

# NASCOM 280 NYT

NASCOM BRUGERGRUPPE  
2730 Herlev

Sidevolden 23  
Giro 6742602

NR: 6  
2. årgang

JUNI 1981.

Hermed sidste nummer inden sommerferien. Der vedlægges et giroindbetalingskort til medlemmer med numre mellem 1 og 147 incl. Beløbet bedes indbetalt omkring den 1.7.81. Medlemmerne med numre fra 148 og op har betalt for perioden 1.7.81 til 1.7.82, mens interesserede med "II" foran navnet bedes indbetale på allerede tilsendte girokort, hvis de fortsat vil modtage noget fra foreningen.

Flere bestyrelsesmedlemmer holder sommerferie i Juli måned, så derfor er der nok ikke den største aktivitet fra vores side. Men skulle du en regnvejrsdag alligevel ikke have noget at foretage dig, må du gerne tænke på os med hensyn til programmer og indlæg til bladet.

Vi starter op med medlemsaktiviteter omkring 1. september, men nærmere oplysninger i næste nummer af Nascom Nyt, som jeg regner med udkommer mellem den 15 og 20 august.

Med venlig hilsen og god sommerferie

si'r

ASBJØRN.

## INDHOLD

side 2	Bufferkort til N1
side 6	Teknisk brevkasse
side 8	Overskriftprogram (objektkode)
side 9	Softwarespalten
side 14	Subrutine til toner (basic/assembler)
side 16	Nyheder til programbiblioteket
side 18	Rubiks magiske terning
side 20	Medlemsannoncer
side 21	meddelelser om programbiblioteket
side 22	Ekstra til side 2-5. Afkodning af I/O-porte
side 23	EPR0M programmer fra 5 og 5.2
side 24	Othello (objektkode)
side 25	Sletning af EPR0M
side 26	Foreningsoplysninger

Hjemmelaget bufferkort til NASCOM 1.

Af Ole Brandt

Et blik på diagrammet af NASCOM's originale bufferkort, var nok for mig til at beslutte, at sådan et måtte kunne laves både smartere og billigere.

På diagrammet (fig. 1) ses hvordan.

DIRECT MEMORY ACCESS.

Alle udgående kontrolsignaler fra CPU'en (minus HALT og  $\overline{\text{BUSAK}}$ ), samt alle adresse- og dataledninger TRISTATES hvis  $\overline{\text{BUSAK}}$  går lav. Herved opnås mulighed for at køre DMA.

MEMEXT.

Når NASCOM 1 skal arbejde over 4k, skal MEMEXT/INT linken på printet flyttes til EXT, og så forventer NASCOM'men at der kommer sådan et signal. Det skal være LO hvis adresserne ligger mellem 0- og 4k, ellers HI.

Sådan et signal findes på det originale NASCOM RAM-kort, men hvis man (som jeg) har tænkt sig at udvide med uoriginale kort (for eks. COMET 2000 kort), skal signalet lige laves, som vist i diagrammet.

STYRING AF DATABUFFER.

Databufferen må kun kunne skrive tilbage til NASCOM'men, såfremt adressen er over 4k, derfor bruges MEMEXT signalet også til at TRISTATE databufferen med.

Ved  $\overline{\text{IORQ}}$  ligger CPU'en adressen på I/O enheden på nederste halvdel af ADRBUS (A0 til A7). samtidig med, at de data der skal overføres ligger på den øverste halvdel af ADRBUS (A8 til A15).

Dette kan få MEMEXT signalet, og dermed TRISTATE signalet til databufferen, til at forsvinde. Det er årsagen til, at  $\overline{\text{IORQ}}$  er gated med i TRISTATE kredsløbet til databufferen.

Ovenstående umuliggør tilslutning af ekstra I/O porte efter bufferen. - Ønsker man mulighed for dette, kan man bruge opstillingen i fig. 2 til at styre databufferen med i stedet for, men læg mærke til, at IOEXT signalet skal forefindes først. - (Det ville være

rart, hvis en eller anden ville skrive et par ord om hvordan NASCOM har lavet IOEXT signalet).

#### MEKANISK OPBYGNING.

Jeg kører selv via ca. 15 cm fladkabel fra NASCOM'men til indgangen af bufferkortet, men da jeg har en fornemmelse af, at dette er en af årsagerne til, at det blev nødvendigt med de modstande der går fra indgangene til stel, vil jeg anbefale at man opbygger bufferen i umiddelbar nærhed af stikket på NASCOM'men. (Da vil disse modstande sandsyneligvis kunne spares. Det samme gælder kondensatoren fra MRQ til stel).

Selve kortet har jeg opbygget på et stykke hulprint v.h.a. lodbar laktråd, men da jeg har haft visse problemer med kortslutninger mellem disse, vil jeg anbefale at man monterer v.h.a. wire-wrap tråd eller lign.

Udgangen af mit bufferkort går direkte ind i et EURO-kort backplane, hvor der sidder et 32k RAM kort, modificeret til 48k (bytes forstås), samt et 2.12kbytes EPROM kort af eget design.

Bemærkninger til Morten Kolbæk's EPROM programmer program.

Af Ole Brandt.

I Mortens program køres PIO'ens port A i byte mode. Det kræver HANDSHAKE/INTERRUPT styring, hvorfor det ikke ville virke hos mig.

Løsningen er, at køre port A i BIT mode ligesom port B.

Det betyder at følgende skal ændres:

OC91	LD	A,OCFH	og	OE67	LD	A,OCFH
	OUT	(DPORTK),A			OUT	(DPORTK),A
	LD	A,OFFH			XOR	A
	OUT	(DPORTK),A			OUT	(DPORTK),A

Begge steder indskydes 4 bytes ekstra, hvilket man selvfølgelig skal tage hensyn til hvis man håndassemblerer.

-----  
Hvis man anvender et andet program, vil programeren også kunne programmere 4kbytes EPROM's af typen 2732 (TEXAS 2532).

Så vidt jeg kan se, kan man styre alle nødvendige signaler til sådan en også.

NASCOM-1

PL 1

- 1 DB0
- 2 DB1
- 3 DB2
- 4 DB3
- 5 DB4
- 6 DB5
- 7 DB6
- 8 DB7

- 25 RD
- 26 WR
- 27 MRQ
- 29 TORQ
- 30 RFSH
- 28 FI

31 HALT

38  $\emptyset$

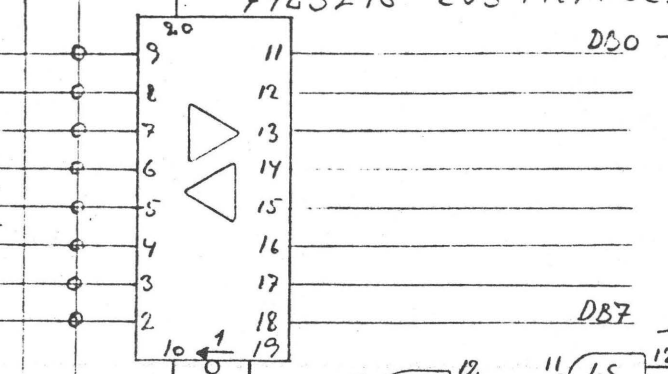
- 21 A0
- 23 A1
- 24 A2
- 22 A3
- 19 A4
- 18 A5
- 17 A6
- 16 A7

- 15 A8
- 1 A9
- 14 A10
- 13 A11
- 9 A12
- 10 A13
- 11 A14
- 20 A15

74LS245 BUS TRANSCEIVER

+5V

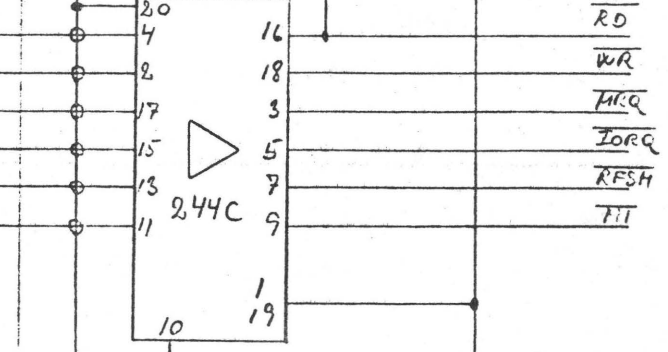
+5V



BUFFED DATA BUS

74LS244

+5V



BUFFED CONTROL BUS

+5V

+5V

1K $\Omega$

1K $\Omega$

LS 00

LS 00

BUSAK

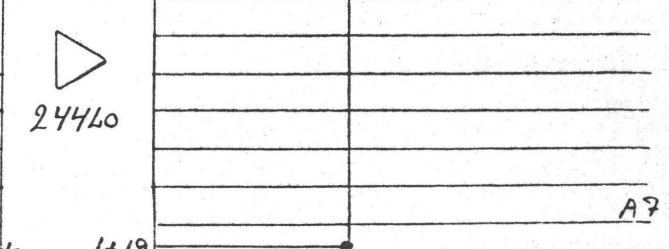
BUSAK

32A

32A

74LS244

+5V



BUFFED ADRESSE BUS

74LS244

+5V

+5V

1K $\Omega$

1K $\Omega$

LS 00

LS 00

BUSAK

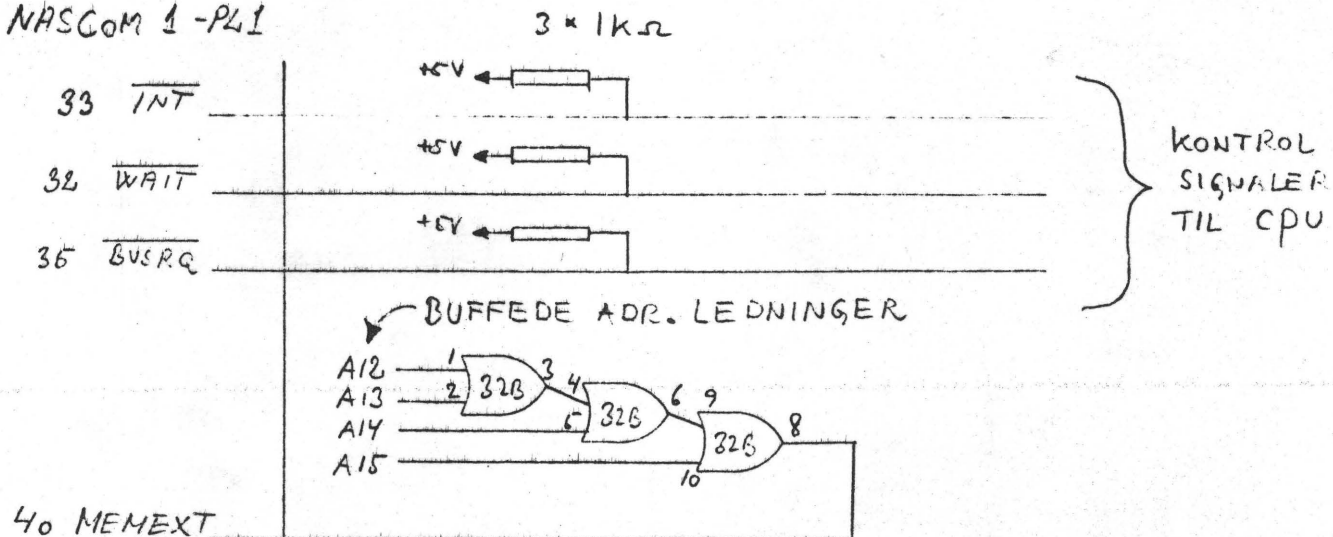
BUSAK

244L0

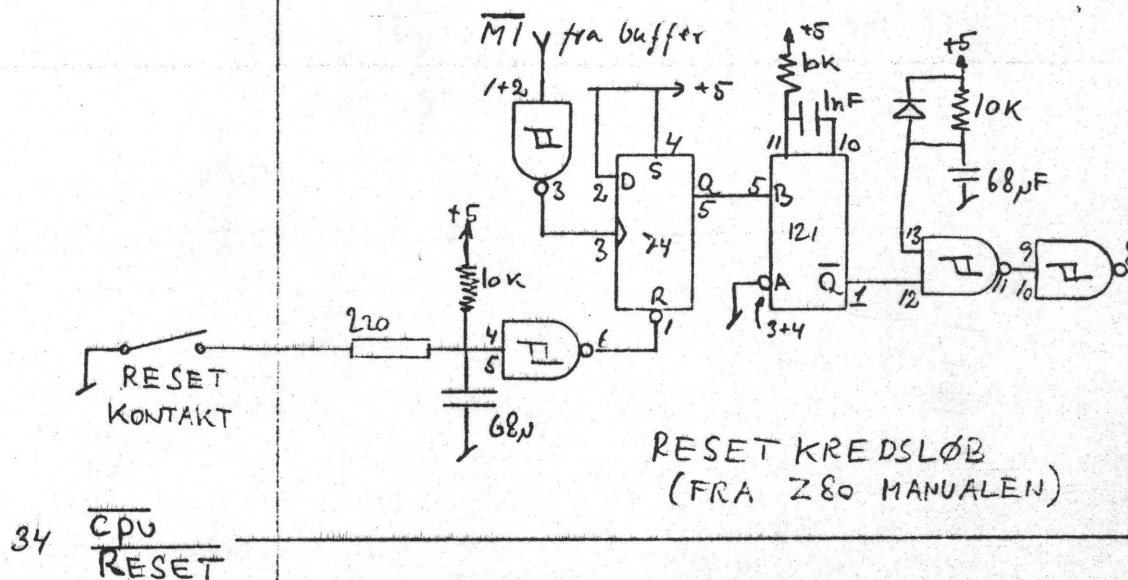
244H1

FIG 1 side 1

NASCOM 1-PL1

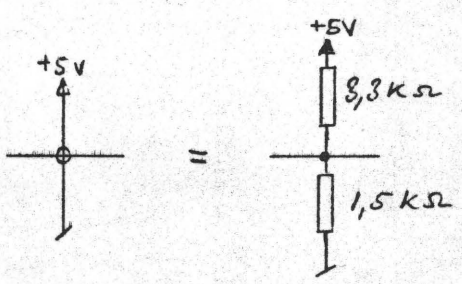


40 MEMEXT



34 CPU RESET

+5V p. 41  
GND p. 42+43



- |      |           |                     |
|------|-----------|---------------------|
| LS00 | = 74LS00  | +5V p. 14, GND p. 7 |
| LS32 | = 74LS32  | -  -, -  -          |
| 132  | = 74LS132 | -  -, -  -          |
| 74   | = 74LS74  | -  -, -  -          |
| 121  | = 74LS121 | -  -, -  -          |

BUFFER TIL NASCOM 1  
028V6

FIG 1 side 2.

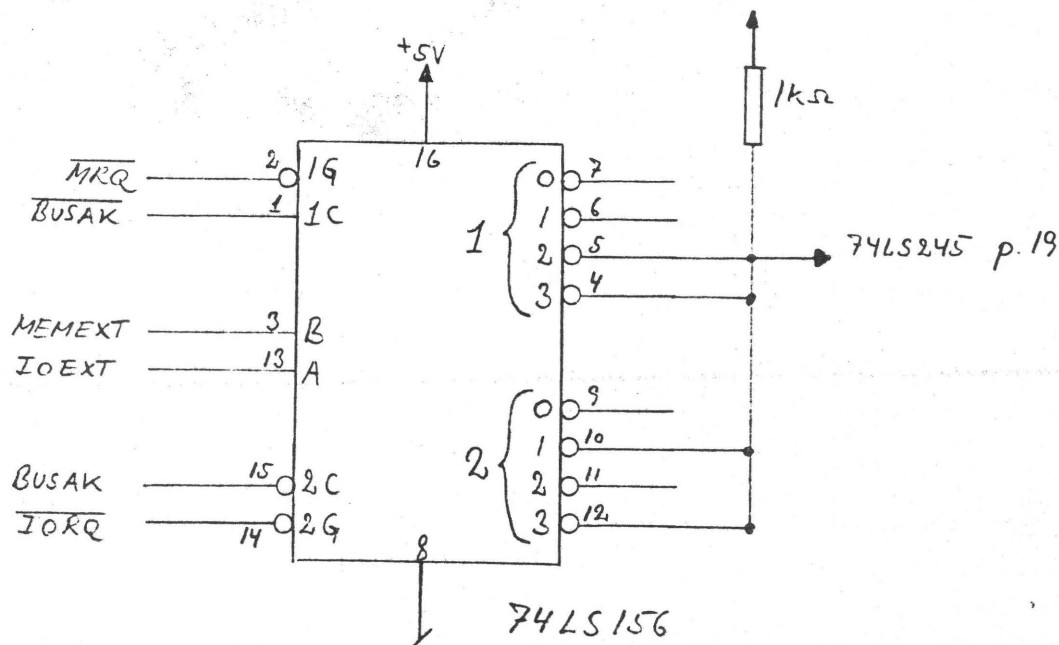


FIG. 2 TRISTATE STYRING AF DATA BUFFER  
MED I/O MULIGHED.

TEKNISK BREVKASSE Ved Christen Fihl

Hej brevkasse

Vil du hjælpe mig med min VDU ram? I punktet OBD3 skifter alle tegn fra små til store bogstaver. Det sker øjeblikkeligt, når jeg POKER til denne memory adresse. Det ser ud til at bit 5 i denne adresse er sat. Kan du fortælle mig hvilken memorykreds det drejer sig om? (Det er på en Nascom I).

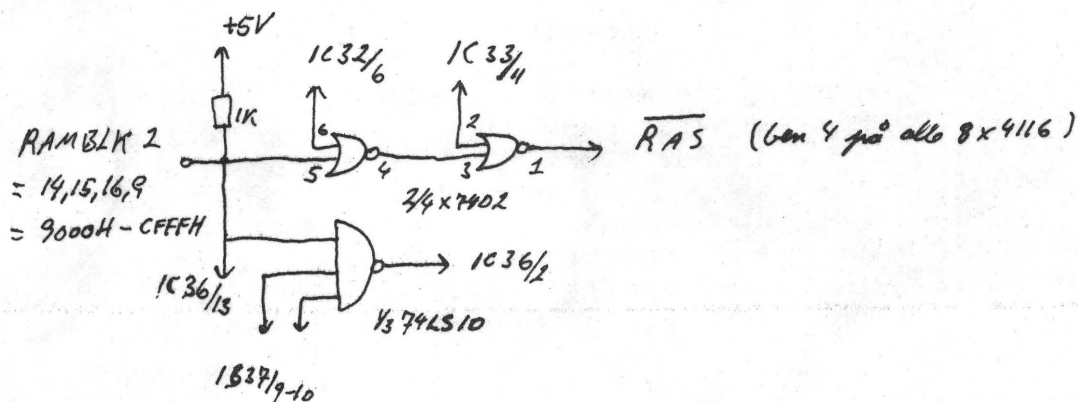
Svar:

Bit nr 5 gemmes i IC 26, og fejlen tyder mest af alt på, at det er kredsen, der er defekt. Da du sikkert ikke skal læse fra adresse OBD3, der ligger i øverste linje, kan du bytte kredsen om med IC 20, der gemmer på bit 7. Der vil da stå det rigtige på skærmen, forudsat at bit 7 ikke bliver brugt til styring af grafiktegnset eller lignende.

Hvordan får man 48 Kbytes på et RAM-A kort?

Svar:

Studerer man diagrammet over ramkortet, ser man, at ramkredse 4116 to og to er parallel forbundet, bortset fra ben 4 RAS indgangen. Der skal derfor laves et signal, der kan styre dette ben på alle 8 nye kredse.



Dette kredsløb kan bruges til styring af RAS'en . Der skal foruden de 8x4116 bruges en 7402 og en 74LS10 (eller 7410).

#### Montering:

Printbanen mellem ben 13 og ben 14 på IC 36 fjernes og printbanen fra ben 8 på IC 37 skæres over. Benene 6,7 og 14 på 7402'eren og ben 7,9,10 og 14 på 7410'eren bukkes rundt, så den tynde del af benet peger opad langs den tykke del. Ben 3 og 4 på 7402 loddes sammen. De to IC'er monteres nu på bagsiden af printet med ryggen mod printet, således at 7402'eren monteres under IC 32 og 7410'eren monteres under IC 37, idet ben 1 er ud for ben 1. Man lodder nu de ombukkede ben fast til de tilsvarende ben på printet, mens resten skal kontrolleres ikke at røre benene på printet. Resten af forbindelserne trækkes mellem kredse og printet. Ben 8 på 7410'eren skal f.eks. til det gennem pletterings hul som før havde forbindelse med ben 8 på IC 37. Nu mangler blot at få monteret de 8 ram kredse. Disse kan monteres på mange forskellige måder. Det eneste krav er, at ben 2 og 14 på een kreds skal tilsluttes til ben 2 og 14 på samme kreds på printet.

#### Montering af de 8 ram kredse :

Benene på 8 IC-sokler bukkes en smule udad og ben 4 og 14 bukkes helt opad langs siden på soklerne. De syv af soklerne loddes nu på undersiden af printet, således at soklens ben 1-8 loddes til ben 1-8 på f.eks. IC 12, mens ben 9-16 loddes til ben 9-16 på IC 13, (bortset fra ben 4 og 14 på soklen). Ben 14 på soklen forbindes med ben 14 på samme IC på printet som ben 4 er tilsluttet (her IC 12). Den sidste sokkel bliver man nødt til at montere mellem IC 10 og IC 11, hvor det nu er ben 4 og 2 der bukkes helt op ad siden på soklen. Her er det ben 2 der skal forbindes med samme kreds som ben 14. Til slut forbindes alle ben 4 med RAS udgangen fra før (7402 ben 1).

## Overskriftprogram.

af Christen Fihl

Dette program gemmer programmer på bånd med indspillet overskrift. Det kan bruges til både maskinkode- og basicprogrammer. Det fylder fra C80 til EA4, hvortil der skal lægges en tekstbuffer på op til 48 bytes.

Programmet startes fra C80 og kan varmstartes fra C82 (det husker da forrige valg og overskrift). Programmet kan frit relokeres, men husk at der skal være plads til max. 48 bytes efter sidste adresse.

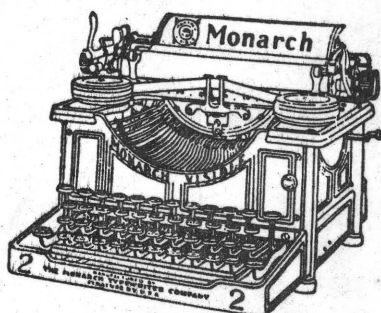
Programmet er selvforklarende!

```

OC80 18 19 D7 08 2A 06 08 11 8B FE 19 E9 D7 00 E1 11
OC90 16 02 19 22 06 08 FD 21 08 08 C9 D7 EF FD 36 00
OCA0 00 EF 18 42 61 73 69 63 2F 4D 61 73 6B 69 6E 20
OCB0 3F 20 28 42 2F 4D 29 20 00 DF 7B F7 CB AF FE 4D
OCC0 20 08 2A 06 08 11 5B FF 19 E9 FE 42 20 CD FD CB
OCD0 00 C6 FD CB 00 CE EF 61 73 69 63 00 EF 18 4E 61
OCE0 76 6E 20 3A 20 00 DF 7B F7 32 09 08 EF 18 52 75
OCF0 6E 20 3F 20 00 DF 7B F7 CB AF FE 4A 20 04 FD CB
OD00 00 DE EF 18 54 69 74 65 6C 20 3A 0D 00 DF 63 06
OD10 31 1A FE 20 20 03 13 10 F8 2A 06 08 44 4D 77 23
OD20 13 B7 28 09 FE 20 28 02 44 4D 1A 18 F1 03 02 DF
OD30 4E DF 5F EF 3C 54 52 59 4B 3E 00 DF 61 30 FC EF
OD40 1B 00 DF 5D 2A 73 0C 11 FB FF 19 22 73 0C AF F7
OD50 F7 EF 0C 00 F7 F7 F7 2A 06 08 06 31 7E B7 28 03
OD60 23 10 F9 CB 38 04 05 28 05 3E 20 F7 10 FB 2A 06
OD70 08 7E F7 B7 28 03 23 10 F8 FD CB 00 46 28 42 EF
OD80 18 1B 43 6C 6F 61 64 20 22 00 3A 09 08 F7 EF 22
OD90 0D D3 D3 D3 00 3A 09 08 F7 DF 6A 21 D6 10 22 0C
ODA0 0C 2A D6 10 22 0E 0C DF 60 DF 5F DF 57 DF 5F FD
ODB0 CB 00 5E 28 06 EF 52 75 6E 0D 00 DF 5F DF 4E DF
ODC0 5B 2A 00 08 22 0C 0C 2A 02 08 22 0E 0C EF 18 1B
ODD0 52 0D 00 FD CB 00 4E 28 0B EF D3 D3 D3 00 3A 09
ODE0 08 F7 DF 6A DF 60 DF 5F DF 57 DF 5F FD CB 00 56
ODF0 28 C9 EF 1B 45 00 2A 04 08 DF 66 DF 6A 18 BC EF
OE00 61 73 6B 69 6E 00 EF 18 46 72 61 20 3A 20 00 D7
OE10 4D 38 F3 22 00 08 EF 18 54 69 6C 20 3A 20 00 D7
OE20 3D 38 F3 22 02 0B EF 18 45 78 65 63 75 74 65 20
OE30 3F 20 28 4A 2F 4E 29 20 00 DF 7B F7 CB AF FE 4A
OE40 20 29 EF 18 46 72 61 20 3A 20 00 D7 11 38 F3 22
OE50 04 08 FD CB 00 D6 2A 06 08 11 5E FE 19 E9 DF 63
OE60 06 05 13 10 FD DF 64 2A 21 0C C9 FE 4E 20 B7 EF
OE70 18 42 61 73 69 63 20 6E 61 76 6E 20 3F 20 28 4A
OE80 2F 4E 29 20 00 DF 7B F7 CB AF FE 4A 20 C8 EF 18
OE90 4E 61 76 6E 20 3A 20 00 DF 7B F7 32 09 08 FD CB
OEAO 00 CE 18 B2

```





## SOFTWARESPALTEN

af Anders Hejlsberg.

For fremtiden vil jeg efter bedste evne prøve at besvare softwarespørgsmål via NASCOM NYT. Alt, lige fra BASIC til maskinkode har interesse, så skriv endelig hvis du har et problem, et godt tips eller gerne vil have et enme uddybet.

Som det første "indlæg" har jeg skrevet om nogle af de ting jeg i tidens løb har disassembleret mig frem til omkring NASCOM's ROM BASIC.

## Fakta om NASCOM ROM BASIC:

## BASIC'ENS ARBEJDSLAGER:

Lad det være sagt med det samme: Denne tabel beskriver ikke alle adresser i arbejdslageret og er ikke med garanti rigtig. Imidlertid vil jeg mene at de fleste af de ting der eventuelt kan være af interesse er med.

Adrs	Ant	Indhold	Betydning
1003	3	C3 xx xx	Hop til USR-rutine.
1006	3	D3 xx C9	Rutine til OUTput på port nummer xx. Den bliver kaldt af OUT kommandoen der først retter xx.
103E	3	DB xx C9	Rutine til INPut fra port nummer xx. Den bliver kaldt af INP kommandoen der først retter xx.
1041	1		Antal nuller (OOH) der skal udskrives efter hvert CR (ODH). Rettes af NULL kommandoen. Værdien har et offset på 1. Således vil kommandoen NULL 1 gøre at ingen nuller udskrives efter CR.
1042	1		Udskriftslinielængde. Rettes af WIDTH kommandoen. Hvis denne byte sættes til 255 bliver WIDTH funktionen fjernet.
1045	1		Output disable flag. Hvis man under programkørsel retter denne byte til en værdi forskellig fra nul vil PRINT sætninger ikke udskrive noget.
1048	2		Antal linier der skal LISTes ad gangen. Rettes af LINES kommandoen.
104D	1		Singlestepflag. Hvis denne byte gøres forskellig fra nul vil BASIC kalde BLINK rutinen i NAS-SYS efter endt udførsel af den næste kommando. Samtidig bliver der igen lagt et nul ud i byten.
104E	3	C3 xx xx	Hop til INPUT rutine. Rutinen skal returnere med HL pegende til adressen umiddelbart før den første byte af en tekstbuffer. Teksten skal være afsluttet med et nul (OOH) og B skal indeholde tekstens længde.

1051	3	C3 xx xx	Hop til SET rutine. Ved indgangen peger HL på den byte i programlinien der efterfølger SET kommandobyten. Rutinen skal returnere med HL pegende på den næste byte i programlinien (typisk 00H eller 3AH=":").
1054	3	C3 xx xx	Hop til RESET rutine. Parametrene er de samme som ved SET.
105A	2		Adresse på "bunden" af stringospace. Ved kommandoen CLEAR xx sættes denne adresse til adressen i 10FAH minus xx.
105C	2		Nummeret på den linie der i øjeblikket udføres. OFFFFH indikerer at der arbejdes i direct mode.
105E	2		Programmets startadresse (normalt 10FAH).
1060	74		Tekstbuffer.
10AB	1		Cursorens position på linien. Indholdet af denne byte returneres ved kald af POS.
10AD	1		Typeflag. Hvis indholdet er 0 indeholder 10E4H-10E7H et floating point tal. Hvis indholdet er 1 indeholder 10E4-10E5 adressen på en beskriverblok for en strengværdi.
10AF	2		Memory size.
10C3	2		Stringospace stackpointer. Peger til den næste fri byte i stringospace.
10D6	2		Programmets slutadresse og startadresse for lager til enkeltvariable.
10DB	2		Slutadresse for lager til enkeltvariable og startadresse for lager til arrayvariable.
10DA	2		Slutadresse for lager til arrayvariable.
10E4	4		Floating point akkumulator.
10E9	13		Buffer til ASCII repræsentation af tal.
10FA			Programmer starter her.

#### FORMAT AF PROGRAMLINIER:

Programlinier starter i 10FAH. En linie har følgende format:

```
bb aa mm nn xx xx xx xx ..... xx xx 00
```

hvor aabb er adressen på den efterfølgende linie, nmm er liniens nummer og xx'erne er de bytes der udgør linien. En linie er altid afsluttet med nul. Den sidste linie i et program består udelukkende af bb og aa, der i dette tilfælde begge er nul.

#### FORMAT AF VARIABLE:

Der er som bekendt to variabeltyper i BASIC: Talvariable og strenge. En "beskriver" til hver af disse værdier fylder fire bytes.

Beskriver til numeriske værdier:

En numerisk værdi repræsenteres ved en 24 bits mantisse og en 8 bits 2's exponent. Beskriveren er opbygget på følgende måde:

```
byte 0-2: Mantisse
byte 3: Exponent
```

Mantissen gemmes med mindst betydende byte i den laveste adresse. Exponenten er et binært tal med et offset på 80H. Således svarer en exponent på 84H til, at mantissen skal multipliceres med  $2^{(84H-80H)}=2^4=16$ . Hvis exponentens værdi er 00H opfattes hele tallet som værende nul. Værdien af mantissen kan opfattes som det fortegnsløse heltal, bestående af de 3 første bytes, divideret med  $2^{24}$ . Mantissen er altid normaliseret, det vil sige at den mest betydende bit skal altid opfattes som værende sat. Fortegnet gemmes i den mest betydende bit: Er den sat er tallet negativt, er den nul er tallet positivt.

Beskriver til strenge:

En strengbeskriver har følgende format:

byte 0: Længde  
 byte 1: Bruges ikke  
 byte 2-3: Adresse på første tegn i strengen.

Når en strengvariabel tilskrives en værdi bliver string-space stackpointeren (10C3-10C4) flyttet ned i lageret og værdien bliver gemt i de reserverede bytes. På denne måde optager en streng aldrig flere bytes end dens aktuelle længde. Imidlertid vil der efterhånden akkumuleret sig en del "strengaffald", dvs. værdier der førhen var tilskrevet en variabel, men nu er ubrugte, da variabelen er blevet tilskrevet en ny værdi. Når string-space stackpointeren efterhånden når bunden af string-space (denne adresse står i 105A-105B) vil BASIC'en inlede en oprydningsskema, der, hvis string-space er stort, kan tage en pæn tid. Det er dette der er grunden til de mærkelige pauser der pludselig kommer under udførsel af et program med mange streng-operationer.

FORMAT AF VARIABELLAGER:

Variabellageret er opdelt i to lejre: Enkeltvariabellageret, der strækker sig fra (10D6-10D7) til (10D8-10D9), og arrayvariabellageret, der strækker sig fra (10D8-10D9) til (10DA-10DB).

Enkeltvariabellageret:

En enkeltvariabel optager 6 bytes lager, og kan være en ud af tre muligheder:

1) Numeriske variable.

byte 0: Andet tegn i variabelnavn.  
 byte 1: Første tegn i variabelnavn.  
 byte 2-5: Floating point værdi.

2) Strengvariable.

byte 0: Andet tegn i variabelnavn med bit 7 sat (OR'et med 80H).  
 byte 1: Første tegn i variabelnavn.  
 byte 2-5: Strengbeskriver.

3) Funktionsvariable (FN'er).

byte 0: Andet tegn i funktionsnavn.  
 byte 1: Første tegn i funktionsnavn med bit 7 sat.  
 byte 2-3: Adresse på funktionsudtryk der skal evalueres ved funktionskaldet.  
 byte 4: Andet tegn i funktionsvariabelnavn.  
 byte 5: Første tegn i funktionsvariabelnavn.

## Arrayvariable:

En arrayvariabel (DIM) gemmes i det følgende format:

Offset	Ant	Betydning
0	1	Andet tegn i variabelnavn. Bit 7 er sat hvis variabelen er et strengarray.
1	1	Første tegn i variabelnavn.
2	2	Offset fra byte nummer 4 til næste array.
4	1	Antal dimensioner (kaldet n).
5	2	Antal elementer i dimension n (kaldet a(n)).
7	2	Antal elementer i dimension n-1 (kaldet a(n-1)).
2n+3	2	Antal elementer i dimension 1 (kaldet a(1)).
2n+5	4	Element xx(0,...,0,0)
2n+9	4	Element xx(0,...,0,1) etc.

Et element er enten en floating point værdi eller en strengbeskriver.

## FLOATING POINT RUTINER:

Floating point rutinerne benytter alle BASIC'ens floating point akkumulator (kaldet FPA) der udgøres af de 4 bytes 10E4-10E7. Rutiner der tager to argumenter (+, -, \*, /, ^) benytter desuden registrene BC og DE (som det første argument). Her udgøres mantissen af CDE (C er mest betydende byte) og exponenten af B. Resultatet placeres altid i FPA.

Adrs	Betydning
F5CD	FPA=EDCB+FPA
F5CA	FPA=EDCB-FPA
F708	FPA=EDCB*FPA
F769	FPA=EDCB/FPA
FAB7	FPA=EDCB^FPA
F822	FPA=SGN(FPA)
F8E6	FPA=INT(FPA)
F838	FPA=ABS(FPA)
FAAC	FPA=SQR(FPA)
FB8B	FPA=RND(FPA)
F6C7	FPA=LOG(FPA)
FAFA	FPA=EXP(FPA)
FC00	FPA=COS(FPA)
FC06	FPA=SIN(FPA)
FC67	FPA=TAN(FPA)
FC7C	FPA=ATN(FPA)

Ved brug af floating point rutinerne fra et maskinkodeprogram bør man sikre sig at der ikke er mulighed for overflow (fx. en division med 0) da BASIC'en således ikke returnerer men i stedet udskriver en fejlmeddelelse og returnerer til direct mode.

## ANDRE BASIC SUBROUTINER:

BASIC'en er naturligvis fuld af subrutiner og denne gennemgang behandler dem der efter min mening kan have interesse.

Kode	Funktion
CALL OE3C1H	Udskriver en fejlmeddelelse og returnerer til direct mode. E indeholder fejllens nummer: 00=NF 02=SN 04=RG 06=OD 08=FC 0A=OV 0C=OM 0E=UL 10=BS 12=DD 14=/O 16=ID 18=TM 1A=OS 1C=LS 1E=ST 20=CN 22=UF 24=MO
CALL OF101H	Konverterer A registeret til floating point format og gemmer resultatet i FPA.
CALL OF0F2H	Konverterer AB registerparret til floating point format og gemmer resultatet i FPA. AB betragtes som værende et 16-bits tal mellem -32768 og 32767 i 2's komplement form.
XOR A LD B,98H CALL OFB2AH	Konverterer DE registerparret til floating point format og gemmer resultatet i FPA. DE betragtes som værende et fortegnsløst heltal.
CALL OE98BH	Konverterer FPA til et 16-bits binært tal og gemmer resultatet i DE.
CALL OED5AH	Evaluerer et udtryk. HL indeholder adressen på det første tegn i programlinien der indgår i udtrykket. Rutinen returnerer med HL pegende på det næste tegn i programlinien. Adressen 10AD indeholder et flag der afgør udtrykkets type: 00H indikerer numerisk, ikke 00H streng. Hvis udtrykket var numerisk er værdien i FPA. Hvis udtrykket var en streng indeholder 10E4-10E5 adressen på en strengbeskriver.
CALL OF988H	Konverterer tallet i FPA til ASCII format. Tekststrengen placeres i ASCII-bufferen, der starter i adresse 10E9. Tekststrengen er afsluttet med 00H.
CALL OE9A5H	Konverterer en talstreng til et 16-bits binært heltal og placerer resultatet i DE. HL peger på det første ciffer i tallet, der skal være i området $0 \leq n \leq 65529$ .

## Annonce

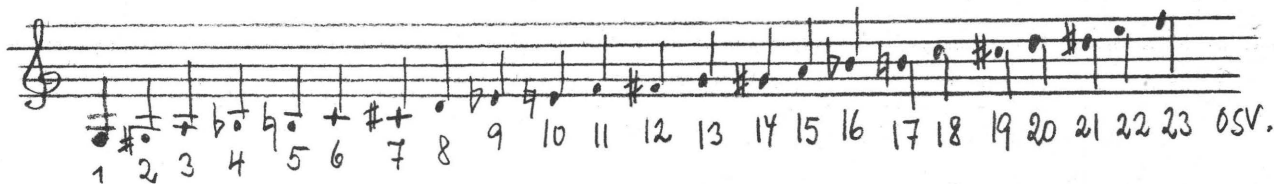
Grundet køb af RAM64 (C-kort), sælges et færdigbygget og perfekt fungerende RAM "A" kort incl. 8 stk. 4116 til højeste bud over 1400 kr.

Henvendelse til  
Jørn Jensen  
Skovbrynet 4  
2800 Lyngby  
02 87 82 57 efter KL. 16

SUBROUTINE FOR TONER

Programmet kan bruges til nemt at indføje korte melodier eller melodistumper i et større program. For eksempel til at angive forskellige dele af programmet, eller angive et rigtigt svar, mens et forkert svar har en anden melodistump. Hvis der er tale om et langt program med mange beregninger, kan subrutinen gøre opmærksom på, når man kan gå videre, osv., osv.

I hovedprogrammet skriver man antallet af toner i melodien, tempoet (i metronomtallet for fjerdedelsnoderne) samt tonehøjde og rytme (dvs. nodeværdi) for hver tone. Tonehøjden fastlægges ved at tildele en værdi, som er nummeret på tonen i en række startende med tonen lille g:

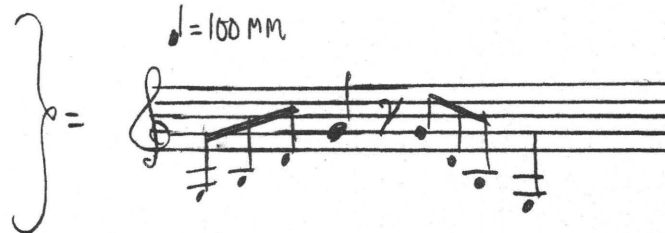


Rytmen er den omvendte værdi af nodeværdien: helnode=1, halvnode=2, ottendedel=8 osv. Pauser kan indlægges ved at udnytte, at de meget høje toner ikke kan høres! F.eks. tone 95 og opefter.

Eksempel på hovedprogram:

```

10 ANTAL=9:TEMPO=100
20 T(1)=1:R(1)=8:T(2)=5:R(2)=8
30 T(3)=8:R(3)=8:T(4)=13:R(4)=4
40 T(5)=95:R(5)=8:T(6)=13:(R(6)=8
50 T(7)=8:R(7)=8:T(8)=5:R(8)=8
60 T(9)=1:R(9)=4
70 GOSUB 10000
80 END
    
```



"Hovedprogrammet" kan naturligvis selv være en subrutine i et større program, sådan at forskellige melodier nemt kan kaldes som subrutiner.

NB.: 1. Hvis der skal anvendes mere end 10 toner må T() og R() dimensioneres.

2. Hvis TEMPO >= 86 er den længste mulige nodeværdi  
 >=172                    - " -  
 >=343                    - " -

Hvis der vælges for lange nodeværdier, forkortes de i subrutinen.

3. Rytmerne angives med flg. værdier:

o = 1		
d = 2	d. = 1.333	$\overset{3}{d} = 3$
♩ = 4	♩. = 2.666	$\overset{3}{\overset{3}{d}} = 6$
♩♩ = 8	♩♩. = 5.333	$\overset{3}{\overset{3}{\overset{3}{d}}} = 12$
♩♩♩ = 16	♩♩♩. = 10.666	$\overset{3}{\overset{3}{\overset{3}{\overset{3}{d}}}} = 24$

o.s.v.

u.a.o. den reziproke værdi af brøkdelen af en hel takt.  
 ( F.eks. d. =  $\frac{7}{8}$ ,  $\frac{8}{7} = 1.142857$  )

Kommentarer til subrutinen:

Linie 10090: Afstanden mellem tempererede toner =  $\sqrt[12]{2} * (\text{forrige tones svingningstal})$ . Da værdien af  $T(TN)$  er omvendt proportional med tonehøjden, benyttes den reciproke værdi,  $1/\sqrt[12]{2} = 0.9438743$ .  $256.9 * 0.9438743 = \text{tonen lille g på computeren.}$

Linie 10100: Med  $R(TN)=240$  i linie 10160 opnås 91 slag i minuttet, altså  $\downarrow = 91$  i metronomangivelse. Da værdien af  $R(TN)$  også er omvendt proportional med tempoet, gælder formelen:  
 $(240 * 91 * 4)/(R(TN) * TEMPO) = 87360/(R(TN) * TEMPO)$   
 hvor TEMPO er metronomtempo for fjerdedele fra hovedprogrammet, og hvor  $R(TN)$  er den reciproke værdi af nodeværdien.

Venlig hilsen Henrik Dyhr

```

10000 REM ***** Tone-subrutine *****
10010 DATA 15065,3328,11879,14848,3329,5711
10020 DATA 16640,31787,10421,4106,31481,4334
10030 DATA -11433,6144,-20496,211,-13863,27227
10040 RESTORE 10010
10050 FOR I=3330 TO 3364 STEP 2
10060 READ J:DOKE I,J
10070 NEXT
10080 FOR TN=1 TO AN
10090 T(TN)=256.9*0.9438743^T(TN)
10100 R(TN)=87360/TE/R(TN)
10110 IF T(TN)>255 THEN T(TN)=255
10120 IF R(TN)>255 THEN R(TN)=255
10130 NEXT TN
10140 DOKE 4100,3330
10150 FOR TN=1 TO AN
10160 POKE 3328,R(TN)
10170 POKE 3329,T(TN)
10180 U=USR(0)
10190 NEXT TN
10200 RETURN
  
```

Programmet benytter sig af en højttaler, der er sat parallelt over readlysdioden. Til almindelig lytteniveau er det udemærket. Tidsforsinkelserne er beregnet til en 4 MHz maskine med et wait state og man kan "liste" subrutinen ind ved at bruge kassettebåndoptageren, som en hurtig maskinskriverske ! (Se et tidligere nr. af NN).

A.

## Nyheder til programbiblioteket.

Jeg har i de forløbne måneder modtaget følgende programmer til programbiblioteket. Hvordan programmerne bliver nummeret og pris-sat kan Jesper oplyse om.

### MORSEPROGRAM af Peter Ø Jensen.

Programmet er beregnet til sending og modtagning af morseprogrammer. Skriv din tekst på skærmen med almindelige bogstaver og send det i morse. Eller modtag en morsesender og få det oversat til almindelig dansk på skærmen. Programmet bruger PIO'en som in og out. Hardwaremodifikationer er kun en buffer på PIO-porten.

### MEMORY sammenligner (rev.4)

En gammel kending i nyt tøj (NN3). Programmet har fået følgende options: oversætter memoryindhold til ASCII eller Decimal, således at der ved forskel udskrives Adr.1, Adr.2, Indhold af 1 og 2 i Hex, ASCII og Decimal. Udskriver 8 linier af gangen og indsætter blank linie før næste 8 linier ved tastning vises. Konstruktor Mogens H. Jørgensen.

Programmet findes i assemblersource beregnet til nassys 1/3. Virkelig godt kommenteret med rutediagram.

### DATABASE.

Også en gammel kending, men med nye rutiner. Programmet kan kun køre under NASSYS 3. Det har længe irriteret mange, at sortering-rutinen altid starter fra "home"-position. Dette har Mogens klaret ved at "rulle" datalinierne i hver tekstsiderundt, så man selv kan vælge denøverste og da sorterer. "Rulle" tilbage og resultatet er at f.eks., alle københavnske medlemmer står efter hinanden lige klar til en udskrift. Palsbo har udvidet programmet med en read og write, der læser data fra bånd bag nuværende data og skriver data til bånd uden styreprogram. Programmet er i assembler.

### DOKE-POKE-liniedanner.

Programmet er udført af Lars Jensen i assembler. Det er tænkt som en hjælp til dem, der skal bruge assemblerrutiner i et basicprogram, og som vil undgå besværet med at konvertere data og adresser fra hex til decimal, og derefter indtaste basic-linier med DOKE-sætninger.

Programmet danner automatisk basiclinier med "DOKE"-instruktioner, adresse og data. Linierne lægges ud sidst i et evt. eksisterende basicprogram med et linespring på 10.

Ved kald af programmet spørges efter en start og slutadresse (i Hex) på den maskinkoderutine, der skal oversættes til DOKE/POKE-sætninger. Derefter finder programmet den sidste basicsætning og sidste linienummer. Der undersøges ikke for overflow i RAM eller linienumre. Ved afslutningen hoppes til Warm start af basic. Programmet er godt kommenteret.



**SORTERING.**

Vi har fra Jørgen Kristensen modtaget en SHELL-METZNER sorteringsrutine i assembler. Programmet kan benyttes i andre programmer til en hurtig sorteringsalgoritme. Programmet er et godt kommenteret.

**DEBUG** af Kim Nygård.

Programmet findes i en udgave til hver monitor NASSYS 1 og 3. Det kører absolut bedst på NASSYS 3, hvor det udnytter, at den er interrupt-sikker. Programmet er skrevet i assembler og fylder i source 18 K, mens objekt-koden kun fylder 1.75 K byte.

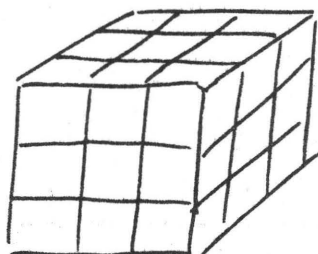
Programmet er specielt udviklet til brug under programudvikling, da man kan lave rammer om PC og SP, så kommandoen overgår til DEBUG, hvis disse rammer overskrides. Det er ligeledes let at se og ændre i alle registre (også skyggereg.), samt at sætte flere breakpoints.

Ved hjælp af TRACE kan man følge programafviklingen med valgbar "menneskelig" hastighed. Det er under dette muligt at ændre og vise reg.

Følgende kommandosæt er til rådighed:

A	Display top af stak
B xxxx,yyyy	Breakpointadresser max 5
C	Nulstiller alle reg, SP=1000
D	Displayer bruger registre
D(RP,R)	Displayer aktuel reg, evt. ændringer indtastes
E xxxx	Start bruger program
L	Initialiser Trace-hastighed
M(A - Z)	Udfører NASSYS-rutiner
O	Test af subrutiner, returadr. på stakken
O-	Sletter O
P	Sæt op til 4 PC-områder
S xxxx	Singelstep
T xxxx	Trace, hvorunder man kan
	A Vis top af stak
	B Break, retur DEBUG
	C Fortsæt
	D Vis registre
	F Hurtigere
	H Halt, stop TRACE
	N Next, udfør næste instruktion
	O Gå til NASSYS
	S Langsommere
V xxxx yyyy	Sæt stak-kontroladresse.
W (M,A,I)	Varm start af NASSYS,NAP,NIP
Z	Ombyt reg. med skyggereg.

*A.*



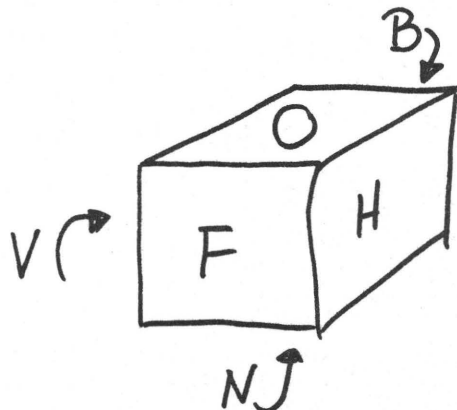
### RUBIKS magiske terning.

Denne lille omtale falder nok lidt uden for Z80's rammer ! ! ! !

Men har man en gang haft en denne terning i hånden, bliver man besat af den. Men avis- og TV reklamernes vås om stærke nerver, skal man som altid tage med et gran salt. Det kan betale sig med lidt systematik - finde de gode drejninger, der ændre uden at ødelægge.

Før jeg løfter sløret for disse gode flytninger, må vi aftale en notationsform, så det bliver let at forklare drejningerne.

Man kan vælge mellem to fra ref 1) eller ref 2). Jeg synes absolut af første er bedst. Den går ud på følgende: Man navngiver terningens sider, efter hvordan man ser dem (herved frigør man sig af farverne på siderne !). F(oran), B(agtil), H(øjre), V(ænstre), O(p), N(ed):

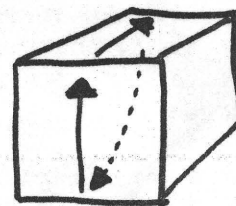
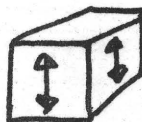


Ved F menes en drejning 1/4 omgang med uret; ved F' menes 1/4 mod uret og F2 betyder så en halv omgang. Hvis flere drejninger efterfølger hinanden, skal man udføre dem i den beskrevne rækkefølge. Gentages et drejningsmønster bruges følgende notation (FH)3 betyder FHFHFH.

De gode flytninger deles i 4 grupper: 1) Kantflytninger, 2) Hjørneflytninger, 3) Kantvendinger og 4) Hjørnedrejninger.

**Kantflytninger**

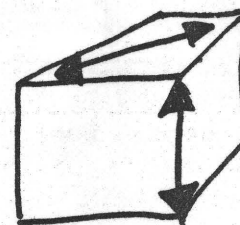
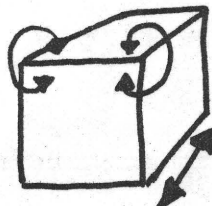
a) ombytning af to par  
(F2H2)3



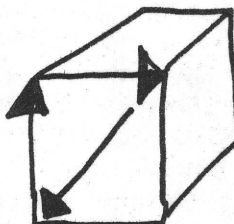
b) Flytning af 3  
O2H'VF2HV'

**Hjørneflytninger**

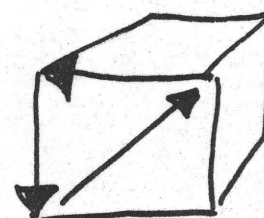
a) ombytning af 2 par  
(FHF'H')3



(FH'F'F)3



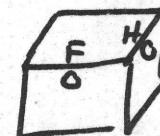
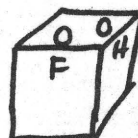
b) flytning af 3  
V'FHF'VFH'F'



O2V2OH0'V2OH'0

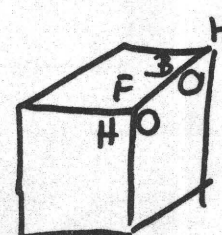
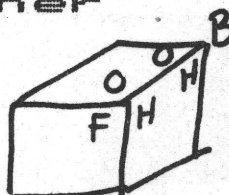
**Vending af kanter**

H' O2H2OH' O' H' O2VFHF' V'



**Drejning af hjørner**

(FNF'N')20 (NFN'F')20'



Angrebstaktik: Det kan betale sig at starte med samme farve hver gang på første side, da man så lærer, hvilke der skal ned i bunden.

Man starter med en endeflade, hvor "krydset" først skal på plads, dernæst hjørnerne. Når dette er klaret, skal man bringe orden i midterringen, hvorefter "krydset" i bunden laves og til sidst vendes hjørnerne på plads.

Hvis alt andet skulle svigte, kan man skille terningen ad på følgende måde: Drej øverste 1/3 1/8 omgang, hvorefter midterterningen presses opad med et fladt instrument.

Hvis det ikke havde noget med vores maskiner at gøre, ville jeg selvfølgelig ikke have taget dette emne op, men da der i flere af ref. nævnes, at det er muligt at dreje sig færdig på 52 kvarte drejninger, kommer sommerens opgave:

## SKRIV ET PROGRAM

der løser dette problem.

### Referanceliste:

H.C. ØRSTED's Kantine publikation ved Mogens Esrom Larsen  
Der Spiegel nr. 4/1981  
David Singmaster: Notes on Rubik's 'Magic Cube'  
Mathematical Science and Computing  
Scientific American 1981/

A.

---

## Efterlysning

Søren efterlyser folk, der også gerne vil købe Philips minidigitalkassette maskine MDCR 220. Søren vil gerne forsøge at lave et fælles indkøb, da prisen falder kraftigt ved selv mindre mængder.

Hvis du er interesseret kontakt Søren (telefonnr. bag i bladet).

---

## SÆLGES monteret i 19" rack med stik og kontakter: NASCOM 1 med

Dobbelt monitor NASSYS1/3  
Grafik Rom  
Udvidet tastatur i blåt kabinet  
Nascom 1 kassetteinterface  
Nascom 2 - - - - (300/600/1200/2400 baud)  
Bufferkort til udvidelse af bussen  
Motherboard med 3 sokler til 8"x8"  
RAM A kort med 32K RAM + sokler til 4K ROM  
EPROM-brænder med software

Dette sælges med eller uden strømforsyning for 6800 kr./6300 kr.

Derudover tilbydes

EPROM-kort med 8K basic og sokler til 16 EPROM  
for 900 kr.

FIRMWARE i 12 2708 EPROM for 800 kr.

SOFTWARE på tape leveres ad libitum !!

Al fornøden dokumentation er tilstede og medleveres.

SÆLGES PÅ GRUND AF OPDATERING TIL STØRRE ANLÆG.

Henvendelse til redaktionen 02 91 71 82

## MEDDELSER FRA PROGRAMBIBLIOTEKET.

Af Jesper Skavin.

### Rettelser til udsendte programmer

Basicprogrammerne i programbiblioteket er alle sammen skrevet til NASCOM ROM BASIC med NASSYS 1 som monitor. Dog er der nogle, som kan anvendes med NASBUG monitoren, idet programmet selv finder ud hvilken monitor, der anvendes. Problemet optræder i forbindelse med maskinkoderrutinerne til scanning af tastaturet. Her er der nemlig stor forskel på NASBUG og NASSYS. Det er dog kun i et enkelt program, jeg har opdaget en fejl og det kun med NASBUG som monitor. Fejlen optræder i STAR TREK (B3), så hvis du kører med NASBUG T2 eller T4, skal følgende rettes:

Linie 100 ændres til:  
100 DATA 27085,14336,-20735,-20665,3370,-5664

To nye linier:  
105 DATA 0,0  
163 CR=13 : IF PEEK(1)<>0 THEN CR=31

Til slut ændres linie 5340 til:  
5340 IF I8<>CR GOTO 5360

RETTELSE TIL "MAGIC LABYRINTH" (B6): Punktummerne i linie 2440 skal rettes til tegnet for potensopløftning (SHIFT/0).

MODIFICERING AF "CHASE Ver 2" (M12): Programmet forudsætter at en PIO dataport er tilsluttet port 4 på Nascom'en og at der står 00 i den ved læsning. Dette krav kan man komme udenom ved at foretage følgende:

1. Indlæs CHASE.
2. Tast: M1673 <enter>
3. Tast: AF 00. (HUSK PUNKTUM)
4. Gem CHASE på bånd ved at taste: W 1400 1800 <enter>.

FORBEDRING AF "MASTERMIND II" (M11). Har man spillet dette program nogle gange, opdager man hurtigt hvordan man kan vinde hver gang. At det er så nemt at vinde skyldes, at programmets random-rutine er temmelig primitiv. Ved at foretage nedenstående ændringer kommer det til at virke meget bedre. Samtidigt rettes nogle skønhedsfejl. Ændringerne udføres med NASSYS "I" kommando og "M" kommando således:

BEMÆRK: PUNKTUMMERNE I PKT. 3,4,5,6 OG 11 SKAL INDFASTES.

- 1 Indlæs MASTERMIND II.
- 2 Tast: I 12BB 12BC 55 <enter>
- 3 Tast: M 12BB efterfulgt af: 22. <enter>
- 4 Tast: M 11BD efterfulgt af: C9. <enter>
- 5 Tast: M 1075 efterfulgt af: D9. <enter>
- 6 Tast: M 1174 efterfulgt af: 93. <enter>
- 7 Tast: M 104D
- 8 Tast på samme linie flg. hex-tal:  
21 11 13 01 07 04 ED 5F 86 38 01 3D 77
- 9 Tast: <enter>
- 10 Indtast hvor cursoren står flg. hex-tal (på en linie):  
91 30 FD 81 3C 77 23 10 FO C9 00
- 11 Tast: . <enter> hvorved M-kommandoen forlades.
- 12 Gem MASTERMIND på bånd med kommandoen: W 1000 1400 <enter>

**GENERELT.**

I er stadig meget velkomne til at sende programmer ind til programbiblioteket. Alt modtages med tak. Hvis programmerne ikke er trykt, så ring og bed om at få et bånd tilsendt til indspilning af dem. Du vil så automatisk modtage en udskrift af indholdet af båndet, ca. 1-2 uger efter du har sendt det retur.

Programmer indsendt inden 1. august vil komme med i den nye indholdsfortegnelse over programbiblioteket, som ventes at udkomme sammen med septembernummeret af Nascom Nyt.

Bestilling af programmer og programbånd sker fortsat til nedenstående adresse (skriftligt), men husk IKKE AT MEDSENDE PENGE. Betaling for ydelserne SKAL ske v.h.a. det GIROKORT, som vedlægges programmerne (båndende).

INDSENDELSE OG BESTILLING AF PROGRAMMER/PROGRAMBÅND SAMT AL ØVRIG HENVENDELSE VEDRØRENDE PROGRAMBIBLIOTEKET SKAL SKE TIL:

JESPER SKAVIN  
BROHOLMS ALLE 3  
2920 CHARLOTTENLUND

tlf: (01) 64 03 14

(træffes ikke tirsdag og torsdag aftener.)

**Ændring til side 2-5:**

Der skal fordeles mellem 5 og 10 stk. 100nF afkoblingskondensatorer (fra +5 volt til nul) ud over hele opstillingen.

Ben forbindelserne i adressebufferne mangler, men de er IND/UD, 17/3, 15/5, 13/7, 11/9, 8/12, 6/14, 4/2 og 2/18 (input og output bennumre er altid 20).

Ole Brandt.

**Fuld afkodning af I/O-porte på NASCOM-1**

Har man brug for mere end de 8 I/O-porte, der kan vælges på NASCOM-1, må man lave en eller anden form for afkodning.

Med en 74LS138, der styres af A3-A7 og IORQ (evt med en enkelt inverter på A7) kan selekttere bundter på 8 porte lige som man selekterer bunker på 4kBytes i lagermodulerne, men så kommer problemet med IOEXT på CPU-kortet. Der skal ikke flyttes nogen forbindelse på LK1, I/O skal forblive forbundet til int, men ben 2 på IC45 og ben 5 på IC46 vippe op af soklerne og forbindes til stiften mærket ext på LK1. Vil man lave det lidt fikser, kan man skære en ledning over på printet og lodde en enkelt ledning mellem to gennemføringshuller, men ovenstående er den enkleste måde

Christian Laustsen

En rettelse og en kommentar til  
EPROM PROGRAMMEREN fra NN nr 5 og 5.2 1981

Først en rettelse til programmet. I 'MKOM'-rutinen efter instruktionen LD (ARGC),A tilføjes denne instruktion LD (ARGX),A . Dette bevirker, at monitorens READ-kommando også kan bruges under EPROM programmerens styreprogram. Hvis den tilføjede instruktion mangler, vil en READ blive udført som en VERIFY og der kommer derfor ingen data ind i computeren. (ARGX = OC2B H)

Så til kommentaren. EPROM'erne 2708 og 2716 laves af en lang række fabrikater. Næsten alle typerne kan programmeres af EPROM programmeren, men der er undtagelser. Undtagelserne gælder 2 mærker og kun de gamle typer. Det drejer sig flg. tre kredse: Motorola TMS2716, TEXAS INSTRUMENTS TI2716 og TMS2716JL. Det er meget sjældent, at man ser disse kredse, men det hænder dog. De 3 kredse er forløbere for den nuværende 2716 og de har mere tilfælles med en 2708 end med den nuværende 2716. De skal blandt andet have 3 forsyningsspændinger og programmeringen foregår på samme måde som for en 2708.

Her er en liste over hvilke EPROM's der kan programmeres med EPROM programmeren. Den er dog ikke komplet, da der findes en række mindre producenter, som ikke er med.

2708

- AMD 2708
- Electronic Arrays 2708
- Fairchild 2708
- Fujitsu 8518
- Intel 2708 / 8708
- Mitsubishi 58732S / 2708
- Motorola 68708 / 2708
- National 2708
- Signetics 2708
- TI 27L08
- Toshiba 322

2716

- AMD 4716
- Electronic Arrays 2716
- Fujitsu 8516 / 2716
- Hitachi 462716
- Intel 2716
- Mitsubishi 2716
- Mostek 2716
- Motorola MCM 2716
- National 2716
- NEC 2716
- TI 2516
- Toshiba 323

Venlig hilsen

Morten Kølbæk

## OTHELLO

```

OC80 4F 54 48 45 4C 4C 4F EF 0C 00 21 80 0C 11 DA 0B
OC90 0E 07 ED A0 13 B9 20 FA 21 9A 0B 06 0B 0E 41 71
OCA0 0C 23 23 10 FA 21 D8 0B 16 31 D9 06 0B D9 06 0B
OCB0 72 23 23 36 2E 10 FA 14 0E 30 09 D9 10 EF 21 A0
OCC0 09 36 00 23 23 36 FF 21 E0 09 36 FF 23 23 36 00
OCD0 21 19 0B DD 21 00 0B 22 29 0C EF 42 4C 41 43 4B
OCE0 20 4F 52 20 57 48 49 54 45 3F 00 CF FE 42 CC 49
OCF0 0D 28 05 FE 57 CC 4D 0D 20 F1 EF 1B 00 22 29 0C
OD00 EF 42 45 53 54 20 4F 52 20 57 4F 52 53 54 3F 00
OD10 D7 4C FE 42 CC 54 0D 28 05 FE 57 CC 59 0D 20 F0
OD20 00 00 CD 2A 0F EF 46 49 52 53 54 20 4F 52 20 53
OD30 45 43 4F 4E 44 3F 00 CF F5 EF 1B 00 F1 FE 46 CA
OD40 7C 0D FE 53 CA 03 0E 18 D9 0E 00 18 02 0E 7F 79
OD50 DD 77 01 C9 DD 36 00 01 C9 DD 36 00 00 C9 DF 62
OD60 04 30 FB F5 7B ED 4F F1 C9 CD 2A 0F EF 49 20 46
OD70 4F 52 46 45 49 54 00 DF 5D 79 2F 4F DD 36 03 00
OD80 CD 2A 0F EF 59 4F 55 52 20 4D 4F 56 45 20 00 DF
OD90 7B F7 D6 31 FA 81 0D D6 0B F2 80 0D C6 09 47 DF
ODA0 69 DF 7B F7 D6 41 FA 80 0D D6 0B FE 0C 28 56 A7
ODB0 F2 80 0D C6 0B 16 00 5F 21 9A 0B CB 23 19 11 40
ODC0 00 19 10 FD 3E 2E BE 20 05 CD 80 0E 28 16 CD 2A
ODD0 0F EF 42 41 44 20 4D 4F 56 45 00 06 E0 3E F0 FF
ODE0 10 FB 18 9C DD 36 06 00 CD AB 0E 00 00 00 00 3E
ODF0 00 DD BE 04 28 D8 DD 36 06 01 CD 10 0F CD AB 0E
OE00 CD 10 0F 18 04 DD 36 03 01 CD 2A 0F DF 5D 79 2F
OE10 4F 00 DD 36 05 FE 21 DA 0B D9 06 0B D9 06 0B 3E
OE20 2E BE CC 80 0E 28 0F 23 23 10 F4 11 30 00 19 D9
OE30 10 EA D9 C3 65 0E DD 36 06 00 CD AB 0E 3E 00 DD
OE40 BE 04 28 E3 CD 37 0F DD 7E 04 DD 96 05 FA 27 0E
OE50 28 0B DD 7E 04 DD 77 05 22 07 0B 18 CA ED 5F E6
OE60 40 28 C4 18 ED 3E FE DD BE 05 CA 8A 0F 2A 07 0B
OE70 DD 36 06 01 CD 10 0F CD AB 0E CD 10 0F C3 79 0D
OE80 E5 FD E1 79 2F FD BE BE C8 FD BE C0 C8 FD BE C2
OE90 C8 FD BE FE C8 FD BE 02 C8 FD BE 3E C8 FD BE 40
OEA0 C8 FD BE 42 C8 B4 C9 00 C5 E5 E5 FD E1 DD 36 04
OEB0 00 06 0B 11 FF 0E 1A 67 13 1A 6F 13 E5 10 F7 00
OEC0 00 06 0B 79 2F FD E5 E1 D1 19 BE 28 04 10 F4 18
OED0 2B C5 06 01 19 BE 20 05 04 18 F9 00 00 2F BE 28
OEE0 03 C1 18 E9 DD 7E 04 80 DD 77 04 3E 01 DD BE 06
OEF0 20 07 FD E5 E1 19 71 10 FC C1 10 C7 E1 C1 C9 FF
OF00 BE FF C0 FF C2 FF FE 00 02 00 3E 00 40 00 42 00
OF10 06 03 36 2E D9 06 20 3E F0 FF 10 FB D9 71 D9 06
OF20 20 3E F0 FF 10 FB D9 10 E9 C9 21 19 0B 22 29 0C
OF30 EF 1B 00 22 29 0C C9 3E 00 DD BE 00 C8 C5 E5 E5
OF40 06 40 11 C0 FF D7 0B E1 06 30 11 FE FF D7 03 E1
OF50 C1 C9 19 7E E6 F0 B8 20 F9 D7 01 C9 00 7E E6 0F
OF60 FE 01 28 15 FE 02 28 15 FE 03 28 15 FE 0B 28 09
OF70 FE 07 28 09 FE 06 28 09 C9 3E 03 18 06 3E FF 18
OF80 02 3E 02 DD 86 04 DD 77 04 C9 1E 00 3E 01 DD BE
OF90 03 C2 69 0D DD 7E 01 D7 18 53 2F 1E 00 D7 12 CD
OFA0 2A 0F 7A 93 FA CE 0F C2 D3 0F EF 44 52 41 57 00
OFB0 76 21 DA 0B D9 06 0B D9 06 0B BE 28 0E 23 23 10
OFC0 F9 06 30 23 10 FD D9 10 EE D9 C9 1C 18 EF EF 49
OFD0 00 18 05 EF 59 4F 55 00 EF 20 57 49 4E 00 DF 5B

```

PROGRAMMET STARTES MED "EC87"<n1>.

HVIS MAN IKKE KAN TRÆKKE TASTES ET TAL OG BOGSTAVET "U",  
 HVOREFTER NASCOM TRÆKKER. PROGRAMMET KAN KØRE PÅ EN U-  
 UDVIDET NASCOM MED NASSYS 1/3, OG DERFOR ER DET PÆNEST  
 UDEN GRAFIK !!



## Lidt om sletning af Prom's.

Når man i et stykke tid har arbejdet med sin computer kommer der et tidspunkt, hvor man vil til at lave lidt om i den. Dette kan skyldes interesse for selve systemet, men det kan jo også være at man vil spare noget tid. Dette kunne jo være en disassembler eller et skriveprogram. For at kunne slette en prom, kræves der en ultraviolet lampe, der sender et lys ud på bølgelængden 253,7 nanometer.

Jeg vil i en senere artikkelserie komme ind på de nærmere omstændigheder, angående selve opbygningen inde i selve kredsen, men da jeg ikke kun vil afskrive en allerede eksisterende artikel, kommer disse serier i små bidder. Philips har en lille lampe der oprindeligt blev lavet til at kunne dræbe bakterier med. Jeg fik venligst stillet en lampe til min rådighed, og takker herigennem firmaet for udvist venlighed.

Lampen hedder TUV 6 W. Som det medfølgende papir viser, er den ikke vidre stor. Den har en indbygget seriemodstand, således at den kan tilsluttes netspændingen.

Levetiden angives at være 2500 timer. Byg lampen ind i en kasse, og anbring kredsen der skal slettes, direkte under lampen. Efter ca 45 minutter skulle Prommen så være slettet. Dette ser ud til at stemme, idet jeg har samstemmede rapporter flere steder fra.

**ADVARSEL SE IKKE PÅ LAMPEN, OG LAD IKKE HUDEN BESTRÅLE.** Det kan give dig eksem og skade synet.

Hvis Du vil lave det rigtig fint så styr tændtiden på lampen ved hjælp af et tænd og sluk ur, eller endnu finere med en digitalstyret enhed. Jeg kommer med sådan en i et senere nummer.

Grunden til at jeg i øjeblikket ikke er så skrivende skyldes, dels agurketid, men osse at min computer er under en alvorlig ombygning, og dette tager nu en hel del tid.

I et af de kommende numre vil jeg så komme med en beskrivelse af en GEMINI-prombrænder, men denne kan jo ikke prøves, før min computer er ombygget.

OLE H

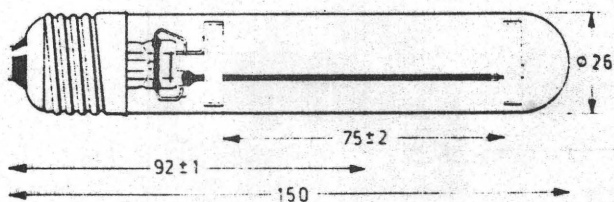
Designation	Nominal voltage <sup>1)</sup> V	Lamp current A	Spectrum	Intensity UV 253.7 nm <sup>2)</sup> $\mu\text{W cm}^2$	Energy output UV 253.7 nm mW	Depreciation <sup>3)</sup> %	Base	Useful life h
TUV 6 W	220	0.027	lines	0.85	85	25	E27	2500

<sup>1)</sup> Consumers' voltage 220 V = 210-230 V.

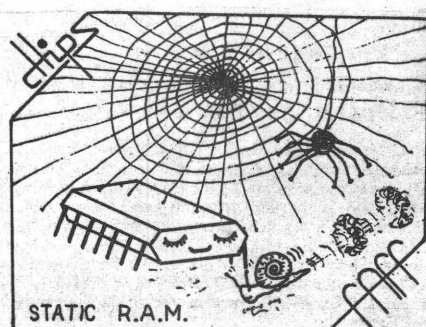
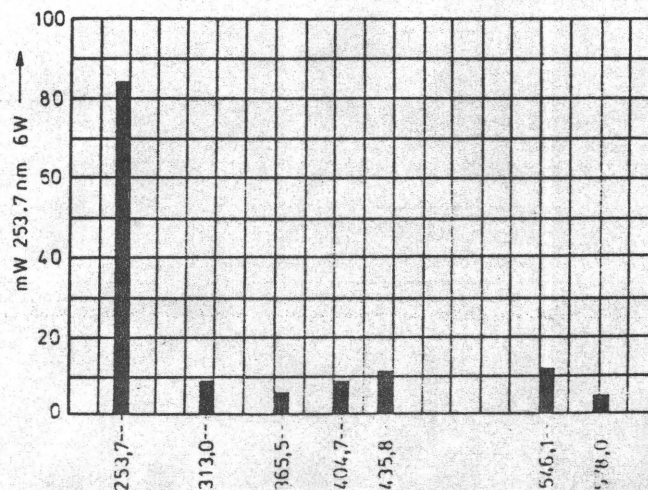
<sup>2)</sup> At a distance of 1 m.

<sup>3)</sup> After 2500 burning hours, with respect to the value of 100 hours.

## DIMENSIONS



## RELATIVE SPECTRAL ENERGY DISTRIBUTION



## ALMINDELIGE OPLYSNINGER OM FORENINGEN :

### Bestyrelsens sammensætning:

Formand	Asbjørn Lind Sidevolden 23 2730 Herlev 02 91 71 82
Næstformand	Jesper Skavin Broholms Alle 3 2920 Charlottenlund 01 64 03 14
Kasserer	Erik Hansen Lyngby Kirkestræde 6.1 2800 Lyngby 02 88 60 55 (mellem 8 og 15.30)
Sekretær	Carsten Senholt Blommevangen 6 2760 Måløv 02 66 19 65
	Ole Hasselbalch Vibeskrænten 9 2750 Ballerup 02 97 70 13
	Søren Sørensen Højlundvej 13 3500 Værløse 02 48 31 01
	Frank Damgård Kastebjergvej 26A 2750 Ballerup 02 97 10 20

### Henvendelse til foreningen:

Indmeldelse, adresseændringer o.l. til kassereren  
Programbibliotek til næstformanden

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

Indmeldelsesgebyr: 25,00 kr.  
Kontingent 1.7.81 - 1.7.82: 80,00 kr.

Oplag: 200

Redaktionen sluttet den 13.6.81

Husk at gamle numre kan købes hos Ole H. til 10 kr./stk.  
Husk printerservice hos formanden af Basic, Pascal, Nap/Nip  
Naspen og hexlistninger.

Annoncer som tillæg til 1,50 kr. pr. medlem pr. A4 ark  
leveret trykt til redaktionens adresse.