

NASCOM Z80 NYT

NASCOM BRUGERGRUPPE
2730 Herlev

Sidevolden 23
Giro 6742602

NR: 3
2. årgang

Marts 1981.

Vores "urafstemning" er afsluttet med følgende resultat: For stemte 73, imod stemte 7. Det har medført, at noget af dette blad er udskrevet på den nye printer. Jeg takker for den høje stemmeafgivelse - det var rart.

Fra næste nr. af åbner vi en brevkasse, hvor en Nascomspecialist vil besvare tekniske spørgsmål omkring Nascom 1 og 2. Spørgsmålene må ikke være alt for rummelige, som f.eks. "Min computer virker ikke !?!". Spørgsmålene må gerne være en opfordring til udvikling, som f.eks. "Kan jeg få mere end 32 K på mit RAM A-kort ?". Indsend dine spørgsmål til foreningens adresse, mærket teknisk brevkasse, allerede nu.

Den 25.3 og 13.5 vil jeg holde "ABENT HUS" aften på Sidevolden. Jeg lukker kælderindgangen op kl. 19.00. Alle er velkomne, men husk drikkevarer, hvis du skulle blive tørstig. Meningen er, at vi (I) skal snakke frit sammen om fælles interesser. Printeren vil selvfølgelig også være i funktion. Der er plads til ca. 20 mennesker - og du ved jo - der er altid plads til en til, der bruger NASCOM.

Vi har indledt et samarbejde med ABC 80 klubben, der har adresse i Tåstrup. Foreløbig har vi holdt et møde, hvor vi udvekslede ideer og erfaringer. Vi aftalte, at vi ville udveksle blade og programmer.

Husk medlemsmøde i Lyngby den 5.4.81 kl.12.30

si'r Asbjørn

INDHOLD

side 2	Interrupt
side 11	Sprogstamtræ
side 12	Båndoptager over RS232
side 13	Ordlisten
side 14	Karakterpakker
side 16	De bevægelige højtidsdages placering
side 17	Tegning !
side 18	Medlemsreaktioner på stemmeseddel
side 19	TV --> Monitor
side 21	Netfilter, Morse og Tilbud

INTERRUPT!!

=====

I forbindelse med sidste medlemsmøde vil jeg her oversætte en artikel fra det engelske tidsskrift Personal Computer World (juni 80). Det skyldes, at jeg ikke kunne være tilstede i Lyngby under Steen Lærkes foredrag om interrupt den 1.3. (og jeg heller ikke kunne få andre til at tage et indholdsreferat).

En interrupt mulighed tillader en CPU at udskyde programafviklingen, mens den udveksler signaler med den omkringliggende verden. Z80 i forbindelse med Mosteks 3881 parallel ind- og udgang (PIO) tilbyder alsidige og kraftfulde interrupt muligheder - specielt i Z80's interrupt mode 2. De mange muligheder der tilbydes betyder, at man behøver kompleks programmering af både CPU og PIO. Så meningen med denne artikel er frembringe en rimelig indføring i brug af interrupt. Samt at give et demonstrationsprogram, der kan køre direkte på en Nascom. Det er ikke hensigten, at alle sider af emnet skal belyses, så smid ikke manualen for PIO'en væk!

Hvorfor skal man bruge interrupt ?

=====

Brug af interrupt kræver ekstra software og ofte også ekstra hardware. Og programmer med interrupt er betydelig sværere at fejlfinde når og hvis der opstår fejl. Så hvorfor bruge det overhovedet ? Et simpelt svar er, at det kan være mest formålstjenligt og ofte den eneste metode til at behandle en række praktiske situationer, specielt hvis man ønsker at kommunikere med den omkringstående verden. En væsentligt forudsætning for interrupt er, at den skal kunne afbryde en igangværende programafvikling, udføre et andet program for derefter at vende tilbage til det oprindelige igen, hvor det så fortsætter, som om intet var håndt. En simpel benyttelse af interrupt kan vises ved hjælp af et eksempel. Det er relativt let at skrive et program med et tidsloop, som omdanner din mikro til et digitalur - et dyrt et!! - og det bruger din CPU hele tiden. Men ved f. eks. at danne et udvendigt signal hvert sekund, der udfører et interrupt på CPU'en, kan den udføre en interrupt service rutine (ISR), hvor den efterlader CPU'en til andre ting det meste af tiden. En udvendig oscillator eller et tællekredsløb kunne danne interrupt-signalet. I andre sammenhænge kunne interruptet sættes i gang af

"udlæsning klar"-signal fra en A/D konverter, hvorefter spændingen læses ind i hukommelsen under ISR, hvorefter hovedprogrammet behandler de indkomne data.

Udvalget af mulige anvendelsesmåder er ubegrænset, men alle indvolverer en kommunikation mellem computeren og et eller andet udvendig apparat. CPU'en reagerer meget hurtigt på et interrupt-ønske (10 mikrosekunder før ISR er i gang), og selv en Concorde bevæger sig kun 1 cm på 10 us.

Kombinationen mellem Z80 og PIO.

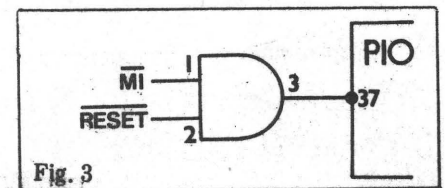
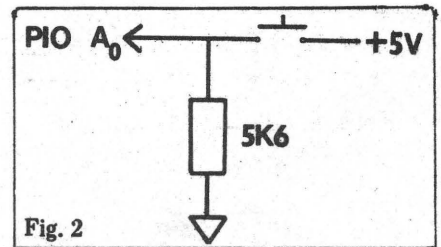
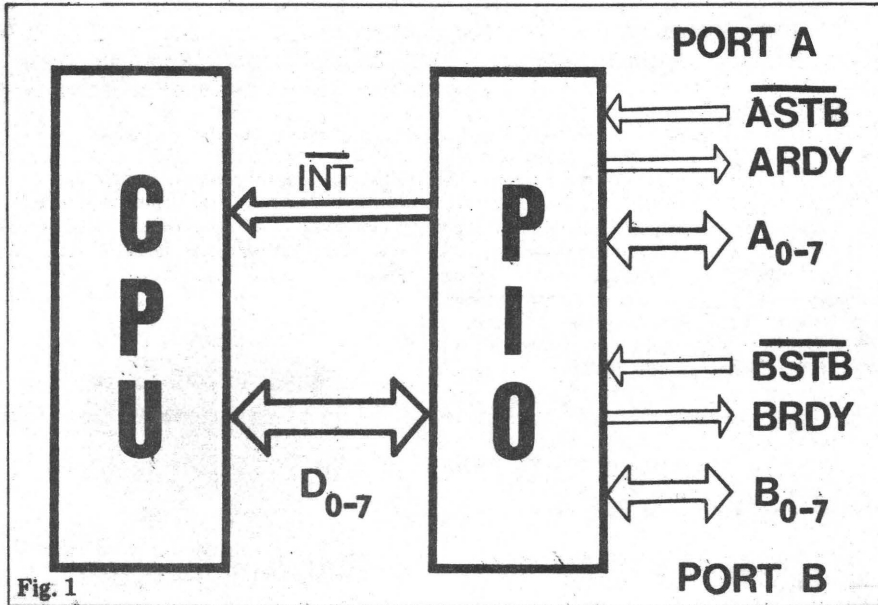
=====

Vi skal nu se, hvordan kombinationen mellem Z80 og PIO'en fungerer ved interrupt, og hvilken software det kræver. Begrænser dog emnet til interrupt mode 2. Når en bestemt betingelse fra den ydre verden bliver opfanget af PIO'en - det kan være en aktiv strobe impuls fra \overline{ASTB} eller \overline{BSTB} eller et bestemt logisk niveau på databussen på en af portene af PIO'en - vil PIO'en, hvis den er den er "enable", danne et \overline{INT} signal (se fig. 1). CPU'en må også være i stand til at modtage et interrupt; når den modtager interruptet, vil den færdiggøre den igangværende instruktion og sætte programtælleren på stakken. (Det vil sige adressen på den næste instruktion i programmet). Derefter vil den hoppe til den ønskede ISR. Den vil udføre ISR og derefter returnere til det oprindelige program og fortsætte. En af PIO'ens vigtige roller er ved en intelligent kontrol, at afgive \overline{INT} -signaler til CPU'en.

Den nødvendige software må 1) definere den ydre natur ved hvilken, der skal afgives et \overline{INT} -signal, 2) definere en 16 bit pegepind til startadressen på ISR (den mest betydende halvdel i CPU'ens i register, og den mindst betydende halvdel i PIO'en som interrupt vektor), 3) åbne og lukke af for muligheden for interrupt i både CPU og PIO, når det er nødvendigt. En fornuftig metode til programmering er at dele den nødvendige software op i fem dele.

- a) programmere PIO'en (SETPIO)
- b) programmere CPU'en
til interrupt (SETCPU)
- c) hovedprogram (MAIN)
- d) interrupt rutiner (ISR)
- e) ISR pegetabel (IPT)

En typisk oversigt over indholdet af disse afdelinger finder du i tabel 1. PIO'en kan danne interrupt fra port a og/eller port B, og hver port har sin egen interruptvektor. Port A er over port B i serviceniveau ved interrupt - dette må tages i betragtning, når begge porte er programmerede til at danne et INT-signal. På dette sted er det nok bedre, at gennemse et aktuelt program, end at generalisere videre om interrupt.



INTRPT
=====

Programmet INTRPT er skrevet til med ovenstående opdeling af software i baghovedet. Det er skrevet til en Nascom med NASSYS monitor, men indeholder generelle træk for Z80. Programmet fylder mindre end 100 bytes. For de der ikke er bekendt med Nascom maskinen, er port 4 og 5 udgange og port 6 og 7 er kontrolporte til henholdsvis port A og B.

Programmet udskriver uendeligt de små bogstaver i alfabetet og skifter linie på skærmen; ved et interrupt, som dannes ved at trække bit 0 i port A høj, udskrives et f-tegn af ISR. Figur 2 er tilstrækkeligt til demonstration af interrupt, men vil sikkert give flere interrupt ved hver tryk på grund af kontaktprel. En almindelig regel er, at sikre sine input til PIO ved at gate signalerne (f.eks. ved en plip-flop).

INTRPT er ikke noget særskilt spændende program, men det illustrerer en simpel form for interrupt og giver basis for ydeligere eksperimenter. Programmet er godt kommenteret, så videre forklaring er ikke nødvendig. Det udføres fra D00, eller

efter en hardware reset af PIO'en (se nedenunder), kan det udføres fra d20.

PIO'ens kontrolord er skematisk angivet i tabel 2. INTRPT bruger mode 3, som er særlig brugbar ved positions anvendelse. I mode 3 kan PIO'en kontrollere skift af niveauer på enkelte bit eller på resultatet af logiske operationer (or eller and) på enhver samling af bit på hver port A eller B. STB og RDY bliver ikke benyttet i mode 3. Dette interrupt mode kræver større programmering end 0, 1 og 2, nemlig en definering af I/O kontrolord og muligvis en afmaskning.

RESET PIO'en
=====

Z80's PIO har ikke nogen reset pind, men kan resettes ved enten at slukke for maskinen (ikke særlig attraktivt!) eller ved at anvende et $\overline{M1}$ signal i fravær af \overline{RD} eller \overline{IORQ} signaler, en metode, der kræver ekstra gates. Nascom 2 bruger en AND-gate (figur 3) for at give umiddelbar reset af CPU og PIO; Nascom 1 har ikke denne facilitet, skønt den kan tilføjes. Hvis PIO'en ikke er resettet ordentlig, kan det være, at den ikke reagerer på kontrolordene på forventet måde. Det er ikke en fejl i PIO'en, men en konsekvens af forudgående programmering! F.eks. kunne den vente på et I/O ord, da det foregående ord, der var sendt, var FF (tabel 2). Dette problem kan undgås ved resette Nascom 2 (eller modificere Nascom 1) før udførelsen af det nye program, eller ved at bruge en softwarerutine, der i begrænset omfang resetter PIO'en. Sådan to software resetrutiner bliver benyttet i INTRPT.

Rutinen

```
3E 0F          LD A,0FH
D3 06          OUT(06),A
D3 07          OUT(07),A
```

sikre, at PIO'en reagerer på det næste kontrolord, der bliver sendt til enten port A eller port B. Dette er "dummy"-instruktioner, der udelukkende sikre, at den næste instruktion ikke bliver opfattet som del af kontrolord eller maske. De skal derfor foranstilles de rigtige PIO-instruktioner.

Den anden rutine

```
06 02          LD B,02
21 XX XX START: LD HL,LOOP
E5             PUSH HL
ED 4D          RETI
XX XX 10 F8 LOOP: DJNZ START
```

indeholder 2 RETI-instruktioner, som sikre at hverken port A eller port B afventer færdiggørelsen af tidligere ISR. Instruktionen er at anbefale, hvis du ikke har en hardware reset eller ikke ønsker at benytte en sådan. Skønt en hardware reset er at foretrække før en kørsel, specielt hvis man har flere end en PIO i brug.

Ønsker/ønsker ikke interrupt

=====

Interrupt må være ønsket (enable) eller ej ønsket (disable) for både CPU'en og PIO'en hver for sig. CPU'ens status for interrupt er indeholdt i to interne flip-flop, IFF1 og IFF2, som begge er 0 (reset), når interrupt er uønsket; og begge er 1 (set), når interrupt er muligt. Disse flag bliver sat af EI-instruktionen og slettet af DI-instruktionen. De bliver også automatisk resettet, når CPU accepterer et ønske om interrupt.

PIO'en bliver enable ved at sætte b7=1 i interrupt kontrolordet (tabel 2) og disable ved at sætte b7=0. Et simpelt eksempel på dette er indeholdt i ISR, hvor benyttes EI og RETI. Hvis flere interrupt er anbragt inde i hinanden, kræver det, at man anbringer EI i starten af ISR, som derved tillader højere prioriterede interrupt adgang til CPU'en. I INTRPT bliver interruptønsker afvist under ISR, men RETI-instruktionen meget vigtig for at åbne for interrupt efter ISR.

Gem dine registre.

=====

Da et interruptønske kan forekomme på et vilkårligt tidspunkt, er det vigtigt, at indholdet af CPU'ens registre ikke mistes og at de det stede igen, når programmet genoptages. Undladelse af dette vil forårsage totalt kaos. Man må omhyggeligt sørge for at gemme og gendanne registrene, som bliver brugt i både det oprindelige program og i interruptrutinen. Sektion d) i tabel 1 illustrerer en måde at gøre det på, men læg mærke til, at den ikke gemmer registrene fra ISR. I visse tilfælde er det meget nemt at sikre sine registre, men i f. eks. BASIC programmer (og i andre højniveausprog) kan man være sikker på, at alle registre bliver brugt i Z80 og derfor handle i overensstemmelse hermed. Det kan med nogen ret hævdes, at gemme og gendanne alle registre, dvs. både skygge- og almindelige registre i Z80, omhyggeligt. Det vil bringe en vis standard over din software (skønt det i flere tilfælde vil være overflødig).

Andre interrupts mode.

=====

I de fleste tilfælde er PIO'ens databus A0 - A7 og B0 - B7 benyttet til overførsel af data til og fra udvendigt udstyr, derfor kan mode 3 ikke benyttes ved undersøgelse af interruptønsker. I disse tilfælde bruges strobe og ready linierne fra PIO'en til at videregive interruptønsker i mode 0,1 og 2. Den aktuelle timing er angivet i diagrammer i manualen for PIO'en og det er ret simpelt at følge de almindelige programmeringsregler ved interrupt i disse modes, skønt det nok er nødvendigt med noget ekstra logik, for at tilslutte udvendigt udstyr til PIO'en.

For at tage et enkelt eksempel, en PIO i mode 1 (indgangs mode) og et aktivt STB signal bliver dannet af udvendigt udstyr, får PIO'en til anbringe data på udgangs registrene, og en stigende kant af STB pulsen danner et interrupt. Readylinien RDY bliver sat lav af PIO'en og forbliver inaktiv indtil CPU'en har læst porten under ISR. Den stigende kant af readysignalet kan da bruges til at fortælle, at CPU'en er færdig og flere data kan sendes til PIO'en. Ved at bruge både det udvendigt dannede strobe og det i PIO'en dannede ready signal, kan man opnå "hand-shaking" forbindelse mellem microen og den udvendige verden, derved bliver kommunikationen ^{mellem} disse enheder meget hurtig.

Slutning.

=====

Det gennemgæede er kun et kort blik og ikke en komplet gennemgang af den kraftfulde forbindelse mellem Z80/PIO ved interrupt. Men det vil måske tilskynde flere til at kigge en ekstra gang på manualen over PIO'en og derved bruge interrupt noget oftere. En af fordelene ved mikrocomputere er netop den lette tilgang til "real-time" computing, som man ikke får ved større flerbruger maskiner.

TABEL 1

Z80 CPU/PIO interrupt program.

Det er godt, hvis man bryder sit program ned i selvstændige afdelinger. En mulighed er vist nedenunder og i det følgende demo program INTRPT. PIO'ens kontrolord bliver vist i tabel 2.

SETPIO

I denne del programmeres PIO'en til den ønskede interrupt betingelse. PIO'en skal først resettes (se tekst). De følgende kontrolord bliver sendt til PIO'en. 1) En interrupt vektor (b0=0), der er dem mindst betyden halvdel af ISR pointeren, 2) et kontrolord for modes (b0-b3=111)+(kun i mode 3) et I/O kontrolord, 3) et kontrolord for interrupt (b0-b3=0111) med b7=1 for ønskelig interrupt + (hvis b4=1) en interrupt maske. De i parentes markerede ord bliver genkendt af PIO'en, på grund af deres bestemte mønster, og de kan derfor sendes i vilkårlig rækkefølge. I/O ordet og masken kan tage vilkårlige værdier, men må sendes som angivet i tabel 2.

SETCPU

Denne del vælger modes og ønsker et interrupt. 1) Vælger Z80 interrupt mode 2, dvs. IM2, 2) tilføjer CPU register I den mest betydende halvdel af ISR pointeren, 3) lukker CPU'en op til interrupt, dvs. EI.

MAIN

Det er programmet, der skal kunne interruptes og skal udføres efter SETPIO og SETCPU.

ISR

Interruptrutinen skall 1) gemme registrene, der bliver brugt i hovedprogrammet og i ISR, dvs. PUSH AF, PUSH BC, 2) udføre den aktuelle rutine, 3) gendanne registrene fra hovedprogrammet, dvs. POP BC, POP AF, 4) åbne Z80 for interrupt, dvs. EI, 5) retur med PIO åben for interrupt, dvs. RETI. Angående 5) Z80 er automatisk lukket for interrupt ved overgang til ISR. I nogle tilfælde med overlappende interrupts er det nødvendigt, at genåbne CPU for interrupt.

IPT

Det er en tabel for startadresse(r) for ISR. Ved interrupt bruger CPU'en interruptvektoren og dens I register til at danne en 16 bits pegepind, der peger på en tabel, der indeholder startadressen på den ønskede ISR.

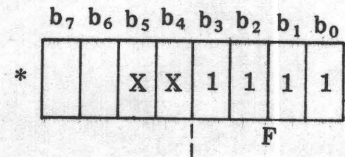
Tabel 2

Kortfattet oversigt for kontrolord til Z80's PIO.

Valg af mode:

Mode b7 b6

0	0	0	Udgang
1	0	1	Indgang
2	1	0	Begge veje
3	1	1	Kontrol

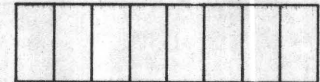


I/O (mode 3):

bn=0 bn=1

udg. indg.

(Dette må være det næste ord efter valg af mode 3)



Interrrupt kontrolord (ICW):

bn=0 bn=1

b7 Disa Ena PIO inter.

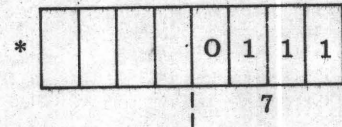
b6 OR AND funktion

b5 Lav Høj niveau

b4 ingen maske følger

(b4,b5,b6 kun ved mode 3)

PIO's enabel flip-flop kan sættes og slettes ved at ændre b4,b5,b6 ved hjælp af et kontrolord.

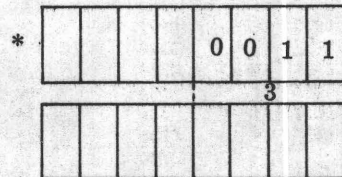


Interrupt maske (hvis b4=0 i ICW):

bn=0 bn=1

Ingen Interrupt(intelligent)

Dette må være det næste ord efter ICW, hvor b4=1.



Interrupt vektor:

Den mindst betydende halvdel af ISR startadresse. Bemærk at b0=0



*) Kontrolord der indeholder et bestemt bitmønster.

```

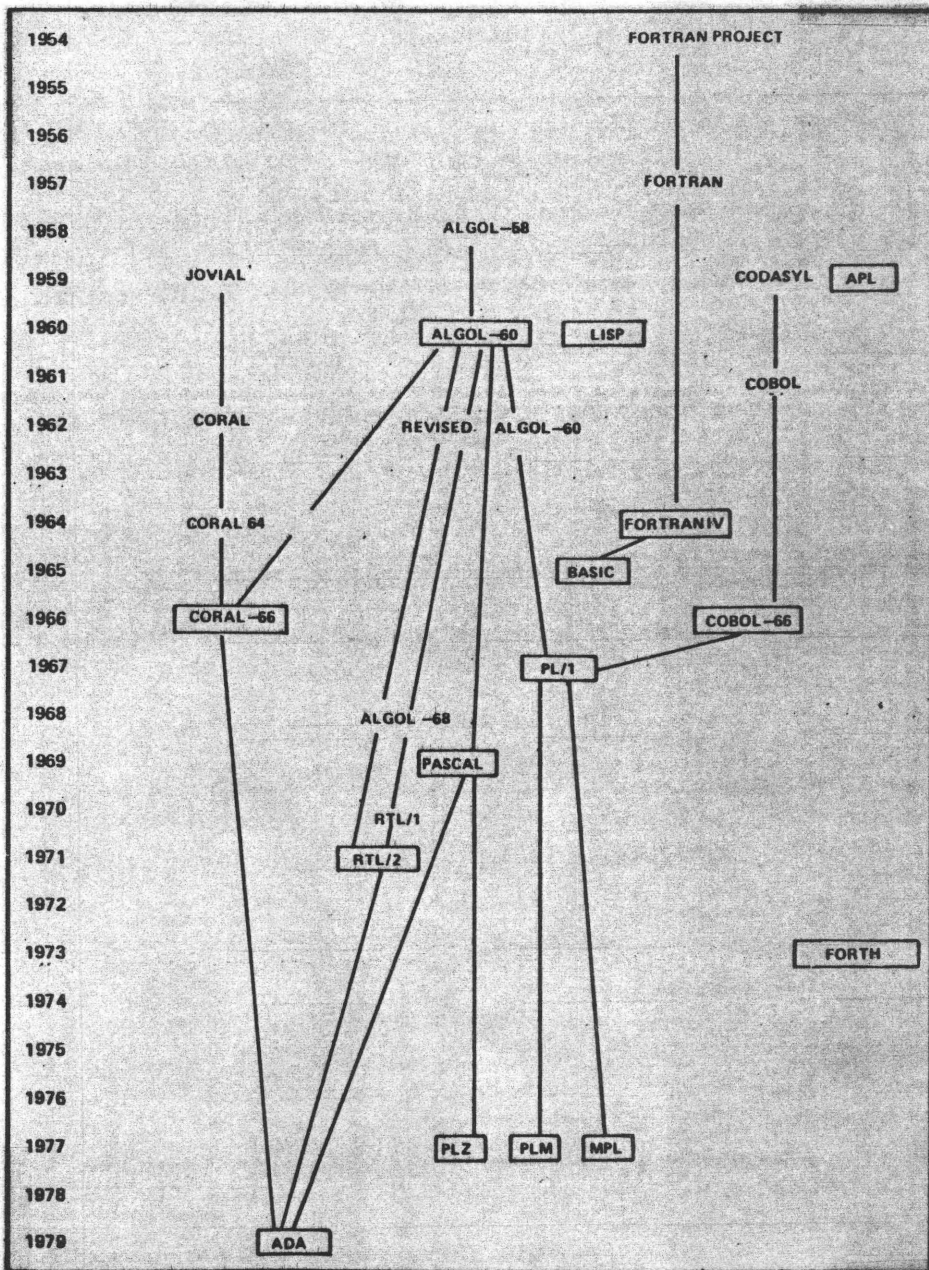
0001
0002           ;           INTRPT
0003
0004           ;Et interruptprogram til Nascom med NASSYS
0005
0006 0D00           ORG 0D00H           ;resetter PIO
0007 0D00           MEM 0D00H
0008
0009 0D00 3E0F      RESET: LD  A,0FH           ;denne del udfører en
0010 0D02 D306      OUT  (06),A           ;software reset af
0011 0D04 D307      OUT  (07),A           ;PIO'ens port A og B
0012 0D06 0602      LD  B,02           ; (se teksten)
0013 0D08 210E0D    START: LD  HL,LOOP        ;kun port A bliver
0014 0D0B E5        PUSH HL           ;brugt i programmet
0015 0DOC ED4D      RETI           ;efter hardware reset
0016 0DOE 10F8      LOOP: DJNZ START        ;kan der startes i
0017 0D10 180E      JR  SETPIO          ;D20
0018
0019 0D20           ORG 0D20H           ;setter PIO og CPU
0020 0D20           MEM 0D20H
0021
0022 0D20 3E80      SETPIO: LD  A,80H           ;interrupt vektor:LSB
0023 0D22 D306      OUT  (06),A           ;af pegepind til portA
0024 0D24 3EFF      LD  A,0FFH          ;valg af mode 3
0025 0D26 D306      OUT  (06),A
0026 0D28 3E01      LD  A,01           ;I/Oord valg af a0 som
0027 0D2A D306      OUT  (06),A           ;indg,al-7 som udgang
0028 0D2C 3EB7      LD  A,0B7H          ;ICW: enabel, OR, høj
0029 0D2E D306      OUT  (06),A           ;maske følger
0030 0D30 3EFE      LD  A,0FEH          ;interrupt maske: kun
0031 0D32 D306      OUT  (06),A           ;a0 kan danne interrpt
0032
0033 0D34 ED5E      SETCPU: IM  2           ;set CPU intrpt mode 2
0034 0D36 3E0D      LD  A,0DH           ;MSB af interruptpege-
0035 0D38 ED47      LD  I,A           ;pind ind i reg. I
0036 0D3A FB        EI           ;enabel CPU for interr
0037 0D3B 1803      JR  MAIN
0038
0039 0D40           ORG 0D40H           ;hovedprogram
0040 0D40           MEM 0D40H
0041
0042 0D40 3E61      MAIN: LD  A,'a'         ; ASCII a
0043 0D42 061A      LD  B,1AH          ;26 bogstaver
0044 0D44 F7        ALPHA: RST 30H          ;skriv bogstav
0045 0D45 3C        INC  A           ;følgende bogstav
0046 0D46 D706      RCAL DELAY        ;pause
0047 0D48 10FA      DJNZ ALPHA        ;alfabetet slut ?
0048 0D4A DF6A      SCAL 6AH          ;skift linie
0049 0D4C 18F2      JR  MAIN          ;start igen
0050
0051 0D4E F5        DELAY: PUSH AF          ;pauserutine, der
0052 0D4F C5        PUSH BC           ;giver 1/4 sek. og ej
0053 0D50 0620      LD  B,20H          ;smadre registrene
0054 0D52 3EFF      SLOW: LD  A,0FFH
0055 0D54 FF        RST 38H
0056 0D55 10FB      DJNZ SLOW
0057 0D57 C1        POP  BC
0058 0D58 F1        POP  AF
0059 0D59 C9        RET
0060

```

```

0061 0D60          ORG 0D60H          ;interrupt service
0062 0D60          MEM 0D60H          ; rutinen
0063
0064 0D60 F5       ISR:  PUSH AF          ;gem register A
0065 0D61 3E23     LD   A,23H          ;ASCII for f
0066 0D63 F7       RST  30H          ;skriv det
0067 0D64 F1       POP  AF          ;gendan A
0068 0D65 D7E7     RCAL DELAY         ;pause
0069 0D67 DF6A     SCAL 6AH          ;skift linie
0070 0D69 FB       EI                ;enabel CPU interrupt
0071 0D6A ED4D     RETI               ;retur fra interrupt
0072
0073 0D80          ORG 0D80H          ;interrupt pointer
0074 0D80          MEM 0D80H          ;tabel
0075
0076 0D80 600D     IPT:  DW   ISR          ;peger paa ISR
0077
0078 0D82          END

```



Det vil den ikke. Min Nascom 2 vil ikke snakke 1200 Baud med andre datamater over RS 232 tilslutningen. Det går imidlertid godt med 110 Baud. Men hvorfor kun 110?

Et diagram kig viser, at TTY er TTY, altså beregnet for gøe gamle teletypes der højest kunne køre 110 Baud. Godt så får den 1200 Baud fra cassette. Det går heller ikke.

Men hvorfor vil UART'en (IC 20) ikke æde cassette clock'en på 1200 Baud? Vel, man må måle sig frem. Hvad er forresten 1200 Baud i menneske hertz? Jo, Baud(udtales Bo), bit pr sek, og Hertz er det samme. Og for at tage tegn pr. sek med, så vil der med et startbit, 8 databit, et checkbit og to stopbit altså gå 12 bit pr tegn. Så bliver 1200 Baud altså til 100 tegn pr sek.

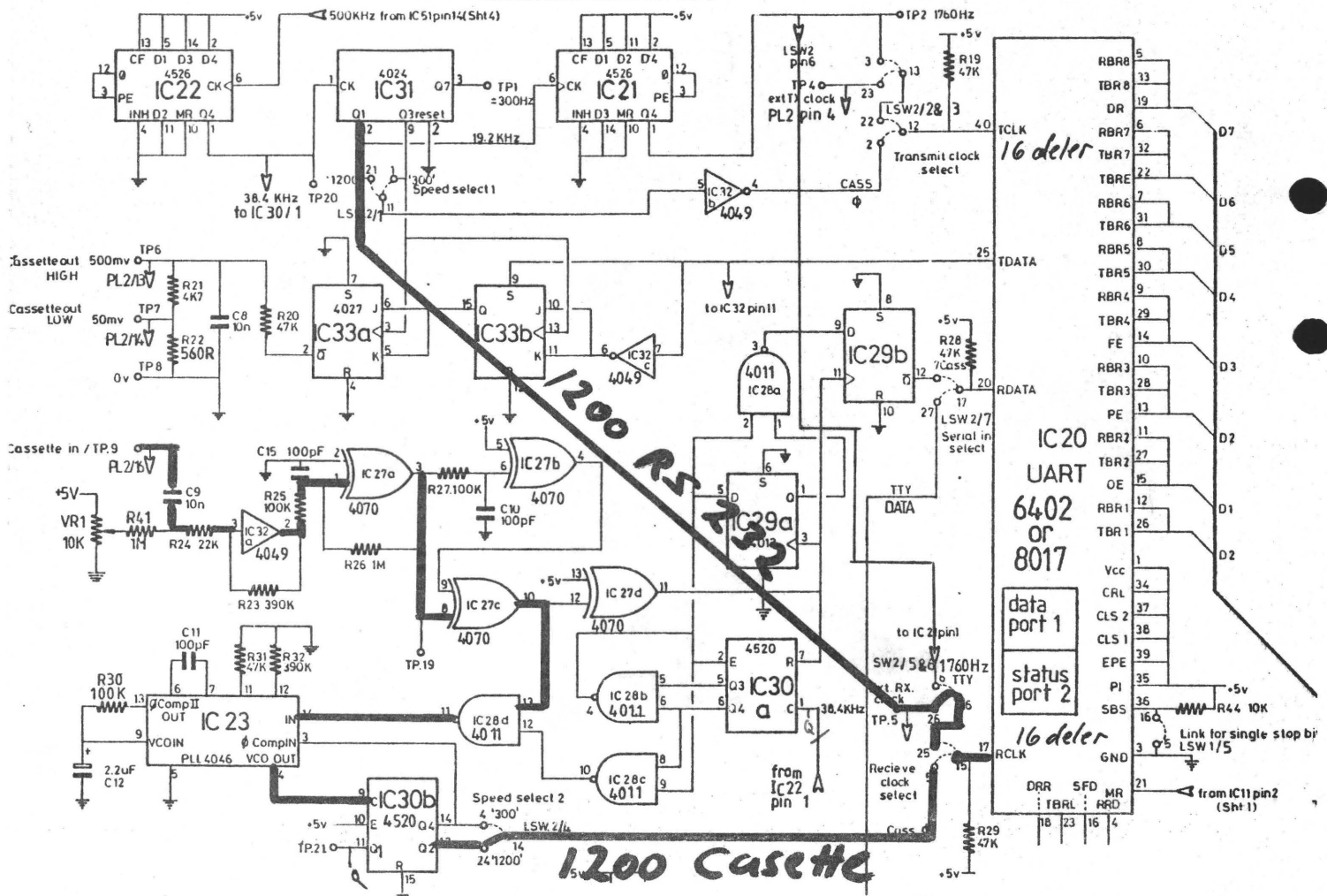
Nuvel, hvad bestiller så de høje frekvenser i min Nascom? Jeg søger efter 1200 Hz eller for den sags skyld 110 og 300 Hz. Men der er kun 19.2kHz 4800Hz og 1760Hz? Er der et skjult mysterium i UART'en? Jo, Hængemt på nederste linie i UART'ens manual står der faktisk at den har indbygget 16 deler på både RX og TX clock indgang. Godt, så forstår jeg de høje frekvenser. Men hvad er der i vejen med cassette clock'en? Oh ve Den dannes jo af data fra cassettebåndet. Uden databånd kører den i tomgang. På en frekvens 50% højere.

Løsningen er derfor at sætte TX clock til RX clock som vedlagte diagram viser.

- 1 Man lodder en ledning fra IC31 ben 2 til TP5
- 2 LSW2/ 6 op. Extern clock.
- 3 LSW2/ 5 op. Extern clock.

Så kører data ind som smør. Det vil osse gælde 1200 Baud modem.

Gerhard Lohse



micro prefix der betyder en milliontedel, f.eks. et microsekund = en milliontedel sekund.
mnemonic code data instruktioner, skrevet på en kortfattet, symbolsk eller forkortet form, der er let at huske. F.eks. kan ADD symbolisere addition.

modem et udstyr til signal omsætning ved datatransmission, navnet er en forkortelse af modulator + demodulator.

monitor et program - typisk en del af operativsystemet - der overvåger, kontrollerer eller styrer computerens funktion, f.eks. ind- og udlæsning af data.

MOS Metal Oxide Semiconductor, en fieleffect transistor med meget høj indgangsimpedans.

mother board et stort kredsløbskort hvorpå man kan isætte andre mindre kortmoduler.

N

nano en milliarddel, f.eks. et nanosekund er en milliarddel sekund.

nonvolatile (ikke flygtig) når termen benyttes om memory, betyder det, at de lagrede data ikke forsvinder, når forsyningsspændingen fjernes. En nonvolatile memory er f.eks. kassetebånd, PROM og ROM.

O

object program den binære form af et source program (source program er det, som programmøren har skrevet) produceret af en assembler eller compiler. Object program består af instruktioner i maskinkode, hvormed computeren kan arbejde.

octal talsystem med grundtal 8, bestående af tallene 0-7.

on-line et styringsforhold, der medfører at en enhed styres af en anden uden manuel indgriben. En dataterminal styres ofte direkte af en computer.

operativsystem software der gør det muligt for brugeren at kommunikere med computeren uden besvær.

P

paper tape hulstrimmel, en sædvanligvis 25 mm bred strimmel med stansede huller i en bestemt kode. Bruges som bærer af data.

parallel I/O en metode til samtidig at overføre samtlige bit i en gruppe, f.eks. en hel byte . . . det sker over parallelle ledninger en for hver bit.

Program Counter. (PC)

Et register, der indeholder information om, hvor den næste instruktion er at hente. Tælles under den "normale" programudførelse op instruktion for instruktion, men loades med et andet indhold ved branchinstruktioner.

port en kommunikationskanal igennem hvilken der kan sendes data til og fra computeren.

printer (skrivemaskine) et elektromekanisk udstyr der fødes med signaler fra computeren og skriver grafiske tegn på papir. Modsatningen er en *plotter* der kan tegne kurver.

program en sekvens computerinstruktioner, nødvendige for at kunne løse en bestemt opgave ved hjælp af computeren.

PROM Programmable Read Only Memory en ROM der kan programmeres af brugeren. Se også ROM og EPROM.

R

RAM Random Access Memory en memory hvor man kan læse og skrive et vilkårligt sted i memory'en.

real time et system kører real time når operationerne i computeren udføres så hurtigt i forhold til et fysisk system, at resultaterne kan anvendes af det fysiske system.

register midlertidig lagerplads i en processor.

reset at bringe en tæller tilbage til udgangspositionen, normal 0 stillingen.

ROM Read Only Memory en hukommelse, hvor man kun kan læse og ikke skrive data. programmeringen sker ved fremstillingen af kredsen.

RS-232 en standard der definerer de elektriske parametre for ind- og udlæsning af data fra periferenheder.

routine en sekvens data, nødvendige for at få udført en bestemt funktion.

S

seriel I/O seriel ind- og udlæsning, en metode til at overføre data mellem computer og periferudstyr en bit af gangen på en enkelt ledning.

simplex en forbindelse konstrueret til at overføre data men kun i en retning ad gangen.

software programmel, en systematisk samling af programmer og tilhørende dokumentation, der anvendes af en computer til at behandle data, f.eks. operativ system, monitor, compiler, editor, hjælpeprogrammer og brugerprogrammer regnes som software.

software documentation listning af programmer og eller tekniske håndbøger manualer der fortæller, hvordan man skal anvende programmerne.

source language det sprog hvori programmøren skriver sit program, og hvorfra oversættelsen sker.

source program et program skrevet i et andet sprog end maskinsprog der kræver oversættelse af en assembler, compiler eller interpreter.

Stack Pointer.

Stacken er en LIFO-hukommelse (last in - first out). Dannes stacken i et RAM-lager, er det nødvendigt at indikere, hvor i RAM'en de sidst indlæste data befinder sig. Denne adresse er til enhver tid indeholdt i stack-pointeren.

static memory en memory RAM der ikke behøver refresh (genopfriskning).

statement i visse programsprog betegnelsen for en bestemt type instruktioner.

string en ordnet række tegn eller symboler.

subroutine et antal instruktioner der udfører en deloperation i et større program; en subroutine kan sættes i arbejde ved en kommando fra hovedprogrammet.

synchronous en metode til at overføre serielle binære data mellem en computer og periferenhed. Overførelsen sker med fast hastighed og sender og modtager er synkroniseret med samme clock.

Effektiv karakter hukommelse.

=====

Ved programmering er det ofte nødvendigt, at pakke så mange karakterer sammem på et givet område. Den almindeligste metode er omdanne teksten til 8 bit ASCII kode, og derefter gemme en karakter pr. byte. Dette giver mulighed for 256 forskellige ASCII koder, som kan gemmes i en bestemt byte.

Oftest behøver man kun at gemme karaktererne fra A til Z, 0 til 9, mellemrum og nogle få tegn.

Hvis man tager et sæt på 40 karakterer, kan vi behandle dem på en sådan måde, at 3 karakterer kan gemmes i 16 bit (2 bytes). Dette forøger hukommelsesområdet med godt 30 %.

Man gør det ved at behandle de tre karakterer, som var de tal med basen 40.

Antag at A=3, B=2 og C=1, da vil tallet ABC(40) være lig med $3 \times 40 \times 40 + 2 \times 40 + 1 = 4881$. Dette tal kan rummes i et 16 bits ord.

Maximumværdien der kan gemmes er $39 \times 40 \times 40 + 39 \times 40 + 39 = 63999$, som er mindre end 65535, der er max for et 16 bits tal.

De to subrutiner er skrevet i Z80 kode, men kan principielt overføres til vilkårlige datamaskiner.

ENCODE tager en 3 byte lang ASCII streng, hvis adresse er i BC og returnerer det i base 40 omdannede tal i HL registret.

DECODE subrutinen tager tallet i base 40 ind i HL og returnerer de tre bytes ASCII streng startende ved adressen BC peger på.

Hvordan virker det så?

ENCODE subrutinen renser først HL, derefter kaldes FIND, der finder karakteren, som BC peger på og returnerer dets nummer fra tabellen i register A. BC opskrives til næste karakter.

DE sættes til 1600 (40×40) og subrutinen MULT kaldes. Den multiplicerer indholdes af register A med DE og anbringer resultatet i HL.

Denne proces fortsætter for de to øvrige karakterer med DE sat til 40 og til 1.

De tre karakterer er nu kodet.

DECODE subrutinen udfører den modsatte funktion af ENCODE.

DE er først sat til 1600, hvorefter TRANS bliver kaldt. Denne subrutine foretager en heltalsdivision af HL med DE, efterladende resultatet i A og resten i HL. Den finder så den rigtige karakter i TABLE og fører den til adressen, hvorpå BC peger. BC opskrives og det hele gentager sig med DE sat til 40 og 1.

```

0001
0002           ;      Karakterpakker.
0003
0004 0D00           ORG  0D00H
0005 0D00           MEM  0D00H
0006
0007           ;BC peger paa en 3 karakter lang ASCII streng
0008           ;Ved retur indeholder HL base 40 tallet
0009
0010 0D00 210000    ENCODE: LD   HL,0           ;ryd HL
0011 0D03 D714      RCAL  FIND           ;1. karakter
0012 0D05 114006    LD   DE,1600        ;40*40
0013 0D08 D71F      RCAL  MULT           ;DExA
0014 0D0A D70D      RCAL  FIND           ;2.karakter
0015 0D0C 112800    LD   DE,40
0016 0D0F D718      RCAL  MULT
0017 0D11 D706      RCAL  FIND           ;3. karakter
0018 0D13 110100    LD   DE,1
0019 0D16 D711      RCAL  MULT
0020 0D18 C9        RET                ;slut
0021

```

```

0022 0D19 0A      FIND:  LD  A,(BC)      ;hent karakter
0023 0D1A 03      INC  BC                ;peg paa naeste
0024 0D1B C5      PUSH BC               ;gem
0025 0D1C E5      PUSH HL               ;paa stakken
0026 0D1D 012800  LD  BC,40             ;tabellaengde
0027 0D20 21560D  LD  HL,TABLE         ;
0028 0D23 EDB1    CPIR                  ;find karakteren
0029 0D25 79      LD  A,C
0030 0D26 E1      POP  HL
0031 0D27 C1      POP  BC
0032 0D28 C9      RET
0033
0034 0D29 A7      MULT: AND  A           ;rens flag
0035 0D2A C8      RET  Z               ;slut hvis 0
0036 0D2B 19      ADD  HL,DE
0037 0D2C 3D      DEC  A               ;nedtaeller
0038 0D2D 18FA    JR   vULT            ;om igen
0039
0040              ;HL indeholder base 40 tallet
0041              ;BC peger paa resultat buffer
0042
0043 0D2F 114006   DECODE: LD  DE,1600   ;40x40
0044 0D32 D70B    RCAL TRANS          ;1. karakter
0045 0D34 112800  LD  DE,40
0046 0D37 D706    RCAL TRANS          ;2. karakter
0047 0D39 110100  LD  DE,1
0048 0D3C D701    RCAL TRANS          ;3. karakter
0049 0D3E C9      RET
0050
0051 0D3F AF      TRANS: XOR  A        ;rens A
0052 0D40 ED52   TRANS1: SBC  HL,DE   ;fratraek
0053 0D42 3803   JR   C,DONE         ;hop ved overflow
0054 0D44 3C      INC  A              ;inc taeller
0055 0D45 18F9   JR  TRANS1          ; om igen
0056 0D47 19      DONE: ADD  HL,DE    ;genopret vaerdi
0057 0D48 E5      PUSH HL
0058 0D49 217E0D LD HL,TABEND        ;slutning af tabel
0059 0D4C 5F      LD  E,A             ;flyt A til DE
0060 0D4D 1600   LD  D,00            0
0061 0D4F ED52   SBC  HL,DE          ;hent adressen
0062 0D51 7E      LD  A,(HL)          ;hent karakteren
0063 0D52 02      LD  (BC),A          ;gem den
0064 0D53 03      INC  BC              ;peg paa naeste
0065 0D54 E1      POP  HL
0066 0D55 C9      RET
0067
0068 0D56 20414243 TABLE: DB  ^ ABC ^
0069 0D5A 44454647 DB  ^ DEFG ^
0070 0D5E 48494A4B DB  ^ HIJK ^
0071 0D62 4C4D4E4F DB  ^ LMNO ^
0072 0D66 50515253 DB  ^ PQRS ^
0073 0D6A 54555657 DB  ^ TUVW ^
0074 0D6E 58595A31 DB  ^ XYZ1 ^
0075 0D72 32333435 DB  ^ 2345 ^
0076 0D76 36373839 DB  ^ 6789 ^
0077 0D7A 302E2C3F DB  ^ 0.,? ^
0078 0D7E 00      TABEND: NOP
0079
0080 0D7F      END

```

```

10 CLS
20 PRINT"DE BEVAEGELIGE HØJTIDERS PLACERING"
30 REM FRA DEN GREGORIANSKE KALENDERS INDFØRSEL
40 REM I DANMARK DEN 1.3.1700 √√
50 DIM R(170):PRINT
60 INPUT"HVILKET AAR ØNSKES";AA
70 IF AA/400=INT(AA/400) GOTO110
80 IF AA/100=INT(AA/100) GOTO120
90 IF AA/4=INT(AA/4) GOTO110
100 GOTO120
110 S=1
120 FOR J=32 TO 59+S:R(J)=J-31+.2:NEXT
130 L=J:N=59+S
140 FOR J=L TO L+30:R(J)=J-N+.3:NEXT
150 L=J:N=90+S
160 FOR J=L TO L+29:R(J)=J-N+.4:NEXT
170 L=J:N=120+S
180 FOR J=L TO L+30:R(J)=J-N+.5:NEXT
190 L=J:N=151+S
200 FORJ=L TO L+15:R(J)=J-N+.6:NEXT
210 A=AA-19*INT(AA/19):B=INT(AA/100)
220 C=AA-100*INT(AA/100):D=INT(AA/400)
230 E=B-4*INT(B/4):G=INT((8*B+13)/25)
240 H1=19*A+B-D-G+15:H=H1-30*INT(H1/30)
250 M=INT((A+11*H)/319):I=INT(C/4)
260 K=C-4*INT(C/4):F1=2*E+2*I+M+32-K-H
270 F=F1-7*INT(F1/7):M0=INT((H-M+F+90)/25)
280 D1=H+F+M0+19-M:D0=D1-32*INT(D1/32)
290 IFM0=3 THEN P=D0+59+S
300 IF M0=4 THEN P=D0+90+S
310 PRINT:PRINT
320 PRINT "FASTELAVNS SØNDAG ";R(P-49)
330 PRINT "PAASKE SØNDAG ";R(P)
340 PRINT "STORE BEDEDAG ";R(P+26)
350 PRINT "KR.HIMMELFARTSDAG ";R(P+39)
360 PRINT "PINSESØNDAG ";R(P+49)
370 PRINT:PRINT:PRINT

```

Ok

RUN

DE BEVAEGELIGE HØJTIDERS PLACERING

HVILKET AAR ØNSKES? 1981

FASTELAVNS SØNDAG	1.3
PAASKE SØNDAG	19.4
STORE BEDEDAG	15.5
KR.HIMMELFARTSDAG	28.5
PINSESØNDAG	7.6

R E T T E L S E T I L N N 2

Stock car racing - side 17:

Tilføjelse: 21 M=C:IF M =<2 THEN M=3

Æncring : 37 SCREEN C,3:PRINT 08:GOSUB135

Med venlig hilsen Per


```

Ø REM ***** Tegning *****
1 REM *** Per Busk Jepsen ** 13.1.1981 **
1Ø CLS
2Ø R=1:L=1:A=Ø
4Ø I=INP(Ø)
45 IF I=255 THEN 4Ø
5Ø IF I=223 THEN 5ØØ
55 IF I=251 THEN A=Ø:GOTO 5ØØ
6Ø IF I=239 THEN L=L+1:GOTO 1ØØ
7Ø IF I=254 THEN L=L-1:GOTO 1ØØ
8Ø IF I=253 THEN R=R-1:GOTO 1ØØ
9Ø IF I=191 THEN R=R+1
1ØØ IF R>95 THEN R=95
1Ø2 IF R<1 THEN R=1
1Ø4 IF L>44 THEN L=44
1Ø6 IF L<1 THEN L=1
112 IF S=Ø THEN RESET(T1.L1)
113 S=Ø:IF POINT(R,L)=1 THEN S=1
115 SET(R,L)
12Ø T1=R:L1=L:GOTO 4Ø
5ØØ IF A=Ø THEN XØ=R:YØ=L
51Ø A=1:X1=R:Y1=L:GOSUB 1ØØØ
52Ø SET(R,L):GOTO 4Ø
1ØØØ Her begynder Anders' subrutine fra NN nr.8 side3

```

Det lille program gør flittigt brug af INP(Ø) samt den fine subrutine fra NN nr8 af Anders som skal tilføjes. Efter RUN ses øverst i venstre hjørne en SET og den kan nu styres saaledes:

```

ENTER - til venstre
LF CH - til højre
SHIFT - ned
CSBACK- op

```

og man kan nu prøve sine kreative evner som tegner.

Et tryk paa @ betyder: tegn en linie fra sidste pos. til nuværende pos. (pos. bliver sat til X1/Y1)

Et tryk paa - betyder: nyt startpunkt, ingen linie fra sidste pos. (pos. bliver sat til XØ/YØ)

Pas forøvrigt paa ikke at trykke SHIFT/ENTER samtidig. da man saa selvfølgelig hopper ud af kørslen.

Til sidst et par generelle bemærkninger. Prøv i saa vidtstrækning som det er muligt at tilrettelægge de af klubben arrangerede kurser som brevkurser el.lignende. saa vi alle kan faa glæde af klubbens arbejde og ikke kun en snæver inderkreds fra "staden". Idet hele taget. faa saa meget som muligt, tips ideer og alt muligt paa papir. Det er den eneste kontakt vi har med klubben, os der ikke bor paa Sjælland. Dette ikke ment som en kritik af bladet, det bliver bedre og bedre.

Jeg formoder at indsendte bånd til listning IKKE bliver kopieret og gennemgået af andre. undtagen hvis man giver lov til det. (her er også tænkt på operatoren)

Jeg har et lille forslag til maskinkodning.

Er der ikke en som kan lave nogle små programmer, som praktisk supplerer det teoretiske (tryk fejl) stof som i sender? Hvis man arbejder med b.opt under et program kan man komme ud for at det griber ind i programmet. Det kan undgås ved at skrive DOKE 3189,1926 som lukker for indput fra tape. virkningen hæves ved DOKE 3189,1922

Zees artikler og indlæg er ofte mere kryptiske, end godt er. Overvej undertegnedes situation: uddannet elektroniktekniker, underviser af Nascom I m/ 16K RAM og ATNC BASIC samt ZEP ASSEMBLER. Systemet har været flittigt brugt i ca 1 1/2 år. Kort sagt jeg er ikke en ren novice. Og så læser man følgende: "jeg har med interesse læst Oks artikler om at få b.opt til at køre lidt efter. for at få G. til at virke".

HVAD BETYDER DET EGENLIG?!

P.S. - OG DER ER OGSÅ ANDRE EKSEMPLER!

God ide med printer.

En særlig hilsen til de, der gør det muligt at udsende det gode medlemsblad.

Er der nogen der ved hvorledes alm. TV. ombygges til vidio ?

Er der nogen der har forbindelser til firmaer der sælger billige indbygningskasser ?

trød med beskrivelser og diagram for EPROM-brander?

Nascom - nyt er et indmærket blad på lang, det ikke kun nummerer sig til de første - forsat med okt. , det nemlig hitem og tak

Håber næste medlemsblade
Deres annoncer vil
være hyggelig!

Oks syns jeg, bladet er lidt mærket. Jeg har selv lavet buffer til min NASCOM I samt 48K RAM.

Regner med at bestille buffer til bladet.

Universal fjernsyn til monitor og tilslutning til Nascom.

Når man har købt sig en mikroprosesser, skal man i de fleste tilfælde også have en skærm. Dette koster jo ekstra. Nu kan du ikke bare gå ind og lave en monitor ud af dit tv, uden først at undersøge, om chassis eller stel har forbindelse med net. Er det sidste tilfældet, vil der kunne komme spænding på stel, og dette må ikke ske! Videosignalet fra Nascom fylder mindst 7 MHz, så det er ikke muligt at føre det gennem 5000 pF. Normalt må man regne med en båndbredden ved videoforstærkeren på ca. 7 - 10 MHz, så har du et lille transistorfjernsyn, så fat diagrammet: Se om der er nettrafo ig gør så som følger: Se skematisk diagram (fig. 1). Her afbrydes automatisk, når man stikker et DIN-stik på TV'et. Næste mulighed er en optokopler. Se fig. 2: Her er dioden mellem ben 1 og 3 en lysdiode. Denne bliver moduleret udefra, og det varierende lys påvirker fotodioden mellem ben 7 - 8. Ideen med denne opstilling er at give en vældig god isolation mellem indgang og udgang. Selve modulationen foregår ved hjælp af en lille tottrins forstærker, der kobles til lysdioden, ben 2-3. Se fig.3.

Se nu ramler jeg ind i nogle vanskeligheder. For det er højst forskelligt, hvordan jeg skal koble ind i fjernsynet, så det kræver altså at den person, der vil lave det her, at han ved noget om det!

Alt i alt vil jeg tilråde den, der vil lave noget med et gammelt universal TV, at man sætter en skilletrafo på. Så kan man hurtigt finde videoforstærkeren og gå direkte ind på gitteret; evt. med en lille forstærkningsregulering i form af et potmeter på 100 - 200 Ohm. Se mit forslag på fig. 4. Der findes endnu to metoder, som kan bruges. Den ene er ret bekendt - brug den indbyggede oscillator i Nascom. Den anden går ud på, at man laver en lille oscillator, som er krystalstyret. Den skal ligge på det anvendte TV' mellemfrekvens, ca. 38 MHz. Denne osc. modulerer man så med VIDEOsignalet og indfører ved hjælp af et par vindinger, der er viklet omkring mellemfrekvenstrafoens andre vindinger, men det krævr kendskab til teknikken. Jeg har kort vist det på fig. 5. De sidste to metoder er brugbare og nr. 2 er at foretrække. Her bliver man fri for striber på skærmen. Jeg har været igennem alle metoder, men vil anbefale den direkte indkobling på sidste trin. Send et brev med tegning over dit fjernsyn, hvis der dukker et problem op.

Ole H.

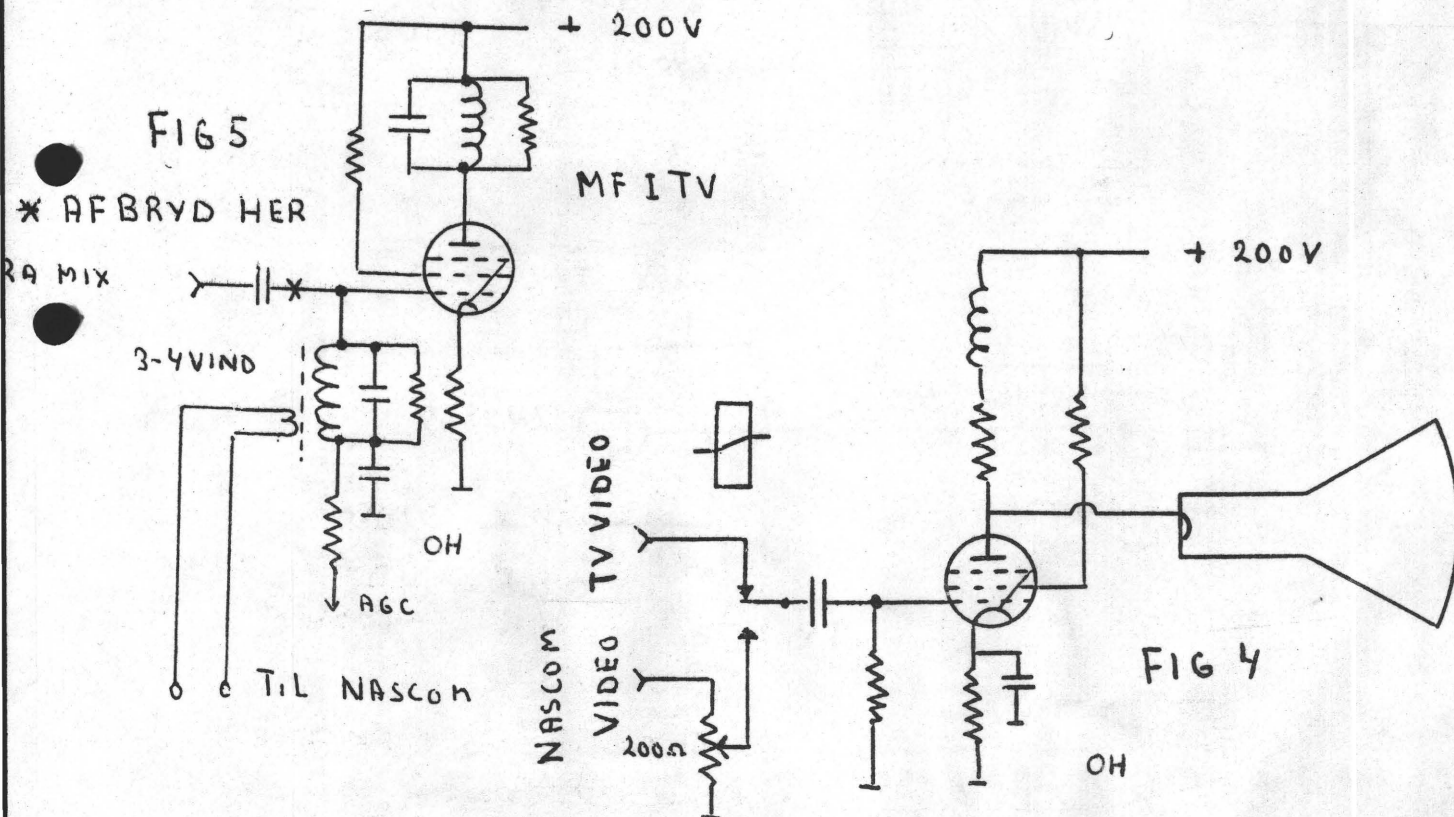


FIG 1

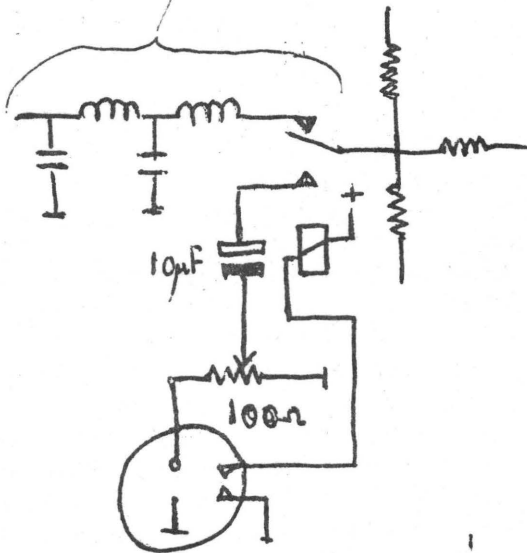
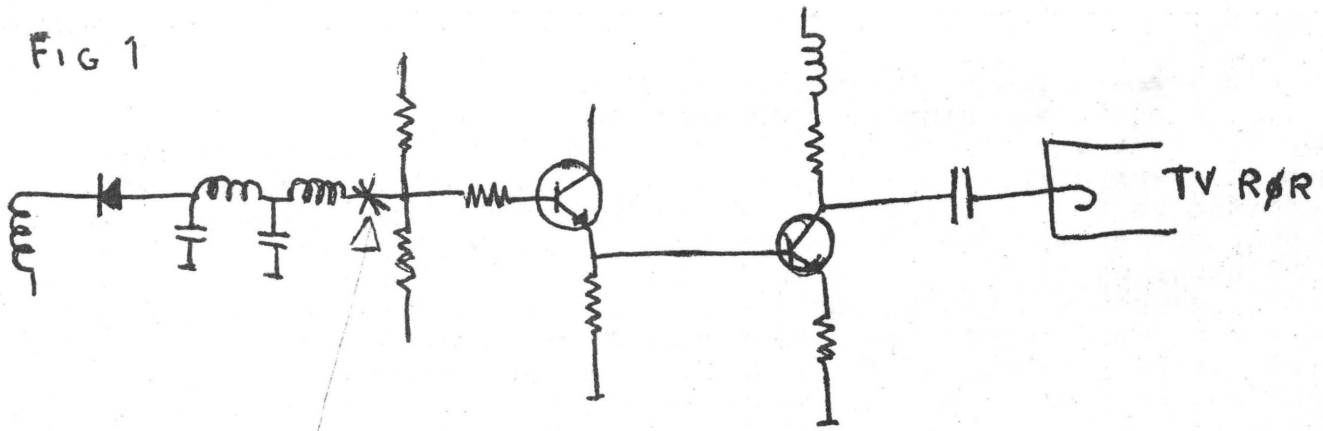


FIG 2

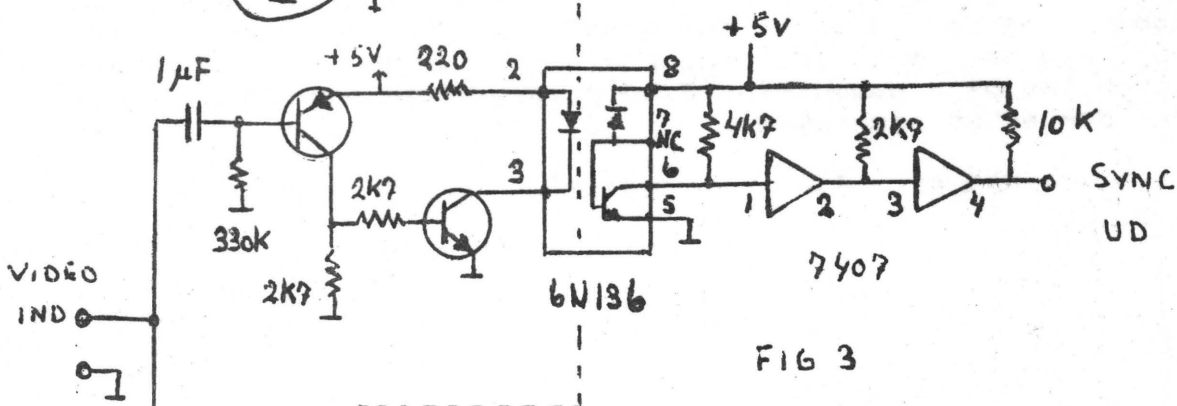
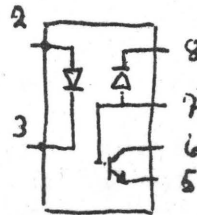
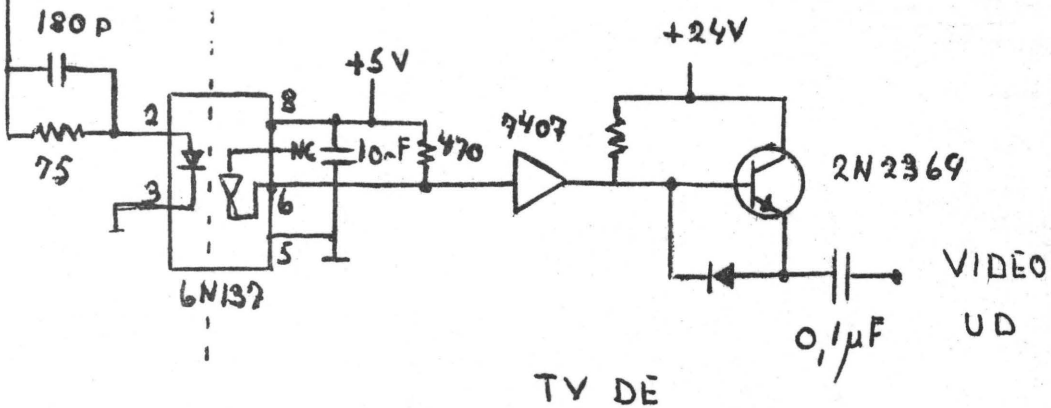


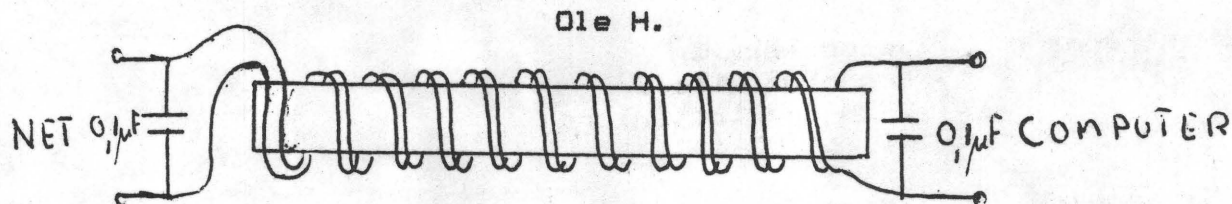
FIG 3



TV DE

Netfilter til computeren.

Adskillige medlemmer har haft problemer med støj. Pludselig forsvinder kontrollen over programmet, og man bliver nødt til at resette og varmstarte basic'en. Dette problem kan i de fleste tilfælde klares med et netfilter. Du tager en antennestav af ferrit og vikler to parallelle tråde på 0.5 mm i diameter og fylder staven op. Lidt lim på enderne og to gode kondensatorer i hver ende (størrelse 100 nF/600V). Det hele lægges i en soleret kasse og sættes i serie med netledningen.



Tråden kan være en toleder højttalerledning.

Morseprogram

Sidste nyheder.

Så lykkedes det: Søndag nat 8-3-81 hørte jeg OZ-1GIP kalde CQ på 2 meter. Klokken var over midnat, men jeg havde OZ-1AON's morseprogram inde i min Nascom 1. Jeg kører til mine forsøg over mit modem, altså FSK 1300 - 2100 Hz. Jeg forsøgte alligevel at svare, og spørme om ikke jeg fik ham ind på min skærm. Programmet er endnu ikke færdigudviklet, men alligevel en tak for godt samarbejde til Peter. Dette bare en lille oplysning fra hjemmefronten.

Ole OZ-60H

<PS> Peter har lovet, at komme med programmet, når det er færdigt, og jeg vil da komme med hardwaretegningerne.

O.

TILBUD fra Poly-Data.

Poly-Data har tilbudt klubben 20% rabat på en serie monitorer fra NEC. Rabatten træder i kraft, når fem (5) har afgivet bestilling; derefter er der også rabat ved køb af enkelte - alt ved forevisning af medlemsbevis (girokort).

Almindelige salgspriser:

9 tommer FVM-95	2200,00 kr. excl. moms
9 tommer JB-902M	2400,00 kr. - -
12 tommer JB-1201M	2800,00 kr. - -

Henvendelse hos:

Poly-Data
Mikrocenter Aps
Strandboulevarden 63
2100 København Ø
01 42 07 05

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 kassereren
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 11.3.81
Oplag: 150

Annonce:

Er du radioamatør med udstyr til transmission af HF - 14 MHz og har du en Nascom, da vil et af vore medlemmer godt være med til at starte et Nascom net, eller bare få kontakt med ligesindede computerinteresserede. Vores medlem bor på Island og hedder Frits O. Eriksson, Dvergholt 4, 270 Varma, ISLAND med kalde kode TF-3CC. Jeg ved at enhver henvendelse - også skriftlig vil være velkommen. Asbjørn