NASCOM Z80 NYT

NR: 2
2. årgang

NASCOM BRUGERGRUPPE
Sidevolden 23
2730 Herlev
Giro 674 26 02

Februar 1981

Som meddelt i sidste nr. af NN afholder vi den første søndag i hver måned medlemsmøde på pædagogisk Central, Rustenborgvej 1, Lyngby kl. 12.30. Søndag den 1.2 afholdt vi møde med emnet Nascoms tastatur og Polydatas nye Pascal. Dette nye sprog vil blive anmeldt i næste nummer af NN af Jesper Skavin.

Cirkeltegningsprogrammet fra NN 1 havde en fejl i linie 340, hvor GOTO skal efterfølges af 380. Det var den eneste fejl i det nummer. Jeg håber for alle at dette og fremtidige numre bliver fejlfri ! (Et stort ønske, men måske ikke uladesiggørligt).

Husk at forslag til generalforsamlingen skal indsendes inden 15.3.1981 til Sidevolden 23

si´r

Asbjørn

Indhold
=======

emnet til medlemsmøde den 1.3: Interrupt.

Maskinkode og

Assemblering.

I fortsættelse af mine tidligere udgydelser har jeg gjort mig nogle overvejelser.

Først om andre end A.L. læser dem,dernæst er vi jo kommet igang med en studiekreds i assemblering,hvor vi også gennemgår Nas-sys rutinerne,foruden en almindelig snakken om de problemer vi går og tumler med.

Dette fører mig til,at jeg vil ændre forudsætningen om at gennemgå T 2 monitoren,men i stedet kombinere et referat af,hvad vi lærer under Erik Palsbo's kyndige ledelse,med en forbindelse til de tanker,jeg tidligere gjorde mig.

Der vil imidlertid ikke ske nogen ændring i grundforudsætningen,at jeg forsøger at tilgodese de nytilkomne µ-ejere (og kommende),samt at jeg vil gøre mig de hæderligste anstrængelser for at komme de mindre eng./amr. sprogmindede til undsætning.

Det skal dog i denne forbindelse siges,at man må indstille sig på at lære (de mange forskellige) udtryk på dette computerkadautervælsk.(Altså for kommunikationens skyld,er der ingen vej udenom.)

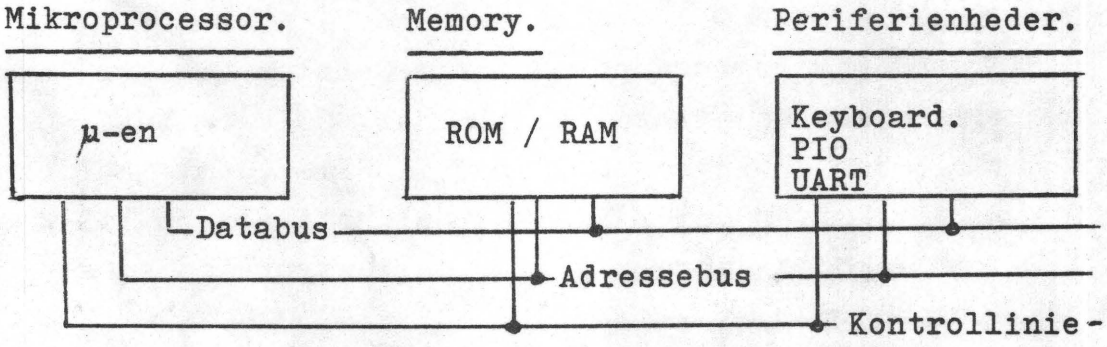
En fortsat forudsætning er også,at vi er en del,der kan lide den "rene"vare d.v.s. detoprindelige Nascom 1. oplæg med kun 1 K. ram,men i erkendelse af,at udviklingen altså går videre med operativsystemet,vil jeg dog forsøge at trække linierne bagud,således at man kan følge udviklingen.

Til sidst i disse overvejelser skal jeg også anføre, at der for mig ikke er nogen tvivl om atdet vil være den mest almentgyldige måde at nærme sig computeren via "maskin/assmble-sprogetidet "højniveausprogene" (ex. basic) oprindeligt er tænkt som hjælp for små hjerner med store maskiner,men at µ-processoren nu giver alle adgang til "små" men meget formående maskiner. (ex. basic betyder jo BEGINNERS o.s.v.)

Introduktion til
µ - computer: (side 1)
(E.C.H.)

Assemblerprogrammering.

1



Ovenstående vises princippet for en computer bestående af:

Microprocessoren der modtager, -behandler og afgiver data fra og til memory (mem. til mem) og periferienheder.

Memory bestående af Read Only Memory(kun læsbar,forud programmeret memory), og Random Access Memory (let tilgængelig,læse til og fra memory).

Periferienheder ex. PIO (parallel ind-og udgang) og UART (universal,asynkron receiver (modtager)transmitter (sender)).Disse skaber forbindelsen til/fra omverdenen.

Man kan dele funktionerne op i OPERATIVSYSTEMET med de forudprogrammerede(µ-ens)operationsprogrammer og MONITOREN (ROM i Nascom:T 2,B-Bug,T 4,NAS-SYS 1 og 3).

Til µ-ens styring og kontrol af systemet behøves en clockimpuls,hvis frekvens (el.cycle) angiver tiden for hver trinvise handling,der sker i systemet.(dette kan for Z-8o´s vedkommende ske ved dc-ændringer,men man stræber for at nå op i så mange MHz. som muligt.)

Foruden den antydede RAM findes der også i µ-en en række registre,der kan opfattes som en del af tilgængelig hukommelse.

Til forskel fra en clockcycle,er der et begreb,der hedder en INSTRUKTIONSCYCLE,der kan se således ud:

ØC8Ø LD (HL),8F

hvor LD (load = lad) den adresse i mem.,som HL (et registerpar i µ-en) peger på med den hexadecimale talværdi 8F.

LD kaldes INSTRUKTIONSKODEN ,mens (HL) og 8F kaldes første og anden operend.

Introduktion til
µ - computer: (side 2)
(E.C.H.)

Under udførelsen af en instruktionsperiode som denne, sker følgende:

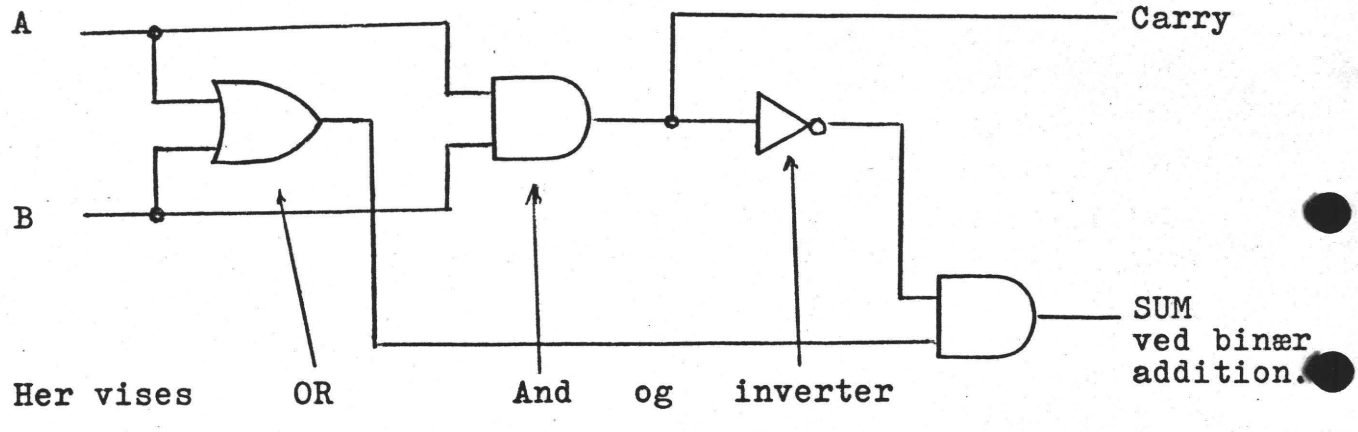
En PC (et registerpar Program Counter (tæller))når under udførelsen af et program til adr. ØC8Ø og herfra kommer via databus 8 binære værdier ØØ11 Ø11Ø (dette kaldes en byte) og vi skriver det i hexadecimal notation 36 til µ-en.

Her dekodes denne byte som en instruktion om at starte et mikroprogram.

Et mikroprogram åbner og lukker en serie elektriske kredsløb i en bestemt rækkefølge.

Således vil nu adressebussen blive sat til næste adr. og via databus hente data 1ØØØ 1111 (8F) til den adresse, som adressebus nu bliver sat til (den HL-registerparret pegede på, der var klammer omkring HL i operend 1).

En anden instruktionskode kunne have været ADC, der bestyder: adder med mente (ADd with Carry) og vi ser her et udsnit af denne operation udført i logiske kredse



Dette kan vises i et sandhedsskema for operationsinstruktionen:

		ADD	
A	B	SUM (binær)	Carry
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

Men det er en helt anden historie (lad mig høre om der er interesse for at høre den.

```
; OPSTART: EC80 aaaa bbbb cccc dd
; hvor dd er linieantal ved udskrift.

;               MONITORRUTINER
B2HEX:  EQU  68H          ;A til VDU
CRLF:   EQU  6AH          ;Carr.ret/Line Feed
MRET:   EQU  5BH          ;Retur til NAS-SYS
SPACE:  EQU  69H          ;Space til VDU
TX1:    EQU  6CH          ;HL og DE til VDU
PRS:    EQU 0EFH          ;EFTERFØLGENDE TIL VDU
RIN:    EQU 0CFH          ;AFVENT INDTAST

;
;               MONITOR RAM
ARGN:   EQU  0C0BH        ;ARGUMENT-ANTAL
ARG2:   EQU  0C0EH        ;Indtastet aaaa
ARG3:   EQU  0C10H        ;     "      bbbb
ARG4:   EQU  0C12H        ;     "      cccc
ARG5:   EQU  0C14H        ;     "      dd
ARG10:  EQU  0C1EH        ;RAMLAGER FOR dd
;
        ORG  0C80H

;
START:  SCAL CRLF
        LD   BC,(ARG5)    ;LINIEANTAL
        LD   IY,ARG10     ;RAMLAGER
        LD   (ARG10),BC
        LD   A,(ARGN)     ;ANTAL ARGUMENTER
        CP   5            ;5 INDTASTNINGER
        JR   NZ,ARGF      ;NEJ TIL FEJLUDSKRIFT
        LD   IX,ENS
        LD   HL,(ARG2)
        LD   DE,(ARG3)
        LD   BC,(ARG4)
        INC  BC           ;For at faa byte"0" med
RUTI:   LD   A,(DE)
        CPI               ;Samm. (HL) og (DE)
        JP   PO,SLUT      ;Hvis BC=0 til NAS-SYS
        CALL NZ,FORSK     ;Hvis uens TIL 'FORSK'
        INC  DE
ITUR:   JR   RUTI         ;Gentag
FORSK:  LD   IX,IENS
        DEC  HL           ;HL er INC ved sammenl.
        PUSH BC           ;Gem byte-counter
        SCAL TX1          ;Udskriv adresser
        SCAL SPACE
        SCAL SPACE
        LD   A,(HL)       ;1.mem. indhold
        SCAL B2HEX        ;(HL) til VDU
        SCAL SPACE
        LD   A,(DE)       ;2.mem. indhold
        SCAL B2HEX        ;(DE) til VDU
        INC  HL
        DEC  (IY)         ;LINIECOUNTER -1
        JR   NZ,VID
        DB RIN            ;AFVENT INDTASTNING
        CP   1BH          ;"ESC" INDTASTET ?
        JR   NZ,IESC
        SCAL CRLF
        SCAL MRET
IESC:   LD   BC,(ARG5)    ;LINIECOUNTER
        LD   (ARG10),BC
VID:    SCAL CRLF
        POP  BC           ;Hent byte-counter
KSROF:  RET
SLUT:   JP   (IX)         ;TIL ENS ELLER IENS
ENS:    DB   PRS,0DH
        DB   'Ens memory - indhold',0DH,0
IENS:   SCAL MRET
ARGF:   DB   PRS,0DH,'Argumentfejl'
        DB   'Indtast:',0DH
        DB   'EC80 1.adr. 2.adr. byteantal  linietal'
        DB   0DH,0
        SCAL MRET

        END
```

81.01.15. MHJ.

Memorysammenligner endnu en gang!

Nu kan du selv vælge antal linier, der skal udskrives. (Fra NW 1 - FORTSÆTTELSE)

EKSTRA Z80 KODER af D. S. PECKETT

Z80 er almindeligvis betragtet som den mest kraftfulde af 8-bit CPU'erne, og den bruges i NASCOM, TRS 80, ABC 80 og MZ 80K m.fl. Zilog beskriver den med 158 typer af instruktioner og med 696 mulige koder (+ data).

Hvis du anser dette for tilstrækkeligt er det uden interesse at vide, at på de fleste Z80'ere, kan man finde 88 flere objektkoder af interesse. Disse giver dig adgang til fire ekstra 8-bit registre - og jo mere du programmerer i maskinkode - jo mere vil du påskønne tilgængeligheden af flere registre.

Denne artikel forklarer om disse ekstra instruktioner og hvorfor de eksisterer. Der vil blive givet en test, der kontrollerer om koderne findes på din maskine.

Z80 arkitektur.

Lad os til en begyndelse se på, hvordan Z80 er opbygget (fig. 1). Kredsen har 2 sæt af registre, og hvert sæt indeholder en akkumulator (A), et flagregister (F) og seks almindelige 8-bit registre (B-L). Disse seks kan kombineres til 3 16-bits registre. CPU'en har instruktioner, der vælger registersæt under programafvikling.

Z80 har yderligere en programtæller (PC) og en stak pegepind (SP) og to 16 bit index registre (IX og IY). Derudover skal vi ikke besvære jer med I og R registret.

Z80 er udviklet fra 8080A, som den har arvet registrene A-L fra. Det andet sæt af registre (A'-L') findes ikke i 8080A - heller ikke IX eller IY !

Konstruktørerne af Z80 har bl.a. ved hjælp af ekstra hardware (registre) fyldt flere instruktioner på kredsen end tidligere mikroprocessorer. De bruger dog de samme koder for samme instruktioner. De ekstra instruktiner dækker områder som: bit test, relativ hop, register skift og flytning af datablokke. Dog mest betydningsfuldt i sammenhæng med denne artikel et væld af index instruktioner.

Disse instruktioner afhjælper den noget tunge programmering, man skulle udføre på 8080A, hvor man altid skal bruge HL som pegepind til hukommelsen. F.eks. hvis man skulle lægge indholdet af adressen 1234H til akkumulatoren måtte man programmere:

```
LD HL,1234H     ;HL=1234H
ADD A,(HL)      ;A=A+DATA
```

hvor Z80's udvider programmeringsmåderne ved hjælp af indes ordrene.

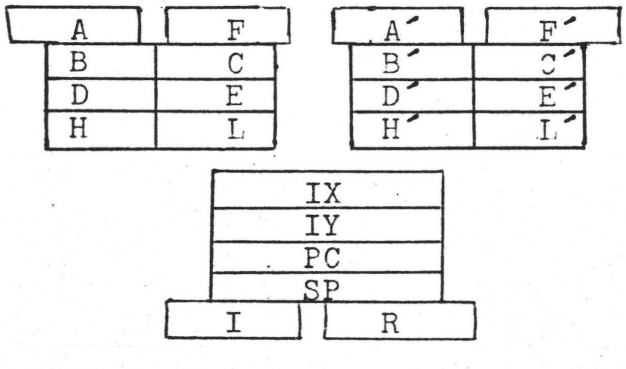


Fig. 1 Sådan ser Z80 ud i følge manualerne

Index adressering.

Hvis du ser på beskrivelsen af Z80´s sprog, vil du sandsynligvis opdage (håber jeg) noget interessant omkring indexadressering. Hver gang en instruktion har formen (HL), har den også en index form. F.eks

har ID A,(HL) denne LD A,(IR+d)
og bit 7,(HL) denne BIT 7,(IR+d).

Hvor IR betegner et af registerne IX eller IY, ydermere er der ingen indexordre, der ikke har en (HL) partner.

Jeg håber, at denne bestrivelse har klartgjort den nære sammenhæng mellem IR og HL. Denne sammenhæng kommer yderligere til udtryk, hvis man ser på maskinkoderne for Z80.

F.eks. (på hex form) for at udføre ADD A,(HL) skal skrives 84H, og den tilsvarende kode for ADD A,(IX+d) er DD 84 dd, hvor dd er afstanden til IX udtrykket i 2´ers komplement.

Et andet eksempel koden for BIT 7,(HL) er CB 7E, og for BIT 7,(IY+d) er FD CB 7E dd. Hvis du ser på listen af instruktioner, vil du opdage, at (IX+d)-kode er dannet af den tilsvarende (HL)-kode med et foranstillet DD og et efterstillet dd. For (IY+d) er der foranstillet FD i stedet for DD.

Den opdagelse forklarer også meget om den langsommere udførsel af indexordre end tilsvarende (HL)-ordre, da objektkoderne er to byte længre. Læsning fra memory tager ekstra tid.

På baggrund af samme opdagelse er jeg helt sikker på, at Z80 bruger samme indre logik for at dekode (HL) og (IR+d) instruktionerne. Det aktuelle valg af registre defineres af tilstedeværelsen eller ej af foranstillet instruktioner (DD eller FD).

Mulige ekstra ordre

Når vi har set, hvordan Z80 henter sine indexinstruktioner, opstår der en tanke. Indtil videre har vi brugt HL som 16 bits register, men vi kan selvfølgelig bruge HL som to selvstændige registre. Hvad sker der, hvis vi foranstiller DD på objektkoden for LD A,H ?

Hvis jeg gjorde dette på min Z80, fandt jeg overraskende at A indeholdt den højeste byte af IX ! Jeg havde en ny instruktion. Selvfølgelig gik jeg videre, ellers var det jo ikke noget at skrive om.

På alle de Z80´ere jeg har undersøgt, tillader det nære forhold mellem HL,IX og IY, at man betragter indexregistrene som 2 ottebits registre.

I almindelighed kan du ikke have for mange interne registre i en CPU, så denne opdagelse er meget værdifuld.

Det afhænger selvfølgelig af, om du benytter indexregistre eller ej. Men det giver dig 2 8-bit registre ekstra for hvert indexregis-ter, du ikke benytter.

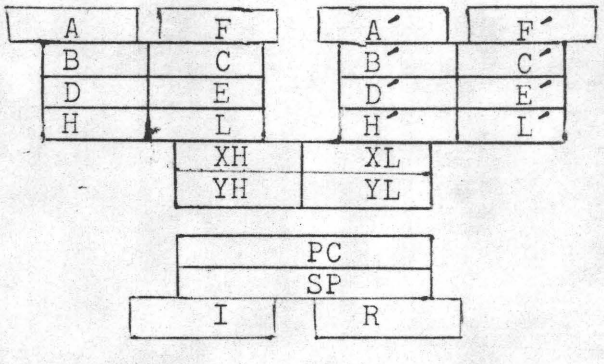


Fig. 2 Hvis du er heldig ser Z80 sådan ud.

Ekstra tilgængelige instruktioner

Lad os se, hvad vi kan bruge af disse ekstra registre. Først vil jeg lige kalde de to byte af IX for XH ogXL og tilsvarende for IY, YH og YL (fig. 2). Med disse registrenavne kan man, som foroven bruge mnemonic LD A,XH for instruktionen med objektkoden DD 7C.

Da jeg først opdagede disse ekstra kommandoer, håbede jeg, at XH osv. kunne bruges i alle Z80 operationer, der benyttede H og L. F.eks. kunne man have 'LD YL,B', 'SUB YH', 'CP XH' osv, men desværre synes Z80 ikke at arbejde helt på denne måde.

For det første er det ikke muligt at have LD XL,H. Det er ikke alt for overraskende !! Instruktionen skulle opbygges ved at foranstille DD for LD L,H (6B). Men Z80 vil da ikke vide om DD 6B betyder LD XL,H eller LD L,XH; den sætter egentlig LD XL,XH. Så vi kan ikke bruge både H og L med de ekstra registre i en enkelt operation.

Den anden begrænsning er mere uklar - dvs jeg ved ikke, hvorfor den eksisterer! De ekstra registre vil kun arbejde med de funktioner, der stammer fra 8080A og ikke i de nye Z80 instruktioner. Så vidt jeg kan se er forskellen, at alle 8080A compatible koder bruger en enkelt byte objektkode (+data), mens de specielle Z80 koder alle bruger to byte. Hvad end grunden er, betyder det, at du ikke kan bruge BIT, SET, RES, rotation eller shift. Stadig er der frie ekstra kommandoer, så vi kan ikke klage!

Tabel 1 viser alle de 'ekstra' instruktioner, som er mulige. Den giver ikke objektkoderne - dem kan du selv danne ved at foranstille DD eller FD som ønsket.

En lille advarsel. Jeg har vist de ekstra kommandoer i almindelig standard Z80 mnemonic. Men prøv ikke, at bruge dem i en assembler. Den vil ikke genkende dem. Enten må du skrive en ny assembler eller ty til håndassemblering. (eller DB 'DD' eller 'FD' foran de almindelig mnemonics. A.).

Det er vigtigt at huske, at disse ordre ikke er officielle. DVS. at de ikke forekommer i den officielle litteratur om Z80, så der er ingen garanti for, at alle Z80 vil udføre dem rigtig. Ydermere skal det nævnes, at ordrene kan forsvinde, hvis Zilog modificerer processorens indre logik. Men hvis en enkelt Z80 en gang har fungeret med disse nye ordre, vil den fortsætte dermed.

Hvis du ønsker at teste din maskine for disse instruktioner, er den bedste måde en række små maskinkodeprogrammer styret af et high-niveau sprog som f.eks. Basic.

MNEMONIC	TEST	
LD r,XR	LD1	'r' - Register A,B,C,D eller E
LD XR,r	LD2	'XR' - 'Register' XH,XL,YH eller YL
LD XR,data	LD3	'XR1','XR2' - Hvilken som helst XR
LD XR1,XR2	LD4	
ADC A,XR	ADDSUB	
ADD A,XR	ADDSUB	
SBC A,XR	ADDSUB	
SUB XR	ADDSUB	
INC XR	INCDEC	
DEC XR	INCDEC	
AND XR	ANDORX	
OR XR	ANDORX	
XOR XR	ANDORX	
CP XR	COMP	

Tabel 1.

Test din Z80

Når man skal lave et selvtestende program, må man først afgøre, hvad der er nødvendigt at checke. Er det nødvendigt at undersøge 'LD A,XH', 'LD B,XH', 'LD C,XH' osv., at alle fungerer tilfredsstillende? Nej, hvis man kan LD B,XH, er jeg helt sikker på, at det også kan lade sig gøre at 'LD' A, C, D og E. Det er altså nok at undersøge, om de ekstra registre kan 'LD' ind i et normalt register.

Det er godt nok for programmet, at checke de ekstra instruktioner i logisk sammenhængende blokke. Jeg foreslår, at vi bruger de 8 viste i tabel 1. Flowchartet viser testrækkefølgen, som går fra de mere simple til de mere komplekse test.

Hver blok undersøger et passende udvalg af mulige operationer. Testen skal gøre to ting 1) undersøge om den ekstra operation virker 2) checke at ingen ubrugte registre bliver ændret. Jeg besluttede, at den bedste måde at udføre dette på var at kalde en maskinkodesubrutine, der så skulle kalde de enkelte test.

Før hver test måtte alle registre sættes til kendte værdier og ved slutningen gemmes i RAM-lageret. Det kontrollerende program i basic skal finde og teste de gemte data mod de forventede.

Program 1 er en assemblerlistning af det kaldende underprogram, TSTALL. (Alle programmer er skrevet til TRS 80, derfor er TSTALL og basicprogrammet omskrevet til Nascom II af A.)

Programmet kalder efter tur de enkelte afsnit efter de er poked ind i samme RAM-adresse af basic. Hvis RET-instruktionen går galt, ved vi, at SP er blevet ændret af testen.

ADDSUB. Idenne test prøves hver af de fire 8-bit aritmetriske operationer een gang. Jeg har valgt værdier - og rækkefølge - så det så vidt muligt ikke er muligt, at flere efterhinanden udligner hver enkelt. FEJL

COMP. Når 'CP' bliver testet, skal man være sikker på, at Z-flaget bliver sat/slettet på de rigtige tidspunkter. Det er ordnet således, at 'LD' af A sørger for, at den forkerte værdi af A kommer med, hvis noget går forkert.

Disse maskinkode test kontrollerer jeg ved hjælp af et basicprogram, som gør det nemmere, at få fat i resultaterne og udskrive dem pænt. Programmet består af flere dele:

a) Indskriver maskinkoden i RAM
b) Udfører maskinkoden
c) Kontrollerer resultaterne
d) Udskriver resultaterne

Program 3 er en listning af det program jeg brugte (ændret til Nascom II af A.). Først er den kaldende rutine lagt i RAM, derefter lægges de enkelte check ud og udføres. Hvis alt går godt fortsættes. Opstår der fejl udskrives fejlmeddelelse og fejlliste. Derefter spørges om man vil fortsætte eller ej.

Konklusion.

De fleste - hvis ikke alle - Z80'ere har disse ekstra instruktioner, som Zilog er meget ømfindtlig overfor. Disse instruktioner giver den udprægede maskinkodemasochist fire ekstra 8-bit frittilgængelige registre at lege med, og de kan være særdeles nyttige.

Det er meget hurtigt at teste din Z80 for disse kommandoer. Hvis den har det, får du en ekstra gevinst, og hvis den ikke har - har du aldrig vidst, at du savner dem !!

```
0001
0002                    ;Program: TSTALL
0003
0004 0D00                       ORG 0D00H
0005 0D00                       MEM 0D00H
0006 0D33               TEST:   EQU 0D33H
0007
0008 0D00 F5            TSTALL: PUSH AF
0009 0D01 C5                    PUSH BC
0010 0D02 D5                    PUSH DE
0011 0D03 E5                    PUSH HL
0012 0D04 DDE5                  PUSH IX
0013 0D06 FDE5                  PUSH IY
0014
0015 0D08 3E75                  LD   A,75H
0016 0D0A 4F                    LD   C,A
0017 0D0B 41                    LD   B,C        ;BC=7575H
0018 0D0C 57                    LD   D,A
0019 0D0D 5F                    LD   E,A        ;DE=7575H
0020 0D0E 213930                LD   HL,3039H   ;HL=12345
0021
0022 0D11 CD330D                CALL TEST
0023
0024 0D14 ED43000E              LD   (0E00H),BC ;GEM RESULTATET
0025 0D18 ED53020E              LD   (0E02H),DE
0026 0D1C 22040E                LD   (0E04H),HL
0027 0D1F DD22060E              LD   (0E06H),IX
0028 0D23 FD22080E              LD   (0E08H),IY
0029 0D27 320A0E                LD   (0E0AH),A
0030
0031 0D2A FDE1                  POP  IY
0032 0D2C DDE1                  POP  IX
0033 0D2E E1                    POP  HL
0034 0D2F D1                    POP  DE
0035 0D30 C1                    POP  BC
0036 0D31 F1                    POP  AF
0037 0D32 C9                    RET
0038
0039 0D33               END
```

↑ Program 1

↓ Program 2

```
0001 0D33               ORG 0D33H
0002
0003
0004 0D33 DD213412 LD1:     LD   IX,1234H   LD   IX,1234H   ;IX=1234H
0005 0D37 FD217856          LD   IY,5678H   LD   IY,5678H   ;IY=5678H
0006 0D3B DD                DB   0DDH       LD   B,XL
0007 0D3C 45                LD   B,L        LD   C,YH       ;BC BØR VÆRE 3456H
0008 0D3D FD                DB   0FDH       LD   D,YL
0009 0D3E 4C                LD   C,H        LD   E,XH       ;DE BØR VÆRE 7812H
0010 0D3F FD                DB   0FDH       LD   A,XL       ;A BØR VÆRE 34H
0011 0D40 55                LD   D,L        RET
0012 0D41 DD                DB   0DDH
0013 0D42 5C                LD   E,H
0014 0D43 DD                DB   0DDH
0015 0D44 7D                LD   A,L
0016 0D45 C9                RET
0017 0D46               END
```

```
0001 0D33                      ORG  0D33H
0002
0003 0D33 014523    LD2:       LD   BC,2345H  LD   BC,2345H  ;BC=2345H
0004 0D36 119078               LD   DE,7890H  LD   DE,7890H  ;DE=7890H
0005 0D39 DD                   DB   0DDH      LD   XH,C
0006 0D3A 61                   LD   H,C       LD   XL,D      ;IX BØR VÆRE 4578H
0007 0D3B DD                   DB   0DDH      LD   YH,A
0008 0D3C 6A                   LD   L,D       LD   YL,E      ;IY BØR VÆRE 7590H
0009 0D3D FD                   DB   0FDH      RET
0010 0D3E 67                   LD   H,A
0011 0D3F FD                   DB   0FDH
0012 0D40 6B                   LD   L,E
0013 0D41 C9                   RET
0014 0D42           END
```

```
0001 0D33                      ORG 0D33H
0002
0003 0D33 DD210000  LD3:       LD   IX,0      LD   IX,0      ;IX=0
0004 0D37 FD210000             LD   IY,0      LD   IY,0      ;IY=0
0005 0D3B DD                   DB   0DDH      LD   XH,17H
0006 0D3C 2617                 LD   H,17H     LD   XL,23H    ;IX BØR VÆRE 1723H
0007 0D3E DD                   DB   0DDH      LD   YH,0F0H
0008 0D3F 2E23                 LD   L,23H     LD   YL,8BH    ;IY BØR VÆRE 0F08BH
0009 0D41 FD                   DB   0FDH      RET
0010 0D42 26F0                 LD   H,0F0H
0011 0D44 FD                   DB   0FDH
0012 0D45 2E8B                 LD   L,8BH
0013 0D47 C9                   RET
0014 0D48           END
```

```
0001 0D33                      ORG 0D33H
0002
0003 0D33 DD216400  LD4:       LD   IX,64H    ID   IX,64H    ;IX=0064H
0004 0D37 DD                   DB   0DDH      LD   XH,XL     ;IX BØR VÆRE 6464H
0005 0D38 65                   LD   H,L       LD   IY,3700H
0006 0D39 FD210037             LD   IY,3700H  LD   YL,YH     ;IY BØR VÆRE 3737H
0007 0D3D FD                   DB   0FDH      RET
0008 0D3E 6C                   LD   L,H
0009 0D3F C9                   RET
0010 0D40           END
```

```
0001 0D33                      ORG 0D33H
0002
0003 0D33 3E90      ADDSUB:    LD   A,90H     LD   A,90H     ;A=90
0004 0D35 DD212080             LD   IX,8020H  LD   IX,8020H  ;IX=8020H
0005 0D39 FD213040             LD   IY,4030H  ID   IY,4030H  ;IY=4030H
0006 0D3D DD                   DB   0DDH      ADD  A,XH      ;BØR:A=10H,CY=1
0007 0D3E 84                   ADD  A,H       ADC  A,XL      ;BØR:A=31H,CY=0
0008 0D3F DD                   DB   0DDH      SUB  YH        ;BØR:A=0F1H,CY=1
0009 0D40 8D                   ADC  A,L       SBC  A,YL      ;BØR:A=0C0H
0010 0D41 FD                   DB   0FDH      RET
0011 0D42 94                   SUB  H
0012 0D43 FD                   DB   0FDH
0013 0D44 9D                   SBC  A,L
0014 0D45 C9                   RET
0015 0D46           END
```

```
0001 0D33                      ORG 0D33H
0002
0003 0D33 DD21FF00 INCDEC: LD   IX,0FFH   LD   IX,0FFH    ;IX=00FFH
0004 0D37 FD2100FF         LD   IY,0FF00H LD   IY,0FF00H  ;IY=FF00H
0005 0D3B DD               DB   0DDH      INC  XH
0006 0D3C 24               INC  H         INC  XH
0007 0D3D DD               DB   0DDH      DEC  XL         ;IX BØR VÆRE 02FEH
0008 0D3E 24               INC  H         DEC  YH
0009 0D3F DD               DB   0DDH      DEC  YH
0010 0D40 2D               DEC  L         INC  YL         ;IY BØR VÆRE FD01H
0011 0D41 FD               DB   0FDH      RET
0012 0D42 25               DEC  H
0013 0D43 FD               DB   0FDH
0014 0D44 25               DEC  H
0015 0D45 FD               DB   0FDH
0016 0D46 2C               INC  L
0017 0D47 C9               RET
0018 0D48          END
```

```
0001 0D33                      ORG 0D33H
0002
0003 0D33 DD211CB5 ANDORX: LD   IX,0B51CH LD   IX,0B51CH  ;IX=0B51CH
0004 0D37 FD21D496         LD   IY,96D4H  LD   IY,96D4H   ;IY=96D4H
0005 0D3B 3E00             LD   A,0       LD   A,0        ;A=0
0006 0D3D DD               DB   0DDH      OR   XH         ;A BØR VÆRE B5H
0007 0D3E B4               OR   H         AND  YL         ;A   -  -   94H
0008 0D3F FD               DB   0FDH      XOR  XL         ;A   -  -   88H
0009 0D40 A5               AND  L         RET
0010 0D41 DD               DB   0DDH
0011 0D42 AD               XOR  L
0012 0D43 C9               RET
0013 0D44          END
```

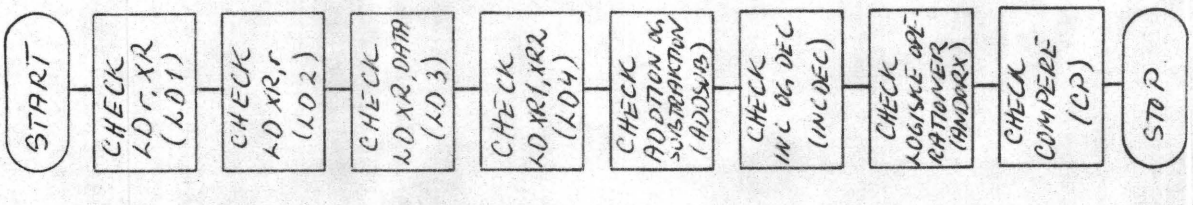
```
0001 0D33                      ORG 0D33H
0002
0003 0D33 DD213412 COMP:   LD   IX,1234H LD   IX,1234H  ;IX=1234H
0004 0D37 FD217856         LD   IY,5678H LD   IY,5678H  ;IY=5678H
0005 0D3B 3E34             LD   A,34H    LD   A,34H     ;A=34H
0006 0D3D DD               DB   0DDH     CP   XH        ;A=XH?
0007 0D3E BC               CP   H        RET  Z         ;RETUR VED FEJL
0008 0D3F C8               RET  Z        LD   A,56H     ;A=56H
0009 0D40 3E56             LD   A,56H    CP   YH        ;A=YH?
0010 0D42 FD               DB   0FDH     RET  Z         ;SKULLE RETURNERE HER
0011 0D43 BC               CP   H        LD   A,10H     ;SÆT FEJLKODE
0012 0D44 C8               RET  Z        RET            ;KUN HER VED FEJL
0013 0D45 3E10             LD   A,10H    END
0014 0D47 C9               RET
0015 0D48          END
```

↓ Program 3

```
100 CLS:PRINT"Test af Z80 ekstra instruktioner"
110 DOKE 4100,3328
120 FOR I=3328 TO 3378 STEP 2
130 READ A:DOKE I,A:NEXT
140 FOR II=1 TO 8
150 READ IT,J1,J2,J3,J4,J5,J6,F$
160 FOR I=3379 TO 3377+IT STEP 2
170 READ B:DOKE I,B:NEXT
180 DUM=USR(0)
190 GOSUB310
200 IFA=J1ANDBC=J2ANDDE=J3ANDHL=J4ANDIX=J5ANDIY=J6THENGOSUB290
210 IF Q=0 THENGOSUB250
220 Q=0
230 NEXT
240 END
```

```
250 PRINT F$;TAB(10)"FEJL":RETURN
260 REM *** En fejlrapport kan udskrives
270 REM *** da alle parametre er tilgængelige
280 REM *** hvis det er nødvendigt ▽▽▽
290 PRINT F$;TAB(10)"GOD"
300 Q=1:RETURN
310 BC=DEEK(3584):DE=DEEK(3586):HL=DEEK(3588)
320 IX=DEEK(3590):IY=DEEK(3592):A=PEEK(3594)
330 RETURN
340 REM DATA TIL TSTALL
350 DATA -14859,-6699,-6691,-6659,30014,16719
360 DATA 24407,14625,-13008,3379,17389,3584
370 DATA 21485,3586,1058,-8946,1570,-754
380 DATA 2082,12814,3594,-7683,-7715,-11807
390 DATA -3647,201
400 REM DATA TIL LD1
410 DATA20,52,13398,30738,12345,4660,22136,LD1
420 DATA 8669,4660,8701,22136,17885,19709
430 DATA 22013,23773,32221,201
440 REM DATA TIL LD2
450 DATA16,117,9029,30864,12345,17784,30096,LD2
460 DATA 17665,4387,30864,25053,27357,26621
470 DATA 27645,201
480 REM DATA TIL LD3
490 DATA22,117,30069,30069,12345,5923,-3957,LD3
500 DATA 8669,0,8701,0,9949,-8937,9006
510 DATA 9981,-528,-29906,201
520 REM DATA TIL LD4
530 DATA14,117,30069,30069,12345,25700,14135,LD4
540 DATA 8669,100,26077,8701,14080
550 DATA 27901,201
560 REM DATA TIL ADDSUB
570 DATA20,192,30069,30069,12345,-32736,16432
580 DATA ADDSUB
590 DATA -28610,8669,-32736,8701,16432,-31523
600 DATA -29219,-27395,-25091,201
610 REM DATA TIL INCDEC
620 DATA 22,117,30069,30069,12345
630 DATA 766,-767,INCDEC
640 DATA 8669,255,8701,-256,9437,9437
650 DATA 11741,9725,9725,11517,201
660 DATA 18,136,30069,30069,12345
670 DATA -19172,-26924,ANDORX
680 DATA  8669,-19172,8701,-26924,62
690 DATA -19235,-23043,-21027,201
700 REM DATA TIL COMP
710 DATA 22,86,30069,30069,12345
720 DATA 4660,22136,COMP
730 DATA 8669,4660,8701,22136,13374
740 DATA -17187,16072,-682,-14148,4158,201
Ok
```

655 REM DATA TIL ANDORX



FLOWCHART.

I

input/output, I/O almindelig betegnelse for periferudstyr der gør det muligt for computeren at kommunikere med omverdenen.

instruction en samling tegn der bestemmer en given computers operation.

interface en forbindelse mellem to apparater med forskellige funktioner.

interpreter (fortolker) et program, der sørger for eksekveringen af et andet program, f.eks. Basic interpreter.

interrupt (afbrydelse) medfører, at det igangværende programforløb afbrydes til fordel for en rutine med højere prioritet; når denne rutine er udført, vender computeren tilbage til det oprindelige programforløb.

J

jump betyder hop til et andet sted i programmet.

K

K læses som »kilo« og betyder 1024. En memory der indeholder 2 K bits rummer altså 2048 bits.

Karaktergenerator
Et kredsløb, der for et givet adresseinput udsender f.eks. et dotmønster for en karakter.

keyboard tangentbord.

Kildeprogram
Det inputmateriale, som man giver en maskine for at få det assembleret.

Komplement
Komplementet til et tal er det "omvendte" tal indenfor det benyttede talsystem.
Eks.: 7 er komplement til 2, idet de er lige langt fra hhv. det højeste og det laveste ciffer i titalsystemet.
I binær form er denne operation yderst let, da alle nuller bliver til ettaller og omvendt.
Ved subtraktion anvendes ofte det såkaldte 2-komplement, der er det normale komplement plus een.

L

Line Feed.
Linieskift.

loader et program der kontrollerer periferudstyrets funktion, medens et andet program indlæses i computerens memory.

logic circuit (logisk kredsløb) er en sammenkobling af logiske elementer, der giver en vis bestemt funktion.

LSI Large Scale Integration, en meget kompleks integreret kreds, der kan udføre komplicerede funktioner. En lille siliciumskive på nogle få mm² indeholder adskillige tusinde transistorer.

M

machine language det binære ækvivalent til en computers instruktionssæt. Det fundamentale sprog hvormed en computer arbejder; grupper af nuller og ettaller.

Maskincycle
Et antal af states fra 1 til 5, svarende til een arbejdsgang i CPU, f.eks. afhentningen af en operationskode i programlageret.

memory (hukommelse) en almindelig benævnelse for en enhed beregnet til at lagre binære data.

memory map en oversigt over hele computerens memory og hvordan den er udnyttet, oversigten starter med address 0000 (hex) og slutter med ffff (hex) for de almindelige 8 bit processorer.

Microcomputer
Betegnelse for et komplet system, bestående af CPU, program- og arbejdslager samt periferiske enheder.

Microprocessor
En komponent, der ved hjælp af et sæt instruktioner kan behandle data.

MSI
Medium Scale Integration.
IC af størrelsesordenen tællere mm.

Multiplexet bus
Et system, der reducerer antallet af ben pr. chip ved at latche busens indhold på de rigtige tidspunkter.

```
60000 REM * RUN FROM 60000 TO SAVE PROGRAM *
60010 REM
60020 A=DEEK(3187)
60030 AA=PEEK(6)
60040 IF AA=178 THEN60110
60050 IF AA=254 THEN60180
60060 PRINT"UNKNOWN MONITOR !!!!"
60070 STOP
60080 REM
60090 REM ********** NASSYS 1 IN USE **********
60100 REM
60110 PRINT"Present monitor: NASSYS 1"
60120 GOSUB60250
60130 DOKE 3187,1914: NULL21
60140 GOTO60360
60150 REM
60160 REM ********** NASSYS 3 IN USE **********
60170 REM
60180 PRINT"Present monitor: -- NAS-SYS 3 --"
60190 GOSUB60250
60200 DOKE 3187,1908: NULL 21
60210 GOTO60360
60220 REM
60230 REM * GET FILE NAME & SET UP VARIABLES *
60240 REM
60250 PRINT"Name of file to be saved:"
60260 INPUTA$
60270 PRINT
60280 PRINT"Set taperecorder in RECORD mode"
60290 PRINT"and press <ENTER>."
60300 AA$=LEFT$(A$,1)
60310 INPUT
60320 RETURN
60330 REM
60340 REM ** WRITE TO TAPE & SAVE PROGRAM **
60350 REM
60360 PRINT:PRINT"CLS"
60370 PRINT:PRINT"REM ";SPC(21-LEN(A$)/2);A$
60380 PRINT:PRINT:PRINT"CLOAD"
60390 CSAVE"AA$" : REM AUTOMATISK START: PRINT "RUN": REM FJERN 1.REM V. AUTOMA-
60400 NULL 1: DOKE 3187,A                                   TISK START.
60410 END
Ok
```

Automatisk start - stop

Jeg har med interesse læst Oles artikler om at få b.opt til at kører lidt efter. for at få G til at virke.
Jeg har selv løst problemet ved at sætte et relæ fra +12v til TP 10 (se manuellen side 15). Parrallelt over relæet har jeg monteret en kondensotor på 100 µF, samt en diode (1n4148).Det har virke upåklagelig i over et år.

Holm

G.R.Wilson: Machine Code Programming for
the Nascom 1 and 2 (Interface).

Maskinkodeprogrammering! Ordet fik det til at risle koldt ned ad ryggen på mig!(Hvis du, kære læser, her smiler overbærende, kan du roligt gå videre til næste artikel!)
Men for nogen tid siden fik jeg lejlighed til at se nærmere i ovenstående bog; den indfører stille og roligt læseren i maskinkodeprogrammeringens verden. Ved grundige forklaringer, tydelige diagrammer og overkommelige programmeringseksempler får forfatteren langsomt opbygget en forståelse for emnet hos læseren. Omhyggeligt repeteres forklaringerne lige dér, hvor man har brug for dem, og kun lidt nyt indføres ad gangen.
Der gennemgås overskueligt talsystemer, computeropbygningen, CPU-opbygningen og dernæst hvordan man noterer og udfører programmer. Hovedparten af bogen udgøres derefter af en fin gennemgang af diverse instruktioner og eksempler på deres brug i små maskinkodeprogrammer - gjort, så man føler sig behagelig sikker i sin (begrænsede) forståelse.
Efter et afsnit om video display (lidt kortfattet) afsluttes med Input/Output, hvor forfatteren pludselig tager et par syvmilestøvler på og stormer gennem stoffet.
Bogen er skrevet i en tør stil, knastør! Her forekommer ikke én henvendelse til læseren - ingen introduktion, ingen afslutning. Den er en brugsanvisning uden sidespring, men som sådan i hovedsagen god.
Med andre ord: Forekommer maskinkodeprogrammering nyt og svært, så er bogen alt i alt en glimrende vejleder at starte med - men man får også lov at betale for det: 120,50 kr, og om man finder den prisen værd er jo et spørgsmål. Det kunne tænkes,at samtaler med mere vidende medlemmer var en hyggeligere og billigere måde at skaffe sig den samme viden på!

Henrik Dyhr

```
  0 REM ********** File R ******************
  1 REM ***** Stock car Racing **************
  2 REM ** Per Busk Jepsen ** 12.1.1981 ******
  3 CLS: GOSUB 2000
  4 C$=CHR$(232)+CHR$(128)+CHR$(197)
  5 O$="    ":X$=CHR$(158)+CHR$(158)+CHR$(158)
  6 A$=CHR$(232)+CHR$(255)+CHR$(197)
  7 FOR X=1 TO 15
  8 PRINTTAB(8)CHR$(199);TAB(40)CHR$(248)
  9 NEXT
 10 SCREEN 1,15:T=500:C=15:M=C
 20 A=RND(1)*30+8:T1=T1+1
 22 IF T1=100 THEN GOSUB 2000:END
 27 PRINTTAB(8)CHR$(199);
 30 PRINTTAB(A)A$;
 35 PRINTTAB(40)CHR$(248)
 36 SCREEN C,4:PRINT C$
 37 SCREEN C,3:PRINT O$
 38 SCREEN 1,15
 40 GOSUB 100
 42 IF T1>60 THEN 20
 45 GOSUB 100
 47 IF T1>30 THEN 20
 50 GOSUB 100
 52 IF T1>10 then 20
 54 GOSUB 100
 60 GOTO 20
100 REM FOR X=1 TO T:NEXT:T=INT(T/2)
110 PRINTTAB(8)CHR$(199);TAB(40)CHR$(248)
115 GOSUB 1000
120 SCREEN C,4:PRINT C$
130 SCREEN C,3:PRINT O$
135 IFPOINT(C*2+2,12)=1 THEN GOSUB 200
136 IFPOINT(C*2-2,12)=1 THEN GOSUB 200
137 IFPOINT(C*2,12)=1 THEN GOSUB 200
140 SCREEN 1,15:RETURN
200 SCREEN C,4:PRINT X$
210 SCREEN C,5:PRINT X$
220 FORX=1 TO 500:NEXT:T=500:T2=T2+1
230 GOSUB 2000
240 SCREEN 1,14
250 PRINTTAB(8)CHR$(199);TAB(40)CHR$(248)
260 RETURN
1000 Q=INP(0)
1005 IF Q=255 THEN RETURN
1010 IF Q=253 THEN C=C-1
1020 IF Q=191 THEN C=C+1
1050 IF C<10 THEN C=10
1060 IF C>38 THEN C=38
1080 SCREEN M,3:PRINT O$:SCREEN C,4:PRINT C$
1085 SCREEN C,3:PRINT O$: M=C
1090 RETURN
2000 SCREEN 1,15
2010 PRINT"      Passeret";T1;"biler ***";
2020 PRINT T2;"sammenstoed";
2030 FOR X=2954 TO 3000 STEP 2
2040 DOKE X+64,DEEK(X)
2050 NEXT:PRINTCHR$(27);:PRINT:RETURN
```

(§ = $)

Med venlig hilsen

Per Busk Jepsen
Haderslev
tlf. eft. 1300
(04) 52 88 53

Hermed et lille bidrag fra det mørke jylland.

Efter jeg fandt ud af at man kan spørge paa INP(0) under programkørslen, lavede jeg fluks dette lille program. 100 biler bliver i stigende tæthed lukket ind nederst paa skærmen ved hjælp af RND, og fortsætter op ad skærmen hvor man selv er kørende paa linie 4. Det gælder nu om at styre udenom, eller at ramme saa mange som muligt (alt efter temperament), og dette gøres med

ENTER - til venstre
LF CH - til højre

ved fjernelse af REM i linie 100 for man langsom opstart efter hvert sammenstød.

****

Hermed et lille bidrag til jeres udmærkede blad.

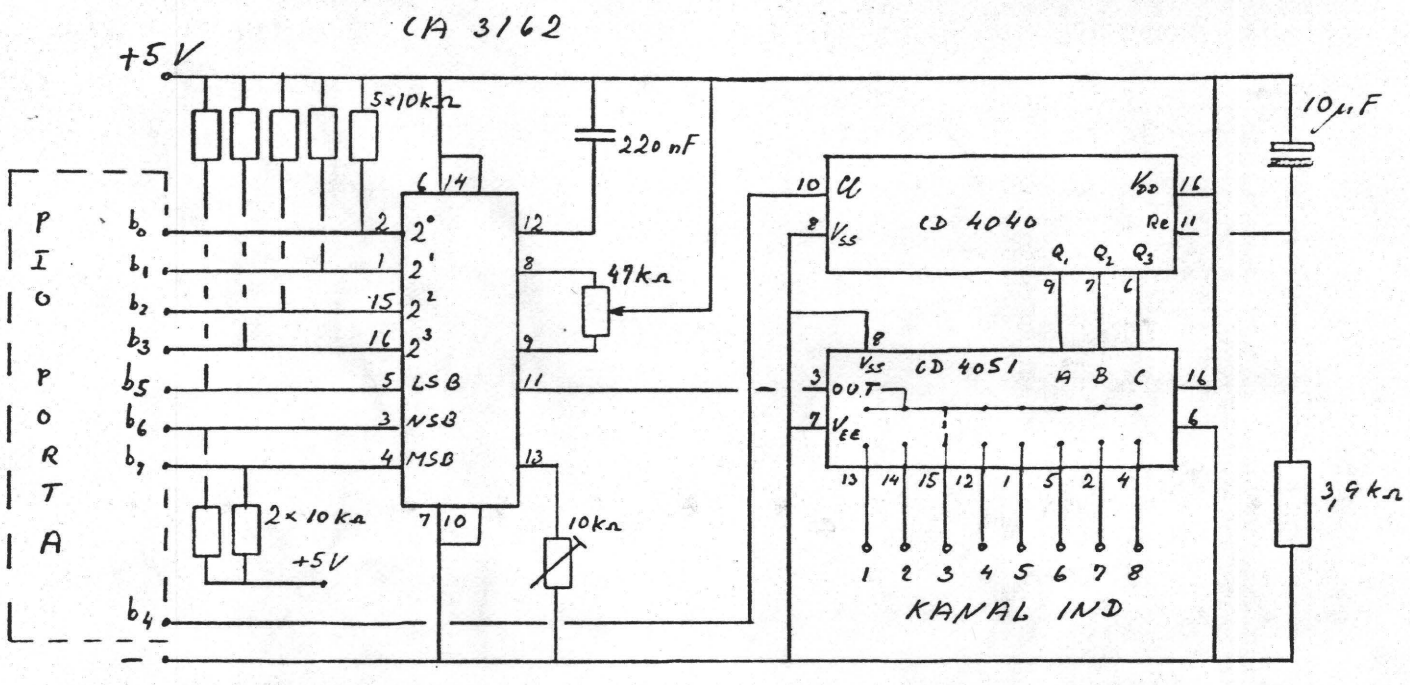
8 KANALS A/D KONVERTER.

Konverteren er opbygget om en digitalvoltmeter-kreds fra RCA. Kredsen hedder CA 3162 og da Josty har den på programmet er den ret let at få fat i(pris ca. 80 kr.). For at kunne udnytte kredsen bedre har jeg lavet en analog multiplexer, som gør,at med denne ene dvm-kreds kan man måle på 8 forskellige punkter på en gang.
På diagrammet ses hvordan man forbinder den til computeren. I denne opstilling foretages der 94 konverteringer i sekundet. Dette er en relativ lav hastighed når man sammenligner med "rigtige A/D-konverterer", men da den samlede pris på opstillingen også er lav er den et værdigt alternativ.
Alle pull-up modstandene på dvm-kredsen er nødvendige da udgangene leverer en strøm og ikke en spænding. De to trimpots er til at justere nulpunkt og finjustere gain. Spændingen, som dvm-kredsen modtager, kommer fra en c-mos kreds, der indeholder 8 analoge switches, og til at bestemme hvilken af de analoge switche der skal være on og hvilke der skal være off bruges en binær tæller også i c-mos. Denne tæller nulstilles når der sættes strøm på opstillingen og man kan så tælle frem ved at pulse med bit 4 på porten. Indgangsspændingen til CA 3162 og dermed også til de analoge switche bør ligge i området 0-999 mV. Hvis man ønsker at måle spændinger uden for dette område må man enten sætte en spændingsdeler foran eller,hvis det er mindre spændinger, sætte en forstærker foran med f.eks. en op-amp.
Hvis man ønsker at måle spændinger, der svæver i forhold til stel , har man også mulighed for dette da CA 3162 har differential indgang. Hvis man har brug for denne differentialindgang skal der kobles endnu en multiplexer på og dette skal gøres til ben 10 på dvm-kredsen, som er den inverterende indgang.

Venlig hilsen Morten Tolbak

DIAGRAM OVER 8 KANALS A/D KONVERTER

MK 31-1-81



$0 \leq E \leq 999\ mV$

```
0002
0003
0004                ø 8 kanals (3 cifret) a/d converter med ca3162
0005
0006                ø programmet indlaeser en serie maaleresultater
0007                ø disse skrives i linie 16
0008
0009                ø ved at erstatte 'scal mret' med 'ret' kan
0010                ø programmet bruges som subrutine og man kan
0011                ø paa den maade kontrollere eller regne paa
0012                ø de maalte vaerdier.
0013
0014
0015 0d00                   org  0d00h
0016 0d00                   mem  0d00h
0017
0018 005b           mret:   equ  5bh        ø monitor retur
0019 0bcd           oadr:   equ  0bcdh      ø output address
0020 0006           kport:  equ  06         ø kontrol for port a
0021 0004           dport:  equ  04         ø data port a
0022
```

```
0023 0d00 21cc0b   start:  ld   hl,oacr    ø
0024 0d03 3ecf             ld   a,0cfh     ø konfigurer port a
0025 0d05 d306             out  (kport),a  ø bit kontrol
0026 0d07 3eef             ld   a,11101111bø 1= input
0027 0d09 d306             out  (kport),a  ø 0= output
0028 0d0b 0608             ld   b,08       ø antal kanaler
0029 0d0d c5       next:   push bc
0030 0d0e cd200d           call msb        ø hent data
0031 0d11 c1               pop  bc
0032 0d12 cbe7             set  4,a
0033 0d14 d304             out  (dport),a  ø skift til naeste
0034 0d16 cba7             res  4,a        ø anal
0035 0d18 d304             out  (dport),a
0036 0d1a 23               inc  hl
0037 0d1b 23               inc  hl
0038 0d1c 10ef             djnz next       ø alle kanaler indlast?
0039 0d1e df5b             scal mret       ø ja ,retur
0040
0041 0d20 db04     msb:    in   a,(dport)  ø hent data
0042 0d22 cb7f             bit  7,a        ø er det msb ?
0043 0d24 20fa             jr   nz,msb     ø nej,sa prov igen
0044 0d26 cd3f0d           call displ      ø ja,sa udlaes tal
0045 0d29 23               inc  hl
0046 0d2a db04     nsb:    in   a,(dport)  ø se msb
0047 0d2c cb77             bit  6,a
0048 0d2e 20fa             jr   nz,nsb
0049 0d30 cd3f0d           call displ
0050 0d33 23               inc  hl
0051 0d34 db04     lsb:    in   a,(dport)  ø se msb
0052 0d36 cb6f             bit  5,a
0053 0d38 20fa             jr   nz,lsb
0054 0d3a cd3f0d           call displ
0055 0d3d 23               inc  hl
0056 0d3e c9               ret
0057
0058 0d3f 0605     displ:  ld   b,05       ø lad data falde til ro
0059 0d41 10fe     ego:    djnz ego
0060 0d43 db04             in   a,(dport)  ø hent data
0061 0d45 e60f             and  00001111b  ø udmask tallet
0062 0d47 c630             add  a,30h      ø konverter til ascii
0063 0d49 77               ld   (hl),a     ø display tal
0064 0d4a c9               ret
0065
0066 0d4b                  end
```

SÆLGES.

1stk. PHILIPS DIGITAL BÅNDOPTAGER DCR-220(arbejder med 6000 baud og 128K på et bånd) 900Kr.

DISPLAY

DL-707-R(fælles anode) 1stk.6Kr. 10stk.50Kr.

LED.

Rød lys DIODE 5mm.(med holder) 10stk.10Kr.. 50stk.40Kr..

Grøn lys DIODE 5mm.(klar plastik hus) 10stk.10Kr. 50stk.40Kr.

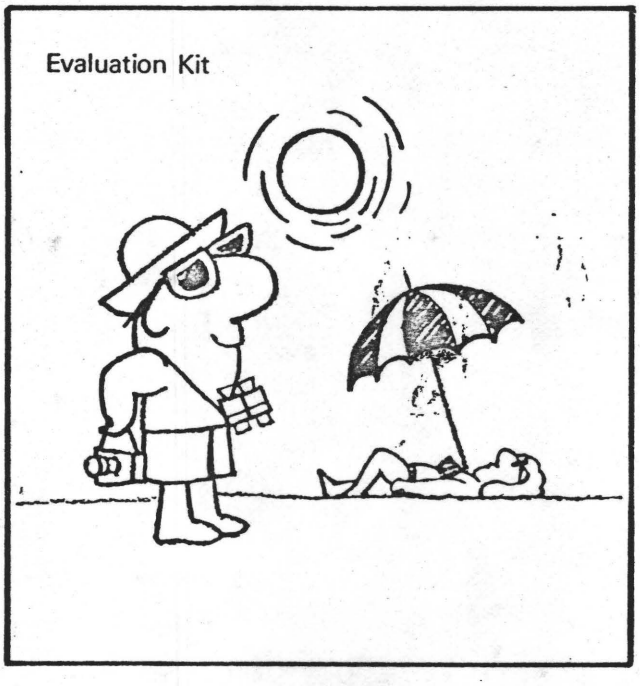
RAM

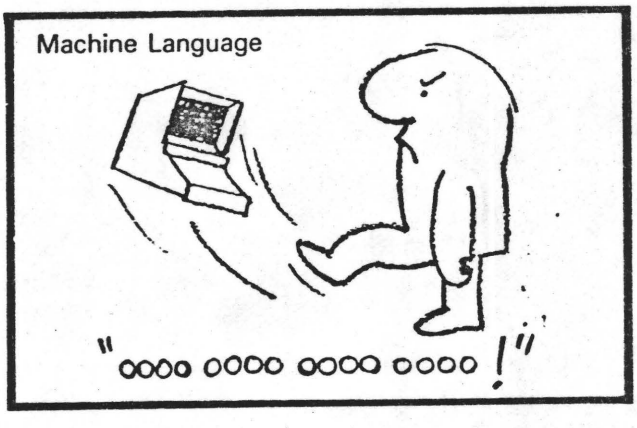
2114L-(1024x4) 1stk. 30Kr.
5101L-(256x4) 1stk. 25Kr.
4042L-(256x4) 1stk. 20Kr.
2112- (256x4) 1stk. 15Kr.

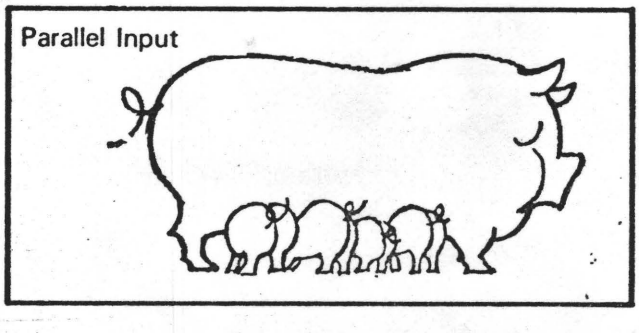
EPROM

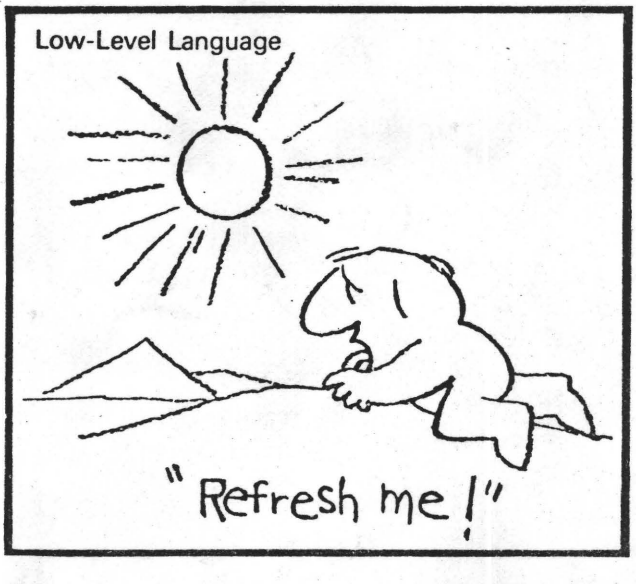
2516 (2716) (2048x8) 1stk. 100Kr.
2532 (4096x8) 1stk. 250Kr.

John Nielsen
Frisersvej 4,II
2920 Charlottenlund









Almindelige oplysninger om foreningen

BESTYRELSENS SAMMENSÆTNING:

Formand: (redaktør)	Asbjørn Lind Sidevolden 23 2730 Herlev 02 91 71 82
Næstformand:	Jesper Skavin Broholms Alle 3 2920 Charlottenlund 01 64 03 14
Kasserer:	Søren Sørensen Højlundvej 13 3500 Værløse 02 48 31 01
Teknisk redaktør:	Ole Hasselbalch Vibeskrænten 9 2750 Ballerup 02 97 70 13
Medlemsmødeleder:	Erik Hansen Lyngby Kirkestræde 6,1 2800 Lyngby 02 88 60 55 (dg. 8 - 15.30)

HENVENDELSE TIL FORENINGEN:

Indmeldelser, adresseændringer o.l. til kasseren
Programbiblioteket til næstformanden.

Øvrige henvendelser til formanden.
(herunder annoncer/stof til NASCOM NYT)

Kontigent for 1.1.81 til 1.7.81 : 80.00 kr.

Reduceret kontigent for studerende efter fremsendelse af gyldigt studiebevis for indeværende år (eller kopi heraf)

65,00 kr.

Indmeldelse 25,00 Kr.

NASCOM BRUGERGRUPPE.

PROGRAMBIBLIOTEKET.
==========
Prisliste pr. 1/2 1981.

MASKINKODEPROGRAMMER:
==========

M1:	Walled Chase.....	11.00 Kr
M1A:	Walled Chase ver 2.....	6.00 -
M2:	Robots.....	8.00 -
M3:	Submarines.....	7.00 -
M4:	Unizap.....	5.00 -
M5:	Sub Search.....	6.00 -
M6:	Attack.....	3.00 -
M7:	Piranha.....	3.00 -
M8:	3D-Labyrint.....	10.00 -
M9:	Space Invaders.....	7.00 -
M10:	Database ver 3.....	2.00 -
M11:	Mastermind II.....	2.00 -
M12:	Chase ver 2.....	1.00 -

BASICPROGRAMMER.
==========

B1:	Hello.....	3.00 Kr
B2:	Russisk Roulette.....	2.00 -
B3:	Star Trek.....	11.00 -
B4:	Cubist Art.....	1.00 -
B5:	Kalender.....	2.00 -
B6:	Magic Labyrinth.....	5.00 -
B7:	Eliza.....	5.00 -
B8:	Camel.....	4.00 -
B9:	Comrade X.....	9.00 -
B10:	Hangman.....	2.00 -
B11:	Dageantal.....	1.00 -
B12:	3D-Kryds & Bolle.....	4.00 -
B13:	Skattejagt.....	7.00 -
B14:	Skydebane.....	4.00 -
B15:	Slangen I Labyrinten.....	1.00 -
B16:	Pædagogiske Tests 1.....	7.00 -
B17:	Pædagogiske Tests 2.....	6.00 -
B18:	Macronoia.....	9.00 -
B19:	Print Using.....	2.00 -
B20:	Grafik Bogstaver.....	3.00 -
B21:	Sypigetips.....	4.00 -
B22:	Mastermind.....	4.00 -
B23:	Tårnene I Hanoi.....	3.00 -
B24:	Nim.....	3.00 -
B25:	Backgammon.....	5.00 -

SPECIALPROGRAMMER.
==========

S1: MAT 4..... 15.00 Kr
S2: Disassembler..... 10.00 -
(S2: Kun på bånd)

PROGRAMBÅND.
==========

(Priserne er uændrede, men medtages her for fuldstændighedens skyld.)

PB1: Maskinkode (M1,2,3,4,5).. 20.00 Kr
PB2: Maskinkode (M1A,7,11,10). 25.00 -
PB3: Maskinkode (M8,9,12,10).. 25.00 -
PB4: Basic (B3,5,2,4)..... 30.00 -
PB5: Basic (B13,12,10)..... 30.00 -
PB6: Basic (B15,19,20,14)..... 30.00 -
PB7: Basic (B16,17)..... 30.00 -
PB8: Basic (B18,21,24)..... 30.00 -
PB9: Basic (B22,23,25)..... 30.00 -

Priserne i denne liste erstatter priserne i listen fra 1. december 1980.

Med hensyn til beskrivelse af programmerne og krav til computeren henvises til listen af 1/12 - 1980.

Betaling for ydelser skal ske over klubbens girokonto v.h.a. det girokort, som vedlægges programmerne eller båndende. Det et således ikke nødvendigt at vedlægge checks, frimærker eller sende postanvisninger ved bestillingen.

Bestillinger af programmer og/eller programbånd skal ske skriftligt til nedenstående adresse med angivelse af nummer samt, for programbåndendes vedkommende, hvilken indspillestandard (KANSAS CITY 1200/300 baud eller NASCOM 1 235 baud).

For bestillinger til et samlet beløb på 10 kr eller derunder pålægges 3 kr i porto.

ALLE HENVENDELSER VEDRØRENDE PROGRAMBIBLIOTEKET SAMT BESTILLINGER SKAL SKE TIL:

JESPER SKAVIN
BROHOLMS ALLE 3
2920 CHARLOTTENLUND.

/J.S

Kære medlemmer.

Vi har siden foreningens start lagt kapital til side til større indkøb. Dette har kunnet lade sig gøre ved 1) at få trykt NN gratis eller til favorabel pris, 2) at samle og kuvertere det selv, 3) at få en lille indtægt fra salg af programbånd.

Bestyrelsen besluttede på sidste bestyrelsesmøde at indkøbe en printer, når vi havde råd, og hvis medlemmerne enten på en generalforsamling eller via en urafstemning havde godkendt indkøbet.

Vi har nu fået tilbudt en printer, vi kan betale (rabat), og af en sådan trykkvalitet, at vi kan kopiere udskrifterne med godt resultat.

Et sådant indkøb ville 1) lette redaktørens arbejde, idet støjniveauet er betydeligt lavere og hastigheden større end redaktørens egen IBM kuglehovedprinter, 2) det ville gøre os uafhængig af en forhandlers åbningstider og transport til og fra, når vi skal have udskrevet vores programlister m.m., 3) der ville blive indført en ny medlemsservice, så alle kunne få glæde af printeren, idet man ville kunne få en udskrift af assembler- og basicprogrammer ved indsendelse af bånd.

Vi agter nu at foretage en urafstemning blandt medlemmer. Så vil du på vedlagte frankerede svarkort udtrykke din mening angående køb af printer af allerede opsparede midler.

Du skal skrive JA, hvis du er for og NEJ, hvis du er imod.

Der vil jo også være plads til supplerende meninger angående foreningens arbejde, bladets udseende og indhold m.m. Men husk et tydeligt JA eller NEJ.

Vi vil gerne have kortet tilbage inden en uge fra modtagelsesdagen.

Bestyrelsen

PS. Her ser du eksempler på udskrifter fra printeren:

EPSON Matrix-printer MX-80 print sample

ABCDEFGHIJKLMNOPQRST
1234567890!§#$%&*()^
EPSON MAKES more pri

ABCDEFGHIJKLMNOPQRST
1234567890!§#$%&*()^
EPSON MAKES more pri

ABCDEFGHIJKLMNOPQRST
1234567890!§#$%&*()^
EPSON MAKES more pri

ABCDEFGHIJKLMNOPQRST
1234567890!§#$%&*()^
EPSON MAKES more pri

ABCDEFGHIJKLMNOPQRSTUVXYZ
1234567890!§#$%&*()^_+@-='
EPSON MAKES more print mechanisms

ABCDEFGHIJKLMNOPQRSTUVXYZ
1234567890!§#$%&*()^_+@-='
EPSON MAKES more print mechanisms

ABCDEFGHIJ
1234567890
EPSON MAKE

ABCDEFGHIJ
1234567890
EPSON MAKE

ABCDEFGHIJ
1234567890
EPSON MAKE

ABCDEFGHIJ
1234567890
EPSON MAKE

ABCDEFGHIJKLMNOPQ
1234567890!§#$%&*
EPSON MAKES more

ABCDEFGHIJKLMNOPQ
1234567890!§#$%&*
EPSON MAKES more
RSTUVXYZ
()^_+@-='
print mechanisms
<>?,./
than ANYONE ELSE
IN THE WORLD.

Program: Side af

	P.C.:	Maskinkode:	Label:	Mne.:	Oprnds.:	Kom.:

Program: ______________________ Side ___ af ___

	P.C.:	Maskinkode:	Label:	Mne.:	Oprnds.:	Kom.: