

Z80-NYT

8. ÅRGANG - NR. 3. 1987

- Side 2 Foreningsoplysninger
- 3 Orientering
 - 4 BDS-C / rutiner
 - 5 BDS-C / minipip
 - 8 Murphy
 - 9 Oscilloskop - brug det rigtigt
 - 10 Lidt om grafik
 - 14 Indkøbsforening Udsalg / tilbud
 - 14 Noter
 - 15 Z80 mc-rutiner
 - 16 Annoncer
 - 16 Bog-Byt
 - 17 Benchmarks
 - 18 Farvebånd/Genbrug
 - 19 Single Density ? Double Density ?
 - 22 Z280 fortsat...
 - 24 Z80 ? eller Z280 ?



Desk Top Publishing

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: pt. Kurt Neierdi se adr. ovenfor. (skriftligt)

Forretn.f: pt. Frank Damgaard, Indkøbsforeningen.

Redaktion: Z80-NYT, Fensmarks Alle 6 3520 Farum

Logo: C. B. Senholt.

Redaktør: pt. Viggo Jørgensen.

Private annoncer gratis for medlemmer.

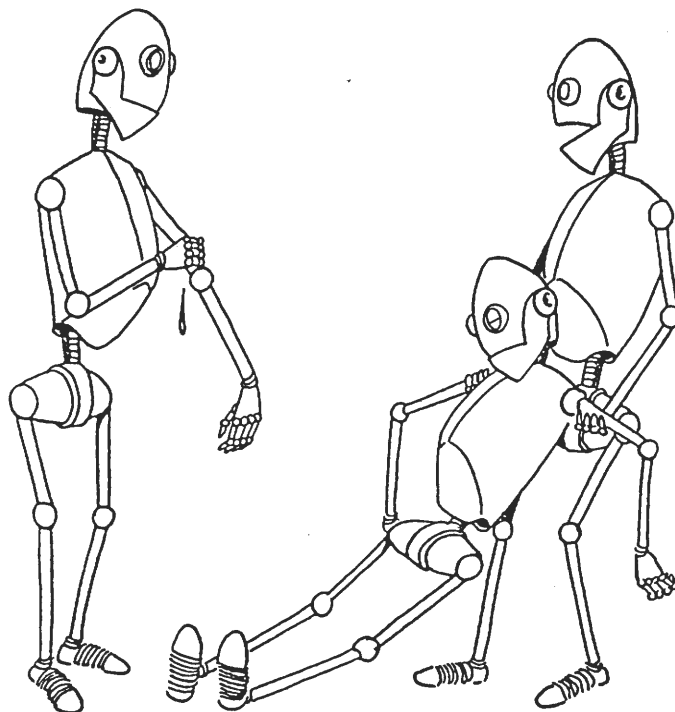
Kontingent årligt: 300 kr. Indmeld.gebyr: 25 kr.

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

```
////////////////////////////////////
/
/ Der køres modemforsøg med data-overførsel på telf. 01 91 38 98, der kan
/ kaldes hele døgnet. 'Linie-op sætning' er 1200 / 1200 fuld duplex
/ 1 startbit, 8 databit, 1 stopbit, ingen paritet.
/
/ Aftal nærmere på telf. 01 11 38 94 helst i kontortiden.
/
/ Formanden.
/
////////////////////////////////////
```

-- o --

3



"He can't stand the sight of oil."

MODEM SIDEN SIDST...

Det nye modem, som er anskaffet af en del Z80-medlemmer, sender og modtager med 1200 / 1200 fuld duplex. Dette modem anvender en seriel kommunikationsport, af typen RS 232C, (samme som V24 + V28) samt et styreprogram der anvender HAYES koder:

f.eks.: KERMIT.COM til CP/M eller PROCOMM.COM til PC'ere.

Er du interesseret så skriv til foreningen, anfør computer og hvilke diskette-formater du har og få tilsendt materiale. Husk at skrive medl.nr. navn, adresse og telf.nr.

PROGRAM-BIBLIOTEKET - får ny adresse.

Foreningens bibliotekar gennem mange år, Asbjørn Lind, har længe ønsket at blive afløst, - det har faktisk ind imellem været et temmeligt krævende job, ikke mindst m.h.t. tid - og for bl.a. at kunne få plads til andre, interessante opgaver har AL. derfor valgt at afvikle nu.

Mange medlemmer har i tidens løb nydt godt af 'en konverterings-hjælp' og med håbet om at foreningen også fremover, kan nyde godt af AL's erfaring, siger medlemmer og bestyrelse Asbjørn Lind tak for veludført arbejde.

CP/M biblioteket skulle pr. første september være flyttet til formandens adresse, men af forskellige tekniske grunde er det imidlertid blevet lidt forsinket, vi håber at kunne ordne det praktiske i løbet af kort tid, uden at det af den grund giver nogen væsentlig, mærkbar forsinkelse.

Det er under overvejelse at udvide/modernisere biblioteks-funktionen, evt. i forbindelse med database og postkassen. (mailbox på 01 91 38 98)

Kurt Neierdi/formand.

BDS-C rutiner

For ca. et halvt år siden fik jeg en BDS-C compiler til min James pc800 og kort tid efter faldt jeg også over en manual, så jeg ikke længere var overladt til "forsøg - fejl" metoden.

Jeg er blevet meget glad for sprogets muligheder, men jeg synes der mangler en facilitet, som jeg ofte har benyttet mig af i PolyPascal, nemlig frit at kunne dirigere input- og output-text til disk eller ydre enheder ved blot at ændre filens navn. Det er let i pascal at give filen navnet "LST:" hvorefter output går på printeren og et øjeblik efter blot ændre navnet til "XXX.ASM" og vupti! ...gik output ud på disk.

Nu har jeg til C barslet med 3 funktioner, der skulle klare dette problem, og har kaldt dem XFOPEN, XFCREAT og XFCLOSE. Fidusen er at BDS-C's standard-funktioner GETC og PUTC giver mulighed for at operere på disk og ydre enheder efter eget valg, men efter et andet princip, der ikke er styret af filnavnet. Ovennævnte to funktioner arbejder ud fra en adresse på en file-buffer. Da en sådan buffer sikkert aldrig vil ligge på adresse 0 til 4 er disse adresser benyttet til at markere hvilke ydre enheder rutinerne skal arbejde på.

adr	CP/M enhed	C-betegnelse
0	CON:	STD_IN
1	CON:	STD_OUT
2	LST:	DEV_LST
3	RDR:/PUN:	DEV_RDR/DEV_PUN (jeg kalder dem AUX: og DEV_AUX).
4	CON:	STD_ERR

Når man normalt benytter buffered I/O defineres et bufferareal, hvis adresse videregives til de funktioner, der skal bruge den. Men i vort tilfælde skal der derudover defineres en pointer til denne buffer, og det er VÆRDIEN af denne pointer, der videregives til læsende og skrivende funktioner. Denne pointer kan nu enten indeholde adressen på bufferarealet, eller antage værdierne fra 0 til 4.

Nu er det blot at skrive funktioner, der kan oversætte "CON:", "LST:" og "AUX:" til 0, 1, 2 og 3. Disse rutiner er vist i forbindelse med artiklen.

For at hjælpe lidt på forståelsen af brugen har jeg bragt rutinerne i et mini-PIP program skrevet i C. Det skal understreges, at XFOPEN, XFCREAT og XFCLOSE kræver ADRESSEN på ovennævnte pointer, da pointeren skal kunne ændres. Husk også at pointeren inden kald af XFOPEN og XFCREAT skal 'pege' på bufferarealet.

(746) Ebbe Sønderhousen. Smedegade 3 B, 5900 Rudkøbing. (09) 51 40 02

ANNONCE!

Er der nogen, der koder i BDS-C og bruger den medfølgende debugger CDB?

Jeg har brugt CDB, og er blevet træt af at tælle linier i C-sourcen (især tælle galt). Årsagen er at CDB tillader at single steppe gennem C-programmet statement for statement, og CDB fortæller hvor den er ved at angive navnet på funktionen og source linienummeret indenfor funktionen.

Da jeg ikke er vidende om noget eksisterende program, der kan linienummerere en C-source funktion for funktion, har jeg derfor lavet et sådant; og resultatet / virkningen kan ses på udskriften af MINIPIP.C

I tilfælde af interesse hører jeg gerne fra vedkommende, idet jeg ikke agter at holde sourcen hemmelig.

Ebbe Sønderhousen. (se adr. ovenfor)

/* minipip

Dette program giver mulighed for at overføre tekst til og fra filer eller ydre enheder.

Kald:

MINIPIP <input-name> <output-name>

Både <input-name> og <output-name> kan være navne på filer eller ydre enheder. De ydre enheder har følgende navne:

Consol = CON:
List = LST:
Aux = AUX:

En kommandolinie kan f. eks. se sådan ud:

MINIPIP BREV LST:

EOF erkendes når en fil er læst til ende eller første gang CPMEOF mødes.
*/

```
#include bdscio.h
```

```
#define STD_IN 0          /* keyboard via getchar */  
#define STD_OUT 1       /* skærm via putchar */  
#define STD_ERR 4      /* skærm via CON (BDOS-kald 2) */
```

```
#define DEV_LST 2       /* printer via LST (BDOS-kald 5) */  
#define DEV_AUX 3      /* input via RDR (BDOS-kald 3)  
                       output via PUN (BDOS-kald 4) */
```

```
char in_name[14],       /* input file/device name */  
     out_name[14];     /* output file/device name */
```

```
FILE in_buffer,        /* bufferareal, der benyttes hvis input kommer  
                       fra en fil */  
     *ptr_to_in_buffer, /* pointer til ovennævnte bufferareal eller til  
                       input device */  
     out_buffer,       /* bufferareal, der benyttes hvis output skal  
                       til en fil */  
     *ptr_to_out_buffer; /* pointer til ovennævnte bufferareal eller til  
                       output device */
```

```
1: main(argc,argv)  
2:   int argc;  
3:   char *argv[];  
4:  
5: { int c, i;  
6:   char *arg;  
7:  
8:   if (argc != 3)  
9:     use_err();  
10:  
11: /* bufferpointerne initieres til at pege på deres buffer */  
12:   ptr_to_in_buffer = &in_buffer;  
13:   ptr_to_out_buffer = &out_buffer;  
14:  
15:   strcpy(in_name,argv[1]);  
16:   strcpy(out_name,argv[2]);  
17:
```

```

18: /* åbning af fil eller ændring af bufferpointer til at pege på input device */
19:   if (xfopen(in_name,&ptr_to_in_buffer) == ERROR)
20:   { fprintf(STD_ERR,"Open error on %s : %s\n",in_name,errmsg(errno)); .
21:     exit();
22:   }
23:
24: /* opret ny fil eller ændring af bufferpointer til at pege på input device */
25:   if (xfcreat(out_name,&ptr_to_out_buffer) == ERROR)
26:   { fprintf(STD_ERR,"Creat error on %s : %s\n",out_name,errmsg(errno)); ....
27:     exit();
28:   }
29:
30: /* kopiering indtil CPMEOF mødes eller fysisk EOF er nået */
31:   while ((c = getc(ptr_to_in_buffer)) != ERROR && c != CPMEOF)
32:   { if (putc(c,ptr_to_out_buffer) == ERROR)
33:     { fprintf(STD_ERR,"Output error on %s : %s\n",out_name,errmsg(errno));
34:       exit();
35:     }
36:   }
37:
38:   if (xfclose(&ptr_to_in_buffer) == ERROR)
39:   { fprintf(STD_ERR,"Close error on %s : %s\n",in_name,errmsg(errno));
40:     exit();
41:   }
42:
43:   if (ptr_to_out_buffer > STD_ERR)
44:     if (putc(CPMEOF,ptr_to_out_buffer) == ERROR)
45:     { fprintf(STD_ERR,"Output error on %s : %s\n",out_name,errmsg(errno));
46:       exit();
47:     }
48:
49:   if (xfclose(&ptr_to_out_buffer) == ERROR)
50:   { fprintf(STD_ERR,"Close error on %s : %s\n",out_name,errmsg(errno));
51:     exit();
52:   }
53: }

1: use_err()
2: { fprintf(STD_ERR,"\nUsage: MINIPIP <input-name> <output-name>\n");
3:   exit();
4: }

1: int xfopen(id,buf)
2:   char *id;
3:   FILE **buf;
4: { if (strcmp(id,"CON:") == 0)
5:   { *buf = STD_IN;
6:     return 0;
7:   }
8:   else
9:   if (strcmp(id,"AUX:") == 0)
10:  { *buf = DEV_AUX;
11:    return 0;
12:  }
13:  else
14:    return fopen(id,*buf);
15: }

```

```

1: int xfcreat(id,buf)
2:  char *id;
3:  FILE **buf;
4:  { if (strcmp(id,"CON:") == 0)
5:    { *buf = STD_OUT;
6:      return 0;
7:    }
8:    else
9:    if (strcmp(id,"LST:") == 0)
10:   { *buf = DEV_LST;
11:     return 0;
12:   }
13:   else
14:   if (strcmp(id,"AUX:") == 0)
15:   { *buf = DEV_AUX;
16:     return 0;
17:   }
18:   else
19:   if (strcmp(id,"ERR:") == 0)
20:   { *buf = STD_ERR;
21:     return 0;
22:   }
23:   else
24:     return fcreat(id,*buf);
25: }

```

```

1: int xfclose(buf)
2:  FILE **buf;
3:
4:
5:
6: { if (*buf <= STD_ERR)
7:   return 0;
8:   else
9:   return fclose(*buf);
10: }

```

/* buf kunne lige så godt være erklæret som FILE *buf og xfclose således kaldt uden '&'. Den her anvendte form er for at bevare samme 'kaldsyntax' som xfoen og xfcreat */

CDBL: Line numbers for CDB debugger, Source: MINIPIP.C

Page 1

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

MODEM: 2400 bps, V22 bis, fuld duplex, auto dial (puls & tone), auto answer, Hayes osv., 'stand alone' type, RS 232 port nødv., pris ved gruppekøb: 6 stk. er (incl. moms.) 2500 kr. at betale ved bestilling, leveringstid 5-6 uger. Eventuelle interesserede bedes henvende sig til redaktionen.

//////////////////////////////////// 02 95 32 01



MURPHY

1'ste lov: Hvis noget kan gå galt, så vil det gøre det.

2'den - : Såfremt der er en mulighed for at flere ting kan gå galt,
vil det, som gør størst skade, ske først.

3'die - : Når absolut ingenting kan gå galt, vil det gøre det alligevel.

4'de - : Hvis du mener, at der er fire mulige måder at få en ting til at
mislykkes på, og du klarer at undgå disse fire, så vil der
øjeblikkeligt dukke en femte op, som du ikke har taget hensyn til.

5'te - : Dersom tingene overlades til sig selv,
tenderer de til at gå fra ondt til værre.

6'te - : Hvis alt ser ud til at gå godt, så har du åbenbart overset noget.

7'de - : Naturen er altid på samme side (parti) som den skjulte fejl.

8'de - : Moder natur er en heks.

O'TOOLEs kommentar til Murphys love: Murphy var en optimist.

FORSYTHs tillæg til Murphys love:

Når du endelig skimter lyset i den anden ende af tunnelen,
- så ramler loftet ned.

PUDDER's love: (1) Alt som begynder godt vil afslutte dårligt.
(2) Alt som begynder dårligt vil slutte endnu værre.

GUMPERSONs lov: Sandsynligheden for at noget skal ske,
er omvendt proportional med ønskeligheden.

HARVARDs lov, anvendt på datamater:

Selv under de strengest kontrollerede forhold med hensyn til
tryk, temperatur, volumen, fugtighed og andre variable,
så gør datamaskinen nøjagtigt, det den selv har lyst til.

CHEOPs lov: Ingenting bliver færdig inden for de fastsatte tids- & budgetrammer.

GILBs 4'de lov: Investeringer i pålidelighed vil øge, indtil de overstiger
den sandsynlige omkostning som fejlene ville have medført,
eller
indtil nogen insisterer på, at man skal gøre noget nyttigt arbejde.

Brug Oscilloskopet rigtigt !

Reprint / 'aktuel elektronik' nr. 21, 1987

Jørgen Jensen.

Oscilloskopet er det mest anvendte instrument i elektronikindustrien. Desværre bliver mange målinger - især i puls- og digital kredsløb - tit forkerte, fordi operatøren ikke tænker sig om.

I de senere år er prisen på et godt bredbåndscilloskop (DC til 30 MHz) faldet betydeligt. Samtidig er der kommet mange nye modeller på markedet. Faktisk er udbuddet nu blevet så stort, at brugeren kan vælge og vrage efter behag. Kvaliteten af oscilloskoperne er der ikke noget i vejen med - heller ikke de billigere modeller fra Japan, Taiwan og Korea.

Men nøjagtighed og reproducerbarhed af en måling ligger helt i hænderne på brugeren. Og her står det skidt til mange steder. De tekniske skoler gør ellers en stor indsats for at meddele nye teknikere et minimum af måleteknik samt forståelse for mulige fejlkilder. Alligevel begås der mange bommerter.

Flest målefejl opstår ved måling på pulser med hurtige stige / faldetider - i størrelsesordenen 1 til 5 nanosekunder. Fejlene skyldes, at brugeren glemmer at bruge prober, dvs. glemmer at kompensere for oscilloskopets indgangsimpedans (bestående af en modstand på 1 Mohm parallelt med en kapacitet på ca. 50 pF) samt kapaciteten i måleledningen. Hvis den er en skærmløsning af RG58-typen, er kapaciteten mellem inder- og yderleder 1 pF per cm. Ved en RG58-ledning, en meter lang, skal man derfor lægge 100 pF til de 50 pF i indgangen (se figur 1 hvor C2 er sat til 150 pF).

Kapaciteten i måleledning og oscilloskopindgang (uden probe) "afrunder" pulsen - mest på hurtige pulser - så skærbilledet slet ikke svarer til de faktiske forhold. Hvis oscilloskopets stigetid er for stor i forhold til pulsens stigetid, bliver pulsen også "afrundet". Så dur oscilloskopet ikke til den aktuelle måling, og det anbefales at finde et bedre. Af nød kan man dog godt få acceptable resultater ved at beregne stigetiden efter rms-metoden.*)

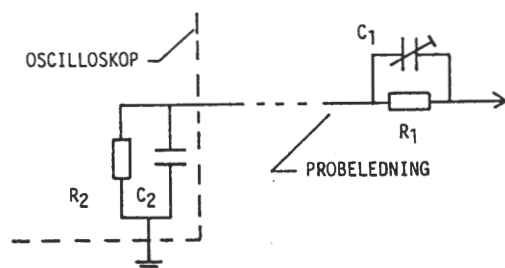
Forudsat at oscilloskopets stigetid er mindre end ca. 10 til 20 % af pulsens stigetid, udskiftes skærmløsningen mellem oscilloskop og puls med en probeledning, dvs. en ledning, der dæmper signalet i forholdet 1:10 og som kompenserer for kapaciteternes fejlvirkning. Probeledningen kalibreres ved at justere en trimmekondensator (C1) i probespids eller BNC-stik, indtil en standardkurve (fx kalibreringssignal indbygget i oscilloskopet) bliver gengivet korrekt (den kompen- serede firkantkurve på figur 2). Spændingsdelingen mellem måleobjekt og oscilloskopets indgang bliver frekvensafhængig og målingerne bliver korrekte.

Hvis probeledningen bruges sammen med et andet oscilloskop, kan en efterindstilling af trimmekondensatoren være nødvendig. Probeledninger er nu standard-tilbehør, når man køber et nyt oscilloskop. De kan også købes i "løs vægt" - og er altså uundværlige, når man vil måle rigtigt.

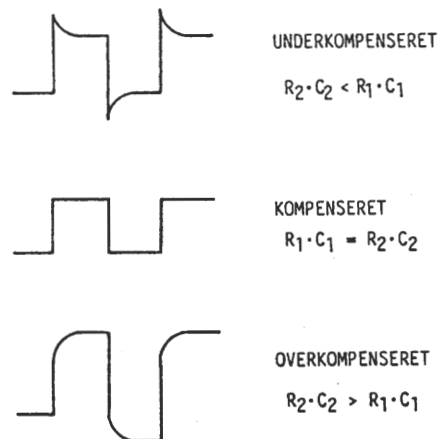
*) Root Mean Square = effektivværdi (eksempel: Veff.) vj.

fig. 2

fig. 1



$R_1 = 9 \text{ MOHM}$ $C_1 = 17 \text{ PF}$
 $R_2 = 1 \text{ MOHM}$ $C_2 = 150 \text{ PF}$
KOMPENSERING FOR $R_1 \cdot C_1 = R_2 \cdot C_2$



Trimmekondensatoren C1 i proben indstilles, så spændingsdelingen er kompenseret. Både ved den over- og den under-kompenserede spændingsdeling opstår der målefejl.

Probeledning og oscilloskop-indgang kan ækvivaleres med denne spændingsdeler. Frekvensafhængig signaltransmission opnås, når spændingsdeleren er kompenseret.

"Lidt om Grafik"

John B. Jacobsen

Denne artikel handler om grafik. For at kunne tegne linier og cirkler samt andre kurveformer er det nødvendigt med en rutine, der sætter en pixel på skærmen. Denne rutine kaldes et grafisk primitiv. Det er her benævnt WRITE_PIXEL og er beregnet til SHARP MZ-80B. Da hastigheden er altafgørende for grafik, er rutinen programmeret i assembler. Proceduren WRITE_PIXEL gemmes i en fil: WRPIXEL.PAS, der så includes. Derfor findes nogle programmer, der benytter WRITE_PIXEL til at tegne cirkler og linier. Hvis man selv har/programmerer det grafiske primitiv WRITE_PIXEL, kan man benytte disse hurtige grafiske programmer

TegnEnLinie, TegnEnCirkel, TegnEnAndenLinie

som følger. Procedureerne clear, taend og sluk er nogle hjælpeprocedurer til SHARP MZ-80B, som er medtaget for en ordens skyld. Af litteratur henvises til:

'Principles of Interactive Computer Graphics' William M.Newman/Robert F.Sproull

procedure clear; (* maskinafhængig procedure til at slette grafik ram 2 *)
begin

```
code
  $3E,$01,      (* LD A,$01      *) (* grafik ram 2 *)
  $D3,$F4,      (* OUT ($F4),A   *)
  $DB,$E8,      (* IN A,($E8)    *)
  $CB,$B7,      (* RES 6,A       *)
  $CB,$FF,      (* SET 7,A       *)
  $D3,$E8,      (* OUT ($E8),A   *)

  $21,$00,$E0,  (* LD HL,$E000   *)
  $36,$00,      (* LD (HL),$00   *)
  $11,$01,$E0,  (* LD DE,$E001   *)
  $01,$40,$1F,  (* LD BC,$1F40   *)
  $ED,$B0,      (* LDIR          *)

  $DB,$E8,      (* IN A,($E8)    *)
  $CB,$BF,      (* RES 7,A       *)
  $D3,$E8;      (* OUT ($E8),A   *)
```

end;

procedure taend; (* maskinafhængig procedure til at vise *)
begin (* grafik ram 2 *)

```
code
  $3E,$0D,      (* $02 grafik ram 1 *)
  $D3,$F4;      (* $0D grafik ram 2 *)
```

end;

procedure sluk; (* maskinafhængig procedure *)

```
begin
code
  $3E,$00,
  $D3,$F4;
```

end;

procedure Write_Pixel(x,y:integer);

procedure plot(x,y:integer); (* maskinafhængig procedure *)

begin

if (x>=0) and (x<=319) and (y>=0) and (y<=199) then

begin

```
code      (* beregn adresse *)
          (* HL:=40*y+E000 *)
          $2A, y, (* LD HL, (y) *)
          $29,   (* ADD HL, HL *)
          $29,   (* ADD HL, HL *)
          $29,   (* ADD HL, HL *)
          $54,   (* LD D, H *)
          $5D,   (* LD E, L *)
          $29,   (* ADD HL, HL *)
          $29,   (* ADD HL, HL *)
          $19,   (* ADD HL, DE *)
          $11, $00, $E0, (* LD DE, E000 *)
          $19,   (* ADD HL, DE *)

          (* HL:=x/8+HL *)
          $ED, $5B, x, (* LD DE, (x) *)
          $CB, $3A,   (* SRL D *)
          $CB, $1B,   (* RR E *)
          $CB, $3A,   (* SRL D *)
          $CB, $1B,   (* RR E *)
          $CB, $3A,   (* SRL D *)
          $CB, $1B,   (* RR E *)
          $19,       (* ADD HL, DE *)

          $3A, x,   (* LD A, (x) *)
          $E6, 7,   (* AND 7 *)
          $47,     (* LD B, A *)
          $04,     (* INC B *)
          $AF,     (* XOR A *)
          $37,     (* SCF *)
          $17,     (* IGEN: RLA *)
          $10, $FD, (* DJNZ IGEN *)

          $47,     (* LD B, A *)

          $DB, $E8, (* IN A, ($E8) *)
          $CB, $B7, (* RES 6, A *)
          $CB, $FF, (* SET 7, A *)
          $D3, $E8, (* OUT ($E8), A *)

          $7E,     (* LD A, (HL) *)
          $B0,     (* OR B *)
          $77,     (* LD (HL), A *)

          $DB, $E8, (* IN A, ($E8) *)
          $CB, $BF, (* RES 7, A *)
          $D3, $E8; (* OUT ($E8), A *)
```

end;

end;

begin

```
y:=200-y;
plot(x,y);
end;
```

```
Program TegnEnLinie;
```

```
VAR X1,X2,Y1,Y2: INTEGER;
```

```
      (*$I B:WRPIXEL *)
```

```
Procedure DDA(x1,y1,x2,y2: integer);
```

```
var
```

```
    length, i: integer;
```

```
    x, y, dx, dy: real;
```

```
begin
```

```
    length:=abs(x2-x1);
```

```
    if abs(y2-y1)>length then length:=abs(y2-y1);
```

```
    dx:=(x2-x1)/length;
```

```
    dy:=(y2-y1)/length;
```

```
    x:=x1+0.5;y:=y1+0.5;
```

```
    for i:=1 to length do
```

```
        begin
```

```
            write_pixel(trunc(x),trunc(y));
```

```
            x:=x+dx;
```

```
            y:=y+dy;
```

```
        end;
```

```
end;
```

```
BEGIN
```

```
    CLEAR;
```

```
    write('Indtast X1 :'); readln(X1);
```

```
    write('Indtast Y1 :'); readln(Y1);
```

```
    write('Indtast X2 :'); readln(X2);
```

```
    write('Indtast Y2 :'); readln(Y2);
```

```
    write(clrhom);
```

```
    DDA(x1,y1,x2,y2);
```

```
END.
```

```
Program TegnEnCirkel; (* Af John B. Jacobsen 22/8 1985 *)
```

```
Var
```

```
    x,y,d : Integer;
```

```
    Radius : Integer;
```

```
      (*$I B:WRPIXEL *)
```

```
Procedure CirclePoints(x,y: Integer);
```

```
Begin
```

```
    Write_Pixel(x,y);
```

```
    Write_Pixel(y,x);
```

```
    Write_Pixel(y,-x);
```

```
    Write_Pixel(x,-y);
```

```
    Write_Pixel(-x,-y);
```

```
    Write_Pixel(-y,-x);
```

```
    Write_Pixel(-y,x);
```

```
    Write_Pixel(-x,y);
```

```
End;
```

```
Begin
```

```
    Clear;
```

```
    Write(ClrHom, 'Hvilken radius skal cirklen have ? '); Readln(Radius);
```

```
    x:=0;
```

```
    y:=Radius;
```

```
    d:=3-2*Radius;
```

```

While x<y Do
  Begin
    CirclePoints(x,y);
    If d<0 Then
      d:=d+4*x+6
    Else
      Begin
        d:=d+4*(x-y)+10;
        y:=y-1
      End;
      x:=x+1
    End;
  If x=y Then CirclePoints(x,y);
End.

Program TegnEnAndenLinie;

VAR X1,X2,Y1,Y2: INTEGER;
    LENGTH      : INTEGER;
    SW: BOOLEAN;

                                (*$I B:WRPIXEL *)

Procedure Bresenham(deltax,deltay,x,y: integer);

var
  e, i: integer;

begin
  e:=2*deltay-deltax;
  for i:=1 to deltax do
    begin
      IF not SW THEN write_pixel(x,y) ELSE WRITE_PIXEL(Y,X);
      if e>0 then
        begin
          y:=y+1;
          e:=e+(2*deltay-2*deltax)
        end
      else
        e:=e+2*deltay;
        x:=x+1;
      end;
    end;
end;

BEGIN
  CLEAR; write(clrhom);
  X1:=0;
  Y1:=0;
  X2:=319;
  Y2:=199;
  LENGTH:=X2-X1;
  SW:=(Y2-Y1)>LENGTH;
  IF SW THEN
    BEGIN
      LENGTH:=Y2-Y1;
      BRESENHAM(Y2-Y1,X2-X1,Y1,X1)
    END
  ELSE BRESENHAM(X2-X1,Y2-Y1,X1,Y1)
END.

```

////////////////////////////////////// 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 -

5 - 3.5" 3M - SSDD a 37 -

10 - 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 50 -

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 //

MODULA-2 / En grundlæggende indføring. (dansk) 288 sider, 225 kr. af Per Amdal Steffensen (medforf. af "Pascal / En grundlæggende indføring i programmerings-sproget Pascal" (PPAS)). Forlaget systime a/s, Klokkebakken 20, Gjellerup, 7400 Herning. 07-11 90 11. ISBN 87-7351-533-7. En diskette med programmer kan købes for 950 kr.

MODULA-2 oversættere findes nu i en række forskellige Z80/ CP/M-80 versioner, bl.a. tilbyder et større, engelsk software firma således følgende:
a) FTL MODULA-2 koster £45 b) Hochstrasser Modula-2 koster £100
c) Turbo Modula-2 - £55 (m. 500 sider manual)
de anførte priser er incl. manual(er) og forsendelse til DK, der betales ikke VAT, men 22% MOMS ved modtagelsen. Levering ca. 2-3 uger.
NB. Turbo Modula-2 kører på alle Z80 ZCPR3, Z-System og CP/M-compatible micro-computers. En special-version kører på Micromint's (BYTE/Steve Garcia) HD64180-baserede SB180 og SB180FX single board computers.

Grey Matter
4 Prigg Meadow, Ashburton
Devon TQ13 7DF / England

Telf. (0364) 53 499

FARSIGHT - et 'Lotus 1-2-3' kompatibelt program, annonceres skrevet i Modula-2.

'GE Solid State' er det nye fællesnavn for GE, RCA og Intersil halvledere.

DALLAS Semiconductor. Nyt stort 1987/88 produktkatalog. (06) 81 65 22 Micronor.

C-128-folk som vil bruge Turbo Modula-2, bør anvende CP/M+ ver. 6DEC ell. 8DEC.

AARHUS RADIOLAGER A/S har udsendt nyt 900-sider katalog fra RS Components Ltd.

Denne artikel er rettet til de maskinkodeinteresserede. Den viser nogle små fif, som måske ikke alle kender. Først de enkle ting:

```
XOR A      nulstiller A registret og sætter Z flaget;
            kun en byte og 4 T states, kortere og hurtigere end LD A,0
OR  A      sætter Z flaget i henhold til indholdet i akkumulatoren, resetter
            carry.
```

Hvis man skal fylde et område i hukommelsen med en konstant værdi, kan dette gøres med følgende Z80 instruktioner :

```
LD  A,N      ; N betegner konstanten
LD  HL,AREA  ; pointer til start på området, der skal fyldes
LD  DE,AREA+1
LD  BC,COUNT ; BC indeholder antal bytes
LD  (HL),A
LDIR
```

Dette kan gøres hurtigere ved at push'e værdien på stakken. Det vil sige, at man flytter stakpointeren til slutningen af det område, der skal fyldes, derefter pushes f.eks. HL (som man har loadet med konstanten) et passende antal gange. Dette gøres ved hjælp af en tæller og eventuelt en subroutine. Herunder et eksempel, der formaterer min ramdisk. Ialt en halv megabyte bliver fyldt op med værdien OE5H. Dette går cirka 3 gange hurtigere end med en LDIR instruktion. REPT 256 er en instruktion til M80 assembleren om at gentage, hvad der står mellem REPT og ENDM 256 gange. I en løkke, der gentages 4 gange, pushes HL 256 gange, dvs. ialt 2048 bytes bliver fyldt op med OE5H. Dette forløb gentages 256 gange; en gang per side i hukommelsen.

; FLUZ VERSION 2

John Jacobsen

```
.Z80
START: LD      HL,OE5E5H
; mapin
      IN      A,(OE8H)
      OR      OCOH
      OUT     (OE8H),A
      LD      A,11H
      OUT     (OF4H),A
      LD      B,0
; page select
IGEN: LD      A,B
      OUT     (OA4H),A
      LD      C,4      ; counter - 4 gange
      LD      SP,6800H
REPT4: REPT    256      ; 256 gange
      PUSH   HL        ; 2 byte = 2048
      ENDM
      DEC    C
      JP     NZ,REPT4
      DEC    B
      JP     NZ,IGEN
; mapout
      XOR    A
      OUT     (OF4H),A
      IN     A,(OE8H)
      AND    3FH
      OUT     (OE8H),A
      LD     C,13
      CALL   5
      JP     0
      END
```

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

EFTERLYSNING! Er der nogen, der ligger inde med viden/dokumentation om hvordan MAIL-MERGE skal håndteres under WORDSTAR?

(746) Ebbe Sønderhousen. Smedegade 3 B, 5900 Rudkøbing. (09) 51 40 02

Gemini computer bestående af:

Videokort GM 812, CPU-kort GM 813, Disccontroller GM 829, 2 stk. Shugart 800 K diskdrev, strømforsyning AC 9231 switch mode, 19" rack GM 610 med 8 slots motherboard, James tastatur og Taxan monitor.

En del disketter med programmer. Meget fyldig dokumentation samt tegninger til alle kort.

Sælges samlet for 5500 kr. eller højeste bud.

Henvendelse: Bent Kristensen, Spøttrupvej 10, 7400 Herning.

Telf. priv. 07 26 73 80 ell. arb. 07125588.

Til SHARP MZ-80B sælges et terminal-emuleringsprogram. Det er specielt udviklet til SHARP MZ-80B og kan ikke køre på andre CP/M maskiner, desværre. Det er muligt, at programmet kan benyttes på MZ-800 familien, men dette vides ikke med sikkerhed i skrivende stund. Skulle en MZ-800 bruger være interesseret bedes han/hun kontakte undertegnede.

Programmet benytter SHARP's standard RS 232C interface, som man altså skal have for at få glæde af det. Af faciliteter kan nævnes :

- 1) Mulighed for at vælge mellem 5,6,7 og 8 bit per tegn.
- 2) Mulighed for kommunikere med eller uden paritetsbit.
- 3) Mulighed for at sætte paritetsbitten til lige eller ulige.
- 4) Mulighed for at vælge mellem 1, 1 1/2 samt 2 stopbits.
- 5) Mulighed for vælge mellem kanal A og B i Z80 SIO'en.
- 6) Programmet benytter interrupt, hvilket giver nogle fordele, som det vil føre for vidt at drøfte her.
- 7) Programmet sender ved tryk på <CR> hele den linie, cursoren er placeret på.
- 8) WORDSTAR lignende editeringsmuligheder i terminalmode.
- 9) Menu-orienteret med hjælpemenuer, som ikke skal hentes fra diskette, men bliver overført direkte til skærmområdet, dvs. uden om BDOS. Dette skal ses, det går lynende hurtigt.
- 10) Forskellige andre options.

Prisen for alle disse herligheder: 250 kr.

Henvendelse til:

John B. Jacobsen / Hans Olriks Vej 8, 3 tv / 2450 SV / Telf. 01 - 22 75 73

ENTERPRISE 64/128 - Er der nogen af Z80-NYT's læsere der har/kender nogen, der har/bruger denne computer, ja, så ring til mig snarest.

Leif Pedersen / 08-14 38 62

//////////////////////////////////// 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.

BENCHMARKS

Reprint / 'Siemens Components XXII (1987)'

Benchmarks er programmer beregnet til at bedømme ydelser af forskellige computere, microprocessorer ell. microcontrollere. Der bruges en blanding af de, fra assembler-områder, hyppigst anvendte instruktioner.

Når der bruges et høj-niveau sprog så udgøres benchmark af ret korte programmer med forskellige formål for at dække et bredt anvendelsesområde.

Eksempler på benchmark programmer er sortering, beregning af primtal, rødder og matricer, floating point og streng-operationer.

Der findes ingen 'industri standard' for disse bedømmelses-programmer.

Uanset hvor vigtig sammenligning af computere kan være, så er det et vanskeligt område af forskellige grunde. Det er næppe muligt at beskrive en computers kompleksitet med et enkelt tal, heller ikke forskel mellem biler kan udtrykkes på denne måde. Faktisk er det kun muligt at sammenligne ydelser mellem computere, der har samme arkitektur, som f.eks. SAB 8088 med SAB 8086 ell. SAB 80286'eren.

Herudover synes kun anvendelses-orienterede benchmarks at være rimeligt egnede. Forskellige anvendelser kan betyde helt andre regne-egenskaber, f.eks. vil et program til at kontrollere et valseværk stille krav, der er helt anderledes end til et grafik- eller bogholderiprogram.

Sammenligning af computere ved brug af benchmark programmer der er skrevet i et højniveau-sprog, inkluderer sammenligning af compiler-effektivitet.

Det kan også være fornuftigt at prøve ens egne programmer på forskellige computere for at finde ud af hvilket system, der er det hurtigste. Resultatet heraf gælder naturligvis kun for disse programmer.

Benchmarks drejer sig kun om 'behandlings-hastighed'/transfer rates og det er blot et af de mange kriterier, som kan være af betydning ved bedømmelsen af computere (eller compilere).

Benchmarks giver ingen information om

- nøjagtighed af real tal,
- nødv. lagerstør. for oversatte programmer,
- compiler-kapacitet og -hastighed,
- compiler-overensstemmelse med standard(er),
- adgang til operativsystem (numerisk co-processor, grafik-egenskaber, lagerstørrelse, etc.
- hjælpeprogrammer, f.eks. editor, linker, debugger, bibliotek osv.
- udgifter, teknisk assistance, udvidelser.

Her en kort forklaring på nogle udtryk som bruges i forbindelse med benchmarks:

Keywords

cpi (clock cycles per instruction)

Det gennemsnitlige antal clock cycles per instruction. Denne enhed har stor betydning ved bedømmelsen af RISC computere (RISC = Reduced Instruction Set Computer).

Dhrystone

Syntetisk program til at sammenligne forskellige computere. R.P. Weicker(2) har udviklet dette program og beskrevet det i 1984.

MFLOPS (Million Floating Point Operations per Second)

Program der tæller hvor mange millioner FP der udføres pr. sekund.

MIPS (Million Instructions Per Second)

Program der tæller hvor mange millioner instruktioner der udføres pr. sekund. Kombination af (statistisk fundne) mest anvendte instruktioner. Mangel på standardisering gør denne metode mindre anvendelig.

Whetstone

Syntetisk program til at sammenligne forskellige computere. Udviklet af H.J. Curnow og B.A. Wichmann (3) sidst i 1960'erne i UK National Physical Laboratory of Whetstone, England. Selvom progr. er baseret på ALGOL bruges det meget i FORTRAN. Der findes både single accuracy(32-bit) og double accuracy(64-bit) udgaver.

Referencer

- 1) Omir Serlin: MIPS, Dhrystone and other Tales. Datamation USA, Vol.32, No.11, June 1986, pp.112 to 118.
- 2) Weicker, R.P.: Dhrystone: A Syntetic System Programming Benchmark. CACM, October 1984, page 1013.
- 3) Curnow, H.J.; Wichman, B.A.: A Syntetic Benchmark. Computer Journal, February 1976, page 43.
- 4) Webster, B.: Benchmarking. Byte, January 1986, pp.371 to 379.

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

FARVEBÅND

Frank Damgaard

Her følger et tip om, hvordan man kan gøre "grå" farvebånd sorte igen.

Det er dog ikke alle typer farvebånd, der kan genfarves. Engangsfarvebånd kan normalt ikke farves, og ligeledes kan der være problemer med visse typer kassetter til farvebånd. Vær iøvrigt opmærksom på at visse farvebånd kun benyttes på den ene halvdel af farvebåndet, så her kan der allerede opnås en mere "sort" udskrift ved at vende farvebåndet i kassetten.

Først skal man finde en brugbar sort farve. Efter en del forsøg er jeg kommet frem til, at sort stempelfarve for gummistempler kan benyttes. Pas på ikke at anskaffe stempelfarve til maskinstempler, idet disse kan indeholde olie og/eller kemikalier, der kan opløse dele af farvebånd eller farvebåndskassetten. Selv benytter jeg en stempelfarve på vandbasis fra "Kores" (30 ml til 15-16 kr.), men jeg antager, at andre tilsvarende produkter også vil kunne bruges.

Dernæst skal farven overføres til farvebåndet. Dette kan være besværligt. Mange kassetter indeholder et uendeligt farvebånd med en lille tap, der trækker båndet rundt. Her kan man opspænde kassetten i en skruestik, og enten med håndkraft eller med en lille boremaskine trække farvebåndet frem. Stempelfarven kan så overføres til farvebåndet, ved at farven dryppes på, eller ved at benytte en lille gummirulle fra en kasseret båndoptager, hvor rullen holdes fugtig med farve samtidig med at farvebåndet føres forbi/rundt om rullen. Pas på, at farvebåndet ikke "drukner" i farve, idet det kan give et noget udflydende resultat på specielt den første side, der udskrives.

Det har vist sig, at de fleste farvebånd kan genfarves mindst 5-10 gange før selve farvebåndet er så slidt, at det ikke kan benyttes mere.

Til slut vil jeg gøre opmærksom på, at der kan benyttes andre farver end sort til at farve farvebånd med. Vil man f.eks. have et rødt farvebånd, benyttes rød stempelfarve og et meget lysegråt farvebånd.

Single Density ?

Double Density ?

Vi kender alle en diskette, sådan da, et firkantet, sort kartonhylster, stort rundt hul i midten, lille rundt hul ved siden af og et hak i den ene side samt aflange åbninger til læse- skrivehoveder og et par små hak til fixering i den kant, som først føres ind i disk. drevet.

Som eksempel for denne artikel er valgt en 'alm.' DS (Dual Side) 5.25" disk. dvs. en diskette, hvor begge sider kan anvendes, men det er principielt uden betydning, der kunne lige så godt have været anvendt en 'Single Side' diskette, med kun een brugbar side (side 0).

På eksempeldisken kan der være 40 eller 80 spor, idet der helt ses bort fra 'eksotiske' disketter/drev med f.eks. 28, 35, 50 spor el. lign., heller ikke sådanne, som med 'diametere' varierende omdrejntal (Sirius), sneglegang (Worm Drives) eller typer hvor visse af bit'ene er inverterede (Sharp, Superbrain).

A.h.t. overskuelighed vil kun et enkelt spor blive gennemgået her, men principet gælder for hele disken uanset antal spor.

Lidt baggrund:

I et vandret monteret diskdrev med en diskette rigtigt anbragt (etiketten opad) kaldes den underste side normalt for side 0, og den øverste side 1.

Under fremstillingen af de første diskdrev / disketter anvendtes en skive, hvor kun den ene side var forsynet med et egnet magnetiserbart lag, (coating), dette vendte nedad mod læse- skrivehovederne og på oversiden behøvede man da kun en lille fjeder-arm med en filtpude til at trykke skiven ned mod læse- skrivespalten nedenunder så der kunne blive magnetisk nærkontakt.

Problemet med sidenumre var ikke aktuel, men det kom - naturligvis- da det blev muligt at fremstille 'Dual Side' disketter, så var der 0-1 / 1-2 diskussioner. 0-1 vandt...

Disken drejer 'med uret' når man ser mod side 1 (etikettesiden).



Sporene på disken er egentlig uendeligt lange (cirkulære og koncentriske), men for alligevel at få en begyndelse og en ende bruges et Index-hul, et lille, ca. 2 mm rundt hul igennem disken, til at markere et start- og slutpunkt. Ofte anvendes en lysdiode + fotodiode til at give et signal for hver omdrejning.

Alle spor på disken er 'lige lange', de yderste såvel som de inderste... når man måler ved det antal bytes, der normalt kan anbringes på sporet.

TPI eller tpi = tracks per inch, dvs. antal spor pr. tomme (1" = 25,4 mm). Dette betyder dog ikke at der er f.eks. 48 eller 96 spor på disken, men det er et mål for, hvor tæt sporene kan/må være på hinanden uden at give anledning til fejl. Spor-arealet er altså lidt mindre end 25,4 mm.

Med betegnelserne SD = enkelt 'tæthed' og DD = dobbelt 'tæthed' menes at der kan anvendes (eller er anvendt) en af disse skrivemetoder, evt. begge former. Skal kapaciteten på en standard-diskette bedømmes, er det nødvendigt at kende hvilken skrivemåde der anvendes, f.eks. Single Density el. Double Density eller måske noget helt tredje.

Kan der skrives med Single Density på en diskette, så kan der på samme diskette også skrives med Double Density, forudsat naturligvis at apparaturet kan skrive på denne måde, på disketterne er der ingen fysiske eller mekaniske forskelle

...der er alene tale om at arrangere bit'ene efter andre regler end ved SD...

Floppy Disc Controller'en og computeren skal ganske vist arbejde lidt mere og hurtigere, men det er ikke noget problem. (måske u765 undtaget).

Formattering:

Inden disketten overhovedet kan bruges skal den formatteres, dvs. der skal dannes et mønster de steder på overfladen / (hm, fladen), hvor sporene skal være, disse førstegangs-spor indeholder oplysning om bl.a. spor-nummer, side 0 eller side 1), sektor-nummer og -størrelse osv. altsammen nødvendige oplysninger til brug ved en efterfølgende skrivning / læsning.

En alm. 5.25" diskette drejer 300 omdr. pr. minut. Dvs. 1 omdr. på 0,2 sekund det er = 200 000 mikrosekunder (uS), og iflg. 'Shugart' standard er den

korteste tid mellem 2 bit = 4 uS, hvilket 'svarer til' en frekvens på 250 kHz,
 længste - - - 2 - = 8 uS, - - - - - 125 kHz.

Ved en byte forstås her 8 på hinanden følgende bit(s) og med 200 000 uS til rådighed skulle der kunne være:

DD: 200 000 / 4 = 50 000 bits, 50 000 / 8 = 6250 bytes.
 SD: 200 000 / 8 = 25 000 bits, 25 000 / 8 = 3125 bytes.

Single Density:

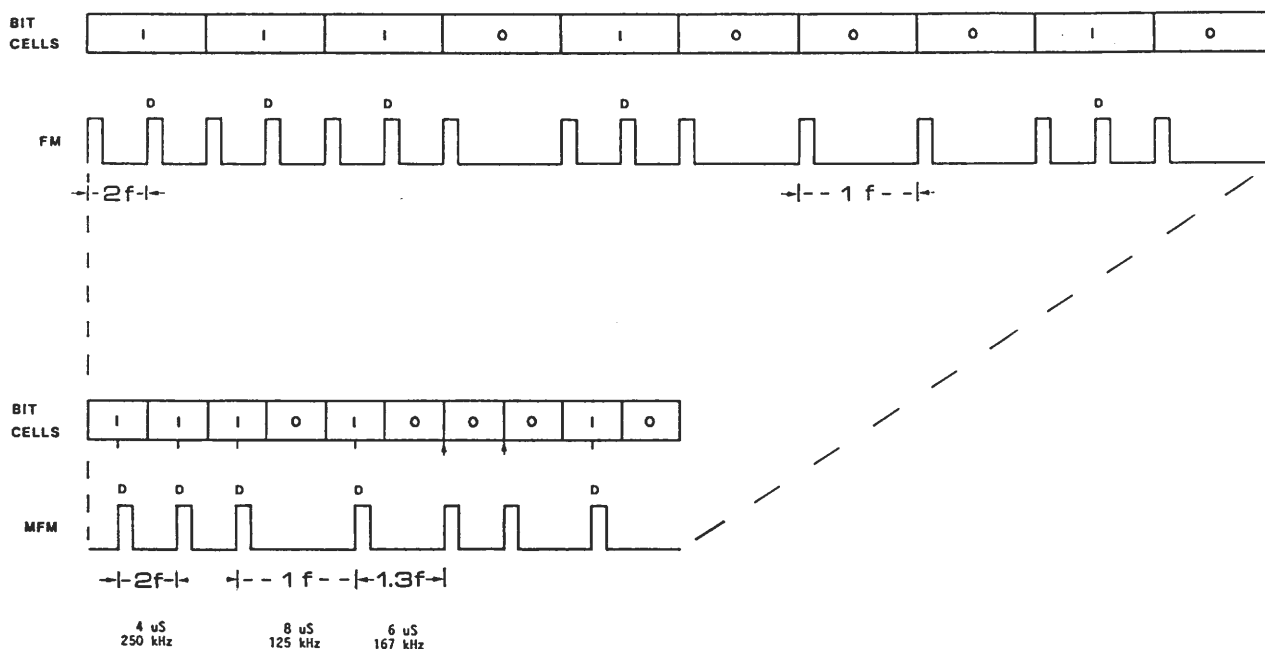
Man tænke sig sporet, som et langt bånd på hvilket L-S-H (Læse-Skrive-Hovedet) kan anbringe 'magnetiske mærker' / bits, de inddeles her i 2 grupper, nemlig

C = (Clock Bit) og D = (Data Bit).

Alle bit'ene er af samme slags, det er alene deres placering, tilstedeværelse eller fravær, som har betydning.

Sporet begynder ved Index signalets første flanke, og disk.drevet betragter hele sporet, som een lang række af 'bit celler' hvori der kan være skrevet en bit. Se fig. For bedre at kunne 'holde takten' undervejs (der er jo dog ca. 25 000 bit-celler, der skal holdes øje med), bliver der ved SD-formatteringen anbragt 1 Clock bit for hver 8 uSekunder.

Hvis der skrives en bit midt i en celle betyder det "1", ellers er det et "0" dvs. bits med afstand på 8 uS imellem = 0 og bits med afstand på 4+4 uS = 1



Disse ændringer kan registreres, som et frekvensskift mellem (i dette tilfælde) en grundfrekvens (1f) og en dobbelt så høj frekvens (2f).*)

Det kaldes Frekvens Modulation (FM).

Samme teknik anvendes også indenfor andre felter. Bl.a. i forbindelse med visse typer modemudstyr, som bruger et offentligt telefonnet, her kan modem'et skifte imellem 2 eller flere toner. Frequency Shift Keying (FSK).

Nu nærmer vi os Double Density, men først et lille sidespring:

På nogle telex-systemer brugte / bruger man en såkaldt 5-bit kode, med den kan opnås 32 tegn, det er dog lige lidt nok, alfabetet kan man få, men nogle tal og andre tegn kan også være nødvendige. Det kan f.eks. klares ved at sende en af de koder, der ikke bruges, den skal sige til dem i den anden ende af ledningen: så kære venner nu betyder signalerne, der følger herefter, ikke bogstaver, men tal og tegn; og når skifte-koden kommer igen næste gang er signalerne atter bogstaver osv.

Double Density:

Da man i nogen tid havde brugt Single Density, været imponeret og vænnet sig til det, kom tanken om, at det ville være rart, hvis der kunne lagres flere informationer, helst på de samme disketter og med samme (kostbare) udstyr/hardware, der allerede var anskaffet.

Nogen fandt ud af, at hvis bit'ene ikke skrives tættere på hinanden end ved SD, (2f), samt at hver bit nu betyder 1 og en 'manglende' bit (1f) betyder 0, så kunne der på samme areal være dobbelt så mange bytes, som ved single density.

Der var bare et lille men, hvis der nu blev sendt en lang række nuller, hvordan kunne man da være sikker på, at man havde 'holdt takten' undervejs og ikke 'var faldet ud af trit' når der igen kom 1'ere? Joh, her ligesom i telex, blev der indført et 'skiftesignal'. Når systemet ser, at der ('forude') kommer nogle 'sammenhængende' nuller, vil der blive anbragt taktholdere (clock bits) imellem nullerne efter følgende regler:

- ingen clock bit i den aktuelle bit celle, hvis enten den foregående bit celle eller den aktuelle bit celle indeholder en data bit,
- data bits skrives i midten af bit celler og
- clock bits skrives i starten af bit celler.

Dvs. blot ved at ændre en bits position i bit cellen kan der skiftes værdisæt.

Det kaldes Modified Frequency Modulation (MFM) - sådan er det.

Viggo Jørgensen

*) - ved en anden, såkaldt digital teknik anvendes 'vinduer' eller 'tid-rammer' til at afgøre om en bit er i midten eller ved begyndelsen af en bitcelle.

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

CMOS MPU'en Z280 er her nu! ...uddrag fra 'New From Zilog' (forts. fra nr. 2)

USER MODE / SYSTEM MODE

En effektiv, dobbelt arbejdsmåde, som medvirker til et systems integritet, idet når CPU'en er i system mode, tillades kun visse instruktioner og der er kun adgang til visse registre. Der kan f.eks. anvendes Z80 application software på bruger niveau, selvom der på system niveau arbejdes med ny 'overordnet' software med multi-task eller multi-user funktioner.

ON BOARD CLOCK

Den integrerede clock oscillator kan forbindes direkte til et krystal eller til et passende ydre clock signal med 2 x den ønskede processor hastighed. Tidlige versioner af Z280 arbejder ved 10 MHz (20 MHz XTAL); andre versioner, som kan arbejde ved op til 25 MHz (50 MHz XTAL) er undervejs.

DRAM REFRESH CONTROL

En 10-bit refresh adresse generator til dynamisk memory er inkluderet (on-chip) og 'refresh rate' er helt programmerbar via software.

2 BUS MODES

Z280 har 2 bus moder, som kan vælges ved power-on med pin 34 (OPT). I Z80 mode bliver 8 data linier multiplexed med 24-bit adresse bussen.

Ved de-multiplexing, hvortil der bruges en enkelt 'address latch package', bliver resultatet et interface, som er helt kompatibelt med Z80 familiens periferikredse, dvs. identiske strobe- og timing forhold til Z80.

I Z-BUS mode bliver 16 data bus linier multiplexed med adresse bussen, og alle control strobes svarer til Zilog's Z-BUS specifikation, det giver en væsentlig lettelse for konstruktøren ved brug af højeffektive periferikredse, som f.eks. SCC'en (Serial Communications Controller). I denne mode kan man endvidere bruge den såkaldte 'burst mode' ved læsning fra dynamiske RAM'er, det vil sige at ved hver access/adressering kan man læse 4 ord, samt at access-tiden faktisk bliver halveret ved læseoperationer. Det er i særdeleshed effektivt når man bruger...

CACHE MEMORY

Z280's on-board værende cache udgøres af 256 high speed registre, hvis man ikke vil anvende cache funktionen kan de anvendes som lokal memory. Når cache'en er aktiveret kan den bruges til cache-instruktioner, data eller begge dele, herved kan koder eksekveres meget hurtigt direkte fra cachen (efter 'opladning'!)

MULTI-MICRO CAPABILITIES

Nogle spec. egenskaber ved konstruktionen af Z280 gør den let at anvende, enten med en slave-processor (f.eks. DMA), eller som del af et multi-CPU design med global memory. Når f.eks. Z280 er i multi-processor mode vil den automatisk skifte til den globale bus (arbitrate), når en ikke-lokal adresse dannes, herved bliver system opbygning mindre kompliceret.

DMA CONTROLLERS

4 DMA kanaler er 'indbygget'. De to af dem understøtter 'fly-by' transactions. Hver kanal bruger 24-bit source og destination adressering med op til 64K data overført pr. operation, mere hvis flere DMA'er kædes sammen. NB. DMA-adresser bliver ikke fortolket af memory management unit.

COUNTER/TIMERS

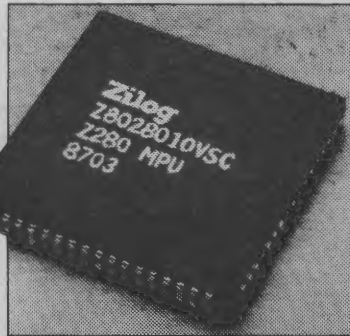
Der er 3 alsidige 16-bit counter/timers til rådighed, hver med 2 multi-funktion programmerbare I/O lines. To kanaler kan kombineres til en 32-bit tæller.

NYHED

Zilog

Lager

Z280 8/16 Bits CMOS MPU



- Z80 code compatible.
- Eight-bit Z80 Bus or 16 Z-Bus controls.
- Three 16-bit counter/ timers.
- Programmable DRAM Refresh controller.
- Three stage instruction pipeline.
- ON-chip full-duplex UART.
- On-chip 256 byte fully associative cache.

- Four on-chip DMA channels supporting flow-through, flyby, and linked operations.
- On-chip MMU supports up to 16M bytes of memory and 64K I/O addressing.

Rekvirer datablad og yderligere information!

DITZ SCHWEITZER



0245 3044

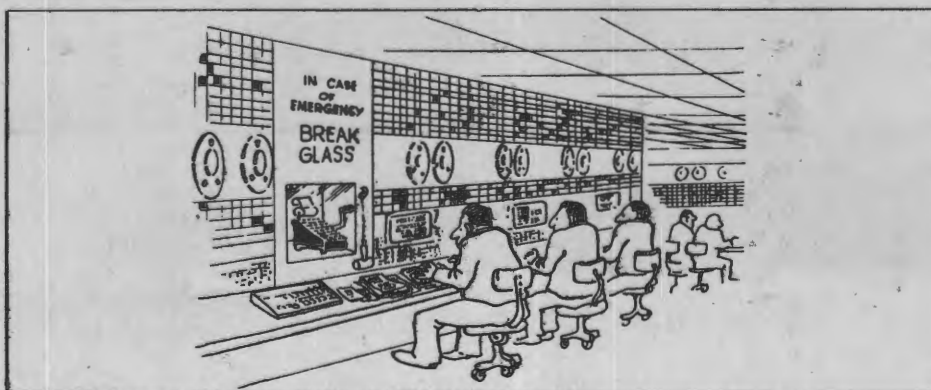
Vallensbækvej 41 - DK 2605 Brøndby

Telefax 02 45 92 06

////////////////////////////////////
Z80 ? eller Z280 ?

; Med denne lille rutine kan et program 'se efter' om det bruges på en Z80 CPU
; eller på en Z280 processor. Instruktionerne ser efter en forskel i resulta-
; tet, når CB37h maskinkoden udføres på Z80 CPU og Z280 MPU, for at tillade et
; program at afgøre hvilken processor det bliver udført på. Instruksjonen sæt-
; ter S flaget på Z80 CPU og stryger (clears) det på Z280 MPU. Rutinen bruger
; A- og F-registrene.

```
LD      A,40h      ; Initialiser operanden.
DEFB    OCBh      ; Denne instruktion vil sætte S flaget på
                ; Z80 CPU og 'stryge' det på Z280 MPU'en.
JP      M,Z80     ; undersøg flaget og hop.
    eller
JP      P,Z280
```



14. novbr. 1987.

Ifølge beslutningen på den ordinære generalforsamling den 14. maj, indkaldes til ekstraordinær generalforsamling med det ene punkt, at forelægge det reviderede regnskab.

Z80 Brugergruppen
Dagsorden for ekstraordinær generalforsamling

Torsdag den 3 december kl. 19:00

Datalogisk Institut
Lokale N-037
Universitetsparken 1
2100 København Ø.

- 1) Valg af dirigent.
 - 2) Valg af referent.
 - 3) Forelæggelse af reviderede regnskab for 1986.
 - 4) Eventuelt.
-

Da den ordinære generalforsamling ikke kunne nå at blive færdig inden kl. 22:00 den 14. maj, indkaldes til ekstraordinær generalforsamling.

Z80 Brugergruppens Indkøbsforening
Dagsorden for ekstraordinær generalforsamling

Torsdag den 3 december

(Samme sted som ovenstående)

- 1) Valg af dirigent.
- 2) Valg af referent.
- 3) Forelæggelse af reviderede regnskab for 1986.
- 4) Afgørelse om Indkøbsforeningens fremtid?
(Skal den afvikles ? Eller kan den få andre opgaver ?)
- 5) Valg af bestyrelse samt 2 suppleanter.
(Skal evt. kun vælges til at afvikle indkøbsforeningen.)
- 6) Valg af revisor samt suppleant til 1987 regnskabet.
- 7) Eventuelt.

Varelager, samt oversigt over Z80 Brugergruppens aktiver.
(pr. 31 december 1986)

Diverse aktiver (nedskrevet).

1	STAR printer (bred) med nyt (forkert) matrix-hoved.
1	MX-80 do. (står hos Ole Hasselbalch)
1	Typehjulsprinter - - Asbjørn Lind
1	GEMINI (købt f. 15.000kr.), 1/2 i 1985, 1/2 i 1986. (Programbibliotek, hos Asbjørn Lind)
1	5.25" 100 TPI disk.drev (hos Rene Hansen)
1	3.5" disk.drev lånt af indkøbsforening til programbiblioteket.
1	XEBEC disk.kontroller (E.Palsbo)

Diverse varer til salg:

7	CP/M mapper (sælges til 700,00 kr.)
20	JRT Pascal manual (- - 2000,00 -)
3	Jesper Skavin bog (- - 450,00 -)
1	kasse data-bånd!

Noter til regnskab:

- Annonceindtægter inkluderer også salg af Z80-NYT.
- Lagerforbrug, bl.a. programbiblioteket, CP/M mapper.
- Ekstraordinære udgifter:
er betaling for anden halvdel af Gemini-computer (7.500 kr.),
første halvdel blev betalt i 1985.
Desuden køb af 1 stk. XEBEC disk.kontroller, som skulle bruges til
database-projektet. (4.900 kr.)

Frank Damgaard

Z80 Brugergruppen. Årsregnskab 1986

1986, Kontingent	2132,00	
Overf. fra tidl. år	44.700,00	
Indmeldelsesgebyr	800,00	
Kontingent/Indmeldelse	>	47.632,00

CP/M Bibl. Indtægter	10.198,56	
CP/M Bibl. Udgifter	-6.209,46	
Lagerforbrug	-3.771,88	
Netto CP/M Bibl.	>	217,22

Renteindtægter	466,75	
Div. indtægter ialt	>	466,75

Indtægter ialt for 1986 48.315,97

Annonce-indtægter	-48,00	
Z80-NYT produktion	15.264,50	
Redaktør-honorar	1.000,00	
Netto Z80-NYT	>	16.216,50

Sekretær/Forretningsfører	6.900,00	
Porto	1.423,77	
Ekstraordinære udg.	12.400,00	
Renteudgifter	230,47	
Div. udgifter ialt	>	20.954,24

Udgifter ialt for 1986 37.170,74

Netto-Resultat 1986 : 11.145,23

Bankindestående ialt	691,67	
Giro 6-74 26 02	1.064,42	
Kassebeholdning	52,50	
Likvide-beholdninger	>	1.808,59

Z80 Brug.Indkøbsforen.	4.093,43	
Debitorer ialt	>	4.093,43

Værdipapirer	2.600,00	
Varelager	3.150,00	
>		5.750,00

Aktiver ialt 11.652,02

Egenkapital primo	796,79	
Overført resultat	11.145,23	
Kursregulering Aktie	-890,00	
Egenkapital ultimo	>	11.052,02

Forudbetalt kontingent > 600,00

Passiver ialt 11.652,02

(se noter)

Frank Damgaard