

Regnecentralen
21.7.62.
B.B. nr. 6.

ALGOL OG MAGNETBAAND.

Procedurer til Magnetbaandsadministration.

I sin oprindelige form tillader DASK ALGOL ikke benyttelse af magnetbaand. Dette er imidlertid muliggjort med NL 5.

Nedenfor er vist eksempler paa en procedure til start af magnetbaandene, en procedure til magnetbaandsadministration og endelig et eksempel paa et lille program (udskrift fra magnetbaand), der benytter disse procedurer.

1. Almindelige bemærkninger.

Ved de fleste indhop til (kald af) procedurerne skal der opgives en adresse i ferritlageret (begyndelsesadresse for en blok). Denne kan selvfølgelig vælges vilkaarligt, men ønsker man at benytte den automatiske lagerkontrol, der findes i DASK ALGOL, kan man erklære sine blokomraader i ferritlageret som arrays. Begyndelsesadressen for et saadant array faar man fat i paa følgende maade (jvfr. A MANUAL OF THE DASK ALGOL LANGUAGE side 53, linie 21 f.o.) idet der forudsættes erklæringer som f eks.:

```
integer array a[0:7,0:7,1:2];  
integer A;
```

a maa være af vilkaarlig dimension, blot skal a optage min. 64 hec.

code

```
60 a      ; AR := 40/60 a0,c  
0c 1      ; fjern c-mærke  
0f 9      ; a0 med enhed pos. 19  
28 A      ; A := beg. adr. for a-array
```

Ved kald af procedurerne bruges A da som begyndelsesadresse for blokken.

Af hensyn til synkronismen skal der som globalt array opgives integer array synkro[1:4], der indeholder numrene paa de blokke, der staar i bufferne. Dette array maa af hensyn til korrekte tællinger ikke erklæres umiddelbart efter array-erklæringen for blokomraaderne i ferritlageret.

2. Baandstart.

Den viste baandstartprocedure er en let modificeret udgave af BA 1, der blot er ændret saaledes, at der ingen udskrift foretages, hvis baandstarten foregaar korrekt, og først ved galt baand paa den ønskede baandstation eller mere end 3 omlæsninger foretages udskrift. Denne kan iøvrigt let ændres, idet den læste daab staar i 0,c og den forlangte i daab (ordreform) eller i opgnr og lbnr (hel-tal).

```
procedure MBstart(station, opgnr, lbnr, FL);
  value station, opgnr, lbnr, FL;
  integer station, opgnr, lbnr, FL;
begin comment proceduren starter baandstation 2000 + station med daaben:
  opgnr, lbnr. FL er arbejdsplads for blokke (64 hec);
  code
    60 lbnr
    0f 20
    20 opgnr
    0c 8
    08 daab ; gem daaben
    60 FL
    0c 8
    29 k + 1
    55 ; irc := FL
    40 -2,c
    08 daab + 2 ; gem FL-celle
    69 fejl ; retabler fejltæller
forfra: 66 fejl ; fejltæller := fejltæller + 2
    21 grænse
    11 udskr 1 ; hvis fejltæller = 8
    60 station
    0c 8
    20 konst
    29 k + 1
    1c ; vælg station
igen: 60 een
fejl: 1e
konst: 1e 2000 ; synkroniser
grænse: 5e 7
    1d -2,c
    12 igen
    61 -2,c
    1e ; søg blok 0
    5e
    1d -2,c ; læs blok 0
    12 forfra ; tobitfejl
    63 -2,c
    51 forfra ; bloknr.fejl
    40 ,c
    01 daab
    08 -2,c
    43 -2,c
    51 udskr 2 ; hvis galt baand
    40 daab + 2
    08 -2,c ; retabler FL-celle
    60 station
    17 595
    40 synkro ; synkro[station] := 0
```

```

17 595
69 fejl
17 158
10 end
=daab: 0
      0
      0
      0
een:   h   1
algol;
udskr 1: skrvvr; skrvtekst(⟨⟨baandfejl paa station ⟩⟩); go to stop;
udskr 2: skrvvr; skrvtekst(⟨⟨galt baand paa station ⟩⟩);
stop:   skrv(⟨nddd⟩, station + 2000);
code
      30 forfra - 1
algol;
end:
end procedure MBstart;

```

3. Baandadministration.

Denne procedure bestaar væsentligst af en forsats, der transformerer parametrene, saaledes som de afleveres ved kaldet af proceduren, til den form, der er forlangt af Baandadministration nr. 3 (KS 61 nr. 2). Den omfatter kun indhoppene MB 1 og MB 3, mens MB 2 er udeladt (samme virkning kan faas ved kald af MB 1 + MB 3). Endvidere er baandstartdelen af KS 61 nr. 2 erstattet af proceduren MBstart. Saafremt man ønsker at bruge MB 2, kan dette ske ved at udvide switch S, der styres af parameteren funktion.

```

procedure MBadm(funktion; station; blok);
      value funktion; station; blok;
      integer funktion, station, blok;
begin comment Sammenhængen mellem værdien af funktion og indhoppene i KS 61 nr. 2
er følgende:

```

funktion	virkning
1	MB 1
2	MB 3 læsning
3	MB 3 skrivning

For funktion = 1 har parameteren blok betydningen bloknr., mens den for funktion = 2 og 3 har betydningen FL (ferritlageradresse for blok).

Som globalt array kræves integer array synkro[1:4], der indeholder de aktuelle bloknr.;

```

switch S := S1, S2, S3;
code
      60 synkro
      20 minus
      29 MBtrim           ; sæt beg.adr. synkroceller
algol; go to S[funktion];
S1: code
      60 station
      28 k + 3           ; sæt parameter
      60 blok
      16 MB 1
      0
      10 kontrol

```

```
algol; S2: code
        60 blok
        0c 9
        0f 1
her:    20 station
        28 k + 2
        16 MB 3
        0
        10 kontrol
algol; S3: code
        60 blok
        0c 9
        0f 1
        20 minus
        10 her
minus:  00 ,c
```

[Her følger koden for KS 61 nr. 2 minus baandstartdelen]

```
        .
        .
        .
kontrol: 11 end
algol; skrvvr; skrvtekst(⟨<Baandfejl paa station ⟩);
skrv(⟨nddd⟩, station + 2000);
code
        30 end
algol; end:
end procedure MBadm;
```

4. Eksempel.

Følgende lille program, der viser anvendelsen af procedurerne MBstart og MBadm, trykker indholdet af visse blokke paa baandet med daaben-2027, 1. Bemærk specielt erklæringen af blokomraadet i ferritlageret, blok, samt begyndelsesadressens beregning.

```
begin comment udskrift af fejlblokke;
integer BLOK; i, j;
integer array synkro[1:4], blok[0:16, 0:7];
procedure MBstart(
        .
        .
        .
end procedure MBstart;
procedure MBadm(
        .
        .
        .
end procedure MBadm;
```

```
code
        60  blok
        0c  1
        0f  9
        28  BLOK           ; sæt beg.adr.
algol;
MBstart(3, 2027, 1, BLOK);
for i := 250, 251, 318, 328, 350, 351, 352, 355, 360, 362, 516, 521,
        528, 549, 573, 582 step 1 until 592, 617, 619, 673, 712, 765,
        767, 785, 789, 797, 798, 817, 829, 853 step 1 until 859, 878 do
  begin
    MBadm(1, 3, i);
    MBadm(2, 3, BLOK); comment læs blok i;
    trykvr;
    trykvr;
    tryk({-ndddd}, synkro[3] - 1); comment tryk bloknr;
    for j := 0, j + 1 while j < 16  $\wedge$  blok[j, 1]  $\neq$  0 do
      begin
        trykvr;
        tryk({-ndddd}, blok[j, 1], blok[j, 3])
      end for j
    end for i
end udskrift af fejlblokke;
```

Bent Bagger