen, maar ook een leuke BASIC-kursus!

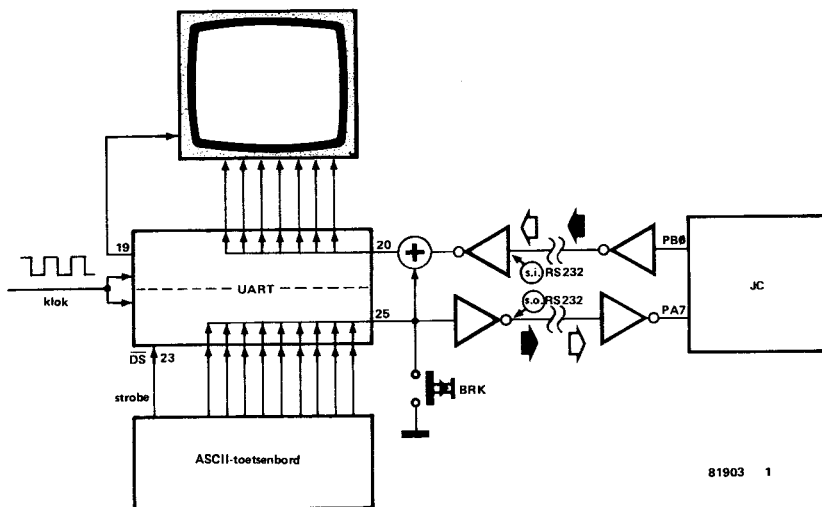
Er is één, bij de genoemde randapparatuur horend onderwerp waar wél wat dieper op zal worden ingegaan: de bitstromen die er lopen van en naar de junior-computer, en tussen de onderdelen van het randapparaat, en de rol die het blok met de science-fiction-achtige naam UART daarbij speelt. Laten we dat eerst maar eens gaan doen.

Junior-computer — randapparatuur vice versa

Seriële, bit-voor-bitse data-overdracht

We hebben vastgesteld dat extra toetsen en het "beter en meer zien wat er gebeurt" (tv-scherm of papier) niet tot de eigenlijke uitgebreide junior-computer horen, maar zijn ondergebracht in één randapparaat. Dat randapparaat ("periferie") wordt aangesloten op de uitgebreide junior-computer. Die aansluiting gaat via een RS 232-interface. Daarover is in hoofdstuk 10 het een en ander verteld. Toetsen hebben iets te maken met onze actie, het tv-scherm met de reactie van de junior-computer daarop. Er is dus sprake van tweerichtingsverkeer. Dat is geheel in overeenstemming met het vraag-en-antwoordspelletje dat er wordt gespeeld. Het is dus niet verwonderlijk dat we in het blokschema van figuur 1 twee lijnen met de naam "RS 232" tegenkomen. Eén lijn voor de van de junior-computer uitgaande berichten en één lijn voor de, de junior-computer binnenkomende berichten. Beide lijnen horen bij een ingang en een uitgang. Het randapparaat, de Elekterminal dus, heeft een ingang s.i. ("serial in") die via via wordt gestuurd uit de als uitgang geschakelde poortlijn PB0 van de junior-computer. De uitgang s.o. ("serial out") stuurt via via de als ingang geschakelde poortlijn PA7.

In het randapparaat zelf zijn er beduidend meer dan twee verbindingen. Het centrale schakelcentrum is de UART. Daar gaan we het direkt nog apart over hebben. Het blok, linksonder in figuur 1 stelt het ASCII-toetsenbord voor. Die is via een 8-bits databus met de UART verbonden. Zodra er een toets is ingedrukt komt de bijbehorende ASCII-kode op die bus te staan. Een negende lijn (strobe) geeft aan of er een toets is ingedrukt of



81903 1

Figuur 1. Blokschema van de data-uitwisseling tussen de junior-computer en het randapparaat: de Elekterminal plus ASCII-toetsenbord.

niet. Deze levert het startsein voor de UART om de van het toetsenbord binnengekomen data verder te verwerken.

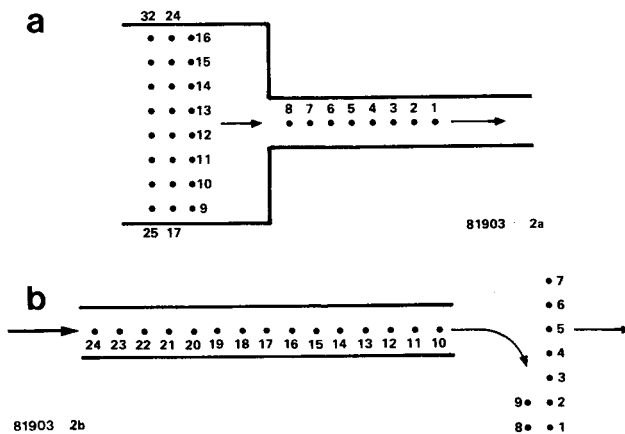
Uit de UART vertrekt een 7-bits databus en verder een stuurlijn naar een blok dat is aangegeven met een tv-scherf. In dit blok is de hele "video-fabriek" van de Elekterminal samengevat. Via de 7-bits databus worden de nieuwste, meest aktuele ASCII-karakters ("updating") naar het tv-scherf verzonden. Die ASCII-karakters zijn de vertaling van een ingedrukte toets of zijn door de junior-computer verzonden. In het geval van een ingedrukte toets van het ASCII-toetsenbord kunnen we dan nog onderscheid maken tussen enerzijds letters, cijfers en leestekens en anderzijds grafische kommando's, zoals CR (terug naar het begin van de regel), LF (verspring naar dezelfde positie op een volgende regel) en nog een dozijn andere, die in het Elekterminal-artikel zijn genoemd. En over "positie" gesproken: deze wordt aangegeven door de zogenaamde "cursor"; zeg maar: de "schrijfwijzer". Deze "aanwijzstok" past zich aan aan de nieuwste, dus aan de meest recente, via de 7-bits databus binnengekomen en in ASCII-kode verpakte afdrukopdrachten of grafische kommando's.

Overigens: we zeggen nou wel dat grafische kommando's uitsluitend op initiatief van het toetsenbord ontstaan, maar dat is niet helemaal waar. Ook de computer kan die opdrachten geven. Voor dit doel heeft het programma PM een paar subroutines in huis.

Dan zien we links in figuur 1 een kloksignaal de UART binnen komen. En waarom dan wel? Om het mogelijk te maken om binnen de UART over te schakelen van een brede databus naar één lijn, of juist omgekeerd, dus van één lijn naar zo'n brede databus. Dus, officieel uitgedrukt: om parallele data om te zetten in seriële data, en omgekeerd.

Het proces is te vergelijken met een optocht van een muziekkapel of drumband, die op een gegeven moment een mansbreed steegje door moet. Hier hebben we temaken met de overgang van parallel (in rijen van meerdere mensen naast elkaar) naar serie: van elke rij van de optocht moet een zich in de looprichting uitstrekkende rij worden gemaakt teneinde de steeg te kunnen passeren. We vergeten nu even de tambour-maître! De gang van zaken is geïllustreerd in figuur 2a. De omzetting van serie naar parallel komt overeen met de situatie na de passage van de steeg door de optocht; zie figuur 2b. De oorspronkelijke situatie wordt hersteld. Nou ja, hersteld... In figuur 2a gaat het om een andere optocht dan in figuur 2b. De ene optocht bestaat uit rijen van acht: de databus van toetsenbord naar UART. De andere optocht bestaat uit rijen van zeven: de databus van UART naar "tv-scherf".

In figuur 1 zien we ook nog een "dwarssteeg": de verbinding tussen de seriële uitgang van de UART (de steeg van figuur 2a) en de seriële ingang van de UART via een soort optelschakeling (de aanduiding "s.i." en "s.o." in figuur 1 betreffen de in/uitgang van de Elekterminal, dus niet van de UART). Met dat plus-bolletje hebben we aangegeven dat de voor het tv-scherf bedoelde seriële data afkomstig kan zijn van of het toetsenbord of de junior-computer. Maar nooit van beiden tegelijk. Door deze dwarsverbinding ziet men een ingedrukte toets direkt op het tv-scherf verschijnen of ziet men de uitwerking ervan op het tv-scherf als het gaat om een grafisch kommando. Deze uitschakelbare verbinding ("half/full duplex") is nodig in al die gevallen waarbij de (junior-)computer geen bevestiging



Figuur 2. Het "optochtmodel" van de omzetting van parallelle naar seriële data (2a) en de omzetting van seriële naar parallelle data (2b).

verzendt (onder supervisie van een systeemprogramma) van een ingedrukte toets, in de vorm van het verzenden van de ASCII-kode van die toets. Een "echo" van de computer blijft dan achterwege. Maar je wilt toch wel graag weten wat je aan het doen bent, nog helemaal afgezien van de reacties van de junior-computer daarop. En levert de software van een systeemprogramma geen echo op dan lukt het wel via de hardware van zo'n dwarsverbinding: een "hardware-echo" dus.

In figuur 1 zien we tevens een toets BRK. Het indrukken ervan heeft een ruwe verstoring tot gevolg van het logische rustnivo van de verbindingstussen randapparaat en junior-computer. Met deze BRK-toets kan "aan de bel worden getrokken": indien de junior-computer bezig is met de verzending van een boodschap (terugmelding, reactie), dan is het met die BRK-toets mogelijk om hem in de rede te vallen. Misschien niet beleefd (in de oudheid werden boodschappers van slecht nieuws een kopje kleiner gemaakt), maar heel vaak wel praktisch. We hebben er dus weer een interrupt-mogelijkheid bij.

Blijven over de vier inverters, nivo-omkeerders, van figuur 1. Ze zijn nodig om de te verzenden of ontvangen seriële data kwa nivo voor te bereiden vóór, respectievelijk terug te vertalen na de verzending over een RS 232-lijn. Iedere verbindingstussen omvat twee inverters en je kunt je afvragen wat het nut ervan is. Immers: twee keer invertieren is toch hetzelfde als helemaal niet invertieren, nietwaar? Inderdaad: niet waar, want de RS 232-norm gaat uit van de negatieve logika van: 1=laag en 0=hoog (zo is de norm nu eenmaal), en zowel de junior-computer als het randapparaat werkt met de vertrouwde positieve logika: 1=hoog en 0=laag. Zonder inverters aan beide kanten van de lijn zou je met geïnverteerde ASCII-kodes moeten werken, dus de code van A zou niet 41 zijn, maar BE. Dat wordt dan allemaal reuze onpraktisch. Nee, die inverters zijn er dus echt niet zo maar voor de lol.

Gaande en komende bits

Laten we eens gaan kijken hoe de bits serieel worden verzonden. De seriële verzending van data betreft eigenlijk een tussenfase van het data-transport. Zowel de oorspronkelijke als de uiteindelijke vorm van de data is parallel: zie de twee databussen in figuur 1 en bedenk verder dat door de junior-computer te verzenden data uit een geheugenplaats komt (acht bits parallel!) of dat door de computer te ontvangen data in een geheugenplaats terecht komt. In de junior-computer vindt de omzetting van serie naar parallel, of omgekeerd, plaats via software. Om precies te zijn: via de carry-vlag. In het randapparaat (Elekterminal) vindt de noodzakelijke omzetting plaats in de UART.

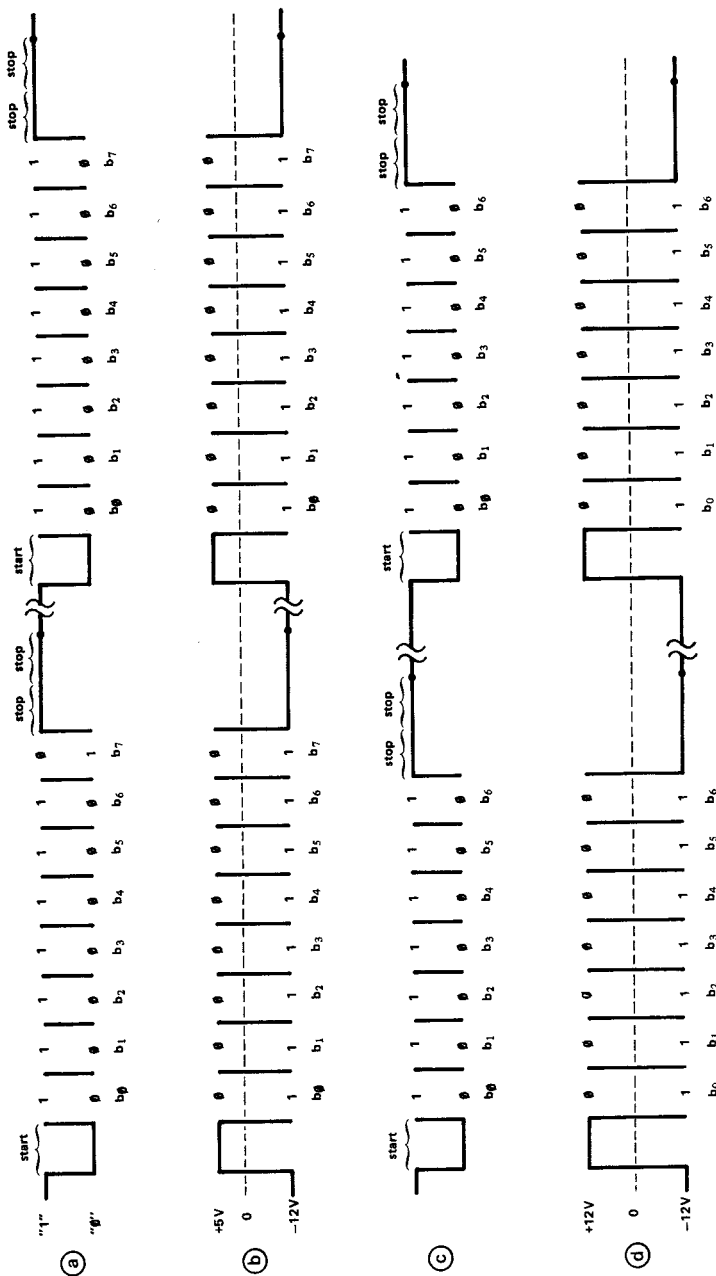
Vergeet we nu even die UART en bekijken we hoe de informatie van een ingedrukte toets in seriële vorm naar de junior-computer gaat (en tevens, via de dwarsverbinding, naar het tv-scherm). Bekijken we dus figuur 3a. Het gaat daarbij om de situatie vóór de aanpassing aan de RS 232-lijn (inversie) aan de Elekterminal-kant en ná de aanpassing van de RS 232-nivo's aan de jc-kant. We gaan uit van een logisch rustnivo één. Er moet een ASCII-karakter worden verzonden. Aan de andere kant van de lijn wacht men hierop en staat klaar voor ontvangst. Hoe komt men nu aan de weet dat er een ASCII-karakter is verzonden? Door het logische nivo gedurende een zekere tijd nul te maken. Die "zekere tijd", dat is de **bittijd** en de nivoverandering heeft te maken met de verzending van het **startbit**.

Het vaststellen van een startbit door de computer (dus de vaststelling van de nivoverandering) betekent dat hij kan zeggen "als ik nou anderhalve bittijd wacht weet ik met zekerheid dat het nivo dat ik dán vaststel hoort bij het bit b0. Wacht ik weer een bittijd lang dan weet ik hoe b1 eruit ziet, enzovoorts". De bits b0 . . . b7 horen bij de door het toetsenbord verzonden parallelle data. Bit b0 is het meest rechtse bit van de 7-bits ASCII-kode en bit b6 het meest linkse bit daarvan. En dat achtste bit b7 dan? Dát is het zogenaamde **pariteitsbit**.

Dit bit wordt verzonden om het verzonden ASCII-karakter te kunnen controleren op goede overkomst na ontvangst. Hoewel we in de junior-computer geen gebruik maken van deze controlemogelijkheid (kijkt u alvast maar stiekum naar figuur 3c) vertellen we er nu iets over, heel in het kort. Je kunt dat bit b7 vóór de verzending een zodanige waarde geven dat het totaal aantal enen van het woord b7 . . . b0 even is (even pariteit), of oneven (oneven pariteit). Bij de ontvangst van het ASCII-karakter kan worden gecontroleerd of dat nog steeds het geval is. Is dat niet het geval, dan kan dat worden gemeld en actie worden ondernomen. Bijvoorbeeld actie in de vorm van het verzoek door de ontvanger aan de zender om dat ASCII-karakter nog eens te herhalen, dus opnieuw te verzenden: "Kunt u dat nog even herhalen? De lijn is slecht". De beschreven controlemethode is overigens verre van waterdicht.

Nadat de acht bits zijn verzonden moet ervoor worden gezorgd dat de verbindingsslijn het rustnivo weer gaat aannemen. Dat gebeurt via de verzending van tenminste één **stopbit**. In ons geval zijn het er twee.

In figuur 3b is het met figuur 3a overeenkomende RS 232-lijnnivo getekend. U ziet dat vanwege de inversie hoog en laag stuivertje hebben verwisseld. We zijn er van uitgegaan dat diode D4 in de Elekterminal (TTL-aanpassing) niet is aangebracht.

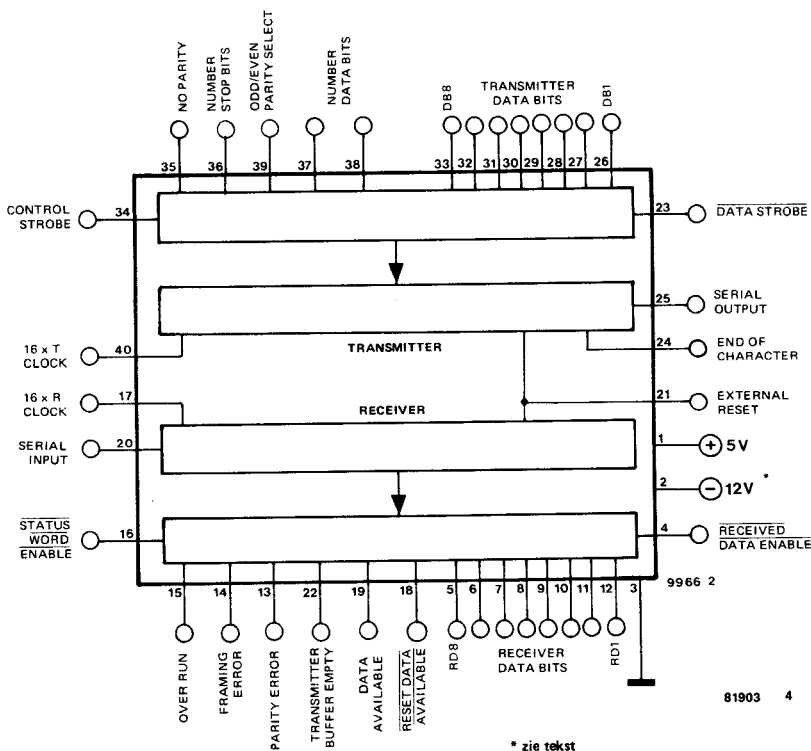


Figuur 3. De logische nive's (a en c) en de spanningsnive's (b en d) tijdens het verzenden van twee ASCII-tekens achter elkaar van het toetsenbord via de UART naar de computer (a en b) en van de computer via de UART naar het tv-scherm (c en d).

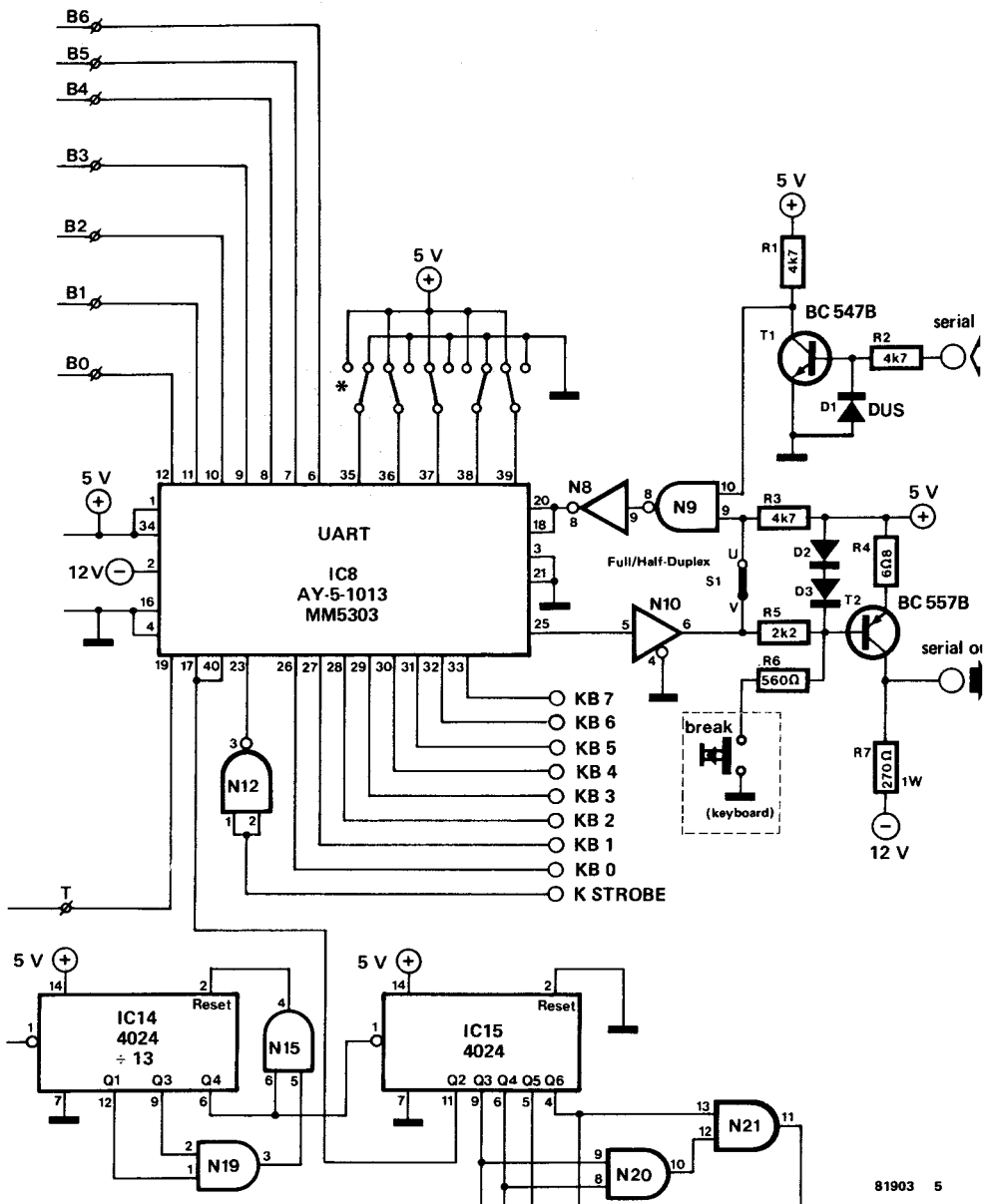
Hoe gaat het in de omgekeerde richting, dus van de junior-computer naar de Elekterminal? Daarover geven ons de figuren 3c en 3d inzicht. We zien dat er geen pariteitsbit wordt meegezonden. Tevens zien we dat het hoge RS 232-nivo op +12 V ligt in plaats van op +5 V.
O ja, voor we het vergeten: het bit b7 wordt weliswaar in overeenstemming met even pariteitscontrole door het toetsenbord via de UART verzonden, maar wordt na ontvangst door de junior-computer nul gemaakt.

De UART

De kreet "UART" is nu al een aantal keren gevallen en het wordt tijd om hem eens apart te nemen. Het gaat om de afkorting van: Universal Asynchronous Receiver/Transmitter. Dat ding met die Engelse naam zorgt voor zowel de omzetting serie-parallel (van junior-computer naar tv-scherm) als de omzetting parallel-serie (van toetsenbord naar de junior-computer en naar het tv-scherm). Het asynchrone schuilt in het feit dat de snelheid van de omzetting is bepaald via de bittijden en niet via een apart mee verzonden kloksignaal.
In het Elekterminal-artikel is al het een en ander verteld over de UART.



Figuur 4. Aanvullende aansluitgegevens van de in de Elekterminal gebruikte UART.



81903 5

Figuur 5. Het gedeelte van het schema van de Elekterminal dat betrekking heeft op de UART IC8 en omgeving is op een aantal punten aangepast aan het gebruik in combinatie met de junior-computer.

Als aanvulling daarop geven we de volledige pinning van het UART-IC in figuur 4. In figuur 5 is het principeschema van de Elekterminal getekend voorzover het gaat om de UART plus directe omgeving. Er is sprake van enige wijzigingen ten opzichte van het oorspronkelijke schema, maar die komen pas in het hoofdstukje met bouwaanwijzingen aan de orde. Figuur 6 is een tekening van het lopende-bandmodel van de UART en van serieel datatransport in het algemeen. Die figuur 6 gaan we nu bespreken.

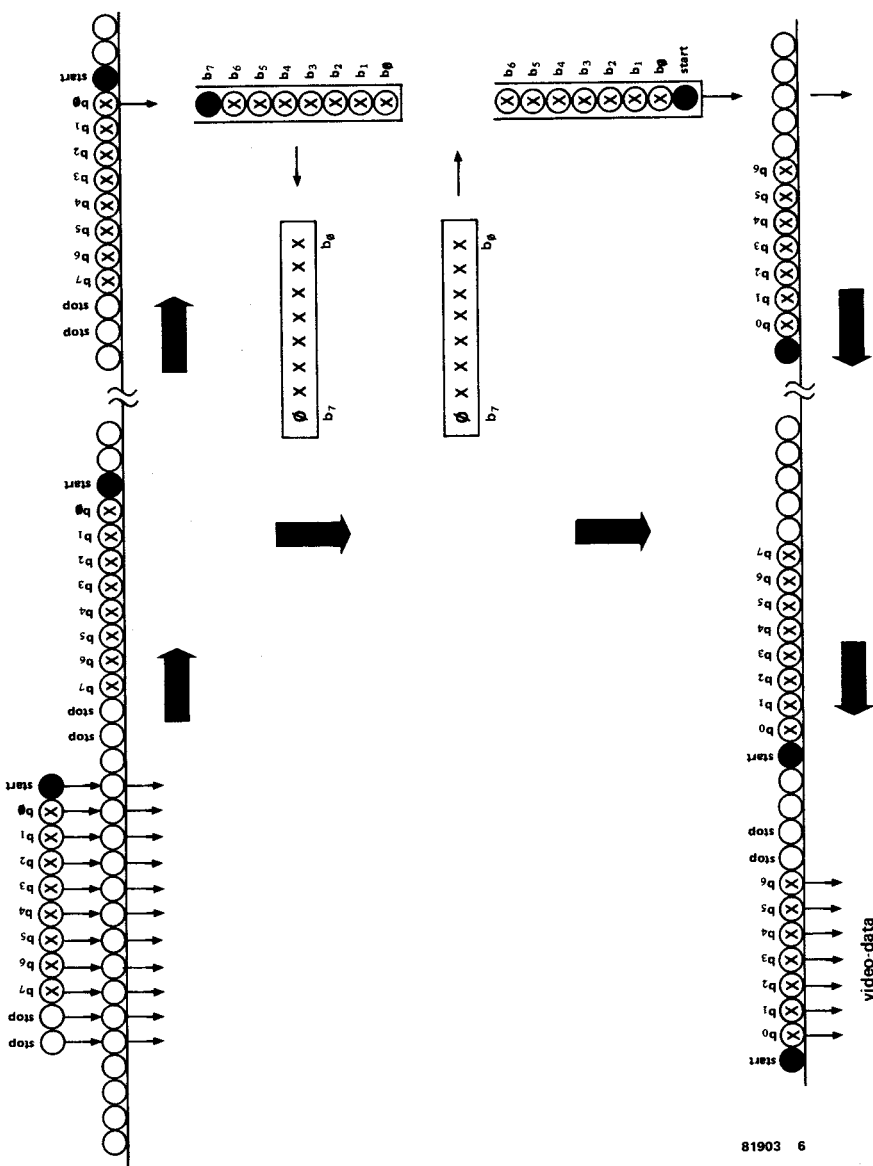
Stelt u zich twee lopende banden voor. De ene band loopt in de richting van de junior-computer naar de Elekterminal, de andere in tegengestelde richting. Op elke lopende band liggen balletjes. Hetzij witte balletjes, hetzij zwarte. Een witte bal stelt een bit één voor, een zwarte bal een bit nul en een bal met een kruisje erin is een zogenaamde "dontcare-bal": hij kan wit zijn of zwart. Als de band een afstand ter grootte van de diameter van een balletje heeft afgelegd is er een bittijd verstreken. De bandsnelheid is dus groter naarmate de bittijd korter is en dus naarmate de Baudrate, dat is de transportsnelheid in bits per seconde, groter is.

In figuur 6 gaat het overigens om de situatie vóór en achter de RS 232-lijnen. De gang door die lijn komt overeen met het witmaken van een zwart bolletje en het na afloop weer zwart maken; een wit balletje daarentegen wordt tijdelijk zwart gemaakt. Dankzij het bestaan van moderne verfspuitbussen is ook in dit opzicht het bolletjesmodel van figuur 6 overeind gebleven, maar zoals gezegd: de RS 232-nivoveranderingen blijven buiten beschouwing.

Beginnen we maar eens met het indrukken van een toets van het ASCII-toetsenbord. Stelt u zich elf pijpjes voor die op een bepaalde plaats (linksboven in figuur 6) boven de band zijn opgesteld. Zodra een toets is ingedrukt en zodra de band zich in een positie bevindt met elf opeenvolgende witte balletjes recht onder de elf met een klep afgesloten pijpuiteinden (er liggen witte balletjes op de band: de verzending van een vorig ASCII-karakter is voltooid), worden de kleppen alle elf tegelijk geopend, de balletjes vallen op de band en nemen elk de plaats in van het balletje dat er lag en dat verdwijnt. De operatie verloopt oneindig snel. Dat mag allemaal in een model, evenals het wegmoffelen van de oorspronkelijke witte ballen.

Het valt op dat het startbit, dus het zwarte bolletje, zich rechts van de daaropvolgende databits $b_0 \dots b_7$ bevindt. Dat is in overeenstemming met de looprichting van de band: richting junior-computer. De databitbolletjes staan in een volgorde die overeenkomt met hun positie in het byte waartoe ze behoren: b_0 helemaal rechts, b_6 (afgezien van het pariteitsbit b_7) helemaal links. Ja, en dat van die twee stopbits is natuurlijk flauwekul. Wat is het hogere nut van het vervangen van twee witte balletjes door weliswaar andere maar nog steeds witte balletjes? U heeft helemaal gelijk en mag van ons in het model gerust twee van de elf pijpjes weglaten.

De bovenste lopende band van figuur 6 loopt naar de junior-computer. Daar zien we een verticale buis die plaats biedt aan acht balletjes. Zodra een zwart bolletje is ontdekt (startbit) vallen de daarop volgende acht balletjes op de een of andere mysterieuze wijze van de band in de buis, in de volgorde van binnenkomst. Het laatste bolletje dat de buis ingaat, b_7 (het pariteitsbit) maken we zwart. Waarom? Omdat we, zoals al eerder opgemerkt, niet zijn geïnteresseerd in de pariteitskontrolle. De volgende



Figuur 6. Het lopende-bandmodel van serieel datatransport, met inbegrip van de omzettingen van parallelle in seriële data of van seriële in parallelle data.

fase, die wordt uitgevoerd via de software van en in de junior-computer, komt erop neer dat de balletjes tot een byte (inhoud van een geheugenplaats) worden omgevormd. De buis gooien we als het ware om in linker richting.

Dan nu de onderste lopende band van figuur 6. Deze loopt van de junior-computer naar de Elekterminal. Het is dus logisch om bij het begin, dus bij de computer te beginnen. De inhoud van een geheugenplaats wordt vertaald in de inhoud van een boven de band en vertikaal opgestelde buis, die plaats biedt aan acht balletjes. Zodra er zich op de band onder de buis witte balletjes bevinden (een vorig ASCII-karakter is volledig verzonden, met inbegrip van twee "witte" stopbits) en telkens wanneer een wit balletje op de band zich precies onder de onderste, met een klep afgesloten opening van de buis bevindt, gaat die klep heel even open en dan weer dicht. Net lang genoeg om één balletje door te laten dat op de band valt en het oorspronkelijke balletje van zijn plaats verdringt. Een bittijd later wordt het volgende balletje losgelaten en dat gaat zo door totdat de buis leeg is.

Ook hier zitten we weer met het probleem van de twee witte balletjes (stopbits), die twee witte balletjes op de band zouden moeten vervangen. In werkelijkheid worden die stopbits verzonden door het herstellen van het rustnivo van de verbinding (als dat al nodig was, afhankelijk van de waarde van het laatst verzonden databit) en wordt de verzending van een volgend ASCII-karakter gedurende twee bittijden geblokkeerd. Bij ontvangst doet zich een soortgelijke situatie voor. Direct na de binnenkomst van het laatste databit (b6 of b7) wordt de ontvangst van een volgend ASCII-karakter gedurende twee opeenvolgende bittijden geblokkeerd.

Terug naar de band en de balletjes daarop van figuur 6. Zodra in de UART in de Elekterminal een startbit wordt vastgesteld (het eerste zwarte balletje na minstens twee witte balletjes) worden de op het zwarte balletje volgende zeven balletjes tegelijkertijd van de band gehaald: de video-data, dat wil zeggen: de meest recente, in ASCII-kode verpakte af te drukken data of grafisch kommando is op de plaats van bestemming terecht gekomen.

Nog één ding over figuur 6. De van het toetsenbord afkomstige seriële data gaat niet alleen naar de junior-computer maar ook "binnendoor" naar het tv-scherm: de al eerder besproken hardware-echo. Het model van figuur 6 is op dit punt niet in detail uitgewerkt. Dat is echter wel mogelijk als men zich bedenkt dat de door het toetsenbord verzonden data het tv-scherm net zo snel bereikt als de junior-computer. En als we ervan uitgaan dat de reactie van de computer ook tijd in beslag neemt zal het niet waarschijnlijk zijn dat de s.i.-ingang **van de UART** uit twee richtingen tegelijk data te verwerken krijgt.

De praktijk

Elekterminal plus ASCII-toetsenbord: de bouw en de aansluiting op de junior-computer

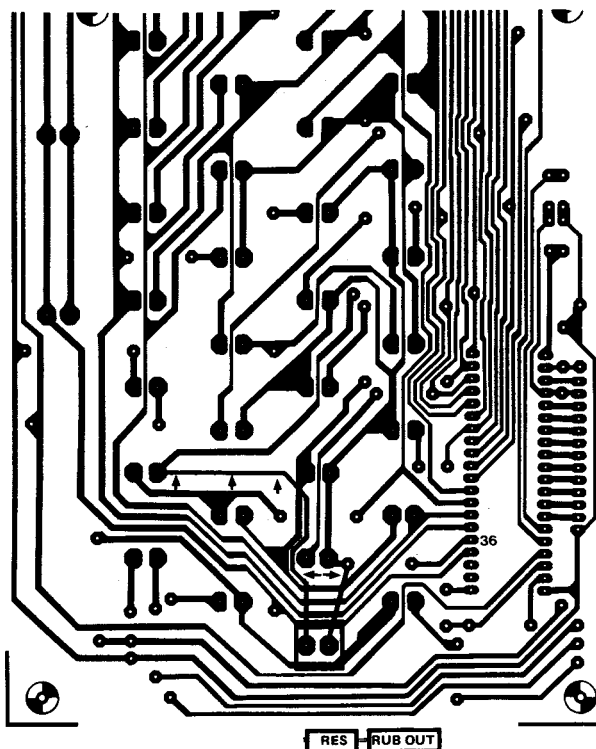
Al eerder in dit hoofdstuk is opgemerkt dat de Elekterminal en het bijbehorende toetsenbord bestaande, destijds in het tijdschrift *Elektuur*

gepubliceerde ontwerpen zijn, die tevens zijn opgenomen in het sinds de zomer van 1981 verkrijgbare Elektuur-boek **SC/MP-computer voor zelfbouw; deel 2**. In dit boek en in de tijdschriftartikelen zijn de volledige schema's te vinden en verder alle printgegevens, een korte uitleg van de diverse schakelingen en een aantal bouwtips. Dus alle informatie die men minimaal nodig heeft is er. Helaas is het niet mogelijk om een schema- en een bouwbeschrijving volgens de volledige junior-metode te geven. Voor de duizenden juniorkonsumenten die via de junior-computer in theorie en praktijk met "computers" hebben kennis gemaakt zou dat zeer nuttig zijn geweest, en de auteurs vinden het best een leuke klus om te doen. Maar ja, het aantal wensen is oneindig groot, de hoeveelheid papier daarentegen niet.

Gelukkig valt de schade wel mee als het gaat om de bouwbeschrijving. Ten eerste heeft u nu de nodige bouwervaringen achter de rug: de standaardversie, de interface-kaart, de voeding. Die benut u nu bij de bouw van de Elekterminal, het ASCII-toetsenbord en (eventueel) de UHF/VHF-modulator. Ten tweede spelen alle verdere bouwactiviteiten zich af op enkelzijdige printen. En ten derde geven we toch een aantal bouwtips, die voor een groot deel noodzakelijk zijn in verband met de toepassing in combinatie met de junior-computer. Al met al moet het dus wel gek lopen, wil u bij de bouw op onoverkomelijke problemen stoten.

	Y0	Y1	Y2	Y3	Y4	Y5	Y6	Y7	Y8	Y9	Y10
X0											
X1											
X2	 = _	 FS							 SP		
X3	 Ø	 *	 P		 @	 ← BS	 {	 }	 CR	 LF ↓	 RUB OUT
X4	 + ;	 ? /	 > .	 < ,	 M	 N	 B	 V	 C	 X	 Z
X5	 L	 K	 J	 H	 G	 F	 D	 S	 A	 Erase FF	 ESC
X6	 O	 I	 U	 Y	 T	 R	 E	 W	 Q	 → HT	 VT ↑
X7	) 9	 (8	 ' 7	 & 6	 % 5	 \$ 4	 # 3	 " 2	 ! 1		

9965 2



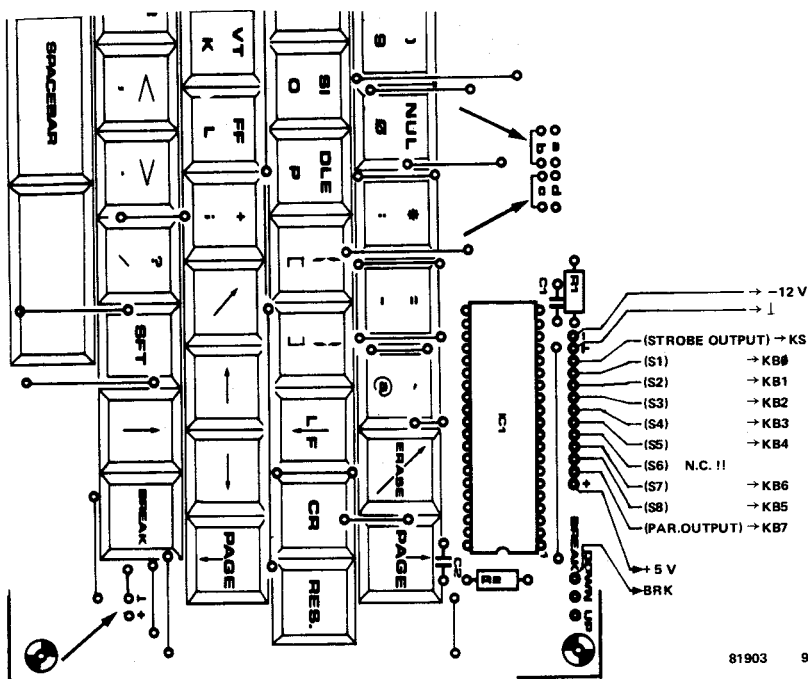
81903 8

Figuur 8. Zó moet men de RES-toets als RUB OUT-toets aansluiten.

Eerst maar het toetsenbord . . .

Bij de bouw van het ASCII-toetsenbord moet u speciaal op het volgende letten:

- De toets RES (reserve) was niet elektrisch aangesloten op de toetsenmatrix $X0 \dots X7/Y0 \dots Y10$, maar moet dat nu, bij gebruik met de junior-computer, wél zijn. In figuur 7 is te zien dat het ene toetskontakt wordt aangesloten op rij X3 en het andere toetskontakt op kolom Y10 van de toetsenmatrix. Toets RES gaat dienst doen als toets DELETE oftewel RUB OUT. Het indrukken van deze toets levert de eigenlijke start op van het systeemprogramma PM. In figuur 8 is een stukje toetsenbordprint te zien met daarop aangegeven de twee verbindingen. Rij X3 is aangesloten op punt 36 van IC1 (AY-5-2376) en kolom Y10 is aangesloten op pen 21 van hetzelfde IC.
- Maak de doorverbindingen b en c, en leg de BRK-toets met één kontakt aan massa. Zie ook figuur 9.
- Op de print van het toetsenbord zijn de aansluitingen (naast IC1) niet met naam en toenaam vermeld omdat daarvoor geen plaats was op de



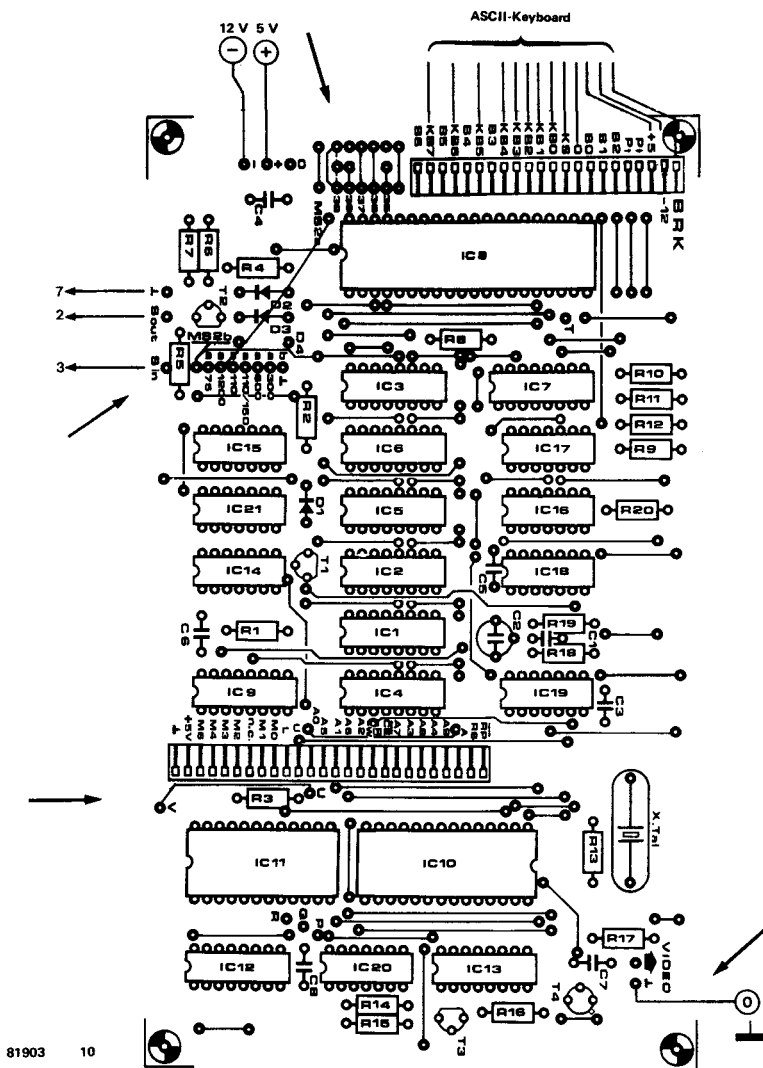
Figuur 9. De draadbruggen en aansluitingen van het ASCII-toetsenbord.

print (witopdruk). In figuur 9 is dat wel gebeurd. Tevens zijn de overeenkomstige aansluitingen op de Elekterminal genoemd. Let erop dat niet S6 maar S8 wordt doorverbonden met het aansluitpunt KB5 van de Elekterminal. Hierdoor wordt bij het indrukken van een lettertoets automatisch de ASCII-kode van de bijbehorende hoofdletter gegenereerd, zonder dat daarvoor het indrukken van een toets SFT (er zijn er twee) nodig is. Trouwens: de karaktergenerator van de Elekterminal heeft toch geen kleine letters in huis. Vergeet de -12 V-verbinding met de Elekterminal niet.

... en dan de Elekterminal ...

In figuur 5 is een stukje Elekterminal-schema te vinden. Het bevat alle informatie die van belang is voor de toepassing van de Elekterminal in combinatie met de junior-computer. In figuur 10 is de (verkleinde!) componentenopdruk van de Elekterminal te zien, met hier en daar enige ingetekende bedradingsaanwijzingen. De speciale junior-zaken, punt voor punt behandeld:

- Allereerst diode D4. Die is niet nodig en dan ook niet meer te zien in figuur 10.
- De baudrate-schakelaar, S2a/S2b, komt te vervallen. Er wordt gewerkt



Figuur 10. Draadbruggen en aansluitingen van de Elekterminalprint. De print is in verband met de afmetingen van dit boek verkleind weergegeven.

met een vaste baudrate van 1200. Het moederkontakt "MS2b" wordt met massa verbonden. Verder wordt er een verbinding gelegd tussen de punten "MS2a" en "1200". Zie figuur 10.

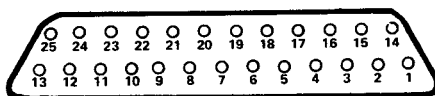
- De full/half-duplex-schakelaar S1 kan worden vervangen door een draadbrug tussen de punten U en V. Indien S1 gehandhaafd blijft moet

deze schakelaar zijn gesloten. Zo ontstaat er een hardware-echo van een ingedrukte toets op het tv-scherm.

- De draadbruggen rond de punten 35 . . . 39 worden gelegd in overeenstemming met de keuzen: wel een pariteitsbit, even pariteit, twee stopbits en zeven bits per karakter (de normale ASCII-kode dus). Weliswaar werkt het systeemprogramma PM niet op basis van een pariteitskontrolle van ontvangen karakters, maar dat neemt niet weg dat het pariteitsbit om andere redenen moet worden meegezonden. In figuur 10 zijn bij het leggen van de draadbruggen 36 en 37 (naar +5 V) de tussenliggende aansluitingen verwijderd, om redenen van overzichtelijkheid.
- In de situatie zonder pagina-uitbreiding van de Elekterminal is er sprake van 13 verbindingen tussen de Elekterminal en het toetsenbord. Deze verbindingen zijn in figuur 10 aangegeven. De aansluiting op de Elekterminal-print vindt plaats via een konnektor. Bekijkt men de printopdruk van de Elekterminal, dan is de konnektorpen -12 V wel aangegeven, maar niet met koperbanen doorverbonden. Gelukkig is dat alleen op papier het geval.
- **UART-keuze (IC8).** Men neme bij voorkeur een type dat uitsluitend +5 V nodig heeft en geen -12 V. Het stroomverbruik voor -12 V is dan minimaal en de Elekterminal kan dan worden gevoed uit de uitgebreide jc-voeding. Het totale stroomverbruik bij uitsluitend +5 V bedraagt ca. 500 mA voor de Elekterminal plus toetsenbord. Er zijn veel UART-typen in de handel met uitsluitend +5 V-voedingsspanning. We noemen de AY-3-1015 (General Instruments), de HM 6402 (Harris), de COM 8017 en COM 8502 (SMC).
- De voedingsaansluitingen massa (nul), +5 V en -12 V zijn in figuur 10 aangegeven. De centrale massa-aansluiting legt men bij voorkeur aan de massa-aansluiting van de video-uitgang.
- De verbinding tussen de video-uitgang en het tv-toestel (monitor-ingang) of de UHF/VHF-modulator moet worden gemaakt met coax kabel met een karakteristieke impedantie van 50 à 75 ohm.
- **Pagina-uitbreiding.** In het SC/MP-boek, deel 2 is aangegeven hoe men het geheugen van de Elekterminal kan vergroten van 1 K naar 4 K. Met als gevolg dat er nu maximaal 64 regels "voorgeschiedenis" kunnen worden onthouden. De besturing vindt plaats via de hardware-toetsen PAGE ↑ en PAGE ↓ van het ASCII-toetsenbord. Dat betekent dat het aantal verbindingen tussen toetsenbord en Elekterminal met twee toeneemt. De aansluiting "DOWN" van het toetsenbord (zie figuur 9) wordt verbonden met het punt "P ↓" van de Elekterminal en de aansluiting "UP" met het punt "P ↑". Zie uiteraard ook figuur 10.
N.B. We lopen nu even vooruit op de bespreking van het programma PM, verderop in dit hoofdstuk. De toetsen PAGE UP en PAGE DOWN zijn hardware-toetsen. Dat betekent dat het indrukken ervan niet de opwekking van een ASCII-kode tot gevolg heeft. Deze toetsen zullen dan ook niet door PM worden herkend als een niet ter zake doende toets. Er zal dus geen foutmelding "WHAT?" of "JUNIOR" volgen.
- **Verbindingen met de junior-computer.** Welke verbindingen zijn er zoal nodig? Allereerst twee RS 232-lijnen, voor het seriële datatransport. Dan minstens één massaverbinding maar liever nog twee stuks: één

"digitale aarde" en een voedingsnul. Verder de verbindingen +5 V en -12 V. We komen dus uit op vijf à zes verbindingen. Neemt men genoeg met één massaverbinding, dan kan men gebruik maken van een vijfpolige DIN-plug plus chassisdeel, vooropgesteld dat men uitgaat van een "losmakelijke" verbinding. Want je kunt natuurlijk óók gewoon direkt de touwtjes leggen en zelfs afzien van de 25-polige D-konnektor op de interface-kaart. Hoe we het in de praktijk ook precies gaan oplossen — de verbindingen moeten korrekt worden gemaakt. Bij de twee seriële dataverbindingen moeten we goed opletten. Eerst even de opmerking dat "serial out" in het schema van de Elekterminal overeenkomt met "Sout" op de print. Idem dito voor "serial in" en "Sin". Laten we figuur 2 van hoofdstuk 10 er even bijnemen. Het punt "Sout" van de Elekterminal-print moet zijn verbonden met D-konnektor-aansluiting 2 op de interface-kaart (bovenkant R5). Het punt "Sin" moet zijn verbonden met D-konnektoraansluiting 3 op de interface-kaart (uitgang IC8). In beide gevallen hetzij direkt gesoldeerd, hetzij via een kabelverbinding met in ieder geval aan de junior-computerkant een male-konnektor, die past in het female chassisdeel, welke laatste of direkt op de interface-kaart is gemonteerd of op de achterkant of zijkant van de jc-kast.

Het mooiste maar ook het duurste is natuurlijk een verbinding tussen de Elekterminal plus toetsenbord enerzijds en de junior-computer anderzijds, die bestaat uit een kabel met aan beide uiteinden een (male) D-konnektor, en met zowel op de junior-computer als op de Elekterminal een 25-polige female-D-konnektor (chassisdeel). In figuur 11 is de plattegrond getekend van een 25-polige male-konnektor. Daaronder is een naar ons beste weten



1. Protective Ground
2. Transmitted Data
3. Received Data
4. Request to Send
5. Clear to Send
6. Data Set Ready
7. Signal Ground
8. Data Carrier Detect
- 9 . . . 14 Not Used
15. Transmitted Bit Clock Internal
16. Not Used
17. Received Bit Clock
- 18,19. Not Used
20. Data Terminal Ready
21. Not Used
22. Ring Indicator
23. Data Signal Rate Selector
24. Transmitted Bit Clock
25. Not Used

81903 11

Figuur 11. De aansluitingen van een 25-polige D-konnektor, met inbegrip van de functie van elke aansluiting.

zo uitgebreid mogelijk overzicht gegeven van de diverse aansluitingen. Dat de EIA-RS 232C-norm van Engels/Amerikaanse oorsprong is zal u, gezien de gebezigde taal, wel duidelijk zijn. Voor de aansluiting op de junior-computer van de Elekterminal spelen uitsluitend de punten 2, 3 en 7 een rol. En bij de punten 2 en 3 moeten we goed opletten. Het gaat om verzonden respectievelijk ontvangen data. Dat wekt verwarring als men bedenkt dat het om twee kanten gaat: verzenden door A betekent ontvangen voor B, en omgekeerd.

Bekijken we de aansluitingen van de D-konnektor op de interface-kaart (figuur 2 van hoofdstuk 10), dan zien we dat pen 2 is gebruikt voor ontvangen (de junior-computer binnenkomende) data en pen 3 voor de junior-computer vertrekkende, uitgaande data. Dat is dus niet in overeenstemming met figuur 11, waar pen 2 is voor "transmitted data", te verzenden data (te verzenden vanuit het randapparaat danwel vanuit de computer), en pen 3 voor "received data" (te ontvangen door het randapparaat danwel de computer). Eerst even een verzuchting: de officiële tekst "Transmitted Data" in figuur 11 wekt ook zo de nodige verwarring op. Bedoeld is: "to be transmitted" en niet: "already been transmitted".

Welnu: gegeven de aansluitingen aan de computerkant (interface-kaart) moet "Sin" zijn aangesloten op pen 3 van hetzij het 25-polige Elekterminal-chassisdeel (volledige, aan twee kanten los te koppelen verbinding), hetzij pen 3 van de 25-polige male-konnektor, die in het bijpassende female-chassisdeel van de computer wordt gestoken. Idem dito "Sout" met pen 2 en de aanpalende massa-aansluiting op de Elekterminal-print (zie figuur 10) met pen 7.

Die verwisseling van de punten 32 en 3 aan de computerkant is toch nog ergens goed voor geweest. Hadden we namelijk wel volledig volgens de norm gewerkt en hebben we te maken met een volledige, aan twee kanten los te koppelen verbinding, dan had punt 2 aan kant A van de kabel moeten worden doorverbonden met punt 3 aan kant B, en evenzo punt 3 aan kant A met punt 2 van kant B. Dat is nu dus niet nodig.

We hebben het ook gehad over **voedingsverbindingen** met de junior-computer. Dat is natuurlijk alleen van toepassing als men de Elekterminal en het toetsenbord wil voeden uit de uitgebreide jc-voeding. Een aparte voeding behoort natuurlijk ook tot de mogelijkheden. Dat verhoogt de universele bruikbaarheid van dit randapparaat. Uiteraard hoeft men dan niet voor voedingsverbindingen te zorgen.

Men kan gebruik maken van twee van de punten 9... 14 van de D-konnektor (zie figuur 11) voor de +5 V- en -12 V-verbindingen. De overeenkomende soldeerpunten op de interface-kaart verbindt men door met het dichtstbijzijnde +5 V- respectievelijk -12 V-punt op de interface-kaart. Een extra ontkoppeling van beide voedingsspanningen naar massa met een tantaalelko van $1\mu/16\text{ V}$ (let op de juiste polariteit!) ter plaatse van de D-konnektor is heel verstandig.

In de bouwbeschrijving van de interface-kaart, in hoofdstuk 10, is opgemerkt dat de punten 4, 5 en 8, alsmede de punten 6 en 20 via draadbruggen met elkaar moeten worden doorverbonden. Deze doorverbindingen zijn niet nodig bij gebruik van de Elekterminal. Toch zijn ze in hoofdstuk 10 bij de bouwbeschrijving genoemd omdat deze verbindingen achteraf, nadat de 25-polige konnektor is gemonteerd, moeilijk zijn aan te

brengen.

Waarom en wanneer zijn deze verbindingen nodig? Die zijn nodig indien men niet de Elekterminal op de junior-computer aansluit, maar een randapparaat (bijvoorbeeld een grote printer met ingebouwd toetsenbord) dat een uitgebreidere vorm van kommunikatie met de computer kent dan de Elekterminal. We bedoelen de kommunikatie op basis van de zogenaamde **handshake**, waarbij er sprake is van een dwingende volgorde waarin de activiteiten zich afspelen, afhankelijk van een stelsel van meldingen en terugmeldingen van zowel de computer als het randapparaat.

Nu is de software noch de hardware van de junior-computer ingesteld op dit "handjes schudden". Gaat echter het randapparaat daar wel van uit, dan moet deze in het geval van een melding aan de computer en dus het wachten van een terugmelding van de computer een "kunstmatig antwoord" krijgen, zodat de werkzaamheden kunnen worden voortgezet. In computer-kretologie: we moeten zorgen voor een "auto-loop back". En dat is de achtergrond van die doorverbindingen tussen de punten 4, 5 en 8, en tussen de punten 6 en 20.

Okay, er is het één en ander uitgelegd over de werking van het één en ander en er is het één en ander gebouwd. De vraag rijst: En wat kun je er allemaal mee doen? Voor het antwoord hebben we de rest van dit hoofdstuk nodig.

PRINTER MONITOR (PM)

Een tv-(systeem)programma

Eigenlijk is de naam Printer Monitor in de meeste gevallen een beetje ongelukkig gekozen. Bij "printer" denk je automatisch aan het afdrukken op papier. Namelijk het afdrukken van een ingedrukte toets na aktie van de gebruiker. En vervolgens het afdrukken van het resultaat daarvan: de reaktie van de junior-computer, in dit geval via de spreekbuis PM. Hoewel er met papier kan worden gewerkt zal er in verreweg de meeste gevallen gebruik worden gemaakt van een video-display (video-interface plus tv-toestel). Bijvoorbeeld de Elekterminal. Dit is geheel in overeenstemming met de gang van zaken bij commerciële "personal computers" (ook wel "ego-computers" genoemd), zoals "de" Apple, "de" PET, enzovoorts. Een betere naam voor PM is wellicht "Video Monitor", maar dat is verwarrend en doet ons te veel denken aan die enge toestanden in banken en warenhuizen.

Het programma PM vormt een alternatief-met-uitbreidingen voor de standaard-monitor. Er is méér tegelijk te zien, aan voorgeschiedenis en/of in de vorm van een reaktie van PM die meerdere regels in beslag neemt. Op de Elekterminal ziet men altijd de laatste zestien regels. Verder zijn er meer toetsfuncties, inclusief twee functies die met het cassette-gebeuren, dus met TM te maken hebben. Het gaat nog steeds om toetsfuncties in het kader van het werken met machinetaal. Eerder in dit hoofdstuk is al opgemerkt dat het, op basis van in principe dezelfde hardware (junior-computer plus randapparatuur) die voor PM nodig is, mogelijk is om "in BASIC" te werken. Mits er natuurlijk een BASIC-vertaalprogramma

("interpreter") als systeemprogramma in het geheugen van de junior-computer aanwezig is.

Het programma PM heeft onderdak gekregen in een EPROM type 2716. Is dus direct na het inschakelen van de junior-computer (en dan uiteraard ná het opstarten) uw bereidwillige dienaar. Het hoeft dus niet eerst van de cassette te worden ingelezen, in RAM. Trouwens: al zou u dat willen, het kan geeneens. Want PM bezet de geheugenplaatsen 1000 . . . 14F3. Kijk het maar na in de hex dump van Aanhangsel 5. Dat is weliswaar minder dan 2 K, maar meer dan 1 K. Er is dus nog plaats voor extra software die men niet wil intoetsen of inlezen. Dat kan op een later tijdstip, via her-programmering van de EPROM. In IC-voet 5, op de interface-kaart, voelt PM zich het beste thuis. Dus niet denken van: het gaat ook wel in IC-voet 4 en ik pas het startadres aan, want dat gaat niet. Van hoofdstuk 10 weten we dat voor IC5 óók 1 K-RAM kan worden genomen. Dat is dus te weinig voor PM. Dus tòch maar "resident", vast aan huis, in plaats van het telkens in huis halen via en van de cassette.

In het kader van de ESS is het mogelijk om uw 2716-EPROM door Elektuur om te laten toveren in een PM-module. Het nummer is ESS 507. Raadpleeg de nieuwste uitgave van het tijdschrift Elektuur voor nadere informatie. U kunt het natuurlijk ook zelf doen of laten doen, aan de hand van de hex dump (zie Aanhangsel 5). Controle van de inhoud van de EPROM kan plaatsvinden met behulp van de standaard-monitor (toetsen AD en +, en byte voor byte bekijken).

De standaard-monitor kent een aantal subroutines die kunnen worden gebruikt in een gebruikersprogramma. Denk daarbij aan subroutines die iets te maken hebben met het weergeven van iets op het display, of met het wachten op en identificeren van een ingedrukte toets. Voor PM geldt iets dergelijks. Er is een subroutine voor het verzenden van een ASCII-karakter naar het tv-scherm of de printer. En er is er één voor het inlezen van een door het toetsenbord verzonden karakter (dus het wachten op een ingedrukte toets). En dan heb je nog allerlei grafische subroutines. Bijvoorbeeld: druk een spatie af. Of: begin op een nieuwe regel. En ga zo maar door. De gebruiksmogelijkheden van al deze subroutines komen eerst in boek 4 volledig aan het licht, in het hoofdstuk over de beschrijving van de PM-software.

Het is mogelijk om, onder supervisie van PM, een gebruikersprogramma zowel ineens als stap voor stap te doorlopen. Dat kennen we al van de standaard-monitor. Maar wordt nu extra interessant omdat het via één enkele druk op een toets mogelijk is om het software-dashboard op te hoesten. Er zijn dus minder uitstapjes nodig om op de hoogte te zijn van allerlei tussenstanden.

We zijn nu zo ver dat we meer in detail kennis kunnen maken met PM en doen dat aan de hand van figuur 12, een plaatje van het ASCII-toetsenbord met daarin aangegeven de toetsen die bij gebruik van PM een rol spelen.

① Toets RES=RUB OUT

De reservetoets RES van het ASCII-toetsenbord is gebruikt als RUB OUT-toets. Druk je hem in, dan wordt het ASCII-teken 7F (zeven énen achter-elkaar) van het toetsenbord via de UART naar de junior-computer

verzonden. Dat is van essentieel belang voor de **tweede start** van PM. Er is dus kennelijk ook een eerste start? Inderdaad, een juiste konklusie. Die **eerste start** vindt plaats via de standaard-monitor:

(inschakelen)

RST 1 0 0 0 GO

De PM-motor is gestart. Het kontaktsleuteltje is a.h.w. omgedraaid. Een deel van het opstart-voorprogramma van PM wordt afgewerkt. Nu is het wachten op het laten rijden van de PM-auto. Dus nu volgt het eigenlijke startsein, door u te geven. De tweede start dus:

RUB OUT(=RES)

Met als gevolg dat het **startkarakter 7F** wordt verzonden naar de computer. Dit maakt het mogelijk om de rest van het PM-voorprogramma af te werken. Er wordt vastgesteld met welke snelheid (Baudrate) die 7 F is verzonden. Alle toekomstige, door de computer te verzenden karakters worden net zo snel verzonden als ontvangen. Als de daarvoor noodzakelijke tijdmetingen zijn verricht, meldt PM zich terug met de boodschap:

JUNIOR

(blanco regel)

Zo'n terugmelding heet een "prompt" (tsja, die computer-kretologie...). Er wordt dus de tekst "JUNIOR" afgedrukt en daarna volgen twee kommando's "begin vóóraan op een nieuwe regel" CR = Carriage Return plus LF = Line Feed). Dat betekent dat er een regel spatie zit tussen "JUNIOR" en de latere tekst.

Na de terugmelding staat PM klaar voor gebruik. Nu is het een kwestie van wachten op een opdracht, dus wachten op een ingedrukte toets, of meerdere toetsen achter elkaar.

N.B. Wordt er geen gebruik gemaakt van de Elekterminal, dan kan in de meeste gevallen een startkarakter 7F worden verkregen via het indrukken van de "verdieping-toets" "control", en vervolgens DEL (delete).

② Toetsen 0 . . . 9 en A . . . F

③ Toets SP (spatie)

④ Toets "." (punt)

Het gaat hier om de meest elementaire toetsen en toetsfuncties. De numerieke toetsen 0 . . . 9 en A . . . F spelen exakt dezelfde rol als bij de standaard-monitor. Dus een rol als onderdeel van een adres, van data op of voor een adres of als onderdeel van een programmanummer (ID). Al met al het bekende hexadecimale repertoire.

Voor het opgeven van een adres of van data zijn speciale ingangsbuffers aanwezig, namelijk de geheugenplaatsen INL en INH (zie tabel 1). Het opgeven van een **werkadres** en het opgeven van data gaat heel anders in zijn werk dan bij de standaard-monitor (toetsen AD, DA, + en numerieke toetsen):

- Eerst wordt de numerieke data opgegeven, daarna pas zegt de gebruiker tegen de computer dat het om de opgave van een werkadres gaat of om de opgave van data op de bij het werkadres horende geheugenplaats.
- De beslissing: het gaat om een werkadres wordt gemaakt door het indrukken van de spatietoets SP.
- Gaat het om het opgeven van data op een werkadres (dat dan wel daarvoor moet zijn opgegeven), dan drukt men op de toets ".".

Tabel 1. De RAM-plaatsen die een rol spelen bij het programma PM.

TEMPORARY DATA BUFFERS		
TEMP	* \$00FC	
TEMPX	* \$00FD	
STPBIT	* \$1A59	NUMBER OF STOP BITS
CNTL	* \$1A5A	BIT TIME BUFFER
CNTH	* \$1A5B	
CNTHL	* \$1A5C	HALF BIT TIME BUFFER
CNTH	* \$1A5D	
TIML	* \$1A5E	COUNT DOWN BUFFER
TIMH	* \$1A5F	
TEMPA	* \$1A60	TEMPS
TEMPB	* \$1A61	
CHA	* \$1A62	CHARACTER BUFFER
PARAL	* \$1A63	PARAMETER BUFFERS
PARAH	* \$1A64	
PARBL	* \$1A65	
PARBH	* \$1A66	
PRTEMP	* \$1A67	TTY BUFFER
BRKT	* \$1A7C	BREAK TEST VECTOR
* * * BUFFERS & EXTERNAL ADDRESSES * * *		
DISCNT	* \$1A68	DISPLAY COUNTER
COLDST	* \$1CB5	EDITOR COLD START
WARMST	* \$1CCA	EDITOR WARM START
BEGIN	* \$1ED3	EDITOR SUBROUTINE
RESET	* \$1C1D	RESET OF VERSION D
GETKEY	* \$1DF9	COMPUTE THE KEY VALUE
LDAINH	* \$1DA7	PART OF SCAND/SCANDS
BEGADL	* \$00E2	BEGIN ADDRESS POINTER
BEGADH	* \$00E3	
ENDADL	* \$00E4	END ADDRESS POINTER
ENDADH	* \$00E5	
CENDL	* \$00E8	CURRENT END ADDRESS POINTER
CENDH	* \$00E9	

- Eén of meerdere linker nullen van een adres of van data hoeven niet te worden opgegeven. Dus voor een werkadres 0200 bestaat het indrukken van: 2 0 0. En drukt men op de spatiebalk SP, dan is het werkadres 0000. Drukt men "." dan wordt het werkadres geladen met de data 00 (bijvoorbeeld de opcode van BRK). Direkt na het verwerken van numerieke gegevens worden de ingangsbuffers INL en INH weer 00 gemaakt, zodat opnieuw het spelletje van de linker nullen links laten liggen kan worden gespeeld.
- Direkt na het opgeven van data (toets ".") wordt het werkadres met één verhoogd en op een nieuwe regel afgedrukt, met de bijbehorende data. Het is dus niet nodig om daarvoor de plustoets in te drukken, zoals het geval is bij de standaard-monitor. Dat bespaart veel "druk-

werk" bij de opgave van een programma: een reeks van elkaar opeenvolgende adressen.

Een voorbeeldje. En wel het optelprogramma van boek 1, pagina 63. Eerst specificeren we in verband met de BRK de IRQ-sprongvektor (staat gericht op een SAVE-achtige ingang van PM, maar dat komt nog). Daarna geven we de programma-data op.

We gaan ervan uit dat PM is gestart en dat de JUNIOR-terugmelding heeft plaatsgevonden.

(1)1A7E(SP)

(2)1A7E XX

Na het indrukken van de numerieke toetsen 1, A, 7 en E en na het indrukken van de spatietoets of -balk SP reageert de junior-computer door op een nieuwe regel het werkadres, een spatie, de data op het werkadres en nog een spatie af te drukken. Oja, even een paar praktische opmerkingen. De toetsakties van de gebruiker zijn met een normaal lettertype weergegeven, inclusief "(SP)" als het om een "onzichtbare" gaat (niet helemaal onzichtbaar, want de stand van de "cursor" van de Elekterminal verradt de "afdruk" van een SP, een CR, enzovoorts). De reactie van JC bij monde van PM is **vet gedrukt**. Het cijfer tussen haakjes aan het begin van een regel is een regelnummer. Dit is hier, in dit boek, om redactionele redenen gebeurd maar wordt in werkelijkheid niet afgedrukt.

Goed, er is een werkadres en we willen daarop de data CF zetten. Dat gaat zo:

(2)1A7E XX CF.

(3)1A7F XX

Het volgende adres verschijnt dus in beeld. Hierop moet de data 14 komen:

(4)1A7F XX 14.

(5)1A80 XX

Op adres 1A80 hebben we verder niets te zoeken. Het programma kan vanaf het startadres 0100 worden ingetoetst. We moeten dus overschakelen op een ander werkadres:

(5)1A80 XX 100 (SP)

(6)0100 XX

en kunnen vervolgens de data intoetsen:

(6)0100 XX 18.

(7)0101 XX A9.

(8)0102 XX 13.

(9)0103 XX 69.

(10)0104 XX 08

(11)0105 XX .

(12)0106 XX

Op adres 0105 is er letterlijk een punt achter gezet: de laatste instructie BRK (opcode 00) van het optelprogrammaatje, dat het geheugen van de junior-computer nu geheel in zich heeft opgenomen.

⑤ Toets "+"

⑥ Toets "-"

Daar kunnen we heel kort over zijn. Toets "+" zorgt voor een nieuw werkadres dat één hoger is dan het oude werkadres. Toets "-" zorgt voor een nieuw werkadres dat één lager is dan het oude werkadres. In beide gevallen wordt na het indrukken het nieuwe werkadres afgedrukt, met data. Het

zijn ideale kontroletoetsen om delen van het geheugen na te lopen. Bijvoorbeeld het stukje geheugen met daarin het optelprogramma:

(12)0106 XX +

(13)0107 XX

Pardon, we moeten de andere kant op:

(13)0107 XX —

(14)0106 XX —

(15)0105 00 —

(16)0104 08 —

(17)0103 69 —

(18)0102 13 —

(19)0101 A9 —

(20)0100 18

Waarmee we zijn beland op het startadres. Alle data is goed opgegeven en het programma kan worden gestart.

N.B. Voor de selectie van de toets + op de Elekterminal moet tevens de "hoofdlettertoets" SHIFT worden ingedrukt.

⑦ Toets R

De R is de afkorting van RUN, "laat maar lopen". Deze toets lijkt als twee druppels water op de GO-toets van de standaard-monitor. Indrukken ervan zorgt ervoor dat een programma, dat begint op het werkadres wordt afgewikkeld. Hetzij ineens, hetzij instructie voor instructie. Dat hangt af van de stand van de STEP-schakelaar.

Laten we het optelprogramma maar eens gaan starten:

(21)0100 18 R

(22)0107 XX

Wat betekent die regel 22? De IRQ-sprongvektor is gericht op 14 CF. Dit adres betekent voor PM hetzelfde als het adres 1C00 voor de standaard-monitor. In het geval van een NMI of IRQ wordt een interrupt-voorprogramma van PM afgewerkt, dat allerlei programmeergegevens bewaart, maar er tevens voor zorgt dat het adres met data wordt afgedrukt dat zonder de interruptie aan bod zou zijn geweest (stand van de programmataeller PC). Bij de BRK treedt de bekende "remweg" op: er wordt één adres overgeslagen. Vandaar 0107 in plaats van 0106 in regel 22.

⑧ Toets P

Deze toets is kwa toetsfunctie gelijk aan de toets PC van de standaard-monitor. Het indrukken van P zorgt er in het geval van het stap voor stap doorlopen van een programma voor dat de draad van het programma weer kan worden opgenomen na allerlei uitstapjes in het kader van het onderzoeken van allerlei tussenstanden in de betreffende fase van het programma (bijvoorbeeld via het indrukken van toets L, zie ⑨). Het terugkeeradres-na-interrupt wordt na het indrukken van P op een nieuwe regel afgedrukt en geldt als nieuw werkadres. Voor de uitvoering van de volgende stap (instructie) is het dan nodig om vervolgens R in te drukken.

⑨ Toets L

De L heeft iets te maken met het engelse list, dus een lijst, opsomming van

de inhoud van alle interne registers van de 6502, die worden bewaard in RAM-geheugenplaatsen tijdens het interrupt-voorprogramma van PM.

Hoe dat precies gaat kan het beste worden uitgelegd aan de hand van een voorbeeld. We gaan verder op regel 22 van het optelprogramma:

(22) **0107 XX L**

(23) **ACC: 1B**

(24) **Y : XX**

(25) **X : XX**

(26) **PC : 0107**

(27) **SP : 01FF**

(28) **PR : 00110100**

(29) **NV BDIZC**

De C van regel 29 wordt gevolgd door een spatie. Nou, dat gaat allemaal wel wat sneller en eleganter dan bij de standaard-monitor. Je ziet alles in één oogopslag in plaats van bij stukjes en beetjes: het achtereenvolgens bekijken van de adressen 00EF ... 00F5. En het P-register in bits uitgespeld met de namen van de vlaggen eronder, dat is helemaal te gek. In één oogopslag zien we het resultaat van de optelling (1B) in de accu staan. Het X- en Y-register spelen geen enkele rol in het optelprogramma en hebben de bekende X-en als inhoud. De inhoud van PC is 0107 en dat klopt, in verband met de BRK. De inhoud van SP (SPUSER) is 01FF, dus net zo als-ie was bij het opstarten van PM (klopt, want we hebben niets op de stapel gezet of ervan afgehaald en zijn ook niet midden in een subroutine gestopt). De B-vlag is 1 omdat we net een BRK achter de rug hebben en de vlaggen N, Z en C zijn alle drie nul omdat het resultaat van de optelling positief is en kleiner dan FF. Een buitengewoon handige toets, die L.

⑩ **Toets M**

⑪ **Toets ", "**

⑫ **Toets CR**

De M van toets M is de afkorting van "Matrix". Een matrix is hier opgevat als een rechthoek die is opgebouwd uit data. Een normale rechthoek kent een lengte en een breedte, een matrix daarentegen kent een aantal rijen en een aantal kolommen. Elke data behoort tot één bepaalde rij en tot één bepaalde kolom. De kreet "matrix" of "datamatrix" komt u wellicht onbekend voor, de gebruikelijke benaming hiervoor, namelijk **hex dump**, niet. Een hex dump is een datamatrix met maximaal zestien data op een rij, dus met zestien kolommen. Het aantal rijen hangt samen met de hoeveelheid data. Deel die hoeveelheid door zestien, tel er één bij op als die deling een rest oplevert en je hebt het aantal regels van de hex dump. De laatste regel hoeft dus niet altijd met data te zijn gevuld, net als de laatste tekstregel van een alinea. Bijvoorbeeld deze alinea.

Samen met de toetsfuncties S en G behoort de toetsfunctie M tot de groep van de zogenaamde **geparametriseerde toetsfuncties**. Dat betekent dat men niet kan volstaan met het indrukken van de toets M of S of G, maar dat er daarna nog meer moet worden opgegeven. Namelijk de **parameters**: door de gebruiker op te geven data die verband houdt met de toetsfunctie. Na het indrukken van toets M of S of G moeten één of meerdere data worden opgegeven. Is er sprake van meer dan één data (IRQ – even een taalkundig probleem, tussen twee haakjes: wat is de enkelvoudsvorm, als-ie er is, van "data"? Toch niet "datum"? Of: als "data" als enkelvoud wordt opgevat,

dan is het meervoud toch niet "data's" of, nog erger, "datassen"! — RTI), dan wordt de data van elkaar gescheiden door een komma. Dus na het opgeven van parameter 1 en vóór het opgeven van parameter 2 moet de toets "," worden ingedrukt. De eigenlijke uitvoering van de toetsfunctie, na het opgeven van alle parameters, vindt plaats via het indrukken van de "bekerhtigingstoets" CR (Carriage Return: ga terug naar het begin van de regel).

Welke parameters spelen een rol bij het maken van een hex dump? Het zijn er twee: het eerste adres en het laatste adres van het datablok dat in de vorm van een hex dump zichtbaar moet worden gemaakt. Hoe het precies in zijn werk gaat kunnen we het beste illustreren aan de hand van het programmavoorbeeld. We waren op regel 29 gebleven:

(29) NVBDIZC M

(30)HEXDUMP:

We geven eerst het eerste adres van het datablok op, dan drukken we ",", vervolgens geven we het laatste adres van het datablok op en drukken we toets CR in. Linker nullen hoeven we ook nu niet op te geven. De opgave van het eerste adres wordt als voltooid beschouwd als "," is ingedrukt. Het eerste adres moet lager zijn dan het laatste adres. Zo niet, dan volgt een foutmelding. Waarover later meer. Okay, daar gaat-ie:

(30)HEXDUMP: 100, 105CR

(31) 0 1 2 3 4 5 6 7 8 9 A B C D E F

(32) 0100 : 18 A9 13 69 08 00

(33) (blanco regel)

(34)JUNIOR

(35) (blanco regel)

en wordt er begonnen op een nieuwe regel: de toekomstige regel 36. We zien dat er eerst een kolomnummer 0 . . . F wordt afgedrukt. De volgende regel begint met het adres dat hoort bij de eerste data (helemaal links) van de rij, die vervolgens op dezelfde regel wordt afgedrukt. De hex dump van het programmavoorbeeld neemt nog geeneens één volledige rij in beslag. Het zal duidelijk zijn dat er in het geval van een groter datablok zich tussen regel 32 en regel 33 meerdere regels à la regel 32 bevinden.

Voor één ding moeten we goed oppassen: het adres op de eerste regel van een rij van een hex dump hoeft niet altijd XXX0 te zijn (met X = 0 . . . F). Dat is een kwestie van kiezen door de gebruiker. Het verdient echter wel aanbeveling om een hex dump te beginnen op een adres XXX0. Je ziet dan in één oogopslag aan het kolomopschrift en aan het rij-adres op welk adres elk data-byte van de hex dump zich thuis voelt. In het programmavoorbeeld van daarnet (regels 31 en 32) is dat inderdaad het geval.

13 Toets S

Deze toetsfunctie komt in feite overeen met de toetsfunctie SAVE van het programma TM van hoofdstuk 11. Het is dus eveneens met PM mogelijk om een datablok in de vorm van een datazending op de cassette-band te schrijven. Er wordt hier eveneens gebruik gemaakt van de subroutine DUMP. Die hoort bij TM. Dus wil men over de toetsfunctie S beschikken, dan moet dus TM in huis zijn.

Het gaat om een geparametriseerde toetsfunctie: na het indrukken van S

moeten achtereenvolgens ID, SA en EA (=LA+1) worden opgegeven. Het tijdstip waarop men vervolgens de toetsfunctie uitvoert, via het indrukken van de toets CR, komt overeen met het tijdstip waarop men bij gebruik van TM de toets GET indrukt. Voor een gedeelte van de noodzakelijke voorbereidende handelingen verwijzen we naar hoofdstuk 11.

Men mag geen programmanummers 00 en FF opgeven. Doet men dit wel, dan volgt een foutmelding.

Een praktijkvoorbeeld. Het optelprogrammaatje:

(36)S11,100, 106CR

(37)READY

Regel 37 wordt afgedrukt zodra het datablok naar de band is weggeschreven. Een volgende actie van de gebruiker uit zich op het begin van regel 38. Het optelprogramma heeft het programmanummer 11 toegevoegd gekregen. Alle data van SA tot EA (tot en met LA) is op de band geschreven.

⑭ Toets G

Zoals toets GET het broertje is van toets SAVE bij TM, zo is toets G familie van toets S bij PM. Het indrukken van G is een nodige voorwaarde voor het via PM inlezen van een datablok van de band. Het zal duidelijk zijn dat ook nu het programma TM aanwezig moet zijn (subroutine RDTAPE), wil toets G niet een loze toets zijn.

Een parameter die in ieder geval moet worden opgegeven is het programmanummer (ID). Kiest men voor een ID FF, dan moet tevens een startadres SA worden opgegeven; want hoe zat het ook al weer (hoofdstuk 11)? Geef je een ID FF op dan wordt het eerste de beste datablok ingelezen en in het geheugen geschreven op een stuk(je) geheugen dat begint met de opgegeven SA.

Ook nu een praktijkvoorbeeld. Het optelprogramma is dermate interessant dat we het op de band hebben gezet (regels 36 en 37) om later nog eens te kunnen gebruiken zonder dat vreselijke intoetswerk van zes bytes. We halen het dus van de band op om er leuke dingen mee te doen:

(38)G11CR

(39)READY

Regel 39 wordt afgedrukt zodra de datazending met ID 11 is gevonden en het bijbehorende datablok in het geheugen van de junior-computer is geschreven, en wel op de adressen 0100 (SA) tot en met 0106. Nu kan de situatie optreden dat we het optelprogramma nodig hebben en een ander, groot programma, dat begint op 0200. Het optelprogramma moet dus worden verplaatst. Het past ruimschoots op pagina nul (vanaf adres 0010) en er komen geen absolute adresoperanden in voor die moeten worden gewijzigd.

Dus wat doen we? Het volgende (band terugdraaien tot vlak voor het begin van datazending 11; band starten in weergavestand direkt na regel 41):

(40)GFF

(41)SA: 10CR

(42)READY

Het optelprogramma is gekopieerd op de adressen 0010 tot en met 0016. Dit is te controleren door dit hexadecimaal te dumpen, via ondermeer de M-toets.

⑮ Toets BRK

Stel u bent bezig met het afdrukken van een hex dump van het geheugenbereik 0000 . . . FFFF en u bedenkt zich plotseling dat het moest gaan om het geheugenbereik 0000 . . . 000F. De eerste afdruk duurt 4096 keer zo lang als de tweede. Het is dus de moeite waard om te kunnen beschikken over een elegante manier om het afdrukproces af te breken om vervolgens te springen naar een gedefinieerd punt (lees: startadres van de een of andere routine) in het programma. Dus: netschakelaar even uit? Nee, een zeer slechte oplossing. Toets RST indrukken, dus terug naar de standaard-monitor? Een iets minder slechte oplossing. Nee, we zoeken iets in de geest van de toets ST van de standaard-monitor. Het indrukken daarvan resulteert in een NMI. Er volgt een indirecte sprong (JMP -IND) naar de geheugenplaatsen 1A7A en 1A7B, waarin een door de gebruiker opgegeven NMI-sprongvektor is gespecificeerd. Meestal gaat het terug naar de standaard-monitor (1C00).

In het geval van de BRK-toets gaan we iets dergelijks doen, zij het niet via het NMI-mechanisme. Bekijken we het gedeeltelijke schema van de Elekterminal (figuur 5), dan zien we dat deze toets met één kant aan massa ligt (lees: moet liggen) en met het andere kontakt aan de gebufferde s.o.-uitgang van de UART ligt. Indrukken van BRK heeft tot gevolg dat het logische rustnivo één op die s.o.-uitgang en op poortlijn PA7 van de junior-computer verandert in een nivo dat logisch nul is. De bijbehorende RS 232-lijn verandert van het rustspanningsnivo van -12 V in het spanningsnivo +5 V.

Welnu, TM heeft een subroutine die zich bezig houdt met het verzenden van een ASCII-teken naar het randapparaat. Die subroutine heet PRCHAR. Tijdens de laatste fase van de uitvoering ervan wordt gekeken of poortlijn PA7 soms laag is. Is dit het geval, dan wordt er gewacht totdat PA7 weer logisch één is. Dus totdat de BRK-toets weer is losgelaten. Daarna volgt een indirecte sprong (JMP-IND) naar de geheugenplaatsen:

(BRKT) 1A7C voor de ADL, en

1A7D voor de ADH

van de **BRK-sprongvektor**. Direkt na het opstarten van PM is de BRK-sprongvektor gericht op het label LABJUN, met adres 105F. Het programma dat op 105F begint omvat de verzorging van de terugmelding JUNIOR en het aan FF gelijkmaken van de stapelwijzer SP en van de inhoud van de geheugenplaats SPUSER (adres 00F2). Het is mogelijk om een andere BRK-sprongvektor te specificeren, maar dat moet dan wel tijdens het verblijf in PM gebeuren. Let wel: met "BRK-sprongvektor" is hier iets anders bedoeld dan de IRQ-sprongvektor in het geval van een **BRK-instructie!**

N.B. Uit het voorgaande is op te maken dat de BRK-toets uitsluitend effectief is tijdens het verzenden, door de junior-computer, van een ASCII-karakter. Zolang PM bijvoorbeeld aan het wachten is op een nieuwe ingedrukte toets, sorteert het indrukken van de BRK-toets geen enkel effect.

Zo, nu zijn alle toetsen van het ASCII-toetsenbord behandeld, die een rol spelen bij PM. Het indrukken van alle andere toetsen levert slechts een meewarig antwoord "WHAT?" op van PM. Of PM antwoordt met

“JUNIOR”, maar dat levert verder óók geen nieuwe gezichtspunten op. We gaan nu nog eens wat doen met de toetsen R, P en L.

Stap voor stap met PM

Momentopnamen afdrukken

Zoals al eerder opgemerkt: óók met PM kun je “single stappen”, dus een gebruikersprogramma bij stukjes en beetjes, instructie voor instructie uitvoeren. Hoe gaat dat?

- Zorg ervoor dat de hardware van de standaardkaart is aangepast. We bedoelen het IC10-opsteekprintje dat in hoofdstuk 10 is besproken; zie de figuren 8b, 9, 11 en 12 van dat hoofdstuk. De twee K-aansluitingen van die opsteekprint moeten zodanig zijn doorverbonden met aansluitpennen van IC6, dat programma's in het adresblok 1000 ... 17FF (K4 en K5; deze zijn doorverbonden ten behoeve van de EPROM met daarin PM) én in het adresblok 1C00 ... 1FFF (K7) **niet** stap voor stap kunnen worden doorlopen. In één geval is een aanvullende NMI-blok-kade van het door K6 geselecteerde adresblok nodig vanwege een kleine monitor-uitbreiding in PIA-RAM, op pagina 1A. Dat betreft zowel de standaard-monitor (K7 en K6, zie Aanhangsel 2) als PM (K4, K6 en K7) in het geval: decimaal rekenen (zie verderop in dit hoofdstuk).
- Zorg ervoor dat de STEP-schakelaar S24 ON staat. Dan licht de rode LED in de GO-toets op.
- Het is niet nodig (met uitzondering van de situatie met decimaal rekenen) om de NMI-sprongvektor op te geven. Deze wordt bij het opstarten van PM automatisch gericht op het adres 14 CF, met het label STEP. Vanaf dit adres speelt zich een soort SAVE-routine af. Alle registers van de 6502 worden bekeken en genoteerd in de bekende RAM-plaatsen op pagina nul. Verder wordt het adres-met-data afgedrukt waarop de opcode staat van de volgende uit te voeren instructie.

Een praktijkvoorbeeld

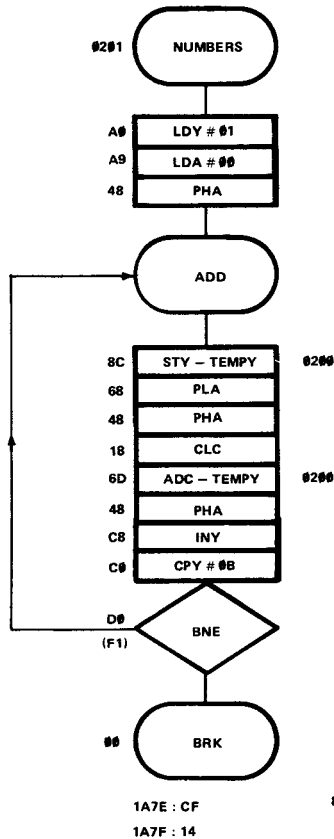
In hoofdstuk 3 van boek 1 is het een en ander verteld over de praktijk van het stapsgewijze doorlopen van een gebruikersprogramma. Ofschoon het hier en nu natuurlijk gaat om de PM-praktijk, is veel van wat er gaat worden behandeld ook van toepassing op de gang van zaken met de standaard-monitor, en kan dit hoofdstukje dus óók worden beschouwd — en toegepast — als een welkome aanvulling op wat daarover in hoofdstuk 3 is gezegd.

Het programmavoorbeeld. Bekijk de volgende reeks getallen eens goed:

0 1 3 6 10 15

+1+2+3+4+5+

We zien dat het verschil tussen het tweede en het eerste getal 1 is; dat het verschil tussen het derde en het tweede getal 2 is; dat het verschil tussen het vierde en het derde getal 3 is; dat dus het verschil tussen twee opeenvolgende getallen één hoger is dan het verschil tussen twee opeenvolgende getallen die elk één positie meer naar links zijn verschoven. We maken een programma waarmee we de eerste elf getallen van die reeks kunnen



Figuur 13. Met het programma NUMBERS is het mogelijk om een bepaalde reeks getallen in de juiste volgorde op de stapel te zetten.

berekenen, inclusief het eerste getal, nul van de reeks, dat het uitgangspunt is. De getallen moeten in de juiste volgorde op de stapel worden gezet, te beginnen met het getal nul op 01FF. Het gaat nu even niet om het hogere nut van het programma, als dat er al mag zijn, maar om de "uitleg-waarde" ervan met betrekking tot de praktijk van het "stappen". Het programma staat in figuur 13. Het heet NUMBERS. Het principe van het programma: eerst wordt de beginwaarde van Y vastgelegd op 01 en gaat het eerste getal, 00, de stapel op. Vervolgens wordt tien keer achter elkaar het programmadeel na het label ADD doorlopen: één keer direct en negen keer via de BNE. De waarde van Y wordt opgeborgen in de geheugenplaats TEMPY, het vorige berekende getal gaat naar de accu (PLA plus PHA) en bij de accu-inhoud wordt de waarde van Y opgeteld (inhoud TEMPY). De instructie PHA is nodig om de stand van de stapelwijzer te herstellen. De berekening van het nieuwe getal van de reeks is voltooid. Dit

FF	01E9	XX	← Y = 0B	D = 1
00	01EA	55	← Y = 0A	
01	01EB	45	← Y = 09	
02	01EC	36	← Y = 08	
03	01ED	28	← Y = 07	
04	01EE	21	← Y = 06	
05	01EF	15	← Y = 05	
06	01F0	10	← Y = 04	
07	01F1	06	← Y = 03	
08	01F2	03	← Y = 02	
09	01F3	01	← Y = 01	D = 0
0A	01F4	00	← Y = 00	
0B	01F5	37	← Y = 0A	
0C	01F6	2D	← Y = 09	
0D	01F7	24	← Y = 08	
0E	01F8	1C	← Y = 07	
0F	01F9	15	← Y = 06	
10	01FA	0F	← Y = 05	
11	01FB	0A	← Y = 04	
12	01FC	06	← Y = 03	
13	01FD	03	← Y = 02	
14	01FE	01	← Y = 01	
15	01FF	00		

81903 14

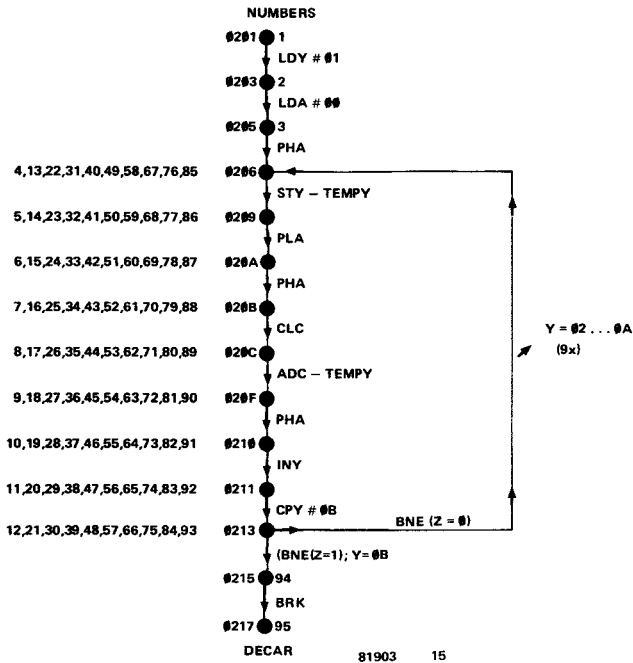
LDA-ABS,Y

Figuur 14. De toestand van de stapel na het binair en het decimaal doorlopen van het programma NUMBERS van figuur 13. De rechter Y-waarden hebben betrekking op het programma NUMBERS en de linker waarden zijn van toepassing voor het programma PRNUMB van figuur 18.

getal gaat naar de stapel (PHA). Dan wordt Y met één verhoogd (INY) voor de berekening van het volgende getal. Een test (CPY #0B plus BNE ADD) leert ons of dat volgende getal nog wel moet worden berekend (springen), of niet (via BRK naar de STEP/SAVE-ingang van PM).

In figuur 14 is de toestand van de stapel en de stapelwijzer geschetst voor de verschillende waarden van Y. De inhoud van de geheugenplaatsen 01FF t/m 01F5 is van toepassing voor binair (hexadecimaal) rekenen. Dat is het geval met het programma van figuur 13. De inhoud van de geheugenplaatsen 01F6 t/m 01EA is van toepassing voor decimaal rekenen. Dáár hebben we het later nog wel over.

In verband met het stap voor stap doorlopen hebben we het programma van figuur 13 in de tot nu toe onbekende gedaante van figuur 15 weergegeven. De zwarte bolletjes stellen adressen met een opcode voor. Dat zijn de "tussenstations", waar de rijdende trein, die de gang door het program-



Figuur 15. De "spoorwegweergave" van het programma NUMBERS (figuur 13). Het aantal verschillende "stations" is één hoger dan het aantal instructies van NUMBERS. Het aantal "stops" is echter aanzienlijk hoger, namelijk 94 stuks. Dat komt omdat er zich op station 0213 een "wissel" bevindt waarvan de stand afhankelijk is van de Z-vlag.

ma voorstelt, een tussenstop maakt: stop voor stop = stap voor stap! U ziet het: het is met recht een boemeltreintje. Bij het ineens doorlopen van het programma stopt de sneltrein maar één keer, namelijk in DECAR, op adres 0217: het laatste stukje van de rit is de BRK-remweg ten gevolge van de BRK op adres 0215.

Het programma NUMBERS heeft u nu leren kennen in de vorm van het gedetailleerde stroomdiagram van figuur 13 en in de vorm van het spoorwegplaatje van figuur 15. Er komt nog een derde vorm bij, die wellicht niet helemaal nieuw voor u is: de weergave in assembler-vorm:

NUMBERS	0201 A0 01	LDY # 01
	0203 A9 00	LDY # 00
	0205 48	PHA
ADD	0206 8C 00 02	STY-TEMPY
	0209 68	PLA
	020A 48	PHA
	020B 18	CLC
	020C 6D 00 02	ADC-TEMPY
	020F 48	PHA
	0210 C8	INY

	0211 C0 0B	CPY #0B
	0213 D0 F1	BNE ADD
BRK	0215 00	BRK
	1A7ECF	
	1A7F 14	

U ziet het: de eerste kolom is bestemd voor de labels. Vervolgens een kolom met de adressen waarop een opcode staat (met uitzondering van de IRQ-sprongvektor, aan het eind). Dan de bytes: één instructie per regel, en dan de mnemonics. De laatste kolom kan eventueel worden gebruikt voor commentaar, toelichtende opmerkingen.

We hebben even voor u uitgerekend dat er bij het stap voor stap doorlopen van het programma NUMBERS sprake is van 93 tussenstops en één eindstop, in DECAR. De beginsituatie van "station" NUMBERS en de eindsituatie van "eindstation" DECAR meegerekend zijn er dus 95 verschillende momentopnamen mogelijk. Dus 95 uitstapjes. De tussenstanden die gedurende die uitstapjes van belang zijn om te weten zijn de inhoud van geheugenplaats TEMPY (0200) en diverse 6502-registers. Het is niet de bedoeling om alle tussenstanden op alle zwarte bolletjes van figuur 15 te gaan noteren in het nu volgende voorbeeld. Dat zou werkelijk meters papier (of — als je daar van kunt spreken — "meters tv-scherm") in beslag nemen. Het staat u vrij om dat wél te doen. We beperken ons tot een aantal momentopnamen. Maar laten we eerst maar eens PM opstarten en het programma NUMBERS gaan intoetsen:

RST 1 0 0 0 GO RUB OUT

JUNIOR

(gevolgd door een blanco regel)

. . . STOP! We kunnen nu wel blijmoedig op gaan schrijven wat u moet doen en hoe de junior-computer daar bij monde van PM op reageert, maar het is natuurlijk veel handiger om gebruik te maken van die mooie papierprinter die Elektuur in huis heeft. Dat hebben we dus maar gedaan, en de weergave daarvan vindt u terug in tabel 2. Eerst is allerlei data opgegeven. Vervolgens is het startadres klaar gezet. De STEP-schakelaar is ON-gezet. De nummers 1 tot en met 95, die voorkomen in de "spoorwegversie" van het programma NUMBERS (figuur 15), vindt men terug in de "listing" van tabel 2. Deze nummers zijn niet via PM tot stand gekomen, maar later aangebracht, ter wille van het overzicht.

Lees verder op pagina 165

Tabel 2. De nu volgende vier-en-een-halve pagina's zijn gewijd aan een PM-listing op papier ("hard copy"), welke verband houdt met het twee maal stap voor stap doorlopen van het programma NUMBERS van figuur 13. Ook de opgave van data is weergegeven. Ter wille van een economisch papierverbruik (in dit boek, wel te verstaan) is de listing weergegeven in de vorm van drie kolommen.

①	②	③
JUNIOR	020A 48 -	0206 8C R 13
	0209 68 -	0209 68 R 14
1A7E	0208 02 -	020A 48 R 15
1A7E 04 CF.	0207 00 -	020B 18 R 16
1A7F 00 14.	0206 8C -	020C 6D R 17
1A80 80 201	0205 48 -	020F 48 R 18
0201 3C A0.	0204 00 -	0210 C8 R 19
0202 3D 1.	0203 A9 -	0211 C0 R 20
0203 3C A9,	0202 01 -	0213 D0 R 21
WHAT?	0201 A0 -	0206 8C R 22
203	0200 3C +	0209 68 L 23
0203 3C A9.	0201 A0 L1	ACC: 03
0204 3C .	ACC: C3	Y : 03
0205 3C 48.	Y : C3	X : C3
0206 3C 8C.	X : C3	PC : 0209
0207 3C .	PC : E6C3	SP : 01FC
0208 3C 2.	SP : 01FF	PR : 10100100
0209 3C 68.	PR : 00000100	NV BDIZC 200
020A 3C 48.	NV BDIZC 200	0200 03 P
020B 3C 18.	0200 3C P	0209 68 R 23
020C 3C 6D.	E6C3 E6 201	020A 48 R 24
020D 3C .	0201 A0 R 1	020B 18 R 25
020E 35 2.	0203 A9 R 2	020C 6D R 26
020F 3C 48.	0205 48 R 3	020F 48 R 27
0210 2C C8.	0206 8C R 4	0210 C8 R 28
0211 3C C0.	0209 68 R 5	0211 C0 R 29
0212 3C B.	020A 48 R 6	0213 D0 L 30
0213 3C D0.	020B 18 R 7	ACC: 06
0214 3C F1.	020C 6D R 8	Y : 04
0215 3C .	020F 48 R 9	X : C3
0216 36 -	0210 C8 R 10	PC : 0213
0215 00 -	0211 C0 R 11	SP : 01FB
0214 F1 -	0213 D0 L 12	PR : 10100100
0213 D0 -	ACC: 01	NV BDIZC 200
0212 0B -	Y : 02	0200 03 P
0211 C0 -	X : C3	0213 D0 R 30
0210 C8 -	PC : 0213	0206 8C R 31
020F 48 -	SP : 01FD	0209 68 R 32
020E 02 -	PR : 10100100	020A 48 R 33
020D 00 -	NV BDIZC 200	020B 18 R 34
020C 6D -	0200 01 P	020C 6D R 35
020B 18 -	0213 D0 R 12	020F 48 R 36

④
0210 C8 R 37
0211 C0 R 38
0213 D0 L 39
ACC: 0A
Y : 05
X : C3
PC : 0213
SP : 01FA
PR : 10100100
NV BDIZC 2000
2000 3C 200
0200 04 P
0213 D0 R 30
0206 8C R 42
0209 68 R 41
020A 48 R 42
020B 18 R 43
020C 6D R 44
020F 48 R 45
0210 C8 R 46
0211 C0 R 47
0213 D0 L 48
ACC: 0F
Y : 06
X : C3
PC : 0213
SP : 01F9
PR : 10100100
NV BDIZC 200
0200 05 P
0213 D0 R 48
0206 8C R 49
0209 68 R 50
020A 48 R 51
020B 18 R 52
020C 6D R 53
020F 48 R 54
0210 C8 R 55
0211 C0 R 56
0213 D0 L 57
ACC: 15
Y : 07

⑤
X : C3
PC : 0213
SP : 01F8
PR : 10100100
NV BDIZC 200
0200 06 P
0213 D0 R 57
0206 8C R 58
0209 68 R 59
020A 48 R 60
020B 18 R 61
020C 6D R 62
020F 48 R 63
0210 C8 R 64
0211 C0 R 65
0213 D0 L 66
ACC: 1C
Y : 08
X : C3
PC : 0213
SP : 01F7
PR : 10100100
NV BDIZC 200
0200 07 P
0213 D0 R 66
0206 8C R 67
0209 68 R 68
020A 48 R 69
020B 18 R 70
020C 6D R 71
020F 48 R 72
0210 C8 R 73
0211 C0 R 74
0213 D0 L 75
ACC: 24
Y : 09
X : C3
PC : 0213
SP : 01F6
PR : 10100100
NV BDIZC 200
0200 08 P

⑥
0213 D0 R 75
0206 8C R 76
0209 68 R 77
020A 48 R 78
020B 18 R 79
020C 6D R 80
020F 48 R 81
0210 C8 R 82
0211 C0 R 83
0213 D0 L 84
ACC: 2D
Y : 0A
X : C3
PC : 0213
SP : 01F5
PR : 10100100
NV BDIZC 20
0200 09 P
0213 D0 R 84
0206 8C R 85
0209 68 R 86
020A 48 R 87
020B 18 R 88
020C 6D R 89
020F 48 R 90
0210 C8 R 91
0211 C0 R 92
0213 D0 L 93
ACC: 37
Y : 0B
X : C3
PC : 0213
SP : 01F4
PR : 00100111
NV BDIZC 20
0200 0A P
0213 D0 R 93
0215 00 R 94
0217 3C R 95
021A 3C R
021D 3C R
0220 47 R

⑦
 0222 C3 L
 ACC: 56
 : 0B
 : C3
 PC : 0222
 SP : 01F4
 PR : 00100101
 NV BDIZC 1FF
 01FF 00 -
 01FE 01 -
 01FD 03 -
 01FC 06 -
 01FB 0A -
 01FA 0F -
 01F9 15 -
 01F8 1C -
 01F7 24 -
 01F6 2D -
 01F5 37 -
 01F4 10 -
 01F3 89 -
 01F2 12 -
 01F1 07 1A7A
 1A7A CF .
 1A7B 14 1A.
 1A7C 5F +
 1A7D 10 +
 1A7E CF .
 1A7F 14 1A.
 1A80 80 1A00
 1A00 00 D8.
 1A01 00 4C.
 1A02 00 CF.
 1A03 00 14.
 1A04 00 217
 0217 3C F8.
 0218 2E 4C.
 0219 3C FF.
 021A 3C 1.
 021B 3C 217
 0217 F8 R 95
 0218 4C L 96

⑧
 ACC: 56
 Y : 0B
 X : C3
 PC : 0218
 SP : 01F4
 PR : 00101101
 NV BDIZC P
 0218 4C R 96
 01FF 00 R 97
 0201 A0 L 1
 ACC: 56
 Y : 0B
 X : C3
 PC : 0201
 SP : 01F4
 PR : 00111101
 NV BDIZC 200
 0200 0A P
 0201 A0 R 1
 0203 A9 R 2
 0205 48 R 3
 0206 8C R 4
 0209 68 R 5
 020A 48 R 6
 020B 18 R 7
 020C 6D R 8
 020F 48 R 9
 0210 C8 R 10
 0211 C0 R 11
 0213 D0 L 12
 ACC: 01
 Y : 02
 X : C3
 PC : 0213
 SP : 01F2
 PR : 10101100
 NV BDIZC 200
 0200 01 P
 0213 D0 R 12
 0206 8C R 13
 0209 68 R 14
 020A 48 R 15

⑨
 020B 18 R 16
 020C 6D R 17
 020F 48 R 18
 0210 C8 R 19
 0211 C0 R 20
 0213 D0 L 21
 ACC: 03
 Y : 03
 X : C3
 PC : 0213
 SP : 01F1
 PR : 10101100
 NV BDIZC 200
 0200 02 P
 0213 D0 R 21
 0206 8C R 22
 0209 68 R 23
 020A 48 R 24
 020B 18 R 25
 020C 6D R 26
 020F 48 R 27
 0210 C8 R 28
 0211 C0 R 29
 0213 D0 L 30
 ACC: 06
 Y : 04
 X : C3
 PC : 0213
 SP : 01F0
 PR : 10101100
 NV BDIZC 200
 0200 03 P
 0213 D0 R 30
 0206 8C R 31
 0209 68 R 32
 020A 48 R 33
 020B 18 R 34
 020C 6D R 35
 020F 48 R 36
 0210 C8 R 37
 0211 C0 R 38
 0213 D0 L 39

ACC: 10
 Y : 05
 X : C3
 PC : 0213
 SP : 01EF
 PR : 10101100
 NV BDIZC 200
 0200 04 P
 0213 D0 R 39
 0206 8C R 46
 0209 68 R 41
 020A 48 R 42
 020B 18 R 43
 020C 6D R 44
 020F 48 R 45
 0210 C8 R 46
 0211 C0 R 47
 0213 D0 L 48
 ACC: 15
 Y : 06
 X : C3
 PC : 0213
 SP : 01EE
 PR : 10101100
 NV BDIZC 200
 0200 05 P
 0213 D0 R 48
 0206 8C R 49
 0209 68 R 50
 020A 48 R 51
 020B 18 R 52
 020C 6D R 53
 020F 48 R 54
 0210 C8 R 55
 0211 C0 R 56
 0213 D0 L 57
 ACC: 21
 Y : 07
 X : C3
 PC : 0213
 SP : 01ED

PR : 10101100
 NV BDIZC 200
 0200 06 P
 0213 D0 R 57
 0206 8C R 58
 0209 68 R 59
 020A 48 R 60
 020B 18 R 61
 020C 6D R 62
 020F 48 R 63
 0210 C8 R 64
 0211 C0 R 65
 0213 D0 L 66
 ACC: 28
 Y : 08
 X : C3
 PC : 0213
 SP : 01EC
 PR : 10101100
 NV BDIZC 200
 0200 07 P
 0213 D0 R 66
 0206 8C R 67
 0209 68 R 68
 020A 48 R 69
 020B 18 R 70
 020C 6D R 71
 020F 48 R 72
 0210 C8 R 73
 0211 C0 R 74
 0213 D0 L 75
 ACC: 36
 Y : 09
 X : C3
 PC : 0213
 SP : 01EB
 PR : 10101100
 NV BDIZC 200
 0200 08 P
 0213 D0 R 75
 0206 8C R 76
 0209 68 R 77

020A 48 R 78
 020B 18 R 79
 020C 6D R 80
 020F 48 R 81
 0210 C8 R 82
 0211 C0 R 83
 0213 D0 R 84
 0206 8C L 85
 ACC: 45
 Y : 0A
 X : C3
 PC : 0206
 SP : 01EA
 PR : 10101100
 NV BDIZC 20
 0200 09 P
 0206 8C R 85
 0209 68 R 86
 020A 48 R 87
 020B 18 R 88
 020C 6D R 89
 020F 48 R 90
 0210 C8 R 91
 0211 C0 R 92
 0213 D0 L 93
 ACC: 55
 Y : 0B
 X : C3
 PC : 0213
 SP : 01E9
 PR : 00101111
 NV BDIZC 20
 0200 0A P
 0213 D0 R 93
 0215 00 L 94
 ACC: 55
 Y : 0B
 X : C3
 PC : 0215
 SP : 01E9
 PR : 00101111
 NV BDIZC 20

⑬		
0200	0A	1FF
01FF	00	-
01FE	01	-
01FD	03	-
01FC	06	-
01FB	0A	-
01FA	0F	-
01F9	15	-
01F8	1C	-
01F7	24	-
01F6	2D	-
01F5	37	-
01F4	00	-
01F3	01	-
01F2	03	-
01F1	06	-
01F0	10	-
01EF	15	-
01EE	21	-
01ED	28	-
01EC	36	-
01EB	45	-
01EA	55	-
01E9	10	

Er zijn tussenstanden genoteerd van de 6502-registers (toets L) en van de geheugenplaats TEMPY (toetsen 2, 0, 0 en SP) op de punten met de nummers 1 (vlak voor de start van NUMBERS 1) en 95 (na de afsluitende BRK; per ongeluk zijn een paar willekeurige instructies uitgevoerd) en verder de nummers 12, 23, 30, 39, 48, 57, 66, 75, 84 en 93: het tussenstation waar zich een "BNE-wissel" bevindt (met uitzondering van 23. We waren wat al te enthousiast bezig met toets R). En tot slot krijgen we een panorama te zien van alle plaatsen van de stapel waarop zich de getallen van de NUMBERS-reeks bevinden. En daar ging het om in dat programma. Overigens: tabel 2 is om redenen van economisch papierverbruik weergegeven in de vorm van drie kolommen per pagina.

Het is niet de bedoeling om aan de hand van de listing van tabel 2 de software-kant van de tussenstanden en de wijzigingen die deze ondergaan te bespreken. Maar niets belet u om dat zelf te gaan doen. Waar het om gaat is dat het systeemprogramma PM van de junior-computer met de toetsen L en P een hoogwaardig stuk gereedschap biedt waarmee het mogelijk is om tussenstanden snel en volledig aan te bieden. De daarop volgende fase van analyse (wat is er nou eigenlijk gebeurd?) kon wel eens meer tijd in beslag nemen, met name voor een gebruikersprogramma dat kwa ontwikkeling nog in de kinderschoenen (Junior?) staat.

En nu nog een keer het programma NUMBERS, maar dan via decimaal rekenen

We doen het nu nog eens dunnetjes over, maar dan nu via een decimale berekening van de NUMBERS-getallenreeks. Let wel: het gaat om precies dezelfde getallen, maar nu niet in de hexadecimale kode, dus in het zestientallig stelsel weergegeven, maar in de decimale kode, dus met ons alledaagse tientallige stelsel. In het eerste geval zijn de getallen weergegeven met de cijfers 0 . . . 9 en A . . . F, in het tweede geval met de cijfers 0 . . . 9.

In Aanhangsel 2 kunt u lezen dat er problemen ontstaan als men via de SAVE-ingang naar de standaard-monitor wil springen als er sprake is van decimaal rekenen, dus als van de beide instructies CLD en SED de laatste het laatst aan bod is geweest. Men kan eveneens allerlei problemen verwachten als men via de STEP-ingang (adres 14 CF) naar PM springt, terwijl SED geldig is. Op een aantal plaatsen binnen PM komt de instructie ADC of SBC voor, en er gebeuren gekke dingen als die instructies decimaal worden uitgevoerd. Bovendien bestaat data met één of twee nibbles A . . . F domweg niet bij decimaal rekenen. In boek 4 komt de PM-hardware byte voor byte aan de orde en dan komt u ze vanzelf wel tegen.

Net als dat voor de standaard-monitor in Aanhangsel 2 is gedaan kan men ook hier het programma PM, of juist: de STEP-routine van PM, uitbreiden met een stukje PIA-RAM, waarin de binaire situatie wordt hersteld en vervolgens naar STEP wordt gesprongen: zie figuur 16b. Er moet dan wel voor worden gezorgd dat de NMI-sprongvektor op BINAR (adres 1A00) wordt gericht als men wil "stappen" door het gebruikersprogramma, of indien men via het indrukken van toets ST terug wil naar PM. Tevens moet de IRQ-sprongvektor op BINAR worden gericht in het geval dat er één of meerdere BRKs voorkomen in een gebruikersprogramma. De omweg over BINAR naar STEP is dus absoluut nodig als de D-vlag logisch één is (decimaal rekenen) vlak voor de sprong naar PM (lees: de sprong naar BINAR) en is eigenlijk overbodig, maar kan geen kwaad als er sprake is van binair, dus hexadecimaal rekenen.

Maar er is nog iets! In hoofdstuk 10 is het een en ander verteld over aanvullende hardware op de standaardkaart, die nodig is om het instructie voor instructie doorlopen van een monitorprogramma (standaard of PM) te blokkeren. Omdat de opdruk "K4,7" op de opsteekprint EPS 81033-3 verwarrend is vertellen we hier nog even voor alle zekerheid hoe de vork in de steel zit:

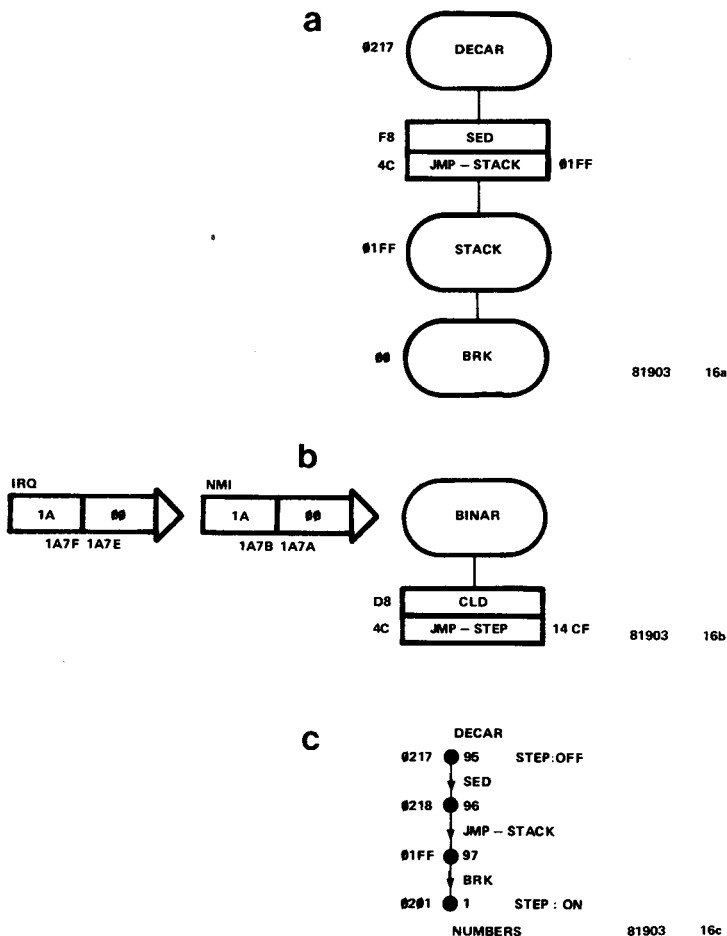
1. **K7 moet altijd zijn aangesloten**, óók als de vektoren RES, NMI en IRQ zelf in buskaartgeheugen zitten! Dit omdat in de standaard-EPROM twee JMP-IND-instructies zitten, die moeten worden uitgevoerd om een NMI of IRQ daadwerkelijk af te handelen (en dat **moet** bij stap voor stap; de effectieve IRQ- en NMI-sprongvektoren zijn in PIA-RAM op pagina 1A gespecificeerd).
2. De tweede aansluiting is K6 in het geval dat men zonder PM, dus met de standaard-monitor decimaal wil kunnen werken. Zie verder Aanhangsel 2.
3. De tweede aansluiting is K4 als men stap voor stap wil kunnen werken met PM, zoals in het NUMBERS-geval van tabel 2 (eerste helft).

Welnu: we willen NUMBERS nu óók **decimaal en stap voor stap** kunnen doorlopen. Gezien de gekozen methode (zie BINAR; figuur 16b) **is er een derde aansluiting, K6, nodig op de opsteekprint**. Hoe dat moet is in figuur 17 te zien.

Toegegeven: de oplossing met de germaniumdiode (DUG) voor K6 verdient in technisch opzicht niet de schoonheidsprijs, maar het werkt en daar gaat het om. Deze voorziening is dus nodig om het decimaal doorlopen van NUMBERS (tabel 2, tweede helft). Het is nu overigens niet de bedoeling dat u nu de junior-computer uitschakelt om K6 aan te gaan sluiten. We gaan er van uit dat dit als is gebeurd vóór het binair doorlopen van NUMBERS. De overschakeling van binair naar decimaal "stappen" door NUMBERS verloopt namelijk bij niet-uitgeschakelde junior-computer! Hoe gaan we nu tewerk in het geval van een decimale uitvoering van het programma NUMBERS?

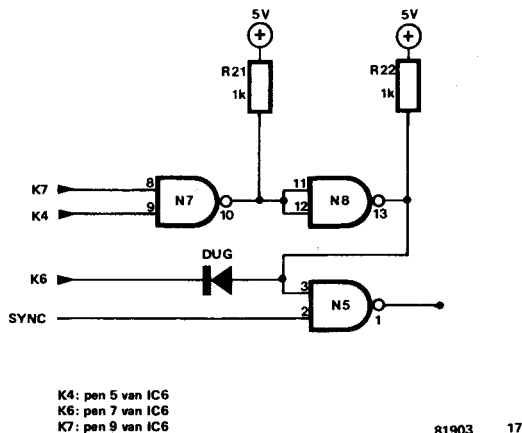
Als volgt:

- Na de uitvoering van de binaire versie van NUMBERS is de programmataeller vanwege de BRK-remweg op adres 0217, met label DECAR beland. Dat er binair is gehandeld blijkt uit de feiten (zie de inhoud van de stapel) en komt voort uit het feit dat tijdens het opstarten van PM de instructie CLD wordt uitgevoerd, net zoals tijdens het RESET-voorprogramma van de standaard-monitor. Dus: CLD, tenzij anders is opgegeven, door de gebruiker.
- Op adres 0217 begint het programma van figuur 16a. Dit programma kan ineens en hoeft dus niet stap voor stap te worden doorlopen. Er wordt overgeschakeld op decimaal rekenen (SED) en gesprongen naar de laagste plaats van de stapel, 01FF, dat voor de gelegenheid het label STACK heeft meegekregen.
- Op het adres 01FF bevindt zich de data 00, namelijk het eerste getal van de NUMBERS-getallenreeks. **Deze 00 doet nu dienst als de opcode van de instructie BRK!** Na de uitvoering van deze BRK staat de programmateller PC gericht op het adres 0201. Dat was het startadres van NUMBERS voor binair rekenen en is dat nu ook voor decimaal rekenen! "Recycling" heet dat!
- De sprong van DECAR naar NUMBERS via de BRK op 01FF (zie ook figuur 16c) maakt het mogelijk om het ongewijzigde programma NUMBERS met een ongewijzigd startadres nu decimaal in plaats van binair uit te voeren. Hoe gaat het allemaal in de praktijk?
- We gaan uit van de situatie aan het eind van het binair doorlopen van NUMBERS. De stand van de programmateller is 0217 (te controleren door het indrukken van toets P). Er is een overzicht gegeven van de inhoud van de 6502-registers, van TEMPY en van de stapelplaatsen met getallen van de reeks.
- Toets de data in volgens de figuren 16a en 16b. Vergeet niet om de NMI- en IRQ-sprongvektoren te wijzigen.
- Zet de STEP-schakelaar OFF (dat hoeft niet persé, maar dan stomen we tegelijk door naar het startadres van NUMBERS). In tabel 2 is stap voor stap tewerk gegaan: de tussenstations 96 en 97 van figuur 16c.
- Druk toets P, voor het klaarzetten van het startadres DECAR.
- Druk toets R. Het programma loopt en men meldt zich via PM terug met het startadres van NUMBERS.



Figuur 16. Het programma DECAR (16a) maakt het mogelijk om na het binair doorlopen van NUMBERS datzelfde programma nogmaals te doorlopen, maar dan decimaal. Daarvoor is het nodig dat PM wordt uitgebreid met een stukje PIA-RAM (16b). De IRQ- en NMI-sprongvektor dienen te worden aangepast. Een en ander houdt tevens een uitbreiding in van het "spoorwegnet" van figuur 15 (16c).

- Er wordt een momentopname gemaakt vóór de start van NUMBERS. Dat gebeurt via het indrukken van toets L, en van de toetsen 200 (SP).
- Zorg dat de STEP-schakelaar ON staat en druk achtereenvolgens de toetsen P (klaarzetten startadres) en R (uitvoering eerste instructie). En nu herhaalt zich het spelletje. Het gaat verder allemaal precies zo als in het binaire geval. Het verdere verloop kunt u nagaan in de tweede helft van tabel 2.



Figuur 17. Indien men decimaal wil rekenen onder supervisie van het programma PM, dan dient óók de aansluiting K6 van de adresdekoder (standaardkaart) op de opsteek-print plaats te vinden en wel met behulp van een germaniumdiode (DUG)

Het zij nogmaals opgemerkt: Bij het besproken programmavoorbeeld NUMBERS, in de binaire en in de decimale versie, is zo te werk gegaan om het een en ander uit te leggen en te illustreren bij het stap-voor-stap-gebeuren. Het was bijvoorbeeld niet nodig om over te schakelen op de decimale versie via het programma DECAR (figuur 16a). We hadden óók al meteen bij het binair doorlopen van NUMBERS elke sprong naar PM via de BINAR-omweg van figuur 16b kunnen laten lopen. Met andere woorden: de enige data die na afloop van het binair doorlopen van NUMBERS moet worden ingebracht betreft het omschakelen op decimaal rekenen. En dat hoeft niet persé te gebeuren via een gebruikersprogramma met ondermeer de instructie SED, maar kan ook direkt via PM tot stand komen: Laad de geheugenplaats PREG (00F1) met zodanige data dat de D-vlag 1 wordt en alle andere vlaggen ongewijzigd blijven.

We hebben aangetoond dat het stap-voor-stap doorlopen van een programma veel meer kan inhouden dan het bekijken van interne registers. Nee, je kunt door het opgeven van data de afwikkeling van het programma beïnvloeden. Verder is aangetoond dat het met PM oneindig veel handiger "stappen" is dan met de standaard-monitor.

PM: Pro Memorie

PM-zaken om te onthouden

Andere NMI-sprongvektor? Dat kan. Na het opstarten van PM wordt deze sprongvektor automatisch, tijdens het voorprogramma gespecificeerd. Geheugenplaats 1A7A krijgt de data CF en 1A7B de data 14; 14 CF is het startadres van het STEP-voorprogramma, waarin de stand van de 6502-registers wordt genoteerd en de inhoud van de programmateller als het

nieuwste werkadres wordt afgedrukt.

De gebruiker kan echter andere ideeën hebben over het gebruik van een NMI. Bijvoorbeeld een speciale toepassing van de toets ST van het standaard-bedieningspaneel van de junior-computer. Denk daarbij overigens wel aan het "vol zetten" van de stapel: iedere keer ST indrukken levert drie gevulde stapelplaatsen op. Het is mogelijk om de NMI-sprongvektor te richten op een door de gebruiker te kiezen startadres. Dat moet dan wel tijdens het verblijf in PM gebeuren, want anders (dus als men het ervoor doet) wordt de data op de adressen 1A7A en 1A7B door het PM-voorprogramma gedikteerd.

De IRQ-sprongvektor (1A7E en 1A7F) kan zowel vóór als na de sprong naar PM worden opgegeven.

Een waarschuwing. Indien men gebruik maakt van de toetsfunctie S of G, dus indien men gebruik maakt van een subroutine van het programma TM, en de STEP-schakelaar ON staat, zal er nooit een datazending op de band worden geschreven of van de band worden gelezen. Men kan dan tot sintjuttemis wachten op de terugmelding "READY". Hoe komt dat? Heel eenvoudig omdat het programma TM in een EPROM zit die door de selectiesignalen K2 en K3 wordt geselecteerd (zie hoofdstuk 10). En voor die K's is het stap voor stap doorlopen van een programma op het overeenkomstige adresblok niet geblokkeerd! Dus in dit geval OFF, die STEP-schakelaar!

Foutmeldingen

Slechts een klein aantal toetsen (zie figuur 12) van het ASCII-toetsenbord betekent iets voor PM: hij herkent ze aan de bijbehorende ASCII-kodes. Let wel: vrijwel alle toetsen kunnen wel degelijk een betekenis hebben in programma's waarin gebruik wordt gemaakt van PM-subroutines. Dat zal verderop in dit hoofdstuk blijken.

Verder heeft PM geen boodschap aan op zich geldige toetsfuncties in het geval dat er wordt gewerkt aan één van de geparametriseerde toetskommando's M, S en G. Neem nou de ingedrukte toetsen:

S11,200,R

Er zal geen datazending op de band worden geschreven, en PM zal ook niet reageren op toets R, maar hij reageert met:

JUNIOR

en produceert vervolgens een blanco regel. Er zijn ook nog andere situaties, die aanleiding geven tot een terugmelding.

a. Terugmelding "WHAT?"

De melding "WHAT?" plus de daarop volgende "afgedrukte" blanco regel (in het geval dat "JUNIOR" volgt) treedt op als PM wacht op een nieuwe ingedrukte toets die niet numeriek is en die ook geen relevante of helemaal geen toetsfunctie vertegenwoordigt. In het geval van het indrukken van een toets met een geparametriseerde toetsfunctie (dus M of S of G) zal de boodschap "WHAT?" altijd worden gevolgd door "JUNIOR" plus een tweede blanco regel.

Een voorbeeld:

**0407 XX 3F,
WHAT?**

We wilden op adres 0407 de data 3F opgeven en in plaats van een punt drukten we een komma in. De kommatoets heeft alleen een betekenis bij de toetsfuncties M en S.

b. Terugmelding "JUNIOR"

De terugmelding "JUNIOR" zijn we al een paar keer tegengekomen. Namelijk in het geval dat er sprake is van:

1. de volledige afronding van het opstarten van PM. Dus de eerste start plus de tweede start;
2. het indrukken van de BRK-toets tijdens het verzenden van één of meer ASCII-karakters door de junior-computer, vooropgesteld dat de BRK-sprongvektor als volgt is gespecificeerd: 1A7C : data 5F; 1A7D : data 10. Dit is automatisch het geval na het opstarten van PM, maar kan worden gewijzigd door de gebruiker;
3. de afronding van een hex dump, na het indrukken van toets M en de opgave van de parameters.

In alle drie gevallen gaat het om normale terugmeldingen, zij het dat de BRK-toets vaak een noodrem-functie heeft. Nu de gevallen waarbij de terugmelding "JUNIOR" een foutmelding betekent:

4. Na het indrukken van toets M zijn de parameters technisch korrekt opgegeven. Dus er zijn geen verkeerde toetsen ingedrukt. Er is echter een eindadres opgegeven dat lager is dan het beginadres van het te dumpen datablok. Er volgt een foutmelding.
5. Toets S is ingedrukt. Daarna wordt een programmanummer (ID) 00 of FF opgegeven. U weet (en PM weet dat dus ook) dat een op de band te schrijven datazending deze programmanummers niet mag bezitten.
6. Na het indrukken van een toets die hoort bij een geparametriseerde toetsfunctie is er een fout gemaakt bij het afronden van de toets-opdracht. Uitsluitend de numerieke toetsen, de toets "," en de toets CR spelen op een bepaald moment een rol (in het geval van G de komma niet). Het indrukken van niet relevante toetsen leidt tot een foutmelding, maar ook het indrukken van bijvoorbeeld CR voordat een tweede parameter is opgegeven (dus voordat ",", is ingedrukt) geeft een foutmelding. Samengevat: iedere afwijking van:

M → **HEXDUMP**: (numerieke data), (numerieke data)CR

S (numerieke data), (numerieke data), (numerieke data)CR

G (numerieke data ≠ FF)CR

GFF → **SA**: (numerieke data)CR

geeft aanleiding tot een foutmelding. Dus óók bijvoorbeeld wanneer men via het indrukken van de spatietoets denkt een in grafisch opzicht betere afdruk te krijgen.

N.B. 1. In een aantal gevallen wordt een foutmelding tijdens de voorbereiding van de uitvoering van een geparametriseerde toetsfunctie voorafgegaan door de foutmelding "WHAT?". Bij de bespreking van de PM-software in boek 4 zal hierop nader worden ingegaan.

N.B. 2. Wat te doen na een foutmelding? In het geval van een geparametriseerde toetsfunctie (M, S of G) gewoon opnieuw beginnen, dus wéér opnieuw M, S of G indrukken, en dán natuurlijk niet opnieuw een fout maken! In alle andere gevallen komt het er meestal op neer dat het werk-

adres waar men mee bezig was opnieuw wordt opgegeven, via het indrukken van numerieke toetsen en van de spatietoets.

Nog wat over S en G. Heeft men S of G ingedrukt, de parameters opgegeven en toets CR ingedrukt, dan wordt uiteraard óók het grafische kommando CR (terug naar het begin van de regel) uitgevoerd. De "cursor" ("schrijfwijzer") van de Elekterminal staat gericht op het begin van de regel waarop als laatste CR voorkwam. Het eerste karakter van deze regel is afwisselend wel en niet afzichtbaar, zolang er nog geen tegenmelding READY heeft plaatsgevonden. Begint men dus aan het begin van een regel met S of G, dan is die S of G na de CR en vóór "READY" er dan weer wel en dan weer niet".

TOEGIFT

Alvast een paar tipjes van de PM-sluier

Welke subroutines heeft PM allemaal in huis? Wat kun je er mee doen in een gebruikersprogramma? Een volledig antwoord op deze vragen krijgt u in boek 4. Dat neemt niet weg dat we dit hoofdstuk willen besluiten met vier toepassingen van PM, die u alvast een blik in de PM-etalage verschaffen. Er zit dus nog glas tussen, maar na de aanschaf van boek 4, in de "Elektuurwinkel", heeft u alles zèlf in huis.

1. Data afdrukken

Zie hoe netjes PM een hex dump afdrukt. Is het nou niet mogelijk om op soortgelijke wijze data van een gebruikersprogramma af te drukken? Denk aan het programma NUMBERS van figuur 13, dat één keer binair en één keer decimaal is doorlopen. Is het mogelijk om de twee getallenreeksen onder elkaar af te drukken?

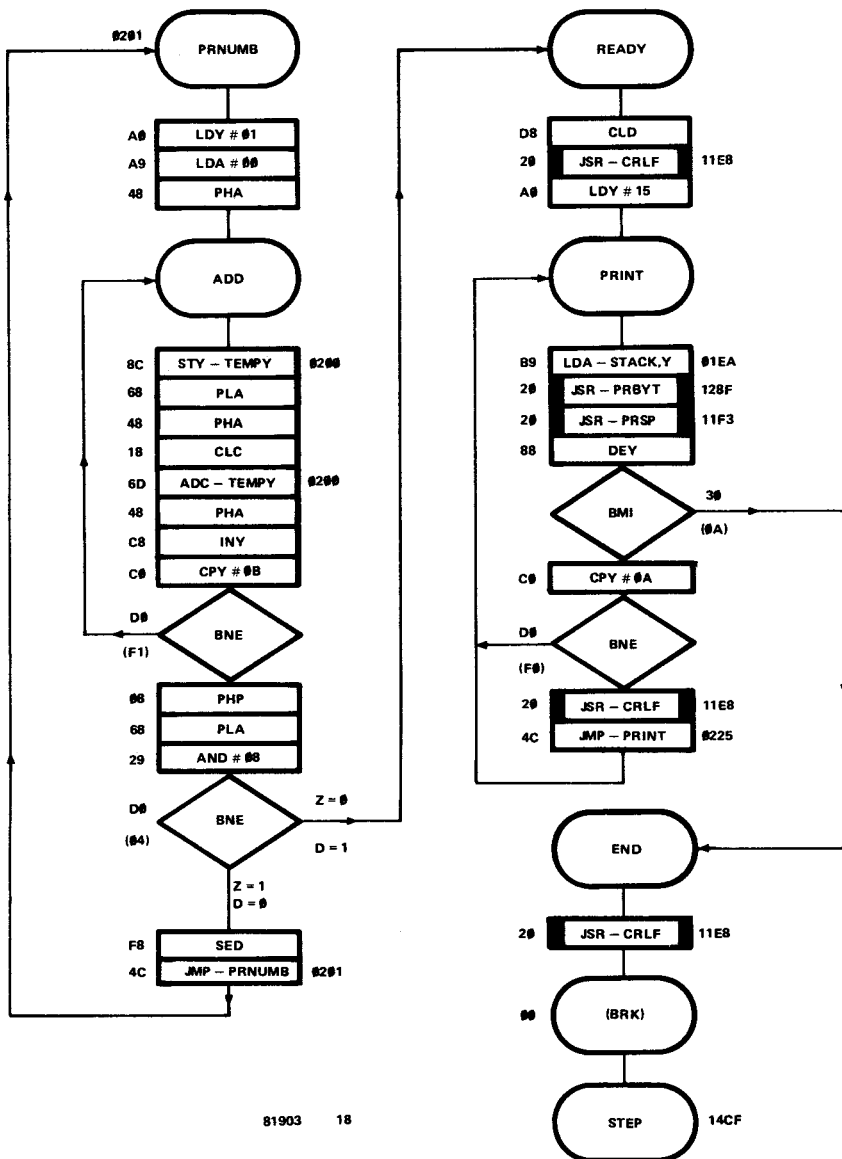
Ja.

Namelijk met het programma van figuur 18. Het heet PRNUMB. Er worden drie verschillende PM-subroutines gebruikt. Dat zijn:

- **CRLF.** Begin op het begin van een nieuwe regel (11E8).
- **PRSP.** Druk een spatie af (11F3).
- **PRBYT.** Verzend de accu-inhoud in de vorm van twee ASCII-bytes (adres 128F).

Hoe zit PRNUMB in elkaar? Het begin is gelijk aan NUMBERS (zie figuur 13). De BNE in het programmadeel na het label ADD staat op springen zolang nog niet alle elf getallen van de NUMBERS-getallenreeks zijn berekend, en op de stapel gezet. Is dat wèl het geval, dan volgt een onderzoek naar de toestand van de D-vlag.

Dat gaat zo: Het P-register wordt in de accu gekopieerd. De PHP plus de PLA zorgen daarvoor. Via de ANDIM 08 maken we alle bits van de accu-inhoud (dus niet van het P-register!) met zekerheid nul, op de D-vlag na. De toestand van de D-vlag geeft antwoord op de vraag of het gedeelte na ADD één keer is doorlopen (binair = hexadecimale berekening van de getallen) of al een tweede keer (decimale berekening). Want als D nul is volgt de overschakeling op decimaal rekenen (SED) en de sprong terug naar PRNUMB.



Figuur 18. Met het programma PRNUMB is het mogelijk om de getallenreeks NUMBERS (zie figuur 13) binair en decimaal te bepalen. De inhoud van de stapel (zie figuur 14) wordt afgedrukt in de vorm van de binaire (hexadecimale) getallenrij met daaronder de decimale getallenrij.

Nu zult u zich misschien afvragen: hoe zijn we er zo zeker van dat de eerste berekening van de getallen binair gebeurt? Omdat het programma eerst moet worden ingetoetst of ingelezen. Hetzij via de standaard-monitor, hetzij via PM (en dat laatste verdient natuurlijk de voorkeur). Na het opstarten van de standaard-monitor of van PM is er automatisch sprake van binair rekenen.

Zodra bij het onderzoek van de D-vlag blijkt dat deze één is, als bewijs dat beide getallenreeksen zijn berekend, volgt de sprong naar READY. Dit tweede gedeelte van PRNUMB houdt zich bezig met het overzichtelijk afdrukken van de getallen.

En heel misschien zult u zich alwéér iets afvragen: was het niet mogelijk om de getallen meteen tijdens de ADD-programmalus, stuk voor stuk af te drukken? Het antwoord: ja en nee. Ja tijdens binair rekenen, nee tijdens decimaal rekenen. Dat komt omdat de kans groot is dat PM-subroutines hun werk niet goed doen als ze via een JSR aan het werk worden gezet terwijl de D-vlag één is! De sprong naar PM (STEP-ingang) via de BINAR-omweg en tijdens D=1 verloopt weliswaar moeiteloos, maar neemt men die omweg niet en sluipt men PM binnen via één van zijn subroutines, en is D één, dan vraagt u om problemen.

Vandaar dat direkt op het label READY orde op zaken wordt gesteld: CLD. Vervolgens zorgen we voor het begin op een nieuwe regel: de sprong naar de subroutine CRLF.

Het is de bedoeling dat de beide getallenreeksen netjes onder elkaar worden afgedrukt: eerst de binaire rij, daaronder de decimale rij. Daartoe maken we Y gelijk aan 15 en laden de accu via absolute Y-geïndexeerde adressering, met als operand-adres 01EA (STACK). Aan het stapelplaatje van figuur 14 kan men zien dat 15 (decimaal: 21) plaatsen van 01EA verwijderd de geheugenplaats 01FF ligt, met het eerste getal van de binaire getallenreeks. Door achtereenvolgens telkens Y met één te verlagen komt na elke omloop via het label PRINT elk getal een keer aan de beurt, dat wil zeggen: in de accu terecht.

Op het laden van de accu volgen de subroutines PRBYT (accu-inhoud afdrukken) en PRSP (spatie als scheiding tussen twee getallen). Daarna gaat Y met één omlaag via DEY. Als Y 00 is geworden is het laatste getal aan afdrukken toe. De waarde van Y na de volgende DEY is FF. Dat betekent dat de N-vlag één is geworden en dat via de BMI de sprong naar het label END volgt: het begin op een nieuwe regel (CRLF), gevolgd door een BRK, dus door een sprong naar de STEP-ingang van PM.

Zodra Y de waarde 0A heeft moet PRNUMB zorgen voor het afdrukken van de rest van de getallen op een nieuwe regel. Immers: Y=0A komt overeen met het eerste getal van de decimale getallenreeks. Kijk het maar na in het stapelplaatje van figuur 14. De instructies CPYIM 0A en BNE PRINT zijn er om bijtijds aan de CRLF-bel te trekken.

Genoeg gepraat. Aan de slag. We toetsen eerst het programma PRNUMB in. Dat gaat heel goed aan de hand van het nu volgende programma-overzicht in assembler-vorm:

```
PRNUMB  0201 A0 01      LDY #01
        0203 A9 00      LDA #00
        0205 48         PHA
ADD      0206 8C 00 02  STY-TEMPY
```

	0209 68		PLA
	020A 48		PHA
	020B 18		CLC
	020C 6D 00 02		ADC-TEMPY
	020F 48		PHA
	0210 C8		INY
	0211 C0 0B		CPY #0B
	0213 D0 F1		BNE ADD
	0215 08		PHP
	0216 68		PLA
	0217 29 08		AND #08
	0219 D0 04		BNE READY
	021B F8		SED
READY	021C 4C 01 02		JMP-PRNUMB
	021F D8		CLD
	0220 20 E8 11		JSR-CRLF
	0223 A0 15		LDY #15
PRINT	0225 B9 EA 01		LDA-(STACK,Y)
	0228 20 8F 12		JSR-PRBYT
	022B 20 F3 11		JSR-PRSP
	022E 88		DEY
	022F 30 0A		BMI END
	0231 C0 0A		CPY #0A
	0233 D0 F0		BNE PRINT
	0235 20 E8 11		JSR-CRLF
	0238 4C 25 02		JMP-PRINT
END	023B 20 E8 11		JSR-CRLF
	023E 00		BRK
	1A7ECF		BRKL
	1A7F14		BRKH

Nu weet u inmiddels zelf wel hoe u dat met de hulp van PM kunt intoetsen.
Laten we het programma maar uitvoeren:

201 (SP)

0201 A0 R

00 01 03 06 0A 0F 15 1C 24 2D 37

00 01 03 06 10 15 21 28 36 45 55

blanco regel

0240 XX

Het programma werkt, en vóór we het vergeten zetten we het op de band:
S1, 201, 23F (CR)

READY

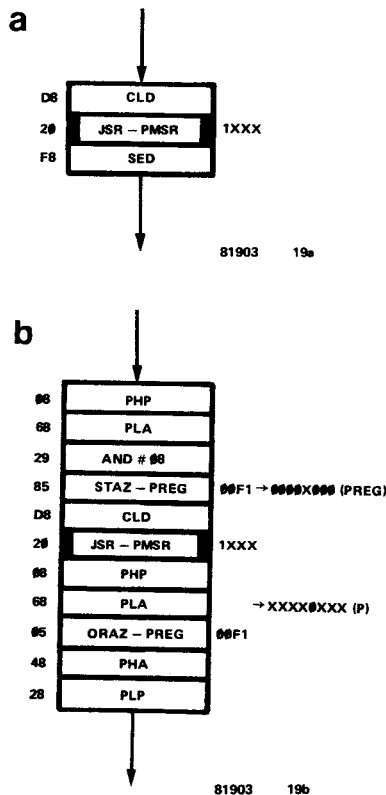
Dat was dat.

Oja: het is in principe mogelijk om een programma als PRNUMB, waarin PM-subroutines voorkomen, te laten uitvoeren onder supervisie van de monitor. Mits de BRK-sprongvektor (dus de IRQ-sprongvektor) op 1C00 is gericht (de terugmelding "0240 XX" verschijnt dan op het zeven-segment-display en niet op het tv-scherm), en verder mits tijdens de computersessie (junior-computer ononderbroken ingeschakeld) PM ooit een keer is opgestart geweest. Dit omdat bepaalde bittijden, die in RAM-plaatsen zijn gespecificeerd, bekend moeten zijn bij die PM-subroutines die iets te

maken hebben met het verzenden of het vaststellen van een ASCII-karakter.

Een belangrijke tip

We hebben gezien dat het aanroepen van PM-subroutines tijdens de situatie: decimaal rekenen problemen oplevert. Zoals besproken zijn die problemen omzeild in het programma PRNUMB. Het is ook mogelijk om, voordat naar een PM-subroutine gaat worden gesprongen, ervoor te zorgen dat wordt overgeschakeld op binair rekenen, als dat nodig is, en na afloop de oorspronkelijke situatie te herstellen. Er kunnen zich twee situaties voordoen:



Figuur 19. De subroutines van PM mogen nooit worden aangeroepen terwijl er sprake is van decimaal rekenen. Is dat permanent het geval tijdens het gebruikersprogramma, dan moet tijdelijk worden overgeschakeld op binair rekenen (19a). Indien beide situaties kunnen optreden (uiteraard nooit tegelijk!), dan moet de oorspronkelijke waarde van de D-vlag na terugkeer uit de PM-subroutine worden hersteld (19b). Daarbij wordt de inhoud van de rest van het P-register beïnvloed door de doorlopen PM-subroutine.

- a. **Er is altijd sprake van decimaal rekenen.** Nou, dat is heel eenvoudig. Laat de sprong naar een PM-subroutine (label PMSR) voorafgaan door een CLD en schakel na afloop terug via een SED. Zie figuur 19a.
- b. **Er is sprake van binair of decimaal rekenen.** Dat is wat minder eenvoudig; zie figuur 19b. Via een PHP en een PLA kopiëren we het P-register in de accu en maken alle bits met zekerheid nul (AND # 08), met uitzondering van de D-vlag. Het resultaat gaat naar de bewaarplaats PREG (00F1) die toch niets te doen heeft omdat PM-subroutines evenals de subroutines van de standaard-monitor niet stap voor stap kunnen worden doorlopen.

Dan volgt de CLD, die eigenlijk niet altijd nodig is, en de aanroep en afwikkeling van de PM-subroutine. Veel van die subroutines hebben ondermeer als eindresultaat een bepaald logisch nivo van één of meer vlaggen. De details komen in boek 4 aan de orde. Met andere woorden: de toestand van het P-register na afloop van JSR-PMSR is belangrijk. We mogen en moeten alleen de oorspronkelijke toestand van de D-vlag herstellen.

Hoe gaat dat? We halen weer via een PHP en een PLA het P-register naar de accu. We weten dat de D-vlag nul is na afloop van PMSR. Via de instructie ORAZ-PREG herstellen we de oorspronkelijke waarde van de D-vlag. Maar dat is pas het geval nadat de inhoud van de accu via de instructies PHA en PLP in het P-register is gezet.

2. Grafisch drukwerk

Kent u de PM-subroutine PRCHA al? Ja en nee. Hoezo? Wel, deze sub-routine wordt twee keer achter elkaar aangeroepen tijdens het verblijf in de subroutine PRBYT, die we in het voorbeeld nummer 1 (PRNUMB, figuur 18) hebben leren kennen. En wat doet deze subroutine? Niets meer of minder dan het verzenden, dus afdrukken van de accu-inhoud. Daarbij is het belangrijk om te weten dat de inhoud van het X-register vóór en na PRCHA er precies hetzelfde uitziet en dat ondanks een intensief gebruik van X tijdens deze subroutine. Het startadres is 1334.

We hebben al numerieke data afgedrukt (voorbeeld 1), gaan straks tekst afdrukken (voorbeeld 4) en willen tijdens het nu aan de orde zijnde voorbeeld 2 eens wat grafisch stoeien. En ook nu geldt: niet omdat dat allemaal kwa praktisch gebruik zo vreselijk zinvol is, maar uitsluitend om kennis te maken met een paar grepen uit de oneindig grote software-trukendoos.

We willen een schaakbord op het tv-scherf weergeven. Maar dan wel een supersimpel schaakbord. In de vorm van acht rijen van elk acht vakjes moeten 32 witte en 23 zwarte vakjes in het juiste patroon op het scherm worden gestrooid. Een wit vakje geven we aan via een spatie, een zwart vakje valt samen met een afgedrukte "X". De hokjes zelf (de vierkantjes) zijn niet zichtbaar gemaakt; een spatie of X geeft het centrum van zo'n hokje weer.

We willen het schaakbord graag vierkant op het tv-scherf weergeven. Een "hifi-schaakbord" dus. Omdat er sprake is van een (vertikale) regelspatie (die ongeveer even groot is als een letterspatie) moeten er naast de spaties die een wit vlak weergeven spaties worden afgedrukt die zorgen voor een

"vierkante weergave". Dat lukt overigens niet helemaal. Het is wel voor- elkaar te krijgen door de beeldhoogte van het tv-scherm te verkleinen. Sommige tv-toestellen hebben hiervoor een "knopje". Het gevolg van die hoogte-afregeling is dat er horizontale zwarte balken onder en boven in het beeld verschijnen. We weten nu wat er moet gebeuren. Nu het "hoe".

Bekijk de nu volgende rijen data eens goed:

```
00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F 10
20 20 58 20 20 20 58 20 20 20 58 20 20 20 58 0D 0A
```

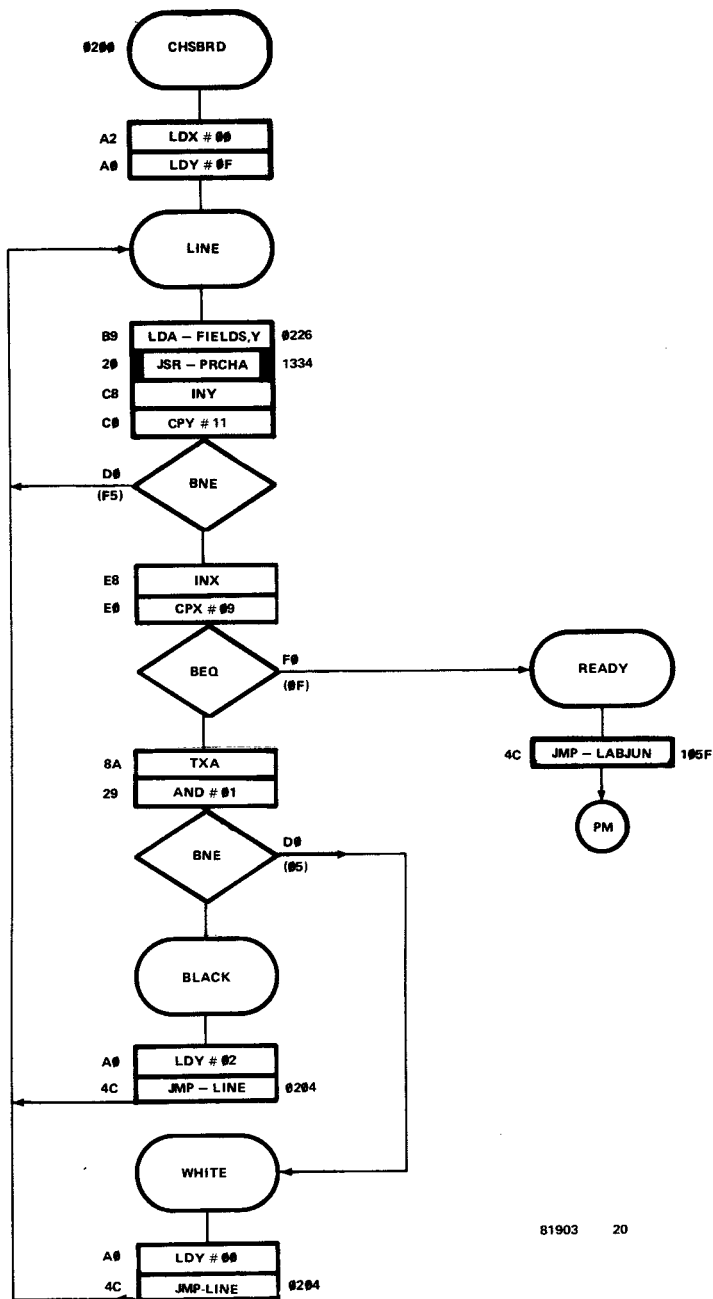
De eerste rij bevat volgnummers, de tweede ASCII-gecodeerde data: 20 hoort bij een spatie, 58 bij de hoofdletter Y, 0D is de ASCII-kode van het grafisch kommando CR en 0A de code van LF. De eerste rij bevat een aantal waarden die het Y-register achtereenvolgens kan aannemen. De tweede rij bevat data, die stuk voor stuk in de accu wordt gezet via absolute Y-geïndexeerde adressering en vervolgens wordt afgedrukt met behulp van de PM-subroutine PRCHA. De datarij hoort bij het afdrukken van een regel van het schaakbord. Het is niet zo dat voor elke regel van het schaakbord alle data van de dataregel wordt gebruikt. De genoemde dataregel wordt negen keer afgewerkt: de eerste keer om ervoor te zorgen dat er op het begin van een nieuwe regel wordt begonnen, de volgende acht keren voor de weergave van een veldenrij van het schaakbord. De stand van zaken wordt bijgehouden in het X-register. Dat doorloopt achtereenvolgens de waarden 00 ... 09:

- Voor $X=00$ is Y gelijk aan 0F; er wordt op een nieuwe regel begonnen.
- Voor de waarden 01, 03, 05 en 07 van X is Y gelijk aan 00: er wordt met een wit veld begonnen.
- Voor de waarden 02, 04, 06 en 08 van X is Y gelijk aan 02: er wordt met een zwart veld begonnen.
- Zodra X de waarde 09 heeft aangenomen weet de computer dat hij met het schaakbord klaar is. Hij zegt dan: "JUNIOR".

Hiermee is al bijna alles verteld over het schaakbordprogramma CHSBRD van figuur 20. De programmalus tussen het label LINE en de eerstvolgende BNE wordt achtereikar doorlopen tijdens het afdrukken van één regel van het schaakbord. De al eerder genoemde dataregel vinden we terug in de vorm van de opzoektabel FILEDS. Uit één van de tot deze tabel behorende geheugenplaatsen wordt de af te drukken accu-inhoud opgehaald. Nadat dit is gebeurd volgt de verhoging van Y met één. Zodra Y de waarde 11 heeft is de regel afgewerkt en is men aan een volgende regel toe. Het zal u niet verbazen dat allereerst X wordt opgehoogd (INX) en wordt gekeken of X soms al 09 is.

Als er nog tenminste één regel moet worden afgedrukt volgt het onderzoek naar de waarde van X. Is-ie even of oneven? Moet er opnieuw een regel worden afgedrukt, te beginnen met $Y=02$ (label BLACK; beginnen met een zwart veld) of te beginnen met $Y=00$ (label WHITE; beginnen met een wit veld)? Hoe kom je aan de weet of X even is of oneven? Door de waarde van X te kopiëren in de accu (TXA) en vervolgens op het meest rechtse bit na alle bits van de accu-inhoud nul te maken (AND #01). Een oneven X levert een Z-vlag nul op, een even X een Z-vlag 1. En de instructie BEQ is nu eenmaal een soort wissel waarvan de stand afhangt van de Z-vlag.

Zodra het hele schaakbord netjes op het tv-scherm is gezet ($X=09$) volgt de sprong naar het label READY van figuur 20 en vervolgens de sprong naar



Figuur 20. Het programma CHSBRD is bestemd voor de weergave van een uiterst primitief schaakbord op het tv-scherm.

het label LABJUN van PM. Het programmeel na LABJUN zorgt voor de terugmelding "JUNIOR", om vervolgens naar het centrale punt van PM te springen (wachten op een ingedrukte toets).

Tot zover de theorie. Nu de praktijk. We geven de (slechts éénmaal!) in te toetsen data ook nu weer in assembler-vorm:

```
CHSBRD 0200 A2 00      LDX #00
        0202 A0 0F      LDY #0F
LINE    0204 B9 26 02   LDA-FIELDS,Y
        0207 20 34 13   JSR-PRCHA
        020A C8         INY
        020B C0 11      CPY #11
        020D D0 F5      BNE LINE
        020F E8         INX
        0210 E0 09      CPX #09
        0212 F0 0F      BEQ READY
        0214 8A         TXA
        0215 29 01      AND #01
        0217 D0 05      BNE WHITE
BLACK   0219 A0 02      LDY #02
        021B 4C 04 02   JMP-LINE
WHITE   021E A0 00      LDY #00
        0220 4C 04 02   JMP-LINE
READY   0223 4C 5F 10   JMP-LABJUN
FIELDS  0226 20
        0227 20
        0228 58
        0229 20
        022A 20
        022B 20
        022C 58
        022D 20
        022E 20
        022F 20
        0230 58
        0231 20
        0232 20
        0233 20
        0234 58
        0235 0D
        0236 0A
```

Eén keer intoetsen is meer dan genoeg. Vandaar:

S7,200,237CR

READY

Het startadres van CHSBRD is het oude vertrouwde 0200. Het is de bedoeling dat we een schaakbord krijgen in de vorm van:

```
  X X X X
X X X X
  X X X X
X X X X
  X X X X
```

```

X X X X
X X X X
X X X X,

```

maar dan zo goed mogelijk aangepast aan allerlei spaties. De praktijk ziet er zó uit:

```

000SPR

```

```

X X X X
X X X X
X X X X
X X X X
X X X X
X X X X
X X X X
X X X X

```

3. Maak zelf een ASCII-tabel!

In ieder computerboek is sprake van de onvermijdelijke ASCII-tabel. Zo ook in dit boek: Zie Aanhangsel 7. En in het artikel/hoofdstuk over het ASCII-toetsenbord staat een heel uitgebreide tabel, met alle 128 mogelijkheden bit voor bit uitgespeld. Nu is dat papier allemaal prachtig natuurlijk, maar is het nou niet mogelijk om een programma te maken dat het mogelijk maakt om de ASCII-kode van een ingedrukte toets af te drukken, en dat dan zowel hexadecimaal als in bits uitgespeld?

Ja, dat is mogelijk.

Dat zullen we u nu laten zien.

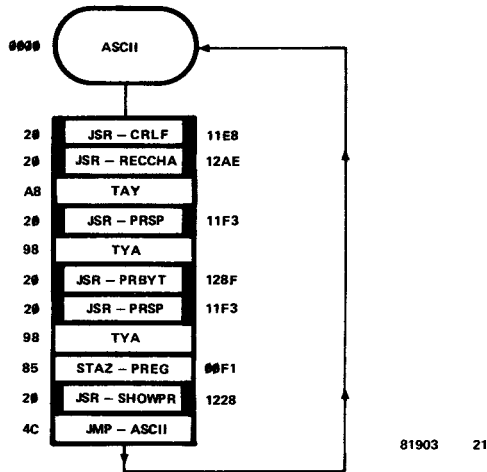
In de voorafgaande twee programmavoorbeelden zijn uitsluitend PM-subroutines aan de orde gekomen waarmee iets werd afgedrukt. Willen we een programma maken waarmee de ASCII-kode van een ingedrukte toets moet worden weergegeven, dan hebben we een subroutine van PM nodig die zich bezig houdt met het wachten op een ingedrukte toets.

Die subroutine is er en heet RECCHA.

Wat doet hij? Hij stelt de binnenkomst vast van een door het ASCII-toetsenbord verzonden karakter (verzonden naar de junior-computer en naar het tv-scherm). De functie zit in de naam verpakt: RECeive CHAracter. De bij de ingedrukte toets horende ASCII-kode ontstaat mede onder invloed van de vraag of het de normale toetsfunctie is of die bij het gelijktijdig indrukken van SFT (Shift) of die bij het gelijktijdig indrukken van CTRL (Control). Na afloop van RCCHA staat de ASCII-kode van de ingedrukte toets (lees: de gekozen toetsfunctie) in de accu.

Het programma voor de weergave van de ASCII-kode van een gekozen toetsfunctie heet — hoe kun je het bedenken! — ASCII en staat in figuur 21. Het startadres is 0000. Het programma doet een niet al te sterk beroep op uw RAM: het omvat slechts 26 bytes. Een redelijke hoeveelheid, dachten we zo.

Het begint met de subroutine CRLF: begin op het begin van een nieuwe regel. Dan volgt het wachten op een ingedrukte toets oftewel uw keuze van een toetsfunctie waarvan u de ASCII-kode wilt leren kennen. Voorzover het een toets betreft die zich zichtbaar in de vorm van een afdruk op het tv-scherm uit, komt die "uiting" op het begin van een regel te staan.



Figuur 21. Het voorbeeldprogramma ASCII. Hiermee kan de ASCII-kode worden weergegeven van een willekeurige ingedrukte toets.

De ASCII-kode van de gekozen toetsfunctie staat nu in de accu en wordt gekopieerd in het Y-register (TAY). Waarom? Omdat bij de nu volgende JSR-PRSP de inhoud van de accu verandert: er wordt een spatie afgedrukt als scheiding tussen de afgedrukte toetsfunctie en de daarop volgende hexadecimale weergave van de ASCII-kode. Die weergave vindt plaats via het in de accu zetten van de oorspronkelijke accu-inhoud (TYA) en de aanroep van de subroutine PRBYT. Die kennen we al van het program-ma-voorbeeld 1 : PRNUMB. Hij zorgt ervoor dat eerst het linker nibble en dat het rechter nibble van de accu-inhoud wordt afgedrukt. De waarden van de nibbles variëren van 0 tot en met F. Vóór de eigenlijke verzending wordt elk nibble vertaald in de bijbehorende ASCII-kode.

Na de afdruk van beide nibbles, dus na de afdruk van de hexadecimale weergave van de ASCII-kode, volgt voor de tweede keer de afdruk van een spatie. En dan zijn we toe aan de bit-voor-bit-weergave van de ASCII-kode. Eerst halen we opnieuw de oorspronkelijke accu-inhoud binnen (TYA). En dan kopiëren we de accu-inhoud in de welbekende geheugenplaats PREG (00F1) en springen naar de subroutine SHOWPR.

Waarom moet zo nodig de accu naar PREG? Het antwoord is eenvoudig: Tijdens het maken van een listing wordt de inhoud van het P-register bit voor bit weergegeven. Kijk het maar na bij de bespreking van de L. Voor die weergave is de subroutine SHOWPR er. Nu zijn we niet geïnteresseerd in de inhoud van PREG. Want die heeft uitsluitend betekenis als die na een sprong naar PM (STEP-ingang) wordt gevuld met de nieuwste programmagegevens. En aangezien we programma's waarin PM-subroutines zijn gebruikt tòch niet stap voor stap kunnen doorlopen mogen we rustig de **geheugenplaats** PREG voor oneigenlijke doeleinden gebruiken, door er een ASCII-kode in te zetten.

Nadat alle bits van de ASCII-kode zijn afgedrukt volgt de terugkeer uit SHOWPR en de sprong terug (JMP-ASCII) naar het begin. Na het klaarzetten van een nieuwe regel is het wachten (RECCHA) op een nieuwe ingedrukte toets, al dan niet in combinatie met een ingedrukte toets SFT of CTRL. Het programma ASCII bevat dus een wachtlus (zit in RECCHA). Men kan het verlaten via het indrukken van toets RST en het vervolgens opnieuw opstarten van PM, of door het indrukken van de toets BRK tijdens het afdrucken van een ASCII-kode.

Aangezien we nu geen bemoeienis hebben met het **hoofdprogramma** van PM zal er geen foutmelding "WHAT?" volgen na het indrukken van een niet-PM-toets (zie figuur 12). Dus alle toetsen van het toetsenbord, dat wil zeggen: alle toetsen waarvan het indrukken de verzending van een ASCII-kode tot gevolg heeft, doen mee.

We zijn gauw klaar met het intoetsen van het programma ASCII van figuur 21:

```
ASCII 0000 20 E8 11 JSR-CRLF
      0003 20 AE 12 JSR-RECCHA
      0006 A8      TAY
      0007 20 F3 11 JSR-PRSP
      000A 98      TYA
      000B 20 8F 12 JSR-PRBYT
      000E 20 F3 11 JSR-PRSP
      0011 98      TYA
      0012 85 F1    STAZ-PREG
      0014 20 28 12 JSR-SHOWPR
      0017 4C 00 00 JMP-ASCII
```

Het is weinig, maar de moeite waard om op de band te zetten:

S8,,1A(CR)

READY

We starten het programma:

(SP)R

en kunnen vervolgens naar hartelust toetsen indrukken. In tabel 3 is, in de vorm van drie kolommen naast elkaar, een aantal toetsfuncties te zien met daarnaast de bijbehorende ASCII-kode in hexadecimale vorm en daarnaast dezelfde code, maar dan in acht bits weergegeven. Het meest linkse bit is altijd nul; de eigenlijke ASCII-kode omvat zeven bits.

Tabel 3 is tot stand gekomen met behulp van het programma ASCII van figuur 21. In plaats van de Elekterminal is er een grote papierprinter op de junior-computer aangesloten. Deze heeft wat meer toetsfuncties in huis dan de Elekterminal. Het indrukken van de spatietoets levert uitsluitend de ASCII-kode (20) op; dat heb je met spaties. Ook de grafische kommando's CR (0D) en LF (0A) zijn uitsluitend kwa uitwerking te zien.

Het programma ASCII kan meteen al heel goed van pas komen bij het nu volgende vierde programmavoorbeeld, waarbij weer te geven teksten eerst in ASCII-kode vooraf moeten worden opgegeven.

4. Tekst afdrucken

Het vierde en laatste programma uit de PM-etalage heet FINAL en staat in figuur 22. Waarom het zo nodig FINAL moet heten, daar komt u nog wel

Tabel 3. Een "hard copy" van een ASCII-tabel, die is gemaakt met behulp van het voorbeeldprogramma ASCII van figuur 21.

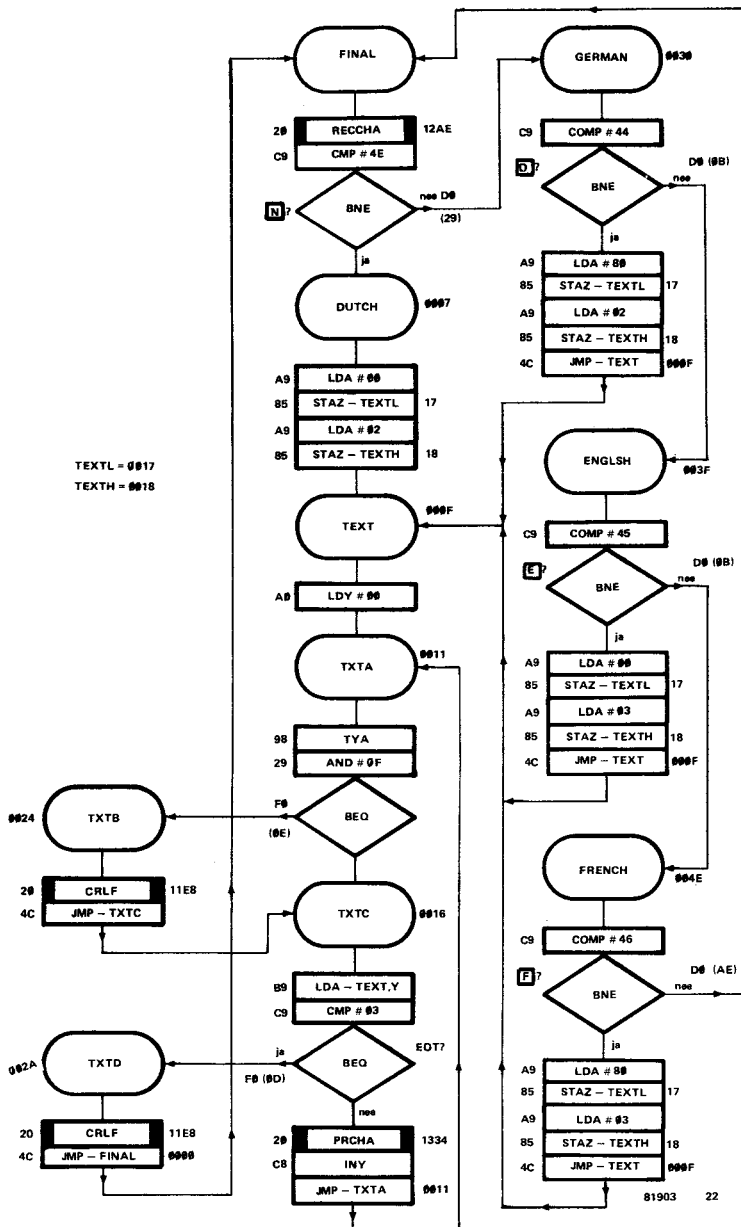
0 30 00110000	P 50 01010000	# 23 00100011
1 31 00110001	Q 51 01010001	\$ 24 00100100
2 32 00110010	R 52 01010010	% 25 00100101
3 33 00110011	S 53 01010011	^ 5E 01011110
4 34 00110100	T 54 01010100	& 26 00100110
5 35 00110101	U 55 01010101	* 2A 00101010
6 36 00110110	V 56 01010110	(28 00101000
7 37 00110111	W 57 01010111	) 29 00101001
8 38 00111000	X 58 01011000	5F 01011111
9 39 00111001	Y 59 01011001	5F 01011111
A 41 01000001	Z 5A 01011010	+ 2B 00101011
B 42 01000010	, 2C 00101100	~ 7E 01111110
C 43 01000011	. 2E 00101110	5D 01011101
D 44 01000100	20 00100000	! 7C 01111100
E 45 01000101	/ 2F 00101111	: 3A 00111010
F 46 01000110	{ 7B 01111011	" 22 00100010
G 47 01000111	' 27 00100111	} 7D 01111101
H 48 01001000	; 3B 00111011	< 3C 00111100
I 49 01001001	\ 5C 01011100	> 3E 00111110
J 4A 01001010	¡ 5B 01011011	? 3F 00111111
K 4B 01001011	` 60 01100000	0D 00001101
L 4C 01001100	= 3D 00111101	0A 00001010
M 4D 01001101	- 2D 00101101	
N 4E 01001110	! 21 00100001	
O 4F 01001111	@ 40 01000000	

achter. Ook nu maken we weer gebruik van de PM-subroutine RECCHA voor het vaststellen van een ingedrukte toets, en verder van twee afdruk-subroutines van PM.

De computer, of juister: de junior-computer is, via PM, uitstekend geschikt voor het weergeven van teksten. En dan bedoelen we niet van die eeuwig vaststaande teksten als "WHAT?" "JUNIOR" en zo, maar teksten die men als gebruiker zelf kan bedenken, ze vervolgens aan de computer kan opgeven en ze door hem kan laten afdrukken, zo vaak als men wil.

Welnu, het programma FINAL is een soort internationale tekstmonitor. De toetsen N (Nederlands), D (Deutsch), E (English) en F (Francais) krijgen een speciale betekenis. Het indrukken ervan levert het ophoesten, door de junior-computer, van een tekst in de betreffende taal. De teksten moeten vooraf worden opgegeven, in RAM.

Waarom vier talen? Omdat de junior-computerboeken in vier talen verschijnen. Tevens verschaft het de lezer inzicht in de wijze waarop je zelf te



Figuur 22. Met het programma FINAL kan een keuze worden gemaakt uit vier verschillende teksten, die op het tv-scherm kunnen worden weergegeven.

werk moet gaan als je verschillende toetsen voor eigen software wil bestemmen.. Een soort privé-monitor dus.

De teksten worden afgedrukt in de vorm van regels met maximaal zestien karakters per regel. De pagina's 02 en 03 (standaard-RAM) zijn beschikbaar voor de af te drukken teksten. Iedere taal krijgt even veel ruimte toebedeeld (lang leve de EEG):

N: 0200 ... 027F; TEXTL=00; TEXTH=02

D: 0280 ... 02FF; TEXTL=80; TEXTH=02

E: 0300 ... 03FF; TEXTL=00; TEXTH=03

F: 0380 ... 03BF; TEXTL=80; TEXTH=03

Iedere tekst kan dus bestaan uit maximaal acht regels van maximaal 16 karakters elk. De tekst moet zijn afgesloten met een zogenaamd EOT-teken (End Of Text; ASCII:03), dus elke tekst bestaat uit maximaal 127 karakters. De karakters moeten in ASCII-kode worden opgegeven. Zie daarvoor Aanhangsel 7.

Eerst maar eens iets meer over het eigenlijke afdruk-gedeelte van FINAL. Dat begint bij het label TEXT. En dat begint met het nul maken van Y. Vervolgens de test of Y soms X0 is, met X gelijk aan 0 ... 7. Is dat namelijk het geval, dan moet er op een nieuwe regel worden begonnen, alvorens er nieuwe karakters worden afgedrukt. De rechter nibble-nul-test gaat via het kopiëren van Y in A (TAY), het nul maken van het linker accu-nibble (AND #0F) en het bekijken van de Z-vlag, via een aansluitende BEQ.

Vervolgens het laden van de accu met het (volgende) af te drukken karakter. Dat gebeurt via absolute Y-geïndexeerde adressering. De adres-operand van LDA-TEXT,Y is gespecificeerd op de geheugenplaatsen 0017 en 0018; op adres 0016 staat de opcode, B9. De genoemde geheugenplaatsen, TEXTL en TEXTH, worden afhankelijk van de gekozen taal vooraf geladen met data die overeenkomt met het eerste adres van het betreffende tekstgeheugen.

Iedere tekst moet zoals gezegd zijn afgesloten met een EOT, anders wordt het een tv-schermdisplay vol onzin. Zodra een 03 is ontdekt leidt de BEQ ons naar de begin-op-een-nieuwe-regel-subroutine CRLF en vervolgens terug naar het begin: FINAL. Zo lang dat niet het geval is wordt de accu-inhoud afgedrukt (PRCHA),Y met één verhoogd en het volgende karakter opgehaald uit het tekstgeheugen.

Nu de toetsenroutine. FINAL begint met RECCHA: zet een ontvangen ASCII-karakter in de accu. Ga vervolgens via een aantal CMPs + BNEs na of het ontvangen ASCII-karakter soms hoort bij één van de toetsen N, D, E of F. Zoniet: terug naar FINAL. Zoja: Pas de operand van LDA-TEXT,Y aan aan de taal.

Tot zover de bespreking van FINAL. We kunnen nu het programma intoetsen aan de hand van het volgende overzicht:

FINAL	0000	20	AE	12	JSR-RECCHA
	0003	C9	4E		CMP #4E
	0005	D0	29		BNE GERMAN
DUTCH	0007	A9	00		LDA #00
	0009	85	17		STAZ-TEXTL
	000B	A9	02		LDA #02
	000D	85	18		STAZ-TEXTH

TEXT	000F	A0	00		LDY # 00
TXTA	0011	98			TYA
	0012	29	0F		AND # 0F
	0014	F0	0E		BEQ TXTB
TXTC	0016	B9	XX	XX	LDA-TEXT,Y
	0019	C9	03		CMP # 03
	001B	F0	0D		BEQ TXTD
	001D	20	34	13	JSR-PRCHA
	0020	C8			INY
	0021	4C	11	00	JMP-TXTA
TXTB	0024	20	E8	11	JSR-CRLF
	0027	4C	16	00	JMP-TXTC
TXTD	002A	20	E8	11	JSR-CRLF
	002D	4C	00	00	JMP-FINAL
GERMAN	0030	C9	44		CMP # 44
	0032	D0	0B		BNE ENGLISH
	0034	A9	80		LDA # 80
	0036	85	17		STAZ-TEXTL
	0038	A9	02		LDA # 02
	003A	85	18		STAZ-TEXTH
	003C	4C	0F	00	JMP-TEXT
ENGLISH	003F	C9	45		CMP # 45
	0041	D0	0B		BNE FRENCH
	0043	A9	00		LDA # 00
	0045	85	17		STAZ-TEXTL
	0047	A9	03		LDA # 03
	0049	85	18		STAZ-TEXTH
	004B	4C	0F	00	JMP-TEXT
FRENCH	004E	C9	46		CMP # 46
	0050	D0	AE		BNE FINAL
	0052	A9	80		LDA # 80
	0054	85	17		STAZ-TEXTL
	0056	A9	03		LDA # 03
	0058	85	18		STAZ-TEXTH
	005A	4C	0F	00	JMP-TEXT

Zo, dat zit er in. Er hoeft geen IRQ-sprongvektor te worden opgegeven omdat er geen BRK voorkomt: het programma FINAL bestaat uit een in principe oneindig lang durende toetsenwachtlus. Het programma FINAL kan worden verlaten door het indrukken van de toets RST (terug naar de standaard-monitor); via 1 0 0 0 GO RUB OUT terug naar PM. Je kunt het programma ook afbreken door tijdens het afdrukken van tekst de BRK-toets in te drukken. Een derde manier is om de toets ST van het standaard-toetsenbord in te drukken. De BRK- en ST-oplossingen lukken alleen dan indien de BRK- en NMI-sprongvectoren zijn gedefinieerd in overeenstemming met de opstartroutine van PM, dus indien tijdens de computer-sessie PM is opgestart. En dat ligt voor de hand, want met PM geef je nog makkelijker data op met de hand of van de band, dan met de standaard-monitor, respektievelijk met TM.

En dan de tekstdata. We beginnen met de Nederlandse tekst:

M

HEXDUMP: 200,230

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0200:	42	45	53	54	45	20	4D	45	4E	53	45	4E	21	44	49	54
0210:	57	41	53	20	48	45	54	20	44	41	4E	2E	54	4F	54	20
0220:	5A	49	45	4E	53	20	49	4E	20	42	4F	45	4B	20	34	21
0230:	03															

JUNIOR

Vergeet vooral niet het einde-tekstteken, 03, op adres 0230. Doet men dat niet, dan wordt er na de "echte" tekst willekeurige data afgedrukt en is de kans groot dat er oneindig lang wordt doorgegaan (tenzij ergens toevallig 03 staat); de BRK-toets biedt dan uitkomst. Die willekeurige, af te drukken data kan de ASCII-kode betreffen van het een of andere grafische kommando, zodat bijvoorbeeld op een gegeven moment bovenaan het tv-scherm met afdrukken wordt begonnen: de bestaande tekst wordt overschreven. Dát gebeurt allemaal niet als de tekst is afgesloten met 03.

Dan de Duitse tekst:

M

HEXDUMP: 280,2BF

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0280:	4C	49	45	42	45	52	20	4C	45	53	45	52	21	44	41	53
0290:	57	41	52	20	45	53	20	46	55	45	52	20	48	45	55	2D
02A0:	54	45	2E	41	55	46	20	57	49	45	44	45	52	53	45	2D
02B0:	48	45	4E	20	49	4E	20	42	55	43	48	20	34	21	21	03
02C0:																

JUNIOR

Vervolgens zwemmen we de Noordzee over:

M

HEXDUMP: 300,340

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0300:	44	45	41	52	20	4D	52	2F	4D	52	53	2F	4D	49	53	53
0310:	4A	55	4E	49	4F	52	21	49	54	27	53	20	54	49	4D	45
0320:	54	4F	20	47	4F	2E	53	45	45	20	59	4F	55	20	20	20
0330:	41	47	41	49	4E	20	49	4E	20	42	4F	4F	4B	20	34	21
0340:	03															

JUNIOR

De Fransen zijn wat langer van stof:

M

HEXDUMP: 380,3E0

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0380:	43	48	45	52	53	20	4C	45	43	54	45	55	52	53	2C	20
0390:	43	27	45	53	54	20	46	49	4E	49	20	50	4F	55	52	20
03A0:	41	55	4A	4F	55	52	44	27	48	55	49	21	20	20	20	20
03B0:	52	45	4E	44	45	5A	2D	56	4F	55	53	20	41	55	58	20
03C0:	50	52	45	4D	49	45	52	45	53	20	50	41	47	45	53	20
03D0:	44	55	20	4C	49	56	52	45	20	51	55	41	54	52	45	20
03E0:	03															

JUNIOR

Na al deze "nationale bytes" volgt nu nog de hex dump van het programma
FINAL:

M

HEXDUMP: ,5C

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0000:	20	AE	12	C9	4E	D0	29	A9	00	85	17	A9	02	85	18	A0
0010:	00	98	29	0F	F0	0E	B9	00	02	C9	03	F0	0D	20	34	13
0020:	C8	4C	11	00	20	E8	11	4C	16	00	20	E8	11	4C	00	00
0030:	C9	44	D0	0B	A9	80	85	17	A9	02	85	18	4C	0F	00	C9
0040:	45	D0	0B	A9	00	85	17	A9	03	85	18	4C	0F	00	C9	46
0050:	D0	AE	A9	80	85	17	A9	03	85	18	4C	0F	00			

JUNIOR

Na al dat toetswerk gaan we eerst het een en ander op de band zetten. We
starten de cassette-recorder in de opnamestand:

S2,,5D(CR)

READY (pauzetoets indrukken)

S3,200,231(CR)

READY (pauzetoets indrukken)

S4,380,3E1(CR)

READY (pauzetoets indrukken)

S5,280,2C0(CR)

READY (pauzetoets indrukken)

S6,300,341(CR)

READY (cassette-recorder stoppen)

Zo, dat kan maar gebeurd zijn.

Nu het startadres opgeven, het programma FINAL starten en toets N indrukken. Later mag u van ons óók de toetsen D, E en F indrukken. Daar gaat-ie:
(SP)R

**N
BESTE MENSEN!DIT
WAS HET DAN.TOT
ZIENS IN BOEK 4!**

Hetgeen helemaal klopt.

Basiskaart plus één RAM/EPROM-kaart

Uitsluitend geheugenuitbreiding

Er zijn bepaalde situaties denkbaar waarbij kan worden volstaan met een vergroting van de junior-computer in de vorm van uitsluitend geheugenuitbreiding. Men kan dan de interface-kaart nemen en daar alle onderdelen op zetten die van belang zijn voor het extra geheugen op die kaart. Het is echter handiger om in dat geval één RAM/EPROM-kaart direkt aan te sluiten op de basiskaart. Slechts één kaart kan worden aangesloten omdat de uitbreidingskonnektor van de standaard-junior-computer niet is gebufferd.

Er gelden de volgende gebruiksaanwijzingen:

- ① **Leg de draadbrug D-EX op de standaardkaart.** Dit vanwege het feit dat er nu niet uitsluitend iets op de basiskaart moet kunnen worden geadresseerd, maar ook op de aan te sluiten RAM/EPROM-kaart. De vrije selectiesignalen K1 . . . K5 kunnen namelijk niet worden gebruikt.
- ② **Maak op de RAM/EPROM-kaart een draadverbinding tussen pen 3 van IC29 of pen 4 van IC8 (uitgang van N5; zie figuur 25 van hoofdstuk 10) en de EX-aansluiting (pen 30c van de uitbreidingskonnektor en van de aansluitkonnektor van de RAM/EPROM-kaart).**

Waarom? Het uitgangsnivo van N5 wordt bepaald door de ingangsnivo's van N1 en N2. En dat zijn de aansluitingen X, Y, V en W, waarvan er één of meerdere worden gebruikt voor (een) doorverbinding(en) met (een) aansluitpunt(en) van de hoofdadresdekoder IC5. Zolang geen der ingangen van N1 en N2 logisch nul is zijn de uitgangen van N1 en N2 logisch één en is dus de uitgang van N5 logisch nul. Dus is EX logisch nul en dus werkt de adresdekoder IC6 op de standaardkaart net zo als vroeger: er kan iets op de standaardkaart worden geadresseerd en geselecteerd. Zodra één van de ingangen van N1 en N2 logisch nul wordt, omdat er een 4K-adresblok op de RAM/EPROM-kaart wordt geselecteerd, is de uitgang van N5 logisch één en wordt de D-ingang van IC6 op de basiskaart logisch één: er kan iets worden geadresseerd dat niet op de standaardkaart zit (en ook niet op de interface-kaart; die speelt hier geen enkele rol). Voor de werking van de standaard-adresdekoder: zie figuur 4 en tabel 1 van hoofdstuk 10.

Dit heeft gevolgen. Namelijk:

- ③ **Maak geen gebruik van de aansluitpunten "0", "1" en – indien de vektoren van de standaard-EPROM op de standaardkaart moeten komen – "F" van de hoofdadresdekoder!** Zou je dat namelijk wel doen, dan zou in het geval van de adressen 0XXX, 1XXX (en eventueel

FXXX) het signaal EX logisch één worden en dat betekent dat de voor die adressen relevante selectie op de standaardkaart (via IC6) niet tot stand komt.

Nog los daarvan is er een tweede reden om de aansluitingen "0" en "1" niet te gebruiken. Stel men wil het gat in de memory map, dat bij de standaardkaart ligt tussen 0400 en 17FF, opvullen met geheugen op die RAM/EPROM-kaart (het vullen van het gat 1800 . . . 19FF gaat sowieso niet omdat dit tot door K6 geselecteerd gebied hoort). Dan zal er altijd wel een 4K-gebied zijn dat voor een deel wordt bezet door geheugen op de RAM/EPROM-kaart en voor een ander deel door geheugen op de standaardkaart. Leest men geheugen op de standaardkaart dat tot een 4K-blok hoort waarvan een ander deel op de RAM-EPROM-kaart zit, dan zullen daardoor de databuffers van de RAM/EPROM-kaart in de leesrichting actief zijn: er treedt de gevaarlijke data-parallelsituatie op die in hoofdstuk 10 (zie ook figuur 6) is besproken.

- ④ In het geval dat de vektoren NMI, RES en IRQ uit de standaard-EPROM op de standaardkaart worden gehaald hoeft men er alleen nog maar voor te zorgen dat de EPROM is aangesloten. Aansluiting "F" van IC5 op de RAM/EPROM-kaart hoeft en mag zelfs niet worden gebruikt. De adreslijnen A13, A14 en A15 zijn nu niet werkzaam en de vektoren worden uit de standaard-EPROM (pagina 1F in plaats van FF) gehaald. **De adressen F000 . . . FFFF kunnen niet worden gebruikt voor geheugen op de RAM/EPROM-kaart.** Voor één kaart blijft er echter nog genoeg over: de pagina's 20 . . . EF. Dat is 52K en meer dan de 24K, die een RAM/EPROM-kaart maximaal (8K-RAM plus 16K-EPROM) kan herbergen.

- ⑤ In het geval dat de vektoren uit EPROM op de RAM/EPROM-kaart moeten komen mag, of sterker: **moet aansluiting "F" worden gebruikt.** Tevens moet op pagina FF EPROM zijn aangesloten, met gespecificeerde vektor-data. Dat betreft dan de adressen:

FFFA voor NMIL;
FFFB voor NMIH;
FFFC voor RESL;
FFFD voor RESH;
FFFE voor IRQL;
FFFF voor IRQH.

Voor meer details verwijzen we u naar het laatste deel van hoofdstuk 10, dat gaat over de RAM/EPROM-kaart. Het bruikbare adresgebied op de RAM/EPROM-kaart omvat de pagina's 20 . . . FF.

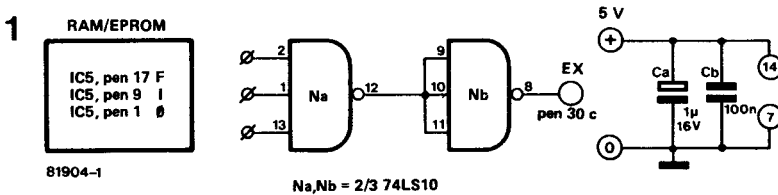
Even een vraag: hoe zit het nu met die **achtvoudige adressering**? U weet wel, die kwestie dat bij de standaard-junior-computer de pagina's 0X samenvallen met de pagina's 2X, 4X, 6X, 8X, AX, CX en EX, en dat de pagina's 1X samenvallen met 3X, 5X, 7X, 9X, BX, DX en FX (X = 0 . . . F). Welnu, er is sprake van standaarddekodering op de standaardkaart (IC6) zolang het om een adres gaat dat niet behoort tot een 4K-adresblok dat op de RAM/EPROM-kaart wordt gedekodeerd. Het hangt er helemaal van af hoeveel van die 4K-blokken (dus IC5-aansluitingen) op de RAM/EPROM-kaart zijn gebruikt als het gaat om de vraag hoe het zit met die meervoudige adressering. Achtvoudig zal hij in ieder geval niet meer zijn. En als de vektoren uit de EPROM op de

standaardkaart komen is er voor de pagina's 1X ($X = 0 \dots F$) sprake van een minstens tweevoudige adressering.

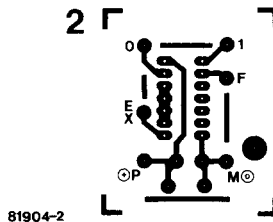
- ⑥ De **aansluitingen** tussen de beide kaarten kunnen plaats vinden via korte draadjes, via konnektors of via konnektors, gekombineerd met flat cable. De figuren 19 en 20 van hoofdstuk 10 geven informatie over de praktische realisatie.
- ⑦ Voor de **voeding** van één RAM/EPROM-kaart kan men nog wel terecht bij de standaard-voeding van de junior-computer. Eventueel de koeling wat verzwaren. De 2708 heeft naast +5 V -5 V en +12 V nodig, de 2716 doet het zonder -5 V en +12 V. De extra RAM doet alleen een beroep op +5 V. Dus met name bij gebruik van 2716's zijn er nauwelijks problemen te verwachten bij het gebruik van de standaardvoeding.

Alternatief

De voorbereiding van de RAM/EPROM-kaart voor gebruik met de standaard-junior-computer bestond uit het leggen van de verbinding tussen de uitgang van N5 op de RAM/EPROM-kaart via EX met punt D van IC6 op de standaardkaart. Het kan ook anders, met de schakeling van figuur 1.



Deze heeft het voordeel dat nu wél de achtvoudige adressering volledig is opgeheven (met uitzondering van een tweevoudige adressering in het geval dat de vektoren van de standaardkaart worden gehaald: pagina's 1X vallen samen met FX, en dat is dan juist de bedoeling vanwege die vektor-bedotterij). Een nadeel is dat de praktische realisatie wat meer voeten in de aarde heeft: óf op een printje (figuur 2), óf op een stukje gaatjesprint, aan de hand van figuur 2.



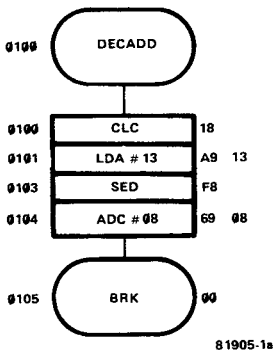
De verbinding met aansluitpunt F van IC5 komt te vervallen als de vektoren uit de standaard-EPROM worden opgehaald. Mijd ook nu de aansluitingen Ø,1 voor doorverbindingen met X, Y, V of W, omdat de bijbehorende 4K-adresblokken toch niet helemaal kunnen worden benut (een

PIA-RAM-uitbreiding van de standaard-monitor

Stap voor stap decimaal rekenen

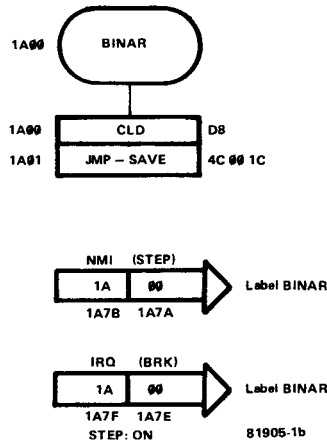
Stel, we maken geen gebruik van de Printer Monitor (PM), dus van de standaard-monitor. We willen een programma waarin decimaal wordt gerekend, dus waarin de instructie SED (F8) voorkomt, stap voor stap oftewel instructie voor instructie uitvoeren. Nemen we als voorbeeld het programma DECADD van figuur 1a.

1a



81905-1a

1b



We toetsen de data in, richten de NMI- en de IRQ-sprongvektor op 1C00 (SAVE-ingang van de monitor), zetten de STEP-schakelaar ON, en toetsen het volgende:

toetsen

AD 0 1 0 0 GO
(uitstapje) PC
GO
(uitstapje) PC
GO
(AD) 0
0
F
1

display

0101 A9 (instructie CLC uitgevoerd)
0101 A9
0103 F8 (instructie LDA uitgevoerd)
0103 F8
0104 69 (instructie SED uitgevoerd)
1040 XX
0400 XX
0400 XX hé, wat is dat nou!?
4001 XX

We wilden zo graag na de uitvoering van de instructie SED (decimaal rekenen) de inhoud van het P-register (00F1) bekijken om na te gaan of de D-vlag inderdaad 1 is geworden. De toets AD staat tussen haakjes omdat er automatisch sprake is van de adresmode als men via SAVE of RESET de monitor binnen komt. Het blijkt dat de monitor niet reageert op het indrukken van toets F, en ook niet op toets PC, als men besluit om dan maar zonder controle van het P-register verder te gaan.

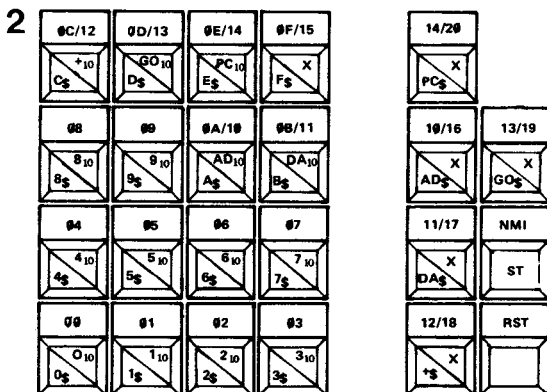
Wat is er aan de hand?

Na de uitvoering van de instructie SED is er sprake van decimaal rekenen. Voor de korrekte werking van de monitor is het nodig dat er binair (hexa-decimaal) wordt gerekend. Vandaar dat de RESET-ouverture van de monitor de instructie CLD bevat, zie figuur 3a. De SAVE-routine echter niet! Op zich is er geen bezwaar tegen om ook in de SAVE-routine een CLD op te nemen. Immers, het P-register van het gebruikersprogramma wordt bij elke sprong naar SAVE bewaard en weer hersteld na de sprong terug van monitor naar programma, tijdens de uitvoering van het programma GOEXEC van de monitor (na het indrukken van GO; zie hoofdstuk 7 van boek 2); er mag dus best een vlag (trouwens: alle elementen van het software-dashboard) worden veranderd tijdens het verblijf in de monitor, omdat dat later allemaal weer op zijn pootjes terecht komt.

Gevolgen

Het is niet zo dat bij decimaal rekenen de monitor compleet onbruikbaar is geworden. Zie figuur 2. We zien dat een aantal **toetsen** een andere **toetsfunctie** heeft gekregen, inclusief de toetsfunctie "zo dood als een pier", voor de toetsen F en AD, DA, +, GO en PC. De toetsen RST en ST blijven als vanouds beschikbaar; de numerieke toetsfuncties A . . . F zijn niet meer beschikbaar. De index dollartekens in figuur 2 duidt op de normale toetsfunctie van de betreffende toets, de index 10 op de toetsfunctie bij decimaal rekenen.

Hoe komt dat? De monitor werkt op basis van de toetswaarde van een ingedrukte toets. Zie figuur 16 op pagina 122 van boek 2, en de bijbe-



81906-2

horende tekst. Die toetswaarden kwamen via binair en komen nu via decimaal rekenen tot stand, met als gevolg voor een deel gewijzigde toetswaarden, die in figuur 2 zijn aangegeven (achter de schuine streep).

De kommando-toetsfuncties hebben dus een andere toets toegewezen gekregen. De numerieke toetsfuncties A . . . F zijn niet beschikbaar en dat betekent dat tijdens het verblijf in de monitor geen adressen kunnen worden opgegeven die één of meerdere nibbles A . . . F bevatten. Je moet dan het dichtstbij gelegen lagere adres intoetsen dat uitsluitend 0 . . . 9 bevat en dan maar net zo lang op de plustoets (= toets C!) timmeren totdat het gewenste adres voorstaat . . . Dat is natuurlijk een hopeloze zaak. Verder is het lastig dat de kommando-toetsen zijn verhuisd. Dáár moet iets aan gedaan worden!

Oplossing één: EPROM wijzigen

Alle decimale problemen voor de monitor zijn de wereld uit als óók de binnenkomst via de SAVE-ingang van de monitor tot een CLD leidt, dus als we van figuur 3a naar figuur 3b gaan. De oplossing schuilt in het verplaatsen van het label START en in het stuivertje wisselen van de instructies SEI en CLD. De laatste instructie hoort nu niet meer bij de RESET-routine maar bij het hoofdprogramma van de monitor, dat begint bij START.

De volgende drie geheugenplaatsen van de EPROM moeten worden gewijzigd:

1C1B 32 (was 33)

1C31 78 (was D8)

1C32 D8 (was 78)

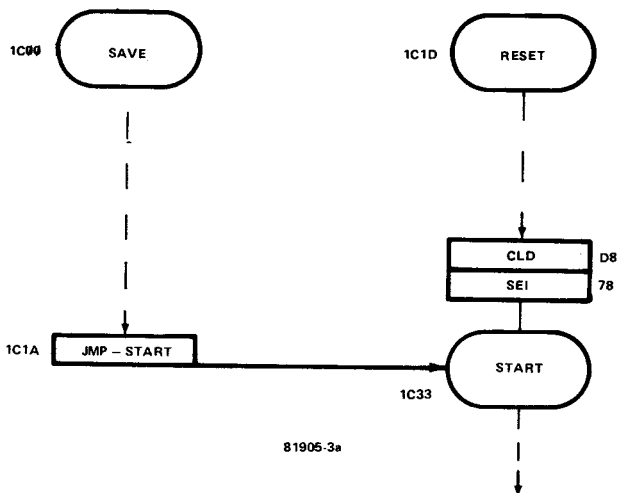
Dit moet worden gerealiseerd via herprogrammering van de EPROM. De EPROM versie D blijft uitsluitend in de ongewijzigde vorm door Elektuur leverbaar via ESS 503. Er is namelijk nog een andere oplossing mogelijk.

Oplossing twee: een stukje monitor op PIA-RAM erbij

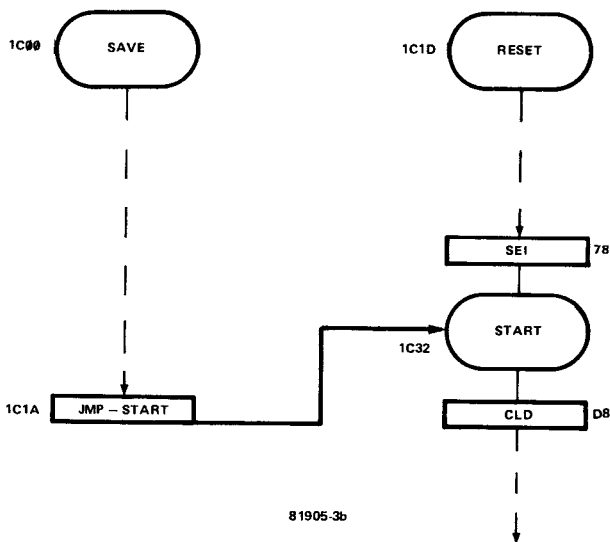
Wat moeten we doen?

- ① Monteer de opsteekprint van hoofdstuk 10 (figuren 8b, 11 en 12 van dat hoofdstuk) op de standaardkaart van de junior-computer. Praktische details staan in de tekst vermeld.
- ② Verbind de K7-aansluiting van de adresdekoder IC6 van de standaardkaart met de ene K-aansluiting van de opsteekprint en de K6-aansluiting van de adresdekoder (weerstand R15 biedt een aanknopingspunt) met de andere K-aansluiting.
Het kan ook worden gedaan volgens figuur 17 van hoofdstuk 12. Dan is ook K4 aangesloten; K6 is aangesloten via een germaniumdiode.
- ③ Toets het programma BINAR van figuur 1b in (BINAR: **BI**Nary **AR**ithmetic oftewel binair rekenen), op pagina 1A (of 1B, dat komt op hetzelfde neer). Het gaat om RAM, dus elke keer na het inschakelen van de junior-computer (en uitsluitend als we decimale rekenprogramma's willen uitvoeren) die twee instructies van figuur 1b intoetsen!
- ④ Richt de NMI-sprongvektor op het startadres van BINAR (stap-voor-stapsituatie). Wordt het programma afgesloten met een BRK, zoals in

3a

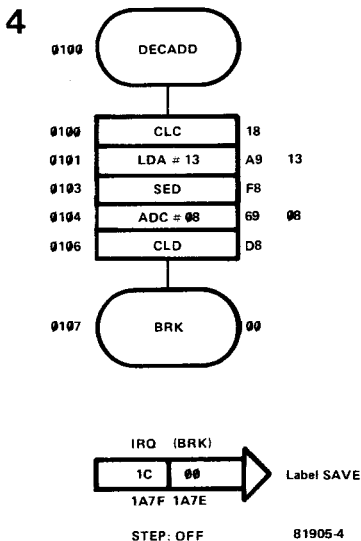


3b



het geval van figuur 1a, doe dan hetzelfde voor en met de IRQ-sprongvektor.

- ⑤ Zet de STEP-schakelaar ON, geef het startadres op en druk GO in. Het zal blijken dat nu het stap voor stap doorlopen van het programma van figuur 1a geen enkele gebruiksbepijking meer oplevert voor de monitor; alle toetsen doen, indien ingedrukt, normaal hun werk.



Overigens: Ook bij het ineens doorlopen van figuur 1a, waarbij de IRQ-sprongvektor (BRK) op 1C00 staat gericht, zou de monitor zonder de zojuist genomen maatregelen deels onbruikbaar zijn, omdat na de BRK de D-vlag nog steeds logisch één is. Bij het **ineens doorlopen van een decimaal rekenprogramma** is er natuurlijk nog een **andere oplossing**. Zorg voor de overschakeling van decimaal naar binair rekenen na de opteloperatie en vóór de BRK, dus vóór de sprong naar de monitor. Dit is in figuur 4 aangegeven. Nu kun je echter zonder maatregelen het programma niet geheel stap voor stap doorlopen.

Wat hebben we eigenlijk gedaan in de vorm van de punten ① . . . ⑤? We hebben alsnog de binaire korrektie (CLD) aan de SAVE-routine toegevoegd en we hebben het stap voor stap doorlopen van een programma in PIA-RAM geblokkeerd. Sla de betreffende tekstgedeelten van hoofdstuk 10 (inclusief figuur 9) er maar op na. Dat die opsteekprint echt nodig is bewijst het nu volgende experiment. Voeren we de punten ① en ② niet uit (de andere drie wél), dan zal bij het stap voor stap doorlopen van bijvoorbeeld figuur 1a de eerste de beste NMI ons weliswaar naar BINAR op 1A00 voeren, maar de daarbij behorende instructies worden niet uitgevoerd omdat slechts één instructie tegelijk wordt uitgevoerd en dat was de instructie van het programma die aanleiding gaf tot de NMI. Er volgt dus geen sprong naar SAVE, dus naar de monitor, en de uitstapjes en de GO-toets voor de uitvoering van de volgende instructie zijn niet mogelijk respectievelijk toegankelijk. Het display is gedoofd.

Die opsteekprint (of de gewijzigde EPROM) is hier dus gewoon nodig.

Vektoren en buskaartgeheugen

Alternatieven zonder nieuwe EPROM

Zoals in hoofdstuk 10 beschreven is het noodzakelijk dat, zodra gekozen wordt voor buskaartgeheugen, op pagina FF EPROM moet zijn aangesloten, met vektordata op de adressen FFFA . . . FFFF. Dat kan buitengewoon vervelend zijn. Want stel dat je uitsluitend geïnteresseerd bent in RAM-buskaartgeheugen, bijvoorbeeld één of meerdere 16K-dynamische RAM-kaarten (niet in dit boek 3 beschreven). Vanwege die zes domme EPROM-bytes zou je dan verplicht zijn om zo'n dure RAM/EPROM-kaart te nemen. Is dat niet een beetje al te gek en is daar nou helemaal niets aan te doen? Jazeker wel! Er zijn zelfs twee oplossingen mogelijk. Beide oplossingen zijn betrekkelijk goedkoop; in ieder geval goedkoper dan die vektor-RAM/EPROM-kaart. Beide oplossingen vereisen echter ook van de junior-bouwer een stukje zelfwerkzaamheid. Maar het prijsverschil is het waard.

De twee oplossingen zijn:

- ① Mèt aangesloten buskaartgeheugen de vektoren uit de standaard-EPROM op de standaardkaart halen;
- ② Een op de buskaart aan te sluiten **vektor-geheugenkaart**.

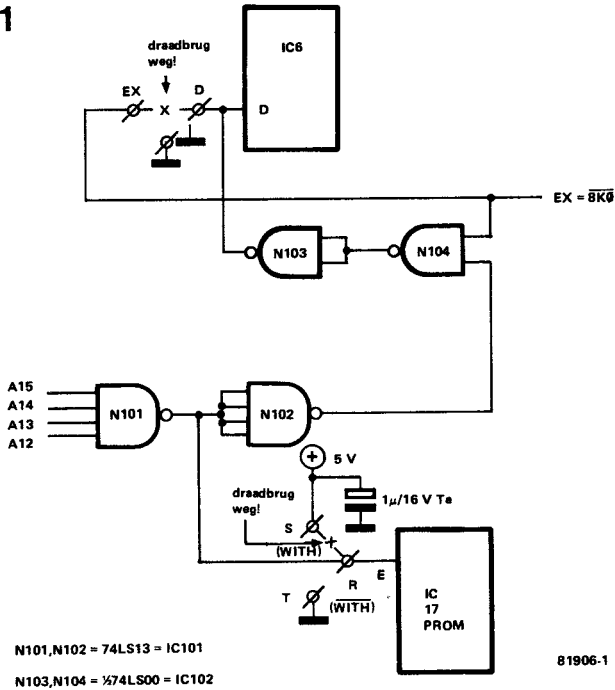
We bespreken nu de beide oplossingen.

Oplossing één: haal de vektoren uit de standaard-EPROM

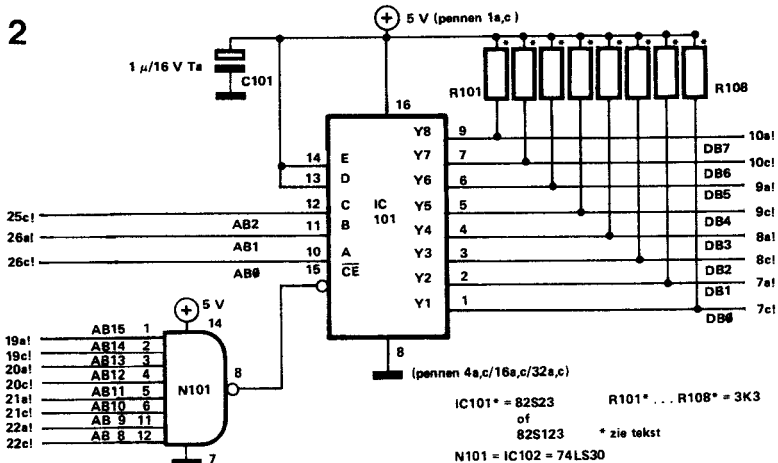
Dus doe net alsof er geen buskaartgeheugen is aangesloten

Van hoofdstuk 10 weten we dat mèt buskaartgeheugen de draadbruggen D-EX en R-S (WITH) moeten worden gelegd en dat zonder buskaartgeheugen de draadbruggen D-massa en R-T (WITH) moeten worden gelegd. In het laatste geval komen de vektoren uit de standaard-EPROM op de standaardkaart. Welnu, deze situatie wordt nu mèt buskaartgeheugen eveneens gerealiseerd. Zie figuur 1. **De draadbrug R-S moet worden verwijderd, evenals de draadbrug D-EX.** Vier poorten (twee IC's) worden toegevoegd. De uitgang van N103 gaat naar punt D, de uitgang van N101 naar punt R. **Hoe werkt het?** Zolang niet alle vier ingangen van N101 logisch één zijn is de uitgang van N101 logisch één (NAND-poort werking). In al die gevallen zou je dus net zo goed de draadbrug R-S kunnen handhaven. Zodra nu alle vier adreslijnen A12 . . . A15 logisch één zijn (dat is het geval voor alle adressen FXXX, dus óók voor de vektoradressen FFFA . . . FFFF) is de uitgang van N101 logisch nul en is er in feite sprake van de draadbrug R-T.

1



2



81906-2

Dat is voorwaarde nummer één voor het ophalen van vektordata uit de standaard-EPROM.

Nu voorwaarde twee. Indien alle vier adreslijnen A12 . . . A15 logisch één zijn moet punt D van IC6 op de standaardkaart logisch nul zijn. Dat gebeurt. En wel via de inverter N102 (N101 plus N102 is een AND-poort) en de AND-poort, gevormd door N103 plus N104. Het signaal $\overline{8K\emptyset}$ mag dan logisch één worden als een vektor wordt opgehaald, dit leidt nu niet tot een logische één, maar een nul op punt D van IC6.

De praktische realisatie van figuur 1 hangt voor een groot deel af van uw knutselgaven. De schakeling kan het beste worden opgebouwd op een stukje gaatjesprint, in de buurt van de verticale verbindingssprint tussen standaard- en interface-kaart. De plattegronden van IC101 en IC102 zijn te vinden in figuur 18 van hoofdstuk 10. Er mag gebruik worden gemaakt van de ongebufferde adreslijnen A12 . . . A15. Vergeet niet om de draadbruggen R-S en D-EX te verwijderen.

Het zal duidelijk zijn dat bij deze oplossing het adresbereik FXXX (een 4K-adresblok) niet kan worden gebruikt om er buskaartgeheugen op aan te sluiten. Immers, voor al die adressen zijn de databusbuffers op de interface-kaart geblokkeerd.

Deze oplossing doet wel een heel erg zwaar beroep op uw knutselkunde en waarschijnlijk is de nu te bespreken oplossing beter geschikt, zij het dat hij iets duurder is en dat er moet worden geprogrammeerd.

Oplossing twee: Vektor-geheugenkaart(je)

Neem een stukje gaatjesprint van 10 cm breed en 5 à 6 cm lang, monteer op een lange kant een mannetjeskonnektor (volgens figuur 19a van hoofdstuk 10), zet de schakeling van figuur 2 erop en steek de zo ontstane vektor-geheugenkaart in een daarvoor gereserveerde vrouwtjeskonnektor van de buskaart. **En let daarbij op de aansluitingen van de kaartkonnektor! Hoe werkt het?** Er is een 32-bytes-PROM genomen van hetzelfde type als is gebruikt op de interface-kaart (een mogelijke versie zonder pullup-weerstanden, met IC101 = 82S123 daargelaten). Nu worden alle acht bits gebruikt. Op zes van de 32 plaatsen wordt de nodige vektordata geprogrammeerd. Zolang de aansluiting $\overline{CE}$ (chip enable) logisch één is, zijn de data-uitgangen van de PROM geïsoleerd van de databus en oefenen daarop geen enkele invloed uit. Dat is het geval als niet alle adreslijnen die op N101 zijn aangesloten logisch één zijn. De situatie verandert radikaal als al die acht adreslijnen wel logisch één zijn. Dat is het geval als een tot pagina FF behorend adres aan bod is. Bijvoorbeeld één van de vektoradressen FFFA . . . FFFF. Nu komt data uit de PROM op de databus te staan. Tenminste: als het om een leesoperatie gaat. En dat is het geval bij het ophalen door de 6502 van een vektor.

De adresingangen D en E van de PROM liggen aan de +5 V. Er blijven drie, op de gebufferde adresbus aan te sluiten adresingangen over en dat is meer dan genoeg om een keuze te maken uit zes vektor-geheugenplaatsen. De PROM moet als volgt worden geprogrammeerd:

adres EDCBA (hex)	data (hex)
00 ... 19	00
1A	2F (NMIL)
1B	1F (NMIH)
1C	1D (RESL)
1D	1C (RESH)
1E	32 (IRQL)
1F	1F (IRQH)

Hoewel de standaard-EPROM nu niet wordt gebruikt voor het ophalen van een vektor is deze nog steeds nodig voor de RESET-routine en voor de overgang van vektor in PROM naar sprongvektor in PIA-RAM (IRQ, BRK, NMI!).

Omdat de PROM nu wordt gedekodeerd door acht adreslijnen bedraagt het verlies aan buskaartgeheugen nu slechts één pagina van FF = 256 bytes. Pagina FF maakt echter deel uit van een 4K-blok, dat niet voor RAM/EPROM-kaartgeheugen kan worden gebruikt.

Neemt men het type 82S123 voor IC101, dan kunnen de acht weerstanden R101 ... R108 vervallen. En nogmaals: let goed op de konnektorpinning!

N.B. Slechts zes van de 32 beschikbare geheugenplaatsen van de PROM worden gebruikt (adressen FFFA ... FFFF). Het is mogelijk om ook de resterende 26 geheugenplaatsen (adressen FFE0 ... FFF9) te gebruiken. Dan moet men punt D van IC101 verbinden met AB3 (pen 25a!) en punt E met AB4 (pen 24C!).

Hex dump van het programma Tape Monitor (TM)

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0800:	20	72	0C	A5	FA	8D	70	1A	A5	FB	8D	71	1A	4C	4E	08
0810:	A2	00	8E	79	1A	8E	70	1A	8E	71	1A	8E	72	1A	8E	73
0820:	1A	8E	68	1A	A9	06	8D	82	1A	A9	1E	8D	83	1A	20	36
0830:	09	D0	FB	20	36	09	F0	FB	20	36	09	F0	F6	20	F9	1D
0840:	C9	14	D0	0E	20	02	0B	A2	FF	EC	79	1A	F0	B2	E8	4C
0850:	21	08	C9	10	D0	08	20	DF	09	A2	00	4C	21	08	C9	12
0860:	D0	11	EE	68	1A	A0	09	CC	68	1A	D0	C2	A0	00	8C	68
0870:	1A	F0	BB	C9	13	D0	25	A5	E2	8D	70	1A	A5	E3	8D	71
0880:	1A	A5	E8	8D	72	1A	A5	E9	8D	73	1A	20	DF	09	20	D3
0890:	1E	A9	1E	8D	83	1A	A2	FF	9A	4C	CA	1C	C9	11	D0	03
08A0:	4C	B5	1C	06	F9	06	F9	06	F9	06	F9	05	F9	85	F9	A0
08B0:	00	CC	68	1A	D0	06	8D	79	1A	4C	2E	08	C8	CC	68	1A
08C0:	D0	06	8D	71	1A	4C	2E	08	C8	CC	68	1A	D0	06	8D	70
08D0:	1A	4C	2E	08	C8	CC	68	1A	D0	06	8D	73	1A	4C	2E	08
08E0:	C8	CC	68	1A	D0	06	8D	72	1A	4C	2E	08	C8	CC	68	1A
08F0:	D0	05	85	E3	4C	2E	08	C8	CC	68	1A	D0	05	85	E2	4C
0900:	2E	08	C8	CC	68	1A	D0	05	85	E5	4C	2E	08	C8	CC	68
0910:	1A	D0	05	85	E4	4C	2E	08	4C	10	08	B9	BB	09	8D	80
0920:	1A	8E	82	1A	98	A0	7F	88	D0	FD	A8	E8	E8	C8	E0	10
0930:	D0	E9	20	A7	1D	60	A9	7F	8D	81	1A	A2	08	A0	00	CC
0940:	68	1A	D0	09	AD	79	1A	85	F9	20	1B	09	60	C8	CC	68
0950:	1A	D0	09	AD	71	1A	85	F9	A0	04	D0	ED	C8	CC	68	1A
0960:	D0	09	AD	70	1A	85	F9	A0	08	D0	DE	C8	CC	68	1A	D0
0970:	09	AD	73	1A	85	F9	A0	0C	D0	CF	C8	CC	68	1A	D0	09
0980:	AD	72	1A	85	F9	A0	10	D0	C0	C8	CC	68	1A	D0	08	A5
0990:	E3	85	F9	A0	14	D0	B2	C8	CC	68	1A	D0	08	A5	E2	85
09A0:	F9	A0	18	D0	A4	C8	CC	68	1A	D0	08	A5	E5	85	F9	A0
09B0:	1C	D0	96	A5	E4	85	F9	A0	20	D0	8E	66	21	7F	7F	52
09C0:	08	09	7F	52	08	47	7F	06	08	09	7F	06	08	47	7F	03
09D0:	06	42	09	03	06	42	47	06	2B	21	09	06	2B	21	47	A9
09E0:	7D	8D	6C	1A	A9	C3	8D	6D	1A	A9	03	8D	76	1A	A9	02
09F0:	8D	77	1A	A9	47	A2	FF	8D	82	1A	8D	78	1A	8E	83	1A

0A00: A9 00 A2 7F 8D 80 1A 8E 81 1A 8D 6E 1A 8D 6F 1A
 0A10: AD 70 1A 85 FA AD 71 1A 85 FB A2 FF 8E 74 1A A9
 0A20: 16 20 A3 0A CE 74 1A D0 F6 A9 2A 20 A3 0A AD 79
 0A30: 1A 20 8B 0A AD 70 1A 20 7A 0A AD 71 1A 20 7A 0A
 0A40: A5 FB CD 73 1A D0 23 A5 FA CD 72 1A D0 1C A9 2F
 0A50: 20 A3 0A AD 6E 1A 20 8B 0A AD 6F 1A 20 8B 0A A9
 0A60: 04 20 A3 0A A9 04 20 A3 0A 60 A0 00 B1 FA 20 7A
 0A70: 0A E6 FA D0 CB E6 FB 4C 40 0A A8 18 6D 6E 1A 8D
 0A80: 6E 1A AD 6F 1A 69 00 8D 6F 1A 98 A8 4A 4A 4A 4A
 0A90: 20 9A 0A 98 29 0F 20 9A 0A 60 C9 0A 18 30 02 69
 0AA0: 07 69 30 A2 08 8E 75 1A 4A 48 90 0C 20 C8 0A 20
 0AB0: E5 0A 20 E5 0A 4C C1 0A 20 C8 0A 20 C8 0A 20 E5
 0AC0: 0A 68 CE 75 1A D0 E1 60 AE 76 1A 2C D5 1A 10 FB
 0AD0: AD 6C 1A 8D F4 1A AD 78 1A 49 80 8D 82 1A 8D 78
 0AE0: 1A CA D0 E7 60 AE 77 1A 2C D5 1A 10 FB AD 6D 1A
 0AF0: 8D F4 1A AD 78 1A 49 80 8D 82 1A 8D 78 1A CA D0
 0B00: E7 60 A9 32 8D 82 1A A9 00 8D 6E 1A 8D 6F 1A A9 FF 8D 6B
 0B10: 7F 8D 81 1A A9 00 8D 6E 1A 8D 6F 1A A9 FF 8D 6B
 0B20: 1A 20 C2 0B 6E 6B 1A AD 6B 1A 20 E8 0B C9 16 D0
 0B30: F0 A0 0A 8C 69 1A 20 36 0C 20 5D 0C C9 16 D0 DC
 0B40: CE 69 1A D0 F1 20 36 0C 20 5D 0C C9 2A F0 06 C9
 0B50: 16 F0 F2 D0 AD 20 5D 0C 20 F3 0B CD 79 1A D0 3E
 0B60: 20 F3 0B 20 4B 0C 85 FA 20 F3 0B 20 4B 0C 85 FB
 0B70: 20 F3 0B 30 8D F0 13 20 4B 0C A0 00 91 FA E6 FA
 0B80: D0 02 E6 FB 20 64 0C 4C 70 0B 20 F3 0B CD 6E 1A
 0B90: D0 09 20 F3 0B CD 6F 1A D0 01 60 4C 02 0B AD 79
 0BA0: 1A C9 00 F0 BB C9 FF D0 F2 20 F3 0B 20 4B 0C 20
 0BB0: F3 0B 20 4B 0C AD 70 1A 85 FA AD 71 1A 85 FB 4C
 0BC0: 70 0B 2C 82 1A 10 FB AD D4 1A A0 FF 8C F6 1A A0
 0BD0: 14 88 D0 FD 2C 82 1A 30 FB 38 ED D4 1A A0 FF 8C
 0BE0: F6 1A A0 07 88 D0 FD 60 48 A9 36 8D 80 1A 68 20
 0BF0: 64 0C 60 20 36 0C C9 2F D0 01 60 20 19 0C 30 FA
 0C00: 0A 0A 0A 0A 8D 6A 1A 20 36 0C C9 2F F0 EC 20 19
 0C10: 0C 30 E7 0D 6A 1A A0 01 60 C9 30 30 0C C9 3A 30
 0C20: 0B C9 14 30 04 C9 47 30 03 A0 FF 60 C9 40 30 03
 0C30: 18 69 09 29 0F 60 A2 08 20 C2 0B 6E 6B 1A CA D0
 0C40: F7 2E 6B 1A 4E 6B 1A AD 6B 1A 60 48 18 6D 6E 1A
 0C50: 8D 6E 1A AD 6F 1A 69 00 8D 6F 1A 68 60 48 7F
 0C60: 8D 80 1A 68 48 AD 78 1A 49 02 8D 82 1A 8D 78 1A
 0C70: 68 60 38 A5 FA E9 01 85 FA A5 FB E9 00 85 FB 60

Hex dump van het programma Printer Monitor (PM)

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
1000:	D8	78	A9	67	8D	82	1A	A9	00	8D	80	1A	A2	FE	8E	5A
1010:	1A	E8	8E	5B	1A	9A	86	F2	A9	7F	8D	81	1A	8D	83	1A
1020:	A2	02	8E	59	1A	A9	5F	8D	7C	1A	A9	10	8D	7D	1A	A9
1030:	CF	8D	7A	1A	A9	14	8D	7B	1A	2C	80	1A	30	FB	20	E0
1040:	12	4E	5F	1A	6E	5E	1A	AD	5E	1A	8D	5C	1A	AD	5F	1A
1050:	8D	5D	1A	A2	08	20	03	13	20	BE	12	C9	7F	D0	A1	20
1060:	46	12	20	E8	11	A2	FF	9A	86	F2	20	59	12	20	68	12
1070:	20	AE	12	C9	2B	D0	09	20	13	12	20	F8	11	4C	6A	10
1080:	C9	2D	D0	09	20	1A	12	20	F8	11	4C	6A	10	C9	20	D0
1090:	0E	A5	F8	85	FA	A5	F9	85	FB	20	F8	11	4C	6A	10	C9
10A0:	2E	D0	0F	A5	F8	A0	00	91	FA	20	13	12	20	F8	11	4C
10B0:	6A	10	C9	52	D0	13	A6	F2	9A	A5	FB	48	A5	FA	48	A5
10C0:	F1	48	A6	F5	A4	F4	A5	F3	40	C9	4C	D0	52	A0	14	20
10D0:	D6	11	A5	F3	20	8F	12	A0	1A	20	D6	11	A5	F4	20	8F
10E0:	12	A0	20	20	D6	11	A5	F5	20	8F	12	A0	26	20	D6	11
10F0:	A5	F0	20	8F	12	A5	EF	20	8F	12	A0	2C	20	D6	11	A9
1100:	01	20	8F	12	A5	F2	20	8F	12	A0	32	20	D6	11	20	28
1110:	12	A0	38	20	D6	11	20	F3	11	4C	6A	10	4C	B0	11	C9
1120:	50	D0	0E	A5	EF	85	FA	A5	F0	85	FB	20	F8	11	4C	6A
1130:	10	C9	4D	D0	E7	A0	52	20	D6	11	20	87	13	10	03	4C
1140:	5F	10	38	AD	65	1A	ED	63	1A	AD	66	1A	ED	64	1A	90
1150:	EE	20	E8	11	A2	06	20	F3	11	CA	D0	FA	A0	00	98	20
1160:	9B	12	20	F3	11	20	F3	11	C8	C0	10	D0	F1	AD	63	1A
1170:	85	FA	AD	64	1A	85	FB	20	E8	11	A2	10	A5	FB	20	8F
1180:	12	A5	FA	20	8F	12	A0	17	20	D9	11	38	AD	65	1A	E5
1190:	FA	AD	66	1A	E5	FB	B0	06	20	E8	11	4C	5F	10	A0	00
11A0:	B1	FA	20	8F	12	20	F3	11	20	13	12	CA	D0	DD	F0	C7
11B0:	C9	47	D0	0B	20	8A	14	30	03	4C	6A	10	4C	5F	10	C9
11C0:	53	D0	08	20	3B	14	30	F4	4C	6A	10	20	6F	12	D0	03
11D0:	4C	70	10	4C	6A	10	20	E8	11	B9	BD	13	C9	03	F0	07
11E0:	20	34	13	C8	4C	D9	11	60	A9	0D	20	34	13	A9	0A	20
11F0:	34	13	60	A9	20	4C	EF	11	20	E8	11	A5	FB	20	8F	12
1200:	A5	FA	20	8F	12	20	F3	11	A0	00	B1	FA	20	8F	12	20
1210:	F3	11	60	E6	FA	D0	02	E6	FB	60	38	A5	FA	E9	01	85
1220:	FA	A5	FB	E9	00	85	FB	60	A5	F1	8D	67	1A	A2	08	0E
1230:	67	1A	90	09	A9	01	20	9B	12	CA	D0	F3	60	A9	00	20

1240: 9B 12 CA D0 EA 60 A0 00 20 D6 11 20 E8 11 60 A0
1250: 07 4C 48 12 A0 0E 4C 48 12 A0 00 8C 63 1A 8C 64
1260: 1A 8C 65 1A 8C 66 1A 60 A0 00 84 F8 84 F9 60 20
1270: 1E 14 30 10 A2 04 06 F8 26 F9 CA D0 F9 05 F8 85
1280: F8 A0 00 60 A0 46 20 D6 11 20 E8 11 A0 FF 60 48
1290: 4A 4A 4A 4A 20 A4 12 20 34 13 68 29 0F 20 A4 12
12A0: 20 34 13 60 C9 0A 18 30 02 69 07 69 30 60 2C 80
12B0: 1A 30 FB 8E 61 1A A2 08 20 03 13 20 12 13 2C 80
12C0: 1A 10 0A 38 6E 62 1A CA D0 F1 4C D4 12 18 6E 62
12D0: 1A CA D0 E7 20 12 13 AD 62 1A 29 7F AE 61 1A 60
12E0: 18 AD 5A 1A 69 01 8D 5A 1A AD 5B 1A 69 00 8D 5B
12F0: 1A 2C 80 1A 10 EA AD 5A 1A 8D 5E 1A AD 5B 1A 8D
1300: 5F 1A 60 AD 5C 1A 8D 5E 1A AD 5D 1A 8D 5F 1A 4C
1310: 1E 13 AD 5A 1A 8D 5E 1A AD 5B 1A 8D 5F 1A 38 AD
1320: 5E 1A E9 01 8D 5E 1A AD 5F 1A E9 00 8D 5F 1A EA
1330: EA B0 EB 60 8E 60 1A 8D 62 1A AD 82 1A 29 FE 8D
1340: 82 1A 20 12 13 A2 07 4E 62 1A 90 30 AD 82 1A 09
1350: 01 8D 82 1A 20 12 13 CA D0 ED AE 59 1A AD 82 1A
1360: 09 01 8D 82 1A 20 12 13 CA D0 F2 2C 80 1A 10 04
1370: AE 60 1A 60 2C 80 1A 10 FB 6C 7C 1A AD 82 1A 29
1380: FE 8D 82 1A 4C 54 13 20 AE 12 C9 2C F0 07 20 6F
1390: 12 30 29 F0 F2 A5 F8 8D 63 1A A5 F9 8D 64 1A 20
13A0: 68 12 20 AE 12 C9 0D F0 07 20 6F 12 30 0E F0 F2
13B0: A5 F9 8D 66 1A A5 F8 8D 65 1A A0 00 60 4A 55 4E
13C0: 49 4F 52 03 45 44 49 54 4F 52 03 41 53 53 45 4D
13D0: 03 41 43 43 3A 20 03 59 20 20 3A 20 03 58 20 20
13E0: 3A 20 03 50 43 20 3A 20 03 53 50 20 3A 20 03 50
13F0: 52 20 3A 20 03 20 20 20 20 20 4E 56 20 42 44 49
1400: 5A 43 03 57 48 41 54 3F 03 52 45 41 44 59 03 48
1410: 45 58 44 55 4D 50 3A 20 03 53 41 3A 20 03 C9 30
1420: 30 0C C9 3A 30 0B C9 41 30 04 C9 47 30 03 A0 FF
1430: 60 C9 40 30 03 18 69 09 29 0F 60 20 AE 12 C9 2C
1440: F0 07 20 6F 12 30 3F F0 F2 A5 F8 8D 79 1A C9 00
1450: F0 35 C9 FF F0 31 20 68 12 20 87 13 30 28 AD 63
1460: 1A 8D 70 1A AD 64 1A 8D 71 1A AD 65 1A 8D 72 1A
1470: AD 66 1A 8D 73 1A 20 DF 09 20 BC 14 A0 4C 20 D6
1480: 11 20 E8 11 A0 00 60 A0 FF 60 20 A2 13 30 17 8D
1490: 79 1A C9 FF F0 11 20 02 0B 20 BC 14 A0 4C 20 D6
14A0: 11 20 E8 11 A0 00 60 A0 5C 20 D6 11 20 EB 14 30
14B0: F5 8D 70 1A A5 F9 8D 71 1A 4C 96 14 A9 67 8D 82
14C0: 1A A9 00 8D 80 1A A9 7F 8D 81 1A 8D 83 1A 60 85
14D0: F3 68 85 F1 68 85 EF 85 FA 68 85 F0 85 FB 84 F4
14E0: 86 F5 BA 86 F2 20 F8 11 4C 6A 10 A9 00 85 F8 85
14F0: F9 4C A2 13

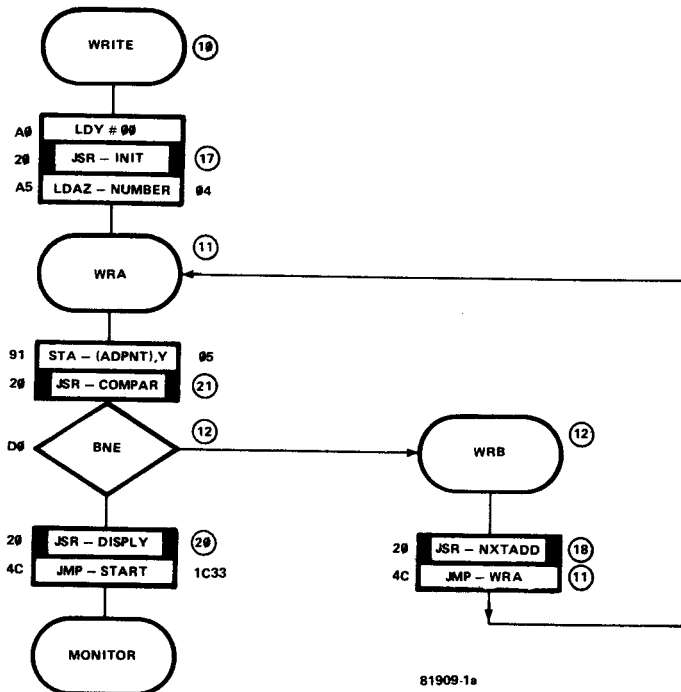
RAM-testprogramma

Schrijf- en leestest

Bij de uitbreiding van de junior-computer hoort uitbreiding van de hoeveelheid RAM-geheugen; hetzij op de interface-kaart, hetzij eveneens als buskaartgeheugen, op een of meerdere RAM/EPROM-kaarten en/of 16K-dynamische RAM-kaarten. Het is buitengewoon belangrijk dat alle RAM-geheugenplaatsen (minstens 1024 stuks; met buskaart-RAM een veelvoud daarvan) korrekt kunnen worden beschreven en gelezen.

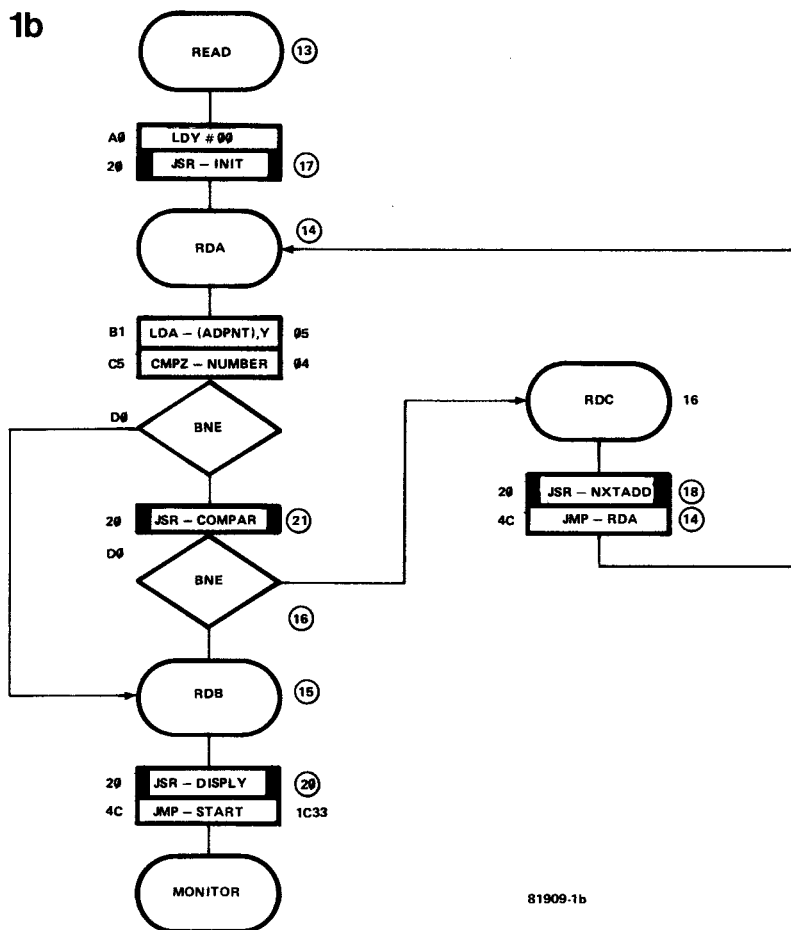
Vandaar het testprogramma van figuur 1. Het moet via de editor en de assembler worden ingetoetst in standaard-RAM (pagina's 02 . . . 03 of een deel van pagina 00). Men is binnen zekere grenzen vrij in de keuze van de

1a



81909-1a

1b

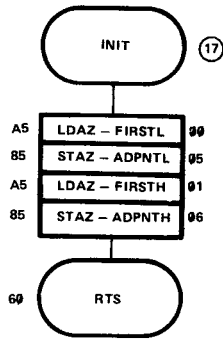


81909-1b

startadressen, vandaar de labels in figuur 1. De keuze voor standaard-RAM voor de huisvesting van het testprogramma betekent dat het standaard-gedeelte van de uitgebreide junior-computer in orde moet zijn.

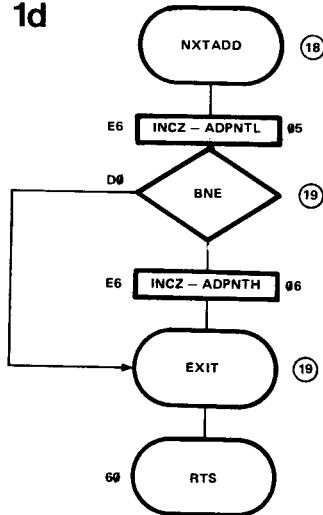
Het gaat om een schrijfroutine (figuur 1a), een leesroutine (1b) en vier sub-routines (1c . . . 1f). Tijdens het schrijven worden alle geheugenplaatsen, van adres FIRST tot en met LAST, geladen met data die is gespecificeerd in de geheugenplaats NUMBER. Nadat alle adressen van het testblok zijn beschreven verschijnt het adres LAST in beeld, met de data van NUMBER. Met het terugleesprogramma READ van figuur 1b worden alle geheugenplaatsen van het testblok gelezen. Nou ja, alle . . . dat hangt er maar van af. De gelezen data wordt vergeleken met de geschreven data (inhoud NUMBER), Zodra er een verschil is gekonstateerd springt men terug naar de monitor. Het adres met de foute data verschijnt in beeld.

1c



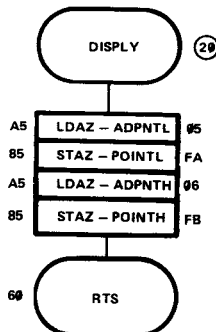
81909-1c

1d



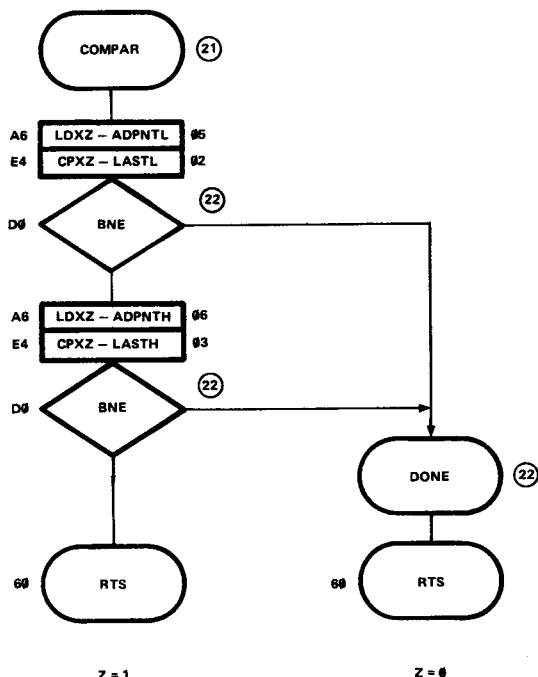
81909-1d

1e



81909-1e

1f



Gebruiksaanwijzing

- ① Toets de programma's van figuur 1 in op standaard-RAM, met behulp van de editor en de assembler. Noteer na afloop hiervan het startadres van WRITE en van READ.
- ② De volgende geheugenplaatsen van pagina nul worden gebruikt:

0000	FIRSTL	eerste adres testblok
0001	FIRSTH	eerste adres testblok
0002	LASTL	laatste adres testblok
0003	LASTH	laatste adres testblok
0004	NUMBER	testdata
0005	ADPNTL	schrijf/leesadreswijzer
0006	ADPNTH	schrijf/leesadreswijzer

Geef FIRST en LAST op aan de hand van uw keuze van een testblok. Neem liever geen testdata 00 of FF.
- ③ Start het schrijfprogramma WRITE. Na afloop is het laatste adres te zien, met de testdata.
- ④ Start het terugleesprogramma READ. Indien na afloop het adres LAST is te zien is alle RAM van het testblok in orde. Verschijnt er een adres in beeld dat lager is dan LAST (mèt foute data; dat wil zeggen: van NUMBER afwijkend), dan is het gedeelte tot aan dat adres in orde. Noteer het foute adres.
- ⑤ Indien alles in orde is, neem dan voor de zekerheid andere testdata

(inhoud NUMBER) en herhaal de schrijf/leestest voor dat testblok. Je zal zien dat een RAM-geheugenplaats die niet in orde is vóór het schrijven een inhoud heeft die gelijk is aan NUMBER; wie zegt ons dan dat die geheugenplaats goed kan worden beschreven? (Na het inschakelen van de netspanning neemt RAM-geheugen willekeurige data aan).

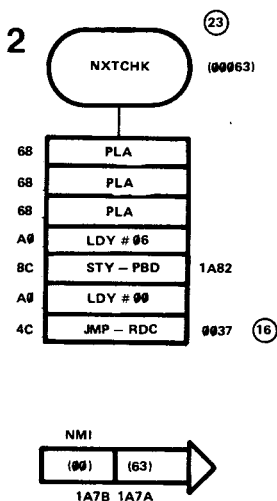
- ⑥ Indien men op een foute geheugenplaats is gestoten kan men nagaan of er soms nog meer van die foute gevallen zijn. Maak daartoe LAST gelijk aan een adres dat één hoger is dan het "foute" adres en start het programma READ opnieuw. Men kan ⑥ zo vaak herhalen als nodig blijkt.

Even een gewetensvraag. Misschien vindt u plotseling achter elkaar vele foute adressen. Is daar wel RAM op aangesloten? Gaat het soms om EPROM? Of PIA? Of VIA? Is er dus sprake van nodeloze paniek? Dit RAM-testprogramma geeft alleen uitsluitsel over adresseerbare zaken die beschreven en gelezen kunnen worden.

- ⑦ Gaat het wel degelijk om niet (geheel) in orde zijnde aangesloten RAM, dan moet de betreffende hardware nog eens grondig worden gecontroleerd. Kijk vooral de adresselektie/dekodering goed na; zijn de te kiezen draadbruggen wel (goed) gelegd?

Zorg er voor dat bepaalde delen van de pagina's 00 en 01 geen deel uitmaken van een testblok! Het is jammer als het testprogramma in de soep loopt, de stapel wordt overschreven en allerlei bewaarplaatsen in pagina nul worden opgescheept met gekke data.

Maak gebruik van de toets ST. Nadat men een fout heeft gevonden zoekt men er nog meer op (punt ⑥). In plaats van de aanpassing van FIRST en het opgeven van het startadres van READ kan men gebruik maken van de toets ST, in combinatie met de interrupt-routine van figuur 2. Na het indrukken van ST gaat het leesprogramma verder vanaf het label RDC: de voorbereiding voor het lezen van het volgende adres. Bij elk fout adres wordt gestopt. Noteer het adres en druk ST in, enzovoorts. Het enige nadeel van deze aanpak is dat men voorbij het laatste adres LAST van het



81909-2

Tabel 1	WRITE	0007	A0 00	LDY # 00
		0009	20 3D 00	JSR-INIT
		000C	A5 04	LDAZ-NUMBER
	WRA	000E	91 05	STA-(ADPNT),Y
		0010	20 56 00	JSR-COMPAR
		0013	D0 06	BNE WRB
		0015	20 4D 00	JSR-DISPLY
		0018	4C 33 1C	JMP-START
	WRB	001B	20 46 00	JSR-NXTADD
		001E	4C 0E 00	JMP-WRA
	READ	0021	A0 00	LDY # 00
		0023	20 3D 00	JSR-INIT
	RDA	0026	B1 05	LDA-(ADPNT),Y
		0028	C5 04	CMPZ-NUMBER
		002A	D0 05	BNE RDB
		002C	20 56 00	JSR-COMPAR
		002F	D0 06	BNE RDC
	RDB	0031	20 4D 00	JSR-DISPLY
		0034	4C 33 1C	JMP-START
	RDC	0037	20 46 00	JSR-NXTADD
		003A	4C 26 00	JMP-RDA
	INIT	003D	A6 00	LDAZ-FIRSTL
		003F	85 05	STAZ-ADPNTL
		0041	A5 01	LDAZ-FIRSTH
		0043	85 06	STAZ-ADPNTH
		0045	60	RTS
	NXTADD	0046	E6 05	INCZ-ADPNTL
		0048	D0 02	BNE EXIT
		004A	E6 06	INCZ-ADPNTH
	EXIT	004C	60	RTS
	DISPLY	004D	A5 05	LDAZ-ADPNTL
		004F	85 FA	STAZ-POINTL
		0051	A5 06	LDAZ-ADPNTH
		0053	85 FB	STAZ-POINTH
		0055	60	RTS
	COMPAR	0056	A6 05	LDXZ-ADPNTL
		0058	E4 02	CPXZ-LASTL
		005A	D0 06	BNE DONE
		005C	A6 06	LDXZ-ADPNTH
		005E	E4 03	CPXZ-LASTH
		0060	D0 00	BNE DONE
	DONE	0062	60	RTS
	NXTCHK	0063	68	PLA
		0064	68	PLA
		0065	68	PLA
		0066	A0 06	LDY # 06
		0068	8C 82 1A	STY-PBD
		006B	A0 00	LDY # 00
		006D	4C 37 00	JMP-RDC

testblok kan komen door maar vaak genoeg ST in te drukken. En dat kan allemaal maar weer nodeloze paniek veroorzaken. Dus: ophouden zodra LAST aan bod is geweest!

Het kan ook anders

De werking van RAM kan eveneens worden gecontroleerd met behulp van de monitor. In de datamode wordt telkens na het indrukken van een numerieke toets de instructie STA-(POINTL),Y uitgevoerd. En tijdens de periodieke sturing van de displays (multiplexing) wordt regelmatig de instructie LDA-(POINTL),Y uitgevoerd. Dus schrijven en lezen.

Men kan nu het volgende doen:

AD X X X X XXXX YY

DA ZZ XXXX ZZ?

Hierbij is XXXX een adres dat hoort tot het testblok. De data ZZ moet zodanig worden gekozen dat deze totaal afwijkt van YY. Is na het intoetsen van de data geen ZZ te zien dan is er iets fout (de eerder besproken misverstanden buiten beschouwing gelaten). Men kan als men zin heeft op deze manier een heel testblok afwerken: telkens de plustoets indrukken en afwijkende data opgeven.

P.S. Een geassembleerde versie van het testprogramma volgens de figuren 1 en 2 (dus inclusief de ST-toetsroutine NXTCHK) is opgegeven in tabel 1. Een en ander is op pagina nul gelokaliseerd, aansluitend op de bewaarplaatsen 0000 . . . 0006. Het startadres van WRITE is 0007, dat van READ 0021. De ST-routine start op 0063; vergeet niet om de NMI-sprongvektor hierop te richten! En verder moet het adres FIRST hoger dan 006F zijn!

ASCII-kodetabel

karakter	ASCII-kode (hexadecimaal)	karakter	ASCII-kode (hexadecimaal)
Ø	30	V	56
1	31	W	57
2	32	X	58
3	33	Y	59
4	34	Z	5A
5	35	spatie	20
6	36	.	2E
7	37	(	28
8	38	+	2B
9	39	i	7C
A	41	&	26
B	42	!	21
C	43	\$	24
D	44	*	2A
E	45	)	29
F	46	;	3B
G	47	— of ~	7E
H	48	—	2D
I	49	/	2F
J	4A	,	2C
K	4B	%	25
L	4C	— of ←	5F
M	4D	>	3E
N	4E	?	3F
O	4F	:	3A
P	50	#	23
Q	51	@	40
R	52	,	27
S	53	=	3D
T	54	“	22
U	55	<	3C

