

**MASTER**

**De koppeling van een PDP 11/20 computer aan het Burroughs B6700 systeem van de THE**

Woensdregt, D.M.P.

*Award date:*  
1973

[Link to publication](#)

**Disclaimer**

This document contains a student thesis (bachelor's or master's), as authored by a student at Eindhoven University of Technology. Student theses are made available in the TU/e repository upon obtaining the required degree. The grade received is not published on the document as presented in the repository. The required complexity or quality of research of student theses may vary by program, and the required minimum study period may vary in duration.

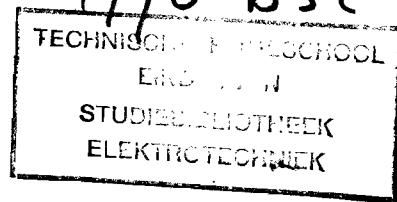
**General rights**

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain

**Take down policy**

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.



AFDELING DE ELEKTROTECHNIEK  
TECHNISCHE HOOGESCHOOL  
EINDHOVEN  
Groep Meten en Regelen

DE KOPPELING VAN EEN PDP 11/20  
COMPUTER AAN HLT BURROUGHS B6700  
SYSTEEM VAN DE THE.

door D.M.P. Woensdregt

Rapport van het afstudeerwerk  
uitgevoerd van april 1972 t/m juni 1973  
in opdracht van prof. ir. F.J. Kylstra  
onder leiding van ir. J.J.H. van Nunen

INHOUDSOPGAVE

	pag.
<u>Lijst van figuren</u> . . . . .	3
<u>Samenvatting</u> . . . . .	4
1. <u>Inleiding</u> . . . . .	5
2. <u>De gekoppelde computersystemen</u> . . . . .	7
2.1. De PDP 11/20 computer	7
2.1.1. De Unibus	7
2.1.2. De processor	8
2.1.3. De master-slave relatie	10
2.1.4. Verandering van busmaster	10
2.2. Het Burroughs B6700 datakommunikatiesysteem	13
2.2.1. De hardwarestructuur	13
2.2.2. De softwarestructuur	14
3. <u>De verbinding</u> . . . . .	18
3.1. De lijn	18
3.2. De modems	19
3.3. De lijndiscipline	20
3.3.1. De masterstatus	21
3.3.2. Het datatransport	22
3.3.3. De tekstblokken	24
3.3.4. Mogelijke fouten	26
4. <u>Het PDP 11/20 interface</u> . . . . .	28
4.1. De taken	28
4.2. De opzet	28
4.3. Het datatransport	31
4.3.1. De adresselektor	31
4.3.2. De transceivers	32
4.3.3. Dubbel bufferen	33
4.3.4. De multiplexor	34
4.3.5. Pariteitsbit-generatie	35
4.3.6. De schuifregisters	35
4.3.7. Start en einde van het zenden	37
4.3.8. De karaktersynchronisatie	38
4.3.9. De distributor	38
4.3.10. De foutendetektie	39
4.4. De statusregisters	40
4.4.1. De indeling	41
4.4.2. De interruptbehandeling	45
4.5. De praktische uitvoering	48
5. <u>De interface driver</u> . . . . .	50
6. <u>Mogelijke uitbreiding van de hardware</u> . . . . .	52
<u>Literatuurlijst</u> . . . . .	56

LIJST VAN FIGUREN

	pag.
1. De computersystemen van de groep ER	5
2. Schematische voorstelling van de UNIBUS	7
3. Bezetting van de Unibusadressen	9
4. Indeling van een enkel-operand instructie	10
5. Gang van zaken bij interrupt	12
6. Schema van het Burroughs B6700 systeem	13
7. Softwarestructuur van het B6700 datakommunikatiesysteem	15
8. Signalen tussen interface en modem volgens CCITT, aanbeveling V.24	19
9. Procedure voor het verkrijgen van het beheer over de lyn	22
10. Procedure voor het datatransport	23
11. Het blokschema van het PDP 11/20 interface	30
12. Het blokschema van de adresselektor	32
13. Het schema van een transceiver	33
14. De sturing voor het dubbel bufferen	33
15. Het schema van de multiplexor	34
16. De schakeling voor de pariteitsbit-generatie	35
17. Het schema van het schuifregister voor zenden	36
18. Het schema van het schuifregister voor ontvangen	36
19. De schakeling voor het starten en beëindigen van het zenden	37
20. De schakeling voor karaktersynchronisatie	38
21. Het schema van de distributor	39
22. De indeling van het statusregister zenden	41
23. De indeling van het statusregister ontvangen	43
24. De schakeling voor het in volgorde doorgeven van de interruptaanvragen	46
25. Het blokschema van de interruptmodule	47
26. Het bepalen van het interruptvektor-adres	48

### SAMENVATTING

Het rapport beschrijft de koppeling van de PDP 11/20 computer van de groep ER aan het Burroughs B6700 systeem van de THE. Het real-time hybride rekenwerk, dat op de PDP 11 / Hitachi 505 combinatie verricht zal worden, vraagt een grote rekenkapaciteit en een groot achtergrondgeheugen voor schattingsalgoritmen. Deze behoefte maakte een koppeling met het grote B6700 systeem gewenst.

Voor de verbinding is een interface nodig, waardoor de PDP 11 aan de telefoonlijn, in casu de modem gekoppeld kan worden. Verder moeten regels vastgelegd worden, volgens welke de konversatie tussen beide computersystemen zal plaatsvinden. Hiervoor is een pakket PDP 11 software ontwikkeld.

In het rapport wordt een beschrijving gegeven van de gekoppelde computersystemen. De regels voor de konversatie worden uiteengezet. Verder geeft het rapport een overzicht van de ontwikkelde hardware en software. Er wordt aangegeven, hoe een aantal functies van de software in de toekomst door hardware kan worden overgenomen.

## 1. INLEIDING

De groep ER van de Technische Hogeschool Eindhoven beschikt momenteel over de volgende computersystemen:

- Een PDP 11/20 digitale rekenmachine met 16 k kernegeugen (16 bits woorden), een dual DECTape en verder een Tektronix storage tube display (4010) en een ASR 33 teletype.
- Een PDP 8i digitale computer met 8 k kernegeugen (12 bits woorden), een disk (256 k woorden), een snelle ponsbandlezer/ponser, een ASR 33 teletype en verder A/D en D/A converters voor interactie met analoge apparatuur.
- Een Hitachi analoge rekenmachine met controlepaneel en 2 consoles, waarop 108 versterkers, 36 integratoren, 12 vermenigvuldigers, 16 comparatoren, 36 servopotentiometers en een aantal normale potentiometers. Als conversie-apparatuur zijn aanwezig een 16-kanaals ADC en DAC van Adage.

Deze systemen zijn via de Unibus van de PDP 11/20 met elkaar verbonden (zie fig. 1), zodat elk systeem de randapparatuur van de andere kan gebruiken.

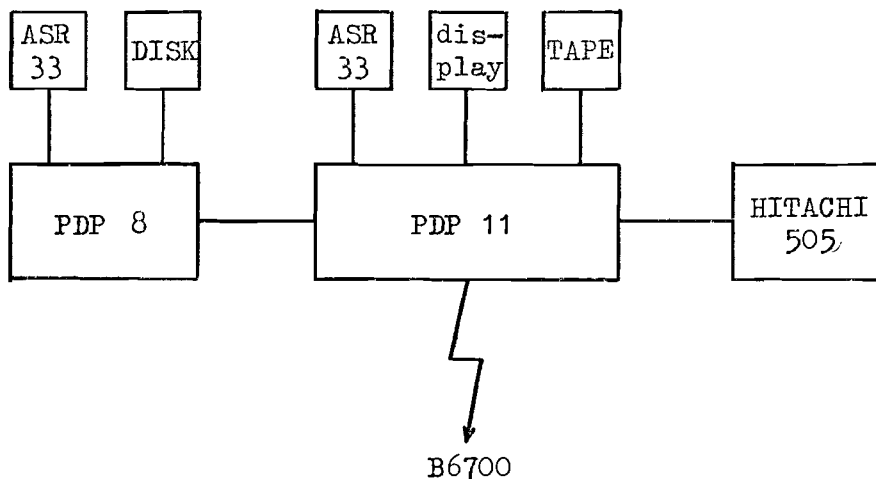


Fig. 1: De computersystemen van de groep ER.

De in de groep meest voorkomende rekenwerkzaamheden zijn parameterschatting t.b.v. modelbouw voor medisch en ergonomisch onderzoek en simulatie van computersystemen. De laatste worden vooral op de PDP 8 uitgevoerd. De parameterschatting gebeurt meestal aan modellen, die op de analoge computer zijn geïmplementeerd. Deze programma's eisen een real-time omgeving voor een juiste interactie met het analoge model. Bovendien vergen de schattingsalgoritmen een grote hoeveelheid ruimte en rekentijd. Door het laatste ontstond de behoefte aan een grotere rekenkapaciteit en een groter achtergrondgeheugen. In deze behoefte kan worden voorzien door het Burroughs B6700 systeem, dat in het rekencentrum van de Technische Hogeschool Eindhoven staat opgesteld. Eén van de faciliteiten van dit systeem is een Remote Job Entry, waarin een kaartlezer, een regeldrukker en een console zijn ondergebracht. Vanuit deze satelliet heeft de gebruiker in principe een gemakkelijke toegang tot het gehele systeem. Door de software voor dit RJE-systeem enigszins uit te bouwen, kon een geschikte mogelijkheid gecreëerd worden voor het koppelen met een satellietcomputer, in dit geval een PDP 11/20. De gebruiker van de PDP 11 krijgt door de koppeling tevens de mogelijkheid, gebruik te maken van het time-sharing programma "Command and Edit" (CANDE) van het B6700 systeem. Cande biedt dezelfde mogelijkheden als het reeds aanwezige Philips P9200 systeem.

Door de koppeling van de PDP 11/20 aan het B6700 systeem en de toekomstige uitbreiding van het PDP 11 softwarepakket door de aankoop van een real-time operating system (RSX), staan voor de gebruiker van het computersysteem van de groep ER grote mogelijkheden open.

## 2. DE GEKOPPELDE COMPUTERSYSTEMEN

### 2.1. DE PDP 11/20 COMPUTER

De PDP 11/20 computer van Digital Equipment Corporation (DEC) (lit. 1 en 2) heeft de mogelijkheid tot het adresseren van zowel 16-bits woorden, als 8-bits bytes. Het meest opvallend aan deze computer is het zgn. UNIBUS-koncept, waardoor randapparaten op dezelfde wijze behandeld kunnen worden als het kerngeheugen. Daarom zijn er geen speciale in- en uitvoer instructies nodig, zodat de instructieset anders is dan in de meeste computers. Verder beschikt de PDP 11/20 over 8 general purpose registers, een hardware stack en een hardware prioriteitenstructuur voor interrupts.

#### 2.1.1. De Unibus

In de PDP 11/20 zijn alle systeemcomponenten (processor, geheugen, randapparatuur) verbonden met hetzelfde snelle datapad (zie fig. 2), genaamd UNIBUS.

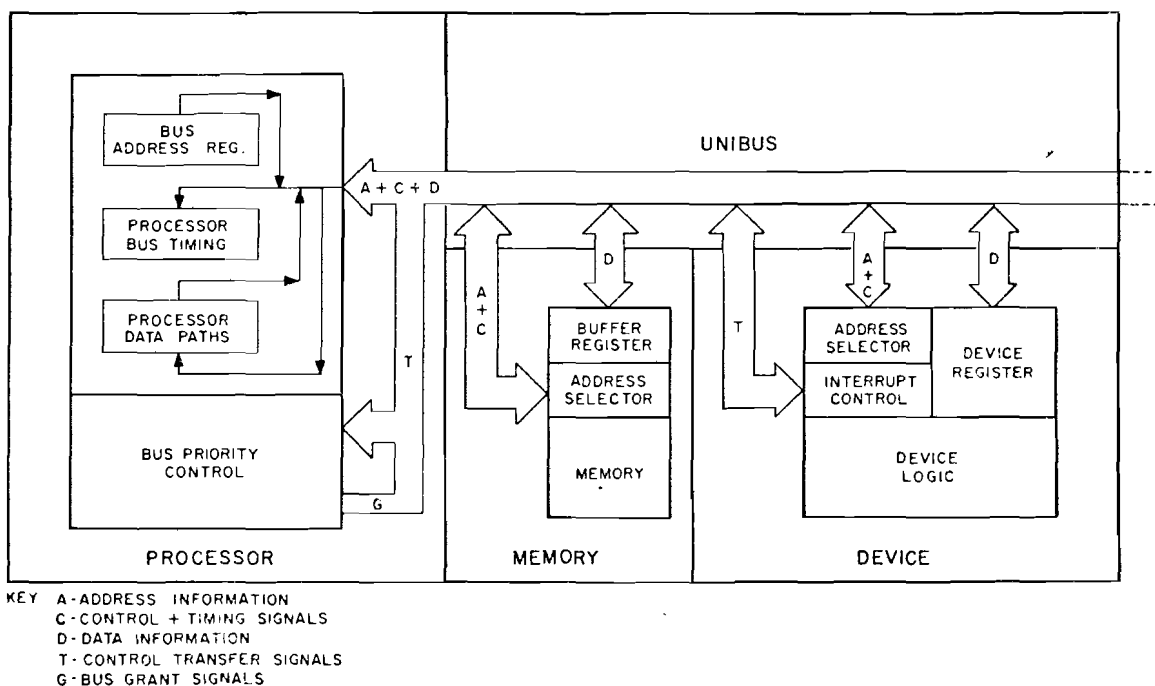


Fig. 2: Schematische voorstelling van de UNIBUS.



De Unibus bestaat uit 56 lijnen, waarover adressen, data en besturingsinformatie worden gezonden. Daar de vorm van communicatie voor alle systeemcomponenten dezelfde is, heeft de processor op dezelfde eenvoudige manier toegang tot randapparatuur als tot het geheugen. Elke geheugenplaats en elk deviceregister heeft een eigen, uniek adres op de Unibus. Omdat de PDP 11 zowel 16-bits woorden als 8-bits bytes kan adresseren, komt het aantal adressen niet overeen met het aantal woorden. Een PDP 11-woord is verdeeld in een high en een low byte. Low bytes hebben even adressen en high bytes oneven. Een woord heeft hetzelfde adres als zijn low byte.

Daar er 18 adreslijnen in de Unibus lopen, is het in principe mogelijk 128k woorden te adresseren. De fabrikant heeft echter 2 adresbits gereserveerd voor toekomstige uitbreiding, zodat momenteel 32k 16-bits woorden of 64k 8-bits bytes geadresseerd kunnen worden. Hiervan zijn de hoogste 4k adressen gereserveerd voor registers in randapparatuur (zie fig. 3).

#### 2.1.2. De processor

De PDP 11/20 heeft een hardware stack-mechanisme waardoor genneste interrupts en een automatische reentrant subroutine structuur mogelijk zijn. De processor van de PDP 11/20 bevat 8 general purpose registers, genummerd 0 t/m 7. Deze registers kunnen worden gebruikt als akkumulator, autoindex registers, of als pointers. De registers R6 en R7 hebben speciale toepassingen. R6 wordt gebruikt als hardware stackpointer, terwijl R7 de programmateller is. De processor kan data op een willekeurig adres behandelen alsof het een akkumulator was. Hierdoor kunnen de general purpose registers voor andere doeleinden worden gebruikt. Het processor statusregister bevat informatie over de prioriteit van de processor en het resultaat van de vorige instructie. De instructieset van de PDP 11/20 omvat enkel- en dubbel-operand instructies, programmacontrole instructies, conditiecode operators en een aantal speciale instructies. Een groot aantal van deze instructies kan

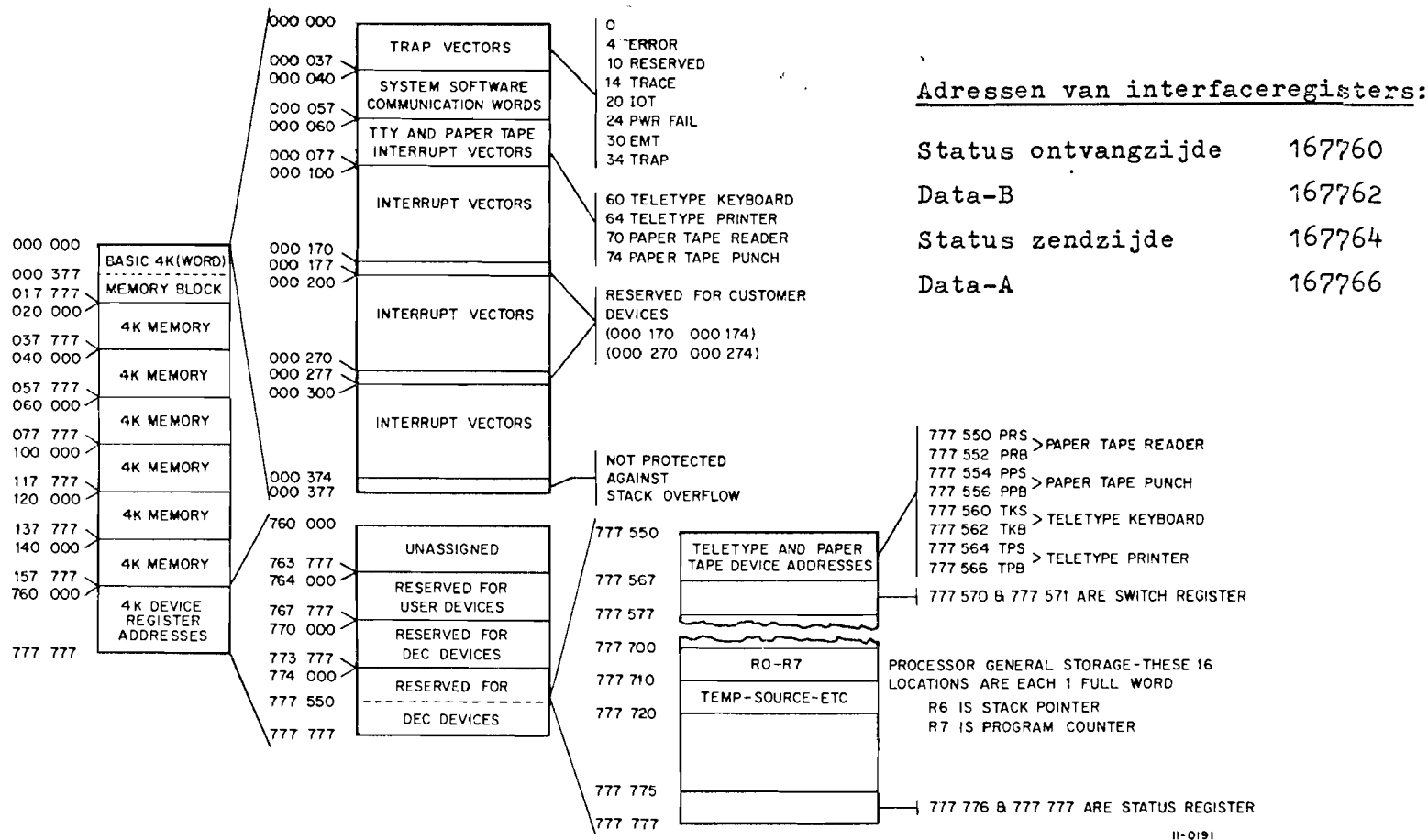


Fig. 3: Bezetting van de Unibusadressen.

zowel op woorden als op bytes werken. De operanden worden geadresseerd via een general purpose register. In het instructiewoord zijn voor elke operand zes bits gebruikt (zie fig. 4). Drie hiervan geven het te gebruiken register aan. De overige drie specificeren de manier, waarop dit register moet worden gebruikt. Het kan bijv. zijn, dat het register de operand zelf bevat, of dat het register het 16-bits adres van de operand bevat. Verder zijn er modes, waarbij het register automatisch wordt opgehoogd of afge-laagd. Ook indexering is mogelijk.

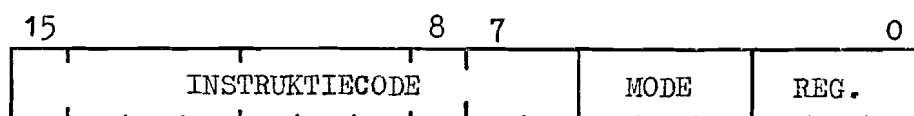


Fig. 4: Indeling van een enkel-operand instructie.

### 2.1.3. De master-slave relatie

De communicatie tussen twee systeemcomponenten via de bus geschiedt in de vorm van een master-slave relatie. Op elk moment is er slechts één device, dat de bus beheert. Dit device is dan "busmaster" en controleert de bus als het met een ander device, de "slave", communiceert. De master-slave relatie is dynamisch, d.w.z. dat de processor het beheer over de bus kan afstaan aan een randapparaat dat hierom vraagt. Dit apparaat kan dan zonder tussenkomst van de processor data transporteren van of naar het geheugen of naar een ander randapparaat. Het datatransport over de Unibus is "interlocked", d.w.z. op een kommando van het masterdevice moet een antwoord van de slave volgen om een transport af te sluiten. Hierdoor is de communicatie onafhankelijk van de fysische afstand en de responsnelheden van master en slave.

### 2.1.4. Verandering van busmaster

Aangezien de Unibus wordt gebruikt zowel door de processor als door randapparaten, is er een prioriteitenstructuur om vast te

stellen welk device het beheer over de bus krijgt. Elke systeemkomponent die busmaster kan worden, heeft een prioriteit. Een device anders dan de processor, vraagt in het algemeen het beheer over de bus om één van de volgende redenen:

- a) om zonder tussenkomst van de processor een datatransport te verzorgen.
- b) om het lopende programma tijdelijk op te schorten en de processor eerst een interrupt service routine te laten uitvoeren.

Het aanvragen en toekennen van het beheer over de bus wordt uitgevoerd parallel aan het datatransport, d.w.z. terwijl het ene device de busgebruikt, worden nieuwe aanvragen in behandeling genomen en wordt de volgende gebruiker aangewezen. Als geen ander device met hogere prioriteit de bus vraagt, is de processor busmaster. Een aanvraag voor het beheer van de bus kan worden ingediend op één van de vijf verschillende prioriteitsniveaus. De hoogste prioriteit heeft het "non-processor request" (NPR). Dit is een aanvraag voor een non-processor datatransport, dat tussen de verschillende buscycli, nodig voor de uitvoering van een instructie, plaats kan vinden. Aanvragen op de vier lagere niveaus, aflopend van BR 7 tot BR 4, worden door de processor alleen gehonoreerd nadat een instructie volledig is afgewerkt. De prioriteit van ieder randapparaat is in de hardware vastgelegd. De prioriteit van de processor kan onder programmakontrolle worden gezet op een niveau, lopend van 0 tot 7, door de bits 5, 6 en 7 van het processor statusregister te wijzigen. Als de processor prioriteit op een bepaald niveau is gebracht, worden alle aanvragen voor de bus op dat, en alle lagere niveaus, genegeerd. Als een device eenmaal het beheer over de bus heeft verkregen, kan het ofwel een datatransport verzorgen, ofwel een interrupt aanvragen. Non-processor datatransporten kunnen plaatsvinden tussen elke twee apparaten. Gewoonlijk echter geschiedt een dergelijk transport tussen een groot secundair geheugen (bijv. een disk) en het kerngeheugen. Zo'n transport kan gebeuren met maximale snelheid, d.w.z. ca. 2,5 mil-

joen woorden per seconde.

Een device dat, na het verkrijgen van het beheer over de bus, een interrupt aanvraagt, kan dan de macht en flexibiliteit van de processor gebruiken. Het device stuurt de processor een interrupt kommando, samen met het adres van de woorden in het geheugen die het startadres en de status van de gewenste interruptroutine bevatten (zie fig. 5). De processor redt daarop de geldende processorstatus en de programmateller op de stack. De nieuwe status en programmateller worden van de opgegeven interruptvektor gehaald en de servicerroutine wordt gestart. De routine wordt beëindigd door een Return from Interrupt (RTI) instructie. Hierdoor worden de oude status en programmateller van de stack gehaald en het onderbroken programma hervat. Een interrupt servicerroutine kan op zijn beurt worden onderbroken door een tweede interrupt van voldoende hoge prioriteit. De gang van zaken is in dit geval precies als boven beschreven. Nadat deze tweede interrupt is afgewerkt, wordt de eerste routine hervat en na beëindiging hiervan gaat de processor verder met het oorspronkelijke programma.

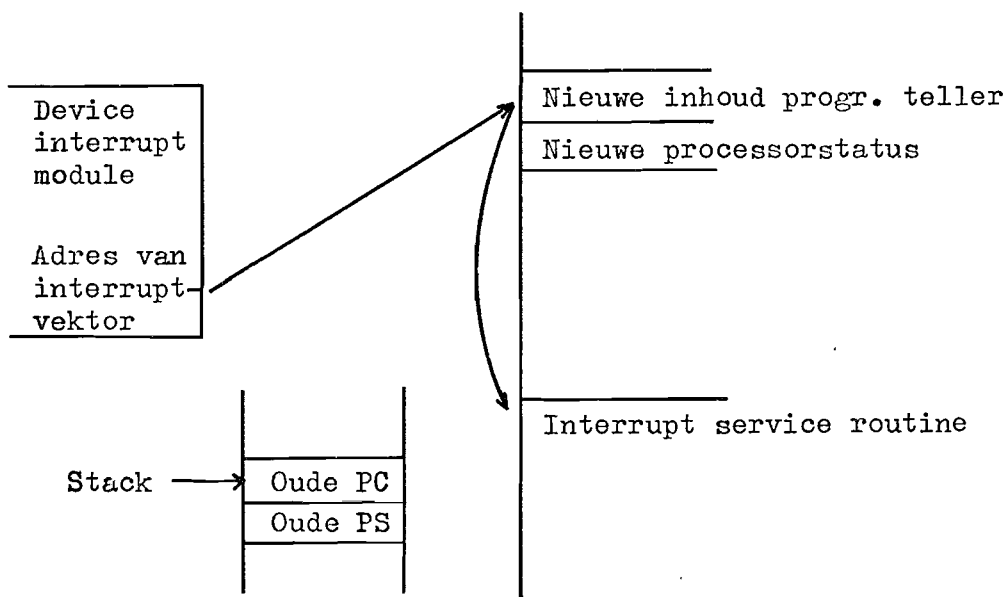


Fig. 5: Gang van zaken bij interrupt.

## 2.2. HET B6700 DATAKOMMUNIKATIESYSTEEM

### 2.2.1. De hardwarestructuur

Het centrale gedeelte van het Burroughs B6700 systeem, zoals dat op het rekencentrum van de THE in gebruik is, bestaat uit twee processoren en vier geheugenbanken, welke onderling zijn verbonden. Een schematische voorstelling hiervan is gegeven in figuur 6.

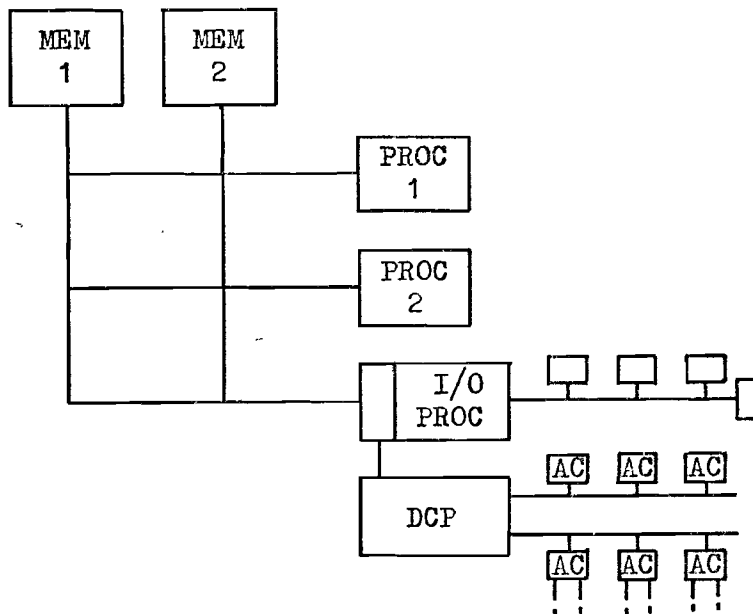


Fig. 6: Schema van het Burroughs B6700 systeem.

Voor de in- en uitvoer van gegevens zijn twee onafhankelijke perifere processoren aanwezig, te weten de I/O processor en de Data Communicatie Processor (DCP) (zie fig. 6 en lit. 3). De I/O processor verzorgt, via data switching channels, het transport van en naar de konventionele randapparatuur. Dit kunnen zijn kaartlezers, regeldrukkers, diskunits, enz. De DCP regelt het transport van en naar het datakommunikatienetwerk (zie lit. 4 en 5). Deze processor heeft d.m.v. de Memory Exchange en de Memory Interface, die met de I/O processor worden gedeeld, toegang tot het centrale ge-

heugen. Aan de andere kant is de DCP via Adapter Clusters, Line Adapters en lijnen verbonden met een aantal verschillende terminals. Elke binnenkomende lijn is aangesloten op zijn eigen lijnadapter. Deze adapters zijn gegroepeerd in groepen van maximaal 16, de clusters. Elke DCP kan maximaal 16 adapterclusters en dus maximaal 256 lijnen besturen. De functie van de verschillende onderdelen van de transmissieketen zal hierna worden uiteengezet.

Als een op het datacommunicatienetwerk aangesloten terminal informatie naar het centrale systeem stuurt, zendt het de bitrepresentatie van karakters over de lijn. De adapter, die op die lijn is aangesloten, neemt de inkomende bits op en vormt zo weer karakters. Telkens als een karakter is binnengekomen, geeft de adapter een teken aan de DCP. De DCP verzamelt de binnenkomende karakters tot boodschappen, die dan worden doorgestuurd naar het centrale systeem. De DCP heeft verder tot taak een aantal controlefuncties uit te oefenen, zoals het beantwoorden en beëindigen van oproepen, het verzorgen van polling operaties en het handhaven van de formele lijndisciplines.

### 2.2.2. De softwarestructuur

Het uiteindelijk doel van het datacommunicatiesysteem is het vereenvoudigen van datatransport tussen een object programma en een terminal, of tussen twee terminals. De gebruiker moet met datacommunicatie-terminals kunnen praten als met konventionele randapparatuur. Hiertoe moet de data een aantal transformaties ondergaan. Om dit te beschrijven worden een viertal niveaus onderscheiden. Deze niveaus verschillen in complexiteit van datastructuur. Op het hoogste niveau, niveau 3, bestaan de object programma's. Deze manipuleren met files d.m.v. lees- en schrijfoopdrachten. Een dergelijke instructie aktiveert in het operating system, het Master Control Program (MCP), een I/O intrinsic (zie fig. 7). Door het specificeren van een file van de soort "datacom" kan een programma communiceren met het datacommunicatienetwerk. Een datacommunicatiefile omvat één of meerdere stations. "Station" is een logisch begrip, in tegenstelling tot "terminal". Aan één terminal kunnen meerdere stations toegekend zijn.

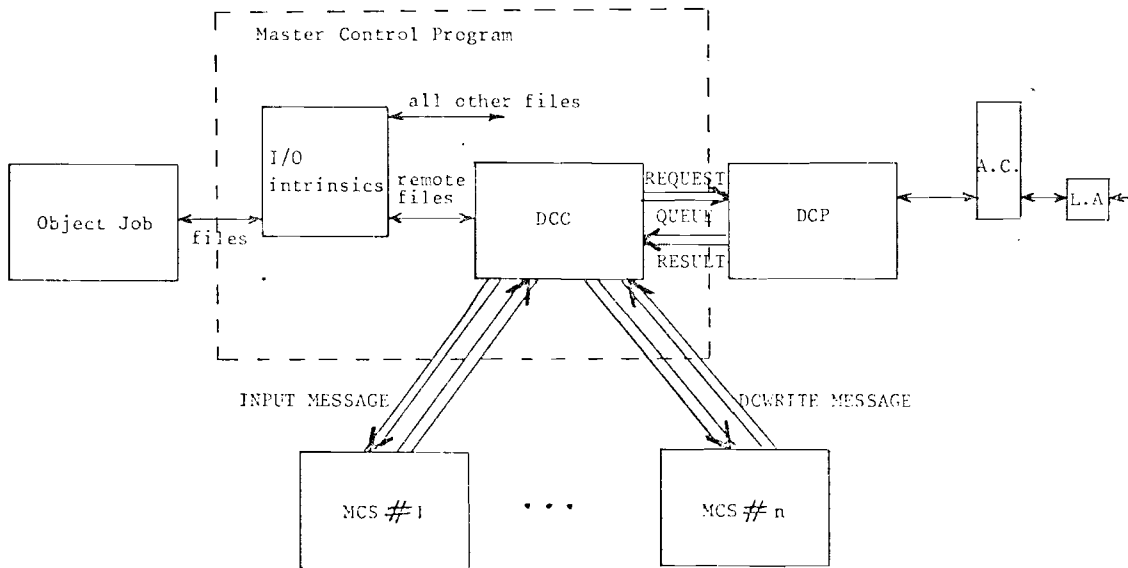


Fig. 7: Softwarestructuur van het B6700 datakommunikatiesysteem.

Een lees- of schrijfofdracht in een object programma heeft tot gevolg, dat een boodschap van of naar de opgegeven file wordt getransporteerd. Op niveau 2 wordt de datatructuur beschreven in termen van dergelijke boodschappen. In het B6700 systeem is een boodschap een array van woorden, bestaande uit een tekstgedeelte en een header (lit. 6). De header bevat informatie over oorsprong, bestemming, lengte enz. Wordt door een object programma een boodschap naar een datakommunikatiefile verstuurd, dan geeft de I/O intrinsic deze boodschap door aan een ander gedeelte van het MCP, de Data Communicatie Controller (DCC). De DCC verzorgt het contact tussen de onafhankelijke programma's, die in het centrale systeem draaien, en de stations van het datakommunikatienetwerk. De DCC stuurt de voor een datakommunikatiefile bestemde boodschap naar de DCP via de zgn. "Request Queue". Een Queue is in het B6700 systeem een lijst van boodschappen, waarbij aan beide zijden van de lijst bood-



schappen toegevoegd of verwijderd kunnen worden (lit. 4 en 6). De DCP kan aan de hand van deze lijst de boodschappen uit het centrale systeem halen en verder verwerken. Een boodschap, afkomstig uit het datakommunikatienetwerk, wordt door de DCP op dezelfde manier via de "Result Queue" aan de DCC doorgegeven.

De stroom van boodschappen door de DCC wordt gecontroleerd door één of meerdere Message Control Systems (MCS). Een MCS is gekoppeld aan één of meerdere stations. Normaal gaan alle goede boodschappen rechtstreeks naar het object programma en komen alleen de foutmeldingen in het MCS. Het MCS kan echter ook besluiten om alle boodschappen te zien en kan dan alle mogelijke bewerkingen op deze boodschappen uitvoeren. De belangrijkste functies van het MCS zijn:

- het openen en sluiten van datakommunikatiefiles, inclusief log-in procedures.
- het reageren op opgetreden fouten.
- het doorsturen van boodschappen naar hun uiteindelijke bestemming (message switching).
- het editten en formatten van boodschappen.

De kommunikatie tussen MCS en DCC geschiedt ook d.m.v. boodschappen. De boodschappen van MCS aan DCC worden "DCWRITE" boodschappen genoemd, terwijl de boodschappen van DCC aan MCS "Input Messages" heten.

Burroughs levert een aantal standaard MCS's. Zo is er een MCS voor Remote Job Entries, dat maximaal vier RJE's kan behandelen (lit. 7) en een MCS voor langzame terminals (bijv. teletypes), genaamd CANDE (lit. 8).

De boodschappen, die via de "Request Queue" en de "Result Queue" met de DCP worden uitgewisseld, worden daar verder verwerkt. Dit houdt in, dat boodschappen worden opgebouwd uit ontvangen karakters of ontleed in te verzenden karakters. Op niveau 1 wordt het datatransport dan ook beschreven in termen van karakters. De DCP is geprogrammeerd voor het verwerken van karakters. Daarvoor is informatie

over de fysische en logische karakteristieken van het datakommunikatienetwerk nodig. Deze informatie wordt gegeven in het Network Definition Language (NDL) programma (lit. 9). M.b.v. NDL wordt het netwerk beschreven. Er wordt gespecificeerd welke lijnadaptors worden gebruikt, wat voor terminals zijn aangesloten, welke regels er gelden voor de konversatie over de verschillende lijnen (de lijndiscipline), enz. Het NDL programma is verdeeld in acht sekties. In elk van deze sekties wordt een bepaald aspekt van het netwerk beschreven.

De DCP stuurt dus karakters naar de lijnadaptors, via de adapter clusters. Over de lijn kan de data in verschillende codes worden getransporteerd, zoals EBCDIC, ASCII, BAUDOT, enz. Alle codes worden in de DCP echter vertaald naar EBCDIC, de interne code van het B6700 systeem. Vanuit de adapter worden de karakters bit voor bit over de lijn verzonden, het minst signifikante bit voorop. Het fysische transport van de bits over de lijn wordt beschreven in het laagste niveau, niveau 0.

### 3. DE VERBINDING

#### 3.1. DE LIJN

Voor een fysieke verbinding van een terminal met een computer staan een aantal mogelijkheden open. Voor kleine afstanden kan meestal worden volstaan met gewone aderpennen. Dit heet "direct connect". Voor grotere afstanden wordt gewoonlijk gebruik gemaakt van telefoonlijnen, aangezien die meestal al ergens in de buurt liggen. Dit kunnen normale geschakelde (switched, dial in/ dial out), of speciaal voor dit doel gereserveerde (leased) lijnen zijn.

De transportsnelheid op de lijn wordt in het B6700 datacommunicatiesysteem mede bepaald door de lijnadaptors. De hoogste snelheid (9600 baud) heeft een adapter van klasse 5. Deze adapter kan zowel synchroon als asynchroon zenden en ontvangen. Bij synchroon transport worden de karakters onmiddellijk na elkaar over de lijn gezonden. Bij asynchroon transport is er tussen opeenvolgende karakters een tussenruimte. Dit soort transport komt voor bij terminals, die geen buffercapaciteit hebben. Voor de synchronisatie van zender en ontvanger op bitniveau is timing-informatie nodig. Deze wordt samen met de data over de lijn gestuurd. De karaktersynchronisatie wordt bij asynchroon transport verkregen door aan elk karakter een start- en een stopbit toe te voegen. Hierdoor kan de ontvanger begin en einde van een karakter bepalen. Bij synchroon transport hoeven zender en ontvanger alleen bij het begin van elk transport te worden gesynchroniseerd. Hiertoe wordt, voorafgaand aan de eigenlijke data, een aantal speciale synchronisatiekarakters verzonden, waardoor de ontvanger in de pas kan komen. Voor de verbinding van de PDP 11/20 met het datacommunicatienetwerk is gekozen voor synchroon transport, omdat dit bij grotere transporten sneller werkt.

De lijnadaptors kunnen slechts half-duplex werken, d.w.z. er kan op elk moment slechts in één richting gezonden worden.

### 3.2. DE MODEMS

Bij het transporteren van data over een telefoonlijn is het in de meeste gevallen nodig een modulator-demodulator (MODEM) systeem te gebruiken om een acceptabele lijnkaracteristiek te verkrijgen. De informatiebits worden omgezet in een gemoduleerd signaal en zo verzonden. Aan de ontvangzijde wordt het signaal gedemoduleerd en de zo herwonnen informatie doorgestuurd. De door het reken-centrum gebruikte modems werken volgens het principe van frequency-shift-keying. Deze modems zijn oorspronkelijk ontwikkeld door R.C. Daleman (lit. 10). F. Simonis heeft een aantal wijzigingen aangebracht, voornamelijk i.v.m. de door het rekencentrum gewenste standaardisatie (lit. 11).

De koppeling met de modem geschiedt geheel volgens de aanbevelingen V.24 van CCITT. De volgens deze aanbevelingen gebruikte signalen zijn gegeven in fig. 8. Het signaalniveau dient te liggen tussen +3 en +25 V (data 0, ON-toestand) of tussen -3 en -25 V (data 1, OFF-toestand). De in fig. 8 genoemde pinnummers zijn de aansluitingen op de gestandaardiseerde 25-polige CCITT-connector.

pinnr.	signaalnaam	afk.
7	aarde	-
2	transmit data	TMD
3	receive data	RCD
4	request to send	RTS
5	ready for sending	RFS
6	dataset ready	DSR
20	connect dataset to line	CDSL
15	transmitter signal element timing	TSET
17	receiver signal element timing	RSET
8	data carrier detector	DCD

Fig. 8: Signalen tussen interface en modem volgens CCITT, aanbeveling V.24.

### 3.3. DE LIJNDISCIPLINE

Er zijn verschillende mogelijkheden om het datatransport over de lijn te controleren (lit. 12). Bij zgn. "polled lines" scant de DCP de verschillende lijnen in een bepaalde volgorde en met een bepaalde frekwentie af. Aan elk van de aangesloten stations wordt gevraagd of het iets te zenden heeft. Het station kan dan al of niet een boodschap sturen. Heeft het centrale systeem een boodschap voor een bepaald station, dan wordt dit met een "select" procedure eerst gevraagd of het een boodschap kan ontvangen. Luidt het antwoord bevestigend, dan wordt de boodschap verzonden. Is het antwoord ontkennend, dan wordt de procedure na enige tijd herhaald.

Voor sommige stations is de bovenstaande procedure niet geschikt, bijv. indien geen buffercapaciteit aanwezig is. In dat geval kan een andere procedure gevolgd worden, die wel "hybride" wordt genoemd. Hierbij kan een station dat een boodschap wil verzenden, eerst een zgn. "wake-up code" naar het centrale systeem sturen. Dit systeem heeft dan de plicht om onmiddellijk de polling procedure te starten om uit te vinden welk van de aangesloten stations een boodschap wil verzenden.

Wil men van te voren niet vastleggen welke van de partners in de konversatie het beheer over de lijn heeft, dan kan de "contention mode" worden gebruikt. Hierbij vraagt het station dat een boodschap wil verzenden eerst het beheer over de lijn ("mastership"). Wordt dit door de andere stations goed gevonden, dan wordt de aanvrager "master" en kan zijn boodschap verzenden. Na afloop geeft de master de lijn weer vrij. Deze is dan "idle".

Voor de konversatie tussen PDP 11/20 en B6700 is gekozen voor de laatste methode, die daarom hierna zal worden beschreven. De voor de konversatie te gebruiken code is de American Standard Code for Information Interchange (ASCII '67). Voor de juiste betekenis en

het gebruik van de in deze code gedefinieerde controlekarakters wordt verwezen naar lit. 12.

De konversatie volgens de hieronder te geven regels is een vraag-antwoord spel. Op elk transport van de master moet een kort antwoord van de slave komen. Dit betekent, dat de richting van het transport over de lijn telkens wordt omgeschakeld. Hierbij moeten zender en ontvanger steeds opnieuw gesynchroniseerd worden. Daarom wordt elk transport voorafgegaan door vier synchronisatiekarakters (SYN). In de diagrammen is dit niet aangegeven.

### 3.3.1. De masterstatus

De gekozen procedure voor het aanvragen en verkrijgen van het beheer over de lijn is gegeven in fig. 9. Als een station een boodschap wil verzenden moet het eerst de masterstatus verkrijgen. Hiertoe zendt het een Enquiry (ENQ) karakter. Het andere station zendt hierop als antwoord een "Affirmative Acknowledgement" (ACK) als het een boodschap kan ontvangen. Een "Negative Acknowledgement" (NAK) als antwoord betekent, dat het station niet in staat is om iets te ontvangen. In het laatste geval kan het ENQ-karakter worden herhaald (maximaal 20 maal). Als dan nog steeds geen ACK is ontvangen, wordt de poging opgegeven en een EOT gezonden. Ontvangt het station, dat de masterstatus aanvraagt, geen of een ongeldig antwoord (hetgeen gedetekteerd wordt d.m.v. een time-out), dan kan het de ENQ herhalen (tot maximaal 20 maal). Als dan nog geen geldig antwoord is ontvangen, wordt een error recovery procedure gestart. Als het station, dat de masterstatus aanvraagt, een ENQ ontvangt, laat het zijn eigen aanvraag vallen en zendt een passend antwoord. Om te voorkomen, dat beide stations tegelijkertijd blijven vragen om de masterstatus, is de time-out van de PDP 11 korter dan die van het B6700 systeem (0,5 tegen 1 sec.).

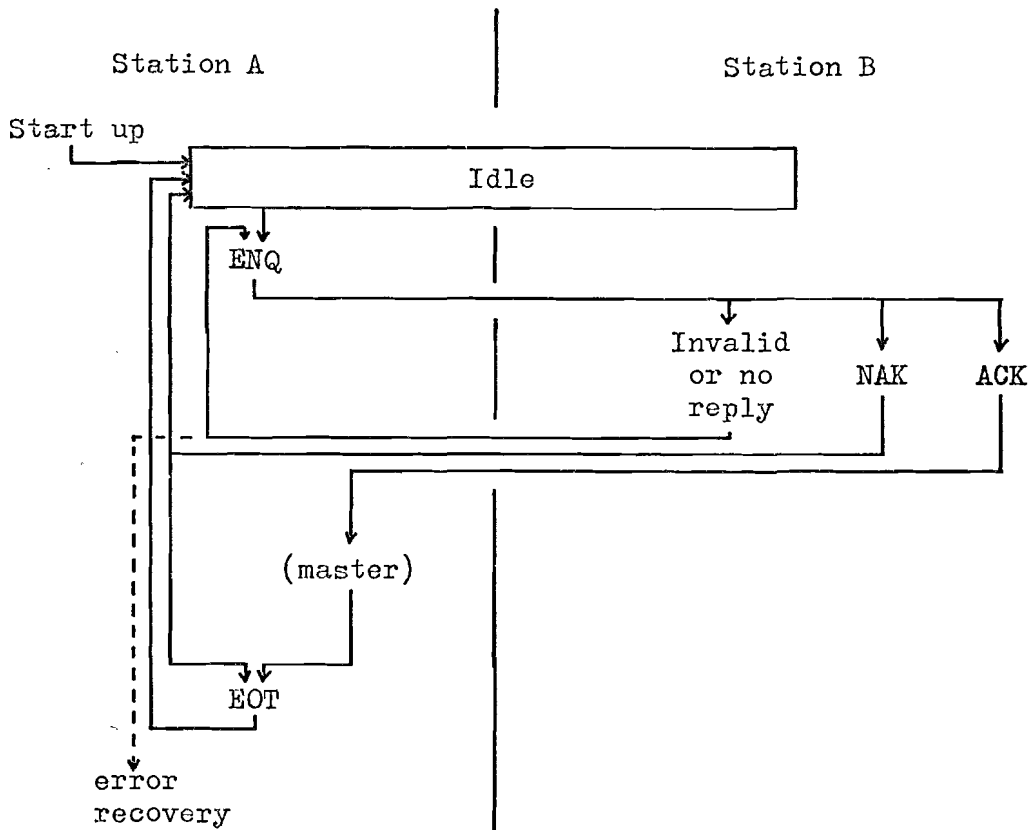


Fig. 9: Procedure voor het verkrijgen van het beheer over de lijn.

### 3.3.2. Het datatransport

Als één van de beide stations volgens de beschreven methode de masterstatus heeft verkregen, kan het eigenlijke datatransport plaatsvinden. Ook dit gebeurt volgens bepaalde regels. De voor de verbinding met de PDP 11/20 gekozen procedure is gegeven in fig. 10. Er dient op gewezen te worden dat de hier beschreven methode slechts één van de vele mogelijkheden is. Binnen het B6700 data-kommunikatiesysteem kunnen vele andere procedures worden gerealiseerd. Voorbeelden hiervan zijn te vinden in lit. 12 en 13.

De master verzendt de data in blokken. Na elk blok geeft het ontvangende station d.m.v. een ACK of een NAK aan of het blok goed, dan wel foutief is overgekomen. Zijn er tijdens het transport

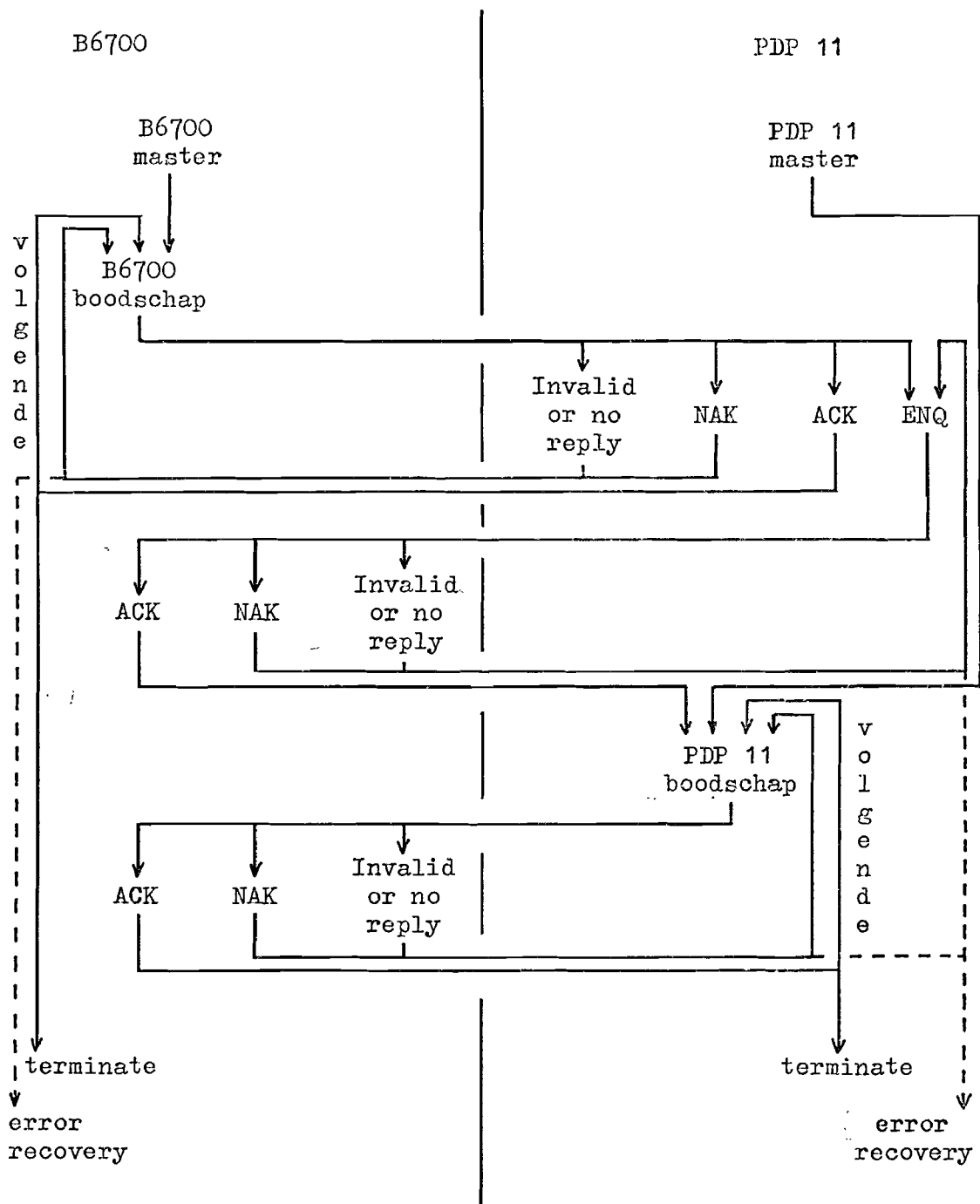


Fig. 10: Procedure voor het datatransport.



fouten opgetreden of ontvangt de master geen of een ongeldig antwoord, dan wordt het blok herhaald. Als het na 20 maal nog niet is gelukt het transport foutloos te doen verlopen, dan start de master een error recovery procedure. Krijgt de master als antwoord ACK, dan kan een volgend blok worden verzonden. Heeft de master op deze manier alle data overgestuurd, dan kan na de laatste ACK de masterstatus worden opgegeven door het zenden van het End of Transmission (EOT) karakter. Hierdoor komt de verbinding weer in de "idle" toestand.

In de procedure voor de verbinding van de PDP 11/20 met het B6700 systeem is nog een extra mogelijkheid ingebouwd. Behalve ACK en NAK kan de PDP ook ENQ sturen als antwoord op een blok. Hiermee wordt te kennen gegeven, dat het blok goed is ontvangen en tevens dat de PDP data wil verzenden. Het B6700 systeem geeft nu de masterstatus op door het zenden van een ACK, waarna de PDP zijn data kan oversturen. Deze gang van zaken wordt "reverse interrupt" genoemd.

### 3.3.3. De tekstblokken

Een blok, zoals dat over de lijn wordt verzonden, bestaat uit een kop en een tekst. De tekst is het belangrijkste gedeelte en bevat de over te dragen informatie. Voor het transport tussen de PDP 11 en het B6700 systeem is de maximale lengte van de tekst gesteld op 512 karakters. Het begin van een blok wordt aangegeven door het Start Of Header (SOH) karakter. Dan volgt de kop, die wordt afgesloten door het Start of Text (STX) karakter. Vervolgens komt de tekst. Het einde hiervan wordt aangegeven door een End of Text (ETX) karakter. De tekst kan twee verschillende structuren hebben, nl. normaal of transparant. In de normale mode geldt voor de tekst de ASCII-code en hebben controlekarakters hun gewone betekenis. In de transparante mode worden de karakters alleen als bitpatronen behandeld, zodat er ongelimiteerd gebruik kan worden gemaakt van de beschikbare acht bits per karakter. Dit is nodig

om binaire data of programma's in machinecode te kunnen oversturen. Om toch controlefuncties te kunnen aangeven zijn in de transparante mode speciale sequenties gedefinieerd, bestaande uit een controlekarakter voorafgegaan door het Data Link Escape (DLE) karakter. Zo wordt het begin en einde van de transparante tekst aangegeven met DLE STX, resp. DLE ETX. Als in de binaire tekst een bitpatroon, gelijk aan DLE, voorkomt wordt dit eengegeven door een extra DLE-karakter in te voegen. Het ontvangende station herkent deze sequentie, zeeft één van beide DLE's uit de datastroom en behandelt de andere als data.

Als een station om de één of andere reden tijdens het zenden geen data meer heeft en toch de synchronisatie wil vasthouden kan het als tijdruimte een synchronisatie-sequentie zenden. In de ASCII-mode bestaat zo'n sequentie uit SYN-karakters. In de transparante mode moet telkens de groep DLE SYN worden gestuurd. Het ontvangende station zeeft deze sequenties uit de datastroom.

De kop bestaat uit vier karakters. Twee hiervan geven de bron of bestemming van het blok en heten daarom adreskarakters. De waarde en betekenis van deze karakters hangt af van de richting waarin het transport plaatsvindt en van het MCS waarmee de verbinding wordt onderhouden. Voor transport van PDP naar RJE/MCS hebben de karakters de waarde

- 01 voor een kommando van het PDP 11 operators console,
- 02 voor ASCII-tekst, niet afkomstig van het console, en
- 17 voor binaire data.

Voor een transport van RJE/MCS naar de PDP 11 zijn de waarden:

- 01 voor een melding op het operatorsconsole en
- 18 voor blokken die niet voor het console bestemd zijn.

Voor de konversatie met CANDE/MCS, waarbij het transport steeds tussen het console en MCS plaatsvindt, hebben de adreskarakters in beide richtingen de waarde 99. In een blok dat door RJE/MCS naar de PDP 11 wordt gezonden en dat niet voor het console is

bestemd, bevatten de eerste zes karakters van de tekst nadere informatie over de aard en bestemming van de boodschap.

Het laatste karakter van de kop is het transmissienummer. Hiermee worden de verzonden blokken modulo 10 genummerd, om duplicering of verlies van blokken te kunnen detekteren. Het eerste karakter in de kop heeft nog geen betekenis. Deze plaats is gereserveerd voor mogelijk toekomstig gebruik.

#### 3.3.4. Mogelijke fouten

Teneinde het ontvangende station in staat te stellen te bepalen of een blok data foutloos is ontvangen, wordt door het zendende station een hoeveelheid redundantie toegevoegd. In de normale ASCII-mode wordt aan elk karakter een achtste bit, het pariteitsbit, toegevoegd. Dit bit is zodanig, dat het aantal enen in het karakter oneven wordt. In de transparante mode is het gebruik van deze verticale pariteitscontrole niet mogelijk, omdat alle bits gebruikt worden voor de informatieoverdracht. In beide modes wordt een horizontale (of longitudinale) controle toegepast. Hierbij worden aan elk blok, na ETX, 16 redundante bits toegevoegd. Voor het genereren van dit zgn. Cyclic Redundancy Character (CRC), wordt het gehele blok, beginnend na SOH, t/m ETX, met uitzondering van synchronisatiesequenties en DLE's, opgevat als een polynoom (lit. 14). Door dit polynoom te delen door het generatiepolynoom  $x^{16} + x^{15} + x^2 + 1$ , ontstaat een rest. Deze rest wordt als CRC met het blok overgezonden. De ontvangzijde voert dezelfde deling uit en vergelijkt de beide resten.

Behalve pariteitsfouten, kunnen tijdens het ontvangen van een boodschap ook andere fouten optreden. Hiervan moeten het wegvallen van de draaggolf (loss of carrier) en het ontvangen van een te lange boodschap (bufferoverflow) genoemd worden. De oorzaak van de eerste fout kan zijn een kabelbreuk of het wegvallen van de spanning van de zendende modem. Een verminking van het ETX-

karakter heeft tot gevolg, dat het ontvangende station het einde van de tekst niet meer kan detekteren. Hierdoor worden meer dan 512 karakters als tekst aangezien. Naast deze fouten geeft het overschrijden van de in de lijndiscipline vastgestelde time-outs aanleiding tot het nemen van speciale maatregelen. Meestal houdt dat in, dat om herhaling van het blok gevraagd wordt. Indien bij het transport het maximaal aantal keren, dat een blok mag worden herhaald, wordt overschreden, wordt een error recovery procedure gestart. Dit kan tot gevolg hebben, dat de verbinding wordt verbroken en dat een melding op het operatorsconsole wordt uitgetypt. Zonodig dient de operateur maatregelen te nemen om de fout op te heffen.

#### 4. HET PDP 11/20 INTERFACE

In dit hoofdstuk wordt de hardware beschreven, die nodig is om datatransport tussen de PDP 11/20 Unibus en de modem mogelijk te maken. Gedetailleerde dokumentatie hierover is te vinden in lit. 15.

##### 4.1. DE TAKEN

Uit het voorgaande zijn de taken van het interface eenvoudig te destilleren. Het asynchrone, parallelle transport over de Unibus moet worden omgezet in een synchroon, serieel transport over de lijn. Er moet een niveau-omzetting plaatsvinden van het Unibus-signaal naar de CCITT-niveaus en de modem moet worden bestuurd. Verder moeten er pariteitscontroles worden uitgevoerd en moet de lijndiscipline worden gehandhaafd. Hoewel het in principe mogelijk is om al deze taken uit te voeren met hardware, is dit niet gedaan. De voornaamste reden hiervoor is dat noch binnen de groep ER noch binnen het rekencentrum ervaring met een koppeling als de onderhavige bestond. Hierom werd besloten om eerst alleen het hoognodige in hardware uit te voeren en de rest van de taken met software af te wikkelen. Immers, software is flexibeler, eenvoudiger op te bouwen en sneller te wijzigen. Het is ook om deze reden, dat het oorspronkelijke plan van een Direct Memory Access (DMA) interface is opgegeven voor een interrupt serviced interface. Het blijft echter mogelijk om later, als met deze configuratie enige ervaring is opgedaan, een aantal taken alsnog met hardware uit te voeren (zie hoofdstuk 6).

##### 4.2. DE OPZET

Het eigenlijke datatransport tussen een randapparaat en de Unibus geschiedt d.m.v. één of meerdere registers in het interface. Deze registers kunnen zowel flipflop geheugenregisters zijn, als dynamische signalen, die eenvoudig aan de bus worden geschakeld tij-

dens een transport. Het is niet nodig, dat de aard van alle bits van een register hetzelfde is. Zo kan er onderscheid worden gemaakt tussen bits die alleen gelezen kunnen worden, bits die alleen geschreven kunnen worden en bits die beide bewerkingen toelaten. DEC hanteert voor de standaard randapparaten een voorkeursindeling voor de registers. Dit betreft zowel de verdeling van de status- en errorbits in een register, als de adrestoewijzing aan de verschillende registers. In het interface voor de koppeling met het B6700 systeem is deze voorkeursindeling zoveel mogelijk aangehouden.

DEC heeft voor de randapparaten een modulaire opbouw ontwikkeld. De fabrikant levert een grote scala van zelfstandige modules, waarmee de gebruiker op eenvoudige wijze zijn interface kan samenstellen. Twee van deze modules zijn in de verbinding met het B6700 systeem gebruikt, nl. de M105 Adresselektor en de M7821 Interrupt Control.

De taken, die aan de hardware zijn toegewezen, worden als volgt uitgevoerd: De omzetting van parallel naar serie en omgekeerd geschiedt met twee schuifregisters. De aanpassing tussen asynchroon Unibustransport en synchroon transport over de lijn wordt verkregen door de data dubbel te bufferen. Hierdoor wordt bereikt, dat bij het zenden met een snelheid van 10 kbaud gemiddeld eens in de 1,6 ms een datawoord in het interface moet worden geplaatst. Hierbij mag zo nu en dan een vertraging van maximaal 2,4 ms optreden. Het aanpassen van de signaalniveaus tussen Unibus, interface logica en modem, gebeurt met speciale niveau-omzetters.

Een blokschema van het interface is in fig. 11 gegeven. De data, die van de PDP 11 naar het B6700 systeem moet worden getransporteerd, wordt vanuit de Unibus in het 16-bits flipflopregister DATA-A geplaatst. Vandaar wordt het datawoord doorgeschoven naar het bufferregister DATA-A'. De multiplexor zorgt dat eerst het low byte en vervolgens het high byte in het schuifregister komt. Dit

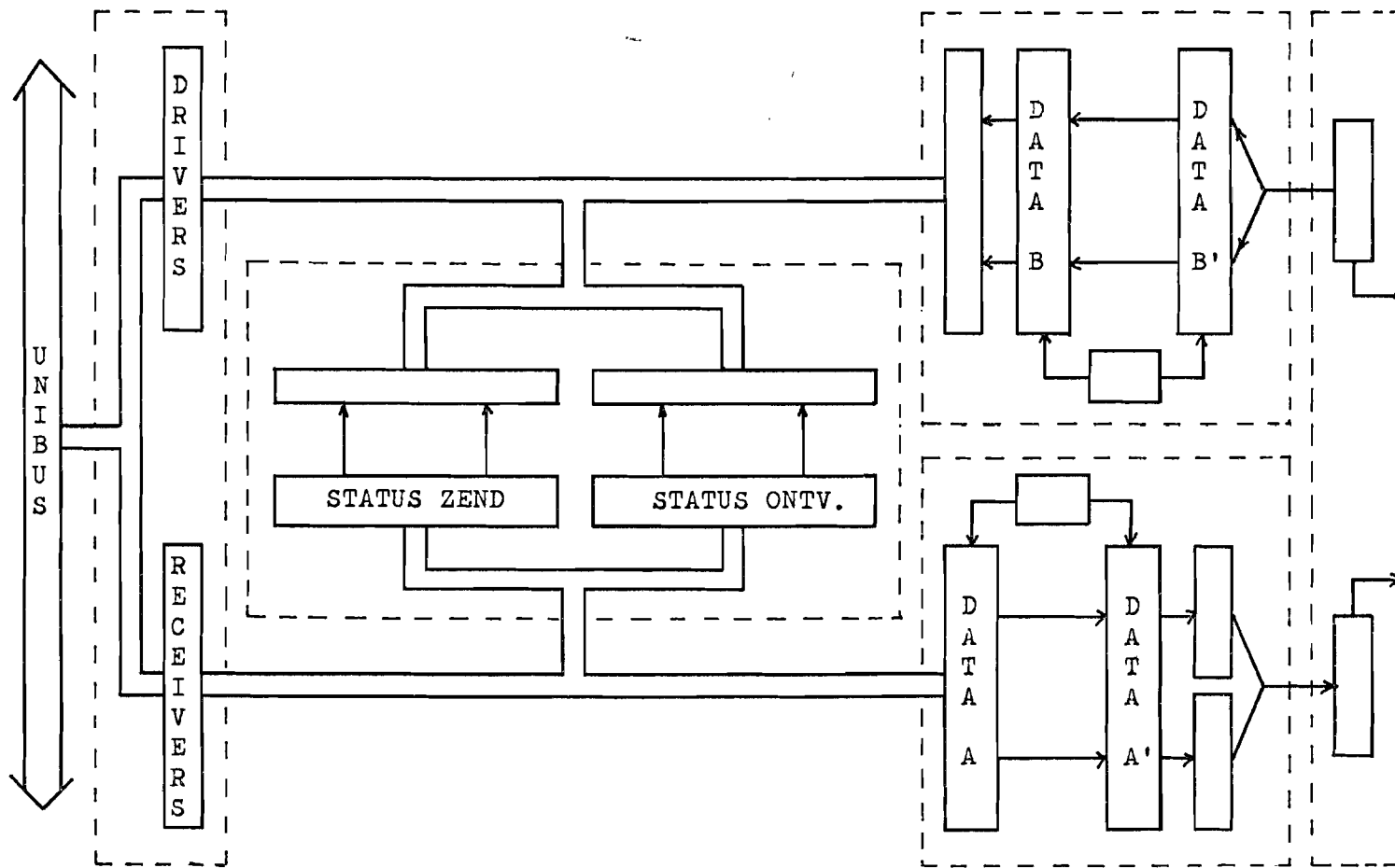


Fig. 11: Het blokschema van het PDP 11/20 interface.

schuifregister schuift, op klokpulsen van de modem, de bits één voor één naar buiten. Via een TTL- naar CCITT-niveau-omzetter worden de bits dan aan de modem toegevoerd.

Aan de ontvangzijde komen de bits één voor één uit de modem, samen met een kloksignaal. Via een CCITT- naar TTL-niveau-omzetter worden deze aan het schuifregister toegevoerd. Is in dit register een byte verzameld, dan wordt dit overgenomen in het 16-bits bufferregister DATA-B'. De distributor zorgt ervoor, dat daarbij eerst het low byte en vervolgens het high byte wordt gevuld. Is DATA-B' vol, dan wordt het datawoord overgenomen in het register DATA-B. Vandaaruit kan het dan over de Unibus worden getransporteerd.

Teneinde vanuit de Unibus de gang van zaken binnen het interface te kunnen controleren, zijn statusinformatie en foutmeldingen in de beide statusregisters beschikbaar. Via deze registers kan het interface worden bestuurd. Vanuit de statusregisters wordt ook de Interruptmodule bestuurd, als er een interrupt moet worden aangevraagd.

#### 4.3. HET DATATRANSPORT

##### 4.3.1. De adresselektor

Om als slave aan datatransporten te kunnen deelnemen, moet een rand-apparaat weten wanneer het aangesproken wordt. Het moet dus een adres kunnen herkennen als het op de adreslijnen van de Unibus verschijnt. Verder moet het dan op de juiste wijze op het door de master, via de controlelijnen, gegeven kommando reageren. De M105 adresselektor voorziet hierin. Deze module wordt aan de adres- en controlelijnen van de Unibus gekoppeld en levert dan een aantal signalen, waarmee op eenvoudige wijze vier 16-bits registers bestuurd kunnen worden. Het blokschema van de M105 is gegeven in fig. 12. De vier "select"-signalen worden gebruikt om één van de vier registers te selekteren. De "in"- en "out"-signalen geven aan of er data van het register naar de Unibus moet, of andersom. Bij het schrijven



in het register wordt bovendien aangegeven of het het low of het high byte, dan wel het hele woord betreft.

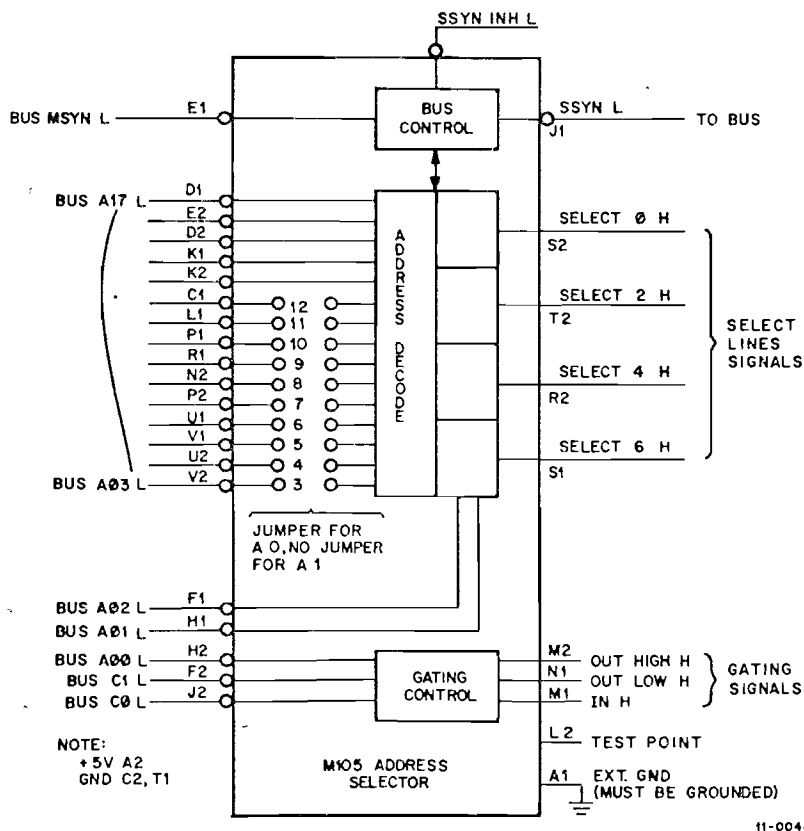


Fig. 12: Het blokschema van de adresselektor.

De adressen, die aan de registers zijn toegekend, zijn:

Statusregister ontvangzijde	167760
Register DATA-B	167762
Statusregister zendzijde	167764
Register DATA-A	167766

#### 4.3.2. De transceivers

Om de snelheid van het datatransport te behouden, moet bij het aansluiten van interfaces op de Unibus enige punten in acht worden genomen. De eigenlijke Unibus is een aangepaste en afgesloten transmissielijn die met speciaal voor dat doel ontworpen circuits moet worden bedreven. De hiervoor gebruikte IC's zijn de 380 receiver en 8881 driver van Signetics.

Teneinde niet voor elk register apart een driver en receiver-set nodig te hebben, is door I. Holtz een print ontworpen waarop zich 16 stel drivers en receivers bevinden. Het schema van zo'n stel is gegeven in fig. 13.

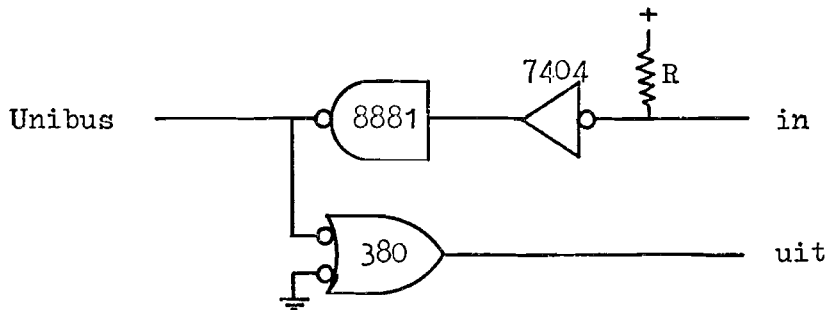


Fig. 13: Het schema van een transceiver.

De signalen lopen via de receivers (380) naar de registers. De uitgangen van de registers zijn d.m.v. poortgroepen verbonden met een wired-or bus, welke via inverters en drivers aan de Unibus is gekoppeld.

#### 4.3.3. Dubbel bufferen

Zowel aan de zenzijde als aan de ontvangzijde wordt de data dubbel gebufferd. Het doorschuiven van de data van het eerste naar het tweede register gebeurt door de in fig. 14 gegeven schakeling.

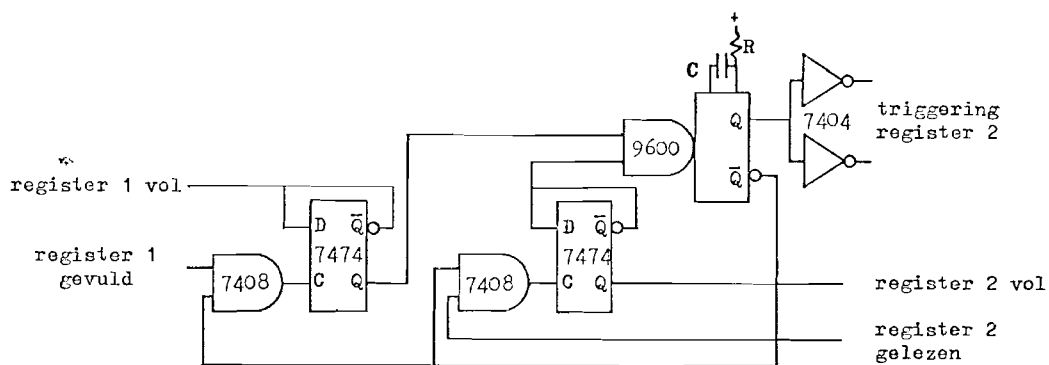


Fig. 14: De sturing voor het dubbel bufferen.

De data wordt doorgeschoven als het eerste register zinvolle data bevat en het tweede niet. Met elk register is een flipflop geassocieerd, die aangeeft of het betreffende register vol is of leeg. Als aan de genoemde voorwaarde is voldaan, wordt de monostabiele multivibrator gestart. De uitgangspuls klokt de data in het tweede register en zet ook beide flipflops om. Bij het vullen van het eerste of het leeghalen van het tweede register wordt tevens de bijbehorende flipflop omgezet.

#### 4.3.4. De multiplexor

De multiplexor dient om afwisselend het low en het high byte uit register DATA-A' in het schuifregister over te nemen. Hiertoe zijn achter DATA-A' twee poortgroepen opgenomen, die als wired-or verbonden zijn. Om de beurt wordt één van de groepen opengezet. Een flipflop houdt bij, welke groep aan de beurt is. Deze flipflop wordt omgezet telkens als een byte in het schuifregister is overgenomen. Verder is er de mogelijkheid om beide poortgroepen dicht te zetten. Op die manier kunnen eventueel karakters in de datastroom gevoegd worden. Het schema van de multiplexor is gegeven in fig. 15.

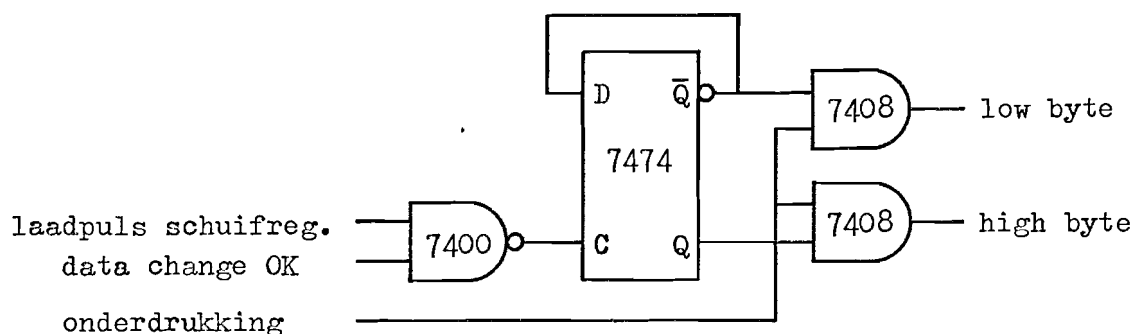


Fig. 15: Het schema van de multiplexor.

#### 4.3.5. Pariteitsbit-generatie

Bij het zenden van ASCII-tekst wordt in het interface het pariteitsbit gegenereerd. Bij het zenden van binaire data moet deze generatie kunnen worden uitgeschakeld. Om onderscheid te kunnen maken tussen ASCII en binaire data, is in het statusregister een codebit opgenomen. Dit bit wordt aan elk byte, dat in register DATA-A wordt geladen, toegevoegd. Aan de hand van dit extra codebit wordt bepaald, of voor het betreffende byte wel of niet een pariteitsbit moet worden berekend. De schakeling voor deze bewerking is in fig. 16 gegeven.

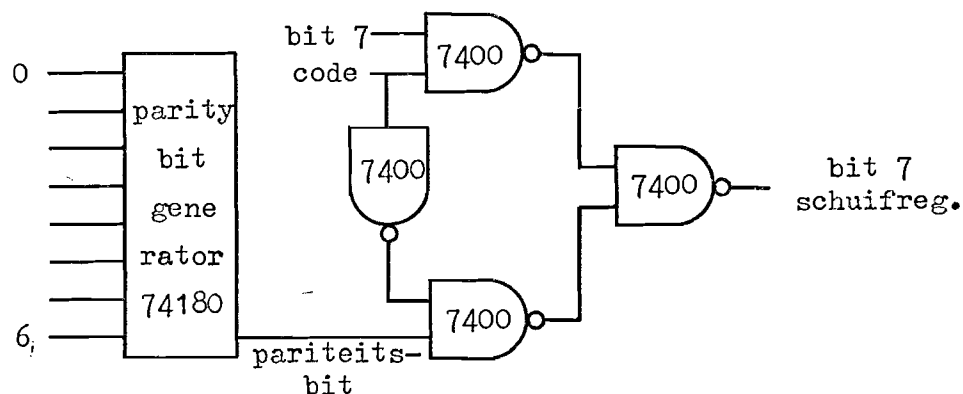


Fig. 16: De schakeling voor de pariteitsbit-generatie.

#### 4.3.6. De schuifregisters

Het omzetten van parallelle data in seriële data en omgekeerd geschiedt d.m.v. schuifregisters. Aan de zenzijde worden de bytes parallel in een register opgenomen en vervolgens bit voor bit verzonden. Het schuiven gebeurt op de klokpulsen van de modulator. Elke keer, als een heel byte is verzonden, wordt onmiddellijk het volgende geladen en verzonden. De schakeling is overgenomen uit lit. 16 (blz. 541) en is weergegeven in fig. 17.

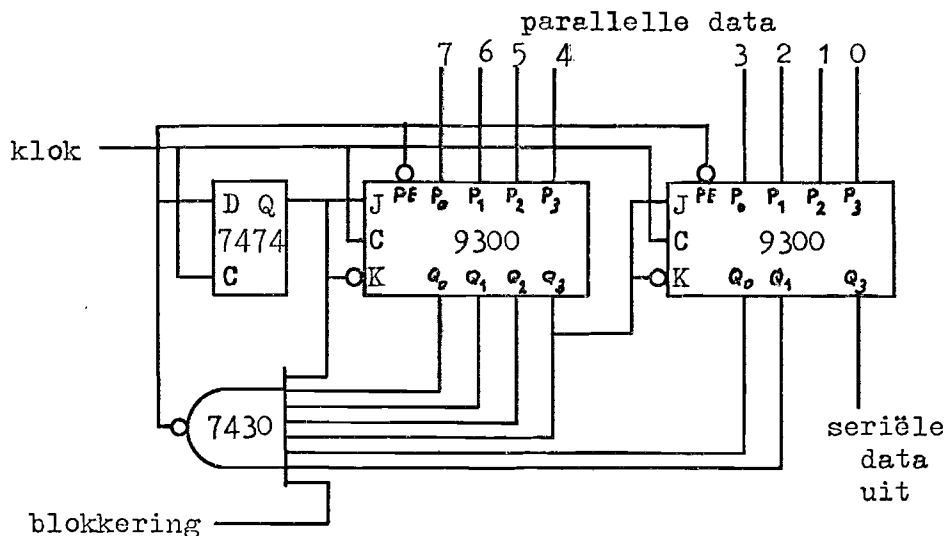


Fig. 17: Het schema van het schuifregister voor zenden.

Het schuifregister aan de ontvangzijde neemt, op de klokpulsen van de demodulator, de binnenkomende bits één voor één op. Als zo een byte is verzameld, wordt een signaal naar de distributor gestuurd. Deze zorgt dan dat het byte wordt overgenomen in register DATA-B', zodat een nieuw byte kan worden ontvangen. De schakeling van dit schuifregister is ook overgenomen uit lit. 16 (blz. 542) en is in fig. 18 weergegeven.

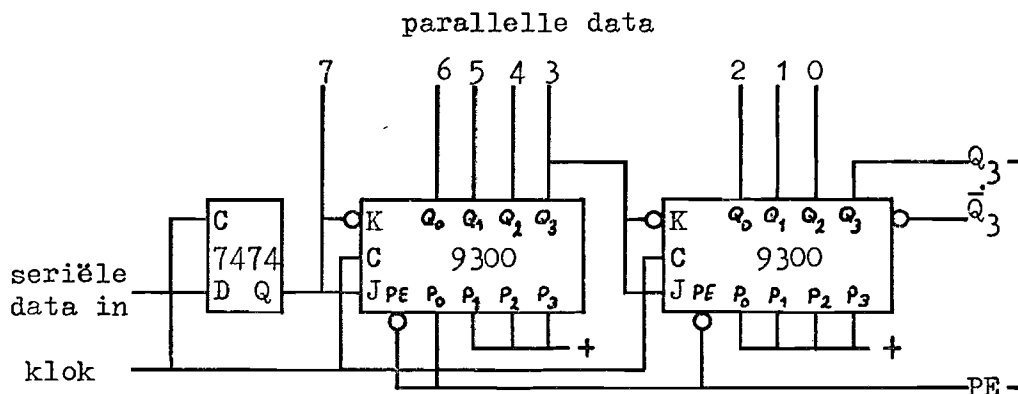


Fig. 18: Het schema van het schuifregister voor ontvangen.

#### 4.3.7. Start en einde van het zenden

Vanuit de Unibus kan het zenden worden gestart en gestopt d.m.v. een bit in het statusregister, genaamd "Request to send" (RTS). Zolang dit bit nul is, worden de pulsen voor het laden van het schuifregister geblokkeerd (zie fig. 17 en 19). Het schuifregister zelf bevat nu enen, die aan de modem worden toegevoerd. Geheel volgens de CCITT-normen zendt het interface dus enen als er geen data wordt verzonden. Voordat het zenden gestart kan worden, moeten eerst de dataregisters worden gevuld. Vervolgens wordt het RTS-bit één gemaakt. Hierdoor wordt de blokkering van het schuifregister opgeheven, en het transport begint.

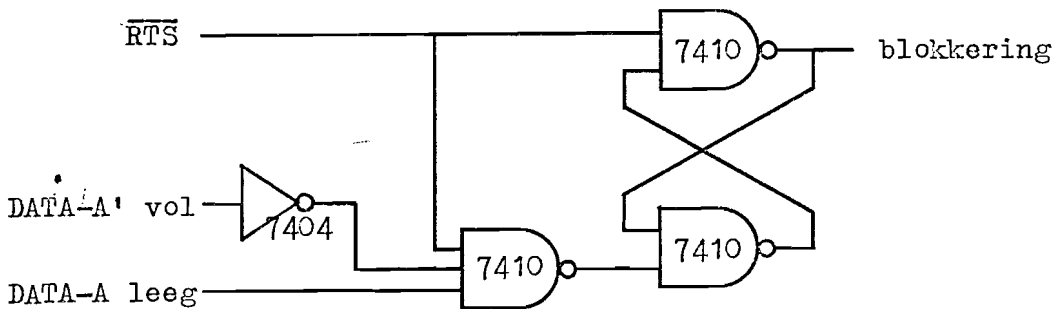


Fig. 19: De schakeling voor het starten en beëindigen van het zenden.

Het zenden kan vanuit de Unibus nu weer worden beëindigd door het RTS-bit nul te maken. Voordat echter de laadpulsen voor het schuifregister mogen worden geblokkeerd, moet eerst alle nog in de registers aanwezige data worden verzonden. De in fig. 19 gegeven schakeling stelt de blokkering uit, totdat de sturing voor het dubbel bufferen aangeeft, dat beide registers leeg zijn. Het laatste byte bevindt zich nu in het schuifregister. Als ook dit is verzonden, wordt een melding in het statusregister gegeven.

#### 4.3.8. De karaktersynchronisatie

Een belangrijke functie van de ontvangzijde is de karaktersynchronisatie. Zolang er geen synchronisatie is bereikt, wordt de inkomende datastroom afgezocht op het voorkomen van synchronisatiekarakters. De standaardmethode hiervoor is de "schuivend venster"-methode. Hierbij worden in het schuifregister acht bits verzameld. Dit bitpatroon wordt vergeleken met het SYN-karakter. Komen beide niet overeen dan wordt de inhoud van het schuifregister over een bit verschoven, waarbij een nieuw databit wordt opgenomen. Dit wordt telkens herhaald, totdat de patronen gelijk zijn. Dan is karaktersynchronisatie verkregen en kunnen steeds acht bits als een karakter worden genomen. Het schema van de schakeling is gegeven in fig. 20. Zie ook fig. 18.

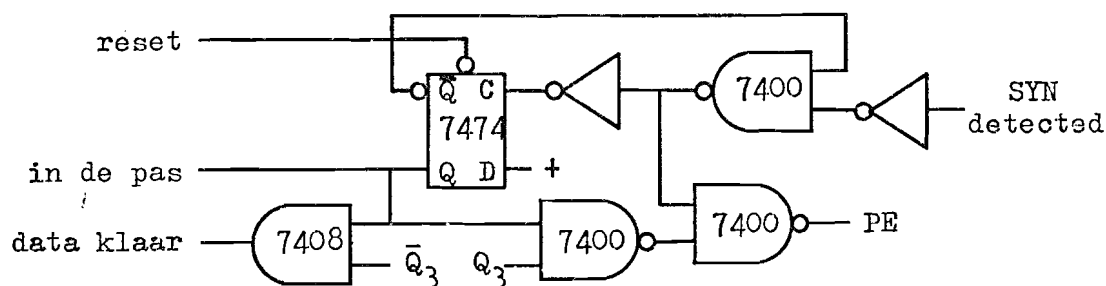


Fig. 20: De schakeling voor karaktersynchronisatie.

De werking van deze schakeling is als volgt: Wordt een SYN-karakter gevonden, dan wordt de flipflop gezet. Tevens wordt de schakeling uit fig. 18 kortgesloten, zodat nu telkens na acht klokpulsen het signaal "data klaar" wordt gegeven. Als een hele boodschap is ontvangen, of als de draaggolf wegvalt, wordt de flipflop gereset. Hierdoor wordt de terugkoppeling van het schuifregister verbroken, zodat de bitstroom weer bit voor bit bekeken kan worden.

#### 4.3.9. De distributor

De in het schuifregister verzamelde bytes worden overgenomen in register DATA-B'. De distributor zorgt, dat eerst het low byte en

vervolgens het high byte wordt gevuld. Het schema is gegeven in fig. 21. De deelflipflop wordt omgezet, telkens als een karakter klaar is. Dit kan eventueel worden belet door "OK" laag te maken. Hiermee kunnen karakters uit de datastroom worden gehaald. Door het omslaan van de flipflop wordt het karakter overgenomen in dat deel van register DATA-B', dat is verbonden met de negatief gaande uitgang.

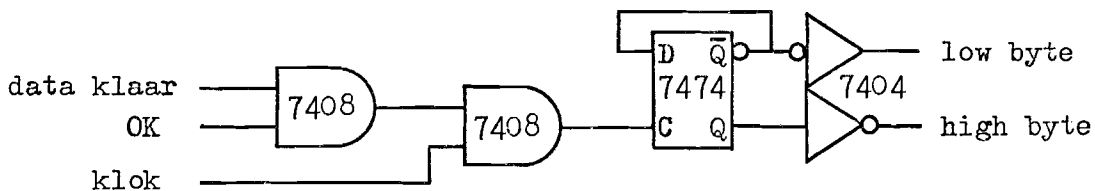


Fig. 21: Het schema van de distributor.

#### 4.3.10. De foutendetektie

In het interface wordt op een aantal plaatsen het optreden van fouten gedetekteerd. Deze fouten zijn:

- gebrek aan data. Dit treedt op als er tijdens het zenden niet op tijd data vanuit de Unibus wordt aangevoerd.
- data overrun. Dit gebeurt als de ontvangen data niet snel genoeg over de Unibus wordt afgevoerd.
- loss of carrier. Dit betekent, dat tijdens het ontvangen van data de draaggolf op de lijn wegvalt.

Een vierde fout, nl. een foute verticale pariteit van een ontvangen ASCII-karakter, wordt momenteel nog niet met hardware gedetekteerd. Dit behoort tot de in hoofdstuk 6 besproken mogelijke uitbreidingen.

Bij synchroon transport moeten de karakters onmiddellijk na elkaar verzonden worden. Als register DATA-A niet op tijd van data wordt voorzien, wordt de continuïteit verbroken. Telkens als er een laadpuls voor het schuifregister komt, wordt gekeken of register DATA-A' leeg is. Is dit het geval, dan wordt het bit "gebrek aan data" in het statusregister voor zenden opgezet.



Als het schuifregister een karakter heeft verzameld en register DATA-B' is nog vol omdat de data niet snel genoeg kan worden afgevoerd, dan zal het nieuwe karakter over het low byte in DATA-B' gezet worden. Om dit verlies van data te detekteren, wordt de bij DATA-B' behorende flipflop bekeken op het ogenblik dat een karakter is verzameld. Is op dat moment DATA-B' nog vol, dan wordt in het statusregister voor ontvangen een foutindikatie gegeven.

Loss of carrier is de situatie die optreedt als de draaggolf op de lijn wegvalt, terwijl de flipflop van de karaktersynchronisatie aangeeft, dat er data wordt ontvangen. Dit betekent, dat het contact met de zender is verbroken. Hierdoor gaat data verloren. Er wordt dan ook een foutmelding gegeven in het statusregister voor ontvangen. Er moet opnieuw karaktersynchronisatie plaatsvinden, dus de schakeling hiervoor wordt gereset.

#### 4.4. DE STATUSREGISTERS

Vanuit de Unibus kan de toestand van interface en modem in de gaten worden gehouden door statusbits in de beide statusregisters. Deze bits zijn zoveel mogelijk volgens de voorkeursindeling van DEC over de twee registers verdeeld.

Binnen het interface wordt uitgemaakt of er interrupts zullen worden aangevraagd. De afhandeling van de aanvragen gebeurt door de Interrupt Control van DEC. Deze module kan busrequests indienen op twee verschillende niveaus. Door een wijziging is het tevens mogelijk om bij een interrupt vier verschillende interruptvektoren op te geven.

4.4.1. De indeling

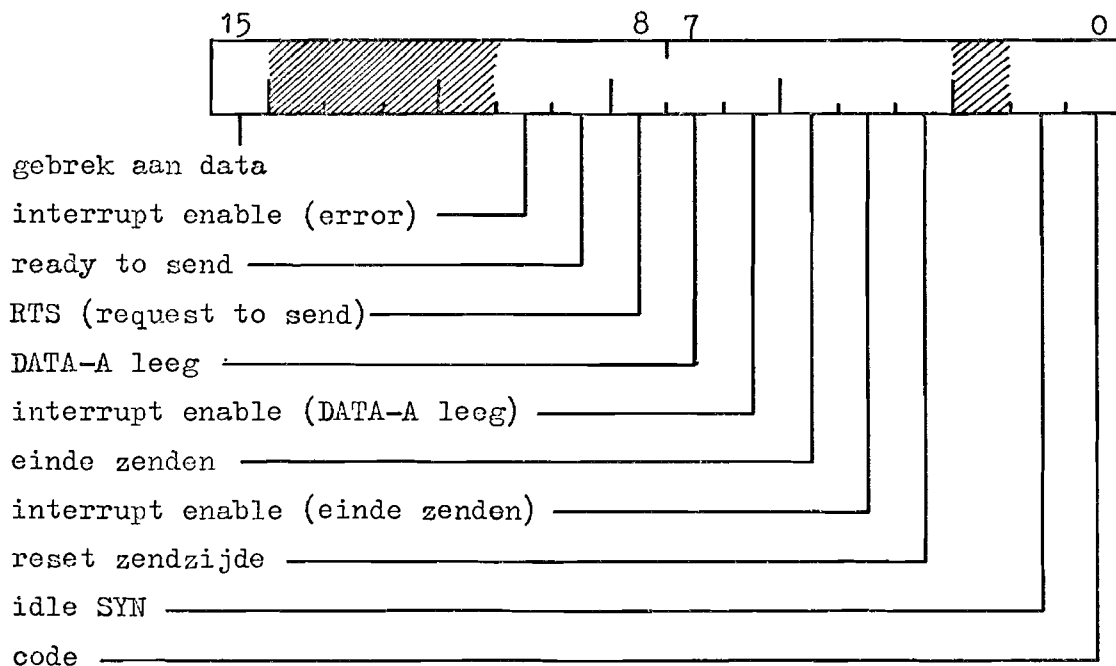


Fig. 22: De indeling van het statusregister zenden

In fig. 22 is de indeling van het statusregister voor zenden gegeven. De betekenis van de verschillende bits is als volgt:

bitnr.	naam	funktie
15	gebrek aan data	wordt geset als register DATA-A niet op tijd wordt geladen. Wordt gecleared door init en reset zenzijde. Read/write C.
10	int. enable	als dit bit opstaat, veroorzaakt "gebrek aan data" een interrupt. Wordt gecleared door init. Read/write.
9	ready to send	geeft de toestand van het signaal "ready to send" van de modem. Read only.
8	RTS	zodra dit bit wordt opgezet, begint het schuifregister data te verzenden. Wordt het afgezet, dan wordt eerst alle nog in de registers aanwezige data verstuurd, voordat het zenden wordt beëindigd. Wordt gecleared door init en reset zenzijde. Read/write.
7	DATA-A leeg	Dit bit wordt door de sturing voor dubbel bufferen opgezet zodra register DATA-A leeg wordt. Wordt opgezet door init en reset zenzijde. Read only.

bitnr.	naam	funktie
6	int. enable	als dit bit opstaat, veroorzaakt het opkomen van "DATA-A leeg" een interrupt. Wordt gecleared door init. Read/write.
5	einde zenden	dit bit wordt door de interface opgezet zodra, na het afzetten van RTS, alle data is verzonden. Wordt gecleared door init en reset zenzijde. Read/write 0.
4	int. enable	als dit bit opstaat, veroorzaakt het opkomen van "einde zenden" een interrupt. Wordt gecleared door init. Read/write.
3	reset zenzijde	reset de zenzijde van het interface tot "power up"-toestand. De interrupt enable bits blijven echter ongemoeid. Write only.
1	idle SYN	indien dit bit opstaat, worden in ASCII-mode bij gebrek aan data SYN-karakters in de datastroom gevoegd. Wordt gecleared door init en reset zenzijde. Read/write.
0	code	dit bit geeft de code van de in DATA-A geplaatste karakters t.b.v. de pariteitsgeneratie. Wordt gecleared door init en reset zenzijde. Read/write.

N.B. De idle SYN mogelijkheid is nog niet geïmplementeerd. Zie hiervoor hoofdstuk 6.

---

Het statusregister voor de ontvangzijde is ingedeeld als in fig. 23 is weergegeven.

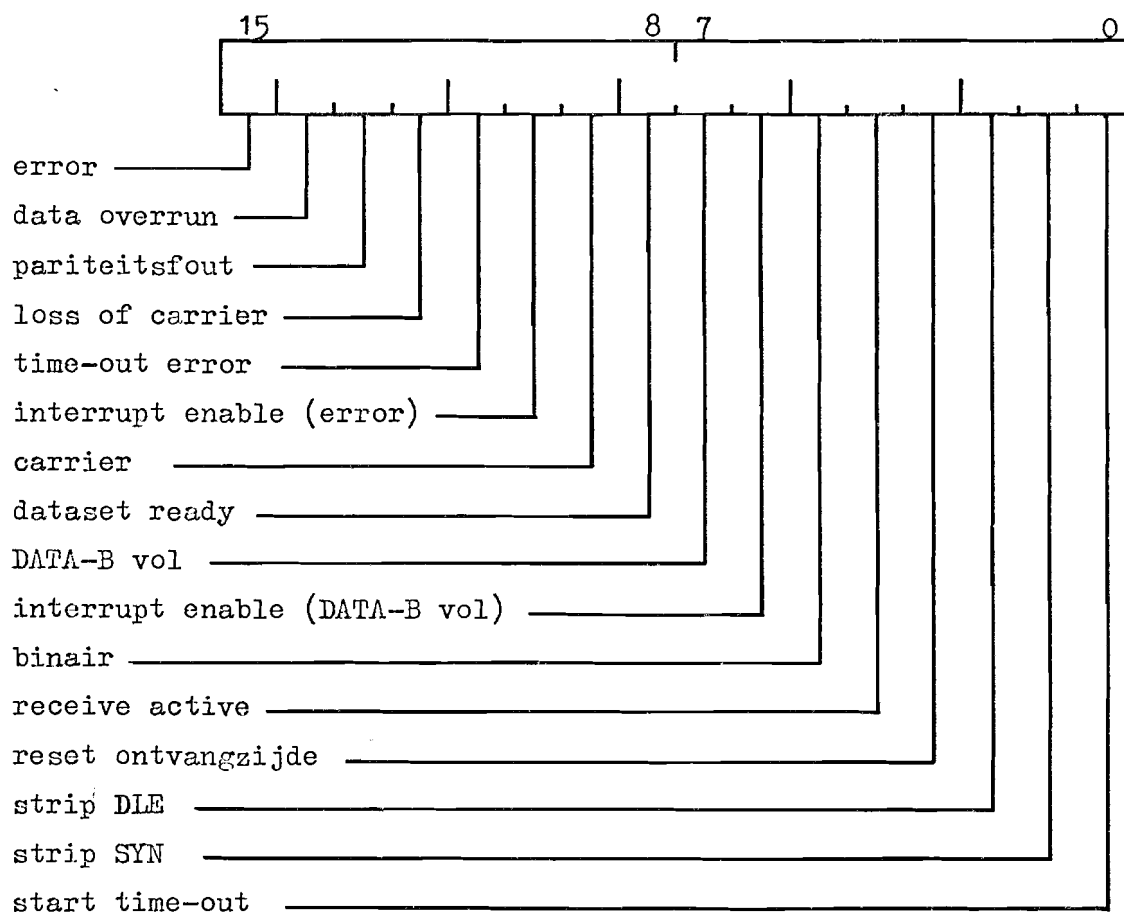


Fig. 23: De indeling van het statusregister ontvangen.

De betekenis van de verschillende bits is als volgt:

bitnr.	naam	funktie
15	error	logische 'of' van bits 14 t/m 11.
14	data overrun	dit bit wordt geset als register DATA-B niet op tijd wordt gelezen. Een "1" op deze plaats betekent dat minstens één datawoord verloren is gegaan. Wordt gecleared door init en reset ontvangzijde. Read/write 0.
13	pariteitsfout	dit bit wordt opgezet als tijdens de ontvangst van ASCII-karakters een foute verticale pariteit gevonden wordt. Wordt gecleared door init en reset ontvangzijde. Read/write 0.

bitnr	naam	funktie
12	loss of carrier	dit bit wordt opgezet als het modemlijntje "carrier detector" hoog wordt terwijl "receive active" opstaat. Wordt gecleared door init en reset ontvangzijde. Read/write 0.
11	time-out error	wordt door het interface geset, als de time-out cyclus verloopt zonder dat data in register DATA-B wordt ontvangen. Wordt gecleared door init en reset ontvangzijde. Read/write 0.
10	int. enable	als dit bit "1" is veroorzaakt "error" een interrupt. Wordt gecleared door init. Read/write.
9	carrier	geeft de toestand van het modemlijntje "carrier detector". Een "1" betekent dat er een carrier ontvangen wordt. Read only.
8	dataset ready	dit bit geeft de toestand van het modemlijntje "dataset ready", wat aangeeft of de modem al of niet voedingsspanning heeft. Read only.
7	DATA-B vol	als dit bit "1" wordt bevindt zich een ontvangen woord in het register DATA-B. Als dat register niet op tijd gelezen wordt, wordt de data overrun vlag geset. Dit bit wordt "0" door: een Unibus initpuls reset ontvangzijde lezen van DATA-B Read only.
6	int. enable	als dit bit "1" is veroorzaakt het volraken van DATA-B een interrupt. Wordt gecleared door init. Read/write.
5	binair	dit bit wordt door de hardware opgezet, als er binaire data is ontvangen. Wordt gecleared door init en reset ontvangzijde. Read/write 0.
4	receive active	receive active wordt opgezet zodra karakter-synchronisatie wordt verkregen. Het schrijven van een "0" op deze plaats betekent: stop ontvangst, wacht op nieuw synchronisatiekarakter. Wordt gecleared door init en reset ontvangzijde. Read/write 0.
3	reset ontvangzijde	wordt dit bit geset dan worden de sturing voor dubbel bufferen, codedetectie en karaktersynchronisatie gereset. Write only.

bitnr	naam	funktie
2	strip DLE	als dit bit opstaat worden DLE-karakters, waar geoorloofd, uit de datastroom gezeefd. Wordt gecleared door init. Read/write.
1	strip SYN	als dit bit "1" is worden SYN karakters, waar mogelijk uit de datastroom gezeefd. Wordt gecleared door init. Read/write.
0	start time-out	door het opzetten van dit bit wordt de time-out cyclus gestart. Wordt gecleared door init en reset ontvangzijde. Read/write.

N.B. De funktiebits 13, 5, 2, en 1 zijn nog niet geïmplementeerd.  
Zie hiervoor hoofdstuk 6.

De time-out cyclus wordt verkregen door een monostabiele multivibrator met een pulstijd van ca. 1 sec. De cyclus wordt gestart door het opzetten van bit 0 in het statusregister voor ontvangen en wordt beëindigd doordat register DATA-B volraakt, of door een reset- of initpuls. Wordt de cyclus geheel voltooid dan wordt het foutbit "time-out error" gezet.

#### 4.4.2. De interruptbehandeling

Er zijn vijf kondities waaronder een interrupt kan worden aangevraagd. Dit zijn:

- het register DATA-A is leeg,
- het register DATA-B is vol,
- het zenden is beëindigd,
- de foutkonditie "gebrek aan data" is opgetreden en
- er is een fout gesignaleerd tijdens het ontvangen.

Elk van deze kondities heeft zijn eigen interrupt enable bit. De eerste twee kondities geven aanleiding tot een interruptaanvraag op prioriteitsniveau 7, de laatste drie op niveau 4. Aanvragen op hetzelfde niveau worden na elkaar ingediend bij de interrupt module. Aanvragen op verschillende niveaus kunnen wel tegelijkertijd worden ingediend. De schakelingen die hiervoor zorgen, zijn gegeven in fig. 24.

Een aanvraag voor een bepaald niveau blokkeert andere aanvragen voor dat niveau, zolang de veroorzakende konditie blijft bestaan. Wordt als gevolg van het toekennen van de interrupt de oorzaak weggenomen dan kan na een kleine vertraging de volgende aanvraag worden afgehandeld. De aanvragen worden verwerkt door de interrupt module.

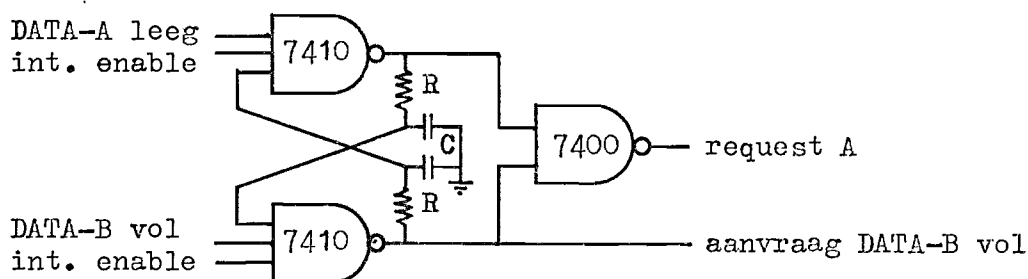


Fig. 24: De schakelingen voor het in volgorde doorgeven van de interruptaanvragen.

De M7821 Interrupt Control module van DEC bevat de circuits en logika, nodig om het beheer over de Unibus te krijgen (busmaster te worden). Hiervoor zijn twee volledig gescheiden circuits beschikbaar. Het blokdiagram van de module is in fig. 25 weergegeven. Als de requestlijntjes van één van beide circuits hoog worden, wordt op de busrequestlijn een aanvraag ingediend. Als deze aanvraag door de processor wordt ingewilligd, maakt deze de bus grant lijn hoog. Hierop zendt de interruptmodule een "selection acknowledge" (SACK) en trekt het zijn busrequest in. Dan wacht de module tot de huidige busmaster de Unibus vrijgeeft (BBSY en SSYN worden hoog). Zodra dit gebeurt, bezet de module onmiddellijk de Unibus door BBSY laag te maken. Als teken voor de interface, dat het beheer over de Unibus is verkregen, wordt "master" laag.

Nu kan een interrupt worden gegenereerd. Daarvoor moet eerst het interruptvektor-adres worden bepaald. Hierbij worden de foutmeldingen van zend- en ontvangzijde samengenomen, zodat in totaal vier verschillende adressen nodig zijn. Omdat de standaard inter-

rupt module slechts twee vektoradressen kan geven, moest hiervoor een wijziging worden aangebracht. De schakeling voor het bepalen van het vektoradres is gegeven in fig. 26. M.b.v. het signaal "master A" wordt het adresbit 3 bepaald. De interrupt veroorzakende konditie zorgt dan voor de bepaling van bit 2.

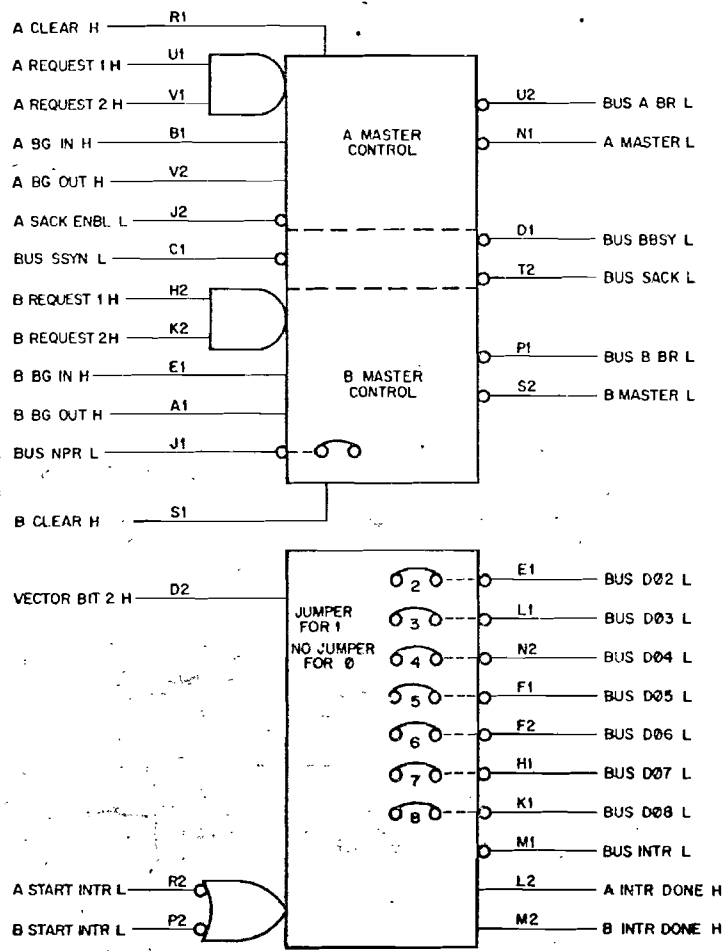


Fig. 25: Het blokschema van de interruptmodule.

Nu wordt "start interrupt" hoog gemaakt. Dit heeft tot gevolg, dat het vektoradres op de datalijnen van de Unibus wordt gezet terwijl tegelijkertijd het INTR-lijntje wordt geactiveerd. Als de processor het adres heeft overgenomen, wordt een SSYN gestuurd. Zodra de module deze ontvangt, is deze interrupt afgehandeld en kan een volgende worden aangevraagd.



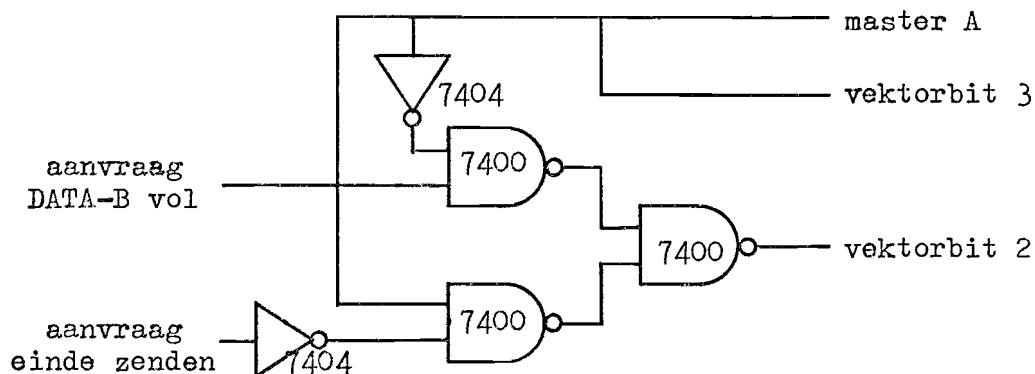


Fig. 26: Het bepalen van het interruptvektor-adres.

De interruptvektoren bevinden zich op de volgende adressen:

- DATA-B vol 320
- DATA-A leeg 324
- Foutkondities 330
- Einde zenden 334

#### 4.5. DE PRAKTISCHE UITVOERING

De interface is opgebouwd uit zeven printen. Elk hiervan vormt een functionele eenheid. De indeling is als volgt:

1. Adresselektor Bevat de logika voor de werking als slave.
2. Interrupt module Bevat de logika voor het verkrijgen van het beheer over de Unibus en het genereren van interrupts.
3. Transceivers Bevat 16 stel Unibusdrivers en -receivers.
4. Print zenzijde Bevat de registers DATA-A en DATA-A', de sturing voor dubbel bufferen, de multiplexor en de detektie van gebrek aan data.
5. Schuifregisters Bevat de beide schuifregisters, de schakelingen voor start en einde zenden en karaktersynchronisatie en de niveauomzetters.
6. Print ontvangzijde Bevat de registers DATA-B en DATA-B', de sturing voor dubbel bufferen, de distributor en de detektie van data overrun.

7. Statusregisters    Bevat de beide statusregisters en de schakelingen voor het doorgeven van de interruptaanvragen, het bepalen van het vektoradres en de time-out cyclus.

Deze zeven printen zijn ondergebracht in een systemunit in de extensionbox van de PDP 11/20. De verbindingen tussen de verschillende printen en met de Unibus zijn op de achterzijde van de systemunit aangebracht d.m.v. de wire-wrap techniek. De modem is bevestigd in de achterdeur van het rek, waarin zich de extensionbox bevindt.

Gedetailleerde dokumentatie over de praktische uitvoering, zoals schema's van de printen, lay-out, stuklijsten en bedradingslijsten zijn te vinden in lit. 15.

## 5. DE INTERFACE DRIVER

Een driver is een geheel van routines, nodig om in- en uitvoertransporten te verzorgen tussen de PDP 11 en één van zijn randapparaten. Het basisprincipe van alle drivers is, dat ze een gemeenschappelijk aanzien moeten hebben voor de routines die er gebruik van maken, zodat een device-onafhankelijke aanroep mogelijk is. Een uitgebreide beschrijving van PDP 11 drivers wordt gegeven in lit. 16. Het eerste deel van elke driver bestaat uit een tabel, die in een standaard formaat informatie bevat over de aard en capaciteiten van het randapparaat. Een programma kan deze tabel gebruiken, ongeacht welk randapparaat wordt aangeroepen.

Een driver bestuurt zijn apparaat onder het PDP 11 interruptysteem. Als een driver wordt aangeroepen door een programma, start deze enkel de gevraagde actie door de hardware registers van het apparaat op de juiste wijze te vullen. Door interrupts wordt de voortgang in de gaten gehouden. Als de gevraagde actie geheel is uitgevoerd, wordt het aanroepende programma hiervan op de hoogte gebracht.

Voordat een programma een driver aanroept moet het een lijst maken in een standaard formaat. In deze lijst staan de parameters die de driver nodig heeft. De verbinding tussen het programma en de driver loopt via het eerste woord van de drivertabel. Hierin zet het programma een wijzer naar de parameterlijst. De driver gebruikt deze wijzer om parameters op te halen en om eventueel informatie in de lijst te plaatsen.

Foutmeldingen van het randapparaat kunnen op twee manieren worden behandeld. Voor sommige fouten kan een oplossing worden geprogrammeerd. De driver kan in dat geval zelf proberen de fout te herstellen of er wordt een foutmelding aan het programma doorgegeven. Voor andere fouten is externe actie nodig, bijv. van de operateur. In dat geval wordt een gemeenschappelijke foutroutine aangeroepen, die

zorgt voor het uittypen van een foutmelding op het operatorconsole.

De driver die het transport tussen de PDP 11 en het Burroughs B6700 systeem moet verzorgen, moet dus aanvragen voor het transport van boodschappen zelfstandig kunnen verwerken. Dit betekent, dat deze driver geheel volgens de lijndiscipline het beheer over de lijn moet verkrijgen, alvorens de boodschap kan worden verzonden. Pas als een bevestiging van een goede ontvangst is binnengekomen kan de opdracht als uitgevoerd worden beschouwd.

Voor het eigenlijke transport zijn twee interrupt service routines gemaakt. De routine voor het zenden heeft de volgende taken:

- het transport van data uit het geheugen naar het interface register DATA-A.
- het bepalen van de code van de verzonden data en het overeenkomstig bijhouden van het codebit in het statusregister.
- waar nodig het invoegen van DLE's bij het verzenden van binaire data.
- het genereren van het CRC-karakter.

De routine voor het ontvangen heeft als taken:

- het transport van data van het interfaceregister DATA-B naar het geheugen.
- het bepalen van de code van de ontvangen data.
- het uitzeven van SYN's en DLE's voor zover mogelijk.
- het controleren van de verticale pariteit van ASCII-karakters.
- het genereren van het CRC-karakter.

De verdere taak van de driver bestaat uit het volgens de lijndiscipline reageren op optredende fouten. In de meeste gevallen betekent dit het herhalen van boodschappen of het vragen daarom. Als een ernstiger fout optreedt wordt dit aan het aanvragende programma meegedeeld via het statuswoord in de parameterlijst. Eventueel wordt een foutmelding op het console uitgetypt.

## 6. MOGELIJKE UITBREIDING VAN DE HARDWARE

Eerder in dit rapport werd al opgemerkt, dat het mogelijk is om een aantal taken, die nu met software worden uitgevoerd, over te brengen naar de hardware. Als mogelijkheden zijn onderzocht:

Aan de zenzijde:

- het invoegen van synchronisatiesequenties (SYN, DLE SYN) bij gebrek aan data.
- het invoegen van DLE's bij het zenden van binaire data.

Aan de ontvangzijde:

- het bijhouden van de code (ASCII of binair).
- het uitzeven van de synchronisatiesequenties uit de datastroom.
- het uitzeven van DLE's bij ontvangst van binaire data.
- het controleren van de verticale pariteit van ASCII-karakters.

Algemeen:

- het genereren van het CRC-karakter, zowel bij het zenden als bij het ontvangen.
- het overgaan van interrupt service routines op een Direct Memory Access interface.

De mogelijke realisering van al deze punten zal hierna kort worden besproken.

Het invoegen van karakters in de datastroom kan gebeuren door extra poortgroepen aan te sluiten op de "wired-or" verbinding tussen multiplexor en schuifregister zenzijde. Als een karakter moet worden ingevoegd, moeten de poortgroepen van de multiplexor d.m.v. het lijntje "onderdrukking" worden dichtgezet en de groep voor het in te voegen karakter open. Tevens moet gezorgd worden dat de laadpuls voor het schuifregister de flipflop van de multiplexor niet omzot. Dit kan door het lijntje "data change OK" laag te houden. Op deze manier kan op een willekeurige plaats in de datastroom een willekeurig aantal karakters worden ingevoegd.

In eerste instantie kan deze mogelijkheid worden gebruikt om bij gebrek aan data synchronisatiesequenties in te voegen. Als het signaal "gebrek aan data" van de print zenzijde hoog wordt, is er bij een seinsnelheid van 9600 baud nog ca. 50  $\mu$ s tijd om de nodige actie te ondernemen. Om te bepalen of SYN dan wel DLE SYN ingevoegd moet worden, is het nodig de code van het laatst gezonden karakter te onthouden. Is deze code ASCII, dan kan het SYN-karakter (inclusief pariteitsbit) op de "wired-or" lijnen worden gezet totdat "gebrek aan data" weer afvalt. Tijdens het zenden van binaire data moet afwisselend DLE en SYN worden ingevoegd, hetgeen wat meer logika vereist.

DLE's in binaire tekst moeten worden verdubbeld om ze te onderscheiden van DLE's in controlesequenties. Een extra DLE kan op zeer eenvoudige wijze worden ingevoegd door de klokpuls voor de flipflop van de multiplexor m.b.v. het lijntje "data change OK" éénmaal te blokkeren. Als er voor wordt gezorgd, dat DLE's van controlesequenties bij het laden van register DATA-A de code "ASCII" meekrijgen, moet dit verdubbelen gebeuren bij DLE's met code "binair".

Het bijhouden van de code tijdens het ontvangen gebeurt door op de datalijnen tussen schuifregister ontvangzijde en de print ontvangzijde te letten op het voorkomen van DLE's. Er is een flipflop nodig om de code te onthouden. Deze staat in de stand "ASCII" zolang er niet wordt ontvangen (er geen karaktersynchronisatie is bereikt). De ontvangst van een DLE in ASCII zet de flipflop in de stand "binair". Wordt in de binaire data een DLE gedetekteerd, dan is de situatie meer gekompliceerd. Of de code in dit geval verandert hangt nl. af van het volgende karakter. Is dit een DLE of een SYN dan blijft de code binair. Is het een controlekarakter, dan gaat de code terug naar ASCII. Er is dus een flipflop nodig om te onthouden dat er een DLE is gedetekteerd. Voor het bepalen van het volgende karakter kan gebruik worden gemaakt van de SYN-detektor op de print schuifregisters.

Als de code op bovenstaande manier tijdens het ontvangen wordt bijgehouden, is het ook mogelijk om de verticale pariteit van de ASCII-karakters te controleren. Hiervoor kan de 74180 parity checker/generator worden gebruikt. Verder kunnen ook op eenvoudige wijze de synchronisatiesequenties en de DLE's uit de datastroom worden gezeefd. Hiervoor wordt gebruik gemaakt van het lijntje "OK" van de print ontvangzijde. Wordt dit lijntje laag gehouden, dan worden de signalen voor de distributor en daarmee de laadpulsen voor register DATA-B' geblokkeerd (zie fig. 21).

Het genereren van een CRC-karakter met hardware is op zich redelijk eenvoudig (zie lit. 14). Dit gebeurt m.b.v. een schuifregister, waarin de seriële data wordt opgenomen. De problemen die zich voordoen zijn:

- niet alle karakters van de datastroom mogen worden opgenomen in de CRC, zodat er een mechanisme nodig is voor het stoppen en starten van de generatie.
- bij het ontvangen is er een tijdsvertraging tussen het ogenblik waarop het eerste bit van een karakter binnenkomt en het moment waarop aan de hand van het parallelle karakter kan worden bepaald of dat karakter wel of niet mag worden opgenomen. Dit kan worden opgelost door een 9 bits buffer-schuifregister.
- het laatste karakter dat nog moet worden opgenomen is ETX. Bij het zenden moet het CRC-karakter echter onmiddellijk na ETX worden overgestuurd. Hiervoor is een speciale schakeling nodig.
- teneinde het gegenereerde CRC-karakter te kunnen vergelijken met het ontvangen CRC-karakter, moet de inhoud van het generatieschuifregister voor de software toegankelijk zijn. Hiervoor moet de capaciteit van de adresselektor worden uitgebreid. Eventueel kan geprobeerd worden deze vergelijking met hardware uit te voeren.

Teneinde de interruptservice routines uit te schakelen, kan worden

overgegaan op non-processor transport met een Direct Memory Access interface. De taak van het DMA-interface zou moeten zijn het zelfstandig verzenden of ontvangen van blokken data. De lijndiscipline zal nog met software gehandhaafd worden. De winst die wordt bereikt door over te gaan op een DMA-interface blijkt uit de volgende ruwe berekening. Bij een seinsnelheid van 10 kbaud moet gemiddeld eens in de 1,6 ms een datawoord over de Unibus worden getransporteerd. Een DMA-interface heeft hiervoor één buscyclus van  $1\ \mu\text{s}$  nodig. De bezettingsgraad is hier dus ca. 0,06 %. De kortst mogelijke interrupt service routine, die slechts het woord transporteerd en een tellertje bijhoudt, heeft minstens  $25\ \mu\text{s}$  nodig, d.w.z. een bezettingsgraad van 1,5 %.

Om het interface als busmaster datatransporten te kunnen laten verzorgen, is nodig:

- een DMA-sturing voor het besturen van de Unibus tijdens de datatransporten.
- een register waarin het adres wordt bijgehouden van de geheugenplaats waar de data vandaan moet komen of naartoe moet worden gebracht.
- een register waarin het aantal nog te transporteren woorden (bytes) wordt bijgehouden.
- een mogelijkheid voor het bepalen van het einde van een boodschap tijdens het ontvangen.

Eindhoven, juni 1973

D.M.P. Woensdregt



LITERATUURLIJST

1. PDP 11/20 processor handbook.  
Digital Equipment Corporation, 1972.
2. PDP 11 peripherals and interfacing handbook.  
Digital Equipment Corporation, 1972.
3. B6700 Data Communications.  
RC-bulletin 28, Technische Hogeschool Eindhoven, 1972.
4. B6700 Data Communications Functional Description.  
Information manual 5000060, Burroughs.
5. B6700 Data Communications Processor.  
Reference manual 1054385, Burroughs.
6. B6700 Data Communications extended Algol.  
Information manual 5000052, Burroughs.
7. B6700 Remote Job Entry (RJE) System.  
Information manual 50000300, Burroughs.
8. B6700 Command and Edit (CANDE) Language.  
Information manual 50000318, Burroughs.
9. B6700 Network Definition Language.  
Information manual 5000078, Burroughs.
10. Daleman, R.C. ; Een flexibel 600-9600 baud modem voor in-plant datatransmissie. THE afstudeerrapport, Eindhoven, 1972.
11. Simonis, F. ; Cretha, Modem 9600 baud. Handleiding RC 153'9, Technische Hogeschool Eindhoven, 1973.
12. Communications procedures.  
Burroughs System Standard 12849006, 1971.
13. Binary Synchronous Communications Procedures.  
Burroughs System Standard 12849022, 1971.
14. Ruiterkamp, W. ; Communicatie voor een verkeersregeling.  
THE afstudeerrapport, Eindhoven, 1972.
15. Woensdregt, D.M.P. ; Schema's en tekeningen van B6700 interface.  
Technische Hogeschool Eindhoven, 1973.
16. PDP 11 device driver package.  
Information manual DEC-11-NIZB-D, Digital Equipment Corporation.