

UDGIVET AF

Z80 BRUGERGRUPPEN

6. ÅRGANG NR. 9

## MEDLEMSMØDE MEDLEMSMØDE

*Julehygge.*

Vi vil lave lidt julehygge, som vi plejer på denne årstid, med gløg og andet. Vi håber endvidre at kunne nå at fremvise det færdig modem og fortælle lidt om dettes virkemåde og brug.

*Torsdag den 12 - 12 - 85 Kl. 19.30 til ca 22.30*

### RECKU

Vermundsgade 5.

Auditorium 18 AB(C) på 1. sal

2100 København Ø

P.B.V. Rene' Hansen.



### INDHOLD:

- 3: Programmerbare funktionst
- 7: Z80-o-Man
- 8: Programbiblioteket - Nyheder
- 10: Autostart under CP/M
- 11: Pascaltest?!
- 12: BDOS del 2
- 18: Billede af MODEM
- 19: Ole hjørne: Bemærk
- 21: STAT - et CP/M program
- 22: Læserbreve
- 24: Resten af annoncerne

December 1985

ALMINDELIGE OPLYSNINGER OM FORENINGEN

HENVENDELSE TIL FORENINGEN TIL FORRETNINGSFØREREN:

Ulla Hansen  
Dronning Dagmarsvej 116, 3650 Ølstykke  
Mandag - fredag kl. 19.00-21.00 på telefon 02 17 77 05

Hertil skal rettes henvendelse om indmeldelse, adresseforan-  
dring, salg af foreningens blade, CP/M mapper og lign.

Øvrige henvendelser af generel art til formanden. Stof og  
annoncer til foreningens blad, samt køb af CP/M programmer ved,  
henvendelse til Asbjørn Lind.

Indmeldelsesgebyr: 25.00 kr.  
Kontingent 1.7.85 - 1.7.86. 200.00 kr.

Annoncering for medlemmer er gratis i Z80 NYT. For andre 250 kr.  
pr. A4 side.

\*\*\*\*\*

\* Bestyrelsesmedlemmer: \*

\*\*\*\*\*

*Formand:	René Hansen	*	
*	Dronning Dagmarsvej 116	*	
*	3650 Ølstykke	*	
*	Tlf. 02 17 77 05	*	
*	Kl. 19.00 - 21.00 alle dage	*	
*		*	
*Næstformand:	Jesper Skavin	*	
*	c/o Kate Hansen	*	
*	Tagensvej 205 4 th	*	
*	2400 København NV	*	
*	Tlf. 01 85 59 66 (Ikke weekend)	*	
*		*	
*Teknisk redaktør:	Ole Hasselbalch	*	
*	Vibeskrænten 9	*	
*	2750 Ballerup	*	
*	Tlf. 02 97 70 13	*	
*		*	
*	Frank Damgaard	Per Thomsen	*
*	Kastbjergvej 26A	Ulspilager 75	*
*	2750 Ballerup	2791 Dragør	*
*tlf. 02 97 10 20 (inden 20.00)		01 53 50 30 (inden 21.00)*	*

\*\*\*\*\*

Redaktør for Z80 NYT:

Sidste frist for ind- Asbjørn Lind  
levering af stof til Sidevolden 23  
de næste numre er: 2730 Herlev  
11.12.1985 Tlf. 02 91 71 82 (man, ons og tors 20-21)

**Programmerbare funktionstaster på Nascom tastaturet (fortsat).**

Denne artikel er en fortsættelse af min artikel fra forrige nummer om programmerbare funktionstaster på Nascom-tastaturet. I denne del vil jeg omtale, hvorledes man kan omprogrammere tasterne.

Den første betingelse for at kunne omprogrammere funktionstasterne er, at man fra et vilkårligt program kan finde tabellen. Jeg har valgt at klare dette problem ved at lægge tabellen et fast sted i BIOS'en (mere præcist i adresse 3EH relativt til starten af BIOS).

Hvis man har et selvskrevet program, der har brug for sit eget sæt værdier af funktionstasterne, er det nu ikke noget problem i starten af dette program at indsætte en lille stump kode, der indsætter en ny tabel i BIOS'en.

Men hvad skal man gøre, hvis man har et "købeprogram", som gerne skulle have sine egne særlige værdier af funktionstasterne? Ja her må man lave et lille program i assembler eller Pascal, der først indlæser den ønskede tabel og herefter starter det egentlige program. I slutningen af denne artikel er vist et assembler-program, som jeg bruger i forbindelse med WordStar.

Nu er der kun et enkelt problem tilbage. Det skulle jo gerne være således, at når man afslutter et program som WordStar og vender tilbage til CCP'en, ja så skulle CCP-versionen af funktionstasterne gerne være bragt på plads igen. Man kan selvfølgelig klare dette ved at have en uberørt version af tabellen liggende et andet sted i BIOS'en, men ærlig talt synes jeg, at det er pladsspild, når den ligger et sted på disketten. I stedet finder vi det sted i BIOS'en, hvor systemet indlæses fra disketten under warmboot. I MAP80-BIOS'en ser det således ud:

```

;Read/Write CCP and BDOS
DBOOT: LD C,A                ;Save read/write
        PUSH BC
        CALL CLEAN           ;Clean up 2797 and Xebec
        LD HL,CCP16         ;CCP load address
        LD DE,0001H        ;Start at track 0 sector 1
        POP BC
        LD B,CPMLN/HSTSIZ   ;Sectors to load
LDCPM:  PUSH DE
        PUSH HL
        PUSH BC
        XOR A                ;Logical drive A
        CALL RWSEC          ;Read sector
        POP BC
        POP HL
        LD DE,HSTSIZ
        ADD HL,DE           ;Next DMA address
        POP DE
        RET NZ              ;I/O error
        INC E                ;Next sector
;Test for next track
        LD A,E
        CP SPTF            ;Sectors per track
        JR NZ,SECOK
        LD DE,0100H        ;Track 1 sector 0
SECOK:  DJNZ LDCPM
        XOR A
        RET
    
```



Bemærk, at både CCP og BDOS indlæses, dvs at den næste sektor på disketten nu indeholder starten af BIOS'en og dermed også vores tabel. Hvis vi nu læser denne sektor over i sektorbufferen (i MAP80-BIOS'en kaldet HSTBUF), kan vi kopiere tabellen på plads.

I stedet for de to sidste instruktioner (XOR og RET) indsætter vi derfor følgende:

```

;Set default function key table
  XOR A
  LD HL,HSTBUF           ;Address of BIOS buffer
  CALL RWSEC            ;Read 1. sec of BIOS TO HSTBUF
  LD HL,HSTBUF+(FNCSTO-BIOS) ;Point to table in HSTBUF
  LD DE,FNCSTO          ;Point to table in BIOS
  LD BC,TABLEN
  LDIR                  ;Move it

  RET

```

I Geminis BIOS er der ikke valgt de samme navne, men selve koden ser omtrent ligesådan ud, så det skulle ikke være svært at finde stedet.

Og så til programmet, der starter WordStar. Det er skrevet til M80 assembleren.

```

.Z80
; PROGRAM TO SET FUNCTION KEY TABLE AND START WordStar
;
; EQUATES

TABOFF EQU 003EH
TABLEN EQU 100H

CPMBUF EQU 0080H

FNCSTO: DEFW 0 ;Pointer

;FUNCTION KEY TABLE
FNCTAB: DEFB 21H+80H ,16H ;! -> ^V
        DEFB 22H+80H ,11H,11H ;" -> ^Q^Q
        DEFB 23H+80H ,02H ;# -> ^B
        DEFB 24H+80H ,0BH,0BH ;$ -> ^K^H
        DEFB 25H+80H ,0BH,03H ;% -> ^K^C
        DEFB 26H+80H ,0BH,16H ;& -> ^K^V
        DEFB 27H+80H ,0BH,0BH ;' -> ^K^K
        DEFB 28H+80H ,0BH,02H ;( -> ^K^B
        DEFB 29H+80H ,11H,13H,0DH,05H ;) -> ^Q^S CR ^E
        DEFB 2AH+80H ,11H,06H ;* -> ^Q^F
        DEFB 2BH+80H ,11H,01H ;+ -> ^Q^A
        DEFB 2DH+80H ,12H ;- -> ^R
        DEFB 30H+80H ,03H ;0 -> ^C
        DEFB 31H+80H ,11H,17H ;1 -> ^Q^W
        DEFB 32H+80H ,11H,1AH ;2 -> ^Q^Z
        DEFB 33H+80H ,17H ;3 -> ^W
        DEFB 34H+80H ,1AH ;4 -> ^Z
        DEFB 35H+80H ,11H,10H ;5 -> ^Q^P
        DEFB 36H+80H ,11H,04H ;6 -> ^Q^D

```



```

DEFB 37H+80H ,11H,13H ;7 -> ^Q^S
DEFB 38H+80H ,11H,18H ;8 -> ^Q^X
DEFB 39H+80H ,11H,05H ;9 -> ^Q^E
DEFB 3AH+80H ,11H,12H ;: -> ^Q^R
DEFB 3BH+80H ,11H,03H ;; -> ^Q^C
DEFB 3DH+80H ,0CH ;= -> ^L
DEFB 5FH+80H ,09H ;_ -> ^I
;GRAPH KEY DOWN
DEFB 38H+0COH ,0BH,19H ;7 -> ^K^Y
DEFB 39H+0COH ,11H,19H ;8 -> ^Q^Y
DEFB 30H+0COH ,11H,0BH ;9 -> ^Q BS
DEFB 2DH+0COH ,14H ;- -> ^T
DEFB 3BH+0COH ,01H,14H ;; -> ^A^T
DEFB 3AH+0COH ,07H ;: -> ^G
DEFB 21H+0COH ,0BH,13H ;! -> ^K^S
DEFB 22H+0COH ,0BH,04H ;" -> ^K^D
DEFB 23H+0COH ,0BH,18H ;# -> ^K^X
DEFB 24H+0COH ,0BH,11H ;$ -> ^K^Q
DEFB 3DH+0COH ,0BH,06H ;= -> ^K^F
DEFB 2BH+0COH ,0BH,0CH ;+ -> ^K^L
DEFB 2AH+0COH ,0BH,12H ;* -> ^K^R
FTABE: DEFB 80H

DEFS TABLEN-(FTABE+1-FNCSTO),80H

FILNAM: DEFB 'WORDSTAR'
FILNME: DEFS 8-(FILNME-FILNAM),0

START: LD HL,(0001H) ;Get address of BIOS+3
LD DE,-3 ;Subtract 3 to get BIOS
ADD HL,DE

EX DE,HL
LD HL,-1600H ;Subtract 1600H to get CCP
ADD HL,DE

LD SP,HL ;Set stack
PUSH HL ;and store address of CCP

LD HL,TABOFF+TABLEN-1
ADD HL,DE ;End of table ...
LD (FNCSTO),HL ;.. into pointer
DEC H
INC HL ;Start of table space in BIOS

EX DE,HL ;DE=destination
LD HL,FNCSTO ;HL=source
LD BC, TABLEN ;BC=length of table
LDIR ;Move table to BIOS

EX DE,HL ;DE=address of file name
POP HL ;CCP address to HL
PUSH HL
LD BC,7 ;Add 7 to get address of command
ADD HL,BC ;buffer
PUSH HL ;Store address
INC HL
INC HL
LD A,(HL) ;Any drive specifikation?
DEC HL
CP ':'

```



```

LD      A,0                ;Number of bytes in buffer
JR      NZ,NODRIV         ;Jump if not

INC     HL
INC     HL
INC     A
INC     A                ;Two bytes in buffer

NODRIV: LD      BC,FILNME-FILNAM ;BC=length of file name
ADD     A,C              ;Add to buffer length
EX     DE,HL
LDIR                    ;Move to CCP command buffer

;Move parameters
LD      HL,CPMBUF        ;Default file buffer
LD      C,(HL)           ;Length of parameter string
INC     HL
ADD     A,C              ;Add to buffer length
INC     C                ;Length zero?
DEC     C
JR      Z,NOPARA        ;Jump if so
LDIR                    ;Move parameter string

NOPARA: EX     DE,HL

LD      (HL),0           ;Set trailing zero

POP     HL               ;Command buffer
LD      (HL),A           ;Set length

DEC     HL               ;Point to bufer length ..
LD      A,(HL)          ;.. and get it.
ADD     A,0BH+1         ;Add buffer offset + 1 ..
LD      L,A              ;.. to get address of pointer
LD      (HL),0BH        ;LSB of pointer (buffer start)
INC     HL
LD      (HL),H           ;MSB of pointer

LD      C,0              ;Drive/user no
RET                    ;Go to CCP

END     START

```

Svend Daugaard Pedersen  
nr. 333      ★

#### Bedstefar til salg

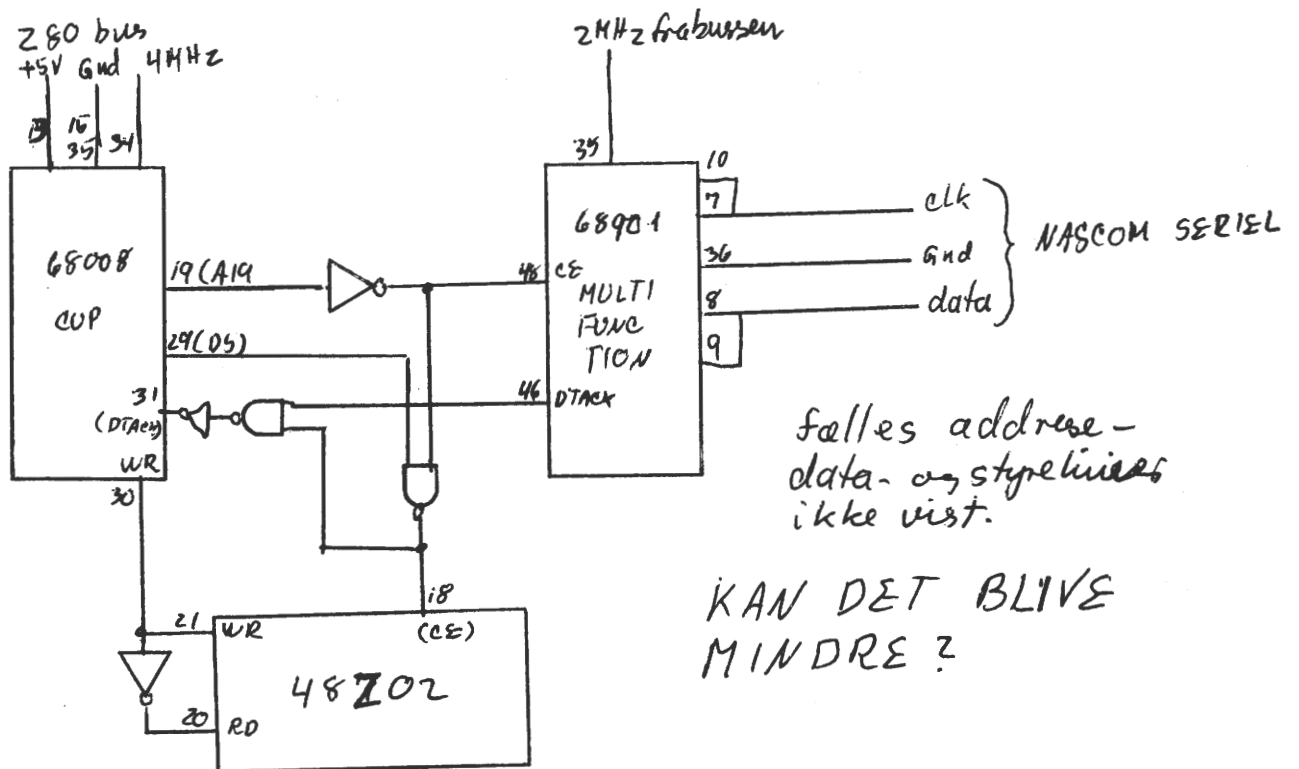
Datapoint 2200 med rc2240 matrix printer benyttes med Nascom 1 eller 2, mikroprofessor eller anden Z80-maskine (Nyværdi: 3-5.000 kr.) Til Datapoint medleveres basic, assembler og databus, samt en række utility programmer og FULD dokumentation af maskinen. Det vil være en betingelse at denne dokumentation stilles til rådighed for andre interesserede medlemmer af klubben. Maskinen vil være oplagt som terminal til moderne processorkort, idet den har 12\*80 tegns display, dobbelt båndstatiin med læsning i begge retninger (blok opdelt) og meget felksibel ekstern bus. Programmering i maskinens assembler er som at gå med for små sko, men det kan lade sig gøre. (En kaput maskine følger med som reservedel for en halv flaske gammel dansk) E. Sjørlund 03 47 90 97 mellem 15 og 21.



# Z80-O-MAN

Hvad laver en Z80-o-man sådan en søndag morgen efter krydderen og sidste nummer af Z80-Nyt er læst?

Hvis man vil være med i ræset, er det åbenbart nødvendigt med en RAM-disk og en 16/32 bits processor. Costgold til 7550 kr kan måske lige gå an, men er det nu nok? Er fremtiden ikke en 68000 (Typenummeret kommer mere og mere til at ligne prisen) Krokodillen er for længst arkiveret i den volatile ram og det der med at gå ud og kigge på damer må vist også opgives. Men hvis man nu alligevel hopper på 16/32 bits vognen hvad så? De færdige fabriksløsninger synes mig uoverkommelige i penge, altså selvbygger og -konstruktion. Men så 68000, Z8000 (nå nej den er da vist gået i sin grav igen) eller 8086 (forleden hørte jeg for resten tallet 80486) eller måske 8-bits udgaverne af de samme? 68008 ser umiddelbart mest tillokkende ud. Det er jo i grunden en 32 bits processor - jo, men den har jo ingen segmenteringsfidus - nej men til gengæld en hulens masse adresseringsmuligheder og 17 registre eller endda 18 når PC tælles med. 32 bits forstås, og 8 MHz selv med et sløvt lager. En ihærdig søgning i Life-Ware-Memory bringer et minimissimum system for dagen (se en skitse senere). Det er ikke meget det kan, men man får chancen for at se begavet og erfaren ud ved kommende medlemsmøder og det er da også noget værd. Og NASCOM'en har stadig en mening med tilværelsen:



Jeg har fløttet mig med en 68901 der er en helt masse bl.a. både serial og parallel ind/ud. 6850 UART er helt tilstrækkelig.



2 kB lager er ikke meget. Husker veteranerne gode gamle dage med NASCOM 1's 4 kB incl. videoram og 2 kB O/S? 48Z02 er en statisk RAM med batteriback-up. Det er nødvendigt, da man starter processen ved at anbringe dimsen i sit Z-80 system og læser en amputeret operativsystem ind (reset, serin, serout), men resten af de 2 kB efter flytning til den "store" computer bruges til arbejds-lager incl stak.

Med en fornuftigt udlægning på et prototypekort er vejen for senere udbygning med 256 kB dynamisk ram mulig, og holder man ørerne stive under studiet af håndbogens afsnit om Bus-Arbitration er det efter min bedste overbevisning endda muligt at sætte hele gøgemøget på Z-80 bussen og bruge lageret som ramdisk for Z-80 systemet og endda bruge dette som interface til 68008 systemet. Der bliver nogle problemer med at få CPU'en til at læse fra ramdisken men der kan man evt bruge en DMA, eller lade 68008 ekspedere over på det normale lager

Et sidste spørgsmål i hele historien er, om man vil understøtte 68000-ordren TAS (test and set), som forudsætter read-modify-write lager og altså bliver et spørgsmål om - så vidt jeg umiddelbart kan overskue - 8stk 74LS 244. Det er dyrt for en enkelt ordre.

Når nu early-write er så enkelt!

Er der nogen der overvejer de samme problemer, eller er jeg helt alene i verden?

*Christian Laustsen*  
Christian Laustsen



#### Programbiblioteket

\*\*\*\*\*

Der er kommet nye volumen til programbiblioteket (20 stk.), de fordeler sig på SIG/M og UK User Group. Sidstnævnte har vi nogle af i biblioteket i forvejen, men de er blevet omdøbt til CPMZ80, men foreløbig vil de optræde under CPMUK.

Her følger en kort oversigt over disse nye volumen, som jeg personlig anser for meget spændende.

#### CPMUK:

- 018 ProPascal programmer: Sorteringsmetode, grafisk anskuelse, ASCII til EBCDIC, File compere, Crossreference, Read sectorer, uafhængig terminal disk doktor, sorteringsrutine uafhængig af memorystørrelse m.m.
- 020 Z8 Cross Assembler (der virker, den har allerede vist sin styrke!), Z80 assembler. Begge med fuld dokumentation på disk.
- 021 Edinburgh Compatible Context Editor, winchester backup, Z80 disassembler og Gemini utilities: Diskeditor, filecompare
- 025 Creator & reporter til Mbasic v. 5.2 (eller nyere)- database-program (professionelt fremstillet!!), Basic statistikanalyse-program, Dbase program, der kan udskrive label, skrive breve og fylde navne i.
- 026 Karactersæt til Gemini SVC, diskutility DU3 (CP/M Plus), PLINK II terminal og fileoverføringsprogram, 68000 monitor, udvidet filsøgningsprogram, ny udgave af SHOW file viewer.





SIG/M

- 176 Nye udgave af SD, CPM3 utility og datorrutiner. Betinget submit fil, Z80 USQ. Alt i LBR-filer incl LU pgr. (1.6.1984)
- 177 Fra Australien: Accounts i SC2, emulator for CDOS 2.35 under CP/M 2.2, PIP patch for handshake file transfer, viser memory map, ny sort og pack, MSDOS 1 & 2 bad block locator, patch for WS 3 og Epson MX80 (12.6.1984)
- 193 Fra Toronto RCP/M: bidirectional pager for både squeezed og almindelige filer. Oki, Epson kolonne udskrift i 2 kolonner, memory til DATA sætninger i Mbasic, sæt MX80, SC2 skabelon pgr., password for .COM fil, rename file af samme ext., Z80 debugger. (7.9.1984)
- 204 Forth-83 med Metacompiler seneste udgave (16.11.1984)
- 210 CP/M-86 utilities: kø til printer, kommandolinje interpreter, katalogopbygger, kopierer mellem userområder, squeezer og unsqueezer. (18.1.1985)
- 217 CP/M 80 emulator til CP/M 86, DbaseII toolkits bibliotek, CP/M 86 LU. (18.1.1985)
- 222 Nye udgaver af flere utilities (16.3.1985)
- 225 FX-80 karaktersæt til brug i forbindelse med WS, videnskabelig font til MX-80, Biokemisk engineering games and simulation (19.4.1985)
- 226 Tilføj eller slet ':' i ASM-filer, skift mellem MM/DD/YY og DRI format, era, allocation dump, Sigi Kluger utilities. Shell Metzner sort, (19.4.1985)
- 231 100k CP/M + bibliotek (Bringing up), C & disk-controller(God-bout), Assembler udvidelse til MSbasic, logosk navne translator (19.7.1985)
- 232 Big Board Bios, Kermit v. 3.9 Z80, Modem7 til Big Board, ny udgave af Xlate2: 8080 til Z80 oversætter (19.7.1985)
- 235 BASICCODE radiokommunikation på hollandsk i Mbasic og i Cbasic med translator, udvidelse til Fig Forth, Tic-Tac-Toe, Games, Screen Digital clock, holidays, kalender, sort (19.7.1985)
- 236 Bios til CP/M+ til Disk-Jockey controller, 25 Kb artikel om CP/M+, print spooler, Ny LISP (19.7.1985)
- 237 Seneste PILOT, Solving Deductive reasoning Puzzles (19.7.1985)
- 239 Symbol Z80 debugger (full screen animated), 16 breakpoint, 20 kommandoer, fuld dokumentation. (16.8.1985)

Der vil i de kommende numre komme uddybende omtale af indholdet på flere af disketterne, har man lovet mig!

A.L.



**SÆLGES:**

Sælges 1 stk originalt RC 702 Picolo Diskdrive af typen YD-274, som er et 40 spors dobbelt sidet 5.25" drive. Fabriksnyt.

Pris.. 1700,-

Forretningsførren. 02 177705.



## Autostart i CP/M.

For nogen tid siden blev jeg spurgt om, hvorledes man lavede autostart under CP/M, altså fik indlæst et program automatisk ved "power on" og RESET. Svaret har sikkert interesse for flere end denne ene person, så jeg har valgt at fortælle lidt om det her i bladet.

For at forstå, hvad man skal gøre, er det nødvendigt at se lidt på indledningen af CCP'en. Den ser således ud:

```

ORG CCP      ;START PÅ CCP'en

JP  COLD
JP  WARM

DB  BUFLen-1 ;INPUTBUFFERENS LÆNGDE
DS  1        ;ANTAL TEGN I AKTUELLE KOMMANDO
DS  BUFLen   ;INPUTBUFFEREN
DS  2        ;PEGEPIND TIL KOMMANDO

```

Man ser, at der to indgange til CCP'en. Den ene ligger i adresse CCP og den anden i adresse CCP+3. Hvis man hopper ind i CCP'en via koldstartindgangen i adresse CCP, undersøger CCP'en først, om der er en kommando i inputbufferen før den scanner tastaturet. Er der det, udføres denne kommando. Hvis man benytter varmstartindgangen i adresse CCP+3, negligeres en eventuel kommando i inputbufferen.

Når et program slutter eller når man trykker ^C, indlæser BIOS'en CCP'en og hopper til denne via varmstartindgangen. Når man derimod tænder for sin maskine eller trykker RESET, boot'es hele CP/M'en påny og BIOS'en vil efter at have foretaget de nødvendige initialiseringer hoppe ind i CCP'en via koldstartindgangen og en eventuel kommando i inputbufferen vil blive udført. Hvis man ønsker at få udført en eller anden kommando (f.eks. at få startet et program), må man altså sørge for at få denne kommando placeret i CCP'ens inputbuffer.

Lad os nu lige se på de næste adresser i CCP-starten. I adresse CCP+6 (altså umiddelbart efter varmstart-hoppet) ligger et tal, der angiver inputbufferens længde. I den normale CCP står der her 7FH, men i visse CCP-erstatninger har man måttet inddrage lidt af inputbufferen for at få plads til de ekstra faciliteter, således at et mindre tal kan forekomme. Herefter følger en byte, der angiver hvor mange tegn den aktuelle kommando i inputbufferen fylder. Her står et nul i den sædvanlige CCP.

Nu følger selve inputbufferen, der strækker sig fra adresse CCP+8 og BUFLen bytes frem (altså normalt frem til og med adresse CCP+87H). I den normale CCP ligger der, som det er fremgået, ikke nogen kommando. Inputbufferen er i stedet fyldt med en copyright-meddelelse fra Digital Research. Guderne (eller DR) må vide, hvad den skal bruges til, for det er mest computer-hacker's der overhovedet opdager, at den står der, og den type mennesker plejer ikke at tage den slags meddelelser alt for bogstaveligt. Under indtastning af en kommando fra tastaturet, bliver kommandoen anbragt i denne buffer med en 0-byte som afslutter.



Umiddelbart efter inputbufferen (normalt i adresse CCP+88H) følger en pegepind, der peger på det sted i inputbufferen, som CCP'en er kommet til i sin fortolkning af en kommando. Her skal normalt stå adressen på inputbufferens start.

For at få anbragt en kommando i inputbufferen, må man rette i MOVCPM eller eventuelt i CPM.COM. Indlæs en af disse ved kommandoen DDT MOVCPM.COM (eller DDT CPM.COM). CCP'en starter nu i adresse 980H, A00H eller B00H afhængig af, om ens system kører med 128, 256 eller 512 bytes pr. sektor (Gemini og MAP80 bruger 512). Nu kan kommandoen og kommandolængden indføres v.h.j.a. S-kommandoen. Kommandoen skal slutte med en O-byte, som dog ikke skal medregnes i længden af kommandoen. Herefter skal man hoppe tilbage til CCP'en (GO) og SAVE det rettede (husk at læse længden ved indlæsningen).

Svend Daugaard Pedersen  
nr. 333



Næste nummers deadline er allerede den 11.12.85, det skyldes, at det skal trykkes inden jul. Det næste blad udkommer altså først i januar 1986, men I ved jo, at der kommer nogle helligdage på denne tid af året. Det næste blad kommer måske først i slutningen af februar, fordi det er redaktørens mening at holde en lille vinterferie!

Asbjørn Lind



Hr. Villadsen prøver medlemmer igen. Sidste gangs pascalsprogram fik ingen løsning, da der ikke var et eneste medlem, der reagerede på udfordringen. Men fat mod, her er et nyt! Der er jo mange dygtige pascalprogramører i blandt os - lad os nu se?

(\* Pascal test 2 \*)

Program test2;

```
function plus1 (i:integer):integer;
begin
  plus1 := i + 1
end;
```

```
function F (function F1 (x,y : real;
                        function f2 (i : integer) : integer
                        ) : real;
begin
```

```
  f:=F1 (3,4,plus1)
end;
```

```
function sum (x,y : real; function F (i:integer):integer): real;
begin
  sum := x + y + F (round(x))
end;
```

```
begin (* Main *)
  writeln (f (sum))
end.
```



BDOS      2.del

Denne gang skal vi se på kontrolfunktionerne til systemet. Disse kald tilbyder programmøren en stor handlefrihed, når han skal bruge hele CP/M systemet.

**System reset: funktion 0**

Denne funktion er beregnet til genstart af CP/M kommandoprocessor (CCP) efter endt programudførelse eller ved brugerafbrydelse. System reset svarer til en JP 0 (jump 0H) eller et Ctrl-C, som tvinger en varmstart (i modsætning til en koldstart, hvor alle parametre og ydre enheder sættes eller initieres). Funktionen afaktiverer alle drive og indsætter A: som det indloggede drev. Funktionen er meget simpel som følger:

```

RESET      EQU    000H      ;SYSTEM RESET NUMMER
            EQU    005H      ;SYSTEM INDGANG
            ORG    0100H
            LD    C,RESET
            JP    BDOS      ;DER KUNNE OGSÅ BRUGES CALL, MEN
                            ;DER VENDES IKKE TILBAGE TIL BRU-
                            ;GERPROGRAMMET!!
    
```

**Get og Set IObyte: Funktion 7 og 8.**

Den almindelige måde, hvorpå CP/M taler med omgivelserne, er gennem logiske enheder. Dette betyder at konsollen, skriveren (list), reader og puncher bliver behandlet som helt almindelig ydre enheder. CP/M tillader, i et vist omfang, at systemet består af flere fysiske ender inden for hver af de logiske. Forbindelsen mellem fysiske og logiske ender foregår gennem IOBYTEN, der normalt er 'gemt' på adresse 0003H. Forfatterne til BIOS-delen af CP/M styresystemet må derfor gennem snedig programmering, undersøge hvilken af f.eks. to forskellige printere, der skal skrives på, når BDOS sender besked om at skrive tegn til printerens. Der er efterhånden udviklet en vis standard, som er kopieret fra en 8-bits maskine udviklet af Intel, som er som følger:

		(SKRIVER)	(PUNCH)	(READER)	(KONSOL)
Logisk enhed		LST:	PUN:	RDR:	CON:
IOBYTE bit		7 6	5 4	3 2	1 0
-----					
Bit mønster					
dec. binært					
0	00	TTY:	TTY:	TTY:	TTY:
1	01	CRT:	PTP:	PTR:	CRT:
2	10	LPT:	UP1:	UR1:	BAT:
3	11	UL1:	UP2:	UR1:	UC1:



Følgende tabel fastlægger de enkelte fysiske enheders tilknytning til ovenstående navne:

- TTY: Teletype konsol med tastatur og printer, som display og mulighed for indbygget papirtape læser og skriver.  
 CRT: En interaktiv katodestråle skærm type med indbygget tastatur og skærm.  
 BAT: En batch processor (bunkeprogramafvikler?) med kortlæser som input og skriver som outputenhed.  
 UC1: En alternativ CRT: enhed.  
 LPT: En linje skriver  
 UL1: En bruger defineret list enhed.  
 PTR: Papir tapelæser  
 UR1: Brugerdefineret 'læser' af tegn enhed.  
 UR2: Brugerdefineret 'læser' af tegn enhed.  
 PTP: Papirtape huller.  
 UP1: Brugerdefineret 'huller' af tegn enhed.  
 UP2: Brugerdefineret 'huller' af tegn enhed.

BDOS understøtter en aflæsning af IOBYTE'ens nuværende værdi og en metode til at ændre den til andre værdier. Hvis en CP/M-80 computer har to forskellige printere sat på systemet på henholdsvis LST: og UL1:, kunne man tænke sig følgende programmering, hvis et program skulle skifte mellem de to enheder.

```

SETIOB EQU 006H ;SET IOBYTE
GETIOB EQU 007H ;GET IOBYTE
BDOS EQU 0005H ;SYSTEM ADRESSE
LSTMASK EQU 11$00$00$00B ;IOMASKE FOR LIST ENHEDEN
LPT EQU 10$00$00$00B ;BITVÆRDI FOR LPT NR. 1
UL1 EQU 11$00$00$00B ;BITVÆRDI FOR LPT NR. 2
ORG 0100H ;PROGRAM START
LD C,GETIOB ;GET NUVÆRENDE IOBYTE VÆRDI
CALL BDOS
AND (NOT LSTMASK) AND OFFH ;BEHOLD ALLE ANDRE BIT
OR UL1 AND LSTMASK ;SÆT FOR SKRIVER 2
LD E,A
LD C,SETIOB ;FUNKTION FOR RESET IOBYTE
CALL BDOS
RET ;UMIDBART TIL CCP
    
```

#### Hent versionsnummer på CP/M-80 system: Funktion nr 12

Ved særlige lejligheder er det nødvendigt for programskrivere at vide, hvilken version af CP/M programmet kører under. Det skyldes, som jeg skrev i sidste nummer, at CP/M styresystemerne ikke er opad kompatible. F.eks. blev random acces filer først indført i version 2.2. Ved retur indeholder HL registret oplysningerne om version og CP/M efter følgende plan:

- hvis H=0 er det et CP/M-80 system
- H=1 er det et MP/M system
- og hvis L=00 er systemet ældre end 2.0
- L=20 er det system 2.0
- L=21 er det system 2.1
- L=22 er det system 2.2
- L=31 er det system PLUS



```

Programmet der aflæser dette er som følger:
GETVERS EQU 00CH ;FUNKTION 12
BDOS EQU 0005H ;SYSTEM ADRESSE
ORG 0100H ;PROGRAM START
LD C,GETVERS
CALL BDOS ;HENT VERSIONSNUMMER
LD A,L
LD (VER),A ;GEM VERSIONNUMMER
RET
VER: DS 1 ;TIL VERSIONSNUMMERET
END

```

**Reset disksystem: Funktion nr: 13**

CP/M systemet har faciliteter til at kontrollere filtilgangen på disketten. Et directory checksum skema, skjult for brugeren, tillader systemet at afgøre, om en diskette har været skiftet inden der skrives på den igen. Men flere brugerprogrammer kan forlange et disketteskift midt i programmet. Det er derfor nødvendigt at gennemføre en resetprocedure, der sletter mindet om forrige diskette og sætter bufferen til 80H, som ved almindelig opstart. Følgende programstump resettet disksystemet.

```

RESET EQU 0DH ;FUNKTION 13
BDOS EQU 0005H SYSTEM . . .
ORG 0100H
LD C,RESET
CALL BDOS ;RESET DISKSYSTEM
RET
END

```

**Get og set usernummer: Funktion 32.**

CP/M systemet tillader disketten at blive opdelt i 15 forskellige brugerområder med hvert sit directory. Dette kan benyttes så man isolerer tekstfiler, programmer og utility på hvert deres område. Fordelen er bedre overskuelighed og mindre mulighed for sammenfald af navne. (I PPAS (eller COMPAS) er det muligt at anvende op til 32 brugerområder, hvis de skiftes fra programniveau og ikke fra kommandoniveau, her vil der vælges med modulus 15 af ønsket nummer). Programstumpen sætter brugerkoden op med 1.

```

GSUSR EQU 020H ;FUNKTION 32
GET EQU OFFH ;GET FLAG
BDOS EQU 0005H ;SYSTEM . . .
ORG 0100H
LD E,GET ;LÆG AN TIL HENT
LD C,GSUSR
CALL BDOS ;HENT NUVÆRENDE NUMMER
INC A ;OPSKRIV MED 1
AND OFH ;MASK TIL MOD(15)
LD E,A ;FLYT TIL SÆT NUMMER
LD C,GSUSR
CALL BDOS
RET
END

```



## SYSTEMFUNKTIONER, DER STYRER DISKETTESTATIONERNE.

## Vælg disk: Funktion 14.

Den simpleste funktion for valg af diskettestation er fra kommandoniveau, hvor man skriver B:, hvis der skal skiftes til drev B: Et program, der udfører den samme funktion følger. Drevnumret følger følgende skema: Drev A: = drev nr. 0, B: = drev nr. 1, aktivt og bruges som det aktuelle drev (nuværende eller default). Det vil det blive ved med indtil næste varmstart, koldstart eller diskreset BDOS funktion.

## Hvilken diskstation? Funktion 25.

Et program kan ved dette kald finde ud af hvilket drev, der er aktivt. BDOS returnerer i A numret på det aktuelle drev. Følgende programstump undersøger om B: er aktivt, og hvis det ikke er tilfældet logges det ind.

```

SELECT EQU 0EH ;FUNKTION 14
ASKDRV EQU 19H ;FUNKTION 25
BDOS EQU 0005H
ORG 0100H
LD C,ASKDRV ;ER DET B:
CALL BDOS
CP 'B'-'A'
RET Z ;RETUR HVIS DET VAR TILFÆLDET
LD E,'B'-'A' ;SÆT TIL LOG DREV
LD C,SELECT ;VÆLG B:
RET
END

```

## Drev status: Funktion 28 og 37:

Diskettedrevets status kan kontrolleres hver for sig ved funktion 28, kan drevet sættes til read only (skrivebeskyttet). Det fås ved LD C,01CH, CALL BDOS. Denne skrivebeskyttelse kan fjernes ved brug af funktion nr. 37. Man kan aktivere alle drev bort fra read only eller man kan aktivere et eller flere ad gangen ved et '1' (bit) i et 16 bit word, der for hver bit relaterer til et drev med drev A: til mindst betydende bit (LSB). Følgende kode skulle resette drev B:

```

RESDSK EQU 025H
BDOS EQU 0005H
LD C,RESDSK
LD D,0000$0000$0000$0010B ; DREV B: SAT BIT
CALL BDOS
RET
END

```

## Hent drive login og read only vektorer: Funktion 24 og 29

BDOS holder øje med hvilke drev, der har været logged ind siden start eller reset. Disse drev betragtes som online, fordi systemet umidbart ved besked om allocations mappen for forbrugte sektorer på disketten og om det er read only eller ej. Funktion 24 tillader programmer at få oplysning om, hvilke der er logget ind af samtlige drev. Vektoren kommer retur i HL og er BIT-map, som ovenstående, hvor '1' betyder, at drevet er bekendt. Mest betydende BIT er drev P: og mindst A: Følgende henter vektoren og gemmer den i et internt dataområde.



```

LOGIN      EQU  018H      ;FUNKTION 24
BDOS       EQU  0005H
           ORG  0100H
           LD   C,LOGIN
           CALL BDOS
           LD   (LOCLOG),HL ;GEM VEKTOR I LOCLOG
           RET
LOCLOG:    DS    2
           END
    
```

På tilsvarende måde afgør BDOS hvilket drev, der er i read only eller i read write status. Et '1' i den tilbageleverede vector betyder read/only. Den følgende kode finder vektoren.

```

ROVEC     EQU  01DH      ;FUNKTION 29
BDOS      EQU  0005H
           ORG  0100H
           LD   C,ROVEC
           CALL BDOS
           LD   (LOCROV),HL ;GEM VEKTOR
           RET
LOCROV:   DS    2
           END
    
```

#### Hent allocation vektor og Diskparameter pointer: Funktion 27 og 31

Disse to funktioner tillader forskellige ting at blive udført. Den første (27) returnerer en adresse i HL, der peger på en BIT-streng i hukommelsen, der holder reder på, hvilke blokke, der er brugt på det valgte drev. Mappen indeholder en BIT for hver blok, der er blevet benyttet startende med det mest betydende bit (MSB) af første byte af strengen. Længden af strengen afhænger af den totalekapacitet på drevet i allokatjonsblokke. Funktion 31 tillader programmer at finde ud af, hvilke karakteristika der er for det valgte drev. BDOS returnerer en adresse i HL, som peger på en tabel på 33 byte, der beskriver udstyret. Oplysningerne i denne tabel indeholder information om: antal direntry, antal allokerbare blokke og indirekte en oplysning om blokkenes størrelse. Det følgende program er et eksempel på, hvordan man kan finde restplads på en diskette. Programmet gemmer oplysningen om restplads i drevet, der er nævnt i FCB'en, i memoryadressen 'KPDISK'.

A.L.

```

;program til at finde restkapacitet på disketten
LOGDRIV   EQU  0004H      ;BESKRIVER NUVÆRENDE DREV
BDOS      EQU  0005H
SLCTDSK   EQU  14        ;VÆLG DISK
GALVEC    EQU  27        ;HENT ALLOKERINGS VEKTOR
GDSKP     EQU  31        ;ADRESSE PÅ DISKPARAMETER
FCB       EQU  005CH
    
```





```

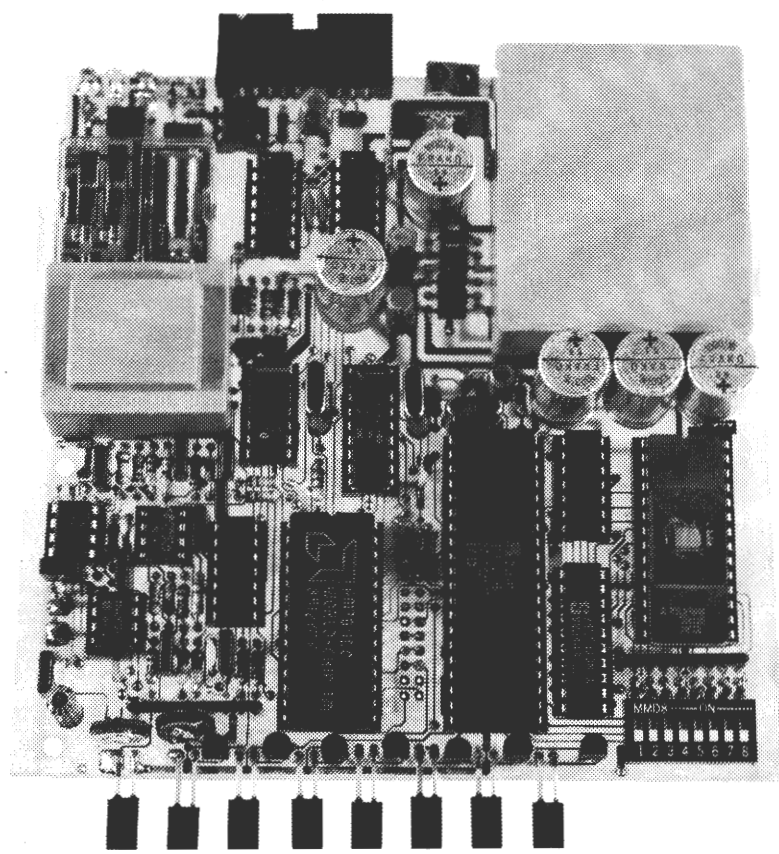
ORG 0100H
SPCGET: LD A,(LOGDIV) ;HENT LOGGED DREV
AND OFH ;AFMASK USERNUMMER
LD (SAVDRIV),A ;OG GEM DISKNUMMER
LD A,(FCB) ;ER DET DEN SAMME
DEC A ;JUSTER FCB-DREV
LD E,A ;VÆLG I BDOS
LD C,SLCTDSK ;SELECT DISK RUTINE
CALL BDOS
LD C,GDSKP ;FIND ADRESSE
CALL BDOS
LD BC,0002 ;INDEX TIL BLOCK SHIFT
ADD HL,BC
LD B,(HL) ;B=BYTE BLOCK SHIFT FAKTOR
INC HL
INC HL
INC HL
LD E,(HL) ;DE=ORD AF DISK BLOCK
INC HL ;TÆLLER
LD D,(HL)
INC DE
LD A,B ;JUSTER SHIFT FOR KB
SUB 3
SPCCAL: OR A ;ANTAL KB PR. BLOCK
JR Z,SPCKNW
ADD HL,HL ;DOBBELT ANTAL SECTORER
DEC A NEDSKRIV BLOCK SHIFT
JR SPCCAL
SPCKNW: LD C,L ;BC=KB PR. BLOCK
LD B,H
LD HL,0 ;NULSTIL KPDISK
LD (KPDISK),HL
PUSH BC ;GEM KB PR. BLOCK
PUSH DE ;GEM ANTAL BLOCK
LD C,GALVEC ;PEG TIL ALLOCATION VEKTOR
CALL BDOS ;HL=ADRESSEN
POP DE
POP BC

LD (ALLSAVE),HL ;GEM ADRESSEN
LD HL,1 ;MIN. START BITTÆLLER
UALLOC: DEC HL ;NEDSKRIV BITTÆLLER
JR NZ,STACT ;STADIG FLERE AKTIVE BIT
LD HL,(ALLSAVE) ;HENT POINTER
LD A,(HL)
INC HL ;GEM NY POINTER
LD (ALLSAVE),HL ;PÅ ADRESSEN
LD HL,08 ;MAX. BIT ANTAL
STACT: RLCA ;ALLOCATION BIT TIL CARRY
JR C,ALLOC ;UNDLAD AT TÆLLE OPTAGET
PUSH HL
LD (HL),KPDISK ;HENT KB TIL REST
LD HL,BC ;EN BLOCK MERE
LD (KPDISK),HL
POP HL
ALLOC: DEC DE ;NEDSKRIV TOTAL BLOKTÆLLER
LD L,A
LD A,D
OR E ;ALLE BLOCKS UNDERSØGT
LD A,L ;GENDAN BIT MØNSTER
JR NZ,UALLOC ;MERE SOM SKAL TÆLLES
LD A,(SAVDRIV) ;RETUR TIL TIDLIGERE DISK
LD E,A ;SÆT
LD C,SLCTDSK
CALL BDOS
RET
;DATAAREALER
BLKSIZ: DS 2
ALLSAV: DS 2
SAVDRIV: DS 1
KPDISK: DS 2
END

```



Function Number DEC: HEX	Function	Entry Value to BDOS Passed in (DE) or (E) regs	Return Value from BDOS Passed in (HL) or (A) register
0 00	System Reset	****	****
1 01	Console Input	****	(A) = character
2 02	Console Output	(E) = character	****
3 03	Reader Input	****	(A) = character
4 04	Punch Output	(E) = character	****
5 05	Printer Output	(E) = character	****
6 06	Direct Console I/O	(E) = 0FFH is input (E) = chr is output	(A) = character
7 07	Get IOBYTE	****	(A) = IOBYTE
8 08	Set IOBYTE	(E) = IOBYTE	****
9 09	Display Console String	(DE) = string addr	****
10 0A	Input Console String	(DE) = string addr	(A) = # chr input
11 0B	Get Console Status	****	(A) = 000H idle (A) = 0FFH ready
12 0C	Get CP/M Version Number	****	(HL) = Version #
13 0D	Reset Disk Subsystem	****	****
14 0E	Select Disk Drive	(E) = disk number	****
15 0F	Open a File	(DE) = FCB address	(A) = dir code
16 10	Close a File	(DE) = FCB address	(A) = dir code
17 11	Search for File	(DE) = FCB address	(A) = dir code
18 12	Search for Next	****	(A) = dir code
19 13	Delete File	(DE) = FCB address	(A) = dir code
20 14	Read next Record	(DE) = FCB address	(A) = error code
21 15	Write next Record	(DE) = FCB address	(A) = error code
22 16	Create New File	(DE) = FCB address	(A) = dir code
23 17	Rename File	(DE) = FCB address	(A) = dir code
24 18	Get Login Vector	****	(HL) = login vector
25 19	Get Logged Disk Number	****	(A) = logged disk
26 1A	Set R/W Data Buff Addr	(DE) = buffer addr	****
27 1B	Get Allocation Vector	****	(HL) = alloc vector address
28 1C	Write Protect Disk	(E) = disk number	****
29 1D	Get Read Only Vector	****	(HL) = R/O vector
30 1E	Set File Attributes	(DE) = FCB address	(A) = dir code
31 1F	Get Addr of Disk Parms	****	(HL) = parm addr
32 20	Get/Set User Select	(E) = 0FFH get	(A) = current user
33 21	Read Random Record	(DE) = long FCB adr	(A) = error code
34 22	Write Random Record	(DE) = long FCB adr	(A) = error code
35 23	Get Size of File	(DE) = long FCB adr	(r0-2 = rec cnt)
36 24	Set Random Record Num	(DE) = long FCB adr	(r0-2 = rec numb)
37 25	Reset Drive	(DE) = drive vector	****
38 26	Not used		
39 27	Not used		
40 28	Write Random with	(DE) = long FCB adr	(A) = error code



## \*\*\* BEMÆRK \*\*\*

På sidste bestyrelsesmøde blev vi enige om, at modem nu skulle klares, og det vil kunne ses i dette nummer. Det ligger lidt tungere med databasen, men et håb ligger jo i, at jo flere modemer medlemmerne, køber desto bedre bliver vor økonomi, og derefter håb om at få den op at stå. De medlemmer, der har et modem, vil indbyrdes kunne høste erfaringer, når de vil overføre programmer til hinanden.

Da en direkte annonce ikke virker, vil jeg nu ad denne vej opfordre det medlem fra Hellerup, der for snart 2 år siden kom med en strømforsyning, om at henvende sig. Jeg fik aldrig at vide, hvad jeg skulle med det, og jeg er på bar bund vedrørende adressen.

Asbjørn skal have en stor tak fra bestyrelsens side vedrørende bladet. Vi kan kun være glade for den måde, bladet ser ud på nu. Pudsemiddel til glorien vil senere komme.

Det kunne forresten være rart med lidt respons fra medlemmerne!!, men det er svært at vide noget, når vi aldrig hører fra jer. Jeg vil forresten bede de medlemmer, som kører på 2 meter, om at undgå negative udtalelser om bladet, når det ikke er saglig kritik!. Det er jo således, at mange ikke medlemmer også lytter, og da de ikke ved så meget om bladet, vil en positiv omtale eller ingen, være noget mere på sin plads. Personligt kommer jeg kun frem med de positive ting, og undgår at komme med priser og den slags ting, vi kan snakke om via telefon.

Angående undervisning står de ikke godt til. Der er ikke en eneste, der har meldt sig, og vi kan ikke tilbyde noget. Det står jo godt nok i vores love, men når ingen melder sig som lærer, er vi jo ikke i stand til at gøre noget.

Jeg vil sige tak til de få trofaste medlemmer, som leverer artikler, men samtidig bede jer andre om at støtte os lidt med små oplysninger. Det er dette som kan gøre bladet alsidigt, og værd at læse. Desuden har Asbjørn en større valgmulighed, når næste blad skal laves, når han kan vælge og vrage lidt.

Jeg vil forsøge mig med gennemgangen om CP/M instruktionerne, men et par positive eller negative læserbreve ville nu ikke være værst, idet dette vil give mig mere blod på tanden. Hvad er det i grunden I forventer jer af bladet? Vi må jo også tilgodese de nye medlemmer, og det er aldeles vanskeligt at gøre jer alle tilfredse. Der er en masse gamle maskiner i cirkulation, og da det ofte er begyndere, der får fat i disse anlæg, må vi også skrive noget til denne gruppe, men så er faren den, at de dygtige melder sig ud. Lad os høre fra begynderne og få at vide, hvad det er, de kunne tænke sig.

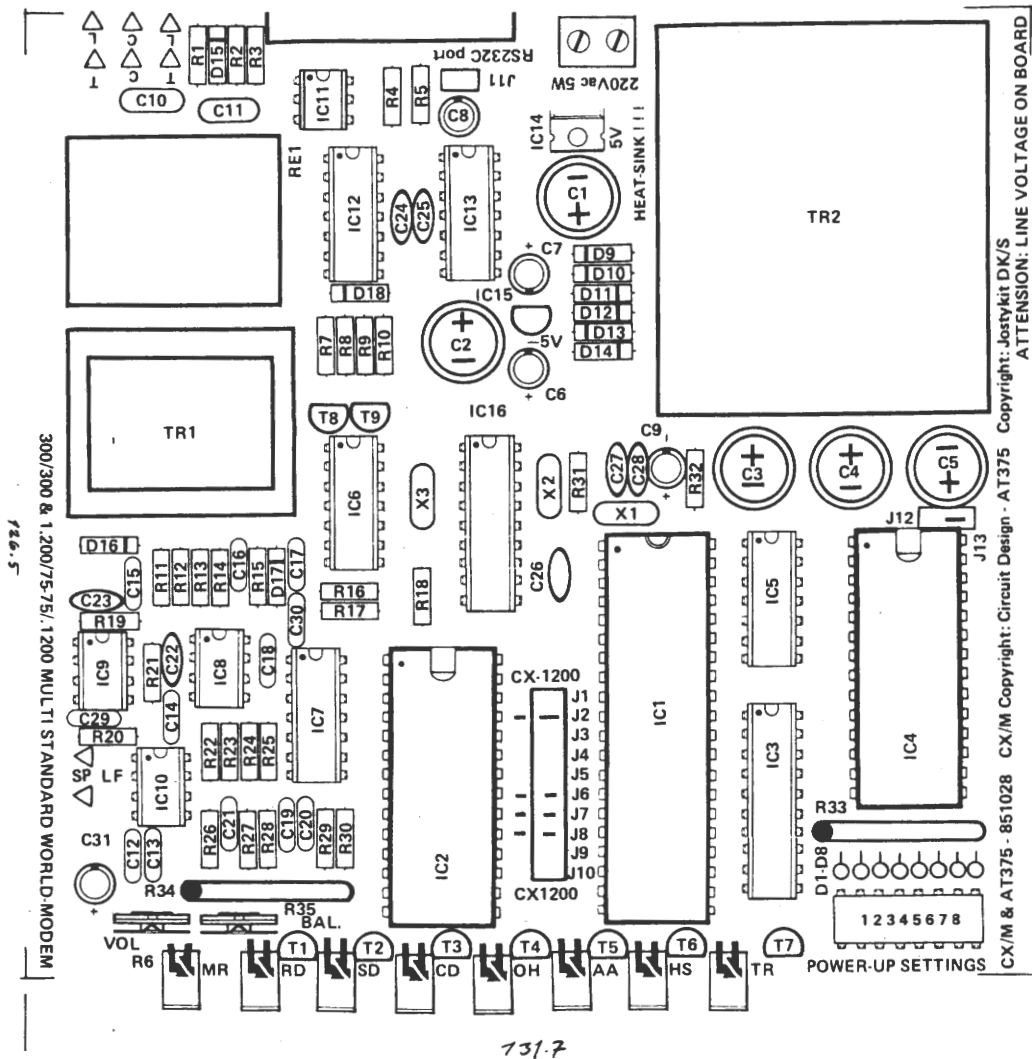
→

Med hensyn til udstilling, så har jeg henvendt mig til en af de andre grupper, og de var indstillet på at få noget igang, når der var valgt en ny bestyrelse, men der er ikke noget nyt om dette før senere på året.

Hvad har I i grunden tænkt jer, når flere og flere af os får fat på en 16 Bit maskine. Dette er også noget vi skal have frem.

Dette var Oles hjørne, som Asbjørn kalder det. Jeg håber på lidt mere kontakt fra jer alle, så vi kan få bladet endnu mere alsidigt.

O.H.



\*\*\* STAT \*\*\*

Så er vi nået til STAT, som er et ganske nyttigt program. Det svarer lidt til SET og SHOW i CP/M 3. Jeg omtaler ikke brugen af RDR PUN TTY BAT o.s.v, men viser bare, hvordan man kommer til de udskrifter, der kommer på skærmen, når man bruger STAT. Lad os først se på et normalt bibliotek, som ses, når vi har skrevet DIR på skærmen.

```
B: WS      COM : WSMMSG  OVR : WSOVLUI  OVR : WSINS  COM
B: STAT    COM : PIP     COM : DUMP     COM : DEBUG  COM
B: BIOS    ASM : MUP     ASM : SYSGEN  SYS : MOVE   PAS
B: BIN     PAS : OHELLO  PAS : OHELLO  COM : MODSTANDBAS
```

Nu er der flere muligheder, hvis vi vil have beskyttet vore programmer mod overskrivning. Det vil jo ikke være så slemt, hvis f.eks OTHELLO.COM blev slettet. Den kunne jo gendannes fra PAS filen.

Derimod ville det være en skam, hvis WS filerne forsvandt, så lad os skrivebeskytte dem sammen. Man kan da nøjes med at skrive:

```
STAT WS*.* $R/O      ( SET WS*.* ÆROÅ )
Hvis PAS filerne skulle skrivebeskyttes taster du:
STAT *.PAS $R/O      ( SET *.PAS ÆROÅ )
```

Disse få eksempler skulle nok give læseren en ide om, hvordan dette virker.

Hvis en bestemt fil altid skal kunne ændres, skriver man bare: STAT NAVN.EXT \$R/W Navnet er filens navn og Ext er de 3 sidste TEGN efter navn.

Du kan beskytte alt på en gang, ved at taste:

```
STAT *.* $R/O      ( SET *.*ÆROÅ )
STAT kan også give andre oplysninger: Prøv at taste:
STAT DSK:          ( SHOW Æ DRIVE Å )
STAT VAL:          ( DEVICE )
STAT USR:          ( SHOW ÆUSERSÅ )
```

STAT kan også sætte dig i forbindelse med den serielle indgang og udgang, og dette gælder også for skærmen. Det du ser skrevet på skærmen, kan samtidig udskrives til en printer eller til en eller anden enhed, der ligger uden for din computers nærhed. Du kan altså nøjes med at bruge din computer som en simpel terminal, hvis det skulle blive nødvendigt.

Dette gælder forresten også for den parallelle del af computeren. Det kan blive aktuelt at kalde disse enheder fra et assembler-, basic- eller pascalprogram, men her går du uden om STAT og kalder direkte.

Jeg vil ellers ikke komme nærmere ind på I/O enhederne her, idet det vil blive ret omfattende, men ganske kort fortælle, at hvis du vil have et program med kildetekst over fra en anden computer skriver man: PIP NAVN.EXT=RDR:ÆVÅ < cr>

Se så er vi allerede ovre i PIP og dette bliver senere omtalt, men det viser jo bare, at der er en god grund til at gennemgå disse ordre. De ordre, der står i parentes, gælder for CP/M 3, og du skal naturligvis udelade parenteserne. Æ er klammestart, mens Å er klammeslut.



Dato: 8/11/85

Til: Z80 NYT v. Asbjørn Lind

Fra: Medlem nr. 407, Hans-Henrik Jensen

Ang: Artikel om datatransmission i Z80 NYT nr. 6, september 85.

Jeg har nogen bemærkninger til din ellers udmærkede artikel. Først gælder det din definition af "høj", "lav", "0" og "1".

Logisk "0" er repræsenteret ved positiv spænding mellem +3 V og +25 V. Denne tilstand kaldes også for "space".

Logisk "1" er repræsenteret ved negativ spænding mellem -3 V og -25 V. Denne tilstand kaldes også for "mark".

Området mellem +3 V og -3 V er undefineret. Værdierne +/- 25 V er uden belastning. Med belastning er de +/- 15 V. Alle disse niveauer er fastsat af både CCITT og EIA.

Jeg har også nogen bemærkninger til dine definitioner af halv og fuld duplex.

Fuld duplex (FDX) betyder at man kan transmittere i begge retninger samtidig.

Halv duplex (HDX) betyder at man kan transmittere i begge retninger, men kun i een retning ad gangen.

Det begreb som du beskriver som FDX er "ekkoplexing".

Ekkoplexing er et funktionsprincip for asynkrone FDX-terminaler, og ikke forskellen mellem FDX og HDX. Til dit forsvar kan jeg nævne at jeg har set terminaler, som man ikke kunne konfigurere som FDX eller HDX, men man kunne sætte "echo on" eller "echo off". Disse terminaler er uberørte af FDX / HDX. Hvis man på en sådan sætter "echo on", og man har en FDX-linje, vil man få dobbelte karakterer på skærmen.

Jeg er ellers enig med dig i dine synspunkter om forvirringerne som "følger" RS232C. Jeg kunne sagtens give 3-4 "standard-sammenkoblinger" i tillæg til dine 4.

Jeg vil gerne at disse rettelser bliver offentliggjort i Z80 NYT, da jeg ikke ønsker at brugergruppens medlemmer bliver lige så forvirrede som så mange andre, når det gælder datatransmission.

Venlig hilsen



Hans-Henrik Jensen

Referencer: "EDB DATATRANSMISSION", Henning Haugård  
WIND Databøger, 1983

"HANDBOOK OF DATA COMMUNICATIONS", The National  
Computing Centre, NCC Publications, 1982

+ diverse publikationer fra Televerket i Norge og  
Sverige.



Indlæg til Z80 nyt

I Z80 nyt 6 aargang nr 8 laeser jeg til min store forbavselse paa side 13, et indlaeg af Asbjoern hvori han spoerger om ikke skal forny os ogomtale andre processorer end Z80 typerne.

Hvad taenker han dog paa, glem ikke at vi alle har meldt os ind klubben fordi vi bruger Z80, og at der stadig kommer nye medlemmer til.

Selvom det ligger i tiden, at der kommer mange nye og "FANTASTISKE" maskiner med baade 16 og 32 bit maa vi ikke glemme at meget af dette kun er glimmer og legetoej. Bladets navn er stadig "Z80 nyt" og lad det blive ved det.

Stop al den snak om at inkludere andre maskintyper, og overlad det til de hundrede hobby blade der florerer, og lad os blive ved med det serioese arbejde, (Hvis jeg havde en anden maskine, f.eks. en IBM PC ville jeg da abbonere paa PC NYT og ikke paa Z80 nyt).

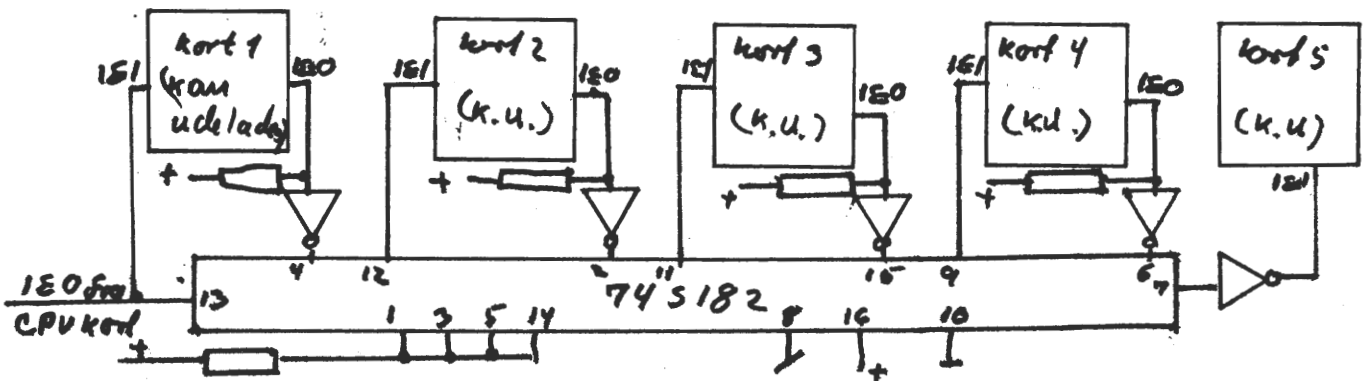
Hvis ogsaa dette blad skal behandle andre maskiner, ja saa ved vi jo godt hvordan det vil gaa, og om kort tid er der ingen plads i Danmark for os Z80 folk.

I oevrigt er det meste jo alligevel CP/M og man glemmer alle de der stadig koerer med NASCOM systemets monitor.

--- Nej lad os holde os til bladets sande ordlyd-- Z80 NYT --

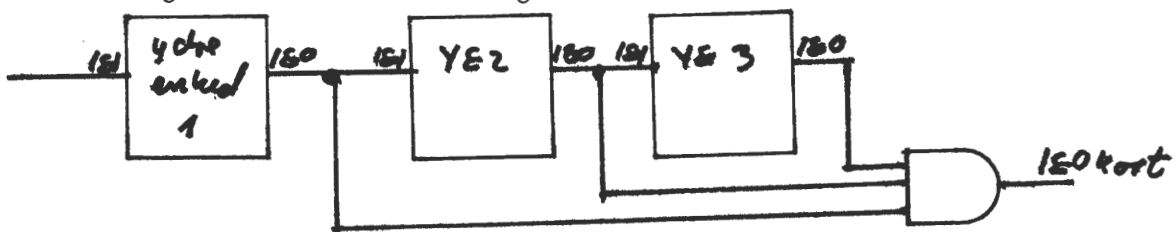
Venlig hilsen no. 338 T. Bundgaard.★

Jeg faldt forleden over en bemærkning i et Z-80 Nyt (glemte at notere nummeret) om interrupt, at man skulle huske at have kort i alle stik hen til CPU-kortet for at sikre daisy-chain-linien. Det er altsaa ikke nødvendigt, hvis man indretter motherboard fornuftigt. En let modificeret Zilog-konstruktion klarer sagen:



Jeg er specielt glad for denne konstruktion fordi jeg på bagsiden af motherboard har anbragt et ekstra stik, så jeg kan sætte kort i til gennemmåling, sådan at de sidder under samme betingelse, som når de bliver sat rigtigt på plads.

Med den viste konstruktion kan man også glemme alt om, hvor mange ydre enheder (SIO, PIO, CTC etc.), der er på de foregående kort, især hvis man på det enkelte kort med mere end to ydre enheder bruger en enkelt AND-gate således:



Her er vist tre ydre enheder, men jeg har et kort med fire, og det kører udmærket.



Har du engang prøvet at køre med memorydisk er du fortabt, hvis du ikke har prøvet, kan du købe 256 Kbyte af mig incl. software til styring af samme. Map80 256 Kb-kort A. Lind 02 91 71 82 2500,-kr.

Nascom 2 incl NASSYS 3, div pgr. (evt. PolyDos eller CP/M 2.2) og backplane. Tastatur i lækker metalkasse 2000 kr.  
 1 stk. G802 RAM64 kort incl page option - 64 Kb 1000 kr.  
 1 stk. G803 EPROM/ROM kort incl. BLS Pascal v. 1.3, NIP/NAP, Poly-Sys 4.3 og div. småprogrammer alt i 2716 EPROM's 800 kr.  
 1stk. TEAC FD50A SS 48 TPI floppy, incl MK-kasse og manual 800 kr.  
 2 stk. SHUGART SA410 SS 96 TPI floppy, incl MK-kasse ialt 2000 kr.  
 (372) Michael Brouer 02 27 50 31

Gemini floppy-system: 2 stk. Pertec FD250, strømforsyning, kontrollerkort til singel-density. 1800,00 kr.

MiroGDC grafikort: NEC 7220, 128 Kb, programmerbart skærmformat f.ex. 512\*512, 16 farver (gråtoner) fra paletter med 4096 nyancer, diverse software. Kan udbygges til 1024\*1024\*4 (512 Kbyte) 6000 kr.  
 Henvendelse til Lars Loldrup på telefon nr. 02 26 76 97

2 stk. 64 Kb RAM-kort, fuld monteret (incl .page mode option) sælges for 600 kr./stk. (531) Kim Chrsitensen 02 54 42 31

MODEM IC Am7911 digital modem kreds - klarer alle standarder, også teledata. Bruges i mange konstruktioner incl datablad 380,00 kr.

NEC PinWriter P2 med centronics interface, 180 tegn/sek. grafik og skønskrift. Næsten ubrugt (fabriksgaranti) incl manual 8900,00 kr.

CP/M Plus Computer: Z80B, 6 MHz, 128 Kb, grafik 640\*320 eller 24-40 linjer af 80 tegn, FDC, printerport, 2\*rs232, keyboardport, timer Switch mode PSU, 2\*360 Kb floppy. Stik. 5 disketter med system og utilities. Alt fabriksnyt (uåbnet). Oprydningssalg 9900,00 kr.  
 Medl nr. 600 Mads Aarup 02 35 80 16

Nyt Diskkopieringsprogram fra MAP80. Kan sættes op til over 100 formater - også PCDOS.

Oprydningssalg (inc. moms)

Software:

GemPen (Tekstbehandlingsanlæg, CP/M).....	350,- kr.	DATARAMA 01 86 12 17
SVC Grafpac (Links Mbasic) .....	650,- kr.	
GemZap (assembler CP/M) .....	400,- kr.	
GraphPac (Nascom Video) .....	250,- kr.	

Hardware:

AVC-Kort .....	1900,- kr.	BERNHARD BANGS ALLE 17
MIMI CP/M Computer .....	7000,- kr.	
Nascom 3 kasse .....	600,- kr.	2000 F
Gostgold 16 bit CPU-kort .....	3800,- kr.	
ROM Kort .....	600,- kr.	11.00 + 17.30
STAR printer 515 .....	3500,- kr.	
Black Box Typeprinter .....	300,- kr.	
Keyboard (AST I) serie .....	800,- kr.	

Et medlem ønsker at nogen skriver om følgende emner:  
 HPiB / GPIB / IEC 625 / IEEE 488 Hardware og software



**SÆLGES:**

Gemini G 812 IVC 'Intelligent Video Card' med SMC 812 Graphics Expander, incl. software til dette skrevet af Anders Hejlsberg, samt noget demo software til Compas.  
Usersmanual diagrammer m.m følger selvfølgelig med.

Pris.. 2500,-

Rene' Hansen. 02 177705.



**SÆLGES:**

Foreningens STAR DP 515 printer sælges i original emballage og incl usersmanual.

Printeren har lige fået nyt matrixhoved og virker dermed perfekt, printeren er af den brede type (15"), og har både gummivalse og traktor.

Pris.. ca. 2500,-

Ring til Rene' Telefon. 02 177705.




---

**Indkøbsforeningens Vare Salg**

Priser excl. moms.  
pr. 1.11.1985

**EPSON**

*Den professionelle printer.*

EPSON er på meget kort tid blevet leverandør nummer ET, af højkvalitets printere til 'personal computer' markedet. Vi vil derfor tilbyde medlemmerne disse printere til følgende meget lave priser.

**Matrix printere.**

EPSON RX-80.	100 cps, 80 cpl og traktor..	Kr. 4020.-
EPSON RX-80 F/T	100 cps, 80 cpl, traktor og friktion..	Kr. 4420.-
EPSON RX-100	100 cps, 136 cpl, traktor og friktion..	Kr. 7640.-
EPSON FX-80	160 cps, 80 cpl, pin feed og friktion..	Kr. 7240.-
EPSON FX-100	160 cps, 136 cpl, traktor og friktion..	Kr. 9652.-

*Tilbehør.*

Farvebånd i kasette til MX-80, RX-80 og FX-80 (1 stk.)..	Kr. 95.-
Farvebånd i kasette til MX-100, RX-100 og FX-100 (1 stk.)..	Kr. 175.-
Traktor til FX-80..	Kr. 655.-
Technical Manual til MX-80, RX-80 og FX-80..	Kr. 350.-

Varer bestilles hos forretningsføreren. 02 177705.



### Skønskrift printere.

EPSON DX-100 P	Standard parallel version..	Kr. 7905.-
EPSON DX-100 S	Standard seriel RS232C..	Kr. 7905.-
EPSON LQ-1500	incl. 1 interface efter eget valg..	Kr. 17585.-

#### Tilbehør.

Farvebånd i kasette til DX-100 (1 stk.)..	Kr. 65.-
Farvebånd i kasette til LQ-1500 (1 stk.)..	Kr. 125.-
Traktor til DX-100..	Kr. 1950.-
Single sheet feeder til DX-100..	Kr. 4490.-
Keyboard til DX-100..	Kr. 3450.-
Traktor til LQ-1500..	Kr. 990.-
Single sheet feeder til LQ-1500..	Kr. 6650.-
Double sheet feeder til LQ-1500..	Kr. 10990.-

## 3M DISKETTEN

### Den professionelle diskette.

3M giver *livsvarig garanti* på alle deres disketter. En defekt diskette kan returneres til 3M, som tester disketten på deres laboratorium og vurderer, om det er håndteringsfejl eller produktionsfejl.

! 3M Disketter. ! Medlem !			
! Type	! 8" !	! 5.25" !	! Pris. !
!740-0	! *	!	! 27.61 !
!740/2-0	! *	!	! 34.29 !
!741-0	! *	!	! 34.29 !
!741/2-0	! *	!	! 40.48 !
!743-0	! *	!	! 40.48 !
!744D-0	!	! *	! 25.56 !
!745-0	!	! *	! 32.20 !
!747-0	!	! *	! 36.10 !
!96tpi HD!	!	! *	! 42.00 !
!3.5" 135 tpi SS DD	!	!	! 46.00 !

! 3M Rensdisketter. ! Medlem !			
! Type	! 8" !	! 5.25" !	! Pris. !
!7400	! *	!	! 127.50 !
!7440	!	! *	! 127.50 !

! Scotch Dataarkiv system uden lås ! Medlem !			
! Type	! Antal !	! 8" !	! 5.25" !
! Arkivbox til	! 80 !	!	! * !
! Arkivbox til	! 80 !	! *	!
! Arkivbox til	! 10 !	!	! * !
! Arkivbox til	! 10 !	! *	!
! Ringbind til	! 20 !	!	! * !

! ABS Dataarkiv med lås og løst låg ! Medlem !			
! Type	! Antal !	! 8" !	! 5.25" !
! Arkivbox til	! 40 !	!	! * !
! Arkivbox til	! 80 !	!	! * !
! Arkivbox til	! 40 !	! *	!
! Arkivbox til	! 80 !	! *	!

Minimum bestilling af disketter er 10 stk.  
Portoen pr. forsendelse udgør Kr. 20.-

## TEAC

### Floppy disk drive

TEAC FD 55 B....	40 spor. dobbelt side. .5 Mb. ufm..5.25"...	Kr. 1661.00
TEAC FD 55 F....	80 spor. dobbelt side. 1 Mb. ufm..5.25"....	Kr. 1900.00
TEAC FD 55 GF...	80 spor. dobbelt side. 1.6 Mb. ufm.5.25".	Kr. 2414.50
TEAC FD 35 F....	80 spor. dobbelt side. 1 Mb. ufm. 3.5".....	Kr. 1900.00
TEAC FD 135 F...	som FD 35 F men kun 25 m.m. højt. (2/3)....	Kr. 2000.00