

BYGG SJÄLV



ABC 80 till 64 kbyte

Med några timmars arbete och för 650 kr kan du enkelt modernisera din ABC 80 så att den får 64 kbyte RAM-minne utan att påverka den normala funktionen. Utbyggnaden består i stort sett av att byta ut några minneskapslar så inget kretskort behövs. Några av de nya applikationerna är: större BASIC-program, möjlighet att använda ABC800-BASIC, förbättra den ursprungliga BASIC:en och anpassa program från andra datorer. I det här första avsnittet visas hur själva ombyggnaden går till, i det andra behandlas ändringar och förbättringar av ABC80-BASIC och i det tredje och sista hur man anpassar ABC800:s BASIC till ABC80.

Av Christer Ekman, Foto Peter Rosén

V arför ska man ha 64 kbyte minne när de flesta program man skriver inte är större än några tusen bytes, kan man fråga sig. Men även om man på köpet med den här utbyggnaden får 16 kbyte ytterligare minne för lagring av BASIC-program så är det inte det som är den stora finessen. Normalt har ABC80 nedre halvan av minnet fyllt med ROM som innehåller bl a BASIC-tolken och DOS (DiskOperativSystem). Flyttar man över innehållet i dessa minnen i RAM på samma adress får man samma funktion som normalt men med möjlighet att ändra i det. Man har då friheten att kunna skriva om delar av BASIC-tolken för att passa sina egna behov. Irriterar man sig t ex på småsaker som prompten "ABC80" ändrar man den enkelt genom att lägga in en egen text. Ett annat exempel är om man vill höja hastigheten på överföringen till kassettbandspe-

laren dubblar man den enkelt genom att ändra 3 bytes.

ABC800 har en mycket liknande minneskarta jämfört med ABC80 (bildminne, BASICtolk, programminne ligger på ungefär samma adresser) som gör det enkelt att föra över ABC800:s kraftfullare BASIC.

Ett annat exempel på applikation är lagring av data (RAM-minnet ger mycket snabbare accesstid jämfört med en diskett). Den som behärskar assemblerprogrammering kan utveckla eller anpassa egna operativsystem eller språk t ex CP/M, FORTH för att kunna använda mer programvara.

Själva utbyggnaden är relativt enkel så att även den som inte har någon erfarenhet av elektronik bör kunna klara av det om man är noggrann. I stycket nedanför förklaras lite om funktionen men den som inte är intresserad av den kan gå direkt på stycket "Byggbeskrivning".

KRETSLÖSNING

För att passa olika applikationer kan man välja mellan fyra olika sätt att koppla in 64 K-RAM:et. Valet sker programmässigt med instruktionen OUT 7,x där x är ett tal mellan 0 och 3 och väljer minneskarta enligt *figur 1*.

Adressavkodningen i ABC80 består av ett PROM som fungerar som en tabell över vilket minne som skall vara inkopplat (RAM, ROM, bildminne eller externt minne) vid olika adresser. Genom att byta det PROM:et och tillföra signaler från en utport får man fyra olika minneskortor enligt *figur 1*. Vid spänningspåslag är alltid karta 0 inkopplad som motsvarar en normal ABC80 fast med 16 kbyte extra minne. I minneskarta 1 har BASIC-ROM och DOS bytts ut mot RAM för att kunna föra över ABC80:s eller ABC800:s BASIC i RAM och göra ändringar. Minneskarta 2 och 3 är för den avancerade assemblerprogrammeraren. Man har då tillgång till hela 64 k-RAM-arean som fyller hela kartan. Om utskrifter skall göras kopplas karta 1 eller 2 in för att komma åt bildminnet.

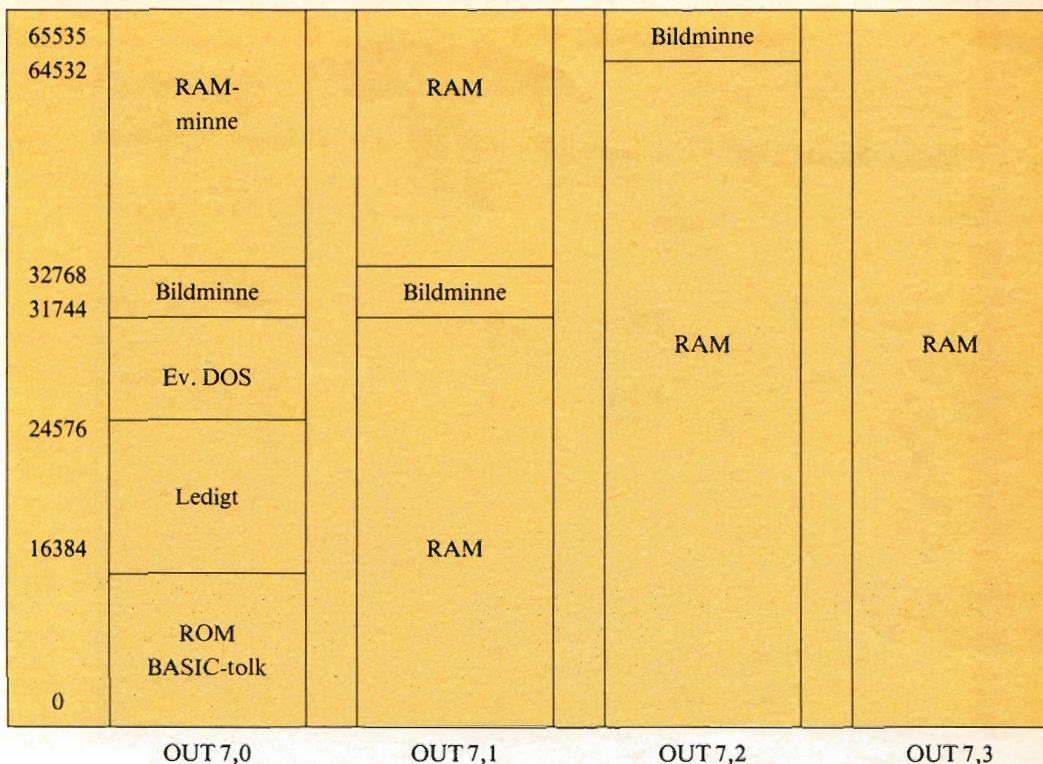
Ljudgeneratoren i ABC80 styrs av en utport bestående av en 74LS273 som är en 8-bitars låskrets (*se figur 2*). Vid instruktionen OUT 6,x läses värdet x fast på utgångarna 1Q—8Q (Dataingångarna D1—D8 är anslutna till databussen) som går till ljudgeneratoren. Med ingången CLEAR (pinne 1) nollställs samtliga utgångar. För att åstadkomma utporten för styrningen av minneskartorna monteras en extra 74LS273 ovanpå (så kallad Piggy Backing) men med 2 av utgångarna kopplade till PROM:et som styr adressavkodningen. Kockingången (pinne 11) kopp-

las så att utporten styrs med instruktionen OUT 7,x. Vid spänningspåslag eller intryckning av RESET-knappen kommer alltid karta 0 vara inkopplad eftersom CLEAR-ingången nollställer 74LS273.

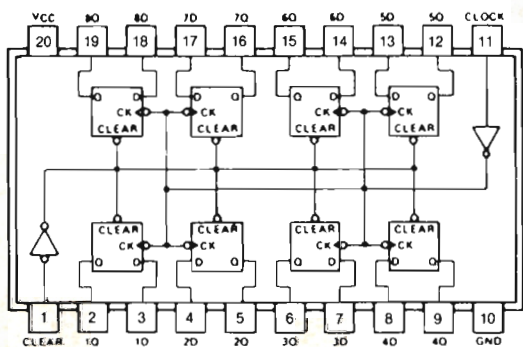
RAM-minnet i ABC80 består av 8 stycken 4116, ett 16 kbit dynamiskt RAM-minne vars pinkonfiguration visas i *figur 3*. Jämför man det med 4164, ett 64 kbit dynamiskt minne vars pinkonfiguration visas i *figur 4*, ser man att det endast är fyra pinnar som skiljer. På 4116 är pinne 1,8,9,16 strömförsörjning med spänningarna —5V, +5V, +12V men eftersom 4164 endast behöver +5V så blir det 2 pinnar över, 1 och 9. Pinne 1 brukar vara en refreshfunktion som inte används här eftersom ABC80 sköter den biten.

Pinne 9 är en extra adressledning för att kunna adressera 64 Kbit istället för 16 Kbit (adressen multiplexas in i två steg därför behövs bara 7 adressgångar på 4116 och en extra för att adressera 4 gånger så mycket minne).

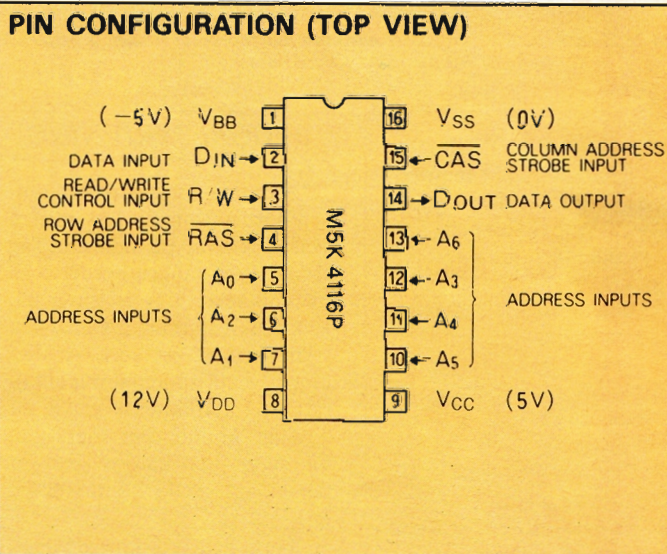
För övrigt är funktionen identisk för 4116 och 4164. Genom att byta ut 4116 mot 4164 och andra dessa 4 pinnar fyr-dubblas minnet enkelt. För att få den extra adressledningen A7 på minneskretsarna monteras en extra 74LS257 på en av de två som ursprungligen multiplexar in adressen i RAM-minnena. 74LS257 är en "Quad 2-line to 1-line multiplexer" (*se figur 5*). Den kan liknas med fyra stycken elektroniska omkopplare som styrs av en kontrollsignal SELECT. Inläsning av adress till minnet sker i två faser, den ena med omkopplarna i ett läge och den andra faser i andra läget. Selectingången styrs av samma signal som de två ursprungliga och OUTPUT CONTROL används ej och ligger på konstant låg nivå.



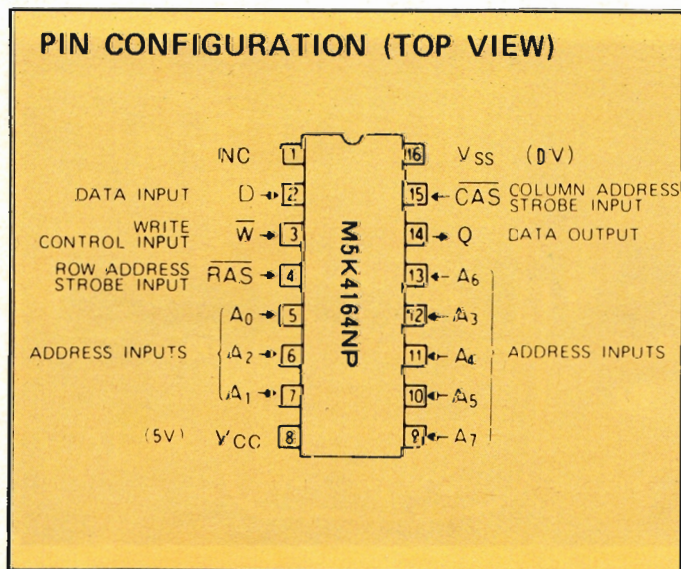
Figur 1. De fyra olika minneskartorna. De två första använder man till den ursprungliga BASIC-tolken. Den första ligger inne vid uppstart och till den andra kan man flytta över tolken i RAM. De två sista minneskartorna används när man har något annat operativsystem, t ex CP/M. Normalt ligger då den sista kartan inne och vid utskrift på skärmen kopplar man in den näst sista.



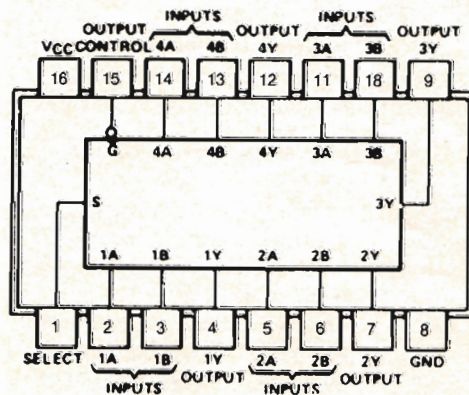
Figur 2. Kretsen som här används som utport heter 74LS273 och är en 8-bitars låskrets av D-typ. Vid en positiv flank på klockingången (pinne 11) läses Q-utgångarna till D-ingångarnas värde. Med CLEAR-ingången (pinne 1) kan Q-utgångarna nollställas.



Figur 3. 4116, 16 kbit dynamiskt RAM-minne. Kräver -5V, +5V och +12V.



Figur 4. 4164, 64 kbit dynamiskt RAM-minne. Matningsspänningen är bara 5V.



Figur 5. 74LS257, en 2 till 1 multiplexen fungerar som 4 stycken 2-poliga strömställare. Med SELECT (pinne 1) väljs om utgången 4 ska ha A- eller B-ingångarnas värde.

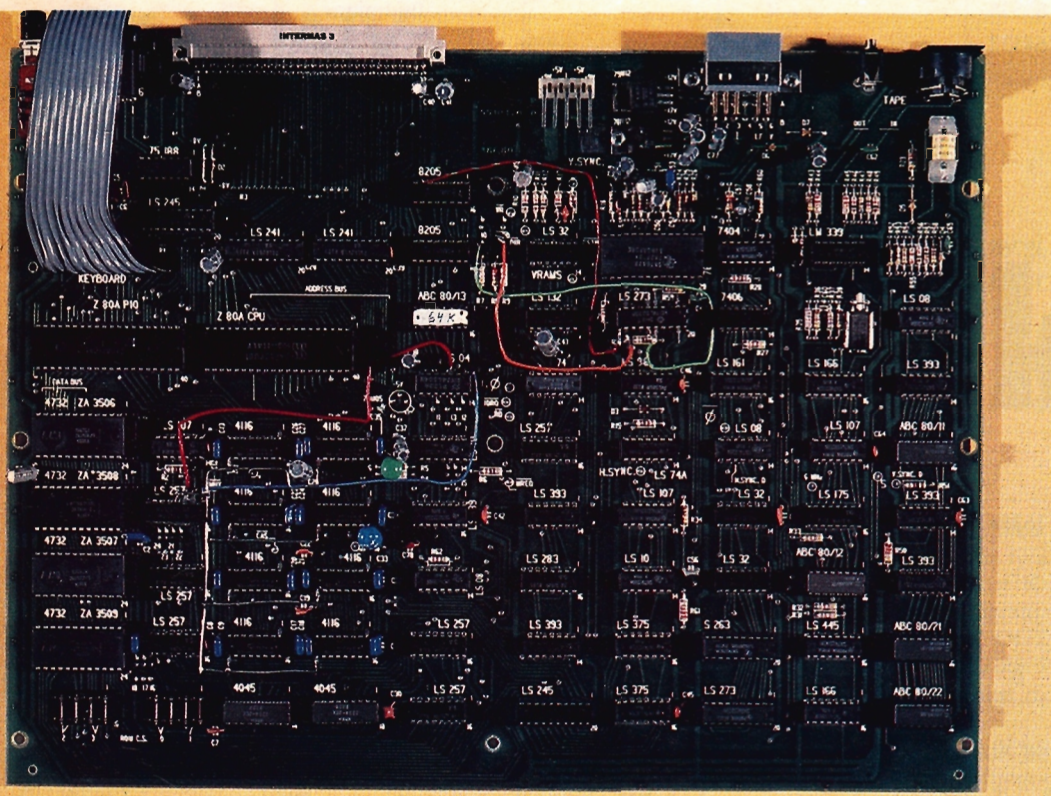
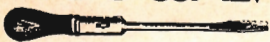


Bild 1. Hela ABC 80:s kretskort när man har monterat ur det. I nedre vänstra hörnet sitter de 8 minneskapslarna som har bytts ut för att få 64 Kbyte minne.

DET HÄR BEHÖVER DU

I lödkolv, helst temperaturreglerad, eller cirka 15–30 W med fin spets.

Lödtenn, cirka 1 mm tjockt.

Blanktråd för bygglar cirka 1 mm tjock.

Tråd i olika färger för förbindningar mellan IC-kretsar cirka 0,25 kvadratmillimeter.

Plattång att bocka ut IC-kretspinnar med.

En vass kniv att skrapa av kretskortledningnar.

1 74LS273 8-bitars D-latch.

1 74LS257 Quad 2-line to 1-line multiplexer.

1 bipolärt specialprom.

8 st MSK4164NP-15, 64 kbit dynamiskt RAM-minne 300ns eller snabbare access-tid.

Komponentsats bestående av 8 st 4164 (150ns), 1 st 74LS273, 1 st 74LS257 och ett specialprom kan erhållas från MIKO KOMPONENT AB, BOX 1004, 126 10 HÄGERSTEN, Tel 08-88 08 80, för 534 kr exkl moms. Endast specialprom:et kostar 136 kr exkl. moms. Ring för prisuppgift på färdig monterning.

BYGGBESKRIVNING

Montera av kåpan på tangentbordet som är fäst med 5 svarta skruvar på sidorna. Tag sedan loss tangentbordet och plåtvinklarna som det sitter på. Kylflänsen som är fäst med 4 st stjärnskruvar tas bort. Anslutningarna till spänningsregulatorerna på kylflänsen sker via ett 5-poligt kontaktdon (när man monterar ihop datorn igen går det ej att vända kontaktdonet fel).

Nu är hela kretskortet frilagt. Lägg märke till raden av siffror i vänstra kanten och raden med bokstäver i kretskortets nedre kant. De används för att enkelt beskriva IC-kretsarnas position. IC-krets K1 är t ex kretsen i nedre kant. De används för att enkelt beskriva IC-kretsarnas position. IC-krets K1 är t ex kretsen i nedre högra hörnet.

Först monteras IC-kretsen 74LS273 som är en 8-bitars D-latch. Bocka med hjälp av en plattång ut pinne 2, 5, 6, 9, 11, 12, 15, 16, 19. Löd fast kretsen på krets G7 (74LS273) i de övriga pinnarna (se bild 2). Se till att den vänds åt rätt håll, alla kretsar skall vara monterade med pinne 1 åt höger. Löd först "slarvigt" en av pinnarna så att kretsen

fixeras. Lös sedan de övriga pinnarna ordentligt och till sist den "slarvigt" lödda pinnen. Lödkolven skall helst vara temperaturreglerad eller ha en effekt på 15–30 W och ha en finkalibrig spets.

Löd en tråd från pinne 11 som är utbockad till pinne 7 på krets E9 (ca 13 cm, rödvit). Mellan krets E8 och E9 finns en bygel märkt B20 på kretskortet (se bild 2) som skall kapas. Bygeln går till R9 i ena änden och till +5V i den andra. Skrapa bort

några mm så att det helt säkert är avbrott. Dra sedan en tråd från B20 (den nedre änden mot R9) till den utbockade pinnen 12 på 74LS273 (9 cm, orange). Från B10 bredvid B20 skall det gå en tråd till pinne 15 på 74LS273 (15 cm, grön).

PROM:et som sitter i hållare i position E7 byts ut mot det nya vitmålade märkt "64K". Bocka ut alla pinnar utom 1, 8, 15, 16 på IC-kretsen 74LS257. Klipp av pinne 5, 6, 7, 8, 9, 10, 11, 12, 13, 14 och löd på samma sätt som 74LS273 fast 1, 8, 15, 16.

Pinne 1 på E6 (74LS04) förbinds till pinne 2 på 74LS257 (blå, 13 cm) och pinne 3 på E6 förbinds med pinne 3 på 74LS257 (14 cm, röd).

Hittills är inga "permanenta" ändringar gjorda. Om man vill ha tillbaka ABC80:n i originalutförande är det bara att byta tillbaka PROM:et i position E7. För att kontrollera att allt är rätt hittills är det dags för ett test. Montera ihop datorn och slå på den. Tag bort eventuell floppy eller annan anslutning till den 64-poliga busskontakten. Mata in följande program:

- 1 PRINT CHR\$(12)
- 2 I=52000
- 3 ONERRORGOTO 5
- 4 READ A: POKE I,A : I=I+1: GOTO 4
- 5 PRINT CALL(52000)
- 6 END
- 7 DATA 243,62,237,50,0,102,62,71,50,0
- 8 DATA 103,62,0,211,7,17,0,16,33,255
- 9 DATA 254,0,32,6,221,54,0,87,24,14

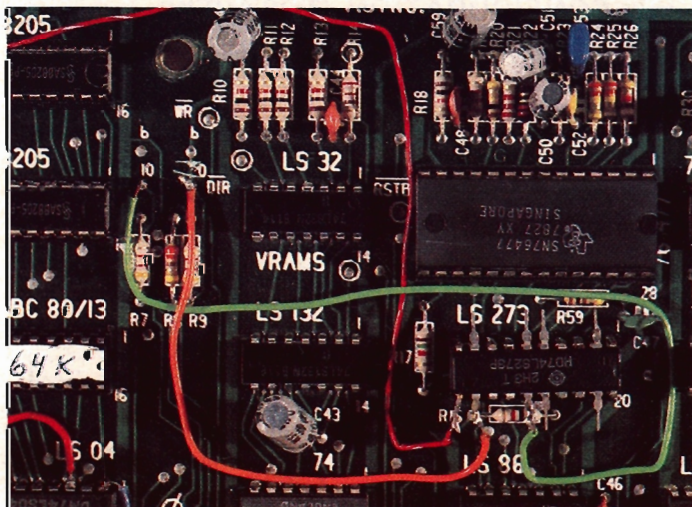


Bild 2. I nedre högra hörnet sitter utporten som styr vilken minneskarta som skall vara inkopplad. Den vitmålade kretsen till vänster är det utbytta PROM:et.

- 10 DATA 254,0,32,6,221,54,0,87,24,14
- 11 DATA 254,255,32,6,221,54,0,78,24,4
- 12 DATA 221,54,0,82,221,35,25,16,224
- 13 DATA 24,254

Det är ett assemblerprogram som går igenom hela minnet och kontrollerar på var 4 096:e minnescell om det är RAM eller ROM inkopplat. För ROM skrivs ett R, för RAM W och om ingenting är inkopplat ett N.

Skriv RUN. På skärmens övre rad skall det stå RRRR-NNNNWWWWW som är minneskarta 0. Eventuellt kan det komma fram tecken på någon annan rad också.

Tryck RESET kortast möjlig tid. För att prova minneskarta 1 skriv: POKE 52012,1 : ;CHR\$(12)CALL(52000). På skärmen skall det stå WWWWWW- WWWWWW.

Tryck RESET igen och skriv: POKE 52012,2 : POKE 52026,252 : ;CHR\$(12)CALL(52000). På skärmen skall det stå WWWWWW- WWWWWW.

Tryck RESET igen och skriv: POKE 52012,3 : ;CHR\$(12)CALL(52000)—. Skärmens övre rad skall vara tom.

Om allting gått bra är det bara att montera isär datorn igen och fortsätta. Annars gå tillbaka och kontrollera steg för steg för att hitta fel.

De 8 ursprungliga 16 kbit RAM-kapslarna 4116 sitter i hållare i position C2—5 och D2—5. Ta ur dem och sätt dem helst i ledande skumplast.

Till vänster om E5 sitter —5V regulatören 79M05 (ser ut som en transistor). Eftersom det

endast är 4116-kapslarna som behöver —5V klipps den bort (se bild 3). Även +12V till 4116 måste bort på grund av att de nya 4164 skall ha +5V. Kapa +12V (som går till D5 pinne 9) till vänster om —5V-regulatorn, se även bild 3. Skrapa bort några mm med en vass kniv och se till att ingen kontakt finns. Mät gärna med en ohmmeter på ett lågt område mellan pinne 32 på den 64-poliga busskontakten och pinne 9 där 4116-kretsarna satt.

Lossa kretskortet genom att skruva bort de 3 stjärnskruvarna och trycka ihop de 3 vita plastdistanserna som håller fast det. På baksidan av kretskortet bygglas pinne 8 och 9 ihop på kretsarna C2—5 och D2—5 med blanktråd, se bild 4. Byglarna har till uppgift att mata pinne 8 på kretsarna med +5V (jämför även pinkonfigurationen på 4116 och 4164). Montera sedan fast kretskortet.

Bocka ut pinne 1 och 9 på de 8 4164 IC-kretsarna och sätt dem där 4116-kretsarna satt. Se till att de vänds åt rätt håll och kontrollera samtidigt att de övriga kretsarna sitter rätt. Pinne 9 på 4164:orna förbinds parvis med blandtråd och förbinds gemensamt till pinne 4 på 74LS257, se bild 5.

Montera ihop datorn, när plastkåpan skruvas fast se till att skruvarna gängas på rakt, det är lätt hänt att de hamnar snett. Prova datorn genom att köra ovanstående testprogram, resultatet skall bli likadant. □

I nästa nummer kommer det lite tips om hur man kan modifiera basic-tolken med hjälp av 64K extra-minnet.

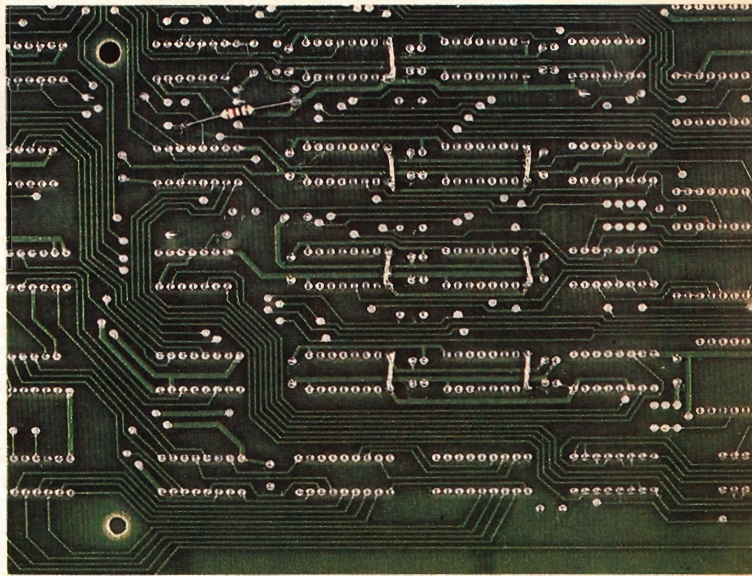


Bild 4. Baksidan av kretskortet. Här skall 8 stycken byglar lödas fast mellan pinne 8 och 9 på RAM-kretsarna. Lägg märke till bokstäverna i nedre kanten som gör det lättare att orientera sig. Tyvärr har bygeln till krets C5 fallit bort på bilden men det påverkar inte funktionen. Motståndet på bilden satt från början och har INTE med utbyggnaden att göra.

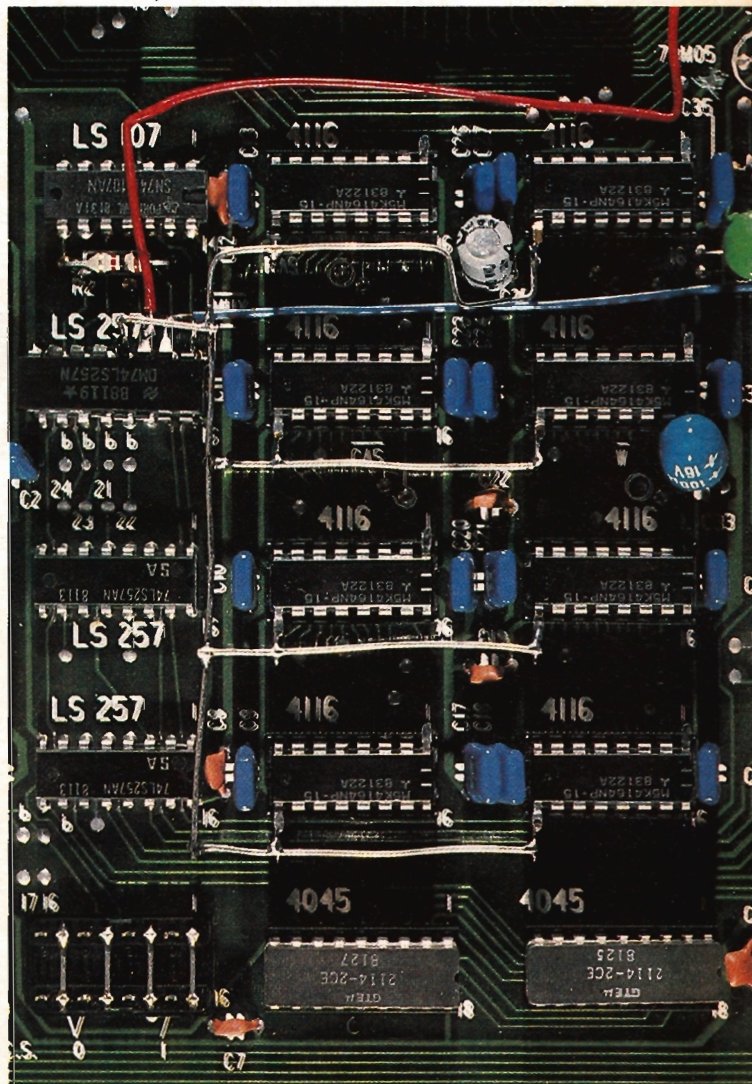


Bild 5. Här är de nya utbyttas RAM-kapslarna. Pinne 1 klipps bort och ansluts ej och pinne 9 förbinds till de andra 4164-kapslarna till pinne 4 på krets B4. I övre högra hörnet sitter den bortklippta —5V-regulatorn som visas på bild 3.

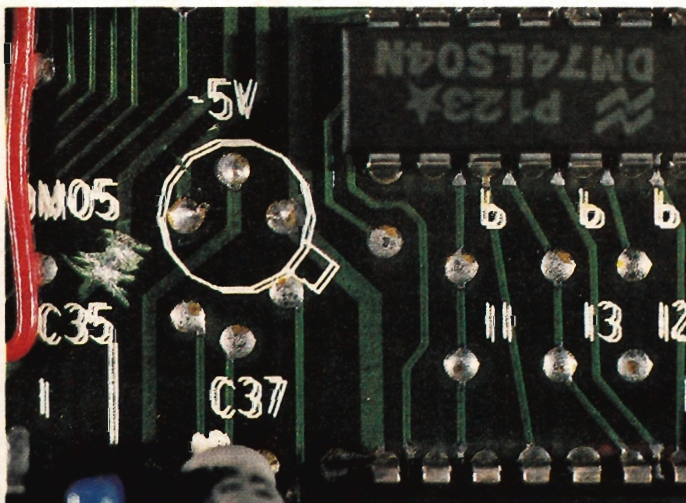
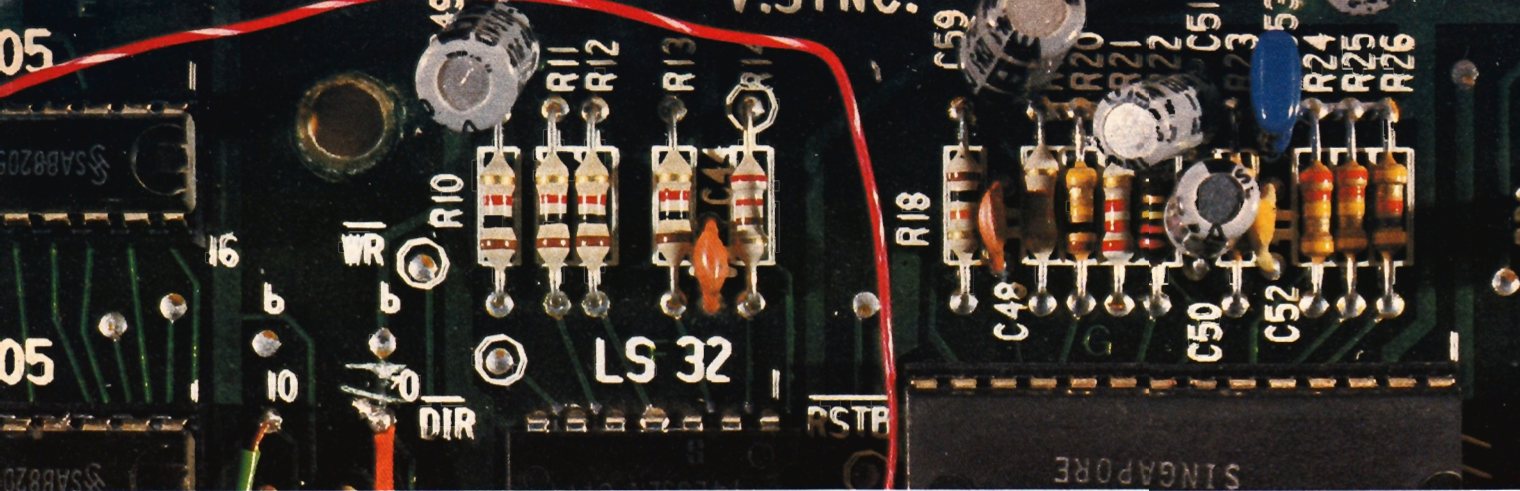


Bild 3. I ringen i mitten har —5V-regulatorn sittit i 3 st ben. Strax till vänster om den är den kapade +12V-ledningen till RAM-kretsarna (under femman i 79M05).



BYGG SJÄLV



ABC 80 till 64 kbyte

Av Christer Ekman

DEL 2

Många nya och spännande applikationer öppnar sig när man bygger ut sin ABC 80 till 64K-RAMminne. I förra avsnittet beskrevs hårdvarubiten av utbyggnaden och i det här avsnittet beskrivs ett av de många användningsområdena, ändringar och förbättringar av ABC 80:s BASIC.

Den tredje och sista delen kommer i MD nr 1/1984 och handlar om hur man anpassar ABC 800:s BASIC till ABC 80.

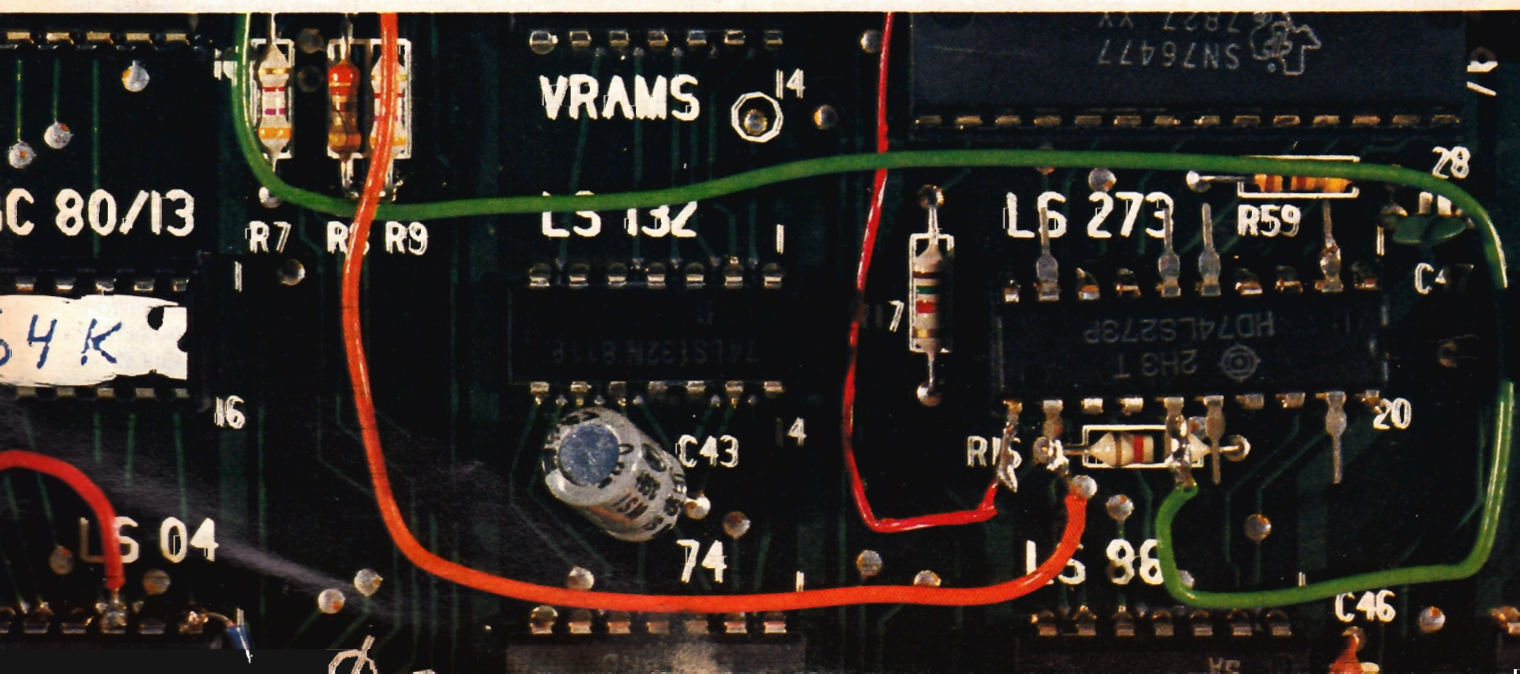
Ett mycket intressant och spännande område att tränga in i är hur BASIC-tolken fungerar. De kunskaper man skaffar sig när man flyttar över tolken till RAM-minne gör att man kan gå in och göra ändringar och förbättringar efter eget behag.

Här visas några användbara ändringar. Tyvärr finns det inte utrymme att ge några utförliga

förklaringar till funktionen utan det blir mer färdiga lösningar.

Ett bra hjälpmedel för att förstå funktionen i BASIC-tolken är disassembleringen med kommentarer utgiven av ABC-klubben (Vidängsv. 1, 161 33 Bromma) där de flesta och viktigaste funktionerna finns beskrivna och kommenterade.

Innan några ändringar kan göras måste innehållet i BASIC-



ROM:arna och DOS:et föras över till RAM-minnet. *Programmet i listning 1a*, TOLKMOVE. BAC, flyttar över innehållet i adresserna 0-31743 från minneskarta 0 till minneskarta 1. För att inte överföringen skall störas av exempelvis interrupt sker den i en assemblerrutin (*program 1b*) som stänger av interrupt. Dessutom går det givetvis mycket snabbare, cirka en sekund tar överföringen.

Programmen i listning 1a, 2, 3, 4, 5a, 6, 8, 9a kan sättas ihop till ett program för att få samtliga ändringar och förbättringar.

PROMPTÄNDRING

När BASIC-tolken väntar på att ett kommando skall matas in skrivs den välbekanta prompten "ABC80" ut först. Vill man ändra prompten till något annat visas i *program 2* ett exempel för det. Där innehåller p– den nya texten som skall skriva ut (lagras från adress 16430 och uppåt, strax efter BASIC:ens slut).

UTÖKAD FELHANTERING

Hur ofta har man inte irriterat sig på att ERROR 62 "Dotadress utanför skärmen" eller liknande fel inte går att hantera med ONERRORGOTO? I BASIC-tolken ligger en tabell med de fel som skall kunna hanteras med ONERRORGOTO.

Flyttas den till ett annat ledigt ställe i minnet kan man själv avgöra vilka fel som kan hanteras.

I *program 3* har felen 62, 48 och 9 lagts till och tabellen är flyttad till adress 16630. Observera att felkoderna ligger lagrade +128 (error 9 lagras som 137 t ex), och att på rad 2170 anges hur många felkoder tabellen innehåller.

FÖRDUBBLAD KASSETHASTIGHET

Normalt ligger hastigheten på in- och utmatning av data till kassetbandspelaren på cirka 700 baud. Det ger en säker dataöverföring som oftast fungerar bra.

Ibland kan man ge avkall på säkerheten till fördel för hastigheten. Farten på in- och utmatning bestäms av endast tre stycken tidsfördröjningsloopar.

```
10 REM SAVE TOLKMOVE.BAC 830704 C.E
20 REM * FLYTTAR ADDRESS 0 TILL 31744
30 REM * TILL MOTSVARANDE ADDRESS I
40 REM * RAM MINNET
50 REM
60 POKE 65408%,243%,118%,33%,0%,0%,1%,124%,0%,62%,0%,211%,7%,86%,62%,1%,211%
70 POKE 65424%,7%,114%,35%,16%,243%,13%,32%,240%,201%
80 ZZ=CALL(65408%)
```

Program 1a.

```
0000 ;TSAVE TOLKMOVE.TXT
0000 ;*****
0000 ; ASSEMBLER RUTIN FÖR TOLKMOVE
0000 ; FLYTTAR BASIC TOLK+DOS IN I RAM-
0000 ; MINNET
0000 ;
FF80 ; ORG 65408
FF80 F3 ; DI
FF81 76 ; HALT ;VANTA PA NMI
FF82 210000 ; LD HL,0
FF85 017C00 ; LD BC,124 ;31744 GGR
FF88 3E00 ; LOOP LD A,0
FF8A D307 ; OUT (7),A ;MINNESKARTA 0
FF8C 56 ; LD D,(HL) ;HANTA EN BYTE
FF8D 3E01 ; LD A,1
FF8F D307 ; OUT (7),A ;MINNESKARTA 1
FF91 72 ; LD (HL),D ;SPARA EN BYTE
FF92 23 ; INC HL
FF93 10F3 ; DJNZ LOOP
FF95 0D ; DEC C
FF96 20F0 ; JR NZ,LOOP
FF98 C9 ; RET
0000 ; END
```

Program 1b.

```
1400 REM *****
1410 REM ANDRA PROMPTEN "ABC80" TILL PR
1420 REM
1430 PR=CHR$(13,10)+'READY'
1440 FOR IX=1% TO LEN(PR)
1450 POKE IX+16430%,ASC(MID$(PR,IX,1%))
1460 NEXT IX
1470 POKE 211%,16430%,16430%/256%
1480 REM
```

Program 2.

```
2000 REM *****
2010 REM ANDRA SÅ ATT FLER FEL KAN HANTERAS
2020 REM AV ONERRORGOTO
2030 REM FELEN SOM SKALL HANTERAS
2040 REM LAGRAS FRÅN ADDRES 16630
2050 REM OBS FELKODEN LIGGER LAGRAD +128
2060 REM
2070 REM
2080 REM DE ORDINARIE FELEN
2090 POKE 16630,135,132,140,158
2100 POKE 16634,162,147,149,163
2110 POKE 16638,164,165,166,167
2120 POKE 16642,168,169,170,171
2130 REM
2140 POKE 16646,128+62,128+48,128+9
2150 REM
2160 REM
2170 POKE 1591,16630,16630/256
2180 POKE 1594,0,19 ; REM ANTAL FEL SOM KAN HANTERAS
2190 REM
```

Program 3.



```

700 REM *****
710 REM ÖKA KASSETT INSPELNING OCH AV-
720 REM SPELNINGSHASTIGHET
730 REM KONSTANTEN K BESTÄMMER FARTEN
740 REM NORMALVÄRDE = 150
750 REM
760 K=70
770 IF PEEK(3433Z)<>42Z THEN 820 : REM CHECKSUMMA EJ11273
780 POKE 1144Z,K
790 POKE 1153Z,K
800 POKE 1439Z,K*4/5
810 GOTO 899
820 POKE 1154Z,K
830 POKE 1163Z,K/2
840 POKE 1439Z,K*4/5
899 REM
    
```

Program 4.

```

500 REM *****
510 REM ÄNDRA REALTIDSKLOCKAN SÅ ATT
520 REM DEN RÄKNAR UPP 1 GANG PER SEKUND
530 REM (ADRESS 65009,65010)
540 REM *****
550 REM
560 REM LÄGG IN NYA ASSEMBLER PROG.
570 REM (SE KÄLLKOD NMI50.TXT)
580 POKE 16384Z,245Z,58Z,240Z,253Z,61Z,40Z,6Z,50Z,240Z,253Z,241Z,237Z,77Z,197Z,2
37Z,75Z
590 POKE 16400Z,241Z,253Z,3Z,237Z,67Z,241Z,253Z,62Z,50Z,193Z,24Z,235Z
600 POKE 65408Z,243Z,118Z,62Z,195Z,50Z,102Z,0Z,33Z,0Z,64Z,34Z,103Z,0Z,251Z,201Z
610 ZZ=CALL(65408Z) : REM INITIERA
620 REM
    
```

Program 5b.

```

0000 ;TBAVE NMI50.TXT
0000 ;*****
0000 ; RÄKNAR UPP REALTIDSKLOCKAN 1 GANG
0000 ; PER SEKUND.
0000 |
4000 ORG 16384
4000 F5 PUSH AF
4001 3AF0FD LD A,(65008)
4004 3D DEC A
4005 2806 JR Z,COUNTUP
4007 32F0FD SLUT LD (65008),A
400A F1 POP AF
400B ED4D RETI
400D C5 COUNTUP PUSH BC
400E ED4BF1FD LD BC,(65009)
4012 03 INC BC
4013 ED43F1FD LD (65009),BC
4017 3E32 LD A,50 ;50 TICK/SEKUND
4019 C1 POP BC
401A 18EB JR SLUT
401C ;*****
401C ; INITIERINGSROUTIN
401C |
FF80 ORG 65408
FF80 F3 DI
FF81 76 HALT ;VANTA PÅ NMI
FF82 3EC3 LD A,195 ;INSTRUKTION JUMP
FF84 326600 LD (102),A ;BÖRJAN AV NMIRUTIN
FF87 210040 LD HL,16384 ;ADRESS TILL RESTEN AV NMIRUTINEN
FF8A 226700 LD (103),HL
FF8D FB EI
FFBE C9 RET
0000 END
    
```

Program 5a.

ger på olika ställen i olika varianter beroende på vilken tolk (annan checksumma) kollar programmet på rad 770 vilken variant det är.

REALTIDSKLOCKAN

Realtidsklockan i ABC 80 består av en pulsgenerator som ger en puls 50 gånger per sekund till en ingång på CPU:n, Non Maskable Interrupt. Vid varje puls avbryter CPU:n det den arbetar med för att köra en interruptrutin på adress 102.

Där räknas innehållet i adresserna 65008-65010 ner med ett.

Därefter återupptar CPU:n det den höll på med.

Att ta reda på innehållet i adresserna 65008-65010 och använda resultatet är ganska krångligt. Ett naturligare sätt vore att innehållet räknades upp 1 gång per sekund.

I program 5a har interruptrutinen flyttats till adress 16384 och ändrats till att räkna upp innehållet i 65009, 10 1 gång per sekund.

För att ta reda på antalet sekunder skriver man "PRINT PEEK(65009) + PEEK(65010) × 256". Här skulle man kunna lägga in någon annan assembler-

snutt som då skulle utföras ”i bakgrunden” 50 gånger per sekund.

**INMATNINGS-
RUTINEN**

Inmatningsrutinen på adress 426 matar in en textsträng från tangentbordet. Den är ganska lätt att följa och kan rekommenderas att studera. En ändring som kan göras här framgår av *program 6*.

Efter ändringen kan även kontrolltecken direkt matas in i strängar och kommandon. Vill man mata in tecknet för start av grafik (ASCII-kod 23) trycker man C-Q (Control-tangenten och Q samtidigt).

Ett exempel är om man önskar grafik i REM-satser: 10 REM C-Q 0123456789 ger en rad med grafiktecken för 0, 1, 2, 3... På liknande sätt kan C-L (töm skärmen), C-G (Beep) och C-X (töm raden) direkt matas in i printsatser, strängar eller REM-satser.

**SVENSKA
INSTRUKTIONER**

För att dels lista och för att kompilera inmatade rader finns det ett antal texttabeller i tolken vilka innehåller alla instruktioner och kommandon. Tabellerna är uppbyggda på följande sätt: först en byte = 128 följt av ASCII-tecknen för första instruktionen.

Efter det följer en byte = 129 med nästa instruktion osv till slutet som indikeras med en byte = 255. Det finns sedan en rutin på adress 68 som letar efter en text i en sådan här tabell och returnerar värdet av den inledande byten. Resultatet används ofta för att slå i en hopptabell över adresserna till instruktionerna.

Med programmet KOMREP. BAC i lista 7 visas alla kommandon och instruktioner tillsammans med inledande byte och startadress. Byter man ut ASCII-tecken i de här tabellerna kommer man döpa om instruktionerna.

I *program 8* t ex har några instruktioner blivit omdömda till svenska varianter.

Ett steg vidare vore att utöka tabellerna med egna instruktioner men det kräver stora kunskaper i både assemblerprogrammering och tolkens funktion.

```
1100 REM *****
1110 REM ÄNDRA SÅ ATT KONTROLLTECKEN KAN
1120 REM MATAS IN DIREKT FRÅN TANGENT-
1130 REM BORDET
1140 REM
1150 POKE 722%,0%,0% : REM ERSÄTT JR C, MED 2 ST NOP
1160 REM
```

Program 6.

```
10 REM SAVE "KOMREP.BAC" 830909 CE
20 REM LISTA TILLGÄNGLIGA KOMMANDON
30 REM OCH INSTRUKTIONER
40 REM
41 ; 'FILNAMN: (RETURN=SKÄRM)';
42 INPUT FR
43 IF FR=' ' FX=0% ELSE FX=1%
44 OPEN FR ASFILE FX
50 ; CHR$(12)
51 ; #FX,'ADRESS', ' NR', 'KOMMANDO'
60 ; #FX,'=====',' ==','=====';
70 IF PEEK(3433)=42 THEN 180
80 A=2592 : GOSUB 280
90 A=6415 : GOSUB 280
100 A=6513 : GOSUB 280
110 A=A+1 : GOSUB 280
120 A=A+1 : GOSUB 280
130 A=A+1 : GOSUB 280
140 A=A+1 : GOSUB 280
150 A=A+1 : GOSUB 280
160 A=A+1 : GOSUB 280
170 END
180 A=2592 : GOSUB 280
190 A=6417 : GOSUB 280
200 A=6515 : GOSUB 280
210 A=A+1 : GOSUB 280
220 A=A+1 : GOSUB 280
230 A=A+1 : GOSUB 280
240 A=A+1 : GOSUB 280
250 A=A+1 : GOSUB 280
260 A=A+1 : GOSUB 280
270 END
280 IF FX=1% THEN 290
281 IF PEEK(65011)=23 THEN GET R# : REM SISTA RADEN ?
290 ; #FX : ; #FX,A,PEEK(A),
300 A=A+1
310 IF PEEK(A)=255 THEN 350 : REM SLUT PÅ TABELL
320 IF PEEK(A)>127 THEN 280 : REM NÄSTA INSTR.
330 ; #FX,CHR$(PEEK(A));
340 GOTO 300
350 ;
360 RETURN
```

Program 7.

```
1200 REM *****
1210 REM ÄNDRA INSTRUKTIONSNAMN TILL SVENSKA NAMN
1220 REM
1230 REM
1240 POKE 2597,ASC('K'),ASC('Ö'),ASC('R')
1250 POKE 2606,ASC('N'),ASC('Y'),ASC('T'),ASC('T'),131
1260 POKE 6416,ASC('L'),ASC('Ä'),ASC('T')
1270 POKE 6425,ASC('S'),ASC('K'),ASC('R'),ASC('I'),ASC('V')
1280 POKE 6433,ASC('O'),ASC('M')
1290 POKE 6452,ASC('F'),ASC('Ö'),ASC('R')
1300 POKE 6499,ASC('P'),ASC('Ä')
1310 POKE 6585,ASC('S'),ASC('T'),ASC('Ä'),ASC('N'),ASC('G')
1320 POKE 6638,ASC('D'),ASC('Ö'),ASC('D'),ASC('A')
1330 POKE 6648,ASC('S'),ASC('P'),ASC('Ä'),ASC('R'),ASC('A')
```

Program 8.



TANGENTBORDS- KLICK

När en tangent trycks ner på ABC 80:s tangentbord genereras ett interrupt och CPU:n hoppar till en rutin som sätter en flagga att en tangent har tryckts och kollar även om det är kontrol-C som har tryckts.

Här kan man ändra så att även andra uppgifter införs. I *program 9a* läggs ett "klick" till i högtalaren vid varje nedtryckning. Ljudet och dess längd kan ändras på rad 1690 och 1695.

```

1600 REM *****
1610 REM LÄGG IN ETT "KLICK" VID NED-
1620 REM TRYCK AV TANGENT
1630 REM
1640 POKE 16530%,245%,229%,62%,70%,211%,6%,33%,0%,4%,43%,124%,181%,32%,251%,175
,211%
1650 POKE 16546%,6%,219%,56%,254%,131%,32%,3%,50%,7%,254%,50%,245%,253%,62%,70%
50%
1660 POKE 16562%,247%,253%,225%,241%,251%,237%,77%
1670 POKE 798%,195%,16530%,16530%/256%
1680 REM
1690 POKE 16533%,67% : REM LJUD 67 T EX
1695 POKE 16538%,12% : REM LÄNGDEN PÅ LJUDET
1700 REM
    
```

Program 9a.

OLIKA CHECKSUMMOR

Som bekant finns det 3 olika varianter på BASIC-tolken med checksummorna 9913, 10042, 9913. Funktionen är nästan helt identisk men några adresser har ändrats. Det kan vålla problem för assemblerprogram som använder rutiner i tolken när det körs på en maskin med en annan checksumma.

För att utprova programmen är man tvungen att byta ut BASIC-ROM:arna eller att byta maskin. Nu när man har RAM-minne i botten kan man istället ladda in en tolk med annan checksumma från en fil.

Programmet "TOLKSAVE. BAC" i *lista 10* sparar en tolk på enklast möjliga sätt på en fil T9913, T10042 eller T11273. Lagringen sker genom att varje byte tolkas som ett ASCII-tecken och skrivs in med randomaccess. Ingen kontroll av innehållet sker men eftersom det är mycket data sparas mycket plats.

Inladdning av en tolk sker med programmet LOADTOLK. BAC i *lista 11*. Där laddas först hela filen in från och med adress 32768 (det finns ju 32 kbyte programminne efter utbyggnaden). Sedan flyttas data med ett assemblerprogram till adress 0.

Överföringen måste ske i assembler för att inte störas av interrupt och därför att BASIC:en inte kan användas när den håller på att laddas in. Källkoden för *assemblerprogrammet* finns i *lista 12*. Samma program kommer användas för att ladda in ABC 800:s BASIC i nästa avsnitt.

```

0000      ;TSAVE KLINK.TXT
0000      ;*****
0000      ;INTERRUPTROUTIN MED KLICK
0000      ;*****
4092                ORG 16530
4092 F5                PUSH AF
4093 E5                PUSH HL          ;SPARA REGISTER
4094 3E46              LD A,46H        ;LJUD
4096 D306              OUT (6),A
4098 210004            LD HL,1024      ;LÄT ETT TAG
409B 2B      DELAY    DEC HL
409C 7C                LD A,H
409D B5                OR L
409E 20FB              JR NZ,DELAY
40A0 AF                XOR A          ;SLUTA LÄTA
40A1 D306              OUT (6),A
40A3 DB3B              IN A,(56)      ;LÄS TANGENTBORD
40A5 FE83              CP 131        ;KOLLA OM CONTROL-C
40A7 2003              JR NZ,EJCC
40A9 3207FE            LD (65031),A    ;MARKERA CONTROL-C
40AC 32F5FD  EJCC     LD (65013),A    ;MARKERA TANGENTTRYCK
40AF 3E46              LD A,70        ;SÄTT "LÄNG" TID I TIMER
40B1 32F7FD            LD (65015),A    ;FÖR TANGENTBORD
40B4 E1                POP HL          ;HÄNTA REGISTER
40B5 F1                POP AF
40B6 FB                EI
40B7 ED4D              RETI
0000      END
    
```

Program 9b.


```

10 REM SAVE"VOLKSAVE.BAC" CE 830708
20 REM *****
30 REM * SPARAR EN BASICDOLK SOM *
40 REM * EN FIL *
50 REM *****
60 REM
70 DIM TX=253
80 ; 'VILKEN TOLK AR DETTA (11273,10042,9913) ' ; INPUT AR
90 REM
100 PREPARE 'T'+AR ASFILE 1
110 QOR=SPACE(253)
120 Z=CALL(28666,1)
130 Z=CALL(28670,0)
140 FOR I=0 TO 16444 STEP 253
150 TX=''
160 FOR A=0 TO 252
170 TX=TX+CHR(PEEK(I+A))
180 NEXT A
190 QOR=TX
200 Z=CALL(28666,1)
210 Z=CALL(28670,1/253+1)
220 NEXT I
230 CLOSE 1
240 END
    
```

Program 10.

```

10 REM SAVE"LOADDOLK.BAC" 830806 CE
20 REM * LADDAR IN EN ABC80 TOLK MED ANANN CHESKSUMMA I RAM-MINNET
30 REM *
40 REM *
50 REM *
60 REM
70 REM * KONTROLLERA ATT BOFA AR UPP-
80 REM * FLYTTAD SA ATT DET FINNS MINNNE
90 REM * ATT MELLANLAGRA 16384 BYTES
100 IF PEEK(65052)+PEEK(65053)*256<>57600 THEN POKE 65052,57600,57600/256 : CHAI
N 'LOADtolk.bac'
110 REM
120 PRINT CHR(12%)'VILKEN TOLK (11273,10042,9913) ' ; INPUT AR
130 OPEN 'T'+AR ASFILE 1
140 REM
150 REM * FÖRBERED FILEN FÖR DIREKTACCESS
160 PRINT CHR(12%)' ***** LADDAR BASICDOLK 'AR' *****'
170 REM
180 Z=CALL(28666,1)
190 REM
200 FOR IX=0% TO 16444% STEP 253%
210 ; CUR(5%,15%)IX;
220 Z=CALL(28668%,IX/253%+1%) ; REM LAS 253 BYTES TILL QOR
230 REM
240 FOR AX=0% TO 252%
250 TX=MID(QOR,AX+1%,1%)
260 POKE AX+IX+32768%,ASC(TX) ; REM MELLAN LAGRA I ÖVRE MINNET
270 NEXT AX
280 REM
290 NEXT IX
300 REM
310 CLOSE 1%
320 REM
330 REM * NU LIGGER BASICDOLKEN FRÅN
340 REM * ADRESS 32768 OCH UPPÅT
350 REM
360 REM * FLYTTA NER TOLKEN TILL ADRESS 0
370 POKE 65408%,33%,0%,128%,17%,0%,0%,1%,0%,64%,62%,1%,243%,118%,211%,7%,237%
380 POKE 65424%,176%,199%
390 ZX=CALL(65408%)
    
```

Program 11.

Program 12.

```

0000
0000 ;TSAVE BLCKMOVE.TXT CE 830809
0000 ; FLYTTAR 16384 BYTES FRÅN ADRESS 32768 TILL 0 OCH KÖR
0000 ; FRÅN ADRESS 0 ;
FFB0 ORG 65408
FFB0 210080 LD HL,32768
FFB3 110000 LD DE,0
FFB6 010040 LD BC,16384
FFB9 3E01 LD A,1 ;MINNESKARTA 1-
FFBB F3 DI ;STÅNGA AV INTERRUPT
FFBC 76 HALT ;VÄNTA PÅ NMI
FFBD D307 OUT (7),A
FFBF EDB0 LDIR
FF91 C7 RST 0 ;EXEKVERA FRÅN ADRESS 0
0000 END
    
```

Har du några intressanta applikationer som andra kan vara intresserade av? Skriv i så fall gärna och berätta. Adressen är:
 Christer Ekman
 Mikrodatorn
 Värtavägen 55
 115 38 Stockholm
 Märk kuvertet "ABC80 applikationer".

BYGG SJÄLV



ABC80 till 64kbyte

Av Christer Ekman

Som en final på utbyggnaden av ABC80 visas här programvara till 64Kbyte-minnet för att köra ABC800:s BASIC i ABC80. Fördelarna behöver väl kanske inte nämnas men exempelvis autonummering, ökad räknegranthet, långa variabelnamn är några av dem. Dessutom ökar tillgången på programvara. Den "gamla" ABC80:n blir alltså ett strå vassare.

Den hårdvarumässiga biten av ombyggnaden beskrevs i MD 9/83, ändringar och förbättringar av ABC80:s BASIC i MD 10/83.

I korthet går det till så att man kör programmet "SAVEDTC.BAC" på en ABC800. Programmet sparar innehållet i BASIC:en på en fil på diskett som sedan kan laddas in på en ombyggd ABC80 med programmet "LOADDTC.BAC". Alla ABC800:s instruktioner kan användas förutom de som rör ABC800:s speciella in/utenheter (t ex grafik) och kassett- och printerrutiner. Vidare får man också se upp med program som använder PEEK och POKE då de ibland ställer om t ex interna pekare för BASIC-tolken.

Program 1, "SAVEDTC.BAC" lägger upp två datafiler på

en diskett, "DTCTOLK.DTA" och "DOS800.DTA" vilka innehåller BASIC-tolken respektive DOS. Datorn som programmet körs på skall vara en ABC800 eller FACIT DTC med 80 teckensskärm, dock ej av den allra tidigaste modellen. DOS:et behöver inte vara originalet utan kan vara s k SuperDos som klarar 8-tums floppy och Winchester. När programmet körs, se då till att det är samma format på floppyn som på ABC80:n ifråga om densitet och antal spår.

Programmet "LOADDTC.BAC" finns i två olika varianter för ABC80 med och utan 80-teckens tillsats. Program 2a är för

```

10 REM "SAVEDTC.BAC" CE 830829
20 REM -----
30 REM PROGRAM SOM SPARAR ABC800-
40 REM TOLKEN PÅ DATAFIL "DTCTOLK.DTA"
50 REM -----
60 REM
70 PREPARE "DTCTOLK.DTA" AS FILE 1
80 FOR I%=0% TO 24575% : REM 24576 BYTES
90 PUT #I,CHR$(PEEK(I%))
100 NEXT I%
110 CLOSE I%
120 PREPARE "DOS800.DTA" AS FILE 1%
130 FOR I% = 0% TO 4095% : REM 4096 BYTES
140 PUT #I%,CHR$(PEEK(I%))
150 NEXT I%
160 CLOSE I%
170 END

```

Program 1. Sparar ABC800:s Basic och DOS på en datafil.

```

10 T$=SPACE$(253)
20 A2%=PEEK(65065%)+PEEK(65066%)*256%+6% : REM adress till t$
30 REM SAVE"LOADDTC.BAC" 830916 CE
40 REM * LADDAR IN EN DATAFILER
50 REM * DTCTOLK.DTA SOM INNEHÅLLER
60 REM * ABC800-TOLKEN OCH ÖVERFÖR DEN
70 REM * TILL RAMMINNET
80 REM *
90 REM *
100 REM *
110 REM
120 REM * KONTROLLERA ATT BDFÄ ÄR UPP-
130 REM * FLYTTAD SÅ ATT DET FINNS MINNNE
140 REM * ATT MELLANLAGRA 24576 BYTES
150 IF PEEK(65052)+PEEK(65053)*256<>57600 THEN POKE 65052,57600,57600/256 : CHA
N 'ALDADDTC.bac'
160 REM
170 OPEN 'DTCTOLK.DTA' ASFILE 1
180 REM
190 PRINT CHR$(12%)' **** LADDAR BASICOLK ABC 800 ****'
200 REM
210 REM * FÖRBERED FILEN FÖR DIREKTACCESS

```

Program 2a. Programmet laddar in ABC800:s BASIC på en ABC80 med 40-teckens skärm.

forts

40-tecken och 2b för 80-tecken. Programmet reserverar först plats för datafilen "DTCTOLK.DTA" mellan adress 32768 och 57600. Inladdning sker sedan med direktaccess och tar cirka 15 sekunder. Genom att ändra pekaren till T \bar{X} :s värde (rad 280) och sedan tilldela T \bar{X} värdet av QO \bar{X} sparas lite tid jämfört med metoden som användes i program 11 i MD10-83.

När inladdningen är färdig sker lite ändringar med POKE-satser vilka ändrar lite i tolken för att passa till ABC80. Efter det sker överföring av data till minneskarta 1 från adress 0 med hjälp av ett assemblerprogram vars källkod finns i MD10-83.

För att även DOS:et skall laddas in finns ett tillägg i program 4 som sätts efter program 2. Inladdningen av BASIC och DOS måste ske i två omgångar på grund av att den stora datamängden inte får plats i minnet samtidigt.

ÄNDRINGARNA

Tack vare ABC800:s och ABC80:s liknande placering av BASIC-tolk, DOS, bildminne och RAM i minneskartan är det inte speciellt mycket ändringar som krävs. De ändringar som behöver göras är de rutiner som har med in- och utenheter, t ex utskrift på skärm och inmatning från tangentbordet.

DOS:et kommer passa direkt eftersom samma typ av kontrollkort används. Om printer eller kassarutiner skall användas måste även de skrivas om men det kommer inte att visas här. En bakväg för att använda printer eller kassett är att lagra filen som skall skrivas på floppy för att sedan mata ut den i ABC80-mod. Övergång från ABC800 till ABC80 kan ske med raden POKE 65532,62,0,211,7; ; CALL(65532).

För att komma från ABC80-mod till ABC800-mod POKE 65532,62,1,211,7; ; CALL(65532).

För att tränga djupare in i hur BASIC-tolken fungerar för att göra ändringar eller anpassningar har man stor hjälp av en kommenterad disassemblering. Blackebergs Dataförening har tidigare gett ut en mycket omfattande sådan med utförliga kommentarer. Tyvärr är den slutsåld men en ny version kommer eventuellt göras senare (Blackebergs Dataförening, Wergelandsgatan 16-24, 161 58 Bromma).

```

220 Z=CALL(28666,1)
230 REM
240 FOR IX=0% TO 24575% STEP 253%
250 ; CUR(5%,15%)IX;
260 Z=CALL(28668%,IX/253%) : REM LÄS 253 BYTES TILL BÖR
270 REM
280 POKE A%,IX,SWAP%(IX+32768%)
290 T $\bar{X}$ =BÖR : REM ladda över 253 bytes
300 REM
310 NEXT IX
320 REM
330 CLOSE IX
340 REM
350 REM * NU LIGGER BASIC TOLKEN FRÅN
360 REM * ADRESS 32768 OCH UPPÅT
370 REM
380 REM *****
390 REM * BÖR ÄNDRINGAR SÅ ATT DET
400 REM * PASSAR I ABC80
410 A%=32768%
420 POKE A%+102%,237%,69%
430 POKE A%+168%,62%,176%,211%,57%,62%,183%,211%,57%,62%,127%,211%,57%,0%
440 POKE A%+1602%,201%
450 POKE A%+665%,62%,255%,211%,57%,211%,57%,24%,17%
460 POKE A%+690%,229%,197%,33%,0%,0%,34%,82%,255%,17%,0%,120%,1%,0%,8%,98%,107%
470 POKE A%+706%,54%,32%,19%,11%,237%,176%,193%,225%,201%
480 POKE A%+996%,0%,120%,0%,121%,0%,122%,0%,123%,0%,124%,0%,125%,0%,126%,0%,127%
490 POKE A%+1012%,80%,120%,80%,121%,80%,122%,80%,123%,80%,124%,80%,125%,80%,126%
,80%,127%
500 POKE A%+1028%,160%,120%,160%,121%,160%,122%,160%,123%,160%,124%,160%,125%,16
0%,126%,160%,127%
510 POKE A%+1044%,1%,80%,0%,221%,94%,0%,221%,86%,1%,61%,40%,14%,221%,110%,2%,221
%
520 POKE A%+1060%,102%,3%,237%,176%,221%,35%,221%,35%,24%,230%,221%,225%,195%,19
2%,2%
530 POKE A%+554%,175%,211%,6%,62%,131%,211%,6%,201%
540 POKE A%+427%,14%,80%,167%,201%
550 POKE A%+509%,24%
560 POKE A%+647%,229%,197%,221%,229%,221%,33%,228%,3%,62%,24%,195%,20%,4%
570 POKE A%+731%,229%,42%,82%,255%,125%,108%,38%,0%,41%,17%,228%,3%,25%,134%,35%
,102%
580 POKE A%+747%,111%,235%,225%,201%
590 POKE A%+900%,235%,203%,254%,251%,58%,226%,255%,135%,56%,25%,219%,56%,135%,56
%,5%,62%
600 POKE A%+916%,70%,50%,227%,255%,16%,254%,58%,227%,255%,61%,32%,232%,62%,8%,50
%,227%
610 POKE A%+932%,255%,24%,4%,175%,50%,226%,255%,203%,190%,219%,56%,230%,127%,193
%,225%,209%
620 POKE A%+948%,221%,225%,201%
630 POKE A%+963%,245%,229%,219%,56%,254%,131%,32%,9%,42%,133%,255%,125%,180%,40%
,2%,203%
640 POKE A%+979%,198%,205%,135%,1%,225%,241%,251%,237%,77%
650 POKE A%+991%,245%,219%,56%,24%,244%
660 POKE A%+358%,963%,963%/256%
670 REM *****
680 REM NU LIGGER TOLKEN FRÅN ADRESS
690 REM 32768 OCH UPPÅT
700 REM FLYTTA NER DEN TILL ADRESS 0
710 REM OCH EXEKVERA
720 POKE 65408%,33%,0%,128%,17%,0%,0%,1%,0%,96%,62%,1%,243%,118%,211%,7%,237%
730 POKE 65424%,176%,199%
740 Z%=CALL(65408%)

```



```

10 TX=SPACER(253X)
20 A2X=PEEK(65065X)+PEEK(65066X)*256X+6X : REM adress till tx
30 REM SAVE'LOADDTC.BAC' 830916 CE
40 REM * LADDAR IN EN DATAFILEN
50 REM * DTCTOLK.DTA SOM INNEHÅLLER
60 REM * ABC800-TOLKEN OCH ÖVERFÖR DEN
70 REM * TILL RAMMINNET
80 REM *
90 REM *
100 REM *
110 REM
120 REM * KONTROLLERA ATT BOFA ÄR UPP-
130 REM * FLYTTAD SÅ ATT DET FINNS MINNNE
140 REM * ATT MELLANLAGRA 24576 BYTES
150 IF PEEK(65052)+PEEK(65053)*256<>57600 THEN POKE 65052,57600,57600/256 : CHAI
N 'LOADDTC.bac'
160 REM
170 OPEN 'DTCTOLK.DTA' ASFILE 1
180 REM
190 PRINT CHR$(12X)' ***** LADDAR BASICTOLK ABC 800 *****'
200 REM
210 REM * FÖRBERED FILEN FÖR DIREKTACCESS
220 Z=CALL(28666,1)
230 REM
240 FOR IX=0X TO 24575X STEP 253X
250 ; CUR(5X,15X)IX;
260 Z=CALL(28668X,IX/253X) : REM LÄS 253 BYTES TILL BOX
270 REM
280 POKE A2X,IX,SWAPX(IX+32768X)
290 TX=BOX : REM ladda över 253 bytes
300 REM
310 NEXT IX
320 REM
330 CLOSE IX
340 REM
350 REM * NU LIGGER BASICTOLKEN FRÅN
360 REM * ADRESS 32768 OCH UPPÅT
370 REM
380 REM *****
390 REM * GÖR ÄNDRINGAR SÅ ATT DET
400 REM * PASSAR I ABC80
410 AX=32768X
420 POKE AX+102X,237X,69X
430 POKE AX+168X,62X,176X,211X,57X,62X,183X,211X,57X,62X,127X,211X,57X,0X
440 POKE AX+1602X,201X
450 POKE AX+665X,62X,255X,211X,57X,211X,57X,24X,17X
460 POKE AX+690X,229X,197X,33X,0X,0X,34X,82X,255X,17X,0X,124X,1X,0X,4X,98X,107X
470 POKE AX+706X,54X,32X,19X,11X,237X,176X,193X,225X,201X
480 POKE AX+996X,0X,124X,128X,124X,0X,125X,128X,125X,0X,126X,128X,126X,0X,127X,1
28X,127X
490 POKE AX+1012X,40X,124X,168X,124X,40X,125X,168X,125X,40X,126X,168X,126X,40X,1
27X,168X,127X
500 POKE AX+1028X,80X,124X,208X,124X,80X,125X,208X,125X,80X,126X,208X,126X,80X,1
27X,208X,127X
510 POKE AX+1044X,1X,40X,0X,221X,94X,0X,221X,86X,1X,61X,40X,14X,221X,110X,2X,221
X
520 POKE AX+1060X,102X,3X,237X,176X,221X,35X,221X,35X,24X,230X,221X,225X,195X,19
2X,2X
530 POKE AX+554X,175X,211X,6X,62X,131X,211X,6X,201X
540 POKE AX+427X,14X,40X,167X,201X
550 POKE AX+509X,24X
560 POKE AX+647X,229X,197X,221X,229X,221X,33X,228X,3X,62X,24X,195X,20X,4X
570 POKE AX+731X,229X,42X,82X,255X,125X,108X,38X,0X,41X,17X,228X,3X,25X,134X,35X
,102X

```

Program 2b. Programmet laddar in ABC800:s BASIC på en ABC80 med 80-teckens skärm.

I programlista 3a, 3b visas källkoden till de ändringar som gjorts i ABC800-tolken. Program 3a är för 40-teckens skärm och 3b för 80-teckens. En kort beskrivning av de olika momenten:

Adress Kommentar
102 Vid Non Maskable Interrupt sker ett hopp hit och här kan ett hopp till en interruptrutin göras. I ABC80 sker NMI 50 ggr per sekund.
168 Initiering av ABC80:s PIO som sköter in/utmatning från tangentbord, kassettbandspelare och printer.
620 I ABC800:s initiering av kassettrutiner sker flera Out och Inp som skulle

```

580 POKE AX+747X,111X,235X,225X,201X
590 POKE AX+900X,235X,203X,254X,251X,58X,226X,255X,135X,56X,25X,219X,56X,13
X,5X,62X
600 POKE AX+916X,70X,50X,227X,255X,16X,254X,58X,227X,255X,61X,32X,232X,62X,6
X,227X
610 POKE AX+932X,255X,24X,4X,175X,50X,226X,255X,203X,190X,219X,56X,230X,127X
X,225X,209X
615 POKE 948X+AX,221X,225X,201X
620 POKE AX+963X,245X,229X,219X,56X,254X,131X,32X,9X,42X,133X,255X,125X,180X,
2X,203X
630 POKE AX+979X,198X,205X,135X,1X,225X,241X,251X,237X,77X
640 POKE AX+991X,245X,219X,56X,24X,244X
660 POKE AX+358X,963X,963X/256X
670 REM *****
680 REM NU LIGGER TOLKEN FRÅN ADRESS
690 REM 32768 OCH UPPÅT
700 REM FLYTTA NER DEN TILL ADRESS 0
710 REM OCH EXEKVERA
720 POKE 65408X,33X,0X,128X,17X,0X,0X,1X,0X,96X,62X,1X,243X,118X,211X,7X,237X
730 POKE 65424X,176X,199X
740 ZX=CALL(65408X)

```

programmera om ABC80:s PIO.
665 Ytterligare initiering av PIO.
690 Rutinen för att tömma skärmen ersatts med ABC80:s. På adress 02BDH bestäms om 1024 eller 2048 tecken på skärmen skall fyllas med blanka.
996 Här ligger en tabell över adresserna till början av de 24 raderna i bildminnet. Adresserna är olika i 40- och 80-teckens versionen.
414H Rutinen för att "scrolla" en rad blir annorlunda p g a skillnaderna i ABC800 och ABC80:s videodel.
22AH ABC80:s mer avancerade ljudgenerator ligger

forts.

- 509 på port 6 istället för 5.
Aldrig färgskärm på ABC80.
- 731 Rutin för att leta rätt på var i bildminnet cursorn befinner sig (använder tabellen på adress 996).
- 900 I ABC800 sker inmatning från tangentbordet via en SIO (Serial In Out) seriellt. Rutinen ersätts med ABC80:s inmatning via PIO.
- 961 Även interruptrutinen som anropas vid tangentnedtryckning blir annorlunda.

Komponenter till utbyggnaden som beskrevs i del 1, MD9-83 kan köpas från MIKO Komponent AB, Box 1004, 126 10 Hägersten, tel 08-88 08 80.
Komponentsats med samtliga komponenter 534:—
Endast avkodarrom (40 eller 80 tkn) 136:—
Färdigmonterad utbyggnad 884:—
På samtliga priser tillkommer moms.

730 POKE 65424%,176%,62%,0%,211%,7%,201%
740 Z% = CALL(65408%)
750 OPEN "DOS800.DTA" AS FILE 1%
760 Z% = CALL(28666%,1%): REM Förbered för direkt access
770 FOR I%=0% TO 4095% STEP 253%
780 Z% = CALL(28668%,1%/253%)
790 POKE A2%,1% + 32768%,SWAP%(1% + 32768%)
800 T~~X~~ = QO~~X~~
810 NEXT I%
820 CLOSE 1%
830 POKE 65408%,33%,0%,128%,17%,0%,96%,1%,0%,16%,62%,1%,243%,118%,211%,7%,237%
840 POKE 65424%,176%,199%
850 Z% = CALL(65408%)

Program 4. Tillägg till program 2a och 2b för att ladda in DOS.

```

0000 ;TSAVE "ABC800.TXT" CE 030902
0000 ;*****
0000 ; ÄNDRINGAR FÖR ATT ABC800-TOLKEN
0000 ; SKA PASSA I ABC80 MED 64K-MINNE
0000 ;*****
0000 ;
0066 DRG 102
0066 ED45 RETN ;VID NMI SKA INGET HÄNDA
0068 ;
00AB DRG 168
00AB 3E80 LD A,0B0H ;INTERRUPTVEKTOR TANGENTBORD
00AA D339 OUT (57),A
00AC 3E87 LD A,10110111B ;INTERRUPTKONTROLL
00AE D339 OUT (57),A
00B0 3E7F LD A,01111111B ;INT. PÅ BIT7
00B2 D339 OUT (57),A
00B4 00 NOP
00B5 ;
0642 DRG 1602
0642 C9 RET ;INITIERA EJ KASSETTRUTINER

```

```

0643 ;
0299 DRG 665
0299 JEFF LD A,255 ;INITIERA TANGENTBORDET
029B D339 OUT (57),A ;CONTROLMODE PIO A
029D D339 OUT (57),A ;ALLA BITAR INPUT
029F 1811 JR FORMFEED ;ENDA INITIERING AV SKÄRM
02A1 ;
02B2 DRG 690
02B2 FORMFEED ;TÖM SKÄRMEN
02B2 E5 PUSH HL
02B3 C5 PUSH BC
02B4 210000 LD HL,0 ;CUR(0,0)
02B7 2252FF LD (65362),HL ;
02BA 11007C LD DE,31744 ;BILDMINNESSTART
02BD 010004 LD BC,1024 ;ANTAL TECKEN
02C0 62 BLANKL LD H,D
02C1 68 LD L,E ;HL=DE
02C2 3620 LD (HL),32 ;(HL)=SPACE
02C4 13 INC DE
02C5 08 DEC BC
02C6 EDB0 LDIR ;TÖM
02C8 C1 POP BC
02C9 E1 POP HL
02CA C9 RET
02CB ;
03E4 DRG 996
03E4 TBADDR ;ADRESSER TILL RADBÖRJAN I
03E4 ; ;BILDMINNET
03E4 00 DEFB 0
03E5 7C DEFB 124
03E6 80 DEFB 128
03E7 7C DEFB 124
03E8 00 DEFB 0
03E9 7D DEFB 125
03EA 80 DEFB 128
03EB 7D DEFB 125
03EC 00 DEFB 0
03ED 7E DEFB 126
03EE 80 DEFB 128
03EF 7E DEFB 126
03F0 00 DEFB 0
03F1 7F DEFB 127
03F2 80 DEFB 128
03F3 7F DEFB 127
03F4 28 DEFB 40
03F5 7C DEFB 124
03F6 AB DEFB 168
03F7 7C DEFB 124
03F8 28 DEFB 40
03F9 7D DEFB 125
03FA AB DEFB 168
03FB 7D DEFB 125
03FC 28 DEFB 40
03FD 7E DEFB 126
03FE AB DEFB 168
03FF 7E DEFB 126
0400 28 DEFB 40
0401 7F DEFB 127
0402 AB DEFB 168
0403 7F DEFB 127
0404 50 DEFB 80
0405 7C DEFB 124
0406 D0 DEFB 208
0407 7C DEFB 124
0408 50 DEFB 80

```

Program 3a. Källkoden till de ändringar som krävs i ABC800:s BASIC för att passa på ABC80. (40-teckens skärm.)

forts.

forts


```

0409 7D      DEFB 125
040A D0      DEFB 208
040B 7D      DEFB 125
040C 50      DEFB 80
040D 7E      DEFB 126
040E D0      DEFB 208
040F 7E      DEFB 126
0410 50      DEFB 80
0411 7F      DEFB 127
0412 D0      DEFB 208
0413 7F      DEFB 127
0414          SCROLL          ;Fortsättning pa scroll
0414 012B00   LD BC,40
0417 DD5E00   LD E,(IX+0) ;LÅG BYTE
041A DD5601   LD D,(IX+1) ;HÖG BYTE
041D 3D      DEC A      ;24 66R ?
041E 280E    JR Z,SCRSLUT
0420 DD6E02   LD L,(IX+2)
0423 DD6603   LD H,(IX+3) ;NÄSTA RAD
0426 EDB0    LDIR      ;KOPIERA RADEN
0428 D023    INC IX
042A DD23    INC IX
042C 18E6    JR SCROLL ;Fortsatt scrolla
042E          SCRSLUT
042E DDE1    POP IX
0430 C3C002   JP BLANKL ;Töm sista raden
0433          ;
022A          ORG 554
022A AF      BELL      XOR A
022B D306    OUT (6),A
022D 3E83    LD A,131 ;LJUDET TILL 6EN.
022F D306    OUT (6),A
0231 C9      RET
0232          ;
01AB          ORG 427
01AB 0E28    LD C,40 ;40TKN SKÄRM
01AD A7      AND A    ;CLEAR CARRY
01AE C9      RET
01AF          ;
01FD          ORG 509 ;SKRIV SOM MONOCHROME SKÄRM
01FD
01FD 18      DEFB 24 ;ALLTID JR
01FE          ;
0287          ORG 647
0287 E5      PUSH HL ;SCROLLA SKÄRMEN
0288 C5      PUSH BC
0289 DDE5    PUSH IX
028B DD21E403 LD IX,TBADDR ;TABELL MED ADRESSER
028F 3E18    LD A,24 ;24 RADER
0291 C31404   JP SCROLL ;Fortsatt där det finns utrymme
0294          ;
02DB          ORG 731
02DB          CURXY ;DE=ADRESS I BILDMINNET
02DB E5      PUSH HL
02DC 2A52FF   LD HL,(65362) ;CURSORPOSITION
02DF 7D      LD A,L ;KOLUMN
02E0 6C      LD L,H ;RADPOS
02E1 2600    LD H,0
02E3 29      ADD HL,HL ;HL=2*RADPOS.
02E4 11E403   LD DE,TBADDR
02E7 19      ADD HL,DE
02EB 86      ADD A,(HL) ;A=KOLUMN+LÅGA BYTEN AV POS
02E9 23      INC HL
02EA 66      LD H,(HL)
02EB 6F      LD L,A

```

```

02EC EB      EX DE,HL
02ED E1      POP HL
02EE C9      RET
02EF          ;
0384          ORG 900 ;ANDRINGAR I INCHAR
0384 EB      EX DE,HL ;HL=ADRESS I BOLDMINNET
0385 CBFE    SET 7,(HL) ;SÄTT PÅ BLINK
0387 FB      EI
0388 3AE2FF   TESTKEY LD A,(65506) ;FLAGGA FÖR TRYCKT TANG.
038B 87      ADD A,A
038C 3819    JR C,GETSLUT ;OM TANG. TRYCKT
038E DB38    IN A,(56) ;NEDTRYCKT TANGENT
0390 87      ADD A,A
0391 3805    JR C,LAB1 ;OM TANG. TRYCKT
0393 3E46    LD A,70 ;TIMER "LÅNG TID"
0395 32E3FF   LD (65507),A
0398 10FE    LAB1   DJNZ LAB1
039A 3AE3FF   LD A,(65507)
039D 3D      DEC A
039E 20EB    JR NZ,TESTKEY
03A0 3E08    LD A,8 ;TIMER "KORT TID"
03A2 32E3FF   LD (65507),A
03A5 1804    JR LAB2
03A7 AF      GETSLUT XOR A
03AB 32E2FF   LD (65506),A
03AB CBBE    LAB2   RES 7,(HL) ;STANG AV BLINK
03AD DB38    IN A,(56) ;NEDTRYCKT TANGENT
03AF E67F    AND 127
03B1 C1      POP BC
03B2 E1      POP HL
03B3 D1      POP DE
03B4 DDE1    POP IX
03B6 C9      RET
03B7          ;
03C3          ORG 963 ;INTERRUPT, TANGENTBORD
03C3 F5      PUSH AF
03C4 E5      PUSH HL
03C5 DB38    IN A,(56) ;LÅS AV TANGENTBORD
03C7 FE83    CP 3+128 ;CONTROL C ?
03C9 2009    JR NZ,EJCC
03CB 2AB5FF   LD HL,(65413) ;PEKARE TILL KONTROL C
03CE 7D      LD A,L ;KOLLA OM C-C SKA MARKERA
03CF B4      OR H
03D0 2802    JR Z,EJCC
03D2 CBC6    SET 0,(HL)
03D4 CDB701   EJCC   CALL 391 ;MARKERA TANGENTTRYCK
03D7 E1      POP HL
03D8 F1      INTSLUT POP AF
03D9 FB      EI
03DA ED4D    RETI
03DC          ;
03DF          ORG 991
03DF F5      PUSH AF
03E0 DB38    IN A,(56) ;LÅS AV TANGENTBORD
03E2 18F4    JR INTSLUT
0000          END

```

forts.


```

0000 ;TSAVE "ABC800.TXT" CE 830902
0000 ;*****
0000 ; ANDRINGAR FÖR ATT ABC800-TOLKEN
0000 ; SKA PASSA I ABC80 MED 64K-MINNE
0000 ;*****
0000 ;
0066          ORG 102
0066 ED45     RETN          ;VID NMI SKA INGET HANDA
0068          ;
00A8          ORG 168
00A8 3EB0     LD A,0B0H    ;INTERRUPTVEKTOR TANGENTBORD
00AA D339     OUT (57),A
00AC 3EB7     LD A,10110111B ;INTERRUPTKONTROLL
00AE D339     OUT (57),A
00B0 3E7F     LD A,01111111B ;INT. PÅ BIT7
00B2 D339     OUT (57),A
00B4 00       NOP
00B5          ;
0642          ORG 1602
0642 C9       RET          ;INITIERA EJ KASSETTRUTINER
0643          ;
0299          ORG 665
0299 3EFF     LD A,255    ;INITIERA TANGENTBORDET
029B D339     OUT (57),A  ;CONTROLMODE PID A
029D D339     OUT (57),A  ;ALLA BITAR INPUT
029F 1811     JR  FORMFEED ;ENDA INITIERING AV SKÄRM
02A1          ;
02B2          ORG 690
02B2          FORMFEED    ;TÖM SKÄRMEN
02B2 E5       PUSH HL
02B3 C5       PUSH BC
02B4 210000   LD HL,0          ;CUR(0,0)
02B7 2252FF   LD (65362),HL ;
02BA 11007B   LD DE,30720    ;BILDMINNESSTART
02BD 01000B   LD BC,2048     ;ANTAL TECKEN
02C0 62       BLANKL LD H,D
02C1 68       LD L,E          ;HL=DE
02C2 3620     LD (HL),32    ;(HL)=SPACE
02C4 13       INC DE
02C5 0B       DEC BC
02C6 EDB0     LDIR          ;TÖM
02C8 C1       POP BC
02C9 E1       POP HL
02CA C9       RET
02CB          ;
03E4          ORG 996
03E4          TBADDR      ;ADRESSER TILL RADBÖRJAN I
03E4          ;BILDMINNET
03E4 00       DEFB 0        ;(30720 för 80tkn)
03E5 78       DEFB 120
03E6 00       DEFB 0
03E7 79       DEFB 121
03E8 00       DEFB 0
03E9 7A       DEFB 122
03EA 00       DEFB 0
03EB 7B       DEFB 123
03EC 00       DEFB 0
03ED 7C       DEFB 124
03EE 00       DEFB 0
03EF 7D       DEFB 125
03F0 00       DEFB 0
03F1 7E       DEFB 126
    
```

forts.

```

03F2 00
03F3 7F      DEFB 0
03F4 50      DEFB 127
03F5 78      DEFB 80
03F6 50      DEFB 120
03F7 79      DEFB 80
03F8 50      DEFB 121
03F9 7A      DEFB 80
03FA 50      DEFB 122
03FB 7B      DEFB 80
03FC 50      DEFB 123
03FD 7C      DEFB 80
03FE 50      DEFB 124
03FF 7D      DEFB 80
0400 50      DEFB 125
0401 7E      DEFB 80
0402 50      DEFB 126
0403 7F      DEFB 80
0404 A0      DEFB 127
0405 78      DEFB 160
0406 A0      DEFB 120
0407 79      DEFB 160
0408 A0      DEFB 121
0409 7A      DEFB 160
040A A0      DEFB 122
040B 7B      DEFB 160
040C A0      DEFB 123
040D 7C      DEFB 160
040E A0      DEFB 124
040F 7D      DEFB 160
0410 A0      DEFB 125
0411 7E      DEFB 160
0412 A0      DEFB 126
0413 7F      DEFB 160
0414          DEFB 127
0414          SCROLL
0414 015000   LD BC,80      ;Fortsättning på SCROLL
0417 DD5E00   LD E,(IX+0)
041A DD5601   LD D,(IX+1)  ;LÄG BYTE
041D 3D       DEC A        ;HÖG BYTE
041E 280E     JR Z,SCRSLUT ;24 GGR ?
0420 DD6E02   LD L,(IX+2)
0423 DD6603   LD H,(IX+3)  ;NÄSTA RAD
0426 EDB0     LDIR        ;KOPIERA RADEN
0428 DD23     INC IX
042A DD23     INC IX
042C 18E6     JR SCROLL    ;Fortsätt SCROLLA
042E          SCRSLUT
042E DDE1     POP IX
0430 C3C002   JP BLANKL    ;Töm SISTA RADEN
0433          ;
022A          ;
022A AF      BELL
022B D306    XOR A
022D 3E83    OUT (6),A
022F D306    LD A,131
0231 C9      OUT (6),A
0232         RET
01AB          ;
01AB 0E50    ORG 554
01AD A7      XOR A
01AE C9      OUT (6),A
01AF         LD A,131
01FD        OUT (6),A
01FD         RET
01FD         ORG 427
01FD         LD C,80
01FD         AND A
01FD         ;BÖTKN SKÄRM
01FD         ;CLEAR CARRY
01FD         RET
01FD         ORG 509
01FD         ;SKRIV SOM MONOCHROME SKÄRM
    
```

Program 3b. Källkoden för ändringarna skiljer sig lite för 80-teckens skärmen. Resten av ändringarna är likadana som i 3a.

ABC80 med 80 tecken inbyggd

För ABC80 med 80-teckens skärm krävs det ett annorlunda avkodarrom för att bildminnet skall hamna på rätt plats. Minneskartorna kommer få ett lite annorlunda utseende vilket framgår av figur 2.

I minneskarta 0 har 2 kbyte bildminne lagts till på adress 22528 där bildminnet för 80-teckensskärmen normalt ligger.

För att kunna köra ABC800:s BASIC har bildminnet utökats till 2 kbyte från adress 30720 i minneskarta 1.

När ABC80:s BASIC körs i minneskarta 1 måste adresserna i tolken till bildminnet ändras eftersom annars kommer tolken försöka skriva från adress 22528 där det inte finns något bildminne. Adresserna som skall ändras ligger från adress 884 och pekar till början av de 24 raderna.

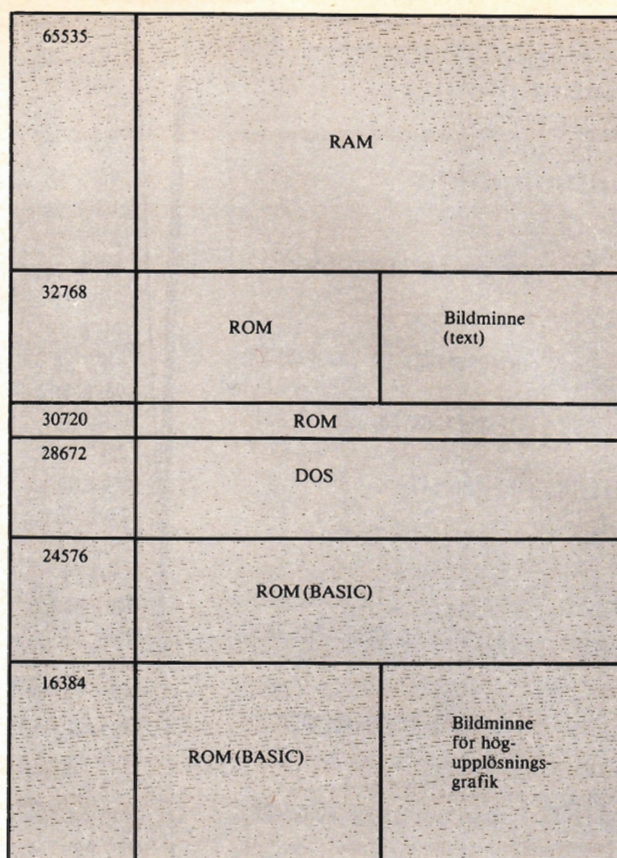
Med raden "POKE 884,0,120,0,121,0,122,0,123,0,124,0,125,0,126,0,127,80,120,80,121,80,122,80,123,80,124,80,125,80,126,80,127,160,120,160,121,160,122,160,123,160,124,160,125,160,126,160,127" (jämför tabellen i program 3b) kommer adresserna bli riktiga (lägg in raden på rad 8 i programmet "TOLKMOVE.BAC" i MD10-83).

ABC80 med 16 kbyte extraminne

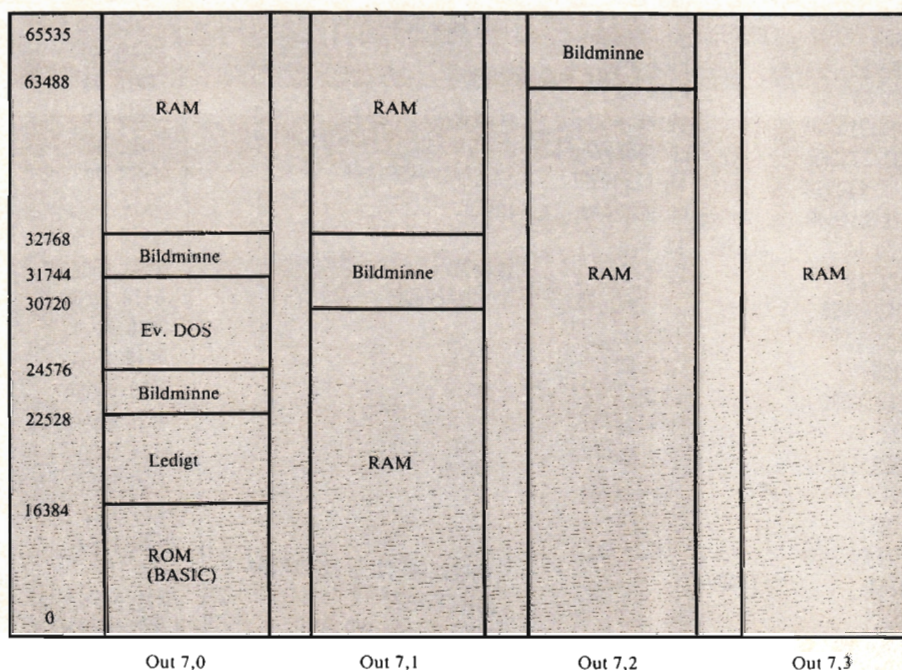
Om ABC80:n redan är utbyggd till 32 kbyte RAM måste utbyggnaden tas bort då det nya 64 kbyte RAM:et ersätter det. Är extraminnet ett kort på bussen är det bara att plugga ur det. Är det fast inmonterat i datorn kan det bli lite krångligare.

Det finns i huvudsak två olika varianter på utbyggnaden. Den ena är ett kretskort monterat i de gamla RAM-socklarna. Då kretskortets pinnar är tjockare än normala IC-kretsars är de gamla IC-hållarnas hål för stora. IC-hållarna måste lödas bort och bytas. Sedan tas de ändringar som gjordes vid monteringen bort så att ABC80:n blir som i originalutförande.

En annan variant på utbyggnad är ett kretskort som sitter i närheten av ljudgeneratoren 76477. I det fallet är det enklare att återställa ABC80:n i originalutförande genom att montera bort kortet. (Den bortmonterade utbyggnaden bör kunna användas till en annan ABC80.)



Figur 1. Minneskartan för ABC800, Facit DTC med 80 tecken monokrom skärm. Som synes är den mycket lik ABC80:s.



Figur 2. De fyra minneskartorna med det nya ROM:et för 80-teckens bildskärm. Platsen för bildminnet har utökats till 2Kbytes. (Jämför gärna med figur 1 i del 1.)

Rättelse

I del 1, MD 9-83, har det framkommit tre fel:

1. I testprogrammet på sid 60-61 har rad 9 ersatts med rad 10.

Rad 9 skall vara:

9 Data 1,6,16,221,33,0,124,54,0,126

2. +12V som kapas på kretskortet går till pinne 8 på krets D5 ej pinne 9.

3. Beskrivningen av det nya PROM:ets montering var en

miss. Där står att en av de pinnar som ska klippas bort är nummer 8 och att samma pinne i nästa andetag ska lödas fast igen. Hoppa över bortklippningen så slipper Ni onödigt arbete med fastlödning!