NASCOM Z80 NYT

NASCOM BRUGERGRUPPE
2730 Herlev

Sidevolden 23
Giro 6742602

NR: 5
2. årgang

MAJ 1981

På medlemsmødet i april, hvor bl. a. EPSON'en blev vist, blev det første PASCAL program til dette blad også udskrevet. Programmet er på side 19. (Tak til A.H. for lån af software den eftermiddag).

Hvordan skal vi huske hinanden på, at man skal henvende sig til den rigtige person, hvis der er noget man er utilfreds med ? Og ikke bare gå rundt i krogene og fremkomme med sin utilfredshed til dem, der gider at høre derpå. Jeg er helt sikker på, at den samlede bestyrelse gerne modtager kritik, men det er ikke rart at høre kritikken om personer, der ikke er til stede og derfor ikke kan argumenterer for sine handlinger eller mangel på samme !!

Jeg skal minde om generalforsamlingen den 4.5 på Køllegårdskolen (hvis bladet da kommer ud inden da - ellers: tak for en god generalforsamling!). Ligeledes vel mødt den 13.5 kl. 19.00 på Sidevolden

si'r ASBJØRN

INDHOLD

Kassetteinterface til N 1.

(Kansas City standard).

Når man skal gemme hukommelsesceller på bånd, skal man først lave 8 bit parallel om til 8 bit seriel. Dette kan laves synkront, hvor alle bit flyder i en jævn strøm efter hinanden. Eller det kan ske asynkront, hvor tidsimpulser sendes samtidig eller efter hver 8. bit. Af forskellige grunde sendes som regel asynkront, fordi det tillader variation af transmissionshastighed(mellem 5 og 9 %). Hvis man brugte synkron transmission, ville alle resterende data være tabte, hvis der var en ujævnhed i transmissionshastigheden !

Den kurveform hvormed en databyte sendes er som følger: Ved tomgang sendes altid et højt niveau, før den første bit sendes et startbit (lav), derefter følger de 8 bit fra ordet, hvorefter der afsluttes med paritetsbit og en eller to stopbits. Dette ville være vanskeligt for en os at programmere et modtagerprogram, der kunne skille disse informationer ad, men heldigvis har halvlederfolket lavet en kreds, der udfører alt arbejdet for os.

Det er en UART (Universiel Asynkron Resiver Transmitter), hvis benkonfiguration kan ses i manualen, der fulgte med i Nascom manualen. Forskellige kontrolinput kan give UART'en besked på at sende fire til 8 bit af data, lige eller ulige paritet, et eller to stop bit. Transmissionshastigheden bliver kontrolleret af to clockindgange, der skal køre 16 gange den ønskede hastighed.

Optagelse.

Når data nu er omdannet til en flydende strøm af høje og lave logiske niveauer, må det omdannes til en form, som kan optages på en almindelig båndoptager. En direkte optagelse af udgangen fra UART'en ville ikke være lykken, på grund af de store spring i DC-niveauer. Det næste man kunne gøre var at omdanne det til toner. En tone for høj og slukket for lav. Men det ville en moderne automatisk optagende kassettebåndoptager ikke kunne klare (signalstrukturen). Det næste var da at sende to forskellige toner , en for høj og en for lav. Og det er det princip, der ligger til grund for Kansas City standard. Denne standard bruger to toner på henholdsvis 2400 Hz og 1200 Hz. For at sende et "1" skal der sendes 8 perioder a 2400 Hz, men et "0" består af 4 perioder af 1200 Hz. Disse frekvenser er beregnet til en transmissonshastighed på 300 baud, og det er også nemt at danne UART'ens clockfrekvens på 4800 Hz. Dette er stadig langsomt, så for at forøge transmissionshastigheden, kan antallet af perioderne af 2400/1200 Hz formindskes med to for at opnå 600 baud osv. Ved maximum sendes en periode af 2400 Hz og en halv periode af 1200 Hz (2400 baud). Det her beskrevne interface kan klare 300, 600, 1200 og 2400 baud uden vanskelighed, det må dog bemærkes, at båndoptageren skal være af en forholdsvis god kvalitet, hvis der skal køres med 2400 baud.

Sendekredsløb.

Sendeskredsløbet bliver styret fra en 555'er, som svinger på 76.8 kHz, men den kan erstattes af en krystalkredsløb, der ved neddeling skal svinge inden for max 5% af 76,8 kHz.

Det 16 gange større transmissionsclock til UART'en tages fra en multiplekser (IC 2a), der styres fra to indgange. Udgangen kontrolleres af to J-K flip-flop (IC 5), som deler clockindgangen med fire og to, for at opnå den ønskede standard. Dette signal bliver ført gennem et simpelt lav-pas filter, for at fjerne de værste kanter på det firkantede signal. For højere hastighed vælges større UART clock, som derved afkorter varigheden af hvert bit. Den til 300 baud hørende kurveform ses under sendekredsløbet.

Modtagerkredsløb.

At anbringe data på tape er nemt, men at få dem tilbage igen er lidt vanskeligere ! Det første man skal gøre, er at skaffe rene og firkantede signaler, der er befriet for støj. Dette opnås ved hjælp af IC 3 og IC 10a i forbindelse med de tilknyttede kredsløb. C2 og R5 danner et høj pas filter, hvorefter signalet bliver forstærket i IC 3c, brugt i analog form. Det forstærkede signal bliver til firkantsignaler i smitt triggeren IC 10a. De to exclusive-or gate (IC 10 b og c) danner en detektor, der giver en negativ puls, hver gang indgangs skifter niveau. Denne puls bl;iver vendt og trigger en "digital" monostabil (IC 9a). Når man sørger for at den monostabile er blevet trigged (dvs. at tælleren er resat) før hver 12 clockimpuls, vil indgangen til D flip-flopen (IC 7) forblive høj. Det inverterede signal fra den anden D flip flop (IC 7b) vil ligeledes forblive høj. Denne tilstand bliver ikke ændret under udsendelse af 2400 Hz. Nå et 1200 Hz signal bliver modtaget vil tælleren nå 12 omkring 3/4 igennem den halve periode og bringer derved NAND gaten (IC 6d) lav. Den næste impuls IC 10c/d gør udgangen på IC 7a og IC 3d lav. To 2400 Hz pulse mere skal der til før IC 10c/d gør IC 7b høj igen.

UART clocken.

Når vi på denne måde har skaffet os de rå data, må vi skaffe et signal, der er 16 gange større end modtagelsenhastigheden af data til brug for UART'en. I en verden uden variationer på båndoptagere kunne vi bruge den samme clock, som sendekredsløbet, men dette ville være galmandsværk, da ingen kassettebåndoptager har konstant gang. For ikke at tale om udveksling af bånd indspillet med lidt forskelligt sendehastighed.

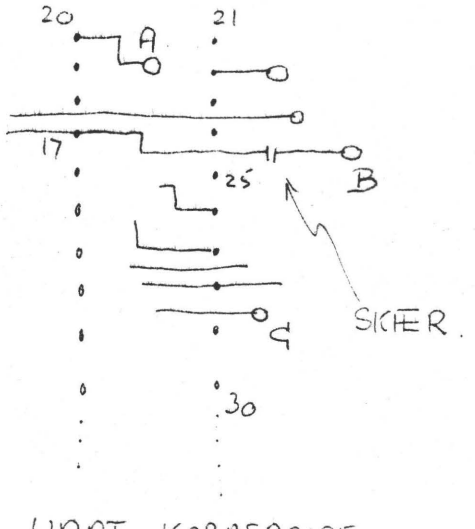
Heldigvis indeholder signalet fra båndoptageren tilstrækkelige informationer til, at vi kan rekonstruere clocksignalet. Når et 2400 Hz signal bliver modtaget vil IC 10c give et 4800 Hz signal fra sig. Når der modtages et 1200 Hz signal vil udgangen kun være 2400 Hz, så der skal findes en ekstra impuls. Sådan en impuls forefindes ved tælleren (IC 9a).

Der må bestemt være nogle før os, der har haft brug for et fase-låse kredsløb (PLL), for man kan købe en kreds, der sammenligner en bestemt frekvens med en udefra kommende frekvens og derefter justere udgangen. Men man kan også sætte en tæller mellem indgang og udgang, og derved opnå at faselås indtræder ved en frekvens der er et helt antal gange større end indgangs-frekvensen. I dette kredsløb er faselåsen konstrueret til 76.8 kHz og de ønskede frekvenser på 38.4, 19.2, 9.6 og 4.8 kHz kan fås fra deleren IC 9b. Den korrekte frekvens bliver valgt af multi plekseren IC 2b, på samme måde som i sendekredsløbet. De forskellige signalers kurveform kan ses under diagrammet.

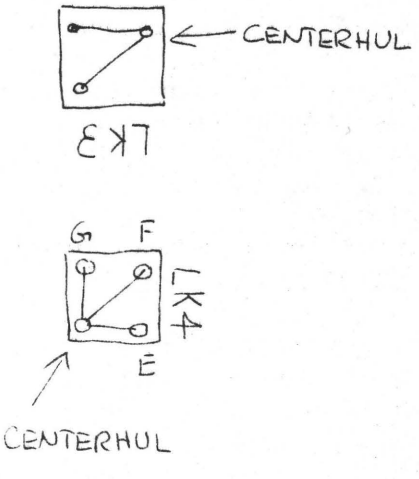
Konstruktion.

Du kan benytte dig af fotokopi af min printplade eller du kan lave dit eget lay-out. Men printudlæget er beregnet på følgende måde: Til soklen i midten kan du føre alle ud/ind- gange, så du ved hjælp af et DIL-stik kan forbinde interfacet med din computer på følgende måde, hvis du for fremtiden kun vil bruge Kansas City standard (ellers skal du gøre som skitsen sidst i denne artikel).

1) Afmonter link 3 og link 4. 2) Skær en kobberbane over som angivet på figuren over UART. 3) Find 16 bens soklen til seriel I/O, denne sokkel har allerede nogle ben benyttet, men ben 6,7,9 og 10 er ubrugte (bl.a.) og anbefales til denne konstruktion. 4) Forbind en ledning fra ben 6 til 'B' nær UART'en. Dette er transmissions clocken. 5) Forbind ben 7 på soklen til 'C'. Dette er Transmit Data. 6) Forbind ben 10 til centerhullet i link 4 (det er det alle tre kan forbindes til). Det er Reciver clock. 7) Forbind ben 9 til center hullet i link 3. Det er Receive Data. Dermed er der ikke flere ændringer på moder boardet.



UART KOBBERSIDE



KOMPONENTSIDE

Lav en ledning mellem de to sokler, således benene 6,7,8,9,10 og 16 er forbundet. Forbind nu på interfacet ben 6 til Transmit clock, ben 7 til Transmit Data, ben 9 til Receive Data og ben 10 til Receive clock. Bemærk at ben 18 og 8 allerede er forbundet som strømforsyning til kortet. ((Men husk at vende DIL stikket rigtigt, da du ellers vender spændingen til interfacet !!)).

Denne forbindelse tillader dig ikke at læse dine gamle bånd. Hvis du vil have den mulighed, skal du (endnu engang) benytte dig af diagrammet sidst i denne artikel. Du skal dog stadig skære en kobber bane over !

Justering.

Den eneste justering der skal foretages er indstilling af de 76.8 kHz på IC 4 ben 3. Dernæst kan du justere følsomheden på indgangen ved at ændre R6. Forstærkningen kan findes ved at divere R6 med R5. Indgangskondensatoren er i første omgang en DC-adskilning, men kan formindskes til omkring 1 nF, for at reducere indgangsstøj og netbrum.

<PS>

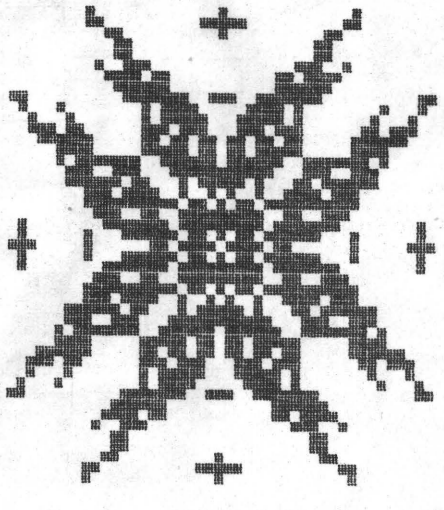
Denne konstruktion er beregnet til folk, der kan læse og fejlrette diagrammer. Det skal forstås derhen, at der kræves indsigt i Nascom virkemåde og en forståelse af virkemåde af interface for at forbinde tingene korrekt til hinanden. Til beroligelse skal det siges, at det er afprøvet af 3 forskellige mennesker - og det virkede alle tre steder. Og de værste fejl er blevet rettet !

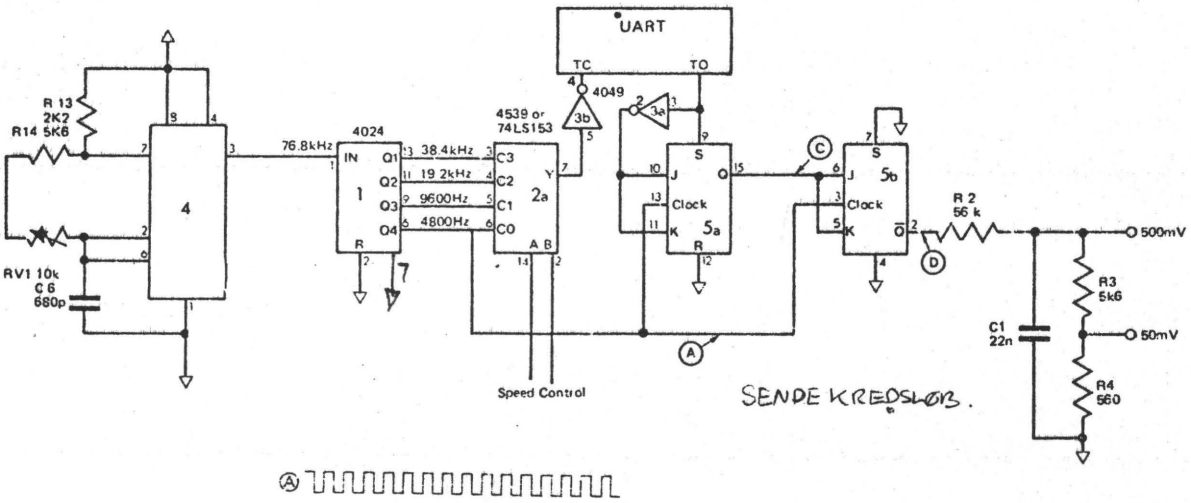
God fornøjelse si'r

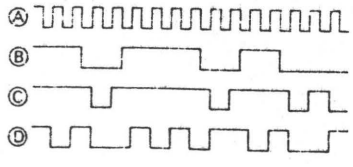
ASBJØRN

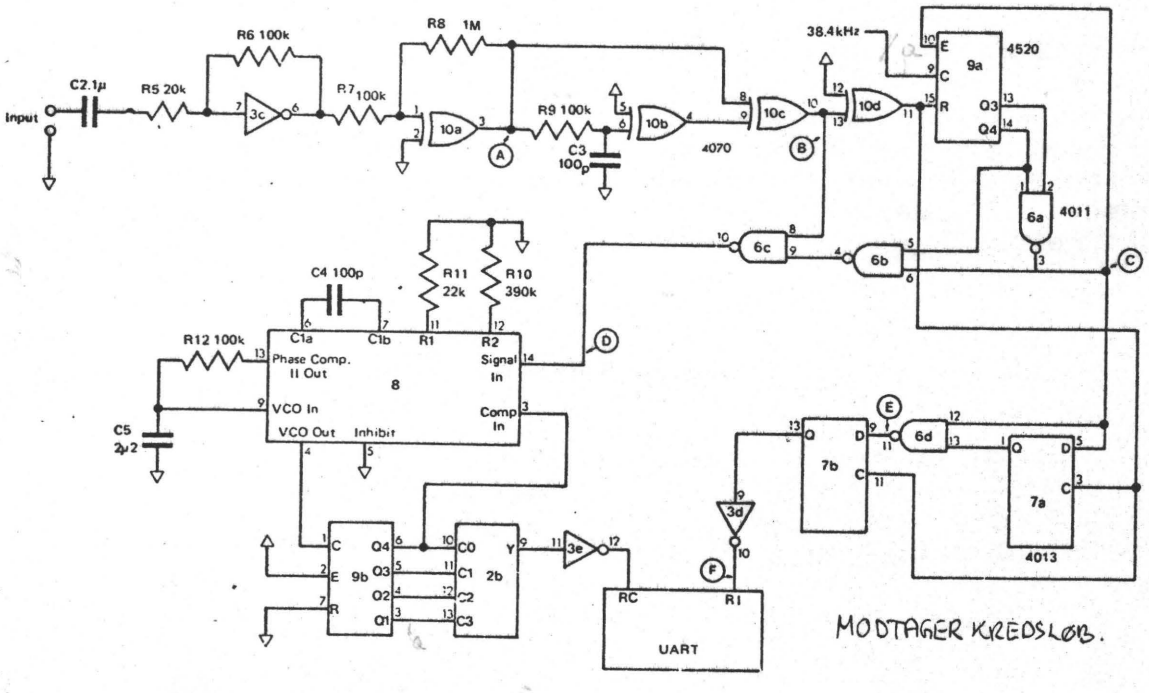
MØNSTERTEGNING

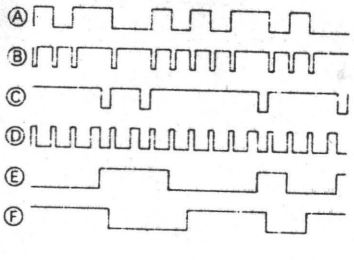
```
100 K=0:CLS:DX=0:DY=0:TI=INT(RND(1)*3+2)
110 X=INT(RND(1)*24+1):Y=INT(RND(1)*24+1)
120 IF X<Y THEN Y=25-Y
130 DX=INT(RND(1)*TI):DY=INT(RND(1)*TI)
140 X=X+DX:Y=Y+DY
150 IF X<25ANDX>0ANDY<25ANDY>0ANDY<=XTHEN170
160 GOTO110
170 IF POINT(X,Y)=0 THEN190
180 GOTO110
190 GOSUB250
200 Z=X:X=Y:Y=Z
210 GOSUB250
220 Z=X:X=Y:Y=Z
230 K=K+1:IF K<175 THEN120
240 FOR T=0 TO 5000:NEXT:GOTO100
250 A=X+22:IF Y<4 THEN B=Y+44:GOTO270
260 B=Y-4
270 SET(A,B):SET(70-X,B)
280 P=X+22:Q=44-Y
290 SET(P,Q):SET(70-X,Q)
300 RETURN
```

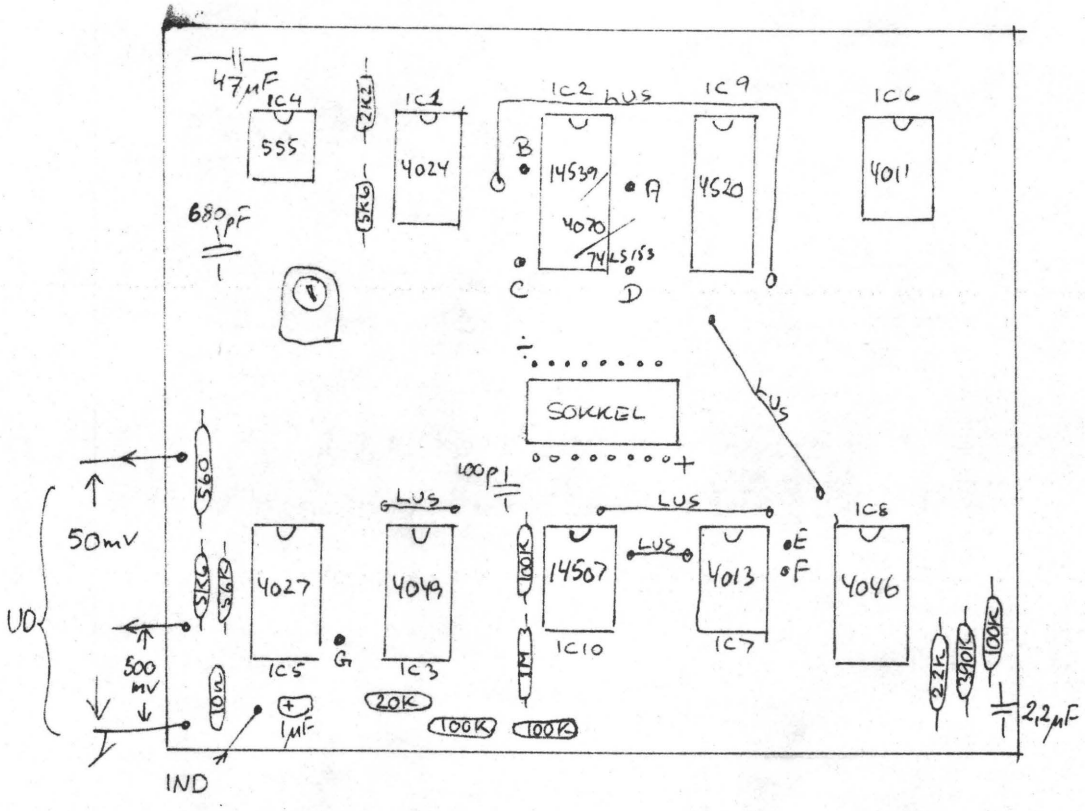



UART
TC
TO
4049
3b
3a
4539 or
74LS153
4024
R 13
2K2
R14 5K6
RV1 10k
C 6
680p
4
76.8kHz
IN
Q1
Q2
Q3
Q4
R
38.4kHz
19.2kHz
9600Hz
4800Hz
C3
C2
C1
C0
A B
2a
1
Speed Control
S
J
Clock
K
R
5a
5b
R 2
56 k
500mV
R3
5k6
50mV
C1
22n
R4
560
SENDE KREDSLØB.

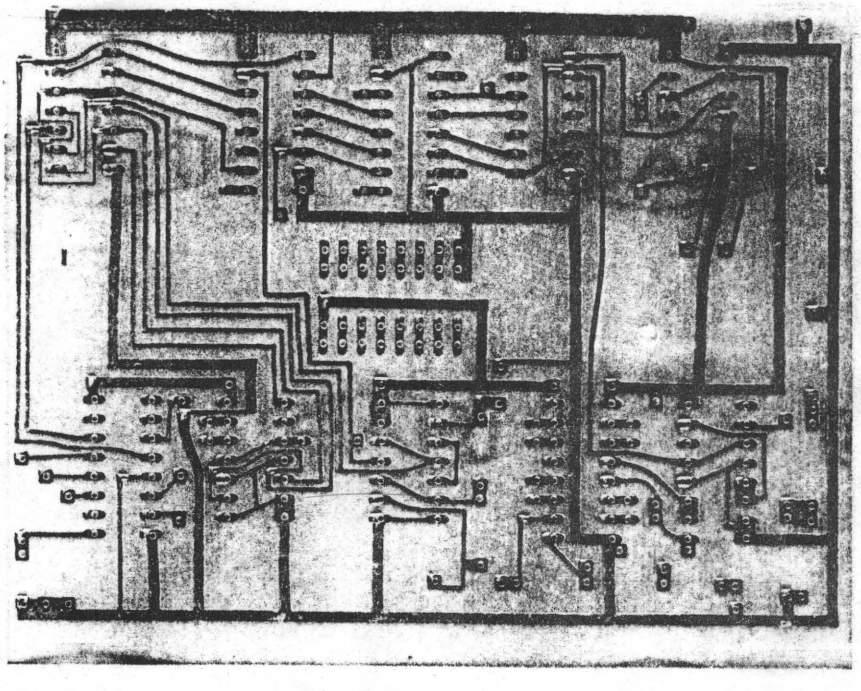

A
B
C
D


Input
C2.1µ
R5 20k
R6 100k
3c
R7 100k
R8 1M
10a
R9 100k
C3
100p
10b
4070
10c
10d
38.4kHz
9a
4520
E
C
R
Q3
Q4
6a
4011
6c
6b
C4 100p
R11
22k
R10
390k
R12 100k
C1a
C1b
R1
R2
Phase Comp.
II Out
Signal
In
8
VCO In
VCO Out
Inhibit
Comp
In
C5
2µ2
9b
2b
3e
RC
RI
UART
3d
6d
7b
7a
4013
MODTAGER KREDSLØB.

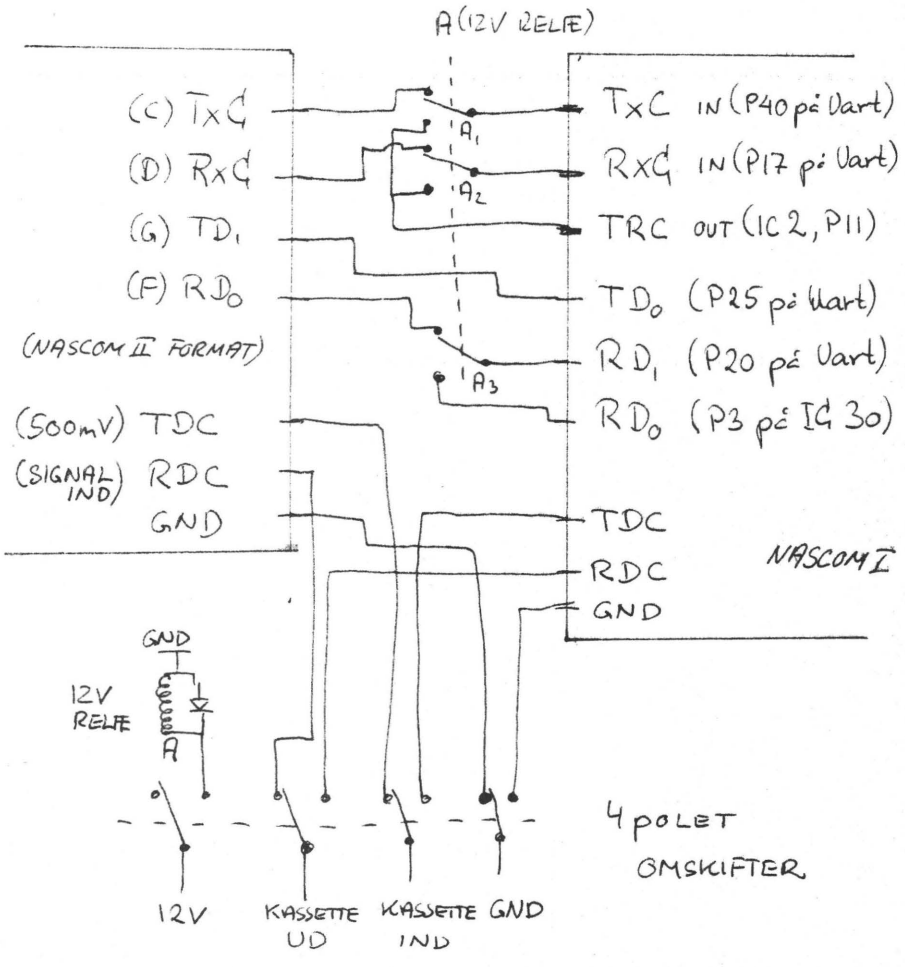

A
B
C
D
E
F



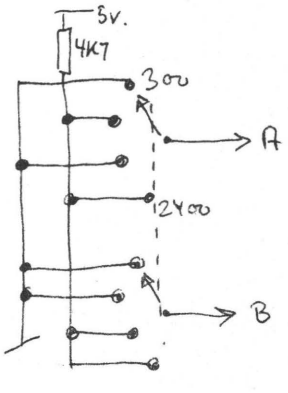
A HASTIGHED
B HASTIGHED
C $\overline{Tx}$ CLOCK
D Rx CLOCK
E Q Rx DATA
F $\overline{Q}$ Rx DATA
G $\overline{Tx}$ DATA



SAMTIDIG BRUG AF NASCOM I OG NASCOM II FORMAT



OMSKIFTERNE VIST I KANSAS CITY STILLING.



OMSKIFTER TIL VALG AF HASTIGHED. (2x4)

EPROM programmer til 2708 og 2716.

De velkendte EPROM'er, der anvendes utallige steder, er faktisk relativ lette at programmere. Det eneste der kan være lidt vanskeligt at omgåes er spændingerne under programmeringen. De ligger i området 25-26 volt og det er jo lidt højere end man er vant til i computersammenhæng.

Da jeg lavede denne EPROM programmer havde jeg bl. a. den ide at den skulle være så simpel som muligt i hardwaren. Softwaren, programmerne, er altid nemmere at ændre i og med blot et lidt større styreprogram kan man faktisk spare en del komponenter.

Resultatet er,at man kan lave en EPROM programmer med 4 standard TTL kredse og 4 transistorer + et par modstande og dioder. De 3 TTL kredse indeholder hver en 4 bits binær tæller, der bruges til at danne den adresse der skal programmeres i EPROM'en. Den sidste kreds er hex inverter med højvolts åben kollector i udgangen.

Til at styre EPROM programmeren anvendes en Z80 PIO. Den ene port bruges som bidirektional dataport og den anden som kontrolport. Ved at lægge et passende bitmønster på kontrolporten kan man frembringe de spændinger der nødvendige til læsning/programmering af 2708 eller 2716 EPROM's. Valg af EPROM type foregår med den to-polede omskifter og på b_7 kan computeren læse hvilken type der er valgt.

Da det er styreprogrammet der skal sørge for,at der er de rigtige spændinger på soklen i EPROM programmeren, er det absolut nødvendigt at vælge EPROM type og starte styreprogrammet FØR der sættes en EPROM i soklen. I modsat fald kan man risikere at EPROM'en signalerer om forglemmelsen med små røgskyer ! Styreprogrammet kan laves meget avanceret, men denne version indeholder kun de 3 grundlæggende rutiner programmering, læsning og sammenligning af EPROM og bufferen, som ligger fra adresse 1000h. Disse 3 rutiner arbejder alle på samme buffer, som fyldes med data v.h.a. monitoren.

Samling og afprøvning.

Hardwaren samles lettest på et print og hvis man monterer et stik på ledningerne over til PIO'en er det nemt at forbinde EPROM programmeren når man har brug for den. Forsyningsspændingerne kan hentes fra computeren undtagen +27 V. Denne spænding bruges kun under programmering og må i det tilfælde hentes fra en ekstern strømforsyning. Hvis man kun vil læse en EPROM er den altså ikke nødvendig. Der anvendes 27 volt da man må regne med ca. 1 volt spændingsfald over transistoren der switcher spændingen. Der er også brug for 25 volt og den spænding dannes ved at der er sat en siliciumdiode i serie med forsyningsspændingen.

Programmet er delt op i nogle ret små moduler og da det også er kommenteret skulle det være relativ let at forstå og eventuelt ændre. For at få programmet til at køre på en anden computer er det kun nødvendigt at ændre på delay-rutinen. Talkonstanten der lægges i B i adresse EA1h skal ændres afhængig af clocken på CPU'en. 53 decimalt giver de krævede 250 µS ved 3 Mhz clock. Ved andre frekvenser kan man bruge forholdsregning.

Slettet / ny EPROM :

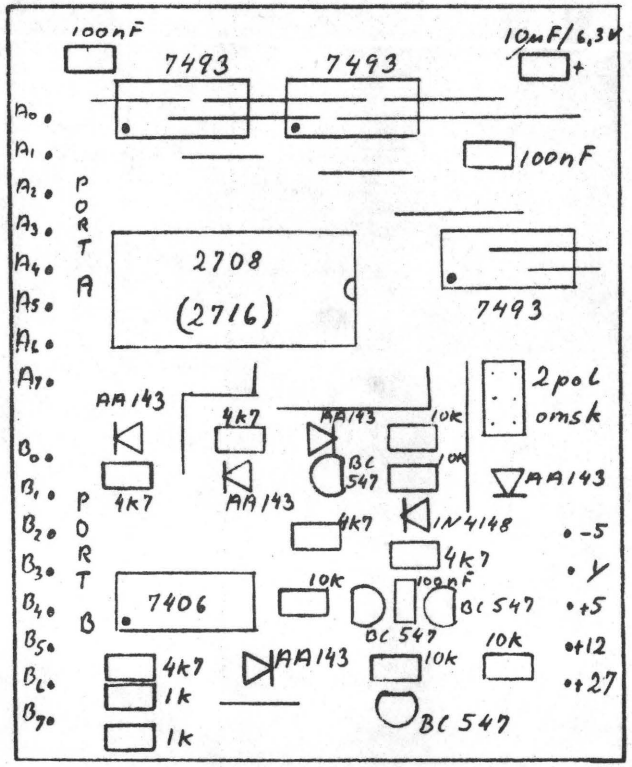
Indeholder FF i alle celler. Kan kontrolleres ved at fylde bufferen med FF og verificere.

Programmering af dele af EPROM'en :

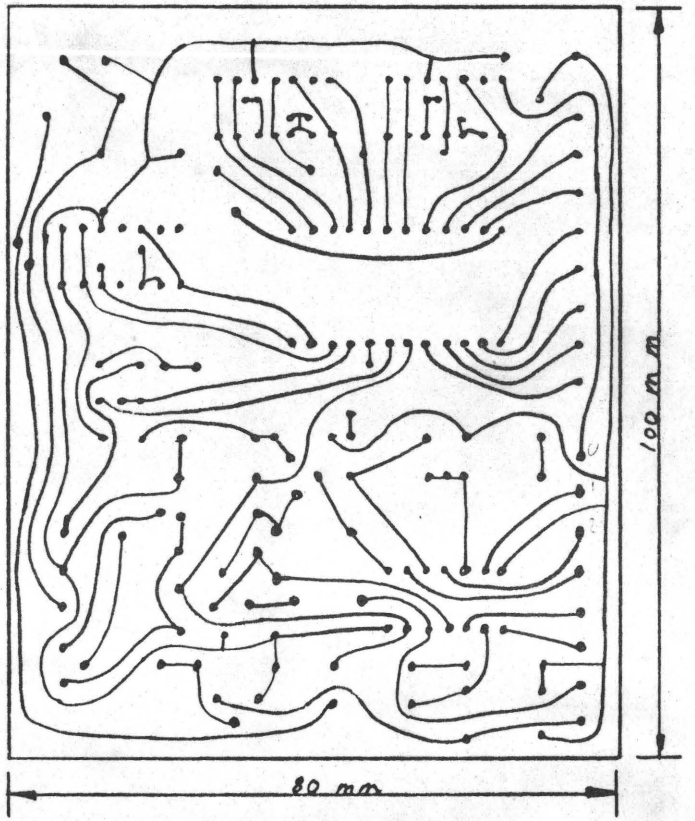
De områder man ikke ønsker programmeret skal fyldes med FF og ved en senere programmering af disse områder skal man huske at lægge FF i de områder som blev programmeret i første omgang.

Jørgen Nielsen Morten Kalbak

KOMPONENTSIDE



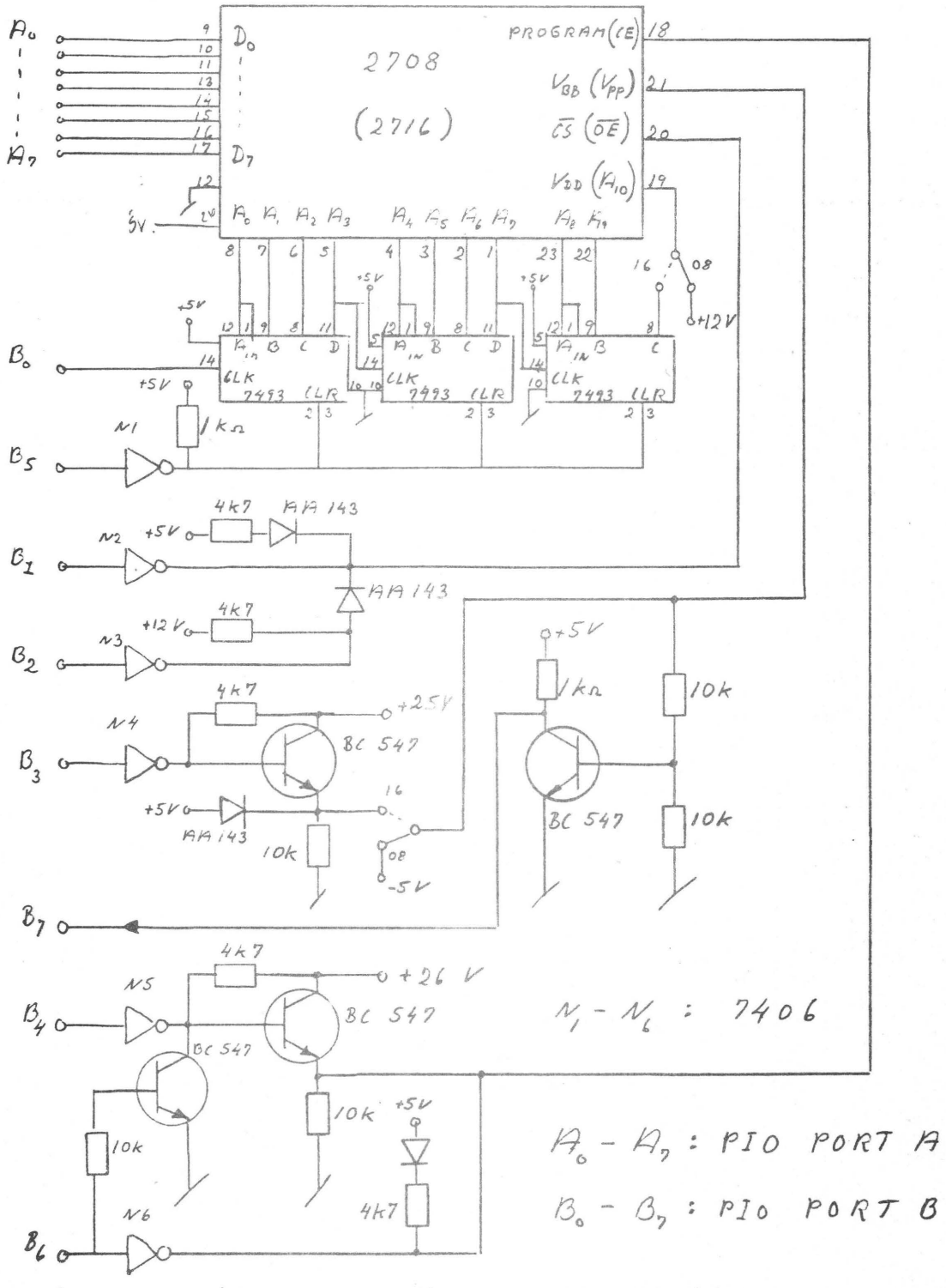
KOBBERSIDE



			EPROM PROGRAMMER	Konstr.
				Tegn.
				Godk
				Tegning nr.
Målestok:				
Materiale:				

EPROM PROGRAMMER

MK 10-3-81



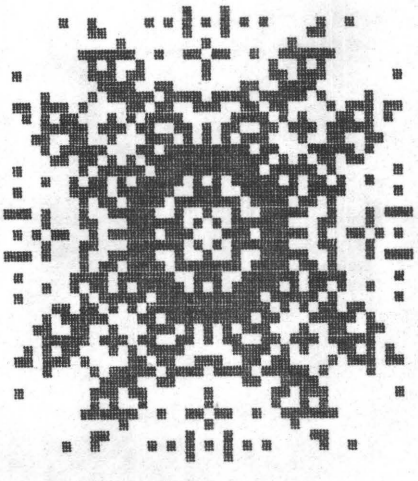
TABEL OVER HVILKE SPÆNDINGER DER ÆNDRES VED PROGRAMMERING

Ben nr	18		19		20		21	
Mode	R	W	R	W	R	W	R	W
2708	0	0→26	12	12	0	12	-5	-5
2716	0	0→5	A_{10}	A_{10}	0	5	5	25
Bit nr	B_6 B_4	B_6 B_4			B_2 B_1	B_2 B_1	B_3	B_3
2708	1 1	1 1 → 0 0			1 1	0 0	0	0
2716	1 1	1 1 → 1 0			1 1	1 0	1	0

SOFTWARE I NN 6

```
LNNR ADDR OBJ.KODE LABEL   MNEM OPERAND    COMMENT

0001              ; ************************
0002              ; *  NASCOM DEMONEN    *
0003              ; ************************
0004
0005              ;KOMMENTERET AF ERIK PALSBO
0006
0007 0063         INLIN:  EQU  63H
0008 0079         RLIN:   EQU  79H
0009 0060         ARGS:   EQU  60H
0010 0028         PRS:    EQU  28H
0011 000D         NL:     EQU  0DH
0012 000C         CS:     EQU  0CH
0013
0014 0C23         BRKADR: EQU  0C23H
0015 0C25         BRKVAL: EQU  0C25H
0016 0C7E         NMIAD:  EQU  0C7EH
0017 1000         STACK:  EQU  1000H
0018
0019 0D00         ORG     0D00H          ;STARTADRESSE
0020
0021 0D00 00              NOP
0022 0D01 00              NOP
0023 0D02 00              NOP
0024 0D03 00              NOP
0025 0D04 00              NOP
0026 0D05 00              NOP
0027 0D06 00              NOP
0028 0D07 310010  START:  LD   SP,STACK  ;LOAD STACKPOINTEREN
0029 0D0A 210000          LD   HL,0      ;FLYT 0 TIL
0030 0D0D 227E0C  STOP:   LD   (NMIAD),HL ;- NMI - JUMPADRESSEN
0031 0D10 21FF0F          LD   HL,STACK-1 ;FLYT ADR OFFFH
0032 0D13 22230C          LD   (BRKADR),HL;- TIL BREAKPOINT ADR
0033 0D16 3E0D            LD   A,0DH     ;- OG GEM 0DH
0034 0D18 32250C          LD   (BRKVAL),A ;- I BREAKPOINTVÆRDI
0035
0036 0D1B EF              RST  PRS
0037 0D1C 0C4A4547        DB   CS,'JEG ER NASCOM DEMONEN'
0038 0D20 20455220
0039 0D24 4E415343
0040 0D28 4F4D2044
0041 0D2C 454D4F4E
0042 0D30 454E
0043 0D32 0D564145        DB   NL,'VAER RAR IKKE AT RØRE MIG!'
0044 0D36 52205241
0045 0D3A 5220494B
0046 0D3E 4B452041
0047 0D42 5420524F
0048 0D46 45524520
0049 0D4A 4D494721
0050 0D4E 0D284A45        DB   NL,'(JEG VIL HELST VAERE ALENE)'
0051 0D52 47205649
0052 0D56 4C204845
0053 0D5A 4C535420
0054 0D5E 56414552
0055 0D62 4520414C
0056 0D66 454E4529
0057 0D6A 0D0D4856        DB   NL,NL,'HVEM ER DU FOR RESTEN?'
0058 0D6E 454D2045
0059 0D72 52204455
0060 0D76 20464F52
0061 0D7A 20524553
0062 0D7E 54454E3F
0063 0D82 0D00            DB   NL,0
0064 0D84 DF63            SCAL INLIN     ;LNDLÆS EN LINIE
0065 0D86 1A              LD   A,(DE)    ;UNDERSØG 1. BOGSTAV
0066 0D87 FE25            CP   '%'       ;ER DET %
0067 0D89 2016            JR   NZ,AGURK  ;-NEJ- GÅ AGURK
0068 0D8B 13              INC  DE        ;PEG PÅ NESTE BOGSTAV
0069 0D8C DF79            SCAL RLIN      ;UNDERSØG ARGUMENT
0070 0D8E 3811            JR   C,AGURK   ;HVIS EJ OK - GÅ AGUR
0071 0D90 DF60            SCAL ARGS      ;LOAD ARGUMENT I HL
0072 0D92 7C              LD   A,H
0073 0D93 B5              OR   L
0074 0D94 2007            JR   NZ,GEMHL  ;HVIS EJ NUL - GEM ADR
0075 0D96 210000          LD   HL,0
0076 0D99 22230C          LD   (BRKADR),HL;SÆT BREAK-ADR. TIL 0
0077 0D9C C7              RST  0         ;RESTART 0
0078 0D9D 22A50D  GEMHL:  LD   (HELP),HL ;GEM HL I HELP
0079 0DA0 E9              JP   (HL)      ;SPRING TIL DENNE ADR
0080 0DA1 2AA50D  AGURK:  LD   HL,(HELP) ;HENT ADRESSEN FRA HELP
0081 0DA4 E9              JP   (HL)      ;- OG SPRING TIL DEN
0082 0DA5 A70D    HELP:   DW   L0DA7     ;GEMMEAREAL TIL ADRESSE
0083 0DA7 EF      L0DA7:  RST  PRS
0084 0DA8 48454C50        DB   'HELP',0
0085 0DAC 00
0086 0DAD 18F8            JR   L0DA7
0087
0088 0DAF         END
```



MEDLEMSMØDE APRIL.

På mødet blev vist EPSON printer, disksystem a la Anders, debug/trace fra Benny og Johnny og en lidt ældre seriel printer.

Derefter gennemgik Anders NASSYS ind og ud rutiner. Jeg skal her gengive hovedindholdet af foredraget. Der er 3 forskellige kaldemåder for subrutiner under NASSYS.

1. RST kald
2. Kald til absolutte adresser
   (hvad der absolut må undgås !!)
3. Monitorkald ved hjælp af
   RST 18H (SCAL #)
   DB #

Virkemåde:
Rutinens # hentes i koden og slåes op i kommandotabellen, ved at gange # med 2 og lægge (STAB). Herefter hoppes til den absolutte adresse. Kan udadtil betragtes som et almindeligt kald.

Rutinen SCALJ (5CH):

Virker som ovenfor, men rutinens nr er i A registeret.

Rutinen ATE (73H)

Kaldes med HL pegende til et word, der indeholder adressen på en tabel af rutiner, der skal kaldes efter tur. Der afsluttes med 0. Afviklingen standser enten ved 0 eller ved carry.

```
HL      ta. l   -->     #1
        ta. h           #2
                        ..
                        .

                        0
```

Ved udskrift af enkeltkarakter benyttes: RST ROUT (30H) F7.

Rutiner der initialiseres efter brug af RESETKNAP er=> Ud: CRT, Ind: KBD, SRLIN (der muliggør Generate).

Udtabellens adresse ligger i $OUT
Ind - - - - - - - - i $IN

Bruger indrutinens adresse ligger i $UOUT
- - ud- - - - - - - - i $UIN

Indlæsning har flere muligheder:

RST RIN (8)	CF
SCAL IN	DF 62
SCAL BLINK	DF 7b
SCAL INLIN	DF 63

Man kan styre sine ind og ud vej v.h.a.

CRT	65H	SKÆRM
SRLX	6FH	TAPE/PRINTER
XOUT	6EH	ASCII ENHED (PARITET)
UOUT	75H	BRUGERDEFINERET

Ind enheder:

KBD	61H	TASTATUR
SRLIN	70H	TAPE
XKBD	74H	EXTERNT TASTATUR
UIN	76H	BRUGERDEFINERET

Indsætning af ny tabel kan ske ved at benytte en subrutine

```
        LD HL,XX      ; XX er adressen på tabel
        SCAL NOM/NIM  ; ud eller ind tabel
```

og man normaliserer igen ved

```
        SCAL NNOM/NNIM
```

man kan aktivere en brugerrutine ved følgende:

```
        LD   HL,RUTINE
        LD   ($UOUT),HL
        LD   HL,TABEL
        SCAL NOM
```

Derefter gennemgik og kommenterede Anders en repeterende subrutine, som et eksempel på input rutine under NASSYS.

```
ORG   0000                ;PROGRAMMET ER RELOKER
MEM   $                   ;BART.

        LD   HL,KEYIN     ;SÆT BRUGER INPUT HOP
                          ;VEKTOR
        LD   (UIN),HL
        LD   HL,INTBL     ;FLYT INPUT TABEL
        SCAL NIM
        SCAL MRET         ;GODDAG IGEN NASSYS

KEYIN:  LD   HL,RESTBL    ;PEG HL TIL RESETTABEL
        LD   DE,KMAP+1    ;PEG DE TIL KEYBOARDTBL
        LD   B,8          ;GØR DET 8 GANGE
RESET:  LD   A,(DE)       ;HENT FRA KEYBOARDTABEL
        AND  (HL)         ;NULSTIL NOGLE BITS
        LD   (DE),A       ;GEM DET IGEN
        INC  HL           ;PEG TIL NÆSTE OFFER!
        INC  DE
        DJNZ RESET        ;GENTAG
        SCAL KBD          ;SCAN TASTATURET
        LD   HL,KEYS      ;PEG HL TIL STATUSORD
        JR   C,KEY        ;NOGET ER NEDE => HOP
        XOR  A            ;NULSTIL A OG CF
NEWK:   LD   (HL),A     ;GEM SIDSTE TASTEDE TAST
        INC  HL
        LD   (HL),LONG    ;LANG PAUSE
        RET
KEY:    CP   (HL)         ;VAR UÆNDRET TAST ?
        SCF           ;HVIS IKKE SÅ BRUG DET HER!
        JR   NZ,NEWK      ;NEJ => HOP
        INC  HL           ;PEG TIL TÆLLER
        OR   A            ;NULSTIL CF
        DEC  (HL)         ;TÆL NED
        RET  NZ           ;VI ER IKKE KLAR ENDNU!
        LD   (HL),SHORT   ;KORT PAUSE NÆSTE GANG
        SCF               ;BRUG DET HER
        RET               ;SLUT
RESTBL: DB 80H,80H,80H    ;DENNE TABEL NULSTILLER
        DB 80H,0C0H,80H   ;IKKE SHIFT,CONTRROL OG
        DB 80H,0B8H       ;GRAPH
INTBL:  DB   UIN,0      ;INDLÆS KUN FRA TASTATUR
NIM:    EQU  72H
MRET:   EQU  5BH
KBD:    EQU  61H
KMAP:   EQU  0C01H
UIN:    EQU  0C7BH
LONG:   EQU  60
SHORT:  EQU  LONG/3
KEYS:   EQU  0C2CH
        END
```

```
10 REM ****************************************
20 REM * ZAMBIELAND bearbejdet til NASCOM *
30 REM * af Asbjørn Lind                  *
40 REM ****************************************
50 CLS:INPUT"INSTRUKTION";Q$:IFQ$<>"JA" THEN180
60 PRINT"Du er lige landet i Zombieland, hvor alle"
70 PRINT"indbyggerne er blinde! De kan kun HØRE !"
80 PRINT"Du kan kun overleve, hvis du lokker dem"
90 PRINT"i de dybe elefanthuller!"
100 PRINT"Du kan ikke LØBE fra dem, kun ved omtanke"
110 PRINT"kan du overleve! Du kan flytte dig i disse"
120 PRINT"retninger:"
130 PRINT
140 PRINT"7 8 1"
150 PRINT"6 * 2"
160 PRINT"5 4 3"
170 INPUT"KLAR - TRYK NL";P$
180 DIMB(12,22),Z(25,2),P(8),Q(8)
190 FOR N1=1 TO 8
200 READP(N1),Q(N1)
210 NEXT
220 DATA1,-1,1,0,1,1,0,1,-1,1,-1,0,-1,-1,0,-1
230 FORN1=1 TO 25
240 FORN2=1TO2
250 Z(N1,N2)=0
260 NEXT N2
270 NEXT N1
280 Z1=0:PRINT"FINDER NU ET ZAMBIELAND"
290 FOR N1=1 TO 12
300 FORN2=1 TO22
310 B(N1,N2)=4
320 NEXT N2
330 NEXT N1
340 FOR N1=2 TO 11
350 FOR N2=2 TO 21
360 R=20*RND(1)
370 IFR>18.5 THEN450
380 IF R<17.92 THEN440
390 Z1=Z1+1
400 Z(Z1,1)=N1
410 Z(Z1,2)=N2
420 B(N1,N2)=2
430 GOTO450
440 B(N1,N2)=1
450 NEXT N2
460 NEXT N1
470 X=5+INT(10*RND(1))
480 Y=3+INT(5*RND(1))
490 B(Y,X)=3
500 FOR N1=Y-1 TO Y+1
510 FOR N2=X-1 TOX+1
520 IF ABS(Y-N1)+ABS(X-N2)=0 THEN540
530 B(N1,N2)=1
540 NEXT N2
550 NEXT N1
560 FOR N1=1 TO Z1
570 IF B(Z(N1,1),Z(N1,2))=2 THEN640
580 FOR N2=N1 TO Z1
590 Z(N2,1)=Z(N2+1,1)
600 Z(N2,2)=Z(N2+1,2)
610 NEXT N2
620 Z1=Z1-1
630 GOTO560
640 NEXT N1
650 PRINT
660 PRINTTAB(5)"*  *  * * ***Zambieland*** * *  *  *"
670 FOR N1=1 TO 12
680 FOR N2=1 TO 22
690 ON B(N1,N2) GOTO700,720,740,760
700 PRINT"  ";
710 GOTO770
720 PRINT" Z";
730 GOTO770
740 PRINT" X";
750 GOTO770
760 PRINT" O";
770 NEXT N2
780 PRINT
790 NEXT N1
800 PRINT
810 INPUT"HVOR VIL DU HEN";A
820 IF A>8 THEN840
830 IF A>=1 THEN860
840 PRINT"DET SKULLE VAERE MELLEM 1 OG 8!!!"
850 GOTO810
860 B(Y,X)=1
870 Y=Y+Q(A)
880 X=X+P(A)
890 ON B(Y,X) GOTO900,920,900,940
900 B(Y,X)=3
910 GOTO960
920 PRINT"LIGE IND I EN ZOMBIE!"
930 GOTO1240
940 PRINT" LIGE NED I ET HUL !!!!!!"
950 GOTO1240
960 Z2=1
970 Z8=Z(Z2,1)
980 Z9=Z(Z2,2)
990 B(Z8,Z9)=1
1000 Z8=Z8+SGN(Y-Z8)
1010 Z9=Z9+SGN(X-Z9)
1020 ON B(Z8,Z9) GOTO1150,1120,1100,1030
1030 PRINT"DER ROG EN ZAMBIE I ET HUL!"
1040 FOR Z3=Z2 TO Z1
1050 Z(Z3,1)=Z(Z3+1,1)
1060 Z(Z3,2)=Z(Z3+1,2)
1070 NEXT
1080 Z1= Z1-1
1090 GOTO1190
1100 PRINT"DU ER FANGET .. .  .  !"
1110 GOTO1240
1120 PRINT"HER KOMMER DER NOGLE ZOMBIER"
1130 B(Z(Z2,1),Z(Z2,2))=2
1140 GOTO1180
1150 B(Z8,Z9)=2
1160 Z(Z2,1)=Z8
1170 Z(Z2,2)=Z9
1180 Z2=Z2+1
1190 IF Z2<=Z1 THEN970
1200 IF Z1>=1 THEN650
1210 PRINT
1220 PRINT"DET VAR DEN SIDSTE"
1230 PRINT"DU UNDSLAP!"
1240 INPUT"ET SPIL TIL";A$
1250 IF A$="JA" THEN280
1260 IF A$="NEJ" THEN1280
1270 PRINT"DU SKAL SVARE JA ELLER NEJ";GOTO180
1280 END
Ok
```

```
LNNR ADDR OBJ.CODE LABEL    MNEM OPERAND     COMMENT

0001
0002                ;                Sammenligningsrutine
0003
0004                ;sammenligner to 8 bits tal, der befinder sig
0005                ;i lokation LOKA og LOKB
0006
0007 0000           ORG      0
0008
0009 0000 3A0102             LD    A,(LOKB)
0010 0003 47                 LD    B,A
0011 0004 3A0002             LD    A,(LOKA)
0012 0007 CD0001             CALL  SAMLGN      ;sammenlign A med B
0013 000A 1824               JR    MINDRE      ;hvis A<B
0014 000C 1842               JR    LIGMED      ;hvis A=B
0015 000E 1860               JR    STORRE      ;hvis A>B
0016 0030           MINDRE:  EQU   $+20H
0017 0050           LIGMED:  EQU   $+40H
0018 0070           STORRE:  EQU   $+60H
0019
0020 0100           ORG      100H
0021
0022 0100 B8        SAMLGN:  CP    B           ;A mod B
0023 0101 E3                 EX    (SP),HL     ;hent returadresse
0024 0102 2810               JR    Z,LIGE      ;hvis A=B
0025 0104 F5                 PUSH  AF          ;gem A og Flag
0026 0105 A8                 XOR   B
0027 0106 F20E01             JP    P,ENS       ;hop hvis Sign er ens
0028 0109 F1                 POP   AF          ;gendan A og Flag
0029 010A 380A      TEST:    JR    C,MIN       ;A<B
0030 010C 1804               JR    STOR        ;A>B
0031 010E F1        ENS:     POP AF
0032 010F 3F                 CCF
0033 0110 18F8               JR    TEST
0034 0112 23        STOR:    INC   HL          ;forskyd returadr.+4
0035 0113 23                 INC   HL
0036 0114 23        LIGE:    INC   HL          ;forskyd returadr.+2
0037 0115 23                 INC   HL
0038 0116 E3        MIN:     EX    (SP),HL     ;gendan retur
0039 0117 C9                 RET
0040
0041 0200           ORG      200H
0042
0043 0001           LOKA:    DS    1
0044 0001           LOKB:    DS    1
0045
0046                ;Hvis man vil have en test om A>=B, kan man
0047                ;i 3 JR-tabel kalde den samme rutine for det
0048                ;ønskede.  F. eks
0049
0050 0000           ORG      0H
0051 0000 3A0102             LD    A,(LOKB)
0052 0003 47                 LD    B,A
0053 0004 3A0002             LD    A,(LOKA)
0054 0007 CD0001             CALL  SAMLGN
0055 000A 1824               JR    MINDRE      ;A<B
0056 000C 1842               JR    LIGMED      ;A=B
0057 000E 1840               JR    LIGMED      ;A>B
```

Hvis du kommer i den situation, at du ønsker at sammenligne indholdet af register HL med register DE uden at ændre indholdet af nogle af registrene, kan det klares på denne måde:

```
B7      OR      A
ED 52   SBC     HL,DE
19      ADD     HL,DE
```

Hvis HL=DE vil Z-flaget være sat ellers er det slettet. Hvis HL>=DE er C-flaget slettet og hvis HL<DE er C-flaget sat.

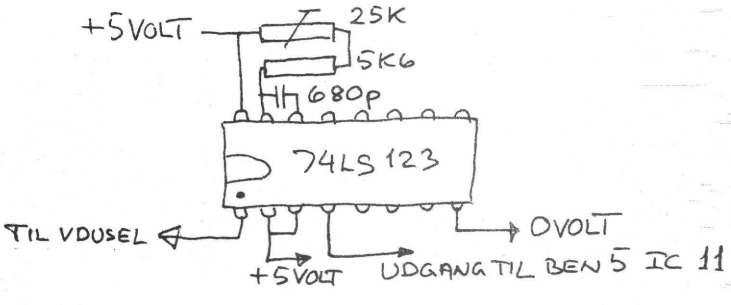
Alt dette på kun 4 bytes !

"SNEPLOV"

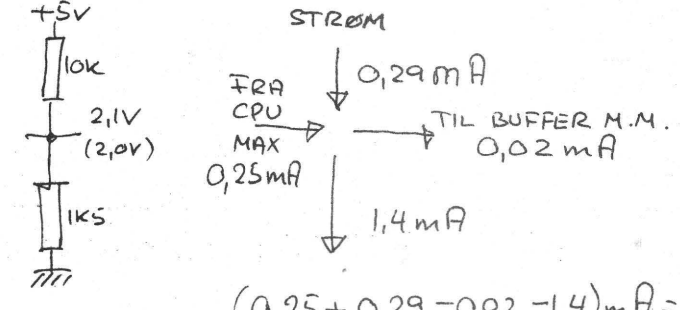
Sneploven bliver brugt i forbindelse med med IC 11 (på Nascom I) til at forøge VDUSEL blanking time for at eliminere 'sne' på skærmen under memory access til videoram. (se tegning).

Den hurtigste metode til denne konstruktionsændring er at montere en 14 bens dil sokkel på et stykke vero-board sammen med en 16 ben sokkel. Derefter forbinde udgangen af 74LS123 med ben 5 på IC 11. Lod derpå alle ben, undtagen ben 5, til på undersiden af vero-boardet. Forbind ben 5 på soklen til indgangen på 74LS123. Husk at parallelforbind spændingsforsyning til 74LS123. Dette lille modul passer fint ind på IC 11 plads.

Justeringen foretages under tabulering 0 FFFF, hvorunder man drejer potmeteret indtil stregerne netop forsvinder.



MEDLEMSREAKTION PÅ NN 4 S.12 !



$(0{,}25 + 0{,}29 - 0{,}02 - 1{,}4)\,mA = -0{,}88\,mA$??? Hvor f..... kommer de elektroner fra?

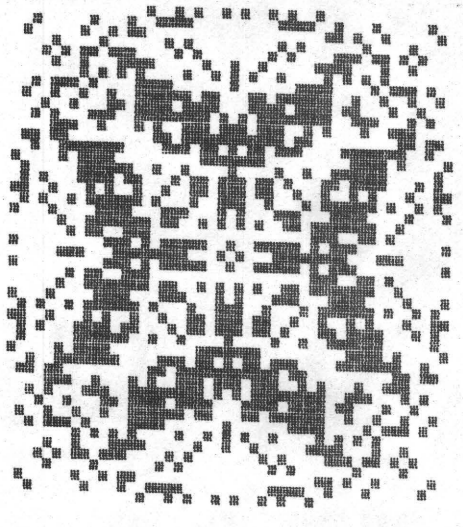
C.L.

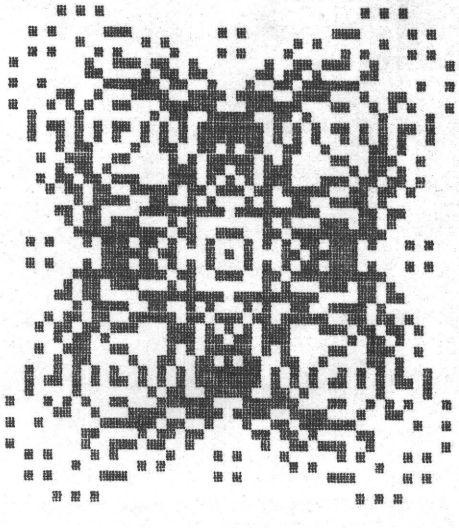
```
PROGRAM Anagram;

  VAR
    i,len: INTEGER;
    bogstav,nyt.ord: ARRAY [1..10] OF STRING[10];
    brugte: ARRAY [1..10] OF BOOLEAN;
    ord: STRING[10];

  PROCEDURE Permuter (taellerned: INTEGER);
    VAR j: INTEGER;
    BEGIN
      IF taellerned=0 THEN
        BEGIN
          writeln;
          FOR i:=len DOWNTO 1 DO
            write(nyt.ord[i])
        END
      ELSE
        BEGIN
          FOR j:=1 TO len DO
            BEGIN
              IF NOT brugte[j] THEN
                BEGIN
                  brugte[j]:=true;
                  nyt.ord[taellerned]:= bogstav[j];
                  Permuter(taellerned-1);
                  brugte[j]:=false
                END
            END
        END
    END;

  BEGIN
    write('Skriv et ord ---> ');
    readln(ord);
    len:= length(ord);
    FOR i:=1 TO len DO
      BEGIN
        brugte[i]:= false;
        bogstav[i]:= mid(ord,i,1);
      END;
    Permuter(len)
  END.
```





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	Søren Sørensen Højlundvej 13 3500 Værløse 02 48 31 01
Teknisk red.	Ole Hasselbalch Vibeskrænten 9 2750 Ballerup 02 97 70 13
Medlemsmøder	Erik Hansen Lyngby Kirkestræde 6,1 2800 Lyngby 02 88 60 55 (mellem 8 og 15.30

Henvendelse til foreningen:

Indmeldelse, adresseændringer o.l. til **kasseren**
Programbibliotek til **næstformanden**

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

Indmeldelsesgebyr: 25,00 kr.
Kontigent til den 1.7.81: 80,00 kr.
Reduceret kontigent for studerende: 65,00 kr.
(Fremsend gyldigt studiebevis eller kopi)

Redaktionen sluttet den 19.04.81

Oplag: 160