

NASCOM 780 NYT

NASCOM BRUGERGRUPPE
2730 Herlev

Sidevolden 23
Giro 6742602

NR: 6
3. årgang

JUNI/JULI 1982

Med dette sidste nummer i denne sæson vil jeg ønske Jer alle en god sommerferie, hvor man kan få sluppet af og måske få gode ideer til aktiviteter til nytte for foreningens medlemmer.

Jeg vil her gøre Jer opmærksomme på den konkurrence vi har udskrevet, nærmere beskrivelse inde i bladet. Jeg håber på mange gode og morsomme løsningsforslag. God jagt

si'r

ASBJØRN

Indhold

side 2	Referat fra generalforsamlingen
side 4	Meddelelser fra programbiblioteket
side 5	Interrupt
side 11	Pædagogiske test
side 12	Konkurrence
side 13	Sammenligning mellem 8080 og Z80
side 17	8080 <=> Z80
side 20	Simpel EPROM brænder
side 23	Annoncer

Referat af generalforsamling i Nascom-brugergruppe Den 2.5.82

DAGORDEN:

1. Valg af dirigent
2. Formandens beretning
3. Fremlæggelse af regnskab
4. Indkomne forslag
5. Fastsættelse af kontingent for 82/83
6. Valg af: Næstformand, sekretær, 2 bestyrelsesmedlemmer, 2 suppleanter, revisor samt revisorsuppleant.
7. Eventuelt

Ad 1.) Den af bestyrelsen foreslåede dirigent Henrik Dyhr blev valgt af forsamlingen.

Ad 2.) Dirigenten henviste til den skriftligt udsendte beretning, og forespurgte om kommentarer til de enkelte afsnit i beretningen. Ingen ønskede ordet.

Formandens beretning blev enstemmigt godkendt.

Ad. 3) Kassereren fremlagde det uddelte regnskab.

F. Thomsen ville gerne have en begrundelse for aktiekøbet i Handelsbanken. Kassereren begrundede investeringen med, at vi derigennem var berettiget til en aktionærkonto (på anfordring til en højere forrentning).

Regnskabet blev derefter enstemmigt godkendt.

Ad. 4) Ingen forslag var indkommet. Punktet bortfaldt.

Ad. 5) Kassereren fremlagde de 2 udsendte budgetforslag. Hvorefter Erik Hansen måtte meddele, at begge forslag var 20 kr. for lave på grund af takstforhøjelsen på tryksager med 50% og ikke med de 25%, som der var indregnet i forslaget.

C. Lautsen ønskede at få uddybet, hvordan programbiblioteket figurerede i budgetforslagene. Formanden kunne oplyse, at programbiblioteket hviler i sig selv (med et lille overskud).

J. Jacobsen ønskede en redegørelse for de 2 forslags konsekvenser. Formanden sagde, at udgangspunktet for begge forslag var en generel aflastning af bestyrelsesmedlemmerne. I forslag <1> fjernes hovedparten af det manuelle arbejde ved at indføre en lønnet forretningsfører. Hvis der ikke afsættes midler til denne hjælp, må bestyrelsesmedlemmerne aflastes ved at få trykt Nascom Nyt i byen, hvilket der er taget hensyn til i Forslag <2>.

Forslag <1> og dermed 80kr.+ 20 kr.=100 kr. i kontingent for 82/83 blev enstemmigt vedtaget.

Ad. 6) Valg i overenstemmelse med foreningens vedtægter:

a) Næstformand: Jesper Skavin blev genvalgt (uden afstemning)

b) Sekretær: Carsten Senholt blev genvalgt (uden afstemning)

c) Menige bestyrelsesmedlemmer: 1.
Frank Døngård blev genvalgt (uden afstemning)

2. Knud Ytteborg blev valgt (uden afstemning)

d) Supleanter:

1. Kaj Ole Hansen blev valgt (uden afstemning)

2. Mogens Jørgensen blev valgt (uden afstemning)

e) Revisor: Jan Jacobsen blev genvalgt (uden afstemning)

f) Revisorsuppleant: Erik Nielsen blev valgt (uden afstemning)

Ad. 7) Kassereren udbad forsamlingen om tilkendegivelser omkring en evt. udgivelse af Nascom-nyt i A5-format (ligesom sidst reviderede medlemsliste).

O. Hasselbach og P. Thomsen m.fl. var ikke positivt instillet overfor denne ide, da en sådan nedfotografering vil forringe evt. diagrammers o.l. kvalitet urimeligt.

Resultatet blev en fortsættelse med A4-format.

Formanden underrettede forsamlingen om bestyrelsens forslag på fraktionering af foreningen. Meningen var, at medlemmer i provinsen samler sig på visse knudepunkter (f.eks. Århus, Odense og Ålborg), så de vil få et udbytte svarende til medlemsmøderne i København.

J. Kjær foreslog, at man kunne give folkene i provinsen en saltvandsindspøjning ved f.eks. at arrangere et møde på Fyn.

sign sekretær

Carsten Senholt

IDEER ØNSKES:

Bestyrelsen vil gerne have ideer til mødevirksomheden i efteråret. Så hvis I kender nogen, der er i besiddelse af viden, og som gerne vil dele den med andre, kan I så ikke ringe til bestyrelsen. Vi vil også gerne have ideer til studiekredsemner og studiekredsledere.

A.

MEDDELELSER FRA PROGRAMBIBLIOTEKET.

Af Jesper Skavin.

BIDRAG FRA MEDLEMMERNE ?

Til trods for vi er små 300 medlemmer i klubben, kan medlemsbidrag til programbiblioteket siden september 1981 tælles på to (2) hænder. Hvad kan forklaringen være ?? Mangel på kreativitet og/eller interesse, dovenskab, frygt for at ens programmer ikke er "gode nok" ????? Jeg håber ikke nogen af de nævnte forhold gør sig gældende, for så ser det sort ud for programbiblioteket i fremtiden.

Derfor vil jeg hermed opfordre alle medlemmer, også dig, til at gå skuffer/div. arkiver/båndsamling og skraldespand igennem for programmer af enhver art (Basic, Pascal, assembler o.a.), og sende mig alt hvad I synes, der kan have interesse for andre. (Adresse på bladets bagside.) Send helst på bånd (hvis du ikke har nogen til overs, så ring til mig og bed om at få tilsendt nogen), men listninger er også velkomne. Hvis dine programmer skal med på den nye liste til efteråret (sep./okt.), så skal de være mig i hænde senest lørdag d. 24 juli, idet jeg er bortrejst hele august, men send dem hellere i dag end i morgen.

ÆNDRINGER TIL EKSISTERENDE PROGRAMMER.

Hvis der er nogen, som har opdaget fejl ved modtagne programmer fra programbiblioteket, hører jeg også gerne fra dem. Dette gælder også udvidelser/forbedringer af de eksisterende programmer. Send det/de ajourførte programmer med en angivelse af fejlen/tilføjelsen, men helst inden d. 24 juli.

SOMMERFERIELUKNING.

Da jeg som nævnt er bortrejst i hele august måned, er det ikke muligt at købe noget fra programbiblioteket i denne periode. Sidste dag for bestillinger er torsdag d. 29 juli. Skriftligt kan der dog godt bestilles i august, men der ekspederes først fra d. 1. september.

NYHED:

Piezodan forhandler nu et krystal, der kører med den rigtige hastighed, således at al flimmer på skærmen er borte. Samtidig har de lavet en indbygningskasse til Nascom 2 helt i aluminium, hvortil der også sælges et tastatur i samme farver og materiale.

INTERRUPT

Inden jeg fortsætter med printerrutinen, vil jeg sige noget mere generelt om interrupt (på dansk afbrydelser). Rutiner kan deles i to grupper, nemlig programstyrede og interruptstyrede.

På selve rutinen er det såmænd ikke så let at se forskel. I Z-80 afsløres det ved et I eller et N i enden på RET. Det er i grunden den eneste forskel; men andre steder i det kørende program kan man finde ordren CALL RUTINE eller JP RUTINE, der viser, at det er et program, der starter rutinen, der altså er programstyret. En interruptstyret rutine startes af en ydre begivenhed, ikke ved et CALL eller et JP, og den allerenkleste interruptrutine, som alle kender til bevidstløshed, er den der startes ved et tryk på en knap med påskriften RS - I ved knappen øverst til højre på tastaturet.

Sådan som NAS-SYS er indrettet (med første ordre LD SP,1000H) er det helt det samme der sker, om man trykker på reset-knappen eller udfører en af ordrene CALL 0, JP 0 eller RST 0, og ser man helt strengt på det er et tryk på resetknappen i virkeligheden en RST 0 ordre. Der er da også nogle datafolk, der betragter RST-ordrerne som en, ganske vist lidt særlig, form for afbrydelser, der kaldes traps (på dansk: programmerede afbrydelser). Der er lidt rod omkring ordbrugen her, for ser I efter i NAS-SYS, vil I finde, at TRAP er den rutine, der startes ved et non-maskable-interrupt. Nå nu er jeg kommet lidt for langt med interrupt. Lidt tilbage på sporet igen.

På Z-80 findes der fire ben, som Z-80 ikke selv styrer, nemlig 26 - RESET (som vi har snakket om) 25 - BUSRQ (som er meget spændende, men spørgsmålet om busrequest er en helt anden sag, som jeg måske vil skrive noget om ved en anden lejlighed) 17 - NMI og 16 - INT.

NMI står for non-maskable-interrupt, en afbrydelse som ikke kan sættes ud af kraft. Det var vist nok Zilogs mening, at den skulle bruges til noget helt specielt, som f.eks. en strømafbrydelse, hvor man så med et signal til NMI kunne få afbrudt det igangværende program og få ryddet op efter sig, mens der endnu var nogle elektroner i udglatningskondensatorerne; men NASCOM's konstruktører har haft andre tanker. Gennem et system af gates og flip-flops sender der signaler til NMI ved step-kommandoen (Ja, jeg er klar over det også sker ved execute, Anders). Så vidt jeg ved, kan man også ved et tryk på en knap på NASCOM 2 aktivere NMI til udskrift af registerindhold; men jeg har ikke gransket nærmere i det, jeg har helt enkelt ikke interesseret mig synderligt for NMI. Af hensyn til det følgende vil jeg lige notere, at et signal til NMI medfører et øjeblikkeligt hop til lageradresse 0066H, en slags RST.

INT er også en interrupt, men den kan sættes ud af kraft enten ved tryk på RESET eller ved udførelse af ordren DI. Det er altså den normale udgangssituation for z-80 at INT er virkningsløst, og først får betydning efter udførelse af ordren EI. Efter modtagelse af et signal på INT fuldføres den igangværende ordre og så udsendes en interrupt-acknowledge = kvittering for afbrydelsesanmodning udefra. På nogle mikroprocessorer findes der et særligt ben til denne kvittering. Det har man ikke kunnet finde plads til på Z-80; her bruger man et kombineret signal, så både M1 og IORQ skal gå lav. Vi har altså på benene M1, RD og IORQ følgende kombinationer:

M1	RD	IORQ	virksomhed
0	0	0	umulig
0	0	1	læs ordre fra lager
0	1	0	interruptkvittering
0	1	1	umulig
1	0	0	læs data fra port
1	0	1	læs data fra lager
1	1	0	skriv til port
1	1	1	noget andet

Man kan så spørge sig selv, om man ikke kunne have sparet WR i stedet; men det har i grunden kun teoretisk interesse, så lad det ligge.

Hvis man ikke har skrevet andet end EI efter reset, så vil Z-80 efter interruptkvittering læse næste ordre fra porten. Det kan være alt fra 1 til 4 tegn, men da det kan være lidt besværligt at få leveret mere end et tegn, er det i reglen en RST-ordre, man sender (Husker I at der var nogle, der betragtede RST-ordrer som en slags interrupts?) men andre muligheder kunne altså være CALL xx xx eller JP yy yy, men husk: kun en ordre.

Nu findes der også en anden Z-80 ordre, der hedder IM med operanden 0,1 eller 2. Skriver man IM 0, får man den ovenfor beskrevne virkning, mens IM 1 giver et direkte hop til adresse 0038H ved interrupt (helt svarende til et NMI hop til adresse 0066H). Det er muligvis den hurtigste reaktion. Den varer 13 T (T er taktimpulslængden på de 2 eller 4 MHz), mens IM 0 med en RST-ordre varer 12 T og IM 0 med en CALL varer 19T. Endelig giver IM 2 en virkning, som er den eneste, jeg vil beskæftige mig med herefter. Den fylder ordremæssigt noget mere og varer, ud over initialiseringen længst, nemlig 19T, men er til gengæld fantastisk fleksibel.

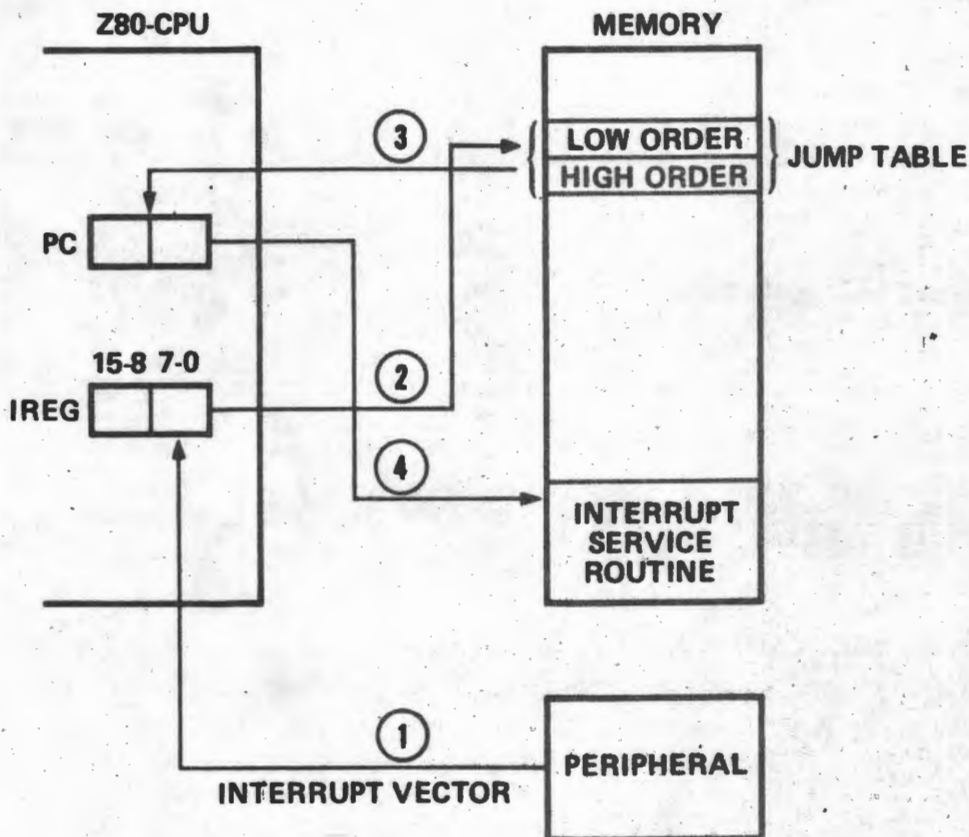
Ud over ordrene IM 2 og EI, skal man også huske at lægge en værdi ind i I-registret med ordrene LD A,n: LD I,A, og endelig skal den ydre enhed, som får lov til at afbryde have lagret en såkaldt interrupt-vektor i et særligt register i den ydre enhed selv. Hele ordrebunken bliver altså, ud over hvad man ellers har lyst til at lægge i programmet:

```

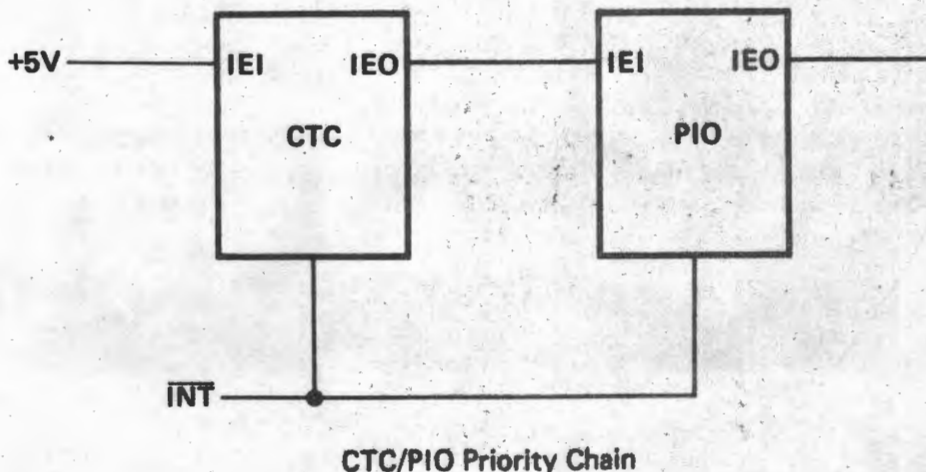
IM 2
LD A,ii
LD I,A
LD A,vektor
OUT (CTR),A
LD A,tilladelse
OUT (CTR),A
EI

```

Efter en afbrydelseskvittering læser ① Z-80 så fra den ydre enhed, der afbryder, denne vektor (lad os kalde den vv) og sætter den sammen med byten ii i I-registret til en 16 bit adresse ② iivv; på denne adresse i lagret findes ③ så endelig den adresse, hvorfra Z-80 skal ④ hente sin næste rutine. Var det indviklet? Så prøv at følge forløbet på tegningen på næste side.



Nu kan man ikke have, at flere periferienheder samtidigt kalder hvert sit interruptprogram. Hvad skal den stakkels CPU så gøre? Derfor er der en mekanisme, som giver den ene periferienhed fortrinsret frem for den anden, som igen står foran den tredje i køen. Det er den såkaldte prioritetskæde (se figur nedenfor). For at en periferienhed kan afbryde CPU'en, må to betingelser være opfyldt. For det første skal denne enhed have fået en speciel tilladelse til at afbryde; for det andet skal der stå +5V på benet mærket IEI. Sådan rundt regnet $\frac{1}{2}$ T efter interruptkvikteringen begyndelse sættes så IEO til 0V og hindrer dermed efterfølgende enheder i at sende interruptsignal og 4 T senere (det er den tid, man regner med går, inden alle enheder ned gennem kæden har registreret, at de er rykket nedad i køen) afleveres interruptvektoren. Læg mærke til, at tilladelse til afbrydelse skal være givet to steder. Dels skal CPU'en selv acceptere afbrydelse ved at have modtaget ordren EI, dels skal den enkelte periferienhed have fået speciel tilladelse til at afbryde ved en særlig styreordre, der er forskellig for de forskellige enheder.



CTC/PIO Priority Chain

Når alle disse tilladelser er givet, er der ikke en sjæl der aner, hvornår en afbrydelse kommer, og det har en række følger for programmeringen for en interruptrutine. CPU'en sørger ved en afbrydelse for, at adressen på den næste ordre, der ellers skulle udføres, gemmes på stakken, nøjagtigt som ved et CALL, og at der afbrydes for den generelle tilladelse til afbrydelser (der udføres altså ordren DI) men andet sker der ikke af sig selv. Hvad der ellers står på stakken og i registrene, ved man intet om, men det skulle gerne stå der igen, når afbrydelsesrutinen er afsluttet. Man må derfor ikke i et interruptmiljø give sig til at flytte rundt på stackpointeren på andre måder end ved PUSH og CALL og nøjagtigt samme antal POP og RET; det sidste RETI i interruptrutinen æder den adresse, interruptkaldet gemte. Vil man absolut lave om på stackpointeren, skal det ske i et monitorprogram, der ikke må kunne kaldes fra afbrydelsesrutinen. Det, at der når som helst kan blive lagt en adresse ud på stakken, betyder, at det der står i stakken under stackpointeren er værdiløst. Det kan jo når som helst blive ændret. Det er lige netop det, man har forsynet sig imod ved SCAL og RCAL i NAS-SYS 1.

Så meget om stakken, men også registerindholdet skal reddes. Derfor skal alle de registre der bruges i interruptrutinen pushes på stakken og poppes igen inden rutinen afsluttes. Hold især øje med F-registret Operationer på de andre registre kan godt ændre flagene, uden man helt tænker over det, så gem for en sikkerheds skyld altid AF, ikke så gerne med EX AF, AF', for kommer der så en ny afbrydelse, der laver samme nummer, er man jo lige vidt. Man kan selvfølgelig overveje, om man altid skulle gemme rub og stub (det er et forbrug på 24 bytes og 242 T, men særlige rutiner PUSHALT og POPALT lidt færre bytes og lidt længere tid), men det er efter min mening at skyde spurve med jord-til-luft-missiler.

Når en afbrydelsesrutine kan aktiveres, må der være givet generel tilladelse til afbrydelser. Selve kaldet sletter denne tilladelse, og den skal derfor gives tilbage af interruptrutinen før denne forlades, men tænk en gang over, om det nu også er nødvendigt at opretholde forbuddet til sidste øjeblik, og dermed også spærre for højere prioriterede rutiner. Som en tommelfingerregel plejer jeg at regne med, at jeg får vrøvl, hvis den tid, der er lukket for afbrydelser er længere end ca 20% af den tid der går mellem der kommer afbrydelser fra den travleste periferienhed, men det er et skøn, som jeg ikke har noget stensikkert grundlag for, og det må ses i sammenhæng med, at jeg har installeret to PIO'er, to CTC'er og en DMA, der alle kan interrupte. Med een PIO og ikke andet gælder reglen ganske sikkert ikke, og hvad jeg må regne med i fremtiden, når jeg om kort tid har fået installeret endnu en CTC plus en SIO er jeg spændt på.

Inden for interruptrutinen kan man tænde og slukke for de ene enheds interruptmulighed - hvis altså PIO-kanal A kom med en afbrydelse kan man slukke for kanal A's interruptmulighed - men lad hellere være med at tænde og slukke for andre periferienheder. Det fører meget let til kaos, hvis man ikke er helt sikker på, hvad det er man gør. En undtagelse er dog hvor det drejer sig om en enhed, der er direkte underlagt den enhed, der lige nu regerer.

Når man vil slukke for en ydre enheds interruptmulighed, gøres det hurtigst med:

```
LD   A,DISABLE ; for PIO har DISABLE værdien 7
OUT  (CTR),A
```

men det går også hurtigst galt, for fra begyndelsen af den sidste ordre til periferienheden effektivt er lukket, går der sådan noget som 10 - 15 T, og hvis enheden så i den tid lige når at sende et interruptrequestsignal, så vil CPU'en kvittere, men nu kan periferienheden ikke aflevere nogen vektor, og CPU'en opsamler, hvad der tilfældigvis hvad der står på datalinierne. Med pull-up modstande bliver det til FE (den sidste bit bliver altid sat til nul), men hvad står der mon her? Klog af skade har jeg på adresse iiFE (altså med mit valg af interrupttabel OCFEH) anbragt en henvisning til en fejludskrift. Også selv om jeg har fået lært, at den rette ordrekæde er denne:

```
DI   ; !!!!!
LD   A,DISABLE
OUT  (CTR),A
EI
```

så er man garderet mod den fejl.

Et lille historisk tilbageblik, inden jeg slutter den generelle del. Kan I huske gamle dage, før vi fik lavet eller fik fat i en assembler? Så sad vi med assemblerkoden i den ene hånd og manualen i den anden hånd og oversatte møjsommeligt til maskinkode. Der er vi altså nu igen med disse periferienheder. Well, hvis det kun er en PIO, er det måske til at overse, for der findes ikke så mange muligheder; heller ikke CTC'en er slem, men vent bare til I vil i gang med DMA'en, så bliver der noget at se til.

Jeg har gjort det, at jeg for hver enhed har set ret nøje på, hvilke styreord der kunne blive tale om at sende ud, og sat dem op i en tabel, som sikkert har en del lighed med SYSEQ i PZAP. Lad mig tage den (lille) del, der handler om PIO

```
UDAD      EQU    0FH
INDAD     EQU    4FH
BIDIR     EQU    8FH ; 0=ud,1=ind
CONTR     EQU    0CFH
INTEN     EQU    87H
ANDPIO    EQU    40H
HIGHPI    EQU    20H
MASKPI    EQU    10H ; 0=ja, 1=nej
RESINT    EQU    10H
IENOM     EQU    83H ; interrupt enable uden modifikation
IDNOM     EQU    3
```

Desuden har jeg lavet en lille rutine PUT inspireret af NASCOM's RST PRS. Desværre skal jeg være forberedt på også at sende et nul ud til periferienheden, så jeg kan ikke som ved RST PRS slutte listen af med et nul, men har i stedet sat et bytetæller forrest, altså:

idet der forudsættes en interrupttabel med dette udseende

ORG nnnnH

SIOINT EQU #
ITXEB DEFS 2

PIAINT DEFS 2

og styreportens adresse stående i reg. C

```
CALL PUT
DEFB ANTAL
DEFB CONTR
DEFB OFH ; f eks
DEFB INTEN+ANDPIO+MASKPI
DEFB OCCH ; f eks
DEFB PIAINT ; se fodnote
ANTAL EQU 5
```

Rutinen PUT ser sådan ud:

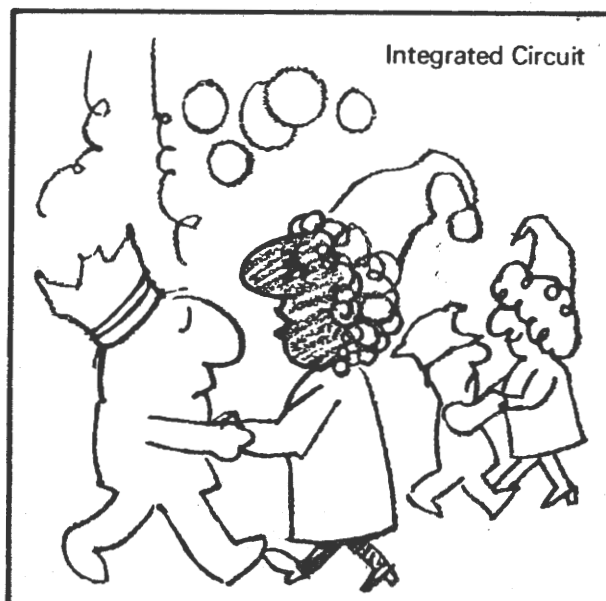
```
PUT EX (SP),HL
LD B,(HL)
INC HL
OTIR
EX (SP),HL
RET
```

Når jeg engang får mig taget sammen til at lave en ny monitor til erstatning for NAS-SYS 1, bliver denne rutine lagt ind i stedet for den vederstyggelige RCAL.

Fodnote: Brugeren man ZEAP-assembler giver denne ordre udskriften Truncation Error, men maskinkoden er god nok, så det burde være Truncation Warning. Min assembler æder ordren uden kommentarer, og det kan siges at være en skønhedsfejl. Iflg Anders skal brugere af NAP tilføje &255, og så vidt jeg kan se af anmeldelser af PZAP skal man der skrive LOW PIAINT, men det kan redaktøren måske bekræfte.

J.A.

Christian Laustsen



=====
 Pædagogisk Tests.
 =====

De Data der bliver indtast i pædagogtesten, bliver slettet igen, idet øjeblik programmet er slut eller der bliver breek in. Jeg har klaret det problem ved at liste til bånd eller hvad det nu hedder eller kaldes, da det tilsidst var kedelig, når man var færdig med pædagogtesten, JA så er alt glemt af det man har indtastet. Denne fremgangsmåde kan også benyttes i andre programmer, hvor det der bliver vist på skræmen skal gemmes.

Læs programmet ind i basic.

```

skriv      MONITOR      (NL)
           X             (NL)
           Z             (NL)
NAS.skriv.OK
           RUN      Start til optag og tryk
                   NEW LINE.
           HUSK:      Afbryd den ledning der går
                   til in i NASCOM. Ellers har
                   du et problem.
  
```

Nu vil alt der kommer på skræmen blive optaget og kan gemmes på bånd.
 Når man vil læse hvad der er optaget

```

skriv      MONITOR      (NL)      (Tilslut ledning til in igen)
           X             (NL)
           H             (NL)
  
```

Start båndoptageren for afspilling, nu vil alt det der blev optaget, blive gengivet på skræmen på nøjagtig samme måde som da man kørte programmet undtagen grafiske søjler, det der kommer på skræmen KAN IKKE gemmes i memory.

=====
 Et lille program der kan få et ord til at løbe tværs hen over skærmen, tyvstjålet fra PRACTICAL COMPUTING 1981.

```

10 CLEAR 1000:D1=5000
20 PRINT"Skriver et ord"
30 INPUTA$:A$=""+A$+"":A=1:CLS
40 X=INT(23-(LEN(A$)/2))
50 N$=MID$(A$,A,(LEN(A$)-A)):M$=MID$(A$,1,A-1)
60 Z$=N$+M$:SCREEN X,6:PRINTZ$
70 FOR C=0 TO D1:NEXT:D1=0:FOR D2 TO 100:NEXT
80 A=A+1:IFA=LEN(A$)THENA=1
90 GOTO 50
  
```

=====
 Hvis man vil skrive en oplysning der fylder så meget som disse linier her. I et program og ikke vil bruge så meget tid på det, så prøv at skriv dette på skræmen og derefter gå op til den første linie og skriv 10 PRINT".der-næst til linie to, 15 PRINT". o.s.v.....

234
 Arne Augustesen
 Postboks 55
 4220 Korsør

KONKURRENCE:

På bestyrelsesmødet den 17.6 besluttede vi, at vi ville prøve at åbne for andre Z80 ejere. Her tænker vi specielt på Gemini, Galaxi, og MIMI ejere, der alle er Nasbus-compatible og derigennem andre CP/M brugere.

Konkurrencen skulle da gå ud på 1) at finde et nyt navn til bladet, som skulle være knyttet til begge produkter (f.eks. Nasbus Nyt), eller bare et godt navn til et organ, som vores (f.eks. Combibit) - eller - 2) Beholde vores nuværende navn, men forskønne det med et godt lay out og evt. en undertitel med henvisning til interesseområder.

Derefter en afdeling der går ud på at finde et godt Logo til de enkelte dele bladet kunne deles op i (f.eks. Z80, NASSYS, PolyDos, MidiCos, CP/M, Basic/Comal, Hardware og Brevkassen). Det skulle da være sådan, at rækkefølgen i de enkelte numre var den samme og samlet under de forskellige logo'er.

Forslag til forside og logo'er sendes til Sidevolden 23 inden den 15.8. Som præmie udsættes der 300 kr. i form af gavekort til en Nascomforhandler efter eget valg. Bestyrelsen er ikke forpligtiget til at anvende og præmiere indsendte forslag og det kan lade sig gøre at dele præmien i 2, hvis 2 personer vinder.

Hc

Til redaktionen af NASCOM NYT.

at være orienteret, hvis nogen skulle spørge.

Ved samme lejlighed vil jeg gerne udtrykke tilfredshed med bladet og det store arbejde, der ligger bag hvert nummer, men tillad mig også at komme med lidt kritik. Det kniber ofte at finde rundt i artiklerne, fordi der mangler en overskrift, en henvisning eller en forklaring på programeksempler m.m. Forfatteren til en artikel forudsætter ofte, at læserne ved ligeså meget om emnet, som han selv gør. Det er jo ikke altid tilfældet, og så kniber det med forståelsen.

Kunne man ikke prøve, at lade et "almindeligt" menneske gennemlæse bladet, før det går i trykken? Jeg tror, at det med ganske få ændringer kunne gøre et godt blad endnu bedre.

med venlig hilsen



Sammenligning af maskinsprogene for 8080 og Z80:

Formålet med denne artikel, er at give en sammenligning af Z80 og 8080 maskinsprogene og specielt vise forskellene mellem de to "CPU"-er, set fra "software"-siden, det skulle derved blive nemmere at overføre maskinkode og assemblerprogrammer mellem de to typer "CPU". Det vil nok være en fordel at have kendskab til enten 8080 eller Z80 "assembly language" inden man læser denne artikel.

De symbolske maskinsprog (assembly languages) for 8080 og Z80 er meget forskellige, på trods af at alle 8080-maskinordrer eksisterer i Z80 uden større forskelle i virkning. Dette afsnit vil derfor bestå af en sammenligning af (1) den anvendte notation, (2) betegnelser for registre og operationer, og (3) virkningen af ens maskinordrer (ordrer med ens bitmøster), men hvor virkningen er forskellig, samt (4) en kort beskrivelse af de nye ordrer i Z80. I (1) og (2) er 8080-"assembly language", af nemheds grunde, forkortet til 8080, og Z80-"assembly language" forkortet til Z80. Alle de steder hvor der henvises til en 8080-ordre er dette angivet med en "*". Da der benyttes forskellige betegnelser for adresseringsmåder, er der her en oversigt over dem der benyttes af Zilog (Z80) :

Z80		eksempel
implied addressing	register adressering	LD B,H
immediate addressing	direkte adressering	LD C,93
absolute addressing	absolut adressering	LD A,(32080)
indexed addressing	indeks adressering	LD (IX-3),E
register indirect addr.	speciel indeks adressering hvor forskydning er nul	LD (BC),A
bit addressing	adresseringsmåde der muliggør tilgang til enkelte bit	BIT 3,D
relativ addressing	en art relativ adressering der benyttes ved hop-ordrer	JR Z,-69
modified zero-page addr.	benyttes kun ved 'RST p'	

(1) Anvendt notation : For begge gælder at STORE bogstaver beskriver de dele af den symbolske ordre, der er fast, og små bogstaver de dele der kan varieres, som f.eks. registre. Generelt beskrives en symbolsk ordre (for både 8080 og Z80) således : (hvor operand kan være register, data)

OPERATION	(operand)	(,operand)	
EX	(SP)	,HL	(Z80)
LD	r	,(HL)	(Z80)
LD	(HL)	,n	(Z80)
XTHL			(8080)
MOV	r	,M	(8080)
MVI	M		(8080)

I 8080 kan det ikke umiddelbart ses, på hvilken måde registre og data eller adresser tilhørende en maskinordre skal benyttes, da det fremgår af den symbolske betegnelse for maskinordren. F.eks. 'MOV r1,r2'* angiver register adressering, idet værdien af r2 flyttes over i r1. 'MVI r,data'* angiver absolut adressering ('I' i MVI står for "Immediate"). 'LDA addr'* angiver absolut adressering. Der eksisterer også en "register indirect addressing" idet der i stedet for register 'r' benyttes 'M', som angiver det ord (byte) i lageret der adresseres af registerparret HL (ved 2 operationer 'LDAX B'* , 'LDAX D'* , 'STAX B'* og 'STAX

Dette er det også muligt at benytte registerparrene 'DE' og 'BC' i stedet for 'HL'. Nogle eksempler er 'INR M', 'CMP M', og 'ANA M'.

I Z80 fremgår adresseringsmåden af notationen, idet værdien af registre, data, og adresser betegnes henholdsvis 'r', 'n', og 'nn' (register adressering, og direkte adressering). Nogle eksempler er: 'LD r,r' som svarer til 'MOV r1,r2', og 'LD r,n' som svarer til 'MVI r,data'. Absolut adressering og "register indirect addressing" angives ved at sætte parenteser om henholdsvis adressen og register-parret, som f.eks. (nn), (HL), og (BC). Et eksempel er 'LD A,(nn)' hvor 'nn' angiver det ord (byte) i lageret som flyttes over i register 'A'. En undtagelse er kun 'JP (HL)', 'JP (IX)', og 'JP (IY)', da ordretælleren, 'PC', tildeles værdien af hhv. 'HL', 'IX', og 'IY', og ikke '(HL)', '(IX)', og '(IY)'. Det må formodes at det skyldes, da det er en hop-ordre, at det er noget ulogisk at hoppe til et registerpar, men bedre at hoppe til den 'byte' i lageret der adresseres af registerparret. I Z80 er der medtaget en ny adresseringsform i forhold til 8080, idet registerparrene 'IX' og 'IY' kan benyttes til indeks adressering. Det angives ved '(IX+d)' eller '(IY+d)', hvor 'd' er en forskydning (displacement) og 'd' er et to-komplement 8-bits heltal. Et eksempel er 'LD r,(IX+119)'.

(2) I 8080 er der syv 8-bits registre, A,B,C,D,E,H og L, der betegnes 'r' (evt 'r1' og 'r2'), og tre 16-bits registre, som er de tre registerpar B, D, og H, der betegnes 'rp', og som hhv. består af registrene (B og C), (D og E), og (H og L). Der findes også en stakpil, 'SP' (Stack Pointer), og en ordretæller, 'PC' (Program Counter). Der eksisterer fem flag ("flag"/"condition bits" eller tilstandsbits), hvoraf de fire kan benyttes til betingede hop. Flagene betegnes 'Z', 'S', 'C', 'P', og har hhv. tilstandene 'NZ', 'NC', 'PO' hvis tilstanden er falsk (nul), og hhv. tilstandene 'Z', 'M', 'C', 'PE' hvis tilstanden er sand (et). Flagenes forskellige tilstande betegnes 'condition'. Det femte flag er 'AC', der dog ikke direkte kan testes, men bruges internt til 'DAA'-ordren. Flagregisteret er den mindst betydende del af 'PSW' (Processor Status Word), hvor registeret A er den mest betydende del.

I Z80 betegnes de syv 8-bits registre 'r' (evt 'r'), og navnene er ellers de samme som i 8080. De tre registerpar betegnes normalt 'dd' og er 'BC', 'DE', og 'HL'. Desuden er der tilføjet to nye registerpar som er 'IX', og 'IY'. Det er muligt at benytte 'IX' og 'IY'-registerparrene ligesom 'HL'-registerparret benyttes af 8080, bortset fra ordrer der benytter '(HL)', da 'IX' og 'IY' i disse tilfælde benyttes med forskydning ; '(IX+d)', '(IY+d)'. Stakpilen 'SP' og ordretælleren 'PC' er de samme. I flagregisteret er der medtaget et ekstra flag 'N', der ligesom 'AC', der i Z80 hedder 'H', normalt kun bruges internt i processoren til 'DAA'-ordren. Flagregisteret betegnes 'F' (for Flag), og PSW hedder nu 'AF'. Desuden betegnes flagenes forskellige tilstande ikke 'condition', men blot 'cc'. Der er yderligere tilføjet to 8-bits registre, 'I' (har ingen forbindelse med 'IX'/'IY') og 'R', og to "interrupt" flag, der benyttes ved "interrupt". Register 'I' benyttes ved visse interrupt måder og indeholder den mest betydende del af adressen for interruptproceduren. Register R ("Refresh register"), benyttes ved dynamisk RAM-lager.

En oversættelse mellem de symbolske betegnelser af operationer for Z80 og 8080, kan ses af tabellerne. Generelt viser en sammenligning, at hvor 8080 har op til flere forskellige symbol

ske betegnelser for mere eller mindre samme operation, har Z80 den samme symbolske betegnelse for de samme operationer. F.eks. hedder alle additions operationer i Z80 'ADD --' hvorimod de i 8080 har forskellige betegnelser, f.eks. 'ADD --', 'ADI --', eller 'DAD --'.

(3) Af forskelle mellem 8080 og Z80 i udførelsen af ens maskininstruktioner (= maskinordrer), dvs instruktioner med ens bitmønster, er der følgende :

- Z80 benytter paritetsflaget (p-flaget) til at indikere overløb/underløb ved visse 8-bits og 16-bits aritmetiske operationer (kun ADC HL,ss og SBC HL,ss ved 16-bits aritmetik). Flaget betegnes ofte 'P/V' (evt P/O i visse bøger over Z80). 8080 benytter derimod ved disse operationer altid paritetsflaget til paritet. Det drejer sig om følgende ordrer, der bruger 'P'-flaget til over/underløb. (der benyttes kun Z80-notation, og "s" betegner enten 'r', '(HL)', eller 'n') :

ADD A,s ADC A,s CP s DEC s INC s SUB s SBC s

- I 8080 retter 'DAA' instruktionen kun ved decimal addition og forøgelse af register 'A', hvorimod 'DAA' vil give et udefineret resultat efter subtraktion eller formindskelse af 'A'. I Z80 retter 'DAA' både ved decimal addition/forøgelse og subtraktion/formindskelse af 'A'. Det er derved muligt at udføre decimal ("BCD") aritmetik. For at mikroprocessoren så kan vide om forrige instruktion var addition eller subtraktion, benyttes flaget 'N', sådan at 'N'=1 angiver subtraktion/formindskelse, og 'N'=0 angiver addition/forøgelse af register 'A'. For 8080 er den tilsvarende bit i flagregisteret, 'F' (PSW*), altid 1.

- De to ikke brugte bit i flagregisteret er i 8080 altid nul, hvor de i Z80 er udfinerede, da de faktisk ændres af næsten hver eneste instruktion.

- Z80 sletter (nulstiller) 'H'-flaget ('AC'*) ved cyklisk skift: 'RLCA', 'RRCA', 'RRA', 'RLC', hvorimod 8080 ikke berører 'H'-flaget ved disse instruktioner.

- Ved 'ANA r'* og 'ANA M'* har det ikke været muligt at finde ud af, hvordan 'H'-flaget berøres af instruktionen, men for 'ANI data'* nulstilles flaget, så det må antages at det samme gælder for de to andre instruktioner. I Z80 sættes 'H'-flaget ved alle 'AND' operationer.

- I 8080 berøres 'H'-flag ikke af 'ADD HL,ss' ('DAD rp'*), hvorimod flaget i Z80 angiver om der var mente ud af bit 11 (bit 3 i den mest betydende del af registerparret).

- I Z80 sættes 'H'-flag ved 'CPL' ('CMA'*), hvor 8080 ikke berører flaget.

- I Z80 nulstilles 'H'-flag ved 'SCF' ('STC'*), hvor 8080 ikke berører flaget.

- I Z80 indeholder 'H'-flaget efter udførelsen af 'CCF' ('CMC'*) det foregående 'CY'-flag (C-flag). 8080 berører ikke 'H'-flaget ved denne instruktion.

(4) Af operationer der er nye i Z80 i forhold til 8080 er der følgende:

- Nye hop-ordrer. I stedet for at angive 2 byte til adressen

efter instruktionen, angives en forskydning, 'e' (på 1 byte), så det er muligt at hoppe fra -126 til +129 ord (bytes) i forhold til start af instruktion: 'JR e', 'JR NZ,e', 'JR Z,e', 'JR NC,e', 'JR C,e', 'DJNZ e'.

- En ny 16 bit addition; 'ADC HL,ss', og en ny 16 bit subtraktion; 'SBC HL,ss'. Disse benytter flagene 'S', 'Z', 'C', og 'P/V'.

- Ordrer til at kopiere en del af lageret til en anden del (blok flytning): 'LDD', 'LDI', 'LDIR', 'LDDR'.

- Ordrer til at sammenligne en del af lageret med register 'A': 'CPD', 'CPI', 'CPDR', 'CPIR'.

- 'LD dd,(nn)' og 'LD (nn),dd', hvor 'dd' nu også kan være 'DE', eller 'BC', og ikke som tidligere kun 'HL'.

- Nye ordrer til ind- og udlæsning fra porte: 'OUTD', 'OTDR', 'OUTI', 'OTIR', 'IND', 'INDR', 'INI', 'INIR', 'IN r,(C)', 'OUT (C),r'.

- Nye operationer til cyklisk skift (cyklisk rotation) og almindelig skift: 'RLC s', 'RL s', 'RRC s', 'RR s', 'SLA s', 'SRA s', 'SRL s', hvor 's' kan være: 'r', '(HL)', '(IX+d)', '(IY+d)'.

- Operationer til at teste, slette, eller sætte enkelte bit i et register: 'BIT b,s', 'RES b,s', 'SET b,s', hvor 's' er det samme som før, og 'b' angiver nummeret på den bit der skal opereres med (0-7).

- 'NEG', der tager to-komplementet til register 'A' og placerer resultatet i 'A'.

- Nogle nye operationer til "interrupt", så flere "interrupt" måder kan tillades, b.l.a. "interrupt" med vektor adressering: 'IM m', 'RETI', 'RETN'.

- Registrene 'A', 'B', 'C', 'D', 'E', 'F', 'H', og 'L' er dubleret sådan at det ved to operationer er muligt at ombytte de to sæt af registre: 'EXX', og 'EX AF,AF'.

Til denne artikel er brugt følgende litteratur:

MOSTEK	MOSTEK
Z80 Microcomputer Software	MK 3880 CPU
Programming Guide	Technical Manual
Z80 Programmingmanual	

INTEL-8080 Microcomputer Systems
Userers Manual

Rodnay Zaks "Programming the Z80"

Lance A. Leventhal "Z80 Assembly Language Programming"

Denne artikel er en modificeret udgave af et afsnit i rapport DAT1 2/81 af:

Kim Ritter Wagner
Peter Herzberg
Frank Damgaard

Øversættelse fra 8080 til Z80 og omvendt, af de symbolske maskinordre.

8080 til Z80 :

8080	Z80	8080	Z80
ACI data8	ADC A,n	MOV M,r	LD (HL),r
ADC M	ADC A,(HL)	MOV r,M	LD r,(HL)
ADC r	ADC A,r	MOV r1,r2	LD r,r
ADD M	ADD A,(HL)	MVI M,data8	LD (HL),n
ADD r	ADD A,r	MVI r,data8	LD r,n
ADI data8	ADD A,n	NOP	NOP
ANA M	AND (HL)	ORA M	OR (HL)
ANA r	AND r	ORA r	OR r
ANI data8	AND n	ORI data8	OR n
CALL addr	CALL nn	OUT data8	OUT (n),A
CC addr	CALL C,nn	PCHL	JP (HL)
CM addr	CALL M,nn	POP pr	POP qq
CMA	CPL	PUSH pr	PUSH qq
CMC	CCF	RAL	RLA
CMP M	CP (HL)	RAR	RRA
CNC addr	CALL NC,nn	RC	RET C
CNZ addr	CALL NZ,nn	RET	RET
CF addr	CALL P,nn	RLC	RLCA
CPE addr	CALL PE,nn	RM	RET M
CPI data8	CP n	RNC	RET NC
CPO addr	CALL PO,nn	RNZ	RET NZ
CZ addr	CALL Z,nn	RP	RET P
DAA	DAA	RPE	RET PE
DAD rp	ADD HL,ss	RPO	RET PO
DCR M	DEC (HL)	RRC	RRCA
DCR r	DEC r	RST n	RST p
DCX rp	DEC ss	RZ	RET Z
DI	DI	SBB M	SBC A,(HL)
EI	EI	SBB r	SBC A,r
HLT	HALT	SBI data8	SBC A,n
IN data8	IN A,(n)	SHLD addr	LD (nn),HL
INR M	INC (HL)	SPHL	LD SP,HL
INX rp	INC ss	STA addr	LD (nn),A
JC addr	JP C,nn	STAX B	LD (BC),A
JM addr	JP M,nn	STAX D	LD (DE),A
JMP addr	JP nn	STC	SCF
JNC addr	JP NC,nn	SUB M	SUB (HL)
JNZ addr	JP NZ,nn	SUB r	SUB r
JPE addr	JP PE,nn	SUI data8	SUB n
JPO addr	JP PO,nn	XCHG	EX DE,HL
JZ addr	JP Z,nn	XRA M	XOR (HL)
LDA addr	LD A,(nn)	XRA r	XOR r
LDAX B	LD A,(BC)	XRI data8	XOR n
LDAX D	LD A,(DE)	XTHL	EX (SP),HL
LHLD addr	LD HL,(nn)		
LXI rp,data16	LD ss,nn		

Z80 TIL 8080 :

Z80	8080		
ADC A,(HL)	ADC M	LD A,I	-
ADC A,(xy+d)	-	LD A,R	-
ADC A,n	ADC data8	LD BC,(nn)	-
ADC A,r	ADC r	LD DE,(nn)	-
ADC HL,ss	-	LD HL,(nn)	LHLD addr
ADD A,(HL)	ADD M	LD I,A	-

ADD A, (xy+d)	-	LD R,A	-
ADD A,n	ADD data8	LD SP,(nn)	-
ADD A,r	ADD r	LD SP,HL	SPHL
ADD HL,ss	DAD rp	LD SP,xy	-
ADD IX,pp	-	LD r,(HL)	MOV r,M
ADD IY,rr	-	LD r,(xy+d)	-
AND (HL)	ANA M	LD r,n	MVI r,data8
AND (xy+d)	-	LD r,r'	MOV r1,r2
AND n	AND data8	LD ss,nn	MVI rp,data16
AND r	ANA r	LD xy,(nn)	LXG rp,data16
BIT b,(HL)	-	LD xy,nn	-
BIT b,(xy+d)	-	LDD	-
BIT b,r	-	LDDR	-
CALL nn	CALL addr	LDI	-
CALL C,nn	CC addr	LDIR	-
CALL M,nn	CM addr	NEG	-
CALL NC,nn	CNC addr	NOP	NOP
CALL NZ,nn	CNZ addr	OR (HL)	ORA M
CALL P,nn	CP addr	OR (xy+d)	-
CALL PE,nn	CPE addr	OR n	ORI data8
CALL PO,nn	CPO addr	OR r	ORA r
CALL Z,nn	CZ addr	OTDR	-
CCF	CMC	OTIR	-
CP (HL)	CMP M	OUT (C),r	-
CP (xy+d)	-	OUT (n),A	OUT data8
CP n	CPI data8	OUTD	-
CP r	CMP r	OUTI	-
CPD	-	POP qq	POP rp
CPDR	-	POP xy	-
CPI	-	PUSH qq	PUSH rp
CPIR	-	PUSH xy	-
CPL	CMA	RES b,(HL)	-
DAA	DAA	RES b,(xy+d)	-
DEC (HL)	DCR M	RES b,r	-
DEC (xy+d)	-	RET	RET
DEC r	DCR r	RET C	RC
DEC ss	DCX rp	RET M	RM
DEC xy	-	RET NC	RNC
DI	DI	RET NZ	RNZ
DJNZ e	-	RET P	RP
EI	EI	RET PE	RPE
EX AF,AF'	-	RET PO	RPO
EX DE,HL	XCHG	RET Z	RZ
EX (SP),HL	XTHL	RETI	-
EX (SP),xy	-	RETN	-
EXX	-	RL (HL)	-
HALT	HLT	RL (xy+d)	-
IM m	-	RL r	-
IN A,(n)	IN data8	RLA	RAL
IN r,(C)	-	RLC (HL)	-
INC (HL)	INR M	RLC (xy+d)	-
INC (xy+d)	-	RLC r	-
INC r	INR r	RLCA	RLC
INC ss	INX rp	RLD	-
INC xy	-	RR (HL)	-
IND	-	RR (xy+d)	-
INDR	-	RR r	-
INI	-	RRA	RAR
INIR	-	RRC (HL)	-
JP (HL)	PCHL	RRC (xy+d)	-
JP (xy)	-	RRC r	-
JP C,nn	JC addr	RRCM	RRC
JP M,nn	JM addr	RRD	-
JP NC,nn	JNC addr	RST p	RST n

JP	NZ,nn	JNZ	addr	SBC	A, (HL)	SBB	M
JP	P,nn	JP	addr	SBC	A, (xy+d)	-	
JP	PE,nn	JPE	addr	SBC	A,n	SBI	data8
JP	PO,nn	JPO	addr	SBC	A,r	SBB	r
JP	Z,nn	JZ	addr	SBC	HL,ss	-	
JP	nn	JMP	addr	SCF		STC	
JR	C,e	-		SET	b, (HL)	-	
JR	NC,e	-		SET	b, (xy+d)	-	
JR	NZ,e	-		SET	b,r	-	
JR	Z,e	-		SLA	(HL)	-	
JR	e	-		SLA	(xy+d)	-	
LD	(BC),A	STAX	B	SLA	r	-	
LD	(DE),A	STAX	D	SRA	(HL)	-	
LD	(HL),n	MVI	M,data8	SRA	(xy+d)	-	
LD	(HL),r	MOV	M,r	SRA	r	-	
LD	(nn),A	STA	addr	SRL	(HL)	-	
LD	(nn),BC	-		SRL	(xy+d)	-	
LD	(nn),DE	-		SRL	r	-	
LD	(nn),HL	SHLD	addr	SUB	(HL)	SUB	M
LD	(nn),SP	-		SUB	(xy+d)	-	
LD	(nn),xy	-		SUB	n	SUI	data8
LD	(xy+d),n	-		SUB	r	SUB	r
LD	(xy+d),r	-		XOR	(HL)	XRA	M
LD	A, (BC)	LDAX	B	XOR	(xy+d)	-	
LD	A, (DE)	LDAX	D	XOR	n	XRI	data8
LD	A, (nn)	LDA	addr	XOR	r	XRA	r

"-" angiver at der ikke er en tilsvarende instruktion.

Register-betegnelser:

Z80	8080	kommentar
n	data8	8-bit heltalsværdi
nn	data16/addr	16-bit heltalsværdi
r	r	register A,B,C,D,E,H,L
ss	rp	registerpar BC,DE,HL,SP
qq	rp	registerpar BC,DE,HL,AF ved PUSH/POP
xy	-	registerpar IX,IY
pp	-	registerpar BC,DE,IX,SP
rr	-	registerpar BC,DE,IY,SP
p	n	ved 'RST'
e	-	forskydning ved hop 'JR ...'
d	-	forskydning ved (IX+d), og (IY+d)

Sammenligning af Z80 og 8080 benævnelserne for registre og registerpar :

registre:				registerpar:			
Z80	8080	Z80	8080	Z80	8080	Z80	8080
A	A	E	E	IY	-	BC	B
A'	-	E'	-	L	L	DE	D
B	B	F	PSW*	L'	-	HL	H
B'	-	F'	-	R	-	AF	PSW
C	C	H	H	PC	PC		
C'	-	H'	-	SP	SP		
D	D	I	-				
D'	-	IX	-				

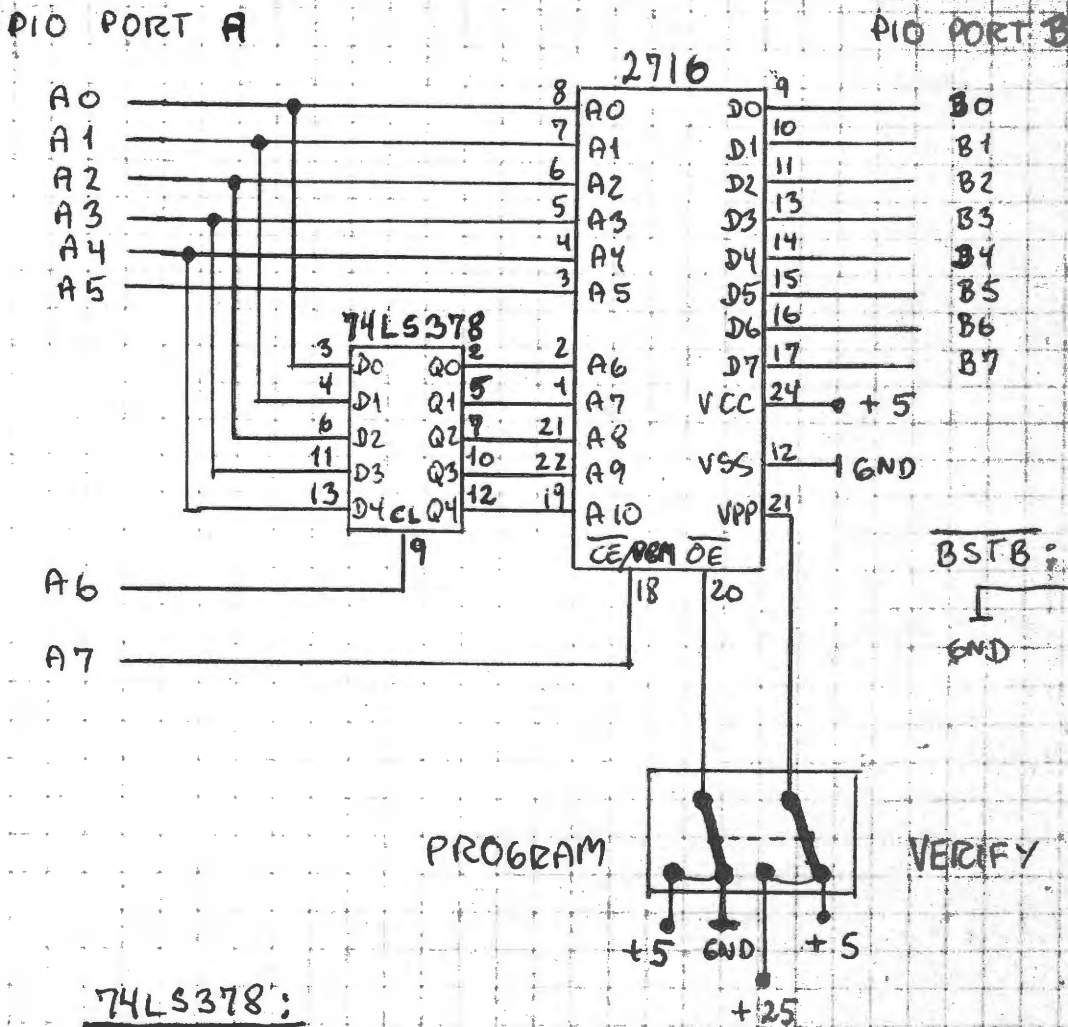
* mindst betydende del af PSW

Simpel EPROM-brænder.

Under arbejdet med at omdanne en NASCOM-2 med Gemini video-kort til en Hewlett-Packard 3000 kompatibel terminal skulle jeg en sen natte time bruge en 2716 - EPROM brænder. Resultatet ses på vedlagte diagram/programliste. Kun en kreds ekstra bruges - en 74LS378 (6 bit latch). De 25 volt fås fra 3 stk. 9 volts batterier, der serieforbindes (+ evt. 3 dioder til et spændingsfald på 2,1 volt - 27 volt virker dog godt nok). Eprommerne skal være af single-supply typen. Opstillingen dner også til 2732 - så tages blot endnu en data bit via latch til A11.

Med venlig hilsen

Lars Rugård Jensen



74LS378:

PIN 16 = +5
 PIN 1,8 = GND (1 = ENABLE)

PORT A = OUTPUT
 PORT B = INPUT (VERIFY)
 OUTPUT (PROGRAM)


```

; ** RUTINE TIL AT BRÆNDE 2716-EPROM (SINGLE SUPPLY)
; ** DATA HENTES FRA 7000H:
; **
BRINI: LD HL,7000H ;HERFRA HENTES DATA
      LD BC,800H ;ANTAL BYTES
      LD A,15 ;SÆT PORT B=OUTPUT
      OUT (7),A
      OUT (6),A R;SÆT PORT A=OUTPUT

BREND: LD A,(HL) ;HENT DATA
      OUT (5),A ;UD TIL EPROM D0-D7
      CALL BRADR ;SEND ADR. OG START BRÆND
      LD D,16 ;PAUSE V. 4 MHZ+WAIT (= 9 VED 2 MHZ)
      XOR A
PAUSE: RST 38H ;PAUSE
      DEC D
      JR NZ,PAUSE

      OUT (4),A ;STOP BRÆNDING
      CPI ;INC HL,DEC BC
      JP PE,BREND ;FLERE SÅ FORFRA
      RST 28H
      DB 'BRÆNDING SLUT',13,0
      SCAL 5BH

SETADR: LD A,H ;HENT ADR/HIGH
      AND 7 ;BRUG KUN 3 LAVESTE BITS
      ADD A,A ;FLYT DEM 2 PLADSER
      ADD A,A
      LD D,A ;GEM DET
      LD A,L ;HENT ADR/LOW
      AND 11000000B ;ISOLER 2 HØJESTE
      ADD A,A ;FLYT TIL PLADS 1/0
      ADC A,A
      ADC A,A
      OR D ;'OR' MED ØVERSTE BITS
      OUT (4),A ;SÆT ADR. 6-10 PÅ EPR.
      OR 40H ;SÆT LATCH-BIT
      OUT (4),A
      LD A,L ;HENT ADR/LOW
      AND 3FH ;BRUG DE LAVESTE 6 BITS
      OR 40H ;INGEN CLOCK
      OUT (4),A ;SÆT ADR. 0-5 PÅ EPROM
      RET ;FÆRDIG

BRADR: CALL SETADR
      OR 80H ;SÆT BRÆND-BIT
      OUT (4),A ;START BRÆNDING
      RET

; **
; ** RUTINE TIL AT INDLÆSE DATA FRA 2716-EPROM.
; ** DATA LÆGEES UD FRA ADR. 6000H OG KAN SÅ
; ** KONTROLLERES VIA TABULERING (ELLER TESTPROGR.):
; **
START: LD A,15 ;SÆT PORT A=OUTPUT
      OUT (6),A

```

```

LD HL,6000H ;HERTIL LÆSES DATA IND
LD BC,800H ;ANTAL BYTES (2K)
INDLES: CALL SETADR ;SÆT ADRESSE OP
IN A,(5) ;HENT DATA FRA EPROM
LD (HL),A ;GEM DATA
CPI ;INC HL,DEC BC
JP PE,INDLES ;FLERE SÅ FROFRA

RST 2BH
DB 'INDLÆSNING SLUT',13,0
SCAL 5BH

SETADR: LD A,H ;HENT ADR/HIGH
AND 7 ;BRUG KUN 3 LAVESTE BITS
ADD A,A ;FLYT DEM 2 PLADSER
ADD A,A
LD D,A ;GEM DET
LD A,L ;HENT ADR/LOW
AND 11000000B ;ISOLER 2 HØJESTE
ADD A,A ;FLYT TIL PLADS 1/0
ADC A,A
ADC A,A
OR D ;'OR' MED ØVERSTE BITS
OUT (4),A ;SÆT ADR. 6-10 PÅ EPR.
OR 40H ;SÆT LATCH-BIT
OUT (4),A
LD A,L ;HENT ADR/LOW
AND 3FH ;BRUG DE LAVESTE 6 BITS
OR 40H ;INGEN CLOCK
OUT (4),A ;SÆT ADR. 0-5 PÅ EPROM
RET ;FÆRDIG

```

Tilbud: 1 Luma farve-TV. (R.G.B.) kan afhentes hos Kassereren.

ANNONCE:

Samleindkøb af floppydiscs:

Med den hårde konkurrence i grossistledet er prisen kommet ned på et rimeligt niveau. Ved et samlet indkøb kan vi yderligere få rimelige kvantumsrabatter.

Der tænkes især på 5 1/4 " drives fra Olivetti (ex. FD 502), ved 10 stk. er prisen 2100 kr. FD 502 er D.S.D.D., det vil sige at et drive vil kunne indeholde 368,6 Kb formateret. En 'mix-samle ordre' er også muligt.

Henvendelse til Søren Sørensen 02 48 31 01

(Hertil skal bemærkes at Søren handler helt privat og ikke på foreningens vegne. A.)

ANNONCE:

Et styk 9 tommer NEC S/H monitor sælges for 1400 kr. (ganske ubrugt normalpris: 2684 kr.)

Morten Støvring

Bakkehøj 47

2740 Skovlunde

Tlf.: Arb. 01 12 32 00 (9-16)

Hj. 02 84 12 70 (17.30-18.30)

ANNONCE:

1 kabinet til Nascom 2	300 kr.
1 keyboardkasse (blå)	50 kr.
I/O Grafik board	375 kr.
2 Strømforsyninger (-trafo)	400 kr.

(Kabinet er en modulbox med blæser, afbryder, netfilter og lysdioder)

Lars Petersen 01 20 80 08 efter 17.

ANNONCE:

Disketter til 8 tommer sælges for 25 kr. pr. stk meget lidt brugt.

Per Thomsen 01 53 50 30

Annonce

Færdigsamlet Nascom 2 med TV, båndoptager, 48 K RAM, Nassys 3 og grafik sælges. Masser af dokumentation og anden litteratur. Af software nævnes i flæng: 8 K ROM Microsoft Basic, 12 K TAPE BLS Pascal, NAP assembler, NIP disassembler, Logichess, Månelander, Monster, Basketball o.m.m. Desuden gammel fjernskriver med software og diagram til interface. 2 TV-spil håndtag med software. Henv.: Steen Iversen, Teglgårdsvej 42, 2920 Charlottenlund, tlf.: (01) 644125.

ALMINDELIGE OPLYSNINGER OM FORENINGEN :

Bestyrelsens sammensætning:

Formand Asbjørn Lind
Sidevolden 23
2730 Herlev
02 91 71 82
(Helst mellem kl. 20.00 og 21.00)

Næstformand Jesper Skavin
Broholms Alle 3
2920 Charlottenlund
01 64 03 14

Kasserer Erik Hansen
Lyngby Kirkestræde 6.1
2800 Lyngby
Helst skriftlig henv. Ellers
torsdag mellem kl. 17 og 18.
02 88 60 55

Sekretær Carsten Senholt
Blommevangen 6
2760 Måløv
02 66 19 65

Teknisk red. Ole Hasselbalch
Vibeskrønten 9
2750 Ballerup
02 97 70 13
(Helst mellem 17 og 19)

Frank Damgård
Kastebjergvej 26A
2750 Ballerup
02 97 10 20

Knud Ytteborg
Dyssegårdsvej 71B
2860 Søborg
01 87 75 23

67

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 1.7.82 - 1.7.83: 100,00 kr.

Oplag: 350

Redaktionen sluttet den 18.06.82

Husk at gamle numre kan købes hos Ole for 10 kr./stk +porto.

Printerservice hos formanden.

Bånd og bokse kan købes hos Carsten til følgende priser:

10 bånd 45 kr., ekstra etiketter 0,25 kr./stk og bokse
1,50 kr./stk + porto.

Annoncepris 0.75 kr. pr. A4 side. Indlevering foreningens adresse.