



ARGENTINA - 1987

- 3
- 4
- 5
- 11
- 11
- 12
- 14
- 14
- 15
- 19
- 20
- 20
- 22
- 24

Number of items

HELP

UDGIVES AF Z80-BRUGERGRUPPEN, Nørre Farimagsgade 1, 1364 Kbh. K.

Bestyrelsen:

Formand	Kurt Neierdi,	Nørre Farimagsgade 1, 1364 Kbh. K.
Tlf.tid	(mandag + torsdag)	01 11 38 94 (1700 - 1800)
Næstform.	Frank Damgaard,	Kastebjergvej 26A 2750 Ballerup
Kasserer	- -	02 97 37 47 (1730 - 1830)
Bestyr.m.	Carsten Busk Senholt,	Blommevangen 6/1 2765 Smørum
-	Anders Otte,	Frederiksdalsvej 40/2/th 2830 Virum
-	Henrik Jensen,	Markskellet 6/ST/TV 2720 Vanløse

CP/M Bibliotek: Asbjørn Lind, Sidevolden 23 2730 Herlev
(skriftligt)

Forretn.f: pt. Frank Damgaard, Indkøbsforeningen.

Redaktion: Z80-NYT, Fensmarks Alle 6 3520 Farum

Logo: Carsten Busk Senholt.

Redaktør: pt. Viggo Jørgensen.

Private annoncer gratis for medlemmer.

Oplag: > 500.

Kontingent for 1987: 300 kr. Indmeld.gebyr: 25 kr.

GIRO 6 74 26 02, Z80-Brugergruppen, c/o Frank Damgaard, (se adr. ovenfor).

MODEM MAIL-BOX 01 91 38 98

forventes klar til prøvekørsel ultimo juni

flere oplysninger i næste nr. af Z80-Nyt.

- = o = -



Fornylig afholdtes foreningens ordinære generalforsamling og selvom ikke alle foreningens medlemmer var mødt frem så lykkedes det alligevel at behandle et bredt udsnit af relevante emner / problemer. På grund af et stort arbejdspress har Lennart Søby ikke set sig i stand til at fortsætte på posten som formand og forsamlingen har nu valgt Kurt Neierdi som sin nye formand. K.N. har foruden en teknisk baggrund, også den 'nødvendige' computer-interesse.

Til bestyrelsen blev genvalgt Frank Damgaard (næstformand), Anders Otte og Carsten Senholt, nyvalgt blev Henrik K. Jensen og som suppleant lovede Jørgen Hjelme at træde til, hvis behov skulle opstå. John Jacobsen ønskede ikke at genopstille denne gang.

Indkøbsforeningens generalforsamling kunne, bl.a. på grund af tidnød desværre ikke gøres færdig, der vil derfor på et senere tidspunkt blive afholdt ekstraordinær generalforsamling for indkøbsforeningen. Thomas Corell har givet tilsagn om assistance.

- - -

I dette nummer af Z80-NYT berøres et antal emner, som formodes at have interesse for en del af foreningens medlemmer, ikke alt er lige dybsindigt, men adskilligt har dog 'fået en tanke' ekstra.

En ny-nyhed er ZILOG's Z280, eller som man nok bedst kender den: Z800, Zilog benævner selv Z280 MPU'en som "vel dette årti's mest omtalte produkt"!; den er så ny, at selv om den pt. kan leveres her i landet, har det været vanskeligt at skaffe tilstrækkeligt materiale, egnet til gengivelse for en teknisk kreds, som her i bladet.

Resultatet er blevet en foreløbig 'sammenstykning' fra mange kilder, og der skal nok komme mere, men læs selv nærmere om denne interessante IC, og hvis nogen af læserne har viden om dette eller andre emner, som kan være af interesse, (f.eks. har der været en del spørgsmål ang. Modula-2, Pascals efterfølger?) så er redaktionen lutter øre, ideer eller spørgsmål er også meget velkomne.

- god ferie! VJ

For nogle år siden opdagede jeg en fejl i JRT-PASCAL, der resulterede i, at sammenligninger af mængder (SET) gav et forkert resultat. Det tog en vis tid, inden det gik op for mig, at fejlen måtte ligge i PASCAL-systemet, og ikke i mit program. Når først fejlen er indkredset, kan man programmere sig uden om problemet (kan nogle gange være svært).

For nogen tid siden læste jeg så en kort artikel i tidsskriftet MC (nr. 6 1984) hvori det førnævnte problem også blev omtalt. Yderligere var der foreslået en rettelser til EXEC.COM, så denne fejl skulle forsvinde. Problemet er, at den løkke, der skal sammenligne de to sæt af 16 byte, "glemmer" at forøge HL og DE med 1, således at det er de samme 2 byte, der sammenlignes op til 16 gange. Der er så heldigvis mulighed for at indføje en rettelser, som ikke fylder mere end den oprindelige løkke, idet det udnyttes, at register C ved adresse 2E2F har værdien OF0H. Rettelsen er som følger:

Kald DDT med "DDT EXEC.COM". Husk at udregne antallet af sider (xx), som programmet fylder, idet værdien skal bruges senere. Benyt nu f.eks. kommandoen "L2E2F,2E3A".

Adresserne fra 2E2F skulle så gerne indeholde:

	<u>8080</u>		<u>MC</u>		<u>Z80</u>
	2E2F	MVI C,10	OE 10	LD	C,10h
	2E31	LDAX D	1A	LD	A,(DE)
	2E32	CMP M	BE	CP	(HL)
	2E33	JNZ 2E50	C2 50 2E	JP	NZ,2E50
	2E36	DCR C	OD	DEC	C
	2E37	JNZ 2E31	C2 31 2E	JP	NZ,2E31
Med "A2E2F" rettes					
ovenstående					
til:	2E2F	LDAX D	1A	LD	A,(DE)
	2E30	CMP M	BE	CP	(HL)
	2E31	JNZ 2E50	C2 50 2E	JP	NZ,2E50
	2E34	INX D	13	INC	DE
	2E35	INX H	23	INC	HL
	2E36	INR C	0C	INC	C
	2E37	JNZ 2E2F	C2 2F 2E	JP	NZ,2E2F
	2E3A				

Herefter kan returneres til CCP med "CTRL C", hvorefter man med kommandoen "SAVE <xx> EXEC.COM" gemmer rettelserne i filen. (Husk, gem ikke rettelserne på originaldisketten, men benyt en kopi!) Ovenstående rettelser ser ud til at virke, og jeg har endnu ikke kunnet finde nogle eventuelle bivirkninger. Vedlagte testprogram kan benyttes til at undersøge fejlen. Programmet skal udskrive værdierne TRUE,FALSE,FALSE i angivne rækkefølge.

```

program test_set;
type set_type=set of 0..30;
var a,b: set_type;
    c:boolean;

begin
  a:=[0,4,8,13,19,27]; b:=[0,4,8,13,19,27]; c:= a=b;
  writeln(' a = b :',c);
  b:=[0,4,8,13,28]; c:= a=b; writeln(' a = b :',c);
  b:=[0,8,13,19,27]; c:= a=b; writeln(' a = b :',c);
end.

```

Frank Damgaard.

dBASEII Telefonregister Vers. 2

På utallige opfordringer skal jeg her bringe anden version af mit telefon register program til dBASE II. Foruden at have forbedret kontrolstrukturen i programmet, har jeg tilføjet to nye felter til databasefilen: SKJUL og KOMMENTAR. Formålet med denne ændring var at kunne have mulighed for at skrive kommentarer til hver enkelt person i telefonregistret såsom alternative telefonnumre, åbningstider og så videre. Ved at sætte feltet SKJUL til TRUE eller FALSE har brugeren mulighed for at få udskrevet den omtalte kommentar. Derfor er der kommet to nye kommandofiler til:

OPDATER.CMD og FINDDATA.CMD

Ønsker man at slette en person fra registret, kan dette gøres med kommandoen

. DELETE RECORD n

Denne record er nu slettemarkeret og vil ikke blive udskrevet ved brug af START.CMD. Den (og eventuelt andre !) kan fjernes helt med ordren

. PACK eller gøres aktive igen med . RECALL

Brugsforvirring til programmet :

Man indtaster de fire kommandofiler: START.CMD, UDSKRIV.CMD, FINDDATA.CMD samt OPDATER.CMD nøjagtigt. Dernæst oprettes databasefilen TELEFON.DBF som har nedenstående struktur. Dette gøres med ordren

. CREATE Nu kan programmet så køres med ordren DBASE START eller

. DO START

Har man allerede oprettet en TELEFON.DBF efter den gamle opskrift kan man hente de gamle data over i den nye database ved hjælp af ordren

. USE A:TELEFON
. APPEND FROM B:TELEFON

Det tilrådes at starte med at indtaste START.CMD, og derefter rette en kopi af denne til, så man kommer frem til FINDDATA.CMD. De to nævnte filer er som det vil ses næsten ens.

. list struc ~~~~~ GOD FORNØJELSE !!! ~~~~~

STRUCTURE FOR FILE: B:TELEFON .DBF
NUMBER OF RECORDS: 00231
DATE OF LAST UPDATE: 00/00/00
PRIMARY USE DATABASE

John B. Jacobsen

FLD	NAME	TYPE	WIDTH	DEC
001	FORNAVN	C	015	
002	EFTERNAVN	C	020	
003	GADE	C	030	
004	POSTNUMMER	C	004	
005	POSTHUS	C	015	
006	TLFNUMMER	C	008	
007	SKJUL	L	001	
008	KOMMENTAR	C	050	
** TOTAL **			00144	

```
* START PÅ FILEN <START.COMD>
* PROGRAM TIL AT SLÅ OP I EN TELEFON REGISTER DATABASE
SET INTENSITY ON
USE TELEFON
ERASE
SET TALK OFF
SET DELETED ON
SET COLON OFF
SET EXACT OFF
STORE ' ' TO KEY
DO WHILE T
STORE 1 TO GALT
@ 0,0 SAY ' ' @
TEXT
```

Hvilket felt vil Du søge på ?

1. Fornavn
2. Efternavn
3. Gade
4. Postnummer
5. Posthus
6. Telefonnummer

Indtast det ønskede feltnummer !

eller

- 0 Opdatering af databasen
X,Q Exit eller Quit

```
ENDTEXT
@ 16,55 GET KEY
READ
DO CASE
```

```
CASE KEY='1'
STORE 'fornavn' TO FELT
```

```
CASE KEY='2'
STORE 'efternavn' TO FELT
```

```
CASE KEY='3'
STORE 'gade' TO FELT
```

```
CASE KEY='4'
STORE 'postnummer' TO FELT
```

```
CASE KEY='5'
STORE 'posthus' TO FELT
```

```
CASE KEY='6'
STORE 'telefonnummer' TO FELT
```

```
CASE !(KEY)='X'  
SET COLON ON  
SET TALK ON  
SET EXACT ON  
SET DELETED OFF  
@ 23,2 SAY ' '  
RETURN
```

```
CASE !(KEY)='O'  
DO OPDATER
```

```
CASE !(KEY)='Q'  
QUIT
```

```
OTHERWISE  
@ 18,65 SAY 'Galt valg !'  
STORE 0 TO GALT
```

```
ENDCASE
```

```
IF GALT=1  
ERASE  
@ 5,5 SAY 'Indtast '+FELT+' herunder'  
?  
ACCEPT ' Søgedata ' TO NOGLE  
STORE !(FELT) TO FELDT  
@ 23,16 SAY 'Det tager lige et øjeblik !'  
LOCATE FOR !(&FELDT)=!(NOGLE)  
DO WHILE .NOT. EOF  
DO UDSKRIV  
CONTINUE  
ENDDO  
ERASE
```

```
ENDIF
```

```
STORE ' ' TO KEY
```

```
ENDDO  
* SLUT PÅ FILEN <START.CMD>
```

```
* START PÅ FILEN <UDSKRIV.CMD>
```

```
ERASE
```

```
?
```

```
?
```

```
?
```

```
? ' Navn : '+TRIM(FORNAVN)+' '+EFTERNAVN
```

```
? ' Adresse : '+GADE
```

```
? ' By : '+POSTNUMMER+' '+POSTHUS
```

```
?
```

```
STORE TLFNUMMER TO T
```

```
? ' Telefon : '+$(T,1,2)+'-'+$(T,3,2)+' '+$(T,5,2)+' '+$(T,7,2)
```

```
?
```

```
IF .NOT. SKJUL
```

```
? ' Kommentar : '+KOMMENTAR
```

```
ENDIF
```

```
@ 22,20 SAY 'Tryk på en knap når Du har læst data !'
```

```
WAIT
```

```
RETURN
```

```
* SLUT PÅ FILEN <UDSKRIV.CMD>
```

```
* START PÅ FILEN <OPDATER.CMD>
SET INTENSITY ON
USE TELEFON
ERASE
SET TALK OFF
SET DELETED ON
SET COLON OFF
SET EXACT OFF
STORE ' ' TO VALG
STORE 0 TO GALT
STORE 0 TO N
DO WHILE T
@ 0,0 SAY ' '
TEXT
```

Hvad ønsker Du ?

1. Finde og rette data i telefonregistret
2. Indsæt nye data i telefonregistret

Tast et af cifrene 1,2 eller

X,Q Exit eller Quit

```
ENDTEXT
```

```
@ 10,17 GET VALG
READ
DO CASE
```

```
  CASE VALG='1'
  DO FINDDATA
  IF N<>0
    EDIT N
  STORE 0 TO N
  ENDDIF
  ERASE
```

```
  CASE VALG='2'
  SET COLON ON
  APPEND
  SET COLON OFF
  ERASE
```

```
  CASE !(VALG)='X'
  ERASE
  @ 12,20 SAY 'Du valgte at returnere !'
  RETURN
```

```
  CASE !(VALG)='Q'
  ERASE
  @ 12,20 SAY 'Du valgte at returnere til CP/M niveau !'
  QUIT
```

```
  OTHERWISE
```

```
  @ 20,27 SAY 'Du indtastede et ugyldigt valg !'
  @ 22,27 SAY 'Mulige valg er cifrene 1 til 2 !'
```


ENDCASE

STORE ' ' TO VALG

ENDDO

* SLUT PÅ FILEN <OPDATER.CMD>

* START PÅ FILEN <FINDDATA.CMD>

SET INTENSITY ON

USE TELEFON

ERASE

SET TALK OFF

SET DELETED ON

SET COLON OFF

SET EXACT OFF

STORE ' ' TO KEY

DO WHILE T

STORE 1 TO GALT2

@ 0,0 SAY ' '

TEXT

Hvilket felt vil Du søge på ?

1. Fornavn
2. Efternavn
3. Gade
4. Postnummer
5. Posthus
6. Telefonnummer

Indtast det ønskede feltnummer !

eller

X,Q Exit eller Quit

ENDTEXT

@ 16,55 GET KEY

READ

DO CASE

CASE KEY='1'

STORE 'fornavn' TO FELT

CASE KEY='2'

STORE 'efternavn' TO FELT

CASE KEY='3'

STORE 'gade' TO FELT

CASE KEY='4'

STORE 'postnummer' TO FELT

CASE KEY='5'

STORE 'posthus' TO FELT

```

CASE KEY='6'
STORE 'tlfnummer' TO FELT

CASE !(KEY)='X'
ERASE
STORE ' ' TO KEY
@ 12,20 SAY 'Du valgte at returnere !'
RETURN

CASE !(KEY)='Q'
QUIT

OTHERWISE
@ 18,65 SAY 'Galt valg !'
STORE 0 TO GALT2

ENDCASE

IF GALT2=1
ERASE
@ 5,5 SAY 'Indtast '+FELT+' herunder'
?
ACCEPT '      Søgedata ' TO NOGLE
STORE !(FELT) TO FELDT
@ 23,16 SAY 'Det tager lige et øjeblik !'
LOCATE FOR !(&FELDT)=!(NOGLE)
DO WHILE .NOT. EOF
DO UDSKRIV
STORE ' ' TO KEY
? 'Er det viste de data Du er interesseret i ? (J/N) '
@ 24,50 GET KEY
READ
IF !(KEY)='J'
STORE # TO N
RETURN
ENDIF
CONTINUE
ENDDO
ERASE
ENDIF

STORE ' ' TO KEY

ENDDO
* SLUT PÅ FILEN <FINDDATA.CMD>

```

////////////////////////////////////// MODEM //

Formanden meddeler: at MODEM-restance-puklen nu (næsten) er afviklet !!!

----- INDKØBSFORENINGEN -----

Udsalg / Tilbud ...chancen er her nu ! Priser incl. moms. + porto.

40 stk. 8" 743 3M disketter DSDD a 30 kr.

50 - 5.25" 747 3M - DSDD a 25 -

30 - - HD 5314 3M - a 45 -

10 - 3.5" 3M - SSDD a 37 -

12 - CP/M mappe a 100 -

3 - "Maskinkodeprogrammering
med Z80" (bog) a 120 -

1 - "Systime PolyPascal" (bog) a 140 -

15 - JRT-Pascal (manual I+II) a 75 -

3 - 3M Head Cleaning Set a 95 -

2 - diskette-box 80 stk. 5.25" a 100 -

3 stk. 5.2 Megabyte harddisk SHUGART a 1100 -

Henv. Frank Damgaard.

Til Hard-diskene kan der evt. leveres 1 stk. Xebec-Controller,
(for 2 Hard-diske) til SASI-interface. Prisen kan der tales om.

Henv. Kurt Neierdi.

////////////////////////////////////// NOTER //

ZILOG: i DK DITZ SCHWEITZER A/S. Vallensbækvej 41, 2600 Glostrup (02 45 30 44).

RADIO-PARTS A/S er kommet med et nyt katalog / håndbog nr. 8, > 1000 sider,
pris 200 kr. + moms. (01 33 33 11).

'GE/RCA' (General Electric / RCA) & Intersil repræsenteres (1 Januar 1987) af
Tage Olsen A/S. (02 65 81 11). (RCA overtog Intersil, GE overtog RCA)

AARHUS RADIOLAGER A/S formidler komponenter fra det engelske RS Components Ltd.
og udsender i den anledning dette firmas katalog til sine kunder; > 800 sider,
pris pt. ukendt spørg selv. (06 12 62 44).

COBOL "Programmering i struktureret COBOL" af Inge Sørensen. ISBN 87-88799-01-8
Lærebog på dansk for begyndere. Forlaget NB i Odense har nu udsendt 3. udgave,
1. oplag 1985. Pris 198 kr. Bog for efter-sommerferien-kursister + mange andre.
Bogen kan bestilles hos: Forlaget NB / Klingstrupvænget 97 / 5230 Odense M.

Er dette mon 'det' nye kommunikationsmiddel? uden formatvanskeligheder? ja, det mener nogen. Hvis man blot køber The Softstrip Reader (ca. 200 £ + VAT). På en strimmel kan der være op til 5500 bytes iflg. Cauzin Systems & Eastman Kodak.

////////////////////////////////////

COMPAS PASCAL (PolyPascal) SPARE-PROGRAM

Inspireret af artiklen i sidste nummer angående spareprogrammet til TurboPascal, har jeg hevet det vedlagte program (RUN) ud fra skuffen. Det minder i funktion om programmet TUPAS fra samme nummer med den umiddelbare forskel at programmet er beregnet til COMPAS (og vel også PolyPascal).

Den primære årsag til at lave programmet var, at .COM-filer, udover at bruge plads til det oversatte program, også bruger ca. 8 kb til en "run-time" del. For at spare "run-time" delen, kan man oversætte programmet til en "object"-fil (.OBJ), til gengæld kræver .OBJ-programmer, at de kaldes fra et andet pascal-program med 'CHAIN' kommandoen i COMPAS.

Derudover kan programmet RUN også udføre de fleste .COM-programmer. Dette kan synes lidt overflødigt, men er ganske praktisk, hvis man ikke altid kan huske, hvilke programmer de er oversat til .OBJ eller .COM.

Dernæst er programmet forberedt på, at det kaldte program muligvis benytter nogle argumenter fra kommandolinjen, således at resten af kommandolinjen overføres til programmet. Nedenstående eksempel viser et kald af et udskriftsprogram, med angivelse af en tekstfil, der skal udskrives.
Eksempel:

```
RUN UDSKRIV C:TEKST.TXT
```

Ovenstående gør det muligt, at Pascal programmer kan udføres med SUBMIT, idet alle relevante oplysninger kan placeres i kommandolinjen og ikke behøver at angives interaktivt.

Programmet (RUN) skal oversættes til .COM-fil. Hvis alle programmer skal køre på det samme CP/M-system, behøver man ikke angive <TOP> feltet i PROGRAM eller OBJECT kommandoen i COMPAS. Men hvis man vil have, at programmerne skal kunne køre på flere maskiner, bør man angive <TOP> til en passende lav værdi, så der er plads nok i lageret. Jeg benytter selv værdien "CFFF".

Eksempel: COMPILE RUN,,CFFF eller OBJECT UDSKRIV,,CFFF

Vær opmærksom på, at der helst bør benyttes den samme værdi af <TOP> feltet for oversættelsen af RUN-programmet og oversættelsen af "object"-programmerne.

Om selve programmet skal der nævnes, at de variable der er mærket med "(* se tekst *)" er beregnet til nogle senere udvidelser, som jeg håber jeg kan nå at skrive om til efterfølgende numre af Z80-NYT. Herunder kommer blandt andet et MENU-program, så programmer kan kaldes ved blot at flytte markøren (cursor) hen til det ønskede program, og nogle procedurer som kan kaldes fra Pascal programmer, således at der fra kommandolinjen automatisk indlæses to filnavne, samt eventuelle parametre.

Frank Damgaard (13)

```

PROGRAM RUNOBJCOM; { udfører object/com filer fra Pascal ('COMPAS' m.v.) }
TYPE  comstrtp = string[255];
      charset = SET of CHAR;
VAR
  CHAINRETST : STRING[20];      (* se tekst *)
  CHAINLINE  : INTEGER;        (* - - *)
  CHAINSTR   : string[255];    (* - - *)
  CHAINFLAG  : byte AT $80;    (* - - *)
  f          : file;
  fname     : string[8];      boot      : INTEGER AT $0001;
  i, m      : Integer;      cpmname  : comstrtp AT $0080;
  tmpstr    : comstrtp;     cpmargs  : byte AT cpmname;
  ccpstr    : ^comstrtp;    skipchars, filechars : charset;
BEGIN
  ccpstr := ptr(boot -$15FC);      { CP/M ver 2.2 !! }
  filechars := ['0'..'9', 'A'..'Z', 'a'..'z', ':'];
  skipchars := [' ', ';'];
  IF cpmargs = 255 THEN BEGIN
    { hvis cpmargs = 255 er programmet kaldt fra et Pascal-program }
    WRITE('Fejl: RUN kan kun udføres som .COM fil!');
  END ELSE BEGIN
    { hvis cpmargs > 30 er der > 30 tegn i argumentet og da COMPAS nogle gange }
    { bruger dataområdet $80-$FF, hentes den rigtige tekst derfor fra CCP !! }

    tmpstr := ccpstr^ + ' '; m := len(tmpstr); i:=1; { Hop over blanktegn }

    WHILE (i < m) AND (tmpstr[i] IN skipchars) DO i := succ(i);
    tmpstr := copy(tmpstr, i, m-i+1); i := 1; m := len(tmpstr);
      { overspring progr.navnet (RUN) }
    WHILE (i < m) AND (tmpstr[i] IN filechars) DO i := succ(i);
    tmpstr := copy(tmpstr, i, m-i+1); m := len(tmpstr);
      { Fjern blanktegn i slut }
    WHILE (tmpstr[m] IN skipchars) AND (m > 1) DO m := pred(m);
    tmpstr := copy(tmpstr, 1, m); i := 1;

    WHILE (i <= len(tmpstr)) AND (tmpstr[i] IN skipchars) DO i := succ(i);
    tmpstr := copy(tmpstr, i, len(tmpstr) -i +1); fname := ''; i := 1; m := 8;
      { Find filnavnet, der skal kaldes, udfra argument: }

    WHILE (i <= len(tmpstr)) AND (tmpstr[i] IN filechars) DO
      BEGIN IF NOT (tmpstr[i] IN filechars) THEN m := 0;
            IF i <= m THEN fname := fname + tmpstr[i];
            i := succ(i);
      END;
    IF i <= len(tmpstr) THEN tmpstr[i] := ' ';
    IF len(tmpstr) > 0 THEN tmpstr := copy(tmpstr, i, len(tmpstr) -i +1)
    ELSE tmpstr := '';
    ASSIGN(f, fname + '.OBJ');
    WRITELN('Run: ', fname, tmpstr);
    CHAINSTR := tmpstr; ccpstr^ := tmpstr;
    IF len(tmpstr) <= 30 THEN cpmname := tmpstr ELSE
      BEGIN cpmname := copy(cpmname, 1, 30); cpmargs := len(tmpstr); END;
    CHAINRETST := ''; CHAINLINE := 0; m := 0;
    {$I- } chain(f); { $I+ } { Kald program, hvis det var .OBJ }
    IF IORES > 0 THEN ASSIGN(f, fname + '.COM'); { Dummy }
    {$I- } EXECUTE (f); { $I+ } { Kald program, som .COM }
    IF IORES > 0 THEN WRITE('Run-fejl: ', fname, ' eksisterer ikke!');
  END;
END.

```

//////////////////////////////////// ANNONCER //////////////////////////////////////

NYT GM-832 SVC-videokort + GM827E-tastatur : 3000,- kr.
 Nascom-2 mother board + keyboard + strømforsyning + GM802 64k RAM-board
 + GM803 EPROM / ROM-kort + Nascom I / O-kort + wrappt GM829-floppy
 controller kort : 5000,- kr.
 2 stk. TEAC FD55F 5.25" diskdrev : 1250,- kr. pr. stk.
 NEC 12" ravgul monitor : 1000,- kr.
 Gemini-kasse m. strømforsyning til diskdrev : 500,- kr.
 Extra GM802 64k RAM-kort med ombygninger : 750,- kr.

Ovenstående sælges til den anførte pris eller højeste bud. Desuden medfølger al dokumentation og software. Er du interesseret i det hele kan vi snakke om pris. Henv.: Jesper Skavin på 01 85 59 66 el. arb. 02 88 85 11 indtil 31/7. Bortrejt fra 10/7-26/7. Adresse: Tagensvej 205, 4th, 2400 KBH NV.

- - -

KONSTANT-SPÆND.TRAFO til indbygning.
 190-250V / 24V 50VA 180,- kr.
 190-250V / 12V 50VA 180,- kr.
 190-250V / 6V 25VA 150,- kr.
 Henv.: Viggo Jørgensen. 02 95 32 01.

- - -

//////////////////////////////////// BOG-BYT //////////////////////////////////////

Denne rubrik er et forsøg på at formidle f. eks. bøger, litteratur af teknisk art mv. Man kan forestille sig at nogen ved skift fra een 'model' til en anden har en del overskuds-litteratur, som andre medlemmer måske endnu ikke har læst osv. Der er muligheder nok, her er som eksempel:

- - -

CP/M HÅNDBOGEN, Thom Hogan, overs DK, 293 sider, pæn, sælges 120 kr. < 1/2 pris evt. byttes med ... Henv. (18-19) 02 95 32 01 Viggo Jørgensen.

- - -

////////////////////////////////////

+ ERIK HANSEN +

Søndag aften 28. juni sad jeg, efter invitation, hos min gode ven Erik Hansen og fik et måltid mad. Besøget gjaldt egentlig også computerens 'C'drev, et mindre, teknisk problem, som hurtigt blev klaret. Omkring kl. 22.30 tog jeg hjem... Næste dag blev jeg ringet op af Eriks kone, og hun fortalte, at Erik ikke var mere... Så hurtigt kan det gå! - Vi er mange, der mindes Erik fra hans arbejde i foreningen, ikke mindst i 'Pædagogisk Central-perioden'.

På manges vegne Ole Hasselbalch.

Modula-2

Peter Villadsen.

Modula-2 er et generelt anvendeligt sprog til programmering af datamater. Det indeholder så mange gode ideer og tanker og kan køre på maskiner med så begrænset kapacitet, at det er værd at investere den tid, det tager at læse denne artikel for at lære lidt om det.

Sproget er fostret af den østrigske ingeniør Niklaus Wirth, der også skabte Pascal. Ligesom Pascal afspejlede det stade, som datalogien var nået til, da det kom frem for 10 år siden, således afspejler Modula-2 det stade, som datalogien havde for et par år siden. Den største forskel er nok at finde indenfor modulstrukturen (som vi kommer tilbage til), men der er også væsentlige syntaktiske forbedringer, der gør sproget lettere at læse. Man kan sige, at Modula-2 er den logiske videre udvikling af Pascal, hvor der er foretaget generaliseringer og fjernet begrænsninger i Pascal.

Modulstrukturen

Som navnet Modula-2 antyder, er det i det faktum, at programmer kan opbygges af moduler, at de væsentlige forbedringer i forhold til ældre sprog skal findes. Lad os kort ridse op, hvad denne modul struktur indebærer:

En programdel består af en definitionsdel og en implementeringsdel. Definitionen er det, der fortæller omverdenen, hvad den pågældende programdel stiller til rådighed af funktioner, procedurer, typer o.s.v. Implementeringen er den del af programmet, der faktisk udfører programdelens funktioner. Denne simple opdeling har langt mere betydningsfulde konsekvenser, end man umiddelbart skulle tro. Den betyder blandt andet, at man kan opdele et program i en række mindre programdele. En programdel kan opbygges, bare de andre programdele af hvilken den er afhængig, er givet ved deres definitionsdele (men ikke nødvendigvis ved deres implementationsdele). Dette betyder, at man typisk kun har brug for at oversætte en lille del af et stort program (nemlig et implementeringsmodul) for at rette en fejl.

Definitionsdelene kan betragtes som en kontrakt, der skal opfyldes af implementeringsdelen. Det er ikke nødvendigt (eller ønskeligt), at pakkens klienter (dvs. de programdele, der bruger pakken) er vidende om, hvordan pakken udfører sit arbejde. Analogien mellem en kontrakt og en definitionsdel kan måske videreføres dertil, at hvis en kontrakt om et stykke arbejde indgås, er udbyderen af arbejdet principielt ikke interesseret i, hvordan arbejdet udføres, bare det bliver udført efter specifikationen.

Mekanismen med definitionsdel og implementeringsdel kan for eksempel bruges til at implementere datastrukturer på en tilfredsstillende måde. En datatype er som bekendt givet ved værdier og operationer på objekter af den pågældende type. Ideen er så at lave et definitionsmodul for hver datatype. Dette modul skal stille alle operationer, der logisk set kan foretages på objekter af typen samt definere datatypen, til rådighed. Et eksempel vil nok klare begreberne lidt. Vi har medtaget et eksempel på, hvordan man på denne måde kan implementere en stak. Følgende operationer er gyldige for stakke: PUSH, der stakker en værdi, POP, der afstakker en værdi og IsEmpty, der afgør om stakken er tom. Det er ikke defineret at afstakke en værdi fra en tom stak.

Modula-2 indeholder også andre nyheder for Pascal programmøren: En af de vigtigste er muligheden for at have flere processer, som skeduleres ved frivillig tidsdeling. Ved brug af denne facilitet kan mange algoritmer, der er vanskelige at lave strengt sekventielt, udformes let og elegant.

En anden forskel på Pascal og Modula-2 er, at de fleste af runtime rutinerne fremstår for brugeren, som om de var skrevne i Modula-2. De inkluderes i brugerens program, som var det et ganske almindeligt Modula-2 modul. Hvis man har brug for at lave en speciel version af en eller anden rutine i runtimebiblioteket, kan dette gøres helt smertefrit. Dette er en egenskab, som Modula-2 har til fælles med Ada og C.

Borland Modula-2

Til glæde for alle, der bruger Z80 eller Hitachi's nye 64180 kreds, har Borland International udviklet Turbo Modula-2. Dette er et aldeles fremragende system, der er opbygget ligesom Turbo Pascal, blot med mange forbedringer. Det er uden enhver tvivl den bedste Modula-2, man kan få til CP/M-80. Vi vil blot antyde nogle af de mange features, som denne implementering stiller til rådighed.

Som navnet antyder, er Turbo Modula-2 opbygget næsten ligesom Borlands Turbo Pascal. Der er en WordStar Editor, der kan bruges til at editere i kildeteksten. Selve oversætterdelen kan så oversætte denne kildetekst. Hvis der findes fejl under oversættelsen, hopper man ind i editoren igen på det sted, hvor fejlen blev fundet. Når fejlen er rettet, genoptages oversættelsen fra et sted bestemt af den foretagne rettelser, men altså ikke nødvendigvis forfra. Der kan oversættes enten til en intermediær kode, kaldet M-kode eller til rigtig Z80 kode. M-koden udmærker sig ved ikke at fylde ret meget, men til gengæld skal den udføres af en M-kode fortolker (som kan kaldes direkte fra oversætteren). M-koden kører forbløffende hurtigt. Man kan også vælge at generere Z80 kode, der så kan linkes sammen med programdele, der er skrevne i et format, der er kompatibelt med MicroSoft's L80 linker. Systemet omfatter også en linker og en librarian. Linkeren kan håndtere overlays, og konstatere konflikter imellem et definitionsmodul og dets implementeringsmodul (ved hjælp af en CRC kode). Kun de moduler, der er nævnt (transitivt) i programmets IMPORT sætninger, bliver medtaget i den færdige kode. Librarian-programmet kan vedligeholde biblioteker, som er filer, der indeholder de mange små filer, som fremkommer ved oversættelse af store programmer. Linkeren kan gennemsnøge sådanne biblioteker.

Borlands Modula-2 indeholder flere udvidelser til standard Modula-2 (som er defineret i bogen "Programming in Modula-2" af Niklaus Wirth, fra forlaget Springer Verlag). Hvis man er interesseret i at skrive flytbar kode, kan man ved en option sætte oversætteren i en tilstand, hvor disse udvidelser til sproget giver anledning til fejl.

En af de nyttigste udvidelser til sproget er de såkaldte exceptions, der er en avanceret kontrol struktur. Exceptions, der iøvrigt også findes i Ada, er en mekanisme, der kan "grave" sig ud af et eller flere underprogrammer, som givet ved den dynamiske kaldesekvens (og ikke det statiske virkefelt, som Pascals GOTO). Disse exceptions bruges oftest til at komme graciøst ud af en fejlsituation, for eksempel i det tilfælde, at man midt i en algoritme til bestemmelse af løsningen til n ligninger med n ubekendte opdager, at matricen er singular. Når man har lært dette system at kende, vil man se, at det er en meget elegant struktur. Alle runtimefejl (for eksempel division med 0, åbning af en fil, der ikke eksisterer etc.) giver ikke bare en fejlmeddelelse og stopper, men rejser en passende systemdefineret exception, der kan behandles af det omgivende program på en fornuftig måde.

De fleste af de datatyper, som Modula-2 stiller til rådighed, vil være alle Pascal programmører bekendt. Ud over disse har Turbo Modula-2 LONGINT (32 bit heltal der kan antage værdier i intervallet -2147483648 til 2147483647) og LONG REAL (flydende tal i intervallet -3.595386269724D+308 til 3.5953862697246D+308 med 14 betydende cifre).


```
IMPLEMENTATION MODULE StakHandler;  
FROM STORAGE IMPORT ALLOCATE;
```

```
CONST
```

```
  StakMax    = 101;
```

```
TYPE
```

```
  IndexRange = [ 1 .. StakMax ];
```

```
  StakType   = RECORD
```

```
    StakBody : ARRAY IndexRange OF Element;
```

```
    StakPtr  : IndexRange
```

```
  END;
```

```
  Stak      = POINTER TO StakType;
```

```
PROCEDURE Push (VAR s : Stak; e : Element;  
BEGIN
```

```
  IF s^.StakPtr = StakMax THEN
```

```
    RAISE StakOverflow;
```

```
  END;
```

```
  INC (s^.StakPtr);
```

```
  s^.StakBody [ s^.StakPtr ] := e;
```

```
END PUSH;
```

```
PROCEDURE Pop (VAR s : Stak) : Element;  
VAR
```

```
  e : Element;
```

```
BEGIN
```

```
  IF s^.StakPtr = 1 THEN
```

```
    RAISE StakIsEmpty;
```

```
  END;
```

```
  e := s^.StakBody [s^.StakPtr];
```

```
  DEC (s^.StakPtr);
```

```
  RETURN e
```

```
END Pop;
```

```
PROCEDURE IsEmpty (s : Stak) : BOOLEAN;  
BEGIN
```

```
  RETURN s^.StakPtr = 1
```

```
END IsEmpty;
```

```
PROCEDURE NewStak () : Stak;  
VAR
```

```
  s : Stak;
```

```
BEGIN
```

```
  NEW (s);
```

```
  s^.StakPtr := 1;
```

```
  RETURN s
```

```
END NewStak;
```

```
END StakHandler.
```

```

DEFINITION MODULE StakHandler;
  TYPE Stak;                (* Vi specificerer ikke detaljerne om *)
                           (* hvordan en stak er implementeret *)
  TYPE Element = INTEGER;
  PROCEDURE Push (VAR s : Stak; e : Element);
  (* Stak elementet e på stakken s *)
  PROCEDURE Pop (VAR s : Stak) : Element;
  (* Afstak s, og returner elementet *)
  PROCEDURE IsEmpty (s : Stak) : BOOLEAN;
  (* Returnerer TRUE hvis stakken er tom *)
  PROCEDURE NewStak () : Stak;
  (* Initialiserer et stak objekt *)
  EXCEPTION
    StakOverflow,          (* Rejses hvis stakkens kapacitet overskrides *)
    StakIsEmpty;         (* Rejses hvis tom stak pop'pes *)
END StakHandler.

```

```

-----
MODULE StakTest;
  IMPORT StakHandler;      (* Importerer alt fra stakhandleren *)
  FROM   InOut   IMPORT WriteString, WriteInt, WriteLn,
                    Read, ReadInt, EOL;
  VAR
    s : StakHandler.Stak;
    i : INTEGER;
    ch : CHAR;
  BEGIN
    S := StakHandler.NewStak();
    LOOP
      Read (ch);
      CASE CAP (ch) OF
        ! 'U' : ReadInt (i);
                  StakHandler.Push (s, i)
        ! 'O' : i := StakHandler.Pop (s);
                  WriteInt (i, 5)
        ! 'E' : IF StakHandler.IsEmpty (s) THEN
                  WriteString ("Stakken er tom");
                ELSE
                  WriteString ("Stakken er ikke tom");
                END (* if *)
        ! 'Q' : EXIT
        ! EOL : (* Ingenting *)
      ELSE
        WriteString ("Kun U, O, E, Q tilladt");
      END; (* if *)
      WriteLn;
    END
  EXCEPTION
    StakHandler.StakOverflow :
      WriteString ("For mange elementer på stakken");
      WriteLn !
    StakHandler.StakIsEmpty :
      WriteString ("Elementer kan ikke afstakkes fra tom stak");
      WriteLn;
  END StakTest.

```

De matematiske funktioner er selvfølgelig defineret både på REAL og LONGREAL typer. Yderligere kan man have variable af typen PROC (<typelist>), altså procedurevariable. Dette er en generalisering af det begreb, der kendes (i en indskrænket form) i Standard Pascal (ikke i Poly/Turbo Pascal), hvor underprogrammer kan overføres som argument til underprogrammer.

Kørende Turbo Modula-2 programmer har mulighed for at bruge redirection, og har tilgang til de argumenter, der blev givet på kommandolinien. Dette system (der iøvrigt kendes fra UNIX og MS-DOS) gør det muligt at designe programmer med en homogen brugersnitflade. Store programmer kan skrives ved hjælp af overlays, næsten ligesom i Turbo Pascal.

Dokumentationen til systemet er en over 500 sider lang gedigen bog, der kommer ud i alle detaljer. Det er ikke en lærebog i Modula-2 programmering, men en glimrende referencemanual for netop denne implementering. Bogen er iøvrigt umuligt at kopiere uden at ødelægge den.

Systemet kan bestilles hos VilliSoft, Postboks 164, 2650 Hvidovre.

Pris kr. 850,00 + evt. diskettekonvertering + moms.

////////////////////////////////////
Kopiering af CP/M disketter i non Commodore format.

Der har været tvivl om Commodore kunne klare, hvad de lover med hensyn til overførelse af filer fra andre formater. I den anledning har Commodore skrevet et brev, som gengives her.

Kopieringen foregår som beskrevet i vejledningen. Hvis CP/M selv præcist kan identificere det pågældende format, vil kopieringen forløbe som med diskette i Commodore format. Det er i vejledningen til Commodore 128, afsnit 11 nævnt, hvilke formater C1571 understøtter.

Hvis CP/M ikke selv kan identificere formatet under forsøg på læsning, vil der nederst på skærmen fremkomme navnet på det format, som CP/M formoder, det er. Her kan man så vælge at godkende format-forslaget eller skifte til det nærliggende format med piletasterne øverst på tastaturet. Eksempelvis kan CP/M foreslå Kaypro II, selvom formatet er Kaypro IV. Her kan man så skifte til dette format ved at trykke på pil-til-højre-tasten. Formatet godkendes endeligt ved tryk på RETURN-tasten. Man kan også låse formatvalget ved at trykke på CTRL-tasten og holde den nede, mens man trykker på RETURN-tasten. Hvis man gør det, vil CP/M huske valget næste gang den skal læse fra disketten.

NB: Diskettestation 1571 kan kun læse de i vejledningen nævnte formater. IBM-8 formaterne skal være formatteret under CP/M operativsystemet. Programmer kan kun udføres på C128, hvis de er lavet til en Z80 processor, eksempelvis kan programmer fra CP/M på en IBM PC ikke udføres, idet denne maskine bruger en 8088 processor. Husk at formattere en diskette før kopieringen sættes i gang.

Husk, at tilsvarende gælder for overførelse mellem Piccoline og CP/M 80. Det er i første omgang kun datafiler & prog.source i ren ASCII, der er nyttige. Men der eksisterer programmer, der emulerer den anden processor. Det kan blandt andet lade sig gøre at afvikle Z80 programmer på en IBM PC!!!
Det er jo nærmest at gå over åen for at hente vand, men de findes.

Asbjørn.

Z280 = 8-bit CP/M med 2x IBM-AT hastighed ?

Langt om længe... Z280'eren er her! Her er alt hvad der oprindeligt blev lovet om Z800 og mere til. I de sidste 4 år har der været allehånde rygter om Z800, en super-chip, der skulle være kompatibel med Z80. Det trak imidlertid ud og i mellemtiden kom så Hitachi's HD64180, som en 'udvidet' Z80, og da selv Zilog indgik aftale om salg / produktion af henholdsvis HD64180, som er delvis kompatibel med Z80 og en variant Z180, der er helt 'nedad' kompatibel også til Z80's periferi-kredse (hvilket HD64180 ikke er!), da var der vist ikke mange som troede på Z800. Den har fået nyt navn, men nu er den klar, og vi får se...

Z280 har foruden alle Z80-instruktioner (og dermed også 8080's) yderligere et udvidet sæt, bestående af 45 nye instruktioner samt flere adresseringsmåder. Der er både 8 & 16 bit multiplikation og division med eller uden fortegn! ialt mere end 600 instruktioner.

Med en clock frekvens på 10 MHz eller mere, 16 M byte adresse-rum og 8-bit Z80 Bus eller 16-bit Zilog Z-Bus, giver det langt flere muligheder end med Intels's 80286, der anvendes i IBM PC AT.

Nye adresseringsmåder betyder også, at man nu kan opnå en væsentlig bedre effektivitet ved anvendelse af højniveau-sprog end tidligere.

Anvender man f.eks. Z-Bus'en, har man et udvidet sæt af Z80 instruktioner til sin rådighed og "en ægte 16-bit'er" (hvad jo også 80286 er).

Zilog Z280 er en højt integreret MPU, med hovedpunkter / funktioner såsom: Paged Memory-Management afdeling med en 256 bytes data- og instruktions- Cache; en tre-trins instruktions 'Pipelining', (dvs. en anordning, som forsyner f.eks. en CPU med "tygget mad" og derfor kan bearbejdes meget hurtigere); en 4-kanal DMA controller; 3 stk. 16-bit Counter/timers; en 6-trin Wait-state generator; DRAM refresh controller; samt en high-speed UART med hastighed op til 2,5 MHz.

For at få lidt bedre hold på / indtryk af, hvad denne nye MPU kan bruges til, er her valgt en beskrivelse af et projekt, som synes at være realistisk, det lyder i det mindste interessant og vi er straks gået igang med at afprøve 'sandhedsværdien', for i et kommende nr. af Z80-NYT evt. at kunne bringe en mere detaljeret beskrivelse og måske en praktisk demonstration.

Indtil videre må vi nøjes med at fortælle om MPU'en, og det følgende er i væsentligt omfang baseret på pressemeddelelser m.v. fra de involverede firmaer!

Men her er et 'typisk' eksempel på hvad der kan opnås ved en 'ombygning' af en almindelig, ikke helt ny CP/M Z80-computer:

Et amerikansk firma planlægger at bruge Zilog Z280 i et "Zuperboard", dvs. et printkort beregnet til indbygning i en eksisterende Z80 CP/M computer, og for tiden er man igang med de afsluttende prøver inden en frigivelse til salg, formentlig allerede i sommer 1987 til en forventet pris på under \$ 500.

Brugere af denne CP/M computer der køber et sådant kort, kan selv foretage 'ombygningen' idet man blot skal tage dækslet af computeren, fjerne Z80'eren, anbringe Zuperboard'et istedet, forbinde monitorkablet til det nye Zuperboard og man har nu en computer med arbejdshastighed baseret på 12 MHz mod før 4 MHz. Det giver op til 10x større effektivitet i forhold til den tidligere computer i situationer, som ikke er afhængig af I/O operationer (f.eks. disk-drev).

Kortet leveres med 1 M byte RAM (der kan udvides til 16 M byte!) og kan af brugeren anvendes som RAM-disk. Har man en ekstra RGB farveskærm tilsluttet, kan man selv bestemme 'forgrunds- og baggrundsfarver'. Endvidere giver kortet mulighed for at arbejde med Virtuel Memory og Cache Memory.

Z280'erens Cache Memory gemmer automatisk alle de senest anvendte instruktioner og/eller data, derved har processoren hurtig adgang til disse, idet der bl.a. ikke kræves en fuldt opsat adresse, men kun en 'mindre lokal-instruktion' derved kan memory- eller diskoperationer reduceres. På den måde er det i mange tilfælde muligt at spare indtil 75% af normal busbrug. (hvis man er heldig! *)

Instruktions pipelining virker, som om man har 3 co-processorer til rådighed på samme chip til intern behandling af flere instruktioner samtidig; f.eks. medens den første processor bearbejder een instruktion er processor nr.2 og nr.3 igang med de næste to instruktioner. Pipelining bidrager derfor meget væsentligt til Zuperboard'ets forøgede præstationer.

Under udviklingsarbejdet viste det sig nødvendigt at udskifte computerens skærm-driver så skærmen kunne følge med til Z280's arbejdshastighed. Zuperboard indeholder derfor nu en ny, 18 MHz no-wait-state, grafik skærm-driver med egen 256 byte Cache Memory til hurtig skærm-opdatering. 25 gange hurtigere end med den gamle skærm; så nu skulle skærmen kunne følge med programmerne.

Den nye skærmdriver leveres med et RGB-stik så der er mulighed for at forbinde en extra farvemonitor til computeren. Man kan lave eget tegnsæt og skifte mellem dette og computerens standardsæt. Grafik-delen (GKS) på Zuperboard'et indeholder et subset af den nye GKS standard.

Zuperboard'et understøtter multi-tasking og networking. Men ligesom for -286 og -386 processorerne afventes færdiggørelse af et operativsystem specielt til den nye processor for at kunne udnytte alle dens muligheder.

Strømforsyning/Batteri sørger for at RAM'en hele tiden kan holdes 'i live' dvs. aktiv, klar til brug, under cold boot, ved netsvigt eller når der er slukket for computeren. Zuperboard'et har endvidere pin-outs for en SCSI bus (Small Computer System Interface), det giver mulighed for op til 300 Megabytes hard disk memory. I RAM'en vil der iøvrigt også være plads til en rimelig stor 'lommeregner og notesbog' mv.

Herudover er der under udvikling et 'high speed interface' til brug mellem Zuperboard'et og en PC, man kan f.eks. forbinde CP/M-computeren med en billig PC-klon og bruge den som hard-disk lager og evt. som en 'network file server'.

Såvidt så godt... med hensyn til hurtighed alene skulle Zuperboard'et være sammenlignelig med en standard AT'er, men hvad med software til at udnytte alle de nye muligheder? Standard CP/M programmer skulle uden videre kunne køre, men de udnytter jo ikke den ekstra lagerplads, som er til rådighed, derfor må denne bruges f.eks. som RAM disk. Konstruktørerne er dog i forbindelse m. forskellige grupper, som er interesserede i at konvertere programmer til den nye IC. Afhængig af chip'ens fysiske muligheder på kortet skulle det nok være muligt at omforme et program så det 'ser efter', om der er mere RAM udover 64 K og i givet fald så bruge det, mener specialister.

(samlet fra div. kilder: VJ)

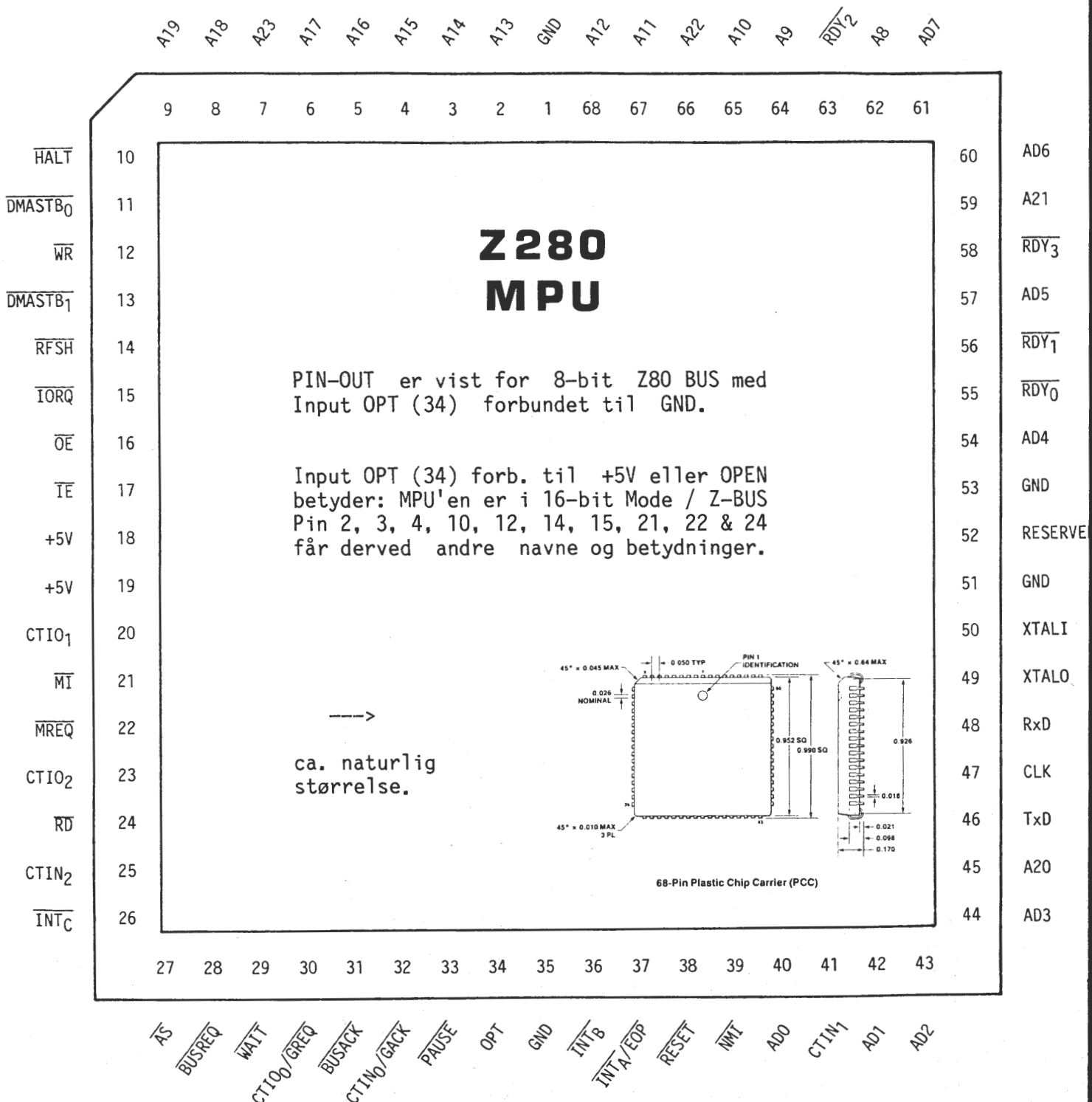
*) hit-rate (=træf %), man kan aldrig være helt sikker på at et forbehandlingsarrangement, her pipelining, altid vil tage de rigtige instruktioner/data frem, når der i cpu'en kan frembringes resultater, som f.eks. betyder uforudsete hop; bl.a. derfor kan man ikke opnå en hit-rate på 100%, og iøvrigt er der blandt 'specialister' megen uenighed om hvor stor en hit-rate bør være for at kunne betale sig; noget tilsvarende gælder om størrelsen på Cache memory.

Det har vist sig at der også i Tyskland findes mindst en fabrikant, som er ved at være klar med et CPU-kort idet de meddeler at deres ECB-Z280 allerede venter på levering af Z280'ere "i antal" og de tager gerne imod forudbestillinger:

10 MHz Z280 CPU mit Cache, MMU und 4xDMA Speichersockel 32-polig bytewide, davon 4 akkugepuffert. Max. 768k Eprom oder 256k und 512k CMOS-RAM. Watchdog, Spannungsüberwachung, Tageszeit, V.24 mit RTS/CTS oder 20mA TTY, User-LED's, Normgerechter SCSI-Bus (DMA) mit WD33C93. Karte mit SCSI 1137,72 DM, ohne 997,50 DM Sandwich mit 2 MB dyn RAM 1137,72 DM.

Priserne er benævnt 'vejledende'. Formentlig incl. 15% Mehrwert Steuer (MOMS).

////////////////////////////////////



0	000000000	128	100000000
1	000000001	129	100000001
2	000000010	130	100000010
3	000000011	131	100000011
4	000000100	132	100000100
5	000000101	133	100000101
6	000000110	134	100000110
7	000000111	135	100000111
8	000001000	136	100001000
9	000001001	137	100001001
10	000001010	138	100001010
11	000001011	139	100001011
12	000001100	140	100001100
13	000001101	141	100001101
14	000001110	142	100001110
15	000001111	143	100001111
16	000100000	144	100100000
17	000100001	145	100100001
18	000100010	146	100100010
19	000100011	147	100100011
20	000101000	148	100101000
21	000101001	149	100101001
22	000101010	150	100101010
23	000101011	151	100101011
24	000110000	152	100110000
25	000110001	153	100110001
26	000110010	154	100110010
27	000110011	155	100110011
28	000111000	156	100111000
29	000111001	157	100111001
30	000111010	158	100111010
31	000111011	159	100111011
32	001000000	160	101000000
33	001000001	161	101000001
34	001000010	162	101000010
35	001000011	163	101000011
36	001000100	164	101000100
37	001000101	165	101000101
38	001000110	166	101000110
39	001000111	167	101000111
40	001010000	168	101010000
41	001010001	169	101010001
42	001010010	170	101010010
43	001010011	171	101010011
44	001010100	172	101010100
45	001010101	173	101010101
46	001010110	174	101010110
47	001010111	175	101010111
48	001100000	176	101100000
49	001100001	177	101100001
50	001100010	178	101100010
51	001100011	179	101100011
52	001101000	180	101101000
53	001101001	181	101101001
54	001101010	182	101101010
55	001101011	183	101101011
56	001110000	184	101110000
57	001110001	185	101110001
58	001110010	186	101110010
59	001110011	187	101110011
60	001111000	188	101111000
61	001111001	189	101111001
62	001111010	190	101111010
63	001111011	191	101111011
64	010000000	192	110000000
65	010000001	193	110000001
66	010000010	194	110000010
67	010000011	195	110000011
68	010000100	196	110000100
69	010000101	197	110000101
70	010000110	198	110000110
71	010000111	199	110000111
72	010001000	200	110001000
73	010001001	201	110001001
74	010001010	202	110001010
75	010001011	203	110001011
76	010001100	204	110001100
77	010001101	205	110001101
78	010001110	206	110001110
79	010001111	207	110001111
80	010100000	208	110100000
81	010100001	209	110100001
82	010100010	210	110100010
83	010100011	211	110100011
84	010100100	212	110100100
85	010100101	213	110100101
86	010100110	214	110100110
87	010100111	215	110100111
88	010110000	216	110110000
89	010110001	217	110110001
90	010110010	218	110110010
91	010110011	219	110110011
92	010111000	220	110111000
93	010111001	221	110111001
94	010111010	222	110111010
95	010111011	223	110111011
96	011000000	224	111000000
97	011000001	225	111000001
98	011000010	226	111000010
99	011000011	227	111000011
100	011001000	228	111001000
101	011001001	229	111001001
102	011001010	230	111001010
103	011001011	231	111001011
104	011010000	232	111010000
105	011010001	233	111010001
106	011010010	234	111010010
107	011010011	235	111010011
108	011011000	236	111011000
109	011011001	237	111011001
110	011011010	238	111011010
111	011011011	239	111011011
112	011100000	240	111100000
113	011100001	241	111100001
114	011100010	242	111100010
115	011100011	243	111100011
116	011100100	244	111100100
117	011100101	245	111100101
118	011100110	246	111100110
119	011100111	247	111100111
120	011110000	248	111110000
121	011110001	249	111110001
122	011110010	250	111110010
123	011110011	251	111110011
124	011111000	252	111111000
125	011111001	253	111111001
126	011111010	254	111111010
127	011111011	255	111111011

////////////////////////////////// BITS ////////////////////////////////////

hvordan er det nu? Nåh, ja det er klart

nå! - heller ikke?

...HER ER DEN lille TABEL !!!

40 spor på 80 spor-drev? Flere læsere har spurgt om det kan lade sig gøre at anvende en 40 spors diskette på et 80 spors drev. I nogle tilfælde er det muligt. En hindring er ofte at det pågældende drev ikke har en direkte omstillingsmulighed (mekanisk/elektr.), men det er der råd for! Dels kan man jo pr. software dobbelt-stæppe, og endelig er det tit muligt ved hjælp af et mindre, operativt indgreb at opnå det ønskede resultat. Nogle disk-drev har endog en medfødt omstillingsmulighed!

Vi har på diagram-lageret bl.a. Shugart 465, som er omstillelig, og håber på velvillig assistance fra tekn. interesserede/erfarne læsere med andre drevtyper til næste nr.

Fornylig kom posten med brochurer over disketter, det er ikke usædvanligt, men man studsede over at der forlangtes en højere pris for Double Density disketter end for Single Density ditto. Firmaet gav, på forespørgsel, den begrundelse at der kan 'være meget mere' på en sådan! Det er jo da også rigtigt. Men måske er det en god ide med en lille forklarende artikel om SD / DD? (i nær fremtid)

Lysledere. De omtales tit, som en fin løsning på mange opgaver, men når man vil prøve denne teknik i praksis til privat brug finder man snart ud af at det ikke er helt billigt! Et par hundrede kr. i hver ende af et 10m lysleder kabel til stik/fatn. + kabel til måske 500 kr.! - Tjah, sølv- og guld-tråds kabler bruges jo som højttaler ledninger af 'audio-folk', - men billigt, det er det nu ikke. Men der er lys forude! Der er nu kommet lysdioder og fotodioder på markedet, som letter situationen de ligner små plastrør med en indstøbt LED eller fotodiode, som 'en prop' i den ene ende, i modsat ende passer hullet til et 1mm standard plast-lysleder kabel med 2,1mm kappe, det kan afskæres med en skarp kniv til ønsket længde. (ingen polering)

CCD & Telefax er emner, som vi evt. kan tage op.

Har du noget, som måske kan være af interesse for andre medlemmer så fortæl os lidt om det, og hvis nødvendigt hjælper vi med 'indpakningen'

MVHVJ