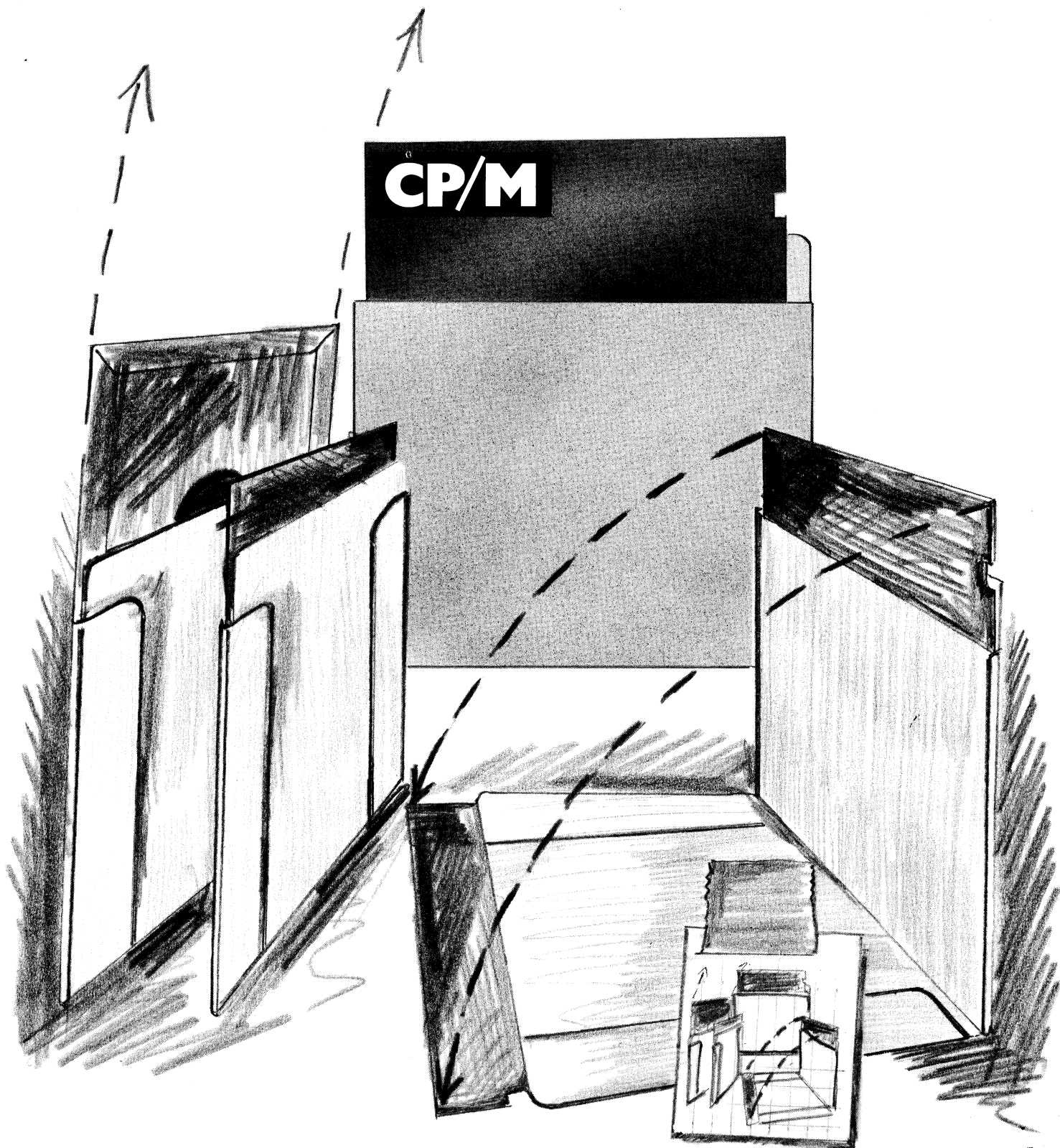


ABC BLADET

ABC-KLUBBENS MEDLEMSBLAD FÖR BLANDAD INFORMATION TILL BÅDE NYTTA OCH NÖJE

NUMMER 4 1984



PUAB Process & Underhåll AB

från design till idrifttagning

PUAB

Process & Underhåll AB
från design till idrifttagning

Vi söker kontakt med duktiga programmerare.
I första hand BASIC-ABC80/ABC800.
Även andra system kan komma ifråga.
Maskinspråk, styr och regler, hårdvara m m.

Skriv eller ring Christer Bolt
Fågelgatan 25, 149 00 Nynäshamn
0752-177 31, MBS-0047 50567, Bil 010-706620

OBS! OBS! OBS! OBS! OBS!
Fel telefonnummer
i tidigast annons.
OBS! OBS! OBS! OBS! OBS!

MODEM

Svensktillverkade och godkända av televerket.

Nu till introduktionspris.

TYP: 300/300 (även som pluggin för VIC-datorer)
300/300 + autosvar
300/300 + 1200/75 + autosvar +
en programmerbar kanal

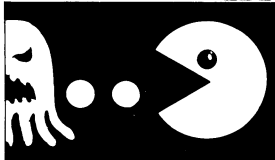
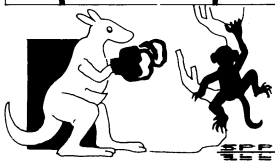
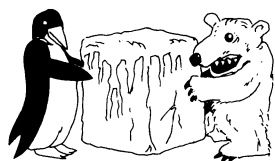
Dessutom har vi splitspeedkonvertrar för
terminal och datorände

Se även artikel på annan plats i detta nummer.

Skriv eller ring för information

SELIC AB

Box 44002, 400 76 Göteborg, Tel: 031/22 33 10



ALFA

Spelskivan för ABC-80 med
16 K-bytes minne.
Alfa innehåller marknadens
fyra tuffaste spel:
PINGVIN CITYDEFEND
KÄNGURU och EATER

Priset 235:- inkluderar
skiva eller kassett, utförliga
instruktioner samt porto
och postförskottsavgift.

Snabbhetspremie: Vid be-
ställning inom 10 dgr med-
sändes en "ljudplopp" utan
extra kostnad. Ljudploppen
förbättrar spelens ljudef-
fekter avsevärt.

Jag beställer st ALFA - Spelskiva för 235:-/st
Jag önskar ☐ skiva ☐ kassett

Namn

Adress

Postadress

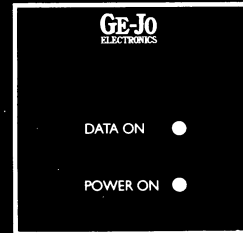
Kupongen sändes till:

Ken Verlage, Kåsättersvägen 7, 732 00 ARBOGA

NYHET!

ABC

SUPER-RAM



Dekal "ABC SUPER RAM" att klistera över ABC80 ingår.

128k-minnesutbyggnad med oanade möjligheter:

Supersnabb RAM-FLOPPY
Primärminne för maskinkod
Hela arbetsminnet i RAM
Emulerar ABC800-BASIC
Utbyggbar till 2 M-byte
Enkel montering i tangentbordet

Bara 2495:- inkl moms

Ring eller skriv för närmare information

GE-Jo
ELEKTRONIK

Box 30, 520 30 Ljung. Tfn 0513-50673

Medlemsorgan för ABC-klubben

Vidingsvägen 1
161 33 Bromma
ISSN 0349-3652

Ansvarig utgivare: Gunnar Tidner
Redaktör: Rune Mattsson
I redaktionen: Odd Rolander
Magnus Hedner
Claes Schibler

ABC-klubbens postgiron:
Medlemsavgifter: 15 33 36-3
Publikationer: 62 93 00-5
Q-Zentralen: 43 51 74-8
telefon 08-80 15 22
(automatisk telefonsvarare med aktuell
klubbinformation)
telefon 08-80 15 23
(modem med ABC Monitor)

Tryck: Märstattryck AB 1983

Lämnat till tryck 1 december 1984.

Nummer 4, 1983

INNEHÅLL

Omslagsillustration Rune Mattsson	
LEDAREN av Gunnar Tidner	4
Medlemsavgiften 1984	4
Vill du publicera i ABC bladet	4
KALLELSE till ABC-dagen med årsmöte 1984 i ABC-klubben	5
Valberedningens UPPROP	5
Programdistribution för ABC800 av Bo Kullmar	5
Forth på ABC80, del 2 (forts från föreg nr.) av Robert Claesson	6
Booles algebra för Svenne & Co av Sven Wickberg	12
UFD-doset av Bo Kullmar	13
Ny monitor av Monitorgruppen genom Bo Kullmar	13
ABC806 av Bo Kullmar	14
Systemvariabler i ABC800 av Carl-Axel Pettersson/ Bo Kullmar	15
ABC-Stockholm	16
Inläga: Kort och Brett om DEC10	
ABC-Kassetterna av Ulf Sjöstrand	17
CP/M till ABC80, ABC800/FASIT DTC av Odd Rolander	18
WordStar som ordbehandling av Odd Rolander	19
Överföring av CP/M-filer av Odd Rolander	20
"Bildern" av Odd Rolander	21
SuperCalc II som kalkylprogram av Odd Rolander	24
CP/M på ABC-datorer några synpunkter och erfaren- heter av Torbjörn Alm	25
DIV:	25
Annonssbilaga särtryck ur Mikrodatorn nr 7, 1983	28

Annonspriser fr o m nr 4, 1983

1/1-sida 185 × 260 mm	2.400:-
1/2-sida 185 × 128 mm, eller 90 × 260 mm	1.450:-
1/3-sida 185 × 85 mm, eller 60 × 260 mm	975:-
1/4-sida 90 × 128 mm	780:-
2 st 1/1-sidor i uppslag	5.400:-
2:a omslagssida	3.000:-
3:e omslagssida	2.875:-

4:e omslagssida 185 × 225 mm 3.300:-
Begärd placering 10% förhöjning.

Tidningen ansvarar ej för att införda programlistningar är
korrekta.

Särskild prislista vid best. av flera ex. tillhandahålls på
begäran.

Upphovsrätt gäller för införda program om inget annat anges.

Medlemsavgifter 1983

Seniorer 125 Skr
Juniorer 70 Skr

Junior räknas man t o m det kalenderår man
fyller 18 år. Ange därför personnummer
när Du betalar medlemsavgiften.
Medlemskapet är personligt. (Fysisk person.)
Särskild prislista för prenumeration
tillhandahålls på begäran.

Medlem blir Du enklast genom att sätta in
medlemsavgiften på ABC-klubbens
postgirokonto 15 33 36-3.

ABC-klubbens styrelse för 1983

Ordförande: Gunnar Tidner
Vice ordförande: Kjell-Åke Johansson
Kassör: Marianne Forsman
Sekreterare: Ulf Sjöstrand
Ledamöter: Joe Johnsson
Bo Kullmar
Stig Löfgren
Rune Mattsson
Suppleanter: Björn Sjöborg
Claes Schibler
Kalle Lindström

Postgirokonton:
Medlemsavgifter: 15 33 36-3
Publikationer: 62 93 00-5
Q-Zentralen: 43 51 74-8

LEDAREN.

Hur står det egentligen till med kom-pabiliteten inom ABC-familjen?

Ordet kompatibel kommer som så många andra från engelskan och kan närmast översättas med "förenlig". När det gäller data-utrustning är det givetvis önskvärt att program som utvecklats på en utrustning skall kunna köras på en annan utrustning och därvid ge samma resultat. För dator-program är det snarare regel än undantag att så inte är fallet när det är fråga om utrustning av olika fabrikat. Om man vill göra portabla program dvs sådana som med mycket små ändringar kan flyttas över till och köras i en helt annan dator, får man lov att programmera i ett språk som utvecklats med detta mål för ögonen t ex FORTH eller PASCAL, och i görligaste mån undvika rutiner som är maskinberoende.

BASIC finns i en mängd dialekter som skiljer sig tillräckligt mycket för att program inte blir portabla. På ABC-datorerna har vi BASIC I för ABC80 och BASIC II för ABC800. Program som är skrivna i ren BASIC I kan köras på alla förekommande modeller av ABC-80. Vi har dock fått många sk smarta program som utnyttjar olika rutiner i Basicolken. Det är inte säkert att dessa rutiner ligger på samma adresser i en modell med en annan checksumma. Inom ABC-klubben har det lagts ner ett omfattande arbete för att modifiera de program klubben distribuerar så att de skall kunna köras på alla förekommande varianter av ABC80 och utrustning (kassett/flexskiva/extra minne).

BASIC II har många trevliga egenskaper, t ex långa variabelnamn och flerradiga funktioner med möjlighet att använda lokala variabler. Det underlättar skrivandet av mer lättlästa och väl strukturerade program men det sker till priset av att kompatibiliteten går förlorad. Program skrivna i BASIC II med utnyttjande av dessa och andra finesser blir inte flyttbara till ABC80 med enbart BASIC I. Därför är det mycket angeläget att vi snarast möjligt får tillgång till BASIC II även på ABC80.

Vad gäller möjligheten att flytta ett program skrivet på ABC80 till ABC800 så är den inte så god som det ibland görs gällande. Ett nödvändigt villkor för portabilitet är att programmet inte gör anrop till rutiner i basitolken, inte pokar i systemvariablerna och undviker att använda sådana in- och utportar som är speciella för hårdvaran i ABC80.

Att programmet (som måste vara lagrat i BAS-format) kan laddas in i ABC800 utan att syntaxkontrollen ger något felmeddelande är inte en tillräcklig garanti för att det går att köra med avsedd funktion. Det kan uppstå tokigheter som beror på de små skillnader som finns i basitolknas sätt att arbeta. Skillnaderna kan ha att göra med anpassningen av Basic II till ANSI-standard, olika felkoder, nollställningen av variabler etc. Här gäller det att verkligen vara uppmärksam.

Det finns även vissa små skillnader i BASIC II mellan de olika modellerna: ABC800 M, ABC800 C, ABC802 och ABC806 som kan göra att ett program utvecklat på en modell inte kan köras på någon av de andra modellerna.

Under den tid som ABC800 funnits har vidare ändringar införts t ex när det gäller

den sk Poke-arean. Det har kommit en rad olika versioner både av DOS-et och det sk optionsprommet med printer/terminal-rutinerna.

Den mångfald av olika kombinationer som kan förekomma ute bland användarna skapar problem när det gäller att modifiera programmen så att de skall kunna köras på alla förekommande kombinationer. Bara uttestningen är nu ett svårlost problem. Det gäller ju att ha tillgång till hela mångfalden av utrustningar. Mycket talar för att vi tvingas ge avkall på ambitionerna och överlåta åt medlemmarna/användarna att själva göra de nödvändiga anpassningarna av programmen.

Glädjande nog finns kompatibiliteten kvar mellan ABC80 och ABC800 när det gäller datalagringen på flexskiva och kassett. Sättet att lagra filer har inte ändrats. Textfiler skrivna på ABC80 kan utan vidare läsas av ABC800. Det går bra även i andra riktningen om ABC800 begränsar radlängden till max 117 tecken i stället för 157. Kompatibiliteten gäller även filer i ABS-format och random access på flexskiva.

När det gäller olika flexskiveutrustning så är det så och så med kompatibiliteten. Den tekniska utvecklingen har gjort det möjligt att lagra alltmer på en flexskiva. Det har medfört att det nu finns en rad olika flexskiveformat, enkel eller dubbel densitet, enkel eller dubbelsidig lagring, enkel eller dubbel spårthet. ABC830, Data-disc 82 och 84, FD2D och FD4D använder alla 5 1/4 disketter och är sinsemellan kompatibla i den mening att det går att skriva filer på en utrustning som sedan kan läsas av en annan. Att sända en diskett är ett smidigt sätt att utbyta data mellan utrustningar på olika platser. På liknande sätt är ABC838 och Datadisc 88 kompatibla när det gäller 8 tums disketter. ABC832 använder också 5 1/4 disketter men i ett format som skriver med dubbel spårthet. Denna utrustning kan visserligen läsa en

diskett som är skriven med enkel spårthet, t ex av en ABC830. En diskett som är skriven på en ABC832 kan dock bara läsas av en utrustning som arbetar med dubbel spårthet. Det man vinner i förmåga att lagra mycket på en skiva forlorar man i minskade möjligheter till datautbyte med andra utrustningar. Tidigare har det vid bristande kompatibilitet mellan flexskiveformaten gått att använda kassett för datautbyte men den möjligheten är stängd när det gäller ABC806 som tyvärr saknar kassett-interface. Man blir i detta fall hänvisad till att gå över modem eller få hjälp med kopiering till annat format.

Vad gäller övrig kringutrustning så utgör printrarna ett sorgligt exempel på bristande kompatibilitet mellan olika fabriker och modeller. Det finns en etablerad standard när det gäller överföringsprotokoll och anslutning av printer till dator (V24 eller RS-232 för serieöverföring och det sk centronicsinterfacet vid parallellöverföring). Men när det sedan gäller att styra printerutskriften är det slut på standarden. Varje fabrikant tycks gå sin egen väg i valet av styrkoder för olika funktioner.

Gunnar Tidner

Medlemsavgiften 1984.

Ny rutiner kommer att gälla för inbetalning av medlemsavgift från och med 1984. Klubben kommer till varje medlem skicka ut en avi med inbetalningskort. Detta innebär att avgiften skall betalas först när inbetalningskortet kommer.

Vi har för avsikt att skicka ut inbetalningskorten omkring den 15 januari 1984. Inga medlemskort kommer att skickas ut mot postförsäkring.

Styrelsen kommer att föreslå årsmötet att medlemsavgiften skall vara betald senast den 15 mars (ändring av stadgarnas paragraf 5).

Medlemsavgiften för 1984 har styrelsen fått fullmakt av årsmötet att fastställa upp till ett tak på 140 Skr för seniorer och 80 Skr för juniorer.

Betala alltså medlemsavgiften när ett inbetalningskort kommer!

ABC-klubben/Styrelsen

Vill du publicera i ABC bladet.

Allt kan vara stoff för artiklar i ABC-bladet

- föredrag om smådatorer som Du har liggande
- ideer som Du vill sprida
- hur Du kom att intressera Dig för mikrodatorer
- hur Du använder Din ABC-dator
- hur man skulle kunna använda den i Ditt arbete

Låt oss andra ta del av Dina erfarenheter så lär vi oss och vissa misstag blir ej gjorda på nytt.

Hur Du går till väga?

Enklast möjligt. Kan Du använda TV-editorn är det utmärkt. Då kan Du redigera Din text med rubriker och annat.

Du kan lagra texten på en kassett som Du sänder oss. Kom då bara ihåg att spara texten två-tre gånger för att vi säkert

skall kunna läsa tillbaka texten.

Var generös med blanka rader. Det är lättare för oss att ta bort dem vid redigeringen än att vi måste sätta dit dem.

Dela raderna mellan ord och stava inte av med bindesstreck. Låt radernas höger kant bli ojämn. Det rättar vårt utskriftsprogram till. Vi är väldigt tacksamma om det inte är alltför mycket stavfel. Kan Du skicka med en utskrift på papper är det till stor hjälp.

Det är klart att Du får skicka en diskett, men förpacka den väl. Vi returnerar självklart kassetter och disketter så fort vi har hunnit göra en kopia. Sätt därför ut namn, medlemsnummer och adress ordentligt.

Du kan också lägga in texten på ABC-Monitorn så hämtar vi den där. Som sista utväg kan Du även höra av Dig till oss och vi tar över texten med hjälp av 300-bauds modem och FILTRANS.

KALLELSE

till ABC-dagen med årsmöte 1984 i ABC klubben.

Tid: Lördagen den 18 februari 1984

Plats: Brommasalen, kommunalhuset i Alvik
Gustavslundsvägen 168, 161 33 Bromma
T-banestation: ALVIK

I likhet med tidigare år räknar vi med att på ABC-dagen arrangera en utställning av utrustning och tillbehör. En del av leverantörerna kommer också att ha försäljning av enklare tillbehör och förbrukningsmaterial.

Utställningen kommer att vara öppen 9.30-13.00

Program

11.00 Utställningen öppnar

13.00 Årsmötesförhandlingar

14.30 (c:a) Paus för besök på utställningen

Om tid finnes följer efter årsmötesförhandlingarna en frågestund där du kan få tillfälle att ställa alla möjliga och omöjliga frågor till en expertpanel på ABC-datorer.

19.00 Gemensam middag följer för medlemmar och gäster Pris cirka 70 kr

Förslag till dagordning för
ABC-klubbens årsmötesförhandlingar 1984

1. Val av mötesordförande
2. Val av mötessekreterare
3. Val av två justeringsmän, tillika rösträknare att jämte mötesordförande justera årsmötesprotokollet
4. Fråga om mötet är behörigen utlyst
5. Fastställande av dagordning
6. Styrelsens redovisningshandlingar
7. Föredragning och godkännande av revisionsberättelse
8. Fråga om ansvarsfrihet för styrelsens medlemmar
9. Fastställande av balansräkning
10. Beslut med anledning av vinst eller förlust enligt balansräkningen
11. Fastställande av budget och medlemsavgift

12. Val av ordförande och vice ordförande samt övriga styrelseledamöter och suppleanter för ett år

13. Val av revisorer och en revisorssuppleant för ett år

14. Val av valberedning om minst två personer

15. Behandling av ärenden som styrelsen vill förelägga årsmötet

16. Behandling av motioner som medlemmarna inkommit med senast sex dagar före mötet

17. Övrig frågor

Motioner skall ha inkommit senast fredagen den 10 februari 1984 till

ABC-klubben
Vidängsvägen 1
161 33 Bromma

ABC-klubben/Styrelsen

VALBEREDNINGENS UPPROP.

Vid årsmötet 1983 utsågs undertecknade till valberedning, med uppgift att förbereda personvalen av klubbfunktionärer till årsmötet 1984.

Valberedningen vill ha in namn på personer villiga att arbeta i och för ABC-klubben och önskar kontakt med personer som är beredda att engagera sig i ABC-klubben.

Sammankallande

Dag Issjö
Hästhagsvägen 14
131 33 Nacka
08-716 68 22

Göran Sundqvist

Filippavägen 19
178 00 Ekerö
08-23 52 10 (dagtid)
0756-303 10 (kvällstid)

Magnus Hedner

Allevägen 80
191 77 Sollentuna
08-754 33 30

Programdistribution för ABC800.

Klubbens distribution av program sker främst på kassett. För ABC800 serien är detta ett problem eftersom man normalt inte använder kassettbandspelare till dessa mikro-datorer. Till ABC806 går det inte ens att ansluta en kassettbandspelare!

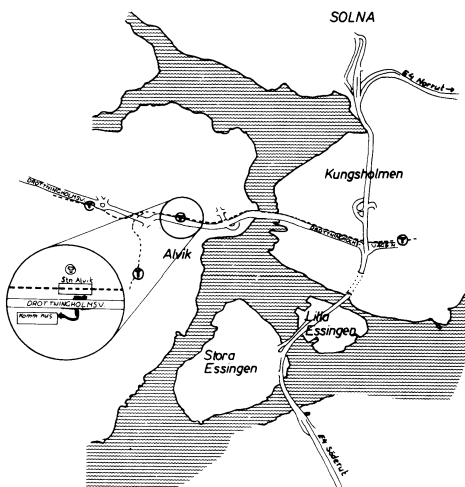
Kopiering av flexskivor är tyvärr inte möjlig, eftersom man inte kan snabbkopiera dessa. Dessutom finns det många olika typer av floppydiskar. Distributionen av FORTH79 på flexskiva får ses om ett undantag, tyvärr!

Därför måste programdistributionen ske via klubbens monitor. Flertalet av de 800-program som vi har fått in finns där. ABC-Klubben arbetar nu med att bygga ut det nuvarande monitorsystemet med ytterligare en linje för 300 bps. På sikt vill vi försöka att få en 1200 bps/V23 linje eftersom denna hastighet är den snabbaste "fria" hastigheten.

En annan enkel metod att få tag i de 800-program som klubben kommer att distribuera på kassett är att be någon som har tillgång till ABC80 om hjälp med kopieringen. I medlemsmatrikeln kan man se vem som bor inom samma postnummerområde.

Skicka gärna in 800-program till ABC-Klubben. Vi har ännu inte fått in så många program. Skriv då ett programhuvud som förutom vanliga uppgifter som programmets namn också talar om vilka 800-datorer som programmet är avsett för. Om disk används, ange då för vilken typ av disk och diskoperativsystem som programmet är avsett. Främst är det viktigt att ange om programmet är avsett för UFD-doset eller för gammalt dos. Ange också på vilken 800 dator som programmet är testat.

Av Bo Kullmar



Forts från föregnr.

FORTH.

FORTH PÅ ABC 80, DEL 2

INTERPRET är betydligt mer komplicerad. Först söker det efter ordet TIB i ordlistan. Finns det inte där, är det antagligen ett tal. Är det inte ett tal heller, är det fel, och följdaktligen ges ett felmeddelande.

INTERPRET ska dessutom kunna skilja mellan interpretering (tolkning) och kompilering.

```
: INTERPRET ( -- )
  BEGIN
  -FIND
  IF STATE E <
    IF CFA ,
    ELSE EXECUTE
    THEN
  ELSE HERE NUMBER
    DPL E 1+
    IF /COMPILE/ DLITERAL
    ELSE DROP /COMPILE/ LITERAL
    THEN
  THEN ?STACK
  AGAIN ;
```

-FIND letar i ordlistan efter det ord som finns i TIB, med startposition tagen från IN. Om ordet finns där, returneras parameter-fält-adressen (parameterfältet är den area i ett ord där de ord det består av finns), längden på namnet, och en sann flagga, annars en falsk. Ordet CFA konverterar parameter-fält-adressen till kod-fält-adressen, vilket är den adress där en pekare finns, vilken pekar på den maskinkodsrutin som ska utföras när ett ord utförs. Denna rutin är olika för kolon-definitioner, variabler och konstanter.

Ordet , (komma), lagrar ett 16-bitars värde i ordlistan, i det här fallet en adress, i ett ords parameterfält. DPL är en variabel som sätts av NUMBER. Den anger det eventuella antalet siffror till höger om decimalpunkten. En decimalpunkt anger att talet ska tolkas som ett dubbeltal. Om DPL är -1, finns det ingen decimalpunkt, varför talet konverteras till ett enkeltal. ?STACK kontrollerar att stacken inte har svämmat över. Om den är överfull eller "undertom", ges ett felmeddelande. Både NUMBER och ?STACK ger själva felmeddelanden.

Observera att DLITERAL och LITERAL föregås av ordet /COMPILE/, eftersom de är IMMEDIATES. /COMPILE/ letar igenom ordlistan efter det ord som följer /COMPILE/ (återigen används ordet -FIND). Om det inte finns där, ges ett felmeddelande. I annat fall kompileras ordet (adressen till kodfältet läggs efter HERE, med hjälp av ordet "komma", samt att ordlistepekaren (den heter DP för Dictionary Pointer, och är en variabel) ökas med 2 bytes.

```
: /COMPILE/ ( -- )
  -FIND 0= 0 ?ERROR
  DROP CFA , ;
```

?ERROR skriver ut ett felmeddelande (i det här fallet nummer 0) om det näst-översta värdet på stacken är sant. Felmeddelandena tas med offset (startposition) från skärm 4 (om variabeln WARNING är noll, ges ett numeriskt felmeddelande). Om felkoden överstiger 21 (största radnumret på en skärm), sätts rad 0 på nästa skärm (nr. 5) till rad 22 o.s.v.

LITERAL kompilarar ett 16-bitars värde (DLITERAL kompilarar ett 32-bitars värde). En numerisk konstant kompileras genom att ordet LIT kompileras, sedan läggs ett 16-bitars värde ner efter LIT. När koden sedan utförs, ökar LIT programräknaren (det register som håller reda på var i minnet den kompilerade programkod som körs finns, programräknaren kallas IP för Interpretative Pointer), samtidigt som de 2 bytes som LIT stegar förbi läggs på stacken. DLITERAL använder LITERAL 2 gånger genom satsen /COMPILE/ LITERAL.

```
: LITERAL ( N -- )
  STATE E
  IF COMPILE LIT ,
  THEN ;
  IMMEDIATE

: DLITERAL ( D -- ) STATE E
  IF SWAP /COMPILE/ LITERAL /COMPILE/ LITERAL
  THEN ;
  IMMEDIATE
```

COMPILE kompilarar det ord som ligger efter COMPILE i den kompilerade programkoden. Det här var ju krångligt. Men det fungerar i alla fall så här. Antag att du definierar ett IMMEDIATE ord, som ska lägga ner värdet 7 i ordlistan (från HERE). Ordet heter SEVEN, och definieras på följande sätt:

```
: SEVEN ( -- )
  COMPILE LIT
  7 , ;
```

IMMEDIATE

SEVEN får ingå i ett ord som heter .SEVEN, d.v.s. ett ord som ska skriva ut talet 7 när det exekveras (utförs). Punkten i .SEVEN talar om att något skrivs ut. .SEVEN är inte en IMMEDIATE.

```
: .SEVEN ( -- )
  SEVEN . ;
```

Naturligtvis skulle det gå lika bra att använda en konstant i stället för SEVEN, men för exemplens skull så...

Seven utförs när .SEVEN kompileras, eftersom det är ett IMMEDIATE ord. SEVEN lägger först ner ordet LIT i definitionen av .SEVEN, sedan talet 7. Adressen till ordet SEVEN kompileras inte. Om man vill använda ett IMMEDIATE ord i en egen definition, måste man skriva /COMPILE/ ccc, där ccc är ett IMMEDIATE ord.

Beräkningsstacken är den stack som vanligtvis kallas för "stack-en". Beräkningsstackens startposition bestäms av variabeln S0. Beräkningsstackens nuvarande adressfås med SPÉ, som returnerar adressen till stackens översta värde (egentligen det understa, eftersom alla stackar växer neråt) som den var innan SPÉ utfördes.

Om man har fyllt stacken, och vill nollställa den, kan man, i stället för att skriva DROP DROP DROP ..., skriva SP!, vilket sätter stackpekaren (naturligtvis finns det en sådan, för att hålla reda på var i minnet man kan lägga upp på resp. hämta värden från stacken) till S0. Som stackpekare används mikroprocessorn Z80As stackregister SP. Med hjälp av S0 och SPÉ kan man definiera sina egna stack-mainpulerings-ord.

Returstacken används för att lagra hoppadresser. När ett ord anropas (och det sker ju nästan jämt), måste man naturligtvis hålla reda på varifrån anropet skedde. När ett FORTH-ord anropas, läggs därför IP på returstacken (IP är i Z80 FORTH mikroprocessor-register BC), därefter laddas IP med adressen till det första ordet i det anropade ordet. Detta nya ord anropar i sin tur andra ord, varför IP återigen läggs på returstacken o.s.v. Returstackens startposition anges av variabeln R0. RP! använder det värde som finns lagrat i R0 för att nollställa returstacken. Returstacken har också en andra uppgift. Man kan nämligen spara undan värden tillfälligt på returstacken. Man får bara inte glömma att hämta tillbaka värdena innan man gör retur (när definitionen är slut), eftersom IP då kommer att laddas med fel värde från returstacken. >R lägger det tal som finns överst på beräkningsstacken på returstacken. R kopierar det värde som finns på returstacken till beräkningsstacken. Värde på returstacken finns alltså fortfarande kvar. R> hämtar ett värde från returstacken, och lägger det på beräkningsstacken.

Var försiktig med dessa ord. Det kan krascha FORTH-systemet om de används fel.

Det finns ytterligare ett sätt att avbryta en BEGIN -- AGAIN-loop än det som visades i förra avsnittet, med hjälp av de ord som manipulerar returstacken. Du kan stoppa den med R> DROP.

Definierande ord är sådana som skapar ett nytt ord. Exempel på sådana ord är : (kolon), VARIABLE och CONSTANT. När ett sådant ord utföres, skapas ett s.k. headerfält. Det innehåller, i nämnd ordning, namnets längd i en byte, namnet i varierande antal bytes, en pekare till nästa ord i ordlistan, en kodpekare till den assemblerkod som ska utföras när ett ord anropas (denna kod är olika för kolon, VARIABLE och CONSTANT), samt ett fält med pekare till de ord detta ord består av. Dessa fält kallas namnfält, länkfält, kodfält och parameterfält. Sedan headerfältet har skapats (det består av namnfält och länkfält), läggs kodpekaren ner, den s.k. run-time-koden, för aktuellt ord ner vid HERE (HERE är definierad som DP E). VARIABLE och CONSTANT sparar sedan ett 16-bitars tal, medan kolon sätter FORTH i kompileringsläge, vilket gör att INTERPRET kommer att spara adressen (till kodfältet) för de ord som räknas upp i ordlistan, fram till semikolon, vilket sätter FORTH i interpreteringsläge.

Man kan göra egna definierande ord, t.ex. för olika datatyper. Ett definierande ord har alltså dels en egen kod, dels en kod som utförs när ordet som skapades med detta ord utförs. Det är denna senare kod som kallas "run-time-kod".

Den egna koden inleds med ordet <BUILDS, och run-time-koden inleds sedan med DOES>. Run-time-koden skapas här i FORTH-kod, men kan också skapas i assemblerkod, men det tar vi i kommande avsnitt. Ett lämpligt exempel är ordet 2VARIABLE, som används för att lagra ett 32-bitars tal. Det kan definieras som 0 VARIABLE 2 ALLOT (ALLOT flyttar fram ordlistepeka-

ren det antal bytes som anges före), men för exemplets skull ska vi definiera allt själva.

```
: 2VARIABLE ( D -- )
  <BUILDS , ,
  DOES> ;
```

Eftersom adressen till första byten efter run-time-kod-pekaren i det definierade ordet läggs på stacken innan 2VARIABLES run-time-kod utförs (den som står efter DOES>), anger vi ingen run-time kod. Efter run-time pekaren ligger ju det värde vi ville nå, och adressen dit får vi s.a.s. på köpet. Lite värre blir det med 2CONSTANT.

```
: 2CONSTANT ( D -- )
  <BUILDS , ,
  DOES> 2E ;
```

Här ska vi ju också hämta det värde som konstanten motsvarar. På liknande sätt kan man göra arrays ("indexerade variabler"), flerdimensionella matriser, strängar m.m.

Kompilerande ord är ord som gör något under kompileringen. Dessa är IMMEDIATES. Exempel på kompilerande ord är SEVEN (!), IF, ELSE, WHILE, REPEAT, UNTIL och AGAIN.

```
: IF ( -- )
  COMPILER OBRANCH
  HERE 0 , 2 ; IMMEDIATE
```

CASE-satsen är skapad av Charles E. Eaker (nej, han är INTE medlem i ABC-klubben). Det CASE-satsen ska göra är att jämföra ett argument, som ges CASE-ordet, med olika jämförelsevärden, ett åt gången, som ges ett annat villkorsord. Om argumentet överensstämmer med testvärdet, ska det som står efter villkorsordet utföras, annars ska jämförelsen fortsätta till CASE-satsens slut (här kallat ENDCASE). Alla ord i CASE-satsen måste vara IMMEDIATES, av kompilerande typ.

```
: CASE ( N -- )
  ?COMP CSP É !CSP 4 ; IMMEDIATE
```

CASE-satsen inleds med ordet CASE. Det testar om FORTH befinner sig i kompilersmode, och avbryter om så inte är fallet. CSP är en variabel som används främst av kompilerande ord. Den är avsedd att spara stackens position i. Sedan kan man testa om stackens aktuella position motsvarar den som finns i CSP, och ge ett felmeddelande om så inte är fallet (med ordet ?CSP). Här hämtar vi CSPs värde (det kan ju ha satts av t.ex. en BEGIN -- WHILE -- REPEAT-sats), och lagrar det på stacken (stacken används inte till något speciellt just nu - vi kompilerar ju). Sedan sparas stackens position i CSP för senare felkontroll av ordet ENDCASE. Slutligen lämnas en fyra på stacken för typkontroll (se även IF-satsen, den lämnar en 2a, vilken används av ELSE och THEN för att kontrollera att man inte skriver IF -- WHILE -- UNTIL, eller något annat galet. Den villkorliga satsen heter här OF.

```
: OF ( N1 N2 -- N1 at no match, otherwise nothing )
  4 ?PAIRS ( KOLLA OM DET ÄR RÄTT TYP )
  COMPILER OVER COMPILER = ( LÄGG NER TEST-SEKVENSEN )
  COMPILER OBRANCH ( VILLKORLIGT HOPP )
  HERE 0 , ( SPARA ADRESSEN )
  ( TILL HOPPADRESSEN )
  ( OCH RESERVERA PLATS )
  ( FÖR ADRESS )
  COMPILER DROP 5 ; IMMEDIATE
```

Först kontrolleras att OF är rätt ord i detta fall (med ordet ?PAIRS), sedan kompileras ordet OVER, = resp. OBRANCH in. HERE lämnas på stacken för senare bruk av de avslutande orden, varefter address 0 anges som hoppadress tills vidare. Det är för att kunna lägga in rätt address här senare, som HERE läggs på stacken. OBRANCH gör ett hopp om en falsk flagga ligger på stacken när koden utförs. Sedan läggs ett DROP in för att ta bort CASEs argument om jämförelsen stämde.

OF ska naturligtvis ha ett avslutande ord. Det heter, mycket logiskt, ENDOF.

```
: ENDOF ( -- )
  5 ?PAIRS
  COMPILER BRANCH HERE 0 ,
  SWAP 2 /COMPILE/ THEN 4 ; IMMEDIATE
```

Som vanligt sker först en typtest. Därefter läggs ett ovillkorligt hopp in. Ett SWAP utförs för att hämta adressen där OFs villkorliga hoppadress ska läggas. En tvåa läggs på stacken (det är den som THEN testar mot), varefter THEN utförs, med alltså vad det innebär av ytterligare kompilering. Till sist läggs en fyra på stacken att användas av OF eller ENDCASE som testvärde vid kompileringen. Nu är det bara ENDCASE kvar.

```
: ENDCASE ( -- )
  4 ?PAIRS
  COMPILER DROP
  BEGIN
  SPÉ CSP É = NOT
  WHILE
  2 /COMPILE/ THEN REPEAT
  CSP ! ; IMMEDIATE
```

Först testas, som vanligt, att ENDCASE är rätt ord i sammanhanget. Därefter läggs ett DROP ner, för att ta bort eventuellt argument som kan ligga kvar, om inget av fallen stämde. En WHILE-loop utförs tills alla jämförelser har försetts med hoppadresser. SPÉ hämtar stackpekarens aktuella värde, därefter hämtas stackens position som den var innan CASE-satsens början. Om dessa två värden inte överensstämmer, utförs THEN, som lägger ner hoppadresser alldeles gratis. Slutligen återställs CSP med det värde som lagrades på stacken. CASE-satsen används på följande sätt:

Antag att du vill att den som använder programmet bara ska få mata in A, B och C. Om någon annan tangent trycks ner, ska en 'bell'-signal ges. De (giltiga) tecken som matas in ska inte sparas, bara skrivas ut.

```
: GET-ONLY-ABC ( -- )
  5 0 DO ( 5 TECKEN )
  KEY CASE
  65 OF 65 EMIT ENDOF
  66 OF 66 EMIT ENDOF
  67 OF 67 EMIT ENDOF
  ( OTHERWISE )
  7 EMIT
  ENDCASE
  LOOP ;
```

Observera att om en definition inte får plats på en skärm, kan du göra tre saker.

- 1) "Packa" programmet, d.v.s. skriv bara ett mellanslag mellan varje ord, utan att byta rad vid DO-LOOP, IF-ELSE-THEN, BEGINUNTIL o.s.v.
- 2) Dela upp definitionen på flera skärmar med ordet -->. Efter som --> är en IMMEDIATE fungerar det utmärkt. Att låta en definition sträcka över flera skärmar brukar betecknas som "dålig FORTH-stil" (en FORTH-definition ska vara rätt kort), men jag anser att det är betydligt bättre att göra så än att packa programmet.
- 3) Dela upp definitionen i flera, små definitioner. Det går dock inte alltid att göra så (exempelvis vid en CASE-- sats med många jämförelser). I sådana fall är alternativ 2 det bästa.

En IMMEDIATE-definition som är praktisk att ha när man skriver program är ASCII. ASCII kompilerar ascii-värdet av det tecken som står efter ASCII. Ex:

ASCII A ger samma resultat som att skriva 65, vilket är ascii-värdet för A. ASCII ligger på systemskärmarna, som levererades i filen SCREEN.TXT. Skriv 6 LOAD för att ladda in ASCII.

På samma skärm ligger WHERE, som används för felsökning vid laddning från skärmar. När ett fel har uppstått, kan man skriva WHERE. Då skrivs skärmens nummer ut, samtidigt som den rad där felet uppstod skrivs ut. Det felaktiga ordet markeras med ett delete-tecken, "fylld ruta", på raden under.

```
: ASCII ( -- )
  BL WORD HERE 1+ CÉ
  /COMPILE/ LITERAL ; IMMEDIATE
```

Stränghanteringen i FORTH sker på samma sätt som i assembler. För som inte har någon erfarenhet av assemblerprogrammering, kan jag nämna, att det innebär full frihet att skjuta in tecken, dra ihop, flytta, skarva o.s.v.

En strängvariabel (man lägger inte gärna strängar på stacken, bara dess address) kan man definiera med ordet CVARIABLE (fåö Charakter VARIABLE), med strängens maximala längd och ordet ALLOT efter.

```
0 CVARIABLE STRÄNG 28 ALLOT
```

reserverar plats för en strängvariabel med namnet STRÄNG, som inte får innehålla fler tecken än 28 (i första byten ska strängens längd ligga), med antalet tecken satt till noll från början. Med hjälp av <BUILDS och DOES> kan man skapa egna datatyper, t.ex. den datatyp som kallas CHAR i Pascal. Orden CMOVE och <CMOVE används för att flytta minnesareor. Med

HAUSSE PÅ SVENSKA

ADMINISTRATIVA PROGRAM

Argus 800. Ett programpaket konstruerat för projektkalkylering i små och medelstora företag.

Baslager. Ett komplett program för rationell lagerredovisning.

Basregister 800. Vidareutveckling av Basregister ABC.

ABC 800 Bild. Bild 800 är ett program till ABC 800 och 806, som hjälper dig att rita bilder på datorns skärm. Bilder som du sedan sparar, fotograferar av eller skriver ut på en skrivare eller färgskrivare.

Byggadministration. Lokal uppföljning på byggarbetsplatser.

Byrå. Projekt/lönsamhetsuppföljningsprogram för annonsbyråer och andra konsultföretag.

Comp-Graphics. Produktion av bildserier för information, undervisning och reklam.

Comp — se om. Datorprogram som utför företagsanalyser.

Dublett. Program för borttagning av dubletter i basregistret.

Eagle Eye. Registerhantering och informationssökning utan programmering.

Etikett. Tilläggsprogram till basregistret som skriver ut etiketter (upp till fem i bredd med 10 valmöjligheter samtidigt).

Fakturerings 800/Kundreskontra 800. För alla företag med behov av ett system, där man registrerar in posterna samtidigt som faktura framställs.

FH-data. Registerprogram för företagshälsovårdscentraler och mindre läkarmottagningar.

Frakt och inköpsadministration. Program för grossister eller industriföretag med behov av exakt planering av frakter.

Grafik 800. Presenterar bild och text i färg och högupplösningsgrafik.

Hotell 800. Bokningsprogram för mindre och medelstora hotell.

Hyresservice. Ett program som tar hand om allt rutinarbete i samband med hyresavisering.

Importrutin med tulldeklarering. Program för alla importerande företag.

Inkasso 800. Inkassosystem anpassat till gällande lagstiftning.

Inventarie- och anläggningsregister. Programmet lagrar grunddata för varje inventarie, t.ex. konto, benämning, tillverkare, inköpsår m.m.

Kvalitetsgradering — kvalitetsuppföljning. Program för kvalitets- och produktionstekniker som arbetar med uppläggning av kvalitetskontrollrutiner.

Kontek Data Lön 800. Ett avtalsberoende lönesystem för små och medelstora företag.

Lager 82. Generellt lagerhanteringssystem, integrerat med Faktura 82.

Lager 800. Innehåller bl.a. inventeringsrutin, lagervärde och automatisk prisjustering.

Leasingsystem. Program för leasing och uthyrningsföretag.

Leverantörsreskontra 800. För alla företag med behov av leverantörsreskontra.

Lön F. Lön F är en specialvariant av lön III. Avsedd för servicebyråer, banker m.fl. som administrerar löneutbetalningar åt många företag med få anställda.

Mack 800. Löser administrationen för varje bensinmack.

Mat och bespisning. Program för ekonomiföreståndare vid skolor.

Mediadistribution. Program för företag som distribuerar ljus- och videoband, typ AV-centraler.

**MPS-Modul Order, Inköp, Lager/För-
råd, Kommunikation med stordator.** Program för tillverkande företag, handelsföretag.

Mäklardata. Program för fastighetsmäklare.

Ord 800. En vidareutveckling av ABC Ord.

Order 800. Program för kundregister, orderregister och leverantörsregister.

**Planering avskrivning av inventarier/
maskiner, till Basregister 800.** Läs/skrivrutin till Basregister 800.

Press supply print. Verksamhetsplaneringssystem för tryckerier, innehållande förkalkyl, orderhantering, produktionsuppföljning, efterkalkyl och fakturering.

Produkt. Ett effektivt hjälpmedel för alla tillverkande mindre företag.

Produktionsplanering och analys. Programmet är avsett för bearbetningsindustrier.

Redovisning 800. Ingår som en del i ABC 800:s administrativa paket, men kan även köras separat.

Register 800. Register för adresser, prislister, lager, kunder m.m.

Repoff. Program för företag som säljer mot postförskott.

Resultatberäkningar vid tävlingar. Programmet utförs som variationer på Luxor/Digits program tävling. Ex skyttetävling bana/fält.

Schemaregistrering. Program för rektorsexpeditioner.

Skatt. Beräknar preliminärskatteavdrag, med tillstånd av Riksskatteverket.

Struktur 800. Strukturregister i en nivå till Lager 800. Recept, behovsberäkningar och automatiska uttag.

Sum-II. Summeringsrutin för Basregister II med decimaltabulering.

Tandläkar 800. Komplet t tandläkaradministration.

Taxilön. Specialanpassat löneprogram för taxiföretag.

Tidredovisning. Tids- och kostnadsredovisning för projekt/uppdrag för konsultbyråer, ingenjörsfirmor, advokatbyråer, lego-tillverkare m.fl.

Det finns två kategorier smådatortillverkare. Dom som satsar på fantastiska maskiner för maskinernas egen skull. Och dom som satsar på maskiner som skall passa användaren. Den första gruppen kan vi glömma. Den andra gruppen kan delas in i tre olika klasser. De som saknar programvara eller har nyskrivna, ännu inte utprovade program. De som påstår sig ha massor av

program, men bara ett fåtal på svenska. Och så Luxor Datorer.

Luxor Datorer har ett unikt programutbud. Med svenska Standardprogram för företagets vanliga rutiner. Testade och utprovade under lång tid och in i minsta detalj. Av oss själva och inte minst av slutanvändarna. Alla de 1000-tals svenska företag som redan idag använder våra program.

NÄRMASTE TEAM 100-HANDLARE HITTAR DU I TELEFONKATALOGENS

PROGRAMBÖRSEN.



KOMMUNIKATIONS-PROGRAM

ABC 1004. Synkront kommunikationsprotokoll för UNIVAC 1004.

Allfset. Specialversion av det synkrona kommunikationspaketet RJE 800.

ABC 2780. Program för synkron-kommunikation enligt IBM 2780/3780 protokoll.

Asynkron terminal VT100 – VT102. Detta kommunikationsprogram förvandlar en ABC 800 M-dator till en asynkron terminal.

ABC 3270 BSC. Kommunikationsprotokoll för emulering av IBM 3270 BSC.

ABC UTS 400. Synkront kommunikationsprotokoll för UNIVAC UTS 400.

Trace. Testutrustning och kontroll vid dataöverföring (asynkron).

ABC UTE 800. Kommunikationsprotokoll till bl.a. följande asynkrona terminaler: ANSI X3.41, X3.64, VT 100, VT 52, Datamedia Elite 1520/21, 2500, DT 80, Infoton Vistar, Facit 4430, Tandberg TDV 2230. (ADM-3A protokoll ingår som standard i ABC 800-familjen).

TTY 800. Asynkron terminalrutin för filöverföring.

ABC 3270 SDLC/SNA. Synkront kommunikationsprotokoll till IBM 3270 SDLC/SNA.

ABC USER 200. Synkront kommunikationsprotokoll till ControlData CDC USER 200.

BERÄKNINGSPROGRAM

Assemblerad FFT – Fast Fourier Transform. Digital signalbehandling.

Beteendemätning. Programvara för neurologisk forskning förmätning av reaktionstider, navigeringsminne och olika typer av aktiviteter hos försöksdjur.

Diagram II 800. Programmet ritlar kurv-, trapp-, punkt-, stapel- och cirkeldiagram. Dessutom finns möjlighet att kombinera olika diagramtyper i samma diagram.

Husköp 83–85. Program för mäklare, banker och hustillverkningsföretag.

Kalkyl 800. Förvandlar ABC 800 till ett kraftfullt beräkningsverktyg.

MPS (material- och produktionsstyrning). För mindre verkstadsindustrier och som komplement vid stora verkstadsföretag.

Omega. Program för tidmätning i styr- och reglersammanhang samt vid tävlingar.

Planetarium. Program som presenterar solens upp/nergång för valfri ort.

Plotter. Ett av marknadens mest avancerade och allsidiga programpaket för "business graphics", tekniskt-vetenskaplig grafik och grafik för undervisning.

Reglerteknik. Program för ingenjörer och konsulter samt för undervisning i reglerteknik inom teknisk utbildning.

Reseräkning (ARA). Beräknar rese-räkningar.

Skatt 83–85. Skatteplaneringsprogram för tiden 1983–85.

Tax 84. Program för banker, revisorer och skatterådgivare.

VVS-paket. Tvåror, ettrör, K-värde, KV-värde, tappvatten, värmebehov och luft.

Väglinjer. Väglinjeberäkningssystem för markbyggare.

SYSTEMPROGRAM

ASM 800. Avancerad assembler.

Assembler 800. Z80-assembler skriven i maskinkod med editor.

Bridos. Ett CPM kompatibelt operativsystem.

Doc. Program som automatiskt dokumenterar Basic-program.

Enterlib. Program för systemansvariga.

Galmod. Byggsten för applikationsutveckling.

GEO1. Geometriprogram.

HRbild. Composer för att producera, lagra och visa högupplösningsgrafik på ABC 800 M.

JSP 800. Datorstöd vid analys och konstruktion av system/program enligt JSP-metoden "Jackson Structured Program Design".

L & L Forth. Ett mycket snabbt högnivåspråk.

Mini-workbench. Systemrutinerna i administration 800.

Ritsub. Ett subrutinpaket för enkel hantering av HP's plotter 7470A.

Pascal 800. Programpaket för att skriva och köra program i språket Pascal på ABC 800.

ORANGE=Luxor Datorers Standardprogram



Vi har idag levererat över 40.000 smådatorer till de mest skiftande användningsområden. Därför har det utvecklats en unik flora av program till våra datorer. Flertalet av dessa program har vi sammanställt i en katalog "Svenska Programbörser".

Känner du till eller om du själv har program som du tycker borde vara med i "Svenska Programbörser", så skicka en kort beskrivning och uppgifter om vem som har gjort programmet till Luxor Datorer AB, Dokumentation, Box 923, 591 29 Motala.

Men inte nog med det. Till alla dessa Standardprogram kan vi lägga den guldkantade "Svenska Programbörser". Med flera hundra svenska program, från de mest skiftande branscher och verksamheter. Detta sammantaget ger oss en unik position på den svenska smådatormarknaden.

Därför vågar vi lova dej att haussen på "Svenska Programbörser" inte är någon tillfällighet.

Den har kommit för att stanna. Så vill du göra det riktiga klippet, utan att ta onödiga risker. Vänd dig till närmaste Team 100-handlare, våra egna "börs-mäklare" och köp Luxor Datorer. Säkra papper som står högt i kurs. Och bara ger mer och mer.

LUXOR
Datorer

GULA SIDOR UNDER RUBRIKEN "DATAMASKINER OCH TILLBEHÖR".

desså kan ordet ASSIGN skapas, vilken tilldelar en sträng en annan strängs innehåll.

```
: ASSIGN ( FROM TO -- )
OVER CÉ 1+ CMOVE ;
```

ASSIGN flyttar hela strängen vid address FROM till address TO. Hela strängen flyttas, inklusive längdbyten.

Skilnad mellan CMOVE och <CMOVE är att CMOVE flyttar lägsta byten först och går mot högre adresser, medan <CMOVE gör tvärt om. Om du ska flytta isär tecken i en sträng ska du använda <CMOVE, medan CMOVE ska användas om du ska dra ihop.

Strukturerad programmering är ett område som det har yttrats många dumheter om. Här kommer några till.

I många böcker och manualer om FORTH står det att FORTH är lämpligt att användas för att konstruera program enligt Top/Down metoden. Top/Down är inte en metod. Det är ett sätt att skriva och konstruera program. JSP (Jackson Structured Programming) är en metod. Top/Down kan användas för att konstruera program i JSP. Bottom/Up går lika bra att använda. Top/Down går till på följande sätt. Man börjar med att definiera huvudordet. Det ska beskriva vad programmet gör. För ett applikationsprogram, t.ex. ett databasprogram, kan detta se ut så här:

```
: DATABAS      ( MAIN )
  INITIERA
  BEGIN
      MENY
      UTFÖR
  UNTIL
  AVSLUTA      ;
```

INITIERA ska initiera databasen. Det består i sin tur av andra ord, som på en lägre nivå beskriver hur denna initiering ska gå till. Var och ett av dessa ord består av ytterligare ord på en ännu lägre nivå, som beskriver en avgränsad process, t.ex. att fråga efter behörighet till databasen, o.s.v. På samma sätt är MENY, UTFÖR och AVSLUTA nerbrutna i ord på lägre nivå.

Bottom/Up går till på det omvända sättet, d.v.s. att man börjar med att definiera orden på den lägsta nivån.

I ett strukturerat program får ord på olika nivåer inte anropa varandra. Det kan vara svårt att hålla reda på om man enbart använder Top/Down eller Bottom/Up. Anledningen till att man håller så strikta regler är att det ska vara enkelt att följa programmet, ändra (utan att det blir grötigt), felsöka och göra tillägg. JSP är en metod. Det finns bestämda regler för hur man får göra i JSP. Det finns det inte i Top/Down eller Bottom/Up, eftersom de inte är metoder att utveckla program, utan sätt att mata in programmet i datorn. Men, som sagt var, Top/Down och JSP går utmärkt att kombinera.

I FORTH använder man en trädstruktur (det kallas så eftersom strukturdiagrammen, med lite fantasi, ser ut som ett träd med roten högst upp). Högst upp, det som kallas "roten", ska program-mets eller delrutinens namn stå. För att se vad denna är av för typ, måste man titta på närmast lägre nivå. Det finns bara 3 olika typer av strukturer, nämligen sekvensen (utförs efter varandra), iterationen (upprepnig, t.ex. en loop), och selektionen (ett val, t.ex. IF -- ELSE -- THEN). Med dessa tre typer kan man konstruera ALLA program. Det bevisade två herrar vid namn Jacopini och Böhm det minnesvärda året 1966. En sekvens markeras inte alls, en iteration markeras med en stjärna eller en asterisk, och en selektion markeras med en ring. Ovanstående exempel, DATABASE, kan uttryckas på följande sätt i ett JSP-strukturdiagram.

[illegible][illegible]

Man ska inte skriva större diagram än vad som ryms på en A4-sida (jag brukar använda liggande A4or). De ord som är utan vidare förgreningar (de s.k. löven, i detta fall MENY och UTFÖR), kan man beskriva närmare på särskilda sidor (naturligtvis i JSP-strukturdiagram). DATABST ovan utför ingenting. Det är bara en stomme som måste stå där. Eftersom DATABAS är en sekvens, kan det inte finnas en iteration mitt i. Man måste använda "rena" strukturer. Sedan alla diagram är färdiga, ska man inte sätta igång och skriva FORTH-kod på en gång. Vi har nämligen ett viktigt steg kvar, den s.k. schematiska logiken, vilket är ett slags mellan-programmeringsspråk som till sin struktur liknar Pascal. Observera att löven har semikolon efter, som procedur-anrop i Pascal.

Exemplet ovan skulle uttryckas så här:

```
DATABAS seq
  INITIERA;
  DATABST iter until VAL=AVSLUT
    MENY;
    UTFÖR;
  DATABST end
  AVSLUTA;
DATABAS end
```

Nu kan man skriva den FORTH-snutt som visades tidigare. JSP visar sina starka sidor framför allt vid mer komplicerade program som det var ett tag sen man skrev, eller som man kanske inte alls skrivit själv. Har man dessutom tillgång till strukturdiagrammen och den schematiska logiken, kan man mycket snabbt sätta sig in i programmet. De höga underhållskostnaderna för program inom framför allt mini- och stor-dator området gör att hundratals programmerare undervisas i JSP varje år i Norden.

Hur långa ska definitionerna vara i FORTH ? Det beror på vad de utför, men helst bör de inte vara längre än några få rader. I BASIC använder man subrutiner för att ersätta en sekvens som förekommer på flera ställen. I FORTH sätter man en avgränsad operation i en definition. Har man skrivit JSP-diagrammen bra, är det lagom att låta varje JSP-ruta vara en definition.

Innan vi tar i tu med utility-modulen, ska vi göra ett mycket, mycket enkelt spel. Det brukar kallas "rysk roulette", och går ut på att tävla mot ett slumptal. Det gäller att klara sig så länge som möjligt. Har man otur, kommer slumptalet upp i ett för högt tal, och sedan är man förlorad.

```

0 VARIABLE RÄKNARE
: SLUTA      ( AUBRYT SPELET )
  1 ;

: TRYCK      ( SE OM HJÄLTEN BLEV TRÄFFAD ELLER INTE )
  CLOCK ON 215 >
  IF CR ." Det är över nu..." SLUTA
  ELSE CR ." -KLICK-" 0 1 RÄKNARE +!
  THEN CR ;

: AUBRYT     ( HJÄLTEN VILL AUBRYTA )
  CR ." FEGIS!" CR CR SLUTA ;

: SPELA      ( SPELA TILLS HJÄLTEN AUBRYTER ELLER BLIR TRÄFFAD )
  0 RÄKNARE ! ( NOLLSTÄLL RÄKNAREN )
  BEGIN
    KEY
    CASE
      13 OF TRYCK ENDOF
      45 OF AUBRYT ENDOF
    ENDCASE
  UNTIL ;

: J/N        ( GE 1 FÖR J, 0 FÖR N )
  BEGIN
    KEY DUP 223 AND DUP DOP
    74 = SWAP 78 = OR
    IF 74 = SWAP EMIT 1
    ELSE 2DROP 0
    THEN
  UNTIL ;

```



```

: SPELA-IGEN? ( SKRIV UT HUR MÅNGA VÅR HJÄLTE KLARADE, OCH FRAGA
              OM HAN/HON VILL FORTSÄTTA
CR ." DU KLARADE " RÄKNARE ? ." GÅNGER." CR CR
." VILL DU SPELA IGEN ? " J/N NOT CR CR ;

: REGLER ( SKRIV UT VILKA REGLER SOM GÄLLER )
PAGE 7 SPACES
." ***** RYSK ROULETTE ***** CR CR CR
." DIN UPPGIFT ÄR ATT SPELA RYSK ROULETTE" CR
." MOT DEN TVELIVADE TSAREN" CR
." DU KAN KOMMA UN DAN GENOM ATT SKRIVA" CR
." BOKSTAVEN 'A' I STÄLLET FÖR ATT TRYCKA AV." CR
." DU KOMMER DA UN DAN MED SKAMMEN," CR
." OCH BLIR HANAD AV DEN TVELIVADE TSAREN." CR CR
." TRYCK PA NÅGON TANGENT FÖR ATT BÖRJA : " KEY DROP
CR CR ;

: HEJDA ( VÄLKOMNA TILL FORTSATT SPELANDE VID TILFÄLLE )
CR ." VÄLDIGT TRAKTIGT." CR
." DEN TVELIVADE TSAREN HÄLSAR DIG VÄLKOMMEN" CR
." VID SENARE TILFÄLLE FÖR FORTSÄTTA FESTLIGHETER."
CR CR ;

: ROULETTE ( HUVUDLOOP )
REGLER
BEGIN
  SPELA
  SPELA-IGEN?
UNTIL
HEJDA
QUIT ;

```

Du får själv klura ut hur ovanstående program fungerar. Det är konstruerat med JSP.

Utility- (hjälp-) modulen däremot, ska vi gå igenom lite närmare. Du har väl, liksom jag, irriterats över att behöva skriva CLEAR innan en ny skärm ska tas i bruk. Programmet i sej är inte särskilt avancerat, bara några få ord långt, men desto mer användbart. Dessutom kan delar av det vara användbart i andra sammanhang.

```

0 VARIABLE FRÅN
0 VARIABLE TILL

: #IN ( TA IN ETT ENKELTAL FRÅN TANGENTBORDET )
TIB # SWAP EXPECT
BL WORD
HERE NUMBER DROP ;

: FRÅN? ( TA REDA PÅ STARTSKÄRM )
." FRÅN VILKEN SKÄRM SKA RADERINGEN SKE ? " 10 #IN
DUP DUP 1 < SWAP 1000 > OR
IF QUIT THEN FRÅN ! ;

: TILL? ( TILL VILKEN SKÄRM )
." TILL VILKEN SKÄRM ? " 10 #IN
DUP DUP FRÅN # < SWAP 1000 > OR
IF QUIT THEN TILL ! ;

: MELLAN-VILKA? ( MELLAN VILKA SKÄRMAR SKA DET RENSAS ? )
PAGE 7 SPACES
." ***** SUDDA SKÄRMAR ***** CR CR CR
FRÅN? CR
TILL? CR ;

: SUDDA-SKÄRMAR ( SUDDA SKÄRMARNA )
TILL # 1+ FRÅN # DO
  I BLOCK 768 BL FILL I UPDATE
  LOOP
  FLUSH ;

: SUDDA ( HUVUD-DEFINITIONEN, ELLER MAIN SOM MAN SÄGER )
MELLAN-VILKA?
SUDDA-SKÄRMAR
QUIT ;

```

\$IN tar in ett enkeltal från tangentbordet. Innan anropet sker, ska det maximala antalet siffror som får matas in anges. FRÅN? tar in ett värde, och kontrollerar rimligheten i svaret. TILL? gör samma sak, men kontrollerar att detta värde inte är lägre än från-värdet.

SUDDA-SKÄRMAR är det ord som sköter om själva suddningen. BLOCK ger adressen till den skärm som anges som parameter. Om skärmen inte finns i minnet läses den in. FILL suddar sedan 768 bytes från denna adress med blanktecken (mellanslag). Sedan markeras att denna skärm har ändrats (uppdaterats), så att den sedan skrivs tillbaks till skivan när den buffert den upptar i minnet behövs för andra skärmar. Till sist sköter FLUSH om att skriva tillbaka de återstående skärmarna. SUDDA är det ord man skriver för att starta programmet. QUIT i denna definition gör att inget "ok" skrivs ut efteråt.

De numeriska felkoderna är som följer:

- 1 - EMPTY STACK (stacken är tom)
- 2 - DICTIONARY FULL
- 3 - HAS INCORRECT ADDRESS MODE
- 4 - ISN'T UNIQUE
- 6 - DISC RANGE ?
- 7 - FULL STACK
- 8 - DISC ERROR ! (READ ERROR ?)
- 9 - DISC DOOR OPEN/NO DISC
- 10 - DISC IS WRITE PROTECTED
- 11 - DISC ISN'T FORMATTED
- 12 - NO DISC !

Starting FORTH

Det har för många varit mycket svårt (och dyrt) att få tag i Leo Brodies utmärkta lärobok *Starting FORTH* här i Sverige. ABC-klubben har därför som en form av medlemservice köpt hem ett parti av boken direkt från Amerika. Samtidigt har Bob Johnsen gjort en speciell version av FORTH som helt överensstämmer med den FORTH 79 Standard som finns i exemplen i boken. Du som vill lära dig FORTH får härmed bästa tänkbara hjälpmedel! Boken säljes i ett paket tillsammans med en programkassett med FORTHprogrammen. På kassetten finns även de nya FORTH-versioner som klubben fått tillgång till. Dessa innehåller kompletteringar med nya screens och rättelser av vissa buggar. De som har ABC800 får i stället för kassett programmen på diskett, och då anpassat till ABC800.

Priset för detta paket är 220 SEK som insättes på:

PG 62 93 00-5, ABC-klubben Publikationer. Ingen tillkommande moms.

Du kan även göra skriftlig beställning till: ABC-klubben Publikationer, Vidängsvägen 1, 161 33 Bromma. Varvid tillkommer postförskottsavgift.

GLÖM INTE ANGE DITT NAMN, DIN EGEN ADRESS SAMT DATORTYP vid beställning.!

220 SEK
PG 62 93 00-5
ABC-klubben
Publikationer

- 15 - Här ligger en kommentar som används vid listning på printer
- 22 - COMPILATION ONLY, USE IN DEFINITION
- 23 - EXECUTION ONLY
- 24 - CONDITIONALS NOT PAIRED
- 25 - DEFINITION NOT FINISHED
- 26 - IN PROTECTED DICTIONARY
- 27 - USE ONLY WHEN LOADING
- 28 - OFF CURRENT EDITING SCREEN
- 29 - DECLARE VOCABULARY

FORTH på ABC 800 med DOS 6-1x för ABC 830 har tagits fram av Mats Knuts. Vi som använder UFD-DOS och/eller ABC 832 får vänta tills en sådan version tagits fram. UFD-DOS är inte särskilt lämplig att använda tillsammans med den nuvarande lagringsmetoden. Direktfiler, eller, ännu hellre, textfiler vore här det riktiga. Textfilerna har dessutom den fördelen att programmen utan problem kan överföras via tråd, eller skrivas via vilken texteditor som helst. Man får dessutom fritt format på programmen och är inte begränsad till skärm-formatet, något som har betydelse när man ändrar i programmen. Man behöver inte förlora det virtuella minne som skärmarna utgör. Man kan simulera detta via direktfiler som kontrolleras via de vanliga orden FLUSH, BLOCK, UPDATE m.fl. ord. Men detta är utökningar för framtiden.

Till nästa avsnitt, GO FORTH!

Robert Claeson <3492>

Synpunkter på fig-FORTH

Robert Claesson <3492> kommer under vintern att utveckla en ny FORTH enligt FORTH83-standard. Den som har upptäckt buggar i fig-FORTH, saknar funktioner eller har andra synpunkter kan ta kontakt med ▶

<3492>

Robert Claeson
 Dommarvägen 47
 961 40 Boden
 0921-501 87

BOOLES ALGEBRA

FÖR SVENNE & CO.

I programsammanhang stöter man ofta på problemet att göra små bokstäver till stora. Men kan inte hindra att en användare svarar med små där man väntar sig stora. Antingen det sker av slarv eller av okunnighet måste man göra något åt saken.

Här ett förslag:

```
100 INPUT A$
110 A%=ASC(A$) AND 95%
120 ;CHR$(A%)
```

Rad 110 gör om alla små bokstäver till stora. Hur fungerar det?

För att förstå det behöver man titta litet närmare på hur datorn skiljer mellan "sant" och "falskt".

I satser som IF...THEN undersöker datorn om det som står efter IF är sant. I så fall utförs det som står efter THEN, i annat fall icke.

Uppgiften sant-falskt lagras i en byte som består av 8 minnesceller. Var och en av dessa celler kan ha ett av två tillstånd: "på"=1 eller "av"=0. Tillstånden kan åstadkommas på olika sätt som vi inte behöver befatta oss med. För oss räcker det att innehållet i cellen av datorn uppfattas om en nolla eller en etta.

Om alla cellerna = noll betyder det "falskt". Ber man datorn skriva vad som står i denna byte blir det förstås en nolla.

Om alla cellerna = 1 betyder det "sant". Detta skrivs av datorn som -1. (Datorn lagrar för övrigt också heltalet 65535 på samma sätt.)

För ABC80 gäller att allt som inte är falskt betraktas som "icke falskt"="sant". Allt utom noll uppfattas därför som sant.

Man kan skriva t ex

IF Q% THEN ---

Om Q% inte är noll kommer resten av satsen att utföras. Man behöver alltså inte skriva IF Q%=-1%...

Q%=NOT Q%

förvandlar Q-värdet noll till -1 och tvärtom, vilket kan användas som "switch" dvs en markering som växlar mellan sann och falsk varje gång raden utförs.

Nu kommer vi till AND. Det betyder OCH på engelska och verkan av detta ord stämmer för det mesta ganska bra med ordets språkliga innebörd.

IF A<B AND C<D THEN X=Y

betyder att om både A<B och C<D så utförs X=Y.

Men datorn arbetar inte med det språkliga. Den undersöker först om A<B. Om detta är sant sätter den -1 i den byte som skall hålla reda på resultatet, annars sätter den 0. Sedan undersöker den på samma sätt om C<D.

Ifall resultatet också där är en etta då sätter datorn en etta som totalresultat av undersökningen, dvs "sant" som blir signalen till att X=Y utförs.

Om endera villkoret inte stämmer, blir slutresultatet en nolla, alltså "falskt", och satsen efter THEN utförs inte.

Krångligt? Ja, det kan vara svårt att få grepp om det här i början. Låt oss nu

titta på våra små bokstäver i raden 120 ovan.

Vad är skillnaden mellan små och stora? Om du tittar i en ASCII-tabell finner du att A=65 och a=97. Skillnaden är alltså 32, och det gäller hela alfabetet.

Datorn jobbar hela tiden med celler som bara kan ha två värden, 0 och 1. Den använder det binära talsystemet och betecknar 65 som 01000001 medan 97 blir 01100001.

Bitarna betyder inte 1, 10, 100 osv som i decimalsystemet, utan 1, 2, 4 o s v:

Bittabell

värde	128	64	32	16	8	4	2	1
65D=	0	1	0	0	0	0	0	1
97D=	0	1	1	0	0	0	0	1
95D=	0	1	0	1	1	1	1	1

65D betyder 65 decimalt. Om man är noga skriver man binära tal som 01000001B, där B betyder binärt.

En skillnad på 32 mellan lilla och stora A innebär ett etta i 32-biten. Ta bort den och du får ASCII-värde för stora A.

Som du ser har det magiska talet 95D också skrivits in här ovan. Det har ettor i alla positioner UTOM 32 (och 128).

AND åstadkommer nu en jämförelse mellan talen bit för bit. Om motsvarande bit är 1 ("sann") i båda talen blir slutresultatet också "sant", alltså 1. Om endera talet har en nolla i motsvarande bit, blir slutresultatet noll.

Härav ser man att efter jämförelse med 95D kommer alltid en ev etta i 32-biten att försvinna. Och detta betyder att ASCII-värdet för en liten bokstav förvandlas till ASCII-värdet för motsvarande stora.

128-biten då? Jo den kommer att vara 1 så länge tangenten är nedtryckt. Sedan blir den noll. Men datorn jobbar mycket snabbt. Innan man hinner släppa tangenten har den kanske hunnit göra sin jämförelse. 95D tar bort den ettan också.

Förfarandet kallas ibland att "lägga på en mask", i det här fallet 95D, och "maska bort", i det här fallet 32-biten.

Det här är ett exempel på vad som i matematiken kallas Boole's algebra. För AND gäller följande sanningstabell:

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

där alltså som förut 0 och 1 betyder falskt och sant.

Det finns fler "ord" i sammanhanget.

OR används för att säga att en sak är sann om ANTINGEN ena villkoret ELLER det andra (eller båda) är sant:

Sanningstabellen ser ut så här:

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

XOR (exclusive or) betyder däremot att antingen den ena eller den andra MEN INTE BÅDA skall vara sanna för att slut-

resultatet skall bli sant. 1+1 är alltså 0, annars är det likadant som OR.

Det finns ytterligare några uttryck av detta slag, men dessa är de vanligaste.

XOR kan användas för att avläsa ABC80:s klocka.

Klockan finns lagrad i adresserna 65009, 65009 och 65010. Den första byten ändrar sig ett steg var femtiondels sekund, den andra ändrar sig ett steg varje gång den första gått ett varv, och den tredje när den andra gått ett varv.

Med följande program kan man se hur klockan jobbar:

```
500 ;PEEK(65008%), PEEK(65009%), PEEK(65010%)
600 GOTO 500
```

Då ser man att klockan går baklänges. Det kan man använda för tidtagning. Med POKE kan man fylla varje adress med ettor, sedan avläser man med PEEK().

Genom att jämföra det ursprungliga värdet (255) med det avlästa vet man hur lång tid som förflutit.

```
100 POKE 65008%,255%,255%,255%
```

fyller alla bitar i klockan med ettor.

```
200 IF INP(56%)>127% THEN 500:GOTO 200
```

INP(56%) avläser tangentbordet. Den har ASCII-värdet på senaste nedtryckte tangent. Så länge en tangent trycks ned, blir också 128-biten=1. Rad 200 avvaktar alltså tills en tangent tryckts ned. Då blir det första uttrycket sant och det sker hopp till rad 500 som avläser klockan.

Om man startar detta program med RUN, hinner man inte släppa tangenten innan programmet hunnit till rad 200, och då kommer man aldrig i gång. Sätt därför in:

```
10 FOR I=1 TO 1000:NEXT I;CHR$(7%)
```

Det ger en sekunds "försprång" och säger pling när klockan startas.

För att få ut tiden byter man rad 600:

```
600 T=(PEEK(65008%) XOR 255%)/50 + (PEEK(65009%) XOR 255%)*5.12
```

```
610 ;T' sekunder'
```

Den snabbaste byten rör sig var femtiondels sekund. Den nästföljande rör sig efter 255/50 sekunder, dvs efter 5.12 sekunder. Den långsammaste byten har utelämnats. Den rör sig först efter ungefär 20 minuter.

Vad gör nu XOR här? Anta att det hunnit gå 65 perioder när tangenten trycks ned och klockan avläses. Då har den snabbaste byten hunnit minska från 11111111B till 10111110B. XOR innebär att följande bitjämförelse görs:

.	128	64	32	16	8	4	2	1
.	=====							
255D=	1	1	1	1	1	1	1	1
190D=	1	0	1	1	1	1	1	0
.	=====							
XOR =	0	1	0	0	0	0	0	1

Resultatet bit för bit blir en etta om endast den ena av bitarna är 1. Man får ut 65, vilket är exakt så många perioder som klockan rört sig.

Hängde du med? Det här är svåra saker att förklara i skrift. Vare sig du förstod allt eller inte, kan du använda den här visade klockrutinen för tidtagning.

Sven Wickberg
<1384>

UFD-DOSET.

I ABC806 och ABC802 finns det ett nytt dos, UFD-doset. Det kallas även för superdoset. Utmärkande för detta är att man kan köra olika typer av floppydiskar. UFD-doset kan dessutom erhållas som option till ABC800 M och C.

Drive:arna adresseras inte som DR0 eller DR1. De olika typerna av drive:ar har i stället olika enhetsnamn. Man kan vid uppstart köra ett initieringsprogram som gör att DOS:et tolkar DRX som en valfri typ av drive.

Sektoradresseringen är helt linjär i UFD-doset. Detta innebär att gamla program som läser och skriver direkt på en sektor inte längre fungerar. Man kan med UFD-doset välja gammal sektoradressering för en viss typ av drive:ar. Detta görs vid initieringen av drive:arna.

Enheter:

MF motsvaras av ABC832, MF är en förkortning för minifloppy. Bitkartorna finns i sektorerna 14-15. Klusterstorleken är normalt 4. Flexskivan är en 5 1/4 " skiva med 80 spår och dubbelsidig.

MO motsvaras av ABC830, MO är en förkortning för minifloppy ordinary. Bitkartorna finns i sektorerna 6-7. Klusterstorleken är normalt 1. Flexskivan är en 5 1/4 " skiva med 40 spår och enkelsidig.

SF motsvaras av ABC838, SF är kanske en förkortning för superfloppy? Bitkartorna finns i sektorerna 14-15. Klusterstorleken är normalt 4. Flexskivan är en 8 " skiva.

Klusterstorleken:

DOS:et kan inte adressera alla sektorer direkt. Därför hanteras sektorerna "i klump". För t ex ABC832 drive:ar är sektoradresseringen normalt 4. Detta innebär att en bit i bitkartorna motsvaras av 4 sektorer för ABC832. Bitkartorna kan förklaras som två kartor över skivan. En bitkarta anger vilka sektorer som är upptagna och den andra anger vilka sektorer som är användbara. Även om man kan ändra klusterstorleken så bör man undvika detta. Dels kan man förlora kompatibiliteten med andra skivor skrivna av andra ABC-datorer. Dels kan man kanske få problem om en bitmapp inte rymmer inom en sektor.

Initiering av doset:

Rent allmänt gäller att relokerbara rutiner kan laddas genom att de anropas av en ADDOPT.ABS fil. Att rutinerna är relokerbara innebär att de kan placeras i minnet på vilken adress. Alla relokerbara rutiner är objektkod i ASSEMBLER, som har extensionen ".REL". För att skapa en ny ADDOPT.ABS kan lämpligen Mats Knuts program CONFIG.806 användas. Det finns på klubbens monitor. Observera att en gammal ADDOPT.ABS fil måste finnas på en av drive:arna.

Med hjälp av programmet SYS.BAC kan man ändra parametrar för diskenheterna. Sedan kan parametrarna sparas i en fil med namnet DEVDES.REL. Programmet SYS.BAC visar först vilka enheter som finns och vilka parametrar som gäller. Man väljer drivetyp, sedan kan man ange typ av styrkort, kortadress och sektoradressering, samt

gammal eller ny sektoradressering. Dessutom väljer man vilka enheter som skall tolkas som DRX.

Under förutsättning att parametrarna för diskenheterna laddas in genom att en ADDOPT.ABS laddas in i filen DEVDES.REL kan programmet DRINI.BAC sedan se till så att DOS:et tolkar vald enhet som DRX.

Man kan med hjälp av programmet PREABS.BAC skapa en autostartsekvens i filen BASICINI.SYS. Det går att få autostart både på en ABS-fil och en BASIC-sträng. Normalt anger man autostart på filen ADDOPT.ABS och sedan anger man RUN DRINI i BASIC-stängen. Vill man ha vissa options inställda så kan man sedan skriva in dessa antingen i programmet DRINI eller i autostartsträngen. Om man skriver det i autostartsträngen, så måste man se till så att man har blanktecken ifall man anger EXTEND. Man kan skriva så här:

```
INTEGER : ATTRIBUTE 3 : EXTEND :  
CHAIN 'DRINI.BAC'
```

Observera att man måste skriva CHAIN och inte RUN! RUN kan bara skrivas först på raden.

I DRINI finns det en möjlighet att länka in ett vanligt program. Om man vill att DRINI skall avslutas utan CHAIN, kan man ange enheten 'NUL:' som nästa fil i DRINI. CHAIN 'NUL:' har alltså samma effekt som om man kunde skriva NEW i ett program.

LIB-spår:

Spår 1 användas inte, som på gammalt DOS för ABC830, till backup för libspåret (spår 2). Dessutom utnyttjas även spår 3 som ett libspår. Detta innebär att man får plats med 240 filer på en skiva, dvs 2*8*15.

Storleken på filerna finns i LIB spåret, omedelbart framför filnamnet! En pokearea saknas eftersom UFD-doset använder sig av den area som var pokearea i en ABC800 med gammalt dos.

User File Directory - UFD

Om man använder en winchesterdisk går det inte att ha ett och samma bibliotek för några tusen filer. Därför har man skapat ett system för användardefinierade bibliotek, som kallas för User File Directory, UFD. Biblioteket för ett UFD finns i en vanlig fil, som tar upp 18 sektorer, vilket är lika med 2 bitkartor och 16 bibliotekssektorer. Namnet på UFD:et framgår av filnamnet för punkten. Extension är alltid ".ufd". Observera att det är gemena bokstäver bara för att man inte skall kunna öppna filen under systemet. Det är ju så att operativsystemet inte kan "hitta" en fil med gemena bokstäver eftersom filnamn alltid konverteras till versaler.

Man skapar, ändrar namn på, listar, aktiverar UFD med programmet UFD.BAC. UFD.BAC anropar i sin tur programmet UFD.ABS.

Normalt använder man UFD i ett ABC-NET, men det kan även användas i en användarsystem.

Ett UFD anropas som enheten UFD:. I minnet på adress 65529 finns det en uppgift för vilken drive som aktuell UFD finns på.

På adressen 65527 finns i 2 byte adressen till den första bibliotekssektorn för ett aktuellt UFD. Är dessa 0, så är inget UFD aktuellt. Adress 65526 kallas för UFD-reset. Denna byte har den funktionen att om det står 165 där, nollställs inte adressen till ett aktuellt UFD. Det innebär alltså att ett aktuellt UFD inte deaktiveras vid reset. Däremot nollställs UFD-resetbyten vid reset, så man måste efter varje reset, skriva 165 i adress 65526 för att få samma effekt vid nästa reset.

Denna lösning gör att datorn inte kan adressera mer än ett UFD i taget, vilket innebär att man inte kan kopiera mellan olika UFD. Kopiering måste alltid ske mellan ett UFD och ett MFD, dvs Master File Directory, huvudbiblioteket.

NY MONITOR.

På grund av att den enda telefonlinje som idag finns till klubbens monitorsystem idag utnyttjas till bristningsgränsen har klubbens styrelse för avsikt att bygga ut systemet så att fler medlemmar kan utnyttja det. ABC-Klubbens styrelse har därför tillsatt en grupp för att arbeta fram en ny monitor.

Det har varit monitorgruppens avsikt att modifiera det nuvarande systemet temporärt så vi skulle kunna använda två ABC80, med hjälp av ett nätverk, till en gemensam floppydisk. Detta har emellertid visat sig vara omöjligt. Nu blir det alltså inget tillfälligt system, utan vi startar direkt med att utveckla ett helt nytt system.

Klubbens styrelse har när detta skrivs (831113) inte fattat några beslut om hur det nya systemet skall se ut. Det jag skriver här är alltså monitorgruppens egna ambitioner.

Monitorgruppen har för avsikt att använda ABC802 som datorer i det nya monitorsystemet, samt en 40 Mbyte winchesterdisk. Vi beräknar att vi kan öppna ett nytt system tidigast i mars 1984. Vi kanske kan visa upp ett testsystem vid årsmötet den 18 februari 1984. När systemet tas i bruk kommer det att finnas två stycken modemingångar för 300 bps (V21). Senare hoppas vi även kunna erbjuda fler ingångar. Vi tänker oss att dessa nya ingångar skall vara för split speed 1200/75-110 bps (V23 i tillämpliga delar) och 1200 bps full duplex (V22). Full duplex kommer att användas för alla typer av ingångar.

Det nya systemet kommer att bestå av två huvuddelar. Dels det egentliga monitorsystemet med klubbens programarkiv. Här kommer alltså alla program som klubben har att finnas tillgängliga, både sådana som har givits ut på kassett och andra. Dock har vi tänkt oss att nya program som vi får in, inte skall kunna hämtas från systemet förrän klubben har granskat dessa.

I en senare etapp har vi för avsikt att utveckla ett konferanssystem som skall ha funktioner liknande KOM på Q-Zentralen. Tills vidare kommer monitorsystemet att innehålla speciella textbibliotek som kommer att kunna utnyttjas ungefär som i det nuvarande systemet.

Medlemmar i monitorgruppen är Gunnar Tidner, Stig Löfgren, Kalle Linström, David Aronson, Lars-Göran Göransson, Benny Löfgren och Bo Kullmar.

Monitorgruppen gm Bo Kullmar

ABC806.

ABC806 är den största datorn i ABC800 serien. Till det yttre skiljer den sig markant från sina släktingar. Tangentbordet är som på ABC802 löst, men datorn finns i en burk med måtten 500 x 350 x 70 mm (b x d x h). Bildskärmen är lös och kan väljas fritt. Normalt använder man ABC812, som är en färgskärm. Man kan även använda ABC815, som används till ABC800 M. Dessutom går det att använda ABC810/ABC811 som normalt används till ABC800 C, samt även den lilla ABC815.

Inom skåpet finns en del nyheter och vissa saker är borttagna. Här är en sammanfattning:

Bättre högupplösningsgrafik, 4 grafikbilder kan samtidigt hanteras

Bättre hantering av bildskärmen, med softscroll och attributminne för styrtecken

Dataninnet för grafikbilder kan när det inte användes till detta utnyttjas som en RAM-floppy

Batteriförsörjd klocka

Lågupplösningsgrafik saknas (SET DOT, TX-POINT mm)

Kassettrutin saknas

Terminalrutinen måste laddas i RAM-minne från disk

Datorn är utrustad med UFD-dos (superdoset)

Med datorn skall det följa en systemskiva som bland annat skall innehålla rutiner för RAM-floppy, softscroll och terminalrutin. När detta skrivs finns det bara en preliminär version av systemskivan för ABC806. När ABC-BLADET har gått ut till medlemmarna bör den slutgiltiga versionen vara klar.

För den preliminära versionen anger Luxor i ett servicemeddelande att det finns en bug (ett fel) vardera i RAM-floppyn och terminalrutinen. Jag har den erfarenheten att man inte skall använda softscrollrutinen tillsammans med terminalrutinen på grund av en bug.

Högupplösningsgrafik:

Högupplösningsgrafik kan genereras på tre olika sätt.

Kompatibel mode, dvs grafik enligt ABC 800 med en upplösning av 240 x 240 punkter, där färgerna väljs med FGCTL nr. Nr är ett tal mellan 0 och 255. Observera att FGPAINT är "förbättrad", vilket innebär att gamla grafikprogram kan ge ett felaktigt resultat.

Grafik i fyra färger med 512 x 240 punkter. Färger väljs genom att man anger dem i klartext, t ex FGCTL RED+YEL+BLU+CYA.

Grafik 16 färger med 256 x 240 punkter. Färger väljs på samma sätt som för grafiken med 512 x 240 punkter. Det finns inte mer än 8 färger i datorn, om man räknar in vitt och svart. Däremot kan man för varje färg få både bakgrundsfärg och förgrundsfärg. Det senare motsvar färgnummer 127 i den kompatibla moden.

Maximalt kan man arbeta med 4 grafikbilder. Man kan uppdatera en bild och visa en annan. Detta styrs med instruktionen FGPICTURE.

Vid skifte mellan den nya grafikmoden som väljes med FGCTL RED+YEL+BLU+GRN och den gamla som väljes med FGCTL nr raderas den grafikbild som är i uppdaterings-

mode. Detta kan utnyttjas för att blanka ut grafikbilden!

Bildskärmshantering:

Genom att ladda in en rutinen SOFTOPT.REL kan man få softscroll. Detta innebär att bildskärmen rullas, inte rad för rad, utan den "flyter" uppåt utan att "hacka" uppåt radvis. Softscollen är ganska långsam, men den är en oerhörd skönhetsupplevelse för ögonen! Det går inte att beskriva den, den måste upplevas - ses!

Argument för färg mm lagras i ett speciellt attributminne. Detta minne är inte ett "vanligt mappat minne" utan nås genom INP(53) och OUT,53,nr. Man kan skicka ett OUT kommando t ex OUT 53,1 Detta får som resultat att promptern ABC806 skrivs ut i röd färg. Hur man skall läsa av attributminnet för en utskrift av bildskärmen känner jag inte till. Hur stryptecknen ser ut i attributminnet har jag räknat ut, men hur DBLE och EL behandlas har jag inte kommit på.

Bit 0 är 1 om = FLSH

" 1 " 1 " = ULN (understryckning)

" 2-4 innehåller färgnummer för bakgrundsfärg

" 5-7 " " " textfärg

Färgnummer är kodat enligt följande:

0 BLK

1 RED

2 GRN

3 YEL

4 BLU

5 MAG

6 CYA

7 WHT

Man kan alltså prova att skicka attribut till attributminnet med kommandot OUT 53,Nr. Nr skall sättas ihop enligt ovan.

Med ATTRIBUTE 0 kan man få kompatibel mode med ABC800 C, dvs varje styrtecknen tar upp en position på bildskärmen. Däremot finns fortfarande styrtecknen i attributminnet och inte i bildminnet som på ABC800 C.

RAM-floppy

Genom att ladda in rutinerna OPTROSH.REL och OPTROSL.SYS kan man utnyttja det extra minnet som en RAM-floppy. Det är en disk typ ABC832 som simuleras. Den hanteras då som en vanlig floppy, med selectkoden 29 om man vill skriva eller läsa direkt på en sektor. Man får då tillgång till 512 sektorer, 32 sektorer tar doset åt sig, så kvar har man 480 sektorer. Rutinen för RAM-floppy kan hantera upp till 512 Kbytes minne, om man kompletterar med 3 st 128 Kbytes kort i en expansionslåda. RAM-floppyn adresseras med enhetsnamnet RAM:

Vid reset raderas endast drivrutinen, själva RAM-minnet är intakt. Givetvis förstörs innehållet i RAM-floppyn när strömmen bryts till datorn. Man bör komma ihåg att lagra över filer som man skall spara till floppydisken innan man stänger av datorn. Vid uppstart kan man automatiskt kopiera över ett lämpligt antal filer till RAM-floppyn.

Vissa standardprogram såsom ORD800 och KALKYL800 använder sig av RAM-floppyn.

Extraminnet kan även kopplas in med OUT instruktioner, i segment om 8 Kbytes, så att datorn adresserar det i stället för det normala minnet. Hur man gör detta bör framgå av bruksanvisningen till ABC806:s RAM-floppy.

RAM-floppy kan erhållas till de andra 800-orna också, samt även till ABC80. Jag vill betona att RAM-floppyn verkligen är

en bra sak, som jag varmt kan rekommendera! Med en RAM-floppy kommer man förbi svagheten med långsamma floppydiskar till ett pris som avsevärt understiger vad man skulle få betala för en winchesterdisk. Jag jämför här med en winchesterdisk eftersom den typen av sekundärminnen är mycket snabba.

Klockan:

En batteriförsörjd klocka finns i datorn. Den utnyttjar CMOS minne. Man måste med ett speciellt program ställa datorns klocka från CMOS klockan. Detta gör man lämpligen med Luxors program DRINI, där denna rutin finns inlagd. En speciell rutin finns för att ställa CMOS klockan. Klockan går såvitt jag kan se mycket exakt. Om man skulle vilja köra en ABC806 kontinuerligt, så skulle jag därför rekommendera att man ställer datorns klocka från CMOS klockan en gång per dygn.

Lågupplösningsgrafik saknas:

Eftersom det inte finns plats för lågupplösningsgrafik i datorn, har man tagit bort den. Instruktioner som SET DOT, CLR DOT och TXPOINT ger felmeddelande nr 200, Enheten ej ansluten. För ABC80 användare kan jag nämna att TXPOINT är det samma som SET DOT, med den skillnaden att origo finns i nedre vänstra hörnet. ASCII-grafiken finns däremot kvar.

I bildminnet utnyttjas ASCII 0-31 och 128-160 för ASCII-grafik. Genom att skriva PUT \$0,Chr\$(1) får man ASCII-grafikttecknet som mostvaras av ASCII 33. Det kan även erhållas genom ;GWHt Chr\$(33) eller ;GWHt "!".

ASCII grafik nr	ASCII i bildminnet
33 - 63	0 - 31
96 - 127	128 - 160

Kassettrutin:

Även kassettrutinen har tagits bort av platsbrist! Man kan om man vill skriva en egen kassettrutin och utnyttja en av V24 ingångarna. Men varför skall man använda kassett?

Terminalrutin:

Terminalrutinen får inte heller plats i PROM: en i datorn, därför måste den laddas in genom filen TERMOPT.REL.

UFD-doset:

Datorn är utrustad med UFD-doset, vilket även kallas för superdoset. Se speciell artikel.

Några personliga reflektioner:

Jag är mycket nöjd med ABC806. De nackdelar som finns för ABC800 C har lösts med denna dator. Framför allt är färgskärmen med softscroll en stor njutning! Även RAM-floppyn förtjänar att nämnas, den är mycket bra!

Det börjar bli trångt i minnet, eftersom RAM-minnet måste utnyttjas för systemrutiner. Därför kan man kanske säga att ABC 806 är för bra för en 8 bitars dator! Man skulle behöva lite större direktadresserat minne än 64 Kbytes! Man tänker då på en 16 bitars dator från Luxor!

Det är så att man med just ABC806 är ganska förberedd för en ny kraftfullare mikrodator, genom att man endast behöver byta datorburk om man vill byta upp sig till en ny 16 bitars dator från Luxor! När den nu kommer, 1984 ???

Av Bo Kullmar

Systemvariabler i ABC800.

ABC800 datorer använder sig av en area högst upp i minnet för att lagra data som systemet använder sig av. Denna area omfattar minnesadresserna 64768 - 65535. Om man inte har UFD-doset är dock 65024-65279 ledigt för t ex egna assemblerrutiner. En omfattande beskrivning av adresserna finns i boken Bit för Bit. Här skall jag endast nämna en del adresser och lite om hur de kan användas. B betecknar en byte (ett tecken) och W betecknar ett ord (två tecken).

65280

Denna adress erhålles med SYS(10). Enligt Carl-Axel Petterson sätts adressen till 1 om CTRL-C har tryckts.

65286 W BOFA

Pekare till programmets början, dvs egentligen programinformationsblocket. Det är lika med värdet som fås med SYS(11).

65288 W EOFA

Pekare till sista byten i BASIC-programmet.

65290 W HEAP

Pekare till första lediga byte i minnet.

65292 W BOTTOM

Pekare till första byte av minnet som används till program m m. Om man höjer denna pekare kan man få minnesutrymme ledigt för t ex assemblerprogram.

65294 W TOP

Pekare till sista byten av minnet som används till program m m. Om man sänker denna pekare kan man minnesutrymme bli ledigt för t ex assemblerprogram.

65309 B FL

Diverse flaggor

Bit 2 = 1 då är blanktecken signifikant (EXTEND mode)

Bit 1 = 1 indikerar EXTEND mode

Bit 0 = 1 indikerar INTEGER mode

65310 B PREC

Flyttalsprecision. Innehåller antalet bytes lagringsutrymme som de två varianterna av flyttal upptar. Vid SINGLE precision är värdet 4 i 65310. Vid DOUBLE precision är värdet 8 i 65310.

65311 B DIGITS

Innehåller antal siffror som skrivs ut med PRINT, standardvärde är 6/16 beroende på flyttalsprecision. Ändras med BASIC-instruktionen DIGITS. Observera att maxvärdet är 7/17 siffror.

65314 B DEFLOW

OPTION BASE. Innehåller lägsta vektorindex. Ändras med BASIC-instruktionen OPTION BASE.

65315 B INT

Interrupt-byte

Bit 3 = 1 anger SINGLE STEP mode

Bit 2 = 1 anger DIREKT mode, = 0 anger programkörning

Bit 1 = 1 anger TRACE mode

Bit 0 = 1 anger att CTRL-C har skrivits in

65318 B PRSTAT

Programstatus

Bit 7 Ny errorhantering används

Bit 6 HR-grafik används

Bit 5 Program högt i minnet

Bit 4 Flyttal finns i COMMON

Bit 3 Flyttal är allokerade

Bit 2 Dubbel precision

Bit 1 Listskydd, att nollställa denna byte på fil gör normalt inte att man kan läsa upp listskydd

Bit 0 Fixed-up

65326 W VARBAS

Pekar på variabelistan

65332 B USERCS

Senast valda I/O kort.

65333 B TRCLU

Logisk enhet vid TRACE, innehåller nummret på den fil som TRACE § skriver på.

65348 B ERRCOD

ERRCODE, obs ERRCODE nollställs ej vid RUN.

65362 B

Aktuell kolumn för bildskärmen

65363 B

Aktuell rad för bildskärmen

65364 B WID

Aktuell radbredd för bildskärmen

65403 W DEVTBA

Pekare till länkad enhetslista. Denna lista används om man vill länka in egna enheter.

65405 W STMTBA

Pekare till länkad instruktionslista. Denna lista kan användas om man vill lägga in egna instruktioner.

65407 W FNKTBA

Pekare till länkad funktionslista. Liknar ovanstående men avser funktioner i stället för instruktioner.

65413 W CTRLCPTR

Pekare till CTRL-C. På adressen (normalt 65315) som denna systemvariabel pekare på sätts bit 0 till 1 när CTRL-C skrivs in.

Stäng av CTRL-C-funktionen = POKE 65413,0
Återställ CTRL-C-funktionen = POKE 65413,35,255 (65315).

65415 W CMDUNSA

Hopp till UNSAVE-rutinen. Hit hoppar datorn när den får kommandot UNSAVE. Genom att lägga in RET (Poke 65415,210) här kan man stänga av kommandot UNSAVE. Man kan också lägga in ett hopp via en egen rutin, som då utföres när kommandot UNSAVE ges.

65421 W HRCRLR1

Släck högupplösningsgrafik, anropas bl a av LIST-kommandot. Hopp till denna rutin innebär att systemet gör FGCTL 0. Funktionen kan stängas av genom att skriva RET i 65421 (POKE 65421,201). För att sätta på funktionen igen måste man återställa de ursprungliga värdena. Detta kan ske genom att först ta reda på vad det finns för adress där genom Pekare=PEEK2(65421) och sedan återskriva adressen med POKE 65421, Pekare, SWAP%(Pekare).

65424 W CONSI

Hopp till rutinen som läser ett tecken (GET) och lägger resultatet i register A.

65430 W USERTRAC

Hopp till en subrutin som skriver ut radnummer vid TRACE. Genom att i stället göra hopp till en egen TRACE-rutin så kan man använda en annan TRACE-rutin än den som finns normalt. Se Bit för Bit.

65433 B CASSPEED

Hastighet vid skrivning på CAS. Normalt är innehållet 40 vilket avser 2400 bps. 0 är lika med 700 bps, d v s den hastighet som ABC80 normalt använder. Carl-Axel Petterson rekommenderar en test med t ex 25.

65460 W

Interrupt-adress för tangentbordet. Används vid interrupt från tangentbordet (när någon trycker på en tangent).

65494 W

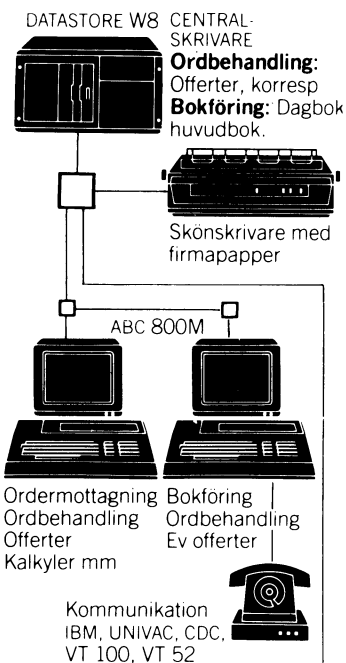
Interruptvektor för klocka. Ett hopp hit sker var 1/100 sekund. Hoppet sker till en rutin som ökar innehållet i 65525 med 1.

NYA ABC NET

MULTIUSER

EXEMPEL

Fleranvändarsystem.
Administration.



Fler exempel på fleranvändarsystem i vår specialbrochyr. Ring, skriv eller slink in!

T-D-X

SmåDatorer AB

T-D-X NORR: SOLLENTUNA-VAGEN 225, 111 23 SOLLENTUNA

T-D-X CITY: KUNGSGATAN 79, KUNGSHOLMEN STOCKHOLM

GEMENSAMT TELNR 08-96 01 80

Om man låter hoppet gå till en egen rutin kan man t ex låta datorn kontinuerligt skriva ut klockan på bildskärmen. Se programmen SYSTIME.800 eller TIME.806 som utnyttjar detta. Programmen finns på klubbens monitor.
65506 B

Tangentbordsflagga. Denna byte innehåller 128 om en tangent är nertryckt. Erhålles också med SYS(5).
Nollställs med GET, INPUT och INPUT LINE. Kan givitvis också nollställas med POKE 65506,0
65507 B

Tangentbortsbuffert. Innehåller ASCII-väret för senast nedtryckta tanget. Jmf SYS(6).
65519 B YEAR

År för Time\$ (0-99)
65520 B MNTH

Månad för Time\$ (1-12)
65521 B DAY

Dag för Time\$ (0-31, 0 = stopp klocka)
65522 B HOUR

Timme för Time\$ (0-23)
65523 B MIN

Minut för Time\$ (0-59)
65524 B SEC

Sekund för Time\$ (0-59)
65525 B TIC

Räknare för Time\$ (0-93)

65526 B

UFD-RESET. Om innehållet är lika med 165 nollställs ej ett ev aktivt UFD vid RESET. Observera att 65526 alltid nollställs vid RESET, så man måste skriva dit 165 efter varje RESET som sker för att få funktionen kvar.
65527 W

UFD-offset. Adress till ev aktivt UFD-bibliotek. UFD är ett användardefinierat bibliotek som kan användas om man har UFD-DOS eller ABCNET-DOS.
65528 B

UFD-drive. Drivenummer för ett aktivt UFD.
65529 B

RAM-floppy. Finns en RAM-rutin innehåller adressen 165 vill man att RAM-floppyn skall formateras (nollställas) skrivs 0 här och formateringen sker vid nästa RESET om drivrutinen finns på tillgänglig vid RESET.

PS Adresserna 65526-65528 är enbart aktuella för datorer utrustade med UFD-doset.

Carl-Axel Petersson <2229>

Bo Kullmar <1789>

ABC-Stockholm på Tekniska Museet eller hur QZ kan användas

Lördagen den 12 november kom ett meddelande på QZ från Tekniska Museets Ungdomsförening, som på fars dag - d v s dagen efter meddelandet - skulle ordna en s k aktivitetsdag med praktiska demonstrationer av olika slag. I år ville man gärna ha hjälp med att visa datorer och vad de kan användas till och vände sig av den anledningen till ABC-klubben.

Det var flera som läste meddelandet och gjorde sig beredda att agera. Förberedsetiden var ju inte så lång, men en grupp formerades sig snabbt med Göran <2771>, Allan <3435> samt Kjell <700> med sonen Håkan <3673>. De gjordes ett antal snabba ryck och med klubbens och privata datorer ställde man upp på museet och tänkte visa administrativa program, grafik, frågesport och kommunikation för att ge en mångsidig belysning av datorn som praktiskt hjälpmedel.

Museets unga besökare visade sig dock vara inne på en något annan linje, och det tog inte många sekunder innan de anstörde entusiasterna hade slagit CTRL C och laddat in 'Starfighter' och satt full fart på rymdkampen!

Det som därutöver intresserade var främst dator-kommunikation med QZ och hela övningen var ett bra exempel på nyttan av denna form av samband.

I ABC-Stockholms bibliotek finns följande litteratur:

Böcker

ABC om användardokumentation
ABC om BASIC
ABC om mätsystem
ABC om programmering och dokumentation

Avancerad programmering på ABC80
Basic Computer Games
Basic Handboken
Bygg ut ABC80 med DataBoard 4680
Mikrodatorns ABC
The Basic Handbook
Your First Computer

Bruksanvisningar och manualer

ABC80 bruksanvisning
Datadisk 82-84 d:o
Datadisk 88 d:o
Flexskiveenhet ABC d:o
Servicemanual ABC80
Servicemanual ABC800
Z80-Z80A CPU Technical Manual

Ovanstående litteratur förvaras i klubb-lokalen och är tillgänglig under tisdagskvällarna. Ingen hemlåning tillåts.

ABC STOCKHOLM

Vidängsvägen 1
161 33 BROMMA Postgiro 40 42 52-9

Ordf Stig Löfgren <872>
sekr Bo Hjulström <557>,
kassör Kjell Järbin <700>.

Flera medlemmar i stockholmskretsen

Det under hösten utsända informations- och värvningsbrevet till ABC-klubbens stockholmssektion har gett en glädjande stort medlemsökning. 1983-11-09 var antalet 314. Alla som bor i Stor-stockholm eller inom bekvämt reseavstånd från Alvik är välkomna att bli medlemmar. Alla får plats i klubben. De fördelar du får via lokalklubben är i korthet följande:

Informationsmöten med föredrag och demonstrationer i olika ämnen. Som lokalmedlem kommer du in gratis, annars kostar det 10:- per kväll.

Öppet hus på klubblokalen varje tisdag kl 1830 - ca 2130. Där finns möjlighet att diskutera med andra medlemmar, köra program på klubbens datorer, köpa visst förbrukningsmateriel, läsa klubbens böcker och tidskrifter etc. Så småningom kommer vi att kunna köra ABC-klubbens monitor från en lokal dator till hjälp för alla som inte har telefonmodem.

I klubblokalen finns också en anslagstavla för klubbaktiviteter, köp och försäljning etc. Dessutom bjuder klubben på te.

Kommande stockholmsaktiviteter

Onsdag 25 jan 1984

Information om JSP (Jackson Structured Programming).

Onsdag 7 mars

Årsmöte i ABC Stockholm.

Där ej annat sägs äger alla informationsmöten rum kl 1900 i Vidängssalen, Vidängsvägen 1.

Aktiviteter 1984 under förberedelse

Februari

Studiebesök. (program ej fastställt)

Onsdag 11 mars

Informationsmöte. (program ej fastställt)

Kursverksamhet

Vi planerar att arrangera datororienterad utbildning i form av studiecirklar. Den första torde bli assemblerprogrammering. Problemet för närvarande är att vi saknar cirkelledare, och vi skulle mycket gärna vilja få kontakt med en person med administrativ och pedagogisk intresse för organisation av kursverksamheten. Lärare och tekniska experter finns, men det är svårt att få tiden att räcka till även för den organisatoriska överbyggnaden.

Är det någon som har ideer till ämnen för informationsmöten och studiebesök eller som känner för att hjälpa till med kursverksamheten, går det bra att ringa till någon i styrelsen.

Tekniska projekt inom sektionen

Bosse <557> har en konkret ide till en hårdvarutillsats som ger cursor-styrning och funktions-tangenter via ABC 80:s V24-kontakt. En "standard" joystick (t ex Atari) kan användas som styrdon. Beskrivning planeras i ABC-bladet.

En klubbmedlem har tagit fram en billig 32 K minnesexpansion för ABC 80, som kan byggas för runt 200:-. Beskrivning kommer troligen i ABC-bladet eller i en ABC-rapport.

Att rädda flexskivor

På informationsmötet den 19 okt berättade Rune Sagnell <1602> om hur data är organiserade på 5 1/4 " flexskivor och demonstrerade en nytt program "FILTEST" med vars hjälp man kan läsa och skriva direkt i skivan. Programmet är skrivet av Kai Liljeblad <806> och är uppbyggt på ett sådant sätt att det är lätt att både lära sig och förstå skivans inre hemligheter. Det finns också ett antal funktioner för analys av felaktigheter, "reparation" av bitmap, biblioteket etc.

Ett ofta förekommande inslag på informationsmötena är och demonstration av program från ABC-kassetterna. Den 19 okt visades programmet i högprecisionsmatematik och hur man kan använda detta för bl a statistiska beräkningar. Det var ett verkligt intressant exempel på hur en mikrodator kan utnyttjas inom ett område som inte kan hanteras manuellt och där man tidigare fått ta till stordatorer.

Kort och Brett om

DEC-10.

Här kommer nu ytterligare 16 sidor av skriften Kort och Brett om DEC-10.

Skriften kommer att publiceras efter hand i mån om plats.

Den som vill ha tillgång till skriften i original kan beställa den direkt från QZ, 08-67 92 80. Tyvärr kan man inte beställa den genom ABC-klubben eller Q-Zentralen.

VAD FINNS PÅ

ABC-KASSETTERNA.

Som ny medlem i ABC-klubben kan det vara svårt att veta vad ABC-klubbens distribuerade program innehållit och var dokumentation och beskrivningar till dessa står att finna. Nyinträdande medlemmar får alltid retroaktivt det innevarande kalenderårets tidningar och kassetter. Om Du blir medlem nära ett årskifte meddela därför vilket kalenderår Du avser.

För att översiktligt kunna beskriva vad som finns på ABC-kassetterna har här gjorts en indelning till några få typer. Antalet programfiler på de hittills distribuerade 8 kassetterna uppgår till 196.

Några av grupperna överlappar varandra och för en del program kan tillhörigheten diskuteras. Extensionen har tagits med då det behövs för att kunna särskilja en fil från en annan. Först på varje kassett ligger alltid ett program som möjliggör en enkel överföring till diskett. Detta har inte tagits upp i denna förteckning.

I ABC-bladet publiceras fortlöpande en noggrannare dokumentation över hur de olika programmen fungerar och vad de åstadkommer. Här nedan upptas huvuddragen av vad som hittills har publicerats i ABC-bladet av dokumentationer och beskrivningar till program-paket.

ABC-blad 2,1982:Kassett 1, 2 och 3.
ABC-blad 3,1982:Kassett 4, 5, 6, matematik-programmen, LISP 1.5, assembler-paketet och HJÄLPARE.
ABC-blad 1,1983:Kassett 7, 8.
ABC-klubbens uppringbara Monitor-system presenteras i ABC-bladet 4-5,1981.
ABC-klubbens terminalrutin ABCV24 presenteras i ABC-bladet 2,1982.
ABC-klubbens textbehandlingsprogram, TV-editor, behandlas utförligare i ABC-bladet 1 och 2, 1982.
ABC-klubbens implementering av fig-FORTH på ABC80/ABC800 presenteras i ABC-bladet 2,1982 samt 1 och 2,1983.

Utility

Inom denna grupp finns program som hjälper vid programmering och hantering av program.

CAS 1:
FILSUM, FILCOM, CONCAT, SORTCORE, KONVERT, DÖP, IDMÄRK, CKOLON
CAS 2:
CASLIB, NYMEMDMP, EXTDEL, HUVUD, MENY, TIME, LÄSDISK, FILOVM, DISKST, DIRFIX, TVSET, SNYGGBAS, COPYLIB.BAC, COPY.BAC, LIB.ABS, DOSGEN
CAS 3:
CHECK, ASSIGN, DISASEM.REM, DISASEM.BAS, OPLIST, OPCODE1.TXT, OPCODE2.TXT
CAS 4:
HJÄLPARE.REM, HJÄLPARE.I6K, HJÄLPARE.32K, HJÄLP16K.ASM, HJÄLP32K.ASM, RESETFIX, FILDIS.REM, FILDIS
CAS 5:
CRTOS.INF, CRTOS.BAS, NOCTRL.TXT, NOCTRL.TXT, CMDINT.CAS, SD10
CAS 6:
CASCOPY.REM, CASCOPY.BAS, CASCOPY1.BAC
CAS 7:
RUNONLY, NOTONLY, DEVELOP.TXT, DEVELOP.BAS
CAS 8:
HJÄLPARE.REM, HLP16.BAC, HLP32.BAC,

HLPB16.BAC, HLPB32.BAC, ASSTRANG.BAC, ASSTRANGH.BAC, ASSTRANGL.BAC, BOFA, SÄNK, SÄNKIF, GOLVTAK, ERRORSYS.BAC, BYECHAIN

Ord- och textbehandlingsprogram

Programmen inom denna grupp framställer och hanterar rena textfiler på en mängd olika sätt.

CAS 1:
MAKETEXT, VISA
CAS 2:
TV.TXT, TV.BAC, TVMAIN.BAC, TVSUBR.ABS
CAS 3:
TVEDIT.REM, TVCAS.BAC, TVSUBR.BAC, TV80.BAC, TVMAIN2.BAC, RADJUST
CAS 4:
INUTFIL, TV.BAS, TVMAIN3.BAC, TVSUBR.ABS, TVLIB.MRG

Kommunikationsprogram

Inom denna grupp finns program som används i samband med att ABC80 - ABC800 skall fungera som terminal, främst över ett modem, asyckron trafik med 300 baud.

CAS 1:
ABCMINI, CASMINI, ABCTTRANS, LOGAIN
CAS 3:
ABCV24, CASSEND.BAS
CAS 4:
ABCTTRANS.80K
CAS 5:
CASMON, SAFT, FILTRANS.REM, FILTRANS.BAS
CAS 8:
CAS80.BAC

Registerprogram

CAS 2:
REGISTER
CAS 3:
CASPROMP.BAC, CASBLOCK.REM, REGCASPR.BAS

Läroprogram

Denna grupp innehåller program med pedagogisk inriktning.

CAS 1:
EUROFRÅG, MATTEN, ROTEN, PRIM, LÅN, STÅLTEMP, OHMDATA
CAS 2:
DECROM, OMVANDLA
CAS 3:
MORSE.BAS
CAS 6:
MORSEÖVN

Spelprogram

CAS 1:
MAXEN, SLALOM, REAKTION, GUBBEN, KRIG, LUFFAR, VIRRE, MYSKO
CAS 2:
MAXEN, LABYRINT, GOMUKO, INVANDER, HINDLOPP
CAS 3:
LANTERNA, UBÅT, ELMÄTARE
CAS 4:
GLIPP.XXX, GLIPP.STY, GAPHALS
CAS 5:
FYLLE, TVSPEL, DEFENDER, HUSET
CAS 7:
GODJUL82, BREAKOUT, SOLITÄR, SKYTTE, LUFTVÄRN
CAS 8:
GENESIS, ROADRACE, GODJUL2

Grafikprogram

Program som huvudsakligen använder sig av grafiken.

CAS 1:
MÖNSTER, KULGRAF
CAS 4:
GRAFPRIN.PRT
CAS 5:
BIGTEXT.BAS, BIGTXT12, BIGTXT69, BIGTXT57
CAS 6:
PAUSBIRD, GRAFEDIT.HLP, GRAFEDIT.BAS, ABC80.BAS, EMBLEM.PIC

Ljudprogram

Program som huvudsakligen använder sig av ljudgeneratorn.

CAS 1:
SLOOP
CAS 2:
BLINKA, GUBBENOA
CAS 3:
MUSIK, BRAMUSIK
CAS 5:
TONTXT, TONE, ROCK, TONI

Matematikpaketet

Matematikpaketet är en samling program som gör att ABC80 kan användas för högprecisionsberäkningar med upp till 4000 säkra siffror.

CAS 7:
HÖGPREC.INF, PROG1.BAS
CAS 8:
INFO.BAS, MASK.BAS, RPN.BAC

fig-FORTH

Ett helt nytt språk implementerat på alla ABC80 med olika checksummer och minnesutbyggnader. **CAS 4:**
DOSCREEN.BAS, SCREEN.TXT, FORTH16.ABS, FVISA.BAS1

CAS 5:
SCREEN2.TXT, FORTH32.ABS
CAS 6:
DOSCREEN.MRG
CAS 7:
CASFORTH.REM, CASCREEN.BAS, INSCREEN, SCREEN.C, CMDINT.CAS, CASFORTH.ABS, CASCRUT.BAS
CAS 8:
SCREEN3.TXT

Assembler

En Z-80 assembler kompletterad med en disassembler.

CAS 6:
ASM.BAS, ASM2.BAS, ASMCAS.REM, ASMCON

Lisp

Ett litet programpaket som gör det möjligt att prova på hur programspråket LISP fungerar.

CAS 6:
LISP.ABS, LISP.BAS, SAVE.LSP, LOAD.LSP, EDITF.LSP, FAC.LSP, GENEALOG.LSP, BAGLOD.LSP, BAGGINS.LSP

Hur får man tag på dem?

ABC-bladet, Samlingsnummer 1980-81 inkluderar ABC-kassetterna 1 och 2, är en sammanställning av ABC-bladets åtta första nummer. Detta omfattar c:a 150 sidor och kostar 100 Skr inkl frakt.

ABC-bladet 1982, inkl ABC-kassetterna 3,4, 5,6,7 och 8 har satts samman till en volym och finns tillgänglig med alla sex kassetterna för 125 Skr.

ABC-klubben rabatterar för nya medlemmar vid samtidigt köp av båda dessa samlingsnummer och tar då 210 Skr. Av praktiska skäl kan inte någon uppdelning göras i mindre delar än som angivits ovan.

Om Du är intresserad av någon av ABC-klubbens publikationer får Du dem enklast genom att sätta in rätt belopp på ABC-klubbens postgirokonto 62 93 00 - 5 och ange vad Du vill ha samt en tydlig och läslig leveransadress.

CP/M

CP/M till ABC80 ABC800/FACIT DTC.

Mikroprogram

För att en dators mikroprocessor skall kunna hantera någon form av instruktioner från en användare, måste ett mikroprogram finnas inbyggt i processorn, som styr dess funktioner. Mikroprogrammet är olika utformat för skilda typer av processorer t ex 8080, Z80, 6502, 6300, 8086, 8088, Z8000, 65000, för att här nämna några av de vanligaste typerna. Med hjälp av mikroprogrammet förses processorn således med en instruktionsrepertoar som är unik, i flertalet fall skild från andra leverantörers processorer.

Datorprogram

Mikroprogrammets instruktioner förutsätter att ett datorprogram är skrivet i maskinkod eller att datorn med hjälp av ett inladdat programspråk kan omvandla programmets instruktioner till maskinkod. Mikroprogrammet arbetar vanligtvis enligt det binära talsystemet, således på bitnivå med olika kombinationer av ettor och nollor, som betyder ström eller inte ström. Mikroprocessorer idag kan beroende på två arbeta med 8, 16 eller 32 bitars ordlängd.

Eftersom datorns processor endast kan förstå binära tal och människan som användare inte är särskilt väl lämpad att kommunicera med processorn i form av ettor och nollor, behövs någon form av programspråk som möjliggör att vi kan skriva instruktioner i klartext som av programmet översätts till en form som kan hanteras av processorn.

Operativsystem

Om världen vore så enkel som skissats här ovan skulle vi med ett bra program kunna klara allt som vi vill göra med en dator. Vi behöver emellertid även ha tillgång till ett operativsystem i datorn (ett fast program) för att kunna använda ett flertal olika programspråk i maskinen men även för att kunna hantera själva hårdvaran (datorn) och den kringutrustning vi kopplar till maskinen. Vi skulle annars på ett tämligen komplicerat sätt i våra program ge instruktioner till datorn hur den t ex skall lagra program på disketten, göra utskrift på skrivare och bildskärm mm.

Operativsystemet kan således i förenklad form beskrivas som ett fast program som automatiskt sköter om hanteringen av den programvara som kan läsas in i datorn och den kringutrustning som är ansluten till maskinen, bildskärm, tangentbord, flexskiveenheter, skrivare mm. Det sköter om att ladda program och datafiler från diskett eller annat yttre minnesmedia, lista biblioteket på disketten (som visar vilka program och datafiler den innehåller), spara program och datafiler, radera bort program och filer mm.

Även operativsystemen är olika till olika fabrikat av datorer beroende på vilken typ av processor som sitter i maskinen. Detta medför att programvara inte på ett enkelt sätt kan flyttas mellan olika datorer. Systemprogramvaran, operativsystemet, är i allmänhet något mycket diffust för användaren.

Man bör därför när man väljer dator välja en maskin med ett väl utprovat operativsystem och vara försiktig med nyheter som dyker upp på marknaden. Varför?

Jo, operativsystem är mycket svåra att få felfria. En och annan sk "bug" har vi som användare av ABC-80 och ABC-800 väl hittat under åren, i vart fall de som varit med från allra första början med ABC-80. Trots allt är nog operativsystemet ett av de bästa som finns på marknaden. Ett gott tecken på ett bra operativsystem är i allmänhet att det är väl utprovat, vilket betyder att en dator är såld i ett stort antal exemplar och därmed har använts av många slutanvändare. De svenska ABC-datorerna får väl i viss mån sägas uppfylla detta kriterium. I viss mån?

Ja, det har sålts mer än 30 000 ABC-80 och kanske ABC-800/FACIT DTC i liknande storleksordning. Jag har inte tillgång till några aktuella siffror från leverantörssidan just för tillfället. Men ser vi på datorer som använder CP/M som operativsystem så står sig inte jämförelsen. Antalet sålda maskiner med detta operativsystem får räknas i hundratusental om inte rent av i miljoner exemplar. CP/M har blivit vad man kan kalla industristandard. Till detta operativsystem finns även mer än 10000 olika applikationsprogram att tillgå (de flesta dock utvecklade i USA).

Man kan uttrycka det så, att man i dag bör välja en dator som kan utnyttja CP/M eftersom all programvara av någon betydelse finns utvecklad under CP/M.

Jag använder avsiktligt detta något provocerande uttryckssätt därför att vissa kritiker av CP/M har angripit att operativsystemet är trögt att använda. Och visst är det trögt att tex använda kopieringskommandon och en del annat. Men en slutanvändare upplever i allmänhet inte själva operativsystemet då man i allmänhet kör färdiga applikationsprogram som startar, om man så vill, med automatstart.

Även om operativsystemet är något trögt för programutvecklare att använda är det enkelt. I många fall skall applikationsprogrammen användas av personal som inte alls är insatta i programmering eller som har små kunskaper på detta område, och då är enkelhet vid hanteringen att se som en fördel. Den största fördelen är den mycket kraftfulla programvaran som nu finns att tillgå.

ABC-80 och ABC-800/FACIT DTC kan förses med CP/M-kort som medför att all CP/M-programvara kan användas på datorn. MYAB och CAT har tex under några års tid liksom FACIT marknadsfört CP/M-kort till maskinerna. Det kostar ca 3 500:- att bygga om maskinerna vartill kommer kostnaden för ett 80-teckens kort för ABC-80 för 80-teckens bildskärm, totalt omkring 5 000:-.

De svenska ABC-datorerna är inte rena RAM-maskiner avsedda för CP/M men ombyggnaden medför att datorn blir en RAM-maskin som kan arbeta både med det vanliga

operativsystemet och med CP/M. All ABC-programvara kan således köras förutom den programvara som finns under CP/M. Men?

Ja, det finns ett litet frågetecken för den ombyggda ABC-datorn. Datorn blir inte lika snabb som en ren CP/M-maskin, vilket har betydelse för den professionella användaren. En företagare som skall använda de vanliga smör- och brödrutinerna för order - lager - fakturering - reskontra mm kanske skall välja en snabbare maskin om man hanterar lite större volymer av transaktioner. Maskinerna är för mer vanligt bruk tillräckligt snabba, och avsevärt mycket snabbare än som vanliga ABC-maskiner när man kör kompillerad Basic eller de nya superhögnyårspråken som tex Supercalc och dBASE.

Ett basicprogram skrivet i vanlig interpreterande basic som skall sortera 1000 siffervärden tar 4 min 20 sek att köra, vilket är något lite långsammare än ett vanligt ABC-80 program, men sedan programmet har kompilerats till maskinkod körs sorteringen på ca 3 sek, och där hänger inte en vanlig ABC-80 eller 800 med på långa vägar. Skillnaden är drastisk, drygt 4 min jämfört med ca 3 sek för samma program.

Det som blir något långsammare i den ombyggda ABC-maskinen är framför allt läsningar och skrivningar som skall göras till disketten.

Vad är CP/M?

CP/M betyder Control Program for Microcomputers och operativsystemet utvecklades av Gary Kildall i mitten på 70-talet. Kildall grundade ett företag, Digital Research Corp. som i USA växt till ett av de största programvaruhusen.

CP/M är ett operativsystem som förutsätter att programmet självt kan laddas in från diskett (kassetanvändare gör sig inte besvär). I datorn som är en ren ram-maskin finns endast en liten boot-rutin i prom som läser in operativsystemet från disketten, automatiskt när datorn slås på.

CP/M består av fyra delar, Console Command Processor (CCP), Basic Input/Output System (BIOS), Basic Disk Operating System (BDOS) och Transient Program Area (TPA).

BIOS är den del av CP/M som måste göras specifikt för varje dator och sköter hanteringen av maskin- och programvara. Genom att skriva BIOS-delen till en viss dator (gjort av leverantören) får man alla maskiner som innehåller en Z80-processor att se likadana ut sett från programmets synvinkel. En dator som skall kunna köra CP/M måste förses med Z80-processorn från Zilog, vilket kan göras genom att ett kort sätts in i datorn om denna processor inte finns från början.

Genom att endast biosdelen måste göras speciellt för en viss dator, möjliggör detta att man relativt lätt kan flytta program mellan olika maskiner. Praktiskt taget alla leverantörer måste idag förse sina maskiner

med CP/M för att överhuvudtaget kunna vara med och konkurrera på marknaden. Här kommer CP/M 86 och MSDOS att få motsvarande betydelse på 16-bitarsmaskinerna.

På datorer med 8-tums diskettstationer flyttas programmen i allmänhet endast genom att sätta in disketten i den nya datorn. Man använder då IBM-format, singel density, som är standard för de flesta maskiner. På 5 1/4-tums diskettstationer finns ingen motsvarande standard utan varje leverantör har i stort sett sin egen. Här måste programmen oftast föras över med kabel mellan maskinerna.

CCP är den del av CP/M som tillåter användaren att ladda program från diskett eller Winchester, lista biblioteket, spara program och datafiler, radera program och filer mm.

BDOS är den del av operativsystemet som används för lagring av program och datafiler på yttre minnesmedia på effektivast möjliga sätt.

TPA är den del av minnet som användaren kan använda att programmera i eller för inladdning av ett färdigt program.

Den stora fördelen med CP/M kan således förenklat uttryckas som att valfritt program eller programspråk kan läsas in från diskett och programvaran kan flyttas mellan olika datorer. Det sista är inte minst viktigt vid professionell användning av datorer, där de stora investeringarna inte ligger i själva den burk som användaren kör sina program på utan i programvaran och de register och datafiler eller databaser som användaren lagt upp. Att konvertera stora datamängder därför att burken blivit för långsam och måste bytas är alltför kostnadskrävande och ibland inte alls möjligt att göra.

Ta ett relativt litet praktiskt exempel, ett kundregister med namn och adressuppgifter på 10 000 kunder som måste knappas in på en ny maskin. Detta är troligen minst 3 veckors heltidsjobb för en van dataoperatör, som dessutom först måste ha konstruerat den nya databasen där uppgifterna skall

läggas in efter att ha lärt sig den nya programvaran. Den här typen av jobb brukar kosta som man säger "skjortan".

Att kunna köra de gamla rutinerna oförändrade på den nya maskinen till dess man fått igång de nya rutinerna och därefter låta datorn föra över uppgifterna från gamla register till nya är helt klart ett tilltalande alternativ.

Fördelen med att vara maskinoberoende måste emellertid vägas mot nackdelen av att den mest avancerade programvaran inte är svensk. Användaren måste lära sig de kommandon som finns i programmet och tala engelska till programmet. Problemen med engelska kommandon är inte alltför svåröverkomliga då de flesta leverantörer försett programvaran med svenska hjälptexter och svensk manual.

Gör några av ABC-klubbens användare anspråk på att kunna skriva Basic-program med kommandon och instruktioner på svenska? Knappast.

För att effektivt kunna utnyttja avancerade program behöver man också kunna en del programmering. Det torde inte heller vara något problem för klubbens medlemmar. Man bör emellertid för att effektivisera sig själv glömma all Basic-programmering för framtiden och övergå till de nya fjärdegenerationsspråken typ dBASE. Vad sägs om att bara behöva skriva SORT ON NAMN TO "registernamn" för att sortera ett register med 5 000 adresser efter fältet namn.

Motsvarande sorteringsrutin i Basic med möjlighet att välja vilket fält jag vill sortera efter blir nog ett par hundra programrader lång och kräver dessutom assemblerrutiner för själva sorteringen om den inte skall ta timmar eller dagar i anspråk. Basic-rutinen som klarar sorteringen på ca 20 minuter, som dBASE gör är väl värd att publiceras i tidningen. Endast ABC-basic är giltig. Inte program med CP/M-basic som är kompillerade till ren maskinkod, där kan problemet med sorteringstiden klaras med en SHELL-METZNER sort-rutin utan assemblersnuttar.

Odd Rolander

WordStar ^{CP/M}

som ordbehandlingsprogram

WordStar-programmet medför att datorn kan användas som ordbehandlingsutrustning. Redan när du första gången skriver i texten på tangentbordet kan du rätta nedslagsfel och andra mindre skrivfel på ett enkelt sätt.

Tangentbordet

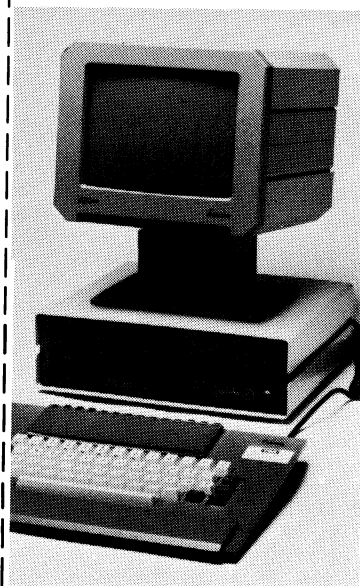
Tangentbordet på datorn ser ut ungefär som ett vanligt skrivmaskinstangentbord. Det är emellertid försett med ett antal tangenter som är specifika för datoranvändning, som vi skall återkomma till senare. Tangentbordet används dels till att mata in text med till ordbehandlingsprogrammet, men även för att kunna ge styrkommandon till programmet så att det utför de funktioner som finns inbyggda och möjliggör avancerad ordbehandling.

Texten du skriver hamnar i det interna minnet på datorn och alltså inte direkt på papper. Det interna minnet kallas ofta för primärminne för att skilja det från sekundärminnet - det yttre minnet. Inskrift av text och utskrift sker alltså vid två olika tillfällen.

När du trycker ned en tangent på tangentbordet skickas ett tecken i kodad form (ett siffravärde) via sladden från tangentbordet till datorns interna minne. Datorn skickar därefter tecknet vidare, man brukar kalla det för att tecknet "ekas", till videoskärmen så att tecknet blir synligt. Datorn har emellertid lagrat undan tecknet i en viss position i sitt interna minne och kan i nästa position i minnet lagra ett nytt tecken från tangentbordet.

NYHET!

SLIM LINE FLOPPY TILL ABC



Liten men naggande god.
Se bara:

DataDisc 52—2 × 160 kB

DataDisc 54—2 × 320 kB

DataDisc 56—2 × 640 kB

Passar till ABC 80, 800, 802, 806.

Välkommen till T-D-X Norr
eller till T-D-X City!

T-D-X

SmåDatorerAB

T-D-X NORR: SOLLENTUNA-
VAGEN 225, 111 23 SOLLENTUNA

T-D-X CITY: KUNGSGATAN 79,
KUNGSOLMEN STOCKHOLM

GEMENSAMT TEL NR 08-96 01 80

Videoskärmen

På videoskärmen kan du läsa den text du har slagit in från tangentbordet. Man kan således säga att bildskärmen speglar vad som finns lagrat i minnet på datorn. Detta är emellertid inte helt sant, eftersom bildskärmen endast kan visa 24 rader med 80 teckens längd på varje rad. Man kan därför säga att bildskärmen är ett "fönster" mot det interna minnet genom vilket man kan se en del av texten i taget. Det finns flera olika sätt i programmet att flytta fönstret så att du kan se den text du önskar. Du kan få texten att åka uppåt eller nedåt på skärmen. Detta brukar kallas för att texten "scrollar". Programmet kan även scrolla i sidled, vilket underlättar när man skriver breda tabeller för t ex liggande A4-format.

Bildskärmen används även av ordbehandlingsprogrammet för att visa viss information för operatören t ex felmeddelanden och hänvisningar, men även uppgifter om på vilken rad och i vilken position på raden den blinkande markören befinner sig. Markören visar i vilken position nästa tecken skall skrivas in.

Programmet

Programmet tar hand om instruktioner från tangentbordet som möjliggör att du på ett enkelt sätt kan rätta felaktigheter, göra strykningar och tillägg i texten och lätt kan göra större ändringar.

Arbetsgången blir att du efter inskrift av texten tar ut ett korrektur på papper och på korrekturet för in rättningar. När du sedan skall införa rättningarna i den ursprungliga texten behöver du inte skriva om något av ursprungstexten utan bara ändra felaktigheter, ta bort de delar som skall bort eller skjuta in eventuella tillägg. Programmet har instruktioner för automatisk rad och sidjustering och kan flytta ord mellan rader för att jämna ut stycken. Programmet kan vid utskrift även ge texten rak högermarginal, men detta förutsätter att man går igenom texten och "formaterar" rader och stycken. Du kan vid detta arbetsmoment även få avstavningshjälp av programmet. Vid utskriften skjuter programmet in blanktecken jämnt fördelade över raden för att sprida ut texten till rak högermarginal. Det finns även möjlighet i programmet till proportionell utskrift (mikrojustering) av skrivena tecken om du har en skrivare som kan klara detta.

Du kan med programmet kopiera hela texter eller textsidor, kopiera, flytta eller radera ord, hela meningar eller block av text. Du kan även ta bort eller lägga till textsidor och även hämta sidor eller textblock ur andra texter för att mixa ihop med den text du håller på med.

Du kan vidare med programmets hjälp sätta marginaler och tabulatorer, formatera om texten så att text kan läggas ut på olika bredd, automatiskt söka efter text och ersätta den funna texten med text av annat innehåll, mixa en text med ett adressregister (fordrar Mailmerge) för utskrift av brev, kuvert etiketter mm.

I samband med utskrift kan programmet hantera styrkoder som lagts in i texten för understrykningar, utskrift av rubriker med halvfet stil etc.

Programmet kan även fås att automatiskt sätta in rubriker och fotnoter på sidor, numrera sidorna i ett dokument samt skriva ut texten i spalter på sidan.

För att kunna göra utskrifter uppställda på spalter på sidan är det praktiskt att med programmet kunna se spalterna uppställda bredvid varann även på bildskärmen.

Detta låter sig göras med programmet på samma sätt som du även kan påverka kolumner i tabelluppställningar och flytta om kolumnerna efter behov på bildskärmen. Emellertid finns det en del restriktioner som måste iaktas när man vill skriva text i flera spalter. Texten kan inte skrivas ut med rak högermarginal eftersom programmet då justerar hela radbredden på bildskärmen och flyttar om texten. Detta leder till att spalterna inte längre blir raka. Texten måste även skrivas in med word-wrap funktionen avstängd eller läsas in från en färdig fil på disketten när funktionen har stängts av. Datorn betar sig annars så att den flyttar för långa ord till ny rad och utjämnar högermarginalen.

Texten måste först skrivas in och därefter läggas om till lämplig spaltbredd. När hela texten lagts om till lämplig bredd kan kolumnfunktionen kopplas till och spalterna flyttas upp vid sidan av varandra.

Här bör även nämnas att utskrift på papper är möjligt att göra på spalt med rak högermarginal, men att man då måste skriva en spalt i taget, backa tillbaka skrivaren och därefter skriva nästa spalt o s v.

Effektivitet

Effektiviteten i ett ordbehandlingsprogram jämfört med konventionell maskinskrivning ökar i proportion till hur många gånger dokumentet skulle ha behövt skrivas om, om du använt vanlig skrivmaskin. Det är inte bara inskrivningen av text som underlättas utan även korrekturläsningen, eftersom man kan lita på att inga nya skrivfel tillförs för varje ny version. Du behöver bara kontrollera de senast införda ändringarna. Du får även då den slutliga utskriften görs ett dokument helt utan synliga rättningar.

Om man till WordStar kopplar t ex programmet Ordet+ görs automatiskt även rättstavningskontroll av texterna.

Följande önskelista kan uppställas på ett effektivt ordbehandlingssystem.

Systemet bör kunna skriva de tecken som är vanligt förekommande inom den branch användaren hör till utan att man utnyttjar dyrbara specialskrivare eller stoppar utskriften för att byta typhjul.

Formelskrivning bör kunna ske på ett enkelt sätt om detta behov finns. Man skall kunna skriva både upphöjda och nedsänkta siffror, bråkstreck mm.

Tabellskrivning är ett område som ofta är förbiset och som förorsakar avsevärda problem om inte maskinen är lämplig.

Vid ifyllande av blanketter krävs att man lätt kan markera de platser på bildskärmen där olika uppgifter skall skrivas in för att utskriften skall bli korrekt. Helst bör blankettens utseende kunna visas på bildskärmen i form av horisontella och vertikala linjer. Detta är ej möjligt med WordStar-programmet.

Systemet bör kunna hantera olika pappersorter och format utan tidsödande ändringar av skrivaren. Detta är dock en svaghet hos så gott som alla system idag och medför att det är svårt för två eller flera sekretariat att dela på en skrivare som är placerad

på något annat ställe än nära intill skrivplatserna.

Programvara bör finnas att tillgå för rättstavningskontroll av texter där ordbanken som programmet arbetar emot enkelt kan uppdateras med nya ord.

Möjlighet till datakommunikation är i de flesta fall ett krav när det gäller användning av ordbehandlingssystem. Text kan då sändas mellan olika företag, avdelningar eller förvaltningar för redigering och bearbetning i samband med författande, utrednings- och kommittéarbete mm. Via de datakommunikationsnät som redan finns eller är under uppbyggnad kan man lätt sända meddelanden, brev, protokoll, tabeller mm både inom och utom landet med en snabbhet som inte är möjlig när man måste utnyttja posten. Man har härvid även möjligheter att utnyttja andra datorer med deras databaser (lagrade fakta) och beräknings- och sorteringsmöjligheter.

För att systemet skall kunna användas till datakommunikation krävs i allmänhet att datorn kan programmeras t ex i BASIC eller i vart fall att den kan köra andra färdiga program som möjliggör kommunikation via telefonlinje. Även kommunikation via telex kan i vissa fall vara möjligt.

Slutsatser

Ett generellt system medger större flexibilitet eftersom det kan användas både som dator och som ordbehandlare. Det kan ofta vara av värde att ha samma fabrikat av dator för beräkningar som för ordbehandling. Förutom fördelen att man har två identiska system - om något går sönder - kan man använda systemen till det som för tillfället är mest angeläget. Det finns i allmänhet till datorer även fler olika programvaror att välja bland så att användaren kan köra det program han eller hon trivs bäst med (förutsatt att data kan överföras mellan de olika systemen).

Odd Rolander

CP/M

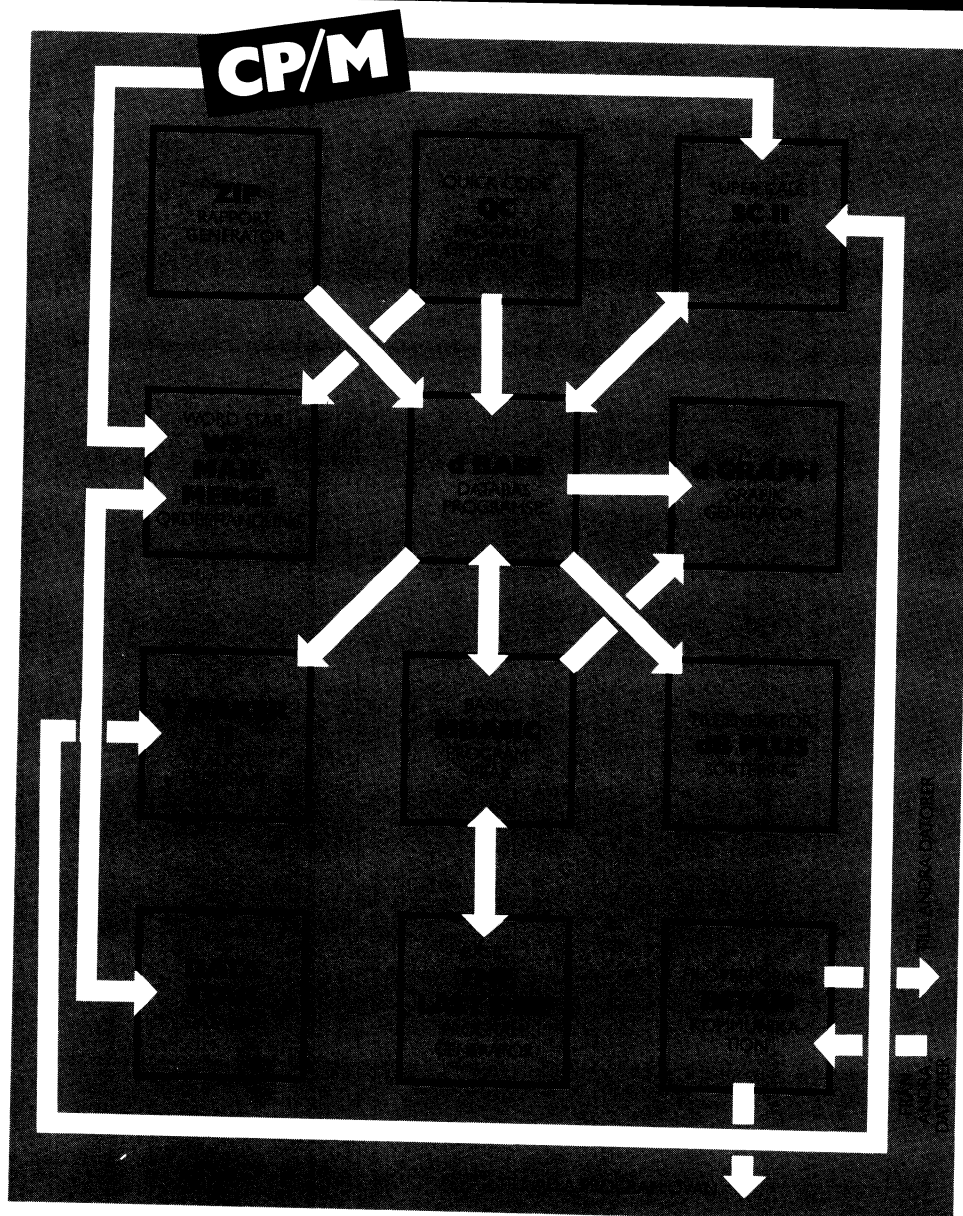
Överföring

av CP/M-filer

Texten om Wordstar som ordbehandlingsprogram har skrivits med WordStar alltså under CP/M. Eftersom redaktionen använder en ABC-80 med ett alldeles speciellt utskriftsprogram för att producera texter skrivet i ABC-80 Basic har filen överförts från den TRS80 modell II där filen skrevs med programmet BSTAM till min ABC-80 som har CP/M. Därefter har CP/M-filen konverterats till vanlig Basic-fil med ett litet konverteringsprogram jag själv knäpat ihop innan styrkoder för utskrift med redaktionen program använts. Rättstavningskontroll har inte körts på texten vilket kanske borde ha gjorts. Vi får se!

I CP/M-programvaran på ABC-80 finns en färdig rutin att gå andra vägen d v s att överföra en Basicfil till en CP/M-fil. Detta innebär för redaktionen del att man nu kan ha möjligheter att överföra tidnings-sändare och artiklar till CP/M, köra WordStar med rättstavningskontroll av texterna, konvertera texterna tillbaka till Basic och köra utskrift. Arbetsbesparingen att slippa nästan all korrekturläsning är väl värd tiden som går åt, ett par minuter, för konverteringen fram och tillbaka.

Odd Rolander



Bilden

Bilden är avsedd att ge en illustration av hur man kan arbeta med CP/M-programvara. De programvaror som upptagits är endast några få av de vanligaste programmen (det finns mer än 10 000 programvaror att välja på).

CP/M som arbetsmiljö gör att man kan arbeta samman med en mängd programvaror. Filer kan utvecklas med ett program, läsas in i ett nytt program, där man kan fortsätta bearbetningen, för att därefter kanske lämpa över filen till ett tredje program o s v. Vanligtvis slutar kanske jobbet i WordStar där jag vill ha en avancerad utskrift på skrivaren. Själv är jag helt fångad av WordStar eftersom få ordbehandlingsprogram ens kommer i närheten av den kraftfullhet som WS besitter.

Detta beror naturligtvis på att WS kan hantera nästan alla filer som produceras i andra programspråk, andra programvaror mm eller oftast kan konverteras till ett filformat som är lämpligt för WS. En rad i WS får max vara 32 000 tecken lång. Programmet kan hantera hur många A 4-sidor som helst, bara de rymms på disketten. Filen rullas nämligen fram och tillbaka mot disketten och endast drygt en A 4-sida finns i datorns minne åt gången.

Näval åter till bilden. Det centrala arbetsredskapet är i de flesta fall ett databaspro-

gram, här representerat av dBASE, som förutom att det är en kraftfull databashanterare även är ett eget programmeringsspråk. Det finns konsulter som jobbar avancerat i dBASE som anser att det är det enda program man behöver. Med dBASE kan man göra **allt**.

Jag tror för min del att en liten överdrift kan ligga i detta. Ordbehandling går säkert bättre än i ett databassystem om jag använder ett ordbehandlingsprogram. I många fall föredrar jag att använda ett kalkylprogram t ex SuperCalc II för beräkningar, men visst går det att räkna även i dBASE. Nackdelen är att jag då liksom då jag skall göra avancerade beräkningar i Basic behöver mer än rudimentära kunskaper i programmering.

Till dBASE finns en ZIP-rutin, som är en rapportgenerator. Det innebär att jag kan åstadkomma layouter till formulär och rapporter eller en bildskärmslayout, som jag trivs att jobba med på ett enkelt sätt. Jag ritat helt enkelt med markören på bildskärmen hur jag vill att det skall se ut, och ZIP översätter vad jag ritat till programkod som dBASE behöver för att kunna skriva ut min rapport eller presentera informationen på skärmen exakt så som bara jag önskar.

QuickCode är en programgenerator som skriver dBASE-program åt mig. Så länge

jag endast jobbar med en databas behöver jag alltså inte kunna programmering utan det kan QuickCode få lära mig genom att det gör hela programmet. Annorlunda blir det dock när jag känner mig mer avancerad och vill börja använda flera databaser (register) för att plocka uppgifter ur och mixa ihop till någonting nytt. Att kunna programmering i det läget underlättar. Du vill kanske göra ett helintegrerat program för orderlager-fakturerings-kundreskontra eller något i den vägen för att nu inte ta till för lite.

Bli inte skräm, en orderregistreringsrutin blir bara ca 40 programrader lång. Inmatningen till ett kundregister kan QuickCode skradda till och likaså uppläggnings och inmatningen till lager och uppläggnings av kundreskontraregistreret. Orderregistreringsrutinen skall sedan hämta uppgifter från kunder och artiklar och kunna lagra i diverse olika filer. Problemet för ett enkelt program av denna typ består sedan i att kunna skradda till filer som lyfter över order till en faktureringsrutin och därefter när fakturan är utskrivnen lyfter över uppgifterna till kundreskontra för bevakning. Du får naturligtvis om du skall ha ett komplett, mycket avancerat system, lägga ned en hel del tid på övriga kringrutiner för att kunna ändra i order, lista alla inläggande order, klara restorderhantering o s v.

Programutvecklingen torde dock gå på ungefär 1/10-del av den tid det tar att skriva motsvarande program i Basic, Pascal, Fortran eller Cobol. Föredrar man dessa programspråk så finns de ju också tillgängliga under CP/M.

Grafik kan köras ut på skrivare med programmet dGRAPH som kan läsa dBASE-filer, basicfiler eller du kan i programmet självt lägga in de register som skall producera diagram på ett enkelt sätt. När det gäller dGRAPH får man se upp med att man har en skrivare som programmet kan hantera. Själv gick jag på en blåsning när jag testade programmet. Min Diablo 1650 med Wordprocessorkort kunde bara rita ovala cirkeldiagram, förmodligen beroende på att VMI och HMI på diablön måste ställas in rätt för att det skall fungera. Skrivhuvudet flyttas 1/48 " i höjddled men kan även flyttas 1/60 " eller 1/120 " i sidled. Förflyttningen i höjddled går inte att variera.

Hur bilden i övrigt är uppbyggd torde vara lätt att förstå. dBASE kan läsa in filer skrivna i Basic. Till Microsoft Basic finns programgeneratören The Last One som gör kompletta Basicprogram, även databaser i Basic för den delen, o s v.

Arbetsområdena är näst intill outtömliga när man har programvarorna till sitt förfogande.

Priserna för programvaror ligger på nivån ca 3000:-. dBASE är dyrare och kostar ca 6 500:-. Det finns dock en Runtime-modul för ca 800:- med vilken man kan köra färdiga dBASE-program, dock inte programmera själv. Det finns i allmänhet även demoversioner av programmen till en billig penning som endast kan hantera en begränsad mängd data, men som kanske kan vara något att börja med för den intresserade.

Redaktionen planerar några häftiga datakör med ovanstående programvara i klubbens lokaler vid Alvik i samråd med Stockholmsavdelningen efter trettonhelgen. Uppgifter om tider och aktiviteter kommer vi att lägga in på klubbens telefonsvarare någon gång efter jul.

Odd Rolander

ELFA mikrodatorinfo

ELFAs sortiment av mikrodatorprodukter har utökats! Se sidorna D1–D39 i ELFA-katalogen nr 32. Här är några av nyheterna, som alla finns i lager:

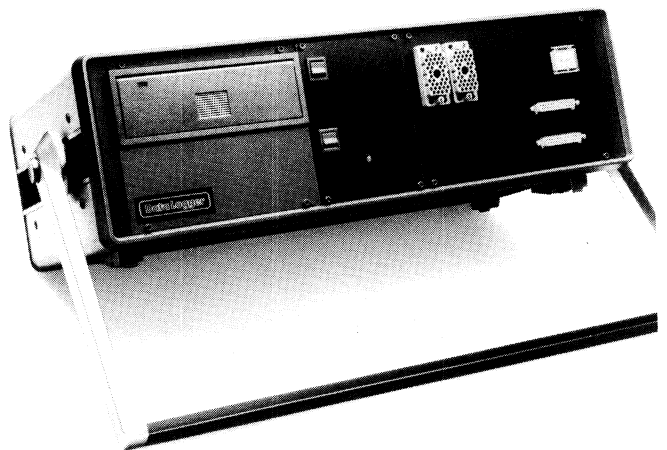
Epson HX-20

Smådatorn du kan ta med dig i portföljen. Den bärbara datorn. Vikt 1,6 kilo. Batteri- eller nät-driven. Minneskapacitet på 16k – expanderbart till 32k. Den kan kopplas direkt eller via modem till en ABC800 som i sin tur är kopplad till en stordator. HX-20 kan också kopplas direkt till en stordator via ett modem.



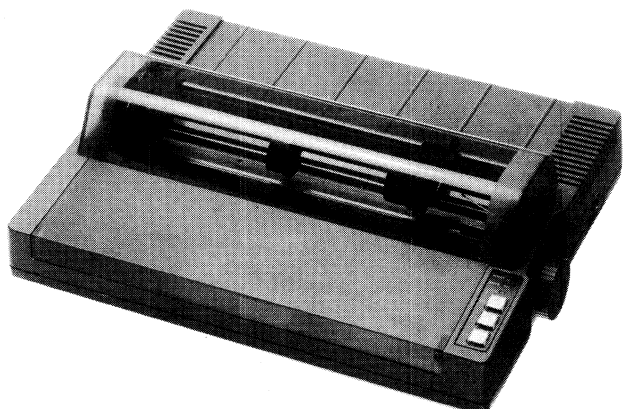
Datalogger

Ett datainsamlingssystem, uppbyggt kring enkortsdatorn 1003/1004 ur Data-Board-serien. Dataloggern kan hämta mätvärden från olika mätplatser vid bestämda tidpunkter, värdena lagras på en flexskiva, som senare kan läsas i en ABC800 för analys. Med Dataloggern medföljer programvara för datainhämtning och analys.



Mini 1200

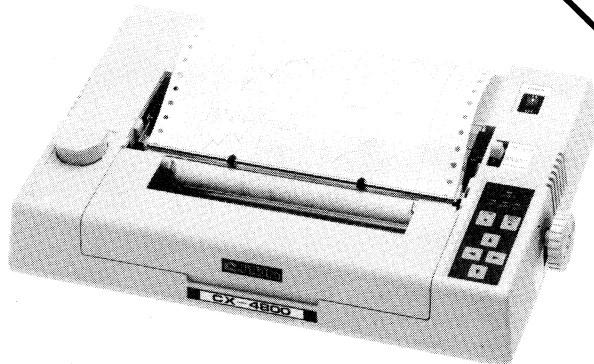
Kompakt matris-skrivare med stor tecken-repertoar. 120 tecken/sek, 80 tecken/rad, skriver i båda riktningarna. Friktion- och piggmatning. RS 232C och Centronic som standard. 96 ASCII-tecken. Epson-kompatibel.



CX4800

Plotter-printer

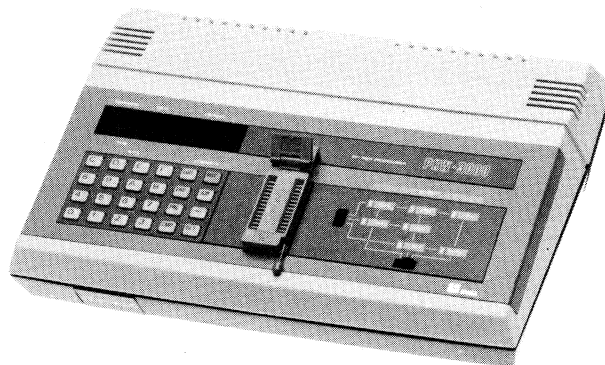
Liten A4 trumplotter med plotthastighet 12 cm/s, 4 pennor i revolvermagasin. Intelligens för generering av vektorer, bågar, cirklar, linjer och tecken. Pigg- och friktionsmatning. RS 232C och parallellinterface. Svensk dokumentation.



PKW 3000

Prom-programmerare

Programmerar Eprom/EEprom från 16 k till 64 kbyte via master, knappsats eller RS232C-snitt, 110–4800 baud. Editeringsfunktion, läschek vid 3 olika spänningar. Sex standardiserade överföringsformat. Nätanslutes. Programmerbara prom: 2716, 2732, 2732A, 2532, 48016, 2564, 2764 eller deras ekvivalenter.



CAD-ABC

CAD-paketet som betalar sig på 5–10 kretskort!

CAD-ABC är en komplett arbetsplats för framställning av ritningar, schemor och mönsterkortsunderlag.

Systemet är baserat på ABC80 i standardutförande kompletterat med ytterligare 16 kbyte RAM och en dubbel flexskiveenhet.

Vi sänder gärna utförligare information eller kommer överens med er om en demonstration hos oss i Solna.

PRESSTOPP



Läs igenom D-sektionen i nya ELFA-katalogen. Där finner du säkert de flesta mikrodatorprodukter du har användning för. Har du en systemtillämpning som kräver specialprogram så hjälper vi på avd. Mikrodatorer till med det också.

ELFA
RADIO & TELEVISION AB
171 17 SOLNA
INDUSTRIVÄGEN 23 • 08-730 07 00

CP/M SuperCalc II

som kalkylprogram

SuperCalc är ett av de många kalkylprogram som finns att tillgå under CP/M på marknaden. Andra exempel är Multiplan, Knowledge Man och Visicalc (det senare kan endast köras under andra operativsystem på t ex Apple och TRS-80). För de nya 16-bitarsdatorerna finns t ex Lotus 1, 2, 3.

Programvarorna är alla exempel på de nya 4:e generationsspråken eller superhögnyväspråken, vilket man nu vill kalla dem. Detta innebär användarvänliga programmeringsspråk, så som jag förenklat vill uttrycka det. Användaren märker inte att han i själva verket programmerar datorn när han eller hon använder kalkylprogrammet.

Bildskärmen kan ses som ett stort rutat arbetsblad, indelat i rader och kolumner. Man brukar även kalla det för ett elektroniskt skissblock. SuperCalc är uppbyggt med 63 kolumner i sidled och 254 rader i höjdlid, där du i varje ruta som bildas kan skriva in text, siffror eller matematiska formler, som refererar till innehåll i andra rutor i matrisen. En kalkyl byggs upp så, att programmet hämtar innehållet ur andra rutor i matrisen utför någon ny beräkning och skriver svaret i den ruta där formeln är inmatad. Data blir på detta sätt relaterade till varandra och en ändring i en ruta påverkar alla övriga rutor i matrisen, som på något sätt är relaterade (kopplade till) den ruta där ändringen gjordes.

Man kan lätt ändra, stryka, spara eller skriva ut kalkylen. Rader eller kolumner kan läggas till eller tas bort i efterhand och i varje situation, när man arbetar med ett kommando, kan man få fram hjälptexter på svenska som visar hur man skall gå vidare med kommandot.

Jag skall inte gå in för att förklara kraftfullheten i dessa typer av programvaror i detalj, det skulle kräva ett eget nummer av ABC-bladet, utan nöja mig med att visa ett exempel på en liten kalkyltillämpning och hur resultatet av ett antal kalkyler kan se ut presenterade av diagramprogrammet dGRAPH.

Kalkylen avser installation av värmepumpar i en bostadsrättsförening innehållande 5 hus och 120 lägenheter för att återvinna energi ur den friskluft, ca 20 grader varm, som evakueras ut via fläktsystemet.

VARMEPUMP

Utrustn	Faktor	kWh	Korr kWh	Oljepris	Energipris	Kr/år	Driftred
VARMEPUMP	1	1,070,000.00	1,070,000.00	1,760.00	.22	235,400.00	
KULVERT	1	70,000.00	70,000.00	1,760.00	.22	15,900.00	
EL VP	1	370,000.00	370,000.00		.29	107,300.00	
EL EPATR	1	5,000.00	5,000.00		.29	1,450.00	
TARIFF		9,450.00				9,450.00	132,500.00

Ar	Faktor	Årlig red	Service	Korr red	Amort	Uppr fakt	Gar ränta	Kapitalkostn	Netto	Accumulerat
Ar 1	1	132,600.00	.00	132,600.00	35,815.00	.03	40,050.00	73,865.00	58,735.00	58,735.00
Ar 2	1.1	145,860.00	20,000.00	125,860.00	36,210.95	.0325	43,387.50	81,596.45	44,261.55	102,996.55
Ar 3	1.1	160,446.00	22,000.00	138,446.00	43,178.37	.035	46,725.00	89,903.37	48,542.63	151,539.18
Ar 4	1.1	176,490.60	24,200.00	152,290.60	48,791.64	.0375	50,062.50	98,854.06	53,436.54	204,975.71
Ar 5	1.1	194,139.66	26,520.00	167,619.66	55,134.47	.04	53,400.00	108,534.47	58,985.13	263,960.84
Ar 6	1.1	213,353.03	29,282.00	184,071.03	62,301.95	.0425	56,737.50	119,039.45	65,232.18	329,193.05
Ar 7	1.1	234,908.99	32,210.20	202,698.79	70,401.20	.045	60,075.00	130,476.20	72,222.59	401,415.64
Ar 8	1.1	258,399.89	35,431.22	222,968.67	79,553.35	.0475	63,412.50	142,965.85	80,002.81	481,418.44
Ar 9	1.1	284,235.80	38,974.34	245,261.46	89,895.29	.05	66,750.00	156,645.29	86,320.24	570,738.68
Ar 10	1.1	312,663.66	42,871.78	269,791.88	101,561.68	.0525	70,087.50	171,669.16	96,122.91	668,861.65
Ar 11	1.1	343,930.25	47,158.95	296,771.30	114,787.30	.055	73,425.00	188,212.30	108,559.00	776,720.65
Ar 12	1.1	378,323.28	51,874.85	326,448.43	129,709.64	.0575	76,762.50	206,472.14	119,976.28	896,696.93
Ar 13	1.1	416,155.60	57,062.35	359,093.25	146,571.80	.06	80,100.00	226,671.80	132,421.37	1,029,116.30
Ar 14	1.1	457,771.16	62,708.57	395,062.60	165,626.25	.0625	83,437.50	249,003.75	145,938.63	1,175,057.15
Ar 15	1.1	503,548.28	69,045.42	434,502.86	185,441.10	.065	86,775.00	272,216.10	162,886.76	1,367,343.90
Summa					1,335,000.00		951,187.50	2,286,187.50	1,367,343.90	

Konsultföretaget som projekterade anläggningen, hade räknat ut besparingen, den årliga driftsreduktionen, det första året. Jag upplevde att jag ville se hur kalkylen tog sig ut under en 15-årsperiod eftersom investeringen skulle genomföras med hjälp av statliga energilån, som skulle förräntas och amorteras under en 15-årsperiod. Genom att ta hänsyn till alla kända fakta och göra en uppräkningsfaktor (= årlig oljeprisökning) borde man få ett bättre underlag för att fatta beslut om investeringen.

I kalkylen intogs även en faktor för att kunna variera effektiviteten på anläggningen d v s 0 till 10 % avvikelse från förutsatt funktion på värmepumpar och elförbrukning.

Vid större avvikelse än 10 % träder leverantörens garanti in och anläggningen skall åtgärdas utan kostnad för beställaren men upp till 10 % felfunktion får man tåla i kalkylen. Eftersom staten aviserat att en snabbare upptrappning av den garanterade räntan kan komma att ske på energilån behövs även uppräkningsfaktorn för garanterad ränta kunna varieras.

Överst i kalkylen räknas den årliga driftsreduktionen det första året fram, där värmebesparing och energibehov redovisas i kWh. Oljepriset i kr/kubikmeter omvandlas till fjärrvärmesatser i kr/kWh varefter årlig besparing i kr redovisas.

I den nedre tabellen finns inflationsfaktor, nu 10 % årlig inflation förutsatt. Årlig reduktion räknas fram och en korrigerad reduktion sedan hänsyn tagits till kostnader för service av anläggningen. Amorteringen på lån (annuitetslån med olika annuiteter på bottenlån och energilån) framräknas jämte garanterad ränta, varefter man ser den totala kapitalkostnaden per år. Längst ut till höger räknas årligt netto fram samt ackumulerad besparing under 15-årsperioden.

Att nu variera de olika faktorerna leder till att datorn producerar en oacceptabel mängd datalistor där det inte blir lätt att få överblick över vad man egentligen håller på med.

Grafikprogrammet kan lättare redovisa kurvor över det ackumulerade nettot, när inflationen sätts till t ex 8 eller 10 % och när upptrappningsfaktorn för garanterad ränta t ex fördubblas. Tre diagram redovisas här med 0 %, 5 % respektive 10 % förutsatt felfunktion på anläggningen. Man ser i det sista diagrammet att 8 % inflation och fördubblad garanterad ränta blir klart förlustbringande.

Genom möjligheterna att på två sekunder räkna om kalkylen kan man lätt se vilka

parametrar kalkylen är känslig för. Störst betydelse får det oljepris jag har när jag startar kalkylen, årlig prisstegring är nödvändig för att en kalkyl överhuvudtaget skall gå runt, brytpunkten ligger vid ca 4 % årlig inflation o s v.

Hur gjorde man förr för att göra enkla investeringskalkyler ?

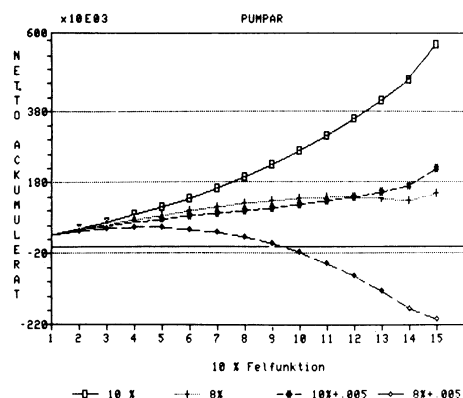
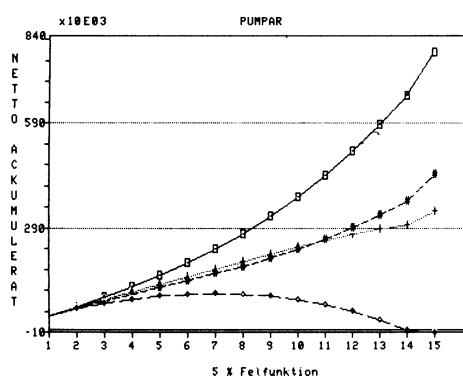
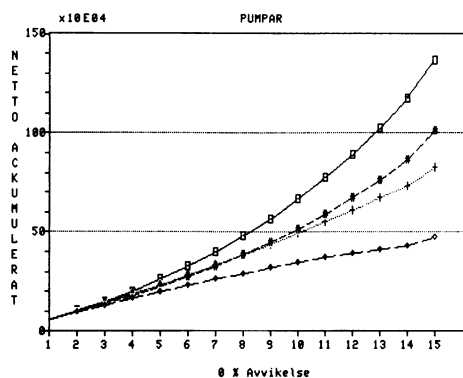
Ja, jag för min del avstod från att räkna på mer än några få alternativ med vanlig fickräknare. Utan fickräknare kan jag dessutom inte alls räkna.

Leder detta till någon förbättring ?

Man kan hävda att den som påstår något nytt, har bevisbördan för att det leder till någonting bättre. Jag har här avstått från bevisning genom att formulera till kalkylen inte redovisats, de hade annars lika enkelt kunnat fås ut på papper på skrivaren.

Frågan överlåter jag till dig, käre läsare att besvara.

Odd Rolander



CP/M på ABC-datorer

några synpunkter och erfarenheter

CP/M är ett av världens spridda operativsystem, och det finns nog inget system som finns på så många olika datorer. Jag kommer att enbart behandla CP/M 2.2, som finns på såväl ABC-80 som ABC-800. CP/M 3.0 eller CP+ som endast finns på ABC-806 och nya DTC-2 kräver extraminne m.m.

Fördelar med CP/M:

det finns
det är mycket spritt
det finns mycket programvara och språk.
Genom CPMUG finns en hel del program till mycket lågt pris, ungefär som via ABC-klubben.

det finns nästan alla programspråk under CP/M för den som vill förkovra sig.

Nackdelar:

på ABC-80 och ABC-800 krävs extra hårdvara med extra minne.
diskhanteringen är långsam jämfört med ABC-DOS.

standardeditorn, ED är litet knölig innan man vant sig. Den är TTY-orienterad och TECO-liknande och trög om man är bortskämd med skärm-orienterade editorer. Det finns givetvis att köpa avancerade sådana editorer, t ex VEDIT.

CP/M är utrymmeskrävande på disk. Att köra på 830-disk innebär ett evigt skivbyte, och att köra på en FD2 torde vara efter värre.

Vad finns det då för möjligheter att köra CP/M på ABC-datorer? För ABC-80 finns det MyAB CP/M. Det består av ett litet kort, som man sätter in mellan CPU-kretsen och det stora kortet. Det är mycket enkelt att montera, man lirkar upp CPU-kretsen och lossar en skruv, som fäster moderkortet i lådan. Sedan sätter man ett distansrör mellan CP/M-kortet och moderkortet och trycker fast CP/M-kortet i CPU-hållaren. Till sist sätter man in CPU:n i CP/M-kortet. Sedan är det bara att montera ihop ABC-80, sätta CP/M-skivan i disken och köra. Det finns mycket bra monteringsanvisningar till alla MyAB:s kort. Om man har TKN-80 inmonterad, utnyttjar CP/M detta automatiskt.

På ABC-800 finns dels MyAB:s CP/M dels numera även Luxors CP/M. Båda kräver ett minneskort om 64 kB, som monteras mellan CPU-kort och video-kort. För att köra MyAB:s CP/M krävs dessutom att man har MyAB UNI-800-kortet som 64kB. På detta finns ett BOOTPROM. Använder man UNI-800/G finns det en monteringsanvisning för att kunna köra HR-kortet i maskinen. Luxors CP/M nyttjar en klurig BOOT via ett 'fusk-bibliotek' i sektor 16, som i sin tur mappar om minnet och laddar in CP/M. Bootprogrammet undersöker även om det ABC-800 eller 802/806 och lägger ut olika kod i Bios, beroende på memory-mapping hårdvara. Luxor CP/M har stora fördelar, det finns dock även fördelar hos MyAB-CP/M.

1. Programmet ABCDISK under CP/M gör det möjligt att flytta ASCII-filer från ABC-DOS till CP/M. Programmet nyttjar CP/M BIOS och fungerar inte under Luxor CP/M.

2. Programmet CPMDISK flyttar filer från CP/M till ABC-DOS under ABC-DOS.

3. Källkoden till BIOS och BDOS finns med, så man kan se hur det är gjort och modifiera BIOS vid behov.

4. BASIC-minnet (29k på ABC-800) används som RAM-disk och kallas C:.

Nackdelen med detta är att det är litet struligt att generera system för andra konfigurationer. I detta fall är SYSGEN under Luxor CP/M ett synnerligen lätthanterligt system. När man startar SYSGEN får man upp aktuell konfiguration, och man kan sedan enkelt uppdatera vilken disk som är A:, B:, C:, D: etc. Har man tillgång till en maskin med flera olika disk-stationer, är det synnerligen enkelt att göra olika system och att flytta filer mellan olika format. Samma skiva kan dessutom köras i ABC-800 med CP/M-kort, i ABC-802 och i ABC-806.

Kör man på en enda maskin, går det på ett ut, behöver man veva runt på olika maskiner med flera diskar är Luxor CP/M ett måste.

På ABC-80 finns bara MyAB CP/M, och likaså om man vill köra med 5" SSSD eller single side, double track (320kB). Vill man använda ABC-802 eller ABC-806 finns det endast Luxor CP/M eller CP/M 3.0 från MyAB. För ABC-802 kommer från MyAB snart ett minneskort med upp till 512 kB för inbyggnad. Wow!

CPMUG är uttytt CP/M Users Group. De har ställt iordning ett bibliotek med f.n. 77 st 8" SSSD skivor med program av varierande kvalitet. De kostar 50 kr. per skiva från MyAB. Vill man ha 830-skivor, får man varje CPMUG-skiva på två skivor för 100 kr. Eftersom programmen är helt fria får man ge bort dem hur som helst. Priset är ju i stort sett mediakostnaden.

Ett antal av de bättre programmen kommer från Ward Christensen, som tycks ha vikt sitt liv åt att skriva program till CPMUG. Här finns t ex MODEM, som är ett generellt filtransfereringsprogram för CP/M-datorer och REZ, som är en mycket avancerad disassembler. Från CPMUG finns diverse assemblers, en enkel Pascal, Ratfor, Adventure, ett enkelt Algol-system och mycket annat. De flesta program är i 8080 assembler, och en del måste yxas till för att kunna köras. Det är till stor del programvara för och av dator-amatörer.

Om man vill ha tag i en Pascal billigt, finns JRT Pascal för 36 dollar. Den har en del begränsningar och vid kompilersfel stoppar den vid första påträffade fel. Den fungerar dock. Den säljs under villkoret att man får ge bort kopior till sina vänner, så om man slår i hop sig några stycken, behöver man inte bli ruinerad på Pascal. Det finns ett antal Pascal-system under CP/M. Det bästa och dyraste är nog Pascal/MT+ för ca 5000:-.

Det finns en hel del C-kompilatorer under CP/M, även om 8080-arkitekturen inte är den bästa för C. En prisvärd sådan är C/80 från Software Toolworks. Den säljs av Elektrokonst, N-3000 Drammen, Norge för 400 NKR. Den är baserad på Ron Cains Small C och innehåller det mesta vad beträffar int och char. I Byte fanns en artikel där man jämförde olika C-kompilatorer under CP/M. BDS-C verkar vara en annan prisvärd kompilator till överkomligt pris, som är utrymmessparande på disken. En del C-kompilatorer, tex. C/80 skapar en mellanfil på disken med Assembler-kod, och denna är ytterligt platskrävande.

För den som vill skriva assembler under CP/M och vill använda Z80-instruktioner finns det dels Z80ASM från CPMUG, dels UVMAC från Software Toolworks. Den senare är en Macro-assembler. Även Microsoft M-80 lär klara Z-80 kod.

Vidare finns givetvis Basic, främst

Microsoft Basic om man har 4000:- över. Skall man få snabba program krävs dessutom BASCOM, Basic kompilatorn för ytterligare drygt 5000:-. Det finns ADA, LISP, FORTRAN, COBOL, FORTH och Modula-2 och mer kommer. Priserna varierar kraftigt, från 30 dollar och uppåt.

Man kan också använda ett CP/M-system som utvecklingssystem för andra mikroprocessorer då det finns korsassembleratorer för de flesta CPU:er från tex. SORCIM. De säljs av Applitron i Göteborg. Det finns t o m C-korskompilatorer under CP/M, men de är ganska ineffektiva.

De flesta program under CP/M kan levereras på 8" SSSD. Många går att köra direkt, under det att en del måste yxas till. För att få över dem till annat format krävs en ABC-800 med flera olika disk-stationer. Kan man få tillgång till en sådan maskin, finns det mycket att välja på. Läser man BYTE finns det sida upp och sida ned med annonser. Man får dock alltid vara beredd på att vänta och på att inget händer.

I Sverige kan man köpa CP/M-program dels från MyAB, dels från Applitron i Göteborg. MyAB kan leverera på alla ABC-format och de har även CPMUG. Applitron har ett stort urval av programvara, men levererar främst på 8" SSSD.

Som avslutning kan jag säga att CP/M är ett bra komplement till ABC-datorerna om man vill gå utanför BASIC och det som finns för ABC-DOS. Det kostar dock en hel del, så man måste vara motiverad. Dyrast blir det på ABC-80 och ABC-800. På ABC-802 och ABC-806 kommer man billigare undan genom Luxor CP/M.

Torbjörn Alm <116>

DIV:

Om TV-editorn

Efter att åtskilliga gånger försökt hämta in en fil från skivan med kommandot ;S med resultat att filen fått skrivas in på nytt eller reparerats med t.ex. programmet LÄSFIL2, har jag nu infört följande ändringar.

```
3135 IF LEN(M$)=0% GOSUB 3820
3820 ;CHR$(7)CUR(23,0)"Varning! Vill du
spara en tom fil?"
3830 FOR I=1 TO 1000 : NEXT I : RETURN
```

Rad 3135 kollar om minnet är tomt och hoppar till subrutinen på rader 3820 och 3830. Klocksignal och text ger varning för eventuell överlagring på redan skriven fil, men tillåter efter ca 1 sek väntan att ändå gå vidare, men då förhoppningsvis mera medvetet.

Ändringarna har använts ett tag och fungerar bra. De lades in i TVMAIN samtidigt med ändringarna enligt ABC-bladet nr 2, därför radnr enligt ovan.

Stig Berlin < 251 >

Denna ändring kan också göras i TVMAIN.800 (versionen för ABC800-datorer, men då måste rad nr 3135 enligt ovan ges radnummret 3130.

Bo Kullmar

DIV:

RÄTTELSE

Priserna som uppgavs i förra nummret av ABC-Bladet var tyvärr lite felaktiga. De riktiga priserna är:

Selic 300/300 + autosvar

1.895 kronor exl moms

Selic 300/300 + 75/1200 + autosvar

2.495 kronor exl moms

Selic 300/300

995 kronor exl moms

För kombinationsmodemet (300/300 + 75/1200 + autosvar) bör switch nr 7 normalt sättas on. Sker inte detta så kopplar modemet ner förbindelsen bl a då datorn går över i lokal mode och vid filöverföring med ABCFIL. Sätts switch 7 on så hålls alltid stift 108 on. Jmf ABC-Bladet nr 2, 1983, där samma problem ventileras, men den lösning som där rekommenderas gäller då televerkets för-
enklade modem.

Slutgiltiga uppgifter om Selics enkla 300 bps modem finns nu tillgängliga. Det blev som tur inte ett enkelt A-kanalsmodem som jag skrev i förra nummret, utan ett A- och B-kanals modem. Modemet finns i två utförande:

Ett som passar terminaler/datorer som har ett V24 interface. Kraften tas om möjligt från terminalen/datorn, annars måste en batterieliminatör (för cirka 100 kronor) anskaffas.

Ett som är avsett för VIC och som så skall pluggas in i dennes "userport".

Bo Kullmar

Adresser och telefonnummer:

I förra ABC-Bladet uppgavs tyvärr Mats Knuts gamla adress och telefonnummer. Aktuell adress är Oshögavägen 186, 230 30 OXIE, vilket också står i matrikeln. Telefonnummer är däremot även fel i matrikeln. Det rätta nummret är 040-54 91 55.

Telefonnummret till Bo Kullmar finns angivet på matrikelns baksida. Det är dock fel där. Det rätta nummret är 08-751 18 29. Det är alltså rätt nummer som står inne i tidningen.

Bo Kullmar

Notis från klubbens monitor:

NYTT MONITOR-SYSTEM ÖPPNAT PÅ ÅLAND Systemet kallas RXMON och använder samma monitorprogram som ABC-Klubben. Systemet är öppet kvällar och helger att använda för alla som vill.

Tfn. från Sverige 009358-2821133

Tfn. Från Finland 928-21133

Modem 300/Bps halv duplex

Data hälsningar från Medlem 1139

Bug i FORTH79

Det har visat sig att ett fel finns i klubbens FORTH79. Felet finns både i den ursprungliga FORTH79 versionen för ABC80 och i de senare versionerna för ABC800, ABC802 och ABC806. Felet yttrar sig så att man under vissa omständigheter kan få filen som skrivs ut med SAVE-FORTH, att hamna på dirve 1 i stället för dirve 0.

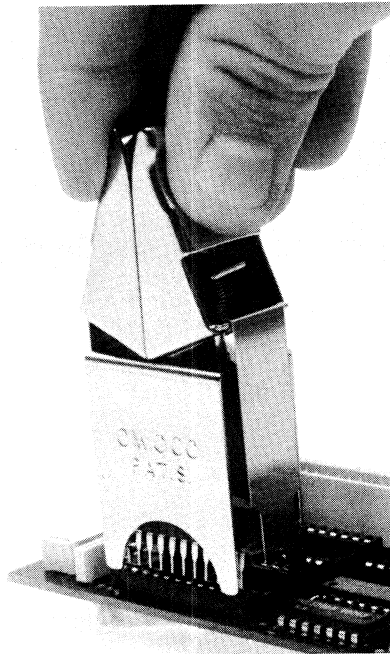
ABS-filen skrivs nämligen ej nödvändigtvis ut på dirve 0 utan på den dirve som man senast läste eller skrev på.

Om man följer anvisningar för 800-versionerna medför detta att allting fungerar om man följer anvisningarna till punkt och pricka. Däremot om man försöker vara smart och sätter i programskivan i dirve 1 och den tomma skivan i dirve 0 så kommer programmet att laddas från dirve 1 och därefter kommer ABS-filen att skrivas ut på dirve 1. Är skivan i dirve 1 då ej skrivskyddad, så har man ingen körbar programskiva längre!

De som har fått sina programskivor sönder-skrivna kan få de utbytta genom att sända in skivan.

Detta gäller alltså även ABC80 versionen.

Mats Knuts och Bo Kullmar



Gammalt problem nu löst, PROM-UT tar lätt ut IC-kapslar ur tröga hållare!!

Den som med mejsel eller annat vertyg försöker sig på konststycket att lossa större IC-kretsar ur hållare har ofta misslyckats och fått skador på dessa eller på kort och hållare.

OWOCO AB har nu tagit fram ett nytt praktiskt hjälpmedel för demontering av större IC-kretsar av PROM-typ ur hållare.

Anbringa PROM-UT över komponenten och med ett lätt handgrepp (ingen kraft-åtgång) lyftes den rakt upp ur hållaren. PROM-UT tar lite utrymme och kan med fördel användas på tätpackade kort.

PROM-UT är liksom IC-monteringsvertyg "DIP-IN" m fl av svensk design och tillverkning och exporteras till ett flertal länder.

Tillverkning och försäljning

OWOCO AB

Kvarnbergsvägen 25, 141 45 Huddinge
Telefon 08-774 02 90

Ett litet floppytips

Om man inte alltid har systemskivan i samma drive som den satt när systemet initierades kan man få konstiga felmeddelande eftersom systemet tror att BASICINI.SYS finns i samma drive på skivan som nu sitter där. BASICINI.SYS är filen som bl a innehåller felmeddelande på svenska för ABC800-datorer. För ABC80 heter motsvarande fil BASICERR.SYS.

Problemet är alltså att man byter skivor ofta och då hittar systemet inte BASICINI.SYS. En enkel lösning är att alltid ha BASICINI.SYS på alla skivor och på samma plats. Man kan t ex se till så att man kopierar över BASICINI.SYS som första fil efter att man har kört DOSGEN.

Bo Kullmar

Interface till Olivetti Praxis 35

Interface till Olivetti Praxis 35 kan man få billigt t ex genom att beställa från en del småfirmor utomlands.

Det var lätt att installera serieinterfacet (har en ABC802), tillverkat av TIMTOM MICRO, Penylan, Cardiff, Great Britain.

Lödningen tog cirka 25 minuter. Kostnad 75 pund, d v s över 3 300 kr billigare än vad Olivetti i Sollentuna kan åstadkomma.

Jag ringde till Olivetti några gånger och frågade om det här men fick ganska luddiga svar trots att jag köpt deras skrivmaskin. TIMTOMs interface har endast en mycket blygsam buffert men med den pris-skillnaden...

Nackdelar då?

Ja det finns. Man blir beroende av leverans och service från ett annat land. Detta tar dock max några veckor enligt min erfarenhet.

Beträffande komponenterna kan nämnas att TIMTOM tagit bort beteckningarna på chipsen så Du kan inte räkna med att själv byta ut ev pajade chips om Du inte är mycket bevandrad i chipsvärlden.

Jag och en kompis gick in i interfacet och trixade litet varvid detta framkom. Tycker själv ändå att beroendet är värt risken. Man kan ju t o m köpa två interface om tio dagars tidspill kan bli kostsamt, ändå tjänar man på affären jämfört med det svenska alternativet.

Om bufferten som Olivetti säljer verkligen motiverar de extra 3 300 kr undrar man varför Olivetti inte tillverkar flera olika typer av interface.

Har någon annan erfarenheter av det här?

Björn Cedervall

<4016>

PS

Det här brevet skrevs med hjälp av det interface som omtalas ovan.

Totalkostnaden för interface, sladdar och kontakter blev cirka 950 kronor. Den mesta tiden av installationsarbetet gick åt till att ta isär och sätta ihop skrivmaskinen samt för bekvämlighetens skull ordna med de extra kontakterna. Sammanlagt tog arbetet cirka två timmar.

Har någon en flexskiveenhet med dubbel densitet (FD2D/DD82) att sälja?

I så fall hör av Er till

<3063>

Henry Lidholm
Vårstavägen 99
140 32 Grödinge

0753-291 05 eller 0753-392 06

Spara attributminnet för ABC806

I en tidigare artikel om hardcopy för ABC800-datorer skrev jag att jag inte visste hur man kan spara undan attributminnet. Det har jag nu fått reda på och kan därför redovisa det här.

När man läser av ett tecken i bildminnet kan man samtidigt få tag på attribut-tecknet genom INP(53). Detta skall enligt uppgift gå bra både i assembler och BASIC. Jag har ännu bara provat det i BASIC.

I BASIC går man så här för att lagra undan bildminnet:

```
Tecken=PEEK(30720) : Attribut=INP(53)
```

För att skriva tillbaka det gör man:

```
OUT 53,Attribut : POKE 30720, Tecken
```

Använd ej Attribut som variabel, eftersom detta är ett BASIC-ord.

Nedan följer två BASIC-rutiner som omfattar hela bildminnet:

```
300 DIM Screen(2048),Attribut(2048)
2000 DEF FNSave806screen
2010 Curpos=PEEK2(65362)
2020 FOR I=1 TO 2048
2030 Screen(I)=PEEK(I+30719) :
Attribut(I)=INP(53)
2040 NEXT I
2050 RETURN 0
2060 FNEND
```

```
3000 DEF FNPrint806screen
3010 POKE 65362,Curpos,SWAP%(Curpos)
```

```
3020 FOR I=1 TO 2048
3030 OUT 53,Attribut(I) : POKE
I+30719,Screen(I)
3040 NEXT I
3050 RETURN 0
3060 FNEND
```

Ovanstående rutiner är dock ganska långsamma. Ungefär 6 sekunder tar varje rutin. Om man använder WHILE-loopar för att få I som lokal variabel, så tar det ytterligare någon sekund. Detta beror på att en WHILE-loop inte är så optimerad för snabbhet som FOR-NEXT-looparna.

Dessutom tar dessa rutiner upp 8 kbyte av minne. Skriver man om rutinerna i assembler så bör de bli avsevärt snabbare. Dock kvarstår då problemet med att det tar upp mycket minne för att lagra undan både bild och attributminnet, även om lagringen kan göras effektivt så att man enbart tar i anspråk 4 kbyte. En lösning på detta kan vara att skriva en assembler-rutin som lagrar undan bildminnet på en fil i RAM-floppyn. Då får man först testa om drivrutinen för RAM-floppyn finns laddad. Om så inte är fallet, ja då föreslår jag att man helt enkelt struntar i att lagra undan bild och attributminnet på detta sätt. Man gör helt enkelt som man gör på de andra 800:orna.

Bo Kullmar

Speciell teckenmode för ABC806

ABC806 kan ställas om med kommandot OUT 34,134 så att tangentbordet lämnar ett speciellt tal när en tangent trycks mer och ett tal när en tangent släpps upp. Detta gör att tangentbordet inte längre lämnar ASCII-koder utan VARJE tangent lämnar ett tal beroende på vilken tangent som har tryckts ner. Siffran 1 på den alfa-numeriska delen av tangentbordet ger alltså inte samma kod som siffran 1 på den numeriska delen av tangentbordet. Återställning till normal mode görs med OUT 34,6.

Denna speciella "up/down mode" kan användas om man vill flytta eller byta funktioner för tangenter. Därvid måste man dock skriva en assembler-rutin för att tolka tangentnedtryckningarna på det sätt som man önskar. Då måste man också själv hålla reda på om någon shift tangent har tryckts ned eller om capslock-tangenten är nertryckt. Lysdioden i capslock kan i "up/down mode" tändas med OUT 34,136 och släckas med OUT 34,8.

För att testa detta kan man skriva ett enkelt basicprogram:

```
10 OUT 34,134
20 FOR I=1 TO 30
30GET Ö$ : ; ASC(Ö$);
40 NEXT I
50 OUT 34,6
```

FOR-NEXT loopen finns med så att man kan komma ur "up/down mode" utan att initierat det från tangentbordet, vilket ju i detta läget är omöjligt.

Dessutom finns följande funktioner om också inte står i bruksanvisningen:

```
OUT 34,9 Ljud, 5 ms klick
OUT 34,10 Ljud, 20 ms klick
```

Bo Kullmar

Funktionstangenter i ABC800-datorer

Pf 1 ger decimalt värdet 192 och Pf 8 ger 199, medan SHIFT + Pf 1 ger 208 samt SHIFT + Pf 8 ger 215.

Detta verkar inte särskilt logiskt! Om man tittar på vilka binära koder som Pf-tangenterna genererar så kommer förklaringen fram!

```
Bit 0 är alltid 1
" 1 " " 1
" 2 - 3 ger ett tal decimalt 0 - 3
" 4 är alltid 0
" 5 - 7 ger ett tal decimalt 0 - 7
Bit 2 - 3 anger alltså om någon shift-
tangent är nedtryckt.
```

Om decimalt: Nedtryckt tangent:
0 Enbart Pf-tangent
nedtryckt

```
1 Pf+SHIFT
2 Pf+CTRL
3 Pf+SHIFT+CTRL
```

Bit 5 - 7 anger vilken Pf-tangent som är nedtryckt. Om man adderar 1 till värdet så får man fram Pf-nummret.

På tangentbordet ABC55 som normalt används till ABC802 generas Pf-koderna med hjälp av vanliga tangenter i kombination med SHIFT och CTRL. På samma sätt sker även generering av Pf-koder på tangentbordet ABC77, som normalt används till ABC806, men eftersom detta tangentbord också innehåller vanliga Pf-tangenter så är det inget som man använder.

Av Bo Kullmar

Hastigheten för klockan i ABC800

Två olika sätt att ändra klockans hastighet i ABC800-datorer har medlemmar kommit på.

Carl-Axel Pettersson, <2229>, föreslår följande:

```
OUT 99,208,99,X,99,Y
```

X kan t ex vara:

```
37 = Stänger av klockan
133 = Klockan går 16 ggr fortare
165 = Normalt värde
```

Y är tidskonstant som normlat är 125.

En annan medlem har hitta följande metod:

Klockans hastighet ändras med OUT 99, 133,99,N. N kan vara 0 - 255. Klockan kan stängas av med OUT 99,1.

Bo Kullmar

Problem med ABC806

På de först producerade ABC806 har upptäckts två fel vid speciella tillämpningar enligt Luxor Datorer Serviceinformation, 1983-09-06 2-01.

1. Vid körning mot expansionslåda kan det uppstå problem att läsa status från databussen.

2. Störningar på inkommande, klocka från tangentbordet. Datorn missar tecken från tangentbordet. Problemet visar sig vid körning med hög hastighet över V24:.

Om man har en ABC806 som berörs av ovanstående fel kan man få maskinen gratis lagad via sin återförsäljare. Det framgår också av nämnda servicemeddelande hur felet åtgäras, men detta måste ske av en tekniker.

Bo Kullmar



Kassetter till salu

ABC-Klubben har köpt ett antal ALME C60 kassetter som vi inte kan använda till ABC-Kassetter eftersom de inte går att snabbkopiera. Detta beror på att mekaniken är dålig i en del av kassetterna.

ABC-Klubben säljer därför ut kassetterna för 300 kronor per sats om 100 kassetter. Kassetterna går att använda, men är av dålig kvalite. De som är mekaniskt felfria kan gå att använda till data. Klubben lämnar ingen garanti på kassetterna, en del kan alltså vara dåliga. Vi säljer dem därför billigt.

Kassetterna erhålles genom att sätta in beloppet på klubbens postgirokonton för publikationer 62 93 00-5. Glöm ej den egna adressen.

ABC-Klubben/Styrelsen

SUPER SMARTAID -kraftfullt hjälpmedel för ABC 80

Starkt beroendeframkallande och med uppenbar risk för tillvänjning; det är omdömen som brukar mana till försiktighet i användningen i många fall. Beskrivningen kan emellertid helt berättigat appliceras även på OWOCO AB:s nya produkt SUPER SMARTAID. Och då är riskmomenten obefintliga, medan nyttan och de positiva effekterna är uppenbara.

SUPER SMARTAID är ett tillbehör till ABC 80 och betecknas av tillverkaren, Huddinge-företaget OWOCO AB, som ett programmeringshjälpmedel. Men det bjuder i realiteten på bekväma funktioner och god hjälp både före, under och efter själva programmeringsfasen. Autostart, KEY-funktioner, spooling, programkörning i "single step" och JOB-strömmar är några av dessa

funktioner. De ger ABC 80 ett helt nytt och intressant arbetsregister.

SUPER SMARTAID omfattar 10K-byte programvara. Den ligger lagrad i minneskapslar av PROM-typ. ABC 80:s eget arbetsminne tas alltså inte i anspråk, utan det kan användas ograverat för programmering. AID:en har dessutom ett eget arbetsminne (CMOS-RAM) på 2K-byte för lagring av olika uppgif-

ter. Allt som ingår i SUPER SMARTAID är "förpackat" i en bastant plåtlåda, klar att anslutas till den 64-poliga ABC-bussen.

SNABB INSTALLATION

Installationen är enkel. Man lossar de bakre gummifötterna som ABC 80:s tangentbord normalt vilar på. Därefter passas hjälpmedlets honkontakt in i kontaktdonet baktill på ABC 80. En plåtskena nedtill på AID:en ansluter då mot bottenplattan på tangentbordet och två små, förhandsborrade hål i plåtskenan ger möjlighet att skruva fast gummifötterna i datorn igen. Därmed sitter även AID:en fast. I mitt fall var passningen inte helt perfekt, men sedan hålen i plåtskenan hade förstörats något, gled skruvarna lätt in på plats.

När vi slår på spänningen är SUPER SMARTAID genast initierad och funktionsberedd. Datorns klarsignal "ABC 80" på bildskärmen ersätts nu med beskedet "SMARTAID" som tecken på att hjälpmedlet står till vårt förfogande.

Vid initieringen gör SUPER SMARTAID ett försök att ladda en fil och utföra ett JOB med namnet SUPER.JOB. Om JOB-filen är så utformad, att den startar ett BASIC-program, har vi på så sätt tillgång till en autostart-funktion.

Ett exempel: Vi vill ha autostart av programmet LIB.BAC och se storleken på filerna (alternativ "S" i LIB-programmens meny). På textskivan lägger vi då in följande textrad: "RUN LIB_M S". Denna rad har vi sparad under namnet SUPER JOB. Det hela resulterar i att SUPER SMARTAID startar och kör det önskade programmet vid initiering av AID:en. (En förkla-

```
SMARTAID
HELP
U      PEEK      FIND      CHANGE  VAR
SYS    KEY       CONT      START   RESUME
SPOOL  DISP      TLOAD    TSAVE   TMERGE
LIB     DIR       HELP     TAB     EX
AUTO   OLD       JOB      PEW     POW
TIME   NOTIME    STACK   BYE     RUN
LIST   NEW       SCR      CLEAR  LOAD
MERGE  REN       ED       SAVE   UNSAVE

SMARTAID
```

Figur 1
"HELP" hjälper oss att komma ihåg vilka kommandon som SUPER SMARTAID erbjuder.

```
SMARTAID
SYS
SUPER SMARTAID

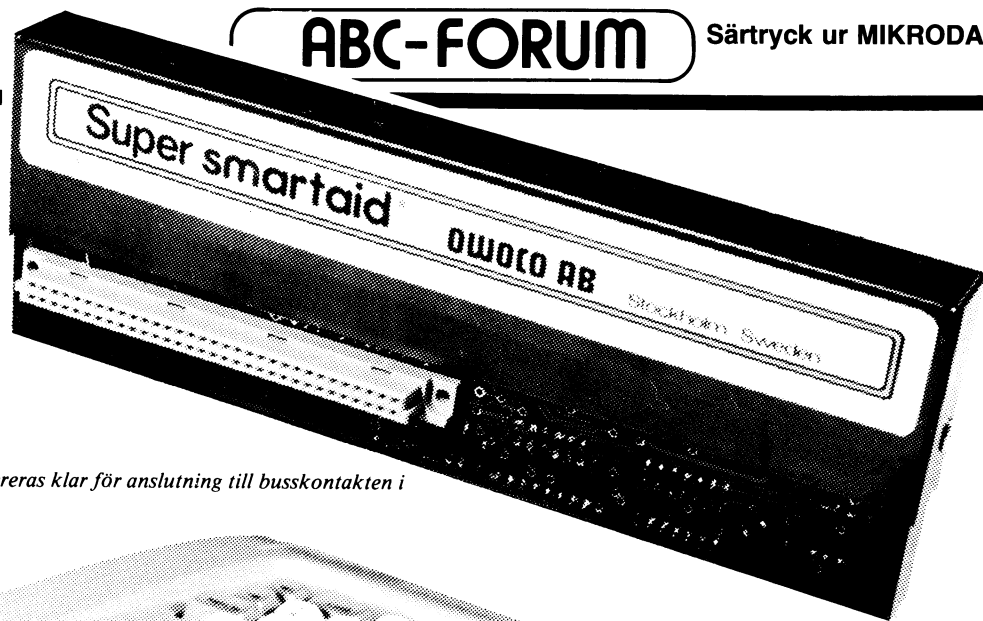
MEMMAP: PROG 1924  VAR 322  FREE 27205

SYSVAR: BOFA 32768 EOFA 34692 HEAP 35515
         STCK 62720 VROT 34693 RAM 32768
         BKGD 0    CMOS 22000 DVRT 28339

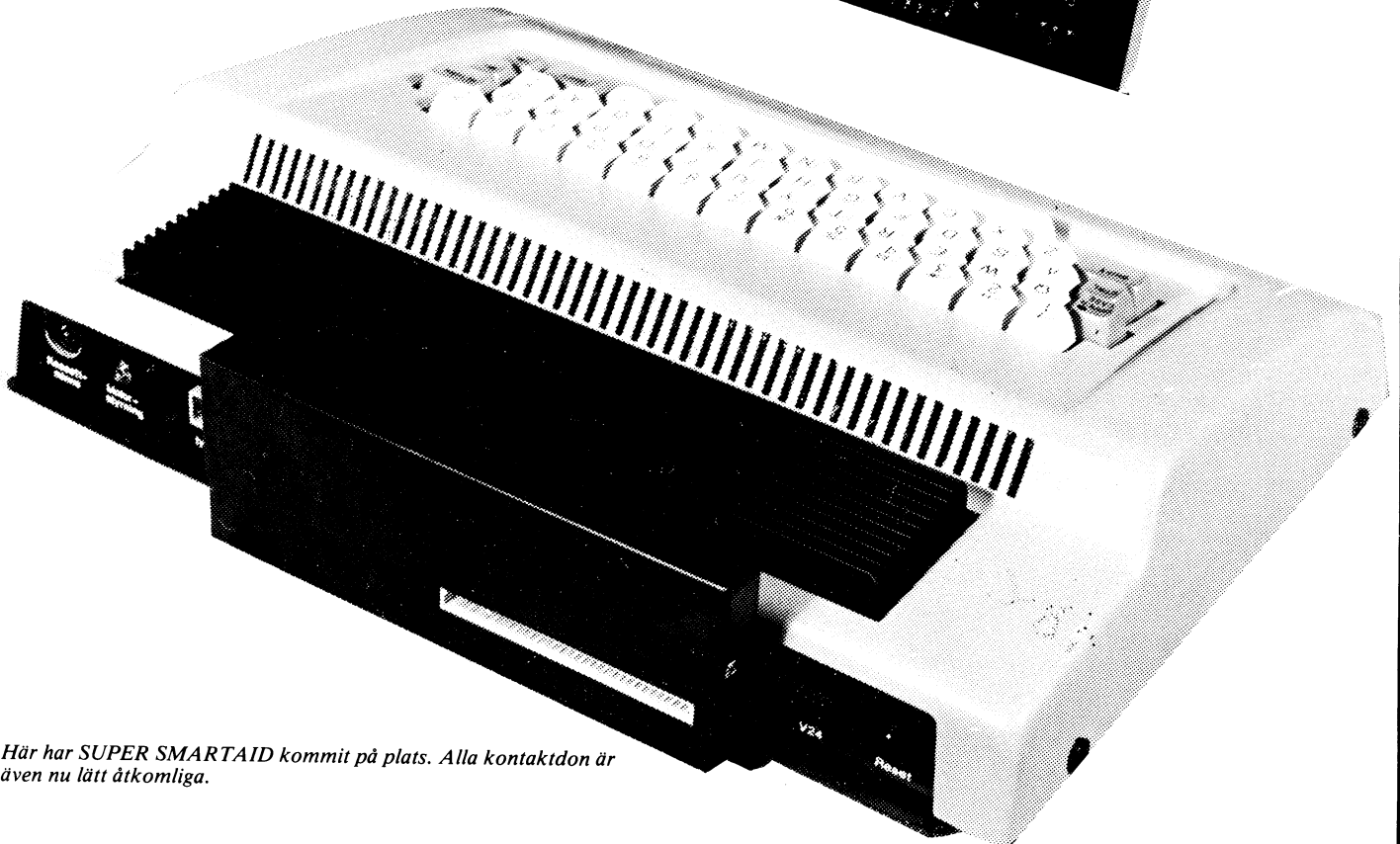
TIME : 19:43:32
DEVICE: DR0, DR1, DR2, DR3, DR4, DR5,
        DR6, , CAS, , PR, IEC

SMARTAID
```

Figur 2
SUPER SMARTAID ger omfattande systeminformation som svar på kommandot SYS.



SUPER SMARTAID levereras klar för anslutning till busskontakten i ABC 80:s tangentbord.



Här har SUPER SMARTAID kommit på plats. Alla kontaktdon är även nu lätt åtkomliga.

ring: __M är koden för CTRL-M = RETURN).

TEXTBEHANDLING

Det utmärkande för SUPER SMARTAID är att den förses ABC 80 med en rad nya direktfunktioner. För att skapa och spara den textrad, som förekom i föregående stycke, använder vi lämpligen AID:ens egen texteditor. Inskrivning av texten startar vi genom att som första programrad skriva en asterisk (*). Så här:

```
10 *
20 RUN LIB__MS
```

Med ED-kommandot tar vi

därefter bort rad 10. Den återstående, önskade, raden sparar vi sedan genom att skriva TSAVE SUPER. JOB. Snabbt och enkelt utan att behöva ladda in ett textbehandlingsprogram från någon yttre enhet! Raden sparas med det här förfarandet utan att föregås av radnummer.

Lika enkelt är det att med direktkommando läsa en textfil. En sådan hämtas genom att skriva TLOAD (filnamn). I detta fall tillfogas radnummer till texten och editering kan göras med ED-kommando. Vill vi slippa radnummer, kan textfilen läggas ut på skärmen med kommandot DISP (filnamn). I det fallet påverkas inte innehållet i datorns internminne av våra åtgärder.

EGNA FUNKTIONER

Hela SUPER SMARTAID:s kommandorepertoar listas på skärmen med HELP-kommandot. (Se figur 1.) Därutöver finns en rad funktioner, huvudsakligen för bildskärmseditering. Dessa styrs med ungefär 20 olika CTRL-(tecken)-kombinationer. Men dessutom håller AID:en fortlöpande reda på en rad systemdata, som vi kan få upp på skärmen, närhelst vi ger kommandot SYS. (Se figur 2.) Om nu inte allt detta skulle räcka till, har vi möjlighet att tilldela enskilda tangenter (eller CTRL + tangent) nya funktioner helt efter vårt eget huvud. Det gör vi genom att an-

vända KEY-funktionen.

Skriver vi exempelvis KEY_T = "TIME__M", tilldelas kombinationen "CTRL-T" funktionen TIME och lägger till RETURN. Detta medför att vi lägger ut datorns realtidsklocka längst upp till höger på skärmen. Dessförinnan har vi lämpligen ställt klockan, vilket vi gör med TIME hh,mm,ss.

De KEY-funktioner, som vi själva har definierat (Ex: KEY_N = "NOTIME__M"), sparas i AID:ens eget CMOS-RAM. SUPER SMARTAID inrymmer också en laddningsbar NICAD-ackumulator som fullladdad klarar av CMOS-minnets strömbehov under ungefär 60 dygn. De KEY-funktioner som vi

har lagt in finns alltså direkt tillgängliga även nästa gång vi slår på datorn. Ackumulatören får sin laddning under den tid som datorn är ansluten till elnätet (datorn påslagen).

NYA MÖJLIGHETER

Med kommandot KEY kan vi göra en listning på bildskärmen av de definitioner som ligger lagrade i CMOS-minnet. (Se figur 3.) Allt i en KEY-funktion som följer efter första "M" ignoreras. Därför kan vi, om vi så vill, skriva en förklarande kommentar efter KEY-definitionen. Exempel på detta framgår av figur 3.

Att KEY-definitionen bara kan innehålla "M" en gång innebär på sätt och vis en begränsning av vad som kan åstadkommas direkt med KEY-kommando. Låter vi emellertid som i figur 3 en KEY-funktion enbart starta en JOB-fil, kringgår vi lätt den begränsningen. En sådan fil kan klara många successiva uppgifter och se ut t ex som figur 4 visar. Där startar den ett program (TEST.BAS) och matar direkt in åtta svar på INPUT-satser i början av programmet. Finessen är alltså att allt detta kan åstadkommas med att bara trycka ned två tangenter CTRL+J. Och heter filen SUPER.JOB, får vi till och med automatstart.

PROGRAMMERINGS-HJÄLP

När vi sätter oss ned för att programmera, sköter datorn själv om radnumreringen, sedan vi gett kommandot AUTO. Gör vi syntaxfel i en lång programrad och får felmeddelande, är inte raden räddningslöst förlorad som med den "nakna" ABC 80. Med CTRL-tangenter kan vi flytta upp markören på raden igen, rätta felet och därefter mata in den korrekta lydelsen. Både INSERT- och DELETE-möjligheter finns vid editering. Vi kan välja på ett flertal sätt för att flytta markören över bildrutan och för att mata in text i inmatningsbufferten. CTRL-L rensar skärmen helt eller delvis.

```
SMARTAID
KEY
_J="JOB SUPER.JOB M" ! Startar fördefinierad JOB-fil.
_T="TIME M" ! Visar realtid i övre högra hörnet av skärmen.
_N="NOTIME M" ! Tar bort utskriften av TIME från skärmen.
_U="U-M" ! Listar från programmets början.
SMARTAID
```

Figur 3

Med KEY får vi besked om vilka KEY-funktioner som ligger lagrade i CMOS-minnet.

```
SMARTAID
TLOAD SUPER.JOB
SMARTAID
U
10 RUN TEST.BAS M J M 12 M 3 M 25 M ; "HEJ" M ; "HOPPSAN" M
; "GÄR" M ; "BRÄ" M
SMARTAID
```

Figur 4

Exempel på hur en JOB-fil kan utformas.

Vill vi veta vilka variabler som har tagits i anspråk i ett program, ger vi kommandot VAR. Med CHANGE kan vi i ett enda moment göra generella ändringar i ett helt program. Exempelvis ändra variabeln B till B% på alla ställen där den förekommer. Det är ett bekvämt sätt att använda i stället för att hela tiden brottas med att behöva skriva %-tecknen manuellt.

REN-kommandot får utökad repertoar med SUPER SMARTAID. Det är t ex möjligt att numrera om enbart ett avsnitt i ett program, medan resten behåller sina radnummer. Den möjligheten kan även användas i text-mode och ger då möjlighet att flytta hela textavsnitt inom den textmassa som ligger i internminnet (BLOCK MOVE).

Med kommandot DEL (DELETE) tar vi bort en eller flera rader ur programmet med ett fåtal tangentnedtryckningar.

MÅNGSIDIGT

Även LIST-funktionen är förbättrad. Som snabbkommando kan vi skriva "Ü RETURN". Listning sker då (första gången) från första programraden. Nästa gång vi använder samma kommando, startar listningen med den skärmbild, där föregående listning avbröts. Listningen kan rullas (scrollas) nedåt eller uppåt på skärmen med hjälp av piltangenterna. Skriver vi "LIST-" eller "Ü-", sker listningen alltid

från första raden. Det kan vara praktiskt att, som i vårt exempel (figur 3), lägga in den varianten som en bekväm KEY-funktion.

FIND är ett kraftfullt kommando att använda när vi söker efter en variabel, en sträng eller ett radnummer i ett program eller i en text. Alla rader där det sökta begreppet återfinns listas då i sin helhet. Med mellanslagstangenten eller RETURN släpper vi fram en sådan rad i sänder, till dess att vi når slutet på programmet. Editering kan vi göra direkt utan att behöva gå ur FIND-funktionen. Det hela är mycket värdefullt vid felsökning och "avlusning" av program.

Vid vanlig programlistning kan editering ske över flera rader i "ett svep". Ett antal fördefinierade CTRL-tangenter ger oss då möjlighet att direktflytta markören ett stycke framåt eller bakåt i texten till ett önskat tecken.

STEGVIS

När programmet är klart att köras, kan vi göra det på vanligt sätt med RUN. Men vi kan också starta från valfri rad. I det fallet skriver vi START (radnr) i stället för RUN.

Efter stopp i programmet (vid STOP eller ERROR) går det att återuppta körningen med behållna variabelvärden. Det sker med CONT eller RESUME. Datorn kör då automatiskt fortsättningen av programmet i

"step-mode". Det innebär att exekveringen sker en rad i sänder i den takt som vi själva bestämmer. Även i det fallet står flera möjligheter till vårt förfogande.

Med CTRL-D får vi den programrad som står i tur att exekveras utskriven i sin helhet. Det sker längst ned på bildskärmen. Med den stegvisa funktionen kan vi läsa rad för rad i den takt vi trycker fram nya och direkt se vad de utför. Det är en funktion som har klara pedagogiska fördelar.

Väljer vi i stället CTRL-T, får vi tillgång till TRACE-funktion. Enbart radnumren visas då på nedersta skärmraden, när vi stegar oss framåt i programmet. CTRL-S, slutligen, använder vi vid stegvis körning, när vi inte behöver något av den hjälpinformation som CTRL-D eller CTRL-T kan förse oss med. Vid normal körning (utan STOP eller ERROR) kan vi direkt sätta datorn i STEP-MODE med tangentkombinationen CTRL-C.

När det är dags att spara vårt program på flexskiva, kan vi först ge kommandot LIB (eller DIR om vi har 8-tums flexutrustning). Då får vi direkt upp information om vilka filer som redan finns på skivan och hur mycket utrymme som återstår. Programmet i datorns primärminne påverkas inte av den åtgärden.

SPOOLING

Med CTRL-SHIFT-O kan vi när som helst "dumpa" skärminnehållet på en ansluten skrivare. Printerparametrarna ligger för övrigt lagrade och klara i CMOS-minnet, sedan de en gång har matats in.

Lista programmet på skrivare kan vi göra på vanligt sätt, med LIST- (eller Ü-) (filnamn) PR:, men här bjuder SUPER SMARTAID återigen på valmöjligheter. AID:en klarar nämligen också spooling. Det är därmed möjligt att överföra textfiler till printer samtidigt som ABC 80 är användbar till andra uppgifter, t ex programmering. Vill vi nyttja den funktionen, skriver vi SPOOL (filnamn), vilket startar utskriften.

ABC 80 delar nu sin tid mellan förgrunds- och bakgrundsjobb. Utskrift på printern pågår hela tiden och datorn har till sy-

nes tid med operatören bara lite då och då. Skriver vi exempelvis med jämn och snabb takt in ett program eller en text, kommer resultatet upp "skurvis" på skärmen. Tecknen kommer med viss fördröjning, flera samtidigt vid varje tillfälle. Detta verkar ovan och något förvirrande, åtminstone i början. Men det fungerar ovedersägligen. Våra tangentnedtryckningar lagras först i en typeahead-buffert för att därefter tas om hand av datorn, när den "får tid". På grund av det här beteendet sätter jag nog ett litet frågetecken i kanten beträffande möjligheten att verkligen kunna dra nytta av SPOOL-funktionen i praktiskt arbete.

MER FINESSER

Med ovanstående beskrivning har jag någorlunda utförligt exemplifierat hur SUPER SMARTAID kan vara till snabb och omedelbar hjälp under ett körpass. Men ytterligare möjligheter återstår. Med PEEK eller PEW (PEEK a Word) kan vi direkt lista och läsa en respektive två byte i taget i primärminnet. Liksom vid listning av program

kan vi scrolla minnesinnehållet (vid PEEK) framåt eller bakåt med piltangenterna. Med POW (address) skriver vi in två byte direkt i primärminnet, medan POKE används som vanligt.

Bland ytterligare specialiteter bör jag slutligen nämna även kommandot OLD. Med det kan vi återkalla och köra ett program även efter NEW eller RESET. En annan finess är körning av FOR-NEXT-loopar även som kommando, utan radnummer.

SLUTOMDÖME

I praktisk användning (under några månaders tid) har SUPER SMARTAID visat sig helt problemfri. Det mest dramatiska som har inträffat är att CMOS-minnet vid ett par tillfällen har tappat sitt innehåll. Det inträffar om ABC 80 "spårar ur" i samband med någon, oftast olämplig, åtgärd. Vid sådana tillfällen skrivs "CMOS JEOPARDIZED" för övrigt ut vid nästa initiering av AID:en. Det är ett besked om att data i CMOS-minnet har gått förlorade eller är otillförlitliga.

Med den stora funktionsrepertoar som SUPER SMART-

AID har, tar det en hel del tid att lära sig att hantera alla finesser med någorlunda fulländning. De 20 CTRL-tangenternas funktion (och ett varierande antal egna KEY-funktioner) kan vara svåra att ha aktuella i huvudet. Men man lär sig snart nog att behärska några favorithandgrepp som klarar elementära men nyttiga grundfunktioner. I övrigt listar man lämpligen övriga funktioner med HELP-kommandot, när det egna minnet sviker.

Slutligen går det bra att rådfråga den bruksanvisning som finns. Den omfattar 25 maskinskrivna A4-sidor. I den beskrivs, omfånget till trots, hjälpmedlets funktioner tämligen koncentrerat och där ges exempel på hur man kan göra. OWOCO AB betecknar själv (juni 1983) bruksanvisningen som preliminär. Men den är helt tillfyllest för den som redan är någorlunda bevandrad med den terminologi som förekommer i datorsammanhang. Nybörjaren kan kanske ha lite svårare att tillgodogöra sig hela innehållet.

Samma omdöme kan för övrigt gälla SUPER SMARTAID som helhet. Alla — inte minst nybörjare i datorsammanhang — har stor nytta av flertalet av

hjälpmedlets bekväma och snabba funktioner. Men det är främst den något mer avancerade tangenttryckaren och programmeraren som kan använda och ha nytta av hela den breda funktionsrepertoaren. Det är med andra ord ett hjälpmedel med "högt i tak". Något för flertalet att växa i.

MYCKET PENGAR — ÄNDÅ BILLIGT

Finns det då inte några nackdelar bland alla de positiva egenskaperna? Ja, det skulle i så fall vara priset. Den som kompletterar sin ABC 80 med SUPER SMARTAID får räkna med att spendera nära 2 800 kronor plus moms. Det är alltså den summa som går åt för att förvandla ABC 80 till en verkligt intressant maskin. Men det är en summa som måste anses i hög grad överkomlig med hänsyn till resultatet — en uppdaterad och i många avseenden "ny" och kraftfullare dator.

Karl-David Höglund
ABC-Forum
Box 8040
900 08 UMEÅ

Meddelande till ABC-klubbens medlemmar!

Som framgår av denna artikel lovordas SUPER SMARTAID av författaren Karl-David Höglund.

Han säger vidare att om något kan räknas till nackdelarna så skulle det möjligen vara priset.

Vi har tagit fasta på detta och kan glädja medlemmarna med ett sänkt pris till 1.985:- inkl moms.

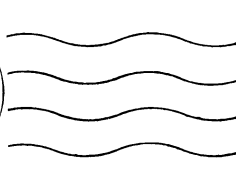
**Beställ Din SUPER SMARTAID direkt från
OWOCO AB, Kvarnbergsvägen 25, 141 45 Huddinge
tel. 08 - 774 02 90.**

**Du kan också kontakt någon av OWOCO:s återförsäljare
som Du får tel.nr till genom att ringa OWOCO.**

**PS. Vi har infört nya finesser till SUPER SMARTAID (okt. -83).
Även SMARTAID 3 har fått sänkt pris 995:- inkl moms.**

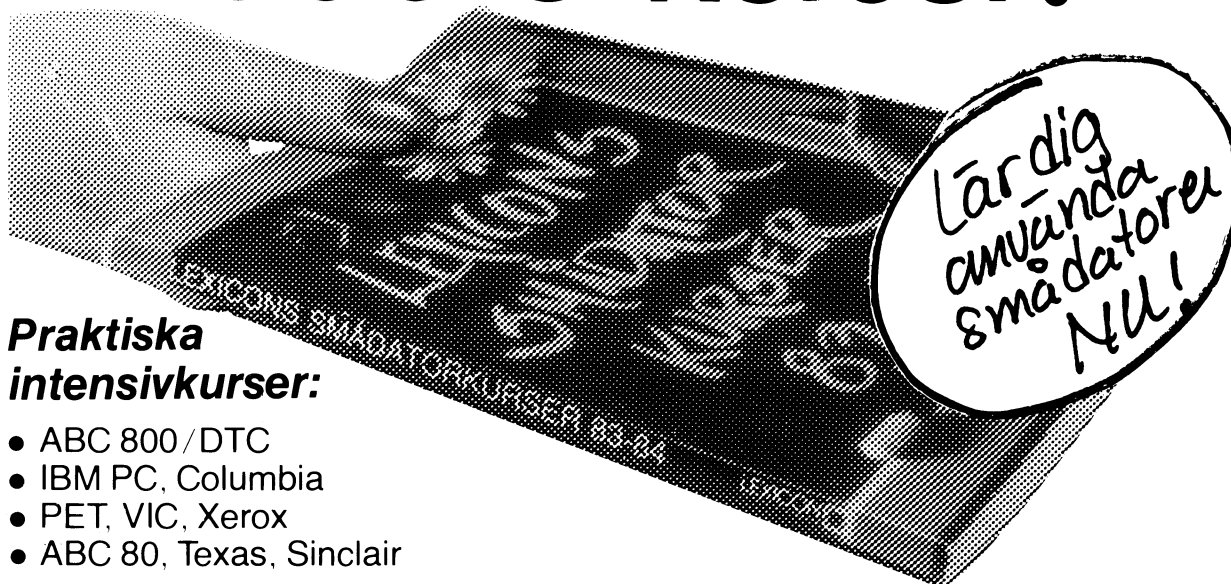
ABO 1982 ABC-bladet 4/1982
LUND OLA

SJÄMMANSVADEN 14
411 00 FALKENBERG



Kom ihåg att anmäla adressändring i tid

Varsågod **700** smådatorkurser!



Praktiska intensivkurser:

- ABC 800/DTC
- IBM PC, Columbia
- PET, VIC, Xerox
- ABC 80, Texas, Sinclair
- CP/M-, MS DOS- och UNIXkurser
- Användar- och programmeringskurser
- Special- och skräddarsydda kurser
- Hemdatorkurser
- Programmeringskurser

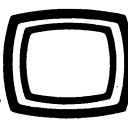
Kursexempel: Nybörjarkurser, Ordbehandling, BASIC, Köpa dator, Teledata, Kalkyl, Basregister, Bokföring, Mät & Styr, Datakomm, Nätverk, Databas, Systemering, JSP, CP/M, Assembler, C, Unix, Pascal, Cobol, FORTRAN.

JA, sänd er 132-sidiga kurskatalog GRATIS.

Namn _____

Företag _____

Postadress _____

LEXICON 
utbildning på smådatorer

Box 136, 182 12 Danderyd. Tel. 08-753 31 40

