

3

4. ÅRGANG

MAJ 1980

data lære

INDHOLD

Fletsortering

"Datalogi som støttefag"

Når man skal købe en mikrodatamat . . .

Anvendelsen af EDB i undervisningen - herunder datalære - ved Odense kommunale Skolevæsen

Et plottersystems udvikling

Udgivet af

FORENINGEN FOR DATALÆRE OG ANVENDELSE AF EDB I UNDERVISNINGEN

ITT 3343

markedets bedste dataterminal, nu også
med dansk karaktersæt



Den ideelle dataterminal til forsknings- og undervisningsmiljøer. Den utroligt driftsikre og stabile opbygning sikrer lang levetid med minimal vedligeholdelse.

• 132 karakterers skrivebredde • almindeligt papir • Buffer sikrer maksimal printhastighed • Margin kan sættes såvel fra linien som fra tastaturet • 7 x 9 matrix • Akustisk kobler til transmission over telefonnettet.

Nu: incl. danske tegn og akustisk kobler kr. 16.800,-
undervisningsrabat 840,-

kr. 15.960,- excl. moms.

ITT data

Standard Electric Kirk A/S, Fabriksparken 31, 2600 Glostrup. Telefon 02 45 00 45
Standard Electric Kirk A/S, Ane Stauingsvej 21, 87 00 Horsens. Telefon 05 62 33 11

FLETSORTERING

Unge Jørgensens problem

Isenkramfirmaet Jørgensen & Søn har et problem. På grund af nogle faglige møder omkring 1. november er en del fakturaer fra sidste uge af oktober ikke blevet lagt på plads i ringbindet med fakturaer for oktober måned. Viggo Jørgensen - også kaldet Unge Jørgensen - får af faderen, Sivert Jørgensen - Jørgensen Senior - tildelt den interessante opgave at få den sidste uges fakturaer sat ind i ringbindet på rette plads blandt de øvrige. Fakturaerne i det pågældende ringbind er ordnet efter kundenumre og indeholder ialt 60 fakturaer fra de tre første uger. Dyngen af fakturaer, som endnu ikke er sat på plads, indbefatter 22 fakturaer. Unge Jørgensen har flere muligheder for at løse opgaven. Han kan tage alle fakturaer ud af ringbindet og derpå stikke de nye fakturaer ind imellem dem én for én. Det gider han ikke!



Han kan også lade de allerede sorterede fakturaer blive siddende i ringbindet og iøvrigt benytte den lige nævnte fremgangsmåde. I så fald har han hele tiden god styr på de fakturaer, som er sat på plads, men må til gengæld åbne og lukke ringbindet for hver faktura, der skal sættes ind. Det gider han slet ikke! I stedet venter han, til Jørgensen Senior er gået til bestyrelsesmøde i Zero-klubben, og sætter sig så til at tænke over tingene (Jørgensen Senior betragter stillesiddende tænk-somhed med mistro som tegn på hang til ikke profitgivende teoretiseringen). Viggo Jørgensen skriver nogle kundenumre ned på et stykke papir. Så tænker han lidt. Derpå skriver han nogle andre kundenumre ned. Og tænker lidt igen. Pludselig giver det et ryk i ham. Han streger de sidst skrevne kundenumre ud, og skriver dem op på en anden måde. Så slår han et par streger på papiret, nikker eftertænksomt og går hen og tager fat på arbejdet med fakturaerne.

Hvad Unge Jørgensen skrev

Mens Viggo Jørgensen sorterer fakturaer, kan vi kigge på det, han har skrevet ned.

← 12 15 17 20 23 23 28 31 35 40
44 14 30 34 19 15 23
14 15 19 23 30 34 44

Den øverste række skal forestille en række kundenumre, som er sorteret, og repræsenterer dermed fakturaer, som er sat i ringbind. Den næste talrække forestiller kundenumre i den "tilfældige" orden, som fremkommer ved, at de udskrives, efterhånden som ordrerne indgår. Den nederste række af tal er tydeligvis et resultat af, at den midterste række er blevet sorteret. Viggo Jørgensen har altså arbejdet med en model af det problem han skal løse, og ved at betragte denne model, har han fundet ud af en metode til løsning af det reale problem.

- og hvad han tænkte

Den øverste talrække er sorteret og i denne sorterede række skal tallene i den anden række indsættes på en sådan måde, at resultatet bliver en ny, sorteret række. Samtidig skal det helst undgås, at der skal "rodes for meget rundt" i de allerede sorterede tal. Dette sidste skyldes en vigtig egenskab ved de reale objekter, der skal arbejdes med, nemlig at der er tale om papirer i et ringbind. Viggo får nu den idé - måske ved at se på de allerede sorterede numre - at man også kunne sortere den anden række af numre, hvilket han derpå gør. Ved at kaste en sidetanke til det reale problem, indser han, at det ikke vil være så farligt at sortere de fakturaer, der er blevet skrevet i sidste uge af oktober, idet der ikke er flere, end man kan lægge ud på et bord ved siden af hinanden. Ved at se på de to rækker af sorterede kundenumre, får han den afgørende idé, og det tilkendegiver han ved at tegne de to pile og derpå gå hen for at sætte fakturaerne ind i ringbindet. I et "genialt glimt" har han set følgende proces for sig:

Først ser man på de to forreste numre (12 og 14). Man vælger det mindste af disse og sætter det på plads. Nu står numrene 15 og 14 forrest. Det mindste af disse er 14, som derfor står for tur. Sådan går det videre, og en del af processen er afbildet her:

12 14 15 15 17 19 20 23 23 ← 12 15 17 20
14 15 19 23 ← 14 15 19 23

Den resulterende række er afbildet således, at man let kan se, om et element er kommet fra den øverste eller nederste række.

- og hvad han så gjorde

Efter således at have gennemtænkt problemet, er det let for Viggo Jørgensen at lægge sin arbejdsplan. Først lægger han de 22 løse fakturaer ud på

kundebordet - som er let at kende på kaffe- og portvinspletterne - og sorterer dem ved hjælp af en metode, han kender godt fra selskabslivet i Nr. Nusserød, hvor familien har domicil, nemlig den man bruger, når man sorterer en hånd med kort (se kortspilsmetoden, Datalærebladet, Februar 1978, side 12). Derpå tager han ringbindet med de allerede sorterede fakturaer, vender hele indholdet over til højre side og åbner ringene. Og nu bruger han den metode, som han netop har opfundet. Han ser på kundennummeret på de to fakturaer, der ligger øverst i hhv. bunken i ringbindet og bunken på bordet. Hvis den, der ligger øverst i ringbindets højre side, har det mindste kundennummer, vender han den blot over på venstre halvdel af ringene, og hvis det er den, der ligger øverst i bunken på bordet, som har det laveste kundennummer, stikker han den ind over ringene til venstre. Sådan bliver han ved, indtil én af bunkerne er tom. Resten af den anden bunke kan da blot sættes ind over ringene på én gang. Metoden viser sig at være så god, at Viggo forklarer Senior om den, og efter at have rystet tilbørligt på hovedet beslutter denne, at man for fremtiden vil bruge den systematisk, så man kun sætter fakturaer på plads én gang om ugen.

Øvelse 1

Afprøv metoden på to bunker af spillekort, der hver for sig er sorteret. Lad fx. den ene bunke indeholde 10 kort og den anden 7 kort.

Fletning

Den proces, som Viggo Jørgensen udfører med fakturaerne, kaldes også fletning. Man kan notere sig to ting i forbindelse med fletning:

- 1) Der er tale om to rækker, som hver for sig er sorteret.
- 2) Der ændres ikke noget på den indbyrdes rækkefølge af elementerne i de to rækker hver for sig.

Alle de sorteringsmetoder, vi indtil nu har undersøgt, har virket på tabeller, dvs. datamængder hvis enkelte elementer vi kan læse eller skrive i vilkårlig rækkefølge, og hvor vi får adgang til en komponent ved at angive dens indeks eller postnummer. Vi har da også i alle tilfælde benyttet, at man fx. kan ombytte elementer i sådanne datastrukturer. Hvis de data, man skal sortere, står skrevet i strukturer, hvis elementer man kun kan få adgang til forfra og i rækkefølge, altså fx. data-køer og sekventielle filer, kan sorteringer af de tidligere nævnte typer ikke anvendes. I sådanne tilfælde må man sætte sin lid til metoder, der bygger på fx. fletning. I det følgende skal en sådan metode gennemgås med henblik på sortering af sekventielle filer.

Sortering ved fletning

Lad os tænke os, at vi ønsker at sortere en sekventiel fil, som har dette indhold:

```
18 32|6 60|14 42 44 68|12 24 30 48|4 8 72|
  3 20 58|38 62
```

Man kan uden videre se, at en sådan fil kan deles op i en række delfiler, som hver for sig er ordnede. På billedet ovenfor er denne opdeling antydnet med lodrette streger. De enkelte ordnede delfiler kalder man for løb, og i vort eksempel har vi altså ialt syv sådanne løb. Vi opdeler nu filen i to nye filer, idet vi skriver hvert andet løb over i den ene fil og resten over i den anden:

```
18 32|14 42 44 68|4 8 72|38 62
  6 60|12 24 30 48|3 20 58
```

Derpå sammenfletter vi de enkelte løb i den ene fil med de tilsvarende løb i den anden til en ny sammenhængende fil - det overskydende løb i den øverste hjælpefil kopierer vi blot med over:

```
6 18 32 60|12 14 24 30 42 44 48 68|3 4 8 20 58 72|
38 62
```

Denne fil opdeles atter i parallelle løb:

```
6 18 32 60|3 4 8 20 58 72
12 14 24 30 42 44 48 68|38 62
```

Den næste fletning giver:

```
6 12 14 18 24 30 32 42 44 48 60 61|
3 4 8 20 38 58 62 72
```

Efter endnu en opdeling og en fletning fremkommer den fuldstændigt sorterede fil.

Øvelse 2

Opskriv resultatet af den sidste opdeling og den endelige fletning. Find antallet af løb efter hver fletning. Hvor mange løb indeholder den fuldstændigt sorterede række?

Som eksemplet viser har vi brug for ialt tre filer til processen: en hovedfil og to hjelpefiler. Endvidere kan vi se, at processen i det væsentlige består af en række opdelinger med påfølgende fletninger. Inden vi kan begynde at skrive algoritmerne for processen, må vi også gøre os klart, hvorledes vi kan formulere afslutningen af et løb ved hjælp af en relation, som et datamatisk system kan afprøve. Ser vi atter på eksemplet ovenfor, konstaterer vi, at det sidste element i et løb er karakteriseret ved, at det efterfølges af et element, der er mindre, eller af fildlut. Det er klart,

at processen er afsluttet, når antallet af løb er lig med 1. Idet vi kalder hovedfilen for C og hjælpefilerne for hhv. A og B, kan vi nedskrive hoved-proceduren for processen samt procedurerne OPDEL og FLETNING for de to vigtigste del-processer.

PROCEDURE FLETTEPROGRAM

GENTAG

gør filen A og filen B klar til skrivning

gør filen C klar til læsning

OPDEL

luk alle filer

gør filen A og filen B klar til læsning

gør filen C klar til skrivning

nulstil løbtælleren

FLETNING

luk alle filer

INDTIL løbtælleren er lig med 1

SLUT FLETTEPROGRAM

PROCEDURE OPDEL

GENTAG

KOPIERLØB(fra C til A)

HVIS ikke filen C er slut SÅ

KOPIERLØB(fra C til B)

SLUT-HVIS

INDTIL filen C er slut

SLUT OPDEL

PROCEDURE FLETNING

GENTAG

FLETLØB //flet et par af løb//

tæl løbtælleren én op

INDTIL filen A eller filen B er slut

MENS filen A ikke er slut UDFØR

//et overskydende løb tilbage i A//

KOPIERLØB(fra A til C)

tæl løbtælleren én op

SLUT-MENS

MENS filen B ikke er slut UDFØR

//et overskydende løb tilbage i B//

KOPIERLØB(fra B til C)

tæl løbtælleren én op

SLUT-MENS

SLUT FLETNING

tekniske detaljer. Vi er ved at sortere filer, og filer består af poster. Indholdet af en enkelt post kan efter omstændighederne være ret så broget. Posten kan bestå af mange felter med mange forskellige informationer. Et af disse felter må imidlertid indeholde den nøgle, man sorterer efter. Det kan fx. være et felt med et kundenummer, et felt med et lønningsnummer, eller et felt med et efternavn. For ikke at gøre denne gennemgang af flet-sortering altfor afhængig af en særlig anvendelse, vil jeg nøjes med at tale om en nøgle (key), som indeholder den information, der danner grundlag for sorteringen. Læseren må så selv modificere programmet efter sit eget behov og herunder selv sørge for, at nøgleinformationen for hver af de tre filer bliver læst og skrevet på rette sted i programmet. I det følgende vil jeg anvende betegnelsen nøgle(X) for værdien af det sidst læste felt med nøgleoplysningen fra filen X. Hvis man altså fx. benytter en persons efternavn som nøgle for sorteringen, må man sørge for, at nøgle(X) indeholder efternavnet fra den sidst læste post i filen X, hvis sorteringen skal virke korrekt. Endvidere vil jeg arbejde med en buffer, som kan indeholde den næstsidste værdi af nøgle(X), og som kan bruges fx. i det tilfælde, hvor man skal undersøge en relation mellem to på hinanden følgende nøgleoplysninger. I såvel FLETLØB som KOPIERLØB kaldes proceduren KOPIER, som på en måde udfører "det grove" for de to procedurer, fra hvilke den bliver kaldt. Denne procedure kopierer en post fra én fil, som vi vil kalde X, til en anden, som vi vil kalde Y, og dernæst undersøger den, om et løb er afsluttet eller ej. Proceduren kan beskrives således:

PROCEDURE KOPIER(fra X til Y)

buffer:=nøgle(X)

SKRIV post(X) i filen Y

LÆS post(X) i filen X

udtag nøgle(X)

HVIS filen X er slut SÅ

slutløb:=sand

ELLERS

slutløb:=(buffer>nøgle(X))

SLUT-HVIS

SLUT KOPIER

Øvelse 3

Afprøv ved en skrivebordskørsel de tre algoritmer, idet der anvendes de data, som er brugt i eksemplet ovenfor. Kontrollér, at løbtælleren stemmer med de i øvelse 2 fundne værdier.

* * *

Procedurerne

Vi skal nu til at se nærmere på de procedurer, der hedder FLETLØB og KOPIERLØB, og vi kommer ikke længere uden om at overveje et par

Ved procedurens start har vi den senest læste post fra filen X - post (X) - stående i arbejds-lageret, og nøgle (X) indeholder værdien af det felt, vi sorterer efter. Nøgleoplysningen tildeles den variable buffer til senere brug. Derpå finder den egentlige kopiering sted, idet hele posten fra X skrives i filen Y, hvorpå en ny post læses i filen X. Denne læsning giver naturligvis også nøgle (X) en ny værdi, nemlig den fra den nye post. Der er nu flere muligheder: Det kan vise sig, at filen X er slut, og så er et løb naturligvis

også slut, og det noterer vi i den Boolske variabel slutløb ved at tildele den værdien sand. Hvis filen ikke er slut, er der atter to muligheder: enten er et løb slut, eller også er det ikke slut. Det afgøres af sandhedsværdien af det Boolske udtryk: $\text{buffer} > \text{nøgle} (X)$. Vi husker, at den variable buffer indeholder nøglen til den post, som netop er blevet skrevet over i filen Y, og at nøgle (X) indeholder nøglen til den post, som netop er blevet læst i filen X. Dersom det viser sig, at denne sidst læste nøgle er mindre end den foregående, betyder det netop, at et løb er afsluttet. Hvis det derimod ikke er tilfældet, fortsætter løbet. Den rigtige sandhedsværdi bliver tildelt den variable slutløb. Når vi forlader KOPIER, "ved" den variable slutløb altså, om et løb er slut eller ej, og denne oplysning kan vi bruge i de øvrige procedurer.

Efter at have skrevet KOPIER er det en let sag at skrive procedurerne KOPIERLØB og FLETLØB:

```
PROCEDURE KOPIERLØB(fra X til Y)
  GENTAG
    KOPIER(fra X til Y)
  INDTIL slutløb
SLUT KOPIERLØB

PROCEDURE FLETLØB
  GENTAG
    HVIS nøgle(A) < nøgle(B) SÅ
      KOPIER(fra A til C)
    HVIS slutløb SÅ
      KOPIERLØB(fra B til C)
    SLUT-HVIS
  ELLERS
    KOPIER(fra B til C)
    HVIS slutløb SÅ
      KOPIERLØB(fra A til C)
    SLUT-HVIS
  SLUT-HVIS
  INDTIL slutløb
SLUT FLETLØB
```

Bemærk, hvorledes den fra KOPIER styrede variabel slutløb anvendes i de to procedurer.

Program FLETSORTERING

Som bilag til denne artikel findes et COMAL program, som udfører sortering ved fletning. Programmet følger nøje de angivne algoritmer, men på grund af visse mangler i COMAL, har det været nødvendigt at føje nogle hjælpesætninger til de oprindelige algoritmer for at få processerne udført af et COMAL-system. De tre filvariable A, B og C er realiseret som numeriske variable med værdierne hhv. 1, 2 og 3. De fleste af de nuværende COMAL versioner tillader desværre ikke parameter-overføring mellem procedurer, og i program-

met er denne overføring af værdier ved parametre erstattet af simple tildelinger, som foretages inden kaldet af proceduren. Vi kan fx. se på PROC OPDEL, hvor man finder tildelingerne: LET X=C (linje 300) og LET Y=A (linje 320) inden kaldet af proceduren KOPIERLØB. Når vi via KOPIERLØB når frem til KOPIER, har X altså værdien 3 og Y værdien 1, og det betyder, at der kopieres fra filen C ind i filen A, hvilket netop er det, vi ønsker. Ved at give X og Y andre værdier, udføres kopieringen med andre afsender- og modtagerfiler. For at holde styr på posterne fra de tre filer, er vi også nødt til at bruge nogle strukturerede buffere, og til dette formål har jeg valgt at bruge fire tabeller: NAVN\$, ADR\$, POSTDSTR\$ og NØGLE\$. Hver af dem har tre komponenter: 1. komponenterne bruges som buffere for poster fra A, 2. komponenterne som buffere for poster fra B, og 3. komponenterne som buffere for poster fra C. Den valgte poststruktur skal blot tjene et demonstrationsformål, og læseren må selv indrette buffere til sit specielle formål. Man skal blot huske at få NØGLE\$ tildelt værdi af det felt, hvis indhold danner grundlag for sorteringen. Der er desuden indsat en "service-procedure" LISTFIL, som kan bruges ved afprøvelsen af programmet. Programmet startes fra afsnittet i linje 1010 - 1120, hvor der også kan indtastes test-værdier til filen C. Også dette afsnit må naturligvis indrettes efter det aktuelle behov.

Afslutning

Hermed slutter artikelsen: ORDEN SKAL DER TIL. Min væsentligste kilde til de i serien gennemgåede algoritmer har været:

N. Wirth: Algorithms + Data Structures = Programs.
(Prentice-Hall 1976)

Det er som bekendt Wirth, der har defineret programmerings-sproget Pascal, og det var med dette sprog som forbillede, COMAL i sin tid blev defineret.

Det kan derfor heller ikke undre, at COMAL programmerne i artiklerne har en påfaldende lighed med Wirths Pascal-programmer. Af andre bøger, som kunne have interesse for mine læsere, kan jeg nævne:

E. Horowitz & S. Sahni: Fundamentals of Data Structures (Pitman 1977)
Peter Naur: Concise Survey of Computer Methods (Studentlitteratur 1974)
Kenneth L. Bowles: Problem Solving Using Pascal (Springer Verlag 1977)

```

0010 PROC LISTFIL
0020 OPEN FILE (C,3),"CFIL"
0030 READ FILE (C),NAVN*(C),ADR*(C),POSTDSTR*(C)
0035 LET NØGLE*(C)=NAVN*(C)
0040 WHILE NOT EOF(C) DO
0050 PRINT NØGLE*(C),
0060 READ FILE (C),NAVN*(C),ADR*(C),POSTDSTR*(C)
0065 LET NØGLE*(C)=NAVN*(C)
0070 ENDWHILE
0080 PRINT
0090 CLOSE FILE (C),
0100 ENDPROC LISTFIL
0110 REM //-----//
0120 PROC KOPIER
0130 LET BUFFER*=NØGLE*(X)
0140 WRITE FILE (Y),NAVN*(X),ADR*(X),POSTDSTR*(X)
0150 READ FILE (X),NAVN*(X),ADR*(X),POSTDSTR*(X)
0155 LET NØGLE*(X)=NAVN*(X)
0160 IF EOF(X) THEN
0170 LET SLUTLØB=TRUE
0180 ELSE
0190 LET SLUTLØB=(BUFFER*>NØGLE*(X))
0200 ENDIF
0210 ENDPROC KOPIER
0220 REM //-----//
0230 PROC KOPIERLØB
0240 REPEAT
0250 EXEC KOPIER
0260 UNTIL SLUTLØB
0270 ENDPROC KOPIERLØB
0280 REM //-----//
0290 PROC OPDEL
0300 LET X=C
0310 REPEAT
0320 LET Y=A
0330 EXEC KOPIERLØB
0340 IF NOT EOF(C) THEN
0350 LET Y=B
0360 EXEC KOPIERLØB
0370 ENDIF
0380 UNTIL EOF(C)
0390 ENDPROC OPDEL
0400 REM //-----//

```

```

0400 REM //-----//
0410 PROC FLETLØB
0420 LET Y=C
0430 REPEAT
0440 IF NØGLE*(A)<NØGLE*(B) THEN
0450 LET X=A
0460 EXEC KOPIER
0470 IF SLUTLØB THEN
0480 LET X=B
0490 EXEC KOPIERLØB
0500 ENDIF
0510 ELSE
0520 LET X=B
0530 EXEC KOPIER
0540 IF SLUTLØB THEN
0550 LET X=A
0560 EXEC KOPIERLØB
0570 ENDIF
0580 ENDIF
0590 UNTIL SLUTLØB
0600 ENDPROC FLETLØB
0610 REM //-----//
0620 PROC FLETNING
0630 REPEAT
0640 EXEC FLETLØB
0650 LET L=L+1
0660 UNTIL EOF(A) OR EOF(B)
0670 LET Y=C
0680 WHILE NOT EOF(A) DO
0690 LET X=A
0700 EXEC KOPIERLØB
0710 LET L=L+1
0720 ENDWHILE
0730 WHILE NOT EOF(B) DO
0740 LET X=B
0750 EXEC KOPIERLØB
0760 LET L=L+1
0770 ENDWHILE
0780 ENDPROC FLETNING
0790 REM //-----//

```

```

0800 PROC FLETTEPROGRAM
0830 REPEAT
0840 OPEN FILE (A,1),"AFIL"
0850 OPEN FILE (B,1),"BFIL"
0860 OPEN FILE (C,3),"CFIL"
0870 READ FILE (C),NAVN*(C),ADR*(C),POSTDSTR*(C)
0875 LET NØGLE*(C)=NAVN*(C)
0880 EXEC OPDEL
0890 CLOSE
0900 OPEN FILE (A,3),"AFIL"
0910 OPEN FILE (B,3),"BFIL"
0920 OPEN FILE (C,1),"CFIL"
0930 READ FILE (A),NAVN*(A),ADR*(A),POSTDSTR*(A)
0935 LET NØGLE*(A)=NAVN*(A)
0940 READ FILE (B),NAVN*(B),ADR*(B),POSTDSTR*(B)
0945 LET NØGLE*(B)=NAVN*(B)
0950 LET L=0
0960 EXEC FLETNING
0970 CLOSE
0980 UNTIL L=1
0990 ENDPROC FLETTEPROGRAM
1000 REM //-----//

```

```

1005 RANDOMIZE
1010 LET A=1; B=2; C=3
1020 LET TRUE=1; FALSE=0
1022 DIM NAVN*(3,30),ADR*(3,20),POSTDSTR*(3,20)
1024 DIM NØGLE*(3,30)
1030 OPEN FILE (3,1),"CFIL"
1040 FOR I=1 TO 10
1050 INPUT "NAVN: ",NAVN*(C)
1052 INPUT "ADRESSE: ",ADR*(C)
1054 INPUT "POSTDISTRIKT: ",POSTDSTR*(C)
1060 WRITE FILE (C),NAVN*(C),ADR*(C),POSTDSTR*(C)
1070 NEXT I
1080 CLOSE
1090 EXEC LISTFIL
1100 EXEC FLETTEPROGRAM
1110 EXEC LISTFIL
1120 END

```

METRIC PRÆSENTERER:

METRIC ABC80

80'ernes Mikrodatamat- system



METRIC ABC 80 - MIKRODATAMATEN til morgendagens undervisning!

- * Let at lære - nem i brug
- * Lynhurtigt og avanceret semikompilerende BASIC, det internationale sprog
- * 16K bruger-RAM (opt. 32K)
- * Kassettebåndstation til program- og datalagring
- * Skandinavisk kvalitetsprodukt
- * 16K ABC 80 BASIC i ROM, altid klar til brug, ingen besværlig indlæsning
- * 12" dataskærm med grafik
- * Professionelt tastatur
- * Har udgang for modem, kan også bruges som terminal
- * Pris: 8.990,- excl. moms

DATAUDSTYR FRA SG **METRIC** AIS

DATAAFDELINGEN, SKODSBORGVEJ 305, 2850 NÆRUM, TLF. (02) 80 42 00

»Datalogi som støttfag«

Af *Gustav Leunbach*,
Danmarks pædagogiske institut.

I studiet til cand psych ved Københavns Universitet skal der inden forprøven afsluttes være bestået en prøve i statistik og metodelære. Som én blandt flere måder at gennemføre denne prøve har jeg i en årrække som ekstern lektor tilbudt et kursus med titlen "statistik fra databehandlings synspunkt" med prøven indbygget som afslutning af et individuelt arbejde.

De fysiske rammer om kurset var et undervisningslokale i universitetets Amagerafdelings terminalsektor med en skærmterminal med tre fjernsynsmonitorer som simultant viste hvad der stod på skærmen, således at hele gruppen kunne følge forløbet. Endvidere blev programmer og resultater udskrevet på terminalsektionens linjeskriver, dels i timerne, dels i en af læreren sammenredigeret form til afhentning dagen efter hver time. Terminalsektionen er tilsluttet edb-centret RECKU's Univac 1100.

Varigheden var et universitetssemester på 5 mdr. I den første halvdel af tiden var der fællesundervisning 2 timer om ugen, i resten af tiden individuelt arbejde enkeltvis eller i små hold med jævnlige lærerkontakter, samlet omfang skønsmæssigt 3-4 timer om ugen.

Emnet for fællesundervisningen var behandlingen af et psykologisk prøvemateriale som forelå kodet på papirstrimmel. Kodningen var i hovedsagen bestemt af en psykologisk klassifikation af reaktioner med betegnelser af mønstret bogstavtal-bogstav, i f. ex. S2A; da en enkelt kode dominerede billedet, indførtes en kode af mønstret tal-bogstav til at angive en ubrudt række på en bestemt længde af denne kode. Disse kodningstekniske overvejelser var det første emne i undervisningen.

Der blev forelagt et ufærdigt algol-program til indlæsning af det kodede materiale, og sideløbende med en gennemgang af grundbegreberne i programmeringssproget algol blev dette program gennemgået og sammenholdt med strukturen af inddata. Studenternes første opgave var at komplettere programmet så meget at det kunne indsættes og eksekveres, den næste at indsætte udskrivningsordrer som viste at det også fungerede.

Som det vil fremgå af beskrivelsen af data drejede programmet sig i høj grad om behandling af tekstvariable, herunder overførsel af talkoder i tekstvariable til heltalsvariable; de numeriske operationer indskrænkede sig stort set til + og - på heltalsvariable; evt. kunne decimal-

tal komme ind hvis en optælling af forekomster af bestemte koder blev efterfulgt af procentudregning. (Det samme galdt stort set om de programmer studenterne udarbejdede holdvis.)

Programmet for indlæsning af data blev efterhånden udvidet til at omhandle kontrol af det korrekte antal koder pr. forsøgsperson, optælling af forekomsten af koder, udskrift af kodebetegnelser og deres hyppigheder i tabelform, samme udskrift efter alfabetisk sortering af koderne mm. De forskellige udvidelser blev redigeret kollektivt i grupper og udført af en deltager på skærmterminalen, idet RECKU's editingsprogrammet også blev gennemgået - ligesom de nødvendige dele af RECKU's styresystem.

Fra begyndelsen anvendtes korrekturlæste inddata, men på et senere tidspunkt indsattes ikke-korrekturlæste data med henblik på en diskussion af fejlsøgningsmuligheder i programmet. Fejlsøgning diskuteredes dels udfra synspunktet om den nødvendige fejlretning for at programmet overhovedet kunne køre, dels fra forholdet mellem fejlprocent og arbejdstid til korrektion.

I timerne suppleredes skærbillederne (med fjernsynsmonitorer) efter behov - og efter de praktiske muligheder - med linjeskriverudskrifter, og efter hver ugentlig dobbelttime redigerede læreren en udskrift som lå til afhentning dagen efter. Det almindelige mønster for denne udskrift var

Programmet ser nu sådan ud

...

og giver følgende udskrift

...

Lav de tilføjelser som også vil give os følgende oplysninger

...

og dette er udgangspunktet for næste times drøftelser.

Til de selvstændige øvelser i små hold efterlyste jeg egne materialer hos studenterne, og enkelte gange var der en som havde et materiale som hun/han benyttede som sit øvelsesmateriale og derved fik en vis bearbejdelse udført "gratis" af et materiale som f. ex. skulle bruges som opgave ved et andet fag i studiet.

Men ellers havde jeg et antal materialer klar med et psykologisk-pædagogisk indhold - og i praksis viste dette indhold sig at være den væsent-

ligste faktor i holdenes valg mellem materialerne; deres datalogiske struktur kunne deltagerne dårligt vide noget om.

Gangen i en holdopgave var: indhulning (normalt på hulkort) af en meget begrænset del af materialet; efterfølgende diskussion af manuelle og datamatiske kontrolmuligheder for hulning, herunder letlæselige udskriftsformater; programmering af indlæsningen med de forømtalte kontroller; programmering af simple optællinger og tabeludskrifter med anvendelse af et større indkodet materiale. Materialerne var i nogle tilfælde rent numeriske men omfattede ofte tekstdata, og tabeludskriften omfattede i alle tilfælde kombinerede tekst- og taludskrifter.

Kriteriet for beståelse af prøven var i hovedsagen tidsmæssigt, så og så mange timers reelt arbejde, under møder med læreren og i det arbejde holdet efter mit skøn måtte have præsteret inden møderne. Arbejdet nåede normalt aldrig frem til egentlig brugbare resultater, og holdene fik altid tilbud om at føre det videre efter det tidspunkt hvor jeg meddelte at de havde bestået; enkelte af dem tog mod tilbuddet.

Problemet med at bedømme de stærke og de svage deltagere på holdet har jeg ikke nogen genial løsning på; i de timer jeg havde med holdet prøvede jeg at aktivere alle, og jeg kan kun sige at hvis ikke alle har præsteret lige meget arbejde, så har i hvert fald alle lært noget om emnet.

De "samfundsmæssige aspekter"

Både de ugentlige redigerede udskrifter og den mini-lærebog jeg uddelte (afsnit om algol, editering og styresystemer) var linjeskriverfremstillet, og dette gav anledning til samtaler dels om HK's og typografernes holdning til de nye teknikker, dels om forholdet mellem teknikken og de mennesker der skal bruge dens produkter (manglende æ, ø og å, manglende små bogstaver osv.)

Derudover var der lejlighedsvis diskussioner om forholdet mellem forskning og datamatik; dels var materialerne udgangspunkt for sammenligninger mellem resultaterne af manuel og datamatisk behandling, dels kunne det dreje sig om forskernes (psykologernes) kendskab til datalogi og forholdet mellem "gør det selv" og det at være i lommen på eksperterne.

Programmeringssprog

Alle programmer, både de læreren leverede og studenterne udbyggede, og de studenterne opbyggede fra grunden, var skrevet i det på RECKU anvendte algol-sprog, Norsk Regnecentrals Nualgol, som har gode tekstbehandlingsmuligheder, og hvor udskrift af programteksten forsynes med en nummerering i marginen af alle "begin" og "end".

I sammenligning med det programmeringssprog algol plejer at sammenlignes med, fortran, er efter

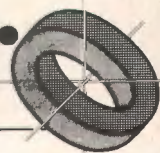
min mening algol langt klarere opbygget og dermed langt bedre egnet til en undervisning som prøver at kombinere den håndværksmæssige læring af sprogelementerne med en rudimentær forståelse af processerne i datamaten.

Fordelene ved fortran er to: at der skal væsentlig færre ændringer til for at overføre et fortran-program fra en maskine til en anden, og at samme opgave på de fleste anlæg køres billigere i fortran end i algol.

Hvis jeg skulle føre kurset videre ville jeg overveje at gå over til simula som er en udbygning af algol med helt omstruktureret tekstbehandlings-system med standardfunktioner i en del af de tilfælde hvor vi var nødt til at skrive egne programmer; simula nærmer sig iøvrigt samme grad af standardisering som fortran; endvidere kan simula behandle store og små bogstaver (forudsat at ens linjeskriver også kan).

★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★

Medlems-Service



Publikationer, som kan fås eller lånes ved henvendelse til SEL, CAI-laboratoriet, Svendborgvej 10, 5762 Vester Skerninge, tlf. (09) 24 22 82, onsdag 9-16.

Fra seminaret i Tønder, marts 1980:

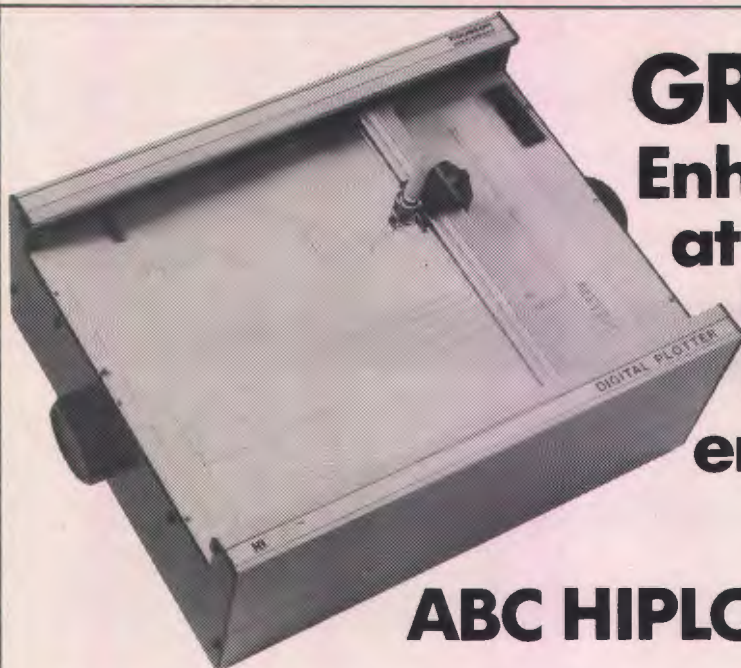
Better Programming with ELAN. Indlæg af Jochen Liedtke, Bielefeld. 13 sider.

Projekt Informatikkurse. Unterrichtsmaterial zu einem Grundkurs in Informatik. Udarbejdet ved Gymnasium Sylt. 34 sider.

EUMEL - Benutzerhandbuch. Af Rainer Hahn et al., Universität Bielefeldt. 129 sider.

Tilsendt:

edb i danske skoler. Studiereise mars 79. Norsk rapport udarbejdet af Jan Wibe og Helge S. Rasmussen.



GRAFIK!

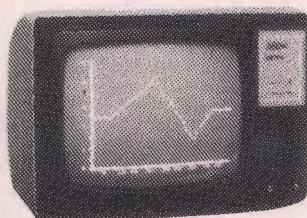
Enhver ved
at grafisk
kommuni-
kation
er bedre!

ABC HILOT-Plotter

- * Udskrivning af kurver, tegninger, bogstaver, tal m.m.
- * Størrelse: A4
- * Opløsning: 0,1 mm
- * Tilsluttes den indbyggede V-24 udgang
- * Driverprogram og kabel medfølger
- * Pris: 9.600,- excl.moms

Måling og styring

Længde
Masse
Temperatur
Hastighed
Spænding
Strøm
Frekvens



- * Til fysik og kemiforsøg
- * Måling, registrering og styring
- * Alle interfacekort kan styres fra ABC 80 BASIC, ingen besværlig assemblerprogrammering
- * Til langtidsforsøg
- * Let at anvende, selv for ikke-teknikere
- * Interfaces til enhver måling (A/D, D/A, PIO, IEEE, RELÆ m.m.)

DATAUDSTYR FRA ^{SC} METRIC ^{AIS}

DATAAFDELINGEN, SKODSBORGVEJ 305, 2850 NÆRUM, TLF. (02) 80 42 00

Når man skal købe en mikrodatamat . . .

Af Carsten Hansen

Denne og en følgende artikel har til formål at give et overblik over, hvilke overvejelser, man bør gøre sig, når man står over for køb af et mikrodatamatanlæg, samt at give en beskrivelse af nogle af de mere interessante produkter, der findes på markedet. En typisk grundkonfiguration udgøres i denne sammenhæng af en CPU med lager, en skærm, et tastatur og en kassettebåndoptager, altså et anlæg i 10.000 kroners-klassen.

Skærm

Skærmen må ikke være for lille, og der skal være passende linjelængde og antal linjer. Mange fabrikater har 24 linjer á 40 tegn, hvilket bestemmer i underkanten. Ligeegyldigt, hvilket programmeringssprog, man benytter, bruges 40 tegn i en linje hurtigt op, ikke mindst, hvor man benytter indrykning på linjen til at markere strukturel opbygning af programmet. For lille linjeantal er også et stort handicap ved udvikling af store programmer, hvis man ikke råder over en printer.

At tegnene skal være letlæselige, kan synes indlysende, men der er naturligvis en konflikt mellem tegnstørrelse og antal tegn pr. linje.

Løvrigt bør man erindre, at skærmen er en vigtig del af "arbejdspladsen". Ikke alle skærme er lige behagelige for øjnene under flere timers arbejde.

Tastatur

Efterhånden er tastatur af sædvanlig kvalitet for elektrisk skrivemaskine ved at være standard på området. Der er således ingen grund til at afstå fra krav om, at skrivning skal kunne foretages i sædvanligt tempo og på samme nemme og behagelige måde, som når man sidder ved en elektrisk skrivemaskine. Eksempelvis er repeterende taster for visse tegns vedkommende ret generende at skulle undvære.

Dansk alfabet med æ, ø og å er forbundet med den ulempe, at nogle af de originale tegn må udgå. Ofte er disse bl. a. skarpe parenteser, hvilket f. eks. er en ulempe, hvis maskinen skal bruges til kørsel af PASCAL-programmer.

Båndoptager

Efter introduktionen af den nye digitalrecorder (som i kraft af sin hurtige acces-tid næsten kan fungere som et traditionelt baggrundslager, såfremt den organiseres med et filsystem),

virker kassettebåndoptageren i sædvanlig udførelse noget antikveret. Man må formode, at alle kassettebåndoptagere med tiden erstattes af digitalrecordere; ikke mindst uden båndtæller er kassettebåndoptageren ret besværlig at arbejde med.

Floppy disk (diskette)

Skal mikrodatamaten benyttes i forbindelse med store programmer eller store datamængder, er anskaffelse af en diskette uomgængelig. Her har en dobbelt-diskette-station væsentlige driftmæssige fordele frem for enkelt-diskette-stationen (uden at koste det dobbelte), ikke mindst, at den muliggør kopiering fra en disk til en anden. - Typiske diskstørrelser er 80-100 kbyte pr. plade.

Printer

Valg af printer afhænger af krav til skriftkvalitet og driftsikkerhed. Ønsker man blot en mulighed for en samlet listning af et program i programudviklingsfasen, kan en termoprinter være en nogenlunde god (og billig!) løsning. Ellers bør man ofre en matrix-printer eller typehjulprinter på formålet. Flere sådanne kan fås med mulighed for forskellige skrifttyper (f. eks. via programmell eller typehjulskiftning).

Da udgiften til en kvalitetsprinter ofte udgør en væsentlig del af anlæggets samlede pris, kan det være værd at overveje at købe en printer, der har opkoblingsmuligheder til flere systemer (f. eks. via V24-port).

Basisprogrammell

Vigtigst her er naturligvis, hvilke(t) programmeringssprog, maskinen giver mulighed for at benytte. Erfaringerne viser, at fabrikaterne på dette område sjældent er i stand til at holde, hvad der i salgssituationen blev lovet om "fremtidige udvidelser". Man må derfor være forberedt på, at man i en lang periode efter anskaffelsen må leve med de(t) programmeringssprog, som var til rådighed fra starten.

Alle maskiner har en BASIC-version, nogle COMAL og enkelte PASCAL.

Især BASIC findes i utallige versioner. I USA har man efterhånden fået udviklet en nogenlunde tålelig version (CBASIC), som bl. a. har lange variabelnavne. Der kan være kommercielle fordele ved BASIC (der findes utallige færdigudviklede programmer), men det virker på mig

begrænsende og perspektivløst at lade sig nøje med BASIC på undervisningsområdet, i hvert fald i datalære-sammenhæng.

På COMAL-området er der håb om en vis grad af standardisering (COMAL80), og med PASCAL går man i hvert fald ikke fejl af UCSD-versionen (fra University of California at San Diego, ved Ken Bowles). (Også dens editor- og filsystem er godt og velgennemprøvet!)

Af det øvrige basisprogrammel bør man hæfte sig ved editor (og editeringsmuligheder i almindelighed, f. eks. under interaktiv kørsel) og filsystem. Nogle vil sikkert også finde, at en god assembler er af interesse.

Service

De erfaringer, man indtil nu har indhøstet med mikro-elektronik-teknologien, giver grund til at formode, at produkterne har praktisk taget ubegrænset levetid. Såfremt man ikke hurtigt konstaterer fejl, synes der derfor at være alle chancer for, at service kun i ringe grad vil være påkrævet og da sædvanligvis i form af udskiftning af et helt modul (f. eks. et defekt printkort), ofte til en overkommelig pris.

Med den stigende anvendelse af nævnte teknologi også på de ydre enheders område (både disketter og printere), kan man håbe på forbedrede forhold også her, hvad angår service. Imidler-

tid er man indtil videre ret afhængig af, at der er troværdige reparationsmuligheder for de mekaniske og elektroniske dele.

Udvidelsesmuligheder, herunder tilslutning til diverse ydre enheder

I denne forbindelse er det især udvidelser af RAM-lageret, der påkalder sig interesse. Da der (endnu!) kun er tale om 8-bits-maskiner, er 64 K øvre grænse for lagerstørrelse. Afgørende er det også, om der tales om arbejdslager (dvs. den del af lageret, som brugeren har til rådighed), idet basisprogrammet i en del konfigurationer (bl.a. dem uden ROM) optager en stor del af den eksisterende RAM.

Vær også opmærksom på, at når man udvider sit anlæg med diskette eller printer, beslaglægger den tilhørende driver tit en del af det forhåndenværende arbejdslager. Dette "glemmes" ofte ved præsentationen af vidunderet, ligesom prisen på driveren!

Skal man bruge anlægget i industriel sammenhæng (eller i fysikundervisning), kan det være af afgørende betydning, at der eksisterer printkort, som gør det muligt at tilkoble specielle (måle-) apparater.

At det er vigtigt at tænke på udvidelsesmuligheder allerede fra begyndelsen, kan man overbevise sig om ved at studere prislistor for forskellige fabrikater for "ens" udvidelser!

Anvendelsen af EDB i undervisningen - herunder datalære - ved

Odense kommunale Skolevæsen

De fleste, der arbejder med en datamaskine - størrelsen underordnet -, kender utvivlsomt til fornemmelsen af at blive tryllebundet heraf. Mener man, at udtrykket er for stærkt, så kender man i hvert fald det fænomen, at man lige skal nå at lave det og det. Inden man ser sig om, er der gået mindst det dobbelte af den tid, man fra starten havde estimeret. Som underviser må man hurtigt erkende den motivation, der ligger i at arbejde ved et tastatur til en Teletype, data-skærm eller microdatamat. Er man nået til denne erkendelse, og har man overstået "programmørens" første trin med programmer til andengradslikninger, printal, CPR-nr. o. lign., melder der sig en idé om at kunne anvende datamaskinen til andet og mere. Man vil gerne kunne anvende den i undervisningen, men hvordan? Typisk er vel - især for folkeskolens vedkommende - at der eksisterer nogle områder, hvor det er nødvendigt at træne

basal viden, f. eks. indenfor områderne regning og dansk. Denne træning kan gribes an på mange måder, herunder at inddrage datamaskinen som hjælpemiddel, hvor der også er mange pædagogiske veje, der kan betrådes. En af de veje, der relativt let kan betrådes, er udformningen af programmel i drill & practice form, hvor eleven på en terminal får stillet nogle spørgsmål, der besvares - eleven får at vide, om det er rigtigt eller forkert, og der gås videre til næste spørgsmål, uden at et evt. forkert svar behandles med det samme, ved at der f. eks. er indlagt nogle for-greningsmuligheder i programmet.

Begyndelsen

Opgaver af drill & practice-typen er ikke lige velsete i alle pædagog-kredse. Jeg vil gerne forsvare denne form. Vi har ved skolevæsenet arbejdet med denne type programmer siden 1974

og - synes vi - haft succes hermed, idet man må erindre, at arbejder man med drill & practice-øvelser, må man tage skyldigt hensyn til det emne, der behandles, og der må finde en eller anden form for efterbehandling sted. Emner, der har vist sig velegnede, er f. eks. taltræning i bred almindelighed (de 4 regningsarter) og dansk. Graden af drill & practice kan være forskellig, spændende fra det mest primitive til de mere integrerede øvelser. En faktor, der i denne henseende spiller en stor rolle, er elevernes motivationsgrad. Er eleven motiveret i en indlæringsituation, vil der normalt komme et positivt udbytte, eller sagt på en anden måde: "Hvad der er lært ved leg, bliver hængende", hvor leg skal tages i sin positive betydning og ikke forveksles med pjat. Desuden er der de positive sider for den, der laver programmer af denne type som en begyndelse, at der indhøstes nogle værdifulde erfaringer og ny viden, der kan anvendes i mere avancerede former for DFU (datamaskine formidlet undervisning). Det element, vi primært kan spille på i folkeskolen, er den rent ud sagt forbløffende og vedblivende motivation, eleverne har i relation til mediet.

Forsøgets start

Ud fra vore første erfaringer med drill & practice programmer i specielt taltræning ønskede vi i 1975-76 at begynde et forsøg i vore specialklasser i samarbejde med folkeskolens forsøgsråd indenfor fagene regning og dansk. Som nævnt i den foregående artikel er specialklasser beregnet for elever med generelle indlæringsvanskeligheder, og med den stigende integration af elever i normalklasserne, der finder sted i disse år, er de elever, som i dag findes i vore specialklasser meget "tunge". Det er elever, der kræver en meget individuel undervisning, og tingene skal gentages (tålmodigt!) igen og igen. Netop her kan datamaskinen komme ind i billedet og udfylde en væsentlig rolle. Det er ikke enhver lærer beskåret altid at være lige ligevægtig og vedblive med at bevare samme tålmodighed, når en ting skal gentages for 117. gang overfor en elev.

Regning

Men hvorledes skulle vi nu tilrettelægge tingene til datamaskinen? Da vi havde besluttet os for fagene dansk og regning, måtte vi først se på, hvad lærerne ønskede set i relation til undervisningen. I regning ønskede man programmer, der udover at give en træning, kunne anvendes til indlæring og til håndtering af akut opståede problemer. Derfor blev programmerne til regning lavet som enkeltstående programmer, der skulle kaldes ved navn hver gang, de skulle anvendes, fordi talområderne skulle fastlægges individuelt

til eleven og situationen. Fælles for alle programmer var - og er - dog, at de har reference til en fælles og meget uddybende statistikfejl på baggrundslageret.

Dansk

Anderledes stillede det sig for programmer til dansk. Vi havde her et meget righoldigt undervisningsmateriale spændende fra de traditionelle boglige materialer til undervisningsmaskiner som Language-Master, Mini-Tutor etc., hvor det var muligt indenfor en periode på f. eks. 1-2 uger at opstille et undervisningsforløb. Hertil var det nu muligt at producere opgaver til datamaskineformidlet undervisning, således at opgaver på terminalen gik hånd i hånd med det traditionelle materiale. Læreren kan nu anvende DFU-materialet i forskellige situationer:

- 1) som en kontrol af allerede indlært stof
- 2) som en støtte til andet materiale
- 3) i en direkte indlæringsituation som indledning til eller uddybning af traditionelt materiale.

En elevs undervisningsforløb i DFU-materialet kan ændres ved kald af program OPDATERING, idet situationen kan være af en sådan art, at der kræves en ændring, - eleven magtede f. eks. ikke en stillet opgave, eller den var for let.

Hvorledes reagerer nu eleverne?

I den ene af klasserne, der specielt har været interessant at følge, da den var den første 1. specialklasse ved Odense Skolevæsen, har elever og lærere altid 2 dataskærme til rådighed. Paren-tetisk skal det lige bemærkes, at vi tidligere startede specialklasser i 2. kl., d.v.s. med elever, der havde lidt et nederlag i løbet af 1. normalklasse. De omtalte elever er altså fundet for påbegyndelsen af normal skolegang i f. eks. børnehavklasser. Eleverne har haft de 2 dataskærme i klassen siden maj-juni 1976, hvor de gik i 2. kl. På dette tidspunkt kendte de ikke de store bogstaver, men kun de små trykte. Forventningerne til "fjernsynene" var store for de små elever, men tastaturet var forsynet med store bogstaver, hvorfor der her var et øjeblikkeligt handicap for brugen. Da en betingelse for at anvende skærmene var ovennævnte kendskab, præsterede børnene at lære de store bogstaver på mindre end 3 uger. Mads Jørgensen, klassens dansklærer, har siden fortalt, at han aldrig har været ude for noget lignende. Mads, der nu i sit 66. år har gjort hele udviklingen med fra at anvende griffel og tavle og sælge Chr. IX-penne til 2 øre stykket til at "sælge" EDB, taler her på baggrund af et helt

(fortsættes side 16)

INGEN DATAMAT ER STÆRKERE END DENS PERIFERIUDDSTYR

Til METRIC ABC 80 findes det
rigtige udstyr...

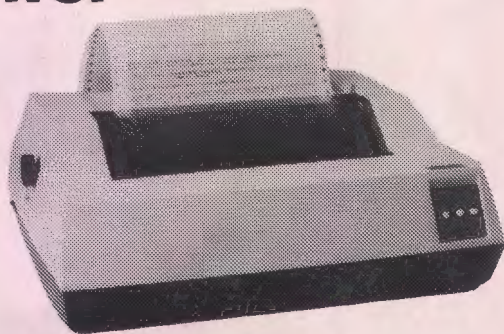
ABC FD2 - Floppydisk



- * Kapacitet 2x80K byte
- * Mere end 10 gange hurtigere end kassettebånd
- * Sekventiel og random access
- * 2 ekstra kortpladser til udvidelser
- * Pris:11.500,- excl.moms

ABC P 80 - Printer

- * Hastighed 125CPS/63LPM
- * Selv-test
- * Skriver på almindeligt papir
- * Dansk og engelsk karaktersæt
- * 40-80 (opt.132) kar./linie
- * Store og små bogstaver
- * Pris:6.900,-excl.moms + interfacekort:1.600,-excl.moms



DATAUDSTYR FRA SC **METRIC** AIS

DATAAFDELINGEN, SKODSBORGVEJ 305, 2850 NÆRUM, TLF. (02) 80 42 00

(fortsat fra side 14)

livs undervisningserfaring af denne type børn. Børnene er i dag 4 år efter starten lige motiverede og koncentrerede under arbejdet ved dataskærmene. De betragter rent faktisk dataskærme som en god hjælper og ven, hvilket mange episoder i tidens løb viser. Som f. eks. da en ny elev i 4. kl. kom ind i klassen - han var da på forhånd blevet lagt ind i systemet med opgaver - og fik lov til at sidde ved skærmen. Da han har skrevet sit nummer, får han spørgsmålet: "Hedder du Kent"? Mads kan fra sin plads se, at Kent bliver helt stiv, løfter armene i vejret, vender sig om og råber: "Hr. Jørgensen, den kender mig allerede!" Det var nærkontakt af 1. grad! Episoder af denne art er med til at styrke forbindelsen mellem eleven, læreren og undervisningsmidlet. For eleverne er der ikke tale om nogen form for fremmedgørelse i forhold til mediet, de behandler det uhyre kompetent og fornuftigt.

Den normale daglige start med at tænde for skærmene, logge sig ind på systemet og kalde programmer frem klares af eleverne selv. I klasserne har vi mange besøgende. Det er også her karakteristisk for elevernes koncentrationsevne ved skærmene, at de er ganske upåvirkede og fortsætter deres opgaveløsning.

Lærerne

Lærerne er jo nok dem, der vil føle den største fremmedgørelse. Ny teknik, computer, automatisering, gloser med en vis "belastning", er nok det, de fleste lærere vil fornemme i første omgang, og det er da også den hyppigste reaktion, jeg får fra lærerside. Fra konstruktørside har vi derfor bestræbt os på at gøre "systemet" så brugsvenligt som muligt. Udover at tænde og slukke for dataskærmen skal læreren kunne skrive flg.:

1. RUN "DANSK" for at køre sit planlagte undervisningsforløb.
2. RUN "OPDATERING" for at tildele nye undervisningsforløb.
3. RUN "STATISTIK" for at kunne få en statistik.

Selve forløbet skal køre bag ryggen på læreren og være transparent for denne. Lærerne har i de 4 år, systemet har fungeret, været meget tilfredse og mener, at det giver eleverne en større stave- og læsefærdighed - en væsentlig forudsætning for indlæring -, en bedre forståelse for det indlærte stof og magter at overføre indlærte færdigheder fra DFU-materiale til traditionelt og omvendt.

Man kunne nu tro, at dataskærmens fremkomst i klassen kunne have en skævhed til følge, at det sociale samspil, der nu engang må være i en klasse, bliver ødelagt, ved at eleverne ved dataskærmene mister kontakten med læreren og kammeraterne.

Vor erfaring er, at det sociale samspil tværtimod øges, hvilket har følgende årsager:

- 1) læreren har bestemt, hvad eleven skal beskæftige sig med i de ca. 10 minutter, en elev bruger ved terminalen.
- 2) eleven kan ikke snyde (!)
- 3) eleven arbejder meget koncentreret ved terminalen med individuelt tilpasset stof; herved får læreren mere tid til de øvrige elever.

Systemets opbygning

Selve systemet er baseret på et hovedprogram med tilhørende driverprogrammer, der hver især er tilpasset den datastruktur og opgave, det skal behandle. Al information fra driverprogrammer vedr. programmets status: utidig afbrydelse, normal afslutning, fejlsituationer m.v. afleveres via random filer til hovedprogrammet. Herved opnås bl. a. en væsentlig standardisering af driverprogrammerne, og man undgår at belaste baggrundslageret med den samme kode i alle programmer, men kan samle den i ét program. Eleverne har deres opgave- og statistikinformation placeret i filer, der opdateres/tømmes ud fra information under opgaveløsning eller ved lærerkommandoer. Opgaverne ligger pakket i random- og sekventielle filer, typen afhænger af art og struktur. Et problem omkring systemer af denne art er beskyttelsen af programmer og data mod uautoriseret brug (sletning/ændring). Her benytter vi RC 8000's base-system, der i denne henseende er overordentlig effektivt. Gennem 4 år har et system af denne art gennemgået store ændringer, både hvad datastruktur og programmer angår. Ændringerne vil stadig komme til - og nyskabelser. Vort største handicap er nok fasen fra opgaveformulering til selve opgaven er i "kassen", idet selve indtastningen og fastlæggelsen af datastrukturen foretages af "programmør"-gruppen. Idéen og fastlæggelsen af opgavetekster og endelig layout er et team-work imellem lærer og "programmører". I øjeblikket arbejder vi meget ihærdigt på at få opgaveforfatterne til at foretage indtastningen af opgaverne selv, principielt er der intet problem heri, men lærerne har det som en hest, der refuserer foran en forhindring, hvor forhindringen her er nervøsiteten ved at begive sig ind i det ukendte land. Løses problemet ikke i denne generation af systemet, vil det helt givet kunne løses i kommende generationer med evt. et andet programmeringssprog som basis (PASCAL?), der bl. a. er orienteret mod strukturering af data.

Ved påbegyndelsen af skoleåret 1980/81 pr. 1./8. vil der ialt være 6 skoler med specialklasser og specialundervisning, der benytter systemet til dansk og regning. Vi vil på ingen måde foregive, at vi her har fundet de vises sten, hverken i systemets udformning eller i anvendelsesmetoder.

Systemet har i sin nuværende udformning vist sig ganske brugbart og er en platform, hvorfra nye idéer og udformninger kan udvikle sig i et roligt tempo. Det er ikke den automatiserede lærer, vi her har opfundet, men - som Mads Jørgensen udtrykker det: "Det er et pragtfuldt hjælpemiddel i en lærers hånd".

Torben Høirup

● ● ●

Tredje runde. . .

I det amerikanske tidsskrift *Recreational Computing* er man nu nået til tredje runde i en debat om BASIC contra PASCAL.

I anden runde var det vores hjemlige Søren Ravn-Jensen, der med et COMAL program ville vise at "BASIC" kunne være mange ting.

Diskussionen er nok ikke færdig endnu, men PASCAL fanatikeren er ikke helt utilfreds med COMAL. Bl. a. er han parat til at købe en micro med COMAL, når den kommer til 600 \$ og er opbygget med datastrukturer osv. (!) En anden morsom ting er, at selv om Søren Ravn-Jensen ikke nævner navnet COMAL i sin artikel, genkendte PASCAL manden Børges "barn" og han kalder det for "rent sofisteri" at bruge navnet BASIC om COMAL(!)

● ● ●

CRYPTARITMER

En "dille" er blomstret op i USA, privamaternes hjemland. Det er egentlig gammelkendte opgaver, der får en ny dimension, ved at de skal løses på en datamat.

Mange kender denne "klassiske":

$$\begin{array}{r} \text{SEND} \\ \text{MORE} \\ \hline \text{MONEY} \end{array}$$

Opgaven er selvfølgelig at finde hvilke cifre, de enkelte bogstaver står for.

Men prøv så at lave et program, der kan løse sådanne CRYPTARITMER.

Her er et par stykker at bide skeer med:

$$1) \begin{array}{r} \text{LSULTUL} \\ \text{ESSLTTY} \\ \hline \text{BMMHHHHH} \end{array}$$

$$2) \frac{\text{CT} \cdot \text{ND} \cdot \text{AA} \cdot \text{TU} \cdot \text{CB} \cdot \text{D} \cdot \text{TU} \cdot \text{RT} \cdot \text{ND}}{\text{UT} \cdot \text{RY} \cdot \text{BE} \cdot \text{TR} \cdot \text{AC} \cdot \text{NT} \cdot \text{ET} \cdot \text{AT} \cdot \text{AB}} = \text{D}$$

I 1) skal man finde hvor mange løsninger der er.
I 2) skal man bare finde værdien af de enkelte bogstaver.

Hvem laver nu nogle gode, evt. generelle, programmer?

NYE BØGER

Der er udsendt en ny udgave af "Bogen om EDB" på Politikens forlag.

Man har taget konsekvensen af den hurtige udvikling på edb-området, og det meste af bogen er nye eller reviderede artikler i forhold til de gamle udgaver.

Bogen har desværre én fejl: I afsnittet om organisationer og foreninger har man på en eller anden mystisk måde overset Datalæreforeningen. Men mon det ikke bliver rettet næste gang?

Trods denne "skønhedsfejl" kan bogen anbefales.

★ ENGLAND IGEN

Den i sidste nummer omtalte gentagelse af studieturen til England i april måned så vi os desværre nødsaget til at aflyse, idet der kun havde meldt sig fire personer som interesserede.

Vi forventer at kunne vende tilbage til efteråret med et nyt tilbud til medlemmerne.

● ● ●

Some are wise and som are otherwise!

Fra pålidelig kilde har vi erfaret, at en kendt sønderjyde nu vil indføre OTHERWISE i COMAL. Mere konkret skal OTHERWISE bruges i CASE-konstruktionen, hvor de sætninger, der står efter OTHERWISE, udføres, hvis ingen af WHEN'erne opfyldes.

I de fleste COMAL-versioner i dag kan man opnå samme effekt ved at sætte alternativet til WHEN'erne ind før den første WHEN-sætning, men selvfølgelig vil et OTHERWISE gøre tingene tydeligere. Så på sin vis er det nok mere "wise" end "otherwise" at indføre OTHERWISE.

● ● ●

ER STREGKORTENE KOMMET TIL KORT? NÆ . . .

For ikke så mange år siden var det temmeligt almindeligt, at man baserede sin undervisning i datalære på stregkort, som man fik kørt på et eller andet fjernt anlæg. Omkring 1973-1975 kulminerede disse aktiviteter, og der blev bl. a. kørt masser af programmer på Danmarks Lærerhøjskole og på ASDA.

Op det fortsætter. I hvert fald på ASDA. Her køres der stregkort som fast service 2-3 gange om ugen, og der er stadig en hel del skoler, der benytter sig af denne mulighed. Specielt skal da nævnes, at Aalborg Katedralskole, der jo er en statsskole og derfor ikke som de andre gymnasier i Nordjylland har fået sin egen datamat, i flere år har drevet forsøgsundervisning baseret på stregkort, som køres på ASDA.

Så kort sagt, kortene er ikke helt kommet til kort...

RC * INFO

RC-NYHEDSORGANET FOR RC-BRUGERE

DE REJSENDE DISKETTER

Regnecentralen har gennem mange år haft et rejsende anlæg, som mange lærere og elever har haft glæde af. Dette anlæg er nu ved at blive afløst af microdatamater, så inden længe vil vi have "De rejsende micro'er" som et nyt begreb indenfor edb og undervisning. Men allerede nu er der noget "rejsende" fra RC79:

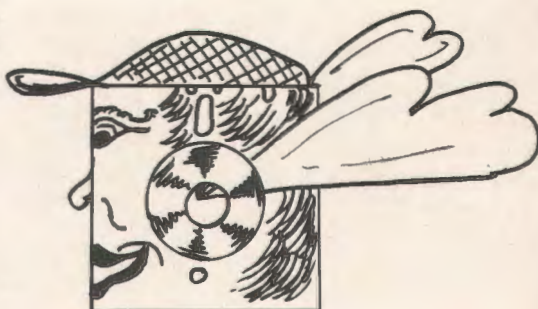
DE REJSENDE DISKETTER

RC701 har i den udsendte version ikke umiddelbar tilslutningsmulighed for f. eks. strimmellæser, og dette bevirker, at der kun er to muligheder for at få et program/ind i maskinen: man kan taste det ind, eller man kan loadet det fra en diskette. Det er jo lidt besværligt med al den tastning, men har man først én gang fået programmet ind på en diskette, kan det kopieres i massevis på få sekunder.

Det samme gælder også for datafiler: første gang er man nødt til at taste, men er indholdet af en fil først inde på disketten, kan man uden større problemer flytte de fleste filer fra diskette til diskette.

Rundt om i landet sidder der jo mange nybagte RC701 brugere, som med sikkerhed vil have glæde af at udveksle programmer. For hvad den ene bruger har fået tastet ind i en RC701, er der ingen grund til at en anden bruger også skal taste. For at få gang i en sådan frugtbar udveksling af programmer på disketter, starter RC79 ordningen med: DE REJSENDE DISKETTER.

De forskellige RC701 brugere vil med jævn mellemrum modtage én af de rejsende disketter. På denne diskette vil der ligge en række programmer evt. med tilhørende datafiler. Brugeren har så nogle dage til at tage kopier af alt, hvad han finder interessant, men han



bedes så samtidigt lægge kopier af egne programmer på disketten.

I kuverterne sammen med disketterne vil der ligge selvklebende mærker med adressen på den næste RC701 bruger i kæden, og efter et par dage sendes disketten så videre.

Med passende mellemrum vil disketterne vende tilbage til nogle "knodepunkter", hvor der dels vil blive "ryddet op" på disketterne, og dels vil blive kopieret programmer over fra forskellige disketter med forskellige rejseruter.

Umiddelbart kan man ikke lade DE REJSENDE DISKETTER lægge ruten forbi f. eks. RC7000 anlæg, idet formatet for disketterne ikke er det samme. Men en tilsvarende ordning for RC7000 disketter er under forberedelse, og man vil så i "knodepunkterne" kunne koordinere og overføre programmer fra den ene type rejsende disketter til den anden.

»KATALOG« OG »SYSTEM«

Hvis man skriver BYE til RC701 vil den spørge om PROGRAM? Man kan derefter vælge mellem flere, bl. a. KATALOG og SYSTEM. Virkningen af disse programmer er beskrevet i kap 6, men et par uddybende bemærkninger er vist på sin plads, specielt fordi det ikke går så galt, som brugsanvisningen siger. I denne står nemlig, at ved brug af KATALOG eller SYSTEM vil disketten blive overskrevet, således at programmer og data går tabt. Dette er ikke tilfældet altid; man kan således godt skifte systemet på en diskette ud til en nyere revision uden at programmer og data går tabt.

Virkningen af KATALOG er altid den samme: der udskrives et nyt katalog på disketten og dermed er alle programmer og data glemt for evigt.

Virkningen af system er afhængig af, om disketten i forvejen har et katalog eller ej. Hvis disketten ikke har et katalog, f. eks. fordi det er en helt ny diskette, skrives først et nyt katalog ud på disketten og derefter udskrives selve systemet, således at disketten kan bruges til opstart af RC701.

Hvis der derimod i forvejen er et katalog på disketten, vil dette katalog ikke blive rørt, men systemet vil blive skrevet ud på disketten, og var der et gammelt system i forvejen (f. eks. en ældre revision), vil dette blive overskrevet. D.v.s. man kan altså skifte revision uden at programmer og datafiler bliver berørt.

På den anden side kan man dog stadig få slettet alle gamle programmer på en diskette. Man skal nemlig bare først bruge programmet KATALOG og bagefter SYSTEM.

SKRIVEBESKYTTELSE

Har man en diskette med vigtige programmer eller data, kan man beskytte disse mod u- eller overlagt ødelæggelse.

I brugsanvisningens kap. 3 er beskrevet, hvordan man med en hullemaskine og/eller saks kan klippe et lille hak i disketten, og dette hak sikrer så mod ødelæggelse af indholdet. Funktionen minder på sin vis om en tilsvarende facilitet på almindelige kassettebånd.

Har man lavet et sådant hak i sin diskette, vil det ikke mere være muligt at save pro-

grammer, ligesom man ikke kan slette programmer.

Ønsker man alligevel på et senere tidspunkt at slette eller save, kan dette også klares; man tager simpelthen et stykke uigennemsigtigt tape (f. eks. lærredstape) og dækker hakket, så er disketten parat til alting igen. Benytter man sig af dobbeltsidige disketter kan man godt indrette dem på den måde, at den ene side er skrivebeskyttet mens den anden side ikke er det.

Ønsker man at beskytte en dobbeltsidig diskette på begge sider, skal der to hak til, ét i hver side.

BILLIG PRINTERMULIGHED TIL RC701

RC701 er udstyret med tilslutningsmulighed for en såkaldt V24 printer, d.v.s. en printer, der modtager signalerne i serie ligesom almindelige terminaler.

Dette åbner mulighed for at benytte en almindelig terminal som printer til RC701, og her er det nok mest aktuelt med en god gammeldags TTY eller en Silent.

Begge disse terminaler kan uden større problemer tilsluttes, og de virker fint dog kun med henholdsvis 10 og 30 tegn/sek.

Det er nødvendigt med en lille mellemløbetid mellem RC701 og terminalen, idet et par af forbindelserne skal "vendes" og et par andre skal kortsluttes. Og da RC701 fra fabrikkens side er stillet til en hastighed på 120 tegn/sek. på printerudgangen, skal man også lige have lukket maskinen op og stillet hastigheden ned til 10 eller 30 tegn/sek.

Alt dette volder ikke de store vanskeligheder, og allerede flere RC701 brugere benytter sig af denne mulighed for udskrifter af og fra programmer.

Yderligere oplysninger omkring tilslutning af terminaler som printere til RC701 kan fås ved henvendelse til:

Regnecentralen 79
Lautrupbjerg 1
2750 Ballerup
Att.: Niels Adler Nissen
(02) 65 80 00

LUFTVÆRN

En af de ting, man kan more sig meget med, er ON-ESC sætningen. I det følgende program benyttes den som "aftrækker" på en luftværnskanon.

Programmet virker ellers på den måde, at der i en tilfældig højde sendes en flyver ind fra en af siderne. I midten af bunden har man derefter en kanon, og man skal så ramme flyveren med en granat.

Programmet hører til blandt "spøg & skæmt"-programmerne, men dem skal der jo også være plads til en gang imellem.

Programmet vil kunne køre på en RC822 dataskærm og forudsætter call-routine til cursoradressering. Hvis man ikke har denne facilitet med i sit system, kan man let rette programmet de steder, hvor der adresseres. God fornøjelse . . .

```

9000 LET FORSØG=0; PLET=0
9010 REPEAT
9020   RANDOMIZE
9030   LET HØJDE=INT(18*RND(0))+2
9040   PRINT "<140>"
9050   CALL "XYADR",40,21
9060   PRINT " I"
9070   CALL "XYADR",40,22
9080   PRINT "(#)"
9090   LET RETNING=-1; POSITION=80
9100   IF RND(0)>.5 THEN LET RETNING=1; POSITION=1
9110   LET RAMT=0; FORBI=0; SKUD=0; GRANAT=20
9120   ON ESC THEN EXEC SKUD
9130   LET SKUD=0
9140   REPEAT
9150     LET POSITION=POSITION+RETNING
9160     CALL "XYADR",POSITION,HØJDE
9170     IF RETNING=1 THEN
9180       PRINT " :-)"
9190     ELSE
9200       PRINT "<60>--:"
9210     ENDIF
9220     IF SKUD THEN
9230       LET GRANAT=GRANAT-1
9240       CALL "XYADR",40,GRANAT
9250       PRINT "^<31>"
9260     ENDIF
9270     IF ABS(POSITION-38)<3 AND GRANAT=HØJDE THEN
9280       CALL "XYADR",37,HØJDE
9290       PRINT "<7><12>♦♦ BOOM! ♦♦<7>"
9300       LET RAMT=1; PLET=PLET+1
9310     ENDIF
9320     IF GRANAT=1 OR (GRANAT-HØJDE)<-4 THEN
9330       PRINT "<140>FORBIER!!!!"
9340       LET FORBI=1
9350     ENDIF
9360   UNTIL RAMT OR FORBI
9370   DELAY=2
9380   PROC SKUD
9390     PRINT "<7>"
9400     LET SKUD=1
9410   ENDPROC
9420   LET FORSØG=FORSØG+1
9430   UNTIL FORSØG=5
9440   PRINT "<140>DU FIK";PLET;"TRÆFFERE UD AF 5 MULIGE

```

RC700 PÅ HANNOVER-MESSEN

RC 700 sælges nu også uden for Danmark. På et møde for RC's datterselskaber og større udenlandske kunder i starten af april, blev startskuddet givet.

Den første præsentation for kunder skete på Hannover-messen, som er verdens største industrimesse. Den henvender sig primært til industrikunder, og det var således første gang man henvendte sig til denne sektor med RC 700. Alle forhåbninger blev indfriet i fuldt omfang, og der ventes et meget stort salg af RC 700 i udlandet.

RC 700 blev udstillet sammen med andre RC produkter, nemlig RC 8000 og RC 3600. Desuden blev præsenteret et andet nyt produkt, DATEX-P, som er en datanettilslutning til det kommende offentlige data-net. RC er én af de første leverandører i verden, som kan vise en sådan tilkobling, og systemet har derfor vakt stor opmærksomhed.

RC700 PÅ INDKØBSMESSEN I FREDERICIA

RC 700 vises på indkøbsmessen for stat, amter og kommuner i Fredericia, den 25 - 29. august 1980. Adgangskort kan rekvireres hos Thorkild Maaetoft, tlf. 02-65 80 00, lokal 311.

NYE RC-BRUGERE

Inden for undervisningssektoren er der siden sidst kommet følgende nye brugere:

RC 7000

Aalborg Skolevæsen

RC 700

Langå Skole

Ravnsborgskolen i Horslunde

Annebergvejens Skole, Aalborg

Lemvig Gymnasium

Pædagogisk Central, Søllerød

Skive Handelsskole

Metalindustriens Fagskole, Ballerup

Nykøbing Falster Katedralskole

RC 8000

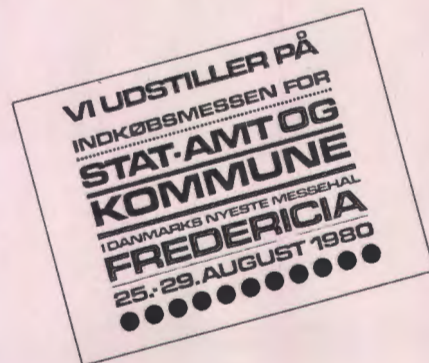
Amtscentralen i Odense

Udvidet med: 21 MB Fixed Media Disk

192 kw halvlederlager

1 MPX (8kanaler)

1 BSC-kanal (4kanaler)



RC ^A **REGNECENTRALEN** af 1979

LAUTRUPBJERG 1 - 2750 BALLERUP

Telegram: rcbalrc - Telex: 35 214 rcbaldk

Telefon: 02 65 80 00

Erhvervsorienteret mikrodatamat

SBS 8000

Microcomputeren:

Z80A baseret, 24k Byte Super-Basic i ROM, 32k Ram lager, (Ram udvides til 60K), 64 tegns skærmbredde, CP/M, C Basic (compiler), Pascal, Cobol, Fortran og mulighed for Comal, tastaturet har talblok. Der er indtil 16 cifres udskrift, RE-NUMBER, AUTO-LINE NO, PASS-WORD, TIMER' MEMORY-PAGING (indtil 15 programmer i hukommelsen samtidig), GRAFIK og indbygget RAM-test.

Disksystemet:

Dobbelt floppy disk system med 5 1/4" diske. Hver diskette indeholder 184 kByte. Der kan tilsluttes 2 stk. dobbelt 5 1/4" diskette stationer. Der kan også leveres 8 1/2" floppy diskstationer og 10, 20 og 40 mByte hard disk station.

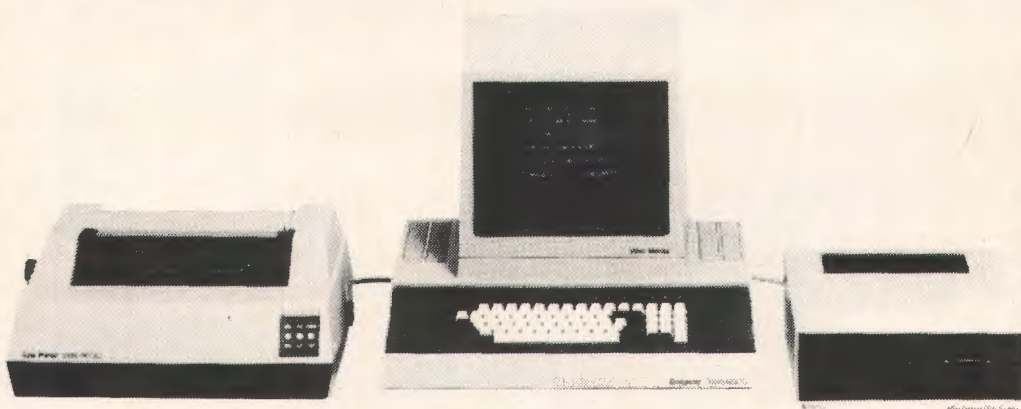
Skrivere:

SBS 8830 har en skrivehastighed på 125 tegn pr. sekund, en skrivebredde på 80 positioner og en matrix på 5 x 7. Skriveren SBS 8703 har en skrivebredde på max. 132 positioner og en hastighed på 180 tegn pr. sekund. Begge skrivere kan skrive dobbeltbredde tegn, der er softwarekontrollerede.

Udvidelser:

S-100 bus-interface, sound-generator, printer-interface, RS-232C & 20 mA current loop interface, D/A-A/D converter (8bit), TTL parallel data interface, IEEE488 interface, Card cage expansion for 7 cards, Real life controller 6 x 220 V AC + 6 x 12 V DC.

Pris ca. 28.000,- samlet.



E.W.S. Electronics ApS, Rolighedsvej 17, 1958 København V, tlf. 01-37 14 12

— Forlang brochure —

Generalagent for Scandinavia

Forhandler søges

Et plottersystems udvikling

Nils R. Pedersen

I det følgende vil jeg beskrive et nyt plottersystem til brug på SAG (Silkeborg Amtsgymnasium), som jeg udviklede i efteråret 79, men samtidig vil jeg prøve at beskrive de tanker og idéer, der ligger til grund for denne konstruktion i et forsøg på at give andre en startimpuls til selv at gå igang med at konstruere forskellige interface's, f. eks. til plottere, elektroniske musikinstrumenter e. l. Der er flere grunde til, at jeg gerne vil motivere andre til selv at gå igang. For det første lærer man selv en forfærdelig masse af det, og når man selv gør hele arbejdet, ser man det hele tage form lidt efter lidt, lige fra den første idé til konstruktionen står klar i et kabinet, og det får man ikke mulighed for nogensomhelst steder ellers, og det giver en meget større forståelse af de forskellige problemer ved en sekventialiseret produktionsmetode, hvor man ellers selv kun får en lille del af processen at se. Desuden er det en stor tilfredsstillelse at vise, at det ikke kun er de store selskaber med masser af penge i hånden, der kan udvikle anvendeligt udstyr, men at man også kan gøre det med ganske enkle midler og lidt omtanke. En væsentlig ting er selvfølgelig også, at slutproduktet bliver en hel del billigere end det kommercielle udstyr (prototyper er dog altid dyrere end de næste kopier af konstruktionen).

Så da jeg startede med at sige, at det var et plottersystem til brug på SAG, jeg her ville beskrive, var det ikke så meget plottersystemet, jeg tænkte på, som de tanker og idéer der ligger bag det, og som forhåbentlig kan hjælpe nogle hjemmekonstruktører over de første små "hurdles" på vej mod målet.

Efterårsprojektets start

Jeg konstruerede i foråret 79 sammen med Carsten Andersen et plotterinterface beregnet til analoge XY-plottere, hvoraf vi på SAG havde en enkelt stående. Det var ikke nogen særlig god plotter - den var ulineær, havde ikke indbygget penlift, var langsom, armene var for tunge, og den kunne kun tegne rette linier i rette vinkler eller i 45°. Penlift mekanismen klarede vi hurtigt, men hvis man skulle tegne rette linier i en arbitrær vinkel, måtte man steppe sig frem, og det tog tid. Dette og andre ting gjorde, at man både fra elev- og lærerside henvendte sig til mig og spurgte, om jeg ville konstruere et nyt system baseret på en HP-plotter (Hewlett Packard "HP9125b Calculator Plotter"), der var en del af et ældre bordregnersystem, og som ikke var blevet benyttet efter RC-7000-anlæggets ankomst.

De første tanker

Når jeg nu igen skulle til at lave et plottersystem, var der jo god grund til at lave konstruktionen, så den meget let kunne ændres til anvendelse sammen med andre plottere end vores HP9125b, altså en slags "generelt plotterinterface". Opløsningsevnen skulle være stor, og jeg skønnede, at en opløsning på 1/10 til 1/25 mm måtte være passende; altså 100 til 400 step/cm. Der skulle være en beeper, der fortalte, om man begik fejl; f. eks. forkert brug af formatet eller brug af for store koordinater (overflow), og sidst, men ikke mindst, skulle det være let at bruge plottersystemet, dvs. at brugerformatet skulle være enkelt og logisk opbygget, ligesom det ville være en fordel, hvis det var selvdokumenterende. Selvfølgelig måtte systemet set fra brugerside heller ikke være væsensforskelligt fra det gamle system, så man kunne undgå at revidere eksisterende programmer alt for meget.

Plotteren måtte ikke ombygges, så den ikke kunne bruges sammen med det gamle HP-anlæg, og det betød, at der måtte konstrueres et omkiftersystem, så plotteren kunne tilkobles både til det nye interface og det gamle HP-anlæg.

Projektet igang (arbejdet på selve plotteren)

Det er ganske indlysende, at selve HP-plotteren måtte undersøges før nogensomhelst andet. Af eksterne oplysninger forelå tegneformatet på 40 x 25 cm, hvor de indbyggede DACs (Digital to Analog Converter) havde en opløsning på 200 step/cm, altså et koordinatformat på 8000x5000 step, altså $X \in [0; 8000[$ og $Y \in [0; 5000[$, idet jeg droppede fortegnfaciliteten, der syntes at være besværlig at anvende i mit system. Desuden indicerede det 44-polede multistik, der dannede forbindelse til HP-bordregneren, at der var tale om et parallelt indlæsningssystem (hver bit i talformatet har sin egen forbindelse), som ville gøre det let at konstruere interfacet.

Det sidste viste sig meget hurtigt at være ønsketænkning, for efter at have åbnet plotteren, lod det sig hurtigt afgøre, at de to DACs (én for hver akse) havde et 4-bit-serielt indlæsningssystem - altså at de enkelte cifres binære koder skulle komme efter hinanden. Efter ved nogle målinger at have lokaliseret de forskellige tilledninger til DACene - de af dem man kunne vurdere, hvad de blev brugt til - stod det også klart, at de to koordinater (X og Y) skulle komme med cifrene skifte-

vis i en bestemt rækkefølge - altså først tusinder for X, så tusinder for Y, så hundreder for X efterfulgt af hundreder for Y osv. At der anvendes en bestemt rækkefølge, var ikke noget krav fra DAC-ernes side, men da de blev nulstillede før en data-indlæsning, betød det for det første, at overførsel af datasæt fra interfacet skulle ske så hurtigt, at pennen ikke nåede at bevæge sig mod origo, før det nye datasæt var indlæst, men også for det andet, at det måtte være logisk at overføre mest betydende cifre først, så bevægelsen mod origo formindskedes yderligere (som systemet kører nu, foregår ciffertransmissionen ved ca. 10 Kbaud, altså en transmissionstid for et datasæt på ca. 1 millisekund).

Når nu kun de 4 forbindelser af de 44 i multistikket var til taloverførsel, måtte det betyde, at de andre måtte være forskellige kontrolsignaler, pen-styring (op/ned) o. l., og at disse kontrolsignaler ville forsvinde, når plotteren ikke længere var i forbindelse med HP-bordregneren. Altså måtte jeg se at finde ud af, hvilke kontrolsignaler der var strengt nødvendige for plotteren, og hvordan de så ud. Det er oplysninger, der findes på et anstændigt servicediagram, men en hurtig opringning til HP afslørede - ikke til nogens forbavelse - at de ikke ville levere en kopi af et sådant. Jeg måtte altså selv igang med at tegne et diagram over plotterens vigtigste dele for at kunne simulere kontrolsignalerne fra det nye interface. Da jeg havde besluttet at bryde ind mellem det gamle styresystem og DACene, diagrammerede jeg fra DACene mod udgangstrinnene, der kontrollerede motorerne (der trækker pennen). Da det var gjort - og det tager sin tid - var det ikke så vanskeligt at rekonstruere de nødvendige kontrolsignaler, og det viste sig, at man kunne nøjes med otte parallelle kontrolkanaler til DACerne samt deres fire datakanaler og desuden et par kanaler til penliftmekanismen samt et par andre til andet brug - alt i alt 18 parallelle kanaler.

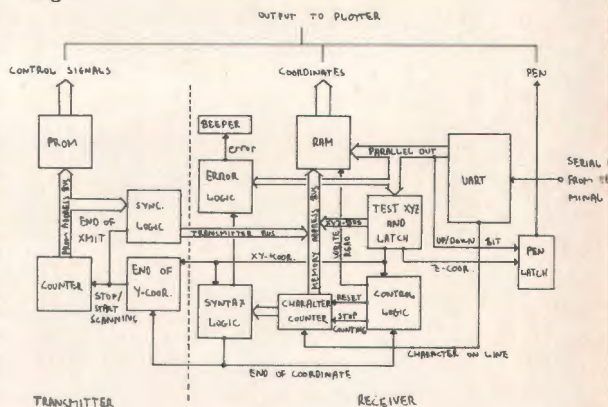
Da jeg nu vidste, hvordan datasættene skulle overføres til plotteren, hvilke kontrolsignaler der skulle til for at indlæse dem, og hvordan de så ud, hvordan man kontrollerede pennen, og hvilke spændingsniveauer de forskellige signaler skulle have (og de var fuldstændig vanvittige i forhold til de niveauer, man arbejder med idag), var den første del af projektet overstået.

Konstruktionen af interfacet

Det første jeg gjorde mig klart, var selvfølgelig syntaksen i brugerformatet, eller rettere det af den, der skulle implementeres i interfacet. Der ville jo blive tale om X-, Y- og Z-værdier, hvor $X \in [0; 8000[$, $Y \in [0; 5000[$ og $Z \in \{0,1\}$ ($Z=0$ betyder at pennen er nede, 1 at den er oppe). For X og Y besluttede jeg at formatet skulle være:

" $X = C_1C_2C_3C_4\Delta\Delta\Delta$ " og " $Y = C_1C_2C_3C_4\Delta\Delta\Delta$ ", hvor Δ er spaces (mellemrum). For pennen skulle formatet være: " $Z=0\Delta$ " og " $Z=1\Delta$ ".

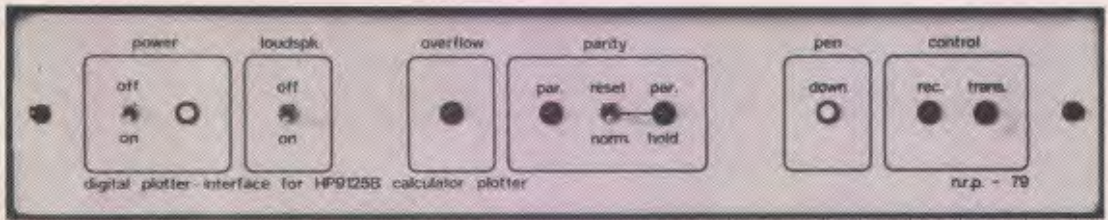
I det følgende vil jeg beskrive konstruktionen ud fra det detaljerede blokdiagram, som godt nok først er tegnet, efter at konstruktionen stod færdig.



Blokdiagram.

De terminaler, vi har på SAG, anvender som de fleste EIA's RS-232C spændingsstandard, og transmissionen foregår asynkront i et syv-bit ASCII-karakter-format med en startbit, en paritetsbit og to stopbit. Modtageren (receiver) i interfacet er derfor en UART (Universal Asynchronous Receiver/Transmitter), som afgiver ASCII-koder parallelt, samtidig med at den checker for transmissionsfejl (paritetstest), og fortæller, når den har modtaget en karakter. Interfacet skal ikke foretage sig noget, med mindre det drejer sig om X-, Y- el. Z-koordinater, så derfor testes de modtagne karakterer, og hvis der kommer et X, Y el. Z, lagres denne koordinat karakter, der jo ellers mistes, når den næste karakter modtages. Nu sættes interfacet automatisk i modtagemode, karaktertælleren nulstilles, og RAMen (hukommelsen: Random Access Memory) stilles i skrive-(Write)-mode, så man kan lagre i den. Hvis koordinat karakteren er X el. Y sker følgende:

Den næste karakter (efter X el. Y) er "=". Når "=" er modtaget, stepper karaktertælleren 1 frem og står altså på 1. Syntaks/kontrol-logikken ignorerer tegnet, som er uden interesse, og som derfor kan være hvadsomhelst, men "=" er valgt af hensyn til formatets selvdokumentarisme. Ved hver af de følgende fire karakterer (som er selve koordinaten), stepper tælleren 1 frem, og de enkelte cifre lagres i RAMens X- eller Y-område. De efterfølgende tre spaces bruges til at sætte RAMen i Læse-(read)-mode, så man kan hente den indskrevne information og til at sætte modtageren i ventemode, så den er klar til næste koordinat. Hvis koordinat karakteren var et Y,



Frontplade.

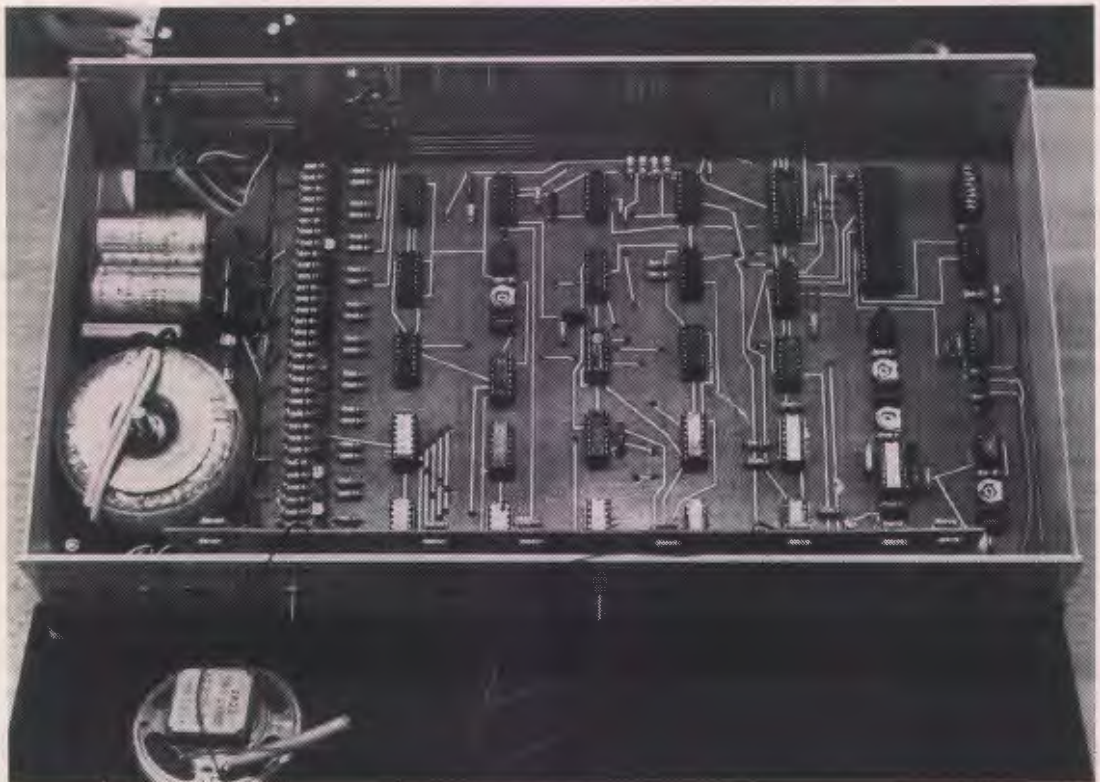
gøres senderen (transmitter) klar til at overføre RAMens indhold til plotterens DACs, samtidig med at modtageren stilles i vente-mode. Senderdelen overtager nu kontrollen med RAMen, der står i læse-(read)-mode, og en tæller skanner igennem en PROM (Programmable Read Only Memory), der indeholder kontrolsignalerne til plotteren, samtidig med at en synkroniseringslogik kombinerer kontrolsignalerne med datasignalerne fra RAMen og bryder koordinaterne op, så de enkelte cifre kan overføres i den rigtige rækkefølge (se tidligere). Når transmissionen er overstået (der sendes 96 bytes signaler via 12 parallelle kanaler), sættes senderen også i vente-mode, og interfacet er parat til at modtage et nyt koordinatsæt.

Hvis nu koordinatkaraktæren var et Z, var følgende sket:

Karaktertælleren ville blive nulstillet og koordinatkaraktæren gemt af vejen, og når "=" var modtaget, ville karaktærtælleren steppe 1 frem til 1. Lighedstegnet ville også her blive ignoreret, men efter at "Z" blev modtaget, klargjordes en speciel hukommelse (pen latch), og når karakteren efter "=" er indlæst, overføres den til denne latch (karakteren er enten 0 eller 1), som styrer pennen. Det ene space sætter også her modtageren i vente-mode.

Hvis man sender for store tal over til interfacet, vil plotteren begå fejl, og beeperen vil hyle for at gøre opmærksom på dette.

Ud fra det her beskrevne ses det, at Z-kommandoer er øjeblikk virkende, hvorimod X- og Y-koordinater først viser sig på plotteren efter



Et kig indenfor i interfacet. (Det er låget med højttaleren (beeperen) man kan se i forgrunden).

at Y-koordinaten er indlæst; så derfor skal X-koordinaten altid indlæses først.

Når man så udfra blokdiagrammet skal lægge et diagram for konstruktionen, tager det altid længere tid end ventet - for mig gør det ihverfald, fordi jeg udfra betragtninger om, at prototyper som oftest kun er til besvær, videreudvikler konstruktionen på papiret og skaber på den måde de nye generationer, før de gamle er blevet realiteter. Det er en procedure, der tager sin tid, men slutproduktet fungerer faktisk taget altid upåklageligt. På dette tidspunkt er størsteparten af papirarbejdet overstået, og tredje fase af projektet kan påbegyndes.

Den praktiske opbygning

Da nu diagrammet lå klart, var det næste skridt at få lagt et print, og det var ikke nogen helt let opgave til så stor en konstruktion, specielt ikke hvis den skulle virke første gang, der blev sat strøm til. Da jeg ville sikre mig, at jeg ikke lavede fejl, tog det et pænt stykke tid at lægge printet og teste det flere gange på papiret, for derefter at tegne selve printet i hånden (der var jo ingen grund til at lave en foto-master, og så fotografere det over på printet, for der kunne jo alligevel være fejl). Efter at printet var færdigt og komponenterne iloddet, blev konstruktionen afprøvet. Den var dog allerede testet på dette tidspunkt, da jeg prøveopbyggede den eksperimentelt på dertil indrettede prøveplader (Deck's), og efter at én enkelt fejl var rettet, fungerede den upåklageligt. Et kabinet blev derefter tilpasset til formålet, og alle delene blev samlet til et hele.

Da nu interfacet var færdigt og omskifterarrangementet indbygget i plotteren, blev systemet en terminal og blev afprøvet til alles tilfredsstillelse. På dette tidspunkt havde projektet optaget godt 280 arbejdstimer.

Anvendelsen

Et af de krav jeg stillede var, at plottersystemet skulle være let og logisk at anvende set fra brugerens side, hvilket igen ville sige, at brugerformatet skulle bestå af simple dele kædet sammen af ofte anvendte ordrer i vores COMAL-sprog. Når man ser på de dele af formatet, der blev implementeret i syntaks-logikken i interfacet, virkede det indlysende at anvende et "PRINT USING"-format, og her opstilles først det generelle format, der benyttes af plottersystemet:

$$\text{PRINT USING } \left\{ \begin{array}{l} \text{"<14>\Delta} \\ \left. \begin{array}{l} X = \text{####} \Delta\Delta\Delta \\ Y = \text{####} \Delta\Delta\Delta \\ Z = \text{\#}\Delta \end{array} \right\} \left[Y = \text{####} \Delta\Delta\Delta \right] \end{array} \right\} \left\{ \begin{array}{l} \left\{ \begin{array}{l} \langle \text{var} \rangle \\ \langle \text{expr} \rangle \\ \langle \text{val} \rangle \end{array} \right\} ; \left\{ \begin{array}{l} \langle \text{var} \rangle \\ \langle \text{expr} \rangle \\ \langle \text{val} \rangle \end{array} \right\} \left[\left[\begin{array}{l} \langle \text{var} \rangle \\ \langle \text{expr} \rangle \\ \langle \text{val} \rangle \end{array} \right] ; \left[\begin{array}{l} \langle \text{var} \rangle \\ \langle \text{expr} \rangle \\ \langle \text{val} \rangle \end{array} \right] \right] \\ \left\{ \begin{array}{l} \langle \text{var} \rangle \\ \langle \text{expr} \rangle \\ \langle \text{val} \rangle \end{array} \right\} \left[\left[\begin{array}{l} \langle \text{var} \rangle \\ \langle \text{expr} \rangle \\ \langle \text{val} \rangle \end{array} \right] ; \left[\begin{array}{l} \langle \text{var} \rangle \\ \langle \text{expr} \rangle \\ \langle \text{val} \rangle \end{array} \right] \right] \end{array} \right\} \end{array} \left. \right\} \text{"<15>"}$$

hvoraf de to mest brugte formater nok vil være:

```
PRINT USING " <14>\Delta X = #### \Delta\Delta # #### \Delta\Delta <15>";
<expr>, <expr>
```

og

```
PRINT USING " <14>\Delta Z = #\Delta <15>"; <var>
```

eller f. eks. en afledning heraf:

```
PRINT " <14>\Delta Z = 0\Delta <15>
```

(sænker pennen)

En ting, der kan være lidt generende ved brug af dette format, er, at billedet "ruller", men man kan ganske let sikre sig, at terminalen skriver på samme linie hele tiden:

Man benytter en PAGE på 0 og bruger så formatet:

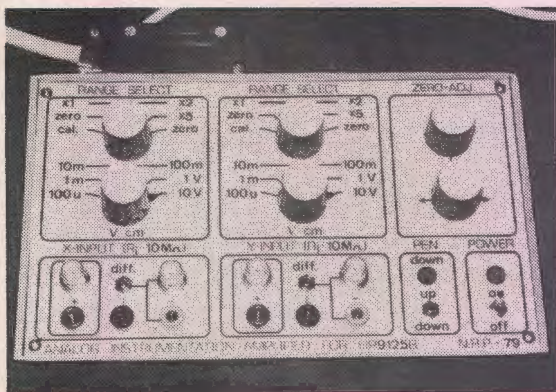
```
PRINT USING " <14>\Delta X = #### \Delta\Delta Y =
#### \Delta\Delta <15> <13>"; <expr>, <expr>!
```

Δ er spaces, <14> åbner printerudgangen på terminalen (den interfacet tapper informationen fra), <15> lukker den igen, <13> fører cursoren til liniens første position, og en PAGE på 0 sørger for at "gøre linien uendelig lang"; alt sammen resulterer det i, at formatet bliver udskrevet på samme linie hele tiden, og billedet derfor ikke "ruller".

Man skal bare huske efter hver udskrift at indskyde en eller anden forsinkende rutine, så plotteren kan nå at følge med.

De videre perspektiver i systemet

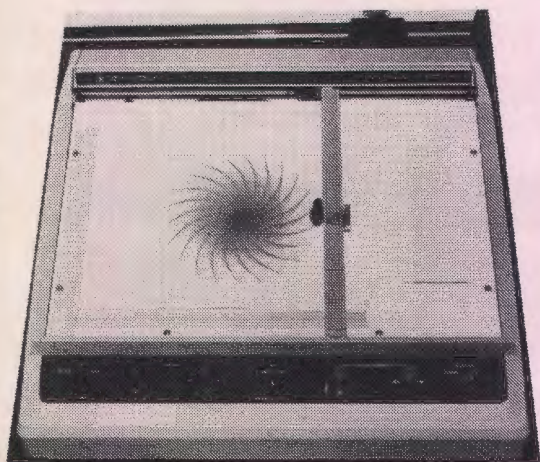
Et andet af de krav jeg stillede, da jeg startede, var, at systemet skulle være "generelt", altså let at ombygge til alle andre plottere end vores egen HP. Systemet har, som det er nu, en fuldstændig generel modtagerdel, men senderdelen er lidt mere speciel, men udgør mindre end en femtedel af konstruktionen og er derfor let at ændre. Systemet er endda så generelt, at det tillader tilkobling af analoge plottere med meget små ændringer. Systemet kan derfor anvendes af enhver, der har en "løs" plotter stående og som gerne vil have den tilkoblet en computer; kravet er dog, at transmissionen foregår med ASCII.



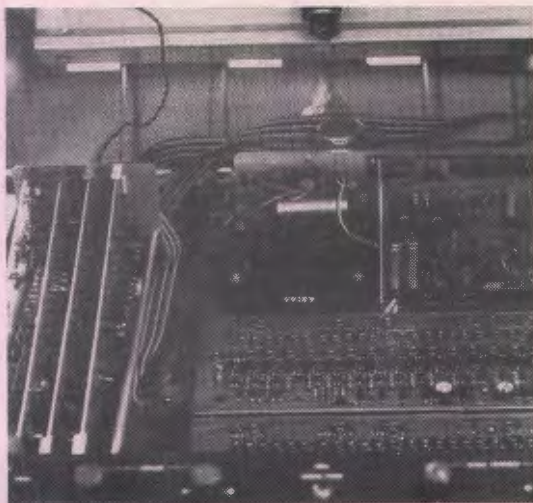
Det analoge interface til naturfagsafdelingen.

Økonomien

Trods alle smukke ord er der dog visse problemer ved at udvikle nyt udstyr. Der ligger jo en masse timer i at udvikle en nykonstruktion på papiret, især når den skal tilpasses noget allerede eksisterende som vores HP-plotter, og når man bliver bedt om et skøn over omkostningerne ved en nykonstruktion, er det fuldstændigt umuligt, da man jo aldrig på forhånd ved, hvilke problemer man støder på, og hvor lang tid det tager at løse dem. Da jeg blev bedt om at gå igang med at konstruere et nyt plottersystem, blev jeg også bedt om et skøn over omkostningerne, og jeg gav et skøn, der dengang virkede realistisk. Den endelige pris blev godt det dobbelte af mit startskøn, og derfor diskuterer visse kræfter her på stedet, om man i det hele taget kan betale for udviklingsarbejde. Nu ligger det jo sådan at 80-90% af arbejdet med en prototype er udviklingsarbejde, men ikke et ondt ord om de diskuterende herrer. Et andet problem er, at man selv er nødt



Plotteren i nærbillede.



Et kig indenfor i plotteren. De to print, der ligger ned i forgrunden, er DAC'erne. De lodretstående til venstre er det gamle styresystem, og de lodretstående i baggrunden til højre er driversystemet til motorerne. De to bændelkabler danner forbindelsen til interfacet.

til at lægge pengene ud til komponenter o. lign., og derefter må vente 3-4-5-6 måneder med at få udlægget ind igen (og jeg er kun en stakkels, fattig gymnasieelev). Dette er ikke skrevet for at så splid, men kun for at belyse de forhold, man som elev ligger under for.

Vurdering

Jeg startede projektet med at stille en række krav til slutproduktet, og alle disse krav må siges at være opfyldt, og brugervenligheden turde fremgå af, at systemet er i næsten konstant brug, og at ingen af brugerne har haft problemer med anvendelsen af det. Systemet har i skrivende stund kørt upåklageligt en måned (her til begyndelsen af januar), og for en gangs skyld er jeg selv tilfreds med en konstruktion, som har kostet flere timer end noget projekt tidligere. Uden at tage munden for fuld tror jeg, at jeg kan sige, at det har været muligt med lidt omtanke, relativt enkelt udstyr og få penge at skabe et semi-professionelt plotter-system, der i sine specifikationer ikke står tilbage for væsentligt dyrere kommercielt udstyr. Det skal her til slut bemærkes, at jeg også har konstrueret et analogt interface til plotteren, så skolens naturfagsafdeling har fået en afløsning for den gamle analoge plotter, jeg omtalte i starten, og det endda til en meget billig penge!

Yderligere information kan indhentes ved at ringe eller skrive til mig. Jeg er meget interesseret i at slå en lille sluder af med folk, som synes at det er ulejligheden værd at snakke med mig, og jeg er meget interesseret i at komme i forbindelse med folk, som har tumlet med de samme idéer som jeg, og evt. er stødt på problemer, da det

DET VIGTIGSTE...



Lærebøger der passer til datamaten

- * Pædagogisk udformede lærebøger der hjælper eleven til hurtigt at forstå datamaten. På dansk - selvfølgelig
- * Også bøger for den, der vil vide mere om datamaten indretning, måledatamatsystemer og programdokumentation
- * Ingen uoverensstemmelser mellem datamat og lærebog
- * Bestil et sæt - læs - og bliv overbevist

Videointerface

- * Lader hele klassen se med på en datamat
- * Tilsluttes skolens videoanlæg eller et almindeligt TV
- * Pris: 650,- excl. moms

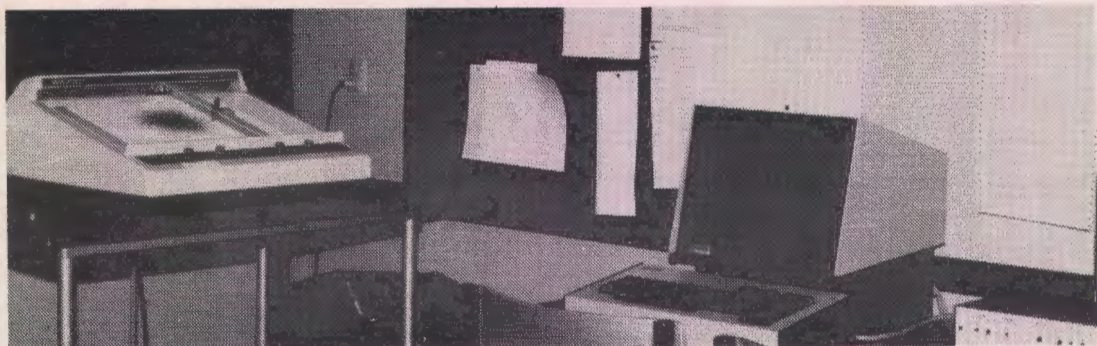
Priseksempel:

Grundsystem m. CPU, tastatur, dataskærm og kassette	8.990,-
System med grundsystem, dobbelt floppy disk (2 x 80K), ABC P80 printer og interface for denne	28.990,-
Klassesæt m. 6 x grundsystem og ABC P80 printer med omskifter for 6 x ABC 80	63.940,-

Alle priser forstås excl. moms

DATAUDSTYR FRA SC METRIC AIS

DATAAFDELINGEN, SKODSBORGVEJ 305, 2850 NÆRUM, TLF. (02) 80 42 00



Et kig ud i datarummet.

kan være meget lærerigt at udveksle erfaringer. Da der givetvis er mange, der går rundt med idéer til lignende projekter, håber jeg med denne artikel at være med til at motivere andre med tilstrækkelig elektronisk baggrund til selv at gå igang.

(I artiklen er bevidst undgået brug af den faglige terminologi undtagen ganske enkelte steder).

Nils R. Pedersen

Blommevej 15, 8600 Silkeborg

Tlf. (06) 81 01 44 eller (06) 82 52 80



Eksempel på hvad man kan lave på det nye plottersystem.

69! 100! 69!

— Eller hvorledes 100 udråbstegn bliver til 158 cifre

Af Chr. Michelsen

Hvad er det største tal man kan tage fakultet af? Det er selvfølgelig et dumt spørgsmål, men hvis vi tænker på en lommeregner, er det som bekendt 69, idet $69! = 1,7112 \cdot 10^{98}$, altså et 99-cifret tal. Så store tal ser man ikke hver dag på gaden, og for de sidste cifres vedkommende ej heller på lommeregneren; men hvad så med computeren da?

Vores mini har ikke nogen fakultetsknap, og grundig inspektion af SYS'erne hjælper heller ikke, så vi må selv programmere. F. ex. således:

```
0010 INPUT "ET TAL: ", N
0020 LET F=1
0030 REPEAT
0040   LET I=I+1
0050   LET F=F*I
0060 UNTIL I=N
0070 PRINT "N! ="; F
0080 END
```

En række skuffelser med negativt og brudent input, for slet ikke at tale om 0, antyder at programmet trænger til nogle forbedringer. Og da vi højst får 6 cifre og desuden ryger i overflow ved $N = 57$, står det klart, at vi må gå helt anderledes radikalt til værks.

Resultatet blev MULTI N!

Først sikrer vi os en fornuftig udgangsværdi, linie 50-70, så opretter vi en begyndelsesværdi, der også gælder når $N = 0$, og noterer lige at den er 1-cifret, linie 90.

Derefter går vi igang med multiplicere med faktorerne i stigende rækkefølge - ciffer for ciffer - ganske som var det med papir og blyant; husk at få menten med, linie 100 - 190.

Ved hver multiplikation finder vi den næste mente ved at fjerne det bageste ciffer i det aktuelle produkt, linie 150, hvorefter vi sikrer os, at det alligevel er dette ciffer, der står tilbage ved at fjerne de eventuelle andre cifre i produktet, linie 160. Antallet af cifre vokser selvfølgelig, linie 180, men til sidst er vi færdige, og det er nu

blot et spørgsmål om at få resultatet til at stå pænt, linie 230 - 350.

(Bemærk hvorledes det kan være lige så omfattende som selve beregningen).

Programmet er ikke ligefrem hurtigt, så hvis

man har travlt, er der sikkert tid at spare ved at udnytte alle 6 cifre i hvert af vektorens elementer. Men så går det ud over den pædagogiske dimension.

```
0020 REM BEREGNER N! I MULTIPRÆCISION
0030 REM REV 28/2 80
0040 DIM FAKULTET(160)
0050 REPEAT
0060   INPUT "HELT TAL N TIL N! : ",N
0070   UNTIL N>=0 AND N<=100 AND INT(N)=N
0080   REM -----
0090   LET FAKULTET(1)=1; CIFRE=1
0100   FOR FAKTOR=1 TO N
0110     LET CIFFER=0; MENTE=0
0120     REPEAT
0130       LET CIFFER=CIFFER+1
0140       LET FAKULTET(CIFFER)=FAKULTET(CIFFER)*FAKTOR+MENTE
0150       LET MENTE=INT(FAKULTET(CIFFER)/10)
0160       LET FAKULTET(CIFFER)=FAKULTET(CIFFER)-10*MENTE
0170       UNTIL CIFFER>=CIFRE AND MENTE=0
0180       LET CIFRE=CIFFER
0190     NEXT FAKTOR
0200   REM -----
0210   PRINT
0220   PRINT
0230   PRINT "   NÅR N =";N; "ER"
0240   PRINT "       N! =";
0250   LET CM50=CIFRE MOD 50
0260   LET TOMME=CM50 DIV 10-(CM50 MOD 10=0)
0270   PRINT TAB(65-CM50-TOMME);
0280   FOR I=CIFRE TO 1 STEP -1
0290     PRINT USING "#",FAKULTET(I);
0300     IF (I-1) MOD 10=0 THEN PRINT " ";
0310     IF (I-1) MOD 50=0 THEN
0320       PRINT
0330       PRINT TAB(11);
0340     ENDIF
0350   NEXT I
0360 END
```

NÅR N = 10 ER
N! =

3628800

NÅR N = 100 ER
N! =

93326215

4439441526 8169923885 6266700490 7159682643 8162146859
2963895217 5999932299 1560894146 3976156518 2862536979
2082722375 8251185210 9168640000 0000000000 0000000000

Almindelige oplysninger om foreningen

Bestyrelsens sammensætning:

- Formand:** ERLING SCHMIDT
Revlingebakken 40, II, 9000 Ålborg, tlf. (08) 18 53 66.
- Næstformand:** WILLY KJELLBERG CHRISTENSEN
Strandpromenaden 32, 4900 Nakskov, tlf. (03) 92 30 34.
- Sekretær:** FRITZ G. KNUDSEN
Kollerupvej 17, 8900 Randers, tlf. (06) 43 49 04.
- Kasserer:** TORBEN HØIRUP
Karl Withsvej 2, 5000 Odense C, tlf. (09) 14 33 53.
- HUGO JØRGENSEN
Olivenvvej 11, Helsted, 8900 Randers, tlf. (06) 42 37 91.
- GERD BELHAGE
Slettebjergvej 7, 2750 Ballerup, tlf. (02) 97 10 46.
- TORSTEN ALF JENSEN
Langemarken 27, 5762 Vester Skerninge, tlf. (09) 24 22 35.

Henvendelser til foreningen:

Indmeldelser, adresseændringer o.l. til kassereren:

FORENINGEN FOR DATALÆRE OG ANVENDELSE AF EDB I
UNDERVISNINGEN
Rismarksvej 80, 5200 Odense V, tlf. (09) 16 86 50.

eller til privatadressen.

Årskontingent: 90 kr. incl. blad. Studerende 45 kr.

Øvrige henvendelser til formanden.

BLADET:

Ansvarshavende redaktør:

TEDDY LANG PETERSEN
Holstedvej 7, 5200 Odense, tlf. (09) 16 90 56.

Henvendelser vedr. annoncer/stof:

Til redaktøren.

Vedr. manuskripter til Datalære

Maskinskrevne manuskripter er velsete. Håndskrevne manuskripter må være let-læselige, og kun den ene side af papiret bør forsynes med tekst.

Ønske om bibeholdelse af afvigelser fra den "gængse" retskrivning og/eller tegnsætning bedes angivet på manuskriptet.

➔ **OBS! OBS!**

Stof til næste nummer af bladet skal være redaktionen i hænde senest mandag, den 18. august 1980.

Dansk Data Elektronik

Drejer det sig om dataanlæg til undervisningssektoren? - køb

GODT GREJ, -DER IKKE KOSTER EKSTRA!

SPC/1 BORDDATAMAT opfylder selv de største krav til en effektiv undervisning i datalogi.

De fleste anskaffer sig model SPC/1 fordi den til enhver tid kvalificerer sig blandt lignende systemer -og ikke så meget fordi

DEN ER DANSK

SPC/1 kan ligeledes anvendes i den administrative sektor. Til f.eks.:

Skemaplanlægning,
Skemaændringer,
Elevdisponeringer,
Vikartilkald, m.v.

SPC/1 dataanlæg består af:
Skærmterminal, en printer
samt en centralenhed, der
er lager for alle infor-
mationerne.

Professionelt ville
man sige, at SPC/1
er et køreklart
system, der kan
støtte både COMAL
og Pascal
multibruger-
konfiguration.

dde

**Dansk Data
Elektronik ApS**
Herlev
Hovedgade 207,
2730 Herlev
02/84 5011

