

**REGNECENTRALEN**

**SCANDINAVIAN INFORMATION PROCESSING SYSTEMS**

82-GSL 400

December 1965

Chr. Andersen,

Kim Andersen

Lærebog i Kodning for Gier  
Bind III

Gier, Programming, Education, Manual

This volume contains the following chapters: The Gier System, The Buffer, The Carrousel, Conventional Magnetic Tape Stations, Magnetic Tape Administration, and The Disk File. 59 pages.

Informatik og Matematisk Modellering  
DTU



380020009559

C.O  
Reg

Hist.

**SYSTEM LIBRARY**

DK-2500 VALBY · BJERREGAARDSVEJ 5 · TELEPHONE: (01) 46 08 88 · TELEX: 64 64 rcinf dk · CABLES: INFOCENTRALEN

INDHOLDSFORTEGNELSE

Forord

17. GIER-systemet

- 17.1 Indledning
- 17.2 GIER-systemet
- 17.3 il og us ordrer

18. Bufferen

- 18.1 Indledende beskrivelse
- 18.2 Transporter mellem ferritlageret og bufferen
  - 18.2.1 Parameterordet
  - 18.2.2 Adressedelen i il og us ordren
  - 18.2.3 Eksempel
  - 18.2.4 Bemærkninger
- 18.3 Bufferen som reservelager

19. Karrusellen

- 19.1 Indledning
- 19.2 Transporter mellem buffer og karrusel
- 19.3 Statusord
  - 19.3.1 Paritetsfejl
  - 19.3.2 Informationsfejl
  - 19.3.3 Statusordets indhold
  - 19.3.4 Transport af statusordet
  - 19.3.5 Venteordre
  - 19.3.6 Bemærkning
- 19.4 Eksempler
- 19.5 Bemærkning

20. Konventionelle magnetbåndsenheder

- 20.1 Indledning
- 20.2 Transporter mellem buffer og magnetbånd
  - 20.2.1 Parameterord
  - 20.2.2 Statusord
  - 20.2.3 Adressen i il eller us ordren
- 20.3 Eksempler
- 20.4 Venteordre
- 20.5 Sikkerhed
- 20.6 Hastighed

21. Magnetbåndadministration

- 21.1 Indledning
- 21.2 Bånddåb
- 21.3 Administrationsprogram
- 21.4 Etikette
- 21.5 Initialiseringssekvens
  - 21.5.1 Kald af initialiseringssekvens
  - 21.5.2 Arbejdsbånd
  - 21.5.3 SLIP-kode til initialiseringssekvens
  - 21.5.4 Eksempel
- 21.6 Båndsekvens
  - 21.6.1 Parameterord
  - 21.6.2 Statusord
  - 21.6.3 Kald af båndsekvens
  - 21.6.4 Funktionsadresser i båndsekvens
  - 21.6.5 SLIP-kode til båndsekvens
  - 21.6.6 Filemark
- 21.7 Eksempler

*Historie / C. O. Reg*  
*D. 3.2*  
Institutet for Datateknik  
Biblioteket  
Danmarks Tekniske Universitet  
2800 Lyngby  
A/S REGNECENTRALEN  
BIBLIOTEKET  
Malmparken 10  
2750 Ballerup - 97 53 66 ✓

2857

## 22. Disk-file

- 22.1 Indledning
- 22.2 Transporter mellem GIER og disk-file
  - 22.2.1 Valg af gruppe
  - 22.2.2 Valg af sektion
  - 22.2.3 Læse og skriveordrerne
  - 22.2.4 Eksempel
  - 22.2.5 Disk-filens opbygning
- 22.3 Effektiv udnyttelse af disk-file
  - 22.3.1 Flytning af læse-skrivehovederne
    - 22.3.1.1 Eksempel
    - 22.3.1.2 Bemærkning
  - 22.3.2 Eksempel
- 22.4 Gemning af grupperregisteret
  - 22.4.1 Eksempel
  - 22.4.2 Bemærkning

December 1965.

F O R O R D.

De to første dele af lærebogen i kodning for GIER indeholder beskrivelser af såvel selve GIER-maskinen og en del af de ydre enheder, der benyttes i forbindelse med GIER, som af de kodesystemer, kaldet HJÆLP og SLIP, der benyttes ved udarbejdelsen af programmer for GIER.

Nuværende tredje del af lærebogen indeholder beskrivelser af det nye udstyr, der i løbet af de sidste par år er udviklet og tilpasset for GIER-maskinen. Det drejer sig dels om et bufferlager, dels om forskellige typer af magnetbåndstationer. Endelig omtales det disk-file lager, der i nær fremtid vil kunne indbygges i GIER i stedet for tromlelageret.

Bortset fra selve beskrivelsen, der vil kunne læses uden videre, er alle eksempler skrevet for læsere med kendskab til SLIP. Det må derfor anbefales de læsere, der endnu ikke er fortrolige med indholdet af anden del af lærebogen, at repetere denne, før de går i gang med nærværende tredje del.

I øvrigt er bogen - i lighed med første og anden del - skrevet for brugere af GIER, d.v.s. for kodere, hvorimod tekniske beskrivelser af enhver art er udeladt.

Bogens kapitler er nummereret i fortsættelse af kapitlerne i lærebøgerne I og II.

Under arbejdet med bogen har vi haft en uvurderlig støtte af Inger Møller, der har gennemarbejdet hele bogen og ved flere lejligheder har haft afgørende indflydelse på den endelige fremstilling. Vi bringer Inger Møller vor varmeste tak.

I øvrigt takker vi de af Regnecentralens andre medarbejdere, der på forskellig måde har hjulpet os.

Chr. Andersen

Kim Andersen

## 17. GIER-systemet.

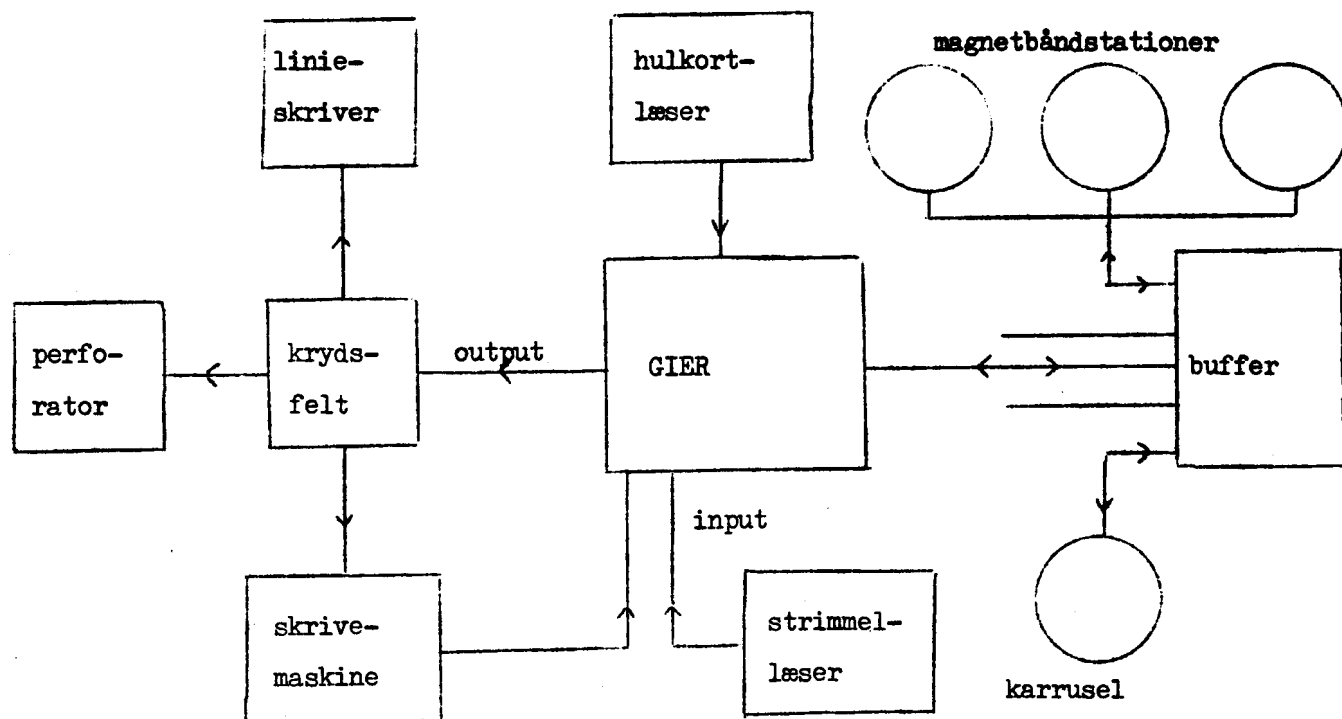
### 17.1. Indledning.

Medens tromlelageret, der er beskrevet i såvel 1. som 2. del af lærebogen, er en del af det anlæg, der betegnes som STANDARD-GIER, er bufferlageret, der skal omtales i det følgende, ligesom hulkortlæseren og linieskriveren, der begge er beskrevet i 2. del af lærebogen, en del af det såkaldte GIER-system. Udover dette består GIER-systemet af et antal ens eller forskellige magnetbåndstationer. Bufferlageret, eller kortere bufferen, tjener dels som reservelager for GIER's ferritlager og dels som gennemgangslager ved transporter mellem ferritlageret og magnetbåndstationerne og ved transporterne mellem de forskellige magnetbåndstationer.

De magnetbåndstationer, der foreløbigt kan tilsluttes, er: konventionelle båndenheder af diverse fabrikater og FACIT KARRUSEL ECM-64. Disse magnetbåndsenheder skal omtales udførligt i det følgende. Inden vi beskriver de forskellige bestanddele hver for sig, bringer vi en skitse af GIER-systemet.

### 17.2 GIER-systemet.

Nedenstående tegning viser de forskellige dele af GIER-systemet og transportlinierne mellem disse.



Det bemærkes, at der er 4 datakanaler mellem bufferen og de ydre enheder. På ovenstående skitse er kun to af disse kanaler benyttet: en til karrusellen og en til samtlige tre båndstationer. Andre muligheder vil sikkert blive anvendt i nogle af de GIER-systemer, der vil blive opstillet i den nærmeste fremtid. For eks. vil systemer uden karrusel, men med fire båndstationer blive benyttet. Om disse fire stationer vil blive tilknyttet via hver sin af de fire kanaler - hvad der vil være bedst, men også dyrest - eller via en fælles kanal, er ikke afgjort på nuværende tidspunkt.

17.3 il og us ordrer.

Administration af bufferen og magnetbånd i GIER foregår med ordrene il og us. Disse ordrer er ikke omtalt i operationslisten i 1. del af lærebogen.

Almindeligt gælder, at:

il-ordren styrer transporter til GIER ( indlæs ), og  
us-ordren styrer transporter fra GIER ( udskriv ).

il har talværdien 44 og us talværdien 45.

Funktionen af en il eller us ordre afhænger dels af ordrens resulterende adresse og i flere tilfælde desuden af R-registerets indhold. For at få udført de forskellige operationer kræves således normalt to ordrer: først en ordre der anbringer et såkaldt parameterord i R-registeret, og dernæst selve il eller us ordren.

Angående muligheden for at anvende flere datakanaler simultant bemærkes det, at transport langs de forskellige kanaler udmærket kan foregå på denne måde. Derimod er der ikke mulighed for, at udføre en il eller us ordre, der benytter en given datakanal, før en eventuel transport langs denne er tilendebragt.

## 18. Bufferen.

### 18.1 Indledende beskrivelse.

Bufferlageret er i fysisk henseende opbygget på samme måde som ferritlageret i GIER, d.v.s. det består af en samling ferritkerner. Disse er ordnet i 4096 celler á 42 kerner. Hver celle kan altså indholde et ord bestående af 42 bit, svarende til et ord i ferritlageret.

Bufferens kapacitet er fire gange ferritlagerets og ca. en trediedel af tromlelagerets.

Som omtalt i 17.1 er bufferen et gennemgangslager og reservelager for GIER. Bufferen kan ikke uden videre benyttes som et forstærket og dermed forbedret ferritlager, idet det ikke er muligt at udføre direkte transporter mellem GIER's registre og bufferen.

### 18.2 Transporter mellem ferritlageret og bufferen.

Der er mulighed for at overføre en sammenhængende del i ferritlageret - kaldet en blok - til bufferen, og omvendt kan en blok i bufferen overføres til ferritlageret.

Før transporterne kan finde sted, skal koderen sørge for at anbringe et parameterord i R-registeret.

#### 18.2.1 Parameterordet.

Indholdet af parameterordet ved transporter mellem ferritlageret og bufferen skal være:

- pos. 0 - 9 blokkens begyndelsesadresse i ferritlageret
- pos. 10 - 19 bloklængden (antal ord der skal transporteres)
- pos. 28 - 37 blokkens begyndelsesadresse i bufferen.

Indholdet af de øvrige positioner - pos. 20 - 27 og pos. 40-41 - er irrelevante.

Ved en transport udover celle 1023 i ferritlageret og celle 4096 i bufferen, vil transporten fortsætte i celle 0 og fremefter.

#### 18.2.2 Adressedelen i il og us ordren.

Den resulterende adresse skal være 0 uanset om transporten er til eller fra GIER.

#### 18.2.3 Eksempel.

En programdel, der står i ferritlageret fra celle 240 til celle 880, ønskes overført til bufferens celle 3000 og fremefter. Derefter skal indholdet af celle 110 til celle 435 i bufferen overføres til celle 250 og fremefter i ferritlageret.

Et program til disse transporter kan f. eks. se sådan ud:

arnr1V	;R:=parameterord
qqm240+641.19+3000.39	;parameterord
us	;udskriv
arnr1V	;R:=parameterord
qqm250+326.19+110.39	;parameterord
il	;indlæs

#### 18.2.4 Bemærkninger.

Under transporter mellem buffer og ferritlager er GIER beslaglagt, d.v.s. der kan ikke udføres andre ordrer i GIER, før end transporten er tilendebragt. Dog vil tromletransporter, der startede, blive udført på normal måde samtidig med buffertransporten. Ligeledes vil transporter mellem bufferen og de til denne koblede ydre enheder kunne foregå simultant med GIER-buffer overførslerne. Man må altså passe på, at det område man bruger i bufferen til overførslerne mellem GIER og bufferen, og det område der benyttes til samtidig transport mellem bufferen og magnetbånd ikke overlapper hinanden. Hvis man imidlertid ønsker at benytte samme område i bufferen til forskellige transporter, må man med en såkaldt venteordre sidre sig, at den ene transport er tilendebragt, førend den næste startes.



Transporter mellem GIER og bufferen udføres med en hastighed på ca. 40 usec + ca. 16 usec pr. ord.

Efter udførelsen af en us-ordre, er R-registerets mærkebit udefinerede. (Som GIER i dag er konstrueret, bliver de sat lig med det sidst transporterede ords mærkebit). Bortset herfra er alle for koderen tilgængelige registre i GIER uændrede efter en il eller us ordre.

### 18.3 Bufferen som reservelager.

Da transporter mellem GIER og bufferen udføres med en hastighed på ca. 16 usec pr. ord eller 40 ord på ca. 0.7 msec, medens en tromletransport af 1 kanal på 40 ord tager ca. 20 msec, vil det i mange tilfælde være naturligt at benytte bufferen som reservelager i stedet for tromlen.

Hvis et program kræver stor plads til lagring af talmateriale, vil det være hensigtsmæssigt at benytte bufferen hertil. Har man programmer, der er for store til at kunne være i ferritlageret på en gang, vil det desuden være fordelagtigt, at benytte bufferen til lagring af de sekvenser, man hyppigt får brug for under kørslen, hvorimod tromlen benyttes til programdele, som kun sjældent skal bruges.

19. Karrusellen.19.1 Indledning.

Ud over konventionelle magnetbåndsenheder, der beskrives i afsnit 20, kan FACIT KARRUSEL ECM-64 indgå i GIER systemet. En karrusel består af en drejelig krans, på hvilken der er anbragt 64 udskiftelige spoler nummereret fra 0 til 63. Hver spole består af et ca. 1. cm. bredt magnetbånd, hvorpå der er plads til lagring af 16 blokke - nummererede fra 0 til 15 - à 512 ord a 42 bit. Karrusellens kapacitet er således

$$64 \times 16 \times 512 \text{ ord} = 524\ 288 \text{ ord}$$

altså godt en halv million ord. Dette svarer til 512 gange ferritlagerets kapacitet.

plads til fotografi

Afgives der fra GIER ordre om skrivning på eller læsning fra en spole drejes kransen, så den pågældende spole kommer til at stå ud for skrive- og læsehovedet. Båndet spoles af, og den ønskede informationsmængde overføres. Derpå spoles båndet på igen, og karrusellen er klar til en ny transportordre.

Blokinddelingen er fast, og der kan kun overføres et helt antal blokke i hver transport. Yderligere kan hver transport kun aktivere en af de 64 spoler. Det bemærkes, at al transport til og fra karrusellen skal ske via bufferen, samt at der i bufferen kun er plads til 8 blokke ad gangen.

### 19.2 Transporter mellem buffer og karrusel.

Ved transporter mellem buffer og karrusel skal der være anbragt et parameterord i R-registeret inden il eller us ordren kan udføres. Indholdet af dette parameterord skal være:

- pos. 0 - 5 spolenummeret,
- pos. 6 - 9 nummeret på den første blok der skal benyttes på den pågældende spole,
- pos. 16 - 19 antal blokke der skal transporteres
- pos. 28 - 39 begyndelsesadressen i bufferen.

Indholdet af de øvrige positioner - pos. 10 - 15, pos. 20 - 27 og pos. 40 - 41 - er irrelevante. Hvis blokantallet sættes lig med 0 er den pågældende ordre blind. (ordren kan bruges som venteordre se 19.3.5).

Den resulterende adresse i il eller us ordren skal være lig med 7, således at ordrene til transport fra karrusel til buffer og omvendt bliver henholdsvis

il7 og us7.

### 19.3 Statusord.

Et af de vanskeligste tekniske problemer i forbindelse med arbejdet med magnetbånd af enhver art er at få gjort overførslerne til og fra bånd helt pålidelige. For at give brugeren mulighed for at gribe ind, såfremt der forekommer fejl ved en transport, er der indbygget forskellige checks i karrusellen, check for paritetsfejl og check for informationsfejl.

#### 19.3.1 Paritetsfejl.

Ved skrivning på en af karrusellens spoler opfattes et ord som sammensat af 7 karakterer à 6 bit. Når et sådant 6-bit tegn skrives på magnetbåndet, suppleres disse 6 bit altid med en ekstra 7. bit. Dette skulle ske således at antallet af bit, der er lig med 1 altid er ulige - såkaldt ulige paritet. Såfremt et tegn ved enten en læsning eller en skrivning af en eller anden årsag indeholder et lige antal 1'er, taler man om paritetsfejl.

### 19.3.2 Informationsfejl.

Såfremt nummeret på den første blok der skal benyttes (parameterordets pos. 6 - 9) plus antal blokke der skal transporteres (parameterordets pos. 16 - 19) er større end 16, taler man om informationsfejl, d.v.s. at der er krævet transporteret flere blokke end der er til rådighed på den pågældende spole.

### 19.3.3 Statusordets indhold.

Informationen om paritetsfejl eller informationsfejl i en transport har koderen adgang til gennem det såkaldte statusord. Dette er et normalt GIER ord bestående af 42 bit. Såfremt der er konstateret enten en paritetsfejl eller en informationsfejl ved sidste transport mellem buffer og karrusel, bliver pos. 0 i statusordet sat lig med 1 ellers bliver den sat lig med 0. De øvrige positioner vil altid være nulstillede.

### 19.3.4 Transport af statusordet.

Statusordet læses til bufferen ved hjælp af ordren

il23.

Denne ordre kræver også, at et parameterord er placeret i R-registeret inden operationen skal udføres. I dette tilfælde skal parameterordet indeholde en bufferadresse placeret i pos. 28 - 39. Indholdet af alle andre positioner - pos. 0 - 27 og pos. 40 - 41 - er irrelevante. Ordren il23 bewirker da, at statusordet for den sidste transport mellem buffer og karrusel bliver placeret i den celle i bufferen, som parameterordets pos. 28 - 39 angiver. Da statusordet først overføres når transporten er tilendebragt, virker il23 ordren som venteordre.

### 19.3.5 Venteordre.

Hvis man vil sikre sig, at transport mellem buffer og karrusel er afsluttet, før end der fortsættes i programmet, kan man anvende venteordren

us23.

Denne ordre kræver ikke noget parameterord, d.v.s. indholdet af R-registeret er irrelevant. Der sker kun det, at GIER venter til en eventuel karruseltransport er tilendebragt. Som nævnt under 19.2 kan il7 og us7 anvendes som venterodrer, hvis pos. 16 - 19 i parameterordet er nulstillede ( f. eks. kan iln7 eller usn7 anvendes).

#### 19.3.6 Bemærkning.

Enhver karruseltransport, uanset om den er i retning til eller fra GIER, bør (skal) testes for eventuelle fejl ved hjælp af statusordet.

#### 19.4 Eksempler.

Til belysning af hvordan karruseltransporterne kan administreret, skal her anføres et par eksempler.

##### Eksempel 19.4.1

En datamængde, der står i bufferen fra celle 100 til celle 3099, ønskes overført til spole 33 fra blok nr. 7 og fremefter.

Først bemærkes, at hver blok på karrusellen kan rumme 512 ord. De 3000 ord, der skal overføres, kræver således 6 blokke, af hvilke der ikke er brug for mere end en del af den sidste. Da blokklængden er fast, og overførsel kun kan foregå med et helt antal blokke, vil vi få overført  $6 \times 512$  ord = 3072 ord.

Det ønskede program skal altså først overføre den ønskede datamængde til karrusellen. Dernæst skal statusordet læses til bufferen (f. eks. til celle 99) og videre til ferritlageret (f. eks. celle 200). Til sidst skal testes om den pågældende transport er foregået uden fejl. I modsat fald skal transporten gøres om, og der skal testes igen. e.s.v. (Erfaringen viser, at en transport, der ikke er lykkedes i 1. omgang efter en eller flere gentagelser vil foregå som ønsket).

```

.
.
.
e:  arna1,usm7          ; skriv på spole 33 fra blok 7 og frem
                                ; 6 blokke fra celle 100 i bufferen
    arna2,ilm23          ; læs statusord til celle 99 i bufferen
    il, arnm200          ; R:=statusord
    hveLT                ; if fejl then go to e
.
.
.
a1: qqm33.5+7.9+6.19+100.39 ; parameterord til transporten mellem
                                ; bufferen og karrusellen
a2: qqm200+1.19+99.39      ; parameterord til læsning og transport
                                ; af statusord
.
.

```

Da det kun er pos. 28 - 39 i parameterordet der er relevante ved læsning af statusord til bufferen, kan vi benytte samme parameterord ved læsning af statusordet til bufferen som ved videre transport af statusordet fra bufferen til ferritlageret. I dette tilfælde er det a2 der bruges som fælles parameterord.

Hvis man ved en fejl f. eks. havde beordret 6 blokke skrevet på spole 33 begyndende i blok 11, ville dette forsøg på at skrive flere blokke end der er plads til på den pågældende spole, blive registreret som en informationsfejl i statusordet, og omskrivning ville blive foretaget i det uendelige.

#### Eksempel 19.4.2

Da bufferen indeholder 4096 celler, og da hver blok på karrusellen består af 512 ord, er der, hvis man transporterer 8 blokke, netop ikke plads i bufferen til transport af statusordet. D.v.s. man må afvente transportens afslutning. Dette kan f. eks. gøres med venteordren us25. Dernæst må man gemme en af bufferens celler i ferritlageret, hvorefter statusordet via denne celle kan transporteres til ferritlageret. Til sidst må den pågældende buffercelle reableres.

Lad os tænke os at vi ønsker at transportere 8 blokke fra spole 19 blok 0 til bufferens celle 0 og fremefter. Et program til denne transport med efterfølgende test af statusord kan f. eks. være:

```

.
.
ita1,pa3          ; etablering af parameterord
e:  arna, ilm7    ; indlæs fra karrusel til buffer
    usm23, arna3  ; vent til transport er forbi
    il,ilm23      ; gem buffercelle 0 i a1
    arm1P         ; læs statusord via celle 0 i bufferen
                    ; til a2 (=a1+1)
    il,arna2      ; R:=statusord
    hveLT         ; if fejl then go to e
    arna3,us      ; retabler celle 0 i bufferen
.
.
.
a:  qqm19.5+8.19 ; parameterord for transport karrusel
                    ; buffer
a1: qq           ; lagercelle for buffercelle 0
a2: qq           ; lagercelle for statusord
a3: qqm1.19      ; parameterord til transport mellem buffer
                    ; og ferritlager
.
.

```

Det bemærkes at havde det drejet sig om en skrivning på karrusellen, skulle celle 0 i bufferen have været retableret inden statusordet blev testet.

#### Øvelse 19.4.3

Skriv et lignende program, der skriver 8 blokke på spole 19 blok 0 og følgende fra bufferens celle 0 og fremefter med test.

Eksempel 19.4.4

På spole nr 35 og 36 fra blok nr. 0 er lagret 10240 tal, altså 20 blokke. Disse tal skal trykkes i den rækkefølge hvori de står på karrusellen. En egnet taltrykningssekvens tænkes lagret i ferritlageret i celle b og fremefter. Til transport mellem karrusel og bufferen er første halvdel af bufferen ledig, medens der kun er reserveret plads til 8 tal i ferritlageret.

I programmet hentes fire blokke over ad gangen til bufferen i celle 0 og fremefter. Transporten testes for fejl, og tallene transporteres videre til ferritlageret i grupper på 8, hvorpå tallene trykkes.

```

.
.
e:  vym32                ; valg af ydre enhed
    ita,paal             ; etablering af parameterord
    pae1t4               ; etablering af tælling
e2:  sym42,grna2
    pta2t8               ; etablering af parameterord
    ita3,paal2
    pae3t255            ; etablering af tælling
e4:  arna4,ilm7          ; indlæs fra karrusel til buffer
    arna1,ilm23         ; læsning af statusord
    il,arna              ; R:=statusord
    hve4LT              ; if fejl then go to e4
e5:  pae6t7             ; etablering af tælling
    ita5,pael7
    arna2,il             ; indlæs til celle a3 i ferritlageret
    ara6,gra2           ; dan nyt parameterord
e7:  arnt1              ; tryk celle a5+1 (=a3) og følgende
    hsb                 ; go to trykning
    qqm10               ; trykparameter
e6:  btt-1              ; 8 tal ad gangen
    hve7
e3:  btt-1              ; tryk 256x8 tal
    sym64,hve5
    arnm4D
    aca4.itm-1          ; dan parameterord til læsning af næste blok

```





```

e2: grb4,usm7           ; c(b4):=parameterord, skriv 1 blok på karrusellen
    can(e1),hve3       ; test af sidste blok
    hsa2               ; indlæs 512 tal til celle a1 og fremskifter
    arnb,itm-512      ; skift mellem bufferadresse 0 og 512
e:  bt,arb5
    us,can(e)         ; udskriv til bufferen
    arnb5,hhe4       ; etablering af parameterord til næste
                    ; skrivning på karrusel
e4:  srnb5,arb4
    arm1DX           ; og bloknr. med 1
e3:  arnb2,ilm23    ; statusord for sidste transport til
                    ; celle b1 i ferritlageret
    il,arnb1        ; R:=statusord
    hhe5NTX         ; if -,fejl then go to e5
    arnb4,usm7      ; skriv om
e5:  hve3,itm-1
e1:  bt,hve2        ; skriv 20 blokke
    .
    .
    .
b:   qqm512.19      ; parameterord til transport ferritlager buffer
b1:  qq             ; lagercelle for statusord
b2:  qqm1.19+4095.39 ; parameterord til transport af statusord
b3:  qqm35.5+1.19  ; parameterord til transport mellem buffer
                    ; og karrusel
b4:  qq             ; lagerplads for parameterord for forrige
                    ; transport mellem buffer og karrusel
b5:  qqm512.39
    .
    .
a1:  qq

```

Det bemærkes at transporten af statusordet sker gennem celle 4095 i bufferen, samt at kun bloknnummeret i parameterordet - pos. 6 - 9 i celle b3 - tælles op, hvorved spolenummeret - pos. 0 - 5 - automatisk øges med 1, når bloknnummeret bliver større end 15.

19.5 Bemærkning.

Så snart opspolingen af et magnetbånd er påbegyndt, kan en ny transport ordre blive behandlet i GIER, således at ordren vil blive udført så snart opspolingen er tilendebragt.

Angående den hastighed, hvormed karrusellen arbejder, gælder det, at denne dels afhænger af det blokantal der skal transporteres, og dels afhænger af det blokantal der skal transporteres, og dels afhænger af, hvor meget kransen skal drejes. Operationstiderne fremgår af følgende tabel:

drejning af kransen 1 plads	ca. 0.1 sec.
drejning af kransen 32 pladser	ca. 1 sec.
nedspoling til første blok	ca. 0.8 sec.
derefter pr. blok	ca. 0.1 sec.
opspoling fra første blok	ca. 1 sec.
opspoling af hele båndet	ca. 2.3 sec.

En simpel mulighed for beskyttelse af en spole på karrusellen mod skrivning er, at indsætte en stift i kransen ud for den pågældende spole. Herved spærres der for skrivning på spolen, hvormed der uden vanskelighed kan læses fra den.

20.1 Indledning

Konventionelle magnetbåndsenheder af forskellige fabrikater kan benyttes i forbindelse med GIER's centralenhed. Den følgende beskrivelse vedrører båndenheder af typen CDC 9100, men bortset fra hastighedsangivelser samt oplysninger af mere teknisk karakter dækker beskrivelsen i det væsentlige også forholdene for båndenheder af andre fabrikater. Dette gælder specielt for den del af beskrivelsen, der angår programmeringen.

I en magnetbåndsenhed findes der to aksler, hvorpå man kan anbringe spoler. Når enheden skal benyttes, anbringes en spole med magnetbånd på den ene aksel hvorefter den frie ende af magnetbåndet føres gennem forskellige mekanismer blandt andet indeholdende et skrive- og et læsehoved til en anden tom spole, hvor båndet fæstnes. Denne spole er i forvejen anbragt på den anden aksel. Mellem spolerne passerer magnetbåndets to såkaldte kapstansruller, en der kan trække båndet frem og en der kan føre det tilbage. Ved hjælp af to vacuumkamre holdes magnetbåndet tilpas stramt.

De forskellige båndoperationer styres ved hjælp af il og us ordrene. Forholdene minder meget om de tilsvarende forhold ved karrusellen.

Det GIER system, der vil blive omtalt her, arbejder med tre identiske båndenheder. Ved hjælp af omskifttere på båndenhederne kan disse tildeles identifikationsnumre, således at det er muligt i programmet at referere til den enkelte båndstation.

Læsning fra og skrivning på et magnetbånd kan kun finde sted medens båndet føres frem. Dette forhold bevirker, at arbejdet med magnetbånd består i skiftevis fremføringer (under hvilke der læses eller skrives på båndet), og tilbagespolinger. Igangsætning og bremsning af magnetbåndet tager forholdsvis lang tid sammenlignet med overførslerne til eller fra bufferen. Det er derfor af afgørende betydning for en effektiv udnyttelse af båndet, at det datamateriale der skal lagres på dette, er ordnet på en hensigtsmæssig måde.

Man har mulighed for at lagre med to forskellige tætheder på et magnetbånd, såkaldt high density og low density. Ved high density lagres med                    tegn pr. tomme, og ved low density lagres med                    tegn pr. tomme. Dette medfører, at ved high density har et enkelt magnetbånd plads til 16 millioner karakterer à 6 bit, eller ca. 2 millioner ord på 42 bit, og ved low density                    millioner karakterer à 6 bit, eller ca.                    millioner ord på 42 bit. Ved hjælp af en trykknop indstilles båndstationen til at arbejde med den ønskede tæthed.

I modsætning til karrussellen arbejdes der her med variabel bloklængde, d.v.s. koderen må i sit program angive hvor mange ord der skal transporteres mellem buffer og magnetbånd. Der kan kun transporteres een blok ad gangen.

På karrussellen havde man kun mulighed for at få skrevet med ulige paritet, d.v.s. at checkbitten i de enkelte tegn altid blev sat således at antallet af 1-bit i hvert tegn er ulige. På konventionelle magnetbåndsenheder har man derimod også mulighed for at få skrevet med såkaldt lige paritet, hvilket altså vil sige at checkbitten bliver sat således at antallet af 1-bit altid er lige. Da der normalt arbejdes med ulige paritet, skal vi ikke her komme nærmere ind på brugen af lige paritet. Dog skal en speciel skrivning med lige paritet omtales. Det drejer sig om det såkaldte filemark. Et filemark er en blok hvor alle tegn er skrevet med lige paritet, og hvor hvert tegn har talværdien 15. Elektronikken i båndstationerne, er indrettet således at båndene kan køres frem til og stoppe ved d.v.s. umiddelbart efter filemark. Ligeledes kan man køre tilbage og stoppe ved d.v.s. umiddelbart før et filemark.

## 20.2 Transporter mellem buffer og magnetbånd.

Som tidligere omtalt kan der tildeles de enkelte båndstationer forskellige numre. Det er disse numre man benytter i koden, når der skal skrives på eller læses fra et magnetbånd. I det følgende går vi ud fra, at vi arbejder med tre båndstationer med numrene 1, 2 og 3.

### 20.2.1 Parameterord.

Udførelsen af de fleste båndordrer styres af et parameterord, der skal være placeret i R-registret inden ordrens udførelse.

Indholdet af parameterordet skal være:

pos. 8 - 19 bloklængden - d.v.s. antallet af ord - i den blok der ønskes transporteret

pos. 28 - 39 blokkens begyndelsesadresse i bufferen.

Indholdet af de øvrige positioner - pos. 0 - 7, pos. 20 - 27 og pos. 40 - 41 - er irrelevante.

### 20.2.2 Statusord.

Ligesom ved karrussellen kan transporterne til og fra magnetbånd testes ved hjælp af et statusord. Angående den sidst udførte transport kan man gennem statusordet få information, om der var paritetsfejl, om den sidste blok var et filemark og endelig om længden af blokken. Disse tre ting kan kun hentes fra båndelektronikken til bufferen een gang og kun hvis der ikke er udført andre operationer på den pågældende båndstation, d.v.s. informationen bliver ødelagt i båndelektronikken så snart den er transporteret til bufferen samt ved en ny båndordre til den pågældende station. Udover denne testinformation indeholder statusordet oplysning om den såkaldte båndstatus. Denne indeholder information om hvor vidt magnetbåndet står i begyndelsen af båndet - load point -, om det er kommet til slutningen - end of tape - og om skrivning er tilladt på båndet. Angående dette sidste bemærkes at der bag på båndspolen kan anbringes en såkaldt skrivering. Såfremt skrivingen er på,

kan der skrives på båndet, i modsat fald er skrivning umuligt. Læsning berøres ikke af dette forhold. Endelig indeholder båndstatus information om hvilken density, der valgt på den pågældende station.

Opbygningen af statusordet fremgår af følgende skema:

position	1- stillet hvis der er registreret:
0	paritetsfejl
1	file-mark
2	load-point
3	end of tape
4	skrivning tilladt
5	high density

Positionerne 7 - 19 indeholder antallet af ord i den skrevne eller læste blok.

Såfremt en eller flere af de forskellige registreringer ikke har fundet sted, 0-stilles de pågældende positioner.

Position 6 og positionerne 20 - 41 er på nuværende tidspunkt ubenyttet, d.v.s. de er altid nulstillede.

### 20.2.3 Adressen i il eller us ordren.

Den resulterende adresse i il eller us ordren skal bestå af summen af to tal, hvoraf det ene angiver hvilken båndstation vi ønsker at benytte, og det andet hvilken funktion vi ønsker at få udført. Følgende tabel forklarer dette nærmere.

Funktion	funktion- adresse	blokering af datakanal	kræver parameterord	il/us
skriv blok med ulige paritet	0	ja	ja	us
læs blok med ulige paritet	0	ja	ja	il
skriv blok med lige paritet	128	ja	ja	us
læs blok med lige paritet	128	ja	ja	il
læs statusord til buffer	224	nej	ja	il
læs statusord til buffer og kør en blok tilbage hvis der er paritetsfejl	160	nej	ja	il
kør en blok tilbage	32	nej	nej	il
skriv blok med ulige paritet efter overspringelse af ca. 5''=12.5 cm. bånd	32	ja	ja	us
skriv blok med lige paritet efter overspringelse af ca. 5''=12.5 cm. bånd	160	ja	ja	us
kør frem til filemark	64 192	nej	nej	il
kør tilbage til filemark	96	nej	nej	il
spol tilbage til load point	64 192	nej	nej	us
spol båndet helt af	96 224	nej	nej	us
venteordre	80	nej	nej	us



Det bemærkes at den resulterende adresse i il eller us ordren skal være lig med summen af funktionsadressen i tabellen og nummeret på den pågældende station. I kolonnen "blokering af datakanal" angiver et "ja", at der ikke kan udføres transporter til eller fra en anden båndstation tilsluttet samme datakanal før de pågældende transporter afsluttede.

### 20.3 Eksempler.

For at vise hvordan transporterne mellem buffer og magnetbånd kan kodes og hvordan disse transporter testes, anføres et par eksempler.

#### Eksempel 20.3.1

Båndet der sidder på station 2 skal køres til load point, og den første blok, der er på 1024 ord, skal læses til bufferens celle 2048 ff. Yderligere skal transporten testes for eventuel paritetsfejl, og i tilfælde af fejl skal transporten gøres om. Et program til dette kan f. eks. være:

```

.
.
ita_paa2           ; etablering af parameterord
usm64             ; kørsel til load point
e: arnal,ilm2     ; 1. blok på station 2 til buffercelle 2048 ff
  arna2,ilm162    ; statusord til buffercelle 4095 og kørsel en
                  ; blok tilbage hvis der er paritetsfejl
  il,arna         ; R:=statusord
  hveLT          ; if paritetsfejl then go to e
.
.
a: qq            ; arbejdscelle for statusord
a1: qqm1024.19+2048.39 ; parameterord
a2: qqm1.19+4095.39  ; parameterord
.
.
.

```

Det bemærkes, at parameterordet til transporten af statusordet til bufferen, kun skal indeholde en bufferadresse i pos. 28 - 39, d.v.s. indholdet af de øvrige positioner - pos. 0 - 27 og 40 - 41 - er irrelevant. Ved denne transport kan derfor godt bruges det parameterord, som anvendes til transporten af statusordet fra bufferen til ferritlageret, da bufferadressen er den samme.

### Eksempel 20.3.2

Bufferens indhold fra celle 2000 til celle 2999 skal skrives på station 3, og transporten skal kontrolleres ved hjælp af statusordet. Efter operationen skal båndet spoles af.

```

.
.
ita,pa2                ; etablering af parameterord
arna1,usm3             ; skriv på station 3 1000 ord fra buffer-
                       ; adresse 2000
e1: arna2,ilm163      ; statusord til buffercelle 4095 og kø
                       ; en blok tilbage hvis der er paritetsfejl
                       ; R:=statusord
il,arna               ; if -, paritetsfejl then go to e
hveNT                 ; skriv på station 3 med overspringelse af
arna1,usm35           ; ca. 12.5 cm. bånd
hve1                  ; go to e1
e:  usm99
.
.
.
a:  qq                ; arbejdscelle for statusord
a1: qqm1000.19+2000.39 ; parameterord
a2: qqm1.19+4095.39   ; parameterord
.
.
.

```

Eksempel 20.3.3

I eksempel 20,3.2 blev der til transporten af statusordet benyttet ordren ilm163. Hvis vi i stedet bruger ordren ilm227, kan programmet se således ud:

```

.
.
ita,pa2                ; etablering af parameterord
arnal,usm3             ; skriv på station 3 1000 ord fra buffer-
                        ; adresse 2000
e1:  arna2,ilm227      ; statusord til buffercelle 4095
     il,arna           ; R:= statusord
     hveNT             ; if -, paritetsfejl then go to e
     ilm35,arna1       ; kør en blok tilbage
     usm35,hve1        ; skriv på station 3 med overspringelse af
                        ; ca. 12.5 cm. bånd, go to e1
e:  .
     .
     .
a:  qq                ; arbejdscelle for statusord
a1: qqm1000.19+2000.39 ; parameterord
a2: qqm1.19+4095.39   ; parameterord
.
.
.

```

Eksempel 20.3.4

Hvis man ved en læsning fra magnetbånd i parameterordet har beordret færre ord transporteret, end der er i blokken, vil læsning af blokken ske på normal måde, altså hele blokken vil blive læst; men kun det i parameterordet beordrede antal ord vil blive overført til bufferen. Hvis man f. eks. kun skal bruge de første 500 ord i en blok der ialt indeholder 1000 ord, behøver man altså kun beordre transport af 500 ord.

```

.
.
arna,ilm2                ; læs fra station 2 500 ord til bufferadresse
                          ; 0 og fremefter
.
.
.
a:  qqm500.19            ; parameterord
.
.

```

Det bemærkes at bloklængden i statusordet i dette tilfælde vil være 1000, altså den faktiske bloklængde på båndet og ikke det antal ord, der er placeret i bufferen.

#### Eksempel 20.3.5

Hvis man beordrer transport fra et bånd til bufferen af flere ord end den aktuelle blok indeholder, vil transporten kun omfatte denne blok, og den øvrige del af bufferen vil være uændret. Også i dette tilfælde vil bloklængden i statusordet være den faktiske bloklængde på båndet.

Hvis vi skal lave et program der skal læse en blok fra station 3, hvor vi ved at blokken er mindre end eller lig med 500 ord, og hente det sidste ord i blokken til en arbejdscelle, kan dette gøres på følgende måde:

Et program, der skal

- 1) læse en blok, der er mindre end eller lig med 500 ord fra station 3,
- 2) hente det sidste ord i blokken til en arbejdscelle,

kan se således ud:

```

.
.
ita,paa1                ; etablering af parameterord
e:  arna2,ilm3          ; læs en blok fra station 3 til bufferadres-
                          ; se 0 (max. 500 ord)
arna1,ilm163           ; læs statusord til buffercelle 4095, og kør
                          ; en blok tilbage, hvis der er paritetsfejl
il,arna                ; R:=statusord
hveLT                  ; if paritetsfejl then go to e
mba3,sra4              ; R:=(bloklængde-1).19
tkm-20,ara4           ; R:=1.19+(bloklængde-1).39
ara5D                  ; R:=R+a5.9
il                     ; transport af sidste celle i blokken til a5
.
.
.
a:  qq                 ; arbejdscelle for statusord
a1:  qqm1.19+4095.39  ; parameterord
a2:  qqm500.19        ; parameterord
a3:  qqm511.19
a4:  qqm1.19
a5:  qq                 ; arbejdscelle for sidste ord i blokken
.
.

```

### Øvelse 20.3.6

Lav et program der læser en blok fra station 1 og skriver de sidste 20 ord i blokken, hvis bloklængden er 300 skal der skrives på station 2 og hvis bloklængden er 400 på station 3. Alle transporter skal efterfølges med test af statusord, og i tilfælde af paritetsfejl skal transporten gøres om. I tilfælde af at bloklængden er forskellig fra både 300 og 400 skal der hoppes til en fejludskriftssekvens.

Eksempel 20.3.7

Der skal læses en blok fra station 2 til bufferadresse 2048. Denne blok skal efter læsning skrives på station 3. Blokken kan enten være en almindelig datablok skrevet med ulige paritet på 500 ord, eller den kan være et filemark, altså en blok skrevet med lige paritet.

```

.
.
ita,paal          ; etablering af parameterord
ita2,paal3        ; etablering af parameterord
e2:  arna4,ilm2    ; læs en blok fra station 2 til bufferadres-
                ; se 2048
                arna1,ilm226    ; hent statusord til celle 4095 i bufferen
                il,pi(a)        ; in:=pos. 0 - 9 i statusordet
                hveLOB          ; if filemark then go to e
                hve1NOA        ; if -, paritetsfejl then go to e1
                ilm34,hve2      ; kør en blok tilbage, go to e2
e1:  arna4,usm3    ; skriv en blok på station 3 fra buffer-
                ; adresse 2048
e4:  arna1,ilm163  ; hent statusord til buffercelle 4095 og
                ; kør en blok tilbage ved paritetsfejl
                il,arna        ; R:=statusord
                hve3NT         ; if -, paritetsfejl then go to e3
                arna4,usm35    ; skriv blok på station 3 efter oversprin-
                ; gelse af ca. 12.5 cm. bånd
                hve4           ; go to e4
e:   arna3,us      ; skriv filemark fra ferritlageret til buf-
                ; feren
                ara2D         ; R:=parameterord
                usm131X       ; skriv filemark på station 3, M:=parameter-
                ; ord
e5:  arna1,ilm163  ; hent statusord til celle 4095 i bufferen
                il,arna        ; R:=statusord
                hve3NT         ; if -, paritetsfejl then go to e3
                xr,usm163     ; skriv filemark efter overspringelse af
                ; ca. 12.5 cm. bånd
                hve5           ; go to e5

```

e3: .  
 .  
 .

a: qq ; arbejdscelle for statusord  
 a1: qqm1.19+4095.39 ; parameterord  
 a2: 3c ; filemark (et filmard er en blok bestående  
 ; af et eller flere tegn med talværdien 15,  
 ; men 3c er et tegn med bitkonfigurationen  
 ; 001111 = 15)  
 a3: qqm1.19+4095.39 ; parameterord  
 a4: qqm500.19+2048.39 ; parameterord  
 .  
 .  
 .

Det bemærkes at test af statusord efter læsning, foregår ved at statusordets pos. 0 - 9 placeres i indikatoren. Dette bevirker at test for de forskellige funktioner kan gøres ved at betinge en ordres udførelse på en indikatorposition. Hvilken pos. der hører til de forskellige funktioner, fremgår af følgende skema:

funktion	Indikatorposition
paritetsfejl	Oa
filemark	Ob
load point	Ta
end of tape	Tb
skrivning tilladt	Pa
high density	Pb

Øvelse 20.3.8

Skriv et program der læser en blok fra station 1. Hvis blokken er på 500 ord skal den skrives på station 2, og hvis den er på 800 ord på station 3. Endelig skal blokken springes over hvis den hverken er på 500 ord eller 800 ord. Dette skal fortsættes indtil der læses et filemark. Dernæst skal skrivningen på station 2 og 3 afsluttes med skrivning af et filemark. Til sidst skal alle båndene spoles af. Alle transporter skal testes ved hjælp af statusordet, og i tilfælde af fejl skal den pågældende transport gøres om.

Eksempel. 20.3.9

I de tilfælde hvor man skal udveksle bånd med en maskine, hvis ordlængde er forskellig fra GIER's, kan det være hensigtsmæssigt at kunne skrive på og læse fra magnetbånd med afkortet ordlængde. Til dette formål kan vi i stedet for at skrive eller læse med de sædvanlige 7 tegns ord, modificere adressen i båndordren, således at operationen kun udføres på de første 4 tegn i ordet altså pos. 0 - 23. Dette gøres ved at addere 16 til adressen i skrive og læseordrene. Ved skrivning skrives altså kun pos. 0 - 23, og de øvrige positioner er irrelevante. Ved læsning læses de 4 første tegn på båndet, og disse tegn placeres i pos. 0 - 23. De øvrige positioner bliver nulstillede.

Hvis man i eksempel 20.3.2 ønskede at arbejde med 4-tegns ord, ville programmet f. eks. blive:

```

.
.
ita, paa2                ; etablering af parameterord
arna1, usm19             ; skriv på station 3 med 4 tegn pr ord 1000
                          ; ord fra bufferadresse 2000
e1:  arna2, ilm163       ; hent statusord til buffercelle 4095 og kør
                          ; en blok tilbage i tilfælde af paritetsfejl
il, arna                 ; R:=statusord
hveNT                    ; if -, paritetsfejl then go to e
arna1, usm51             ; skriv på station 3 med 4 tegn pr ord efter
                          ; overspringelse af ca 12.5 cm. bånd
hve1                     ; go to e1

```



e: usm99 ; spol båndet af  
.  
.  
a: qq ; arbejdscelle for statusord  
a1: qqm1000.19+2000.39 ; parameterord  
a2: qqm1.19+4095.39 ; parameterord  
.  
.  
.

### Øvelse 20.3.10

Skriv et program der kopierer fra men ikke med det tredje filemark til og med det fjerde filemark fra station 2 til station 3. Alle transporter skal testes ved statusordet. Til sidst skal båndet på station 2 spoles af og båndet på station 3 køres til load-point.

### 20.4 Venteordre.

Hvis man vil sikre sig, at en transport mellem magnetbånd og buffer er tilendebragt, inden der fortsættes i programmet, kan man ligesom ved karrusellen anvende en venteordre. Venteordren ved de konventionelle magnetbåndsenheder hedder

us80

hvor der til adressen skal adderes nummeret på den pågældende station. Indholdet af R-registeret er irrelevant, der sker kun det at GIER venter til en eventuel transport mellem bufferen og den station som adressen henviser til er tilendebragt. Overførsel af statusord kan også benyttes som venteordre, men da transport af statusord bruger en celle i bufferen, må man hvis hele bufferen er beslaglagt til data, anvende us80 ordren.

### 20.5 Sikkerhed.

Udover den kontrol der er omtalt i afsnit 20.2.2 under gennemgangen af statusordet, bliver der under skrivning på og læsning fra et bånd dannet et tværgående paritetscheck. Når en blok er skrevet på båndet, indeholder dette en tværgående stribe af karakterer, der hver består af 7 bit, de 6 informationbærende bit og checkbitten. Efter at skrivningen er afsluttet, føres båndet tre pladser frem uden skrivning, hvorefter der skrives en bloksslutkarakter på båndet. Denne karakter er resultatet af det tværgående paritetscheck, og det er indrettet således, at det totale antal 1'er (bloksslut indbefattet) for hvor af de 7 rækker (svarende til de 7 bit en karakter består af) er lige. Ved læsning fra båndet bliver der udover det almindelige paritetscheck tegn for tegn, dannet en tværgående checkkarakter efter samme retningslinier som ved skrivning. Denne karakter sammenlignes med bloksslut karakteren, og i tilfælde af uoverensstemmelse vil der blive registreret paritetsfejl i statusordet.

Ved skrivning på et bånd, bliver hver karakter umiddelbart efter skrivningen læst igen, og der kontrolleres om checkbitten er sat i overensstemmelse med den ønskede paritet. Samtidig undersøges om den tværgående paritet er i orden. Hvis et af disse check viser fejl, vil der blive registreret paritetsfejl i statusordet.

### 20.6 Hastighed.

Den tid det tager at transportere mellem buffer og magnetbånd er forskellige for de forskellige fabrikater. Her skal kun omtales de tider der gælder ved brug af båndenheder af typen CDC 9100.

start af bånd	msec.
overførsel af en karakter ved high density	usec.
overførsel af en karakter ved low density	usec.

Overførselstiderne svarer til tegn i sec ved high density og tegn i sec ved low density. Det bemærkes, at da et ord består af 7 karakterer, skal overførselstiderne ganges med 7 for at få transporttiderne pr. ord.

## 21. Magnetbåndadministration.

### 21.1. Indledning,

Ved arbejdet med magnetbånd, er der især to væsentlige problemer der melder sig.

- 1) hvordan kan man sikre sig et det er de rigtige magnetbånd der anvendes.
- 2) Da det hovedsageligt er de samme ting der skal ske hver gang, hvordan kan man da standardisere brugen af magnetbånd.

Vi vil i dette kapitel forsøge at give løsninger på disse to problemer.

### 21.2 Bånddåb.

Man kan inddele magnetbåndene i to kategorier, sagsbånd og arbejdsbånd. Sagsbånd er bånd, der er reserverede til bestemte opgaver (sager). Sådanne bånd benyttes til opbevaring af information i kortere eller længere tid. Arbejdsbånd benyttes til lagring af data, der kun skal bruges under beregning af opgavens slutresultater og derefter gerne må gå tabt. Disse bånd er derfor - i modsætning til sagsbånd - tilgængelige for alle.

Da et sagsbånd benyttes i forbindelse med en bestemt sag, er det nærliggende at lade båndet få et navn eller et nummer, der på entydig måde henviser til den pågældende sag. I praksis vil der ofte være brug for at løse flere opgaver indenfor samme sag, hvorfor, det er naturligt at lade et løbenummer indgå i navnet på det enkelte bånd. Endelig kan enkelte opgaver være så omfattende, at der skal anvendes flere bånd ved opgavernes behandling. Dette bevirker, at man yderligere arbejder med båndløbenumre.

For på entydig måde at karakterisere et sagsbånd har man da vedtaget, at båndet skal indeholde oplysning om

- 1) sagsnummer
- 2) løbenummer
- 3) båndløbenummer

For at få arbejdsbåndene til at passe i dette system, reserverer man et sagsnummer og et løbenummer til disse bånd. Alle arbejdsbånd vil da have et navn der indeholder dette sagsnummer og løbenummer. For arbejdsbånd vil båndløbenummer være 0.

Hver gang man starter en kørsel med et magnetbånd må man teste, om det er det rigtige bånd der sidder på magnetbåndstationen. Dette test må også udføres hver gang et bånd er kommet i en situation, hvor man kan komme til at skifte det - f. eks. hvis båndet er kørt til load-point.

Man tester ved at undersøge om det aktuelle bånd enten er et sagsbånd med det navn man har benyttet i sit program eller et arbejdsbånd.

Hvis dette test indbygges i et standard administrationsprogram, og alle brugere tvinges til at anvende dette program, er der omtrent 100 % sikkerhed for at det er de rigtige bånd der benyttes.

### 21.3 Administrationsprogram.

For at standardisere brugen af magnetbånd, må vi først gøre os klar, hvad vi ønsker et generelt magnetbåndadministrationsprogram skal udføre. Det første der skal ske, må være en initialisering af selve administrationsprogrammet, der giver oplysning om hvilke bånd der skal sidde på magnetbåndstationerne. Initialiseringen må yderligere sørge for at der arbejdes med rigtige mode på båndet, herved forstås om der skal arbejdes med lige eller ulige paritet og man ønsker at arbejde med 4 eller 7 tegns ord. (Det normale vil være at man arbejder med ulige paritet og med 7 tegns ord).

Som output fra initialiseringen er det rimeligt at man får oplysning om hvilken density der er valgt på den pågældende station, samt om der sidder skrivning på det pågældende bånd.

Dernæst må administrationsprogrammet kunne udføre de båndoperationer, som koderen har brug for, samt udføre de test der er omtalt i afsnit 21.2.

Svarende til de to formål inddeles administrationsprogrammet i to sekvenser:

- 1) initialiseringssekvens
- 2) båndsekvens

Initialiseringssekvensen bruges kun ved starten af et program, eller hvis man under en kørsel ønsker at skifte bånd, hvorimod båndsekvensen skal anvendes hver gang, man ønsker at udføre en båndoperation.

I det følgende vil vi vise hvordan disse to sekvenser kan laves, og hvordan man kan bruge dem.

#### 21.4 Etikette.

Brugen af de to sekvenser vi vil gennemgå, forudsætter at hvert af de magnetbånd der skal benyttes, er forsynet med en etikette, der blandt andet indeholder oplysning om båndets navn.

Den etikette vi her vil arbejde med, består af en blok forrest på båndet (den første blok) bestående af fire 4-tegnsord skrevet med ulige paritet. Første ord består af sagsnummer med enhed i pos. 19. Andet ord består af båndløbenummer med enhed i pos. 5 og løbenummer i pos. 15. Tredie og fjerde ord indeholder information som er uvedkommende for forståelsen af sekvenserne. Opbygningen af første og andet ord giver mulighed for:

sagsnummer  $\leq$  524287

løbenummer  $\leq$  1023

båndløbenummer  $\leq$  63

Når etiketten er skrevet med 4-tegns ord er grunden den, at magnetbånd beregnet for GIER i så fald bekvemt vil kunne benyttes i forbindelse med andre maskiner med en fra GIER afvigende ordlængde. Bl.a. vil man kunne beholde konventionerne for båndnavne.

Læses imidlertid etiketten i GIER på almindelig måde med 7 tegn pr. ord, og forlanges der kun læsning af et ord, vil indholdet af dette ord blive:

sagsnummer med enhed i pos. 19,

båndløbenummer med enhed i pos. 29 og

løbenummer med enhed i pos. 39.

Vi kan altså nøjes med at læse et ord fra den første blok på båndet for at teste, om det er det rigtige bånd vi arbejder med.

21.5 Initialiseringssekvens.

Denne sekvens skal kaldes en gang for hvert bånd, der skal benyttes, og den indsætter de aktuelle data for det pågældende bånd i båndsekvensen. Disse data er

- 1) stationsnummer,
- 2) båndets navn (sagsnummer, løbenummer og båndløbenummer),
- 3) den ønskede mode for det pågældende bånd.

Endelig må kaldet indeholde information om hvor i ferritlageret båndsekvensen ligger.

Værdien af mode fremgår af følgende skema:

mode	4 eller 7 tegn	paritet
0	7	ulige
1	4	ulige
2	7	lige
3	4	lige

Ud over at indsætte de aktuelle data i båndsekvensen, kører initialiseringssekvensen båndet til load-point og tester navnet. Ved udhop fra sekvensen er statusordet for læsning af dåbsblokken placeret i R-registeret.

21.5.1 Kald af initialiseringssekvensen.

Kald af sekvensen foregår på følgende måde:

hs < begyndelsesadresse>

qq < adresse på båndsekvens > t < stationsnummer>

qq < adresse på navn > t < mode>

hvor:

begyndelsesadresse er den ferritlageradresse hvorfra initialiseringssekvensen er lagret,

adresse på båndsekvens er den ferritlageradresse, hvorfra båndsekvensen er lagret,

stationsnummer er nummeret på den aktuelle station,

adresse på navn er adressen på den celle i ferritlageret, der indeholder navnet på det pågældende bånd, hvor navnet skal være pakket med sagsnummer i pos. 19, båndløbenummer i pos. 29 og løbenummer i pos. 39.

mode er den aktuelle værdi man ønsker at benytte på det pågældende bånd. se skema i afsnit 21.5.

Returhop fra sekvensen foregår med ordren

hrs3

### 21.5.2 Arbejdsbånd.

Ved arbejdsbånd benyttes dels et externt navn som koderen bruger og dels et internt navn, som er det faktiske indhold af etiketten på et arbejdsbånd. Anfører koderen det externe navn ved kaldet af initialiseringssekvensen, oversættes dette til det interne navn, og det er dette navn båndsekvensen arbejder med.

Den externe værdi af navnet på et arbejdsbånd er:

sagsnummer = 2047

båndløbenummer = 0

løhenummer = 511

og det er altså dette navn koderen bør anføre, hvis der skal anvendes arbejdsbånd i programmet; men som før bemærket vil et arbejdsbånd altid blive godtaget, selvom man har anført et andet navn.

21.5.3 SLIP-kode til initialiseringssekvens.

```

r
b a1,e2
    arns1,tkm10                ; R:=station.9
    gae2,tkm1                  ; R:=2 × station.9
    aie2,gae1                  ; adr[e1]:=3 × station
    it(s1),pae                 ; adr[e]:=adresse på båndsekvens
    arn(s2),sra                ; R:=navn
    arna1LZV                   ; arbejdsbånd
    ara                        ; R:=navn
e1:  it,itm84                  ; adr[e1]:=3 × station+84
e:   grMt1                     ; adr.[e]:=adresse på båndsekvens+3 × station
                                         ; + 84

    pms2,t1m18
    tkm2,t1m25                 ; R:=paritet.2+tegn.5+station.9
    are2,ude                   ; adr.[e]:=adresse på båndsekvens+3 × station
                                         ; + 85

    nt(e1)-2                   ; adr.[e1]:=3 × station+82
    qq(e)                      ; adr.[e]:=adresse på båndsekvens+3
    hs(e)                      ; R:=båndstatus
e2:  qq,hrs3
a:   qqm2047.19+511.39        ; extern navn for arbejdsbånd
a1:  qqm2047.19+511.39        ; intern navn for arbejdsbånd
e

```

Det bemærkes at vi her arbejder med samme værdi for arbejdsbåndenes interne og eksterne navne.

Da båndsekvensen bruger buffercelle 4095 ved test af navn for til transport af statusord, go da initialiseringssekvensen henter båndstatus ved hjælp af båndsekvensen, vil indholdet af celle 4095 i bufferen være ødelagt ved udhop fra sekvensen.

Sekvensen fylder 18 celler altså fra 0 - 17.



## 21.5.4 Eksempel.

Vi vil vise hvordan initialiseringssekvensen kan anvendes.

Båndet på station 2 har navnet:

```
sagsnummer      = 234,
båndløbenummer  = 5,
løhenummer      = 67.
```

Båndet på station 3 skal være et arbejdsbånd. På station 2 må der ikke være skrivering, hvorimod der på station 3 skal være skrivering. High density på begge stationer. Desuden forudsættes at båndsekvensen er placeret i ferrit-lageret fra celle e og fremefter og initialiseringssekvensen fra celle e1 og fremefter. Endelig skal der arbejdes med ulige paritet og med 7-tegns ord. En fejludskriftssekvens tænkes lagret i celle e2.

```
.
.
hse1                ; Kald af initialiseringssekvens
qget2               ; Båndsekvens i celle e - station 2
qqa                ; Navn i celle a og da tælledele er 0 -
                   ; ulige paritet og 7-tegns ord.
gra1,pi(a1)        ; indikatoren:=båndstatus
hve2LPA            ; if skrivering then go to e2
hve2NPB            ; if low density then go to e2
hse1                ; Kald af initialiseringssekvens
qget3              ; Båndsekvens i celle e - station 3
qqa2               ; Navn i celle a2 - ulige paritet og high
                   ; density
gra1,pi(a1)        ; indikatoren:=båndstatus
hve2NPA            ; if -, skrivering then go to e2
hve2NPB            ; if low density then go to e2
.
```

.  
 .  
 a: qqm234.19+5.29+67.39 ; Navn for båndet på station 2  
 a1: qq ; Arbejdscelle  
 a2: qqm2047.19+511.39 ; Navn for arbejdsbånd  
 .  
 .

Under gennemløbet af disse ordrer, er informationen om magnetbåndene på station 2 og 3 blevet lagret i båndsekvensen. Båndene er blevet kørt til loadpoint, hvorefter det er blevet undersøgt ved hjælp af det test, der er indbygget i båndsekvensen, om det er de rigtige bånd eller arbejdsbånd, der er sat på stationerne. Båndene står derefter ikke i loadpoint, men umiddelbart efter dåbsblokken. Angående fejlreaktion i tilfælde af forkerte bånd henvises til afsnit 21.6

#### 21.6 Båndsekvens.

I båndsekvensen er der indhop for hver af de båndoperationer, som koderen har brug for. Desuden er der et særligt testindhop. Ved indhop hertil vil den sidst udførte båndoperation blive testet for paritetsfejl og statusordet afleveres i R-registeret.

Med disse indhop kunne brugeren af magnetbånd egentlig klare sig, men da enhver skrive- eller læseoperation skal testes for paritetsfejl har man suppleret hver skrive- og læseoperation med to indhop. Et hvor operationen testes umiddelbart efter udførelsen og et, hvor man lige før operationens udførelse tester den transport, der sidst blev udført på samme båndstation. Dette sidste indhop er lavet med henblik på simultankørsel på båndstationerne, idet man da skal udskyde testet af enhver transport så længe som muligt, d.v.s. til umiddelbart før den pågældende båndstation næste gang skal benyttes.

I tilfælde af at et test viser at der har været paritetsfejl i den pågældende transport, gøres denne om indtil der er foretaget en fejlfri transport.

Ved skrivning skrives med overspringselse af ca 12,5 cm bånd

Ud over at få udført transportordrerne kan man ved hjælp af båndsekvensen skrive filemark samt få udført de almindelige kørselsordrer: kør en blok tilbage, kør frem eller kør tilbage til filemark, kør til loadpoint eller spol båndet af.

Ved ethvert indhop testes, om det pågældende bånd står i loadpoint. Er dette tilfældet, vil der blive testet, om båndets navn er i overensstemmelse med det navn, der er givet gennem initialiseringen, eller om båndet er et arbejdsbånd. Hvis dette test ikke giver fejl, kan almindelig kørsel fortsætte. Hvis der imidlertid sidder et forkert bånd på, spoles båndet af og GIER stopper (zq). Ved fornyet start, efter at der er blevet sat et andet bånd på (og forhåbentlig det rigtige bånd), testes igen. Dette fortsætter indtil testet ikke viser fejl.

#### 21.6.1 Parameterord.

Transportordrerne kræver på normal måde et parameterord der skal indeholde bloklængde med enhed i pos. 19 og begyndelsesadressen på det bufferafsnit, som benyttes med enhed i pos. 39. Dette parameterord skal være anbragt i R-registeret ved indhop til båndsekvensen.

#### 21.6.2 Statusord.

For de indhop der medfører test for paritetsfejl, og dermed transport af statusord, vil statusordet ved returhop være placeret i R-registeret.

Til transporten af statusordet benyttes celle 4095 i bufferen. D.v.s. at denne celle ikke umiddelbart kan anvendes til lagring af data.

Hvis man henter statusord mere end een gang for samme transport, får kun hele statusordet første gang. De følgende gange får man ikke information om paritetsfejl, om båndet står ved et filemark og om bloklængden; men man får kun båndstatus.

#### 21.6.3 Kald af båndsekvens.

Kald af sekvensen foregår på følgende måde:

hs <begyndelsesadresse+funktionsadresse>

qq <stationsnummer>

hvor:

begyndelsesadresse er adressen i ferritlageret hvorfra båndsekvensen er lagret,

funktionsadresse er den relative adresse i sekvensen, der bestemmer hvilket funktionsindhop man ønsker at udføre, og

stationsnummer er nummeret på den station man ønsker at køre på.

#### 21.6.4 Funktionsadresser i båndsekvens.

Værdien af funktionsadresse ved kaldet af båndsekvensen fremgår af følgende skema:

funktions adresse	funktion
0	fra bånd med efterfølgende test
1	fra bånd
2	test af sidste transport plus fra bånd
3	test af sidste transport
4	til bånd med efterfølgende test
5	til bånd
6	test af sidste transport plus til bånd
7	skriv et filemark med efterfølgende test
8	skriv et filemark
9	test af sidst transport plus skriv et filemark
12	kør en blok tilbage
13	kør tilbage til filemark
14	kør frem til filemark
16	kør til loadpoint
17	spol båndet helt af

Returhop fra sekvensen foregår med ordren

Ved test af sidst transport forstås test af den sidste transport på den pågældende station som stationsnummer henviser til

### 21.6.5 SLIP-kode til båndsekvens.

```

R
m
b a,b2,d7,e29
e29: xr,hhne3           ; adr. 0 il+test
e9:  xr,hhne4           ; adr. 1 il
    xr,hhne2           ; adr. 2 test + il
    xr,hvne6           ; adr. 3 test
    xr,hvne4           ; adr. 4 us+test
    xr,hvne5           ; adr. 5 us
    xr,hvne3           ; adr. 6 test+us
    xr,hvne            ; adr. 7 filemark+test
    xr,hhne            ; adr. 8 filemark
    xr,arnd2           ; adr. 9 test+filemark, R:=4
e:  ard2,ard3         ; R:=R+4+1
    tkm-1,hve1        ; R:=R/2
    itm-24            ; adr. 12 kør en blok tilbage
    pae11t-23VX       ; adr. 13 kør tilbage til filemark
    pae11t-21X       ; adr. 14 kør frem til filemark
    arnd2,hve6        ; R:=4
    itm-22           ; adr. 16 kør til loadpoint
    pae11t-20X       ; adr. 17 spol båndet af
    arnd2,hve6        ; R:=4
b1: hse16LRB         ; if RB then filemark
    il(e24)LRAV       ; if RA then læs
    us(e24)           ; else skriv
    gr(e9)XMRC        ; gem parameterord
    hve21NQB         ; if -,QB then go to e21
e7: hse25            ; hent statusord
b2: hve17NOA         ; if -, paritetsfejl then go to e17
    hve18LOB         ; if filemark then go to e18
    arn(e9),ilp32    ; kør en blok tilbage
    hse19LB          ; if filemark then go to e19
    us(e24)32NAV     ; if -, NA then skriv med overspringelse

```

```

        il(e24)V                ;           else læs
        qq(e24)-32NB
e17h:
e18h: hve7,arnd1                ; R:=statusord
        hvb1LPBX                ; if PB then go to b1
        hve21,ilp32            ; kør en blok tilbage
        hve21,ilp96            ; kør tilbage til filemark
        hve21,usp64            ; kør til loadpoint
        hve21,ilp64            ; kør frem til filemark
        hve21,usp96            ; spol båndet af
e21:  pp,pi(e29)
e2:   hrs1,arnd3                ; R:=1
e3:   ard3,ard3                ; R:=R+1+1
e4:   ard3,ard3                ; R:=R+1+1
e1:
e5:   ard2,tkm1                ; R:=(R+4) × 2
e6:   gae28,gpe21
        ppn(s1),gie29          ; p:=stationsnummer
e22:  arpD                      ; R:=stationsnummer
        tkm1,ude22            ; R:=3 × stationsnummer
e28:  pi,ude27                 ; R:=d6+3 × stationsnummer
        gae9,arn(e9)          ; R:=navn
        hve10NB                ; if -,B then go to e10
        hse25                  ; hent statusord
        hve10LTA              ; if loadpoint then go to e10
e15:  arn(e9)1
e24:  gaD                      ;adr [e24] :=kørselsparameter
        qqn(e9)1
        hvb1LQAX              ; if QA then go to b1
        hvb2NQB               ; if -,QB then go to b2
e10h:
e11:  hhr,usp64                ; kør til loadpoint
e12:  arnd4,ilp
        araD                   ; c(a):=første ord første blok
        usp80,il
        hse25                  ; hent statusord
        hhe10LOA              ; if paritetsfejl then go to e10
        arna,sr(e9)

```

```

hve13LZ                ; if navn OK then go to e13
arna,srd5
hve14NZ                ; if -, arbejdsbånd then go to e14
e13: acn(e9)MB         ; navn OK
e14h: hve15,usp96     ; spol båndet af
e8:  zq,hhe10         ; stop
e25: arnd1D
ard4,ilp224
il,itm31               ; d1:=statusord
pi(d1),hrs1           ; in [pos.0-4]:=statusord [pos.0-4]
e16: arnd4,itm128
e19: pae20t160
e27: ard6D
us,arnd4               ; skriv filemark evt. med overspringelse
e20: usp,hrs3
[arbejdsceller]
d2:  qqm4
d3:  qqm1
d4:  qqfm1.19+4095.39
d5:  qqm2047.19+511.39 ; navn for arbejdsbånd
d6:  3c                ; filemark
d1:  qq                ; arbejdscelle for statusord
a:   qq
e

```

Denne del af sekvensen fylder 87 celler. Desuden anvendes 3 arbejdsceller for hver station. Disse bruges til følgende:

```

d6+3 × stationsnummer:   navn
d6+3 × stationsnummer+1: kørselsparameter
d6+3 × stationsnummer+2: gammelt parameterord

```

Det antal celler der skal reserveres til hele sekvensen bliver således:  
 $87+3 \times \text{det maximale stationsnummer}$ , altså fra 0 -  $86+3 \times \text{det maximale stationsnummer}$ .

21.6.6 Filemark.

Hvis en læsning testes, og det viser sig at den pågældende blok var et filemark, vil R-registeret, der indeholder statusordet ved udhoppet, være negativt, men statusordet vil være korrekt. D.v.s. man kan i programmet teste for filemark ved at spørge om R-registeret er negativt.

Ved skrivning af filemark kræves ikke noget parameterord. Transport af filemark styres af sekvensen, og det er celle 4095 i bufferen, der benyttes. Sekvensen skriver filemark som eet tegn med talværdi 15.

21.7 Eksempler.

For at vise hvordan administrationsprogrammet kan anvendes, skal her anføres nogle eksempler.

Eksempel 21.7.1

Båndet på station 2 har navnet:

sagsnummer = 98  
 båndløbenummer = 7  
 løbenummer = 6

og båndet skal have skrivering. På station 3 sidder et bånd med navnet:

sagsnummer = 98  
 båndløbenummer = 0  
 løbenummer = 6

og på dette bånd må der ikke være skrivering. High density på begge stationer. Båndsekvensen er lagret fra celle e og initialiseringssekvensen fra celle e1 i ferritlageret. Desuden er der lagret en fejlsekvens fra celle e2. Der skal arbejdes med ulige paritet og med 7-tegns ord.



Efter initialiseringen skal vi læse fra station 3 og skrive det læste på station 2. Dette skal fortsætte indtil vi læser et filemark, og der skal så skrives et filemark på station 2. Begge bånd skal til sidst spoles af. Blok-længden på station 3 er 2000 ord.

```

.
.
hse1 ; kald af initialiseringssekvens
qqot2 ; båndsekvens i celle e - station 2
qqa ; navn i celle a - ulige paritet og 7-tegns
; ord
gra1,pi(a1) ; indikator:=båndstatus
hve2NPA ; if → skrivering then go to e2
hve2NPB ; if low density then go to e2
hse1 ; kald af initialiseringssekvens
qqet3 ; båndsekvens i celle e - station 3
qqa2 ; navn i a2 - ulige paritet og 7-tegns ord
gra1,pi(a1) ; indikator:=båndstatus
hve2LPA ; if skrivering then go to e2
hve2NPB ; if low density then go to e2
.
.
.
e4: arna3 ; R:=parameterord
hse ; il+test
qqm3 ; station 3
hve3LT ; if filemark then go to e3
arna3 ; R:=parameterord
hs4e ; us+test
qqm2,hve4 ; station 2, go to e4
e3: hs7e ; filemark+test
qqm2 ; station 2
hs17e ; spol båndet af
qqm2 ; station 2
hs17e ; spol båndet af
qqm3 ; station 3
.

```

```

.
.
a:   qqm98.19+7.29+6.39      ; navn for båndet på station 2
a1:  qq                      ; arbejdscelle
a2:  qqm98.19+6.39          ; navn for båndet på station 3
a3:  qqm2000.19             ; parameterord
.
.
.

```

Da det maksimale stationsnummer i dette tilfælde er 3, fylder båndsekvensen  $87+3 \times 3 = 96$  celler i ferritlageret fra celle e til celle e+95.

#### Eksempel 21.7.2

På station 2 sidder et bånd, som vi ønsker at få trykt ud. Båndsekvens i celle e, initialiseringssekvens i celle e1, og i celle e2 tankes lagret en udskriftssekvens, der kræver en programparameter med information om, hvorfra i bufferen, der skal udskrives. Vi ønsker at arbejde i to bufferafsnit, således at vi transporterer til det ene medens vi udskriver fra det andet. Blokkene er på 124 ord. Båndet på station 2 har navnet:

43.19+2.29+1.39

Datablokkene på båndet er afsluttet med et filemark, og når dette filemark er læst, skal båndet spoles af. Efter dette filemark er der yderligere en blok på båndet, f.eks. et filemark.

```

.
.
.
hve1                      ; initialisering
qqet2                     ; båndsekvens i e - station 2
qqa                       ; navn i e
pae3,ama1                 ; etablering af talling, R:=parameterord
hs1e                      ; fra bånd (uden test)
qqm2                      ; station 2
pma1,gma3                 ; a3:=parameterord

```

```

e3:  btt-512
      pma1V                ; M:=nyt parameterord
      pma2
      arna3,gma3           ; R:=gl. parameterord, a3:=nyt parameterord
      grn4X                ; a4:=gl. parameterord. R:=nyt parameterord
      hs2e                 ; test + il
      qqm2                 ; station 2
      hve4LT               ; if filemark then go to e4
      hsc2                 ; udskrift
      qqa4                 ; henvisning til parameterord
      hve3                 ; go to e3
e4:  hs17e                ; spol båndet af
      qqm2                 ; station 2
      .
      .
      .
a:   qqm43.19+2.29+1.39   ; navn på båndet på station 2
a1:  qqm1024.19           ; parameterord 1
a2:  qqm1024.19+1024.39  ; parameterord 2
a3:  qq                   ; arbejdscelle for nyt parameterord
a4:  qq                   ; arbejdscelle for gl. parameterord

```

Da det maximale stationsnummer er 2 fylder båndsekvensen  $87+3 \times 2 = 93$  celler fra celle e til celle e+92.

## 22. Disk-file.

### 22.1 Indledning.

Det karakteristiske træk ved magnetbåndene er, at læsning fra eller skrivning på disse skal foregå, medens båndene bliver ført i fremadgående retning. Såfremt man - hvad der i praksis ofte er tilfældet - skal bruge sine data i en anden rækkefølge end den, i hvilken de forekommer på båndene, er det uundgåeligt, at maskinen må vente i kortere eller længere tidsrum, medens båndene gennemløber lange stykker med data, indtil de aktuelle data bliver fundet frem. I andre lagertyper f.eks. i tromlelageret, har man mulighed for at gå mere direkte til de data, man ønsker at arbejde med. Tromlelageret i GIER er imidlertid af stærkt begrænset kapacitet, men der findes en nyere lagertype, en såkaldt disk-file, som på een gang imødekommer behovet for stor kapacitet og behovet for direkte acces.

Den disk-file model, vi her skal omtale, benævnes ANELEX model 80. Det vil være muligt at tilslutte andre modeller til GIER. Bortset fra de mere teknisk betonedede afvigelser mellem forskellige disk-file modeller, vil disse formodentlig minde så meget om hinanden, at den følgende beskrivelse vil være meget generel.

Selve lageret i en disk-file består af en såkaldt disk-kit. En disk-kit er en kasette, der indeholder 6 plader. Hver plade består af en ca. 1 1/4 mm tyk cirkulær aluminiumskive med en diameter på ca. 35 cm. Skiverne er på begge sider belagt med et magnetiserbart materiale. Skiverne roterer om en fælles akse med en hastighed af 2400 omdrejninger pr. minut, hvilket svarer til en omdrejning på ca. 25 msec. Overførsel til eller fra pladerne foregår gennem et antal læse- og skrivehoveder, eet for hver af de 12 overflader. Læse- og skrivehovederne er sammenkoblede og kan bevæget under eet hen over overfladerne.

For anvendelsen af disk-filen er det i første omgang uden betydning, at denne består af et antal plader. Vigtigere er det, at hele lageret er opdelt i 10 grupper nummereret fra 0 til 9. Hver gruppe er igen inddelt i 960 sektioner nummereret fra 0 til 959. En sektion kan ligesom en tromlekanal rumme 40 GIER-ord. Kapaciteten af en disk-file er således:

$$10 \times 960 \times 40 \text{ ord} = 384000 \text{ ord.}$$

### 22.2 Transporter mellem GIER og disk-file.

I GIER med disk-file indgår denne i stedet for tromlelageret. Transporter mellem GIER og disk-file foregår ligesom ved tromlen direkte til eller fra ferritlageret med 40 ord ad gangen - og altså ikke som ved magnetbåndene gennem bufferen. Til transporterne benyttes de samme ordre - vk, lk og sk - som blev benyttet til tromletransporterne. Da der overføres 40 ord ad gangen, kan de programmer, der er udarbejdet til tromlelager, stadig benyttes på en GIER med disk-file.

Kodning af disk-file transporter kræver normalt i alt tre ordre:

1. Valg af gruppe (i alt 10 muligheder)
2. Valg af sektion inden for gruppen  
(i alt 960 muligheder)
3. Selve transportordren.

Operationsdelene i de tre ordrer er de samme, som benyttes i forbindelse med tromletransporterne, d.v.s. at de to førstnævnte er vk-ordrer, og den tredje er enten en lk-ordre eller en sk-ordre.

Vi bemærker, at GIER's ordrestruktur, hvor den resulterende adresse skal være mindre end 1024, gør det naturligt at arbejde med to vk-ordrer, når der skal vælges mellem 9600 forskellige sektioner.

#### 22.2.1 Valg af gruppe.

De 10 grupper er nummereret fra 0 til 9. Valg af en bestemt gruppe sker ved hjælp af en vk-ordre, hvis resulterende adresse er lig med gruppenummeret plus 960. Ønskes gruppe 3 valgt, kodes der således:

vk963.

Ved udførelsen af et sådant valg af gruppeordre sker der det, at nummeret på gruppen indsættes i et grupperegister, hvor det bliver stående, indtil en ny gruppe bliver valgt. Desuden placeres den resulterende adresse - altså 960 plus gruppenummeret - i tk-registeret.

#### 22.2.2 Valg af sektion.

I hver gruppe findes 960 sektioner nummereret fra 0 til 959. Efter valg af gruppe er foretaget, sker valg af sektion ved en ny vk-ordre, hvis resulterende adresse er lig med nummeret på den pågældende sektion. Ønskes sektion 507 i gruppe 8 benyttet i forbindelse med en transport, kodes der da således:

vk968, vk507.

Sektion 507 er derefter valgt, indtil der er udført en ny vk-ordre med en resulterende adresse mindre end 960.

#### 22.2.3 Læse- og skriveordrerne.

Efter valg af gruppe og sektion foregår læsning og skrivning på samme måde som ved tromlen ved hjælp af lk- og sk-ordrerne. Den resulterende adresse angiver begyndelsesadressen på de 40 celler i ferritlageret, som der læses til eller skrives fra.

Det bemærkes, at læse-skrivehovederne ikke flyttes ved valg af ny gruppe eller sektion; først ved udførelsen af lk- eller sk-ordren sker flytning til det aktuelle spor.

Ligesom ved tromlen kan transporterne mellem ferritlageret og disk-filen foregå simultant med andre operationer i GIER. Det gælder her, at ingen af ordrerne vk, lk eller sk kan træde i funktion, så længe der er en transport i gang mellem disk-filen og ferritlageret. Derfor kan vk-ordren benyttes som venteordre, idet den sikrer, at en igangværende transport føres til ende, før det implicerede område i ferritlageret tages i brug.

Det bemærkes, at man ikke må anføre en vk-ordre, hvis resulterende adresse ikke svarer til en eksisterende gruppe eller sektion. Man må altså ikke som venteordre anføre ordren:

vk-1.

Eksempel 22.2.4

Ønsker vi fra gruppe 2 først at læse fra sektion 16 til ferritlageret i celle 500 ff og dernæst at skrive dette i samme gruppe, men på sektion 31, kan dette gøres ved hjælp af ordrene:

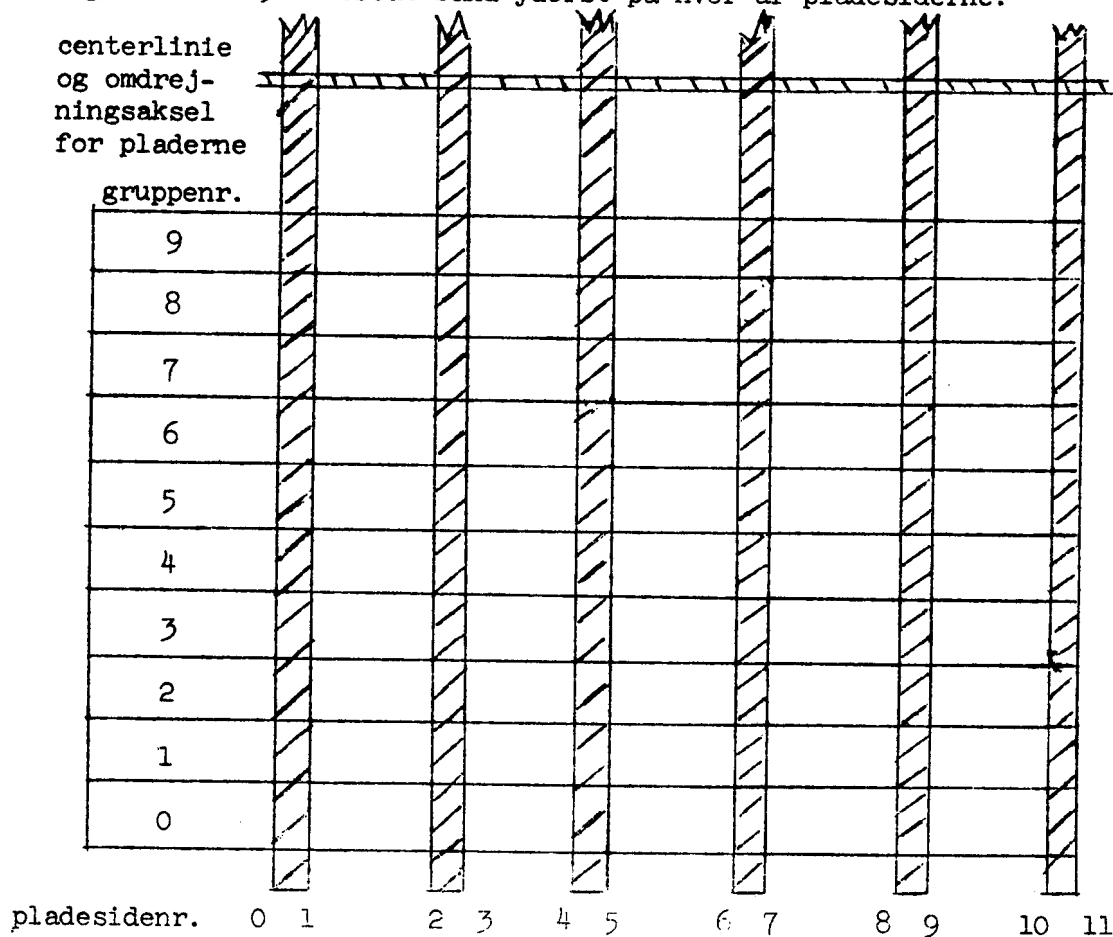
vkm962	; vælg gruppe 2
vkml6	; vælg sektion 16
lkm500	; læs til celle 500 ff
vkm31	; vælg sektion 31
skm500	; skriv fra celle 500 ff
vkm31	; vent.

22.2.5 Disk-filens opbygning.

Programmering i forbindelse med brug af disk-file er således uhyre simpel, og i princippet behøver brugeren kun at vide det, vi indtil nu har omtalt for at kunne anvende en disk-file, der er tilsluttet GIER. For at udnytte en disk-file på en effektiv måde må man dog vide, hvordan de enkelte sektioner er placeret på pladerne. Forholdet er jo det, at en disk-file sædvanligvis kun benyttes, når der er tale om store mængder af data, hvoraf igen følger, at der i praksis vil skulle foregå et stort antal transporter. Af afgørende betydning bliver det da, at den tid, der benyttes til opsøgning af næste sektion, bliver så lille som mulig. Vi vil derfor give en mere detaljeret beskrivelse af disk-filens opbygning.

Hver side af pladen er inddelt i 10 grupper, der hver består af 10 spor. Hvert spor er igen inddelt i 8 sektioner, hvor hver sektion - svarende til en tromlekanal - kan rumme 40 GIER-ord.

De 12 pladesider kan vi tildele numrene fra 0 til 11 og de 10 grupper numrene fra 0 til 9. Gruppernes placering på de 6 plader fremgår af følgende tegning, som dog er stærkt forenklet, idet de 10 grupper kun udgør et ca. 5 cm bredt bånd yderst på hver af pladesiderne.



På hver pladeside består en gruppe af 10 spor, som vi kan tildele numrene fra 0 til 9. De 10 spor er placeret således, at spor med samme nummer ligger ud for hinanden på de 12 forskellige pladesider. Hvert spor består af 8 sektioner, der ligger efter hinanden rundt på pladesiden. Indenfor hvert spor er sektionerne fortløbende nummereret, således at samtlige sektioner i en gruppe er nummereret fra 0 til 959. Denne nummerering er foretaget på følgende måde:

På pladeside nummer 0 i spor nummer 0 ligger sektion 0 til 7 og på pladeside nummer 1 i samme spor sektion 8 til 15. På pladeside nummer 2 og stadigvæk i spor nummer 0 ligger sektion 16 til 23 og på pladeside nummer 3 sektion 24 til 31 o.s.v. Vi har altså på de 12 pladesider 96 sektioner i hvert spor. I spor nummer 0 er sektionerne nummereret fra 0 til 95, i spor nummer 1 fra 96 til 191 o.s.v. til spor nummer 9, hvor sektionsnumrene er fra 864 til 959. D.v.s. at i spor nummer  $s$  ligger sektionerne fra  $96 \times s$  til  $96 \times s + 95$ , og på pladeside nummer  $p$  i spor nummer  $s$  ligger sektionerne fra  $96 \times s + 8 \times p$  til  $96 \times s + 8 \times p + 7$ .

Sektionernes placering indenfor en gruppe fremgår af følgende tegning:

Spør nr.	Sektionsnumre											
9	864	872	880	888	896	904	912	920	928	936	944	952
	-871	-879	-887	-895	-903	-911	-919	-927	-935	-943	-951	-959
8	768	776	784	792	800	808	816	824	832	840	848	856
	-775	-783	-791	-799	-807	-815	-823	-831	-839	-847	-855	-863
.												
1	96	104	112	120	128	136	144	152	160	168	176	184
	-103	-111	-119	-127	-135	-143	-151	-159	-167	-175	-183	-191
0	0	8	16	24	32	40	48	56	64	72	80	88
	-7	-15	-23	-31	-39	-47	-55	-63	-71	-79	-87	-95
pladeside nummer	0	1	2	3	4	5	6	7	8	9	10	11

### 22.3 Effektiv udnyttelse af disk-file.

Den valgte nummerering af sektionerne i en gruppe hænger naturligvis sammen med den måde, hvorpå disk-filen er bygget. Det gælder her, at:

1. læsning af eller skrivning på samtlige 96 sektioner med samme spornummer i samme gruppe kan foregå uden flytning af disk-filens læse- og skrivehoveder, og

2. placering af 8 sektioner på hver af pladesiderne med fortløbende nummerering fra side til side for samme spornummer gør det fordelagtigt at lagre sine data på disk-filen i overensstemmelse med følgende princip: Data, der skal benyttes umiddelbart efter hinanden, bør såvidt muligt lagres i samme spor og i sektioner med fortløbende numre.

### 22.3.1 Flytning af læse-skrivehovederne.

Til nærmere beskrivelse af transporterne mellem GIER og en disk-file vil vi her vise et simpelt eksempel:

#### Eksempel 22.3.1.1

Sektion 115 og sektion 116 i gruppe 7 ønskes læst til celle 100 - 179 i ferritlageret. Dette kan f.eks. gøres på følgende måde:

vkm967, vkm115	; vælg gruppe 7 og vælg sektion 115
lkm100, vkm116	; læs til celle 100 og vælg sektion 116
lkm140, vkm116	; læs til celle 140 og vent.

Den første af disse ordrer bevirker, at den resulterende adresse - 967 - indsættes i tk-registret. Samtidig gemmes gruppenummeret - 7 - i et grupperegister. I den følgende ordre vælges sektion 115. Da adressen er større end 95 og mindre end 192, er det spor nummer 1, der skal arbejdes på. Sektionsnummeret gemmes i tk-registret. På dette tidspunkt er læse-skrivehovederne endnu ikke blevet flyttet. Dette sker først i den næste ordre, hvor læsning bliver udført.

Bevægelsen af læse-skrivehovederne tager ca. 85 msec. uafhængig af, hvor langt de skal flyttes.

Læsning af sektion 115 - ordren lkm100 - begynder, så snart den første celle i sektionen passerer læsehovedet. Til denne opsøgning går der gennemsnitlig ca. 12.5 msec. - tiden for en halv omdrejning.

Mellem to på hinanden følgende sektioner på et spor er der et mellemrum, som det tager ca. 0.2 msec. at passere. Da der er 8 sektioner for hver pladeside i et bestemt spor, tager det altså ca.  $(25/8 \cdot 0.2)$  msec. = ca. 2.925 msec. at læse eller skrive en sektion.

Når læsningen af sektion 115 er tilendebragt, vælges sektion 116 - vkm116 - og læsningen starter umiddelbart efter. Da der i de 0.2 msec., det tager at passere mellemrummet mellem to sektioner, kan nå at blive udført et par ordrer, er der altså ingen fare for, at den første celle i sektion 116 har passeret læsehovedet, før ordren om læsning - lkm140 - er afgivet. Det tager altså kun ca. 3 msec. at læse sektion 116 til ferritlageret regnet fra det tidspunkt, hvor transporten af sektion 115 er afsluttet.

Vi har:

1. Der går ca. 85 msec. med at opsøge et nyt spor, selv om der også skal skiftes gruppe,



2. opsøgning og læsning af - eller skrivning på - en sektion i samme spor, som læse-skrivehovederne er indstillet på, tager ca. 15.5 msec., og
3. efter læsning eller skrivning af den første af en række sektioner med fortløbende numre tager det ca. 3 msec. at læse eller skrive de følgende, såfremt der ikke skiftes spor. Det samme gælder for sektioner, hvis numre vokser i spring på  $8 \times n + 1$ , hvor  $n$  kan være både positiv eller negativ.

Det bemærkes, at den tid, der går til opsøgning af nyt spor samt til læsning eller skrivning, kan udnyttes til simultanregning i GIER.

#### Bemærkning 22.3.1.2.

Skal man transportere flere sektioner, bør man gøre det således, at det samlede antal sektioner, som disk-filen blindt skal køre forbi, bliver mindst muligt. Hvis man således, efter at den aktuelle gruppe er valgt, ønsker at læse sektion 136 og sektion 151 til ferritlageret, bør man - da sektion 136 er den første sektion i spor 1 på pladeside nummer 5, og da sektion 151 er den sidste og dermed den ottende sektion i spor 1 på pladeside nummer 6 - først læse sektion 151 og dernæst sektion 136 til ferritlageret. Gør man det, vil disk-filen ikke komme til at køre blindt forbi nogle sektioner. Hvis man derimod først læser sektion 136 og dernæst sektion 151, vil man komme til at køre blindt forbi 6 sektioner.

#### Eksempel 22.3.2

Vi skal udarbejde et program, der læser 6 sektioner fra gruppe 2 sektion 27 og fremefter, og skrive dem på 6 sektioner fra gruppe 4 sektion 47 og fremefter.

```

.
.
.
vkm962           ; vælg gruppe 2
paet26           ; adr(e):=26
ita-40,pael      ; adr(e1):=a-40
e2: ppm6,ppp-1
e: vktl          ; læs sektion 27 - 32
el: lmt40        ; til celle a ff
    bsp,hhe2
    vkm964       ; vælg gruppe 4
    pae3t46      ; adr(e3):=46
    ita-40,pae4  ; adr(e4):=a-40
e5: ppm6,ppp-1
e3: vktl         ; skriv på sektion 47 - 52
e4: skt40        ; fra celle a ff
    bsp,hhe5
    vkm52        ; venteordre
.
.
.
a: qq           ; ferritlagerplads til transporterne
.
.
.

```

Det bemærkes, at disk-filen efter transporten af første sektion ikke vil køre blindt forbi nogle sektioner. Hvis vi derimod skriver programmet som vist nedenfor, vil disk-filen mellem hver transport køre blindt forbi 7 sektioner.

```

.
.
.
vkm962,it240a      ; vælg gruppe 2
pae,ppm6           ; adr(e):=a+240
el: ppp-1,vkp27    ; læs sektion 32 - 27
e:  lkt-40         ; til celle a+200,a+160,...,a
    bsp,hvel
    vkm964,it240a  ; vælg gruppe 4
    pae2,ppm6      ; adr(e2):=a+240
e3: ppp-1,vkp47    ; skriv på sektion 52 - 47
e2: skt-40         ; fra celle a+200,a+160,...,a
    bsp,hve3
    vkm47          ; venteordre
.
.
a:  qq            ; ferritlagerplads til transporterne
.
.

```

#### 22.4 Gemning af grupperegisteret.

I forbindelse med grupperegisteret ved disk-filen er der indført en ny ordre i GIER, som ikke tidligere er omtalt i de to første bind af lærebogen. Denne ordre, som hedder gg (gem gruppe), gemmer den sidst valgte gruppe plus 960 i adressepositionerne i den celle, som den resulterende adresse i ordren henviser til. Denne ordre har talværdien 46.

##### Eksempel 22.4.1

Hvis vi ønsker at gemme den sidst valgte gruppe for senere at kunne vælge den igen, kan dette f.eks. gøres ved ordrene:

```

.
.
.
gge                ; adr(e):=den sidst valgte gruppe
.
.
e:  vk            ; valg af samme gruppe igen

```

##### Bemærkning 22.4.2

Da den sidst valgte sektion står i tk-registeret, kan dette sektionsnummer gemmes ved hjælp af gk-ordren. Det bemærkes dog, at det - såfremt der har været et gruppevalg siden sidste sektionsvalg - ikke er sektionsnummeret, der står i tk-registeret, men 960 plus den sidst valgte gruppe.