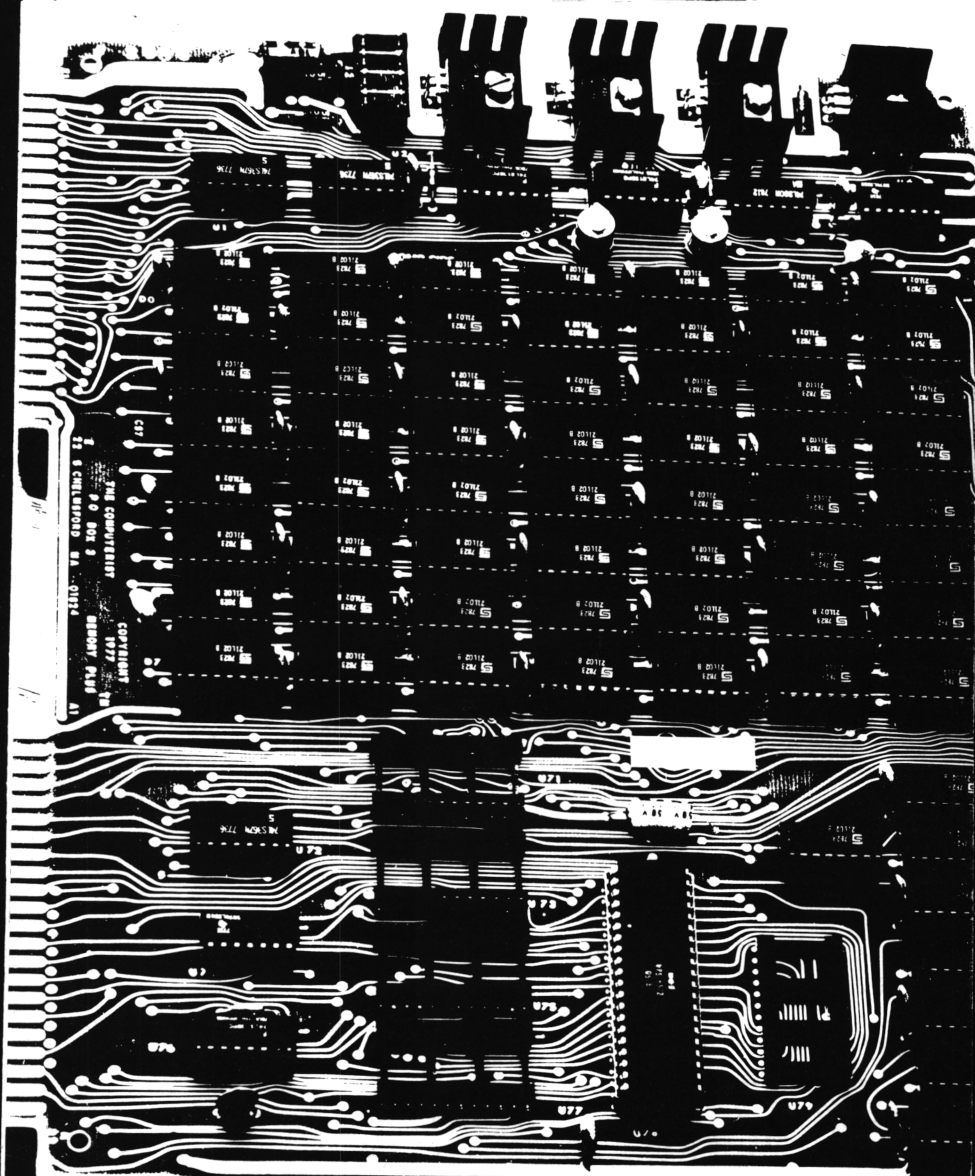


# 9B Håndbog for datamat-amatorer

1978



THE COMPUTER  
9080 2  
RESEARCH / 1978  
33 S. CHAMPLAIN ST. MA. 02148

## INDHOLDSFORTEGNELSE

### ALMENT OM PROGRAMMERBARE

<b>MASKINER</b>	
Sådan begyndte det	A 1
Den forventede udvikling	A 7
Talsystemer	A 21
Binær matematik	A 28
Logiske operationer	A 31

### BIBLIOTEKET – PROGRAMMER

HP-25, Delefilter	B 1
HP-25, Gæt et tal	B 3
HP-25, Likviditet	B 5
HP-25, Mastermind	B 11
KIM-1, Multimaze	B 13
IMSAI 8080, RAM-test	B 17
KIM-1, FUT-FUT	B 19
TI-59, Løn og skat	B 21
TI-59, Kørselsregnskab	B 27
BASIC, Lån, afdrag og renter	B 31
BASIC, Reaktionsid	B 37
TI-58/59, Omregn. mell. talsyst.	B 41
TI-58/59, Mastermind	B 43

### CPU-ARKITEKTUR

CPU-Arkitektur	C 1
Motorola M6800	C 5
Intel 8080	C 7
SC/MP	C 9
Signetics 2650	C 11
Intel 8048	C 13
Zilog Z-80	C 15

### DATAMAT-LITTERATUR

Elementært om Microdatorer	D 1
The first Book of KIM	D 2

### KLUBINFORMATION

Datamatklubber	K 1
Datamatamatører	K 11
Seminar, april 1978	K 17

**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, Vi by 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.

ISSN 0105-581X

### LOMMEREGNERE

TI-Programmer	L 1
HP-25/25C	L 3
TI-59/PC-100	L 5

### MIKRODATAMATER

Valg af mikrodatamat	M 1
Datamatkapacitet	M 5
KIM-1	M 11
KIM-1, Kontakter og dioder	M 15
Motorola M6800	M 19
TK-80, begyndersæt	M 25
Imsai 8048 CC	M 30
Nye datamater, april 1978	M 33
PET 2001	M 37
TELMAC 1800	M 41
KIM-1, udvidet udgave	M 43

### PROGRAMMERINGSTEKNIK

Lær programmering	P 1
Subrutiner	P 49
Splitning af subrutiner	P 51
BASIC	P 55
BASIC, fortsat	P 65

### SELVBYGGERPROJECTER

IMSAI 8080	S 1
Z-80 mikrodatamat	S 11
77-68 selvbyggerdatamat	S 51
Z-80, fortsat	S 57
Z-80, fortsat	S 63

### YDRE ENHEDER

TV-skriver	Y 11
Pocket TTY	Y 13
TV-modulator	Y 16
Digital multiplexer	Y 19
ACT-IV skærmtterminal	Y 25

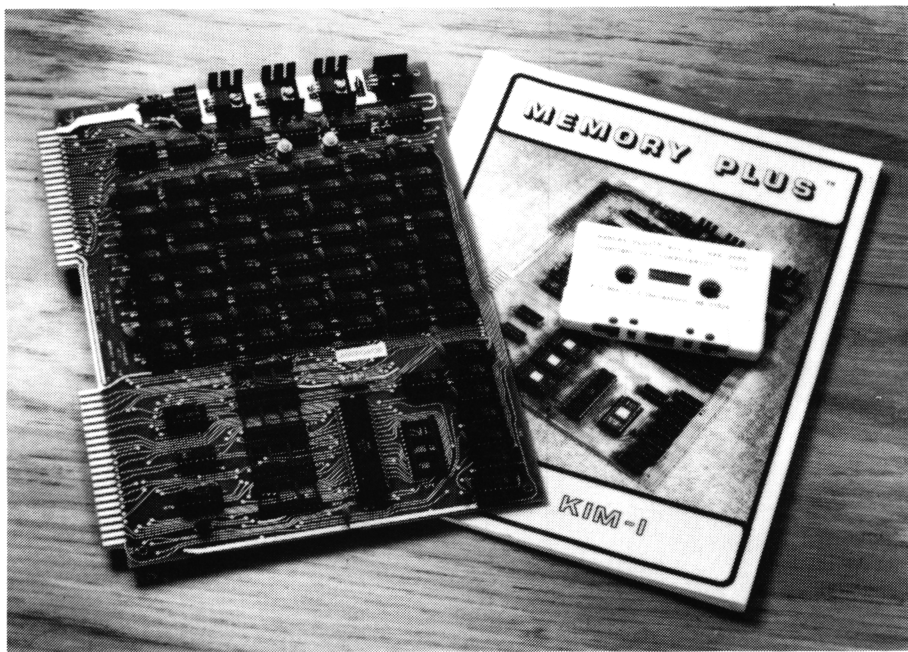
*Beklageligvis var nummernissen på spil i forrige nummer af HFD.*

*Ikke blot forsiden var gal – men én af artiklerne var ligeledes fejlagtigt nummereret.*

*Tag venligst forrige nummer frem – find artiklen af Mogens Pelle om Z-80 projektet – dette er nummereret S-51 til S-56. Og ret tallene til S-57 fortløbende til S-62 incl. TAK*



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



## MEMORY PLUS TIL KIM-1

Memory Plus er en yderst kompakt og effektiv udvidelse – specielt konstrueret til KIM-1.

### MEMORY PLUS indeholder

- 8K bytes low power statisk RAM
- Dekodning og sokler til 8K bytes EPROM
- Speciel sokkel til programmering af EPROM
- 1 stk. R6522 VIA  
(Versalite Interface Adapter)

### MEMORY PLUS leveres:

- med omfattende teknisk manual
- med 3 stk. kantconnectorer
- med 1 stk. kassettebånd  
med test-rutiner
- færdigsamlet og testet

Memory Plus koster **kr. 2.190.-** excl. moms.

Ønsker man at montere Memory Plus i KIM-1 kabinettet, behøver man blot en ny underdel, som mod kr. 85,- excl. moms kan leveres med.

**MICROWOR** ApS

Tindbjergvej 16 • 8600 Silkeborg • ☎ 06-83 60 08



Rockwell  
International

## 16/32K RAM—KORT

Endnu et RAM-kort! Hvorfor det? Der er jo allerede beskrevet et til denne data-mat.

Jo, en hobby, der startede med, at de fleste programmer havde en størrelse på 50-100 bytes og ikke nogen egentlig praktisk værdi, har efterhånden ført til programmer på 10-20 KiloByte eller mere - og vel at mærke programmer, som kan udføre et fornuftigt arbejde.

Vigtigt for denne udvikling har selvfølgelig været de stærkt faldende priser på integrerede kredse, men også det, at man på mange universiteter og læreanstalter har fremstillet gode programmer, som kan erhverves for en ganske lav pris eller gratis, har haft væsentlig betydning.

Disse store hukommelser kan selvfølgelig opbygges med det tidligere beskrevne RAM-kort, men der skal mange til. Det er derfor ønskeligt at kunne få flere K-Byte på hvert kort.

Nu er det jo ikke en udvikling, som er speciel for amatørerne. Også den professionelle verden bruger større og større hukommelser og ønsker, at disse fylder mindre og mindre. Udviklingen har derfor længe gået i retning af, at de enkelte kredse indeholder stadig flere bit.

## STATISK/DYNAMISK

Nu er dette dog lettere sagt end gjort; for selvom de transistorer, der anvendes i en integreret kreds, er utroligt små, er det begrænset, hvor mange, der er plads til. Det bliver derfor af afgørende betydning, hvor mange, der skal anvendes, for at gemme en bit.

Her kommer vi ind på begreberne statisk contra dynamisk.

Den tidligere anvendte hukommelseskreds, 2102 er statisk, det vil sige, at den enkelte hukommelsescelle er opbygget som en flip-flop, bestående af 6 transistorer, der kan stilles i en bestemt stilling og derefter holder denne uden yderligere forholdsregler, blot cellen tilføres strøm.

Det er dog ikke nok med disse 6 transistorer i den enkelte celle, for en kreds

indeholder et stort antal hukommelsesceller, og disse skal på skift, styret af adressen, tilkobles de samme ud- og indgangsbæn. Hertil kræves mindst en ekstra transistor, men i visse kredstyper bruges der flere. Yderligere kommer de transistorer, der skal anvendes for at dekode adressen. Alt i alt et stort forbrug af transistorer pr. opbevaret bit.

Med så mange transistorer er det klart, at det får væsentlig betydning, hvor store de er, og dette er en væsentlig grund til de forskellige fabrikationsmetoder, som ses anvendt - for slet ikke at tale om sidste mode, hvor transistorerne sættes lidt på højkant.

Mere radikalt ville det jo være, om nogle af disse transistorer kunne undværes; Og det er muligt. Principielt kan de alle undværes og erstattes af en kondensator. Som bekendt, kan en sådan gemme en elektrisk ladning, og man kan derfor sige: kondensator med ladning = 1, kondensator uden ladning = 0 eller omvendt. Her kommer dog et problem. Kondensatorer aflader sig selv, og i løbet af kort tid, er vor information forsvundet.

Dette kan ikke forhindres, og man må derfor i stedet genopfriske informationen med passende mellemrum.

Disse principper ligger til grund for den dynamiske hukommelseskreds, som i de sidste år er blevet anvendt i stigende grad, fordi et enkelt hus med 16 ben kan rumme op til 16K bit.

Det der med genopfriskningen (også kaldet Refresh) lyder måske lidt ubehageligt, og for de fleste CPU'er, er det også besværligt at have med at gøre, fordi det skal ske på en bestemt måde, kontrolleret af et ydre kredsløb, medens CPU'en står stille. Det komplicerer altså kredsløbene og nedsætter datamatens hastighed.

Her er vi dog igen heldige med vort valg af CPU. Z-80 har nemlig disse kredsløb indbygget og på en sådan måde, at den kan arbejde med fuld hastighed - også med denne type hukommelse.

For lidt siden erklærede vi, at transistorerne helt kunne undværes. Det er

imidlertid også kun i princippet; for i praksis skal også her de enkelte hukommelseceller kobles til fælles ud- og indgangsbæn. Det kan klares med en enkelt transistor, som altså ikke gemmer informationen, men kun fungerer som afbryder.

#### 4116

Den anvendte kreds (4116 eller 2116) har en temmelig kompliceret opbygning, som det nok er klogt at se nærmere på et øjeblik. Kredsen er indvendigt opbygget som 2 matricer, hver med 64 rækker og 128 søjler. Her må det nok indskydes, at en matrice er en firkant, dannet af et stort antal punkter, der ligger på linie vandret og lodret. De vandrette linier kaldes rækker og de lodrette søjler. Hvis du på et stykke kvadreret papir sætter en blyantsprik i hvert krydspunkt, danner disse punkter en matrice. I kredsen ligger der altså i hvert krydsningspunkt en hukommelsecelle bestående af en kondensator og en transistor. Alle disse celler er forbundet sammen vandret og lodret, og det på så snedig vis, at, hvis man sætter spænding på en vandret og en lodret linie, bliver den kondensator, der ligger i disse 2 liniers krydsningspunkt, gennem sin tilhørende transistor, forbundet til en tredje linie, kaldet "sense"-linien. Der er en sådan linie for hver søjle, og alle disse linier fører hen til hver sin "sense-forstærker". Normalt taler vi jo ikke om forstærkere i digitale kredsløb, men her er ordet på sin plads. Kondensatorer i den enkelte celle er nemlig meget lille (ca. 0,03 pF) og kan kun indeholde en meget lille ladning. Spændingen over kondensatoren falder derfor hurtigt, og for at tingene ikke skal blive alt for kritiske, må en opladet kondensator kunne skelnes fra en uladet i et rimeligt stykke tid. I dette tilfælde kan forstærkeren skelne, blot der er 200 mV over kondensatoren.

Hvor lang tid går der så, før de 12V, som kondensatorer bliver opladet til, når informationen bliver indlæst, er faldet til 200mV? Ikke ret længe. For 4116 anbefales det, at hver celle bliver

genopfrisket hver 2 mS - altså 500 gange i sekundet.

Selve denne "refresh" kan foregå efter forskellige principper, som fastlægges, når man laver en opstilling, men det der sker i kredsen, er under alle omstændigheder temmelig indviklet.

Princippet er dog enkelt nok, idet der simpelthen sker en udlæsning af de enkelte celler, som derefter genindlæses på de gamle pladser. Da der jo er mere end 16000 celler, der hver skal genopfriskes 500 gange i sekundet, bliver det til mere end 8 millioner genopfriskninger pr. sekund.

Så mange refresh ville forhindre datamaten i at bestille andet, og det er derfor meget heldigt at kredsen er udstyret med 128 forstærkere. Herved bliver det nemlig muligt at genopfriske 128 bit ad gangen.

Det næste problem er så at holde styr på, hvilken gruppe, der næste gang skal genopfriskes. Dette klares af et kredsløb kaldet "refresh controller", og her er det, vi slipper billigt med Z-80, idet denne enhed er indbygget i CPU'en, og fungerer helt automatisk. I virkeligheden består den blot af en 7-bit tæller, hvis udgange tilkobles adresselinierne på tidspunkter, hvor CPU'en ikke har brug for disse, samt en kontroludgang kaldet RFSH. Desuden er linien MREQ aktiv. Ram-kortet ved så, at når disse 2 linier er aktive, skal den gruppe, hvis nummer står i de 7 nederste adresselinier genopfriskes.

Til udpegning af en plads blandt 16000 skal bruges 14 adressebit. Det kan derfor virke lidt overraskende, når man ser kredsen, at den kun er udstyret med 16 ben. 20- eller 22 ben ville være mere naturligt. Dette kan lade sig gøre fordi adressen multiplexes, d.v.s. sendes ind af 2 gange. Først sendes de nederste 7 adressebit, og et øjeblik efter de øverste. For at vide, hvornår hvad kommer, er kredsen udstyret med 2 specielle kontrolsignaler, RAS og CAS.

Når adressen er stabil, sendes først de nederste 7 bit ind i kredsen, og RAS bliver aktiv. Herved udpeges den ene ræk-

ke, og lidt efter skiftes til de øverste 7 adressebit, CAS bliver aktiv, og en af søjlerne bliver udpeget.

Den ønskede hukommelsesplade findes så i krydsningspunktet mellem de 2 aktive linier.

De 2 kontrolsignaler er flanketriggede, og generer ikke hinanden.

## DIAGRAMMET

Vi har flere gange erklæret, at diagrammerne til de forskellige kort er meget simple. Det er derfor med nogen sorg, vi føler, at dette kort er temmelig indviklet - og lad det være sagt med det samme: dette kort er ikke egnet for begyndere. Det skyldes først og fremmest, at hvis kortet ikke virker straks efter, der er samlet, er det absolut nødvendigt at have adgang til et godt oscilloskop. CPU'en skal nemlig arbejde med mindst halv hastighed for at genopfriskningen sker korrekt.

4116 er udstyret med separate dataind- og -udgange. De 8 datalinier føres derfor gennem kredsene IC17 og IC20, hvor de 2 retninger adskilles, og videre til kredsene. A0-A13 er mere komplicerede. Som du husker fra foregående afsnit, er adressebene multiplexet på 4116. Det er

de ikke i vores bus, og det må derfor ske, men først går alle adresselinierne gennem IC28 og IC29. Herefter føres IC0-3 og IC7-10 til IC25 og IC4-7 og IC11-13 til IC26.

Disse 2 kredse er såkaldte "Quad 2 to 1 line data multiplexers". Oversat til dansk betyder det, at hver kreds indeholder 4 ens kredsløb. Hver af disse har 2 indgange og en udgang. En af indgangene er forbundet til udgangen. Hvilken afgøres af stillingen på ben 1.


Vi kan altså, med disse 2 kredse, først sende de 7 laveste adressebit videre til 4116 og derefter, ved at skifte stilling på ben 1, sende de øverste 7 bit.

Af drivere har vi til sidst IC27, hvorigennem de 4 kontrollinier føres.

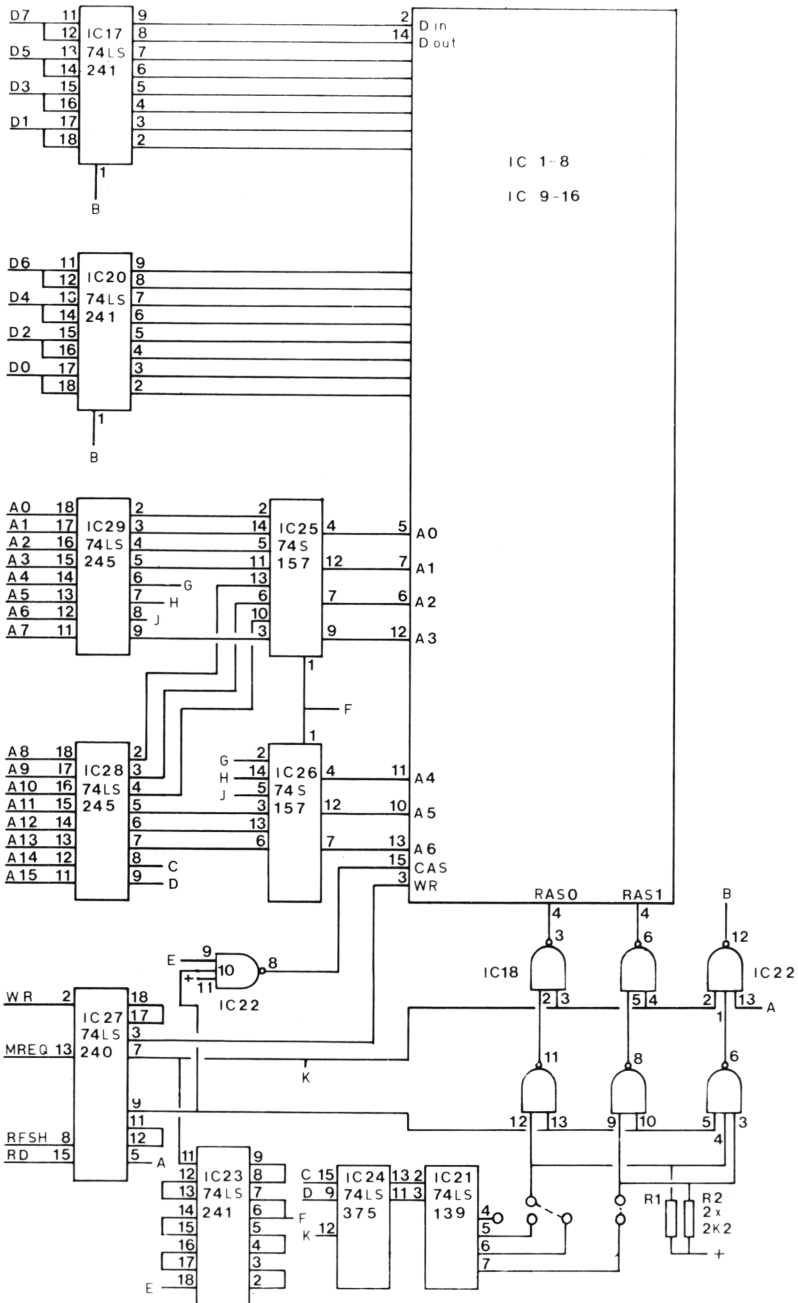
WR sendes 2 gange gennem IC27 for at "vende" rigtigt og føres videre til hukommelseskredsene. RD er også simpelt. Det føres efter driveren til IC22, hvor det er med til at bestemme retningen af datadrivene.

RFSH føres 2 gange gennem IC27 og går derefter til IC18 og dels til IC22.

IC18 aktiverer RAS-signalerne til alle hukommelseskredsene på en gang, mens IC22 blokerer CAS-signalet. Dette



Husk at købe  
Populær Elektronik  
i kiosken.  
Eller send 80 Kr  
til Forlaget Tele-  
press Greve Strand-  
vej 42. 2670 Greve  
Strand, så sender  
vi 1 blad fremover  
i 11 måneder.  
Eller ring 02 90 86 00  
så klarer vi resten.



er en af de 3 måder at gennemføre Refresh på. Metoden kaldes meget naturligt: RAS-only Refresh.

MREQ føres efter driverne videre til IC18 og IC23.

IC23 er en forsinkelseslinie. Når ben 11 går mod 1 vil denne flanke efterhånden løbe gennem hele kredsen. Efter 3 trin og ca. 40 nS vil flanken nå ben 7. Herfra føres det videre op til ben 1 på IC25 og 26 og omstiller disse til at videresende de overste 7 adressebit.

Efter yderligere 5 trin når flanken ben 18, hvorfra den sendes op til IC22. Her er RFSH på den anden indgang i 1 og udgangen går derfor i 0 og bruges som CAS.

Til sidst er der så adresseudpegningen. Denne er lidt anderledes end på de øvrige kort, hvilket skyldes ønsket om at dette kort kun skal fylde den plads i adresseringsområdet, som de indsatte kredse berettiger til. Har du kun indsat 8 hukommelseskredse, kan du altså indsatte f.eks. 44K ROM og RAM, lavet med de tidligere beskrevne kort.

Udpegningen sker ved at A14 og A15 tilføres IC24, som er en firedobbelt latch, hvoraf kun halvdelen bruges, og videre til IC21, som er en 2-4 linier dekode. Udgangene fra denne føres til 4 loddeøer. Fra 2 af disse kan man med lus sende signalerne videre til IC18 og herved få kredsen udpeget.

Ved levering af kortet er lusene lagt således, at kortet fylder de øverste 32K i adresseringsområdet. Ønsker man at ændre herpå, skal disse forbindelser overskæres på kortets loddese ved mærkerne A--A.

Dette var diagrammet for kortets hukommelsesdel. Strømforsyningen vil vi vende tilbage til om lidt. Først skal vi lige se, hvordan kortet virker under normalt arbejde.

## VIRKEMÅDE

Hertil skal vi bruge fig. 1 side S16, idet vi tænker os at starte med en instruction fetch.

Som fig. 1 viser, kommer først adressen. Denne går straks gennem driverne, og de

7 nederste bit går også gennem multiplexerne frem til hukommelseskredse. A14 og A15 når frem til IC24, men først en 1/2 klokpuls senere, når MREQ bliver aktiv, åbnes denne, og de 2 bit føres videre til dekodeerne. Denne sænker en af sine udgange og dette signal føres videre til en af gatene i IC18. Da RFSH på gatens anden indgang er i 1, bliver denne gates udgang 1. Dette går videre til næste gate. Her er MREQ også i 1, og gatens udgang går derfor til 0. Herved er RAS-signalet til den ønskede række dannet.

Idet MREQ bliver aktiv, starter også forsinkelseskæden og starter de tidligere omtalte forløb, som slutter med at CAS-signalet dannes.

Samtidig med MREQ bliver også RD aktiv (se fig. 1). Begge signalerne tilføres IC22 og omstiller datadrivnerne.

Lidt efter kommer de ønskede data fra hukommelseskredse og sendes til CPU'en.

Når læsningen er overstået, bliver både RD og MREQ inaktive, og samtidig skifter adressebussen, således at de 7 nederste linier indeholder refresh-adressen.

Denne trænger direkte frem til hukommelseskredse. Samtidig hermed bliver RFSH aktiv og får de 2 nederste gates i IC18 til at åbne, d.v.s. deres udgange går i 1. Et øjeblik efter bliver MREQ aktiv (se fig. 1) og går til 1 på kortet. Herved bliver begge indgangene på de 2 øverste gates i IC18 logisk 1 og deres udgange logisk 0.

Vi har herved fået dannet RAS-signal til alle hukommelseskredse på en gang.

Når den til refresh udmålte tid er udløbet, forsvinder først MREQ. Herved fjernes RAS-signalet, og desuden læses de 2 øverste adressebit fast i IC24.

Denne læsning er nødvendig, fordi RFSH forsvinder samtidig med at adressebussen skifter til at indeholde en ny adresse. Da det ikke kan garanteres, at RFSH er væk når adressen skifter, kunne udeladelse af latches medføre nogle spidser på adresselinierne, som kunne ødelægge informationen på 128 pladser.

Kortet er herefter klar til en ny cyklus.



**Low Power Schottky + TTL**

74LS00-1.75	7460-1.95	74LS145-5.95	LS244-15.95
LS01-1.75	LS63-11.85	LS147-16.90	LS244-11.95
LS02-1.75	LS70-2.95	LS148-12.75	LS245-26.95
LS03-1.75	LS72-2.95	LS150-9.95	LS247-11.95
LS04-1.95	LS73-2.95	LS151-6.95	LS248-11.95
LS05-1.95	LS74-2.95	LS152-6.95	LS249-11.95
LS06-2.75	LS75-3.95	LS154-9.95	LS251-9.95
LS07-1.85	LS76-2.95	LS155-6.95	LS252-9.95
LS08-1.95	LS78-3.95	LS156-6.95	LS257-9.95
LS09-1.95	LS78-4.95	LS157-6.95	LS258-9.95
LS10-1.95	LS83-3.95	LS158-1.95	LS259-8.75
LS11-1.95	LS85-6.95	LS160-11.95	LS260-2.75
LS12-2.45	LS86-2.95	LS161-11.95	LS261-11.95
LS13-2.65	LS89-2.75	LS162-12.95	LS266-3.75
LS14-5.85	LS90-3.95	LS163-8.85	LS273-9.75
LS15-2.45	LS91-3.95	LS164-8.85	LS275-66.80
LS16-2.95	LS92-3.95	LS165-12.95	LS280-2.75
LS17-2.95	LS93-3.95	LS166-13.95	LS283-9.95
LS20-1.95	LS94-5.95	LS170-13.75	LS290-9.95
LS21-1.95	LS95-5.85	LS173-9.95	LS293-9.95
LS22-1.95	LS96-5.85	LS174-9.95	LS295-9.95
LS26-2.95	LS100-9.95	LS175-9.95	LS298-11.85
LS27-3.95	LS107-3.25	LS177-12.65	LS299-15.85
LS28-2.95	LS109-3.25	LS180-9.95	LS324-11.95
LS30-1.95	LS112-3.25	LS181-13.95	LS325-11.95
LS32-1.95	LS113-3.25	LS182-9.95	LS326-11.95
LS33-1.95	LS114-3.25	LS183-23.75	LS327-11.95
LS37-2.95	LS116-9.95	LS190-11.65	LS352-11.95
LS38-2.95	LS119-16.85	LS191-11.65	LS365-6.75
LS40-1.95	LS121-3.95	LS192-11.45	LS366-6.75
LS42-4.45	LS122-3.95	LS193-11.45	LS367-6.75
LS43-5.95	LS123-5.95	LS194-11.45	LS368-6.75
LS45-6.95	LS124-11.95	LS195-12.65	LS373-23.85
LS47-9.95	LS125-3.95	LS196-12.65	LS375-6.75
LS48-11.95	LS126-3.95	LS197-12.65	LS377-11.95
LS49-11.95	LS128-4.75	LS198-12.95	LS378-12.95
LS50-2.25	LS132-5.95	LS199-14.75	LS386-2.85
LS51-1.95	LS133-4.95	LS201-9.95	LS390-23.85
LS53-1.95	LS136-3.50	LS204-14.95	LS395-13.45
LS54-1.95	LS138-5.95	LS241-16.75	LS399-14.95
LS55-1.95	LS139-5.95	LS242-15.95	LS670-14.95
	LS414-4.75		

Delay per Gate: 7400-10ns/10mW 74LS-10ns/2mW

**IC - AMPS.**

LM301n	-3,-	TBA120a	-8.75
LM318n	-22,-	TBA120s	-9.75
LM324n	-8.75	TBA231	-9,-
LM339n	-6.90	TBA520	-16,-
LF355n	-13.75	TBA550	-26,-
LF356T	-14.75	TAA550	-6.75
LF357n	-11.95	TAA630	-22,-
LM377n	-17.50	TBA800	-16,-
LM378n	-19.75	TBA810s	-18,-
LM380n	-9.95	TCA730	-24,-
LM381n	-14,-	TCA740	-24,-
LM382n	-12,-	TCA940	-19,-
LM386n	-9,-	TCA965	-16,-
LM387n	-9,-	TDA1022	-62,-
LM710t	-9,-	TDA1034	-39.75
LM741n	-2.95	TDA1037	-9.75
LF13741n	-12,-	TDA2002	-24,-
UA739n	-8,-	TDA2020	-38,-
LM748n	-4.95	SO41p	-9.75
LM1303n	-11.95	SO42p	-9.75
LM1307n	-12.95	SAS560	-9.75
LM1351n	-12.95	SAS570	-9.75
LM1458n	-6.50	UAA170	-14.75
MC1496T	-9,-	UAA180	-14.75
MC1496n	-8,-	9582	-45,-
LM1812n	-57.75	95H90	-98,-
LM1830n	-17.75	MM5316n	-42,-
LM2917n	-17.75	CT7004	-78,-
LM3900n	-8.75	ZN1040e	-110,-
LM3911n	-12,-		

**IC Socket**

8 Ben - 1.75	20 "	-3.25	
14 " - 1.95	22 "	-3.75	Wire -Wrap
16 " - 1.95	24 "	-4.25	
18 " - 1.95	28 "	-4.75	14 Ben- 4.65
16 " - 2.25	40 "	-5.50	16 " - 5.15

**EXAR IC**

	Signetics NE/IC	RCA-CA Type
XR 205-59,-	NE510-18,-	CA2111ae-17.90
XR 215-36,-	NE515-23,-	CA3021 -23.75
XR 320-12,-	NE521-17.60	CA3023 -23.75
XR L555-12,-	NE522-17.60	CA3028 -10,-
XR 1310p-10,-	NE526-36.30	CA3046 -9,-
XR 2206-39,-	NE527-14.90	CA3051 -23.75
XR 2207-29,-	NE529-17.40	CA3052 -21.75
XR 2208-44,-	NE531-11.85	CA3059 -18.75
XR 2209-12,-	NE532-6.85	CA3060e -23.75
XR 2212-37,-	NE535-9.95	CA3062 -34.75
XR 2216-54,-	NE536-19.65	CA3075 -21,-
XR 2240-29,-	NE540-16.75	CA3079 -12,-
XR 2242-10,-	NE541-16.75	CA3080e -34.75
XR 2272-10,-	NE544-16.75	CA3081 -12,-
XR 2567-23,-	NE546-11.75	CA3082 -12,-
XR 4136-10,-	NE550-9.90	CA3086 -5.95
XR 4151-23,-	NE555-3.75	CA3089e -27.50
XR 4194-32,-	NE556-8.75	CA3090e -34.75
XR 4195-12,-	NE558-12.75	CA3100e -14,-
XR 4741-14,-	NE559-12.75	CA3118t -14,-
	NE560-24.45	CA3130t -12,-
	NE565-9.75	CA3130n -12,-
	NE566-9.75	CA3140t -9.50
	NE567-10.75	CA3140n -9.50
	NE570-32.70	CA3160e -9.95
	NE571-28.70	CA3161 -13.75
	NE572-11.75	CA3162 -68.25
	NE590-13.75	CA3189e -24.75
	NE592-9.95	CA3240e -9.85
		CA3290e -24.40
		40673 -9.75
		40841 -7.75

**Motorola IC's**

MW 5009 - 78,-	
MW50240 - 83,-	
MW50399 - 112,-	
MW50396 - 112,-	
MW50397 - 112,-	
MW50398 - 78,-	
MW50399 - 78,-	

**100 % fabriksgaranti**

**Lysdioder**

3mm Rød-1.25 stk
3mm Rød-1.50 stk
3mm gul-1.25 stk
5mm gul-1.50 stk
3mm grøn-1.50 stk
5mm grøn-1.75 stk
3mm klar-1.65 stk
Ledholder 5mm-.50 sæt

**Displays**

FND500fk-10,-
FND501+y-10,-
FND507fa-10,-
DL701+y-10,-
DL704fk-10,-
DL707fa-8,-
DL500 - 10,-
DL747fa-8,-
DL750fk-10,-

**LATE NEWS**

1 MHz kristal kr. 48,-  
4 og 5 MHz kun kr. 38,-

**C-MOS HEF/EP - SSS/EP**

4000-1.35	4071-1.55
4001-1.35	4072-1.55
4002-1.45	4073-1.55
4006-4.95	4075-1.55
4007-1.45	4076-3.95
4008-4.95	4077-1.55
4009-8.75	4078-1.55
4010-8.75	4081-1.65
4011-1.35	4082-1.65
4012-1.55	4085-4.25
4013-1.95	4086-4.25
4014-4.95	4093-4.45
4015-8.85	4500-9.95
4016-2.95	4510-9.95
4017-4.75	4511-9.95
4018-4.75	4512-9.95
4019-3.75	4514-16.75
4020-6.50	4515-16.75
4021-9.95	4516-9.95
4022-5.45	4517-19.75
4023-1.45	4518-7.75
4024-4.45	4519-9.95
4025-1.55	4520-7.75
4026-12.75	4521-13.75
4027-2.95	4522-12.85
4028-4.95	4523-4.95
4029-4.45	4528-7.75
4030-1.95	4531-9.95
4031-12.95	4532-13.75
4033-12.95	4539-46.75
4034-6.25	4542-4.95
4036-12.45	4555-7.75
4040-6.90	4556-7.75
4041-4.95	4557-13.75
4042-4.95	4595-7.75
4043-4.95	40161-9.95
4044-4.95	40162-9.95
4045-5.95	40163-9.95
4047-4.95	40174-5.95
4048-2.25	40175-5.95
4049-7.75	4161 - 9.40
4051-7.75	40193-7.75
4052-6.25	40194-7.75
4053-6.25	40195-7.75
4055-14.90	4160 - 9.40
4055a-14.90	4161 - 9.40
4060-13.90	4160 - 9.60
4063-9.90	4163 - 9.60
4064-3.90	4192 - 3.40
4067-10.90	4193 - 3.40
4068-1.45	4194-17.75
4070-1.45	777100

# POP's Elektronik

Åben kl. 12.00 - 18.00 - Telefonordre kl. 10.00 - 18.00  
Bülowsvej 40 . 1870 Kbh. V . Tlf. 01 - 35 19 59 - 37 32 47

Leverandør til det danske elektronik-folk.  
Incl. 18 pct. moms for DK - Norge og Sverige + 18 pct. efterkrav.  
Gebyr kr. 12,-. Postordre minimum kr. 100,-.

# KIM-1: En hobbycomputer med stor support



M  
43

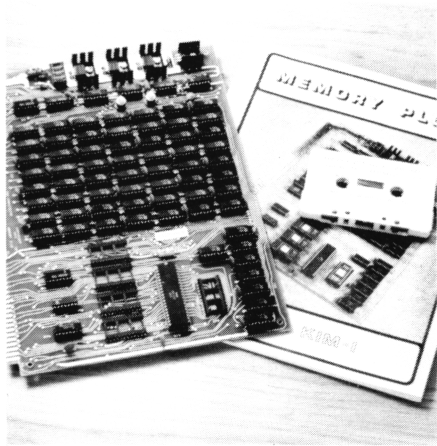
Da KIM-1 for få år siden blev designet omkring 65XX mikroprocessorkomponenterne, var der vel ingen der forestillede sig at denne microcomputer skulle få så stor succes. Verden over er der nu mere end 30.000 KIM-1 i drift, anvendelsesområderne er yderst forskellige.

En stor del anvendes i undervisningsøjemed, og ikke så få anvendes "professionelt" i styrings- og testopgaver og endelig til slut bør vel nævnes, at det hastigt voksende antal data-amatører virkelig har satset på KIM-1.

HFD september 1978

Årsagen til den store interesse omkring netop denne computer må vel først og fremmest ligge i LIM-1's enkle og overskuelige opbygning, kombineret med letfattelige betjening. En del af populariteten må sikkert også skyldes CPU'en i systemet - 6502 er meget velanskrevet i industrien bl.a. på grund af et effektivt instruktionssæt, men også på grund af sin høje arbejdhastighed.

En ting af meget stor betydning for nybegyndere inden for enhver gren af ny teknik, er om der findes tilstrækkeligt med dokumentation - eksempelvis når



det gælder mikrocomputere, er det yderst vigtigt at der i de medfølgende manualer findes praktiske eksempler i form af færdige programmer. Man skal huske på at førstegangsbrugeren er helt ukendt med hvorledes et program konstrueres og opbygges.

På dette punkt var de første KIM-1 brugere nok vel lidt hjulpet, User Manualen var for overfladisk og den mere professionelle Programming Manual er vel for de fleste amatører lidt tung at starte med. De fleste vil jo gerne trykke på nogle knapper i en fart.

Hurtigt blev der udgivet et par KIM-bøger ligesom der også nu findes flere månedsmagasiner. Mængden af færdiglavede "software-pakker" er meget omfattende, i flæng kan nævnes titler som digitalur, tyverialarm, enarmede tyveknagte, frekvenstæller osv. - alle programmer som kan være i KIM-1's IKRAM. Ønsker man at "køre" mere avancerede programmer såsom basic, skak, editor/assembler er det nødvendigt at udvide KIM-1's hukommelse.

#### KIM-1 UDVIDELSE

En bemærkelsesværdig kompakt udvidelse af KIM-1 er netop kommet til Danmark. Printkortet udvider KIM-1 på flere områder først og fremmest forøges hukommelsen med 8K byte EPROM. Også antallet af I/O-linier forøges med

16 stk., der er nemlig udover de 64 RAM kredse monteret en R6522 VIA (Versatile Interface Adapter), hvilket bl. a. giver mulighed for serielt interface, stor timerflexibilitet og mange former for interrupt.

En anden væsentlig feature er at bruge den ved hjælp af Memory Plus, som kortet hedder nu, får mulighed for at programmere sine egne EPROM'er, idet der på printet er monteret en EPROM-programmeringssokkel. Programmeringsrutinen medfølger på kassettebånd, så det nu er lige så enkelt at programmere EPROM'er som det er at overspille KIM-programmer til en kassettebåndoptager. Som det fremgår af foto 1 har Memory Plus samme format som KIM-1. Fordelelen herved er at KIM-1 og Memory Plus på enkel vis kan indbygges i samme kabinet, hvilket fremgår af foto 2.

#### TEKNISK DATA

RAM memory : 8K bytes low power statisk Ram.

ROM memory : Sokler samt dekoding for op til 8K bytes Eprom.

VIA : 2 programmerbare 8 bits I/O porte med mulighed for 2 16-bits timere.

Power Supply : På kortet findes 5 v regulator, så 8-10 v ustab. DC er tilstrækkeligt til at drive datamaten.

Desuden bør nævnes at alle LSI-kredse er monteret i sokler, således at evt. servicering kan foretages uden brug af specialværktøj, men chancen for komponentfejl skulle være minimal, så der her er tale om et færdigsamlet og afprøvet produkt og ikke et kit. SE

Prisen opgives til ca. kr. 2.300,-, excl. moms.

Import: MICRONOR APS  
TINDBJERGVEJ 16  
8600 SILKEBORG  
Tlf. 06-83 60 08

# ACT-IV

Ingen datamat er stærkere end det svageste led i transmissionskæden, og alt for ofte er dette terminalen — eller manglen på samme.

Der er mange muligheder for selv at bygge terminaler eller anskaffe brugt udstyr, men fælles for disse er deres begrænsede muligheder, som rigtigt sættes i relief, når man som her betragter en „professionel” skærmterminal.

Man skal kunne kommunikere med sin datamat — på den ene eller anden måde. Der findes et stort udvalg af ydre enheder, som kan muliggøre denne kommunikation. Tager vi f.eks. en IMSAI 8080, har den selv på fronten en række lysdioder og omskiftere, som tilsammen udgør en simpel — men særdeles praktisk — mulighed. Den bliver dog hurtigt ret trættende, og man ser sig om efter en anden løsning, som er lidt mere humant orienteret.

Skrivemaskinetastaturet og TV-skærmen er blandt de mest effektive og behagelige løsninger på ydre enheder.

Tastaturet giver mulighed for brugen af alle ASCII-karakterer, og TV-skærmen giver mulighed for udlæsning af samme karakterer samt anden grafisk fremstilling som f.eks. søjler og kurver.

Skærmterminalen med tastatur er den bedste løsning på 2-vejs kommunikation med datamaten.

(Blivende data må selvfølgelig udskrives på printer, hulbånd eller andet).

Den billigste løsning på en skærmterminal er et transportabelt TV hængt på en hjemmebygget TV-modulator, et løst tastatur og en TV-skriver (tegn-generator).

Hvis man er tilstrækkelig fingernem, kan man på den måde bygge sin terminal for omkring 3-4.000 Kr.

Resultatet vil fungere, men en sådan terminal vil have en række ulemper i sam-

menligning med en mere professionelt orienteret terminal.

Den selvbyggede skærm vil først og fremmest kræve konstant overvågning fra og styring af datamaten, som derfor ikke kan foretage sig noget fornuftigt, mens terminalen aktiveres.

En professionel terminal vil have en lille CPU indbygget, som kan styre terminalens egne funktioner.

En sådan terminal vil have en omskifter til bestemmelse af, om terminalen er tilsluttet datamaten, eller om den arbejder selvstændigt.

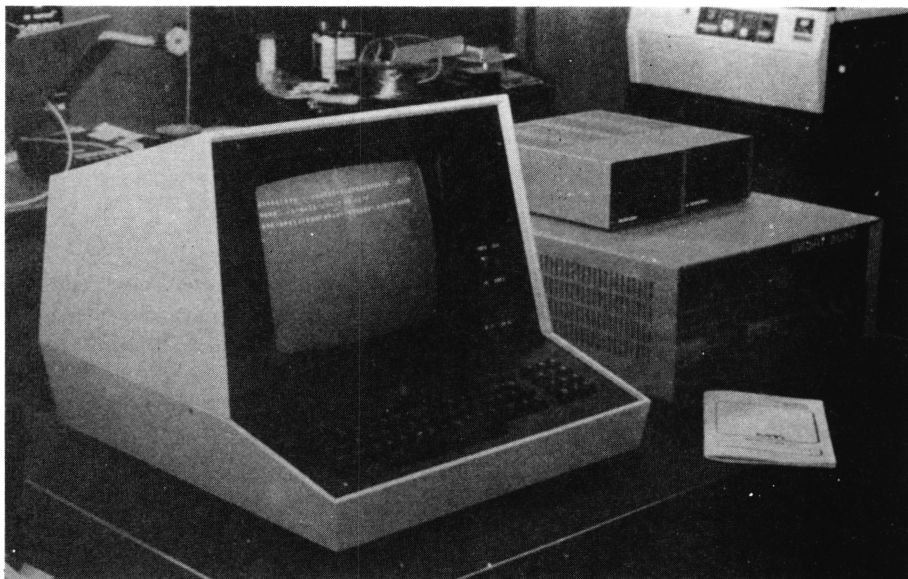
Terminalen kan selvfølgelig ikke foretage nogen form for databehandling, men der er en lang række situationer, hvor man bruger relativ lang tid til at indtaste data, måske i form af kursor-styrede søjlediagrammer, og terminalen kan selv holde styr på disse data i kraft af sin indbyggede hukommelse og CPU.

Når tiden er inde, kobles datamat og terminal atter sammen, og den ønskede behandling foretages.

## ACT - IV

En sådan terminal er ACT - IV, som fremstilles af Micro-Term, Inc. i USA. I Danmark repræsenteret af Piezodan.

Terminalen består af en TV-skærm og et ASCII-tastatur, det hele sammenbygget i et velkonstrueret kabinet med integreret strømforsyning — altså klar til at blive taget i brug.



Det, der i første omgang adskiller ACT-IV fra mere gængse terminaler, er dens indbyggede microprocessor, som sætter den istand til at fungere som selvstændig enhed. Stadig uden mulighed for databehandling, men med fuld redigeringskapabilitet på skærmen, mens datamaten udfører andre opgaver.

Dens alsidighed udbygges yderligere ved muligheden for både halv og fuld duplex operation.

#### REMOTE – LOCAL

På terminalen findes en omskifter, som gør det muligt at betjene terminalen i enten Local – lokalt – eller Remote – fjern – tilslutning.

Når terminalen står i Remote, virker den som direkte input til datamaten, som vil få hver indtastet karakter præsenteret til sin indgang. Altså ganske som en normal terminal.

I denne tilslutning må programmet sørge for at behandle alle karakterer ganske som de indtastes, og resultatet er derfor en kombination af operatørens sikkerhed og programmets smidighed.

I Local-tilslutning er terminalen som sådan afbrudt fra datamaten – med et par undtagelser.

ACT-IV sammen med en IMSAI 8080 med 2 stk. floppy-disk på toppen. Tastaturet er i almindelig skrivemaskinstørrelse og særdeles let at skrive på.

Operatøren kan stille og roligt indtaste sine data, som bliver placeret i terminalens interne hukommelse, som er tilstrækkelig til at kunne rumme 24 x 80 karakterer, som udgør 1 side på skærmen. Den interne CPU kontrollerer kursor, de specielle kontrol-instruktioner, linieskift m.m.

Når den ønskede datamængde er klar, kan operatøren ved tryk på SEND aflevere hele siden til datamaten med op til knap 20.000 bit i sekundet (max 19.200 baud).

Mens terminalen er i Local-mode, kan man ikke levere information via denne til datamaten, men datamaten kan give besked til terminalen over enten en indbygget „klokke“ eller i ren tekst (som så indgår i den indtastede data).

#### I PRAKSIS

Denne form for terminal er den eneste,

Nedenstående fotos fra en ACT-IV er optaget med højeste intensitet på skærmen af hensyn til den fotografiske optagelse — dette slører naturligvis punkterne en smule. I virkeligheden står teksten ved normal intensitet særdeles klart, skarpt og letlæseligt og med perfekt geometri (ingen buede linier).



Vi går i gang med at skrive, men opdager en fejl i sidste ord . . .



. . . Vi trækker kursoren tilbage til det sted, hvor fejlen er . . .



. . . og kan nu blot skrive det rigtige — den gamle tekst slettes automatisk samtidigt . . .

som vil kunne anvendes i praksis i større anlæg, hvor flere terminaler skal tilsluttes samme datamat. Et typisk eksempel er datamaten, som holder styr på billetter og priser for SAS. I alle SAS's kontorer findes terminaler, som er tilsluttet den samme datamat.

Hvis alle disse terminaler skulle optage datamatens effektive tid, mens operatørerne indtastede deres oplysninger og spørgsmål, ville billetbestillingen bryde sammen på kort tid.

Nu sker der det, at indformationen bliver i den lokale terminal, som placerer en request — anmodning — hos datamaten, som derefter på et passende tidspunkt indsamler oplysningerne, behandler dem og placerer svaret i terminalen — og går videre til næste opgave. På denne måde optages datamaten kun få brøkdeler af et sekund ved hvert terminalopkald.

Datamatklubber kan have lignende glæde af sådanne terminaler. Selv en — i sammenligning med virksomhedsdatamater — lille datamat som en IMSAI 80-80 kan med f.eks. 24K RAM og et par floppy-disc med lethed betjene 5 - 10 terminaler så effektivt, at alle operatører opfatter det, som om de havde datamaten for sig selv.

Men også den mindre forretningsdrivende eller amatør med lidt tidskrævende opgaver kan udnytte en terminal som ACT-IV.

Tag f.eks. en automatiseret faktureringsopgave med lagerudskrivning, lagerkontrol og automatisk ordreafgivelse.

Besked til datamaten om at sende en vare til en kunde kan beslaglægge denne i alt fra 5 sek. til måske et helt minut, hvis der samtidig skal udskrives både fakturaer, kreditnotaer, ordresedler, restordre, kundestatistik m.m.

Det vil være særdeles uøkonomisk, hvis operatøren i det ene øjeblik skal vente på maskinen, mens den i det næste skal vente på operatøren.

Med en terminal som ACT-IV indkoder man i programmet et klokkesignal, som lyder, når datamaten er klar til flere oplysninger fra terminalen, og operatøren

slår derefter over i Local, når der opstår ventetid.

Når klokken lyder, skiftes til Remote og trykkes på Send — og den i mellemtiden indtastede information glider over til datamaten. Og mens disse oplysninger behandles, fortsættes nye indtastninger i Local.

Hvis man vil have fuld glæde af sin ACT - IV terminal, bør man i sine programmer indbygge en rutine, som kan indlæse skærmens indhold i tilfælde af problemer med databehandlinger.

Det kan f.eks. ske, at datamaten i en række oplysninger finder et navn, som den ikke kan genkende, og den er selvfølgelig programmeret til at give besked om dette.

Men det ville være ærgeligt, om denne information indgik blandt de oplysninger, som senere blev indlæst fra skærmen. Derfor vil den mest effektive procedure være en bestemt signalering med klokken — f.eks. 3 gange 5 sek —. Den efterfølgende Send-instruktion lagres derefter kritikløst af datamaten i hukommelsen, hvorefter skærmen slettes og fejludskriften dukker op.

Når en tilfredsstillende løsning på fejlen er modtaget af datamaten, placerer den de oprindelige data på skærmen igen og fortsætter med at arbejde.

Det kan umiddelbart lyde tidskrævende, men da skærmen højst kan indeholde 1920 karakterer a 8 bit = 15.360 bit, vil en transport af samtlige karakterer højst vare 0,9 sek. Og da blanke karakterer ikke sendes afsted fra terminalen, viser det sig, at selv en veludfyldt skærm højst rummer 1.000 karakterer — altså højst et halvt sekund i transmissionstid.

Dette tidsforbrug skal sammenholdes med den tid, det vil tage at fjerne evt. fejludskrifter fra skærmen, før dennes indhold transmitteres til datamaten.

## DUPLEX

At opererer i Duplex betyder ganske enkelt, at TV-skærm og tastatur adskilles elektrisk og tilsluttes hver sin hhv. ud- og indgang på datamaten — ganske som ved normal tilslutning.

DETTE ER EN PRØVE  
PAA EN TEKST  
MED FLERE LINIER.

... Nu fik vi skrevet de første 3 linier og et linieskift har sat kursoren klar til tekst på næste linie ...

DETTE ER EN PRØVE  
PAA EN TEKST  
MED FLERE LINIER.

...men vi sprang en linie over, så vi flytter kursoren op til den første linie, som skal rykkes ned ...

DETTE ER EN PRØVE  
PAA EN TEKST  
MED FLERE LINIER.

... og det bliver den med en speciel kontrolinstruktion.

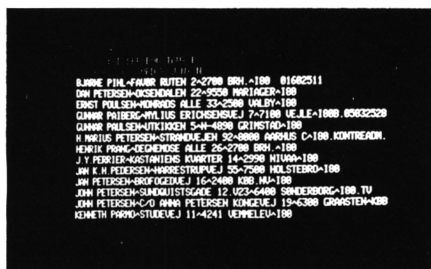
DETTE ER EN PRØVE  
VIA ACT IV  
PAA EN TEKST  
MED FLERE RETTEDE LINIER.  
\*\* MED VARIABEL INTENSITET

Her ser vi det endelige resultat med den indsatte linie. Sidste linie er med nedsat intensitet, som kan ændres både fra skærm og via programmet.





Eksempel på lettere grafisk repræsentation – et biorytme program. Programmet blev kørt den 31 august, hvor undertegnede rent faktisk havde en større general-forkølelse.



Eksempel på udskrift på ACT-IV til bedømmelse af den store mængde information, der kan være på en 80-karakters linie. Og så er der endda kun 63 karakterer på den længste af linierne. I højden er der plads til dobbelt så mange linier.



De øverste rækker viser de almindelige ASCII karakterer og tegn, mens nederste linie viser de grafiske tegn, som yderligere kan genereres. Disse giver mulighed for at lave både store og små firkanter, tomme eller udfyldte felter etc.

Det vil så ikke være tastaturet, som bestemmer, hvad der placeres på skærmen, men datamaten. I virkeligheden vil det være muligt med simultan og totalt separat indtastning og udlæsning – men dette vil sjældent være praktisk for amatører.

De kan der imod benytte Duplex-operationen til f.eks. forenklet indtastning,

HFD september 1978

hvor et indlæsningsprogram sørger for at placere den fulde tekst på skærmen.

Et typisk eksempel er telegram-tekst, der ofte gøres særdeles kortfattet.

Et eksempel er en anmodning om fremsendelse af brochurer, som i fuld tekst vil se sådan ud på engelsk:

Please forward leaflets thank you.

I telex-sprog ser dette således ud:

Pls fwdrlflts tks.

Hvis man først har lært denne form for forkortelser, kan man indtaste en tekst på næsten den halve tid – mens den fulde og korrekte tekst – oversat af datamaten – fremkommer løbende på skærmen. (En strimmel med forkortet telex-tekst kan selvfølgelig produceres samtidig eller efter redigering).

### ACT - IV IØVRIGT

Den mekaniske opbygning adskiller sig også på flere måder fra gængse terminaler. Fjernsynsskærmen og elektronikken til denne har en båndbredde på 10 MHz, hvilket giver mulighed for 80 karakterer pr. linie – og vel at mærke meget letlæselige karakterer.

Til sammenligning kan nævnes, at det skal være et godt „hjemme-TV”, hvis det kan opløse 64 karakterer på 1 linie, og flere producenter af TV-karakterer tager konsekvensen af dette og tilbyder ofte kun 32 karakterer pr. linie – vel vidende, at det tilsluttede TV vil sætte den praktiske grænse for opløsningen.

Tastaturet har foruden de gængse ASCII karakterer et fuldt numerisk tastatur og adskillige grafiske tegn, som gør det muligt at fremstille søjler, kurver og meget andet godt.

De grafiske muligheder er ikke på højde med f.eks. de, der ligger i en PET 2001, men anvendelsesmulighederne iøvrigt er så alsidige, at en sådan sammenligning egentlig ikke er relevant.

Terminalen er i besiddelse af en lang række kontrol-instruktioner, som er medvirkende til at gøre dataopstilling på

skærmen både let og effektiv. Disse instruktioner inkluderer naturligvis fuld kursor-kontrol, line-feed etc. Men også specielle ordre til indsættelse og fjernelse af linier og enkelte karakterer.

Terminalen har sine egne faste tabuleringspositioner, og yderligere kan programmeres fra datamaten.

Det er ligeledes muligt at ændre intensitet på skærmen - og dette kan programmeres fra datamaten, så det i en udskrift let kan skelnes mellem information af større eller mindre vigtighed.

Endelig kan datamaten selv kontrollere en printer, så data kan sendes fra terminalen til printeren, når redigering er overstået.

### KONKLUSION

ACT-IV ligger så afgjort på grænsen til det professionelle, idet den indeholder samtlige de muligheder, man med rimelighed ville forlange af en terminal til rent kommercielt brug.

Dens pris på kr. 7.800 excl. moms kan

måske virke afskrækkende på mange amatører, mens den effektivitet er så stor, at man i mange tilfælde kan spare kostbar programplads i datamaten, og hvis man ikke er inkarneret selvbygger, er det alligevel begrænset, hvor meget ekstra RAM, man får for forskellen på en ACT-IV og en almindelig terminal.

For de, som ønsker at spare lidt på de kostbare håndører tilbyder samme firma en mindre udgave uden skærm og til ca. kr. 3.800,— excl. moms. Denne terminal — ACT I — skriver 64 karakterer på 32 linier, men mangler et par af de mere avancerede muligheder, som vi ser i den store model.

Mange datamat-amatører bruger store summer på at udbygge en datamat med enorme lagre og dyre input-output kort, mens der sjældent bliver ofret større beløb på terminalen.

Hvis man først har arbejdet lidt med en ACT-IV og har gennemskuet dens muligheder, fristes man til at sætte den til en KIM-I eller lignende — det ville give et særdeles slagkraftigt system. PH

### KOMMUNIKATION

kan være besværlig — specielt med en datamat . . .

MEN, med Micro-Term's intelligente ACT-IV terminal går det let. Den har full-size tastatur, fuld Cursor kontrol fra hardware og software, 64 ASCII karakterer og 32 printbare symboler, skærmredigering, tilslutning til ekstra skærm, RS-232 tilslutning, 300 til 19600 baud o.s.v.

ACT-IV burde koste en formue — MEN vi leverer den til

**kr. 7800,— + MOMS**

Og vi kan endda levere en „lillebror“, ACT-1A (uden skærm) og med lidt færre „gimmics“ til

**kr. 3800,— + MOMS**

Begge terminaler demonstreres gerne i vor tekniske afdeling.

Ring til (01) 86 12 17 og få en aftale.

# piezodan aps.

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

### III. LØKKER OG FORGRENINGER

#### 7. Løkker

I alle programmeringssprog har man midler til at **styre udførelsen** af de enkelte dele af et program. Således at nogle afsnit bliver udført **flere gange**, eller sådan at de kun bliver udført **under visse betingelser**. Vi vil begynde med at se på et eksempel, hvor en del af et program bliver udført flere gange.

Vi ønsker at bestemme **gennemsnittet** (middelværdien) af en række tal, der er fremkommet ved, at man har vejet 10 personer. **Inddata** for vort program er altså et **talsæt** med 10 elementer, og vi kan tænke os, at dette talsæt er noteret på en liste, der ser sådan ud:

67 72 70 65 81 75 78 92 55 62

Inden vi begynder at skrive programmet, vil vi overveje hvilken proces, vi kan tænke os at udføre for at få udregnet det nævnte gennemsnit. Det er klart, at vi skal læse tallene eet for eet og lægge dem sammen, efterhånden som vi læser dem. Når vi er færdige med det, skal vi dividere den fremkomne sum med 10. Vi kan til at begynde med beskrive processen således:

```
sum=0
læs det første tal
læg det til sum
læs det næste tal
læg det til sum
læs det næste tal
læg det til sum
læs det næste tal
læs det næste tal
osv.
indtil alle tal er læst
divider sum med 10
```

Vi ser, at de to delprocesser: "læs det næste tal" og "læg det til sum" bliver udført gentagne gange. Derfor er det mere overskueligt at beskrive processen således:

```
sum=0
læs det første tal
```

```
gentag følgende:
  læg tallet til sum
  læs det næste tal
indtil alle tal er læst
divider sum med 10
```

Det er nu let at skrive et program, der kan udføre processen, men først skal vi have listen over talværdierne skrevet op, så den kan læses i et program. I Basic har man en sætning til rådighed for netop et sådant formål. Sætningen består af ordet DATA efterfulgt af en liste med tal, der skal læses. I vort eksempel kommer den til at se sådan ud:

```
DATA 67,72,70,65,81,75,78,92,55,62
```

Hvis man har adgang til en RC 7000, en ID 7000 (fra Dansk Dataelektronik), en Alpha Lsi (fra SC Metric) eller den MPS 2000, som Mogens Pelle beskriver her i bladet, kan programmet skrives således:

```
0010 DATA 67,72,70,65,81,75,78,92,55,62,0
0020 LET SUM=0
0030 READ T
0040 REPEAT
0050 LET SUM=SUM+T
0060 READ T
0070 UNTIL T=0
0080 LET GNS=SUM/10
0090 PRINT "GENNEMSNITTET ER:";GNS
```

I linje 30 finder vi sætningen:

```
READ T
```

Programordet READ betyder "læs", og når sætningen udføres, læses det første tal på listen efter ordet DATA, og det **tildeles den variable T**, der ledsager READ, som værdi. Når linje 30 er udført, har T altså fået værdien 67. Derpå sølger ordet REPEAT, som betyder "gentag", og det, der skal gentages, er naturligvis linjerne 50 og 60. I linje 50 står der:

```
LET SUM=SUM+T
```

# KIM-1/6502

systemer

## STORT UDVALG AF HARDWARE & SOFTWARE F.EKS.

**MEMORY PLUS** Kr. 2024  
8K RAM, PROM programmer,  
sokler for 8K EPROM (2716),  
6522VIA. Samlet, testet, alt incl.

**Skærm Terminal – TTY** Kr. 6169  
20 mA loop interface, komplet  
ascii, cursor adressering m.m.

**HELP for minimal KIM-system**  
**EDITOR** Kr. 120  
**MAILING LIST** Kr. 120  
**INFORMATION RETR.** Kr. 120

**FCL-65** Kr. 175  
Oversætter for programsproget  
FOCAL, 8K memory og terminal.

PRISERNE ER INCL. MOMS

DANSKE VEJLEDNINGER

REKVIRER  
KATALOG OG PRISLISTE  
FRA

# lisco

MICRODATA

Aprilvænget 6, 6000 Kolding

Vi har tidligere set eksempler på sådanne sætninger. Når den udføres, bliver værdien af T lagt til den nuværende værdi af SUM, og den derved udregnede talværdi ( $0 + 67 = 67$ ) bliver tildelt SUM som ny værdi. Derpå udføres sætningen i linje 60, hvorved T får tildelt værdien af det næste tal på listen (72). Vi er nu nået til linje 70, hvor der står:

UNTIL T=0

der kan oversættes direkte til "indtil T er lige med 0". Denne sætning hører sammen med REPEAT i linje 40, og det er faktisk den, der bevirker, at de to linjer 50 og 60 bliver udført flere gange. Hver gang, linje 70 udføres, undersøger systemet, om T har fået tildelt værdien 0. Hvis ikke der er tilfældet, udfører systemet atter linjerne mellem REPEAT og UNTIL. Da T ikke er 0, bliver linje 50 og 60 altså udført igen. Derved får SUM tildelt værdien 139 ( $67 + 72$ ), hvorpå T får tildelt værdien af det tredje tal på listen (70). Sådant bliver vi ved, indtil vi når til det sidste tal på listen. Dette tal er 0, og læseren vil bemærke, at det ikke findes på den oprindelige liste. Vi har føjet det til med det ene formål at kunne standse udførelsen af den mølle, vi har fået startet ovenfor, og det er klart at "stopklodsen" virker. Vi har netop set, at udførelsen af de to linjer 50 og 60 kun bliver gentaget, hvis T ikke har værdien 0. Læg godt mærke til, at der ikke bliver gjort noget forsøg på at lægge det ekstra tal 0 til SUM, idet sætningen, der læser tallet, følger efter sætningen, hvor tallet lægges til SUM. I mange tilfælde bruger man ikke så uskyldige værdier som 0 til at standse en gentagen udførelse, og i så fald ville der naturligvis opstå en fejl, hvis også stop-tallet bliver lagt til. Når T har værdien 0, fortsættes udførelsen af programmet med linjen efter UNTIL-sætningen, hvorved man får udskriften:

GENNEMSNITTET ER: 71.7

Et programafsnit, der udføres flere

HFD september 1978

gange, kaldes også en **løkke**. I forbindelse med en løkke har man altid en **betingelse**, der bruges for at få standset løkken. I vort eksempel lyder betingelsen: "T er lig med 0", der naturligvis på sædvanlig "matematisk" skrives:  $T=0$ . I datalogien kalder man også en sådan betingelse for et **Boolsk udtryk** (efter den engelske matematiker George Boole), og man siger, at et Boolsk udtryk kan have værdien **sand** eller **falsk**. Vi vil benytte denne sprogbrug i næværende fremstilling. Ordet UNTIL skal altid efterfølges af et Boolsk udtryk, og som vi har set ovenfor, standser udførelsen af REPEAT..UNTIL-løkken, når det Boolske udtryk efter UNTIL har værdien **sand**. Hvis man ikke har en datamat, som er udstyret med en avanceret fortolker, men kun kan bruge standard Basic, må man skrive programmet ovenfor lidt om. For det første kan man ikke bruge navnet SUM for en variabel. Man må nøjes med fx. betegnelsen S. Værre er, at man heller ikke har **styreordene** REPEAT..UNTIL. I stedet må man bruge en såkaldt GOTO-sætning:

```
0010 DATA 67,72,70,65,81,75,78,92,55,62,0
0020 LET S=0
0030 READ T
0050 LET S=S+T
0060 READ T
0070 IF T<>0 THEN GOTO 0050
0080 LET G=S/10
0090 PRINT "GENNENSNIETET ER: ";G
```

I linje 70 står der:

```
IF T <> 0 THEN GOTO 50
```

Denne sætning kan oversættes direkte til: "hvis T er forskellig fra 0, så gå til 50", hvor der med "50" menes linjen med nummeret 50. Det lidt besynderlige tegn "<>" betyder egentlig: "er mindre end eller større end", men det er naturligvis det samme som: "er forskellig fra". Programmet virker på helt samme måde, som det, der er beskrevet ovenfor. Udførelsen af linjerne 50 og 60

bliver gentaget, indtil T har værdien 0. Så vil det Boolske udtryk  $T <> 0$  have værdien **falsk** (T er ikke forskellig fra 0), og derfor adlydes **hopordren** GOTO 50 ikke, men udførelsen fortsættes med linje 80, og vi får den samme udskrift som før. Bemærk, at den variable, der tildeles værdien af gennemsnittet, nu kaldes G. En sætning, som den her viste, kaldes en **betinget hopordre**, og den er altid opbygget således: først anføres order IF, derpå følger et Boolsk udtryk, og til sidst en angivelse af, hvilken linje der skal hoppes til. I nogle Basic-versioner må man ikke skrive GOTO, men skal nøjes med THEN, og i andre må man kun skrive GOTO. Se godt efter i beskrivelsen af Basic-versionen for at finde ud af, hvilken syntaks der benyttes. En betinget hopordre udføres altid sådan, at der hoppes, hvis det Boolske udtryk har værdien sand (altså hvis betingelsen er opfyldt), men fortsættes med den næste linje, hvis det Boolske udtryk har værdien falsk (altså hvis betingelsen ikke er opfyldt). Måske er een og anden blandt læserne irriteret over, at sætningen READ T forekommer to gange i programmet. Det kommer naturligvis af, at programmet er opbygget nøjagtigt som det ovenfor. Det kan undgås, at programmet indeholder to READ-sætninger men så skal det omformes en smule. Lad os se på en anden mulighed. Vi kan fx. skrive sådan:

```
0010 DATA 67,72,70,65,81,75,78,92,55,62,0
0020 LET S=0
0030 READ T
0040 IF T=0 THEN GOTO 0070
0050 LET S=S+T
0060 GOTO 0030
0070 LET G=S/10
0080 PRINT "GENNENSNIETET ER: ";G
```

I linje 30 læses som før et tal, hvis værdi tildeles den variable T, men i stedet for straks at lægge dette tal til den variable S, undersøges det først, om T skulle være lig med 0. Hvis det er tilfældet, fortsættes udførelsen med linje 70, hvorefter gennemsnittet udskrives. Hvis ikke T er lig med 0, udføres linje 50, i

hvilken T lægges til S ganske som vi tidligere har set det. Derpå følger i linje 60 en ubetinget hopordre:

GOTO 30

der naturligvis bevirker, at udførelsen af programmet fortsætter med linje 30, hvor et nyt tal læses. Linje 30, 40 og 50 udgør altså det indre af en løkke, men nu har vi mulighed for at **hoppe ud af løkken** i linje 40, hvis det viser sig, at T er lig med 0. Det er altså virkelig lykkedes os at spare en READ-sætning, og det kunne se ud til at være meget fikst gjort. Man vil også bemærke, at programmet er en linje kortere, end det foregående program. Imidlertid har vi til

gengæld fået to hopsætninger: en betinget og en ubetinget. Nu er en ubetinget hopsætning ganske vist meget hurtigt at få udført, men lidt tid tager det dog for systemet at læse den og finde den sætning, den henviser til (**hopadressen**). Hvis løkken i programmet skal udføres mange gange, fx. flere tusinde, vil det betyde, at det sidste program bliver en anelse langsommere end de to første. Ved at udføre prøvekursler med lignende programmer, har forfatteren fundet, at programmet herover tager ca. 5 % længere tid, end det umiddelbart foregående, hvor der kun benyttes een hopordre. Hertil kommer, at mange hopsætninger i et program kan føre til, at man mister overblikket over programmet.

 **commodore**

## BASIC bordcomputer-system

Revolutionen på computerområdet er med PET 2001 en realitet.

PET computerens utallige funktioner og store lagerkapacitet gør den anvendelig indenfor de fleste teknisk-videnskabelige områder såvel som til finanstekniske og kommercielle formål.



Et komplet system til en pris, der giver udtrykket »personal computing« mening.

Tekniske data:  
8k arbejdslager, option til 32k  
8k byte interpreter  
4k byte monitor  
1k byte monitor for maskinsprog  
1k byte testrutine  
IEEE-488 interface  
Programlager på standard kassettebånd

**Pris kr. 8.900,- excl. moms**

Ekstra tilbehør:  
Floppy disk & printer

**INSTRUTEK**

Hovedkontor:

Christiansholmegade  
8700 Horsens

Tlf. 05 - 61 11 00

Øst:

Rødovrevej 155  
2610 Rødovre

Tlf. 01 - 41 34 00



DET KUNNE JEG TÆNKE MIG AT LÆSE MERE OM:

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....



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

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 bestiller 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: .....



BREV

Porto  
120  
øre

Husk afsender

Til:

Telepress ApS

Greve Strandvej 42  
2670 Greve Strand

▲  
**KLIP LANGS STREGERNE HELT TIL BLADETS KANT**  
▼

BREV

Porto  
120  
øre

Husk afsender

Til:

Telepress ApS

Greve Strandvej 42  
2670 Greve Strand