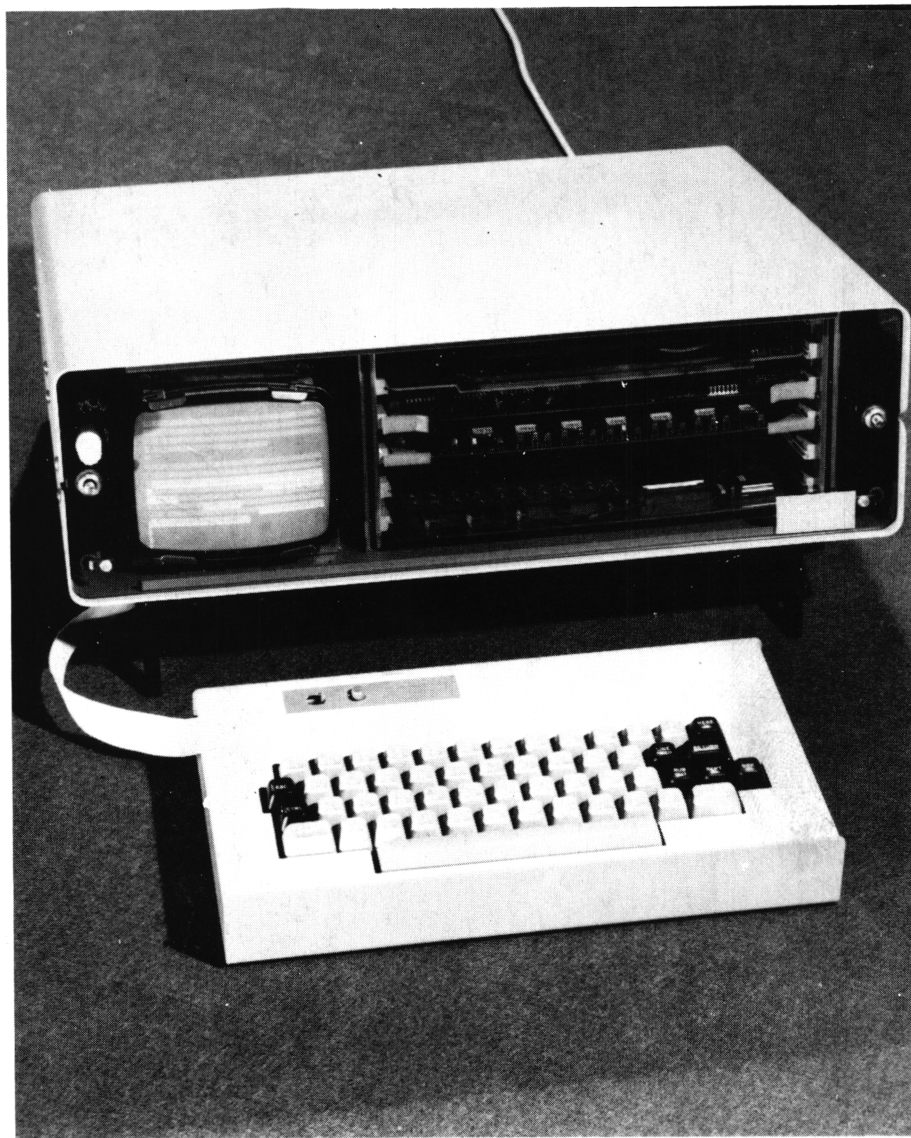


4 Håndbog for datamat-amatorer

1978



INDHOLDSFORTEGNELSE

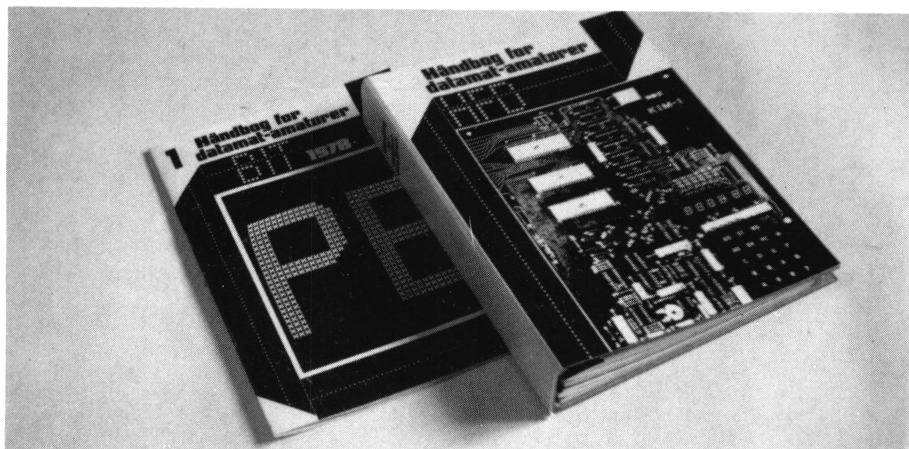
ALMENT OM PROGRAMMERBARE		Datamatamatører	K 11
MASKINER			
Sådan begyndte det	A 1	LOMMEREGNERE	
Den forventede udvikling	A 7	TI-Programmer	L 1
Talsystemer	A 21	HP-25/25C	L 3
Talsystemer, fortsat	A 27	TI-59/PC-100	L 5
Binær matematik	A 28		
		MIKRODATAMATER	
BIBLIOTEKET - PROGRAMMER		Valg af microdatamat	M 1
HP-25, Delefilter	B 1	Datamatkapacitet	M 5
HP-25, Gæt et tal	B 3	KIM-1	M 11
HP-25, Likviditet	B 5	KIM-1, kontakter og dioder	M 15
HP-25, Mastermind	B 11	Motorola M6800	M 19
KIM-1, Multi-maze	B 13	TK-80, begynder sæt	M 25
IMSAI 8080, RAM-test	B 17	Imsai 8048 CC	M 30
KIM-1, FUT-FUT	B 19		
TI-59, Løn og skat	B 21	PROGRAMMERINGSTEKNIK	
		Lær programmering	P 1
CPU-ARKITEKTUR		Splitning af subrutiner	P 51
CPU-arkitektur	C 1	Subrutiner	P 49
Motorola M6800	C 5		
Intel 8080	C 7	SELVBYGGERPROJEKTER	
SC/MP	C 9	IMSAI 8080	S 1
Signetics 2650	C 11	Z-80 microdatamat	S 11
Intel 8048	C 13	77-68 selvbyggerdatamat	S 51
		Z-80, fortsat	S 33
DATAMAT-LITTERATUR		YDRE ENHEDER	
Elementært om Microdatorer	D 1	TV-skriver	Y 11
The first Book of KIM	D 2	Pocket TTY	Y 13
		TV-modulator	Y 16
KLUBINFORMATION		Digital multiplekser	Y 19
Datamatklubber	K 1		

Bemærk: Vigtig meddelelse overfor side P 49

FORSIDEN. Motorola er nu ved at være klar med deres TDS-system, som er en komplet og færdigsamlet micro-datamat med monitor og BASIC i ROM. Der medfølger TV-skriver, og der er direkte tilslutning for kassettebåndoptager. Vi skal senere bringe en mere indgående beskrivelse.

Håndbog for datamat-amatører udgives i løbsbladsformat af Telepress ApS, Greve Strandvej 42, 2670 Greve Strand. Tlf. (02) 90 86 00. Giro nr. 1 15 53 69. Tryk: Fraling Offset, Viby Sj. HFD udsendes til abonnenter som tryksag d. 1. torsdag hver måned. 1. nummer udgivet er nr. 9/1977. Et årsabonnement koster kr. 100,- incl. ringbind og porto. Ansvarshavende udgiver: H. Lind. Redaktør: Peter Holm

Alment om programmerbare maskiner	A
Biblioteket - programmer	B
CPU-arkitektur	C
Datamat-litteratur	D
Interfacing	I
Klubinformation	K
Lommeregnerne	L
Microdatamater	M
Programmeringsteknik	P
Selvbyggerprojekter	S
Tilbud fra læserne	T
Undervisningsudstyr	U
Ydre enheder	Y



SAMLEMAPPE TIL HFD

Det er forbløffende, at så relativ simpel en ting som at fremstille en samlemappe, kan afstedkomme så mange problemer, som de, vi efterhånden har fået refereret.

Ringbindene, som i henhold til det tilbud, som vi har modtaget, skulle være færdige inden påske, ligger idag (d. 1/4) i Jylland med silketryk på halvdelen af oplaget. Det skønnes, at det vil tage 5 dage at få alle mapper færdigtrykt og fremsendt til København, hvor ringene skal isættes. Sidstnævnte operation anlås nu til at ville tage op til 2 uger, hvilket vi står totalt uforstående overfor.

Det korte af det lange er naturligvis, at det ikke skyldes postvæsenet, at ringbindene ikke medsendes dette eksemplar af HFD, og vi kan i dette øjeblik ikke love andet, end at vi vil gøre alt, hvad vi kan for at fremskynde sagen og få de bestilte mapper ud af døren, så snart vi har dem i hus.

Når man betænker, at B&W idag kan bygge et containerskib (hvis de kan få orden) på få måneder, fatter man ikke, at en håndfuld ringbind kan tage så lang tid - de må sandelig blive flotte!

HFD TIL SOMMER

En årgang HFD består af 11 numre, idet bladet ikke udsendes i juli måned.

De nærmeste numre udsendes derfor fra

os på følgende datoer:

d. 5/5

d. 1/6

d. 3/8

HFD udsendes som bekendt den 1. torsdag i hver måned, men i maj må vi flytte til fredagen efter på grund af Kristi Himmelfartsdag.

Vi ændrer meget nødigt adresse på grund af ferieophold - lad venligst postvæsenet eftersende, hvis dette ønskes.

Ved permanent adresseændring skal vi derimod have besked om fulde navn og både den gamle og nye adresse.

MANUSKRIPTER

Vi modtager meget gerne manuskripter fra læserne til offentliggørelse i HFD.

Vi er interesserede i både programmer, beskrivelse af anlæg o.s.v.

Vi kan normalt ikke tilbyde betaling for sådanne artikler, men du er selvfølgelig velkommen til at stille krav om et honorar for offentliggørelse - hvorefter vi kan tage stilling til manuskriptet.

Undgå venligst håndskrevne manuskripter, og benyt stor linieafstand. Vedlæg meget gerne illustrationer i sort streg eller s/h foto.

I er meget velkomne til at kontakte redaktionen vedrørende eksisterende eller planlagte projekter.

DET OKTALE TALSYSTEM

Ganske som det hexadecimaltalsystem er fremkommet som følge af et praktisk behov, benyttes til tider et talsystem med 8 som base - det oktale talsystem.

Det er i princippet opbygget helt som de foranstående systemer, blot kan man nøjes med at benytte de kendte cifre fra 10-talsystemet - fra 0 til 7 incl.

I det oktale talsystem vil de forskellige positioner fra højre repræsentere hhv. enere, ottere, fireogtres'ere o.s.v.

Det betyder, at der også er direkte sammenhæng mellem det binære og det oktale talsystem, på samme måde som vi så det ved det hexadecimaltalsystem.

Hvis vi ønskede at omregne mellem binære og hexadecimaltalsystem, skulle vi benytte 4 binære cifre for hvert hexadecimalt ciffer. Når vi omregner mellem oktale og binære cifre, skal der anvendes 3 binære cifre for hvert oktal-ciffer.

Lad os se på forholdet mellem de første tal fra disse 3 talsystemer:

Denne stribe tal skulle vise med al ønskelig tydelighed, hvorledes de 3 talsystemer hænger sammen.

I alle rækkerne har vi her talt til 32 i det decimaltalsystem.

Da vi så på det hexadecimaltalsystem, bemærkede vi, at der for hver 4 binære tal kunne dannes ét hexadecimalt ciffer. Helt tilsvarende vil et oktalt ciffer kunne erstatte hver gruppe på 3 cifre i det binære talsystem.

Se f.eks. på den nederste række, hvor det binære tal er

100 000

og det oktale er

40

Taget ciffer for ciffer i det oktale tal vil 4 være det samme som 100, mens 0 er helt identisk med 000. Og således vil det være hele vejen igennem.

Der er dog ikke mange maskiner, som benytter sig af det oktale talsystem, men det er godt at vide, hvorledes det er

Binært	Oktal	Hex
000 001	1	1
000 010	2	2
000 011	3	3
000 100	4	4
000 101	5	5
000 110	6	6
000 111	7	7
001 000	10	8
001 001	11	9
001 010	12	A
001 011	13	B
001 100	14	C
001 101	15	D
001 110	16	E
001 111	17	F
010 000	20	10
010 001	21	11
010 010	22	12
010 011	23	13
010 100	24	14
010 101	25	15
010 110	26	16
010 111	27	17
011 000	30	18
011 001	31	19
011 010	32	1A
011 011	33	1B
011 100	34	1C
011 101	35	1D
011 110	36	1E
011 111	37	1F
100 000	40	20

opbygget, så man ikke står helt på bar bund, hvis man en dag falder over det.

Der er selvfølgelig også en sammenhæng mellem det oktale og hexadecimaltalsystem, men en direkte omregning er ikke særlig simpel. Det vil være lettere at nedskrive det tilsvarende binære tal og derefter ændre gruppdelingen og dernæst tage det nye tal.

Hvis man f.eks. ønsker at omskrive et hexadecimalt tal FA3 til oktal, gøres det lettest således.

Først laver vi FA3 om til 3 grupper binære tal på hver 4 cifre:

1111 1010 0011

Dernæst deler vi grupperne om til grupper på hver 3 binære cifre:

111 110 100 011

Og dette omsat til oktalt er lig med:

7643

Hvad er det tilsvarende decimaltal?

Binær matematik

Datamaten kan - som omtalt flere gange - kun arbejde med binære tal, og når vi benytter os af mere tilnærmelige talsystemer og sprog, er dette kun muligt, fordi vi kan få datamaten til selv at oversætte disse informationer til en binær kode, som datamaten kan arbejde med.

En del af datamatens arbejde består i at udføre beregninger, hvori forekommer matematiske funktioner som addition, multiplikation, division o.s.v.

Som følge af datamatenens interne opbygning udføres alle disse matematiske funktioner i binær form, og det er en stor hjælp til både at forstå, hvorledes datamaten fungerer internt, og hvordan de mest effektive programmer skabes, at have kendskab til de mest elementære binære matematiske funktioner.

Lad os først se på den simpleste af alle funktioner:

BINÆR ADDITION

Hvis vi ønsker at addere to binære cifre, er der kun ganske få tænkbare muligheder:

0	+	0	=	0
0	+	1	=	1
1	+	0	=	1
1	+	1	=	10

Da det sidste resultat rummer ét ciffer mere, end de to tal, som blev adderet, må dette være fremkommet som en mente. (På engelsk: Carry = bære).

Brug af mente i binær matematik er identisk med brugen i decimal matematik.

Lad os tage et simpelt regnestykke:

$$\begin{array}{r} 11 \\ 011 \\ + 110 \\ \hline 1001 \end{array}$$

De to små 1-taller er menter.

At regnestykket passer, ses måske lettest ved at opstille det tilsvarende decimale regnestykke. (3 plus 6 = 9).

Når vi adderer to binære størrelser, kan vi altså få mente, som overføres til den nærmeste højere position.

Når vi inkluderer menten, kan vi komme til at addere op til 3 1-taller, og det giver som resultat naturligvis 1 ned og 1 i mente. Vi kan lave lidt om på det forangående stykke, så det svarer til 7 plus 6.

$$\begin{array}{r} 11 \\ 111 \\ + 110 \\ \hline 1101 \end{array}$$

De enkelte additioner i dette regnestykke følger reglerne for addition, som vi viste øverst.

En CPU med 8 bit matematik vil kunne addere 2 tal på hver 8 binære cifre, hvor det højeste tal, som kan fremkomme, er:

$$\begin{array}{r} 11111111 \\ 11111111 \\ + 11111111 \\ \hline 111111110 \end{array}$$

Denne addition svarer til decimaltallene

$$\begin{array}{r} 255 \\ + 255 \\ \hline 510 \end{array}$$

Nu skal der nok være nogle, som er klar til at protestere, for vi har tidligere sagt, at vi med 8 bit kan angive op til 256 forskellige værdier.

Det er skam rigtigt nok - men husk på, at 0 også er en værdi. Det højeste tal, som kan skrives med 8 binære cifre, svarer til decimal-tallet 255.

Mere bemærkelsesværdigt er det, at vi i resultatet har 9 cifre, hvilket en 8 bit maskine egentlig ikke har chance for at arbejde med.

Denne sidste bit - eller forreste bit - står da heller ikke i akkumulatoren, hvor den øvrige del af resultatet er fremkommet, men i et specielt status-register, hvor en bestemt bit hele tiden antager en værdi, som svarer til den ekstra bit, som er nødvendig for at angive 9 cifre.

Resultatet af den sidste binære addition vil altså i maskinen ikke fremstå som 9 cifre, men som med 8 cifre og en indikering i registret af, at der blev en mente til overs.

Dette statusregisters mente-bit kaldes for *Carry*, og det er ét af de registre, som der er mest brug for.

BINÆR SUBTRAKTION

Datamater kan ikke subtrahere, de kan kun addere. Men heldigvis er det muligt at udføre en addition på en sådan måde, at resultatet svarer ganske til en subtraktion.

Dette kan også lade sig gøre med decimaltal, og vi skal måske først benytte os af et eksempel fra 10-tal systemet, så vi kan koncentrere os om funktionen, og ikke behøver at lade os distrahere af de stadig lidt uvante binære cifre.

Vi tager en simpel subtraktion:

$$\begin{array}{r} 9 \\ -2 \\ \hline 7 \end{array}$$

Samme resultat kan opnås gennem en addition, hvis et par ekstra forholdsregler indføres.

Først må vi finde det *komplimentære* tal til 2. Det komplimentære tal opstår, når vi trækker tallet - 2 - fra 10.

Når det komplimentære tal er fundet - det er i dette tilfælde 8 - adderer vi det til det andet tal - og smider menten væk:

$$\begin{array}{r} 1 \\ 9 \\ + 8 \\ \hline 17 \end{array}$$

Resultatet er altså korrekt blevet til 7. Nu er det ikke videre smart at udføre decimal subtraktion ved hjælp af komplimentær addition, da det ikke er nemmere at lægge 8 til, end det er at trække 2 fra, men sagen er noget anderledes, når det drejer sig om binære tal.

Det tilsvarende komplimentærtal i binær form findes ikke ved at udføre en 10-komplimentering, men en 2-komplimentering. Dette involverer også en subtraktion, hvilket maskinen stadig ikke er istand til, så vi må gå en lille omvej - via en 1-komplimentering.

Når vi 1-komplimenterer et binært tal, kan vi blot bytte om på 0'er og 1-taller, så alle cifre i det nye komplimenterede tal får den modsatte værdi. F.eks.

Binært	0101
1-komplimenteret	1010

Hvis et 1-komplimenteret tal skal laves om til et 2-komplimenteret tal, kan vi blot addere 1 til tallet.

Oprindeligt	0100
1-komplimenteret	1011
Addition med 1	<u> 1</u>
2-komplimenteret	1100

Vi kan tage et eksempel, hvor vi først betragter et normalt opstillet binært regnestykke, som vi - men ikke maskinen - kan udføre det.

$$\begin{array}{r} 10001 \\ -01011 \\ \hline 00110 \end{array}$$

Datamaten må gå en lidt mere komplimenteret vej, men vi opstiller regnestykket, som det ser ud efter 2-komplimenteringen.

$$\begin{array}{r} 1 \\ 10001 \\ + 10101 \\ \hline 100110 \end{array}$$

Også her stryger vi menten, så det faktiske resultat bliver: 00110.

Altså det samme, som vi fandt frem til ved almindelig subtraktion. Heldigvis udføres de nødvendige ekstra modifikationer af tallene automatisk i CPU'en, og det går så hurtigt, at langt de fleste datamater er lige hurtige til at udføre en addition og en subtraktion. Det er ikke altid, at vi trækker et mindre tal fra et større - ofte er det omvendte tilfældet, og resultatet bliver da negativt. Datamaten kan dog ikke skelne mellem positive og negative tal, da den slavisk udfører en addition. Heldigvis er der helt klare regler for, hvad der sker i de tilfælde, hvor resultatet bliver negativt. Sagen er i al sin enkelthed, at der ingen mente bliver fra additionen, som foretages på baggrund af subtraktion af et større tal. Til gengæld vil resultatet være i 2-komplimentær form i forhold til det faktiske resultat.

$$\begin{array}{r} 101 \\ -11011 \\ \hline -10110 \end{array}$$

Dette kan vi regne ud, men maskinen kan ikke angive fortegn, så vi må istedet benytte os af informationen fra menten. Således gør maskinen:

$$\begin{array}{r} 1 \\ 101 \\ +00101 \\ \hline 01010 \end{array}$$

Da der ingen mente er kommet med ned til forøgelse af antallet af cifre, ved vi, at resultatet er negativt, men at det fremstår i positiv 2-komplimentær form. Det kan vi ændre, idet det 1-komplimentære tal til 01010 = 10101, hvortil vi skal addere 1, hvorfor resultatet bliver 10110. Fortegnet må vi selv sætte på, men resultatet er det samme, som før, nemlig - 10110. Når vi udfører subtraktion via 2-komplimentær addition, vil den endelige mente fortælle os, om resultatet er positivt eller negativt.

THE FIRST BOOK OF KIM

En bog på 176 sider indeholdende afsnit for begynderen på KIM 1, et utal af både underholdende og praktiske programmer, information om interfacing og avanceret brug etc.

Pris kr. 75,- incl. moms

INSTRUTEK

Hovedkontor:
Christiansholmsgade
8700 Horsens
Tlf. 05 - 61 11 00

Øst:
Rødovrevej 155
2610 Rødovre
Tlf. 01 - 41 34 00

PROGRAM: Løn & skat

I vor artikelserie om programmering har vi beskæftiget os en del med CPR-numre og løn på et ret teoretisk niveau. Det er naturligvis muligt at udføre noget sådant rent praktisk, og vi har benyttet en TI-59 programmerbar kalkulator med PC-100 printer til at udvikle et program, som beregner løn og skat.

Hvis man anskaffer sig en programmerbar kalkulator med printer, er det næppe for blot at spille „månelanding” eller tilsvarende iøvrigt underholdende spil.

Man vil gerne have praktisk anvendelse og gerne tidsmæssig besparelse ud af sin investering.

Vi viser et eksempel på et program, som ikke alene er særdeles praktisk orienteret, idet det vil udregne en lønseddel med oplysning om skattetræk o.s.v., men et program som samtidig er meget bruger-orienteret.

Der er en tendens til, at programmer udføres på en sådan måde, at informationerne til brugerne indskrænker sig til et absolut minimum, hvorfor det altid er nødvendigt at medbringe fuld dokumentation for det pågældende program, når det skal køre.

Et typisk eksempel er Texas Instrument's egne biblioteksprogrammer, som gemmes i et separat, udskifteligt biblioteks modul.

Angiveligt af hensyn til pladsen og dermed den absolutte udnyttelse af modulet er disses programmer udformet, så det er umuligt at benytte dem, hvis ikke den fulde dokumentation ligger ved siden af.

Netop TI-59 (og 58), som har mulighed for tilslutning til printer, burde have in-

spireret programmørerne til mere brugerorienterede programmer. Ganske vidst er det ikke alle brugere, som har tilsluttet printer, men det vil højst betyde, at en information til printeren i alfanumerisk form ikke umiddelbart kan genkendes, når den ses på LED-display'et. Nok om det - lad os se lidt på selve det program, som forhåbentlig skal illustrere en mere bruger-orienteret holdning.

OPLÆGGET

Programmet skulle ikke blot kunne beregne skat af løn o.s.v., men det skulle udformes på en sådan måde, at det selv gav brugeren besked om alle indtastninger, så chancen for fejl var minimal.

Endvidere skulle programmet efter beregningerne udskrive samtlige resultater på en sådan måde, at disse kunne benyttes direkte til baggrund for en lønseddel. Endelig skulle programmet udformes på en sådan måde, at det senere kunne indgå i et større programkompleks, hvori f.eks. kunne indgå månedsvis summering af alle lønposter, automatisk opsøgning af skatteoplysninger etc.

Indtil videre burde programmet dog kunne opfattes som en selvstændig helhed, som skulle kunne benyttes af enhver, som har med løn og skat at gøre.

UDFORMNINGEN

Da kodningen skulle påbegyndes, fastlagdes først det ønskede resultat og dettes format, altså antal oplysninger og udskriftsrækkefølge af disse.

Det var fra starten klart, at alle oplysninger skulle fremstå på en sådan måde, at der ikke kunne herske tvivl om, hvad de repræsenterede. Udgangspunktet blev derfor en standard lønseddel, som de fleste sikkert kender:

LØNKVITTERING		Lønperiode:									
Arbejdsgiverens navn og adresse:		Arbejdsgivernr.									
Indkomstmottager:		Personnummer:									
		<table border="1"> <tr> <td>Dag</td> <td>Må</td> <td>År</td> <td>Leb.nr.</td> </tr> <tr> <td>20</td> <td>12</td> <td>43</td> <td>22516</td> </tr> </table>		Dag	Må	År	Leb.nr.	20	12	43	22516
Dag	Må	År	Leb.nr.								
20	12	43	22516								
Kontant løn med ferieret		10 236 -									
		126 -									
Ferieberett. løn i alt		10 362 -									
Udbetalte sygedagpenge		50 -									
Løn i alt		10 412 -									
÷ ATP eller anden pension		- 12 -									
A-indkomst i alt		10 400 -									
Stat.-Løn	Frdrag iflg. skattekort	4332 -									
	Trækgrundlag	6068 -									
	Nedrundet trækgrundlag	6060 00									
	A-skat 43 % heraf	2606 00 •									
	Overføres til »Frdrag										
Stat.-Feriepenge	Feriegodtgørelse m. v.										
	Frdrag iflg. skattekort for antal dage: ÷										
	Trækgrundlag	00									
	Nedrundet trækgrundlag	00									
	A-skat % heraf	00 •									
Overføres til ferieort eller lign. (Netto-feriepenge)											
Samlet A-skat		2606 00 •									
Frdrag	A-skat af løn (overføres fra skemaet)										
	Kontingent										
	Ferieopsparing										
	Resteindeholdelse										
FRADRAG I ALT		2606 00 → 2606 -									
Godkendt til udbetaling:	I ALT TIL UDBETALING 7794 -										
Udbetalt den:	+ porto 620,- INTERNT BILAG samt. 8414,-										

Grafik nr. 10230

Det ses med det samme, at der er en næsten utrolig mængde forskellige tal ned gennem en sådan lønseddel, og det var lige så lysende klart, at hvis en fuld udskrift skulle følge hvert beløb, ville hovedparten af tiden - og papiret - gå med udskrift af forklaringer.

Det skal lige indskydes her, at den benyttede kombination af kalkulator og

printer muliggør både numerisk og alfanumerisk udskrift, men principielt kun én ting ad gangen. Det betyder, at man normalt burde udskrive først en linie, som forklarede, hvad der fulgte, hvorefter det numeriske resultat kom på næste linie.

Den alfanumeriske information gemmes i alm. hukommelsesregistre, men hvert register kan kun „huske“ op til 5 alfanumeriske karakterer, så en større udskriftsprocedure vil kræve meget stor lagerkapacitet.

Derfor blev det besluttet at udnytte den mulighed for samtidig udskrift på én linie af numerisk og alfanumerisk information, som den specielle ordre OP 06 giver mulighed for. Den tillader nemlig udskrift af numerisk værdi fra displayregistret (resultatet) samtidig med at op til 4 alfanumeriske karakterer trykkes yderst til højre på samme linie.

Det indledende arbejde bestod derfor i at finde frem til de ord og forkortelser, som af brugeren ville give tilstrækkelig information om kørslen, og som samtidig optog minimum af lagerplads. Resultatet blev et mindre bibliotek af ord på op til 4 karakterer.

Det blev til ialt 15 tekster, som lagres i registerne 10 til 24 incl.

Nedenstående udskrift er sammensat af to listninger, så både registernr, numerisk indhold og tilsvarende alfanumerisk udskrift kan vises:

610013.	% A	10
21351635.	FRDR	11
13413732.	AUTO	12
36452233.	SYGP	13
174437.	EXT	14
22132217.	GAGE	15
153335.	CPR	16
37103637.	TAST	17
20133733.	-ATP	18
13243116.	AIND	19
24132797.	IALT	20
37262235.	TKGR	21
31171635.	NEDR	22
36261337.	SKAT	23
41161437.	ÜDBT	24

løvrigt benytter programmet et minimum af registre, og f.eks. den oprindelige information om gagen gemmes ikke, da det som nævnt er forudsat, at programmet skal kunne indgå i et større programkompleks, hvor netop denne oplysning ligger fastindkodet.

Registre 00 til 07 benyttes som alm. lagerregistre og arbejdsregistre, mens lagerregistre 08 og 09 anvendes ved indirekte adressering til at pege på hhv. den celle, hvor informationen skal hen, og hvor den alfanumeriske udskrift skal tages fra til de ledsagende udskrifter.

Et par sub-rutiner viser brugen af indirekte adressering.

Se f.eks. på LBL A', som starter i ordre nr. 20. 22 - 27 incl. kalder adressen fra register 09, placerer indholdet fra denne adresse i udskriftsregister 4 og trykker én linie alfanumerisk tekst ud. Ved at placere forskellige adresser i register 09 kan denne lille rutine udskrive forskellig tekst. Netop denne rutine benyttes til indlæsning af data, hvor maskinen f.eks. skriver TAST CPR.

Ordet tast er i forvejen blevet placeret i udskriftsregister 3 - instruktion 007 - 010.

En anden subrutine, LBL B', ligger i adresse 013. Denne rutine udskriver en linie kombineret numerisk og alfanumerisk information, som hentes fra X og T registrene. Disse registre skal altså være fyldt med den rigtige information, når B' kaldes. Se f.eks. ordrene i 041 til 046, hvor X og T registret fyldes med information. Når B' kaldes herefter, opnås udskrift af CPR-nr. med den passende forklaring.

Indlæsningen af data er iøvrigt interessant, idet der anvendes en løkke med indirekte adressering - ordrer 021 - 039. Når udskrift af indholdet af udskriftsregistret er sket - informationen er hentet via reg. 09 - nulstilles X-registret, så det ikke er nødvendigt at indtaste 0, hvis den følgende information er blank. (ordre 028 - 029). I 030 trykkes en kopi af den indtastede information, så det er muligt senere at se, om evt. fejl skyldes fejlindtastninger. Samme informati-

on, som netop er udskrevet, gemmes i det register, hvis adresse er gemt i lager nr. 08 (ordre 031 - 032).

Nu nedtælles indholdet af celle nr. 09 og 08, idet sidstnævnte sker via en check på indholdet af 09. (033 - 039).

Først når reg. 09 er nedtalt til 0 vil programmet fortsætte.

Effekten af denne løkke er, at den først placerede tekst TAST bibeholdes, mens den efterfølgende tekst (CPR, GAGE, FRDR o.s.v.) skiftes ud, idet der udskrives på tur fra celle nr. 16, 15, 14 o.s.v.

Samtidig vil nedtællingen af reg. 09 betyde, at den efterfølgende indtastede information på skift indlæses i lager 07, 06, 05 o.s.v.

Celle nr. 0 bliver aldrig fyldt, da nedtælling og kontrol af den indirekte adresse i 09 sker efter udskrift og indlæsning.

Når vi støder på ordre 040, er hele indlæsningen overstået, og disse første 40 instruktioner har altså initialiseret hele programmet, udlæs ialt 7 linier alfanumerisk tekst og indlæst lige så mange digitale informationer med efterfølgende kontroludskrift.

Dette burde være en glimrende illustration af, hvorledes brugen af løkker og indirekte adressering kan spare mange indtastede programtrin. I dette tilfælde, hvor løkken kun gennemløbes 7 gange, er besparelsen i kraft af nedtællinger og kontrolordrer selvfølgelig ikke gigantisk, men alligevel i størrelsesordenen 50 trin.

BEREGNINGERNE

Den øvrige del af programmet er helliget beregningerne og de påhæftede udskrifter, og brugen af sub-rutiner er i denne del noget indskrænket, idet B' dog benyttes nogle gange i forbindelse med selve udskriften.

Beregningerne er ganske simple, da der ingen særlige vanskeligheder er forbundet med beregning af skat, men det kan måske være på sin plads at pege på 2 lidt specielle funktioner, nemlig nedrundning og afrunding.

I ordrer 113 - 123 foretages nedrundning til nærmeste hele 10-ere. Dette sker ved en division med 10, fjernelse af evt. de-

cimaldel efter kommaet og efterfølgende multiplikation med 10.

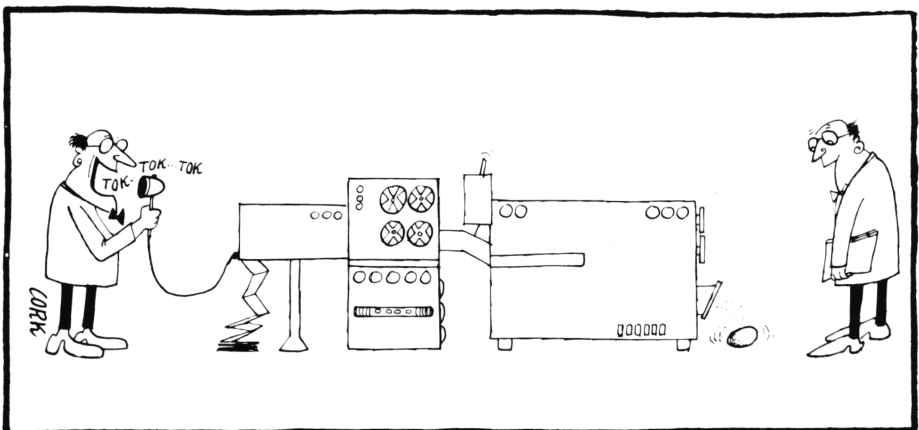
Afrundingen af skattebeløbet sker i 146 - 151, hvor der tillægges beløbet 0,5, og herefter fjernes decimaldelen efter kommaet.

Brugen af T-registret er en besparelse i både tid og programtrin og svarer helt til den brug af interne registre, som finder sted i rigtige datamater.

PROGRAMLISTNING

000	01	1						
001	06	6	036	09	09	071	32	X!T
002	42	STD	037	97	DSZ	072	43	RCL
003	09	09	038	08	08	073	13	13
004	07	7	039	16	A*	074	17	B*
005	42	STD	040	98	ADV	075	43	RCL
006	08	08	041	43	RCL	076	06	06
007	43	RCL	042	07	07	077	32	X!T
008	17	17	043	32	X!T	078	43	RCL
009	69	DP	044	43	RCL	079	20	20
010	03	03	045	16	16	080	17	B*
011	16	A*	046	17	B*	081	01	1
012	76	LBL	047	43	RCL	082	02	2
013	17	B*	048	06	06	083	22	INV
014	69	DP	049	32	X!T	084	44	SUM
015	04	04	050	43	RCL	085	06	06
016	32	X!T	051	15	15	086	32	X!T
017	69	DP	052	17	B*	087	43	RCL
018	06	06	053	43	RCL	088	18	18
019	92	RTN	054	05	05	089	17	B*
020	76	LBL	055	44	SUM	090	43	RCL
021	16	A*	056	06	06	091	06	06
022	73	RC+	057	32	X!T	092	42	STD
023	09	09	058	43	RCL	093	00	00
024	69	DP	059	14	14	094	32	X!T
025	04	04	060	17	B*	095	43	RCL
026	69	DP	061	43	RCL	096	19	19
027	05	05	062	06	06	097	17	B*
028	25	CLR	063	32	X!T	098	43	RCL
029	91	R/S	064	43	RCL	099	02	02
030	99	PRT	065	20	20	100	22	INV
031	72	ST*	066	17	B*	101	44	SUM
032	08	08	067	43	RCL	102	00	00
033	01	1	068	04	04	103	32	X!T
034	22	INV	069	44	SUM	104	43	RCL
035	44	SUM	070	06	06	105	11	11

106	17	B*	132	32	X:T	158	23	23
107	43	RCL	133	43	RCL	159	17	B*
108	00	00	134	10	10	160	43	RCL
109	32	X:T	135	17	B*	161	06	06
110	43	RCL	136	43	RCL	162	32	X:T
111	21	21	137	01	01	163	43	RCL
112	17	B*	138	49	PRD	164	24	24
113	43	RCL	139	00	00	165	17	B*
114	00	00	140	01	1	166	43	RCL
115	55	÷	141	00	0	167	03	03
116	01	1	142	00	0	168	44	SUM
117	00	0	143	22	INV	169	06	06
118	95	=	144	49	PRD	170	32	X:T
119	59	INT	145	00	00	171	43	RCL
120	65	×	146	43	RCL	172	12	12
121	01	1	147	00	00	173	17	B*
122	00	0	148	85	+	174	43	RCL
123	95	=	149	93	.	175	06	06
124	42	STD	150	05	5	176	32	X:T
125	00	00	151	95	=	177	43	RCL
126	32	X:T	152	59	INT	178	20	20
127	43	RCL	153	22	INV	179	17	B*
128	22	22	154	44	SUM	180	98	ADV
129	17	B*	155	06	06	181	98	ADV
130	43	RCL	156	32	X:T	182	61	GTD
131	01	01	157	43	RCL	183	00	00



KØRSEL AF PROGRAMMET

Der behøves næppe mange kommentarer til selve kørslen, da denné ledsages af maskinens egne forklaringer, men det bør måske tilføjes, at maskinen selv fratrækker ATP med 12 kr - dette sker i 081 - 082. Hvis individuel beregning af ATP ønskes, kan her placeres et kald af en subrutine istedet, som sørger for det fornødne.

EXT står for extra, og det kan dække både tilskud og fradrag, idet dette beløb slavisk adderes til den alm. gage. Et fratræk i lønnen foretages altså ved indtastning af negativt beløb.

Endelig skal tilføjes, at der kunne opnås lidt kortere program og kørsel, hvis beregningerne foretoges løbende med indtastning af data og disse blev efterfulgt af mellemresultater med det samme, men dette vil ikke alene give en mindre overskuelig udskrift, men vil også gøre det sværere senere at koble programmet sammen med andre funktioner.

Den benyttes standard-splitning, og både program og data kan gemmes på et enkelt kort, som indlæses fra hhv. side 1 og 4. Mindre end halvdelen af maskinens kapacitet er benyttet til dette program.

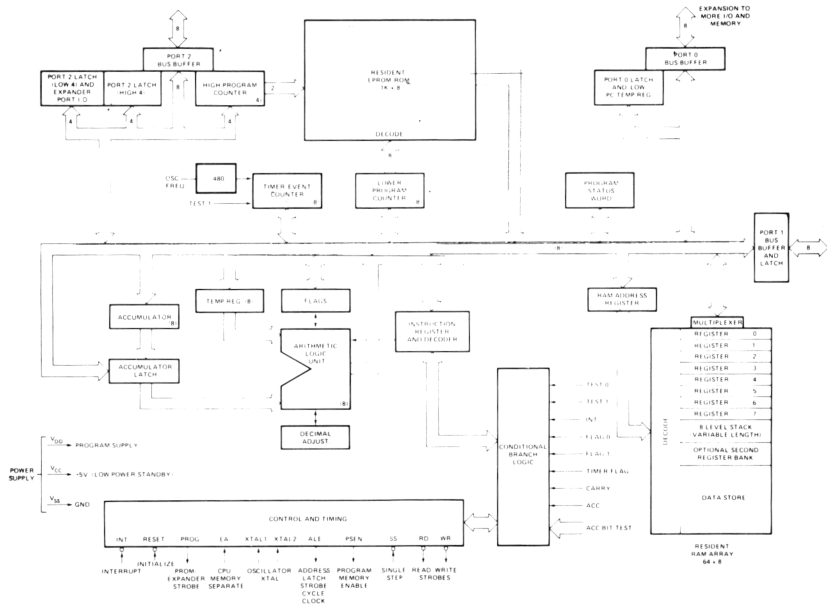
MODIFIKATIONER

Smarte læsere og indehavere af en TI-59 skulle ikke finde det svært at lave eventuelle modifikationer til programmet.

Det kan tænkes, at gagen ikke er fast, men skal indlæses som antal timer og timeløn, ligesom det måske er overflødigt med udskrift af kørselsgodtgørelse. En gennemgang af programmet vil hurtigt vise, hvor der skal (og kan) ind sættes og udelades ordre.

	TAST	CPR
201243.	2256	
	TAST	GAGE
10236.		
	TAST	EXT
126.		
	TAST	SYGP
50.		
	TAST	AUTO
620.		
	TAST	FRDR
4032.		
	TAST	% A
43.		
<hr/>		
201243.	2256	CPR
10236.		GAGE
126.		EXT
10362.		IALT
50.		SYGP
10412.		IALT
12.		-ATP
10400.		AIND
4032.		FRDR
6068.		TKGR
6060.		NEDR
43.		% A
2606.		SKAT
7794.		UDBT
620.		AUTO
8414.		IALT
<hr/>		
	TAST	CPR

INTEL 8048



HARDWARE

- Teknologi** N-kanal MOS
- Ordlængde** 8 bits
- Adressering** 12 bits adressebus (max. 4K programlager). Internt data-lager på separate busledere.
- Interne registre** 1 akkumulator plus 8 temp. reg. plus 56 int. lagerregistre.
- Clock** Intern 6 MHz m. eksternt krystal.
- Belastbarhed** Alle terminaler: ca. 1-2 TTL-lastenheder.
- Spændingsforsyning** Plus 5 volt.
- Statusord** 4 bits stack-pointer, MSB af adressebusen. 2 carry-bits og 1 ekstra flag.
- Typisk cyklustid** 2,5 µs.
- Mulighed for DMA** Nej.
- HFD april 1978**

SOFTWARE

Inst. iktioner	108
Interrupt	1 int. indgang. Flere interups kun ved polling.
Adressestak	Intern i 8 niveauer, hver med plads til PC og PSW.
BCD-aritmetik	Speciel instruktion til justering af akkumulator.
Betingede hop	Kun indenfor samme side (256 bytes)
Input/Output	3 x 8 bits plus 3 bits in/out direkte på chip'en. Ubegrænsede udvidelsesmuligheder.

KOMMENTARER

Med INTEL's nye MCS-48 serie af one-chip serie af micro-controllers er vi kommet ned i et prisleje og et pladsforbrug, der skaber helt nye anvendelsesmuligheder.

Serien består af 3 forskellige typer:

8748 med 1 K internt E-PROM-programlager.

8048 med maskeprogrammerbar 1 K ROM.

8035 uden programlager.

Fælles for dem alle er: Med kun én 40-bens kreds, et krystal og en strømforsyning har man en komplet mikroprocessor med 1 K programlager, en 8 bit programmerbar timer/tæller, en 8-niveaurs adressestak, 3 parallelle 8-bits I/O porte, 1 interrupt-indgang med fast programmeret vektor, samt 2 indgange, som kan testes med conditional jumps ligesom de interne flag.

Uden at være en kopi, så er instruksions-sættet tydeligvis valgt ud fra erfaringerne med 8080/8085, men dette er en del mere effektivt, og erfaringerne viser, at man i 8048 kun bruger 60 % af de bytes, som i en 8080 ville være nødvendigt for løsning af samme opgave.

Desuden findes instruktioner i 8048, som er beregnet til brug sammen med de specielle MCS-48 periferikredse.

Alt i alt er det tydeligt, at 8048 er beregnet som controller i små, fikse systemer, og prisen (omkring kr. 400,- for 8748) skulle vel tiltale de fleste.

YDERLIGERE KREDSE

INTEL har til 8048 lanceret en række kredse specielt designet til at køre sammen med denne med brug af så få ekstra komponenter som muligt.

Således kan man udvide programlageret med 2K med blot 1 ekstra kreds, og man kan med en anden kreds tilføje 256 bytes RAM, 3 output porte og en 14 bit timer/tæller. En tredje kreds udvider antallet af porte med 4.

Fælles for disse kredse er, at der i 8048' instruktionssættet er specielle instruktioner, som er beregnet netop for funktionerne i disse periferikredse.

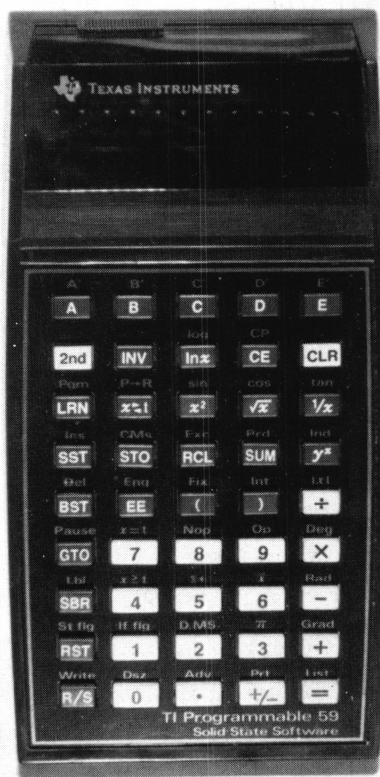
Udover disse specielle kredse kan naturligvis alle andre standard-kredse benyttes sammen med 8048 ved tilføjelse af latches, dekodere o.s.v.

Til udviklingsformål findes en såkaldt PROMPT-48, hvormed man kan udvikle programmer, køre disse i single-step, breakpointing eller real-time, og endelig programmere EPROM'en i 4748.

Sidstnævnte er dog en dyr dreng til omkring 10 af de store sedler, så for de, der er lidt mere sparsommelige, findes 8048 også i et byggesæt fra IMSAI. (Se anmeldelsen side M30). CM

INTEL forhandles af Scansupply.
Tlf. (01) 83 50 90. Det er ikke os bekendt, at 8048 fremstilles af andre end INTEL. Prisdé: Ca. kr. 400,- for 8748.

TI-59 & PC-100



TI-59 i ensom majestæt. På side A5 er den vist sammen med printer PC-100. Pris for hele herligheden ca. kr. 4.690,—. (Det er billigt).

Der er så mange læsere, som har efterlyst programmer til og omtale af Texas Instruments større lommeregnerne, at vi har følt os foranlediget til at tage opfordringerne op. Her beskriver vi - alt for kort - en TI-59 med printer. I dette og følgende numre skal vi yderligere supplere med en række programeksempler.

Det er snart et år siden, at Texas Instruments præsenterede deres nuværende top-model i programmerbare lommeregnerne, TI-59.

TI-59 er en alvorlig affære, som til trods for sit format næsten ikke kan kaldes for en lommeregner, da den - hvis fuld udnyttelse af dens kapacitet ønskes - vil være fastspændt på toppen af printeren PC-100.

Til sammen udgør disse snarere en mini-bordkalkulator med avancerede mulig-

heder, som på mange måder minder om en fuldt udbygget microdatamat.

MASKINENS FUNKTIONER

Der er ikke plads her til at beskrive alle de mange funktioner, som sætter TI-59 i særklasse, men en kort gennemgang af de punkter, hvorved den adskiller sig fra de mere gængse, skulle alligevel give et nogenlunde indtryk af dens muligheder. Maskinen benytter ligefrem notation som de øvrige Texas' maskiner - og helt

på linie med hovedparten af markedets andre lommeregnerne. (Det er faktisk kun Hewlett Packard - os bekendt - der benytter sig af omvendt notation).

Ligeform notation betyder, at matematiske formler indtastes helt som de er nedfældet på papiret - med parenteser og det hele, og den mellemregningsprocedure, som normalt foretages, sker internt i maskinen.

Personligt foretrækker jeg omvendt notation, da det minder mere om det, der sker i en datamat, men selv jeg, der med flere år i selskab med en HP-25 må siges at være lettere miljøskadet, havde ikke særlig svært ved at sætte mig ind i brugeren af en TI-59.

Der er en række punkter, hvor TI-59 adskiller sig fra mere almindelige lommeregnerne, og dem vil vi nu se på.

L

6

PROGRAMMERBAR

Maskinen programmeres på 3 forskellige måder. Hyppigst via tastaturet, hvorved man fremstiller egne programmer, men også ved hjælp af et ROM-modul, som isættes i bunden af maskinen.

Dette ROM-modul indeholder en række biblioteksprogrammer, og udover det modul, som følger med maskinen, kan yderligere moduler anskaffes.

Endelig kan maskinen programmeres via den indbyggede magnetkortlæser, hvor man kan gen-indlæse egne programmer eller andres programmer til kopiering - maskinen kan naturligvis også udskrive sine programmer på magnetkort.

Når maskinen skal programmeres fra tastaturet, nedtrykkes LRN (Learn) og den efterfølgende tastrækkefølge kopieres ind i maskinen hukommelse.

Et efterfølgende tryk på LRN vil bringe maskinen til klar-position, og en RST (Reset) og R/S (Run/Stop) vil få maskinen til at gentage, hvad der netop er indkodet.

Biblioteksprogrammerne i ROM-modulet kan enten benyttes, hvor de ligger, eller de kan kopieres ind til selve laget. Der var ingen tidsforskel at bemærke på kørsel af de samme programmer med de to mulige placeringer, men det

er praktisk at kunne kopiere ROM'en ind i maskinens egentlige lager alligevel, da man derved får mulighed for at ændre eller tilføje på disse programmer, der af hensyn til pladsen er blottet for enhver brug af alfanumerisk udskrift af resultater på printeren.

PRINTEREN

Printer PC-100, som man får megen glæde af, er ikke helt lille, men dens evne til alfanumerisk udskrift kan ikke underkendes. Den kan reproducere ialt 64 forskellige karakterer, hvorved den sættes i stand til ikke alene at lave en komplet programudskrift, men kan ledsage kørsel af programmer med forklarende tekst til supplement af resultaterne - se programmet side B 21.

Programrutinerne omkring alfanumerisk udskrivning er desværre ret langsomme - printeren selv er pænt hurtig. Når man ønsker at fylde en linie op med alfanumerisk udskrift og trykke den, vil der gå omkring 1 sekund med dette.

Printeren kan sættes til at kopiere hvert trin i en kørsel, og det viste sig at være en glimrende hjælp til fejlfinding i nye programmer, hvor mærkelige resultater pludselig dukkede op undervejs.

PC-100 fungerer som strømforsyning for TI-59, og lommeregneren kan fastlåses til printeren, som igen kan fastskrues til skrivebordet - det skulle reducere faren for tyveri noget. (Og næste morgen var skrivebordet forsvundet!)

```

TAST1  INLDATO  +H/E
H=NRB  2  H=ALT  E=EGU
DATO:  C  =  CHECK  DATO
RST  =  SLET  OG  FORFR
                                KLAR

```

Eksempel på alfanumerisk udskrift fra printeren PC-100. Udskriften her er indledende forklaring til et program for kørselsregnskab.

MASKINFORMAT

TI-59 benytter ikke i samme omfang som HP's maskiner flettede instruktioner, og der skal derfor generelt anvend-

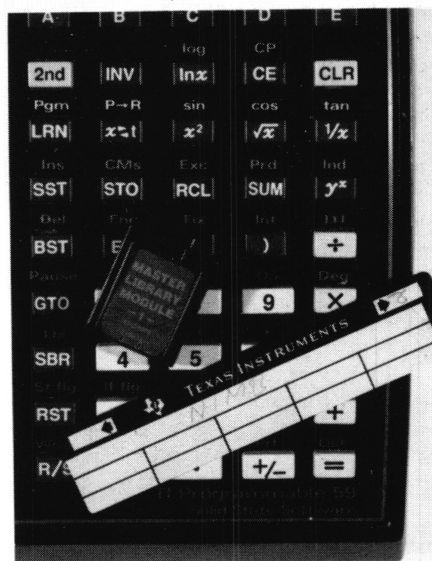
es flere instruktioner i en TI-maskine til at udføre samme opgave.

Dette skyldes til dels den ligefremme notation, som giver mindre mulighed for at udnytte flettede ordrer, og snare-re end at have en maskine med lange programtrin, som ikke fyldes ud, har man taget konsekvensen af dette og la-vet maskinen med korte programtrin - men til gengæld rigtig mange af dem.

Tidsmæssigt betyder det ikke meget for udførelsen af programmerne, da disse maskiner stort set arbejder i serie og flytter og behandler 1 bit ad gangen, men da HP-maskinerne skal skifte adresse lidt sjældnere, bliver de en anelse hurtigere. Et lille program, hvor 2 maskiner skulle tælle til 100, tog på en HP ca. 15 sek, mens det på en TI tog 18 sek.

Når man tænder for en TI-59 har man 480 programtrin og 60 lagerregistre til sin rådighed, men dette forhold kan ændres, idet hvert lagerregister svarer til 8 programtrin.

Hvis man ønsker mange lagerregistre,



De 3 programmeringsmuligheder i TI-59 vist i form af en del af tastaturet, et biblioteks-modul og et magnetkort.

Synes du, det går for langsomt?

Vi kan tilbyde engelsksproget litteratur, hvorfra du kan lære det hele - på én gang. F.eks.:

- Computer LIB, en gennemgang af de vigtigste aspekter kr. 70,-
- Best of Byte, uddrag fra ét af USA's bedste magasiner kr. 90,-
- Best of Creative Computing - fra USA magasin kr. 90,-
- BASIC BASIC - lær sproget BASIC på den nemme måde kr. 90,-

Ring eller skriv efter den nyeste litteraturliste - fremsendes gratis.

piezodan aps.

Bakkedraget 55 - DK 3480 Fredensborg - Tlf. (03) 28 37 44 - Teknisk afd. (01) 86 12 17

kan man få op til 100 af disse på bekostning af programtrin, idet der så „kun“ er plads til 160 af disse. Til sammenligning kan vi igen henvide til programmet om skat og løn.

Hvis der ikke er behov for lagerregistre, kan der opnås ialt 960 programtrin.

Dette format kan varieres i spring på 10 lagerregistre/80 programtrin.

Yderligere fleksibilitet opnås ved brug af magnetkortene, som kan indeholde både data og program. Det er muligt under en kørsel at stoppe op for indlæsning af yderligere data eller programtrin, og når der ses bort fra den tid, som dette involverer, har man i virkeligheden ubegrænset plads til rådighed.

Hele maskinens lager af program og data kan være på 2 magnetkort.

L

8

SÆRLIGE FORTRIN

Som problemløser og programmerer er man især glad for den mulighed for redigering, som findes i DELETE og INSERT funktionerne. Det er altså ikke blot muligt at fjerne én eller flere instruktioner, men det er ligeledes muligt at indskyde nye ordre, uden at slette i det bestående program.

Under en sådan redigering ændrer de forskellige instruktioners adresser sig selvfølgelig, og maskinen er derfor udstyret med mulighed for brug af etiketter, hvilket letter brugen enormt.

Næsten alle taster kan benyttes som etiketter, men især 10 brugerdefinerede

0.	30
0.	31
120.	32
34.	33
10.	34
364114.	35
26271335.	36
16133732.	37
0.	38

TI-59 kan via printeren levere en listning af lagerregisterne, her R 30 - 38 incl.

taster er nyttige, da disse indgår i tests og forgreninger, subrutiner m.m.

TI-59 er forsynet med flag, som kan sættes udefra eller i programmet, og disse flag kan betinge programmets forløb. Man kan således forestille sig, at man til et program kan lade et flag bestemme, om printeren skal lave detaljerede eller sparsomme beskrivelser af programmets forløb.

Maskinen har et særligt register, T-registret, som er særdeles anvendeligt til kortvarige opbevaringer af data, sammenligninger o.lign.

SAMLET VURDERING

TI-59 virker umiddelbart lidt klodset i sammenligning med de meget elegante maskiner fra HP, og materialevalg, knapper o.s.v. fungerer heller ikke helt så lækkert.

Den ligefremme notation er sikkert lettere tilgængelig for de fleste, men der kan med omvendt notation skrives mere effektive programmer - d.v.s. både kortere og hurtigere.

Dette kunne virke som udelukkende negativ vurdering, men der ligger så mange muligheder og så stor kapacitet gemt i en maskine som TI-59, at Texas med denne maskine må siges at holde positionen som fremstiller af markedets mest avancerede lommeregner.

De forskellige styringsmuligheder under programmering og redigering gør sammen med printeren TI-59 til en maskine, som vil kunne være en bogholderimæssig aflastning i mindre virksomheder.

Hvis man stiller lidt mindre krav til kapacitet og alsidighed, kan man evt. vælge „lillebror“, TI-58, som er væsentligt billigere - til gengæld må man give afkald på brugen af magnetkort og en del af lagerkapaciteten.

Personligt vil jeg foretrække TI-59 med PC-100, da denne kombination er så slagstærk, at den bortset fra hastigheden kan opfylde de flestes behov og vil kunne løse næsten enhver tænkelig opgave.

Og så kan man egentlig ikke forlange stort mere.

PH

MG800

TOTAL DEVELOPMENT SYSTEM

KOMPLET KEYBOARD

5 ELLER 9 TOMMER SKÆRM (EL. EGET TV)

BÅND INTERFACE

ASSEMBLER OG EDITOR PÅ PROM

BASIC TOLKER PÅ PROM

8K ELLER 16K LAGER (ELLER MERE)

PROM PROGRAMMERER

POWER SUPPLY OG TDS KABINET

BILLIG LINJE PRINTER 30 TEGN / SEK

ALT ER SAMLET OG AFPRØVET

START MED KR. 6509,00 – BYG 6800 MODULER PÅ

(03) 38 57 16

gds-henckel aps



a franchised **MOTOROLA Semiconductor** distributor

Datamat seminar

Vi har nu fået mulighed for at indbyde til det første hobbydatamat-seminar i Danmark. 3 fremstående amerikanske foredragsholdere rejser gennem landet d. 21 april, og vi har formået dem til at blive en dag ekstra, så vi også i Danmark kan få glæde af deres viden og erfaring.

Selvom der her i landet findes adskillige, som har glimrende kendskab til datamater, er hobby-området så forholdsvis nyt og endnu domineret af indflydelsen fra USA, at det er dér den største viden og erfaring findes.

Det er derfor glædeligt, at det har været muligt at formå 3 af de største kapaciteter på området at forlænge en mellemlanding i Danmark, så vi kan tilbyde alle interesserede at o vervære en række foredrag med efterfølgende spørgetid.

De 3 gæstende foredragsholdere er:

Dr. Portia Isaacson
Mr. Ted Nielson
Mr. Jim Warren

Læsere af amerikanske datamat-tidsskrifter vil utvivlsomt have stødt på disse navne.

De 3 hovedemner bliver:

Datamaten i forretningslivet
Philosofien bag microdatamaten
Programmering af microdatamaten

Hvor de to sidste foredrag er af enorm generel interesse, vil det første, som især henvender sig til de mindre virksomheder, give interesse alle, som har eller påtænker anskaffelse af microdatamat m. økonomisk vinding for øje. Hvert foredrag bliver på ca. 40 min. med efterfølgende spørgetid.

Alle 3 foredrag holdes på engelsk, og de 3 foredragsholdere har lovet at tale både langsomt og med letfatteligt sprogbrug.

Det omtalte seminar bliver afholdt:

Fredag d. 21. april kl. 14.00
Det nye Bella-center på Amager

Det er en fredag, så også provinsboere vil have chance for at komme med.

Stedet er bl.a. valgt af hensyn til de fine parkeringsmuligheder og placeringen i nærheden af lufthavnen.

Firmaet Piezodan er trådt ind som økonomisk garant - det er ikke et helt billigt arrangement - og prisen er derfor så lav som:

kr. 50,-

ved forudbetaling til Piezodan.

Billetter ved indgangen koster **kr. 75,-**.

Ved en evt. aflysning vil alle forudbetalte beløb selvfølgelig blive refunderet og besked herom fremsendt rettidigt.

TILMELDING

Send en check på kr. 50,- eller via giro på konto 1 20 76 60 til:

Piezodan ApS
Bakkedraget 55
3480 Fredensborg

Tilmeldingen må være Piezodan i hænde senest d. **15. april**.

Vi kender de 3 foredragsholdere og kan stille alle, der møder op, en uhyre interessant aften i udsigt. På gensyn. **RED**

Subrutiner

Brugen af subrutiner er en stor lettelse i programmeringen og kan have afgørende betydning for, om et program kan være i en maskine, da subrutiner samtidig betyder en pladsbesparelse.

Subrutiner vil ikke gøre kørslen hurtigere, og man må derfor i hver enkelt tilfælde vurdere, om brugen af subrutiner er ønskeligt.

HVAD ER EN SUBROUTINE?

Det er en programstump, som er udformet på en sådan måde, at datamaten efter udførelse af denne del af programmet vender tilbage til det sted, hvorfra subrutinen blev kaldt.

I opbygning og programmering adskiller den sig ikke fra selve programmet, og man kunne derfor lige så godt kalde en subroutine for et under-program - for det er, hvad det er.

Illustration fra den dansksprogede manual til TI-59, som forklarer brugen af subrutiner.

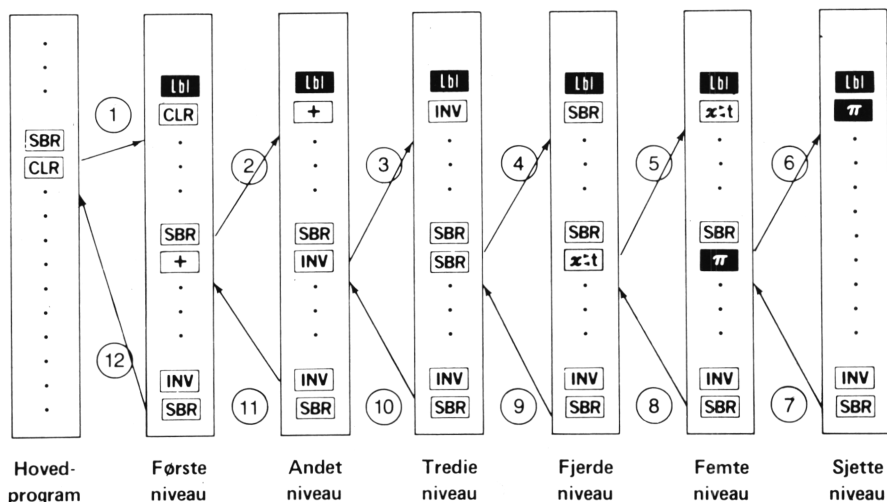
Hvis vi betragter programmerbare kalkulatorer, ser vi, at disse er oplagte emner for brug af subrutiner, da de som regel er mere begrænsede i deres lagerkapacitet, end større datamater. Selv en TI-59, som kan indstilles til at have op til 960 programtrin, vil kunne komme ud for pladsproblemer, hvis ikke et rimeligt snævert udvalg af subrutiner bruges under programmeringen.

BRUGEN AF SUBROUTINER

En subroutine kaldes på forskellig vis afhængigt af den pågældende maskine og sprog.

I f.eks. en TI-59 placeres blot navnet på den pågældende subroutine på passende sted i programmet, hvis det vel og mærke drejer sig om én af de 10 navne, som er beregnet til bruger-definerede funktioner.

Subrutinen indledes derefter med en etikette, som maskinen kan genkende, og for at undgå, at maskinen tror, at navnet igen er et kald af den samme sub-



rutine, skrives i programmet LBL A, hvor LBL er en forkortelse for LABEL, som betyder etiket - og A er navnet på den pågældende subrutine.

Subrutinen afsluttes med instruktionen RTN, hvilket betyder RETURN eller retur til udgangspunktet.

Når der oprettes og kaldes en subrutine første gang, benyttes der altså ialt 4 instruktioner til dette alene, hvoraf de 3 ligger i subrutinen. Hvert yderligere kald kræver 1 ekstra instruktion i programmet.

Det vil aldrig kunne betale sig at oprette en subrutine, som kun kaldes fra ét sted i programmet (hvis ikke specielle løkker gør dette særdeles formålstjenligt), så man må normalt gå ud fra, at en subrutine kaldes mindst 2 gange, hvilket kræver ialt 5 instruktioner blot for at styre disse kald.

Hvis det pladmæssigt skal blive til en besparelse, må selve programindholdet af subrutinen altså være på mindst 6 instruktioner.

Hvis subrutinen kaldes flere gange, kan det set fra et pladmæssigt synspunkt betale sig med også kortere subrutiner. En subrutine i en TI-59, som f.eks. kaldes fra 3 forskellige steder i hovedprogrammet, kræver ialt 6 instruktioner til styringen, så blot 4 instruktioner i selve subrutinen vil betyde en besparelse.

(De 6 instruktioner plus de 4 i selve subrutinen optager ialt 10 programtrin, mens 3 x 4 programtrin kræver ialt 12 programtrin i hovedprogrammet.)

Tidsmæssigt er der ingen fordel ved subrutiner, idet styringen af adresser, opsøgning af rutinen o.s.v. tager relativ lang tid. Tidsforbruget kan reduceres noget ved at placere subrutinerne forrest i lageret, da de fleste maskiner begynder en slavisk opsøgning fra dette punkt, indtil den pågældende subrutines etiket er fundet.

Hvis man laver et kraftigt struktureret program, vil man i f.eks. en TI-59 let kunne komme ud for, at de 10 brugerdefinerede etiketter ikke er tilstrækkelige.

Det betyder, at man må definere i sit

hovedprogram, at en kommende etiket er navnet på en subrutine ved at placere ordren SBR (Subrutine) forinden etiketten.

Og dette betyder igen, at der medgår 1 ekstra instruktion for hver gang den pågældende subrutine kaldes. Konklusionen er naturligvis, at disse subrutiner skal give relativ større programmæssig besparelse, før det kan betale sig at benytte sig af dem.

HVORNÅR?

Der er 2 måder - principielt - hvorpå man finder frem til, at der bør oprettes en subrutine.

I tilfælde af struktureret programmering (bliver grundigt omtalt senere) fastlægger man et enkelt programforløb, som kun indeholder den egentlige styring, de fleste af beslutningerne o.s.v., mens enhver defineret behandling udføres af en subrutine.

Ved mere almindelig programmering, hvor man blot begynder fra en ende af, og fortsætter til man er færdig, vil man ved gennemløb af programmet kunne se programstumper flere steder, som for adskillige trins vedkommende er identiske - eller som ved enkel omskrivning kan blive det.

Disse stumper pilles ud fra hovedprogrammet, hvor der istedet placeres et kald af den pågældende subrutine.

Programmet *Løn og Skat* på side B 21 benytter flere steder subrutiner.

Det efterfølgende program, *Kørselsregnskab*, er udført som struktureret programmering med udstrakt udnyttelse af subrutiner, både komplette afsnit og som splittede subrutiner, hvilket behandles separat i efterfølgende artikel af Jan Pierre Perrier fra Nivå

Vi beklager meget, at navnet på forfatteren af efterfølgende artikel om splitning af subrutiner var „faldet” ud, da artiklen bragtes.

Artiklen blev indsendt af Jan Pierre Perrier fra Nivå.

Z-80 RAM

Sidst beskrev vi ROM-kortet - det kort, som gør det muligt at kommunikere med datamaten, så snart der sættes strøm til. Nu skal vi se nærmere på den del af hukommelsen, der normalt udgør den største del af en moderne amatør-datamat - især i pris - nemlig RAM-lageret.

ROM ELLER RAM

Denne type lager er - især til 8-bit maskiner - normalt udført med halvlederhukommelse, hvorfor det er volatilt (informationen kan overskrives via programmet, og informationen forsvinder, når strømmen afbrydes).

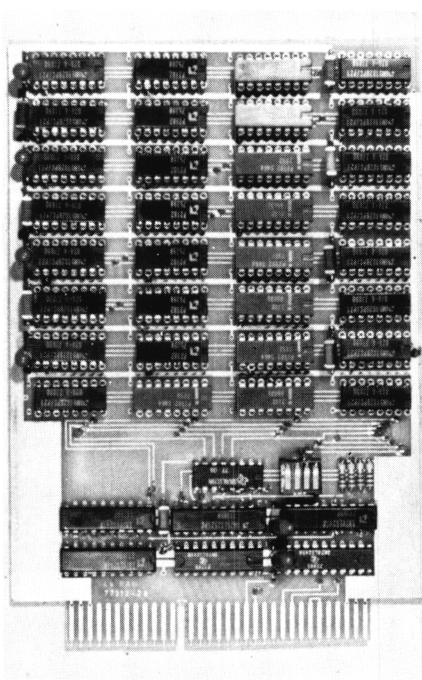
Der findes datamater, som kun har denne lagertype, hvilket ellers i almindelighed frarådes, da det betyder, at man enten skal indlæse selve programmet eller en binær loader (indlæsningsprogram), hver gang, man starter - ved håndkraft! Derefter kan en evt. loader indlæse selve arbejdsprogrammet.

Dette er besværligt og vil let resultere i, at man ikke får den gevinst af datamaten, som man håbede på.

Modsat er det muligt at fremstille en datamat med udelukkende ROM-lager. Denne løsning ses anvendt i tilfælde, hvor der er nøje fastlagte opgaver, der skal løses, men de moderne CPU'er er ikke særlig glade for dette, da de som regel er beregnet til at arbejde med en udvendig stak i RAM-lageret.

Hvad er det særlige ved RAM-lageret?

Jo, navnet står for en forkortelse af *Random Access Memory* - altså et lager,

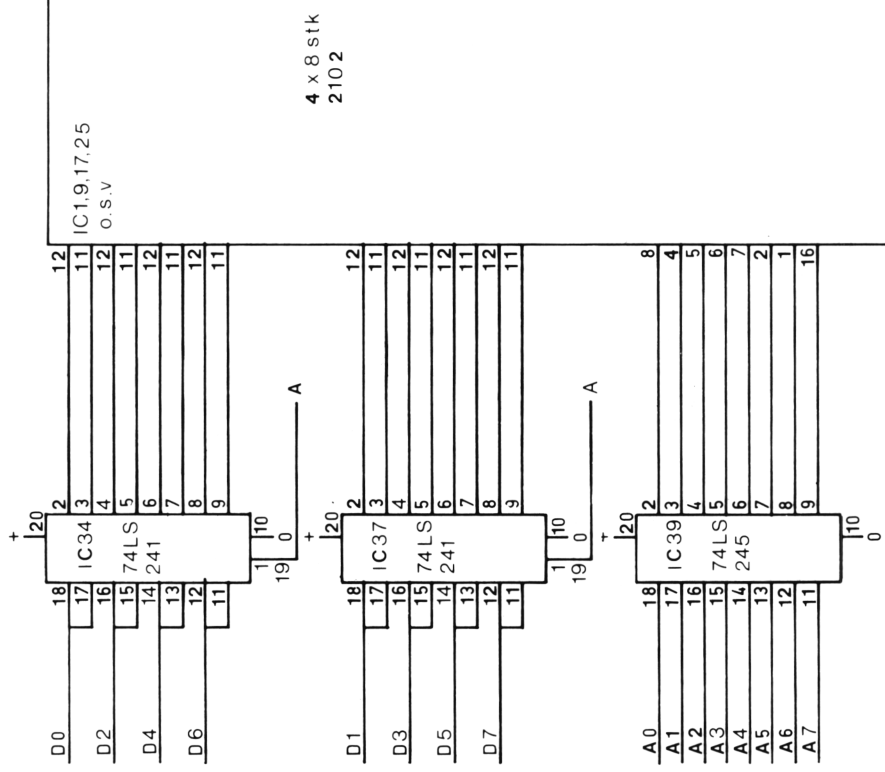


hvor de enkelte pladser kan benyttes i vilkårlig rækkefølge. Dette kan også gøres med ROM-lageret, men ikke med lagertyper som bånd og disk. En anden vigtig egenskab er, at RAM-lageret normalt arbejder med samme hastighed, som CPU'en, hvorfor den maksimale arbejds-hastighed opnås med denne type lager. Vigtigst er det dog nok, at CPU'en kan ændre indholdet af enhver af hukommelsespositionerne.

4K BYTE RAM

Til vor Z-80 datamat vil der blive fremstillet forskellige typer af RAM-kort, og det, vi denne gang skal beskrive, er det billigste. Det indeholder 4K ord og er udført med halvlederkredsen 2102. Det er en kreds, som egentlig er lidt forældet og som næppe benyttes mere i professionel regi, men den er både billig og let tilgængelig - 2 egenskaber, som ikke bør overses.

2102 er organiseret som 1024 x 1 bit. Der skal altså bruges 8 kredse til 1 ord på 8 bit.

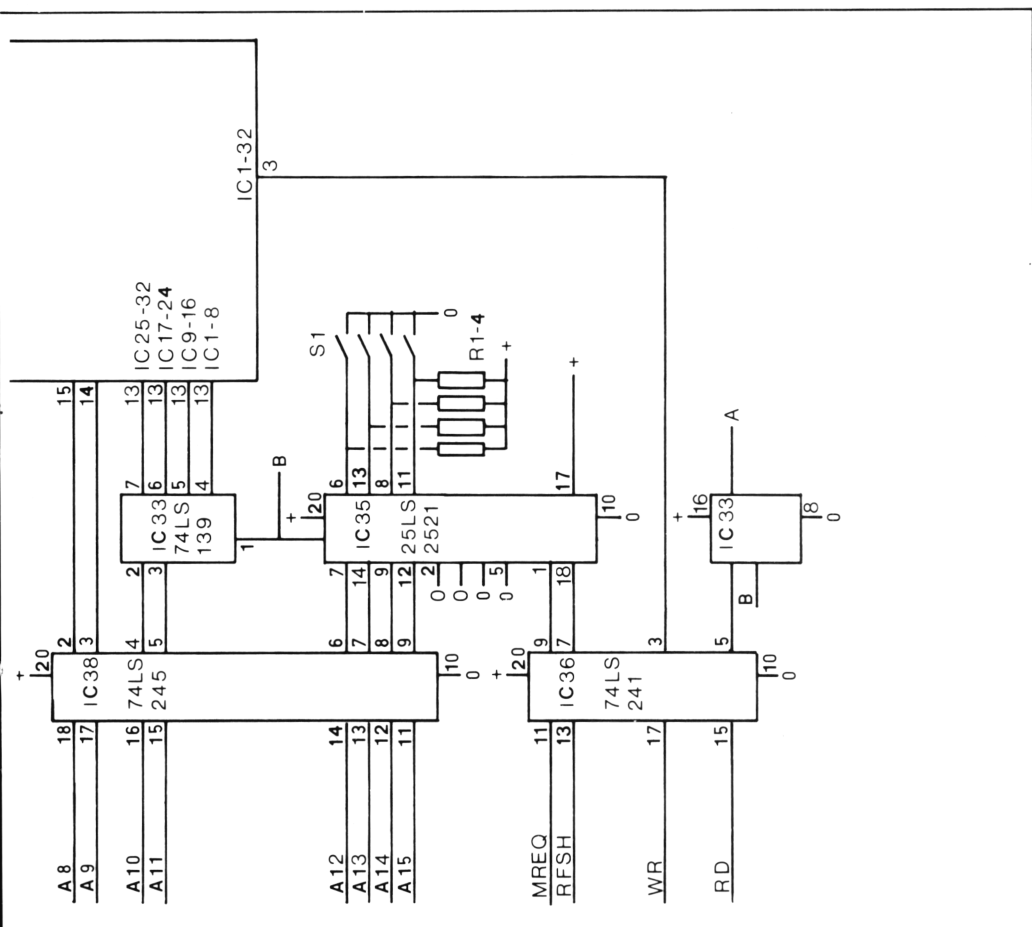


Prisen på disse kredse afhænger meget af, hvor de indkøbes. De amerikanske datamatblade bringer mange tilbud med priser helt ned til 8 kr. stk. Vi har én gang prøvet at købe kredse på den måde med skuffende resultat. De kredse, som ankom, var flere år gamle - og brugte! Herhjemme koster fabriksnye kredse i små stykantal omkring kr. 20,- pr. stk., men denne pris kan dog nedbringes væsentligt ved gruppekøb.

DIAGRAMMET

Som netop nævnt er 2102 organiseret som 1024 x 1 bit. Dette kræver 10 a-

dresselinier til udpegning af én bestemt position. Disse linier føres direkte gennem 74LS245 ud til alle kortets 32 hukommelseskredse (Husk, det er et 4K ord kort). Adresselinierne A10 og A11 bestemmer, hvilken af de 4 rækker kredse, der i et givent øjeblik skal benyttes. Dette sker ved, at linierne - efter at have passeret driveren - tilføres den ene halvdel af en 74LS139. Dette er en 2-4 linier dekoder. Hver af denne halvdels 4 udgange føres til CE (Chip Enable) på en tilsvarende række hukommelseskredse. De 4 øverste adresselinier bruges - som ved ROM-kortet - til at afgøre, hvornår



CPU'en ønsker et givet kort anvendt. Dette sker ved, at de 4 adresselinier efter driveren tilføres også denne kreds, og det er disses stilling, der afgør kortets adresse, idet komparatoren giver besked, når adresselinierne og afbryderne er identiske.

Der er dog yderligere 2 betingelser, som skal opfyldes. CPU'en viser, at den ønsker hukommelsen anvendt ved at sætte linien MREQ i logisk 0. Der er dog to mulige situationer, hvor hukommelsen ønskes benyttet, nemlig dels ved normal hukommelsescyklus, og dels ved en opfriskningscyklus, som benyttes ved dy-

Diagram til RAM-kortet. Til trods for det enkle layout vises et fuldt 4K byte RAM.

namiske RAM'er. Det er kun i første tilfælde, at dette kort skal anvendes, og denne situation kendetegnes ved, at linierne MREQ som nævnt er i logisk 0, men at linien RFSH er i logisk 1. Derfor tilføres disse linier også komparatoren, og de sammenlignes med faste spændinger.

Når alle betingelser er opfyldt, går komparatorens udgang (ben 19) i logisk 0. Dette signal tilføres den tidligere nævnte 2-4 linier dekode og sørger for, at den



kun kan pege på en række på det pågældende kort, hvis kortet allerede er udpeget.

DATALINIERNE

Datalinierne giver lidt problemer. De fleste kredse, der idag anvendes i datasammenhæng, er udstyret med bidirektionel databus, og derfor er vor datamat også forsynet med dette. 2102 er imidlertid fra før den tid, hvor dette blev almindeligt, og den er udstyret med separate dataind- og udgange.

Derfor er der til datalinierne benyttet en anden type drivere end den sædvanlige, nemlig 74LS241. Den er i familie med 74LS245, men er ikke bidirektionel.

Hver kreds indeholder 8 drivere, men disse er på „ydresiden” forbundet sammen 2 og 2, så den ene driver indad og den anden udad mod kortet.

I normalstilling, d.v.s. når kortet ikke er udpeget, er driverne, som peger indad på kortet, aktive.

Driverne skal derfor ikke omstilles, hvis CPU'en ønsker at udføre en „skrive”-cyklus, men kan nøjes med at sætte WR-linien i logisk 0, når først data og adresse er udsendt. Informationen bliver så

Susanne Sønderstrup - én af forfatterne til denne artikel - under afprøvning af Z-80 datamaten.

gemt i de kredse, hvor CE er i logisk 0. En „læs”-cyklus er lidt mere indviklet, da kun ét kort ad gangen må sende information ud på bussen.

Der sker derfor det, at RD-signalet efter driveren tilføres den anden halvdel af den tidligere nævnte 2-4 linier dekode. Hertil føres også udgangen fra komparatoren. Når begge disse linier er i logisk 0, betyder det, at CPU'en ønsker en „læs”-cyklus, samt at kortet er udpeget. Og datadrivene omstilles af en udgang fra dekoderen.

KONSTRUKTION

Når printkortet er undersøgt for evt. fejl og godkendt, monteres de robuste komponenter, d.v.s. modstande og kondensatorer samt evt. fatninger for IC'erne.

Herefter monteres DIL-afbryderen. Sørg for, at den vender således, at afbryderen er lukket, når siden, der vender mod komponenttrykkets 0, er nedtrykket.

Til sidst monteres de integrerede kredse. Også på dette kort er nummereringen

lidt usædvanlig, hvilket er gjort for at lette isætningen af hukommelseskredse. IC1-8 er første række, altså med adressen A10, A11 = 0,0. Derefter følger IC9-16 for A10, A11 = 1,0,0.s.v.

Kortet er beregnet til 4K byte, altså 32 hukommelseskredse, men datamaten kan nøjes med at arbejde med 1K = 8 kredse. Resten kan således evt. isættes på et senere tidspunkt.

AFPRØVNINGEN

Afprøvningen kan fortsat udføres på forskellig måde - afhængig af adgangen til måleinstrumenter.

Det letteste er at koble CPU-, ROM- og RAM-kortene sammen og anbringe en ROM med kendt indhold på ROM-kortet. Husk lige, at ROM- og RAM-kortet skal have forskellige adresser!

CPU'en resettes og bringes i første „memory fetch“. Her kontrolleres, at de data, som ROM'en udsender, når frem til de rigtige RAM-kredse. Data-driverne virker jo i den retning.

Komponenttrykket er udført, så det letter denne operation, idet det viser, hvilke kredse, som indeholder hvilke databit. Desuden kontrolleres, at alle adresse linier overalt er i logisk 0.

CPU'en køres frem til næste „memory fetch“, og målingerne gentages.

Således fortsættes, indtil man er sikker på, at alle linier arbejder frit.

Herefter køres CPU'en frem til et punkt i programmet, hvor den har brug for RAM-kortet. Her kontrolleres, at kom-

paratoren arbejder: Undersøg, om dens ben 19 skifter til logisk 0. Desuden kontrolleres, at den af dekoderens udgange, som A10 og A11 anviser, også skifter til 0, og at dette 0 når ud til alle 8 hukommelseskredse. Dette bør gentages for alle 4 kombinationer af A10 og A11.

WR-signalet undersøges ved sammenligning af CPU'ens ben 22 med hukommelseskredsenes ben 3. De skal til enhver tid have samme logiske niveau.

Til sidst undersøges RD-signalet. CPU'en køres frem til et sted i programmet, hvor der sker en læsning fra RAM-lageret, og det kontrolleres, at kortet udpeges, og at datadrivere skifter stilling.

Dette viser sig ved, at deres ben 1 og 19 går fra logisk 1 til logisk 0. Desuden bør det undersøges, om der står det samme på begge sider af driverne, og at denne information når frem til CPU'en.

Hermed skulle kortet være afprøvet.

TEST-ROM

Vi har nævnt i forbindelse med hukommelseskortene, at man bør benytte en ROM med kendt indhold.

Det er selvfølgelig bedst, hvis en ROM med et specielt test-program kan anvendes, da det vil skabe en række situationer, som fortæller mest muligt om de kort, som er til afprøvning.

Det er dog ikke absolut nødvendigt med netop sådan en ROM, da maskinen arbejder helt normalt, så alene det, at den foretager sig det, som man ud fra kendskabet til programmet forventer, er en tilstrækkelig indikation af, at alt er i orden.

AFKOBLINGER

Vi har i indledningen omtalt, at det er en betingelse for opretholdelse af informationen i et RAM-lager, at dette er konstant strømforsynet.

Dette er et absolut krav, hvilket skal forstås på den måde, at selv en kortvarig reduktion i spændingsforsyningen kan være fatal.

Dette kan meget let blive noget af et problem, for de integrerede kredses strømbehov er stærkt varierende og kan

STYKLISTE TIL RAM-KORT

32 stk.	IC 1-32	2102, 450 nS
1 stk.	IC 33	74LS139
3 stk.	IC 34, 36, 37	74LS241
1 stk.	IC 35	25LS2521
2 stk.	IC 38, 39	74LS245
4 stk.	R 1-4	2,2 kohm, 1/8 W
1 stk.	S 1	4 pol DIL-afbr.
16 stk.	C 1-16	10 µF/10V
20 stk.	C 17-36	0,1 µF
1 stk.	Printkort	77312

skifte under kørslen. Et pludseligt behov for større strøm kan som følge af selv-induktionen i strømforsyningslederne få spændingen til at foretage et kortvarigt dyk, indtil spændingen kan nå at blive stabiliseret.

Denne stabilisering tager dog for lang tid i forhold til RAM-kredsens behov for konstant spænding, og der må derfor tages særlige forholdsregler på udsatte steder i datamaten.

Og RAM-kortene er sådan et udsat sted. De forskellige fabrikanter af RAM-IC'er giver derfor i deres datablade gode råd om, hvorledes dette problem løses ved hjælp af afkoblingskondensatorer.

En afkoblingskondensator er at opfatte som et element, som med sin oplagrede energi skal sørge for, at kredsen til enhver tid får den spænding og strøm, der er behov for.

Desværre er spændingssvingene ofte meget hurtige - frekvenser på op imod 50 MHz er ikke ualmindelige - og en kondensator, som kan reagere ved disse frekvenser, er desværre ikke i stand til at indeholde megen opsparet energi.

Det er derfor almindeligt, at RAM-kortet forsynes med 2 forskellige typer kondensatorer. Den ene er god til at klare hurtige spændingssving, mens den anden reagerer lidt langsommere, men med større mængde energi.

S OPBYGNING AF HUKOMMELSEN

Der er flere, som har spurgt, hvorledes man mest praktisk og økonomisk opbygger sit lager i amatør-datamaten.

Vi mener, at en god begyndelse inkluderer 512 byte ROM og 1K byte RAM. Dette er en billig begyndelse, som alligevel er så effektiv, at man kan have en lille monitor i sin ROM og samtidig have mulighed for at lave f.eks. mindre spil med maskinen og dermed lære noget om datamater.

Hvis lysten og økonomien tillader det, bør derefter RAM-lageret udbygges til 4K byte - når dette er sket, er der mulighed for at arbejde i BASIC.

Næste skridt må nok være en bedre monitor, så ROM'en bør udbygges til 2 K

byte.

De største - og bedste - BASIC'er vil kræve et RAM-lager på 16 K byte, og en yderligere udbygning vil først og fremmest betyde plads til større og flere programmer.

Den her skitserede opbygning og udvikling giver en maskine, som til enhver tid er særdeles fleksibel, og som hurtigt kan skifte fra én opgave til en anden.

Eneste virkelige ulempe herved er, at det program, der skal afvikles, altid skal indlæses i datamaten først. Ved større programmer og langsomme ydre enheder kan dette involvere irriterende lang tid.

Mange, som ofte ønsker at arbejde med ét bestemt program, vælger derfor at nedlægge dette i en ROM, så det altid er klar til brug. Dette kan kun lade sig gøre med programmer, som er specielt udviklet hertil, og antallet af sådanne er ikke overvældende stort, men der findes f.eks. en udmærket 5K BASIC med flydende punkt matematik.

Dette program er yderligere udstyret med faciliteter, der gør det velegnet til styringsformål - det være sig elektriske tog eller værktøjsmaskiner.

Fordelen ved styring via BASIC er naturligvis den hurtige korrektion - man må så acceptere, at programafviklingen ager relativt længere tid. MP/SS

Selvbyggere, som er interesseret i at være medbyggere på det interessante Z-80 projekt, bedes skrive til Mogens Pelle, Birkhøj Terrasserne 416 C, 3520 Farum.

Det samme gælder evt. spørgsmål vedrørende denne datamat.

Mogens Pelle har bedt os om at anmode alle om at benytte sig af skriftlige henvendelser, så hans private telefon ikke blokeres.

BREVKORT

Porto
100
øre

Husk afsender

Til:

Telepress ApS

Greve Strandvej 42
2670 Greve Strand

▲
KLIP LANGS STREGERNE HELT TIL BLADETS KANT
▼

BREVKORT

Porto
100
øre

Husk afsender

Til:

Telepress ApS

Greve Strandvej 42
2670 Greve Strand

KLUBBINATION

Hvis du og/eller dine bekendte vil i forbindelse med andre datamat-amatører, skal I blot udfylde denne kupon og sende den til os – så bringer vi jeres budskab i kommende numre af HFD.

Klubber åbne for medlemmer / interesserede i kontakt med andre klubber:

Klubbens navn:	_____
Adresse:	_____
Postnr.:	_____
By:	_____
Indmeldelsesgebyr, kr.:	_____
Kontingent pr. måned, kr.:	_____
Klubbens udstyr:	_____
_____	_____
Speciel interesse:	_____
Nuværende antal medlemmer:	_____
<i>Datamat-amatører, som er interesseret i at blive kontaktet af klubber og andre datamat-amatører:</i>	
Navn:	_____
Adresse:	_____
Postnr.:	_____
By:	_____
Nuværende udstyr:	_____
Speciel interesse:	_____
Ønsker helst kontakt med:	_____

Giv dette brevkort til en ven, som gerne selv vil have sit eget eksemplar af Håndbog for Datamat-amatører, eller brug det til meddelelser angående flytning.

HFD udkommer med 11 numre om året, og et abonnement kan tegnes når som helst og med start fra valgfri måned. Det anbefales dog, at der tegnes abonnement fra bladets start, da den specielle opbygning bedst udnyttes, hvis alle numre haves. 1. nummer udsendt er nr. 9/1977. Abonnement koster idag kr. 100,- for en årgang og inkluderer 11 numre af HFD, 1 praktisk og solidt ringbind til en hel årgang, porto og moms.

Undertegnede stiller herved et abonnement på HFD for 1 år i henhold til ovenstående for kr. 100,-. Jeg ønsker at abonnementet starter med nr.

Beløbet, kr. 100,-, vedlægges i check.

I bedes fremsende girokort.

Undertegnede ønsker at meddele adresseforandring på mit abonnement på HFD.

Navn:

Gl. gadeadresse:

Gl. postnr. og by:

Ny gadeadresse:

Nyt postnr. og by:

Ovenstående adresseændring træder i kraft d.

Andet:

**KLIP LANGS DE FULDT OPTRUKNE STREGER
SENDES SOM BREVKORT, HUSK PORTO**