

N A S C O M ^Z 8 0 N Y T

NR: 1
2. årgang

NASCOM BRUGERGRUPPE
Sidevolden 23
2730 Herlev
Giro 674 2602

Januar 1981

Er medlemskontigentet betalt? Ellers skynd dig på postkontoret og få afsendt beløbet, hvis du vil have næste nummer af Nascom Nyt. Er vores forening dyr? NEJ !!! Jeg vil hævde, at vores forening er billig med alt. Til sammenligning vil jeg anføre priserne fra Foreningen for private brugere af microdatamater. Med vore priser i parentes.

Programbånd 45-50 kr., (25-30 kr.); programlist ---, (50 øre pr. ark); assemblerkursus 400 kr. for 12 timer, (gratis for ?? timer); basickursus 200 kr. for 6 timer, (hvis, så gratis); før du køber datamat 100 kr. for 3 timer, (ofte gratis telefon konsultation!); blad 15 kr. pr. nr., (med i medlemskontigent); kontigent 110 kr., (80 kr.)

Hertil skal nævnes at ingen har gratis medlemsskab af foreningen hos os! Dette opnås hos F.P.B.M., hvis man kommer med et nyt medlem eller hjælper til i foreningen.

Vær med til at udbrede kendskab til vores forening, hvor du kommer frem. Vær med i foreningens arbejde ved at være åben for nye ideer og impulser - og meddel dem til bestyrelsen. Vær med til at præge Nascom Nyt ved at indsende programmer, artikler, dagbøger, tanker og læserbreve. Vær med til vore medlemsmøder og åben hus aftener. Vær med til at rose og dadle. - Kun derigennem bliver vi bedre og bedre og undgår at stivne i en kliche!

si'r Asbjørn

INDHOLD

=====

Side 2:	Læserbreve
Side 4:	Salg, fejl
Side 5:	Memorytest (assembler)
Side 7:	Memorysammenligner (-"-)
Side 8:	Begynd på maskinsprog IV
Side 10:	Ordlister (A-H)
Side 12:	Krydsreferance (basic)
Side 14:	Tipspræmier (basic)
Side 16:	Cirkeltegning (basic)
Side 17:	Decimal til binær (-"-)
Side 18:	Anmeldelse af BITS og PC's TOOLKIT
Side 19:	Referat fra bestyrelsesmøde
Side 20:	Anmeldelse Math 48
Side 22:	Om foreningen

Christian Dahl Sørensen
Postbox 306
DK-3900 Godthåb, Grønland

19.11.80

Kære Z 80 og NASCOM medamatører.

Jeg følger med stor interesse jeres arbejde, dog er jeg jo desværre ikke i stand til at møde op til jeres arrangementer, men det har jo en naturlig forklaring.

I efterlyser stof, så jeg vil hermed yde et beskedent bidrag til jeres blad.

Først lidt om maskineriet: Det består af NASCOM 2 bestykket med max. RAM, og maskinen kører 4 MHz uden wait-states.

Jeg har haft maskinen i næsten et år, og den eneste fejl, jeg har haft, var en defekt resetknap på keyboardet, den hang. Der er ingen problemer med at få ROM-basic til at køre 4 MHz uden wait, men memory-boardet kræver et par ændringer: IC 37 er erstattet med en 74S00 og IC 31 er erstattet med en 7474. Dette var tilstrækkeligt til at få kortet til at køre fejlfrit på 4 MHz uden wait. Ram-kredsene er 250 ns kredse. Det vedlagte testprogram til NASCOM 2 er fortrinligt til at kontrollere at alt er under kontrol (se vedlagte listning). Ved at vise pass nummer, kan man også fange periodiske fejl i RAM, eftersom fejl vil blive listet med adresse, byte ind, byte ud og pass nr.

RAM adresserne 4000 - BFFF bliver betjent af et 32K RAM-kort fra Mogens Pelles Z80 projekt. Det virker glimrende og er billigere end de originale NASCOM memory-boards.

Det kan dog kun programmeres i 16K banks, men de "skæve" adresser klares jo nemt med lus på NASCOM memory kortet. Række 1 er adresseret til 1000 - 3FFF og C000 - CFFF. 2. række er kun luset til de sidste 4K, D000 - DFFF, så der sidder 12K, som ikke bliver brugt, men det er nemmere at bruge ene 16K kredse. 4K dynamisk RAM er næsten lige så dyrt og så får man problemer med kortet.

Z80 kortet kan kobles direkte på Nasbussen, adresse og datalinier + MREQ, RD, WR, RFSH + power. Den projekterede DC-DC konverter på kortet, kan udelades, da Nasbussen har alle spændinger tilgængelige.

Kortet har kørt i over 6 måneder uden nogen problemer, så er der nogen der søger en billig løsning til udvidelse af RAM lager på NASCOM 2, er dette langt billigere end de originale kort, og så fylder de langt mindre.

Printeren her (det hele var udskrevet på TTY - men omskrevet af A) er en teletype med puncher og reader, tilkoblet gennem et relæ. Denne maskine skulle efter bogen køre RS 232, men det ville ikke makke ret, så jeg har koblet et relæ imellem, og til 110 BAUD er relæet hurtigt nok.

Som I kan se af bilagene, har jeg investeret i ZEAP

assembler, og jeg synes det er de bedste penge jeg har givet ud. Det køre bare derudaf, og det medfølgende memory-test er lavet ved hjælp af ZEAP'en.

(Herfølger en side om den medsendte disassembler, der virker godt, men jeg har videresendt den til næstformanden til vurdering og optagelse i programbiblioteket. A.)

Jeg er ved at prøve at knække ROM-basicen, jeg forsøger at finde ud af hvilken adresse der hoppes til når man skiver "RUN". Så vil der nemlig være mulighed for at få basicprogrammer til at starte sig selv. Er der nogle der kan hjælpe mig med at knække basicen - hører jeg gerne fra jer.

Nu vil jeg slutte for denne gang, fortsat god arbejdslyst

Med venlig hilsen

Christian Dahl Sørensen

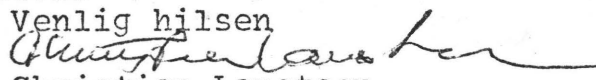
PS. Ang. AUTOstart se NN 7 side 9. A.

Jeg har, som det hedder på EDBsk konkateneret Jesper Skavins opfordring og Steen Lærkes kommentar i sidste nummer af NN og får: Vi sidder allerede i dag dybt begravet i problemer om programudveksling. Der findes mindst een T2-monitor i klubben og vel flere T4; desuden er der NASSYS, tilmed i flere versioner, og nogle af dem er uforligelige med de POLYSYS, der også findes; endelig hender klubben sig også til andre brugere af Z-80 end netop NASCOM-ejere, og alt det disse mennesker har fælles er netop Z-80'eren. Det oplevede jeg helt klart, da jeg med små antydninger fra Asbjørn om NASSYS' virkemåde prøvede at "disemle" Erik Palsbos databaseprogram. Men det slutter ikke her. Der findes også forskellige lagerkonfigurationer, og er Custom Electronics tapeinterface nu også forligelig med de floppydiskinstallationer der findes, og kan printerne styres på samme måde overalt? Eller skal vi samles i små enklaver, der er karakteriseret ved, at alle har samme operativsystem, samme lagerkonfiguration og samme periferiudstyr? Eller skal vi sige, at det der fælles er Z-80 assembler og lade dette være udgangspunktet for programudveksling? Og hvis vi gør det, hvad kan man så tillade sig at bruge af ordrer. Privat bruger jeg rask væk halve indexregistre som nævnt i forrige nr af NN, men det må nok udelades i et udvekslingsprogram. Så er der EXX. Mange af klubbens medlemmer bruger sikkert slet ikke interrupt, så de kan være ligeglade. Så er der nogle, der bruger interrupt alene

til floppydisk (og om kort tid også til tape siger Ole Kiilerich) og hvis de ikke drømmer om at bruge interrupt i andre situationer, er det jo rart, at EXX er reserveret alene hertil. Men efter at have talt med Ole K. sidder jeg i disse dage og sveder over lodningerne til en ny interruptport til min temperaturregulering og pusler med tanken om at gøre noget ved UARTen; om det bliver busrequest eller interrupt ved jeg ikke endnu. Og jeg er vel ikke ene om sådanne tanker.

Hvad angår regnehastigheder er mine eneste sammenligningsgrundlag MM57109 og SR 59 og de er begge væsentligt langsommere end det matematikprogram, jeg stadig pudser af på, i visse tilfælde med en faktor på 10 gange, men "nøjagtigere", hvis antal cifre ellers er en realistisk målestok.

Venlig hilsen


Christian Laustsen

S Æ L G E S:

=====

1 STK. Prof. 20" Philips Videomonitor type:
LDH 2110, båndbredde 10MHz 800 Kr.

16 STK 4116 (16kx1) for 45 Kr. stykket

Preben Asmild 01 75 06 35

RETTELSE

=====

Linie 1000 side 8 i NN 8 (1980) skal have samme udseende som linie 650 side 15 i dette nummer!!!

MØDE

====

Husk medlemsmødet !!!!


```

0001 ;MEMORY TEST FOR NASCOM 2
0002 ;(C) COPYRIGHT 1980 BY DS (GRØNLAND)
0003
0004 ;MEMORY TEST, WRITES HEX CODES
0005 ;00 - FF INTO EVERY MEMORY LOC.
0006 ;AND TESTS THAT EACH LOC. READS
0007 ;THE BYTE CORRECTLY.
0008
0009 ;BY ERROR THE FOLLOWING IS DISPLAYED:
0010 ;ADRESS - BYTE WRITTEN - BYTE READ BACK
0011
0012 ;THE PROGRAM IS ENTERED BY
0013 ;E D00 'FIRST' 'LAST' ADRESS TO BE TESTED.
0014
0015 0C0E ARG2: EQU 0C0EH
0016 0C10 ARG3: EQU 0C10H
0017 0BCA SCREEN: EQU 0BCAH
0018 0C29 CURSOR: EQU 0C29H
0019
0020 0D00 ORG 0D00H
0021 0D00 MEM 0D00H
0022
0023 0D00 AF XOR A
0024 0D01 32BD0D LD (PASSH),A
0025 0D04 3C INC A
0026 0D05 32BC0D LD (PASSL),A
0027 0D08 3E0C LD A,0CH ;CODE FOR CLS
0028 0D0A F7 RST 30H
0029 0D0B 21180D LD HL,TITLE
0030 0D0E 11CA0B LD DE,SCREEN
0031 0D11 011100 LD BC,17 ;BYTES TO BE PRINT
0032 0D14 EDB0 LDIR
0033 0D16 1811 JR CPS
0034 0D18 41444452 TITLE: DB 'ADDR. BYTE. PASS'
0035 0D1C 2E204259
0036 0D20 54452E20
0037 0D24 20504153
0038 0D28 53
0039 0D29 ED52 CPS: SBC HL,DE
0040 0D2B 22BA0D LD (CPOS),HL
0041 0D2E 210A08 LD HL,080AH
0042 0D31 22290C LD (CURSOR),HL
0043 0D34 0E00 MTEST: LD C,0
0044 0D36 2A0E0C OUTER: LD HL,(ARG2)
0045 0D39 ED5B100C LD DE,(ARG3)
0046 0D3D 13 INC DE
0047 0D3E 79 INNER: LD A,C
0048 0D3F 77 LD (HL),A
0049 0D40 46 LD B,(HL)
0050 0D41 B8 CP B
0051 0D42 C4940D CALL NZ,ERROR
0052 0D45 23 INC HL
0053 0D46 B7 OR A
0054 0D47 ED52 SBC HL,DE
0055 0D49 19 ADD HL,DE
0056 0D4A 20F2 JR NZ,INNER
0057 0D4C 0C INC C
0058 0D4D 79 LD A,C
0059 0D4E CD690D CALL PRINT
0060 0D51 B7 OR A
0061 0D52 20E2 JR NZ,OUTER
0062 0D54 F5 PUSH AF
0063 0D55 3ABC0D LD A,(PASSL)
0064 0D58 3C INC A

```

```

0065 0D59 2008      JR    NZ,NONINC
0066 0D5B 3ABD0D    LD    A,(PASSH)
0067 0D5E 3C        INC   A
0068 0D5F 32BD0D    LD    (PASSH),A
0069 0D62 AF        XOR   A
0070 0D63 32BC0D    NONINC: LD  (PASSL),A
0071 0D66 F1        POP   AF
0072 0D67 18CD      JR    OUTER
0073 0D69 F5        PRINT: PUSH AF
0074 0D6A C5        PUSH BC
0075 0D6B D5        PUSH DE
0076 0D6C 2A290C    LD    HL,(CURSOR)
0077 0D6F E5        PUSH HL          ;GEM CURSOR POS.
0078 0D70 F5        PUSH AF          ;GEM TEST BYTE
0079 0D71 010600    LD    BC,6
0080 0D74 09        ADD   HL,BC
0081 0D75 22290C    LD    (CURSOR),HL
0082 0D78 F1        POP   AF
0083 0D79 DF68      DB    0DFH,68H   ;PRINT TEST BYTE
0084 0D7B 010700    LD    BC,7
0085 0D7E 09        ADD   HL,BC
0086 0D7F 22290C    LD    (CURSOR),HL
0087 0D82 3ABD0D    LD    A,(PASSH)
0088 0D85 DF68      DB    0DFH,68H   ;PRINT PASS HIGH ORDER
0089                      ;BYTE
0090 0D87 3ABC0D    LD    A,(PASSL)
0091 0D8A DF68      DB    0DFH,68H   ;PRINT PASS LOW ORDER
0092                      ;BYTE
0093 0D8C E1        POP   HL
0094 0D8D 22290C    LD    (CURSOR),HL
0095 0D90 D1        POP   DE
0096 0D91 C1        POP   BC
0097 0D92 F1        POP   AF
0098 0D93 C9        RET
0099 0D94 E5        ERROR: PUSH HL
0100 0D95 D5        PUSH DE
0101 0D96 C5        PUSH BC
0102 0D97 DF66      DB    0DFH,66H   ;PRINT HL I ASCII
0103 0D99 DF69      DB    0DFH,69H
0104 0D9B C1        POP   BC
0105 0D9C C5        PUSH  BC          ;RESTORE REGISTRES
0106 0D9D 79        LD    A,C
0107 0D9E DF68      DB    0DFH,68H   ;PRINT BYTE WRITTEN IN
0108 0DA0 DF69      DB    0DFH,69H   ;PRINT A SPACE
0109 0DA2 78        LD    A,B
0110 0DA3 DF68      DB    0DFH,68H   ;PRINT BYTE READ BACK
0111 0DA5 DF69      DB    0DFH,69H
0112 0DA7 DF69      DB    0DFH,69H
0113 0DA9 3ABD0D    LD    A,(PASSH)
0114 0DAC DF68      DB    0DFH,68H
0115 0DAE 3ABC0D    LD    A,(PASSL)
0116 0DB1 DF68      DB    0DFH,68H
0117 0DB3 EF        RST   28H
0118 0DB4 0D00      DB    0DH,0
0119 0DB6 C1        POP   BC          ;RESTORE REGISTERS
0120 0DB7 D1        POP   DE
0121 0DB8 E1        POP   HL
0122 0DB9 C9        RET
0123 0002          CPOS:  DS    2
0124 0001          PASSL: DS    1
0125 0001          PASSH: DS    1
0126
0127          ;Oprindeligt skrevet paa ZEAP assembler, men da
0128          ;teksten var utydelig,          så omskrevet direkte
0129          ;til NAP. (A.)
0130 0DBF          END

```

```

0001
0002                                     ;                               80.11.13. MHJ
0003                                     ;
0004                                     ;-----
0005                                     ;
0006                                     ;       Opstart: E C80 aaaa bbbb cccc
0007                                     ;
0008                                     ;       Hvor   : aaaa = Start adr. i 1. mem.omr
0009                                     ;                bbbb = Start adr. i 2. mem.omr
0010                                     ;                cccc = Antal byte
0011                                     ;
0012                                     ; Ved forskel udskrives:  dddd eeee ff gg
0013                                     ;
0014                                     ;       Hvor   : dddd = Adr. i 1. mem.omr.
0015                                     ;                eeee = Adr. i 2. mem.omr.
0016                                     ;                ff = Indhold i dddd
0017                                     ;                gg = Indhold i eeee
0018                                     ;
0019                                     ; Efter sammenligning:  Retur til NAS-SYS
0020                                     ;
0021                                     ;
0022 0068                               ; MONITORROUTINER
0023 006A                               ;
0024 005B                               ;
0025 0069                               ;
0026 006C                               ;
0027                                     ;
0028                                     ;
0029 0C0E                               ; ARG2: EQU 0C0EH      ; Indtastet aaaa
0030 0C10                               ; ARG3: EQU 0C10H      ;      "      bbbb
0031 0C12                               ; ARG4: EQU 0C12H      ;      "      cccc
0032                                     ;
0033 0C80                               ; ARG: EQU 0C80H
0034 0C80                               ; MEM 0C80H
0035                                     ;
0036 0C80 2A0E0C   START: LD HL,(ARG2)
0037 0C83 ED5B1000 LD DE,(ARG3)
0038 0C87 ED4B1200 LD BC,(ARG4)
0039 0C8B 03       INC BC      ;For at faa byte"0" med
0040 0C8C 1A       RUTI: LD A,(DE)
0041 0C8D EDA1     CPI        ;Samm. (HL) og (DE)
0042 0C8F E2AD0C   JP PO,SLUT ;Hvis BC=0 til NAS-SYS
0043 0C92 C4980C   CALL NZ,FEJL ;Hvis forsk. TIL 'FEJL'
0044 0C95 13       INC DE
0045 0C96 18F4     ITUR: JR RUTI ;Gentag
0046 0C98 2B       FEJL: DEC HL  ;HL er INC ved sammenl.
0047 0C99 C5       PUSH BC    ;Gem byte-counter
0048 0C9A DF6C     SCAL TX1   ;Udskriv adresser
0049 0C9C DF69     SCAL SPACE
0050 0C9E DF69     SCAL SPACE
0051 0CA0 7E       LD A,(HL)   ;1.mem. indhold
0052 0CA1 DF68     SCAL B2HEX  ;(HL) til VDU
0053 0CA3 DF69     SCAL SPACE
0054 0CA5 1A       LD A,(DE)   ;2.mem. indhold
0055 0CA6 DF68     SCAL B2HEX  ;(DE) til VDU
0056 0CA8 23       INC HL
0057 0CA9 DF6A     SCAL CRLF
0058 0CAB C1       POP BC     ;Hent byte-counter
0059 0CAC C9       LJEF: RET
0060 0CAD DF5B     SLUT: SCAL MRET
0061 0CAF         END

```

Du kan nu prøve at Single-steppe programmet igennem, altså

0C8E XX). (et punktum fører os tilbage til

>_ og så

S C80 nl.

Reg.:	SP.	PC.	AF	HL	DE	BC
	1000	0C83	XXXX	0D00	XXXX	XXXX
	1000	0C86	XXXX	0D00	0D01	XXXX
	1000	0C89	XXXX	0D00	0D01	0100
	1000	0C8B	XXXX	0D00	0D01	0100

nl.

nl.

nl.

tilsynsladende

er der ikke sket noget, men vi ser jo kun på registrene, og i memory er sket det, vi bad om

LD (HL),0 du kan prøve

>T D00 D08

nl.

0D00 00 XX XX XX XX XX XX XX

nu: >S

nl.

1000 0C8B XXXX 0D01 0D02 00FF

nl.

1000 0C8B XXXX 0D02 0D03 00FE osv. nu kan vi se

at HL DE - registrene tæller op og BC ned, og samtidig

ved vi at 00 lægges i den adr. DE peger på fra den adr.

HL peger på, prøv at single-steppe til:

1000 0C8B XX0D 0D04 0D05 00FC

og så:

>T D00 D08

nl.

0D00 00 00 00 00 00 XX XX XX

og så

>E

nl.

Prøv nu alt andet Re-Set, INTET SKER.

Det skyldes ordren 76 (HALT), og man ser den er effektiv.

Nu! RS. og tabuler D00 til E08. og få en bekræftelse.

Føler du ikke en glæde over nu at have så megen magt over maskinen, at du nemt selv kan bestemme, hvad der skal stå hvor.

Jeg vil herefter sammenstille indholdet af nascoms PROGRAMMING MANUAL (herefter kaldet NPM.) og nascoms Software Notes (herefter kaldet NSN.)

Samtidig vil jeg prøve at sammenstille de oplysninger, jeg sidder inde med, vedrørende systemer, samt kommentere med supplerende viden, jeg finder nyttig. Endvidere vil jeg udelade det, som jeg har erfaret mere forvirrer end gavner, derfor vil denne forklaring (fordanskning) ikke være fuldstændig, og skulle nogen finde anledning til ^{at} komme med yderligere oplysninger vil de blive bragt senere.

Vi har tidligere bragt skemaet over Z 80 kommandoer og tager nu udgangspunkt i nogle af disse nemlig

C7	CF	D7	DF	E7	EF	F7	FF	(Object code)
el. RST 0	8	10	18	20	28	30	38	(Source statement)

idet disse ordrer vil medføre at CPU's PC vil indtage de respektive adresser, således vil maskinkode 1100 0111BCD disse to bytes på binær form eller på hexform C7 eller i assembler sprog: RST 0 altid få Z 80 CPU'en til at stille sin programtæller til at pege på og modtage ordren gemt i adr. 0000.

Alle Z 80 baserede computere må altså have sit operativsystem indrettet således at skal ^{altså} begynde fra en af disse adresser, specielt gælder at ReSet ordren starter CPU'en op med at begynde ved adresse 0.

Specielt for Nascom's operativsystemer (T 2.- B. bug. - T 4 og Nassys gælder at vi ikke behøver at skrive foranstående nuller i hverken data eller kommandoer. Således vil assemblerordre: RST 8 (eller maskinkode: CF (binært 1100 1111)) altid medføre at PC antager værdien 0008.

BEGYND PÅ MASKINSPROG. ERIC HANSEN

Hvis vi derimod ønsker at Stack Pointer skal antage værdien 0008 skal skrive:

LD SP,8 som source statement (LD = load er mne. og SP,8 er oprnds) i assemblersprog(assemble betyder på engelsk "at bringe sammen",her maskinsprog/menneskesprog)

Men vi, der ikke har dette program,må håndassemblere, altså slå op i bøger og tabeller,af hvilke i øvrigt kan anbefales "ZILOG Z-80 CPU,Programming Reference Card," der blev tilbudt deltagerne på sidste møde,og som kan rekvireres fra fa.Ditz Schweitzer.

Vi finder at ordren er 31 0,idet vi altså ikke behøver at skrive det foranstående 0 (31 00).

Prøv at tænde for dine apparater,tryk RS

M. nl. (new line) NB. uden adr.
0000 31>_ nl, (ny linie)
0001 33>_
0002 0C>_

og du ser,hvordan RS først sætter PC til 0000, dernæst at dit operativsystem (Nasbug T 2 - B-Bug - T 4) beordrer CPU'en til at lade SP med adr.angivelsen 0C33. Her vil Nassys dog lade SP med adr. 1000.

Hvis vi selv i et program ønsker at indskrive samme ordre,så:

M C80 nl
0C80 XX>31 nl
0C81 XX>_ og her skriver du så LAVESTE byte først, og det vil ikke have nogen mening at prøve at bruge adr.0C33,da den jo bruges af operativsystemet.(Monitoren)

altså:

0C81 XX>8 nl. (laveste byte)
0C82 XX>D nl. (højeste byte)
0C83>. nl. . fører os tilbage til parat.
>_ Hvor vi udelader 0
>MC80 nl.
0C80 31> nl.
0C81 08> nl.
0C8? 0D>_ punktum og nl.
>S080 nl.

0D08 0C83 XXXX XXXX XXXX XXXX XX XXXX XXXX Flags.

SP. PC. A F H L D E B C I IX IY Register.

Det kan måske være en god ide at skrive registernavnene på et stykke papir,ligesom de fremkommer på skærmen,og lime dette på for neden,så kan man bedre koncentrere sig om funktionerne.

Det skal anføres,at T2 monitoren ikke viser I, IX, IY registerene og ej heller Flags ved bogstaver, men kun i F - registerform.

Det er nu naturligt at gennemgå,hvilke funktioner vi har i vort operativsystem,og her har T 4 monitoren jo en facilitet,der giver den lejlighed til selv at fortælle dette:

>? nl.
A B C D E G I K L M N O Q R S T W X Z ?
>_

Af disse kender T 2 ejere kun B (sæt breakpoint) C (Copy, kopier) D (Dump) E (Eksekuer) M (Modifier/inspicer) S (Single step,enkeltrins gennemgang af program) T (Tabuler.

Så dem tager vi først.

access time den tid det tager at få information ud af en memory.

Accumulator

En μP 's vigtigste register, idet alle aritmetiske, logiske og skifteoperationer sker via akkumulatoren. Den er ofte både operand- og resultatregister.

A/D converter (eng. analog/digital converter) anvendes til at omsætte et analogt signal til et digitalt signal, f.eks. en jævnspænding til et digitalt signal, således at computeren kan forstå signalet. Den »modsatte« enhed er en D/A converter.

address et digitalt tal der angiver en lagerplads i en memory.

alfanumeric en kode der indeholder cifrene 0 til 9, alfabetets bogstaver og visse adskillede tegn.

algoritme en bestemt veldefineret samling af regler eller metoder til løsning af et bestemt problem. En algoritme læses ind i computeren i form af en række instruktioner.

ALU, Arithmetic Logic Unit den del af computeren der udfører de grundlæggende operationer som addition, subtraktion, multiplikation og division.

analog medfører at en fysisk størrelse kan repræsenteres således, at enhver værdi hos størrelsen har en tilsvarende værdi hos den repræsenterende variable.

ASCII American Standard Code for Information Interchange. En standardiseret kode der anvendes meget inden for datatransmission. Koden indeholder 128 små og store bogstaver, cifre og visse andre symbolsymboler . . . alle kodet med hver sit 7 bit binære tal.

assembly language et programmeringssprog med programmeringsvenlige memosymboler (mnemonics = lette at huske), f.eks. betyder ADD addere, SUB subtrahere. Programmer kodet i assembler oversættes ved hjælp af en såkaldt assembler til maskinkode, som computeren kan forstå.

assembler et computerprogram der oversætter et program skrevet i symbolic assembler sprog til maskinkode (binærkode).

asynkron asynkron medfører, at en operation, der er afsluttet, starter sin efterfølger. Modsætningen er *synkron kommunikation*, der skal styres af en taktgenerator (clock).

B

Basic Beginners All-purpose Symbolic Instruction Code. et procedureorienteret programmeringssprog, der bygger på engelske ord og forkortelser. Sproget anvendes meget i homecomputerverdenen og passer bedst i systemer med såkaldte interaktive terminaler, dvs. operatøren konverserer med computeren.

baud signalhastigheden måles i enheden baud, der er lig med antal bit pr. sekund.

BCD Binary Coded Decimal, de binære ækvivalenter til cifrene 0 to 9, består af bit grupper på hver fire bit.

benchmark et testprogram der anvendes til at prøve og sammenligne forskellige computere med hensyn til hastighed, og hvor lette de er at programmere osv.

billedskærm se under CRT.

binær eng (binary) et talsystem med kun to cifre 0 og 1 dvs. grundtallet er to. Computere arbejder i virkeligheden kun med *binære* tal.

bit betyder binary digit. Den mindste enhed i en computer. Otte bit = en byte (ord).

bit parallel en metode til samtidig overførsel af samtlige bit i en bitgruppe, dette sker over parallelle ledninger, en for hver bit.

bit serial medfører at man sekventielt overfører bitsene i en bitgruppe, en efter en, over en eneste ledning.

bootstrap en kort programrutine der indlæses i computeren samtidig med opstart af denne; bootstrapinstruktionerne fortæller computeren, hvor den skal kigge efter data, og hvad den skal gøre med disse.

BS

Back Space.

En kommando, der sletter forrige karakter.

bubble memory en type hukommelse der, trods små dimensioner har stor kapacitet.

bug en fejl i computeren eller i programmet.

bus et antal ledninger, der er kommunikationsvejen for data, adresser og styresignaler mellem forskellige enheder (processor, memory, osv.) i en computer.

byte (udtales bajt) en gruppe af sædvanligvis otte bits der behandles som en enhed.

C

chip indholdet i en transistor eller integreret kreds, men ordet chip anvendes ofte for hele den integrerede kreds.

clock en elektrisk pulsgenerator, der synkroniserer alle signaler i en computer.

compatibel et udtryk siges at være kompatibel med et andet, hvis det i visse henseender er udført med samme standard. Termen anvendes også på programmer for at udtrykke muligheden for at køre programmet i et andet, end det er beregnet til.

computer et system, der kan modtage data og uden menneskelig indgriben udføre sædvanligvis komplicerede behandlinger af disse samt levere resultatet på ønsket form. En computer består af CPU memory samt I/O enheder.

CMOS Complementary Metal Oxide Semiconductor en familie af digital kredse, der kendetegnes ved lavt strømforbrug, høj kompleksitet, men også af begrænset hastighed og følsomhed for statisk elektricitet.

CPU Central Processing Unit.

CR

Carriage Return.

Vogn retur.

CRT Cathode Ray Tube. Hermed menes normalt en videodisplay.

cursor er på skærmen en lille lysende firkant eller streg, der viser, hvor næste tegn skal skrives.

Cycle

Et vist antal states udgør en cycle.

Antallet af states i de enkelte cycle's er forskelligt, afhængig af instruktions-typen.

D

D/A digital til analog.

data repræsentation af fakta eller ideer på en formaliseret måde, der kan bearbejdes eller overføres af mennesker eller maskiner.

debugging opdage og rette fejl i soft- eller hardware.

Decrementere

Tælle een ned; formindske med een.

Antonym: incrementere.

DEL

Delete.

En kommando, der sletter alle tidligere indtastede karakterer.

Disable

Forhindre, gøre utjenestedygtig.

Antonym: enable.

DMA Direct Memory Access er en måde at overføre data på direkte mellem en periferenhed og computerens memory - uden indblanding fra CPU'en. Det er en måde at forøge hastigheden på, og dermed systemets effektivitet.

DIP Dual Inline Package navnet på en af de mest anvendte indkapslingsformer til integrerede kredse.

disk storage (plade lager) en metode til hurtig lagring af store mængder programmer og data. Mediet består af en magnetisk plade. Data skrives og læses ved hjælp af et skrive/læsehoved.

display en enhed der kan vise tegn.

DOS Disk Operating System, et program der gør det muligt at skrive og læse på en disk.

duplex en forbindelse beregnet til at overføre signaler i begge retninger.

dynamic memory er en type halvledermemory, hvor tilstedeværelsen eller fraværet af en ladning i en lille kondensator repræsenterer logisk status i den binære memory-celle. Den dynamiske celle skal »opfriskes« periodisk.

E

editor et program, der anvendes til at redigere andre programmer med, dvs. revidere, tilpasse og korrigere eller komplettere dem, for at de virker som forventet.

edit redigere, forberede data.

Enable

Sætte i stand til.

Antonym: disable

EPROM

Erasable Programmable Read Only Memory.
En hukommelse, der programmeres (normalt udenfor brugerkredsløbet) og herefter kun er tilgængeligt for udlæsning af data. EPROM kan - som betegnelsen antyder - slettes, hvilket sker ved belysning med ultraviolette stråler.

Execute at køre et program i computeren.

F

FET Field effect transistor en type transistor der ikke kræver styrestrøm som en almindelig transistor (unipolar i stedet for bipolar).

fetch (hente) betyder CPU'ens hentning af instruktion fra memory'en.

file (register) en samling data der i en henseende kan betragtes som komplet og derfor behandles samlet.

firmware et program lagret i en ROM.

floppy disk (diskette) en massmemory i form af en blød magnetplade indesluttet i et paphylster. Findes i to størrelser standard og mini.

format beskrivelse af datastrukturen.

Fortran Formula Translator et højniveau-sprog beregnet til videnskabelige beregninger.

H

halvleder hukommelse (eng: semiconductor store) en memory der består af aktive bistabile halvlederelementer. Primærhukommelsen i en micro- eller minicomputer er en halvleder memory.

handshaking, styresignaler der gør det muligt for to elektroniske kredse at synkronisere deres arbejde.

hard copy udskrift af data på papir.

hardware er de mekaniske dele i en computer i modsætning til software, der er programmerne.

hexadecimal et talsystem der har grundtallet 16. Det består af tallene 1 til 9 og bogstaverne A, B, C, D, E, F.

højniveausprog (eng: high level language) et programmeringssprog der stort set er uafhængig af den computer, i hvilken det anvendes. Det mest anvendte sprog er Basic.

High Z

Højimpedanset tilstand i modsætning til de kendte "high"-og "low"-niveauer. Bruges i bussystemer, således at flere kredse benytter samme ledningsnet til dataoverførsel.

KRYDSREFERANCE.

Det følgende basicprogram udskriver alle benyttede variable, samt de linienr. hvori de optræder. Det er meningen at programmet skal listet til bånd og derfra indspilles oven i det værende program. Starten sker ved "RUN 60000" og man besvarer spørgsmålet med det maximale linienr. Hvis man ikke vil have det udskrevet på en printer, der bruger USER-rutinen skal følgende linier slettes: 60001, 60002, 60025, 60026, 60045 og 60054. Linie 60000 og 60005 skal tilføjes tal, så man ikke får OS error, det kan betale sig at gøre tallet i linie 60000 så stort som muligt, da bemærkningen i manualen Appendix B punkt C.5 virkelig kan studeres ved skiftende størelser af tal-værdier i 60000.

Sorteringsmetoden der er anvendt er: Shell-Metznersort, der er den næsthurtigste med et antal gennemløb på ca. (antal poster) opløftet til 1,2.

Asbjørn Lind

```

59994 REM *****
59995 REM * Krydsreferance rutine til NASCOM *
59996 REM * ROM-basic. Ideen er taget fra et *
59997 REM * APPEL-program. Udvidet og tilpas.*
59998 REM * NASCOM: Asbjørn Lind 28.12.80 *
59999 REM *****
60000 CLEAR 20000:REM JO STØRRE JO HURTIGERE
60001 GOSUB60025:REM PRINTER ON
60002 INPUT "HØJESTE LINIENR. (60000)";HLN
60003 GOSUB60026:REM PRINTER OFF
60004 NST$=""
60005 DIM ST$(200)
60006 NL=4346:BT=NL
60007 NL=DEEK(BT):IF NL=0 THEN60034
60008 BT=BT+2:ALN=DEEK(BT):BT=BT+1
60009 IF ALN<0 THEN ALN=ALN+65536
60010 IF ALN>HLN THEN60034
60011 VAR$="":GOSUB60017
60012 GOSUB60024:GOSUB60021
60013 IF KA>64 AND KA<91 THEN60012
60014 IF (KA>47 AND KA<58) OR KA=36 THEN60012
60015 IF KA=40 THEN GOSUB60024
60016 GOSUB60027:GOTO60011
60017 GOSUB60021
60018 IF KA=34 THEN GOSUB60032:GOTO60017
60019 IF KA<65 OR KA>90 THEN60017
60020 RETURN
60021 BT=BT+1:IF BT=NLTHEN60007
60022 KA=PEEK(BT):IFKA=142THENBT=NL:GOTO60007
60023 RETURN
60024 VAR$=VAR$+CHR$(KA):RETURN
60025 DOKE4100,1743:DUM=USR(0):RETURN
60026 DOKE4100,1848:DUM=USR(0):RETURN
60027 PRINT ALN,VAR$
60028 AV=AV+1:ST$=VAR$:ALN$=STR$(ALN)
60029 ST$=LEFT$(ST$+NST$,5)
60030 ST$(AV)=ST$+RIGHT$(NST$+ALN$,5)
60031 RETURN

```



```

60032 GOSUB60021:IFKA<>34 THEN60032
60033 RETURN
60034 S1=AV:S2=AV:PRINT"SORTERER - VENT ∇∇"
60035 S2=INT(S2/2):IF S2=0 THEN60044
60036 S3=S1-S2:S4=1
60037 S5=S4
60038 S6=S5+S2:IF ST$(S5)<=ST$(S6) THEN60042
60039 TST$=ST$(S6):ST$(S6)=ST$(S5)
60040 ST$(S5)=TST$
60041 S5=S5-S2:IF S5>0 THEN60038
60042 S4=S4+1:IF S4>S3 THEN60035
60043 GOTO60037
60044 WI=64:REM WI=46 UDEN PRINTER (ren VDU)
60045 GOSUB60025
60046 FOR I=1 TO AV+1
60047 A$=LEFT$(ST$(I-1),4):B$=LEFT$(ST$(I),4)
60048 IFA$<>B$THENPRINTS$:S$=ST$(I):GOTO60053
60049 S$=S$+RIGHT$(ST$(I),6)
60050 IF LEN(S$)<=WI THEN60053
60051 PRINT LEFT$(S$,WI)
60052 S$=RIGHT$(NST$+RIGHT$(S$,6),10)
60053 NEXT
60054 GOSUB60026
    
```

HØJESTE LINIENR. (60000)? 60055

A\$	60047	60048						
ALN	60008	60009	60009	60009	60010	60027	60028	
ALN\$	60028	60030						
AV	60028	60028	60030	60034	60034	60046		
B\$	60047	60048						
BT	60006	60007	60008	60008	60008	60008	60008	
	60021	60021	60021	60022	60022			
DUM	60025	60026						
HLN	60002	60010						
I	60046	60047	60047	60048	60049			
KA	60013	60013	60014	60014	60014	60015	60018	
	60019	60019	60022	60022	60024	60032		
NL	60006	60006	60007	60007	60021	60022		
NST\$	60004	60029	60030	60052				
S\$	60048	60048	60049	60049	60050	60051	60052	
	60052							
S1	60034	60036						
S2	60034	60035	60035	60035	60036	60038	60041	
S3	60036	60042						
S4	60036	60037	60042	60042	60042			
S5	60037	60038	60038	60039	60040	60041	60041	
	60041							
S6	60038	60038	60039	60039				
ST\$	60028	60029	60029	60030				
ST\$(60005	60030	60038	60038	60039	60039	60039	
	60040	60047	60047	60048	60049			
TST\$	60039	60040						
VAR\$	60011	60024	60024	60027	60028			
WI	60044	60050	60051					

NL : Næste liniestart i RAM
 NST\$: Nulstreng
 S\$: Hjelpevariabel
 S1-S6: Sorteringsvariable
 ST\$: De variable før array
 ST\$(: Array af variable
 TST\$: Midlertidig streng ved sort.
 VAR\$: Variabel som streng
 WI : WIDTH i forbindelse med PRINT

 A\$: Hjelpevariabel
 ALN : Aktuel linie nummer
 ALN\$: som strengvariabel
 AV : Antal variable
 B\$: Hjelpevariabel
 BT : Byte tæller
 DUM : Dummy variabel til USR
 HLN : Højeste linienummer
 I : Tæller
 KA : Karakter i ASCII-værdi

Hvis du er interesseret i tips, skal du vente til mandag aften for at få præmiestørelsen på dine gevinstrækker! Med dette program får du en yderst god vurdering af antal rigtige og pengebeløbet til hver gruppe. Det eneste du skal er at indtaste det antal rækker, du anser, der er tippet for i denne uge, samt procentsatsen for det tegn, der kommer på tipskuponen. Denne sats offentliggøres i de fleste dagblade om lørdagen.

Se skema 1 for udregningsmetoden og se det gennemprøvede eksempel på programkørsel.

A.

ASB. LIND

EFFFD

Ok

RUN

CA. ANTAL MILL. TIPPEDE RAEKKER? 40

- 1 . KAMP ? 29.7
- 2 . KAMP ? 23
- 3 . KAMP ? 42.3
- 4 . KAMP ? 23.9
- 5 . KAMP ? 20.6
- 6 . KAMP ? 77.8
- 7 . KAMP ? 21.5
- 8 . KAMP ? 58.3
- 9 . KAMP ? 29.1
- 10 . KAMP ? 29
- 11 . KAMP ? 50.2
- 12 . KAMP ? 30.7
- 13 . KAMP ? 55

GEVINSTSTØRELSE
FOR 40 MILL. RAEKKER

RIGTIGE	ANTAL	BELØB
13	39.6	51136
12	1097	1844
11	13705	147
10	101828	28

Ok

	(0) A	(1) B	(2) C	(3) D	(4) E	(5) F
1	pro	1/A1 -1				
2	pro	1/A2 -1	B1	C2 *B2		
3	pro	1/A3 -1	C2 +B2	C3 *B3	D2	B3 *E3
4	pro	1/A4 -1	C3 +B3	C4 *B4	E3 +D3	B4 *E4
5	pro	1/A5 -1	C4 +B4	C5 *B5	E4 +D4	B5 *E5
6	pro	1/A6 -1	C5 +B5	C6 *B6	E5 +D5	B6 *E6
7	pro	1/A7 -1	C6 +B6	C7 *B7	E6 +D6	B7 *E7
8	pro	1/A8 -1	C7 +B7	C8 *B8	E7 +D7	B8 *E8
9	pro	1/A9 -1	C8 +B8	C9 *B9	E8 +D8	B9 *E9
10	pro	1/A10-1	C9 +B9	C10*B10	E9 +D9	B10*E10
11	pro	1/A11-1	C10+B10	C11*B11	E10+D10	B11*E11
12	pro	1/A12-1	C11+B11	C12*B12	E11+D11	B12*E12
13	pro	1/A13-1	C12+B12	C13*B13	E12+D12	B13*E13
14			C13+B13		E13+D13	

skema 1

LIST

```
100 REM *****
110 REM * Beregning af tipspraemierne *
120 REM * hvis procentfordeling er kendt *
130 REM * Ide:Kurt Nielsen (Tipsbladet) *
140 REM * 22.12.80 Asbjørn Lind *
150 REM *****
160 DIM T(14,5)
170 CLS
180 INPUT"CA. ANTAL MILL. TIPPEDE RAEKKER";R$
190 IF VAL(R$)<10 THEN170
200 R=VAL(R$)*1E6:T=R
210 REM **** INPUT 13 KAMPE ****
220 FOR A=1 TO 13
230 PRINT A;
240 INPUT ". KAMP ";T$
250 IF VAL(T$)<1 OR VAL(T$)>99 OR T$="" THEN230
260 T(A,0)=VAL(T$)/100
270 NEXT
280 REM **** BEREGNING AF MATRIX ****
290 T(0,0)=1:T(14,0)=1
300 FOR B=1 TO 14
310 T(B,1)=1/T(B,0)-1
320 T(B,2)=T(B-1,2)+T(B-1,1)
330 T(B,3)=T(B,2)*T(B,1)
340 T(B,4)=T(B-1,4)+T(B-1,3)
350 T(B,5)=T(B,1)*T(B,4)
360 NEXT
370 REM **** BEREGNING AF 13'ERE ****
380 FOR C=1 TO 13
390 T=INT(T*T(C,0)*10)/10
400 NEXT
410 G=INT(R/2*.45*.225/T)
420 REM **** BEREGNING AF 12'ERE ****
430 U=INT(T*T(14,2))
440 H=INT(G/T(14,2))
450 REM **** BEREGNING AF 11'ERE ****
460 V=INT(T*T(14,4))
470 I=INT(G/T(14,4))
480 REM **** BEREGNING AF 10'ERE ****
490 FOR D=3 TO 13
500 S=S+T(D,5)
510 NEXT
520 X=INT(T*S)
530 J=INT(G/S*.325/.225)
540 REM **** UDSKRIFT ****
550 CLS
560 PRINTTAB(15)"GEVINSTSTORELSE"
570 PRINTTAB(14)"FOR";VAL(R$);"MILL. RAEKKER"
580 PRINT:PRINT
590 PRINT"RIGTIGE","ANTAL","BELOB"
600 PRINT" 13";:Z=T:D=15:GOSUB650:Z=G:D=32:GOSUB650:PRINT
610 PRINT" 12";:Z=U:D=15:GOSUB650:Z=H:D=32:GOSUB650:PRINT
620 PRINT" 11";:Z=V:D=15:GOSUB650:Z=I:D=32:GOSUB650:PRINT
630 PRINT" 10";:Z=X:D=15:GOSUB650:Z=J:D=32:GOSUB650:PRINT
640 END
650 PRINTTAB(D-INT(LOG(Z)*.4343+1))Z;:RETURN
Ok
```

LIST

```
100 REM *****
110 REM * Cirkeltegningsrutine med *
120 REM * DEMO-program *
130 REM * af Asbjørn Lind dec. 80 *
140 REM *****
150 CLS
160 PI=3.14159:FY=1.53
170 PRINT"CIRKELTEGNINGSPROGRAM"
180 INPUT "AUTOMATISK (J/N)";J$
190 IF J$="J" GOTO420
200 PRINT"SKAERM ER OPDELT X-RETNING TIL HOJRE";
210 PRINT" 95 ENHEDER"
220 PRINT"Y-RETNING NEDAD 66 ENHEDER"
230 PRINT"BEGYNDELSESPUNKT OVERST TIL VENSTRE"
240 PRINT"ALLE STEDER I KOORDINATSYSTEMET ";
250 PRINT"KAN BRUGES SOM CENTRUM f.eks. -25,-1"
260 INPUT"ANTAL CIRKLER - MAX 10";N
270 FOR I= 1 TO N
280 INPUT"RADIUS, centrum X,Y";R(I),X(I),Y(I)
290 NEXT I
300 CLS
310 FOR I= 1 TO N
320 FOR A=0 TO2*PI+.5STEP PI/3/R(I)
330 SX=INT((X(I)+SIN(A)*R(I))+.5)
340 IF ABS(SX-47.5)>47.5 GOTO210
350 SY=INT((Y(I)/FY-R(I)*COS(A)/FY)+.5)
360 IF ABS(SY-21.5)>21.5 GOTO380
370 SET(SX,SY)
380 NEXT A
390 NEXT I
400 SCREEN 1,1
410 END
420 FOR N=1 TO 9
430 READ R(N),X(N),Y(N)
440 NEXT
450 CLS
460 N=N-1
470 SCREEN 21,11:PRINT"( DAVS ∇ )"
480 GOTO310
490 DATA 1.1,48,38
500 DATA 22,48,33
510 DATA 24,48,78
520 DATA 24,48,-19
530 DATA 28,48,-19
540 DATA 6,38,26
550 DATA 6,58,26
560 DATA 9,18,22
570 DATA 9,78,22
580 DATA 40,48,33
Ok
```


ASB. LIND

EFFFFD

Ok

LIST

```

100 REM *****
110 REM *   Decimal til binær omsætter   *
120 REM *         af Jørn Jensen         *
130 REM *****
140 CLS:INPUT"Indtast et tal mindre end 256";A
150 D=A
160 FOR B=7 TO 0 STEP -1
170 B(7-B)=2↑B
180 NEXT
190 CLS:FOR P=0 TO 7
200 D=D-B(P)
210 IF D>=0 THEN C(P)=1:GOTO230
220 C(P)=0:D=D+B(P)
230 NEXT
240 PRINT:PRINT:PRINT:PRINT:PRINT:PRINT
250 PRINTTAB(4);B(0);TAB(10);B(1);TAB(15);B(2);
260 PRINTTAB(20);B(3);TAB(25);B(4);TAB(30);B(5);
270 PRINTTAB(35);B(6);TAB(40);B(7)
280 PRINT
290 PRINTTAB(5);C(0);TAB(11);C(1);TAB(16);C(2);
300 PRINTTAB(21);C(3);TAB(25);C(4);TAB(30);C(5);
310 PRINTTAB(35);C(6);TAB(40);C(7)
320 PRINT:PRINT"          ";A;"="
330 PRINT:PRINT:PRINT:INPUT  "Flere tal ";A$
340 IF LEFT$(A$,1)="J" THEN140
Ok

```

RUN

Indtast et tal mindre end 256? 123

128	64	32	16	8	4	2	1
0	1	1	1	1	0	1	1

123 =

Flere tal ? NEJ

Ok

Anmeldelse af: BITS og P.C.'s TOOL KITT.

Toolkittet er en udvidelse til den eksisterende ROM-basic. Den forøger antallet af kommandoer med 9 + nogle specielle rutiner.

Følgende kommandoer findes:

```
APPEND
AUTO LINE NUMBER
DELETE
DUMP
FIND
HELP
HEX
RENUMBER
STEP
```

I betragtning af at jeg anmeldte POLYSYS i sidste nr. af Nascom Nyt, vil jeg her kun fremhæve forskelle og ligheder.

Af ens fungerende kommandoer er der: AUTO (genererer selv nyt linienr. efter hvert 'ENTER'), DELETE (sletter et bestemt antal linier), HEX (oversætter fra decimal til hexadecimal) og RENUMBER (renumererer linienumre).

Af forskelligt virkende kommandoer findes: APPEND. Funktionen af rutinernerne er den samme, men den forbrugte tid dertil er så sandelig forskellig. Hvad POLYSYS er sekunder om er TOOLKIT minutterom (for store programmer (8K) er det mellem 5 og 8 minutter). TOOLKIT's programmør er dog godt klar over problemet og kommer i manualen med en tåget forklaring, der ikke rigtig har bærekraft!!

DUMP lister alle variable på en gang i modsætning til PS, hvor man jo kan vælge almindelige variable eller strengvariable. Listningen kommer væltende og kan ikke programmeres til et bestemt linieantal.

STEP har en god facilitet, som ikke findes hos PS. Man kan få udskrevet en streng på øverste linie, samt få udskrevet værdien af op til tre variable på samme linie. Det vil sige, at man under step-afviklingen kan følge det aktuelle linienr. og se, hvordan tre variable ændre sig under kørslen.

FIND kan skifte mellem søgning i alle ord eller kun i ikke kommandoord. Det skift foregår ved at taste kontrol-B. Dette fungerer godt.

Så findes der en meget nyttig kommando:HELP, der har to funktioner. For det første vil HELP vise den fejlbehæftede linie ved 'error', samt vise hvor i linien fejlen ligger (cursor blinker på eller lige ved siden af fejlen). For det andet vil HELP nnnn vise denne linie nederst og ved hjælp af kontrol-B kan man opskrive alle kommandoerne til enkelt tegn!! Det vil sige, at man kan presse lange linier sammen, så man undgår at gå over XO-mode, og stadig bibeholde NAS-SYS redigeringsmuligheder.

Af ikke kommandoer findes der INKEY og RINK, der giver en simpel metode til at undersøge tastaturet til tryk eller kun en gang. Man skal inden brug af nævnte rutiner DOKE til 4100.

Derudover er der repiterende tastatur og en PRINT HANDSHAKE, der tester PORT 0, bit 7 under XO-mode.

Asbjørn Lind (30.12.80)

Der indkaldes til BESTYRELSESMØDE mandag den 12.1.81 kl. 19 (Herlev)

Dagsorden:

1) Punkter fra sidste møde:

- a) Anden service,
EPROM-brænder, hvordan
går det med den?
(Erik + Ole)

- b) Printer ; do
(Søren)

- c) MDC 2000

- d) Fælles kassetteinterface

2) Copyright

Referat af Ole og Asbjørns
møde hos KODA.

- 3) Næste medlemsmøder
f.eks. om Interrupt!

- 4) Eventuelt

Asbjørn Lind 30.12.30

Fraværende: Erik hansen

Referat:

Pkt 1a) Der er lavet et optryk af
Nascom Nyt 1980, de kan købes så
længe oplag haves for 10 Kr. pr.
stykk (1,3,4,5,6,7 og 8). De skal
bestilles hos Ole Hasselbalch.

Vedtog at starte et artikelbibli-
otek hos Erik Hansen. Nærmere op-
lysning i et kommende nr.

Ole taler med Ole Stender om et
moderne diagram til en EPROM-bræn-
der.

Pkt 1b) Man vedtog at spørge med-
lemmerne vha spørgeskema, der ud-
sendes til medlemmer.

Pkt 1c) Projekt opgivet indtil vi-
dere af forhandler!

Pkt 1d) Vedtager Kansas City in-
terface (Nascom II-format) i for-
bindelse med til og fra forening.
Ole lover at producere et inter-
face til Nascom I.

Pkt 2) Vi orienterede den sam-
lede bestyrelse om mødet hos
KODA.

Pkt 3) Vi vedtog at placere mø-
de den første søndag i hver
måned, startende med søndag den
1. februar kl 12.30 Pædagogisk
Central, Rustenborgvej, Lyngby.
Emne: Opbygning af Nascom Com-
puter'S KBD.

Pkt 4) Jesper meddelte at prisen
på kopier er steget til 1 kr./ark.



Anmeldelse af MATH48.

For dem der kan lide at arbejde på M/C eller assemblerniveau, har Poly-Data Aps. lanceret en matematikpakke, MATH48.

MATH48 er skrevet for Z-80 baserede systemer, dermed også for Nascom-1 og -2.

Pakken er opdelt i 25 egentlige rutiner, der kan give ens egne M/C eller assemblerprogrammer, en regnenøjagtighed og -hurtighed, der langt overgår talbehandlingen i Nascom 8K Basic.

Rutinerne kan kort opdeles i følgende funktioner:

- 1) De 4 regnearter: plus, minus, gange og divider
- 2) Potensoplyftning og kvadratrodsuddragning
- 3) Logaritme- og eksponentialfunktionen
- 4) Trigonometriske- og deres inverse funktioner
- 5) Modulus, heltals- og decimaltalsuddragning
- 6) Konvertering til og fra 16bit heltal
- 7) Konvertering fra teksstreg til flydende tal
- 8) Formatteret udskrift i fastkomma eller eksponential notation

Regnenøjagtigheden er ca 11 betydende cifre, og rækkevidden fra $2.938735877E-39$ til $1.701411835E+38$.

MATH48 er ikke beregnet på at køre sammen med Nascom Basic. Derfor er man ikke ringere stillet mht. at læse fra og til MATH48. Konverteringsrutinerne sørger selv for at oversætte et ASCII-tal til flydende tal i binær representation og tilbage igen. Det er dog muligt selv at håndkonverterer, så kald af disse rutiner kan spares.

MATH48 fylder $2\frac{1}{2}$ Kbytes, og kræver et arbejdslager på kun 45 bytes, der kan placeres hvor i lageret det ønskes, via SP-registret.

Betjeningen er faktisk meget enkel.

Ens programmer bliver en samling CALL til MATH48 adresser, hvor den kaldte funktion selv tager hånd om tallet i akkumulatoren. Ved brug af programmet, omdannes 6 af 16bit registrene, BC, DE og HL samt deres "skygger" BC', DE', og HL' til 2 flydende talakkumulato-
torer, hver med 40bit mantisse og 8bit eksponent.

"ASCII" læsning til og fra talakkumulato-
torerne, forgår ved at benytte IX-registret som pegepind.

Der skal bruges 2 talakkumulato-
torer, da de funktioner som MATH48

omhandler, indeholder enten 1 eller 2 argumenter.

F.eks vil

\sqrt{A} , $\log A$, $\cos A$ kun indeholde 1 argument (A)

medens: $A \times B$, $A + B$, A/B indeholde 2 argumenter (A og B)

Den regnemæssige behandling i de 2 talakkumulatorer, er kort beskrevet i den medfølgende manual. Den vil ikke blive gennemgået her, da den ikke er nødvendig for at bruge MATH48; men vil blive gennemgået i en senere artikel.

Udover manualen, medfølger en kommenteret assemblerlistning, så der er mulighed for at studere opbygningen af programmet nøjere. Sidst følger et kommenteret programeksempel, der kan give en ide om brugen af MATH48 rutinerne. Log skal værdien af AOUT rettes fra 0003H til 0030H; ellers er ens skærm meget tom under programafviklingen.

Selve manualen omtaler kort de enkelte funktioner og tilgangen til disse, samt test af regneoperationen f.eks: Over- underflow.

Vedbrug opfattes ret hurtigt arkitekturen i MATH48, og det føles som om Z-80 registrene kun består af 1 flagregister, 1 super talakkumulator og dets "skygge" samt IX-registret.

MATH48 behandler kun tal, der allerede er i registrene, og derved vil læsning til og fra skærmen normalt involverer NAS-SYS rutiner. Der kan da opstå den pinlige programmeringsfejl, at en NAS-SYS rutine "sletter" ens tal, hvis ikke det er gemt på stakken, inden en NAS-SYS rutine kaldes. Derfor check NAS-SYS manualen.

Anvendelsesmulighederne for MATH48 vil være mange indenfor beregningsopgaver.

En nærliggende ide, vil være en skærmorienteret regnemaskine, så man kan lave sin egen TI- eller HP regner.

Efter oplysning fra Poly-Data indgår MATH48 i deres PASCALversion, hvorved de der ikke kan lide at arbejde på assemblerniveau, kan ty til BLS PASCAL.

Jørn N. Sundby

Almindelige oplysninger om foreningen

BESTYRELSENS SAMMENSETNING:

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
Vibeskranten 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 kasserer
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.