

EDB - SENTRET

SYSTEMGENERERING FOR MULTITERM-10

AV

SIGURD SJURSEN OG OLA M. JOHNSEN

EDB-RAPPORT 85.MT-10.1

VERSJON 78-03-10

UNIVERSITETET I TROMSØ

INNHOOLD

Innledning	1
1. Generelt	2
2. Generering av MT-10	2
2.1. MULTITERM-modulen	3
2.2. EMULATOR-modulen	3
2.3. BUFFER-POOL modulen	4
3. Systemparametre	8
3.1. Navnekonvensjoner	8
3.2. MULTITERM-modulen	8
3.2.1. Generering av kontrollblokker	9
3.2.2. Installasjonsavhengige parametre	10
3.3. EMULATOR-modulen	11
3.3.1. Generering av kontrollblokker	11
3.3.2. Installasjonsavhengige parametre	12
3.3.3. Antall DCT1000 terminaler	12
3.4. BUFFER-POOL modulen	13
4. Lasting av MT-10	14
5. Innlogging	14
APPENDIKS A	16

Innledning.

Denne rapporten beskriver hvordan MULTITERM-10 systemet (MT-10) skal genereres. Genereringen vil nok kreve en viss kjennskap til hvordan systemet fungerer. Desverre foreligger ennå ikke fullstendig systemdokumentasjon. Denne beskrivelsen må derfor betraktes som foreløpig.

Det foreligger en foreløpig versjon av en brukerhåndbok, "MULTITERM USER'S GUIDE". Denne beskriver systemet for brukerne og gir noen eksempler fra kjøring med MT-10.

1. Generelt

MT-10 leveres som en samling symbolske filer på floppydisk. Filene inneholder symbolske programmer og MODE-filer for systemgenerering og programvedlikehold. Det er dessuten en del tomme symbolske hjelpefiler og tomme BRF-filer. Floppydisken er definert som et direktory med navnet MULT-DIR. Bruker MULTITERM er tildelt all plassen på direktory.

2. Generering av MT-10

MT-10 består av 3 moduler (fig. 2.1):

MULTITERM (MT) RT-programmer for terminalbehandling

EMULATOR (EM) RT-programmer for emulering av
vertsmaskin-protokoller
(foreløpig er en DCT1000 emulator implementert)

BUFFER-POOL (BP) Reentrante subrutiner for reservering av buffere,
oppbygging av meldinger og køer, samt
avsetting av areal til buffere.

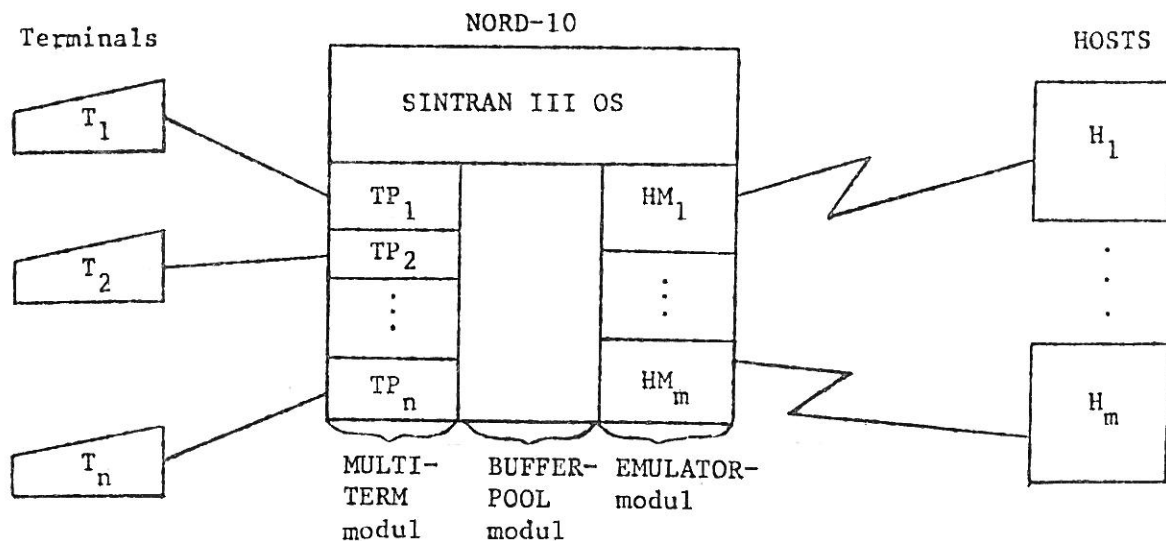


Fig. 2.1 Prinsippskisse av MULTITERM-10

2.1. MULTITERM-modulen

MT-modulen består av 3 symbolske filer. Hoveddelen med N-PL-kode ligger på filen MULT:SYMB.

Dessuten benyttes filene MAC-MACRO, som inneholder nødvendige macro-definisjoner, og MAC-IMULT som inneholder BRF-hodet (definisjoner av ENTRY-punkter, eksterne referanser o.l.).

Generelt gjelder at filer med navn som begynner med MAC-inneholder symbolsk MAC-kode.

Generering av MT-modulen er vist på fig. 2.2. Før generering starter, gjøres eventuelle oppdateringer med editoren (QED). Deretter kan generering av en ny BRF-versjon (MULT:BRF) gjøres ved hjelp av ferdig oppsatte MODE-filer.

Kompilering gjøres med MODE-filen KOMP-MULT (uten listing) eller med MODE-filen LKOMP-MULT (med listing). Begge gir symbolsk MAC-kode ut på filen T1.

T1 sammen med filene MAC-IMULT og MAC-MACRO kjøres så gjennom en redigeringsfunksjon med MODE-filen EDIT-MULT. Resultatet blir fullstendig MAC-kode lagt ut på filen MAC-MULT.

Redigeringsfunksjonen, som gjøres av MODE-filen EDIT-MULT, er innført for å kunne generere 2PASS-listing når man bruker macroer. Dette er en svakhet ved MAC og burde vært unødvendig.

Assemblering gjøres av MODE-filen ASS-MULT som gir ferdig BRF-kode ut på filen MULT:BRF.

2PASS-listing fås ved å kjøre MODE-filen LASS-MULT. Denne gir ikke BRF-kode ut, kun en 2PASS-listing av MT-modulen.

2.2. EMULATOR modulen

EM-modulen består av 3 symbolske filer. Hoveddelen med N-PL-koden ligger på filen EMUL:SYMB. Macrodefinisjonene ligger på samme fil som for MT, MAC-MACRO, mens BRF-hodet ligger på filen MAC-IEMUL.

Generering av EM-modulen er vist på fig. 2.3, som er identisk med fig. 2.1 bortsett fra navn på program-filer og MODE-filer.

Kompilering, assemblering og 2PASS-listing lages etter samme mønster som for MT-modulen. Ferdig BRF-kode legges ut på filen EMUL:BRF.

2.3. BUFFER-POOL modulen

Generering av BP-modulen er vist i fig. 2.4. Denne er noe enklere enn å generere de to andre BRF-modulene fordi redigeringsfunksjonen her er unødvendig. Ingen MAC-filer skal redigeres inn.

N-PL-koden ligger på filen POOL:SYMB, ferdig kode vil legges på filen POOL:BRF. Forøvrig virker MODE-filene likt med tilsvarende MODE-filer for de andre modulene.

Fig. 2.2 MULTITERM maintenance

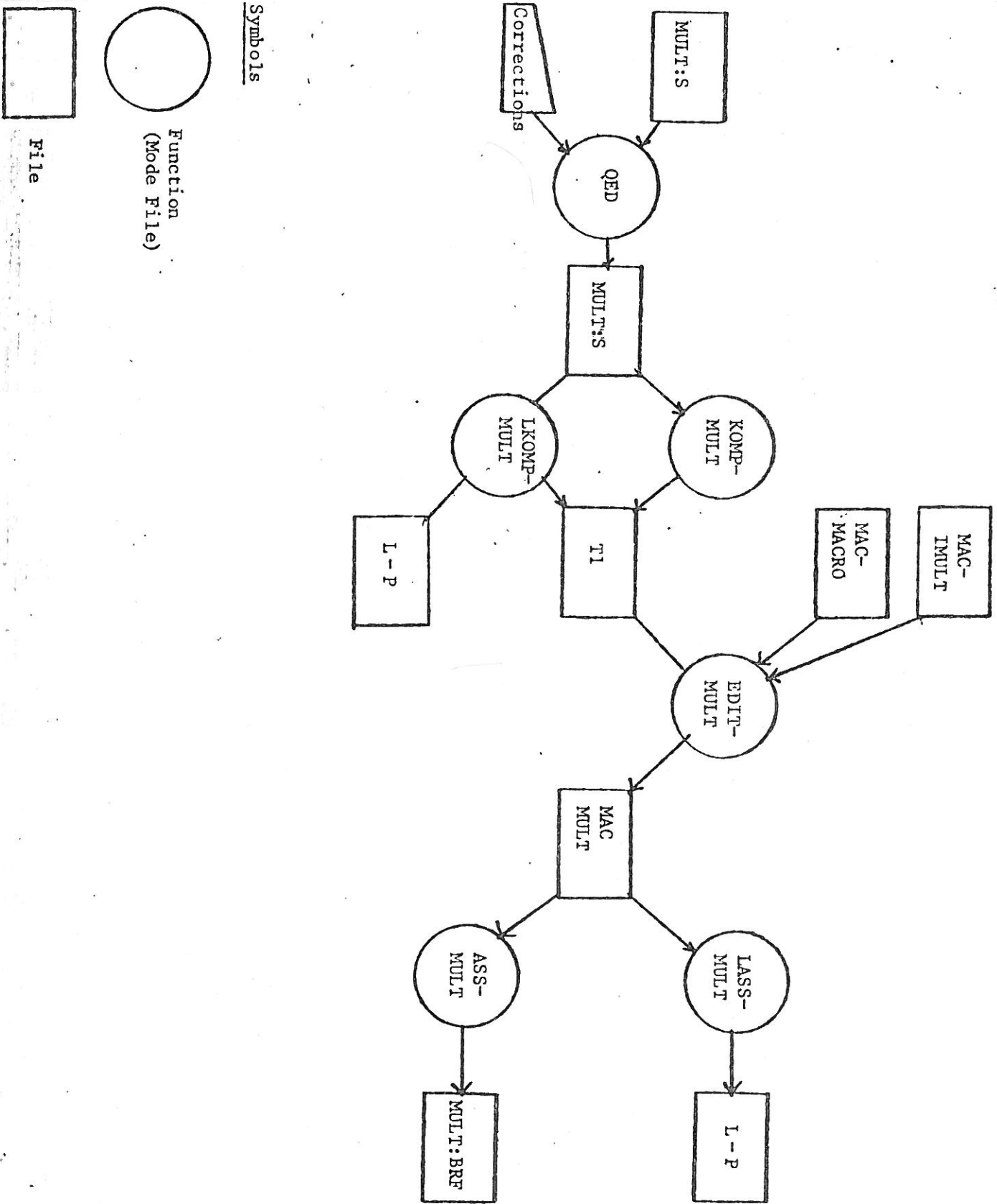


Fig. 2.3 DCT1000 Emulator Maintenance

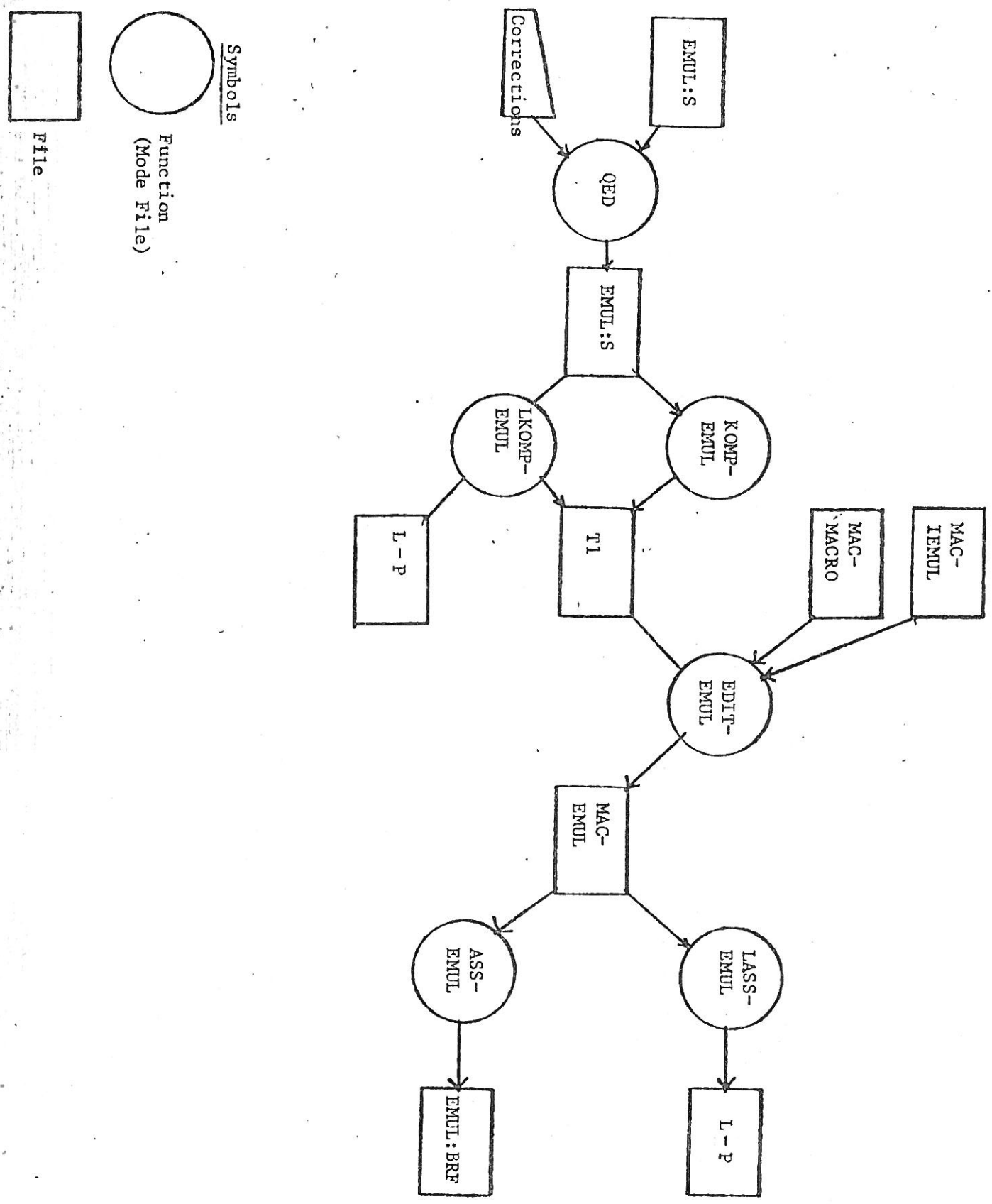
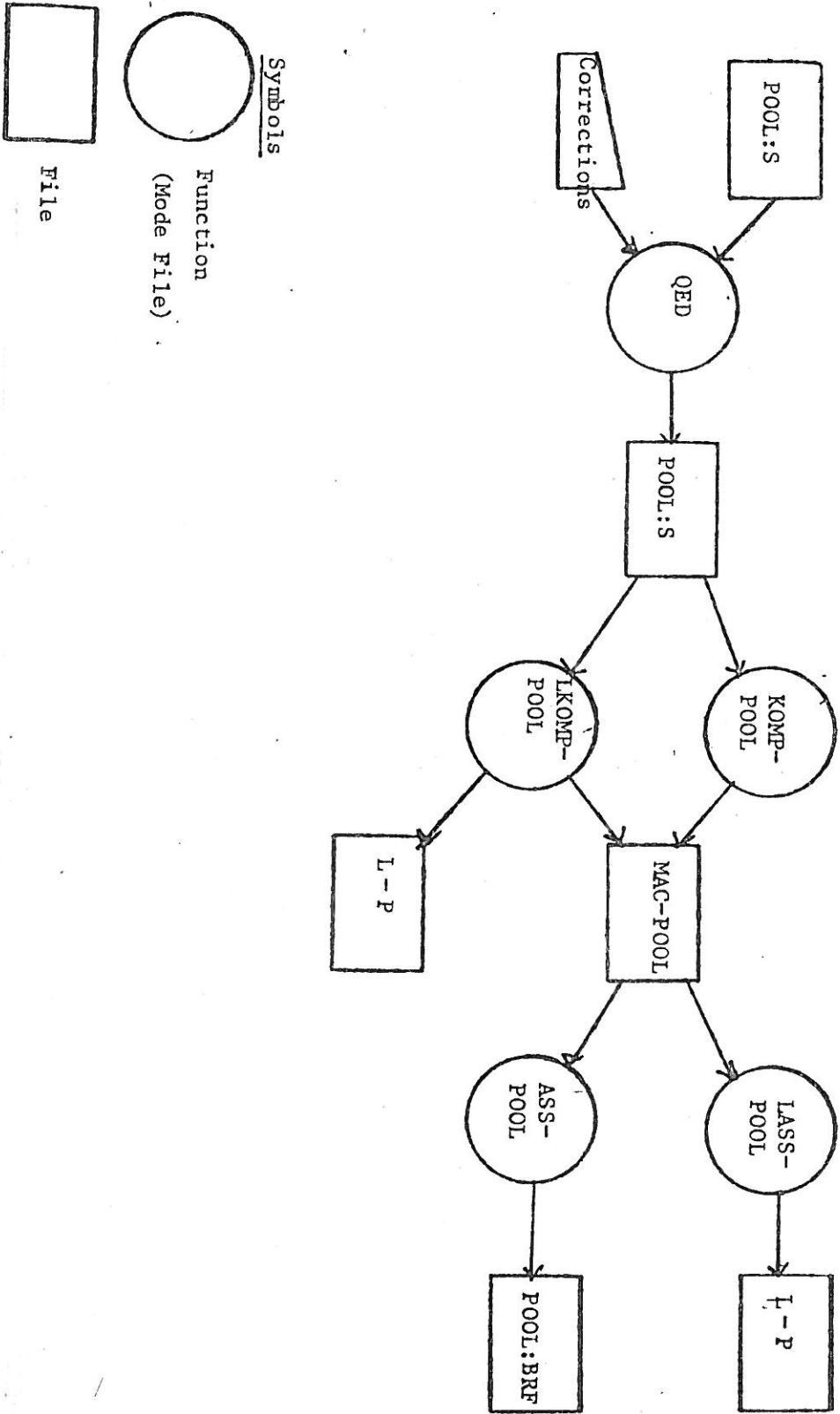


Fig. 2.4 Buffer Pool System Maintenance



3. Systemparametre

Systemkonfigurering gjøres ved å redigere de symbolske filene som MT-10 består av.

3.1. Navnekonvensjoner

RT-programmer benevnes heretter som prosesser.

Prosessene er gitt RT-navn som består av fire tegn. Navnene bygges opp av to parametre til makrokallet GCB (General Control Block). Parametrene består av to tegn hver. GCB-macroen genererer kontrollblokker for prosessene.

De to første tegn i et navn angir prosess type. De to siste brukes til å gi en entydig identifikasjon av prosesser av samme type. Alle macro-definisjoner og parametre er beskrevet på filen MAC-MACRO.

Eksempel 3.1: RT-navn for prosesser

TI00 Terminal Input, terminal nr. 00
TO01 Terminal Output, terminal nr. 01
AITR Aux Input, Tape Reader
HIUN Host Input, UNIVAC

Prosess-kontrollblokkene gis navn som består av tegnene PCB og et to-sifret løpenummer: PCB00, PCB01, ..., PCBnn. Disse navnene genereres ved kallene på GCB-macroen.

3.2. MULTITERM-modulen

MT-modulen inneholder to prosesser for hver terminal og en prosess for hvert input eller output "device" utenom terminaler (f.eks. TAPE-READER, LINE-PRINTER og overføring til/fra filer). Foreløpig er bare terminalprosesser implementert.

Dessuten inneholder modulen RT-programmet MULT som kontrollerer inn- og ut-logging av terminaler.

Alle prosesser (bortsett fra MULT) har dataområder som genereres ved hjelp av macroer. For å innføre en ny terminal eller fjerne en terminal kreves derfor at de rette makrokalene adderes inn eller fjernes. Dessuten må den

globale prosessstabellen PROTB oppdateres. PROTB inneholder pekere til prosess-kontrollblokkene for alle prosesser. Tabellen er organisert som følger:

Først kommer prosesser av type HOST, deretter prosesser av type TERMINAL og sist prosesser av type AUX. Input-prosesser er assosiert med like, og outputprosesser med ulike indekser i tabellen. Pekeren NONE brukes til å markere "huller" i tabellen.

Eksempel 3.2: Layout av PROTB

I dette eksemplet har en regnet med en HOST prosess, to terminaler, en input og en output AUX-prosess.

PROTB/ PCB00	% HOST CONTROL BLOCK, INPUT AND OUTPUT
NONE	% INGEN EGEN OUTPUTPROSESS
PCB02	% TERMINAL CONTROL BLOCK, INPUT
PCB03	% TERMINAL CONTROL BLOCK, OUTPUT
PCB04	% TERMINAL CONTROL BLOCK, INPUT
PCB05	% TERMINAL CONTROL BLOCK, OUTPUT
PCB06	% AUX CONTROL BLOCK, INPUT
PCB07	% AUX CONTROL BLOCK, OUTPUT

Tre symboler, NHOST, NTERM og NAUX, må ha verdier som tilsvarer henholdsvis høyeste HOST PCB indeks, høyeste terminal PCB indeks og høyeste AUX PCB indeks. Dette kan regnes ut som følger:

$NHOST = 2 * (\text{antall HOST'er}) = 2 * 1 = 2$
 $NTERM = NHOST + 2 * (\text{antall terminaler}) = 2 + 2 * 2 = 6$
 $NAUX = NTERM + (\text{antall AUX prosesser}) = 6 + 2 = 10$

Utrekningen er gjort for eksemplet ovenfor. Alle verdier er oppgitt oktalt.

3.2.1. Generering av kontrollblokker

Forkortelsen PCB står for Process Control Block. Hver prosess krever at det genereres en PCB av riktig type. En PCB består av en generell del som i prinsippet er identisk for alle typer prosesser, og en spesiell del som skiller mellom de ulike typer prosesser. Det er definert macroer for generering av hver del i en PCB.

Generering av PCB for HOST-prosesser er beskrevet i 3.3.1.

1. Terminal Input PCB

Bygges opp ved kalli på tre macroer i angitt rekkefølge:

GCB - General Control Block
ICB - Input Control Block
TCB - Terminal Control Block

Bare GCB krever parametre til kallet. Parametrene er beskrevet i forbindelse med macrodefinisjonene på filen MAC-MACRO.

2. Terminal Output PCB

Bygges opp ved kalli på to macroer i angitt rekkefølge:

GCB - General Control Block
OCB - Output Control Block

Bare GCB krever parametre til kallet. Se pkt. 1.

3. AUX Input PCB

Disse vil i prinsippet bygges opp som Terminal Input PCB'er. Ikke implementert pr. idag.

4. AUX Output PCB

Disse vil i prinsippet bygges opp som Terminal Output PCB'er. Ikke implementert pr. idag.

3.2.2 Installasjonsavhengige parametre

Følgende symboler kan eller må defineres for hver individuell MT-10 installasjon.

NHOST Forklaring gitt i eksempel 3.2.

NTERM Forklaring gitt i eksempel 3.2. *NOST*

NAUX Forklaring gitt i eksempel 3.2.

IPRIO SINTRAN prioritet for Terminal Input Prosess
Anbefalt verdi: 41 oktalt.

OPRIO SINTRAN prioritet for Terminal Output Prosess
Anbefalt verdi: 41 oktalt.

MPRIO SINTRAN prioritet for MULT-prosess
Anbefalt verdi: 40 oktalt.

MAXAT Maks. antall simultant aktive MT terminaler (brukes av
MULT-prosessen).

INTDV Logical Unit No for SINTRAN INTERNAL DEVICE. Brukes til
kommunikasjon mellom bakgrunnsprogram og MULT-
prosessen.

SEMNO Logical Unit No for semafor som brukes til
kø-beskyttelse.

3.3. EMULATOR-modulen

EM-modulen skal etter planen inneholde emulatorer og andre
vertsmaskin-avhengige deler for alle vertsmaskiner det er
aktuelt å knytte seg til.

Pr. idag inneholder EM-modulen en DCT1000 emulator prosess.
Dessuten inneholder den en del rutiner som er
vertsmaskin-avhengige, men som kalles fra prosesser i
MT-modulen. Dette er gjort med hensikt for å oppnå størst
mulig uavhengighet mellom modulene, og samtidig ha et "renest"
mulig grensesnitt. Som eksempel på vertsmaskin-avhengige
rutiner som må kalles fra MT-modulen, kan nevnes ILOGR
(innlogging til en HOST) og OLOGR (utlogging fra en HOST).

NB! På grunn av DCT1000-emulatorens struktur er det bare behov
for en HOST input prosess (denne gjør i virkeligheten både
input og output). Pekeren til HOST output prosessen i PROTB
har derfor verdien NONE (se eksempel 3.2).

3.3.1. Generering av kontrollblokker

1. DCT1000 Input PCB

Bygges opp ved kall på fire macroer i angitt rekkefølge:

GCB - General Control Block

HCB - Host Control Block
Felles for alle emulator-prosesser i MT-10.

UNCB - UNIVAC Control Block
DCT1000-avhengig del.

UNTCB - UNIVAC Terminal Control Block
En UNTCB må genereres for hver DCT1000 terminal.

Parametrene er beskrevet i forbindelse med macro-definisjonene på filen MAC-MACRO.

Det er ikke behov for noen Output PCB for DCT1000-emulatoren.

3.3.2. Installasjonsavhengige parametre

Semaforer

Minst 2 semaforer må være definert i SINTRAN med egne Logical Unit No.

SEM00 brukes til beskyttelse av køer. Se parameter-beskrivelse for UNTCB-macro på filen MAC-MACRO.

SEM01 brukes til å sikre eksklusiv aksess for inn- og ut-loggings-rutiner. Se parameter-beskrivelse for UNCB-macro på filen MAC-MACRO.

NB! SEM00 og SEM01 må ha forskjellige Logical Unit No.

Prioritet

SINTRAN-prioriteten til emulatoren bestemmes av verdien til symbolet EPRI0. EPRI0 settes vanligvis over prioriteten for bakgrunnsprogrammer, f.eks. EPRI0=41 oktalt.

Termineringsbetingelse

Valg av termineringsbetingelse i SINTRAN synkron modem driveren bestemmes av parameteren TCOND. I basisversjonen er TCOND=10 oktalt.

Termineringsbetingelse for DCT1000 finnes ikke i SINTRAN's driver. Det må derfor gjøres en patch i driveren for at TCOND=10 skal virke. Patchen er dokumentert i Appendiks A.

3.3.3. Antall DCT1000-terminaler

Antallet DCT1000-terminaler trenger ikke å være det samme som antallet MT terminaler. To eller flere MT terminaler kan dele en DCT1000 SITE-ID. Det vil si at en bruker risikerer å få avslag ved forsøk på å logge inn på UNIVAC (emulatoren gir avslag).

Koplingen mellom MT terminaler og DCT1000 terminaler finnes i tabellen TCBTB. Hvert element i tabellen er en peker til en DCT1000 kontrollblokk. MT terminalnummeret brukes som indeks for oppslag i tabellen.

Antall MT terminaler gis som verdi til symbolet NOFT.
Antall element i TCBTB må være lik NOFT.

Antall DCT1000 terminaler gis som verdi til symbolet NOFTM.
NOFTM må være mindre eller lik NOFT.

Verdien av symbolet MAXSNO må være NOFTM-1.

Eksempel 3.3:

Antall MT terminaler er lik 5.
Antall DCT1000 terminaler er lik 4.
MT terminal nr. 0 og 1 deler DCT1000 terminal nr. 3.

TCBTB/	TCB03	%	MT	TERM	0
	TCB03	%	MT	TERM	1
	TCB00	%	MT	TERM	2
	TCB02	%	MT	TERM	3
	TCB01	%	MT	TERM	4

NOFT= 5
NOFTM= 4
MAXSNO=3

3.4. BUFFER-POOL modulen

BP-modulen består av et sett reentrante subrutiner for generering av meldinger, reservering av buffere og oppbygging og administrasjon av køer.

Før kompilering av modulen må verdien for følgende symboler defineres:

NOFBU= Antall buffere i "poolen".
BSIZW= Bufferlengde i antall ord.
PQSEM= Logical Unit No for "pool" beskyttelses semafor.

NB! PQSEM må være forskjellig fra SEM01 i EM-modulen!
(se kap. 3.3).
PQSEM kan være lik SEM00.

4. Lasting av MT-10.

Lasting av MT-10 må gjøres med RT-loaderen. På direktory (MULT-DIR) finnes en MODE-file for å utføre dette, LOAD-MULTITERM. Denne MODE-filen må imidlertid tilpasses de lokale omgivelser:

Segmentnr. og navn på RT-program må redigeres inn. Eventuelle patcher og start av MT-10 (se nedenfor) kan med fordel også legges på denne MODE-filen. Eksempel kan tas fra den filen som følger med.

Filen LOAD-MULTITERM må kopieres til bruker SYSTEM eller bruker RT før bruk.

MT-10 startes med kommandoen @RT MULT

og stoppes med @ABORT MULT

(Alle MT-10-brukere bør helst være logget ut).

5. Innlogging.

Innlogging skjer ved kommandoen @MULTI-TERM.

MULTI-TERM er navnet på en programfil som ligger under bruker SYSTEM. Filen inneholder absoluttprogrammet MULT-LOGIN. Dette programmet er enkelt; Det skriver en melding på et internt device. Meldingen leses av MULT-prosessen (kap 3.)

MULT-LOGIN-programmet ligger som symbolsk N-PL-kode på filen MULT-LOGIN:SYMB. Det må kompileres, assembleres og legges på filen MULTI-TERM:PROG under bruker SYSTEM.

NB!

Innloggingsprogrammet forutsetter at det interne devicet som benyttes er ordorientert. Dette krever en patch i SINTRAN. Den er beskrevet i Appendiks A.

APPENDIKS A
=====

Patcher i SINTRAN

SINTRAN SYNKRON MODEM DRIVER (76.05.17C)

Patcher for MT-10 er merket av.

```

046624
046624 %XXXXXXXXXXXXXXXXXXXXX C D R 1 XXXXXXXXXXXXXXXXXXXXXXXX
046624
046624 *"MODIN
046624
046624 %XXXXXXXXXXXXXXXXXXXXX
046624 % SYNCHRONOUS MODEM DRIVER
046624 % AND HANDLING ROUTINES FOR SINTRAN III
046624 %XXXXXXXXXXXXXXXXXXXXX
046624
046624 SUBR MODIN,MUSTI,MOTRI,MOSTO,MOTRO,MODUI,MOTMO,MOTMI,MODII
046624 DISP =12
046624 INTEGER MOWSTAT
046624 INTEGER TERSW,MOSW
046624 PSID
046624 % SET STATUS FOR MODEM INPUT
046624 % A=-3: READ MOSW, AND RETURN IT TO THE USER
046624 % A=-2: INITIATE BLOCK RECEIVE (SEARCH FOR SYNC)
046624 % A=-1: CLEAR INPUT BUFFER
046624 % A= 0: UNDEFINED TERMINATE CONDITION
046624 % A= 1: SET OCT-2000 TERMINATE CONDITION
046624 % A= 2: SET CD-200USER TERMINATE CONDITION
046624 % A= 3: SET GERTS=115 TERMINATE CONDITION
046624 % A= 4: SET IBM-3780/2780 TERMINATE CONDITION
046624 % A= 5: SET VIP TERMINATE CONDITION
046624 % A= 6: SET HASP WS TERMINATE CONDITION
046624 % A= 7: SET UNIVAC NTR TERMINATE CONDITION
046624 % A=10: SET MSV2 TERMINATE CONDITION
046624 % A=11: SET IBM-3270 TERMINATE CONDITION
046624 % A=12: SET BTH/VIP7700 TERMINATE CONDITION
046624 %
046624 %
046624 % SET STATUS FOR MODEM INPUT
046624 MOSTI: IF A<0 GO MOSI5
046625 IF A>12 GO MOSI6
046630 A=:TERSW; EXIT % SET TERMINATE CONDITION
046632 MOSI5: IF A+4<0 GO MOSI6
046634 GOSW MOSI4, MOSI3, MOSI2, MOSI1
046641 MOSI4: EXIT
046642 MOSI3: "0"; IF T:=MOSW-4=0 THEN 5=:MOSW;1 FI;EXIT
046653 MOSI2: T:=HDEV;*EXR ST;EXR STX DUMMY READ
046656 2204; T:=HDEV+DCONT; *EXR ST
046662 2205; GO MOSI7
046664 MOSI1:2204
046665 MOSI7: T:=HDEV+DCONT;*EXR ST
046670 0=:MOSW
046671 GO CLBUF
046672 MOSI6: EXIT
046673 %
046673 % MODEM INPUT DRIVER
046673 MODIN:
046673 CALL ID12
046674 MODI1: T:=HDEV+DDR;*EXR ST
046677 A=:LAST
046700 IF T:=CFREE=0 GO MOSWX
046703 T:=MOSW GOSW MOSW0,MOSW1,MOSW2,MOSW3,MOSW4,MOSW4,MOSW6,M
045716 MOSW4: GO MODIN
045717 %TERMINATOR FOUND
046717 MOTERM: LAST

```

```

046720 MOSW3: *PIOF
046721 CALL RBPOT; *PION
046723 MOSWX: CALL RTACT
046724 IF TERSW=7 OR =11 OR =12 GO MOSWY % NTR/3270/B
046736 T:=HDEV+DCONT (10)
046740 2204; *EXR ST OLD NEW
046742 0=:TMK MOSWX+2/171007 17101
046743 4=:MOSW (46725)
046745 GO MODIN
046746 MOSWY: T:=HDEV+DCONT
046750 2204; *EXR ST
046752 2205; *EXR ST
046754 0=:TMK=:MOSW
046756 GO MODIN
046757 MOSW2: GO IBM2
046760 MOSW1: GO IBM1
046761 MOSW6: GO HASP1
046762 MOSW7: GO HASP2
046763 MOSW8: GO HASP3
046764 *)FILL
046773 % TEST FOR TERMINATION CONDITION
046773 eICR;
046773 MOSW0: T:=TERSW GOSW FAR GEN0,FAR DCT0,FAR CDC0,GER0,IBM0,VIP0
047004 NTR0,FAR MSV20,FAR I3270,FAR BTH0;
047010 eCR;
047010 % HASP WS
047010 HASP0: IF A=62 OR =377 GO FAR MORET
047016 *PIOF
047017 CALL RBPOT; *PION
047021 T:=6=:MOSW % NEXT IS HANDLED BY HASP1
047023 GO FAR MODPIN
047024
047024 HASP1:
047024 HASP2: *PIOF
047025 CALL RBPOT; *PION
047027 IF A=377 THEN MIN MOSW FI
047033 GO FAR MODPIN
047034
047034 HASP3: *PIOF % 3. PAD IS HANDLED AFTER A MESSAG
047035 CALL RBPOT; *PION
047037 IF A=377 GO MOSWX % TERMINATE
047042 T:=6=:MOSW
047044 GO FAR MODPIN
047045
047045 %VIP CONTROL (HONEYWELL=BULL INTERACTIV TERMINALS)
047045 VIP0: IF A=0 OR =26 GO FAR MORET; *PIOF
047052 CALL RBPOT;A/0177; *PION
047055 IF A=3 THEN T:=2=:MOSW ; GO FAR MODPIN; FI
047063 IF A=4 GO MOSWX
047066 GO FAR MODPIN
047067 % IBM 3780/2780
047067 IBM0: IF A=62 GO FAR MORET; *PIOF
047073 CALL RBPOT; *PION
047075 IF A=3 OR =46 OR =37 THEN MIN MOSW FI
047107 IF A=377 GO MOSWX
047112 GO FAR MODPIN
047113 IBM1: MIN MOSW; *PIOF
047115 CALL RBPOT; *PION
047117 GO FAR MODPIN
047120 IBM2: 0=:MOSW; *PIOF
047122 CALL RBPOT; *PION

```

```

047124      GO FAR MODPIN
047125      %GERTS=115
047125      GER0:  IF A=0 OR =26 OR =377 GO FAR MORET; *PIOF
047135      CALL RBPOT; *PION
047137      A/0177
047140      IF A=3 THEN A=:MOSW FI
047144      *"BKGS

"047144      GO MODPIN
047145      %UNIVAC NTR
047145      NTR0:  *PIOF
047146      CALL RBPOT; *PION
047150      IF A=3 THEN A=:MOSW FI
047154      GO MODPIN
047155      *)FILL

047170      %CD=200=USER
047170      CDC0:  GO GER0
047171      % NOT DEFINED
047171      GEN0:  GO GER0
047172      %DCT=2000
047172      DCT0:  IF =0 OR =26 OR =377 GO MORET; *PIOF
047202      CALL RBPOT; *PION
047204      A/0177
047205      IF =3 OR =4 OR =6 OR=25 OR=7 OR =21 THEN 3=:MOSW FI
047231      GO MODPIN
047232      % MSV2
047232      MSV20:  IF A=26 GO MORET; *PIOF
047236      CALL RBPOT; A/0177; *PION
047241      IF A=3 OR =27 THEN MIN MOSW FI
047250      IF A=177 GO FAR MOSWX
047253      GO MODPIN
047254
047254      % IBM=3270 EMULATOR
047254      I3270:  IF A=62 GO MORET; *PIOF
047260      CALL RBPOT; *PION
047262      IF A=3 OR =46 OR =37 THEN MIN MOSW FI
047274      IF A=377 OR =67 GO FAR MOSWX
047302      GO MODPIN
047303
047303      % BTH/VIP 7750 INTERACTIVE TERMINALS
047303      BTH0:  IF A=0 GO MORET; *PIOF
047305      CALL RBPOT; *PION
047307      A/0177
047310      IF A=3 OR =27 THEN
047316      2=:MOSW; GO MODPIN
047321      FI
047321      IF A=4 GO FAR MOSWX
047324      GO MODPIN
047325
047325      MODPIN:
047325      MORET:  205; T=:HDEV+DCONT; *EXR ST
047331      =2=:TMR
047333      GO MODIN
047334      %
047334      %
047334      % TIMER ROUTINE FOR MODEM INPUT
047334      NUTMI:  4=:MOSW

```

IF A=3 OR=27 THEN
 3=: MOSW FI

MSV20+14/12400 124005
 (47246) 40770 170403
 171177 4770
 142065 124002
 125102 12400


```

047336          2204/T:=HDEV+DCONT)*EXR ST
047342          GO RTACT
047343          % IOTRANS ROUTINE FOR MODEM INPUT
047343          MOTRI: L+1;IF BHOLD>0 GO RBGET
047347          A:=-1;EXIT;*)FILL
047362          %
047362          %
047362          % SET STATUS FOR MODEM OUTPUT
047362          % A=-1: CLEAR MODEM OUTPUT BUFFER
047362          % A= 0: START SENDING OF CURRENT BLOCK
047362          % A= 1: SET 377 - BYTES TO SEND BETWEEN DATA
047352          % A= 2: SET ASCII SYNC TO SEND
047362          % A= 3: SET EBCDIC SYNC TO SEND
047362          % A= 4: SYNCHRONIZE ON 26 RECEIVED (ASCII)
047362          % A= 5: SYNCHRONIZE ON 62 RECEIVED (EBCDIC)
047362          % A= 6: SYNCHRONIZE ON 226 RECEIVED (ASCII)
047362          %
047362          MOSTN: IF A+1<0 OR >7 THEN EXIT FI
047370          GUSW MOSCL,MOS0,MOS1,MOS2,MOS3,MOS4,MOS5,MOS6
047401          MOS1: 777; GO MOS11
047403          MOS2: 426; GO MOS11
047405          MOS3: 462; GO MOS11
047407          MOS4: 1026; GO MOS11
047411          MOS5: 1062; GO MOS11
047413          MOS6: 1226
047414          MOS11: T:=HDEV-3)*EXR ST;EXIT % UPDATE HARDWARE SYNCREGIST
047420          MOSCL: 0:=:MOWSTAT=:TMR=:DERROR
047423          GO CLBUF
047424          MOS0: 5; T:=HDEV+DCONT)*EXR ST
047430          TTMR=:TMR/2:=:MOWSTAT;EXIT % SET WAIT CONDITION
047435          % IOTRANS ROUTINE FOR MODEM OUTPUT
047435          MOTRU: IF T:=MOWSTAT=2=0 THEN EXIT FI % CALLING PROGR WILL
047442          L+1; GO RBPUT
047444          %
047444          % DRIVER FOR MODEM OUTPUT
047444          MODUT: 00;*)IOF
047445          IF BHOLD=0 THEN 0=:TMR=:MOWSTAT; T:=0; CALL RTACT; GO
047455          CALL RBGET; T:=HDEV+DDW)*EXR ST; PION
047462          TTMR=:TMR
047454          T:=5
047465          HIT: T:=A; T:=HDEV+DCONT)*EXR ST;ION
047472          CALL ID10
047473          00
047474          %
047474          % TIME-OUT SUBROUTINE FOR MODEM OUTPUT
047474          MOTMO: 12=:DERROR
047476          "0"; T:=HDEV+DCONT; *EXR ST
047502          IF ISTATE>0 GO RTACT
047505          0:=:MOWSTAT
047506          EXIT
047507          RBUS
047522
047522
047522          SUBR IOREM
047522          IOREM: IF X:=X.RTRES >< 0 THEN CALL RTENT; CALL FTIMOU FI
047526          CALL STUPR
047527          RBUS
047532

```



NORD SOFTWARE LIBRARY
SOFTWARE SYSTEM REPORT

PROGRAM	NAME SINTRAN		PROGRAM NUMBER(S) 77.05.17
REPORT	NUMBER SIN 271		
REFERENCE	DATE 77.05.25	NAME TJS	
REASONS	<input checked="" type="checkbox"/> INFORMATION <input type="checkbox"/> ERROR <input checked="" type="checkbox"/> PATCH		CORRECTED OR CHANGED IN
COMPUTERS	<input type="checkbox"/> 1 <input type="checkbox"/> 2B <input type="checkbox"/> 5 <input checked="" type="checkbox"/> 10 <input type="checkbox"/> 12 <input type="checkbox"/> 20 <input type="checkbox"/> 50 <input type="checkbox"/> _____		
OPERATING SYSTEMS	<input type="checkbox"/> TSS <input checked="" type="checkbox"/> SIN III M <input type="checkbox"/> SIN III C <input type="checkbox"/> ALONE <input type="checkbox"/> _____		
SUBJECT	INTERNAL DEVICES USING WHOLE WORDS		

DESCRIPTION, ACTION, PATCHES

The following patches must be done on each internal device to be run with whole words. Use the CHANGE-DATAFIELD command of the SINTRAN-SERVICE-PROGRAM. (logical device numbers for byte oriented internal devices are 2008-2388)

	old	new
INPUT:	7/ IGTCH	GETW
	MAX/ xxx	xxx/2
	CFREE/ xxx	xxx/2
OUTPUT:	TRLREG/IPTCH	PUTW

For PUTW, GETW see section 3.5.6.3 in part 1 of the listing.

3-3-P

Eksempel

SS 28/2-78

◆CHAN-DAT

LOG.UNIT NO: 200
INPUT OR OUTPUT? INPUT
MEMORY? YES
IMAGE? YES
SAVE-AREA? YES

MEMORY IMAGE SAVE-AREA

7/	12256	12256	12256	12260				
	12270	12270	12270	MAX/	100	100	100	40
	0	0	0	CFREE/	100	100	100	40
	0	0	0	.				

◆CHA-DAT

LOG.UNIT NO: 200
INPUT OR OUTPUT? OUTPUT
MEMORY? YES
IMAGE? YES
SAVE-AREA? YES

MEMORY IMAGE SAVE-AREA

TRLREG/	12224	12224	12224	PUTW
?				
TRLREG/	12224	12224	12224	12241
	12302	12302	12302	.

III