

"STORDATORN, var god dröj!"

STACKPOINTER



We Specialize
in Computer Letters

REDAKTÖREN HAR ORDET

STACKPOINTER GÅR ÖVER TILL "DATASÄTTNING"

Från och med detta nummer börjar vi att använda textformatteringsprogrammet ZFORM från Zilog. Med hjälp av detta program kan artiklar skrivas in som råtext utan några finesser. Man lägger sedan in styrkommandon i den färdiga texten för att tala om hur man vill ha utskriften redigerad, t.ex. raka högermarginaller, indelning i stycken, centrering av rubriker, fotnötter, automatisk sidnumrering m.m.

Alltså. När Du vill knacka in Din artikel till nästa nr av STACK-
POINTER: knacka in den på Zilog-systemet i Stacken-lokalen som
råtext, så fixar vi på redaktionen så att det blir flott sedan.

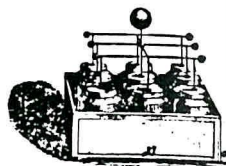
Tips: Börja alltid ny mening på ny rad. På det viset kan vi bryta
texten på godtyckligt ställe.

Är du intresserad av att själv greja med ZFORM finns det diskett och
dokumentation i Zilog-systemets softvarupärm i Stacken-lokalen.

Redaktionen

Björn Ahlén Automatus Sapiens
Hans Nordström tf Redactor Secretarius

COVER: drawn by KENNETH CHAUNDY, 14 bast.



507. Hattori.



Stackpointer utges av mikrodatorföreningen STACKEN,
Box 5079, 141 05 Huddinge, postgiro 433 01 15-9.
Ansvarig utgivare: Nils Söderman, Fredrikssonsv. 12, Hägersten.

STUDIEBESÖK!

Den förträfflige Peter Lagerberg i Stacken har lyckats ordna ett studiebesök på IBM:s Järfälla-fabrik. Studiebesöket går av stapeln - den 6:e november kl. 17.30 - . Eftersom Järfälla ligger ute i vildmarken (busjen), är det svårt att komma dit med allmänna samfärdsmedel. För dig som har punktering på den 18-växlade Peugeot-cykeln, kedjebrott på den gamla fina Turbo-Kawasakin eller förgasarrhosta på 485-hästarnas Corvetten, erbjuds ett gott alternativ: AB Traveller ställer upp gratis med en turistbuss för skjuts till/från Järfälla. Vi samlas kl. 17.00 utanför AB Travellers lokaler på Mäster Samuelsg. 71. (Bredvid Markurells, nära Centralen.)

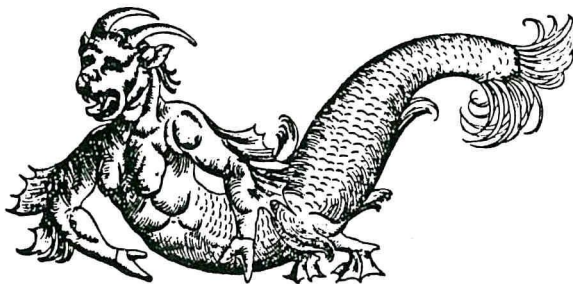
Du som vill följa med skall anteckna Dig på studiebesökslistan i Stacken-lokalen.



Tandem Computers är ett mycket intressant midiätor-företag som ej skall förväxlas med Tandy Computers som tillverkar TRS-80. Tandem är specialister på datorsystem med exceptionellt hög tillförlitlighet, NonStop-system. Både hårdvara och mjukvara är konstruerade för att fungera även under besvärliga Murphy-konditioner. Allting är dubblerat inklusive programmet Du kör! Det i Sverige först installerade systemet finns på Vingresor.

Dag: 22 november, tid och lokal meddelas senare. Anmäl Dig på listan i Stacken-lokalen.

* midi = mitt emellan maxi- och mini-



Z8000

En sportig liten jätte maskinellt toppklassig



Z8000 - EN Z80 MED TVÅ NOLLOR EFTER ?

Några av Stackens medlemmar har nu haft tillfälle att programmera och köra Z8000 i ca en månads tid.

Z8000 är en registerorienterad 16-bits mikro- dator med mycket höga prestanda. Man talar här om en ny generation mikrodatorer med generell arkitektur, kraftfulla adresseringsmoder och ett stort adressområde omfattande flera Megabytes.

Z8000 har 16 generella register som kan användas som ackumulatorer, stackpekare, indexregister etc. En 16-bits registerload tar ca 750 ns vilket är ungefär jämförbart med en mediumstor PDP-11 med kärminne. Man har alltså idag på en enda chip prestanda som tidigare krävdes en hel rack för.

Som vanligt när det gäller Zilog har man en mängd färdiga blockinstruktioner för vanligen förekommande programmeringsuppgifter. Fördelen med blockinstruktioner är att de utförs snabbare än motsvarande kod skrivet som subrutin och att de ger mer lättlästa program. Bl.a. finns instruktioner som flyttar text i minnet, jämför strängar, söker efter vissa ASCII-karaktärer eller t.o.m. finner nästa större tal i en vektor.

Man kan också göra I/O direkt till/från minnet med upp till 0,5 Mbyte i sekunden, översätta mellan olika koder, t.ex. EBCDIC till ASCII direkt i en buffert eller kolla om en textsträng innehåller något av flera specialtecken.

Några exempel:

1.

Sortering av elementen i en vektor av 16-bit integers (med tecken). Man vill hitta nästa element i vektorn som är mindre än ett visst tal.

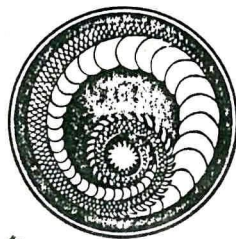
R1 innehåller jämförelsetalet, R2 pekar på vektorn, R3 är dimensionen på vektorn.

```
CPIR    R1,R2,R3,LT    ! Sök elementvis till condition code !
                        ! LT = less than                        !
```

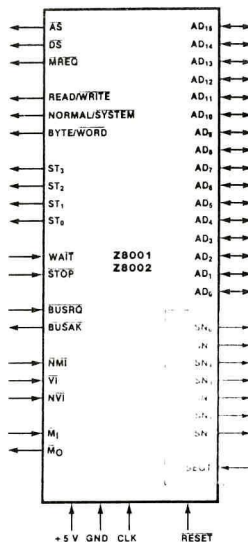
2.

TVå personnamn i minnet. Vem skall stå först i telefonkatalogen? R4 pekar ut första namnet, R5 det andra, R10 är max. antal tecken.

```
CPSTRB  @R4,@R5,R10,UGT ! Compare String, Increment, Repeat, Byte !
                        ! Jämför tknvis tills Unsigned Greater Than !
```



Knäcker det mesta



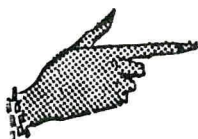
5

Nog med programexempel. De flesta av Z8000:s instruktioner kan verka på 8, 16 och 32-bits operander. Man skiljer på adresseringen med suffixen 'B' för byte (8 bitar) eller 'L' för long (32 bitar).
Ex.

ADDB	RH0,#25	! Addera 25 till 8-bits reg RH0 !
TEST	MAFFE(R8)	! Testa 16 bitar i MAFFE, index i R8 !
SRAL	RR2,#28	! Shift Right Arithmetic Long, 28 snäpp !

Mera om Z8000-programmering i kommande artiklar bl. a. om Z8000 LISP skrivet i PLZ/ASM av Dag Rende/Erik Forsberg och Z8000 Tiny BASIC av Björn Ahlén.

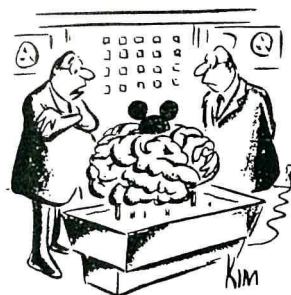
BA



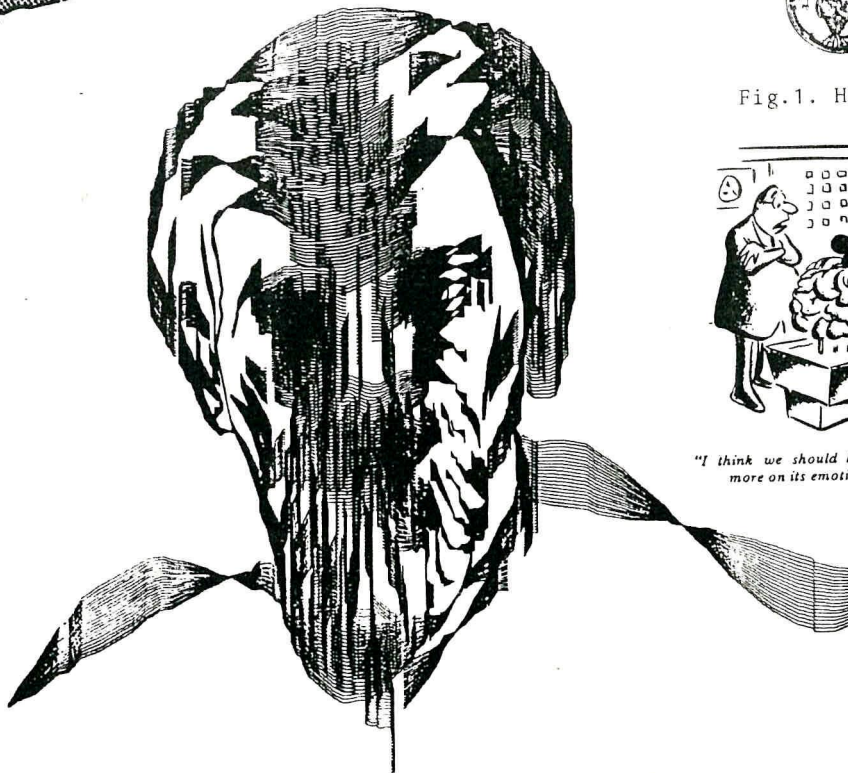
Förr styrdes världen av sina stora ledare,
nu styrs vi av våra små halvledare!



Fig.1. Halvledare.



"I think we should have concentrated a bit more on its emotional development."



"JAG HAR OCKSA GATT ÖVER TILL Z80!"

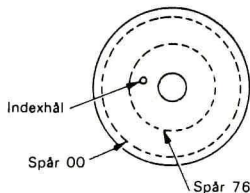
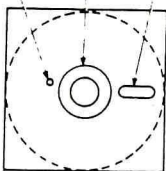


Vi ska titta på två typer av minnesmedia som används i datoranläggningar.

Den första, disketten, består av ett slutet skivomslag med en plastskiva i. I skivomslaget finns öppningar för drivkonan, skriv o läshuvud och indexhål. Se fig. 1. Skivan är några tiondels mm tjock och har en magnetiserbar beläggning. Vid användningen av disketten tänker man sig att skivan har 77 spår (0 t o m 76). Varje spår är uppdelat i ett antal sektorer, vanligen 26 st. Se fig. 2. En viss sektor kommer datorn åt genom att låta drivenheten starta på spår 0 och räkna sig fram. Vid rätt spår räknas rätt sektor fram från indexhålet. En läsning görs för att kontrollera att skriv o läshuvudet hamnat rätt. Under dessa operationer ligger skriv o läshuvudet emot disketten med ett tryck av 25 g.



Öppningar för
index/sector-
markering drivmotor läs/skrivhuvud

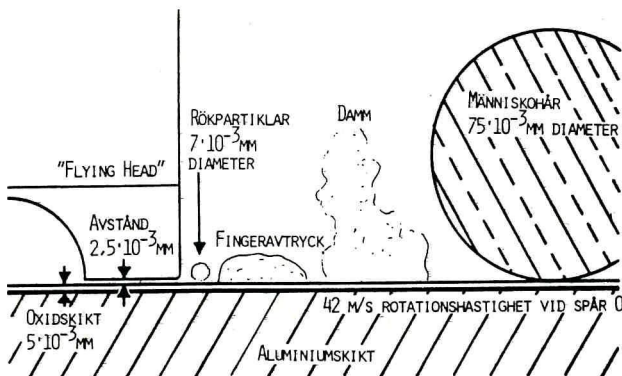
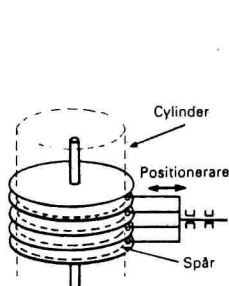


Den andra typen, hårddisken, består av en eller flera staplade skivor av metall. Skivan är omkr. 1 mm tjock. Vid flera skivor, monteras de samman på en axel med mellanrum mellan skivorna. Se fig. 3. På metallskivan finns en magnetiserbar oxidbeläggning. Disken är även här indelad i spår och sektorer. Är det en flerskivig disk brukar man ge spår med samma nummer benämningen cylinder. Till skillnad mot disketten brukar skriv o läshuvudet i drivenheten sväva några tusendels mm över oxidskiktet.



Om det finns något i vägen vid läsning eller skrivning på oxidskiktet så är det normala att datorn försöker göra om operationen, efter att först ha flyttat huvudet till ett annat spår. Lyckas det inte efter ett antal försök så blir det ett avbrott i körningen. Om det händer så brukar det hända när man skall avsluta ett jobb som tagit flera timmar att göra. Det hjälper lite om man tänker på vilka några av marödörerna kan vara, damm, rök el. likn. Se fig. 4.

HN



När man säger att ett programmeringsspråk är ett högnivåspråk menar man ofta att språkets uttrycksmedel (satser, datatyper etc.) är utformade så att man lätt kan använda dem till att beskriva sitt problem, snarare än att de är utformade efter hårdvaran. Enligt detta kriterium skulle assembler (förstås!) vara ett lågnivåspråk, men också BASIC och FORTRAN (är inte det vad man alltid har tyckt??). Se till exempel på FORTRANS aritmetiska IF, "IF (A+2) 10,20,30", som är utformad direkt efter en speciell maskininstruktion på den första maskin som FORTRAN implementerades på. På motsvarande sätt skulle PASCAL, LISP, ALGOL 68 och SIMULA (klassbegreppet!) vara typiska högnivåspråk. Dessa språk har dock i viss grad fortfarande kvar egenskaper från hårdvaran t.ex. utförandet av satser efter varandra (maskininstruktionerna utförs ju i följd så länge man inte har några hopp) och variabler som lådor där saker och ting lagras (=minnes-cellen).

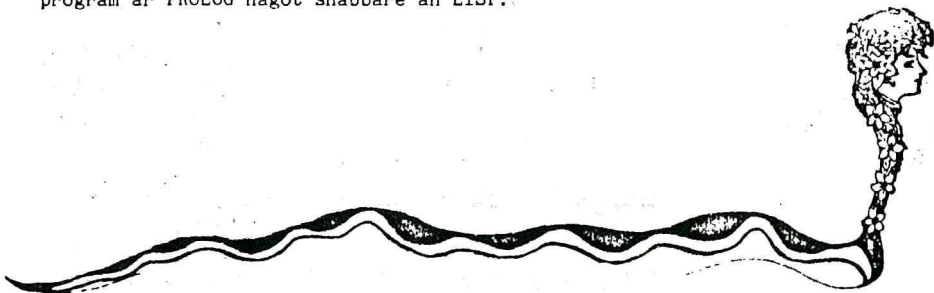
På senare år har det skett en utveckling av programmeringsspråk på mycket hög nivå (s.k. "very high level languages", VHLL). Typiskt för dessa är (bland annat) att de nästan inte alls har kvar några egenskaper från hårdvaran de körs på samt att de tillåter ett uttryckssätt som ligger närmare människans sätt att tänka.

Ett sådant VHLL är PROLOG. PROLOG är praktiskt taget ren logik och vilar därför (liksom LISP) på en fast matematisk grund. Datastrukturerna i PROLOG liknar dem i LISP, men längre än så sträcker sig inte likheterna med LISP (och de flesta andra språk) sträcker sig.

Program i PROLOG utgörs av ett antal så kallade axiom. Dessa anger hur programmets indata skall förhålla sig till dess utdata. Något förenklat kan man säga att ett PROLOG-program består av en serie tester som testar om programmet gett rätt resultat. Nu är emellertid dessa tester hela programmet! I PROLOG talar man alltså inte om hur resultaten skall tas fram utan hur de skall se ut. HUR beräkningarna skall göras listar interpretatorn ut själv. Problemet är alltså att hitta de rätta testerna. Har man väl funnit dem är å andra sidan chansen stor att programmet är riktigt.

I vanliga programmeringsspråk (på låg nivå eller bara "vanlig" hög nivå) måste man använda loopindex, flaggor, tilldelningar, beräkningar etc.. Dessutom är den inbördes ordningen mellan allt detta av stor betydelse. I PROLOG spelar det i stor utsträckning ingen roll i vilken ordning de olika delarna i programmet skrivs. (Ordningen har dock oftast STOR betydelse vad beträffar programmets effektivitet).

Allt detta bidrar till att PROLOG-program både går snabbare att skriva och blir kortare än program i andra språk. Dessutom är ett PROLOG-program i allmänhet lättare att förstå än program skrivna i andra språk. (Om man kan PROLOG förstås...). PROLOG-program kör också med acceptabel hastighet. Kompilerade PROLOG-program (på DEC-10) tar c:a 50% längre tid än motsvarande kompilerade LISP-program när programmen är små. För större program är PROLOG något snabbare än LISP.



Exempel på hur ett enkelt PROLOG-program kan se ut:

Antag att Kurt och Stina har två barn, Kalle och Sigge, samt att Kurts föräldrar heter Adam och Eva. Stina som är hög av PROLOG, vill skriva ett program som sköter om en liten databas för att hålla reda på familjeförhållandena. Hon skriver då på följande sätt: (PROLOG promptar med 'ø'. /* och */ omsluter kommentarer).

```
ø far(kurt, kalle).      /* Tala om att Kurt är far till Kalle. */
ø far(kurt, sigge).      /* Motsvarande */
ø mor(stina, kalle).
ø mor(stina, sigge).
ø far(adam, kurt).
ø mor(eva, kurt).
ø far(gud, adam).
ø                          /* Nu kan man ställa frågor till systemet! */
ø ?-far(kurt, kalle).    /* Är Kurt far till Kalle? */
yes                      /* Det var han! */
ø ?-far(kurt, edvin).    /* Är Kurt far till Edvin? */
no                       /* Nähä */
ø ?-far(kurt, X).        /* Har Kurt några barn? I så fall vilka */
X = kalle;              /* Kalle är ett svar. Genom att skriva ett
                        semikolon får man PROLOG att söka efter
                        flera */

X = sigge;

no                       /* Efter Sigge fanns inga flera svar. */

Nu kan man lägga in lite mera avancerade familjerelationer:

ø farfar(X, Z) :- far(X, Y), far(Y, Z). /* X är farfar till Z om X är far
till någon (Y) och denna någon är far till Z */
ø ?-farfar(adam, sigge). /* Är Adam Sigges farfar? */
yes
ø ?-farfar(adam, kurt).  /* Är Adam Kurts farfar? */
no
ø ?-farfar(adam, X).     /* Vilka är Adam farfar till? */
X = kalle;              /* Kalle till exempel */

X = sigge;

no
ø ?-farfar(X, kalle).    /* Vilka är farfäder till Kalle?? */
X = adam;
no                       /* Bara en förstås */

Håll i er inför nästa exempel!

ø ?-farfar(X, Y).        /* Tala om alla farfäder till alla! */
X = adam
Y = kalle;              /* Adam är farfar till Kalle t.ex. */

X = adam
Y = sigge;
```



X = gud
Y = kurt;

```
no                /* Nu var det inga fler */
ö
```

Observera att programmet "farfar" endast är en halv rad långt, men ändå klarar av fyra olika saker. Nämligen:

- Att testa om person A är farfar till person B.
- Att ta fram farfadern till en person.
- Att ta fram alla personer som har samma farfar.
- Att ta fram alla farfäder till alla personer överhuvudtaget.

Den här mångsidigheten är typisk för PROLOG-program.

Ytterligare några exempel på hur korta och snygga program i PROLOG blir:

För LISP-kunniga: funktionen APPEND. ([och] omsluter listor i PROLOG).

```
ö append([], X, X).
ö append([X,..A], B, [X,..C]) :- append(A, B, C).
```

Ett program som utför en sortering:

```
ö sort([], []).
ö sort([A,..B], C) :- sort(B, D), merge(A, D, C).
ö merge(X, [], []).
ö merge(X, [A,..B], [X, A,..B]) :- X<A.
ö merge(X, [A,..B], [A,..C]) :- X>=A, merge(X, B, C).
```

Den vars intresse för PROLOG skulle ha väckts kan pröva själv om han/hon har lyckan av ett ppn på DEC-10 vid QZ. Info kan fås genom ".HELP PROLOG". Manual finns på DOC:-arean och underverket startas med ".R PUB:PROLOG".

PROLOG är ännu på utvecklingsstadiet och finns endast på ett fåtal datorer. Jag har planer på att utveckla ett PROLOG-system i UCSD PASCAL, vilket skulle göra språket tillgängligt på bl.a. 280 system.

Det finns ingen gud utom PROLOG och PASCAL är hans profet.

Lars-Henrik Eriksson



"Men magistern, bara för att jag skrev GOTO...."

Då undertecknad för en tid sedan planerade för bygget av en digital musiksyntetisator, visade det sig snart att det var orealistiskt att huvuddatorn skulle ansvara för styrningen av de individuella styrparametrarna (grundton, spektrumenvelopp, tidsenvelopp, vibrato- och svepdata etc.) för 16 stycken digitala oscillatorer. Dessutom skulle den administrera in- och utmatning från disk samt övervaka en klaviatur, allt i realtime och helst med en tidsupplösning på 1 millisekund (dock ej för floppyn).

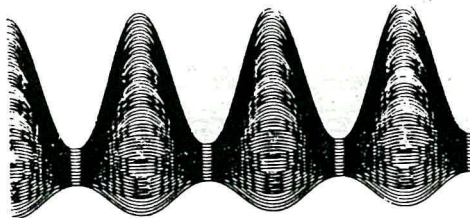
Eftersom detta inte går att klara med en mikrodator verkade det bästa vara att ge själva synthesizern en egen CPU för parameterstyrningen. Denna skulle då få ta emot order på en högre 'intelligensnivå' från huvuddatorn och sedan själv sköta om detaljerna.

Ett elegant sätt att ordna programvaran till slavdatorn är att låta huvuddatorn ladda slavens program i dess minne. På så sätt kan slaven få just de egenskaper man vill ha i en viss situation.

Dock, för att ladda slaven behöver den ett laddningsprogram som 'pratar' med huvuddatorn och tar emot dess data, lagrar i minnet och kollar checksumma samt, när laddningen är klar, hoppar till det just laddade programmet och exekverar det. Efter en stunds funderande verkade det dessutom som om det behövdes något sätt att kommunicera direkt med slaven via en terminal för att kunna studera styrprogrammets funktion i detalj (d.v.s avlusa det). Med andra ord: en PROMmonitor med laddprogram. Eftersom man normalt inte skulle ha någon terminal ansluten till slaven måste laddprogrammet kunna initieras från huvuddatorn.

Efter att ha tänkt ytterligare ett tag uppstod tanken att låta monitorn bli 'generell' i någon mening och uppta ett max PROM-utrymme av 2K byte. Sedan skulle man utan att ändra 'grundPROMet' kunna lägga till ett andra (tredje, fjärde) PROM och få utökade möjligheter.

I mitt fall skulle då laddprogrammet plus diverse andra småsaker ligga i extraPROMmet. Laddprogrammet initialiseras vid power-up och sedan skulle exekveringen av det startas på signal från huvuddatorn. Så länge huvuddatorn inte gav signal skulle man ligga i monitorn på helt vanligt sätt.



I sin nuvarande version har grundmodulen (SI-80 Rev.2.3) följande kommandon:

Break: Sätta brytpunkt (med repetition count)
Compare: Jämföra två minnesareor med varandra
Display: Displaya och ändra i minnet
Fill: Fylla en minnesarea med ett bytevärde
Proceed: Fortsätta från programräknarens aktuella värde
Input: Input från en dataport
IFlop: Displaya och ändra CPU:ns interrupt-enable flip-flop
Jump: Hoppa till en adress i minnet och exekvera därifrån
Move: Kopiera en minnesarea till ett annat ställe
Trace: Single-step av instruktioner (kräver CTC)
Output: Output till en dataport
Reg: Displaya och ändra CPU-register
Set: Skriva i konsekutiva minneslokationer

Övriga kommandon skrivs av användaren och placeras i 'extraPROMmet'.

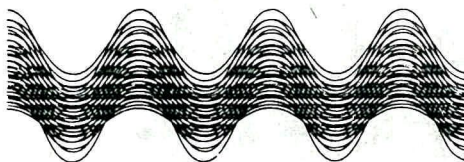
OBS att inga I/O-kommandon till massminne finns i grundmodulen, eftersom dessa är väldigt beroende av vilken systemkonfiguration man har.

De krav som gäller för den här monitorn är följande:

1. Den skall ligga på adress 0 - 800H (0-2K).
2. Om man vill ha ett expansionsPROM skall det finnas plats för detta på adress 800H-0FFFH (2-4K).
3. Det skall finnas RAM från adress 0FC00H-0FFFFH (64-65K).
4. I grundversionen behövs en INTEL 8251 USART på I/O adress 0DEH (data) och 0DFH (control). Detta kan dock ändras med hjälp av extraPROMmet, utan att ändra grundmodulen. (Addresserna har valts för att passa till Jan Gauffins kortserie för S-800-bussen)
5. Om man vill ha single step och multibrytpunkter krävs en CTC på I/O adress 0DBH-0DBH.

Funktionen med 'extraPROM' fungerar så här:

Vid power-up är det första som händer att vissa systemkonstanter som ligger permanent i PROM flyttas upp i monitorns RAMarea. Dessa är av typen vilka ASCII-tecken som används som RUBOUT och LINE DELETE, hur många ASCII 'NULL' som skickas ut efter carriage return o.s.v. Dessutom läggs ett antal hoppinstruktioner upp i RAM, bland annat ett hopp till en felrutin (dit man kommer vid olika typer av inmatningsfel o.d.), ett hopp till den rutin som skriver ett tecken på terminalen (TTYWRO), och ett hopp till den rutin som tar in ett tecken från terminalen (TTYINO). Dessa rutiner ligger i PROM, men eftersom hoppen till dem ligger i RAM kan man ändra i RAM:et och på så sätt få andra I/O-rutiner för terminalen.



Wer viel arbeitet, macht viele Fehler.
Wer nicht arbeitet, macht keine Fehler.
Wer keine Fehler macht, wird befördert und prämiert.

Då systemkonstanterna har flyttats till RAM skickar monitorn ut initialiseringsdata till de portar som den tills vidare antar finns i systemet, nämligen en USART på adress 0DEH-0DFH och en CTC på 0DBH-0DBH.

Då initialiseringen är klar tittar monitorn efter om det i adress 800H (=början av expansionsPROMmet) finns en ASCII-sträng 'PROM'. Detta skall ett expansionsPROM nämligen innehålla. Om så är fallet sätts en bit i en flaggbyte så att monitorn kommer ihåg om det fanns ett extra PROM eller inte. Om det fanns exekverar monitorn en CALL-instruktion till adress 804H. ExtraPROMmet skall där innehålla ett hopp till en subrutin som utför de initialiseringar av portar m.m som behövs. Subrutinen avslutas med en RETURN-instruktion så att man kommer ner i grundmodulen igen.

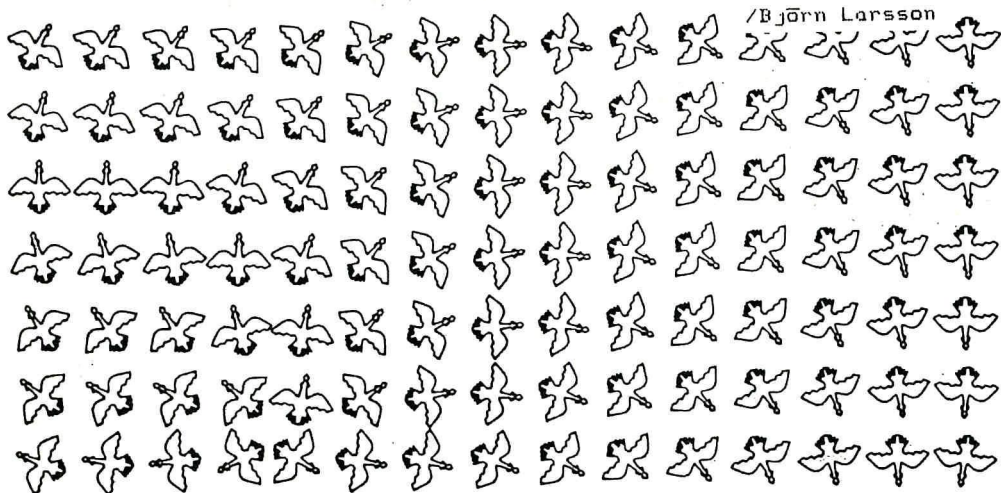
ExtraPROMmets initialisering kan innefatta en omändring av de tidigare uppflyttade hoppinstruktionerna till terminalrutinerna. På så vis kan man låta sina egna terminolrutiner ligga var som helst.

Väl tillbaka i grundmodulen anropas en rutin som tar in en sträng från terminalen. Då strängen knappats in (avslutas med carriage return) gör programmet, om ett expansionsPROM fanns, ett subrutinanrop till adress 807H. Där skall finnas en CLI (Command Line Interpreter) som kollar om det inmatade kommandot var ett av de som fanns i extraPROMmet. I så fall utförs kommandot. Annars exekveras en RETURN-instruktion så att man kommer ner i grundPROMmet igen, och där letar man och ser om det var något av debuggerns standardkommandon. I så fall utförs det. Annars får man ett felmeddelande.

På det här sättet är det enkelt att införa nya kommandon. Extraprommet kan, förutom att ge nya kommandon, även användas för sätta nya namn på standardkommandona. Detta sker då genom att extraPROMmet redigerar om den inmatade textsträngen.

I övrigt ingår en buffrad TTY-driver med line- och character-delemöjligheter. Listningar på skärmen kan stoppas och brytas, och längden på raden i minnesdisplaykommandot är ställbar mellan 15 och 133 tecken per rad.

En assemblerlista finns i STACK-lokalen.



INOMETER AB

MICROPROCESSOR APPLICATIONS

```
LXI SP, STACK POINTER
IF INTERESTED
THEN
  DO;
    CALL TELEPHONE(7528131)
    IF STILL INTERESTED THEN
      START TOMORROW;
  END;
ELSE
  DO;
    HALT;
    EI;
  END;
```

Vi söker omgående en yngre civilingenjör som vill arbeta med mikrodatorer.

Du bör ha erfarenhet av assemblerspråk och något högnivåspråk samt helst byggt något med mikrodatorer.

Dina arbetsuppgifter kommer att omfatta både maskin- och programvara.

Du får stor plats för egna idéer.

Vi är ett litet företag som utvecklar kundanpassad elektronik till industrin. Vi använder processorer som 8085, 8080, 8022, 8748, 8086 och Intel's utvecklingssystem.

Dina ansökningshandlingar sänder Du till adress enligt nedan. Vill Du veta mera så ring gärna civ ing Jan-Peter Ocklind eller civ ing Bo Andersson tel 08-7528131.



HUVUDBRY

TANKEN PÅ MOTION

Det finns en förbryllande historia som har att göra med valutor. Två grannländer, låt oss kalla dem S och N, hade kommit överens om att en S-krona skulle vara värd en N-krona och tvärt om.

Men så en dag bestämde S-regeringen att en N-krona bara skulle vara värd 90 öre i S. Som hämnd svarade N-regeringen med att i N sänka S-kronans värde till 90 öre.

Nu bodde ett ljushuvud i en stad som låg mitt på gränsen mellan de båda länderna. Han gick in i en affär på S-sidan, köpte ett 10-öres motstånd och betalade med en S-krona. Han fick tillbaka en N-krona, som ju representerade ett värde av 90 öre. Därefter gick han över gränsen till en affär på N-sidan, köpte ett 10-öres motstånd och betalade med en N-krona. Här fick han tillbaka en S-krona som i N var värd 90 öre.

När mannen kom hem igen hade han fortfarande sin krona och dessutom två motstånd. Affärsmännen på båda sidor om gränsen hade fått sina 10 öre för motstånden. Vem hade betalt dem?

Att knäcka ovanhängande nöt med huvudet klarar vem som helst. Men att skriva ett datorprogram

Elegantaste programmet belönas med en motståndssats innehållande 2 motstånd.

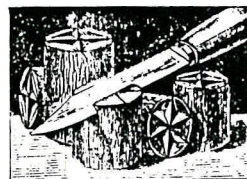
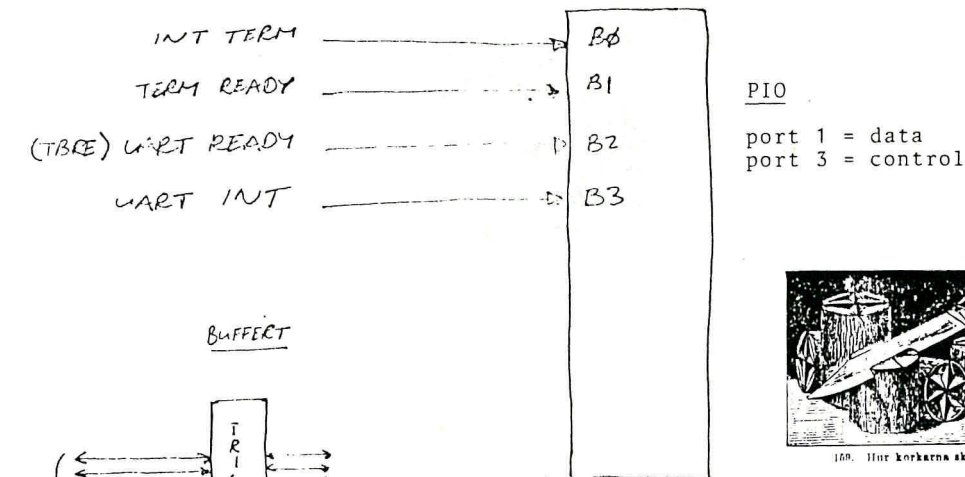
ROS

Operativsystemet ROS som beskrevs i första numret av Stackpointer har crunchats av BA. En någorlunda kommenterad sourcelistning skall finnas i Stacken-lokalen.

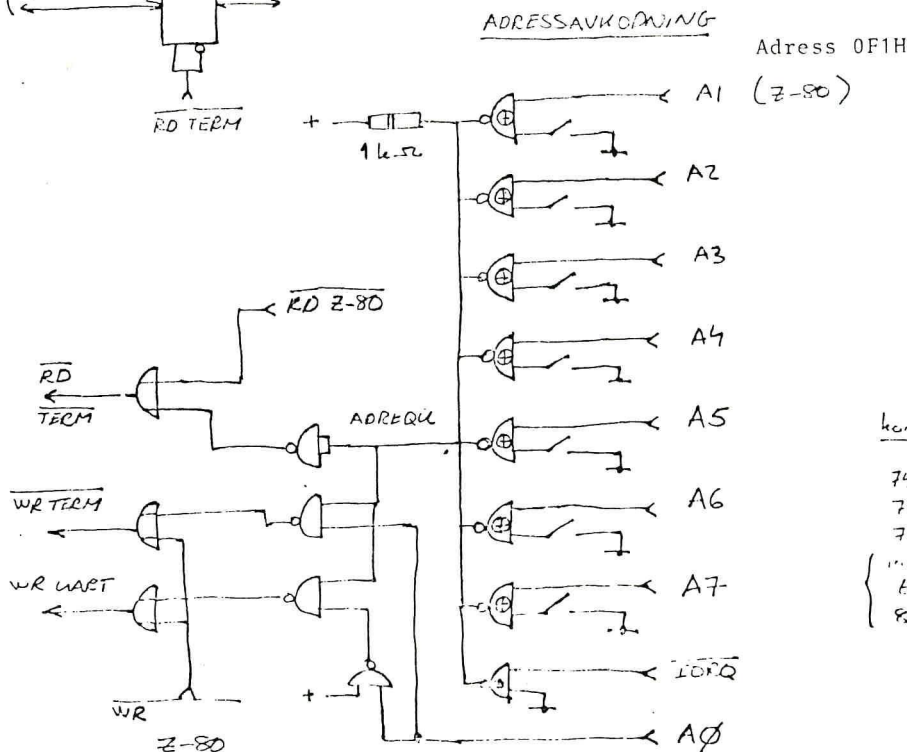
Sven Håkansson på CD i Göteborg har assisterat och lagt in ett HELP-kommando som ger entrypoints till alla de 48 förträffliga färdiga subrutinerna. ROS:en är nu körd med både parallell- och serie-I/O mot CD-terminalen och bikiupe-skärmen.

OBS! STANDARD FÖR INKOPPLING AV CD-BILDSKÄRMEN
TILL VÅRA 800-SYSTEM.

Fördelar: Ger 3 helt lediga PIO-portar på PIO-kortet.
Vid pollad mod ligger statusbitarna på samma sätt som i en 8251 USART, dvs det är lätt att modifiera bef. mjukvara för seriekommunikation.



169. Hux kerkarna skåras.



komponenter:

74LS 216
74LS 00
74LS 32

{ no transys
 tristate
 & bidar

FLOPPY

Chalmers Datorförening, CD, har nu (!) fått de första leveranserna från Siemens av floppydisk-drivar. På begäran har man nu lyckats få option på att få köpa ytterligare ett antal till samma pris, 2150:- inkl moms. Om Du är intresserad, skicka 200:- till Pg 27 01 15 - 9 "Chalmers Datorförening" och ange floppyköp på talongen. Senast den 25 oktober måste pengarna vara inne. Not: Siemens-drivarna är helt Shugart-kompatibla.



**LET ME PROVE HOW YOU CAN
EARN BIG MONEY**

When You Enroll as a Graduate
CORE STRINGER

Core memories are almost obsolete because they can't be assembled cheaply enough to compete with solid-state devices. But now you can earn lush profits stringing cores in your own home. I teach you by simple hypnotism and let you stay under the trance. Send \$1.

CORE CLINIC, Box Q, Chattanooga, Tenn.

MEN - DO YOU HAVE WHAT IT TAKES TO BE A

Chad Collector?

The chad from thousands of Teletype machines across the Nation must be collected daily or Disaster may strike as a result of overflowing plastic chad boxes, stopping punch mechanisms, choking industry, causing a new Depression. You can let America down or be a Man and send us your \$ now!

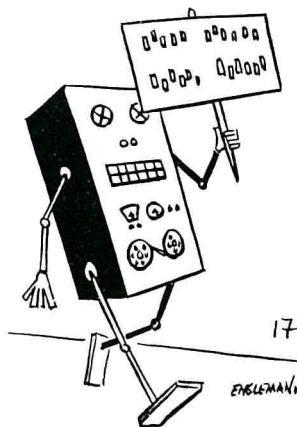
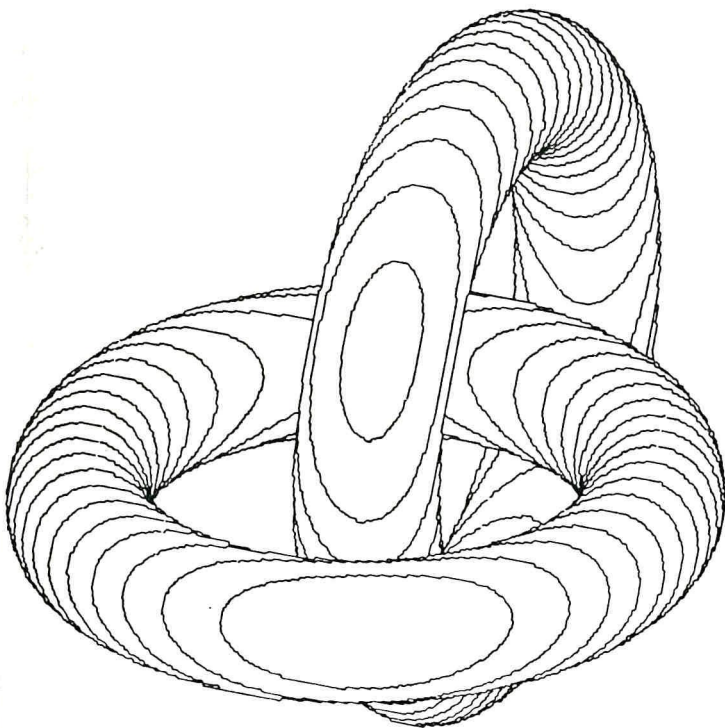
TRAIN ALONE WHILE OTHERS HAVE FUN

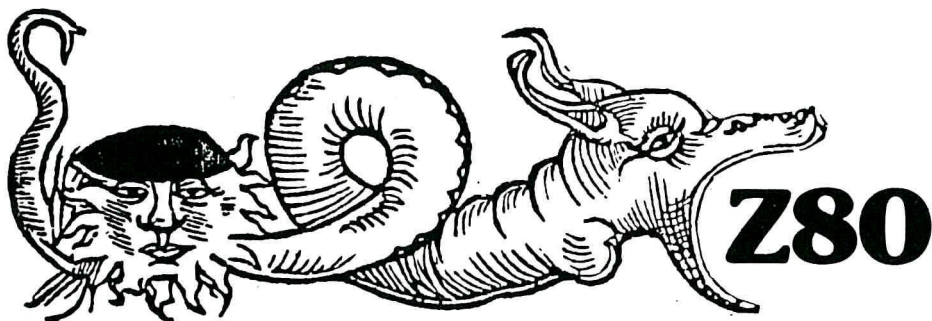
Dreary lessons, back breaking practice, high entrance fee can start now at no risk! Send photo and blank check.

RUSH ME MY FREE BOOK OF MATCHES!

Chad-Vac, Inc., Dept. X, Chicago, Ill.







Z80-SYSTEM

För dig som inte hann komma med i förra Z80-samköpet.
Jan Gauffin i Stacken har gjort i ordning en ny kortserie enligt
följande:

CPU och EPROM kort	150 kr inkl. moms
Z80 CPU samt 2 st EPROM 2758, 2716, 2732 eller 2764. Disable/enable EPROM. Reset CPU. Enable EPROM för NMI på adr. FF.	

USART kort	160 kr
2 st 8251A, 1 st Z80 CTC, serieklockgenerator, V24 buffrar samt spänningsomvandlare för +- 12V.	

PIO kort	140 kr
2 st Z80 PIO	

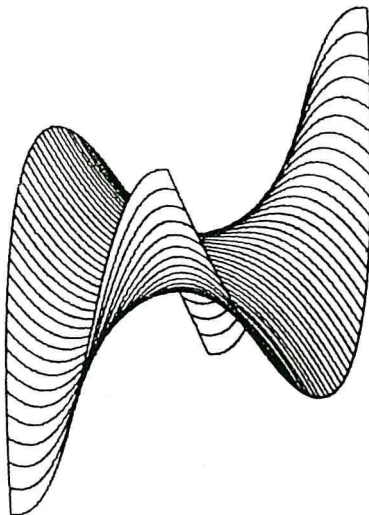
Buss kort	135 kr
Plats för 9 st kontakter, bussavslutning. Kontaktindelning 20 mm. Jordplan på ena sidan. 48-poliga kontakter.	

Floppy-kontroll kort	under utv.
NEC D765C, Z80 CTC.	

RAM kort	160 kr
32k RAM m. card-select på adr. FE.	

SI-80 2k monitor 1 EPROM	dagspris
EPROM 2716	420 kr
Krystaller: 4 MHz	25 kr
16 "	20 kr
19,6608 MHz	20 kr

Tag kontakt med JAN GAUFFIN.

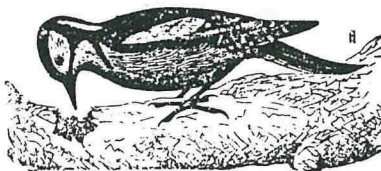


SÄLJES

Bildskärmsterminal HAZELTINE 1200.
24 rader a 80 tecken, TTY-kompatibel,
V24-interface, 110-9600 Baud.

Pris endast 2.000:-

Ring Leffe 38 39 64
eller Erik Forsberg 86 17 35



1 st Facit-terminal typ 6401	1000:-
SP1 parallellsnitt	
tangentbord med å ä ö	
512 byte kärnminne	
12x80 tecken	

Diverse kort till 4680-system

1 st CPU-kort Z-80	1000:-
1 st styrkort till ovanstående	1000:-
2 st 16 kbyte RAM	2000:-
2 st SP1 IN	å 500:-
2 st SP1 UT (virade)	å 300:-
1 st UART (virat)	300:-
1 st kortlåda (19") med virad bus	1000:-
13 kortpositioner för 4680	
1 st Prototyp till Facit datakasettbandsp.	400:-

Göran Boberg

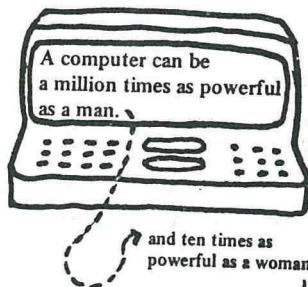
Tel. arb. 08-787 75 65

bost. 08- 82 89 22 (e. 19.00)

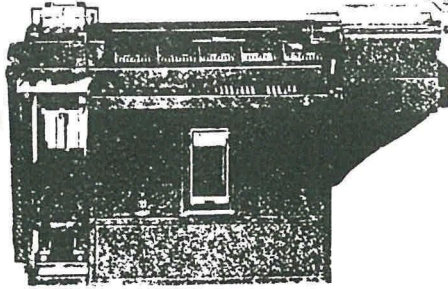
INTRESSE

finns för inhandling av äldre dator-
tidningar som Byte, Interface Age,
Creative Computing m.m.

hans nordström
arb 0760-465 80



Heaven In The Office



Ah, the ecstasy, wonder, elation and pure joy of work now that we've installed a Computermod.

"Amazing," is the word for what this little convenient joy-bringer does for the company and us.

"Marvelous," is the word for how we feel now each and every day.

Before our smart boss had Computermod installed our lot was just more boring office work. Now each day is a fantastic orgy of fun and adventure, a marvelous, breath taking experience of elated happiness. Besides which, our dinky little company has grown to be one of the leaders in the field.

We owe it all to Computermod — our success, our good times, our fame, fortune and the total joy of complete and utter career fulfillment.

If your work days aren't fun days, if "everyday" isn't better than the "office party," its time to progress with Computermod.

COMPUTERMOD

The Joy-Bringer