

LÆREBOG  
I  
KODNING FOR GIER  
BIND II

1. udgave

af

Chr. Gram

REGNECENTRALEN, KØBENHAVN

September 1963





## FORORD

I september 1962 udkom tredje udgave af første del af lærebogen i kodning for GIER. Den indeholdt imidlertid næsten intet om ind- og udlæsning og slet intet om HJÆLP-systemet, og derfor planlagde vi allerede dengang en fortsættelse.

Nærværende anden del af lærebogen forsøger at beskrive hjælpeprogrammerne og indlæseprogrammet SLIP sådan som de tager sig ud for brugeren, d.v.s. koderen. Disse programmer kan gøre umådelig stor nytte under kodningen og afprøvningen af et program, og jeg tror at det er umagen værd at sætte sig grundigt ind i brugen af HJÆLP og SLIP. Derfor er konventionerne for disse omtalt meget detaljeret og belyst med en del eksempler (i kapitel 11, 12 og 13). Derimod vil læseren lede forgæves efter oplysninger om selve koden og den indre struktur for disse programmer.

Desuden indeholder denne bog et kapitel om GIER-bibliotekssekvenser og et par kapitler om tromlelageret, de ydre enheder og manøverbordets knapper og lamper, som altsammen var yderst skematisk beskrevet i Lærebog I. Blandt de ydre enheder er - udover standardudstyret - medtaget linieskriveren og hulkortlæseren, men hverken procesenheden, bufferlageret eller magnetbåndstationer.

I teksten henvises der flittigt til første del af lærebogen under navnet Lærebog I, ligesom kapitlerne er nummereret i fortsættelse af Lærebog I; dette er blandt andet gjort for at under-

strege at de to dele af lærebogen bør betragtes som en helhed. Anden del lider iøvrigt ligesom første del af at være skrevet som en blanding af en lærebog og en håndbog i GIER-kodning; jeg håber imidlertid at eksemplerne i teksten og øvelserne bag i bogen vil lette tilegnelsen af stoffet, som mange steder tynges af håndbogens opremsning af et mylder af detaljer.

Det er mig en glæde at takke alle, såvel på Regnecentralen som udenfor, der har ydet mig hjælp under arbejdet med denne bog. Men frem for nogen vil jeg gerne takke H.Isaksson og P.Mondrup, som har gennemlæst store dele af manuskriptet, foreslået mange forbedringer og med stort tålmod besvaret et utal af spørgsmål. Isaksson har især hjulpet mig med kapitlerne vedrørende GIERS opbygning, mens Mondrup, som har kodet hele HJÆLP-SLIP-systemet, har bidraget til kapitlerne om disse programmer.

Christian Gram

## INDHOLDSFORTEGNELSE

	side
FORORD	
9. TROMLELAGER OG YDRE ENHEDER	1
9.1 Indledning	1
9.2 Tromlelageret	1
9.2.1 Ferritlageradressen	2
9.2.2 Kanaladressen; låsede kanaler	3
9.2.3 Simultan transport; kontrol af kanal aflæsning	6
9.2.4 Kanalhop	9
9.3 Strimmellæser, perforator og skrivemaskine	11
9.3.1 Strimmellæser	11
9.3.2 Perforator	13
9.3.3 Skrivemaskine	14
9.3.4 Operationstider	17
9.4 Anelex linieskriver	17
9.4.1 Margin og tabulator	20
9.4.2 Carriage Return, formularskift og papirformater	21
9.4.3 Linieskriverens kode	21
9.5 Hulkortlæser	22
9.5.1 Mekanisk opbygning	23
9.5.2 Tilslutning til GIER	28
9.5.3 Eksempler på indlæsning	30
10. MANØVREBORD. REGISTRE	34
10.1 Hovedmanøvreboard	34
10.1.1 Lamper på manøvreboardet	34
10.1.2 Start- og Stop-knapper	38
10.2 Betjening af hovedmanøvreboard	40
10.2.1 Stop	40
10.2.2 Start	41
10.2.3 Udførelse af enkelte ordrer	41
10.3 HP-knappen med videre	45

	side
11. INDLÆSNING. SLIP	48
11.1 Indledning	48
11.2 Indlæseprogrammer; et eksempel	49
11.3 Indlæseprogrammet SLIP	55
11.3.1 Indledning. Disponibelt lager	55
11.3.2 Linier	57
11.4 Indlæsning af ordrer med SLIP	59
11.4.1 Grundoperation, variant og indikatoroperation	59
11.4.2 Adressedel	61
11.4.3 Talletal	71
11.5 Indlæsning af tal og tekst med SLIP	73
11.5.1 Talindlæsning	73
11.5.2 Tekstindlæsning	75
11.6 Styrelinier og blokke i SLIP	77
11.6.1 Løbende adresse og løbende kanalnummer	77
11.6.2 Blokke	78
11.6.3 Definitionslinie og etikette	84
11.6.4 Tromleblokke	88
11.6.5 Administratorer	92
11.6.6 Programmer	98
11.7 Indhop og udhop i SLIP	99
11.7.1 Manuelt styret indlæsning	99
11.7.2 Programstyret indlæsning	100
11.7.3 Udhop	101
11.8 Kontroludskrifter fra SLIP	102
11.8.1 Fejludskrift	102
11.8.2 Kontroludskrift	104
11.9 Syntax for SLIP-kode	106
11.9.1 Program og blok	106
11.9.2 Linie	106
11.9.3 Ordrelinie	107
11.9.4 Adressedel og talletal	107
11.9.5 Tallinie og tekstlinie	108
11.9.6 Styrelinier	109
12. UDLÆSNING	110
12.1 Indledning	110
12.2 Taltrykning med sekvensen i HJÆLP	111
12.2.1 Funktion	111
12.2.2 Placering af sekvensen; indhop og udhop	112
12.2.3 Trimningsparametre	113
12.2.4 Skalafaktor	120
12.2.5 Eksempler på trimninger; specielle muligheder	122

	side
12.3 Teksttrykning med sekvensen i HJÆLP	124
12.3.1 Funktion	124
12.3.2 Placering af sekvensen; indhop og udhop	125
12.3.3 Eksempler på benyttelse af sekvenserne i HJÆLP	126
13. HJÆLPEPROGRAMMER. HJÆLP	129
13.1 Kontrolfunktioner	129
13.2 HJÆLPs centrale administration	130
13.2.1 HP-knappen	132
13.2.2 Udhop	136
13.2.3 Programmeret indhop	138
13.2.4 Flydende overløb	139
13.3 Styring af hjælpeprogrammer	140
13.3.1 Hjælpeinformationslinie. Styreparametre	140
13.3.2 Skrivemaskine- eller strimmelstyret udførelse	141
13.3.3 Programstyret udførelse	142
13.3.4 Rettelser	144
13.4 Standard-hjælpeprogrammer	146
13.4.1 Placering af hjælpeprogrammer	146
13.4.2 "kontrol" og "start"	148
13.4.3 "tryk"	152
13.4.4 "kompud"	160
13.4.5 "gem", "hent" og "sam"	167
13.4.6 "ret"	172
13.4.7 "hp ind"	174
13.4.8 "hp ud"	181
13.4.9 "slip"	182
13.4.10 Andre hjælpeprogrammer	184
13.4.11 Konventioner for kodning af hjælpeprogrammer	185
13.5 Fejludskrifter fra HJÆLP	187
13.5.1 Gal styreinformation	187
13.5.2 "tomt hp" og "sumfejl"	188
13.5.3 Fejl på de låsede kanaler	189
14. BIBLIOTEKSSEKVENSER	190
14.1 Indledning	190
14.1.1 Formålet	190
14.1.2 Krav til bibliotekssekvenser	191
14.2 GIER-sekvensbeskrivelser	191
14.2.1 Oversigt	192
14.2.2 Funktion og Metode	193
14.2.3 ALGOL-beskrivelse og Maskinkode	194
14.2.4 Testmetode	195
14.3 GIER-sekvensstrimler	196

	side
15. EKSEMPEL OG ØVELSER	197
15.1 Indledning	197
15.2 Eksempel på simultan tromletransport og beregning	198
15.3 Øvelser	200
15.3.1 Tromle og ydre enheder. Manøvrebord	200
15.3.2 Indlæsning med SLIP	202
15.3.3 Udlæsning	205
15.3.4 Hjælpeprogrammer	206
15.3.5 Bibliotekssekvenser	209
16. OVERSIGTER	211
16.1 Korrespondancen mellem talværdier og typografiske symboler	212
16.2 Indhop og layout ved taltrykning	213
16.3 Understregede bogstaver i SLIP	214
16.4 Fejludskrifter fra HJÆLP og SLIP	215
16.5 Hjælpeprogrammernes virkning og parametertyper	216

FOTO af Manøvrebord og HP-knap

STIKORDSREGISTER



## 9. TROMLELAGER OG YDRE ENHEDER.

### 9.1 Indledning.

I afsnit 2.7 i Lærebog I er de tre enheder strimmellæser, skrivemaskine og perforator nævnt. Vi vil her omtale disse lidt nærmere samt beskrive tilslutning af en linieskriver (Anelex).

Desuden vil vi uddybe beskrivelsen af tromlelageret (smlgn. afsnit 2.3).

### 9.2 Tromlelageret.

Som nævnt i afsnit 2.3 har GIER et magnetisk tromlelager, der er organiseret i kanaler à 40 celler (hver med 42 bits). Der er normalt 320 kanaler eller spor, men GIER kan også forsynes med 960 kanaler (svarende til 3 tromler). Kanalerne er nummereret fra nr.0 til nr.319 (subsidiært til nr.959), og al overførsel til og fra ferritlageret sker kanalvis med 40 celler ad gangen. Dette kræver 2 ordrer, hvoraf den ene - en VK-ordre - angiver nummeret på den ønskede kanal, mens den anden - enten en LK-ordre eller en SK-ordre - angiver det ønskede område i ferrit-

lageret (området er altid en blok på 40 successive celler), samt om transporten skal ske til eller fra ferritlageret.

Disse ordrer behøver ingenlunde at stå umiddelbart efter hinanden, thi ordren

VK <c>

bevirker blot, at kanal nr.<c> udvælges, d.v.s. klargøres til læsning eller skrivning, og dette valg bliver stående (i tromlekanalregisteret tk), indtil der udføres en ny VK-ordre. Enhver læse- eller skrive-ordre vil derfor referere til den kanal, der er valgt med den sidst udførte VK-ordre.

#### 9.2.1 Ferritlageradresser.

Adressen i læse- eller skriveordren er altid adressen på den første af de 40 konsekutive celler i ferritlageret, der berøres af tromletransporten. Hvis denne adresse ligger mellem 0 og 983, udgør de 40 celler en kompakt blok i ferritlageret, men hvis adressen er  $\geq 984$ , indgår de sidste og nogle af de første celler i ferritlageret (ialt 40 celler), idet adresserne simpelthen beregnes modulo 1024.

#### Eksempel 9.1.

Programstumpen

VK 70

-

-

SK 1000

-

-

LK 860, VK 0

bevirker, at indholdet (inklusive mærkebits) i celle 1000 -

1023 samt 0-15 overføres til kanal 70, og derfra læses det til celle 860-899 i ferritlageret. Celle 1000-1023 og 0-15 er naturligvis uændrede, og celle 860-899 (og kanal 70) har nu blot fået samme indhold. Den afsluttende ordre VK 0 er indsat for at sikre, at GIER ikke går videre i programmet før kanaltransporten er afsluttet (se afsnit 9.2.3).

### 9.2.2 Kanaladressen; låsede kanaler.

Adressen i VK-ordren er adressen på den tromlekanal, der tilkobles: Under VK-ordrens udførelse overføres dennes adresse til tk-registeret, og når en SK- eller en LK-ordre skal udføres, bestemmer tk-registerets indhold hvilken kanal, der skal deltage i operationen. Derfor kan en VK-ordre udføres, selvom adressen er  $\geq 320$  subsidiært  $\geq 960$ , og det er først ved en efterfølgende læse- eller skriveordre, at GIER opdager, at den valgte kanal ikke eksisterer. Hvad der herefter sker afhænger af, om der er 1 eller 3 tromler, fordi tromlekredsløbene er ændret lidt i forbindelse med udvidelsen til 3 tromler. Samtidig er mulighederne for at låse visse kanaler for skrivning udvidet:

A) GIER med 1 tromle, 320 kanaler: Er indholdet i tk-registeret  $\geq 320$ , har en SK-ordre kun den virkning, at GIER venter på, at en eventuel igangværende tromletransport afsluttes. Bortset herfra er en sådan skriveordre blind (men opholder dog GIER i ca. 36  $\mu$ s, hvis adresseberegningen er simpel).

En LK-ordre med kanaladresse  $\geq 320$  vil heller ikke bevirke stop, men den har til følge, at tromleadgangen spærres, og at TR-lampen lyser permanent (som tegn på, at den paritetscheck, der udføres under læsningen af en kanal, ikke stemmer, hvorfor den samme "kanal" læses om og om igen). GIER vil da gå i stå på

den næste tromleordre, fordi tromlekredsene melder optaget.

Kanal 0 er normalt låset og kan kun frigives, ved at man slukker for maskinspændingen og fjerner et kontaktstykke fra en af kredsløbspladerne. Kanal 1-31 kan låses (og frigives) ved hjælp af en omskifter på betjeningspanelet øverst i GIER-skabet. Virkningen af at låse en kanal er, at indholdet på kanalen ikke kan ændres, altså at SK-ordrer til denne kanal er virkningsløse (i samme forstand som SK-ordrer til ikke-eksisterende kanaler). En læse-ordre fra en låset kanal udføres naturligvis helt normalt.

#### Eksempel 9.2.

På en GIER-maskine med 1 tromle vil programstumpen

```
[m] VK -1, SK 120
[m+1] LK 400
      -----
[m+17] VK 20
      -----
```

bevirke, at GIER ikke kan udføre VK-ordren i celle m+17, fordi læsningen af "kanal -1" aldrig bliver afsluttet. Lampen TR lyser, men GIER er ikke stoppet, og tryk på NORMAL START har ingen virkning; kun et tryk på RESET kan frigøre tromlekredsløbene, og først derefter vil NORMAL START få GIER til at udføre ordren VK 20 og gå videre i programmet.

GIER kan også bringes over det døde punkt ved annullering af paritetskontrollen (ved hjælp af en kontakt på betjeningspanelet øverst i GIER-skabet. Ved kontakten er anbragt en rød lampe, som lyser, når paritetskontrollen er slået fra), og maskinen vil da straks gå videre i programmet.

B) GIER med 3 tromler, 960 kanaler: Er indholdet i tk-regi-  
steret  $\geq 960$ , vil både en SK-ordre og en LK-ordre bevirke stop,  
idet

- a) GIER stopper på tærsklen til den næste ordre uden at have udført nogen transport.
- b) Fejllamperne TO og TR lyser.
- c) Klar-lampen lyser.

Ved tryk på NORMAL START kører GIER videre.

På betjeningspanelet i GIER-skabet sidder der foruden to trykknapperegistre, hver med 10 knapper, som kan indstilles binært på ethvert tal mellem 0 og 1023. Man kan nu låse et vilkårligt antal successive kanaler ved at indstille det nedre register på den første låsede kanals adresse og det øvre register på den sidste låsede kanals adresse. Dog er kanal 0 låset på samme måde som på GIER med 1 tromle.

Er de to registre indstillet på den samme adresse, er kun een kanal låset, nemlig kanalen med den pågældende adresse (samt kanal 0), og hvis adressen i det nedre register er større end adressen i det øvre register, er kun kanal 0 låset.

Læsning fra en låset kanal foregår helt normalt, men en SK-ordre med reference til en låset kanal bevirker stop, nøjagtig som hvis kanalen ikke eksisterede (se ovenfor).

Resumé: Hvis de to trykknapperegistre er indstillet på henholdsvis adresserne  $K_{min}$  og  $K_{max}$ , vil følgende ordrer bevirke stop (og reaktionerne a, b og c ovenfor):

LK-ordrer såfremt  $960 \leq tk \leq 1023$

SK-ordrer såfremt  $tk = 0, 960 \leq tk \leq 1023$

eller  $K_{min} \leq tk \leq K_{max}$ .

Eksempel 9.3.

I en GIER-maskine med 3 tromler afhænger virkningen af ordrerne

VK 1, SK 200

af trykknop-registrene: Hvis det nedre er indstillet på 0 eller 1 og det øvre på alt andet end 0, er kanal nr.1 låset, og GIER stopper efter SK-ordren uden at have skrevet på nogen kanal. Ellers udføres ordrerne normalt således, at indholdet i celle 200-239 overføres til kanal nr.1.

9.2.3 Simultan transport; kontrol af kanal aflæsning.

En kanaltransport, det være sig læsning eller skrivning, varer normalt 20 millisekunder (tiden for 1 omdrejning), men så snart transporten er sat i gang af en LK- eller en SK-ordre, fortsætter GIER med de følgende ordre i programmet simultant med, at transporten foregår mellem tromle og ferritlager. Kun hvis GIER møder en ny tromleordre - enten VK, LK eller SK - vil den vente på, at den foregående transport afsluttes. Selve SK- eller LK-ordren gennemløbes på 9  $\mu$ s (+ tiden for adresseberegningen), men udførelsen af de følgende ordre i programmet forsinkes med ca. 1 millisekund, fordi den centrale enhed beslaglægges i korte øjeblikke ad gangen under tromletransporten, ialt i 1 millisekund fordelt over de 20 millisekunder, som transporten varer.

Da transporten påbegyndes i det øjeblik, GIER møder en LK- eller en SK-ordre, afhænger det af tromlens øjeblikkelige stilling i forhold til læse- og skrivehovederne, hvilken celle på

kanalen der først aflæses (eller skrives på); derfor må GIER beregne den tilsvarende adresse i ferritlageret (inden for den blok på 40 celler, som er defineret af adressen i LK- eller SK-ordren). Derefter fortsætter læsningen fra (eller skrivningen på) de følgende kanalceller, indtil et helt omløb er fuldført; de 40 celler i ferritlageret vil således ofte ikke blive behandlet forfra, men i en cyklisk forskudt orden.

Forholdene kompliceres yderligere af den automatiske paritetskontrol under læsning fra en tromlekanal: Hver celle på tromlen har en ekstra position, der under skrivning i den pågældende celle 1-stilles eller 0-stilles, afhængig af om antallet af ettere i cellen er ulige eller lige.\*) Under læsningen undersøger GIER nu, om denne paritetscheck stemmer, og hvis det ikke er tilfældet, fortsætter GIER kanal aflæsningen, lige indtil det er lykkedes at aflæse 40 celler i træk uden fejl.

Når paritetscheck'en ikke stemmer, lyser TR-lampen på manøverbordet som tegn på tromlefejl, men den slukkes igen, hvis fejlen ophører. Hvis man ser lampen lyse, er det derfor tegn på vedvarende læsefejl.

En følge heraf er, at mens udførelsen af en SK-ordre altid varer 20 millisekunder, kan en LK-ordre eventuelt beslaglægge tromlekredsløbene længere tid, såfremt der optræder paritetsfejl. Inden man i et program bruger data, der er overført fra en tromlekanal til ferritlageret, må man derfor sikre sig, at transporten er vel afsluttet, og det kan ske ved i programmet at indskyde en (overflødig) VK-ordre med vilkårlig adresse, thi nyt kanalvalg kan ikke foretages, før tromlekredsløbene er ledige.

\*) I 3-tromle-maskinen har hver celle 3 ekstra positioner, og paritetskontrollen foregår modulo 8 i stedet for modulo 2.

Det samme bør man gøre, inden man lagrer nye data i en blok i ferritlageret, som just er blevet overført til en tromlekanal.

Ved tromlefejl, hvor TR-lampen lyser, kan GIER enten stoppes med RESET-knappen eller bringes til kritikløst at læse kanalen ind i ferritlageret; dette sidste gøres ved at benytte knappen "Annullering af paritetscheck" på betjeningspanelet øverst i GIER-skabet.

#### Eksempel 9.4.

```
I programdelen
VK 210, LK 400
---
VK 0, ARSF 403
---
```

sikrer ordren VK 0, at kanallæsningen er tilendebragt, før indholdet i celle 403 bruges i de følgende beregninger.

Hvis man blot skriver

```
VK 210, LK 400
---
ARSF 403
---
```

er det umuligt at forudsige, hvorvidt det er det gamle eller det nye indhold i celle 403, der bruges i de følgende beregninger (og dette kan ændre sig fra kørsel til kørsel).

#### Eksempel 9.5

```
Lad os betragte programdelen
VK 310, SK 200
---
GM 200
---
```

Hvis man ved, at det tager GIER mere end 20 millisekunder at gennemløbe programstykket mellem SK 200 og GM 200, så vil denne sidste ordre ingen indflydelse have på indholdet i tromlekanal 310. Men hvis det tager mindre end 20 millisekunder, er det umuligt at forudsige, hvorvidt det, der overføres til den første celle på kanal 310, er det gamle indhold i celle 200 eller M-registerets indhold (og dette kan ændre sig fra kørsel til kørsel).



Man bør for en sikkerheds skyld indskyde en VK-ordre,  
for eksempel

VK 310, SK 200

-----

-----

VK 0, GM 200

-----

hvorved man sikrer sig, at GM 200 først udføres, når tromletransporten er afsluttet.

#### 9.2.4 Kanalhop

Som omtalt ovenfor fungerer ordrerne VK, SK og LK blandt andet som spærreordrer, som sikrer, at GIER ikke går videre i programmet, før en igangværende tromletransport er afsluttet. Hensigten hermed er bl.a., at man under en beregning, der foregår simultant med en tromletransport, har behov for at sikre sig, at tromletransporten er afsluttet, før programmet begynder at benytte det område i ferritlageret, der berøres.

Der findes imidlertid også en anden facilitet, der undertiden kan benyttes til at lade GIER selv optimalisere udnyttelsen af den "ledige" tid under en tromletransport, nemlig ordren HK ("kanalhop", beskrevet i Lærebog I, operationslisten side 87).

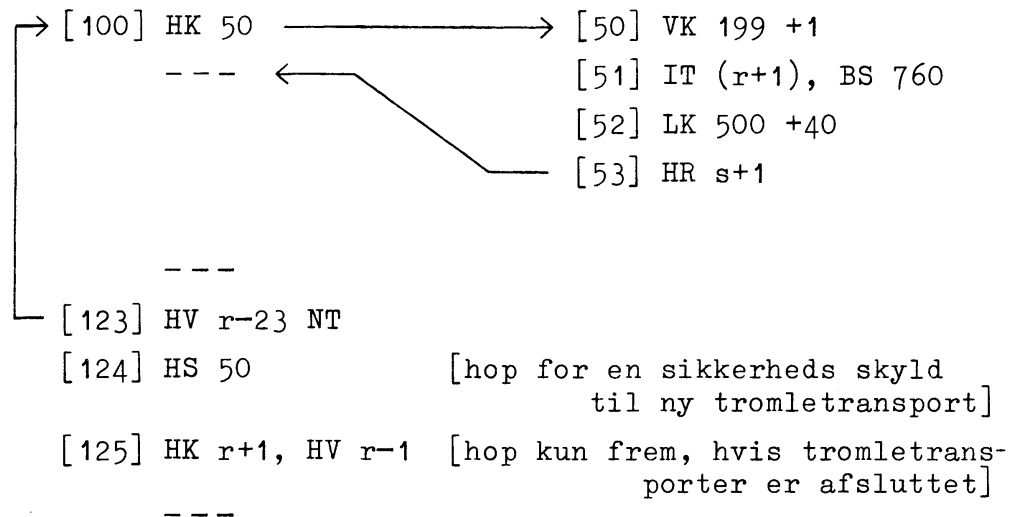
HK-ordrens virkning afhænger af, om der er en tromletransport igang: Hvis dette er tilfældet, er HK-ordren blind (på nær evt. S-variant), og GIER fortsætter øjeblikkeligt med de følgende ordrer; hvis tromlen derimod er ledig, fungerer HK-ordren som et sekvenshop, d.v.s. nøjagtig som en HS-ordre.

Dette kan f.eks. bruges i følgende situation: En programløkke skal gennemløbes mange gange og varer en betragtelig tid,

d.v.s. 0.1 sekund eller mere, og samtidig ønsker man at overføre indholdet i en anden del af ferritlageret til tromlen. Man kan da i den centrale programløkke indføje en HK-ordre, der (når tromlen er ledig) hopper til nogle få ordrer, som starter den "næste" tromletransport, og derpå hopper tilbage i den centrale programløkke (se eksemplet nedenfor).

### Eksempel 9.6.

En programløkke i celle 101-123 skal gennemløbes et stort antal gange, og samtidig ønskes de 7 tromlekanaler 200-206 overført i ferritlageret til celle 540-819. Idet vi antager, at celle 50-53 er ledige, kunne man kode således:



I den centrale programløkke er indføjet ordren HK 50, som bevirker sekvenshop til celle 50, hver gang den sidste tromletransport er afsluttet. Ordrene i celle 50 og 52 sørger for at starte en ny tromletransport, og BS-ordren i celle 51 sikrer, at kun de ønskede 7 kanaler overføres. Imidlertid må man sikre sig, at alle kanalerne er overført, når maskinen er færdig med programløggen, og derfor er der i celle 124-125 anbragt 3 ordrer, der sikrer, at GIER ikke fortsætter, før den sidste kanal er bragt på plads i ferritlageret: Først når tromlen melder ledig, træder ordren HK r+1 i kraft.

HK-ordren er så lumsk, at ovenstående program ikke behøver at fungere korrekt ved trinvis kørsel, thi da vil hver enkelt tromletransport uden tvivl være afsluttet, inden den næste ordre udføres, og derfor vil HK-ordren i celle 125 blive aktiv, første gang den aktiveres, selvom der på det tidspunkt måske mangler flere kanaloverførsler.

### 9.3 Strimmellæser, perforator og skrivemaskine.

GIER er udstyret med en strimmellæser og en perforator til 8-kanal-strimmel samt med en skrivemaskine til både ind- og udlæsning. I dette afsnit omtales kun disse apparaters virkemåde, mens udformningen af ind- og udlæseprogrammer beskrives i senere kapitler.

Såvel indlæsning som udlæsningen sker med én hulsøjle (eller ét tegn) ad gangen, idet en LY-ordre kun bevirker indlæsning, en SY-ordre udlæsning, af én hulsøjle (ét tegn).

Disse ydre enheder er tilkoblet GIER gennem bufferregistre med 10 bits, hvoraf dog kun de bageste 8 (pos.2-9) bruges her. Desuden styres selve tilkoblingen af de forskellige ydre enheder af by-registeret; valg af ydre enheder foregår derfor med VY-ordrer, som kan ændre indholdet af by-registeret. I dette register bestemmer pos.3-6 valg af udskrift-enheder og pos.7-9 valg af indlæse-enhed.

#### 9.3.1 Strimmellæser.

Efter at en strimmel er lagt i læseren, vil et tryk på den sorte knap på læseren bevirke, at én hulsøjle læses til bl-registeret, men der sker ingen transport til selve GIER. Strimlen

vil iøvrigt være stoppet midt imellem den just indlæste hulsøjle og den næste. (En hulsøjle med kun det lille føringshul, Blank Strimmel, opfattes slet ikke af strimmellæseren, som altid vil opsøge den første søjle med hul i mindst én af de 8 betydende positioner; føringshullet er iøvrigt helt overflødigt, da det slet ikke aflæses af strimmellæseren).

En LY-ordre bevirker nu (forudsat at strimmellæseren er valgt) at indholdet i bl-registerets pos.3-9 overføres til R-registerets pos.3-9 og til pos.3-9 i den celle, der udpeges af LY-ordrens resulterende adresse; dernæst indlæses eet tegn fra strimmel til bl-registeret, således at paritetsmærket på strimlen (4.øverste position) sættes i pos.2 i bl-registeret, mens resten af hulrækken anbringes i pos.3-9 i bl-registeret.

Paritetsbit'en overføres ikke til R-registeret og heller ikke til lagercellen, men der udføres en paritetscheck samtidig med transporten fra bl-register til R og celle: Hvis antallet af ettaller i bl-registeret (d.v.s. antallet af huller på strimlen) er lige, stopper GIER efter LY-ordren - som ved normalt stop - med det gale tegn eksklusive paritetsbit i R-registeret og i lagercellen og med det næste tegn i bl-registeret. Samtidig lyser L-lampen på manørebordet (og "str.læs. par.fejl" ved HP-knappen); disse slukkes først ved udførelsen af den næste LY-ordre eller ved betjening af den sorte knap på strimmellæseren (der fører næste tegn til bl).

En undtagelse herfra danner behandlingen af en hulsøjle med huller i alle 8 positioner (All Holes): Til trods for at antallet af huller er lige, stopper GIER ikke; i bl-registeret bliver pos.2-9 et-stillet, og i R-registeret (og i lagercellen) et-stilles pos.3-9 svarende til talværdien 127.

Ved at sammenligne tabellerne 8.3 og 8.4 i GIER Lærebog I (side 154-155) ser man, at hvis man på strimlen ser bort fra paritetsmærket (4. position fra venstre), svarer hullerne netop til ettallene i den binære repræsentation af talværdierne i GIER. Dette udstrækkes iøvrigt også til de hulkombinationer, der ikke umiddelbart kan frembringes på flexowriter'en; f.eks. vil hulkombinationen |oo oo. o| accepteres af GIER og give talværdien 109 i Radr og i adressepositionerne i den pågældende lagercelle.

### Eksempel 9.7.

Som følge af, at GIER ved paritetsfejl på strimlen først stopper, når det gale symbol står i Radr og i en celled adressepositioner, bør ethvert godt indlæseprogram umiddelbart efter hver LY-ordre tage indholdet i R-registeret under behandling, men totalt negligere indholdet i den pågældende lagercelles adresse. Thi når GIER er stoppet på en paritetsfejl under strimmelindlæsning, er det da nemt at rette indholdet i Radr med trykknapperne på manøverbordet, og et tryk på NORMAL START vil derefter sætte indlæseprogrammet i gang med at behandle det korrigerede symbol og derefter fortsætte indlæsningen. Derimod er det besværligt, hvis man også skal rette på indholdet i en celle (vedr. betjeningen af manøverbordet se iøvrigt kapitel 10).

### 9.3.2 Perforator.

En SY-ordre med resulterende adresse c bevirker (forudsat at perforatoren er valgt), at der hulles en kombination på strimlen, der afspejler de 7 bageste binære cifre i adressen c; desuden hulles eventuelt paritetsmærket, således at antallet af huller altid er ulige. Samtidig overføres de 7 binære cifre til pos.

3-9 i bs-registeret, og paritetsmærket sættes i pos.2 i bs (desuden dannes der en sum i celle 1023; se kapitlet om udlæsning).

Også de talværdier  $c$ , der ikke svarer til flexowriter-symboler, vil bevirke perforering af en hul søjle nemlig den til  $c$ 's binære repræsentation svarende hulkombination (plus eventuelt paritetsmærke). Er  $c \geq 128$ , er det tallet  $c$  modulo 128 (de 7 bageste cifre), der bestemmer, hvad der hulles.

#### Eksempel 9.8.

Ordren SY 127 bevirker hulning af kombinationen |ooo o.ooo|; "All holes" kan slet ikke frembringes af GIER.

Ordren SY 128 bevirker hulning af paritetsmærket alene (tegnet for Mellemslag), og i bs-registeret er pos.2 etstillet og resten nulstillet.

#### 9.3.3 Skrivemaskine.

Den til GIER koblede skrivemaskine kan både bruges som input- og outputmedium, og vi vil først beskrive funktionen under indlæsning og derefter funktionen under udlæsning; tilsidst nævnes nogle få forsigtighedsregler, der må iagttages, når skrivemaskinen skiftevis bruges til indlæsning og udlæsning.

A) Indlæsning: Når GIER møder en LY-ordre, mens skrivemaskinen er tilkoblet, tændes den grønne lampe på skrivemaskinen, og GIER venter midt i LY-ordren, indtil der tastes et tegn på skrivemaskinen. Talværdien for det tastede tegn (se tabel 8.4, side 155 i Lærebog I) overføres til pos.3-9 i R-registeret og i den celle, som LY-ordrens adresse udpeger; derefter fortsætter GIER straks med de følgende ordrer, og den grønne lampe slukkes. Bemærk, at der her ikke er noget buffer-register som mellemed,

men at tegnene læses direkte fra skrivemaskinen til R-registeret og cellen.

Skrivemaskinen kan ikke låses i Upper Case (i modsætning til almindelige skrivemaskiner og flexowriter'e), og skal flere tegn i Upper Case indlæses (ved hjælp af flere LY-ordrer), må man holde Upper Case-tangenten nedtrykket, mens tegnene tastes; ganske vist er talværdien den samme i Upper og Lower Case, men idet man slipper Upper Case-tangenten, indlæses et Lower Case-tegn (med talværdi 58), som kan have højst uønskede virkninger det pågældende sted.

#### Eksempel 9.9.

Af tabel 8.4 (Lærebog I, side 155) fremgår det, at med skrivemaskinen som indlæseorgan kan man i Radr kun frembringe tallene mellem 0 og 64 inklusive, dog ikke tallene 10, 11, 12, 15, 26, 28, 31, 42, 44, 45, 46, 47, 61 og 63, da disse enten er ubenyttede eller svarer til symboler, der ikke findes på skrivemaskinen.

B) Udlæsning: Når GIER møder en SY-ordre, overføres de 7 bageste binære cifre af adressen til pos.3-9 i bs-registeret, og paritetsmærket sættes i pos.2 i bs; hvis skrivemaskinen er valgt, skriver denne nu det tegn, der svarer til indholdet i bs, d.v.s. SY-ordrens adresse modulo 128. Samtidig dannes en sum af de udlæste tegn i celle 1023; se kapitlet om udlæsning.

Hvis SY-ordrens adresse (modulo 128) ikke svarer til noget skrivemaskine-tegn, har ordren blot den virkning, at adressen overføres til bs-registeret som ovenfor; derefter fortsætter GIER med de følgende ordrer, uden at der er sket udskrift. (Den

næste SY-ordre kan dog tidligst udføres ca. 0,1 sekund senere, når skrivemaskinen har sendt klar-signal til GIER.)

Lower Case vælges ved hjælp af ordren SY 58 , og alle SY-ordrer bevirker derefter udskrift i Lower Case indtil ordren SY 60 , som bevirker, at alle følgende SY-ordrer giver udskrift i Upper Case (indtil den først følgende SY-ordre med adresse 58). Ved hjælp af en af disse to ordrer vil GIER altså "låse" skrivemaskinen i den ønskede Case-situation.

#### Eksempel 9.10.

Ordreerne

VY 16 +7	[udlæsning på skrivemaskine, indlæsning
SY 58, SY 32	uforandret]
SY 11, SY 84	
SY 129, SY 26	

bevirker udskrivning (i Lower Case) af et minustegn og et 1-tal (SY 129), mens de øvrige SY-ordrer er blinde.

C) Forsigtighedsregler: Hvis skrivemaskinen benyttes til udlæsning umiddelbart efter indlæsning, bør det (sidst) indlæste tegn ikke være

- a) Upper Case eller et tegn i Upper Case
- b) Car.Ret., Tabulator eller Mellemslag.

Dette skyldes, at under indlæsning venter GIER ikke på, at disse typografiske operationer er fuldført, men går videre i programmet, så snart den tilhørende talværdi er anbragt i Radr (og i en celle). En udlæse-ordre kan derfor skabe en konfliktsituation for skrivemaskinen, dersom den ikke har afsluttet den typografiske operation hørende til det indlæste tegn. Der vil dog ofte



blot ske det, at udlåse-ordren bliver virkningsløs, og GIER fortsætter med de følgende ordrer i programmet.

#### 9.3.4 Operationstider.

En enkelt LY-ordre beslaglægger kun GIER i ca. 50  $\mu$ s (inklusive en simpel adresseberegning), nemlig den tid, der bruges til overførsel fra buffer til R og celle; derefter fortsætter GIER med de følgende ordrer i programmet, men en ny LY-ordre kan først udføres, når den pågældende ydre enhed er klar igen (strimmellæseren efter 2 millisekunder, skrivemaskinen efter ca. 0,1 sekund).

Tilsvarende beslaglægger en SY-ordre kun GIER i ca. 60  $\mu$ s, mens perforatoren er optaget i ca. 8 millisekunder og skrivemaskinen i ca. 0,1 sekund.

#### 9.4 Anelex linieskriver.

GIER kan udrustes med en Anelex linieskriver, der kan skrive med en hastighed af maksimalt 1000 linier/minut med maksimalt 119 tegn i hver linie (eventuelt dog 120 tegn/linie).

Linieskriveren er tilsluttet GIER gennem to ligestillede buffer-registre hver med plads til 120 tegn, svarende til en hel linie; mens linieskriveren udskriver indholdet af det ene, opsamler det andet de tegn, der sendes fra GIER. Når linieskriveren både er færdig med at udskrive det ene bufferregister og har modtaget tegnet for CR, ordren SY 64, i det andet bufferregister, bytter de to bufferregistre rolle: Efter at papiret er ført en

linieafstand frem, udskrives der fra det andet, mens tegnene fra GIER opsamles i det første o.s.v.

Om udlæsningen sker på lineskriveren, skrivemaskinen eller perforatoren (eller flere af disse på en gang) styres af indholdet i by-registeret i forbindelse med et sæt trykknapper på det lille manøvrebord med HP-knappen. Der er 16 knapper, som er anbragt i 4 rækker med 4 i hver række i et "koordinatsystem", hvor

	by[3]	by[4]	by[5]	by[6]
Anelex	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Skrivemaskine	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Perforator	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Reserve	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

"abscissen" angiver en bit i by-registeret, mens "ordinaten" angiver et udlæseapparat.

Hvis ingen knapper er nedtrykket, sker der ingen udlæsning overhovedet, d.v.s. alle SY-ordrer er blinde (på nær overførsel af adressen til bs-registeret og summation i celle 1023).

Hvis en knap er nedtrykket, bevirker det, at SY-ordrer giver udskrift på det apparat, der hører til den pågældende række i "koordinatsystemet", forudsat at en forudgående VY-ordre har etstillet den bit i by-registeret, som hører til den pågældende søjle i "koordinatsystemet". Det er tilladt at "tænde" et vilkårligt antal knapper samtidigt; flere knapper i samme række bevirker således, at flere bits i by-registeret aktiverer samme udlæseenhed, og "tænder" man flere knapper i samme søjle, vil en

enkelt bit i by-registeret aktivere flere udlæseenheder og give simultan udskrift på flere apparater.

Med disse trykknapper annulleres den faste virkning, som by[4] og by[5] har på GIER-maskiner uden linieskriver (hvor by[4]=1 aktiverer perforatoren og by[5]=1 skrivemaskinen).

#### Eksempel 9.11.

Et program indledes med ordren

VY 16 +7

som et-stiller by[5] (uden at ændre indlæse-situationen). Under indkørsel af programmet ønskes al udskrift på skrivemaskinen, hvorfor knappen med kryds på nedtrykkes; under rutinekørsel ønskes al udskrift på både linieskriveren og perforatoren, og derfor nedtrykkes de to skraverede knapper:

	by[3]	by[4]	by[5]	by[6]
Anelex	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Skrivemaskine	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Perforator	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Reserve	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

#### Eksempel 9.12.

I et program ønskes 3 slags udskrift: Kontroludskrift (kun under indkørsel), en del perforatorudskrift (der skal bruges som data ved senere kørsler), samt noget linieskriverudskrift (der simpelthen er resultaterne fra den aktuelle kørsel).

Programmet kodes med ordren

VY 32 +7     [sæt by4 =1]

før kontroludskriften, ordren

















VY 16 +7 [sæt by5 =1]

før perforatorudskriften, og ordren

VY 8 +7 [sæt by6 =1]

før lineskriverudskriften.

Under indkørsel nedtrykkes de 3 overkrydsede knapper, hvorved kontroludskriften fås på skrivemaskinen og resten på lineskriveren, og under rutinekørsler nedtrykkes de 2 skraverede knapper:

	by[3]	by[4]	by[5]	by[6]
Anelex				
Skrivemaskine				
Perforator				
Reserve				

Den samme virkning kunne iøvrigt opnås med mange andre kombinationer af VY-ordrerne og trykknapperne.

#### 9.4.1 Margin og Tabulator.

Ved udskrift på lineskriveren må margin og eventuelle tabulatorstop indstilles manuelt ved hjælp af et sæt trykknapper på lineskriveren.

Marginen kan indstilles på ethvert af de første 63 anslag fra venstre, og tabulatorstop kan anbringes på ethvert af de 120 anslag.

#### 9.4.2 Carriage Return, Formularskift og Papirformater.

Som ovenfor nævnt bevirker ordren SY 64 trykning af en linie samt papirfremføring, men linieskriveren er bygget med automatisk kontrol af, om en linie er fyldt op: Hvis dens bufferregister har modtaget 119 tegn uden noget CR imellem, trykkes hele linien, når det 120.tegn modtages, og papiret føres frem (hvis det 120.tegn er et CR-tegn, føres papiret således to linier frem).

Linieskriveren er normalt indstillet på at skifte til en ny "side" (udføre formularskift), når én "side" er fuld. For de 3 standardpapirformater betyder dette følgende:

A4 på højkant:	Automatisk formularskift efter 72 linier	(å 81 anslag)
A4 på tværs:	- - - 48 linier	(å 111 anslag)
Stort format:	- - - 102 linier	(å 120 anslag)

Derudover vil linieskriveren udføre formularskift på ordren SY 42 (42 er en af de ubenyttede talværdier på skrivemaskine og flexowriter). Det bør understreges, at hvis man på papirformaterne A4 på højkant og på tværs prøver at skrive mere end henholdsvis 81 og 111 anslag/linie, vil de sidste tegn på hver linie blot falde uden for papiret; linieskriveren giver ingen indikation heraf, og tegnene går tabt.

#### 9.4.3 Linieskriverens kode.

Linieskriveren har en koblingstavle, ved hjælp af hvilken man kan fastlægge en vilkårlig ønsket korrespondance mellem talværdien i SY-ordrens adresse og det trykte symbol på linieskri-

veren. Denne råder over følgende symboler:

cifrene 0-9 og de store bogstaver A-Å

+ - × / † ( ) [ ]

= ≠ > < . , : ;

\* % ' & α £ \$ 10

Normalt er linieskriveren opkoblet til en kode, der ligger meget tæt op ad skrivemaskinens og flexowriter'ens; der er dog følgende små forskelle:

Bogstaverne trykkes på linieskriveren altid som store bogstaver.

SY med adressen 10, 12, 28, 45, 46, 47 og 61 giver mellemslag i stedet for ingenting.

SY med adressen 29, 62 og 63 er virkningsløs på linieskriveren.

SY med adresse 1, 14, 15, 16, 26, 31, 42 og 44 trykker andre symboler på linieskriveren end på skrivemaskinen og perforatoren.

Iøvrigt henvises til tabel 8.4 i Lærebog I, side 155, eller til tabellen bagest i dette hæfte.

### 9.5 Hulkortlæser.

GIER kan udrustes med en hulkortlæser, der kan læse kort med både huller og blyantmarkeringer. Samtidig kan den fungere som sorteremaskine, men GIER kan dog kun styre en meget primitiv form for sortering (frasortering af een klasse af kort); mere komplicerede sorteringer kan styres ved hjælp af en koblingstavle på hulkortlæseren samt et særligt kriterie-hulkort, der indlæses først. En sådan sortering foregår helt uafhængigt af til-

koblingen til GIER, og hulkortlæseren kan derfor også benyttes helt selvstændigt (off-line) som sorteremaskine.

Den maksimale læsehastighed er ca. 12 kort/sek. eller en kortcyklus på 80 millisekunder. Kører læseren med denne hastighed, skal GIER derfor kunne nå at behandle hvert kort på 80 millisekunder og skal endda have aflæst al den ønskede kort-information i løbet af 72 millisekunder (se afsnit 9.4.2). Hvis GIER ikke kan dette, vil det næste korts information slet ikke nå frem til GIER, og kortet lægges af sorteremaskinen i et restrum som tegn på, at det ikke er behandlet.

Læsehastigheden kan imidlertid varieres kontinuert helt ned til ca. 1,5 kort/sek., således at GIER kan få op til ca. 665 millisekunder til behandling af hvert kort (aflæsningen skal dog ske i løbet af 635 millisekunder).

#### 9.5.1 Mekanisk opbygning.

Kort: De benyttede kort har 80 kolonner, nummereret fra 0 til 79, med hver 12 positioner til hulning og 27 kolonner, nummereret fra 80 til 106 med hver 12 positioner til blyantmarkering. De 12 hullepositioner benævnes fra oven og nedad med numrene 12-11-0-1-2---8-9, hvor positionerne 12 og 11 kaldes overhuller eller kontrolhuller; de 12 markeringspositioner svarer til hullepositionerne og er nummereret på samme måde. Hver markeringskolonne deler plads med 3 hullekolonner, men da  $3 \cdot 27 = 80 + 1$ , deler den sidste markeringskolonne kun plads med hullekolonne nr. 78 og 79 og rager lidt længere ud til højre.

Læser: Læsemekanismen har to børstesæt, hvoraf det første med 81 børster aflæser blyantmarkeringen (3 børster pr.kolonne),

mens det andet med 80 børster aflæser hulningen. Aflæsningen sker rækkevis til et bufferlager, kortbilledbufferen, med 80 + 27 kolonner; når et kort har passeret begge børstesæt, indeholder kortbilledbufferen et fuldstændigt billede af kortet (med 1-taller for alle huller og markeringer og nuller iøvrigt). Dette billede flyttes til et andet bufferlager, udgangsbufferen, og derefter læses det næste kort til kortbilledbufferen \*). Når hele billedet er flyttet til udgangsbufferen, kan GIER hente informationen i denne udgangsbuffer, mens det næste kort indlæses til kortbilledbufferen.

Under den kolonnevise transport af kortbilledet fra det ene bufferlager til det andet, underkastes kortbilledet en undersøgelse, der eventuelt resulterer i en fejllindikation på følgende vis:

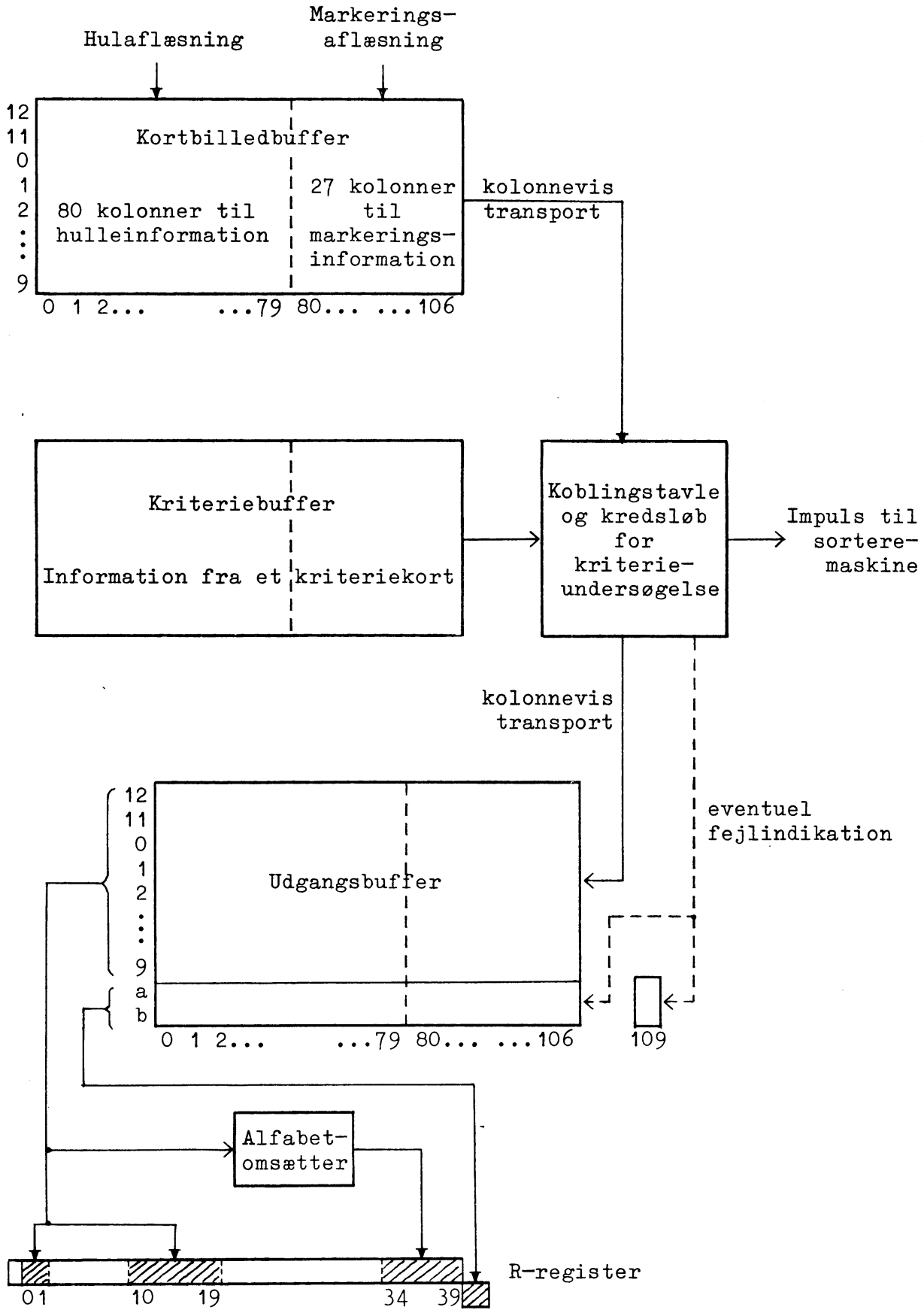
På forhånd kan vælges visse kriterier for korrekt hulning og markering af kort. Afhængig af disse kriterier opkobles en kriterie-koblingstavle, og indholdet af et særligt kriteriekort indlæses til et lager kaldet kriteriebufferen.

Hver kolonne af det indlæste billede undersøges nu ud fra de valgte kriterier: Hvis kolonnen ikke er fejlrant, sendes den blot videre til udgangsbufferen; hvis der er fejl i kolonnen, sættes en af de to fejlbits i kriteriekoblingstavlen, og idet kolonnen (iøvrigt uændret) sendes videre til udgangsbufferen, kan en sådan fejlbit samtidig overføres til en af de to ekstra

---

\*) I virkeligheden har kortbilledbufferen 27 ekstra kolonner, hvortil indlæsning af næste korts blyantmarkeringer påbegyndes samtidig med aflæsningen af forrige korts huller.





positioner, som findes i alle kolonnerne i udgangsbufferen. Man kan imidlertid også lade en sådan fejlbit stå i koblingstavlen under resten af kolonne-overførslen for det pågældende kort, for derpå til sidst at sende fejlbitten til en ekstra kolonne, som kun består af to fejlpositioner, kolonne nr. 109 i udgangsbufferen. I hver kolonne kan man således eventuelt indicere to forskellige fejltyper.

Hvis flere kolonner skal undersøges uafhængigt af hinanden for logiske fejl (enten med samme eller med forskellige kriterier), vil man som regel koble tavlen sådan, at fejlindikationerne sættes i de kolonner i udgangsbufferen, hvor fejlene findes; fejlundersøgelsen begynder da så at sige forfra for hver kolonne, der overføres.

Hvis man derimod ønsker en fælles fejlindikation, hvis blot een ud af adskillige kolonner er fejlramt (logisk addition af en fejlindikation), kan dette gøres ved at bevare fejlindikationerne i koblingstavlen og tilsidst sende den logiske sum deraf til kolonne nr. 109. Samtidig vil denne fejlindikation blive sendt til alle kolonner i udgangsbufferen efter den første af de fejlramte.

Vedrørende nærmere detaljer herom samt om mulighederne for at lade en enkelt fejlindikation afhænge af op til 6 kolonnens information, henvises til beskrivelser af hulkortlæseren.

Aflægning: Når kortene har passeret de to børstesæt, føres de henover 14 aflægningsrum, der benævnes 9-8-...-1-0-11-12-S-R, hvor restrummet R er længst væk fra aflæsningsbørsterne.

Ved hjælp af en koblingstavle og et kriteriekort kan man nu vælge mellem følgende former for aflægning:

a) Successiv aflægning: Rummene nr.9-0 benyttes til normal aflægning uden sortering, som starter i rum nr.9; når dette er fyldt op, fortsættes aflægningen automatisk i rum nr.8 o.s.v. Når rummene 9-0 alle er fulde, stopper hulkortlæseren, men tømmer man et af de fulde rum, skifter aflægningen tilbage hertil, og maskinen fortsætter atter til alle rum er fulde.

Aflægning i rummene 11, 12 og S styres fra koblingstavlen for logiske kriterier, og rummene bruges altså til aflægning af fejlramte kort.

Restrummet R benyttes kun til aflægning af kort, som GIER ikke har nået at behandle.

b) Primitiv sortering styret af GIER: GIER kan dirigere aflægning af udvalgte kort i rum nr.8 på følgende måde: Hulkortmaskinen foretager successiv aflægning som ovenfor, men kun i rummene nr.7-0; fejlrummene 11, 12 og S samt restrummet R benyttes som ovenfor. Et kort aflægges imidlertid i rum nr.8, hvis GIER udfører ordren LY 120 , inden næste kort kommer. Rum nr.9 bruges slet ikke under denne sortering. Ordren LY 120 skal udføres, inden der er gået 9/10 af en kortcyklus (ved maksimal hastighed 72 millisekunder) og inden ordren LY 127 , der betyder, at GIER er klar til næste kort.

c) Fin sortering: Alle finere former for sortering kan kun styres manuelt via kriteriekoblingstavlen og et kriteriekort. Her benyttes rummene nr.9-0 og 11-12 som sorterelommer, S som fejlrum (på et fejlkriterium, der er sat op på koblingstavlen) og R benyttes som restrum for de kort, som GIER ikke når at behandle. Man kan f.eks. sortere på de 12 positioner i en vilkårlig af hulkolonnerne eller på de 12 positioner i en vilkårlig af

markeringskolonnerne; man kan også sortere på mere end een kolonne, men vedrørende nærmere detaljer henvises til beskrivelser af hulkortlæseren.

### 9.5.2 Tilslutning til GIER.

Indlæsning af hulkort-information til GIER sker kolonnevis fra hulkortlæserens udgangsbuffer til R-registeret ved hjælp af LY-ordrer.

Det kræver, at hulkortlæseren er valgt som ydre enhed ved at placere bitkombinationen 010 i by-registerets pos.7-9 (smlgn. operationslisten i Lærebog I, side 89); dette kan f.eks. ske med ordren

VY 18 [indlæsning: hulkort, udlæsning: skrivemaskine]

eller med

VY 2 +120 [indlæsning: hulkort, udlæsning: uforandret]

Derefter bevirker en LY-ordre, at en kolonne indlæses til det tømte R-register, således at

kolonneposition 12, 11 læses til R pos.0,1

                  -          0 - 9          -          - R - 10-19

kolonnens ekstrabits          -          - R's mærkepositioner

Desuden omsættes kolonnens indhold (pos.12, 11 og 0-9) i en koblet alfabetomsætter til et 6-cifret, binært tal, og dette tal læses samtidig til R pos.34-39. På en koblingstavle for denne alfabetomsætter kan man styre den til at transformere et vilkårligt "hulkort-alfabet" til en vilkårlig 6 bits kode. Hver kolonne i udgangsbufferen kan kun aflæses een gang af GIER, da aflæsningen sker destruktivt.

Den resulterende adresse  $c$  i LY-ordren bestemmer bl.a.

hvilken kolonne der læses fra udgangsbufferen til R:

$0 \leq c \leq 79$  : Hullekolonne eller markeringskolonne nr.  $c$  læses  
 eller  
 $80 \leq c \leq 106$  til Rpos. 0, 1 og 10-19; den samme kolonne går via  
 alfabetomsætter til Rpos. 34-39; kolonnens to  
 ekstrabits læses til R's mærkepositioner. Resten  
 af R-registeret (inklusive R00) nulstilles.

$c = 109$  : De to fejlbits, benævnt kolonne nr. 109, læses til  
 R's mærkepositioner, mens Rpos. 00-39 nulstilles.

$c = 120$  bruges kun til den GIER-styrede sortering: Det  
 sidst aflæste kort aflægges i rum nr. 8 (mens alle  
 andre kort aflægges i rummene nr. 7-0, 11, 12, S  
 eller R).

$c = 127$  er klarsignal fra GIER til læseren: Næste gang,  
 kortbilledbufferen er fyldt op, overføres dens  
 indhold til udgangsbufferen; når dette er sket,  
 er læseren klar til at sende til GIER, som ikke  
 kan udføre den næste LY-ordre, før dette er til-  
 fældet.

Hvis læseren ikke modtager klarsignal fra  
 GIER (ved LY 127), inden der er gået 9/10 af en  
 kortcyklus, vil billedet af det næste kort ikke  
 blive overført til udgangsbufferen; desuden af-  
 lægges kortet i restrummet som tegn på, at GIER  
 ikke har nået at behandle det.

$c = 107, 108, 110 \leq c \leq 119$  eller  $121 \leq c \leq 126$  er forbudte  
 adresser; anvendelse vil medføre, at hulkortlæse-

ren stopper (på grund af at en paritetscheck, der udføres ved hver eneste kolonneaflysning, ikke stemmer), og GIER vil da vente på den næstfølgende LY-ordre, fordi hulkortlæseren ikke er klar til at sende til GIER.

Adresser  $\geq 128$  bliver opfattet modulo 128 og giver derfor en af de ovenfor beskrevne virkninger.

### 9.5.3 Eksempler på indlæsning.

Ofte vil et program til indlæsning og behandling af hulkort-information have følgende struktur:

1) En indledningsdel (retablering af adresser m.v.) med bl.a. en VY-ordre, der vælger hulkortlæseren som ydre enhed, samt ordren LY 127, der betyder "klar til første kort".

2) Begyndelsen af en stor løkke, der gennemløbes een gang for hvert kort; hulkortlæserens hastighed skal vælges sådan, at indlæsningen af et kort ikke tager mindre tid end eet gennemløb af løkken.

3) En eller flere løkker inden i den store. Hver løkke indledes med en ordre af typen LYS a t+1, som indlæser "næste" kolonne til det tømte R-register; hver løkke indlæser og behandler en vis gruppe af kolonner fra samme kort.

4) Afslutningen på den store løkke, som består af ordren LY 127, en eventuel afsluttende bearbejdning af det just indlæste samt et tilbagehop. Årsagen hertil er, at så snart den sidste ønskede kolonne er aflæst, bør GIER afgive klarsignalet LY 127; thi derefter varer det 1/10 kortcyklus, før GIER kan be-

gynde at læse næste kort, og dette tidsrum bør udnyttes i GIER til en eventuel afsluttende bearbejdning.

Ved indlæsning af en kolonne varer selve grundoperationen LY ca. 50  $\mu$ sek., og med en skikkelig adressedel med talletal tager udførelsen af en sådan LY-ordre altså ca. 75  $\mu$ sek.

Nedenfor følger nogle få meget primitive eksempler på indlæsning og behandling af kort-information.

### Eksempel 9.13.

På 500 hukort er der hullet positive heltal med maksimalt 3 cifre med enere i kolonne 10 (tiere i kolonne 9 og hundreder i kolonne 8). Disse heltal skal indlæses og summeres med summen som heltal i celle 1022.

Alfabetomsætteren kobles, så talværdien af cifrene 0-9 dannes i den sædvanlige binære form i R's bageste positioner; man kan da kode således (idet vi bruger SLIP-notation med symbolsk adressering o.s.v.):

VY 2 t 120	[vælg hukortlæser]
PM 10 D	[sæt faktoren $10 \cdot 2^{-9}$ i M]
LY 127, GRS 1022	[klar til 1.kort, nulstil sumcelle]
PA a3 t 499	
→a5: PA a0 t 7	[her begynder den store løkke]
GRS a2	[tøm arbejdscelle a2]
→a0: LY 7 t1	[lille løkke: læs næste kolonne]
TL -6, TLS 6	[nulstil de forreste 34 bits i R]
GR a1, MKS a2	[gem det indlæste ciffer, multiplicer det gamle med 10]
AR a1, TK 10	[adder det indlæste ciffer, forskyd p.g.a. faktor $10 \cdot 2^{-9}$ ]
GR a2, IT (a0)	[gem det omformede tal]
BS 10, HV a0	[slut på den lille løkke]
LY 127	[klar til næste kort]
ARS a2, TK -10	[sæt tallet i Rpos.39]
AC 1022, IT -1	[summer i celle 1022]
BT 499, HV a5	[her slutter den store kort-løkke]

ZQ 0

a1: QQ 0 [arb.celle: det delvis indlæste tal, enhed pos.29]

a2: QQ 0 [arb.celle: sidst indlæste tegn]

Et gennemløb af den store løkke (fra celle [a5] og fremad) varer lidt mindre end 2,5 millisekunder, hvilket betyder, at hver gang GIER har udført ordren LY 127, vil den stå og vente i den næste LY-ordre på at kunne begynde behandlingen af næste kort.

#### Eksempel 9.14.

I et primitivt kontrolprogram ønskes udskrift af blyantmarkeringer på kort: Der skal udskrives et "billede" af kortet med 1-taller for alle markeringer og nuller iøvrigt; kolonnerne 80-106 udskrives i omvendt orden på hver sin linie:

b0: VY 34, LY 127 [vælg hulkortlæser og perforator, klar til nyt kort]

PA b1 t 107 [retablering]

b2: SY 64, SY 64←

b1: LY 107 t -1 [læs næste kolonne]

PA b6 t 8 [retablering]

→b4: GA b3, SR b3 [ryd pos.0-9 i R]

TK 1, CA 0 [næste bit af kolonnen undersøges]

SY 16, HH b5 [skriv enten 0]

b5: SY 1, IT -1 [eller 1]

←b6: BT 8, HV b4 [hop tilbage 9 gange]

BS (b1) t 80

HH b2 [hop tilbage til ny kolonne]

HV b0 [hop til nyt kort]

b3: 0

Køres dette program, vil indholdet af det første kort blive trykt som ønsket, men da det varer ca. 2 sek. på perforatoren (der skal udlæses 280 tegn), løber der måske i mellemtiden adskillige kort gennem læseren (afhængig af dennes hastighed); disse aflægges i restrummet, og når GIER melder



klar, vil det kort, som læseren tilfældigvis er nået til, blive skrevet ud.

Tages udskriften derimod på linieskriveren, kan GIER følge med hulkortlæseren, og alle kortene vil blive skrevet ud samt aflagt i et af de første rum.

#### Eksempel 9.15.

Af en samling kort skal nogle frasorteres på basis af hulningen i kolonne 35 og 36, og sorteringskriteriet er sådan beskaffen, at det ikke kan sættes op på koblingstavlen, men må overlades til GIER. Man kan da bruge sorteringsordren LY 120 og f.eks. kode således:

	VY 2 t 120	[vælg hulkortlæseren]
→a0:	LY 127, LY 35	[klar til næste kort, læs kolonne 35]
	GR 0 D	[gem i denne celledes pos.0-19 den læste kolonne]
	LY 36	
	GR 0 D	[gem kolonne 36 i denne celledes pos.0-19]
	-----	} [sorteringskriteriet udregnes, og reduceres til at være angivet af fortegnet i R]
	-----	
	LY 120 LT	[det læste kort frasorteres i rum 8, hvis $R < 0$ ]
	HV a0	[hop til læsning af næste kort]

Forudsat at læserens hastighed sættes, så ovenstående løkke kan gennemløbes på mindre end 9/10 kort-cyklus i læseren, vil de frasorterede kort blive aflagt i rum nr.8, mens de øvrige kort aflægges successivt i rum 7-0.

Hvis løkken tager længere tid, vil det første kort blive lagt i rum 7 uanset sorteringskriteriet (fordi sorteringsordren kommer for sent); næste kort går i restrummet (fordi det ikke bliver læst); tredje kort læses, men går altid i rum 7, fjerde kort går i restrum etc.

## 10. MANØVREBORD; REGISTRE.

### 10.1 Hovedmanøvreboard.

Selvom hovedmanøvreboardet er næsten overflødig ved kørsel med HJÆLP, skal vi alligevel gennemgå betydningen og virkningen af de forskellige lamper og knapper. HP-knappen med videre omtales nedenfor i afsnit 10.3.

Nederst til venstre sidder tændingslåsen, hvormed man tænder og slukker for GIER. Nederst på midten sidder en volumenkontrol for en højttaler, der er tilkoblet R00, og som giver lyd, hver gang R00 ændres.

#### 10.1.1 Lamper på manøvreboardet.

Fotografiet bagest i bogen viser GIERs manøvreboard. Den venstre del af øverste lamperække viser indholdet i indikatorregisteret m.v.; den midterste del af lamperækken fortæller om den øjeblikkelige tilstand (mere teknisk), og den højre del viser, hvilket register operatøren sidst har kaldt frem (ved hjælp af trykknapperne). Den anden lamperække viser, når maskinen er stoppet, det øjeblikkelige indhold af det valgte register, idet cifrene 0 og 1 vises ved henholdsvis slukket og tændt lampe.

Har operatøren valgt et af 10-bitregistrene (f.eks. p-registret), vises dette i den venstre ende af lamperækken betegnet med ADRESSEDEL.

Mere detaljeret har lamperne og knapperne i øvrigt følgende funktioner, idet vi begynder i øverste venstre hjørne:

- O viser indholdet af overløbsregisteret O (idet 1 betyder overløb, og 0 betyder ikke-overløb).
- T viser indholdet af den fortegnbestemmende pos.00 i R-registeret; dette indhold kan indstilles ved hjælp af trykknapperne under T.
- OA, OB, ..., KB viser indholdet af indikatoren. KA og KB kan indstilles ved hjælp af de tilhørende trykknapper.
- YE lyser, når der læses eller skrives med en ydre enhed, samt hvis maskinen venter på en ydre enhed, f.eks. ved indlæsning fra skrivemaskinen.
- $M_1, M_2, M_3, M_4$  viser, hvilken del af den aktuelle ordre, maskinen er i færd med:  $M_1$  lyser under den indledende adresseberegning,  $M_2$  under selve grundoperationen og  $M_3, M_4$  under eventuelle afsluttende modifikationer.
- h lyser, når GIER udfører en venstre halvordsordre, samt når maskinen er stoppet efter udførelsen af en venstre halvordsordre. Ellers er denne lampe slukket.
- KLAR lyser, når GIER er stoppet, d.v.s. maskinen har spænding, men kører ikke.
- SF lyser, når der ikke er maskinspænding (f.eks. under opvarmning af maskinen eller ved maskinfejl).

- TO lyser, når den aktuelle ordre indeholder en ikke-defineret grundoperation (eller ved maskinfejl), og GIER stopper efter den indledende adresseberegning.
- TR lyser, når der er paritetsfejl ved læsning fra tromlen.
- L lyser, når der er paritetsfejl ved indlæsning fra strimmellåseren, og GIER stopper, klar til at udføre næste ordre (se afsnit 9.3.1).

De sidste lamper i øverste række fortæller, hvilket registers indhold der vises i anden lamperække, og operatøren kan vælge at se et nyt register ved hjælp af trykknapperne. Mulighederne er her \*):

- R Resultateregisteret (eksklusive pos.00, der altid ses til venstre i øverste række, benævnet T).
- M Multiplikatorregisteret.
- O Operandregisteret (et af registrene i den aritmetiske enhed).
- H Hjælperegisteret, hvori de egentlige regninger (og adresseberegninger) foregår.
- L Lagerinformationsregister, der fungerer som mellemlid (buffer) mellem ferritlageret og den aritmetiske enhed.
- F Funktionsregisteret, som indeholder den aktuelle ordre på nær adressetallet: Under udførelsen af en helordsordre indeholder pos.20-41 den aktuelle operationsdel med varianter etc., mens pos.10-19 kun under de første mikrotempi af ordren indeholder tælledele. Under ud-

---

\*) Nogle af disse registre er mere detaljeret omtalt i kapitel 2 i Lærebog I.

førelsen af en halvordsordre er pos.30-39 nulstillet.  
Pos.0-9 i F-registeret udgør indexregisteret, p-regi-  
steret.

- r1 Ordretælleren, som viser den aktuelle (eller den næste) ordres ferritlageradresse.
- s1 Sekvensregisteret, som bruges ved s-mærkede adresser.
- r2 Hjælperregister for adresseberegning (adresseregister for ferritlageret).
- s2 Hjælperregister for adresseberegning.
- in Indikatoren (dennes indhold vises ganske vist også øverst til venstre, men indholdet kan kun ændres ved at vælge in og derefter benytte trykknapperne under anden lamperække).
- ta Adresseregister (for ferritlageret) ved tromletransporter.
- tk Tromlekanalregisteret.
- bl Mellemlid (buffer) mellem strimmellæseren og maskinen; bl indeholder altid det tegn, der ved førstkommande læseordre overføres til selve GIER. Umiddelbart efter læseordrens udførelse overføres næste tegn fra strimmen til bl (se også afsnit 9.3).
- bs Mellemlid (buffer) mellem maskinen og skriveenhederne; bs indeholder altid det sidste tegn, der er skrevet på en ydre enhed (se også afsnit 9.3).
- by Ydre enheds register, der indeholder information om, hvilke ydre enheder, der er valgt (se også kapitel 9).

Teksten over den anden lamperække angiver, hvor de enkelte ordrestanddele står, når det valgte register indeholder en

helordsordre eller to halvordsordrer. Det bemærkes specielt, at

a) Pos.0-9, som på manøvrebordet er betegnet ADRESSEDEL, viser kun adresetallet, mens resten af adressedelen - parentes-mærkning eller r-, s- eller p-mærkning - vises i pos.27-29.

b) Pos.33-34, betegnet IO, viser indikatoroperationer (I, M, N eller L), og pos.35-39, betegnet IA, I<sub>A</sub> og I<sub>B</sub>, viser tilsammen indikatoradressen. Se iøvrigt afsnit 4.9 i Lærebog I.

Når GIER er stoppet, kan man med knapperne under lamperækken indstille indholdet i hver enkelt position som ønsket, og med de to knapper yderst til venstre kan hele registeret fyldes med nuller eller ettaller (pos.00 i R-registeret må dog altid indstilles med knapperne under lampen T).

#### 10.1.2 Start- og Stop-knapper.

Forneden på manøvrebordet findes der to sæt af start- og stop-knapper, betegnet med NORMAL og MIKROTEMPI. Disses funktioner er følgende:

NORMAL START må kun benyttes, når maskinen er stoppet (lampen KLAR lyser) som følge af:

- a) udførelsen af en stopordre, eller
- b) brug af knappen NORMAL STOP, eller
- c) brug af knappen MIKROTEMPI STOP.

Et tryk på NORMAL START vil da få GIER til at begynde at regne, idet den begynder med ordren i den celle, hvis adresse står i r1. Holdes NORMAL STOP nedtrykket, vil hvert tryk på NORMAL START bevirke, at een ordre udføres.

Har knappen MIKROTEMPI STOP været benyttet, gælder der særlige regler (se nedenfor).

**NORMAL STOP:** Ved tryk på denne knap stopper maskinen, når den er færdig med den igangværende ordre (se afsnit 10.2). En eventuel igangværende tromletransport afsluttes, før maskinen stopper. (Hvis den igangværende ordre er en venstre halvordsordre, lyser lampen h efter stoppet).

Kører maskinen på grund af kodefejl i en lukket parentes-kæde (en løkke af parentesmærkede ordrer, der "henviser til sig selv") eller på grund af maskinfejl i en løkke i mikroprogrammet (det kan f.eks. ske ved manglende klarsignal fra en ydre enhed), standser GIER ikke ved brug af knappen NORMAL STOP \*). Man kan da trykke på MIKROTEMPI STOP (se nedenfor).

**MIKROTEMPI START:** Forudsat at maskinen er stoppet, vil den ved tryk på denne knap udføre eet trin i mikroprogrammet for den aktuelle ordre. Hver ordres mikroprogram består af mange (d.v.s. 20-30) mikrotrin, som er opdelt i fire hovedafsnit, og lamperne  $M_1$ ,  $M_2$ ,  $M_3$ ,  $M_4$  i øverste lamperække viser, hvilket hovedafsnit af mikroprogrammet maskinen er igang med.

**MIKROTEMPI STOP eller RESET:** Ved tryk på denne knap stopper GIER øjeblikkeligt, eventuelt midt i en ordre, og en igangværende tromletransport standser også (uafsluttet). Da knappen samtidig udløser visse andre funktioner (nulstilling af visse registre m.v.), er det meget farligt at benytte knappen som stopknap under almindelig kørsel, da man derved risikerer at få ødelagt indholdet i en celle i ferritlageret.

---

\*) Maskinen "husker" dog, om der har været trykket på knappen og standser i så fald før udførelsen af den næste ordre, hvis f.eks. et manglende klarsignal pludselig kommer.

Iøvrigt udløses følgende funktioner:

a) Den centrale enhed gør klar til at udføre en ny helordsordre eller en venstre halvordsordre: h-lampen slukkes, hvis den var tændt, og  $M_1$ -lampen tændes. Den næste ordre vil blive hentet fra celle [r1], og afhængig af, på hvilket tidspunkt der er trykket på RESET, kan dette være den igangværende eller den næste ordre.

b) Skrivemaskinen sættes i Lower Case.

c) Funktioner i forbindelse med tromlen standses (er der vedvarende paritetsfejl under læsning fra tromlen eller vælges en ikke eksisterende kanal, kan maskinen kun standses ved brug af knappen MIKROTEMPI STOP, se afsnit 9.2.2).

d) Funktioner i forbindelse med de ydre enheder standses (klarsignalkredsene nulstilles).

e) by[0] nulstilles. (Vedrører HP-knappen).

Ønsker man under kørsel at stoppe GIER, skal man benytte NORMAL STOP. Ønsker man under en kørsel at foretage indgreb i et program, kan man benytte HP-knappen ved skrivemaskinen (se nedenfor).

## 10.2 Betjening af hovedmanøvrebord.

### 10.2.1 Stop.

Maskinen bør under normal kørsel i HJÆLP-SLIP-systemet slet ikke stoppes, men det kan dog gøres på en af følgende to måder:

a) Ved tryk på NORMAL STOP. Maskinen stopper, når den igangværende ordre er fuldført, på tærsklen til den næste ordre (teknisk set stopper GIER i mikrotrin 1 af første afsnit  $M_1$  af



den næste ordre). Den just udførte ordre står i F-registerets pos.10-41; r1-registeret indeholder adressen på den næste ordre.

b) Med en stopordre i programmet. Maskinen stopper som ovenfor på tærsklen til den næste ordre, efter at stopordren inklusive eventuelle modifikationer er fuldført.

I begge tilfælde afsluttes en igangværende tromletransport og ydre enheds-funktion korrekt, og i begge tilfælde bevirker et tryk på NORMAL START, at GIER fortsætter i programmet.

### 10.2.2 Start.

Er maskinen standset på en stopordre, og ønsker man, at maskinen skal fortsætte normalt med den næste ordre, trykkes blot på knappen NORMAL START.

Er maskinen stoppet på en eller anden måde, og ønsker man at starte med at udføre den helordsordre eller den venstre halvordsordre, der står i celle [m], trykkes først på MIKROTEMPI STOP; derefter vælges r1-registeret, og adressen m indsættes bit for bit. Ved tryk på NORMAL START vil maskinen begynde "fra venstre" i celle m.

Iøvrigt bemærkes, at er maskinen standset på en venstre halvords-stopordre, og ændrer man derefter på indholdet i r1, vil GIER ved tryk på NORMAL START begynde med at udføre den højre halvordsordre i den nye celle.

### 10.2.3 Udførelse af enkelte ordrer.

Manuel indsætning af en ordre kan gøres på to måder, idet man enten kan få ordren udført uden at den anbringes i lageret, eller man kan få den lagret i en celle og udført samtidig.

I begge tilfælde forudsættes det, at maskinen er stoppet:

a) En ordre ønskes udført uden at den skrives i ferritlageret:

1. Tryk på MIKROTEMPI STOP.
2. Tryk 3 gange på MIKROTEMPI START.
3. Vælg L-registeret og indsæt heri den ønskede ordre bit for bit, enten som en helordsordre eller som en venstre halvordsordre.
- 4a. NORMAL STOP holdes nedtrykket, og ved at trykke 1 gang på NORMAL START vil den ønskede ordre blive udført (og GIER stopper på tærsklen til den næste ordre).
- 4b. Ønsker man at lade maskinen fortsætte, som om den indsatte ordre stod i den celle, hvis lageradresse stod i r1 ved procedurens begyndelse, trykkes blot een gang på NORMAL START.

b) En ordre ønskes skrevet i ferritlageret og derefter udført:

1. Tryk på MIKROTEMPI STOP.
2. Vælg r1-registeret og indsæt den ønskede ferritlageradresse heri.
3. Tryk 2 gange på MIKROTEMPI START.
4. Vælg L-registeret, som nu har den valgte celledes tidligere indhold. Det ønskede indhold af cellen indsættes i L.
5. Tryk 1 gang på MIKROTEMPI START, hvorved det nye indhold i L-registeret anbringes i den valgte celle i ferritlageret.
6. For at få udført ordren fortsættes som under punkt 4a eller 4b ovenfor.

Det indskærpes, at man ikke må benytte NORMAL START-knappen før efter tredje tryk på MIKROTEMPI START, da maskinen ellers kan regne forkert.

c) Trinvis kørsel: Hvis man holder knappen NORMAL STOP nedtrykket, medens man trykker på NORMAL START, kan man gennemløbe et program ordre for ordre; ved hvert tryk på startknappen udføres normalt een ordre (dog udføres en hel satellitkæde på een gang).

d) Ønskes indholdet i en celle i ferritlageret ændret, udføres først punkterne 1, 2, 3, 4 og 5 under b). Derefter erstattes punkt 6 af et tryk på MIKROTEMPI STOP (hvorefter man på ny kan vælge r1 og ved at ændre dettes indhold begynde et vilkårligt sted i ferritlageret).

#### Eksempel 10.1.

Et program på 40 celler, der er lagret på tromlekanal nr.0, ønskes anbragt i ferritlagerets celle 10-49, hvorefter GIER skal starte med at udføre ordren i celle 10. Fremgangsmåden er da følgende, idet vi antager, at maskinen er stoppet:

1. Tryk på MIKROTEMPI STOP.
2. Vælg tk-registeret og indsæt tallet 0.
3. Tryk 3 gange på MIKROTEMPI START.
4. Vælg L-registeret; nulstil hele registeret og indsæt adressetallet 10 (som ettaller i pos.6 og pos.8) samt operationsdelen LK med talværdien 52 (som ettaller i pos.20, 21 og 23).
5. Med NORMAL STOP nedtrykket trykkes 1 gang på NORMAL START.
6. Tryk på MIKROTEMPI STOP.
7. Vælg r1-registeret og indsæt adressen 10.
8. Ved NORMAL START begynder GIER at udføre ordrerne i celle 10, 11, ...

Eksempel 10.2.

En programstrimmel ønskes indlæst ved hjælp af et indlæseprogram, der står i ferritlageret i celle 512 ff. Idet vi antager, at maskinen er stoppet, er fremgangsmåden følgende:

1. Læg strimmelen i strimmellæseren og tryk på knappen lige under læsebanen (herved overføres det første tegn på strimmelen til bufferregisteret b1).
2. Tryk på MIKROTEMPI STOP.
3. Vælg r1-registeret og indsæt adressen 512.
4. Tryk på NORMAL START, hvorved indlæsningen begynder.

Eksempel 10.3.

Maskinen er stoppet, og af indholdet i celle 203 skal den højre halvordsordre ændres til SR p+4, hvorefter maskinen skal startes i celle 200. Fremgangsmåden er:

1. Tryk på MIKROTEMPI STOP.
2. Vælg r1-registeret og indsæt adressen 203.
3. Tryk 2 gange på MIKROTEMPI START.
4. Vælg L-registeret; sæt adressetallet 4 i tælledele (et ettal i pos.17), operationsdelen SR i den højre operationsdel (d.v.s. talværdien 3 i pos.30-35, altså ettaller i pos.34 og 35), samt p-mærkning i det højre halvord, d.v.s. ettaller i pos.38 og 39.  
Det venstre halvord må ikke røres. Se efter, at indholdet er halvordsmærket ( $Lpos[40] = 1$ ) og ikke F-mærket ( $Lpos[41] = 0$ ).
5. Tryk 1 gang på MIKROTEMPI START, hvorved hele L-registeret sendes ud i celle 203.
6. Tryk på MIKROTEMPI STOP.
7. Vælg r1-registeret og indsæt adressen 200. Ved tryk på NORMAL START vil maskinen begynde at udføre ordrene i celle 200 ff.

#### Eksempel 10.4.

Maskinen er stoppet og ønskes startet på den højre halvordsordre i celle 30. Fremgangsmåden er:

1. Tryk på MIKROTEMPI STOP.
2. Vælg r1-registeret og indsæt adressen 30.
3. Tryk 3 gange på MIKROTEMPI START.
4. Vælg L-registeret og indsæt en uskadelig ordre som venstre halvordsordre; det nemmeste er at indsætte QQ 0 ved at nulstille pos.0-9 og 20-29. (Det højre halvord i L-registeret er den ønskede startordre).
5. Ved tryk på NORMAL START begynder GIER med at udføre ordren QQ 0, som er "blind", og fortsætter derefter med det ønskede program.

#### 10.3 HP-knappen med videre.

Ved siden af skrivemaskinen findes et lille ekstra kontrolbord med nogle lamper og knapper, hvoraf HP-knappen, hjælpeknappen er den vigtigste. Ved hjælp af denne knap kan man under kørslen nårsomhelst standse programforløbet og få adgang til at foretage indgreb i programmet (f.eks. i form af rettelser eller kontroludskrifter), for derefter at fortsætte programets kørsel "som om intet var hændt".

HP-knappen er dog koblet nøje sammen med hele HJÆLP- og SLIP-systemet, så den kan kun bruges ved programmer, der er kodet i dette system (det har f.eks. en ødelæggende virkning at bruge HP-knappen under kørsel med ALGOL-programmer, fordi hele ferritlageret derved lagres sidst på tromlen oven i ALGOL-programmet). Til gengæld er HJÆLP lagt således tilrette, at det egentlige manørebord overflødiggøres, idet al betjening kan

ske med det lille HP-kontrolbord og skrivemaskinen. Dette omtales nærmere i de næste kapitler, og her vil vi blot ganske kort nævne de lamper og knapper, der findes på det lille HP-kontrolbord:

Til venstre er der to store knapper:

RESET, som er identisk med knappen MIKROTEMPI STOP på hovedmanørebordet, omtalt ovenfor.

HP, Hjælpeknappen. Et tryk på denne udløser bl.a. følgende funktioner: Den igangværende ordre fuldføres, hvorefter indholdet i hele ferritlageret og alle registre gemmes sidst på tromlen, og der sker indhop til indlæseprogrammet SLIP, som venter på skrivemaskine-input. (En mere detaljeret omtale af virkningerne findes i kapitel 11 og 13).

Til højre er der en række lamper, som taget fra oven og nedad er følgende:

"K<sub>A</sub>" og "K<sub>B</sub>" med tilhørende knapper, som er identiske med de tilsvarende lamper og knapper på hovedmanørebordet. Man kan således begge steder aflæse og indstille indholdet i KA- og KB-registrene.

"Klar" er en kopi af Klar-lampen på hovedmanørebordet. Den lyser, når GIER er stoppet, hvilket ikke bør ske under kørsel med HJÆLP-systemet. Maskinen kan dog startes ved at trykke på HP-knappen, hvorved der foretages indhop til HJÆLP-administrationen (se de næste kapitler).

"YE" er en kopi af YE-lampen på hovedmanørebordet. Den lyser, når der læses eller skrives med en ydre enhed.

"str.læs. par.fejl" er en kopi af L-lampen på hoved-manøvrebor-  
det. Den lyser, når GIER er stoppet på grund af en pa-  
ritetsfejl på hulstrimlen; se iøvrigt afsnit 9.3.1.

"Tromlefejl" er en kopi af TR-lampen på hoved-manøvrebordet.

Den lyser, når der er paritetsfejl ved læsning fra trom-  
len.

"TO fejl" er en kopi af TO-lampen på hoved-manøvrebordet. Den  
lyser, når den aktuelle ordre indeholder en ikke-definie-  
ret grundoperation.

"HP spærret", som lyser når hjælpeknappen er sat ud af funktion,  
d.v.s. når  $by[0] = 1$ ; se iøvrigt beskrivelsen af HJÆLP  
i kapitel 13 \*).

"i-løkke", som lyser, når GIER cykler længe i en parenteskode  
(og  $M_1$ -lampen lyser vedvarende). Dette kan kun afbry-  
des ved at trykke på RESET, som vil stoppe GIER på  
tærsklen til den næste ordre uden at ordren med paren-  
teskoden er udført; hvis det sker i en venstre halvords-  
ordre, vil GIER dog stoppe klar til at gentage denne  
venstre halvordsordre (fordi ordretælleren  $r1$  ikke er  
talt frem, og fordi halvords-bit'en  $h$  nulstilles).

---

\*) Hvis HP-knappen nedtrykkes, når maskinen kører og medens  
 $by[0] = 1$ , sker der ingenting, førend  $by[0]$  nulstilles  
(ved udførelse af en VY-ordre); først da udløses HP-knap-  
funktionen; GIER "husker" altså, når der har været trykket  
på HP-knappen under kørsel. Dette er derimod ikke tilfæl-  
det, hvis maskinen står stille med HP-knappen spærret (den-  
ne situation bør aldrig forekomme). Ved tryk på RESET vil  
 $by[0]$  blive nulstillet.

## 11. INDLÆSNING; SLIP.

### 11.1 Indledning.

I dette kapitel vil vi først med et simpelt eksempel vise brugen af indlæseordren LY i indlæseprogrammer, men hovedparten af kapitlet handler om konventionerne for brug af indlæseprogrammet SLIP, som blandt andet tillader symbolsk adressering. Imidlertid er SLIP koblet nøje sammen med systemet af hjælpeprogrammer og udgør sammen med disse programkomplekset HJÆLP, som beskrives nærmere i kapitel 13, og nærværende kapitel handler mest om, hvordan ordrer, tal og tekst skal skrives, og hvordan det føjes sammen til et program. Men først når man kender de generelle mekanismer i HJÆLP-kompleksets administration, kan man forstå virkningen af SLIP, specielt indhop til og udhop fra indlæseprogrammet. Derfor må dette kapitel først og fremmest opfattes som en oversigt over reglerne for skrivning af programmer til indlæsning med SLIP.

Det skal bemærkes, at SLIP først og fremmest er beregnet til indlæsning af program, herunder også de talkonstanter og den tekst der indgår i programmet. Men SLIP bør ikke bruges som undersekvens hvortil der hoppes i mange omgange for at indlæse



data under beregningerne; for hvert indhop i SLIP bruges der nemlig ca. 1 sekund til administration (gemning og retablering af hele ferritlageret).

Derfor bør man til indlæsning af data bruge bibliotekssekvenser beregnet hertil (se kapitlet om bibliotekssekvenser).

## 11.2 Indlæseprogrammer; et eksempel.

Kernen i et indlæseprogram er som regel en eller flere LY-ordrer, der hver indlæser eet tegn. For at give nem adgang til at rette eller overspringe tegn med paritetsfejl (ved strimmelindlæsning) bør to regler overholdes: 1) Hvis det indlæste tegn skal lagres i en celle til senere brug, bør man ikke nøjes med den lagring, som sker i selve LY-ordren, men explicit skrive en GR-ordre umiddelbart efter indlæsningen. Rettelse af et galt tegn behøver da kun at foretages i R-registeret. 2) LY-ordren bør stå som en venstre halvordsordre, thi da kan man ved hjælp af RESET-knappen nemt få gentaget LY-ordren, d.v.s. få oversprunget det gale tegn.

Konklusionen af 1) og 2) er, at kernen f.eks. kan se således ud:

```
LY c, GR c
```

hvis celle[c] benyttes til lagring af tegnet.

Omkring kernen er der som regel en ret omfattende administration, fordi gode indlæseprogrammer har to funktioner udover den rent slaviske indlæsning tegn for tegn: Dels en omregning eller bearbejdning fra det benyttede "ydre" kodesprog til GIERS interne repræsentation, og dels en kontrol af, om de indlæste tegn er

meningsfyldte. Denne kontrolfunktion skal gøres så fuldstændig som mulig, fordi man på den måde kan forhindre, at maskinen begynder at arbejde på et datamateriale, der er forkert på grund af hullefejl og lignende. Man bør under planlægningen af et indlæseprogram tage hensyn til alle tilgængelige oplysninger om form og struktur af det datamateriale, der skal indlæses, og udnytte disse oplysninger til at indbygge fejlfindings- og kontrolmekanismer i indlæseprogrammet (jævnfør den systematisk gennemførte fejlsøgning i ALGOL-oversætteren til GIER).

Vi vil betragte følgende konkrete, ret simple eksempel: Der skal indlæses en række heltal med enhed i pos.39 til cellerne  $s+1$ ,  $s+2$ , ...; følgende konventioner skal gælde for hulning af tal-strimler:

1) Mellemslag, Tape Feed og All Holes er helt blinde symboler, som er tilladt hvorsomhelst.

2) Indlæseprogrammet starter i Lower Case, men Upper og Lower Case må gerne optræde på strimlen, blot skal cifrene i heltallet være i Lower Case.

3) Hvert tal består af højst et fortegn (evt. intet) og højst 10 cifre (mindst eet).

4) Som afslutning skal bruges enten Car.Return, komma eller et af bogstaverne a, A, b, B, c, C. Her skal Car.Return og komma medføre nulstilling af mærkebits for det indlæste tal, mens bogstaverne skal bevirke a-, b- eller c-mærkning af tallet.

5) Overholdes disse regler ikke, skal programmet give en fejludskrift, som dog bliver den samme i alle tilfælde.

Under indlæsning af et tal kan programmet være i forskellige "tilstande" afhængig af, hvad der er indlæst tidligere, og vi vil skelne mellem følgende seks:

- nr.1: Lower Case-situation før fortegn er indlæst
- nr.2: Upper Case - - - - -
- nr.3: Lower Case - før første ciffer er indlæst
- nr.4: Upper Case - - - - -
- nr.5: Lower Case - under cifferindlæsning
- nr.6: Upper Case - - -

Når indlæsningen af et tal starter, er programmet i tilstand nr.1, og nedenstående tabel viser så, hvordan det næste indlæste tegn påvirker programtilstanden, idet "ordinaten" til venstre er programmets gamle tilstand og "abscissen" foroven angiver det næste tegn, der indlæses:

Program-tilstand \ Næste tegn	Mellemrum Tape Feed All Holes	Upper Case	Lower Case	For-tegn	Ciffer	a,b,c komma CR	Alle andre tegn
1: LC før fortegn	1	2	1	3	5	Fejl	Fejl
2: UC før fortegn	2	2	1	4	Fejl	Fejl	Fejl
3: LC før ciffer	3	4	3	Fejl	5	Fejl	Fejl
4: UC før ciffer	4	4	3	Fejl	Fejl	Fejl	Fejl
5: LC blandt cifre	5	6	5	Fejl	5	Slut	Fejl
6: UC blandt cifre	6	6	5	Fejl	Fejl	Slut	Fejl

Tilstanden "Slut" betyder indlæsning og lagring af tallet afsluttes, og programmet vender tilbage i tilstand nr.1, klar til næste tal. Tilstanden "Fejl" betyder, at datastrimlen strider mod konventionerne, og programmet skal sørge for en fejludskrift. (Hvad der derefter skal ske vil vi ikke specificere her.).

I øvrigt er opdelingen i de 6 tilstande lidt grov; for eksempel er der ikke taget hensyn til, at et tal højst må have 10

cifre. Denne begrænsning er dog medtaget i nedenstående ALGOL-program og GIER-kode.

I et ALGOL-lignende sprog kan indlæseprogrammet skrives således, idet de boole'ske variable UC, efter fortegn og efter ciffer bruges til at holde styr på de 6 tilstande:

```

Nyt tal:      s:= s + 1;  tal[s]:= 0;
              cifferantal:= 0;
              UC:= efter fortegn:= efter ciffer:= minus:= false;

Næste tegn:  Tegn:= lyn;

Blinde:      if Tegn = 0  $\vee$  Tegn = 63  $\vee$  Tegn = 127 then go to Næste tegn;
Upper Case: if Tegn = 60 then begin UC:=true; go to Næste tegn end;
Lower Case: if Tegn = 58 then begin UC:=false; go to Næste tegn end;
Fortegn:     if Tegn = 32 then
              begin if efter fortegn then go to Fejl;
                minus:=  $\neg$  UC;
                efter fortegn:= true;
                go to Næste tegn
              end Fortegnsbehandling;

Ciffer:      if Tegn = 16 then Tegn:= 0;
              if Tegn  $\leq$  9 then
                begin if UC  $\vee$  cifferantal  $\geq$  10 then go to Fejl;
                  efter ciffer:= efter fortegn:= true;
                  cifferantal:= cifferantal + 1;
                  tal[s]:= tal[s]  $\times$  10 + Tegn;
                  go to Næste tegn
                end Cifferbehandling;
              tal[s]:= if minus then -tal[s] else tal[s];
              if  $\neg$  efter ciffer then go to Fejl;

Slut:       if Tegn = 64  $\vee$  Tegn = 27 then go to Nyt tal;
              if Tegn = 49 then begin sæt amærke; go to Nyt tal end;
              if Tegn = 50 then begin sæt bmærke; go to Nyt tal end;
              if Tegn = 51 then begin sæt cmærke; go to Nyt tal end;

Fejl:       skrvtekst ( $\{\langle$ syntaktisk fejl i talindlæsning $\rangle\}$ );
              ---

```

I GIER-kode kan dette indlæseprogram skrives således, idet vi benytter SLIP-notation, hvor grundoperationerne kan skrives med små bogstaver, og hvor der er benyttet symbolsk adressering (se iøvrigt de næste afsnit i dette kapitel):

```

[Nyt tal:]
  b0: ps s+1 , grn s      ; s:=s+1 , tal[s]:=0
      pp 0 , pi 0        ; cifferantal:=0 , alle boole'ske var.
                          ; i indik. nulstilles
[Næste tegn:]
  b1: lyn r+c1 , gr r+c1 ; Tegn:=lyn . c1 er arb.celle for tegnet
      ↑ ca 0 , hv r+b1
      ↑ ca 63 , hv r+b1
      ↑ ca 127 , hv r+b1 ; nu er de 3 blinde tegn siet fra
[Upper Case:]
      nc 60 , hv r+b2    ; if Tegn ≠ 60 then hop frem
      pi 1 t+6          ; sæt RB:=1 (UC:=true)
      ↑ hv r+b1
[Lower Case:]
  b2: nc 58 , hv r+b3    ; if Tegn ≠ 58 then hop frem
      pi 0 t+6          ; RB:=0 (UC:=false)
      ↑ hv r+b1
[Fortegn:]
  b3: nc 32 , hv r+b4    ; if Tegn ≠ 32 then hop frem
      hv r+c0 LRA       ; if efter fortegn then go to Fejl
      pi 8 t+7 NRB
      pi 0 t+7 LRB      ; minus:= ¬ UC (minus = QA)
      pi 2 t+13         ; sæt RA:=1 (efter fortegn:=true)
      ↑ hv r+b1
[Ciffer:]
  b4: ca 16 , grn r+c1   ; if Tegn = 16 then sættegn(0)
      sr 10 D
      hv r+b5 NT        ; if Tegn ≥ 10 then hop frem
      it p , bs 10      ; if p < 10 then
      hh r+1 NRB        ; if ¬ UC then fortsæt cifferbehandling
      hv r+c0 , it 11   ; else go to Fejl
      pi 4 , pm s       ; sæt QB:=1 (efter ciffer:=true)
      mln r+c2 , tl 9   ; R:=tal[s]×10 (enhed i pos.39)
      gr s , arn r+c1
      tk -30 , ac s     ; tal[s]:=tal[s]×10 + Tegn
      ↑ pp p+1 , hv r+b1 ; cifferantal:=cifferantal + 1

```

```

[tal færdigt:]
b5: hv r+2 NQA      ; if  $\neg$  minus then hop frem
    srn s , gr s    ; tal[s]:=-tal[s]
    hv r+c0 NQB     ; if  $\neg$  efter ciffer then go to Fejl
[Slut:]
pi 0                ; sæt mærkebits RA = RB = 0
arn r+c1 , ca 49; R:=Tegn
pi 2 , hh r+b6     ; if Tegn = 49 then sæt a-mærke i RA
ca 64 , hh r+b6    ; if Car.Return then go to Nyt tal
ca 27 , hh r+b6    ; if komma then go to Nyt tal
ca 50
pi 1 , hh r+b6     ; if Tegn = 50 then sæt b-mærke i RB
nc 51 , hv r+c0    ; if Tegn  $\neq$  51 then go to Fejl
b6: pi 3 , arn s   ; if Tegn = 51 then sæt c-mærke i RA, RB
    gr s MRC       ; forsyn tal[s] med mærkning
    hv r+b0        ; go to Nyt tal
c1: qq 0          ; arb.celle for det indlæste tegn
c2: qq 10         ; konstanten  $10 \times 2^{-9}$ 
c0: ---          ; program for fejludskrift
    ---

```

Til ovenstående program vil vi knytte følgende bemærkninger:

1) Det burde forsynes med lidt administration, som kunne afbryde indlæsningen, når et nærmere angivet antal tal er indlæst, thi programmet vil fortsætte indlæsningen, indtil det støder på en syntaktisk fejl på datastrimlen.

2) Hvis der er paritetsfejl på strimlen, stopper GIER i celle[b1] klar til at udføre GR-ordren i højre halvord. Hvis det gale tegn blot skal overspringes, trykkes på RESET og derefter på Normal Start; GIER vil da gentage LY-ordren og derefter fortsætte normalt. Hvis der skal rettes i det gale tegn, retter man i R-registerets pos.3-9 (hvor det gale tegn står uden paritetsmærke) og trykker på Normal Start; GIER vil da fortsætte med GR-ordren, der nu lagrer det rettede tegn i celle [c1].

3) Det er meget muligt, at programmet kan komprimeres, så det fylder mindre, men samtidig bliver det mindre gennemskueligt. Derfor har vi valgt ovenstående form.

4) Konventionerne for dette indlæseprogram er valgt meget stramme. I praksis bør man nok tillade flere blinde symboler imellem tallene.

### Eksempel 11.1.

Lad følgende datastrimmel blive indlæst med ovenstående program, idet  $s = 399$

+ -470ab-12,

Ved indlæsning af det første minus-tegn fås fejludhop fra celle  $[b3+1]$ . Hvis fejludskriften afsluttes med hop til fornyet indlæsning (i celle  $[b0]$ ), indlæses tallet +470 (minustegnet er gået tabt) og lagres i celle  $[401]$  med a-mærkning. Ved indlæsning af b'et fås fejludhop fra celle  $[b5+1]$ ; hvis maskinen derefter igen hopper til celle  $[b0]$ , vil tallet -12 blive indlæst og lagret i celle  $[403]$ , idet der også blev talt frem i s-registeret ved indlæsning af det gale tegn b.

## 11.3 Indlæseprogrammet SLIP.

### 11.3.1 Indledning. Disponibelt lager.

SLIP (Symbolic Language Input Program) er som nævnt i afsnit 11.1 nærmest en undersekvens i systemet HJÆLP, beregnet til indlæsning af program, konstanter og tekst. Man kan også benytte SLIP til programstyret indlæsning af data, men som regel er det her bedre at benytte en bibliotekssekvens (se kapitlet om bibliotekssekvenser).

Et væsentligt træk ved SLIP er at symbolsk adressering er tilladt, både i adressedele og i talletal; dette letter både kodearbejdet og fejlretningen, og det må i de fleste tilfælde anbefales at bruge symbolske adresser sammen med relativ adressering, fordi det derved bliver lettere senere at indsætte eller at fjerne ordrer.

I ferritlageret kan koderen disponere over celle 10-1022 inklusive, idet SLIPs faste administration ligger i celle 0-9, mens celle 1023 blandt andet benyttes til checksumdannelse under udlæsning. På tromlen kan man disponere over kanal 58-293, idet de første 58 kanaler benyttes til lagring af SLIP og HJÆLP (og heraf er igen kanal 0-31 låsede og indeholder SLIP samt en central del af HJÆLP), mens kanal 294-319 anvendes som et billede af ferritlageret under indlæsning: Alt hvad koderen forlanger indlæst til ferritlageret, indlæses i virkeligheden til en tilsvarende plads på disse kanaler, og først når indlæsningen afsluttes (lige før udhoppet fra SLIP), overføres kanal 294-319 til ferritlagerets celle 0-1023 (man kan altså også indlæse program eller data til celle 0-9, men det må frarådes, da SLIPs faste administration derved ødelægges).

Tilsvarende indledes enhver indlæsning med, at det øjeblikkelige indhold i celle 0-1023 og alle registre lagres på kanal 294-319, og først derefter påbegyndes indlæsningen. Overførslen af disse 26 kanaler - den ene eller den anden vej - tager  $26 \times 0.02 \text{ sek} \approx 0.5 \text{ sek}$ , og man bør derfor foretage indlæsning i så store portioner som muligt for at undgå overflødige kanaltransporter.

På maskiner med 3 tromler med kanalerne 0-959 benyttes kanal



934-959 som ferritlagerbillede; overalt i det følgende hvor der henvises til kanal 294-319, skal man ved en GIER med 3 tromler blot øge kanalnumrene med 640.

I de følgende afsnit omtales de vigtigste konventioner for indlæsning med SLIP, og i afsnit 11.9 findes en fuldstændig syntax for programskrivning (syntaxen er formet på samme måde som i ALGOL-rapporten).

### 11.3.2 Linier.

For at få et program bestående af ordrer, tal og tekst indlæst med SLIP, må SLIP have visse supplerende oplysninger, styringsinformation. Vi skelner derfor mellem programinformation (ordrer, tal eller tekst), som lagres i maskinen og udgør det færdige program, og styringsinformation, som ikke findes i maskinen efter endt indlæsning, men som styrer forløbet af selve indlæsningen.

Al information opdeles i enheder som kaldes linier, der skal være adskilt med Carriage Return (CR), og SLIP skelner mellem seks typer af linier, nemlig

ordrelinie, der består af en eller to ordrer	}	Program-	
tallinie, - - - et eller flere tal			information
tekstlinie, - - - en eller (paradoxalt nok) flere linier tekst			
internkodelinie (udskrift fra hjælpeprogrammet "kompud", som omtales nærmere i kapitel 13)			
styrelinie, der kan	}	Styrings-	
definere adresser (definitionslinie)			information
styre SLIPs lagring af tal (administrator)			
udføre valg af ydre enhed og lignende (administrator)			
hjelpeinformationslinie (der vælger hjælpeprogrammer og parametre til disse, som iøvrigt ikke omtales nærmere her, men i kapitel 13)			

De første tre typer af programlinier omtales i afsnittene 11.4 og 11.5, mens styrelinierne omtales nærmere i afsnit 11.6.

Blind information: 1) Under indlæsning af tekstlinier negligeres kun Tape Feed og All Holes (d.v.s. tegn med hul i alle 8 positioner).

2) Under indlæsning af alt andet end tekstlinier negligeres Mellemslag \*), Tabulator, Stop Code, Punch On, Punch Off, Tape Feed og All Holes; desuden overspringes alt fra og med [ til og med ], herunder også CR. Endelig overspringes alt fra og med semikolon til - men ikke med - CR; dette giver mulighed for at skrive kommentarer til de enkelte linier, f.eks. ved ordrer eller tal: Ønsker man en kommentar først eller midt på en linie, indesluttet den i kantede parenteser; ønsker man en kommentar sidst på en linie, kan man bruge de kantede parenteser eller man kan blot indlede kommentaren med et semikolon.

Symbolerne \_ og | negligeres, medmindre de indleder en linie og efterfølges af et symbol i Lower Case (og i så fald vil kun de i afsnit 11.6 omtalte kombinationer blive accepteret, mens de øvrige virker som s).

Flere CR efter hinanden har samme virkning som eet CR, således at de på nær det første CR er blind information.

---

\*) Under indlæsning fra skrivemaskine vil et mellemslag umiddelbart efter CR dog bevirke kontroludskrift med rødt af løbende adresse i (og eventuelt løbende kanalnummer k ); se iøvrigt kapitel 13.

#### 11.4 Indlæsning af ordrer med SLIP.

I dette afsnit beskrives den enkelte ordrelinie, og det får i det væsentlige karakter af en udvidelse af de konventioner for ordrenedskrivning, der er fastlagt i kapitlerne 3, 4, 5 og 6 i Lærebog I. Hvordan indlæsningen styres til at foregå til et bestemt afsnit af ferritlageret eller tromlen omtales først i afsnit 11.6.

En ordrelinie lagres altid i en helcelle; ordrelinien består enten af to halvordsordrer eller af een helordsordre; som skilletegn mellem to halvordsordrer skal anvendes komma (som tidligere) eller skråstreg. Det må bemærkes, at en halvordsordre kan være tom. Således vil f.eks. ordrelinien

,AR 17

bevirke, at venstre halvord nulstilles svarende til ordren QQ 0 mens ordren AR 17 anbringes som højre halvordsordre.

##### 11.4.1 Grundoperation, variant og indikatoroperation.

Operationsdelen kan skrives med små eller store bogstaver i flæng, og udover de i kapitel 5 omtalte bogstavkombinationer accepteres 7 pseudooperationer\*), for hvilke SLIP vil indsætte følgende bitkonfigurationer i pos.20-25 (eller i pos.30-35):

---

\*) De 4 af disse, nemlig IL, US, GC og PC, er de nye grundoperationer i GIER-systemer med bufferlager (4096 celler's ferritlager) eller procesenhed.

pseudo- operationer	bitkonfigu- rationer	dec. tal- værdi
IL	101100	44
US	101101	45
ZG	101110	46
GC	101111	47
PC	110000	48
ZJ	111001	57
ZL	111110	62

Herved opnås, at man på ordreform kan indlæse enhver af de  $2^{\uparrow 42}$  mulige bitkonfigurationer i en celle.

S-variant kan noteres enten som S (som tidligere) eller som n (mnemoteknik: nulstilling. Bogstavet lille s kan desværre ikke bruges, da det indgår i sekvensmærkede adresser).

F-variant kan noteres med stort eller lille bogstav.

X-, V- og D-varianter skal (som tidligere) noteres med store bogstaver.

Indikatoroperation skal noteres med store bogstaver som fastlagt i Lærebog I.

Ordenen af de forskellige bestanddele af en ordre er vilkårlig på nær følgende regel: Enhver ordre skal indledes med grundoperationen, og adressedelen skal komme før talletallet; men urtherover kan man blande varianter, indikatoroperation, adressedel og talletal efter behag. Dette hænger sammen med, at opbygningen af bitkonfigurationen svarende til en bestemt ordrelinie foregår ved en række logiske additioner, således at hver gang SLIP har indlæst en bestanddel af en ordre, adderes de dertil svarende bits logisk til det i forvejen indlæste fra samme ordrelinie. Denne fremgangsmåde medfører desuden, at der må anbringes et vilkårligt antal indikatoroperationer og varianter i hver ordrelinie; f.eks. vil fireogtyve X-er og tre n-er i een ordrelinie have samme virkning som eet X og eet n.

### 11.4.2 Adressedel.

I kapitel 3 og 4 har vi indført begreberne absolut adresse, relativ adresse, indexmærket adresse, sekvensmærket adresse, parentesmærket adresse samt talletal; alt hvad vi der sagde om notation og betydning af disse begreber gælder naturligvis også for indlæsning med SLIP, og de regler, vi opstiller i det følgende, er udvidelser af de tidligere indførte:

1) Simpel adressedel: En adressedel kan bestå af

a) r, s, p eller ingenting

b) +, - eller ingenting

c) et heltal eller ingenting.

d) Ovenstående kan eventuelt omgives af en parentes.

Hvis fortegnet mangler mellem a) og c), tolker SLIP dette som et plustegn. Således er f.eks. adressedelene

s24 og s+24

helt ækvivalente. Her er rækkefølgen ikke ligegyldig, idet heltallet skal stå sidst. Ethvert heltal er tilladt, men adressetallet udregnes modulo 1024. En helt tom adressedel er ækvivalent med 0.

2) Navne: I adressedele (og talletal) kan indgå navne, der skal bestå af et af bogstaverne a, b, c, d, e efterfulgt af et heltal, d.v.s. man kan bruge navnene

a0, a1, a2, ...

b0, b1, b2, ...

⋮

e0, e1, e2, ...

(I stedet for a0 kan iøvrigt bruges a, i stedet for b0 b o.s.v.).

Et sådant navn kan defineres, d.v.s. få tillagt en talværdi, på

to måder, nemlig enten ved hjælp af en definitionslinie, eller ved at navnet skrives som etikette foran en linie (se afsnit 11.6.3 Definitionslinie m.v.). Men uanset på hvilken af disse måder navnet defineres, spiller det under indlæsningen af en ordrelinie en rolle, om et dér optrædende navn er defineret tidligere eller om navnet først defineres senere i programmet.

I første tilfælde kaldes navnet et defineret navn, og dets talværdi indsættes straks under indlæsningen af den pågældende adresse; i det andet tilfælde kaldes navnet et undefineret navn, og SLIP må da gemme en oplysning om, i hvilke ordrer det undefinerede navn er benyttet for senere under indlæsningen, når navnet bliver defineret, at kunne indsætte den rigtige talværdi i den pågældende ordre. Denne forskel i SLIPs behandling af et defineret og af et undefineret navn medfører visse forskelle i konventionerne for brugen af definerede og undefinerede navne.

3) Adressedel med undefineret navn: Navne, som først defineres senere i programmet, kan indgå i adressedele af følgende struktur:

- a) r, s, p eller ingenting
- b) + eller ingenting
- c) et undefineret navn.
- d) Ovenstående kan eventuelt omgives af en parentes.

Det bemærkes, at minustegn ikke er tilladt, og at et manglende fortegn tolkes som et plus.

Eksempel 11.2.

Eksempler på adressedele med udefinerede navne er følgende:

c0

s+c1

(pd11) som er ækvivalent med (p+d11).

Derimod er

-c0

c1+3

3+c1

s-d11

ulovlige adressedele, hvis navnene er udefinerede, fordi der hverken må være minustegn eller heltal sammen med et udefineret navn. (En helt anden sag er det, at 3+c1 vil blive opfattet som adressedelen 3 samt talletallet c1 ; se nedenfor i afsnit 11.4.3).

Hvis adressen er relativmærket, subtraheres værdien af r (ordretælleren) fra navnets talværdi, inden værdien indsættes som adressetal. Dette er nødvendigt, for at man kan anvende symbolske relativmærkede adresser på samme måde som simple, relative adresser, nemlig til relativ adressering uafhængigt af programmets placering i lageret.

Eksempel 11.3.

I programstumpen

---

HVS r+3 IZA

AR p+2, GR 124

MK 124, GR 125

AR p+4, MK 126

---

kan man ikke indskyde eller fjerne en ordre uden at ændre den relativmærkede hopadresse  $r+3$ . Hvis man skriver

```

---
HVS r+c1 IZA
AR p+2, GR 124
MK 124, GR 125
c1: AR p+4, MK 126
---
```

vil det have samme virkning som det første program, idet SLIP foretager følgende: Navnet  $c1$  er undefineret under indlæsningen af ordrelinien HVS  $r+c1$  IZA og sættes derfor på venteliste. I den sidste ordrelinie  $c1: AR p+4, MK 126$  virker etiketten  $c1$  som definition af navnet  $c1$ , idet  $c1$  derved defineres som adressen på denne celle (se iøvrigt afsnit 11.6.3). Nu finder SLIP i ventelisten, at navnet  $c1$  er benyttet tidligere og indsætter så den fundne værdi i de adresser, hvor  $c1$  har optrådt. Men da adressen i HVS-ordren er relativmærket, subtraheres først adressen på den celle, hvori HVS-ordren står; altså får ordren det endelige udseende HVS  $r+3$  IZA.

I dette program kan for det første ordrer indsættes eller sløjfes uden videre, idet SLIP under indlæsningen stedse vil sørge for, at  $c1$  bliver erstattet med differensen mellem adresserne på ordren  $AR p+4$  og ordren HVS  $r+c1$  IZA. For det andet kan programmet efter indlæsningen flyttes i ferritlageret takket være den relative adressering.

Hvis man derimod skriver

```

---
HVS c1 IZA
AR p+2, GR 124
MK 124, GR 125
c1: AR p+4, MK 126
---
```

bliver  $c1$  erstattet med den absolutte adresse på den sidste AR-ordre, og dermed er programmet bundet til at skulle ligge på det sted i ferritlageret, hvortil det først er indlæst.



4) Adressedel med definerede navne: Navne, som er defineret tidligere i programmet, kan indgå på flere måder i en adressedel, men den grundlæggende struktur af adressedelene er følgende:

- a) r, s, p eller ingenting
- b) +, - eller ingenting
- c1) et vilkårligt antal definerede navne adskilt med + eller - . Det sidste led (men kun dette) i en sådan række kan også være et heltal.
- d) Ovenstående kan eventuelt omgives af en parentes.

Som ovenfor vil et manglende fortegn under b) blive tolket som et plus. Til forskel fra adressedele med udefinerede navne, hvor der i hver adressedel kun kan optræde eet navn, må der her skrives vilkårlig mange.

#### Eksempel 11.4.

Hvis  $a_3$ ,  $a_7$  og  $b_{14}$  er defineret inden de bruges i adressedele, vil følgende adressedele blive accepteret af SLIP,

$$(a_3 - a_7)$$

$$s + b_{14} + a_3 + 1$$

$$pa_7 - a_3 \text{ ækvivalent med } p + a_7 - a_3$$

Derimod udgør

$$b_{14} - 3 - a_3$$

$$pa_7a_3$$

ikke lovlige adressedele, den første fordi tallet 3 ikke kan efterfølges af leddet  $-a_3$ , den sidste fordi der mangler et fortegn mellem  $a_7$  og  $a_3$ . (En helt anden sag er, at SLIP vil acceptere den første af ovenstående symbolrækker, men opfatte den som bestående af adressedel samt talletal. Den vil give adressedelen  $b_{14}-3$  og talletallet  $-a_3$ , mens den anden vil give fejludskrift.

Den i eksempel 11.4 illustrerede begrænsning af placeringen af heltal i en adressedel kan omgås, idet punkt c) skal udvides med

c2) Ethvert af de under c1) optrædende led kan bestå af et heltal efterfulgt af et defineret navn uden fortegn imellem.

Det "manglende fortegn" vil blive tolket som et plus (og ikke - hvad man måske ville vente - som et multiplikationstegn), og heltal plus navn opfattes som omgivet af en sædvanlig aritmetisk parentes.

#### Eksempel 11.5.

Ifølge regel c2) er adressedelene

$s + 5a7 - a3$

$b14 - 3a3$

$p1b14$

lovlige, og SLIP vil indsætte værdierne af

$5 + a7 - a3$

$b14 - 3 - a3$

$1 + b14$

som adrestetal i de respektive ordreceller (og den første adresse vil blive s-mærket, den sidste p-mærket).

Hvis en adressedel er relativmærket, sker der ganske som for undefinerede navne en subtraktion af r-værdien (ordretællerens indhold), inden værdien af et navn indgår i SLIPs beregning af adrestallet. Er der kun eet navn i adressedelen, og er dettes fortegn +, er virkningen helt som beskrevet i eksempel 11.3; men det må fremhæves, at optræder der flere definerede navne i en

relativmærket adressedel, subtraheres  $r$ -værdien først fra hvert af navnene, og derefter adderes og subtraheres leddene som forlangt. (Resultatet vil formentlig kun sjældent være til nogen nytte.)

#### Eksempel 11.6.

Den sidste af ordrelinierne

a1: PS  $s+1$ , AC 100

ARS  $r+a1-2$  X

vil blive lagret med adressetallet  $-3$ , altså som ordren ARS  $r-3$  X.

SLIP holder under indlæsning rede på, hvilken celle og hvilken tromlekanal, der i øjeblikket indlæses til, og de aktuelle værdier af løbende adresse og løbende kanalnummer, betegnet med henholdsvis  $i$  og  $k$ , kan indgå i adressedele på samme måde som definerede navne:

c3)  $i$  eller  $k$  kan indgå i stedet for ethvert defineret navn under punkterne c1) og c2) ovenfor.

Hvis adressen er relativmærket, foretages dog ingen subtraktion fra  $i$ -værdien eller  $k$ -værdien.

I afsnit 11.6 nedenfor omtales løbende adresse og kanalnummer nærmere.

#### Eksempel 11.7.

Hvis ordrelinien

LK  $i+25$ , VK 0

skal indlæses til celle 100, lagres den i celle 100 som ordrerne LK 125, VK 0; ordrelinien LK  $r+25$ , VK 0 vil derimod blive lagret som ordrerne LK  $r+25$ , VK 0.

5) Positionsangivelse: Ved de ovenfor omtalte adressedele behandles alle leddene som heltal med enhed i position 9 eller - under indlæsning af en højre halvordsordre - i position 19. Man kan imidlertid undertiden ønske at indlæse tal med enhed andetsteds (jfr. f.eks. trimmeparametrene ved taltrykning, kapitel 12). Dette kan gøres ved hjælp af positionsangivelser i adressen:

c4) Ethvert led i en adressedel med definerede navne kan være forsynet med en positionsangivelse, hvilket vil sige, at leddet (af typen c1, c2 eller c3) efterfølges af et punktum og et heltal (uden fortegn).

NB: Et heltal med positionsangivelse behøver ikke at stå sidst i adressedelen.

Virksomheden af en positionsangivelse er, at heltalværdien for det pågældende led anbringes med enhed i den angivne position (med en TK-ordre), inden leddet indgår i adressetalberegningen.\*). Der er intet i vejen for ved hjælp af en positionsangivelse at gå uden for adressepositionerne i cellen.

Man må her være opmærksom på behandlingen af adresser i højre halvordsordrer: Under indlæsningen sker opbygningen af en adresse i adressepositionerne 0-9, dog under hensyntagen til eventuelle positionsangivelser; når adressen er færdig indlæst, skif-

---

\*) Der er herunder en ulogisk forskel i behandlingen af heltal  $\geq 512$  (eller negative heltal) og navne, hvis værdi er  $\geq 512$ : Heltallene behandles som negative tal, som ved skift til højre i cellen bliver suppleret med ettaller foran; værdien af et navn behandles altid som et positivt tal, som ved højreskift bliver suppleret med nuller foran.

tes det hele 10 positioner til højre, såfremt det drejer sig om en højreadresse. Herved virker enhver positionsangivelse altså som om den var 10 større. Den samme behandling får talletal (se iøvrigt nedenfor).

Den under c1, c2, c3 og c4 indførte konstruktion betegnes i det følgende som en <defineret adresse>. En <defineret adresse> er altså en samling led adskilt med + eller -, hvor hvert led kan være

- a) et defineret navn (inklusive i eller k) med eller uden foranstillet heltal og med eller uden positionsangivelse.
- b) et heltal med positionsangivelse.

Sidste led i en defineret adresse kan være et heltal uden positionsangivelse.

#### Eksempel 11.8.

Ordrelinien

PI 1.7+1.8, VY 1.4

vil blive indlæst som ordrerne PI 6, VY 32.

#### Eksempel 11.9.

Ordrelinien

AR (p+1.0+1.1) X

indlæses som ordren AR (p-256) X.

#### Eksempel 11.10.

SLIP holder ikke regnskab med, hvorvidt cellens rammer sprænges ved en positionsangivelse, men udfører slavisk den forlangte skifteoperation (uden afrunding): Lad a1 være defineret som 100. Da vil ordrelinien

MKF a1.2, AR a1.50-1

indlæses som ordren MKF -512, AR -1.

Eksempel 11.11.

I et led af typen c2), d.v.s. heltal efterfulgt af navn som f.eks. 17a1, med positionsangivelse, behandles 17a1 som den enhed, der skal forskydes. Således vil ordrelinien

ARn 17a1.8

indlæses som ordren ARS 234, hvis a1 har værdien 100.

Eksempel 11.12.

Trimmeparametre til taltrykning kan bekvemt indlæses med positionsangivelser. Lad os betragte en trimning til den i næste kapitel omtalte taltrykningssekvens, hvor vi vælger parametrene  $b = 6$ ,  $h = 4$ ,  $f1 = 1$ ,  $d = 3$ ,  $n = 1$ ,  $bE = 3$ ,  $f2 = 2$ ,  $g1 = 4$  og  $g2 = g3 = g4 = g5 = 0$ .

Betydningen af parametrene kan findes i næste kapitel, og der vil man ligeledes finde, at parametrene skal placeres således i cellen:

b med enhed i pos.3	h med enhed i pos.7
f1 - - - - 9	d - - - - 13
n - - - - 14	bE - - - - 17
f2 - - - - 19	g1 - - - - 23

Derfor kan disse parametre indlæses med helordsordren

QQ 6.3 + 4.7 + 1.9 + 3.13 + 1.14 + 3.17 + 2.19 + 4.23

Her er positionsangivelsen .9 forsåvidt overflødig, men udelades den, skal heltallet 1 stå som sidste led i adressen; man kunne således også skrive

QQ 6.3 + 4.7 + 3.13 + 1.14 + 3.17 + 2.19 + 4.23 + 1

(og de øvrige positionsangivelser behøver heller ikke at stå i nummerorden).

### 11.4.3 Talletal.

Som talletal kan man skrive de samme kombinationer som i adressedele på nær r-, s- og p-mærkning samt parentesmærkning.

Et talletal kan således være et

Simpelt talletal bestående af

- a) +, - eller ingenting,
- b) et heltal.

Talletal med udefineret navn bestående af

- a) + eller ingenting,
- b) et udefineret navn.

Talletal med definerede navne bestående af

- a) +, - eller ingenting
- b) en <defineret adresse>, som ovenfor fastlagt.

Optræder der positionsangivelser i et talletal, øges positionen med 10 under indlæsningen (se forrige afsnit).

Adskillelse mellem adressedel og talletal. Da adressedelen i en ordre kan bestå af vilkårligt mange led, og da talletallet kan ligne adressedelen til forveksling, kan det være nødvendigt at kunne adskille dem på bekvem måde. SLIP giver her koderen forskellige muligheder:

a) Talletallet kan indledes med t (lille bogstav). Denne mulighed kan altid bruges, og det kan anbefales at gøre det til en vane at skrive t foran alle talletal.

b) Enhver af variant-betegnelserne S, n, F, f, X, V, D kan skrives mellem adressedel og talletal og fungerer som skilletegn.

c) Enhver indikatoroperation herunder også den blinde indikatoroperation I fungerer som skilletegn (forudsat den noteres mellem adressedelen og talletallet).

d) Hvis adressedelen slutter med et heltal, vil eventuelle efterfølgende led blive tolket som tælle- tal (jfr. den i Lærebog I benyttede notation, f.eks. ordren  $AR\ p+4\ +1$  ).

Eksempel 11.13.

I ordrelinien

VK  $r+a17\ t1$

er adressedelen  $r+a17$  og tælle- tallet  $1$  .

Eksempel 11.14.

I ordrelinien

LY  $a21\ t-1$

er adressedelen  $a21$  og tælle- tallet  $-1$  , mens den samme ordrelinie uden skilletegnet  $t$

LY  $a21\ -1$

vil indlæses med adressedelen  $a21-1$  og tælle- tal  $0$  for- udsat  $a21$  er defineret. (Hvis  $a21$  er et udefineret navn, er ordrelinien i strid med SLIP-syntaxen og vil give fejlud- skrift.)

Eksempel 11.15.

Ordrelinien

BT  $s+5\ -2$

indlæses som operationen BT med adressedel  $s+5$  og tælle- tal  $-2$  og kunne iøvrigt også være skrevet som  $BT\ s+5\ t-2$  eller som  $BT\ s+5\ I\ -2$  , idet I er en virkningsløs indika- toroperation og indlæses som et nul i pos.33-34 (jfr. afsnit 4.9.2 i hæfte I).



Eksempel 11.16.

I ordrelinien

ar a1 n a2

virker operationsvarianten  $n$  (nulstilling) som skilletegn mellem adressedelen  $a1$  og tallet  $a2$ , mens ordrelinien  $ar\ n\ a1a2$  vil give fejludskrift, idet der mangler et skilletegn mellem  $a1$  og  $a2$ ; skriver man  $ar\ n\ a1+a2$  får den indlæste ordre adressetallet  $a1+a2$  og tallet  $0$ , forudsat at  $a1$  og  $a2$  er definerede navne.

11.5 Indlæsning af tal og tekst med SLIP.11.5.1 Talindlæsning.

SLIP kan indlæse tal på fire forskellige måder, nemlig som flydende tal, som maskintal, som heltal eller som heltalsgrupper; valget mellem de fire typer styres dels af det aktuelle tals form og størrelse og dels af administratorerne m og f (m og f er eksempler på styrelinier).

Hvert tal eller hver heltalsgruppe lagres i en helcelle, og som skilletegn mellem tallene kan bruges

- 1) CR eller komma, som samtidig bevirker, at de to mærkebits i den pågældende celle nulstilles.
- 2) Bogstaverne a, b, eller c, som samtidig bevirker, at den pågældende celle a-, b- eller c-mærkes.

Tallene skrives som tal i ALGOL, jfr. definitionen af <number> i ALGOL-rapportens afsnit 2.5, d.v.s. med eller uden decimalpunktum og med eller uden titalsexponent. Iøvrigt indlæses et tal som

flydende tal såfremt den nærmest foregående administrator er

f; en eventuel titals-exponent må ikke overstige 152;

maskintal såfremt den nærmest foregående administrator er m og tallet indeholder et decimalpunktum eller en titals-exponent; tallet 1.0 vil blive lagret som det størst mulige maskintal, nemlig  $1 - 2^{(-39)}$ ; hvis et tal uden for intervallet  $-1.0 \leq x \leq 1.0$  forsøges indlæst som maskintal, giver SLIP fejludskrift (se afsnit 11.8 nedenfor);

heltal med enhed i pos.39 såfremt den nærmest foregående administrator er m og tallet hverken indeholder et decimalpunktum eller en titals-exponent.

En heltalsgruppe består af et eller flere heltal med skråstreger imellem og eventuelt til højre for tallene. Virkningen af en skråstreg er for det første, at tallene indlæses som heltal uafhængigt af den rådende administrator, og dernæst at et tals enerposition skydes 10 positioner til venstre for hver skråstreg, der står tilhøjre for tallet. Er et heltal for stort til den på denne vis tilmålte plads, lagres blot så mange af de bageste cifre i tallet, som der er plads til (d.v.s. de 10, 20 eller 30 bageste, binære cifre). Specielt vil de foranstillede ettal-ler, der er kendetegnet for de negative tal, kun blive sat inden for den tilmålte plads.

NB: En heltalsgruppe må ikke indledes med en skråstreg.

#### Eksempel 11.17.

Styre- og tallinierne

m

1, 1.0,  $50_{10}^{-2}$ , -0.5, 300, 250/

vil forårsage lagring (i 6 helceller) af heltallene 1 og 300

med enhed i pos.39, heltallet 250 med enhed i pos.29, samt maskintallene  $1-2\uparrow(-39)$ , 0.5 og -0.5.

Derimod vil

$$\underline{m}$$

$$-0.1_{10^2}$$

give fejludskrift, fordi intervallet for maskintal overskrides.

#### Eksempel 11.18.

Linierne

$$\underline{f}$$

$$1, 1.0, 50_{10^{-2}}, -0.5, 300, 250/, 0/-200$$

vil forårsage indlæsning og lagring (i 7 helceller) af de flydende tal 1.0, 1.0, 0.5, -0.5 og 300.0 samt lagring af heltallet 250 med enhed i pos.29 og heltallet -200 i positionerne 30-39 med enhed i pos.39.

#### Eksempel 11.19.

Uafhængigt af administratorer vil

$$1// -4a$$

lagres i en helcelle som tallet 1 i venstre halvcelle (pos. 0-19) med enhed i pos.19 og tallet -4 i højre halvcelle (pos.20-39) med enhed i pos.39. Endelig a-mærkes cellen.

$$1025///$$

lagres som tallet 1 med enhed i pos.9, mens resten af cellen nulstilles.

#### 11.5.2 Tekstindlæsning.

Hvis en linie indledes med symbolet  $\underline{t}$ , opfattes det følgende (inklusive eventuelle CR-tegn) til og med det første semi-

kolon som en tekststreng, således at alle de indlæste tegn lagres i successive helceller med 7 tegn pr.celle. I hver celle sker lagringen dog "baglæns", idet det først indlæste tegn lagres i pos.36-41, det næste tegn i pos.30-35 o.s.v. I de 6 positioner, hvert tegn fylder, lagres en bitkombination, der er identisk med tegnets hulkombination, bortset fra paritetspositionen, i flexowriterkoden; den eneste undtagelse fra denne regel er tegnet for CR (der som det eneste tegn benytter den øverste position på strimlen), idet CR lagres som kombinationen 111111.

Under tekstindlæsning overspringes kun tegnene for Tape Feed (7 huller) og All Holes (8 huller).

Når SLIP har indlæst et semikolon, lagres et afslutningssymbol (den ubenyttede kombination 001010 ), og så påbegyndes altid indlæsning til den næste helcelle uanset om den sidste celle er fyldt op eller ej. En tekststreng optager således altid et helt antal celler.

I systemet HJÆLP findes der en sekvens til trykning af tekststreng, der er indlæst med SLIP; se næste kapitel.

#### Eksempel 11.20.

Tekststrengen

t Nu gaar det løs.

juli 1963;

består af 31 tegn (inklusive Case shift, mellemrum, afslutningssymbol o.s.v.) og vil blive lagret i 5 helceller.

## 11.6 Styrelinier og blokke i SLIP.

### 11.6.1 Løbende adresse og løbende kanalnummer.

Indlæsning med SLIP foregår automatisk til successive celler i ferritlageret eller til successive kanaler på tromlen, og SLIP benytter to "visere" til at holde rede på disse adresser:

Løbende adresse  $i$  angiver altid til hvilken ferritlagercelle den næste programinformation indlæses, og hver gang en celle er fyldt op, øges  $i$  med 1.

Man kan tildele  $i$  en ønsket værdi (f.eks. den ønskede begyndelsesadresse for et program) på to måder: 1) Man kan i et blokhoved (se nedenfor) skrive  $i=<defineret\ adresse>$ , hvor  $<defineret\ adresse>$  betegner et udtryk af samme slags som indført i afsnit 11.4.2; 2) Man kan et vilkårligt sted i programmet skrive en linie af formen

$i=<defineret\ adresse>$

og  $i$  begge tilfælde sættes den løbende adresse lig med værdien af den definerede adresse på højre side.

Når man starter indlæsning med SLIP, er den løbende adresse  $i=10$ , og er man tilfreds hermed, behøver man ikke at definere løbende adresse; det indlæste vil da lagres fra celle 10 og fremad.

Løbende kanalnummer  $k$  angiver altid nummeret på den kanal, hvortil den næste programinformation lagres, og hver gang en kanal (med 40 celler) er fyldt op, øges  $k$  med 1.

Man kan kun tildele  $k$  en ønsket værdi ved i et tromleblokhoved (se nedenfor) at skrive  $k=<defineret\ adresse>$ . Herved sættes  $k$  lig med værdien af den definerede adresse.

Når man starter indlæsning med SLIP er det løbende kanalnummer  $k = 294$ . Dette skyldes, at al indlæsning - som omtalt i indledningen - i virkeligheden sker til tromlen, og så længe det drejer sig om programinformation, der skal havne i ferritlageret, sker indlæsningen til ferritlagerbilledet, d.v.s. kanal 294-319. Her behøver koderen kun at specificere løbende adresse  $i$ , idet SLIP automatisk holder rede på tilhørende kanalnumre (og automatisk overfører billedet til den rigtige beliggenhed i ferritlageret, når indlæsningen er tilendebragt). Kun hvis en programdel skal lagres andetsteds på tromlen, er det nødvendigt at definere en  $k$ -værdi; iøvrigt omtales disse forhold (og sammenhængen mellem  $i$  og  $k$ ) nærmere i afsnittet 11.6.4 om tromleblokke. Det skal blot nævnes, at  $i$  i ferritlagerbilledet svarer 1.celle på kanal 294 til celle 0 i ferritlageret, 2.celle på kanalen til celle 1 i ferritlageret o.s.v.

#### Eksempel 11.21.

Definitionslinien

$i = 401$

bevirker, at det følgende indlæses til celle 401 og fremad.

Definitionslinien

$i = i+5$  eller  $i = 5i$

bevirker, at 5 helceller overspringes under indlæsningen.

#### 11.6.2 Blokke.

Programmer, der indlæses med SLIP, kan have en blokstruktur, der minder om blokstrukturen i ALGOL; den væsentligste virkning heraf er at navne, som er erklæret i en blok, er lokale for den-

ne blok på samme måde som i ALGOL. Når der i det følgende tales om blokke, tænkes der på program lagret i ferritlageret; der gælder lidt andre konventioner for tromleblokke som omtales i afsnit 11.6.4.

En blok består af et blokhoved, selve programdelen samt en blokafslutning.

Blokhoved: En blok skal indledes med symbolet b samt eventuel styreinformation om blokkens begyndelsesadresse i ferritlager og om, hvilke navne der bruges i blokken. Blokhovedet skal iøvrigt have følgende udseende:

- a) Symbolet b
- b)  $i = \langle \text{defineret adresse} \rangle$ , hvor  $\langle \text{defineret adresse} \rangle$  betegner et udtryk af helt samme art som i afsnit 11.4.2 ovenfor.
- c)  $i = \langle \text{defineret adresse} \rangle$ .
- d) Et eller flere navne adskilt med kommaer; der kan dog højst være 5 navne, da hvert forbogstav kun må optræde en gang.

Punkterne b), c) eller d) kan eventuelt bortfalde; dog er symbolet b helt alene blind information.

Virkningen af et blokhoved er følgende:  $i = \langle \text{defineret adresse} \rangle$  bevirker, at løbende adresse sættes lig med værdien af den definerede adresse på højre side, d.v.s. at den følgende programinformation indlæses til cellen med denne adresse (og de følgende celler) i ferritlageret. Hvis punkt b) mangler, fortsættes indlæsningen hvor den holdt op umiddelbart før blokhovedet.

Bruger man navne i programdelen af en blok, skal SLIP være adviseret herom i det tilhørende blokhoved eller i hovedet af en

2.

udenom liggende blok ved en "erklæring": Optræder der et navn i et blokhoved, kan man i den tilhørende programdel frit bruge alle navne med samme forbogstav og numre mindre end eller lig det i blokhovedet optrædende nummer (eller dette nummer +1, hvis nummeret er lige). Talværdierne for disse navne skal som tidligere nævnt fastlægges ved hjælp af en definitionslinie eller en etikette. (Navnets optræden i et blokhoved fungerer på samme måde som en erklæring i ALGOL).

Eksempel 11.22.

Et navn i et blokhoved svarer til en erklæring i ALGOL i den forstand, at der derved ikke uddeles talværdier, men blot oprettes mulighed for at benytte visse navne.

Eksempel 11.23.

Blokhovedet

b i=250, a3

bevirker, at den følgende indlæsning foregår til celle 250 ff, og navnene a0, a1, a2 og a3 kan optræde. Blokhovedet

b i=250, a4

bevirker samme indlæsning, hvor navnene a0, a1, a2, a3, a4 og a5 alle er tilladt.

I mange programmer optræder der flere blokke inden i hinanden, og rækkevidden for de erklærede navne svarer helt til forholdene i ALGOL: 1) Er et navn erklæret i en "indre" blok, må det kun bruges i programdelen for denne blok (herunder hører også eventuelle blokke inden i den aktuelle). 2) Er et navn erklæret 2 gange, i en ydre og i en indre blok, fungerer det som to



forskellige navne, hvoraf det ene kun må bruges i den inderste blok, det andet kun i den ydre blok. Det vil sige, at navnet skal defineres (have tillagt en talværdi) to gange, en gang i hvert "niveau", samt at den talværdi navnet har fået i den yderste blok er utilgængelig i den inderste blok og omvendt.

Blokafslutning: En blok afsluttes med følgende:

- a) Symbolet e
- b) en <defineret adresse> eller ingenting.

Virkningen af en blokafslutning er for det første at de navne, der er erklæret i det tilhørende blokhoved, nedlægges, d.v.s. de må ikke bruges mere (medmindre de også er erklæret i en udenom liggende blok), og de talværdier, som navnene (måske) har fået tildelt i blokkens programdel, er utilgængelige. Hvad angår kontroludskrift og opbevaring af SLIPs interne "navnekataloger" henvises til afsnit 11.6.5 og 11.8 nedenfor.

For det andet kan det ske, at indlæsningen afbrydes: Hvis der følger en <defineret adresse> efter e, hoppes til den celle, som den <definerede adresse> angiver, og programmet udføres.

Hvis symbolet e står alene efterfulgt af CR, afbrydes indlæsningen kun, såfremt der ialt er indlæst flere e'er end b'er; i så fald hopper GIER tilbage til det sted i ferritlageret, hvorfra SLIP blev påkaldt (smlgn. afsnit 11.7 nedenfor samt kapitel 13). Så længe antallet af e'er (uden adresse efter) er  $\leq$  antallet af b'er, fortsættes indlæsningen. Se iøvrigt afsnit 11.6.4 om tromleblokke og 11.6.6 om programmer.

Eksempel 11.24.

I programmet

```

  [ b i=100, d3
    d0: PM 700 IPA
      MK (d0), GR (d0)
      [ b a1
        ARS (d0), NK r+a0
        a0: PP 0 t+1
      ]
      e
      GR (d0), ZQ 0
    ]
e 100

```

indsættes navnet d0 med værdien 100 over det hele, fordi navnene d0, d1, d2, d3 kun er erklæret og defineret i den yderste blok. Navnet a0 får værdien 103, men kan kun bruges i den inderste blok.

Eksempel 11.25.

Under indlæsning af programmet

```

  [ b i=100, d3
    d0: PM 700 IPA
      MK (d0), GR (d0)
      [ b d0, a1
        ARS (d0), NK (r+a0)
        a0: PP 0 t+1
      ]
      e
      GR (d0), ZQ 0
    ]
e 100

```

vil GIER ved den inderste bloks afslutning protestere mod brugen af navnet d0 i den inderste blok, hvor det kun er erklæret, men ikke defineret. Derimod vil det både i MK- og GR-ordren samt i den afsluttende GR-ordre blive oversat som adressetallet 100.

Med de angivne blokhoveder har d1 helt samme stilling som d0, idet den indre bloks hoved

b d0, a1

spærrer de "gamle" værdier af  $d_0$  og  $d_1$  ude fra hele den inderste blok. Derimod er  $d_2$  og  $d_3$  brugbare i hele programmet (hvis de blot bliver defineret på et eller andet tidspunkt).

### Eksempel 11.26.

Under indlæsning af programmet

```

[ b i=100, d3
  d0: PM 700 IPA
    MK (d0), GR (d0)
    [ b a1
      ARS (d0), NK (r+a0)
      a0: PP 0 t+1
    ]
    e
    QQ 512 t512
    HV i+3
    [ b i=i+2
      IT (a0), GT d0
      ZQ 0, HV d0
    ]
    e 100

```

vil GIER protestere mod brugen af navnet  $a_0$  i den sidste lille blok. Situationen kan reddes ved at flytte erklæringen af navnet  $a_0$  (og dermed automatisk  $a_1$ ) op i det ydre blokhoved, således at programmet indledes med

```
b i=100, d3, a1
```

thi derved bliver værdien af  $a_0$  tilgængelig i hele programmet.

Formålet med denne indviklede mekanik, der begrænser rækkevidden af navne, er det samme som i ALGOL: Det skal være muligt at inkorporere undersekvenser (i ALGOL: procedurer) skrevet af andre i et SLIP-program, uden at man behøver at spekulere over at undgå de navne, som bruges i undersekvensen. Dette kan opnås hvis alle undersekvenser gøres til blokke, der indledes med erklæringer for de benyttede navne.

### 11.6.3 Definitionslinie og etikette.

Som nævnt i afsnit 11.4.2 ovenfor kan et navn (der skal være erklæret i et forudgående blokhoved) tildeles en talværdi, et heltal i intervallet  $0 \leq t \leq 1023$ , på to måder:

1) I en definitionsline, der ser således ud:

$\langle \text{navn} \rangle = \langle \text{defineret adresse} \rangle$

hvor  $\langle \text{navn} \rangle$  er et af de i afsnit 11.4.2 indførte navne eller bogstavet  $i$  (som betegner løbende adresse). Man kan definere flere navne i samme definitionsline, når blot de enkelte definitioner har ovenstående form og er adskilt med kommaer.

Virksomheden er, at talværdien af den definerede adresse på højre side udregnes, hvorefter navnet på venstre side får tillagt denne talværdi (og i det følgende altså optræder som et defineret navn, se afsnit 11.4.2, punkt 2). \*)

En definitionsline med  $i$  på venstre side bevirker, at den løbende adresse får den talværdi, som den definerede adresse på højre side angiver, og indlæsningen vil da fortsætte fra denne adresse og fremad.

---

\*) Desuden er der følgende meget specielle mulighed for henvisning til et højre halvord:

$\langle \text{navn} \rangle h = \langle \text{defineret adresse} \rangle$

der udover den normale virkning (tillægning af en talværdi medfører følgende: I de forud indlæste ordrer, hvor navnet har optrådt i adressedele som et undefineret navn, ændres grundoperationerne HV og PA til HH og PT. Bemærk, at kun disse to grundoperationer ændres, og det sker kun i det hidtil indlæste, ikke i det følgende, hvor navnet optræder som et defineret navn.

2) Et navn kan også defineres ved en etikette der består i, at navnet efterfulgt af kolon indleder en programlinie, d.v.s. en ordrelinie, en tallinie eller en tekstlinie; en etikette, der skal henviser til et højre halvord, skal også skrives forrest på linien og har normal samme udseende, men kan eventuelt bestå af et navn, bogstavet h samt et kolon. Man kan skrive flere etiketter i samme programlinie; hvert navn skal blot efterfølges af et kolon.

Virkningen af en etikette er, at navnet får tillagt den aktuelle værdi af i som talværdi, altså adressen på den efterfølgende programlinie. Bogstavet h har som ved definitionslinier kun den virkning, at forsåvidt der i forvejen er indlæst HV- eller PA-ordrer, hvor navnet står i adressedelene, ændres disse ordrer til HH- eller PT-ordrer.

#### Eksempel 11.27.

I programstumpen

a1=7

ARS 202 IPA a1

får additionsordren talletallet 7, idet den indlæses;  
i programstumpen

ARS 202 IPA a1

a1=7

sker dette først efter indlæsning af definitionslinien.

#### Eksempel 11.28.

Programstumpen

i=25, a0=1, a1=100

ARS 5a1 ta0

MK 100a1 ta0

HV i-2 NT

indlæses i celle 25, 26, 27 som ordrerne

```
[25] ARS 105 t1
[26] MK  200 t1
[27] HV   25 NT .
```

Eksempel 11.29.

I programstumpen

```
b i=100, a2
a0: PPS 10, PP p-1
    PM p+700, MK p+720
    BS p t0
    HH r+a0
a1: ZQ a1 t1
```

får a0 værdien 100, a1 værdien 104, og efter indlæsning ser ordrerne således ud:

```
[100] PPS 10, PP p-1
[101] PM p+700, MK p+720
[102] BS p+0 t0
[103] HH r-3
[104] ZQ 104 t1
```

Da hopordren er relativmærket, er den løbende værdi af *i*, her *i*=103, subtraheret fra adressetallet *a0*=100 under indlæsningen.

Eksempel 11.30.

Programstumpen

```
i=200
AR (r+b3), GR (r+b3)
HV r+b3 NZ
---
```

b3h:---

```
---
HV r+b3 LO
```

vil indlæses som

```

[200] AR (r+3), GR (r+3)
[201] HH r+2 NZ
      ---
[203] ---
      ---
[205] HV r-2 LO

```

idet kun den hopordre, hvor b3 optræder som undefineret navn, ændres til en HH-ordre på grund af etiketten b3h . Adressedelene i celle 200 henviser som altid til den venstre adressedel i celle 203.

Redefinition. I samme blokniveau kan et navn kun redefinere, d.v.s. få tillagt en ny talværdi, såfremt redefinitionen sker i en definitionslinie. (Derimod er der ingen indskrænkninger med hensyn til nye definitioner i et andet blokniveau forudsat at det pågældende navn er defineret påny.)

Eksempel 11.31.

I programmet (hvor stregerne står for ordrer)

```

b d1
---
d1=14
---
i=100, d1=i-2
---

```

vil d1 , hvor det optræder i 1. og 2.sektion af programmet, blive erstattet med værdien 14, mens d1 i 3.programsektion får værdien 98.

Derimod vil SLIP nok protestere imod programmet

```

b d1
---
d1=14
---
d1: AR 400 t1
---

```

fordi redefinitionen foregår ved hjælp af en etikette. Kun hvis AR-ordren netop indlæses til celle 14, vil SLIP acceptere programmet, idet det er tilladt som kontrol at sætte et defineret navn som etikette for den celle, som talværdien af navnet angiver.

Etikette på tom linie: Står en etikette alene på en linie, knyttes den til den følgende programlinie; der overspringes ikke nogen celle under indlæsningen.

#### Eksempel 11.32.

Programdelen

```
i=50
GR r+a4, MKS p+31
AR r+a4 D
a4:
SR (s-1) t1
```

vil blive indlæst til cellerne 50-52 som ordrene

```
[50] GR r+2, MKS p+31
[51] AR r+1 D
[52] SR (s-1) t1 .
```

#### 11.6.4 Tromleblokke.

Er et program for stort til at ligge helt i ferritlageret, kan man dele det op i tromleblokke, som så lagres på tromlen og successivt kan kaldes ind i samme afsnit af ferritlageret. Syntaktisk set ligner tromleblokkene de tidligere omtalte blokke, idet hver tromleblok består af et blokhoved, en programdel og en blokafslutning; heraf har programdelen og blokafslutningen helt samme struktur som i ferritlagerblokke, og blokhovedet afviger kun fra det i afsnit 11.6.2 indførte ved, at der skal være en angivelse af placeringen på tromlekanalerne:



Et tromleblokhoved består af

- a) Symbolet b
- b)  $k = \langle \text{defineret adresse} \rangle$
- c)  $i = \langle \text{defineret adresse} \rangle$
- d) Et eller flere navne adskilt med komma; der kan dog højst være 5 navne, da hvert forbogstav kun må optræde en gang.

Bestanddelene b), c) og d) adskilles med kommaer. En eller to eller alle tre kan mangle.

Et tromleblokhoved bevirker, at det følgende program indtil den tilsvarende blokafslutning indlæses på tromlen, og denne lagring sker på kanal nr.  $k$  og fremad, idet løbende kanalnummer  $k$  under indlæsningen af punkt  $b$  får den værdi, som den  $\langle \text{definerede adresse} \rangle$  har.

Den løbende adresse  $i$  har ingen direkte indflydelse på lagringen af en tromleblok, thi den starter altid i den første celle på den udpegede kanal, men  $i$  skal som regel svare til den senere placering i ferritlageret: Hvis  $i$  benyttes i adresse- og tælledele eller i definitionslinier i tromleblokken, samt hvis der benyttes etiketter, hvortil der henvises uden relativmærkning, må man sætte  $i$  lig den adresse i ferritlageret, hvortil tromleblokkens 1.ordre senere skal overføres under kørslen.

Uanset om  $i$  er defineret i tromleblokhovedet eller ej, øges løbende adresse på normal vis med 1 for hver celle, der fyldes under indlæsningen, og for hver 40.celle øges  $k$  automatisk med 1. Ved afslutningen af en tromleblok stilles  $i$  og  $k$  tilbage til de værdier, de havde inden tromleblokken (inklusive hovedet) blev påbegyndt; d.v.s. at situationen i ferritlageret

er nøjagtig som om tromleblokken slet ikke var indlæst. (Denne tilbagestilling af  $i$  og  $k$  sker ikke ved afslutningen af en ferritlagerblok.)

### Eksempel 11.33.

Programstykket

```

┌ b k=60, i=100, a7
├ --- } program der fylder 100 celler
├ ---
└ e

```

vil blive indlæst på kanal 60, 61 og halvdelen af 62, og under indlæsningen vil  $i$  successivt antage værdierne 100, 101, ..., 199, men efter indlæsningen af  $e$  vil både  $i$  og  $k$  have samme værdier, som de havde før indlæsningen af hele tromleblokken.

Hvis  $i$  ikke optræder i programmet for tromleblokken, vil blokhovedet

```
b k=60, a7
```

besørge nøjagtig samme indlæsning.

### Eksempel 11.34.

Et stort program kan f.eks. deles i en hoveddel, der altid ligger i ferritlageret, samt to tromleblokke, der deler et afsnit i ferritlageret; det kan da skrives således

```

b i=10, a4, b7, c13
--- } hoveddelen af programmet
---

```

```

┌ b k=60
├ --- } 1.tromleblok
├ ---
└ e

```

```

┌ b k=75
├ --- } 2.tromleblok
├ ---
└ e

```

```
e
```

hvorved 1. og 2.tromleblok vil blive indlæst til kanalerne 60 ff. henholdsvis 75 ff., men for dem begge vil løbende adresse i svare til, at de i ferritlageret placeres i fortsættelse af hovedprogrammet, og hovedprogrammet skal derfor indeholde ordrer, der på passende tidspunkter henter først den ene og senere den anden tromleblok ind på dette samme sted i ferritlageret.

Begrænsninger vedrørende brugen af tromleblokke: 1) Hvis der i en tromleblok bruges globale navne, d.v.s. navne der er erklæret uden for tromleblokken, skal disse være defineret inden de bruges; det er dog ligegyldigt, om definitionen er sket før indlæsningen af tromleblokken, eller om definitionen sker i selve tromleblokken før brugen af de pågældende navne. For navne, der erklæres i selve tromleblokken gælder ingen restriktioner af denne art, men de samme regler som omtalt i de foregående afsnit.

Eksempel 11.35.

I programstykket

```

┌ b a2
├ ---
└ ┌ b k=50
    │ HV r+a2
    │ ARS p-100 LZ
    │ a2: GR 496 t+1
    └ ---

```

vil SLIP protestere mod, at det globale, udefinerede navn a2 optræder i en tromleblok. Men hvis man blot flytter erklæringen og skriver

```

  |  b
  |  ---
  |  |
  |  |  b k=50, a2
  |  |  HV r+a2
  |  |  ARS p-100 LZ
  |  |  a2: GR 496 t+1
  |  |  ---

```

går det godt, fordi a2 nu er et lokalt navn. (Hvis det ikke drejede sig om en tromleblok, var der slet intet problem.)

2) SLIP holder ikke regnskab med det benyttede antal kanaler eller hvorvidt løbende adresse i er rimelig eller ej, og koderen må derfor selv påse, for det første at tromleblokkene ikke griber ind over hinanden på kanalerne, og dernæst at løbende adresse i ikke i løbet af tromleblokken ændres til en værdi mindre end begyndelsesadressen eller til en værdi, der ikke findes i ferritlageret.

3) Det har normalt ingen mening at indlæse tromleblokke til kanal 294-319, da dette tromleleafsnit benyttes af SLIP som et billede af ferritlageret.

#### 11.6.5 Administratorer.

Som nævnt i indledningen samt flere gange i det foregående kan man dirigere nogle af SLIPs funktioner ved hjælp af administratorer, som alle består af et understreget bogstav. Enkelte af administratorerne er omtalt i det foregående, men for oversigtens skyld nævner vi dem alle her sammen med de øvrige styrelinier, enten med en beskrivelse af virkningen eller med en henvisning til en nærmere omtale.

1) Administratorer:

f : Indlæsning af flydende tal (se afsnit 11.5.1).

m : Indlæsning af maskintal og heltal (se afsnit 11.5.1).

t efterfulgt af hvadsomhelst undtagen semikolon: Indlæsning af tekst (se afsnit 11.5.2). \*)

l : Strimmellæseren vælges som indlæseenhed (og skrivemaskinen som udlæseenhed), og indlæsningen fra strimmel starter øjeblikkeligt; dette er normalt den situation, der ønskes under indlæsning af et program.

s : Skrivemaskinen vælges som indlæseenhed (og som udlæseenhed); det vil sige, at når GIER - fra strimmel - har indlæst administratoren s, skrives der et CR på skrivemaskinen, og derefter venter GIER på input fra skrivemaskinen, hvis grønne lampe lyser.

Tastes nu et mellemslag, får man kontroludskrift (med rødt) af løbende kanalnummer k og løbende adresse i. Den samme kontroludskrift fås, hvis man umiddelbart efter en blokbegyndelse eller en blokafslutning taster et CR og et mellemslag.

I alle andre situationer vil tastning af et mellemslag umiddelbart efter et CR give kontroludskrift af i alene.

r : Under den følgende indlæsning bliver alle adressedele, der ikke er s- eller p-mærkede, automatisk relativmærkede. Denne automatiske relativmærkning kan dog undgås ved at indlede adressedele med bogstavet m (på samme plads som r, s og p). Administratoren r er indført, fordi der i mangfoldige program-

---

\*) t er egentlig ikke en administrator i syntaktisk forstand, men er dog taget med i oversigten her.

dele, ikke mindst undersekvenser, vil være mange flere relative end absolutte adresseangivelser. Den automatiske relativmærkning fortsætter, indtil administratoren n indlæses.

n : I det følgende indlæses alle adressedele på normal måde, d.v.s. uden automatisk relativmærkning (og bogstavet m er overflødig, men uskadeligt i de absolutte adressedele). Tilstanden fortsætter, indtil administratoren r indlæses. (Se eksempel 11.37 nedenfor.)

## 2) Øvrige styrelinier:

b i = <defineret adresse>, <navn>, ..., CR: Blokbegyndelse med definition af løbende adresse og med erklæringer (se afsnit 11.6.2).

b k = <defineret adresse>, i = <defineret adresse>, <navn>, ..., CR: Tromleblokbegyndelse med definition af kanalnummer og ferritlageradresse og med erklæringer (se afsnit 11.6.4).

b CR : Blind information.

e CR : Blokafslutning (se afsnit 11.6.2 og 11.6.4).

e <defineret adresse> CR: Blokafslutning samt afslutning af indlæsningen med hop til den celle, hvis adresse er angivet (se afsnit 11.6.2).

<navn> = <defineret adresse>, <navn> = <defineret adresse>, ..., CR: Definition eller redefinition af et eller flere navne, eventuelt af løbende adresse i (se afsnit 11.6.3).

d <navn> = <def.adr.>, <navn> = <def.adr.>, ..., CR: Denne linie har fuldstændig samme virkning som den samme linie uden d (se lige ovenfor). Symbolet d er derfor helt overflødig.

c <defineret adresse> CR: Linien har samme virkning som linien i = <defineret adresse> CR , nemlig en definition af lø-

bende adresse. Linietypen er overflødig, men har den lille fordel at kunne skrives helt i Lower Case (med mindre der forekommer plustegn i den definerede adresse).

x <defineret adresse> CR: Katalogerne over alle erklærede navne og disses eventuelle værdier samt visse administrative oplysninger (om blokniveau og lignende) lagres på 7 successive tromlekanaler, hvor den første er angivet ved den definerede adresse, som bør ligge mellem 58 og 287 inklusive. Skrives administratoren x alene uden nogen adresse efter, lagres katalogerne automatisk på kanal 287-293.

z <defineret adresse> CR: Denne administrator kan kun bruges, når x er benyttet tidligere med samme adresse, thi z bevirker, at navnekatalogerne m.v. hentes fra den angivne kanal (og de følgende 6), og at der sker en fuldstændig retablering af indlæsesituationen ved det tilsvarende x, både med hensyn til valg af indlæse- og udlæse-enheder og med hensyn til de øvrige administratorer. (Se eksempel 11.38 nedenfor.)

u <defineret adresse> CR: Denne styrelinie sætter udhopsadressen fra SLIP lig den angivne definerede adresse. D.v.s. at når der er indlæst tilstrækkelig mange e'er til at indlæsningen afbrydes, vil GIER hoppe til den celle, som udhopsadressen angiver (se afsnit 11.7 nedenfor).

h <hjælpeprogramnavn> CR: Denne styrelinie bevirker udførelse af et hjælpeprogram. Efter denne linie følger ofte nogle styringsparametre for hjælpeprogram. Se iøvrigt kapitel 13 nedenfor.

Alle andre små, understregede bogstaver har samme virkning som s, således at fejlskrivning blot medfører, at SLIP sættes

i ventestilling. Dog er g helt blind information (på grund af et specielt forhold i OLGA-DIG-indlæseprogrammet).

Eksempel 11.36.

Hvis et længere program ønskes delt op på flere små strimler, lader man blot hver strimmel slutte med s. Når een strimmel er indlæst, sættes den næste klar i strimmellæseren, hvorefter man taster l på skrivemaskinen; det vil bevirke indlæsning af den næste strimmel i umiddelbar forlængelse af den foregående o.s.v.

Hvis man vil kontrollere, at indlæsningen sker til det rigtige sted i lageret, kan man taste CR og et mellemslag før man taster l. Thi da får man på skrivemaskinen kontroludskrift af løbende kanalnummer k og løbende adresse i, d.v.s. det sted hvortil indlæsningen af den næste strimmel begynder.

I det hele taget bør alle strimler slutte med s, da dette sætter SLIP i den naturlige ventestilling.

Eksempel 11.37.

Programdelen

r  
a1: ARS a4 t1  
SR (a1) t1  
DK m+100, IT p-1  
BT m+7, HV a1  
HR s+1

a4:

n

vil blive lagret som ordrerne

ARS r+5 t1  
SR (r-1) t1  
DK 100, IT p-1  
BT 7, HV r-3  
HR s+1  
QQ 0

og det afsluttende n medfører, at det følgende program



indlæses på normal måde. Bemærk at administratorerne r og n overhovedet ingen virkning har på adressedele, der er p- eller s-mærket.

### Eksempel 11.38.

Når man under indkørsel af et program skal indsætte rettelser, er det meget bekvemt at kunne benytte de samme navne som under programskrivningen. Det kan man imidlertid kun, hvis man benytter administratorerne x og z, thi ellers går alle oplysninger om navnene tabt ved afslutningen af indlæsningen. Man bør derfor ved udgangen af enhver større blok i programmet indføje en linie

x <kanalnummer> CR

lige før blokafslutningen (og de valgte kanalnumre skal afvige mindst 7 fra hinanden, fordi navnekataloger m.v. fylder 7 kanaler).

Før man indlæser rettelser i en bestemt blok, reetableres SLIP-situationen (navnenes betydning m.v.) ved at skrive linien z <kanalnummer> CR, hvor kanalnummeret skal være det samme som ved det tilsvarende x.

Lad følgende program være indlæst:

```

[ b a7, b9
  a1: ---
  a2: ---
    ---
    [ b a5
      ---
      a1: ---
        ---
        b0: ---
          x           [gem navnekataloger på kanal 287-293]
          e
            ---
            b1: ---
            x 280     [gem navnekataloger på kanal 280-286]
            e a1     [indlæsning slut. Hop til celle nr.a1]

```

Hvis der under kørslen skal rettes i celle nr.a1 i den inderste blok, kan man efter at have afbrudt kørslen (med HP-knappen) taste

$$\underline{z}$$

$$i = a1$$

derefter rettelsen og endelig 3 gange e CR , hvorefter kørslen vil blive genoptaget nøjagtig der, hvor den blev stoppet.

Skal man rette i celle nr.a1 i den yderste blok, tastes

z 280 CR

og derved får man i øvrigt reableret navnene b0 og b1 samt a1 og a2 med disses værdier i den yderste blok.

#### 11.6.6 Programmer.

Når indlæsning med SLIP startes, er situationen som om følgende fiktive blokhovede var indlæst:

b k=294, i=0

i=10

n

m

idet alt program til ferritlageret i virkeligheden i første omgang indlæses til tromlekanal 294-319 og først ved afslutning af indlæsningen bringes på plads i ferritlageret. Herunder svarer celle 0 i ferritlageret til den første celle på kanal 294, men indlæsningen vil først ske til celle 10 (egentlig på kanal 294), fordi celle 0-9 er optaget af den faste SLIP-administration.

Endelig betyder administratoren n , at SLIP er indstillet på indlæsning af "normale" adresser altså uden automatisk relativmærkning, og m betyder, at SLIP er indstillet på at læse maskintal.

Hvis man ønsker at indlæse sit program på disse betingelser og uden brug af navne, behøver man intet blokhovede at skrive, men kan gå lige løs på ordrene, og programstrimlen kan f.eks.

slutte med s , således at man efter indlæsning af strimlen kan afslutte SLIPs arbejde ved at skrive e eventuelt efterfulgt af en adresse (og et CR) på skrivemaskinen. (Se også eksempel 11.39 nedenfor.)

I øvrigt betragter SLIP et program som afsluttet, enten når der er indlæst eet e mere end b'er (hvorved blokstrukturen "går op", når der tages hensyn til det fiktive blokhovede ovenfor), eller såsnart der indlæses et e efterfulgt af en <defineret adresse> (uanset antallet af b'er og e'er).

## 11.7 Indhop og udhop i SLIP.

### 11.7.1 Manuelt styret indlæsning.

Uanset GIERS øjeblikkelige situation (hvadenten maskinen kører eller er stoppet på en eller anden måde) vil et tryk på HP-knappen ved skrivemaskinen udløse følgende funktioner \*) :

1. Det øjeblikkelige indhold af ferritlageret og alle registre lagres på kanal 294-319;
2. På skrivemaskinen udskrives med rødt ordet "hp-knap" efterfulgt af udhopsadressen, d.v.s. den adresse hvortil udhoppet fra SLIP vil ske (medmindre den følgende indlæsning giver SLIP andre direktiver);
3. SLIP kaldes ind i ferritlageret og gør klar til indlæsning ved bl.a. at oprette det ovenfor nævnte fiktive blokhoved;
4. SLIP venter på input fra skrivemaskinen.

---

\*) som beskrives mere detaljeret i kapitel 13.

Hvis man derfor skriver l på skrivemaskinen, vil SLIP påbegyn-  
de indlæsningen af en programstrimmel. Sluttes denne med s ,  
kan man derefter på skrivemaskinen give SLIP det afsluttende e  
(eventuelt med en adresse efter) og et CR , som forårsager di-  
rekte hop til programmet; man kan også ved hjælp af skrivemaski-  
nen indkoble hjælpeprogrammer på dette tidspunkt. En program-  
strimmel kan slutte med e med eller uden adresse efter (samt  
CR), men dette umuliggør manuelle indgreb på dette trin og må  
derfor frarådes.

### 11.7.2 Programstyret indlæsning.

Man kan fra et program hoppe til indlæsning med SLIP på to  
måder:

1) Ordren HSF 2, som vil bevirke de samme 4 funktioner som  
HP-knappen, blot skrives ordet "hsf 2" i stedet for hp-knap ,  
og udhopsadressen angiver her altid cellen efter HS-ordren  
(d.v.s. SLIP opfører sig som en normal undersekvens, der slutter  
med ordren HR s+1 ).

2) Ordren HS 2, som kun udløser funktionerne under punkt 1  
og 3 ovenfor; der kommer ingen udskrift, og den bestående situa-  
tion med hensyn til valg af ydre enheder ændres ikke; som oven-  
for indstiller SLIP sig på at foretage udhop til cellen efter  
HS-ordren.

Udførelsen af hver af disse to ordrer tager ca. 0.5 sekund,  
fordi hele ferritlageret skal overføres til tromlen.

Endelig kommer man (ufrivilligt) ind i SLIP, hvis der opstår  
flydende overløb i et program: GIER hopper som nævnt i opera-  
tionslisten til celle 0, og dette udløser de samme 4 funktioner

som et tryk på HP-knappen, idet ordet hp-knap erstattes af ordet "fl.overløb" efterfulgt af adressen på den ordre, der forårsagede overløbet. (Det betyder, at udhoppet fra SLIP vil ske til den samme ordre, med mindre man angiver en anden udhopsadresse.)

### 11.7.3 Udhop.

SLIP er som andre undersekvenser indstillet på "at vende tilbage hvor den blev kaldt fra", og det betyder, at efter indhop indstiller SLIP sig normalt på et udhop til cellen efter den, hvorfra indhoppet skete. Ved brug af HP-knappen er dette således cellen efter den ordre, som GIER var i færd med i det øjeblik, der blev trykket på HP-knappen.

Hvis det indlæste program slutter med e uden adresseangivelse, vil udhoppet ske som ovenfor nævnt, men man kan ændre udhopsadressen på to måder:

1) Ved et vilkårligt sted i programmet at skrive u efterfulgt af en defineret adresse; herved sættes udhopsadressen til den angivne værdi, og indlæsningen fortsætter indtil et e afbryder den.

2) Ved at afslutte programmet med e efterfulgt af en defineret adresse; herved hoppes straks til den angivne adresse.

I alle tilfælde sker der imidlertid umiddelbart før et udhop det, at ferritlagerbilledet overføres fra kanal 294-319 til ferritlageret, hvorved det, der er forlangt indlæst til ferritlageret, anbringes på sin rigtige plads.

En konsekvens af brugen af dette ferritlagerbillede er, at umiddelbart efter indhop til og umiddelbart før udhop fra SLIP er indholdet i ferritlageret identisk med indholdet i kanal 294-319.

Eksempel 11.39.

Udførelsen af det i eksempel 11.38 angivne program skal starte i celle nr. a1-1 i den inderste blok. Desuden skal der være mulighed for at foretage manuelle indgreb efter indlæsningen. Derfor skal programmet skrives således:

```

[ b a7, b9
  a1: ---
  a2: ---
    [ b a5
      ---
      a1: ---
      ---
      b0: ---
      u a1-1      [sæt udhopsadressen = a1-1]
      x
      e
      ---
      b1: ---
      x 280
      e
      s          [fortsæt indlæsning fra skrivemaskinen]

```

Denne strimmel indlæses ved først at trykke på HP-knappen og derefter taster l. Når strimmelen er indlæst, og den grønne lampe lyser, kan man rette i det indlæste eller påkalde hjælpeprogrammer fra skrivemaskinen; og når man taster et e CR, påbegyndes udførelsen af programmet i den ønskede celle.

11.8 Kontroludskrifter fra SLIP.11.8.1 Fejludskrift.

Under indlæsningen foretager SLIP en syntaktisk kontrol af det indlæste program, og når der opdages en fejl udskriver SLIP på skrivemaskinen oplysning om fejlens art og placering, hvorefter indlæsningen i næsten alle tilfælde fortsætter. For de

fleste fejls vedkommende opnår man herved at kun en enkelt celle får galt indhold (hvilket ofte kan rettes fra skrivemaskinen efter endt indlæsning), samt at man finder alle de syntaktiske fejl ved den første prøve kørsel.

SLIP skelner mellem 9 forskellige fejltyper, men fælles for dem alle gælder følgende om fejludskriften: På skrivemaskinen udskrives (efter et CR ) et fejlnummer samt løbende adresse, og ved indlæsning fra strimmel udskrives endvidere det sidst læste symbol samt de næste tre linier; som regel fortsætter indlæsningen herefter normalt.

I øvrigt er fejlnumrenes betydning og SLIPs reaktion kort beskrevet i nedenstående skema.

Fejlnummer	Betydning	SLIPs reaktion
1	Syntaktisk fejl i ordrelinie, tekstlinie eller styrelinie.	Alt overspringes indtil først følgende kolon, komma, skråstreg eller CR. Derefter fortsættes indlæsningen normalt (o: Som regel får kun een celle galt indhold).
2	Ulovlig brug af udefineret navn i adressedel eller talletal.	Som ved fejl nr. 1.
3	Brug af ikke erklæret navn.	Som ved fejl nr. 1.
4	Fejl i erklæring i blokhovede.	Som ved fejl nr. 1. (o: Normalt overspringes kun den fejlramte erklæring).

Fejl-nummer	Betydning	SLIPs reaktion
5	Ubenyttet hulkombination på strimmel.	Symbolet overspringes, og indlæsningen fortsætter normalt.
6	Redefinition i en etikette, hvor værdien <u>ikke</u> stemmer med den tidligere definition.	Redefinitionen er virkningsløs, og indlæsningen fortsætter normalt.
7	Syntaktisk fejl i tallinie.	Den pågældende celle får et ukontrollabelt indhold, og indlæsningen fortsætter.
8	Tal uden for tilladt talområde (maskintal: $-1 \leq x \leq 1$ flyd.tal: Det flyd.talområde samt exponent $\leq 152$ ).	Som ved fejl nr. 7.
9	Der er brugt for mange navne eller blokke på een gang. (Adressebogen har godt 250 celler, hvoraf der bruges $\frac{1}{2}$ celle til hvert navn 1 - - hver erklæring 1 - - hvert ferritblokhovede 2 celler - - tromleblokhovede).	Indlæsning fra skrivemaskine som efter <u>s</u> . Programindlæsning kan i almindelighed <u>ikke</u> fortsættes med held.

### 11.8.2 Kontroludskrift.

Løbende adresse og løbende kanalnummer: Under indlæsning fra skrivemaskine kan man efter hvert linieskift få udskrevet løbende adresse i ved at taste et mellemslag umiddelbart efter CR. Hvis man taster CR og mellemslag efter indlæsning af et blokhovede (b efterfulgt af erklæringer m.v.), en blokafslutning (e uden adresse), en definitionslinie eller et s, udskrives både løbende kanalnummer og løbende adresse (først k og så i). Kontroludskriften kommer på skrivemaskinen og med rødt.



Værdierne af benyttede navne: Hvis KA eller KB eller begge er etstillet, fås ved hver blokafslutning udskrift af alle navne, der er benyttet i den just afsluttede blok, samt af  $i$  og  $k$ . Udskriften kommer på perforatoren, hvis  $KA = 1$ , og på skrivemaskinen, hvis  $KB = 1$ , og den ser således ud:

- a) adressen på første kanal af den tromleblok, man lige har forladt (eller stadig er inde i).
- b) ferritlageradressen på den lige nedlagte bloks første celle.
- c) løbende adresse, d.v.s. adressen på næste ledige celle i ferritlageret i den just afsluttede blok.
- d) adressen på den kanal, hvortil næste programlinie skal indlæses i den just afsluttede blok.
- e) Alle navne, der er erklæret og benyttet i den just nedlagte blok, efterfulgt af deres værdier.

Eksempel 11.40.

Lad en tromleblok være indlæst med blokhovedet

$\underline{b}$   $k=48$ ,  $i=218$ ,  $a_5$

og lad blokken fylde 8 kanaler og lidt til. Hvis  $KA=1$ , kan man f.eks. få følgende perforatorudskrift efter indlæsning af  $\underline{e}$ :

48	218	320	50
a	225		
a1	228		
a4	318	219	
a5	246		

samt skrivemaskine-udskriften (med sort):

a4 318 219

Denne udskrift viser, at tromleblokken er indlæst på kanal 48-50, svarende til en ferritlagerplacering fra celle

218 til celle 319 inklusive. Endvidere ser man, at a0 , a1 og a5 har værdierne 225, 228 og 246, mens a2 og a3 ikke har været benyttet, og a4 nok er benyttet (sidst i celle 318's adressedel og i celle 219's tælledele), men ikke er blevet defineret i blokken.

Med KA = 0 ville man ved samme blokafslutning kun få fejludskriften

```
a4 318 219
```

på skrivemaskinen.

## 11.9 Syntax for SLIP-kode.

### 11.9.1 Program og blok.

```
<program> ::= <liniegruppe>
```

```
<blok> ::= b<blokhoved>CR<liniegruppe><blok slut> | <tromleblok>
```

```
<tromleblok> ::= b k=<defineret adresse>, <blokhoved>CR  
                  <liniegruppe><blok slut> |
```

```
                  b k=<defineret adresse>CR<liniegruppe>  
                  <blok slut>
```

```
<liniegruppe> ::= <linie> | <blok> | <liniegruppe><liniegruppe>
```

```
<blokhoved> ::= i=<defineret adresse> | <navn> | <blokhoved>, <navn>
```

```
<blok slut> ::= eCR | e<defineret adresse>CR
```

### 11.9.2 Linie.

```
<linie> ::= <programlinie><ny linie> | <styrelinie> |  
          <etikette><ny linie> | <etikette><linie>
```

```
<programlinie> ::= <ordrelinie> | <tallinie> | <tekstlinie> |  
                  <internkodelinie>
```

```
<styrelinie> ::= <definitionslinie><ny linie> | <administrator> |  
                  <hjælpeinformationslinie><ny linie>
```

```
<ny linie> ::= CR | ; <vilkaarlig streng uden CR> CR |  
                  <ny linie><ny linie>
```

<etikette> ::= <navn>:|<navn>h:  
 <navn> ::= <forbogstav><index>  
 <forbogstav> ::= a|b|c|d|e  
 <index> ::= <ciffer>|<index><ciffer>|<tom>  
 <ciffer> ::= 0|1|2|3|4|5|6|7|8|9

### 11.9.3 Ordrelinie.

<ordrelinie> ::= <halvordsordre>,<halvordsordre>|  
                   <halvordsordre>/<halvordsordre>|<helordsordre>  
 <halvordsordre> ::= <grundoperation><halvordsmodif><adressedel>  
                   <halvordsmodif>|<tom>  
 <helordsordre> ::= <grundoperation><bioperation><adressedel>  
                   <bioperation><t><tælletal><bioperation>  
 <grundoperation> ::= <bogstav><bogstav>  
 (dog kun de i kapitel 5 samt afsnit 11.4.1 nævnte kombinationer)  
 <halvordsmodif> ::= <sletningsmærke>|<flydende mærke>|<tom>|  
                   <halvordsmodif><halvordsmodif>  
 <sletningsmærke> ::= n|S  
 <flydende mærke> ::= f|F  
 <bioperation> ::= <halvordsmodif>|<helordsmodif>|  
                   <indik.oper><indik.adresse 1><indik.adresse 2>|  
                   <bioperation><bioperation>  
 <helordsmodif> ::= X|V|D  
 <indik.oper.> ::= I|M|N|L|<tom>  
 <indik.adresse 1> ::= K|Z|O|T|P|Q|R|<tom>  
 <indik.adresse 2> ::= A|B|C|<tom>  
 <t> ::= t|<tom>

### 11.9.4 Adressedel og tælletal.

<adressedel> ::= <relativmærke><adressetal>|  
                   (<relativmærke><adressetal>)

$\langle \text{relativmærke} \rangle ::= m|r|s|p|\langle \text{tom} \rangle$

$\langle \text{tælleetal} \rangle ::= \langle \text{adressetal} \rangle$

$\langle \text{adressetal} \rangle ::= \langle \text{undef.navn} \rangle | + \langle \text{undef.navn} \rangle | \langle \text{fortegn} \rangle | \langle \text{defineret adresse} \rangle | \langle \text{tom} \rangle$

$\langle \text{defineret adresse} \rangle ::= \langle \text{heltal} \rangle | \langle \text{led} \rangle | \langle \text{led} \rangle + \langle \text{defineret adresse} \rangle | \langle \text{led} \rangle - \langle \text{defineret adresse} \rangle$

$\langle \text{led} \rangle ::= \langle \text{symbolsk adr} \rangle | \langle \text{symbolsk adr} \rangle \langle \text{position} \rangle | \langle \text{heltal} \rangle \langle \text{position} \rangle$

$\langle \text{symbolsk adr} \rangle ::= \langle \text{def.navn} \rangle | \langle \text{heltal} \rangle \langle \text{def.navn} \rangle | i | k | \langle \text{heltal} \rangle i | \langle \text{heltal} \rangle k$

$\langle \text{position} \rangle ::= . \langle \text{heltal} \rangle$

$\langle \text{fortegn} \rangle ::= + | - | \langle \text{tom} \rangle$

$\langle \text{heltal} \rangle ::= \langle \text{ciffer} \rangle | \langle \text{ciffer} \rangle \langle \text{heltal} \rangle$

$\langle \text{def.navn} \rangle$  er et  $\langle \text{navn} \rangle$ , hvis talværdi er kendt, idet navnet bruges i en ordre.

$\langle \text{undef.navn} \rangle$  er et  $\langle \text{navn} \rangle$ , hvis talværdi ikke er kendt, idet navnet bruges i en ordre.

#### 11.9.5 Tallinie og tekstlinie.

$\langle \text{tallinie} \rangle ::= \langle \text{tal} \rangle | \langle \text{tal} \rangle \langle \text{terminator} \rangle | \langle \text{tal} \rangle \langle \text{terminator} \rangle \langle \text{tallinie} \rangle$

$\langle \text{tal} \rangle ::= \langle \text{number jfr. ALGOL-rapport afsnit 2.5} \rangle | \langle \text{gruppe} \rangle$

$\langle \text{terminator} \rangle ::= a | b | c | , | \langle \text{terminator} \rangle \langle \text{terminator} \rangle$   
(kun den første terminator efter et tal bestemmer tallets mærkning)

$\langle \text{gruppe} \rangle ::= \langle \text{fortegn} \rangle \langle \text{heltal} \rangle | \langle \text{fortegn} \rangle \langle \text{heltal} \rangle \langle \text{skilletegn} \rangle | \langle \text{fortegn} \rangle \langle \text{heltal} \rangle \langle \text{skilletegn} \rangle \langle \text{gruppe} \rangle$

$\langle \text{fortegn} \rangle ::= + | - | \langle \text{tom} \rangle$

$\langle \text{skilletegn} \rangle ::= / | / \langle \text{skilletegn} \rangle$

$\langle \text{tekstlinie} \rangle ::= \underline{t} \langle \text{vilkårlig streng uden ; ;} \rangle$

( $\langle \text{internkodelinie} \rangle$  beskrives ikke nærmere her, men følger konventionerne for hjælpeprogrammet "kompud", se kapitel 13).

### 11.9.6 Styrelinier.

$\langle \text{definitionslinie} \rangle ::= \langle \text{def.linie } 1 \rangle | \underline{d} \langle \text{def.linie } 1 \rangle |$   
 $\underline{c} \langle \text{defineret adresse} \rangle$

$\langle \text{def.linie } 1 \rangle ::= \langle \text{definition} \rangle | \langle \text{definition} \rangle, \langle \text{def.linie } 1 \rangle$

$\langle \text{definition} \rangle ::= i = \langle \text{defineret adresse} \rangle | \langle \text{navn} \rangle = \langle \text{defineret adresse} \rangle |$   
 $\langle \text{navn} \rangle h = \langle \text{defineret adresse} \rangle$

$\langle \text{administrator} \rangle ::= \underline{f} | \underline{m} | \underline{l} | \underline{s} | \underline{r} | \underline{n}$

$\langle \text{hjælpeinformationslinie} \rangle ::= \underline{x} | \underline{x} \langle \text{defineret adresse} \rangle | \underline{z} |$   
 $\underline{z} \langle \text{defineret adresse} \rangle |$   
 $\underline{u} \langle \text{defineret adresse} \rangle |$   
 $\underline{h} \langle \text{hjælpeprogramnavn} \rangle |$   
 $\underline{h} \langle \text{hjælpeprogramnavn} \rangle / \langle \text{heltal} \rangle$

(se iøvrigt kapitel 13).

## 12. UDLÆSNING.

### 12.1 Indledning.

En meget vigtig del af et program er den eller de sekvenser, der besørger udlæsningen af de fundne resultater på en læselig form; der må altid ofres megen omhu på planlægningen af udlæsningen, således at de ønskede oplysninger trykkes på en hensigtsmæssig og overskuelig måde, og således at der ikke udlæses en mængde overflødige tal.

Når man har fastlagt hvilke resultater der skal trykkes, må man planlægge hvordan de skal trykkes, og her er det praktisk at skelne mellem det enkelte tals form - den lokale typografi der fastlægger antallet af trykte cifre, konvention for fortegnstrykning m.v. - og opstillingen af tallene i tabelform - den globale typografi der fastlægger antal tal pr.linie, antal linier pr. side, placering af tekst imellem tallene m.v.

I systemet HJÆLP findes sekvenser til udlæsning af tal og tekst; disse bliver beskrevet nedenfor, men disse sekvenser tager sig kun af den lokale typografi, og brugeren må selv sørge for administrationen af den globale typografi imellem de enkelte indhop til de omtalte sekvenser. Derfor giver vi i afsnit 12.3.3

et eksempel på et program, der sørger for en pæn tabelopstilling og som benytter de ovennævnte sekvenser som undersekvenser.

## 12.2 Taltrykning med sekvensen i HJÆLP.

### 12.2.1 Funktion.

For hvert indhop i sekvensen udskrives eet tal med et layout der fastlægges ved hjælp af en række parametre. Der er fem mulige indhop, som bevirker udlæsning af indholdet i R-registeret (eller en del deraf eller RF-registeret) på fem forskellige måder; idet sekvensen tænkes lagret fra celle [m] og fremad, viser tabellen de 5 muligheder:

Indhop til	Indholdet af	trykkes som
celle[m+0]	Rpos. 0-9	heltal i intervallet $0 \leq h \leq 1023$
celle[m+1]	Rpos. 0-9	heltal i intervallet $-512 \leq h \leq 511$
celle[m+2]	Rpos. 0-39	heltal i intervallet $-2^{39} \leq h < 2^{39}$
celle[m+3]	Rpos. 0-39	maskintal i intervallet $-1 \leq m < 1$
celle[m+4]	RF-registeret	flydende tal

Det ønskede tal omregnes til decimalrepræsentation og afrundes til det ønskede antal cifre, som derefter trykkes; tallet kan eventuelt blive multipliceret med en opgiven titals-potens før trykningen (se nedenfor under trimming og skalafaktor).

Før indhoppet til sekvensen skal den ønskede udlæseenhed vælges med en VY-ordre i hovedprogrammet; sekvensen forudsætter,

at den valgte udlæseenhed er i Lower Case, men efterlader den altid i Lower Case.

Ved udhoppet fra sekvensen er både R-registeret og M-registeret ændret; skal det udlæste tal bruges senere i programmet, må det derfor lagres inden indhoppet til udlæse-sekvensen.

### 12.2.2 Placering af sekvensen; indhop og udhop.

Sekvensen fylder 120 celler og er lagret på kanal 33-35 i den faste del af systemet HJÆLP. Sekvensen kan placeres hvor som helst i ferritlageret, men fylder her 123 celler fordi den benytter 3 arbejdsceller i umiddelbar forlængelse af de 120 celler. Brugeren må selv kode overførslen fra tromle til ferritlager af de tre kanaler 33-35, og begyndelsesadressen i ferritlager kan altså højst være lig med 900, da celle 1023 er reserveret.

Selve indhoppet fra hovedprogrammet skal ske med en HS-ordre, og sekvensen kræver som programparameter adressen på den celle, hvor trimnings-parametrene (til fastlæggelse af det ønskede layout) findes. Programparameteren og sekvenshoppet kan placeres enten som to halvordsordrer

QQ<trimningsadresse> , HS<indhopsadresse>

eller som en helordsordre efterfulgt af en helords- eller halvordsordre

HS<indhopsadresse>

QQ<trimningsadresse> , ---

Indhopsadressen skal være en af de fem der er nævnt i forrige afsnit, og trimningsparametrene hentes i alle tilfælde fra den celle som trimningsadressen angiver. Da sekvensen gør dette ved



hjælp af ordren ARS (s+0) i første tilfælde og ARS (s+1) i andet tilfælde, må trimningsadressen gerne være parentesmærket eller sligt.

Tilbagehoppet fra udlæsesekvensen sker altid med ordren HR s+1 , og ved tilbagehoppet er R- og M-registeret ændret, mens p-registeret og indikatoren er retableret. Hvis sekvenshoppet står som halvordsordre, retableres sekvensregisteret ikke korrekt; bruges udlæsesekvensen som "undersekvens for en undersekvens", er det derfor nødvendigt at placere sekvenshoppet som helordsordre, altså således

HS <indhopsadr.>

QQ <trimningsadr.> , ---

for at sekvensmekanismen skal fungere korrekt.

#### Eksempel 12.1.

I et program, der ikke bruger celle 900-1022, ønsker man at udlæse maskintallet i celle [a2] med et layout der er lagret i celle [a5]. Man kan da kode således:

```

---
VK 33, LK 900
VK 34, LK 940
VK 35, LK 980      [hent udlæsesekvensen fra tromlen]
VK 0, ARS a2       [R:= det rigtige tal]
HS 903             [hop til trykning]
QQ a5 , ---

```

#### 12.2.3 Trimningsparametre.

Til at fastlægge formen - layout - af det udlæste tal bruger sekvensen 12 parametre som skal vælges af brugeren, og som skal pakkes på en bestemt måde i den celle som <trimningsadressen> angiver. Disse parametre benyttes til at specificere antallet

af betydende cifre der ønskes trykt, det maximale antal cifre foran kommaet, det maximale antal decimaler, det maximale antal cifre i titalsexponenten hvis en sådan ønskes trykt, konventionen for fortegnstrykning både for taldelen og for exponentdelen, en eventuel opdeling af taldelen i ciffergrupper med mellemslag imellem, samt en finesse vedrørende trykning af tallet 0.

Endelig kan man få indholdet i R- (eller RF-) registeret multipliseret med en given titalspotens før trykning, men dette sker ikke ved hjælp af trimningsparametrene (se afsnit 12.2.4 nedenfor).

Et layout kan indlæses som en ordre med en lang række positionsangivelser

$$\begin{aligned} QQ \langle b \rangle . 3 + \langle h \rangle . 7 + \langle f1 \rangle . 9 + \langle d \rangle . 13 + \langle n \rangle . 14 + \langle bE \rangle . 17 + \langle f2 \rangle . 19 \\ + \langle g1 \rangle . 23 + \langle g2 \rangle . 27 + \langle g3 \rangle . 31 + \langle g4 \rangle . 35 + \langle g5 \rangle . 39 \end{aligned}$$

hvor de tilladte værdier og betydningen af de enkelte parametre er følgende:

$0 \leq b \leq 15$  angiver antallet af betydende cifre der ønskes trykt; det vil sige at fra og med det første ciffer der er  $\neq 0$  trykkes der  $b$  cifre med korrekt afrunding af sidste ciffer. Dog trykkes der aldrig mere end  $d$  decimaler, og eventuelt "fyldes der op" med nuller foran decimalpunktet og mellemslag efter (se også nedenfor).

$0 \leq h \leq 15$  angiver det maximale antal cifre før decimalpunktet. Hvis det udlæste tal ikke har så mange heltalscifre, trykkes der til erstatning et antal mellemslag foran tallet.

$0 \leq d \leq 15$  angiver det maximale antal decimaler der skal trykkes. Hvis der er trykt  $b$  betydende cifre inden de  $d$  decimaler er "opbrugt", trykkes der et antal mellemslag tilsidst til erstatning. Selvom der ikke er trykt  $b$  betydende cifre, kan der aldrig trykkes mere end  $d$  decimaler. Hvis  $d = 0$ , udelades trykning af decimalpunktet; hvis  $d > 0$ , men alle de betydende er trykt i heltalsdelen, erstattes decimalpunktet af et mellemslag.

$0 \leq f1 \leq 3$  styrer fortegnstrykningen foran positive taldele samt fortegnets placering: Idet der altid trykkes et minustegn foran negative taldele, er mulighederne følgende fire:

1) Fortegnet trykkes umiddelbart før første ciffer (eller decimalpunktet), og der ønskes

Intet anslag foran taldel  $\geq 0$  :  $f1 = 0$

Mellemslag - -  $\geq 0$  :  $f1 = 1$

Plustegn - -  $> 0$  og mellemslag foran  
taldel = 0 :  $f1 = 2$

2) Fortegnet trykkes som første anslag i trykningen som

Plustegn foran taldel  $> 0$  og mellemslag foran

taldel = 0 :  $f1 = 3$ .

$0 \leq n \leq 1$  har kun betydning for trykning af heltalsnullet for taldele mellem  $-1$  og  $+1$ . Da vil  $n = 0$  medføre at der trykkes  $h$  mellemslag foran decimalpunktet, hvorimod  $n = 1$  medfører at der trykkes  $h - 1$  mellemslag og et nul på dette sted.

$0 \leq bE \leq 7$  angiver det maximale antal cifre i eksponentdelen.

Denne trykkes som symbolet  $10$  efterfulgt af et fortegn og selve exponenten med maksimalt  $bE$  cifre.

Har exponenten færre end  $bE$  betydende cifre, indskydes et antal mellemslag mellem taldelen og symbolet  $10$ , således at sidste ciffer i exponenten altid trykkes i anslag nr.  $bE+2$  efter taldelen.

Er exponenten = 0, men  $bE$  valgt  $> 0$ , trykkes exponentdelen som  $bE+2$  mellemslag.

Vælges  $bE=0$  (eller udelades den helt), trykkes det ønskede tal uden udskillelse af exponentdel (og uden erstattende mellemslag).

$0 \leq f2 \leq 3$  styrer fortegnstrykningen foran exponenten efter helt de samme regler som gælder for  $f1$  (se ovenfor).

$g1, g2, g3, g4, g5$  skal alle ligge i intervallet  $0 \leq g \leq 15$  og styrer en eventuel opdeling af taldelen i ciffergrupper adskilt af mellemslag og decimalpunktet:  $g1$  er antallet af cifre (eller anslag) i 1.ciffergruppe,  $g2$  antallet af cifre (eller anslag) i 2.ciffergruppe o.s.v. Taldelen kan således maksimalt opdeles i 6 grupper, og decimalpunktet opfattes altid som gruppedeler. Hvis taldelen indledes med en række mellemslag på grund af "manglende" heltalscifre, tælles disse med i ciffergrupperne; gruppeinddelingen foretages således altid på de  $h+d$  anslag som taldelen fylder (foruden fortegnsanslag og decimalpunkt).

Eksempel 12.2.

Nogle tal ønskes trykt med 4 betydende cifre uden exponentdel, og tallenes absolutværdier ligger mellem  $10^3$  og  $10^{-1}$ ; plustegn skal erstattes af mellemslag, og der skal ikke være anden grupperdeling end den der besørgeres af decimalpunktet. Altså skal følgende parameterverdier anvendes:  $b = 4$ ,  $h = 3$ ,  $d = 4$  (for at give plads til 4 betydende cifre også for de mindste af tallene),  $f_1 = 1$ ,  $bE = f_2 = 0$  (ingen exponent),  $g_1 = h = 3$ ,  $g_2 = d = 4$  og  $g_3 = g_4 = g_5 = 0$ . Der mangler nu kun parameteren  $n$ , og hvis de ægte decimalbrøker skal udlæses med et nul foran decimalpunktet, sættes  $n = 1$ .

Dette layout kan lagres i f.eks. celle [a5] ved hjælp af følgende ordrelinie

```
a5: QQ 4.3+3.7+1.9+4.13+1.14+3.23+4.27
```

og vil for positive tal i det angivne område bevirke trykning af følgende udseende

```

| ***.* |
| **.** |
| *.*** |
| 0.**** |

```

hvor hver stjerne markerer et ciffer og hvor de lodrette streger markerer hvor trykningen af tallene begynder og slutter: Hvert tal fylder  $h + d + 2 = 9$  anslag.

Det valgte layout svarer iøvrigt fuldstændig til følgende ALGOL-layout `{-ndd.d000}`.

Iøvrigt gælder følgende regler om samspillet mellem de forskellige parametre:

1) Trykningen af taldelen med fortegn fylder normalt  $h + 1$  anslag hvis  $d = 0$  og ellers  $h + d + 2$  anslag, hvor nogle af de

første og nogle af de sidste kan være mellemslag; vælges for fortegnstrykningen  $f_1 = 0$ , vil taldele  $\geq 0$  fylde et anslag mindre (dette bruges når man vil spare anslag ved trykning af positive tal).

2) Trykningen af exponentdelen med fortegn fylder intet hvis  $bE = 0$ , og ellers fylder den normalt  $bE + 2$  anslag, hvoraf nogle eller alle kan være mellemslag (ligesom for taldele vil valget  $f_2 = 0$  medføre, at eksponenter  $\geq 0$  fylder et anslag mindre).

3) Man bør altid vælge  $h + d \geq b$ , da der højst trykkes  $h + d$  cifre i taldelen. Ved trykning uden exponentdel (d.v.s. med  $bE = 0$ ) vil tal med absolutværdi  $\geq 10^h$  - altså tal med mere end  $h$  cifre foran decimalpunktet - blive trykt med det nødvendige antal heltalscifre til den korrekte værdi, men tallet fylder altså flere anslag end planlagt (hvis absolutværdien er  $< 10^{15}$ ; ellers ser sekvensen ikke anden udvej end at forsyne tallet med en exponent).

Tal med absolutværdi  $< 10^{d-b-1}$  - altså tal som ikke har  $b$  betydende cifre foran afskæringen af  $d$  decimaler - vil ved trykning uden exponentdel blive trykt med netop  $d$  decimaler og altså mindre end  $b$  betydende cifre. Meget små tal bliver således trykt som  $0.00\dots 0$  med  $d$  nuller efter decimalpunktet (og de opfattes altid som positive).

4) Ved trykning med exponentdel ( $bE > 0$ ) og hvor  $b = h + d$ , vil taldelen altid have netop  $h$  heltalscifre og netop  $d$  decimaler, der alle er signifikante, og den trykte exponent vil være indrettet herefter. Men hvis  $b < h + d$  er der flere mulige titalsexponenter, der alle vil give det ønskede antal betydende cifre i taldelen indenfor den ramme, som angives af  $h + d$ .

Sekvensen vælger da altid den (entydigt bestemte) exponent, der er et multiplum af  $h + d + 1 - b$  og som giver  $b$  betydende cifre i taldelen. \*)

5) Ofte ønsker man ikke nogen grupperdeling af de udlæste taldele, og da skal man ved trykning af heltal (d.v.s. med  $d = 0$ ) vælge  $g_1 = h$ , mens de fire sidste kan udelades. Ved trykning af decimalbrøker uden grupperdeling skal man sætte  $g_1 = h$  og  $g_2 = d$ , mens de tre sidste kan udelades (d.v.s. nulstilles).

Grupperdelingen bruges kun ved trykning af taldele med mange cifre, hvor en deling kan gøre trykbilledet meget mere overskueligt.

### Eksempel 12.3.

For at illustrere punkt 3 ovenfor vil vi betragte følgende layout

$$QQ\ 2.3 + 4.7 + 1.9 + 2.13 + 1.14 + 4.23 + 2.27$$

d.v.s. vi har valgt  $b = 2$ ,  $h = 4$ ,  $d = 2$ , "sædvanlig" fortegnstrykning, ingen exponent og ingen grupperdeling udover decimalpunktet.

Nedenstående tabel viser udseendet af trykning af forskellige tal med dette layout. De lodrette streger markerer det antal anslag, der bruges til hvert tal:

---

\*) Hvis talværdien er så lille, at dette ikke kan gøres med en exponent med  $bE$  cifre, går det ud over antallet af betydende cifre: Sekvensen vælger den mindste exponent med  $bE$  cifre, som er et multiplum af  $h + d + 1 - b$ , og den tilsvarende taldel trykkes da med færre end  $b$  betydende cifre.

GIER-tal	trykkes som
-0.0123...	-0.01
1.234...	1.2
1234.5...	1200
12345.6...	12000
-12500.0...	-13000

idet de to sidste tal sprænger rammerne, fordi de er numerisk større end  $10^4$ .

#### Eksempel 12.4.

For at illustrere punkt 4 ovenfor vil vi betragte følgende layout

QQ 2.3+2.7+1.9+3.13+1.14+2.17+2.19+2.23+3.27

som svarer til et ALGOL-layout  $\{-nd.000_10+dd\}$ . Med denne trimning vil vi trykke tallene  $1.2468 \times 10^p$ , hvor  $p$  varierer fra  $-4$  til  $6$ . De titalsexponenter, som sekvensen vil trykke disse tal med, vil stedse være multipla af  $h+d+1-b = 2+3+1-2 = 4$ ; desuden afrundes tallene til 2 betydende cifre, og vil blive trykt således:

1.2	$10^{-4}$
12	$10^{-4}$
0.012	
0.12	
1.2	
12	
0.012	$10^{+4}$
0.12	$10^{+4}$
1.2	$10^{+4}$
12	$10^{+4}$
0.012	$10^{+8}$

hvor mellemslaget umiddelbart foran exponenten skyldes, at der er tilladt to cifre i exponenten, idet  $bE = 2$ .

#### 12.2.4 Skalafaktor.

Ønsker man under udlæsning at få tallet i maskinen multipliceret med en skalafaktor, der er en titalspotens - hvis f.eks.



en række maskintal skal multipliceres med 100 under udlæsningen - kan dette gøres med følgende indgreb, der er helt uafhængigt af den valgte trimning: Er den ønskede skalafaktor  $10^H$  anbringes heltallet  $H$  som adressetal i celle 14 i sekvensen, før det normale indhop foretages. Dette adressetal er fra starten nulstillet, men efterlades iøvrigt uændret af sekvensen; har man derfor én gang sat en bestemt skalafaktor i celle 14, er denne virksom ved alle følgende indhop, indtil man enten ændrer indholdet i celle 14 eller påny indlæser sekvensen fra tromlen.

#### Eksempel 12.5.

Et maskintal skal multipliceres med 1000 under udlæsningen, og kun cifrene foran decimalpunktet skal trykkes. Lad udlæsesekvensen være anbragt i celle 900-1022, lad tallet være lagret i celle [a2] og lad celle [a5] indeholde et layout

```
a5: QQ 3.3 + 3.7 + 1.9 + 1.14 + 3.23
    [b = h = g1 = 3, f1 = n = 1, resten = 0]
```

Da kan trykningen kodes således:

```
---, ARS a2
PA 914 t3      [sæt skalafaktoren  $10^3$ ]
HS 903        [tryk maskintal]
QQ a5, PA 914 [fjern skalafaktoren]
---
```

Hvis celle [a2] f.eks. indeholder tallet  $-0.03456\dots$ , vil dette blive trykt som

```
| -35|
```

idet stregerne markerer, hvor trykningen begynder og slutter.

Skal et tal under udlæsningen multipliceres med en skalafaktor, der er en 2-talspotens, kan dette gøres ved at ændre passende på indholdet i M-registerets pos.0-9 - som indeholder 2-talsexponenten for flydende tal - og straks efter hoppe normalt ind i sekvensen til trykning af flydende tal. Det må bemærkes, at ved trykning af flydende tal forudsætter sekvensen ikke, at R indeholder en korrekt normaliseret taldel for et flydende tal; sekvensen opfatter simpelthen indholdet af R som et binært tal med komma mellem pos.11 og 12 (og med R00 som fortegnsviser) og trykker dette under skyldig hensyntagen til exponenten i  $M_{0-9}$ .

#### Eksempel 12.6.

Maskintallet i celle [a2] skal trykkes med skalafaktoren  $2^{15}$ . Dette kan gøres ved at anbringe tallet i R-registeret og exponenten 4 (nemlig  $15 - 11$ ) i  $M_{0-9}$ , og derpå hoppe til trykning af tallet som flydende tal. Med samme lagerdisposition som i tidligere eksempler kan man kode

```

---, ARS a2    [R:= det ønskede tal]
PM 4 D        [sæt exponenten = 4]
HS 904        [tryk som flydende tal]
QQ a5, ---

```

#### 12.2.5 Eksempler på trimninger; specielle muligheder.

I nedenstående skema kan man aflæse virkningen af at trykke forskellige tal med 4 forskellige trimninger; disse er valgt således at de blandt andet illustrerer virkningen af  $f1$ , som styrer fortegnstrykningen, og  $n$ , som styrer trykning af heltalsnullet. Bemærk iøvrigt at sekvensen altid trykker den korrekte talværdi (afrundet inden for de afstukne rammer), men hvis

tallet er anderledes end ventet, kan trykningen blot komme til at fylde flere anslag end planlagt.

Foroven i skemaet er værdierne af de enkelte parametre angivet sammen med det tilsvarende layout i GIER-ALGOL (en tom rubrik betyder parameterværdi 0). De tal, der skal trykkes, er angivet forneden til venstre, og forneden til højre ses resultaterne af udlæsningen, idet de lodrette streger markerer begyndelsen og afslutningen af trykningen for hvert enkelt tal.

b	4	4	4	4
h	2	2	4	2
d	2	2		6
f1	1		2	1
n	1			1
bE				1
f2				2
g1, ..., g5	g1 = g2 = 2	g1 = g2 = 2	g1 = 4	g1 = 2, g2 = g3 = 3
tilsvarende ALGOL-layout	{-nd.dd}	{dd.dd}	{+dddd}	{-nd.dd0 000 <sub>10</sub> +d}
0.012345...	0.01	.01		0.012 35
1.2345...	1.23	1.23	+1	1.235
-1.2345...	-1.23	-1.23	-1	-1.235
-123.456...	-123.46	-123.46	-123	-0.001 235 <sub>10</sub> +5

Ønsker man kun at få trykt exponentdelen af et tal, kan det opnås ved at vælge  $b = h = d = 0$ , d.v.s. trykning af en taldel med 0 betydende cifre og 0 cifre både før og efter decimalpunktet. Der er kun den ulempe at alle tal behandles som positive, d.v.s. der trykkes et mellemslag eller et plustegn (afhængig af f1) foran tallet i exponenten uanset tallets fortegn i GIER.

Som det fremgår bl.a. af ovenstående skema er sekvensen normalt indstillet på at trykke tallene med "komma under komma".

Imidlertid kan den også styres til at trykke tal med "første betydende ciffer under første betydende ciffer" på bekostning af at trykningen ikke fylder lige mange anslag for alle tal; vælger man nemlig  $h=0$  men dog  $b>0$ , vil ethvert tal, der er numerisk  $\geq 1$ , sprænge rammerne i det valgte layout, og ciffertrykningen vil altid begynde "helt til venstre". Adressetallet i celle 0 i sekvensen vil iøvrigt ved udhop være øget med det antal anslag, der er benyttet.

#### Eksempel 12.7.

Ved trykning af en række heltal ønskes første betydende ciffer under første betydende ciffer, og alle cifre i tallene skal udlæses. Derfor sættes  $h=0$  og  $b=15$ , det maximale antal der kan vælges; da det drejer sig om heltal, sættes  $d=0$  samt  $n=1$ ,  $f_1=1$  (normal fortegnskonvention),  $b_E=f_2=0$  (ingen exponent),  $g_1=15$  og  $g_2=g_3=g_4=g_5=0$  (ingen gruppdeling). Denne trimning kan f.eks. resultere i følgende udskrift:

0
-1
12
-123
1234

af tallene 0, -1, 12, -123 og 1234 .

### 12.3 Teksttrykning med sekvensen i HJÆLP.

#### 12.3.1 Funktion.

For hvert indhop i sekvensen udskrives en tekstlinie indlæst med SLIP, d.v.s. sekvensen udskriver en tro kopi af, hvad der er blevet indlæst mellem et t og det først følgende semikolon (eksklusive disse tegn).

Før indhoppet skal den ønskede udlæseenhed vælges med en VY-ordre i hovedprogrammet; sekvensen forudsætter at den valgte udlæseenhed er i Lower Case, men efterlader den altid i Lower Case.

Ved udhoppet fra sekvensen er både R-registeret og M-registeret ændret, mens de øvrige benyttede registre er retableret.

### 12.3.2 Placering af sekvensen; indhop og udhop.

Sekvensen fylder 13 celler og er lagret i celle 27-39 på kanal 16 i den faste del af HJÆLP. Sekvensen kan placeres hvorsomhelst i ferritlageret, og brugeren må selv kode overførslen af kanal 16 til ferritlageret; sekvensens begyndelsesadresse i ferritlageret kan altså højst være lig med 1010 (og under indlæsningen af kanal 16 berøres da 40 celler fra 983-1022).

Indhoppet fra hovedprogrammet skal ske med en HS-ordre til celle 0 i sekvensen, og den kræver som programparameter adressen på den første af de celler hvor teksten er lagret. Lige som for taltrykningssekvensen kan programparameteren og sekvenshoppet placeres enten som to halvordsordrer

QQ <tekstadresse>, HS <indhopsadresse>

eller som en helordsordre efterfulgt af en helords- eller halvordsordre

HS <indhopsadresse>

QQ <tekstadresse>, ---

Tekstadressen må gerne være parentes-, r-, s- eller p-mærket, fordi sekvensen benytter en parentesmærket adresse (s+0) eller (s+1) til at hente den lagrede tekst.

Tilbagehoppet sker altid med ordren HR s+1, og da er R- og M-registeret ændret, mens p-registeret og indikatoren er retable-

ret; hvis sekvenshoppet står som halvordsordre, reableres sekvensregisteret dog ikke korrekt (se den tilsvarende bemærkning i afsnit 12.2.2 ovenfor).

### Eksempel 12.8.

Hvis man i et program ønsker udskriften "den gik, Granberg." på skrivemaskinen, kan man kode således:

```

---
VK 16, LK 983      [læs kanal 16]
VY 1.5 t7          [input uændret, output på skr.mask.]
VK 0, SY 58        [vent til kanal 16 er læst, vælg
                    Lower Case]

HS 1010
QQ b2, ---
---
```

b2: t den gik, Granberg.;

Hvis ordren SY 58 blev erstattet af SY 60 [vælg Upper Case], ville sekvensen udskrive "DEN GIK, Granberg.", fordi den af SLIP indlæste og lagrede tekst ikke begynder med et Lower Case-tegn men med tegnet for d respektive D. Det første Lower Case-tegn findes mellem G og r.

### 12.3.3 Eksempel på benyttelse af sekvenserne i HJÆLP.

Vi vil illustrere brugen af udlæsesekvenserne i en større sammenhæng ved at gennemgå dele af et program, der i forbindelse med en vis beregning skal trykke en tabel af følgende udseende:

```

          HOVEDLØS. 28.4.1772

t         x         y         teta
0         _____
0.1       _____
0.2       _____
--        ----
2.0       _____
afvigelse = _____
```

hvor stregerne står for de beregnede værdier af x, y og teta

samt afvigelse; x, y og teta beregnes som maskintal og skal trykkes med følgende layout  $\{-n.dddd\}$ , mens afvigelse (også maskintal) skal trykkes på formen  $\{+n_{10}+d\}$ ; t beregnes som flydende tal og skal åbenbart trykkes som  $\{n.d\}$ .

Lad t blive lagret i celle nr.c0, og lad tilsvarende værdier af x, y og teta blive lagret i cellerne c1, c1+1 og c1+2.

Man kunne da i hovedtræk kode som følger:

```

    b a3, b4, c1
    ---
    VK 16, LK 860 [bl.a.definition af navnet c1]
    VK 33, LK 900 [hent tekst-sekvensen til celle 887-899]
    VK 34, LK 940
    VK 35, LK 980 [hent tal-sekvensen til celle 900-1019]
    VY 1.4 t7 [vælg perforatoren til udlæsning]
    SY 64, SY 64 [2 gange CR]
    HS 887
    QQ b0, SY 64 [tryk overskriften samt CR]
    VK 0 [vent til kanaltransport er afsluttet]
    PA a3 t20 [retabler linietælleren]
  a0: ---
    --- [beregning af en t-, x-, y- og teta-
    værdi]
    SY 64, ARSF c0 [tryk CR]
    HS 904 [tryk en t-værdi]
    QQ b2, SY 0
    PA a2 tc1-1 [retabler adressen på x]
  a1: SY 0, SY 0
  a2: ARS c1-1 t1 [R:= x eller y eller teta]
    HS 903 [tryk R]
    QQ b3, ARS a2
    NC c1+2, HV a1 [hop til a1 de første 2 gange]
  a3: BT 20 t-1
    HV a0 [hop til næste linie de første 20 gange]
    SY 64, SY 64 [2 gange CR]
    HS 887 [tryk den sidste tekst]
    QQ b1, ARS c1+3
    HS 903 [tryk afvigelse]
    QQ b4, SY 64 [tryk et afsluttende CR]
    SY 11, --- [og et Stop Code]
    ---
  b0: t HOVEDLØS. 28.4.1772

    t x y teta;
  b1: t afvigelse = ;
  b2: QQ 2.3 + 1.7 + 1.13 + 1.14 + 1.23 + 1.27 [layout {n.d}]
  b3: QQ 5.3 + 1.7 + 4.13 + 1.9 + 1.14 + 1.23 + 4.27 [layout {-n.dddd}]
  b4: QQ 1.3 + 1.7 + 2.9 + 1.14 + 1.17 + 2.19 + 1.23 [layout {+n10+d}]
    ---

```

Bemærk at i denne kode skal navnet `c1` være defineret inden indlæsningen af det midterste programstykke (nærmere betegnet inden indlæsningen af PA-ordren i celle nr.a1-1).

Den afsluttende udlæsning af tegnet for Stop Code bevirker blot, at når resultatstrimlen senere aflæses i en flexowriter, stopper denne automatisk ved dette tegn.



## 13. HJÆLPEPROGRAMMER; HJÆLP.

### 13.1 Kontrolfunktioner.

Under indkørsel af programmer er det af meget stor betydning at have et system af hjælpe- og kontrolprogrammer, som kan lette fejlfindingen og fejlretningen. For at disse programmer skal være en effektiv hjælp for GIER-operatøren må det være let at indkoble og styre dem under selve kørslen, og derfor er HJÆLP-komplekset opbygget sådan at det kan betjenes udelukkende via HP-knappen og skrivemaskinen. Desuden er det imidlertid ofte bekvemt at kunne foretage forud planlagte indgreb på anden vis, og derfor kan hjælpeprogrammerne også styres via strimmellæseren samt ved hjælp af specielle ordrer anbragt i programmet.

Et sådant system bør kunne udføre mange forskellige kontrolfunktioner på meget flexibel måde, men samtidig må det ikke blive for kompliceret at betjene: Antallet af hjælpeprogrammer må ikke være alt for stort, fordi man så ikke kan huske dem alle, og antallet af parametre til styring af de enkelte programmer må af samme grund ikke være for stort.

Resultatet må derfor blive et kompromis mellem disse ønsker, men systemet bør kunne udføre følgende funktioner:

- 1) Initialisering af hele maskinen.
- 2) Indlæsning og udlæsning af programmer, data m.v. såvel på extern (let læselig) form som på komprimeret form (beregnet til hurtig ind- og udlæsning).
- 3) Indsættelse af rettelser i et indlæst program.
- 4) Kontroludskrift af register- og lagerindhold samt af ændringer, der er sket i lageret under kørslen.
- 5) Overvågning af et kørende program, dels af programmets administrative funktion og dels af programmets numeriske behandling af data.
- 6) Hjelpeprogrammer bør kunne indkobles og udkobles på ethvert tidspunkt af kørslen uden at ødelægge det kørende program.

Som det vil fremgå af det følgende, opfylder HJÆLP-komplekset stort set disse krav, og det er desuden meget nemt at udvide systemet med nye hjelpeprogrammer. I afsnit 13.4 gennemgås de standardhjelpeprogrammer der indgår i komplekset, og sidst i afsnittet beskrives, hvordan komplekset kan udvides vilkårligt; i afsnit 13.5 gennemgås forskellige fejlreaktioner med en omtale af hvordan disse kan imødegås. Men først vil vi beskrive HJÆLPs centrale mekanik samt de forskellige indhop i systemet.

### 13.2 HJÆLPs centrale administration.

Set fra koderens synspunkt er HJÆLP at betragte som en samling af undersekvenser, der kan indkobles på et vilkårligt tidspunkt af kørslen enten ved hjælp af HP-knappen eller ved hjælp af en speciel ordre i programmet (programmeret indhop). Desuden

foretages der automatisk indhop i HJÆLP hvis der opstår overløb under flydende regning.

Samlingen af hjælpeprogrammer styres af en fælles administration, og indlæseprogrammet SLIP er nærmest en undersekvens for denne centrale administration: Al indlæsning - både af programmer og af styringsinformation - sker ved hjælp af SLIP, og al indlæsning afsluttes ligesom al hjælpeprogramaktivitet med et udhop via den centrale administration. Ved indhop i HJÆLP sættes udhopsadressen normalt lig adressen på ordren efter den, hvorfra indhoppet er sket. Hvis udhopsadressen ikke er blevet ændret i mellemtiden, fortsætter GIER altså der, hvor programforløbet blev afbrudt af et indhop i HJÆLP.

Prisen for at kunne benytte faciliteterne i HJÆLP er, at den disponible lagerplads indskrænkes, da systemet jo også skal være i lageret. I GIER med 1 tromle benytter HJÆLP følgende dele af lageret:

Celle 0-9 samt celle 1023 til den del af administrationen,

der altid skal være tilstede i ferritlageret.

Kanal 0 til den helt centrale administration.

Kanal 1-37 til en række faste sekvenser, herunder SLIP og de i kapitel 12 omtalte udlæsesekvenser.

Kanal 38 er en arbejds kanal, bl.a. ved benyttelse af HP-knappen.

De øvrige standardhjulpeprogrammer er normalt lagret på kanal 39-57 eller på kanal 191-209; i det sidste tilfælde kan ALGOL-oversætteren ligge på kanal 39-190 uforstyrret af HJÆLP. Disse hjælpeprogrammer kan eventuelt placeres hvorsomhelst på tromlen (se afsnit 13.4).

Kanal 294-319 bruges som ferritlagerbillede, hvor HJÆLP ved indhop gemmer hele ferritlagerets og alle registres indhold, og hvorfra ferritlageret og registrene reetableres ved udhop fra HJÆLP.

Den disponible lagerplads for koderen er således i ferritlageret celle 10-1023 og på tromlen kanal 58-293, i nødstilfælde kanal 39-293.

Det vil under omtalen af de forskellige hjælpeprogrammer nedenfor fremgå, at hvis et program ikke bruger al pladsen på tromlen kan det desuden være bekvemt at lade nogle af de første kanaler efter nr.57 samt nogle af de sidste før ferritlagerbilledet stå til disposition for hjælpeprogrammerne.

### 13.2.1 HP-knappen.

I kapitel 11 er HP-knappens funktion omtalt ganske kort; mere detailleret bevirker et tryk på HP-knappen følgende:

- 1) Den igangværende ordre fuldføres, herunder også en igangværende tromletransport eller en ydre enheds funktion. (Er GIER i forvejen stoppet udføres straks punkt 2.).
- 2) I by-registeret etstilles by[0] , hvorved HP-knappen sættes ud af funktion, således at et nyt tryk på knappen er virkningsløst.
- 3) Indholdet i celle 0-39 gemmes på kanal 38 og kanal 0 indlæses til celle 0-39.
- 4) Indholdet i r1-registeret (ordretælleren) lagres som adressetal i celle 0, og GIER hopper til celle 1.

Forudsat at kanal 0 indeholder HJÆLPs centrale administration sker der derefter følgende:

- 5) Der udføres sumcheck på indholdet i de låsede kanaler 0-31. (Hvis denne ikke stemmer, skrives "FEJL"; se iøvrigt afsnit 13.4 nedenfor.)
- 6) Der udskrives: CR , "hp-knap" efterfulgt af udhopsadressen (med rødt) og CR . Et h efter udhopsadressen betyder, at den just udførte ordre var et venstre halvord, og at GIER ved udhoppet fra HJÆLP vil starte med det tilsvarende højre halvord. Adresser  $\geq 512$  udskrives som deres negative komplement.
- 7) Indholdet i ferritlageret og alle relevante registre ved indhoppet i HJÆLP gemmes på kanal 294-319. De første 10 celler på kanal 294 får dog altid det faste indhold, som er nødvendigt for at HJÆLP kan fungere.
- 8) I pos.20-39 i celle 1023 i ferritlageret sættes en speciel bitkonfiguration (GK, VY r) , som bevirker at et nyt tryk på HP-knappen ikke forårsager ny gemning af ferritlager og registre.

I by-registeret nulstilles by[0] , hvorved HP-knappen atter bliver aktiv (jfr. punkt 2 ovenfor).

- 9) Der foretages indhop til SLIP, som derefter venter på input fra skrivemaskinen som om følgende blokhovede var indlæst:

```

b k=294, i=0
i=10
m n s

```

(se iøvrigt kapitel 11).

Bemærkninger: A) Der findes altså to forskellige spærringer: Den ene sætter ved hjælp af by[0] HP-knappen ud af funktion under udførelsen af ovenstående punkt 3-8 som tilsammen varer

ca.  $\frac{1}{2}$  sekund; dette er nødvendigt for at sikre at et utidigt tryk på knappen ikke kan have ødelæggende virkninger midt under gemningen af ferritlageret m.v. Tilsvarende er HP-knappen sat ud af funktion i ca.  $\frac{1}{2}$  sekund under retableringen af ferritlageret umiddelbart før udhoppet fra HJÆLP (se nedenfor).

Den anden form for spærring er tilstede fra ferritlageret er blevet gemt (punkt 8 ovenfor) og indtil ferritlageret retableres lige før udhoppet fra HJÆLP; denne spærring sikrer (ved hjælp af indholdet i celle 1023) at når ferritlageret een gang er gemt på kanal 294-319, sker dette ikke mere selvom man trykker på HP-knappen før retableringen er foretaget. Desuden ændres udhopsadressen i celle 9 ikke sålænge spærringen står i celle 1023.

B) Af ferritlageret gemmes og retableres indholdet i celle 10-1023, mens celle 0-9 altid får følgende indhold; celle 0-5 indeholder den fælles indhop-mekanisme, der benyttes ved programmeret indhop i HJÆLP, og celle 6-9 den fælles udhop-mekanisme, der benyttes ved alle udhop fra HJÆLP:

- [0] IT -1, IT -1 [indhop v. flyd.overløb]
- [1] IT -2, PT 0 [indhop v.hp-parenteser og program-  
styret hp-valg]
- [2] GK 1, VY 529 [indhop til SLIP-indlæsning; knap-  
spærring]
- [3] VK 318, SK 960
- [4] VK 25, LK 960
- [5] VK 317, HV 960 [hop til SLIP m.v.]
- [6] LK 960, VK 0 [fælles udgang af HJÆLP]
- [7] QQ 0 N [benyttes ved hp-parenteser]
- [8] VY 0 t511 [frigør knap-funktion]
- [9] QQ , HV (r) [udhop fra HJÆLP]

I celle 9 sættes udhopsadressen som venstre adressetal; hvis indhoppet er sket fra en venstre halvordsordre, ændres hopordren til HH (r) \*).

C) HP-knappen fungerer altså hvis blot indholdet på kanal 0 er intakt. Hvis indholdet af kanal 1-31 er intakt, fungerer også SLIP m.v., og ellers udskrives ordet FEJL.

D) Hvis lampen "HP-knap spærret" ikke lyser og HP-knappen dog ikke virker, skyldes det som regel at GIER står midt i en ordre som ikke kan fuldføres. Det kan dreje sig om en ordre med en udefineret grundoperation, en tromleoperation på en ikke-eksisterende kanal, en uendelig parenteskode, eller det kan skyldes at GIER venter på en ydre enhed. I alle tilfælde vil et tryk på RESET gøre at HP-knappen atter virker.

E) Mens knap-funktionen er spærret, husker GIER dog hvis der trykkes på HP-knappen, og så snart spærringen ophæves, udføres den normale knap-funktion. HJÆLPs centrale administration sørger derfor for, at trykker man to eller flere gange hurtigt efter hinanden på knappen har det helt samme virkning som eet tryk. Desuden sørger administrationen for, at hvis man trykker på HP-knappen under retableringen af ferritlageret lige før udhoppet, kommer man atter ind i HJÆLP, men udhopsadressen er uændret (og ikke altid 9 som man kunne frygte, fordi knapspærringen hæves i celle 8).

F) Ligeledes sørger HJÆLPs administration for, at trykker man på HP-knappen mens man er inde i HJÆLP (f.eks. under indlæsning), sker der næsten ingen skade takket være spærrekonfigura-

---

\*) og desuden b-mærkes celle 9 af kodetekniske grunde.

tionen i celle 1023: Ferritlagerbilledet lades urørt, derunder også udhopsadressen i celle 9, og der gøres som sædvanlig klar til indlæsning fra skrivemaskine til celle 10 i ferritlageret (egentlig ferritlagerbilledet). Dog kan man risikere at noget af det sidst indlæste går tabt, hvis man trykker på knappen under indlæsning: Det indlæste skal havne på tromlen, enten i ferritlagerbilledet eller andetsteds; under indlæsningen opsamles det imidlertid i ferritlageret og overføres først til tromlen, hver gang 40 celler er fyldt op, eller når der indlæses en definition af  $i$ , et blokhovede eller en blokafslutning. I alle andre tilfælde vil et tryk på HP-knappen bevirke, at det, der er indlæst men endnu ikke overført til tromlen, går tabt; det kan i uheldigste fald dreje sig om henved to kanalers indhold (fordi SLIP benytter to kanalbuffersektioner i ferritlageret).

G) De registre, hvis indhold gemmes og retableres, er alle de for koderen relevante: R-registeret, M-registeret, Overløbsregisteret, indikatoren, p-registeret, s-registeret, by-registeret og tk-registeret.

### 13.2.2 Udhop.

Udhop fra HJÆLP (og SLIP) foretages, når der enten er indlæst et  $e$  efterfulgt af en adresse eller når der er indlæst flere  $e$ 'er end  $b$ 'er. Når alt det indlæste er bragt på plads på tromlen, spærres knap-funktionen, hvorefter alle registre retableres og ferritlagerbilledet overføres til ferritlageret; GIER hopper til celle 6, hvor den sidste kanal af billedet indlæses til ferritlageret; derefter frigøres knapfunktionen (i celle 8) og udhoppet sker altid fra celle 9, hvor udhopsadressen



står som venstre adressedel, mens højre halvord enten er HV (r) eller HH (r) afhængig af om returhoppet skal ske til en venstre eller en højre halvcelle.

Mens HP-knappen virker helt uafhængigt af indholdet i ferritlageret og altid sætter det korrekte indhold i celle 0-9, fungerer de i de følgende afsnit beskrevne indhop kun rigtigt, hvis indholdet i celle 0-5 er intakt. I celle 6-8 (eller mere korrekt celle 0-8) sættes umiddelbart efter indhoppet det foreskrevne indhold, men i celle 9 sættes kun udhopsadressen som adressedetal\*). Derfor skal celle 9 være intakt for at udhop fra HJÆLP skal foregå korrekt. (At celle 9 ikke retableres skyldes, at visse hjælpeprogrammer indsætter specielle ordrer her.)

#### Eksempel 13.1.

Skrivemaskineudskriften (med rødt)

hp-knap 25h

viser, at der blev trykket på HP-knappen, mens GIER enten var ved at udføre den venstre halvordsordre i celle 25 eller et højrehop med resulterende adresse 25.

#### Eksempel 13.2.

Hvis man under kørsel trykker på HP-knappen og derefter taster e CR , fortsætter kørslen aldeles uforstyrret.

---

\*) og eventuelt ændres pos.33, der angiver forskellen mellem HV- og HH-operationen, samt pos.41 af kodetekniske grunde.

### 13.2.3 Programmeret indhop.

Programmeret indhop i HJÆLP kan foretages på tre måder, hvoraf den første har næsten samme virkning som HP-knappen:

A) Ordren HSF 2 bevirker:

- 1) HP-knappen sættes ud af funktion, idet by[0] etstilles.
- 2) Der udskrives: CR, "hsf 2" efterfulgt af udhopsadressen (med rødt) og CR . Udhoppet vil altid ske til næste venstre halvcelle eller helcelle, som om det blev foretaget med den normale returhopordre HR s+1 .
- 3) Ferritlageret og alle relevante registre gemmes på kanal 294-319. Dog får celle 0-8 altid det faste indhold, og i celle 9 indsættes udhopsadressen.
- 4) I celle 1023 i ferritlageret sættes spærrekonfigurationen, HP-knappen frigøres, og der foretages indhop til SLIP nøjagtig som punkt 8-9 ved brug af HP-knappen (side 133).

B) Ordren HS 2 bevirker indhop til HJÆLP (og SLIP) uden udskrift ved indhoppet og uden nyt valg af ydre enhed; der udføres altså de samme funktioner som under punkt 1, 3 og 4 ovenfor, idet by-registeret dog er urørt (pånær pos.0).

### Eksempel 13.3.

Anbringes ordren HS(F) 2 som venstre halvord i en celle, bliver den tilsvarende højre halvordsordre oversprunget ved udhoppet fra HJÆLP. Desuden bringes der uorden i sekvensmekanismen, fordi i s-registeret indsættes adressetallet fra denne højre halvordsordre; dette er nøjagtig samme virkning som hvis udhoppet foregik med ordren HR s+1.

C) Den tredje mulighed for programmeret indhop er at benytte ordren HS 1 efterfulgt af styreparametre til et hjælpeprogram. Denne bevirker - uden indhopsudskrift og uden mulighed for skrivemaskineindgreb - følgende:

- 1) HP-knappen sættes ud af funktion, mens ferritlageret og registre gemmes på kanal 294-319.
- 2) Spærrekonfigurationen i celle 1023 sættes, og HP-knappen frigøres.
- 3) Programparametrene efter HS-ordren vælger og styrer udførelsen af et hjælpeprogram.
- 4) Der foretages normal retablering og udhop til den første celle efter programparametrene.

Iøvrigt beskrives dette nærmere nedenfor i afsnit 13.3.3.

#### 13.2.4 Flydende overløb.

Opstår der overløb ved en af de flydende operationer (ARF, SRF, MKF eller DKF), hopper GIER til celle 0; forudsat at celle 0-5 er intakt sker der følgende:

- 1) HP-knappen sættes ud af funktion.
- 2) Der udskrives (med rødt): CR, "fl.overløb" efterfulgt af adressen på den ordre, der har givet overløb, og CR . Hvis dette er en højre halvordsordre, skrives et h efter adressen.
- 3) Ferritlageret og alle relevante registre gemmes på kanal 294-319. Dog får celle 0-8 altid det faste indhold, og i celle 9 indsættes den udskrevne adresse.

- 4) I celle 1023 i ferritlageret sættes spærrekonfigurationen, HP-knappen frigøres, og der foretages indhop til SLIP nøjagtig som punkt 8-9 ved brug af HP-knappen (side 133).

Bemærk, at foretages der udhop fra HJÆLP i denne situation uden at udhopsadressen er ændret, vil GIER gentage den ordre der frembragte overløb; dette er som regel meningsløst, blandt andet fordi R-registerets adressepositioner er ændret (jfr. beskrivelsen af ARF i operationslisten, Lærebog I, side 67).

### 13.3 Styring af hjælpeprogrammer.

#### 13.3.1 Hjælpeinformationslinier. Styreparametre.

Valg af hjælpeprogram sker med hjelpeinformationslinier, som har formen

h <hjulpeprogramnavn>

ved skrivemaskinestyret eller strimmelstyret indkobling af hjælpeprogrammer, og som har formen

h <hjulpeprogramnavn>/<adressetal>

ved programstyret indkobling af hjælpeprogrammer (i forbindelse med ordren HS 1 ).

Efter en hjælpeinformationslinier følger som regel nogle linier med styreparametre for det pågældende hjælpeprogram. Konventionerne for disse styreparametre gennemgås i forbindelse med de enkelte hjælpeprogrammer i afsnit 13.4 nedenfor, men de er alle udformet som sædvanlige ordrelinier eller tallinier og indlæses på normal måde af SLIP.

Løvrigt har denne styringsinformation nøjagtig samme form uanset på hvilken måde hjælpeprogrammet vælges; men det skal understreges, at formen på den enkelte styreparameterlinie er valgt ud fra mnemotekniske hensyn, og enhver sådan linie kan udformes som en vilkårlig SLIP-linie, blot denne ved indlæsningen frembringer samme bitkonfiguration.

### 13.3.2 Skrivemaskine- eller strimmelstyret udførelse.

Efter indhop i HJÆLP enten med HP-knappen, med ordren HSF 2 eller med flydende overløb, kan et hjælpeprogram sættes i sving ved at taste

h <hjelpeprogramnavn>

Passende styreparametre for det valgte hjælpeprogram

e CR

Når e CR er tastet, kontrolleres checksummen for det valgte hjælpeprogram, og derefter udføres dette øjeblikkeligt; da hjælpeinformationslinien her har omtrent samme syntaktiske stilling som et tromleblokhovede, bevirker e CR at denne pseudoblok nedlægges, og HJÆLP er derfor i helt samme situation som før hjælpeinformationslinien: De indlæste styreparametre findes ikke lagret nogetsteds (i hvert fald ikke tilgængeligt for koderen), og løbende adresse og kanalnummer er uændrede.

Strimmelstyring: Ønsker man ikke at taste styreinformationen, kan man indlæse en strimmel med samme udseende blot suppleret med et s (eller eventuelt e CR): Efter indhop i HJÆLP som ovenfor tasteres l, hvorved styrestrimmelen indlæses; hjælpeprogrammet udføres, og det afsluttende s bringer kontrollen

tilbage til skrivemaskinen, idet HJÆLPs situation er helt som ovenfor efter tastning af e CR .

#### Eksempel 13.4.

Efter et tryk på HP-knappen ønskes udskrift af celle 100-108 på ordreform. Når HJÆLP med rødt har skrevet f.eks.

hp-knap -501

som tegn på at udhoppet vil ske til celle 523, taster man på skrivemaskinen

[	<u>h</u> tryk	[som vælger tryk-programmet]	
	gp 100 t 108	[som trimmer "tryk" til at udskrive	celle 100-108]
	<u>e</u> CR	[som aktiverer "tryk"]	

Når denne udskrift er foretaget, er situationen igen som efter benyttelsen af HP-knappen, og tastes endnu et e CR , fortsætter GIER i celle 523 som om intet var hædt.

Hvis man i læseren lægger en strimmel med følgende udseende

h tryk  
gp 100 t 108  
e  
e CR

vil tastning af l frembringe helt samme virkning som ovenfor.

#### 13.3.3 Programstyret udførelse.

Ønsker man på bestemte steder i et program at kunne afbryde kørslen for at foretage indgreb der afhænger af kørslens forløb, anbringer man på disse steder ordren HSF 2 ; hver gang GIER kommer til denne ordre, får man adgang til at dirigere udviklingen fra skrivemaskinen.

Hvis man derimod et sted i programmet ønsker at foretage forud planlagte indgreb, kan man ligeså godt indføje al styreinformation på det pågældende sted i programmet i følgende skikkelse:

```

---
HS 1      [udfør følgende hjælpeprogram]
h <hjulpeprogramnavn>/<antal programparametre>
Passende styreparametre for det valgte hjælpeprogram
---
```

Her betyder <antal programparametre> det antal celler (det vil som regel sige linier), som de følgende styreparametre fylder; styreparametrene opgives på helt samme form som i afsnit 13.3.2 ovenfor.

Derimod skal der her ikke skrives e CR som afslutning på styreinformationen, fordi denne type hjælpeinformationslinie med skråstreg ikke sidestilles med et blokhovede.

Alt det indlæste lagres som normal SLIP-information, hvoraf hjælpeinformationslinien fylder 1 celle \*). Angivelsen af antallet af programparametre sætter HJÆLP i stand til at foretage udhoppet til den første celle efter styreparametrene.

---

\*) Linien lagres som halvordsordrerne NC, QQ hvor adressetallet for NC er "tværsommen" af hjælpeprogramnavnet, mens adressetallet for QQ er antallet af programparametre. Det, der skrives efter skråstregen, indlæses som et normalt talletal.

Eksempel 13.5.

Hvis man efter at have trykket på HP-knappen taster

```
i = 150
hs 1
h tryk /1
gp 100 t 108
ar 45
```

vil dette blive indlæst og lagret i celle 150-153. Når GIER da under programudførelsen når frem til celle 150, bliver indholdet i celle 100-108 udskrevet, hvorefter GIER fortsætter i celle 153.

Hvis man derimod taster

```
i = 150
hs 1
h tryk
gp 100 t 108
e
ar 45
```

vil HJÆLP øjeblikkelig udskrive indholdet af celle 100-108, mens ordrene hs 1 og ar 45 vil blive lagret i celle 150-151 (og hvis GIER senere sættes til at udføre ordren i celle 150, vil man få fejludskrift, fordi den næste celle ikke har det forventede indhold, nemlig en NC-ordre m.v.).

13.3.4 Rettelser.

Som det implicit fremgår af det foregående, behøves der ikke noget specielt hjælpeprogram til indlæsning af rettelser i lageret, fordi man fra skrivemaskine eller strimmel kan indlæse SLIP-information direkte til tromlen eller ferritlageret (egentlig ferritlagerbilledet). Vi skal dog minde om at ved at bruge styrelinierne

x <kanaladresse>    og    z <kanaladresse>

kan man genoprette de under indlæsningen benyttede navne med de



samme værdier, hvilket kan være en stor hjælp ved indlæsning af rettelser (sammenlign omtalen i kapitel 11 side 97 ).

Derimod kan indlæsning af rettelser i registerindhold kun ske ved anvendelse af hjælpeprogrammet "ret" som omtales nedenfor.

Fortryder man under valg eller under udførelse af et hjælpeprogram dette, kan man nårsomhelst trykke på HP-knappen. Så snart den aktuelle ordre er fuldført, foretages det normale knap-indhop i HJÆLP, men takket være spærrekonfigurationen i celle 1023 lades ferritlagerbilledet (og dermed udhopsadressen) urørt fra forrige indhop \*). Derved glemmes det før valgte hjælpeprogram totalt, og SLIP er påny klar til skrivemaskineindlæsning til celle 10 eller til nyt hjælpeprogramvalg.

#### Eksempel 13.6.

Efter indhop i HJÆLP på en eller anden måde taster man f.eks.

```

h tryk
gp 100 t 405
e

```

hvorefter HJÆLP begynder at udskrive 306 cellers indhold på skrivemaskinen. Så snart fejltagelsen opdages, må man trykke på HP-knappen hvorved udskriften afbrydes, og HJÆLP er atter klar som om "tryk" ikke var valgt. Man kan da taste den korrekte styreinformation, f.eks.

---

\*) Dette gælder ikke for brug af knappen under udførelse af et overvågningsprogram, thi når først dette er valgt og startet, er man helt ude af HJÆLP, og derfor er også spærrekonfigurationen fjernet.

h tryk  
gp 400 t 405  
e

der forårsager udskrift af 6 cellers indhold.

### Eksempel 13.7.

Hvis man taster den samme information som i eksempel 13.6, men opdager fejlen før der er tastet e, altså efter indlæsning af

h tryk  
gp 100 t 405

kan man gøre et af to:

Et tryk på HP-knappen efterfulgt af tastning af et vilkårligt tegn (som får GIER til at fuldføre den LY-ordre, den venter i) forårsager tilbagevenden til klar-tilstanden som før "tryk" blev valgt.

Man kan imidlertid også rette i den indlæste styreinfor-  
mation ved at afslutte ordrelinien med CR og derefter  
taste

i = i - 1      [sæt løb.adresse tilbage]  
gp 400 t 405 [den korrekte styring]  
e

hvorefter hjælpeprogrammet udføres.

## 13.4 Standard-hjælpeprogrammer.

### 13.4.1 Placering af hjælpeprogrammer.

Som tidligere omtalt beslaglægger HJÆLP normalt kanal 0-57, men af disse kan man eventuelt disponere over kanal 39-57 ved enten at undvære de hjælpeprogrammer, der normalt ligger på disse kanaler, eller ved at placere dem andetsteds på tromlen.

Den faste del af HJÆLP, kanal 0-37, omfatter HJÆLPs administration, SLIP, sekvenser til udlæsning af ordrer, tal og tekst (de to sidste er omtalt i kapitel 12), samt hjælpeprogrammerne h start, h kontrol og h slip, som omtales nedenfor.

Resten af standardudgaven af HJÆLP består af følgende hjælpeprogrammer (med angivelse af deres normale placering):

<u>h</u> tryk og <u>h</u> sam	fylder 9 kanaler, lagret på kanal	39-47
<u>h</u> kompu	d - 4 kanaler, - - -	48-51
<u>h</u> ret	- 1 kanal, - - -	52
<u>h</u> hent og <u>h</u> gem	- 1 kanal, - - -	53
<u>h</u> hp ind og <u>h</u> hp ud	- 3 kanaler, - - -	54-56
<u>h</u> læs hp	- 1 kanal, - - -	57

Af hensyn til ALGOL-oversætteren, som ligger på kanal 39-190, findes der også en version af HJÆLP, hvor disse hjælpeprogrammer er placeret (i samme orden) på kanal 191-209, mens den faste del stadig ligger på kanal 0-37. Oversættelse af et ALGOL-program vil dog som regel berøre kanal 191-209. Alle disse hjælpeprogrammer kan placeres hvorsomhelst på tromlen og ikke nødvendigvis i forlængelse af hinanden; programmerne findes på hver sin strimmel og kan indlæses til vilkårlige tromlekanaler ved at taste c <begyndelseskanal>CR og l (se også nedenfor). På samme måde kan supplerende hjælpeprogrammer indlæses i vilkårligt antal, gerne oven i allerede indlæste programmer; blot skal hjælpeprogrammet "læs hp" være intakt, da det bruges under indlæsningen.

HJÆLP fører hele tiden et katalog over navnene på de indlæste hjælpeprogrammer og deres placering. Hver gang et hjælpeprogram påkaldes, undersøger HJÆLP om navnet findes i kataloget; i modsat fald fås alarmudskriften "tomt hp", hvorefter HJÆLP

igen er klar til indlæsning. Hvis navnet findes, kontrolleres checksummen, og stemmer denne ikke udskrives "sumfejl", hvorefter HJÆLP atter er klar til indlæsning (se iøvrigt afsnit 13.5).

Man må her være opmærksom på følgende forhold vedrørende kataloget over hjælpeprogrammer:

1) HJÆLP opdager ikke at man indlæser et nyt hjælpeprogram til en allerede benyttet kanal, men udvider blot kataloget med det nye programs data. Først når man prøver at benytte det tidligere indlæste hjælpeprogram, protesterer HJÆLP på grund af gal checksum. Men ønsker man at anbringe nye hjælpeprogrammer oven i tidligere indlæste, går alting således glat, sålænge man ikke prøver at bruge de gamle hjælpeprogrammer.

2) Indlæser man et nyt hjælpeprogram med samme navn som et allerede eksisterende, placeres det nye program på den forlangte plads; der rettes i kataloget så det stemmer med det nye programs placering og checksum.

3) Kataloget kan nårsomhelst kontrolleres af "kontrol" men kan kun initialiseres af "start", som derefter kontrollerer checksummerne for alle standardhjelpeprogrammerne (se nedenfor).

#### 13.4.2 "kontrol" og "start".

1) Som ovenfor nævnt kan man kontrollere hvilke hjelpeprogrammer, der er indlæst på tromlens uaflåse del, ved hjælp af "kontrol". Dette hjælpeprogram kan bruges helt uden eller med 1 parameter, hvis udseende iøvrigt er ligegyldigt, og virkningen er:

Uden parameter: Checksummerne kontrolleres for alle de hjælpeprogrammer der er opført i kataloget, og der udskrives en liste over begyndelseskanalnummeret og navnet på hvert program i kataloget; for de løse hjælpeprogrammer udskrives dog kun begyndelseskanalnummeret.

Hvis en checksum ikke stemmer, udskrives med rødt ordet "sumfejl" foran det pågældende programs kanalnummer og navn.

Med 1 parameter, f.eks. QQ: Den samme kontrol udføres, men nu omfatter udskriften kun de programmer for hvilke checksummen ikke stemmer.

Den første plads i kataloget vedrører den faste del af HJÆLP på de uaflåsede kanaler 32-37. Af kodetekniske grunde behandles imidlertid kanal 28-37 som eet hjælpeprogram med navnet "uaflåsede" (der aldrig bør ændres af brugeren) og den første linie i den udskrevne liste er derfor: "28 uaflåsede". Hvis alle standardhjelpeprogrammerne ligger på deres normale plads og kataloget ikke er udvidet, ser listen iøvrigt således ud:

```

28 uaflåsede
43 sam
39 tryk
48 kompud
57 læs hp
53 hent og gem
54 hp ind
55 hp ud
52 ret
sumfejl 45 algol

```

ALGOL-oversætteren er opført i kataloget som et hjælpeprogram på linie med de øvrige, men da den deler plads med nogle af disse, kan den ikke være intakt samtidig med hjælpeprogrammerne; derfor

får man udskriften "sumfejl 45 algol" \*).

Hvis standardhjælpeprogrammerne er indlæst til kanal 191-209 og ALGOL-oversætterens kanal 45 er intakt, vil listen se således ud:

```

28  uaflaasede
195 sam
191 tryk
200 kompud
209 læs hp
205 hent og gem
206 hp ind
207 hp ud
204 ret
45  algol

```

2) "start" er et initialiseringsprogram, som bringer GIER i en veldefineret start-tilstand før indlæsning af nye opgaver. "start" kan bruges helt uden eller med 1 parameter, hvis udseende iøvrigt er helt ligegyldigt, og virkningen er:

Uden parametre: I den disponible del af lageret (d.v.s. celle 10-1022 samt kanal 58-319) sættes ordren HSF 2 i hver helcelle; alle registre nulstilles (dog sættes 17 i by-registeret og 10 i r1-registeret); kataloget over hjælpeprogrammer initialiseres, d.v.s. i kataloget opføres standardhjælpeprogrammerne placeret på kanal 39-57; listen over hp-parenteser nulstilles (se afsnittet om "hp ind"). Derpå udføres "kontrol" med udskrift af kataloget.

---

\*) Iøvrigt kontrollerer HJÆLP kun checksummen af kanal 45, som indeholder det centrale indhop i oversætteren. Resten kontrolleres af oversætteren selv.

Med 1 parameter, f.eks. QQ: Samme virkning som "start" uden parametre, blot omfatter katalogudskriften kun de programmer for hvilke checksummen ikke stemmer.

Bemærk, at hvis et af standardhjælpeprogrammerne er ødelagt, kan "start" ikke retablere det, men giver udskriften "sumfejl" efterfulgt af navnet og begyndelseskanalnummeret.

Der udskrives altså den på side 149 angivne liste, eventuelt med "sumfejl" foran en eller flere af linierne.

### Eksempel 13.8.

Under prøvekursel af et program benyttes hverken "kompud" eller "ret", men til gengæld et hop-overvågningsprogram "hop", som fylder 5 kanaler.

Efter at have tastet

h start  
e

og sikret sig at standardudgaven af HJÆLP er intakt, lægger man strimlen med hjælpeprogrammet "hop" i læseren og taster

c 48  
l

hvorved det indlæses til kanal 48-52, og det kan nu bruges præcis som de øvrige hjælpeprogrammer. Til gengæld vil et forsøg på at bruge "kompud" eller "ret" give fejludskrift, fordi checksummerne ikke stemmer. Påkaldes "kontrol" senere ved at taste

h kontrol  
e

får man følgende udskrift:

```

28  uaflaasede
43  sam
39  tryk
sumfejl 48 kompud
57  læs hp
53  hent og gem
54  hp ind
55  hp ud
sumfejl 52 ret
sumfejl 45 algol
48

```

### 13.4.3 "tryk".

Hjælpeprogrammet "tryk" kan udskrive vilkårlige dele af lager og registre på mange forskellige former, og derfor er konventionerne for styreparametrene ret indviklede; bl.a. er der pakket flere styreinformationer sammen i hver parametercelle. Derved er "tryk" i stand til at udskrive lager- og registerindhold på ordreform (med forskellige varianter) og som decimale tal på 4 forskellige måder (med mange varianter), men derimod hverken på binær, oktal eller nogen lignende form.

Parametrene skrives på ordreform, og der skelnes mellem 3 typer af parametre, nemlig områdespecifikation (1 helcelle), basistrimming (1 helcelle) og numerisk trimming (2 helceller); disse typer beskrives hver for sig nærmere nedenfor. "tryk" kan kaldes med vilkårligt mange parametre, således at man i een omgang kan få udskrevet flere dele af lageret på forskellige former, hvis det er ønskeligt; blot skal der være mindst een områdespecifikation (som fortæller hvad der skal udskrives).

A) Områdespecifikation kan være to halvordsordrer eller en helordsordre under en af formerne

```

<udskrift-form>, <registernavn>
<udskrift-form> <beg.adresse>t<slutadresse>
<udskrift-form> <kanalnr>.39 + <beg.adresse>t<slutadresse>

```



hvor <udskrift-form> og <registernavn> kan være visse operationsdele, mens de øvrige størrelser kan være vilkårlige adressetal.

Enhver områdespecifikation indledes altså med en operationsdel der angiver udskriftsformen, d.v.s. hvorvidt udskriften skal ske på ordreform, som flydende tal, som maskintal, som heltal eller som heltalsgruppe (formen kan så yderligere specificeres ved hjælp af basistrimming og numerisk trimming). Virkningen af denne venstre operationsdel fremgår af nedenstående skema.

<udskrift-form>	Celle-indholdet udskrives som
gp gpr	ordrer ordrer med tilføjelse af absolut adresse (i kantet parentes) for alle relativ- mærkede ordrer
gr grp grn	flydende tal med 5 betydende cifre - - - 10 - - - - - layout og skalafaktor angivet i en numerisk trimming *)
gm gmp gmn	maskintal med 6 decimaler - - 12 - - - layout og skalafaktor angivet i en numerisk trimming *)
gi gin	heltal (enhed i pos.39) med 13 cifre - ( - - - ) - layout og skala- faktor angivet i en numerisk trimming *)
ga	heltalsgruppe, d.v.s. 4 heltal i intervallet $0 \leq t \leq 1023$

Ved udskrift af tal udskrives altid mærkningen af de respektive celler (eller R-registeret) som a, b, c eller et komma, således at de udskrevne tal eventuelt kan genindlæses af SLIP.

\*) Se afsnit C nedenfor.

Resten af områdespecifikationen angiver hvad der skal trykkes, og her skelnes der mellem registerudskrift, ferritlagerudskrift og tromlelagerudskrift:

Registerudskrift fås med en af områdespecifikationerne

<udskrift-form>, <registernavn> eller , <registernavn>

hvor <registernavn> kan være de i skemaet på modstående side nævnte operationsdele. Hvis <udskrift-form> udelades, gælder den sidst angivne udskrift-form.

Ferritlagerudskrift fås med områdespecifikationen

<udskrift-form><beg.adresse>t<slutadresse>

som bevirker, at indholdet i cellerne fra og med nr.<beg.adresse> til og med nr.<slutadresse> udskrives på den angivne form. Udelades slutadressen (eller sættes den lig med 0), udskrives blot een celle.

Tromlelagerudskrift fås med områdespecifikationen

<udskrift-form><k>.39 + <b>t<s>

som bevirker udskrift (på den angivne form) af cellerne fra og med nr.<b> til og med nr.<s> på kanal nr.<k>. Adresserne <b> og <s> må gerne være >39, idet der så blot tælles frem på de næste kanaler, men både <b> og <s> skal ligge i intervallet  $0 \leq t \leq 1023$ . Udelades <s> (eller sættes den lig med 0), udskrives blot celle nr.<b> på kanal nr.<k>. (NB: Hvis der er brugt en basistrimning med basisadresse  $\neq 0$ , udskrives fra og med celle nr.<b>-<basisadresse> på kanal nr.<k>; dette svarer til at <b> er ferritadressen, hvis kanal nr.<k> placeres i ferritlageret fra celle nr. <basisadresse> og fremad; se også nedenfor under punkt B, basistrimning.)

<registernavn>	Udskrift af	Formen af udskriften
gr grf gm	R - registeret (incl. mærkebits) RF-registeret ( - - - ) M - registeret	} afhænger af <udskrift-form> .
gs gp ga gk	s - registeret p - registeret udhops-adressen ("r1-register") tk-registeret	} heltal, $-512 \leq t \leq 511$ . } Uafhængig af <udskriftform>.
gi	indikatorregisteret incl. KA og KB samt overløb-registeret	1) indikatorrens pos. 0-9 som heltal, $-512 \leq t \leq 511$ 2) - - 0-9 - 5 2-bits-grupper *) 3) indikatorrens KA, KB som en 2-bits-gruppe *) 4) overløb markeret med bogstavet 0, ellers mellemslag.
sy	by-registeret	1) som heltal, $-512 \leq t \leq 511$ . 2) numrene på de 1-stillede bits.
cl	den celle, der fortsættes med (celle nr. [udhops-adresse]).	afhænger af <udskrift-form> .
ar arf	alle ovennævnte undtagen RF alle ovennævnte, idet R og M behandles som RF.	} kun behandlingen af R og M afhænger } af <udskrift-form> .

\*) Hver 2-bits-gruppe udskrives med eet tegn, hvor punktum står for 00

a	-	10
b	-	01
c	-	11

B) Basistrimning. Basistrimningen, som gælder for alle efterfølgende områdespecifikationer indtil en ny basistrimning angives, er en helordsordre af formen

bt<s> <antal>.39 + <outputenhed> t <basisadresse>

Operationsdelen bt står for basistrimning, og de øvrige bestanddele har følgende betydning:

Hvis <s> er tom, indledes hver linie i udskriften med ferritlageradressen i kantet parentes. Hvis <s> er s-mærkning, udelades (glettes) denne adressetrykning.

<antal> er et heltal, der angiver hvor mange tal der skal trykkes pr.linie. Udelades <antal> gælder det sidst angivne antal, og fra starten er dette lig med 3.

<antal> influerer dog ikke på ordretrykningen der altid sker med eet helord pr.linie.

<outputenhed> kan angives som 1.5 ved udskrift på skrivemaskine, 1.4 ved udskrift på perforatoren og 3.5 (eller 1.4 + 1.5) for begge dele. Udelades <outputenhed> vælges det sidst benyttede medium, og fra starten er dette skrivemaskinen.

<basisadresse> er et heltal der kun bør bruges ved tromlelagerudskrift. Det skal da angives som ferritlageradressen for den første celle på den første kanal (jfr. løbende adresse i et tromleblokhovede under programindlæsning); derefter skal adresseangivelserne i områdespecifikationer ske som ferritlageradresser, ikke som cellenumre på kanalen. Udelades <basisadresse>, sættes den lig med 0 \*) .

---

\*) se fodnoten på modstående side.

Ønskes ingen af disse specielle muligheder udnyttet, kan basis-  
trimningen udelades helt og holdent. Da gælder den indbyggede  
basistrimning

bt 3.39 + 1.5 t0

således at der altså skrives 3 tal pr.linie med adressetrykning;  
outputmediet er skrivemaskine og basisadressen er nul, hvilket  
betyder at adresseangivelserne ved tromlelagerudskrift fungerer  
som cellenumre på kanalen.

C) Numerisk trimning. En numerisk trimning gælder for alle  
efterfølgende områdespecifikationer, der indledes med grn, gmn  
eller gin, indtil en ny numerisk trimning angives. Den har form  
som to helordsordrer

nt <2-potens> t <10-potens>

qq <layout>

Operationsdelen nt står for numerisk trimning, og de øvrige  
dele har følgende betydning:

<2-potens> er et heltal som medfører, at lager- eller regi-  
sterindholdet multipliceres med den angivne 2-talspotens  
før udlæsningen. Udelades <2-potens>, svarer det til  
potensen  $2^{\uparrow 0}$ .

<10-potens> er et heltal som angiver multiplikation med en  
10-talspotens før udlæsningen. Udelades <10-potens>,  
svarer det til potensen  $10^{\uparrow 0}$ .

---

\*) Basisadressen bruges til at finde den ønskede celled placering på tromlen. Det er nemlig celle nr.<ferritadresse> ÷ <basisadresse> på den første kanal (dette nummer må gerne være >39).

Ved ferritlagerudskrift subtraheres basisadressen også fra den opgivne adresse, og en basisadresse ≠ 0 bevirker derfor en "parallelforskydning" af udskrift fra ferritlageret.

<layout> er en adresse med op til 12 led, som alle er heltal med positionsangivelser. Et layout fastlægger typografien for de enkelte tal (antal betydende cifre, decimaler, exponentcifre o.s.v.) efter nøjagtig de samme regler som gælder for udlæsning med SLIPs taltrykkesekvens, beskrevet i afsnit 12.2 ovenfor (det er den samme sekvens der bruges af "tryk").

#### Eksempel 13.9.

Efter indhop i HJÆLP ønskes udskrift af R- og M-registret som maskintal med 12 decimaler samt af s-registret.

Man kan da taste

```

h tryk
gmp , gr
gmp , gm
,gs
e

```

hvorved man på skrivemaskinen f.eks. får følgende udskrift:

```

[ R ] 0.123 456 789 101a
[ M ] -0.345 000 678 321
[ s ] 345

```

(Derefter venter GIER atter på skrivemaskine-input).

#### Eksempel 13.10.

Efter indhop i HJÆLP ønskes udskrift af alle registrene (og den ordre der fortsættes med), idet indholdet af RF er et flydende tal. Man kan da taste

```

h tryk
grf , arf
e

```

hvorved man f.eks. får følgende udskrift

```

97 ar r-5 ,gr 34
1.234677a
345 10 61 82=.bb.a . 0 -239= .1.5.9

```

Opstillingen er her følgende

<udhopsadr><celleindhold>

<RF>

<s-reg.><p-reg.><tk-reg.><indik.-reg.><overløb><by-reg.>

Således er i eksemplet  $s = 345$ ,  $p = 10$  og  $tk = 61$ . I indikatoren (pos.0-9) står bitkonfigurationen 00 01 01 00 10 svarende til det decimale tal 82, og det efterfølgende punktum viser at  $KA = KB = 0$ . Bogstavet 0 viser, at der er registreret overløb, og endelig er  $by = -239$ , dvs. positionerne 0, 1, 5 og 9 er etstillet (men pos.0 markeres aldrig i udskriften).

### Eksempel 13.11.

Ønsker man udskrift af programmet i celle 10-12, kan man (efter indhop i HJÆLP) taste

h tryk  
gpr 10 t 12  
e

hvorved man f.eks. får udskriften

```
[10] arf r+7 [17] t1
[11] mkf 94      , dkf s-3
[12] hv r+31 [43] NT
```

Ønskes udskriften på perforator og uden adresserne før hver linie, angives også en basistrimming:

h tryk  
bt s 1.4  
gpr 10 t 12  
e

Det er her væsentligt at basistrimmingen skrives før område-specifikationen, da den ellers ville være virkningsløs.

### Eksempel 13.12

Man ønsker udskrift af de flydende tal lagret på kanal 110-114 med et layout  $\{-ndd.000\}$ ; udskriften tages på perforator med 8 tal pr.linie og uden adressetrykning. Efter indhop i HJÆLP tastes

h tryk  
 bt s 8.39 + 1.4 [ingen adr.; 8 tal/linie; perforator]  
 nt [ingen skalafaktorer]  
 qq 4.3+4.7+3.13+1.9+1.14+4.23+3.27 [layout]  
 grn 110.39 + 0 t 199 [200 celler fra kanal 110 ff]  
e

hvor symbolerne +0 i den sidste parameter iøvrigt ligeså godt kunne udelades.

#### 13.4.4 "kompud"

Hjælpeprogrammet "kompud" udskriver på perforator opgivne dele af lageret i en komprimeret form, der er beregnet til hurtig indlæsning med SLIP. Den udskrevne strimmel er forsynet med den nødvendige styreinformation og kan genindlæses ved efter indhop i HJÆLP at taste l på skrivemaskinen. Hvad der sker som afslutning på denne genindlæsning afhænger af nogle af de parametre man giver "kompud" (se nedenfor under punkt B).

"kompud" skal have en eller flere heltalsgrupper som parametre til at specificere de ønskede områder i lageret, til at angive en speciel behandling af de oversprungne dele af lageret, samt til at specificere den ønskede afslutning ved genindlæsning. Dette besørges af to typer af parametre:

A) Områdespecifikation. Dette er en heltalsgruppe af formen

<beg.kanal>/<beg.celle>/<slutkanal>/<slutcelle><mærke>

hvor <mærke> enten er et a, et b eller ingenting, mens de øvrige størrelser er heltal. Denne parameter bevirker udskrift af et lagerområde efter følgende regler:

<beg.kanal> er nummeret på den første kanal; er <beg.kanal> udeladt (eller sat lig 0), betyder det udskrift ferrit-lageret.



<beg.celle> er nummeret på den første udskrevne celle, målt ud fra celle 0 på begyndelseskanalen; hvis <beg.kanal> = 0, er <beg.celle> adressen på den første udskrevne celle i ferritlageret.

<slutkanal> er nummeret på den sidste kanal; er <slutkanal> udeladt (eller sat lig 0), betyder det at sidste kanal er den samme som første kanal. Hvis <beg.kanal> = 0, skal også <slutkanal> = 0, og udskriften sker da fra ferritlageret.

<slutcelle> er nummeret på den sidste udskrevne celle, målt ud fra celle 0 på slutkanalen; ved ferritlagerudskrift er <slutcelle> ferritadressen på den sidste celle.

Størrelserne <beg.kanal> og <slutkanal> skal ligge i intervallet  $1 \leq k \leq 319$  (og kanalnumre  $\geq 294$  henviser naturligvis til ferritlagerbilledet). Ordene første kanal og sidste kanal ovenfor skal forstås i den vide forstand, at hvis numrene <beg.celle> eller <slutcelle> er  $>39$ , tæller hjælpeprogrammet selv frem til den ønskede celle på en efterfølgende kanal. Men cellenumrene bør ligge i intervallet  $0 \leq c \leq 1023$  (værdierne er egentlig arbitrære, men de opfattes modulo 1024, jfr. indlæsning af heltalsgrupper, side 74).

Hvis der inden for det opgivne område er celler med indholdet HSF 2 (jfr. hjælpeprogrammet "start", der fylder hele lageret med HSF 2), overspringes disse under udskriften; gennem mærkningen af heltalsgruppen med områdespecifikation har man mulighed for at få disse celler behandlet på to forskellige måder under genindlæsningen:

<mærke> = tom bevirker at SLIP fylder op med HSF 2 i alle celler inden for området, der blev oversprunget under udskriften. Det område, som blev specificeret under udskriften, bliver således fuldstændig retableret.

<mærke> = a bevirker at SLIP ved indlæsningen overspringer de celler (inden for området) der blev oversprunget under udskriften.

Endelig bevirker den sidste mulighed

<mærke> = b at det opgivne område slet ikke udskrives, men at SLIP ved genindlæsningen fylder hele området op med ordren HSF 2 .

Man kan ofte ønske at specificere et område ved hjælp af symbolske adresser, men dette er ikke muligt i en heltalsgruppe; derfor er det undertiden nyttigt at opgive parameteren på ordreform med 4 positionsangivelser i adressen:

qq <beg.kanal>.9+<beg.celle>.19+<slutkanal>.29+<slutcelle>.39

Man skal da blot huske at mærkningen skal angives som halvordsmærkning eller F-mærkning.

Der kan specificeres et vilkårligt antal områder ved hjælp af hver sin heltalsgruppe (eller ordre), ligesom områdespecifikation kan helt udelades, hvis en af de specielle muligheder nedenfor benyttes.

B) Udhopsspecifikation. Specielle områder. Denne parameter har form som en c-mærket heltalsgruppe

<u-adr.>/<e-adr.>/<udhop>/<D> c

Bemærk at der højst må være een parameter af denne type, og at den altid skal c-mærkes.

De to første størrelser kan være vilkårlige heltal, <udhop> er et højst 3-cifret tal med cifrene 0 eller 1, og <D> er enten 0, 1 eller 2. Virkningen af disse parametre er iøvrigt:

Indlæsningen af en komprimeret strimmel afsluttes altid med et indhop i SLIP, der fortsætter med normal strimmelindlæsning (angående værdien af i og k se nedenfor), og cifrene i <udhop> bevirker at den komprimerede strimmel til sidst forsynes med en eller flere af følgende normale SLIP-styrelinier:

<udhop> = 100: Den komprimerede strimmel afsluttes med styrelinien u<u-adr.> , som altså sætter udhopsadressen fra HJÆLP lig med <u-adr.> (se kapitel 11). Iøvrigt fortsætter strimmelindlæsningen.

<udhop> = 10 : Den komprimerede strimmel afsluttes med g , således at SLIP venter på skrivemaskineinput efter indlæsningen.

<udhop> = 1 : Den komprimerede strimmel afsluttes med styrelinien e<e-adr.> , således at GIER efter indlæsningen straks hopper til celle nr.<e-adr.>.

Vælges <udhop> lig med 11, 101, 110 eller 111, kombineres de ovennævnte tre effekter, og på den komprimerede strimmel anbringes styrelinierne i samme orden som cifrene, altså først u<u-adr.> , så g og tilsidst e<e-adr.>.

Vælges <udhop> = 0 eller 100, kan den komprimerede strimmel ikke bruges direkte til genindlæsning, da denne ikke bliver afsluttet fornuftigt (og strimmellæseren vil "hyle" efter indlæs-

ningen). Hvis den komprimerede strimmel ikke suppleres manuelt med et eller andet, må man derfor vælge en af de andre muligheder.

Den sidste størrelse, <D> , bruges til at specificere udskrift af hele ferritlageret eller hele tromlelageret:

<D> = 0 er uden virkning og forudsætter en områdespecifikation som under punkt A.

<D> = 1 bevirker, at alle registre og hele ferritlageret udskrives; de celler med HSF 2 , der overspringes under ud-læsningen, vil blive fyldt op med HSF 2 ved genindlæsning.

<D> = 2 bevirker, at alle registre, hele den ledige del af tromlelageret, d.v.s. kanal 58-293, samt ferritlageret udskrives; HSF 2-celler får samme behandling som ovenfor.

Idet ferritlagerbilledet fylder til celle 23 på kanal 319 og registrenes indhold er lagret på denne kanals celle 24-33, svarer

<D> = 1 altså til områdespecifikationen 294//319/33, og

<D> = 2 svarer til specifikationen 58//319/33.

#### Eksempel 13.13.

Ønsker man komprimeret udskrift af hele ferritlageret sådan at GIER efter genindlæsningen straks hopper til celle 10, kan man taste

h kompu  
10/1/1 c  
e

#### Eksempel 13.14.

Ønsker man komprimeret udskrift af celle 10-94, celle 300-451 samt kanal 100-114, således at GIER efter indlæsning

af den komprimerede strimmel skal vente, klar til at hoppe til celle 305, kan man taste

```

h kompud
10//94
300//451
100//114/39
305//110/c
e

```

Når den derved producerede strimmel senere indlæses (ved blot at taste l ), sættes udhopsadressen lig 305, og GIER slutter med at vente på skrivemaskine-input. Herved kan man f.eks. få tid til at klargøre en datastrimmel eller lignende, før man ved at taste e starter programudførelsen.

Rækkefølgen af parametrene til "kompud" er ligegyldig, og i ovenstående eksempel kunne vi lige så godt have tastet f.eks.

```

h kompud
300//451
305//110/c
100//114/39
10//94
e

```

Løbende adresse og kanalnummer efter indlæsning af en komprimeret strimmel. Lad  $i_0$  og  $k_0$  være værdierne af løbende adresse og kanalnummer lige før indlæsningen af en komprimeret strimmel, og lad  $i_1$  og  $k_1$  være værdierne for cellen efter den sidste celle på den komprimerede strimmel. For værdierne af  $i$  og  $k$  efter indlæsningen gælder da reglen: Hvis  $i_1 + 40 k_1 \geq 40 k_0$ , har  $i$  og  $k$  værdierne  $i_1$  og  $k_1$ ; ellers stilles de "tilbage" til værdierne  $i_0$  og  $k_0$  gældende før indlæsningen.

Anderledes sagt bevirker denne regel, at hvis indlæsningen slutter inden for den tromleblok, der var oprettet før indlæsningen, kan man uden videre indlæse mere i forlængelse af den

komprimerede strimmel. Men hvis indlæsningen slutter uden for denne tromleblok (og det vil sige før), stilles  $i$  og  $k$  tilbage til startværdierne for denne tromleblok. Herunder skal ferritlageret opfattes som en tromleblok med  $k=294$  og  $i=0$ .

Umiddelbart efter indhop i HJÆLP-SLIP oprettes en blok med  $k=294$  og  $i=0$ . Indlæsning af komprimeret udskrift fra ferritlageret i denne situation vil da altid slutte inden for denne blok, og indlæsning af andre tromlepartier end ferritlagerbilledet vil da altid slutte uden for denne blok. Derfor kan indlæsning til ferritlageret fortsætte på sædvanlig måde efter indlæsning af sådanne komprimerede strimler.

Udseende af komprimeret udskrift. Den komprimerede strimmel, som "kompud" frembringer, har i det væsentlige følgende opbygning: 1) en introduktion bestående af et startsymbol, definition af startadressen, d.v.s. løbende kanalnummer og adresse, 47 mellemslag og et slutsymbol; 2) en hoveddel bestående af et antal blokke, der hver består af et ord, der angiver hvor mange celler der skal overspringes (eller fyldes med HSF 2), samt indholdet i en række successive celler (hvor hvert ord består af 6 tegn); 3) en checksum samt antallet af tegn på den komprimerede strimmel; 4) en eventuel afslutning som forlangt gennem en udhopsspecifikation under punkt B.

Årsagen til at introduktionen på strimlen indeholder de mange mellemslag (som er blinde symboler på dette sted) er, at starter man indlæsningen med et af mellemslagene, vil indlæsningen ske til SLIPs øjeblikkelige løbende kanalnummer og adresse. Derved bliver hele indlæsningen "parallelforskudt" i lageret, fordi alle adresseangivelser på den komprimerede strimmel er

relative i forhold til startadressen. Har man f.eks. taget komprimeret udskrift af celle 100-155 og ønsker dette indlæst til celle 248-303, definerer man blot løbende adresse  $i=248$  via skrivemaskinen og starter indlæsningen af den komprimerede strimmel efter den første halve snes tegn.

#### Eksempel 13.15.

Der findes en speciel mulighed for at få udskrevet en komprimeret strimmel uden nogen introduktion: Hvis den første parameter for "kompud" er 3c , udelades startadressen (og de 47 mellemslag) på den komprimerede strimmel.

Hvis man f.eks. vil udskrive en bibliotekssekvens, der er indlæst til celle 850-889, sådan at den senere kan indlæses hvorsomhelst i lageret, taster man

```

h kompud
3c
850//889
10/c
e

```

Den frembragte strimmel er da uden introduktion og afsluttes med s , hvorfor den vil blive indlæst til den øjeblikkelige løbende adresse, og indlæsningen vil afsluttes med at GIER venter på skrivemaskine-input.

#### 13.4.5 "gem", "hent" og "sam".

##### "gem".

"gem" udfører flytning af hele ferritlagerbilledet (inklusive registerindhold) - ialt 26 kanaler - til andre tromleafsnit, og da ferritlagerbilledet er identisk med ferritlageret i det øjeblik HJÆLP blev påkaldt, sørger "gem" altså for at gemme ferritlagersituationen.

"gem" kan benyttes uden parametre eller med et vilkårligt antal parametre af formen

sk  $\langle b \rangle$  ,

hvor  $\langle b \rangle$  er et heltal i intervallet  $39 \leq b \leq 319$ . Virkningen er følgende:

uden parametre: Hele ferritlagerbilledet (kanal 294-319) overføres til kanal nr.268-293, d.v.s. de 26 sidste frie kanaler på tromlen.

sk  $\langle b \rangle$  : Ferritlagerbilledet flyttes til kanal nr. $\langle b \rangle$  og de følgende 25 kanaler.  $\langle b \rangle = 0$  (eller  $\langle b \rangle$  udeladt) bevirker, at der læses til kanal 268-293.

Ønsker man kun at gemme en del af ferritlagerbilledet eller at gemme billedet i flere dele spredt på tromlen, kan dette gøres med "hent", som beskrives nedenfor.

### "hent".

Medens "gem" altid læser fra ferritlagerbilledet til andre dele af tromlen, er "hent" beregnet til at udføre transporter den modsatte vej; imidlertid er parameterstyringen for "hent" så flexibel, at enhver transport kan udføres hermed, og hvad man måtte savne af muligheder med "gem" kan derfor udføres med "hent" ved passende parametervalg.

"hent" kan benyttes uden parametre eller med et vilkårligt antal parametre af formen

sk $\langle a \rangle$ .39 +  $\langle b \rangle$ t $\langle c \rangle$

hvor  $\langle a \rangle$  ,  $\langle b \rangle$  og  $\langle c \rangle$  er heltal i intervallerne  $0 \leq a \leq 319$ ,  $39 \leq b \leq 319$  og  $39 \leq c \leq 319$ . Virkningen er følgende:



uden parametre: Indholdet på de 26 kanaler 268-293 overføres til ferritlagerbilledet, altså præcis det omvendte af "gem" uden parametre.

sk <a>.39 + <b>t<c> : Indholdet på <a> kanaler, fra og med kanal nr.<b> overføres til lige så mange kanaler fra og med nr.<c>. Specielt har <a> = 0 samme virkning som <a> = 26 , d.v.s. at 26 kanaler flyttes; <b> = 0 har samme virkning som <b> = 268 , d.v.s. der læses fra tromleafsnittet lige før ferritlagerbilledet; <c> = 0 har samme virkning som <c> = 294 , d.v.s. der læses til ferritlagerbilledet.

Opgives der flere parametre til "hent", udføres flytningerne i den orden hvori de er forlangt gennem parametrene.

#### "sam".

"sam" sammenligner to opgivne tromleafsnit (hver på 26 kanaler) celle for celle, og når indholdene ikke er ens, udskrives disse på opgiven form.

"sam" skal have to typer af parametre, nemlig områdespecifikation på formen

sk <a>t<b>

og udskriftspecifikation på samme form som hjælpeprogrammet

"tryk". Virkningen er følgende:

#### A) Områdespecifikation.

sk <a>t<b> : "sam" sammenligner de to tromleafsnit der begynder på henholdsvis kanal nr.<a> og kanal nr.<b>. Spe-

cielt har  $\langle a \rangle = 0$  samme virkning som  $\langle a \rangle = 294$  (altså ferritlagerbilledet), mens  $\langle b \rangle = 0$  har samme virkning som  $\langle b \rangle = 268$  (altså tromleafsnittet umiddelbart før ferritlagerbilledet).

Udelades denne parameter, sammenlignes ferritlagerbilledet med tromleafsnittet umiddelbart før (d.v.s. samme virkning som med parameteren sk 0 t 0 ).

B) For hver områdespecifikation skal der være udskriftspecifikationer efter de samme regler som for "tryk", side 153, dog med følgende to undtagelser: a) Al adresseangivelse refererer til ferritlageret, hvorfor angivelse af tromleadresser er virkningsløs; d.v.s. at eventuel angivelse af basisadresse og kanalnumre overspringes. b) Hvis der opgives en basistrimning er angivelse af antal ord/linie (side 156) virkningsløs, idet udskriften altid har følgende form \*) :

	$\langle a \rangle$	$\langle b \rangle$
$\langle \text{adresse} \rangle$	$\langle \text{celleindhold} \rangle$	$\langle \text{celleindhold} \rangle$
$\langle \text{adresse} \rangle$	$\langle \text{celleindhold} \rangle$	$\langle \text{celleindhold} \rangle$
.....	.....	.....

hvor den tredje kolonnes placering bestemmes af tabulatorens indstilling, fordi "sam" bruger tabulator-ordren SY 30 mellem anden og tredje kolonne (og kun her).

Hvis man glemmer at specificere udskriftsformen, er "sam" helt virkningsløs.

Det er tilladt at specificere flere sammenligninger af par af tromleafsnit efter hinanden ved hjælp af flere parametre af

---

\*) Adressen udskrives selvom der benyttes en basistrimning med s-mærkning.

den første type sk <a>t<b> , men så skal der umiddelbart efter hver parameter af denne type følge udskriftspecifikation for den pågældende sammenligning.

Eksempel 13.16.

En meget almindelig anvendelse af de tre programmer "gem", "hent" og "sam" bliver formodentlig følgende:

1) Efter indlæsning af et program til ferritlageret (et program som ikke bruger kanal 268-293) taster:

h gem  
e

og derefter gennemføres selve den ønskede kørsel.

2) Hvad enten kørslen går godt eller skidt, kan start-situationen retableres på ethvert tidspunkt ved at trykke på hjælpeknappen og taste:

h hent  
e

3) Hvis man efter en sådan kørsel ønsker at se hvilke af ordrerne i celle 10-150 og 600-1022 der er ændret, samt hvilke af arbejdscellerne 151-559 der er ændret, kan man efter kørslen trykke på hjælpeknappen og taste:

h sam  
bt 1.4  
gp 10 t 150  
gp 600 t 1022  
gr 151 t 559  
e

Da parameteren sk mangler, sammenlignes ferritlagerbilledet (d.v.s. situationen efter kørslen) med tromleafsnittet 268-293 (d.v.s. situationen før kørslen); basistrimningen medfører udskrift på perforator, og de sidste parametre bevirker at i områderne 10-150 og 600-1022 udskrives afvigelserne på ordreform, mens afvigelserne i celle 151-599 udskrives som flydende tal med 5 betydende cifre.

13.4.6 "ret".

Ethvert normalt indhop i SLIP-HJÆLP kan man direkte indtaste hvadsomhelst i en hvilken som helst lagercelle ved først at definere *k* og *i* passende og derefter taste det ønskede indhold. Hvis man derimod ønsker at andre et af registrenes indhold, må dette ske ved hjælp af "ret", der som styringsinformation først skal have navnet på det ønskede register og derefter det indhold der ønskes indsat. "ret" kaldes med to parametre med formen

```
<registernavn><overløb>
<SLIP-information>
```

hvor <registernavn> ligesom for hjælpeprogrammet "tryk" kan være

```
gr for R-registeret,   grf for RF-registeret,
gm - M-registeret,   gs - s-registeret,
gp - p-registeret,   gk - kanal-registeret,
gi - indikator-registeret og
sy - by-registeret.
```

<overløb> må kun bruges ved indlæsning til R-registeret, og den styrer overløbssituationen dels i R, dels i Overløbsregisteret:

<overløb> = 0 eller intet: Efter indlæsning af det angivne tal (eller hvad det nu er), sættes  $R00 = Rpos[0]$  og overløbsregisteret nulstilles.

<overløb> = 1: Efter indlæsning af det nye R-indhold sættes  $R00 = \neg Rpos[0]$  og overløbsregisteret nulstilles.

<overløb> = 10: Efter indlæsning sættes  $R00 = Rpos[0]$  og 0-registeret etstilles.

<overløb> = 11: Efter indlæsning sættes  $R00 = \neg Rpos[0]$  og 0-registeret etstilles.

De normalt anvendelige muligheder er altså gr ved indlæsning til R uden overløb, og gr11 ved indlæsning til R af et eller andet med overløb. <SLIP-information> kan være en ordrelinie eller en tallinie eventuelt med en foranstillet taladministrator (f eller m):

Til R- og M-registeret kan indlæses en ordrelinie eller et tal, enten maskintal, heltalsgruppe eller flydende tal (som lagres sammenpakket).

Til RF-registeret kan kun indlæses flydende tal, som lagres på den normale oppakkede form.

Til de øvrige registre kan heltal (modulo 1024) indlæses enten som heltal (med administrator m) eller som adressetal i en ordrelinie af form qq <defineret adresse>.

Konventionerne for denne SLIP-information er iøvrigt helt som beskrevet i kapitel 11.

Det skal understreges, at man i hvert kald af "ret" kun kan indsætte eet tal.

### Eksempel 13.17.

I et program hoppes der til SLIP med ordren HS 2 , og der skal i RF-registeret sættes tallet  $5_{10}^{-20}$  , hvorefter kørslen skal fortsættes: Når GIER kommer til dette sted i programmet, taster man på skrivemaskinen

<u>h</u>	ret	
<u>g</u>	rf	[sæt i RF-registeret]
<u>f</u>		[det flydende tal]
<u>5</u>	$_{10}^{-20}$	[ $5_{10}^{-20}$ ]
<u>e</u>		
<u>e</u>		[fortsæt i programmet]

Eksempel 13.18.

I indikatoren skal TA og PA etstilles og resten nulstilles. Samtidig skal kanalregisteret sættes lig 140. Man kan da taste:

```

h ret
gi
qq 1.2 + 1.4 [etstil pos.2 og 4 i indikatoren]
e
h ret
gk
140 [sæt heltallet 140 i tk-registeret]
e

```

Administratoren m foran tallet 140 er overflødig, fordi ved indhop i HJÆLP sættes SLIP altid i m-situationen.

13.4.7 "hp ind".

Dette hjælpeprogram indsætter "parenteser" bestående af hjælpeprogrammer i kørende programmer, d.v.s. "hp ind" kan sørge for, at et andet hjælpeprogram (f.eks. "tryk") udføres, idet en bestemt ordre i et kørende program gennemløbes. "hp ind" sørger for at den pågældende ordre (celle) plukkes ud af programmet og erstattes af et hop til HJÆLP (HSF 1), som først udfører det valgte hjælpeprogram, derefter udfører den udplukkede ordre (ordrer) og endelig fortsætter korrekt i det kørende program.

Man kan indsætte flere "hp ind"-parenteser i samme program, dog maksimalt 32 på een gang.

Det valgte hjælpeprogram udføres hver gang GIER kommer til den pågældende celle, men man kan dog angive at det ikke skal ske de første  $n$  gange (for vilkårligt  $n \geq 0$ ). En indsats parentes kan fjernes igen ved hjælp af programmet "hp ud" (se næste afsnit).

I øvrigt skal "hp ind" have 2 typer af styringsinformation på følgende form:

A) Tromlelagringsinformation (som kan udelades) på formen  
 qq <beg.celle>.29 + <ledig kanal>t<basisadr>.29 + <kanalnr>

Her vedrører de to første parametre placeringen på tromlen af den nødvendige information til parentes-indsættelsen, mens de to sidste parametre kun bruges, hvis den udplukkede ordre (i det kørende program) ligger i en tromleblok. Hele tromlelagringsinformation kan udelades, hvis parentesens skal indsættes i en ferritblok samt hvis den nødvendige information må lagres i kanalerne 58 ff eller i fortsættelse af tidligere parentesinformationer (fra forudgående brug af "hp ind" med tromlelagringsinformation).

Betydningen af de 4 parametre er:

<ledig kanal> er adressen på den første af de kanaler (ofte kun een) der skal bruges til lagring af den følgende information til "hp ind". Hvis denne parameter udelades, eller sættes lig 0, lagres informationen på tromlen i forlængelse af den sidst lagrede information for et "hp ind"-kald; hvis parameteren udelades ved den første "hp ind"-anvendelse, lagres informationen automatisk på kanal 58 ff (de første frie kanaler). \*)

---

\*) Den information, der skal lagres, er den nedenfor under pkt. B omtalte hp-information; antallet af beslaglagte celler på tromlen er lig antallet af linier i hp-informationen, inklusive linien med hp-navnet. (Se eksempel 13.18 nedenfor).

- <beg.celle> er nummeret på den første celle på den ovenfor valgte kanal, der skal bruges til lagring af den følgende information. Tilsammen angiver disse to parametre altså den celle på tromlen, hvor lagringen starter. Vælges <beg.celle> = 0 eller udelades den helt, begynder lagringen forrest på den valgte kanal. I øvrigt kan <beg.celle> gerne vælges >39, idet der da tælles frem til en celle på en af de følgende kanaler på sædvanlig vis.
- <kanalnr> angives kun hvis parenteser skal indsættes ved en ordre i en tromleblok. Da skal <kanalnr> være nummeret på den 1.kanal i denne tromleblok (d.v.s. k-værdien angivet i tromleblokhovedet). Hvis denne parameter udelades, betyder det at parenteser indsættes i ferritlageret.
- <basisadresse> angives også kun hvis det drejer sig om en parentes i en tromleblok. Da skal <basisadresse> være tromleblokkens begyndelsesadresse i ferritlageret (d.v.s. den gældende i-værdi under indlæsningen af tromleblokhovedet i programmet; sammenlign afsnittet om tromleblokke, side 88). Udelades denne parameter, sættes basisadressen = 0.

B) hp-information på formen

h <hp-navn> / <n>.29 + <udpluksadresse>

efterfulgt af den nødvendige information til det valgte hjælpeprogram på sædvanlig måde.

Ved hjælp af denne type information angiver man hvad der



skal indsættes som parentes, hvor det skal indsættes og hvornår det skal træde i funktion:

<hp-navn> er navnet på det valgte hjælpeprogram (som skal være et af standardprogrammerne eller være indlæst inden parentesens aktivering).

<udpluksadresse> er ferritlageradressen på den ordre i det kørende program, der skal udplukkes og erstattes af parentesens (også hvis parentesens skal indsættes i en tromleblok, skal man her angive ferritlageradressen for den pågældende ordre). Der udplukkes altid en helcelle og den ordre eller de to ordrer, der oprindeligt stod i cellen, udføres efter at det valgte hjælpeprogram er udført.

<n> er det antal gange som celle nr.<udpluksadresse> skal gennemløbes uden at parentesens udføres, men derefter udføres parentesens hver gang celle nr.<udpluksadresse> passeres. Hvis <n> udelades eller sættes = 0, udføres parentesens allerede første gang (og alle følgende gange), celle nr.<udpluksadresse> passeres.

Man kan indsætte parenteser næsten hvorsomhelst i det kørende program, når blot man overholder følgende begrænsninger:

1) I hvert "hp ind"-kald kan man kun indsætte et hjælpeprogram et sted i det kørende program. Til gengæld kan man ved at kalde "hp ind" flere gange med samme udpluksadresse alligevel få udført flere funktioner på et sted.

2) Der kan maksimalt oprettes 32 parenteser på en gang (idet HJÆLPs katalog over udpluksadresser har 32 pladser); hvis man i

løbet af et program fjerner parenteser (med hjælpeprogrammet "hp ud") ryddes også pladser i kataloget, hvorved man kan operere med mere end 32 parenteser, blot ikke på en gang.

3) Da den udplukkede celle fjernes fra sin oprindelige plads i programmet og lagres og udføres et andet sted i ferritlageret (nemlig i celle 7, hvor den udføres umiddelbart før returhoppet fra celle 9), er visse ordrer og ordretyper forbudt i den udplukkede celle. Denne må ikke indeholde

a) Ordrer, der modificeres af andre ordrer i programmet (thi dette vil gå hårdt ud over den indsatte hopordre).

b) En ordre med tælledel, der tæller i sin egen adressedel (thi tællingen i adressedelen vil ikke blive husket til næste omgang).

c) En HS-ordre, hvor s-registeret bruges til noget som helst andet end til returhoppet på formen HR s+1, HH s eller lignende (thi da HS-ordren sættes i celle 7, bliver indholdet i s-registeret altså 7). Hvis der er anbragt en parentes ved et HS-hop, må man ikke kalde HJÆLP fra undersekvensen.

d) En ordre med V-mærkning.

e) En betingende ordre (BS, BT, CA, NC, CM) som betinger udførelsen af en ordre i næste celle (thi det vil gå ud over celle 8). Derimod kan cellen gerne indeholde en betingende ordre som venstre halvordsordre, da det under alle omstændigheder kun vedrører det tilhørende højre halvord, som plukkes med ned i celle 7.

f) Ordre der er del af en satellitkæde (IS, IT, NS, NT eller ordre umiddelbart efter sådanne), fordi virkningen af en satellitkæde afhænger af at ordrene udføres umiddelbart efter hinanden. (Det kan dog gå godt, hvis venstre halvord er en satellit-

ordre og højre halvord en skikkelig ordre).

Derimod er adressedele med parentes- og r-, s- eller p-mærkning tilladt og vil blive behandlet korrekt.

4) Man må gerne sætte parenteser i forskellige tromleblokke på een gang, men de tilhørende ferritlager-adresser skal være forskellige.

#### Eksempel 13.19.

I et program i ferritlageret ønskes udskrift af celle 400 og R-registeret ved passage af ordren i celle 157. Ved passage af ordren i celle 219 ønskes sammenligning af hele ferritlageret med det oprindeligt indlæste samt udskrift af celle 500-509.

Efter indlæsning af programmet kan man taste følgende:

<u>h</u> gem	
<u>e</u>	[hvorved det indlæste program gemmes til senere sammenligning]
<u>h</u> hp ind	
<u>h</u> tryk / 157	[indsæt tryk-parentes i celle 157]
gm 400	[tryk celle 400 som maskintal]
gm, gr	[tryk R-registeret som maskintal]
<u>e</u>	[afslut "hp ind"]
<u>h</u> hp ind	
<u>h</u> sam / 219	
gp 0 t 1023	[ændringer i ferritlageret udskrives på ordreform]
<u>e</u>	[afslut "hp ind"]
<u>h</u> hp ind	
<u>h</u> tryk / 219	
gm 500 t 509	[tryk celle 500-509]
<u>e</u>	[afslut "hp ind"]
<u>e</u>	[afslut HJÆLP og begynd på selve programmet]

Dette vil bevirke, at cellerne 157 og 219 plukkes ud af programmet og erstattes af ordren HSF 1, samt at alle de tastede oplysninger om de ønskede hjælpeprogrammer lagres på kanal 58 (hvor de iøvrigt vil fylde 10 celler). Under kørslen af programmet vil parenteserne blive udført ved hver passage af cellerne 157 og 219.

I kataloget over parenteser optages 3 pladser, idet celle 219 optræder 2 gange.

Eksempel 13.20.

Under kørsel med et program ønsker man udskrift af RF-registeret ved passage af en ordre i celle 74 som hører til en tromleblok, der er indlæst med blokhovedet  $b k=125, i=50$ . Kontroludskriften skal først begynde ved den 16. passage af ordren.

Efter indlæsning af programmet kan man taste følgende:

<u>h</u> hp ind	
qq t 50.29 + 125	[information om tromleblokken]
<u>h</u> tryk/15.29 + 74	[15. passage af celle 74]
gr, grf	[tryk RF som flydende tal]
<u>e</u>	[afslut "hp ind"]
<u>e</u>	[afslut HJÆLP og påbegynd programmet]

Dette bevirker at den 24. celle i tromleblokken udplukkes, og at RF-registerets indhold udlæses fra og med den 15. gang denne celle gennemløbes. Derimod kan celle 74 i ferritlageret udmærket være passeret mange flere gange, hvis den ligger i et område der også bruges til andre programblokke.

I ovenstående eksempel bruges de 4 første celler på kanal 58 til lagring af den nødvendige information om hp-parenthesen. Hvis denne plads ikke er ledig, men f.eks. de sidste 10 celler på kanal 153 er ledige, kan man i stedet for ovenstående taste

<u>h</u> hp ind	
qq 30.29 + 153 t 50.29 + 125	[lagring på celle 30, kanal 153]
<u>h</u> tryk/15.29 + 74	
gr, grf	
<u>e</u>	
<u>e</u>	

Hvis man senere under samme kørsel - d.v.s. uden at hjælpeprogrammet "start" har været anvendt - påkalder "hp ind" uden at angive <ledig kanal> og <beg.celle>, vil parentesinformationen blive lagret på kanal 153 fra celle 34 og fremad i forlængelse af ovenstående.

### 13.4.8 "hp ud".

Dette hjælpeprogram sletter parenteser som er indsat af "hp ind", idet den oprindelige ordre (eller ordrer) sættes på sin rigtige plads, og den tilhørende henvisning slettes i kataloget over parenteser. Som parameter skal bl.a. angives en udpluksadresse, og "hp ud" fjerner så alle henvisninger til denne adresse fra kataloget.

I øvrigt skal "hp ud" have een parameter på formen

```
qq <basisadresse>.39 + <udpluksadresse>t<kanalnr>
```

hvor

<udpluksadresse> er ferritlageradressen på den celle hvorfra parenteser skal fjernes.

<kanalnr> og <basisadresse> skal kun angives hvis den pågældende celle hører hjemme i en tromleblok, og i så fald på samme måde som for "hp ind", nemlig som k- og i-værdierne fra indlæsningen af tromleblokken.

Det må understreges at har man indsat flere parenteser på samme udpluksadresse (som i eksempel 13.19), fjernes de alle ved een anvendelse af "hp ud".

"hp ud" kan for det første indkobles manuelt under kørslen, når man synes at en hp-parentes har fungeret længe nok; men hvis man på forhånd kan planlægge hvornår parenteser skal fjernes, kan "hp ud" indkobles efter et vist antal gennemløb ved hjælp af en tælling styret af "hp ind" (eksempel 13.21).

Eksempel 13.21. Tælling med "hp ind".

Ønsker man en kontroludskrift af alle registrene ved passage af ordren i celle 843 (der hører til en skikkelig ferritlagerblok), men kun 2. til 10.gang, celle 843 gennemløbes, kan man, efter indlæsning af programmet, taste

```

h hp ind
h tryk / 1.29 + 843    [tryk fra og med 2.passage af
                           celle 843]
gm, ar                 [tryk alle registres indhold, M og
                           R som maskintal]

e
h hp ind
h hp ud / 10.29 + 843 [ved 11.passage af celle 843]
qq 843                [fjernes parenteser i celle 843,
                           også hp ud-parentesen]

e

```

Eksempel 13.22. Programstyring.

Uden at det skal opfattes som en typisk anvendelse af "hp ud" viser vi her et eksempel på en programstyret udkobling:

Lad der i begyndelsen af et program være indsat hp-parenteser i cellerne 314 og 471 (i ferritlagerblokke). Man kan da på et passende sted i programmet fjerne dem igen ved at kode

```

-- --
[m ] HS 1    [hop til HJÆLP]
[m+1] h hp ud / 1 [udfør "hp ud" med 1 cellefuld para-
                           metre]

[m+2] qq 314 [fjern parentes fra celle 314]
[m+3] HS 1   [nyt hop til HJÆLP]
[m+4] h hp ud / 1
[m+5] qq 471 [fjern parentes fra celle 471]
-- --

```

13.4.9 "slip".

Man kan undertiden efter et vist antal gennemløb af en programdel ønske at hoppe til SLIP-HJÆLP for at foretage indgreb af en eller anden art i programmet. Dette er umuligt med de hidtil

indførte mekanismer, thi man kan nok hoppe ind i SLIP-HJÆLP fra programmet (med HSF 2 eller HS 2 ), men det sker allerede første gang en sådan ordre mødes, og man kan også med "hp ind" få udført på forhånd valgte hjælpeprogrammer efter et vist antal gennemløb, men hjælpeprogrammerne skal vælges og specificeres på forhånd.

Derfor er der lavet et meget primitivt lille hjælpeprogram "slip", der blot udfører et opgivet valg af indlæse-enhed og derefter hopper ind i SLIP (som med HS 2 ); programmet er hovedsagelig tænkt anvendt som underprogram under "hp ind". Der kræves en parameter på formen

qq <ydre enhed>

hvor <ydre enhed> skal være = 17, hvis man ønsker indlæsning fra skrivemaskinen, og ellers <ydre enhed> = 16, som giver indlæsning fra strimmellæseren. Udlæseenheden er som sædvanlig skrivemaskinen.

### Eksempel 13.23.

Efter de 100 første gennemløb af celle 127 ønsker man at få mulighed for at gribe ind i udviklingen uden på forhånd at vide hvad man vil gøre. Man indsætter derfor en parentes med hop til SLIP-HJÆLP der først aktiviseres ved passage nr. 100: Efter at programmet er indlæst tastes

```

h hp ind
h slip / 99.29 + 127 [ved 100.passage hop til SLIP]
qq 17 [med skrivemaskineindlæsning]
e

```

13.4.10 Indlæsning af andre hjælpeprogrammer.

Som omtalt i afsnit 13.4.1 side 146 er det meget nemt at ændre placeringen og antallet af hjælpeprogrammer, så længe blot kanal 0-38 lades urørte. Desuden skal hjælpeprogrammet "læs hp", der normalt er lagret på kanal 57, være intakt. "læs hp" kan ikke kaldes som de andre hjælpeprogrammer, men træder i funktion når man indlæser en hjælpeprogramstrimmel ved at taste

c<beg.kanal>

l

Samtidig med indlæsningen af et nyt hjælpeprogram indføres dets navn og placering i kataloget over hjælpeprogram; hvis det har samme navn som et tidligere indlæst hjælpeprogram, slettes det gamle af kataloget, jfr. bemærkningerne i afsnit 13.4.1, side 148.

Under indlæsning af nye hjælpeprogrammer må kanal 57 (eller rettere den kanal hvor "læs hp" er lagret) ikke røres, men så snart selve indlæsningen er overstået, kan denne kanal bruges til hvad som helst.

Eksempel 13.24.

Hvis man partout vil indlæse et hjælpeprogram, der fylder 4 kanaler, til kanal 54-57 kan dette ikke gøres uden først at flytte "læs hp". Man må derfor først indlæse hjælpeprogrammet "læs hp" til en eller anden ubenyttet kanal, f.eks. kanal 117, ved at taste

c 117  
l

Derefter kan man lægge strimlen med det ønskede hjælpeprogram klar og taste

c 54  
l



Det skal bemærkes, at strimlerne "HJÆLP-uaflåsedede" og "HJÆLP-uaflåsedede uden om ALGOL", som indeholder alle standard-hjælpeprogrammerne, indlæses ved blot at taste 1. Strimlen "HJÆLP", som indeholder hele systemet på nær kanal 0, indlæses ved blot at taste et mellemslag (efter at kanal 1-31 er låset op).

#### 13.4.11 Konventioner for kodning af hjælpeprogrammer.

Hjælpeprogrammer, der skal indgå i HJÆLP-komplekset, må kodes under hensyntagen til nogle få konventioner, der er nødvendige for at programmerne kan indkobles og udkobles via den centrale administration i HJÆLP. Disse spilleregler handler dels om hvordan programmet skal kodes og dels om den endelige strimmels udseende:

##### A) Programmet.

1) Længde og placering: Programmet må højst fylde 12 kanaler og er under udførelsen placeret i ferritlageret fra celle 0 og fremad. Programmets to første celler skal have følgende indhold:

[celle 0] en QQ-ordre med adresse 55 (antal kanaler -1) \*)

[celle 1] "komplementær checksum"

Ved "komplementær checksum" forstås en sådan bitkonfiguration, at summen (dannet med AR-ordrer) af alle celler i programmet (et helt antal kanaler) er lig med 0.

---

\*) Indholdet i højre halvord af celle 0 er ligegyldigt.

2) Indhop og udhop: Indhoppet til hjælpeprogrammet sker fra HJÆLPs centrale administration med en HV-ordre; indhopadressen kan vælges vilkårligt og angives på strimmelen med hjælpeprogrammet (se nedenfor). Det normale udhop fra hjælpeprogrammet skal ske med ordren HV 694 , og registrenes indhold på dette tidspunkt er uden betydning. Hvis hjælpeprogrammet af en eller anden grund ikke kan fuldføre sin mission (måske på grund af fejl i parametrene), bør programmet levere en fejludskrift på skrivemaskinen og skal afsluttes med ordrene VY 17, HH 695 ; dette bringer HJÆLP i samme situation som efter et s .

3) Parametre: Når indhoppet til hjælpeprogrammet sker, har den centrale administration indlæst parametrene og lagret dem i celle 642, 643, ...; der er maksimalt plads til 38 parametre. I celle 641 står antallet af parametre som talletal (i cellen står i øvrigt linien h <hjulpeprogramnavn>/<antal parametre> , lagret på samme vis som nævnt i afsnit 13.3.3, side 142). Efter udhoppet fra hjælpeprogrammet er indholdet i celle 641-679 uden betydning.

### B) Strimmelen.

Den strimmel med hjælpeprogrammet, der skal indlæses som beskrevet i afsnit 13.4.10 ovenfor, skal have følgende udseende:

```

b a0
a0: b k=a0, i=0
    --- } komprimeret udskrift af hjælpeprogrammet
    --- } uden introduktion og afslutning
    e
h læs hp
h <hjulpeprogramnavn>/<indhopsadresse>.29 + a0
e
e
s

```

hvor <indhopsadresse> er den ønskede startadresse (ofte sikkert

celle 2), mens <hjælpeprogramnavn> er det ønskede navn på programmet. Navnet kan vælges helt frit, men "tværsommen" bør være forskellig fra "tværsummerne" for de eksisterende hjælpeprogrammer.

### 13.5 Fejludskrifter fra HJÆLP.

Under kørsel med HJÆLP kan man få forskellige fejludskrifter (og fejlreaktioner) udover de i afsnit 11.8 omtalte, som skyldes syntaktiske fejl og lignende under SLIP-indlæsning (se side 102). Disse fejludskrifter, der kan dukke op under brug af hjælpeprogrammerne, omtales nærmere nedenfor.

#### 13.5.1 Gal styreinformation.

Under tastning (eller strimmelindlæsning) af parametre til et hjælpeprogram kan man få fejludskriften

gal information

Det betyder at en eller flere af parametrene ikke er i overensstemmelse med kravene for det pågældende hjælpeprogram. Når det e, der afslutter parametrene, er indlæst bliver hjælpeprogrammet udført mangelfuldt (eller slet ikke), og derefter kan man vælge hjælpeprogrammet på ny og give de rigtige parametre.

Man kan også, så snart man ser fejludskriften, trykke på HP-knappen og derefter vælge hjælpeprogrammet på ny.

Hvis man under tastning af styreparametre opdager at man har tastet galt, kan man rette fejlen som ved sædvanlig SLIP-indlæsning: Afslut den igangværende linie; tast eventuelt CR og et

mellemslag, hvilket giver udskrift af løbende adresse i ; stil i tilbage til den fejlramte linie; tast den korrekte linie; stil i frem igen og fortsæt.

Man kan også trykke på HP-knappen, men så skal man begynde helt forfra med det pågældende hjælpeprogram.

### 13.5.2 "tomt hp" og "sumfejl".

Hvis man vælger et hjælpeprogram og får udskriften

tomt hp

betyder det, at der i HJÆLPs katalog ikke er opført noget hjælpeprogram med det navn (eller rettere sagt med et navn, hvis "tværsum" er den samme som det valgte navn).

sumfejl

betyder, at det pågældende hjælpeprogram står i kataloget, men at checksummen ikke stemmer, d.v.s. at der er sket ændringer på de kanaler hvor hjælpeprogrammet er lagret.

I begge tilfælde venter GIER derefter på skrivemaskine-input, og det må kraftigt tilrådes at taste

h kontrol  
e

for at få fastslået hvilke hjælpeprogrammer der er intakte, og hvor de er placeret. Derefter kan man eventuelt genindlæse de manglende hjælpeprogrammer og fortsætte. Et enkelt hjælpeprogram indlæses ved at taste

c <beg.kanal>  
l

og hele den del af HJÆLP, der ligger på uaflåede kanaler, findes både på strimmelen "HJÆLP-uaflåede" og på strimmelen "HJÆLP-uaflåede uden om ALGOL", og disse indlæses ved blot at taste l;

indlæsningen afsluttes med normalt indhop i HJÆLP, og den afbrudte kørsel kan fortsætte, da indlæsningen kun berører de af HJÆLP beslaglagte kanaler.

### 13.5.3 Fejl på de låsede kanaler.

A) Hvis man efter et tryk på HP-knappen eller ved et andet indhop i HJÆLP får udskriften

FEJL

betyder det, at der er fejl på en af de låsede kanaler 1-31.

Situationen kan reddes ved følgende manøvre: 1) Lås op for kanal 1-31 (en kontakt i GIER-skabet); 2) læg strimmelen "HJÆLP" (som indeholder hele HJÆLP-systemet på kanal 1-57) i læseren og tast et mellemslag, hvorved strimmelen indlæses; 3) lås for kanal 1-31. Ligesom ovenfor er tilstanden i resten af GIER uændret, og den afbrudte kørsel kan fortsætte.

B) Hvis et tryk på HP-knappen ikke engang giver udskriften FEJL, kan det skyldes at der er spærret for HP-knap-funktionen (pos.0 i by-registeret er etstillet). Et tryk på RESET vil frigøre HP-knappen. Hvis denne stadig ikke virker, er der maskinfejl eller fejl på kanal 0, og i alle tilfælde kan kørslen kun fortsættes med kørselslederens hjælp.

Under normal GIER-kørsel kan der ikke på nogen måde opstå fejl på kanal 0, da der er låset for skrivning på denne med en meget vanskelig tilgængelig kontakt i GIER-skabet. Hvis HP-knappen (som kun forudsætter at kanal 0 er intakt) ikke virker, er det derfor tegn på en alvorlig fejl.

## 14. BIBLIOTEKSSEKVENSER

### 14.1 Indledning.

#### 14.1.1 Formålet.

Som nævnt allerede i kapitel 1 i Lærebog I er det af stor betydning at have en stor samling lettilgængelige bibliotekssekvenser ved regnemaskinen. Dette er programmer, der udfører mere eller mindre hyppigt tilbagevendende standardopgaver; disse programmer kan enten være udformet som selvstændige biblioteksprogrammer der løser en afsluttet opgave, eller som bibliotekssekvenser der skal indgå som undersekvens for et større program og som kun udfører en (lille) delproces af den samlede beregning.

Ved at benytte de foreliggende bibliotekssekvenser og -programmer spares man for det første for kodearbejdet og den dertil hørende fejlfinding, fordi bibliotekssekvenserne er gennemprøvet; desuden må man formode at mange af bibliotekssekvenserne er næsten optimale med hensyn til kodelængde, køretid og nøjagtighed. Derfor skal man altid granske GIER-biblioteket (samlingen af bibliotekssekvenser og -programmer), før man giver sig i kast med kodningen af et større problem; thi hvis nogle delprocesser af det foreliggende problem kan klares med bibliotekssekvenser, slipper man jo nemmere fra kodningen af problemet.

### 14.1.2 Krav til bibliotekssekvenser.

Hvis en bibliotekssekvens skal være til nogen almen nytte, er det ikke nok at skrive et stykke genial kode. For det første må sekvensen afprøves meget grundigt, og bl.a. er det væsentligt at teste den logiske struktur helt tilbunds, d.v.s. undersøge om alle forgreninger i koden fungerer som tilsigtet; test-eksemplerne må altså vælges sådan at de dækker alle mulige specialtilfælde.

For det andet er selv den bedste sekvens unyttig uden en ordentlig beskrivelse, og denne bør omfatte: En kort beskrivelse af sekvensens funktion og køretid, de betingelser der skal være opfyldt ved indhop, samt tilstanden ved udhop fra sekvensen; den til grund liggende (matematiske) metode; et rutediagram eller en ALGOL-beskrivelse af maskinkoden; selve maskinkoden; en kort beskrivelse af den foretagne afprøvning og resultaterne herfra.

Kodetekniske krav: Koden skal være skrevet sådan (med flittig brug af relativ adressering) at sekvensen kan placeres hvor som helst i lageret. Hvis der bruges navne i koden skal sekvensen udgøre en blok, således at navne med kun lokal interesse virkelig bliver lokale. Indhop og udhop skal foregå med HS- og HR-ordrer, sådan at sekvensmekanismen kan fungere korrekt.

## 14.2 GIER-sekvensbeskrivelser.

I dette afsnit gennemgås de forskellige elementer i en beskrivelse af en bibliotekssekvens, og da beskrivelserne af biblioteksprogrammerne har omtrent samme struktur, gælder nedenstående

stort set også for disse. Dette afsnit er altså nærmest en vejledning i at læse en sekvensbeskrivelse, men kan måske også virke som inspiration ved udarbejdelsen af nye beskrivelser.

Beskrivelserne er som regel delt op i visse standardafsnit, nemlig: En oversigt med de væsentlige oplysninger i skemaform; en beskrivelse af den matematiske metode der ligger til grund for sekvensen; en beskrivelse af sekvensens funktion; en ALGOL-beskrivelse; maskinkoden med kommentarer; testmetoden og resultaterne heraf.

#### 14.2.1 Oversigt.

Oversigten findes altid på side 2 umiddelbart efter forsiden med indholdsfortegnelsen, og den består af to skemaer og nogle supplerende oplysninger nedenunder.

I det øverste skema findes nummereret de forskellige mulige indhop i sekvensen; for hvert af disse er angivet indhoppets form og de eventuelle parametres placering (i rubrikken location), betegnelse (i rubrikken indhold), lagringsform (i rubrikken form) samt eventuelt yderligere forklaring af parameterens betydning (i rubrikken betydning).

I det nederste skema findes de til hvert indhop svarende udhop (nummereret på samme måde) med de tilsvarende oplysninger om udhoppets form og resultaternes placering og form.

Ved adresseangivelserne i rubrikken location er sekvensen som regel tænkt lagret i cellerne  $m$ ,  $m+1$ ,  $m+2$ , ..., mens indhoppene foretages fra celle nr.  $s$ .

Selve indhoppet skal altid foregå med en HS-ordre og udhoppet fra sekvensen er næsten altid en HR-ordre med adresse  $s+1$ ,



s+2 eller lignende (afhængig af eventuelle programparametres placering).

Under disse skemaer findes en række andre oplysninger:

1) Hvilke registre der ændres af sekvensen - principielt bør en bibliotekssekvens retablere alle de benyttede registre, undtagen hvis resultatet afleveres i et register.

2) Det kodesprog, sekvensen er skrevet i (næsten i alle tilfælde kodesproget SLIP).

3) Længden af maskinkoden og eventuelt specifikation af benyttelse af arbejdsceller uden for selve sekvensen - ofte medregnes sådanne arbejdsceller i selve sekvensen, og i så fald er der ingen uden for sekvensen.

4) Opremsning af eventuelle bibliotekssekvenser, der benyttes som undersekvenser for denne, med specifikation af eventuelle krav om bestemt lagring, bestemte parameterverdier m.v. for disse undersekvenser.

Alt i alt skulle oversigten således indeholde tilstrækkelig mange oplysninger til at man kan afgøre, om man kan bruge sekvensen i en bestemt situation, og hvordan sekvensen kan indpasses i hovedprogrammet.

#### 14.2.2 Funktion og metode.

Rubrikken "Funktion" er et supplement til Oversigten, og her forklares først sekvensens funktion i ord. Indholdet i registre og parameterceller ved indhop og ved udhop præciseres, navnlig hvis der er sket ændringer; eventuelle specialtilfælde (fejludhop og lign.) omtales nærmere. Desuden angives den omtrentlige

køretid samt den maximale fejl på resultaterne for så vidt dette er muligt.

I rubrikken "Metode" forklares grundlaget for sekvensen, f.eks. den matematiske metode sekvensen arbejder efter. Forklaringen bør være tilstrækkelig grundig til at man uden større vanskelighed kan forstå sekvensens virkemåde; eventuelt kan forklaringen bestå af en litteraturhenvisning med nogle få kommentarer.

### 14.2.3 ALGOL-beskrivelse og Maskinkode.

Derefter følger en ALGOL-beskrivelse af maskinkoden. Snarere end at være et formelt korrekt ALGOL-program (eller en procedure) bør denne beskrivelse udformes i pseudo-ALGOL, thi formålet er at det skal være en læsehjælp til maskinkoden. Kodetekniske detaljer, der ikke kan udtrykkes i ALGOL (f.eks. udnyttelsen af diverse indikatorbits eller tromletransporter), kan i nogen udstrækning skrives som kommentarer.

Maskinkoden er skrevet på den form som leveres af hjælpeprogrammet "tryk", når det kaldes med en områdespecifikation af formen gpr 0 t <sidste celle> , idet sekvensen tænkes lagret fra celle 0 og fremad. Udskriften er altså på ordreform, og efter alle relativmærkede adresser er den absolutte adresse noteret i kantet parentes. Koden bør være forsynet med kommentarer, og det er af stor værdi at benytte de samme betegnelser i beskrivelsen af metoden, i ALGOL-beskrivelsen og i kommentarerne til maskinkoden.

Som supplement til maskinkoden eller som et selvstændigt afsnit under overskriften Kodetekniske oplysninger kan man finde oplysning om specielle finesser ved koden; det kan f.eks. dreje

sig om indholdet i visse arbejdsceller, der undertiden kan udnyttes af brugeren, eller måske kan en stump af sekvensen under visse omstændigheder bruges i helt andre forbindelser, eller særlige kodetricks kan nævnes her.

#### 14.2.4 Testmetode.

Sidste afsnit indeholder en kort gennemgang af den afprøvnings, sekvensen har været underkastet; eventuelt er nogle af de opnåede resultater gengivet, ligesom det bør fremgå, hvad resultaterne er sammenlignet med (f.eks. hvilket tabelværk der er konsulteret). Det er nemlig væsentligt at dokumentere på hvilken baggrund sekvensens regnenøjagtighed og køretid er anslået.

#### Eksempel 14.1.

Bibliotekssekvensen EXP-1 har 3 mulige indhop til beregning af henholdsvis  $10^x$ ,  $e^x$  og  $2^x$ . I alle 3 tilfælde tages argumentet  $x$  fra RF-registeret ved indhoppet, og resultatet afleveres atter i RF-registeret. I oversigtens to skemaer, som er gengivet nedenfor, optræder der derfor 3 indhop og de tilsvarende 3 udhop:

#### Indgang:

Nr.	Location	Indhold	Form	Betydning
1	celle[s] RF	HS m+0 x	Ordre Flyd.tal	Indhop til $10^{\uparrow x}$ Argument
2	celle[s] RF	HS m+1 x	Ordre Flyd.tal	Indhop til $e^{\uparrow x}$ Argument
3	celle[s] RF	HS m+5 x	Ordre Flyd.tal	Indhop til $2^{\uparrow x}$ Argument

Udgang:

Nr.	Location	Indhold	Form	Betydning
1	celle[m+22] RF	..., HR s+1 $10 \uparrow x$	Ordre Flyd.tal	Udhop
2	celle[m+22] RF	..., HR s+1 $e \uparrow x$	Ordre Flyd.tal	Udhop
3	celle[m+22] RF	..., HR s+1 $2 \uparrow x$	Ordre Flyd.tal	Udhop

Af det nederste skema kan man bl.a. aflæse at udhoppet i alle normale tilfælde sker fra højre halvcelle m+22. Men først af rubrikken "Metode", som ikke er aftrykt her, kan man se, at i tilfælde af at  $x$  er for stor (så exponential-funktionen sprænger talkapaciteten), foretages der et nødudhop til HJÆLP fra celle m+23.

14.3 GIER-sekvensstrimler.

Ved GIER-maskinen vil der i almindelighed ligge et sæt strimler med bibliotekssekvenser. Hver strimmel indeholder maskinkoden for een sekvens skrevet på samme form som maskinkoden i beskrivelsen (se ovenfor), dog uden kommentarer. Strimmelen slutter med et Stop Code og et s. (Symbolet Stop Code før s er anbragt for at man ved en kopiering af strimmelen på flexowriter kan undgå at kopiere s med.)

For at indlæse en bibliotekssekvens er det således blot nødvendigt at sætte løbende adresse lig med den ønskede begyndelsesværdi og derpå starte strimmelindlæsningen. Denne vil altid slutte med at GIER venter (i HJÆLP) på skrivemaskine-input.

## 15. EKSEMPEL OG ØVELSER

### 15.1 Indledning.

Da de foregående kapitler indeholder en hel del ret udførlige eksempler (bl.a. i afsnit 11.2 og i kapitel 12), vil vi her nøjes med at give et enkelt lidt større eksempel (afsnit 15.2). Derefter findes en række øvelser der stort set er ordnet efter bogens kapitler, og som sammen med eksemplerne skulle lette tilgængelsen af stoffet. Nogle af øvelserne er med vilje ikke udført helt exakt og entydigt, fordi de derved måske svarer lidt bedre til den barske virkelighed, hvor problemer sjældent er entydigt defineret.

### 15.2 Eksempel på simultan tromletransport og beregning.

På en række successive tromlekanaler findes nogle data som skal bearbejdes; vi vil udnytte muligheden for simultan beregning og tromletransport ved at indlæse fra tromlen skiftevis til 2 sektioner à 40 celler i ferritlageret (2 "bufferlagre"). Mens der indlæses til den ene sektion, skal GIER behandle materialet i den anden sektion.

Lad tromlekanalerne være nr.  $a_0, a_0+1, \dots, a_0+a_1-1$ , ialt  $a_1$  kanaler; lad de 2 sektioner i ferritlageret begynde i henholdsvis celle  $[b_0]$  og celle  $[b_1]$  og lad os bruge navnene  $c_0, c_1, \dots$  til programhenvisninger.

For hver gang en kanal er færdigbehandlet, vil vi ombytte adresserne  $b_0$  og  $b_1$  i programmet, og for at gøre det overskueligt optræder de kun i tre ordrer, nemlig en LK-ordre (læsning af næste kanal) og to AR-ordrer (repræsenterende bearbejdningen af materialet). Følgende kodelistump viser en mulig udformning af ombytningen:

```

c1: lk b1, ...      [læs en kanal til celle b1, ...]
c2: arn b0, ...
c3: ar b0 t1       [bearbejdning af celle b0 til b0+39]
   ==
   arn c1, it (c2) [den ene adresse → R, den anden som
                   talletal]
   pa c1, ga c2    [ombyt adresserne i celle c1 og c2]
   ga c3, ...

```

Under bearbejdningen må al adressering foregå med henvisning til adresserne i celle  $c_2$  og  $c_3$ . Hvis dette er ubekvemt kan mekanismen blot suppleres med flere GA-ordrer efter den sidste ordre GA  $c_3$ . Det er nødvendigt at adressen i celle  $[c_2]$  ikke modificeres under bearbejdningen.

Vi vil nu udbygge denne kode lidt så den omfatter den indledende manøvre og løkken der sørger for læsning af netop  $a_1$  kanaler:

	vk a0, lk b0	[læs 1.kanal]
	pa c0 t a0	[retabler kanaladressen i celle c0]
→ c0:	vk a0 t1	[vælg næste kanal]
	c1: lk b1, ...	[begynd læsning af næste kanal]
	c2: arn b0, ...	
→ c3:	ar b0 t1	} [bearbejdning af en kanal]
	===	
	arn c1, it (c2)	[forbered ombytning]
	pa c1, ga c2	[ombyt adresser]
	ga c3, it (c0)	
	bs a0+a1, hv c0	[hop tilbage a1-1 gange]
	---	

Bemærkninger: 1) Ved sidste gennemløb af løkken påbegyndes læsningen af den uvedkommende kanal nr. a0+a1, men det gør ikke anden skade end at denne kanaltransport måske ikke er afsluttet når GIER løber ud af løkken. 2) Skal hele ovenstående program kunne gentages, er det nødvendigt at retablere adresserne i celle c1, c2 og c3 (hvis antallet af ombytninger af adresserne er lige, behøver man kun at retablere adressen i c3).

Indlæsning: Før ovenstående kode kan indlæses med SLIP, skal den forsynes med erklæringer og definitioner. Specielt skal navnene a0 og a1 være defineret inden indlæsning af den næstsidste ordre BS a0+a1, thi en sådan adresse er kun tilladt med definerede navne (se afsnit 11.4.2, side 61); derimod kan de øvrige navne gerne defineres efter indlæsningen af selve programmet - de skal blot være erklæret inden programmet.

Vi forestiller os nu at dette program skal indlæses som en del af en større helhed. For ikke at lægge unødigt beslag på navne i resten af programmet vil vi gøre denne del til en blok. Tidligere i hovedprogrammet er de data, der nu skal bearbejdes, lagret på kanal c17, c17+1, ..., c17+15 (hvor c17 er et navn der defineres forrest i hovedprogrammet). Hovedprogrammet har ligeledes benyttet cellerne c23 til c23+79 som arbejdsceller, og

disse skal nu benyttes som bufferlagre for tromletransporten. Korrespondancen mellem hovedprogrammet og denne blok kan da ordnes med nogle definitionslinier først i blokken:

```

===
  b a1, b1, c3
  a0=c17, a1=16      [begyndelseskanal og antal kanaler]
  b0=c23, b1=c23+40 [buffersektionernes beg.adresser]

|                     |
|---------------------|
| ovenstående<br>kode |
|---------------------|

e [de erklærede navne frigøres igen]
===

```

Bemærk at erklæringen b ..., c3 kun gør navnene c0, c1, c2, c3 lokale, mens navnene c4, c5, ... kan bruges inden i blokken i deres globale betydning.

Under den første indlæsning af programmet bør enten KA eller KB være etstillet, thi da får man udskrift af alle navnenes talværdier (enten på skrivemaskinen eller på perforatoren).

### 15.3 Øvelser.

#### 15.3.1 Tromle og ydre enheder. Manøverbord.

##### Øvelse 15.1.

Et program benytter 3 undersekvenser, der er lagret på tromlen og som hver fylder 1 kanal; undersekvenserne deler en sektion på 40 celler (f.eks. celle nr. b0 til b0+39) i ferritlageret, og for alle 3 sekvenser sker indhoppet til den første celle.

Sekvenserne kaldes hyppigt af hovedprogrammet, men i meget usystematisk rækkefølge, og ofte kaldes den samme undersekvens flere gange i træk.



Indhop til disse sekvenser kunne kodes sådan, at før hvert indhop overføres den tilsvarende sekvens fra tromlen til ferrit-lageret, men dette er uøkonomisk når den samme sekvens bruges flere gange i træk.

Derfor: Skriv en stump kode med 3 forskellige indhop (et for hver sekvens), der for hvert indhop undersøger om den pågældende sekvens er den sidst benyttede, og som kun i benægtende fald henter sekvensen fra tromlen, for derefter at foretage indhop til sekvensen. Benyt f.eks. to (evt. tre) indikatorbits til at holde rede på, hvilken sekvens der sidst er benyttet.

### Øvelse 15.2.

Under indlæsning af en strimmel stopper GIER, og lampen "str.læs. par.fejl" lyser. En undersøgelse af strimmelen viser, at oven i tegnet for CR er der ved en fejltagelse blevet hullet et mellemslag som er helt overflødigt. Hvad skal man gøre for at eliminere virkningen af dette og fortsætte indlæsningen?

### Øvelse 15.3.

Antag at GIER venter på skrivemaskineinput, men at HP-knappen og HJÆLP ikke fungerer. Beskriv hvordan man ved hjælp af manøverbordet kan indsætte følgende tre ordrer i celle 1-3:

```
[1] LY 0, TL -6
[2] CA 0, HV 1
[3] GM 5 M t-1
```

og derefter starte GIER i celle 1 (det er disse ordrer, der skal bruges ved indlæsning af HJÆLPs centrale administration til kanal 0).

15.3.2 Indlæsning med SLIP.Øvelse 15.4.

Afgør, hvilke af følgende ordrelinier der vil blive accepteret af SLIP, og angiv om de benyttede navne kan være udefinerede:

nt a1-3a3+4.9-c13.8+1+c14	SR R+3 t 17a1
ar ra1+a2.19, bssa0	qD -1028 IK
mkLZmentaL	aRD r-a1 XtVn-31
HviLt	AcetOnI

Øvelse 15.5.

Angiv de ordrelinier der svarer til følgende heltalsgrupper (d.v.s. de ordrelinier der giver samme bitkonfiguration ved indlæsning med SLIP):

31/-17//	2/3/33/2 c
----------	------------

Øvelse 15.6.

Hvor mange celler fylder følgende tekst:

t Den glade enke:  
Xantippe;

Øvelse 15.7.

Hvad er værdien af navnet a3 efter indlæsning af følgende "kode":

```

b a7
i=12, a3=5i
  b a3, b3
  a3=512
e

```

Hvad er værdien af i ? Fælde: Hvad er værdien af a3 efter indlæsning af endnu et e ?

Øvelse 15.8.

Find adresse- og talletallene for alle ordrene, når nedenstående program er indlæst:

```

i=100
b a17
a2 = i+1
    AR r+1, SR i+1
    AN a2+1, SN a2+i+1
a2 = a2-i
a3: AC a2.7, SC 26a3.8
e
s

```

Øvelse 15.9.

Hvilken kontroludskrift fås ved indlæsning af programmet i foregående øvelse, når KA = 1 under indlæsningen.

Øvelse 15.10.

Vil følgende program blive accepteret af SLIP:

```

b b3
b = i+3
b1: mk b t1
b2: mkD b1 tb2
b3: mknf b1-b2, m1 r2b.7
b: mt (rb2+1)
e
s

```

Øvelse 15.11.

I et større program skal en tromleblok på 182 celler lagres på kanal 78-82, og under kørslen skal den placeres (af programmet) i celle 520 og fremad; i blokken bruges de lokale navne

$c_0, c_1, \dots, c_{13}$ . Hvordan skal blokhovedet for denne tromleblok se ud? Hvilke værdier har løbende adresse og løbende kanalnummer umiddelbart før og umiddelbart efter indlæsningen af det e der afslutter tromleblokken?

### Øvelse 15.12.

Efter kodningen af et større program med 4 sektioner erkendes det, at de to midterste sektioner, som hver fylder 100 celler, må lagres på tromlen og må dele plads i ferritlageret, mens programmets 4.sektion skal lagres i ferritlageret umiddelbart efter den plads der bruges til 2. og 3.sektion (se figuren).

Angiv de tromleordrer m.v. der må indføjenes i programmet, samt de tromleblokhoveder og -afslutninger der må tilføjes, idet kanal 70-75 kan benyttes til lagring af sektion 2 og 3.

1.sektion
2.og 3.sektion skiftevis
4.sektion

Hvad skal man undersøge vedrørende brugen af navne i 2. og 3.sektion fordi disse nu bliver tromleblokke?

### Øvelse 15.13.

Under SLIP-indlæsning af et program (hvor  $KA = KB = 0$ ) udskriver GIER med sort på skrivemaskinen:

```
b 1 137 0
d 4 155 208
```

hvorefter indlæsningen fortsætter (for at slutte med det obligatoriske s ).

Hvad betyder denne fejludskrift? Hvad kan man gøre for at redde situationen?

15.3.3 Udlæsning.Øvelse 15.14.

Der foreligger en bibliotekssekvens, der kan beregne sinus og cosinus; med maskintallet  $x$  i R-registeret bevirker et hop til sekvensens celle 0 at  $\cos(2\pi x)$  udregnes, og et hop til celle 1 at  $\sin(2\pi x)$  udregnes; i begge tilfælde afleveres resultatet som maskintal i R-registeret. Sekvensen fylder højst 31 celler.

Skriv et program, der trykker en tabel med 4 søjler, nemlig  $v$ ,  $\cos v$ ,  $\sin v$  og afvigelsen  $1 - (\cos^2 v + \sin^2 v)$  for  $v = 0^\circ$ ,  $0.01^\circ$ ,  $0.02^\circ$ , ...,  $0.2^\circ$  og for  $v = 44.5^\circ$ ,  $44.55^\circ$ , ...,  $45.5^\circ$ .

De 4 størrelser skal trykkes med følgende layouts:

$v$	$\cos v, \sin v$	$1 - (\cos^2 v + \sin^2 v)$
{nd.dd}	{-n.ddddd}	{-n.d <sub>0</sub> -dd}

Forsyn programmet med styrelinier m.v., så det er klar til indlæsning.

Øvelse 15.15.

Lav lidt om på ovenstående program således at de benyttede  $v$ -værdier kan indtastes: Programmet skal begynde med at indlæse 3 tal, nemlig startværdien, skridtlængden og slutværdien for  $v$ ; så skal tabellen beregnes og trykkes, og derefter skal programmet være klar til at begynde forfra. Benyt indlæsning med SLIP sådan at man blot skal taste de 3 tal med komma imellem og afsluttet med e.

#### 15.3.4 Hjælpeprogrammer.

##### Øvelse 15.16.

Efter en kørsel ønskes udskrift af heltallene i celle 20-21 på kanal 94. Hvad skal man taste?

##### Øvelse 15.17.

Efter en kørsel skal de flydende tal på kanal 100-139 sammenlignes med tallene på kanal 150-189, og alle afvigelser skal udskrives med 10 betydende cifre. Hvad skal man taste?

##### Øvelse 15.18.

GIER venter på skrivemaskine-input (den grønne lampe på skrivemaskinen lyser). Men så snart der er skrevet eet tegn, skriver GIER "hp-knap 47" (med rødt). Hvad er den sandsynligste forklaring herpå?

##### Øvelse 15.19.

Et program er lagret i celle 10-259 og benytter data lagret i celle 300-600.

Midt under kørslen skriver GIER på skrivemaskinen

HSF 2 371 (med rødt)

Hvad tyder dette på, og hvad bør man foretage sig?

##### Øvelse 15.20.

Efter kørsel med et program ønskes udskrift af celle 900-939 og kanal 70-74 som flydende tal med fuld præcision (9-10 betydende cifre). Hvad skal man taste for at få "tryk" til at udføre dette?

Øvelse 15.21.

Under en kørsel ønsker man at indkoble overvågningsprogrammet "hop", men når man efter et tryk på HP-knappen taster h hop med passende parametre, skriver GIER "tomt hp" (med rødt).

Hvad betyder dette, og hvad bør man gøre?

Øvelse 15.22.

Et program består af 4 dele, der skal indlæses og lagres som følger:

- 1) En hovedsekvens i celle 10-94
- 2) Første bibliotekssekvens i celle 150-189
- 3) Anden - i celle 190-233
- 4) Data på kanal 100-149

Kørslen skal begynde i celle 12. Alle strimlerne slutter med g men ingen af dem indledes med definition af løbende adresse, hvorfor dette må gøres manuelt. Endelig ønskes programmet gemt på kanal 268-293 til senere sammenligning.

Beskriv hvad operatøren må foretage sig før selve kørslen kan starte.

Øvelse 15.23.

Et program er lagret i celle 50-249 og benytter celle 300-647 til data og resultater (lagret som maskintal). Hvad skal man gøre for at få det hele gemt på kanal 268-293?

Efter kørslen ønskes udskrift af alle de celler, hvis indhold er ændret under kørslen. Hvordan kan dette gøres?

Øvelse 15.24.

Et program, der bl.a. benytter navnene d4 og d5, er indlæst og sidst på strimmelen findes bl.a. styrelinien  $x_{261}$ . Under kørslen ønskes indholdet i celle d4 ændret til ca  $r+41$ , hh  $r+d5$ . Hvordan kan dette gøres?

Øvelse 15.25.

Under kørsel med et program, der kun bruger celle 10-319 i ferritlageret, ønskes følgende udskrifter hver gang celle 117 passeres: Indholdet af RF-registeret, celle 109 på ordreform og celle 315-319 som flydende tal. Hvad skal man taste efter indlæsningen af programmet?

Efter nogen tids kørsel ønskes disse kontroludskrifter ikke længere. Hvordan kan de frakobles under kørselen?

Øvelse 15.26.

Et program bruger celle 10-900 i ferritlageret samt tromlekanal 58-200. Hvad skal man taste for at få kontroludskrift af R's indhold som heltal de første 5 gange celle 91 passeres, samt kontroludskrift af celle 750-51 som heltal de første 10 gange celle 213 passeres?

Øvelse 15.27.

Under kodningen af et program forudses det, at på et bestemt sted i programmet vil det være nyttigt at få kontroludskrift af p-registeret, indikatoren og celle b13 til b13+49 de første 6 gange dette programpunkt gennemløbes. Hvordan kan denne kontroludskrift kodes?



15.3.5 Bibliotekssekvenser.Øvelse 15.28.

Skriv en bibliotekssekvens, der finder det numerisk mindste tal blandt en række givne maskintal med følgende konventioner:

1) De givne tal er lagret på tromlen på konsekutive kanaler. Adressen på den første kanal og nummeret på den første benyttede celle på denne kanal står som adressetal og talletal i cellen efter sekvenshoppet.

2) Antallet af tal står som adressetal i den næste celle igen. Indhoppet til sekvensen har derfor formen

$$\begin{array}{l} [s] \quad \text{HS } m \\ [s+1] \quad \text{QQ } \langle \text{beg.kanal} \rangle \text{ t } \langle \text{beg.celle} \rangle \\ [s+2] \quad \text{QQ } \langle \text{antal tal} \rangle , \dots \end{array}$$

3) Ved udhop skal det fundne tal stå i R-registeret; kanalnummer og cellenummer (på kanalen) for dette tal skal stå som adressetal og talletal i celle  $[s+1]$ , og tallets nummer i rækken af de givne tal skal stå som adressetal i celle  $[s+2]$ . Udhopet skal ske med ordren  $\text{HR } s+2$ .

4) Hvis der er flere tal med samme numeriske værdi mindre end de øvrige, skal indikatoren TA etstilles; ellers skal  $\text{TA} = 0$  ved udhoppet. Resten af indikatoren må ikke ændres.

5) Alle registre på nær R-registeret skal være uændrede ved udhop.

Forsyn derefter programmet med de nødvendige styrelinier, så det bliver en rigtig GIER-bibliotekssekvens.

Øvelse 15.29.

Skriv et program der tester sekvensen fra øvelse 15.28 grundigt. Konstruer nogle datasæt, der sikrer at alle ordrer i sekvensen har været gennemløbet, og at alle de logiske betingelser i sekvensen har været i funktion.

16. OVERSIGTER.

### 16.1 Korrespondancen mellem talværdier og typografiske symboler.

Nedenstående tabel er en let udvidet udgave af tabellen i afsnit 8.4 i Lærebog I; den viser både for flexowriter, skrivemaskine og Anelex sammenhængen mellem adressen i en SY-ordre og det udlæste symbol. For Anelex'en gælder nedenstående for den normale opkobling (jfr. afsnit 9.4.3, side 21).

	Flexowr.		Skrivem.		Anelex	
	Low. Case	Up. Case	Low. Case	Up. Case	Low. Case	Up. Case
0	Ml.slag		Ml.slag		Ml.slag	
1	1	✓	1	✓	1	£
2	2	x	2	x	2	x
3	3	/	3	/	3	/
4	4	=	4	=	4	=
5	5	;	5	;	5	;
6	6	[	6	[	6	[
7	7	]	7	]	7	]
8	8	(	8	(	8	(
9	9	)	9	)	9	)
10	ubenyttet		ubenyttet		Ml.slag	
11	Stop Code		ubenyttet		Stop	
12	ubenyttet		ubenyttet		Ml.slag	
13	ubenyttet		å	Å	Å	
14	-		-		*	'
15	ubenyttet		ubenyttet		%	&
16	0	^	0	^	0	↑
17	<	>	<	>	<	>
18	s	S	s	S		S
19	t	T	t	T		T
20	u	U	u	U		U
21	v	V	v	V		V
22	w	W	w	W		W
23	x	X	x	X		X
24	y	Y	y	Y		Y
25	z	Z	z	Z		Z
26	ubenyttet		ubenyttet			†
27	,	10	,	10		,
28	ubenyttet		ubenyttet		Ml.slag	
29	ubenyttet		rød farve		ubenyttet	
30	Tabulator		Tabulator		Tabulator	
31	Punch Off		ubenyttet			\$

	Flexowr.		Skrivem.		Anelex	
	Low. Case	Up. Case	Low. Case	Up. Case	Low. Case	Up. Case
32	-	+	-	+	-	+
33	j	J	j	J		J
34	k	K	k	K		K
35	l	L	l	L		L
36	m	M	m	M		M
37	n	N	n	N		N
38	o	O	o	O		O
39	p	P	p	P		P
40	q	Q	q	Q		Q
41	r	R	r	R		R
42	ubenyttet		ubenyttet		For.skift	*)
43	ø	Ø	ø	Ø		Ø
44	Punch On		ubenyttet			α
45	ubenyttet		ubenyttet		Linespr.	*)
46	ubenyttet		ubenyttet		Ml.slag	
47	ubenyttet		ubenyttet		Ml.slag	
48	æ	Æ	æ	Æ		Æ
49	a	A	a	A		A
50	b	B	b	B		B
51	c	C	c	C		C
52	d	D	d	D		D
53	e	E	e	E		E
54	f	F	f	F		F
55	g	G	g	G		G
56	h	H	h	H		H
57	i	I	i	I		I
58	Low. Case		Low. Case		Low. Case	
59	.	:	.	:	.	:
60	Up. Case		Up. Case		Up. Case	
61	ubenyttet		ubenyttet		Ml.slag	
62	ubenyttet		sort farve		ubenyttet	
63	Tape Feed		ubenyttet		ubenyttet	
64	CR		CR		CR	

\*) "Linespr." betyder at Anelex'en kan styres til at overspringe et ønsket antal linier, når den får ordren SY 45. "For.skift" betyder formularskift, se afsnit 9.4.2, side 21.

## 16.2 Indhop og layout ved taltrykning.

Nedenfor findes en kort oversigt over formen af indhop og layout og betydningen af de enkelte parametre ved udlæsning med standardsekvensen i HJÆLP. En mere detaljeret beskrivelse findes i afsnit 12.2, side 111.

Indhop: HS <adr.>  
 QQ <trimningsadr.>, ... } eller QQ <trimningsadr.>, HS <adr.>

<adr.> = m+0 :  $R_{0-9}$  som heltal  $\geq 0$   
 m+1 :  $R_{0-9}$  - heltal,  $-512 \leq h \leq 511$   
 m+2 :  $R_{0-9}$  - heltal  
 m+3 : R - maskintal  
 m+4 : RF - flydende tal

### Layout:

QQ b.3+h.7+d.13+f1.9+n.14+bE.17+f2.19+g1.23+g2.27+g3.31+g4.35+g5.39

Parameter	Betydning
$0 \leq b \leq 15$	Antal betydende cifre
$0 \leq h \leq 15$	Antal heltalscifre
$0 \leq d \leq 15$	Antal decimaler
$0 \leq f1 \leq 3$	Fortegnstrykning for taldelen: Plustegn foran pos.tal: $f1 = 2$ - erstattet af ml.slag: $f1 = 1$ - helt fjernet: $f1 = 0$ Fortegn som første anslag: $f1 = 3$
$0 \leq n \leq 1$	Nultrykning: Nul foran decimalpkt.: $n = 1$ Ml.slag - - : $n = 0$
$0 \leq bE \leq 7$	Antal betydende cifre i exponenten
$0 \leq f2 \leq 3$	Fortegnstrykning for exponenten: Betydning som ved f1
$0 \leq g1 \leq 15$	Antal anslag i 1.ciffergruppe
$0 \leq g2 \leq 15$	- - - 2. -
$0 \leq g3 \leq 15$	- - - 3. -
$0 \leq g4 \leq 15$	- - - 4. -
$0 \leq g5 \leq 15$	- - - 5. -

16.3 Understregede bogstaver i SLIP.

Nedenfor angives betydningen af alle de understregede bogstaver ved SLIP-indlæsningen; hvor forklaringen er for kortfattet, henvises til kapitel 11, specielt afsnit 11.6.5, side 92.

<u>b</u> <erklæringer>	:	Ferritlager-blokhovede ( <u>b</u> alene er helt blind)
<u>b</u> i=<def.adr.>,...	:	Ferritlager-blokhovede
<u>b</u> k=<def.adr.>,...	:	Tromleblokhovede
<u>c</u> <heltal>	:	Definition af løb.adresse
<u>d</u> <navn>=<def.adr.>,...	:	Definition af løb.adresse og/eller navne ( <u>d</u> er overflødig)
<u>e</u>	:	Blokafslutning
<u>e</u> <def.adr.>	:	Afslutning af indlæsning
<u>f</u>	:	Flydende tal
<u>h</u> <hp-navn>	:	Kald af hp-program
<u>l</u>	:	Strimmelindlæsning
<u>m</u>	:	Maskintal
<u>n</u>	:	Ophæv automatisk relativmærkning af adresser
<u>r</u>	:	Sæt automatisk relativmærkning
<u>s</u>	:	Skrivemaskineindlæsning
<u>t</u>	:	Tekst
<u>u</u> <def.adr.>	:	Definition af udhopsadressen
<u>x</u>	}	: Gem katalog over benyttede navne
<u>x</u> <def.adr.>		
<u>z</u>	}	: Hent katalog over benyttede navne
<u>z</u> <def.adr.>		

Alle øvrige understregede bogstaver har samme virkning som s.

#### 16.4 Fejludskrifter fra HJÆLP og SLIP.

Nedenfor er anført alle de i HJÆLP og SLIP indbyggede fejludskrifter med en kort forklaring af betydningen. Iøvrigt henvises til afsnit 11.8, side 102, og afsnit 13.5, side 187.

Udskrift med rødt	Betydning
1 <løb.adr.>	Syntaktisk fejl
2 <løb.adr.>	Udefineret navn brugt forkert
3 <løb.adr.>	Ikke erklæret navn brugt
4 <løb.adr.>	Fejl i erklæring
5 <løb.adr.>	Ubenyttet strimmelsymbol
6 <løb.adr.>	Forkert kontroldefinition
7 <løb.adr.>	Syntaktisk fejl i tal
8 <løb.adr.>	Talkapacitet sprængt
9 <løb.adr.>	For mange navne og blokke
<navn> <adr.1><adr.2>	Navnet er ikke defineret ved blokslut. Det er sidst benyttet i adressedelen i celle <adr.1> og i talletallet i celle <adr.2>. (Skrives med sort).
gal information	Et af hp-programmerne har fået opgivet en forkert parameter.
sumfejl	Checksummen for det ønskede hp-program stemmer ikke.
tomt hp	Det ønskede hp-program står ikke i hp-kataloget
FEJL	Fejl på en af de låsede kanaler (nr.1-31).

16.5 Hjelpeprogrammernes virkning og parametertyper.

I nedenstående tabel kan man for hvert hjælpeprogram finde virkningen af programmet samt de parametertyper hvert program

Navn	Virkning
<u>h</u> start	Lageret tømmes, hp-katalog initialiseres, checksummer kontrolleres
<u>h</u> kontrol	Checksummer kontrolleres
<u>h</u> tryk	Udskrift af lager og registre
<u>h</u> kompud	Komprimeret udskrift af lager
<u>h</u> gem	Flytning af FL-billedet
<u>h</u> hent	Retablering af FL-billedet
<u>h</u> -sam	Sammenligning af tromleafsnit
<u>h</u> ret	Indlæsning til registre
<u>h</u> hp ind	Indsættelse af hp-parenteser
<u>h</u> hp ud	Fjernelse af hp-parenteser
<u>h</u> slip	Indhop til SLIP
<u>h</u> hop	Udskrift af alle hop under kørsel



kan kaldes med. Derimod er den specifikke virkning af hver parameter ikke medtaget her (men kan findes i kapitel 13). Tabellen omfatter alle standardhjælpeprogrammerne samt hopover-vågningsprogrammet "hop".

Parametre
Ingen eller een af vilk. form
Ingen eller een af vilk. form
<udskr.form>,<registernavn> <udskr.form><kanal>.39 + <beg.adr>t<slutadr> bt<s><antal>.39 + <outputenhed>t<basisadr> { nt<2-potens>t<10-potens> qq<layout>
<beg.kanal>/<beg.celle>/<slutkanal>/<slutcelle><mærke> <u-adr>/<e-adr>/<udhop>/<helt lager> c 3c
Ingen eller sk<beg.kanal>
Ingen eller sk <beg.kanal> sk<antal kanaler>.39 + <beg.kanal 1>t<beg.kanal 2>
sk<beg.kanal 1>t<beg.kanal 2> udskriftspecif. som for <u>h</u> tryk
<registernavn> <SLIP-information>
qq <fri celle>.29 + <fri kanal>t<basisadr>.29 + <beg.kanal> <u>h</u> <hp-navn>/<n>.29 + <udpluksadr> <parametre for det valgte hp>
qq <basisadr>.39 + <udpluksadr>t<beg.kanal>
qq <inputenhed>
bt <lageradr for "hop">t<outputenhed> qq <beg.adr>t<slutadr>



INDIKATOR

KLAR

FEJL

REGISTER

0 t 0A 0B TA TB PA PB 0A 0B RA RB KA KB YE M1 M2 M3 M4 h SF TO TR L R M 0 H L F r1 s1 r2 s2 in ta tk bl bs by

ADRESSEDEL

TÆLLEDEL

GRUNDOPERATION

GRUNDOPERATION

-0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41

S () r s X V D IO IA IA' B / F

HØJTALER

NORMAL

MIKROTEMPI

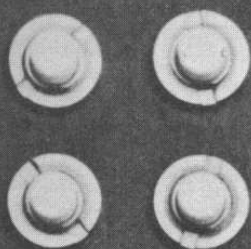


START STOP

START STOP







Reset



HP

— K<sub>A</sub>

— K<sub>B</sub>

— Klar

— YE

— Str.læs. par.fejl

— Tromle fejl

— TO fejl

— HP spærret

— i-løkke

GIERS's hjælpemanøvrebord med HP-knap m.v.



STIKORDSREGISTER

Administratorer	92
Adressedel	61
Adressedel med defineret navn	65
Adressedel med undefineret navn	62
Adressedel, syntax	107
Adresse i højre halvordsordre	68
Aflægning af hulkort	26
Aflæsning, successiv	27
Anelex linieskriver	17
b (i layout)	114
<u>b</u>	79
Basistrimning til "tryk" og "sam"	156
bE (i layout)	116
Betydende cifre (i layout)	114
Bibliotekssekvenser	190
Blind information i SLIP	58
bl-lampe	37
Blok	78
Blokafslutning	81
Blokhoved	79
Blok på tromle	88
Blok, syntax	106
bs-lampe	37
Bufferlager, hulkortlæser	24
by-lampe	37
Ciffergrupper (i layout)	114
Cifre foran komma (i layout)	114

d (i layout)	114
Decimalantal (i layout)	114
Defineret navn	62
Definitionslinie	84
Disponibelt lager i HJÆLP	55, <u>132</u>
<u>e</u>	81
Etikette	85
<u>f</u>	93
f1 (i layout)	115
f2 (i layout)	116
"FEJL"	<u>189</u> , 215
Fejllæsning, hulkort	24
Fejlnummer (SLIP)	<u>103</u> , 215
Fejl på låsede kanaler	189
Fejludskrift (HJÆLP)	<u>187</u> , 215
Fejludskrift (SLIP)	<u>102</u> , 215
F-lampe	36
Ferritlagerbillede	132
Flydende overløb	139
Flydende tal	73
Formularskift, linieskriver	21
Fortegnstrykning (i layout)	114
<u>g</u>	96
g1...g5 (i layout)	116
"gem", hjælpeprogram	<u>167</u> , 216
Grundoperation	59
h (i layout)	114
<u>h</u>	95, 140
Heltal	74
"hent", hjælpeprogram	<u>168</u> , 216
Heltalsgruppe	74
HJÆLP	129
Hjælpeinformationslinie	140



Hjælpeknap	45, 46
Hjælpemanøvrebord	45
Hjælpemanøvrebord, foto	221
Hjælpeprogram: "gem"	<u>167</u> , 216
Hjælpeprogram: "hent"	<u>168</u> , 216
Hjælpeprogram: "hop"	216
Hjælpeprogram: "hp ind"	<u>174</u> , 216
Hjælpeprogram: "hp ud"	<u>181</u> , 216
Hjælpeprogram: "kompud"	<u>160</u> , 216
Hjælpeprogram: "kontrol"	<u>148</u> , 216
Hjælpeprogram: "læs hp"	184
Hjælpeprogram: "ret"	<u>172</u> , 216
Hjælpeprogram: "sam"	<u>169</u> , 216
Hjælpeprogram: "slip"	<u>182</u> , 216
Hjælpeprogram: "start"	<u>150</u> , 216
Hjælpeprogram: "tryk"	<u>152</u> , 216
Hjælpeprogrammer	146
Hjælpeprogrammer, katalog	<u>147</u> , 149
Hjælpeprogrammer, kodning af	185
Hjælpeprogrammer, placering	146
Hjælpeprogrammer, programstyring	142
Hjælpeprogrammer, strimmelstyring	141
HJÆLP, fejludskrifter	187
HJÆLP, placering	131
HJÆLP-uaflåse	185
HJÆLP-uaflåse uden om ALGOL	185
h-lampe	35
H-lampe	36
"hp ind", hjælpeprogram	<u>174</u> , 216
HP-knap	45, 46, <u>132</u>
HP-knap spærret	133
HP-knap spærret-lampe	47
HP-parentes	172
"hp ud", hjælpeprogram	<u>181</u> , 216
HS 1	139
HSF 2	100, 139
Hulkort	23

Hulkortlæser	22
Højre halvordsordre, adresse i	68
i, løbende adresse	67, <u>77</u>
"i-løkke"-lampe	47
Indikatorlamper	35
Indlæseprogram, eksempel	49
Indlæseprogrammet SLIP	55
Indlæsning af komprimeret strimmel	165
Indlæsning af ordrer	59
Indlæsning af tal og tekst med SLIP	73
Indlæsning fra skrivemaskine	14
Indlæsning fra strimmel	11
Indlæsning, manuelt styret	99
Indlæsning, programstyret	100
in-lampe	37
k, løbende kanalnummer	67, <u>77</u>
Kanal 0	4, 189
Kanaladresse	3
Kanalhop	9
Kanallåseregistre	5
Katalog over hjælpeprogrammer	<u>147</u> , 149
KLAR-lampe	35, 46
Kode for linieskriver	21
Kodning af hjælpeprogrammer	185
Komprimeret strimmel	160
"kompud", hjælpeprogram	<u>160</u> , 216
Kontrol af kanalaflysning	6
"kontrol", hjælpeprogram	148
Kontroludskrift (SLIP)	104
<u>l</u>	93
Lamper på hjælpepult	46
Layout	114
Linier i SLIP	57
Linieskriver	17

Linieskriverens kode	21
Linie, syntax	106
L-lampe	36
Lokalt navn	78
Lower Case, skrivemaskineindlæsning	15
Lower Case, skrivemaskineudlæsning	16
Læsehastighed, hulkortlæser	23
"læs hp", hjælpeprogram	184
Løbende adresse, i	67, <u>77</u>
Løbende kanalnummer, k	67, <u>77</u>
Låsede kanaler	4, 189
Låsede kanaler, fejl på	189
<u>m</u>	93
M <sub>1</sub> ...M <sub>4</sub> -lamper	35
Manuelt styret indlæsning	99
Manøverbord foto	219
Manøverbord	35
Manøverbordets lamper	34
Maskintal	74
MIKROTEMPI START	39
MIKROTEMPI STOP	<u>38</u> , 39, 46
M-lampe	36
n (i layout)	115
<u>n</u>	94
Navn	<u>61</u> , 105
Navn, defineret	62
Navn, lokalt	78
Navn, undefineret	62
Navn, værdi af	105
NORMAL START	38
NORMAL STOP	39
Nul, trykning af (i layout)	114
O-lampe	35, 36
Operationstider for LY-ordrer	17

Ordrebestanddele	38
Ordreindlæsning	59
Ordrelinie, syntax	107
Oversigt, Anelextegn	212
- , Fejludskrifter fra HJÆLP og SLIP	215
- , Flexowritertegn	212
- , Hjælpeprogrammer	216
- , Layout	213
- , Skrivemaskinetegn	212
- , Taltrykning	213
- , Talværdier for typografiske tegn	212
- , Understregede bogstaver i SLIP	214
Parentes (hjælpeprogram)	172
Paritetscheck, indlæsning	12
Paritetsfejl	12
Paritetskontrol, tromlelæsning	4, 7
Paritetsmærke	13
Perforator	13
Placering af HJÆLP	131
Positionsangivelse	68
p-register	37
Program	98
Programmeret indhop i HJÆLP	138
Programinformation (i SLIP)	57
Programstyret hjælpeprogram	142
Programstyret indlæsning	100
Program, syntax	106
<u>r</u>	93
r1-lampe	37
r2-lampe	37
Redefinition	87
Registre	34
Relativmærket adresse med navne	63, 66
RESET	38, 39
RESET-lampe	46

Resulterende adresse ved hulkortlæsning	29
"ret", hjælpeprogram	<u>172</u> , 216
Rettelse i lager	144
Rettelse i register	145, 172
R-lampe	36
<u>s</u>	93
s1-lampe	37
s2-lampe	37
"sam", hjælpeprogram	<u>169</u> , 216
SF-lampe	35
Simultan tromletransport	6
Skalafaktor (i layout)	120, 122
Skilletegn mellem adressetal og talletal	71
Skrivemaskine	14
SLIP	48, <u>55</u>
"slip", hjælpeprogram	<u>182</u> , 216
SLIP syntax	106
Sortering af hulkort	27
Spærret HP-knap	133
Start	41
"start", hjælpeprogram	<u>150</u> , 216
Startknapper	38, 39
Stop	40
Stopknapper	38
Strimmellæser	11
Strimmelstyring af hjælpeprogram	141
Styrelinie	<u>77</u> , 94
Styrelinie, syntax	109
Styreparametre til hjælpeprogram	140
Styringsinformation i SLIP	57
Successiv aflægning, hulkort	27
"sumfejl"	<u>188</u> , 215
Syntax for SLIP-kode	106
t	71
<u>t</u>	93

ta-lampe	37
Talindlæsning	73
Tallinie, syntax	108
Taltrykning med HJÆLP	111
Tekstindlæsning	75
Tekstlinie, syntax	108
Teksttrykning med HJÆLP	124
Titalsexponent (i layout)	114
tk-lampe	37
T-lampe	35
TO-lampe	36
"tomt hp"	<u>188</u> , 215
Tre tromler	5
Trimning, særlige muligheder	122
Trimning til taltrykning med HJÆLP	113
Trinvis kørsel	43
TR-lampe	36
Tromleblok	88
Tromleblok, begrænsninger	91
Tromleblokhoved	89
Tromlefejl	<u>8</u> , 47
Tromlelager	1
"tryk", hjælpeprogram	<u>152</u> , 216
Tryknapregister til kanallåsning	5
Tælletal	71
Tælletal, syntax	108
<u>u</u>	<u>95</u> , 163
Udefineret navn	62
Udførelse af enkelt ordre	41
Udhop fra SLIP	101
Udhop fra HJÆLP	136
Udhopsadresse	<u>95</u> , 101, 136, 163
Udlæseenhed, valg af	18
Udlæsning	110
Upper Case, skrivemaskineindlæsning	15
Upper Case, skrivemaskineudlæsning	16

Værdi af navn	105
<u>x</u>	<u>25</u> , 97
Ydre enheder	11
YE-lampe	35
<u>z</u>	<u>25</u> , 97
Øvelser	200 ff.