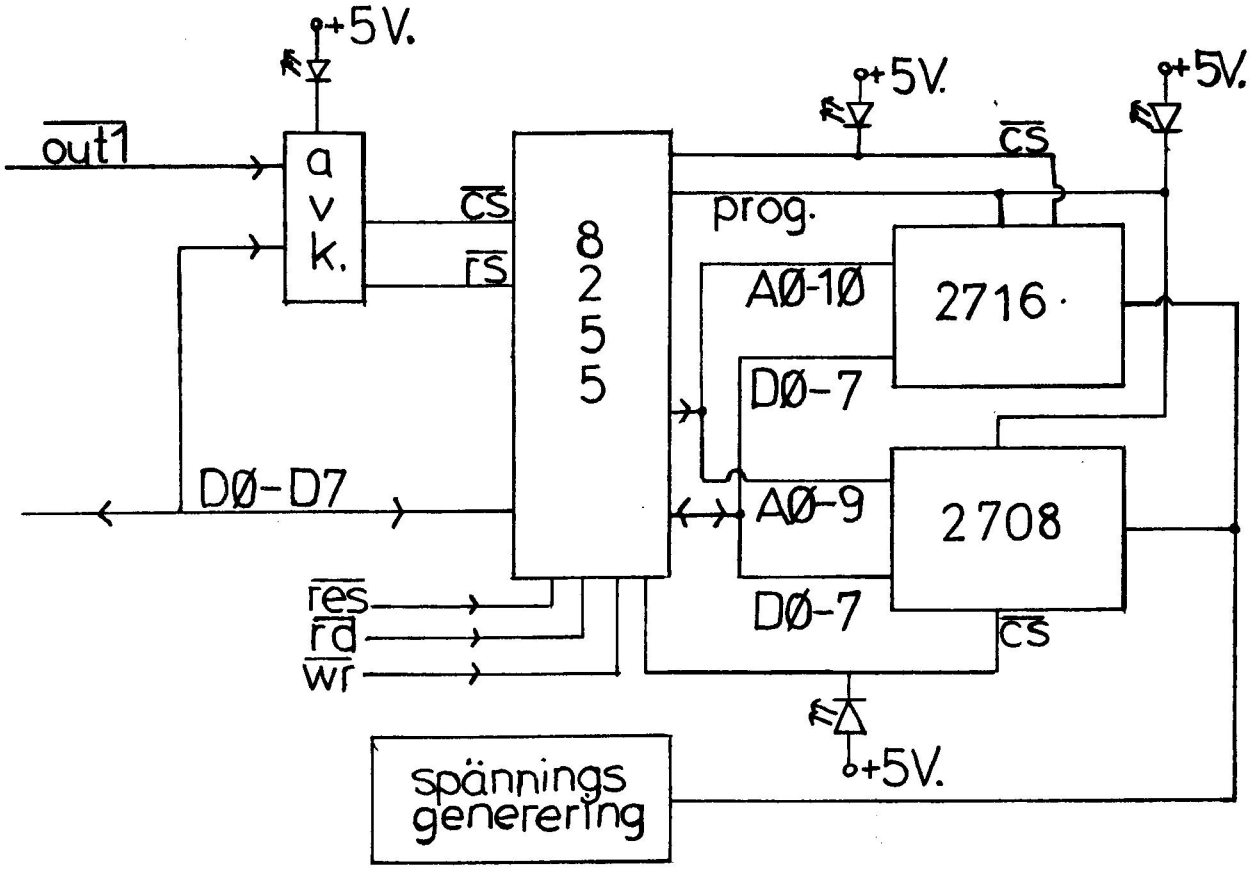
ABC PROG

SCANDIA METRIC AB

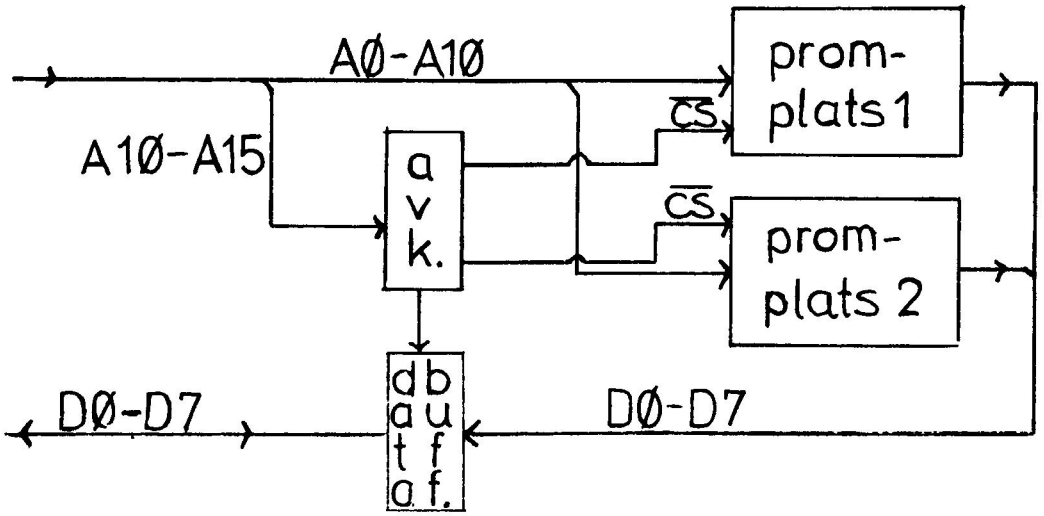
BANVAKTSVÄGEN 20, BOX 1307, 171 25 SOLNA, TEL 08/82 04 00

DANMARK: TEL 02/80 42 00 NORGE: TEL 02/28 26 24 FINLAND: TEL 90/42 39 11

ABC-80 PROMPROGRAMMERARE



Blockschema programmeringsdel



Blockschema minnesexpansion

ABC PROM/PROG

- ALLMÄNT
- PROMNING AV BASIC-PROGRAM, ALLMÄNT
- PROMNING AV BASIC-PROGRAM MED ABC-PROM/PROG
- BESKRIVNING AV PROGRAMVARAN
- AUTOMATISK UPPSTART
- BESKRIVNING AV LYSDIODER OCH SPÄNNINGSBYGLINGAR
- BYGLING AV KORTVALSADRESSER
- ADRESSBYGLING
- BILAGOR

  EXEMPEL PÅ I/O-RUTINER
  EXEMPEL PÅ AUTOSTARTRUTIN
  ABC 80:S MINNESKARTA

Promning av BASIC-program, allmänt

Det finns två olika metoder att promma ett BASIC-program. Antingen kan man välja ASCII-formatet (.BAS) eller också det kompilerade formatet (.BAC).

Av dessa två metoder är ASCII-formatet det enklaste och därför även det mest använda (när det gäller BASIC-program). I denna beskrivning, som medföljer ABC-PROM/PROG, har vi därför valt att använda ASCII-formatet.

ASCII-formatet

Kortfattat kan man beskriva förloppet att promma ett BASIC-program i ovanstående format som

- Man skriver en speciell I/O-rutin, som kan läsa in programmet från PROM till RAM-minnet (där exekvering sker).

- Genom en speciell initieringsrutin läggs sedan denna I/O-rutin in i ABC 80:s tabell över vilka enheter som finns tillgängliga i systemet (den s.k. device-listan). Vanligt är att man använder sig av namnet ROM:.

- När man sedan vill hämta in/köra programmet är det bara att göra "LOAD ROM:" eller "RUN ROM:".

I det exempel på I/O-rutin som finns i bilaga 1 är enhetsnamnet ROM valt.

Kompilerad form

När man prommat ett BASIC-program i BAC-format i stället för i BAS-format vinner man det att programmet inte behöver flyttas över till RAM-minnet för att köras. Men det vållar en del problem. I BAC-formatet finns det pekare till bl.a. variabler. Detta resulterar i att programmet måste vara rätt kompilerat innan det prommas. När man väl "laddat in" och kört det så går det heller inte att direkt göra "NEW" och hämta in ett program från t.ex. kassett och köra det, utan man måste först "nollställa" ABC 80:n (RESET-knappen).

Promning

När vi skall promma ett program i BAC-format måste det först kompileras på rätt adress.

Vi tittar på ett exempel:

Vi har ett BASIC-program som i det här fallet är mindre än två kB. Vi bestämmer oss för att lägga det på adress A800-AFFF (43008-45057).

Sätt in 8K extra minne. Ändra BOFA så att den pekar på A800. POKE 65052, 43008, SWAP%(43008)
NEW

Hämta in programmet
LOAD <programnamn>

Nu måste vi se till att inga variabler lagras i PROM-arean. Höj därför EOFA och HEAP till början på det vanliga RAM-minnet.
POKE 65054, 49152, SWAP%(49152), 49152, SWAP%(49152).

Sedan skall programmet kompileras
RUN

När programmet startat, avbryt med CTRL-C.

Nu höjer vi golvet för att inte skada den kompilerade koden.
POKE 65052, 49152, SWAP%(49152)
NEW

Nu är koden på adress A800-AFFF färdig att prommas.

Start

Även här kan vi använda en "I/O"-rutin för att starta upp programmet.

Det som utförts i "I/O"-rutinen är följande:

Vid OPEN ställs BOFA om så att den pekar på början av PROMMET. Vid INPUT sätts EOFA och HEAP så att de pekar på början av RAM-minnet. Därefter ges Filslut-status till BASIC. Som tidigare nämnts i texten innehåller denna dokumentation endast "support" för promning i ASCII-format. För intresserade har vi dock valt att ge ett exempel på hur en I/O-rutin kan se ut. Rutinen finns på disketten under programnamnet BAC.TXT och BAC.ITH. Listning finns i Bilaga 3.

Promning av BASIC-program med ABC-PROM/PROG

All nödvändig programvara, inte bara för att "promma" ett program, utan även för att ladda in det i ABC 80:s RAM-minne och köra det, medföljer på diskett. Här finns även ett exempel (EXEMPEL.BAS) på ett litet program som ritar grafik på ABC 80:s bildskärm.

Proceduren för att "skjuta" ett BASIC-program med ABC 80:s prom-programmerare kan indelas i följande fyra steg:

1. Skriv och prova ut ett BASIC-program på vanligt sätt. När det fungerar på önskat sätt, spar det som vanligt med:

   SAVE <programnamn>

   Tag sedan bort alla REM-satser (för att spara plats), omnumrera av samma orsak (REN1) och spara programmet med:

   LIST <programnamn>

   Du har nu "samma" program på disketten under två olika namn <programnamn>. BAS och <programnamn>.BAC. Den senare, som även innehåller Dina kommentarer (REM-satser) är Din back-up. Den i .BAS-form, dvs ASCII-format är den Du skall lägga i prom.

2. Nästa steg i proceduren är ett kontrollera vilket minesbehov, som kommer att krävas för programmet. För att få redan på minnesbehovet kör Du därför programmet KONTROLL.

   RUN KONTROLL

   Detta program frågar efter "FILNAMN" och Du svarar med <programnamn>. Du behöver inte ange prefixet .BAS, eftersom programmet förutsätter detta. (Men det gör inget!). Finns inget program med det angivna namnet och prefixet .BAS på någon av disketterna, får Du en ny fråga.

   Kom nu ihåg att:

   2708 rymmer 1024 bytes
   2716 rymmer 2048 bytes
   2758 rymmer 1024 bytes

   (en 2758 är i praktiken endast en 2716, som inte klarat testerna utan låsts till 1k byte).

   Programmet frågar slutligen om Du vill gå direkt till programmering. Om Du svarar JA (ja, j, J räcker också) länkas programmet BASPROM in automatiskt.

3. Nästa steg i kedjan är programmeringen. Om Du inte går direkt från steg 2, startar Du upp med

RUN BASPROM

Programmet frågar nu efter

EPROMTYP

Innan Du besvarar denna fråga bör Du montera Ditt prom på prom-programmeraren. Du måste även försäkra Dig om att Du sätter i Ditt prom på rätt plats, eftersom 2708 och 2716 (2758) har olika spänningar. Felaktig placering "dödar" effektivt Ditt prom.

Du skall även se till att Du vänder kapseln åt rätt håll. Felvändning är annars ett annat effektivt sätt att förstöra prom på. Den markering "halva hål", som finns på prommet, skall vara vänt åt samma håll som sockelns hävarm.

Efter alla dessa "förmaningar" är det nu dags att besvara frågan om EPROM-typ

svara med vald promtyp 2708, 2716 eller 2758

Programmet frågar därefter efter FILNAMN ?

svara med önskat <programnamn>
(även här kan .BAS utelämnas)

Innan själva programmeringen startar, tar programmet först bort alla onödiga mellanslag i filen <programnamn>. När sedan programmeringen startar tänds den lysdiod märkt "programmering" på prom-programmeraren. Programmeringen brukar ta ett par minuter.

Skulle programmet inte rymmas i ett prom, kommer ett meddelande upp när det är dags att byta prom. Sätt då i ett nytt prom (obs måste vara samma typ!) och "ge klartecken" så fortsätter programmeringen.

Slutligen kommer meddelandet "Programmeringen klar" upp på skärmen.

Kontroll sker automatiskt om EPROM:met är raderat. Om så ej är fallet fås meddelandet:

"EPROM EJ RADERAD.
SKALL PROGRAMMERING SKE ÄNDÅ ?"

(Se rubrik "JÄMFÖRA PROM MED MINNE ")

4. Det är nu dags att provköra det prommade programmet. Öppna luckorna till disketterna och slå av spänningen på ABC 80 och flexskivenheten.
Flytta det färdiga prommet till en ledig promplats (vänd kapseln rätt!) och bygla adressen till t.ex. 4000H. (H:et står här för hexadecimal representation, dvs tal med basen 16 i stället för vårt vanliga decimala system, som har basen 10. 4000H motsvarar den decimala adressen 16384 och är alltså den plats som ligger direkt efter ABC 80:s 16k-BASIC i minneskartan).

Slå på spänningen igen. Stäng luckorna på diskettstationen.

Skriv sedan

RUN PROMLINK

Detta program lägger nu in enheten ROM: i ABC 80:s device-lista.

Skriv nu RUN ROM:

varvid det prommade programmet läses in i RAM-minnet och programmet startar.

Beskrivning av programvaran

Den programvara som medföljer ABC-PROM/PROG består f.n. av nedanstående program. Observera att i vissa fall är programmet delat i två delar, dels ett program på bara några rader (flyttar pekare etc) och dels själva programmet. Normalt startar man upp det "lilla" programmet, som sedan automatiskt länkar in nästa program. Här beskrivs endast de egentliga programmen, dvs de man normalt anropar.

1. PROG.

Programmet innehåller följande:

- LADDA INTELHEXFIL:

  Laddar en intelhexfil gjord med ASSEMBLER2 till den adress som anges av "STARTADRESS MINNE" på skärmen. Denna laddare förutsätter att maskinkodsfilen är kontinuerlig, dvs att källfilen ej innehåller odefinierade minnesareor.

  (källfilen får ej innehålla "DEFS", eller flera "ORG").

  Sådana areor måste definieras med NOP el.liknande för att inladdningen skall bli korrekt.

  Om filen är större än vad som ryms i ett prom, erhålls "BLOCK X INLÄST", varefter programmering av första prommet kan ske. En ny tryckning på "1" läser sedan in nästa block. (Glöm inte att byta prom).

- KOPIERA PROM TILL MINNE

  För över informationen från programmet till ABC 80:s RAM-minne. Användbart t.ex. när man vill kopiera ett prom.

- JÄMFÖRA PROM MED MINNE

  Efter programmering av ett prom finns här möjlighet att kontrollera att programmet "fastnat" ordentligt. Ibland om man lyckas få tag på lite mindre bra prom behövs programmeringen göras ytterligare en gång. Om ej full överensstämmelse gäller listas de avvikande positionerna på ABC 80:s bildskärm med utseendet

  MINNE(POS) = VÄRDE PROM(POS) = VÄRDE

  (M står här för minnesinnehållet, P för prommets motsvarande). Observera att omprogrammeringsförsök endast lönar sig om Prommet har högre värde i resp bit.

Tag som exempel följande:

M(67) = 15 P(67) = 143

binärt set det då ut på följande sätt:

M		0	0	0	0	1	1	1	1	
P		1	0	0	0	1	1	1	1	
	128	64	32	16	8	4	2	1	(bit-värden)

M = Minnesinnehåll P = motsvarande prommhåll

D.v.s. ett nytt försök är gångbart !

Om resultatet av jämförelsen i stället hade varit

M(67) = 15 P(67) = 142

är det lönlöst med ett nytt försök, eftersom den lägsta biten (2^0) är satt till noll i prommet och full jämförelse aldrig kommer att bli möjlig. (Prommet måste raderas först).

- KONTROLLERA ATT PROM ÄR RADERAD

  Kontrollerar att samtliga promceller har värdet OFFH, dvs alla bitar = 1. Om inte, erhålls en listning på skärmen.

- ÄNDRA PARAMETRAR

  Här sker val av promtyp och bufferadress. Om adressangivelse avslutas med ett "H" tolkas den hexadecimalt.
  Vid promning av 2716, 2758 kan startadressen i prommet ändras för att möjliggöra att endast en del av prommet programmeras. M.a.o. man kan fortsätta programmera i samma prom. En 2708 programmeras alltid från början till slut.

- PROGRAMMERA

  Programmerar prom.

- AVSLUTA

  Avslutar programmet.

2. KONTROLL

Programmet kontrollerar minnesbehovet av vald fil i ASCII-format. Programmet möjliggör även direkthopp till programmet BASPROM.

3. BASPROM

Program för programmering/promning av BASIC-program i ASCII-format. Programmet plockar själv bort alla onödiga mellanslag.

4. PROMLINK

I/O-rutin som lägger upp enheten ROM: i ABC 80:s tabell över tillgängliga enheter (device-listan). Programmet möjliggör laddning och exekvering av prommat BASIC-program genom kommandon LOAD ROM: resp RUN ROM:.

5. EXEMPEL.BAS

Ett litet BASIC-program som ritar grafik på ABC 80:s bildskärm. Programmet får gott och väl plats i en 2708:a och används i denna dokumentation som exempel tillsammans med AUTSTART.

6. AUTSTART

En färdig maskinspråksrutin (intelhex-format, genererad av ABC 80:s assembler) som kan prommas och användas för automatisk uppstart av prommade program i ABC 80.

Automatisk uppstart

Det finns möjlighet att låta ett prommat BASIC-program starta direkt när spänningen slås på ABC 80. Eller naturligtvis att programmet återstartar efter ett kortare eller längre spänningsbortfall. En s.k. "watch-dog", mycket användbar och i vissa fall nödvändig t.ex. i mätinsamlingssystem.

Förutsättningen för att det skall fungera är dock att INTE en flexskiveutrustning finns ansluten till ABC 80.

Orsaken är att man utnyttjar den normala rutinen för start av flexskiveenheten som finns i ABC 80. Den fungerar på så sätt att ABC 80 vid RESET eller spänningstillslag kontrollerar om det finns en hopp-instruktion på adress 604BH. Finns instruktionen där så gör ABC 80 ett subrutinanrop dit. Om en lämplig initieringsrutin läggs efter detta hopp, kan autostart erhållas.

Ex:

```
            ORG   604BH
604B        JP    START       ; Här tittar BASIC
604E START:                   ; Här börjar initieringen
```

Ett färdigt program för autostart finns på disketten. För att kunna promma detta görs följande:

- Tryck RESET
- Skriv RUN PROG
- Tryck på "1" (= ladda intelhexfil)

- Därefter kommer frågar om FILNAMN ?

  Svara med AUTSTART

  Programmet läser nu in maskinkodsfilen AUTSTART samt beräknar erforderligt minnesutrymme. I detta fall 203 bytes (= hex.CB).

- Tryck nu på "5" för att ändra parametrar. Skriv in "2708" på promtyp och svara med enbart <RETURN> på de övriga.

- Sätt en raderad 2708 på dess avsedda programmeringsplats.

- Nu kan programmering ske genom att trycka på "6" och svara "J" (Ja, ja, j) på frågan om programmering skall ske.
  Lysdioden bredvis promplatserna skall tändas och indikerar att 25 V programeringsspänning är tillslagen.

När programmering är klar, öppnas luckorna till disketterna och spänningen slås av på ABC 80. Det färdiga prommet flyttas till en av promplatserna (vänd kapseln rätt!) som byglas till adressen 6000H.

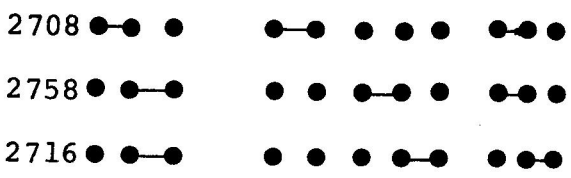
För att sedan prova funktionen hos autostart-prommen kan programmet EXEMPEL.BAS läggas i ett prom. (Se promning av BASIC-program). Detta prom placeras sedan på den lediga prom-platsen. Adressen till denna byglas med adress mellan 4000H-5FFFH (16384-24575) eller 6400H-7BFFH (25600-31743). Enklast är kanske att lägga den på adressen 6400H.

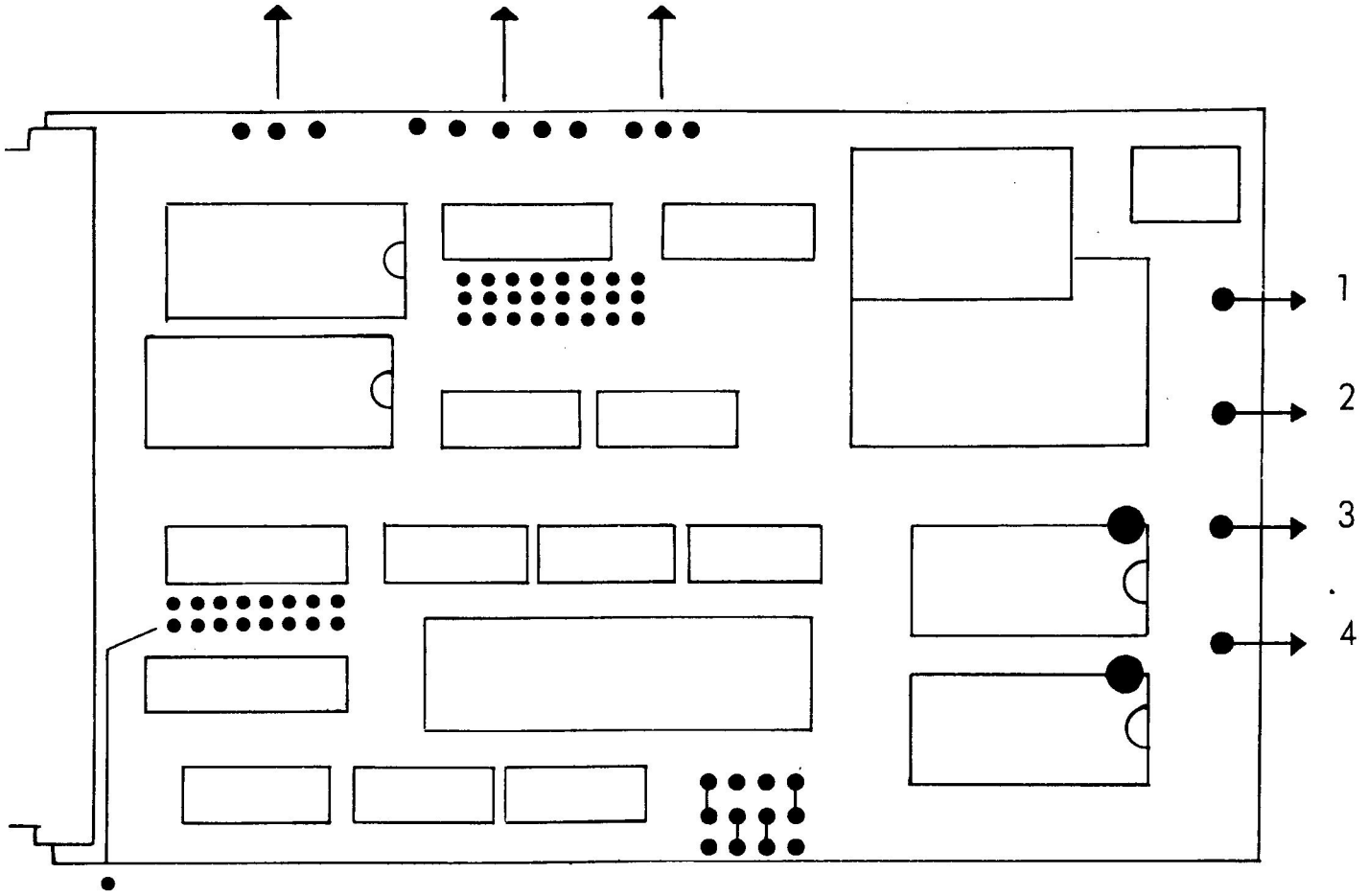
Vid spänningstillslag eller vid RESET kommer AUTOSTART-prommet automatiskt att leta upp "texten", ladda in den och starta programmet. OBS! att flexskivenheten INTE får vara ansluten vid detta prov. Buss-kontakten behöver dock ej tas ur, det räcker med att flexskivenhetens spänning bryts (efter att luckorna har öppnats !).

Ett exempel på I/O-rutin för autostart finns i bilaga 2.

Beskrivning av lysdioder och spänningsbyglingar.

Spänningsbygling (gäller bägge promplatserna).





LYSDIODER

1. Indikerar att programmering pågår

2. Indikerar att kortet är valt (card select)

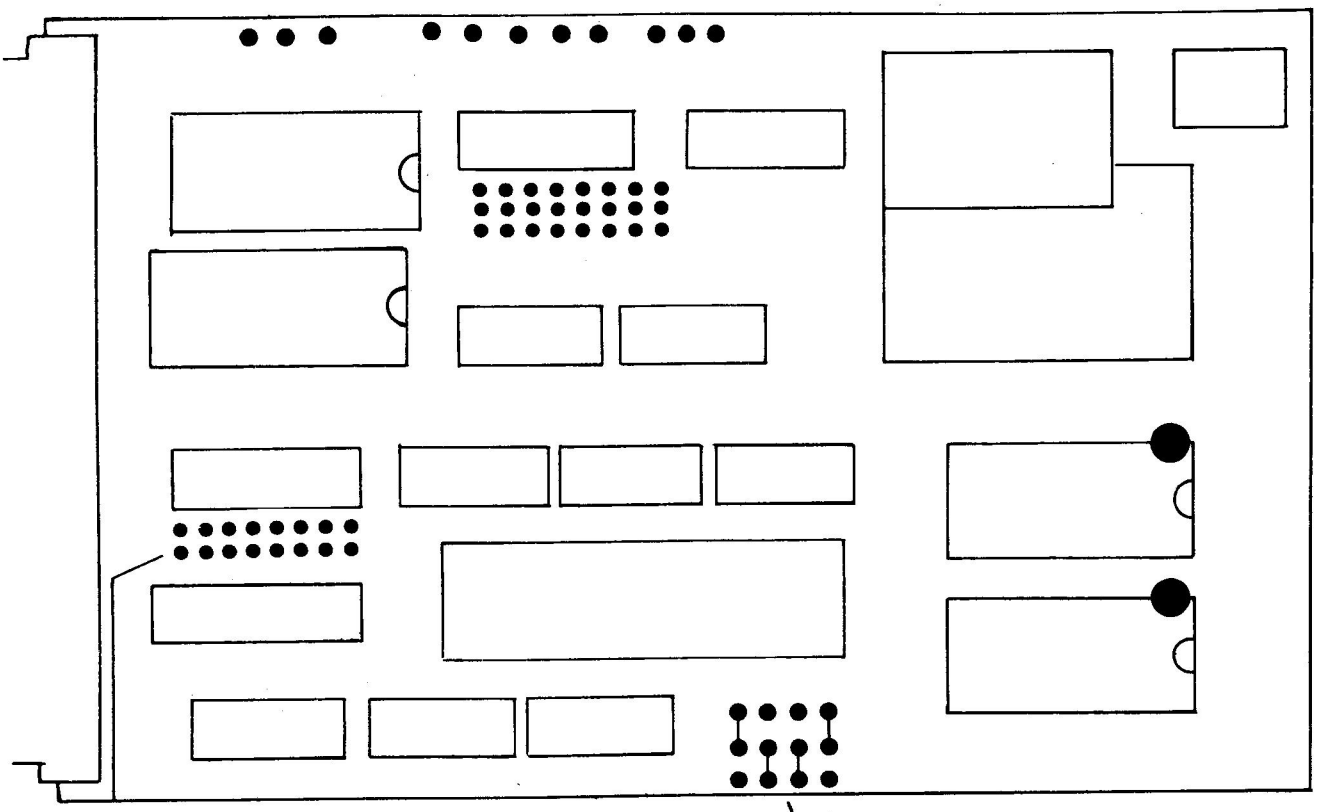
3. Programmeringen avser 2708. Tänds vid spänningstillslag.

4. Som punkt 3, men avser 2716/2758.

OBS ! Montera prommen åt rätt håll, dvs alla "hack" mot lysdioderna.

Beskrivning av kortvalsadresser

Programvaran som medföljer promprogrammeraren förutsätter att kortet är byglat för adresserna 24-27. Dessa är fast inlagda på kortet, men det kan kanske ändå vara intressant att veta hur byglingen är gjord.



Välj kortadress: | 0 | 0 | A | B | C | D | X | X |

Används ej

Fasta byglingar på kort

Stift rad som ger logisk nolla

stift att ansluta

Stiftrad som ger logisk etta

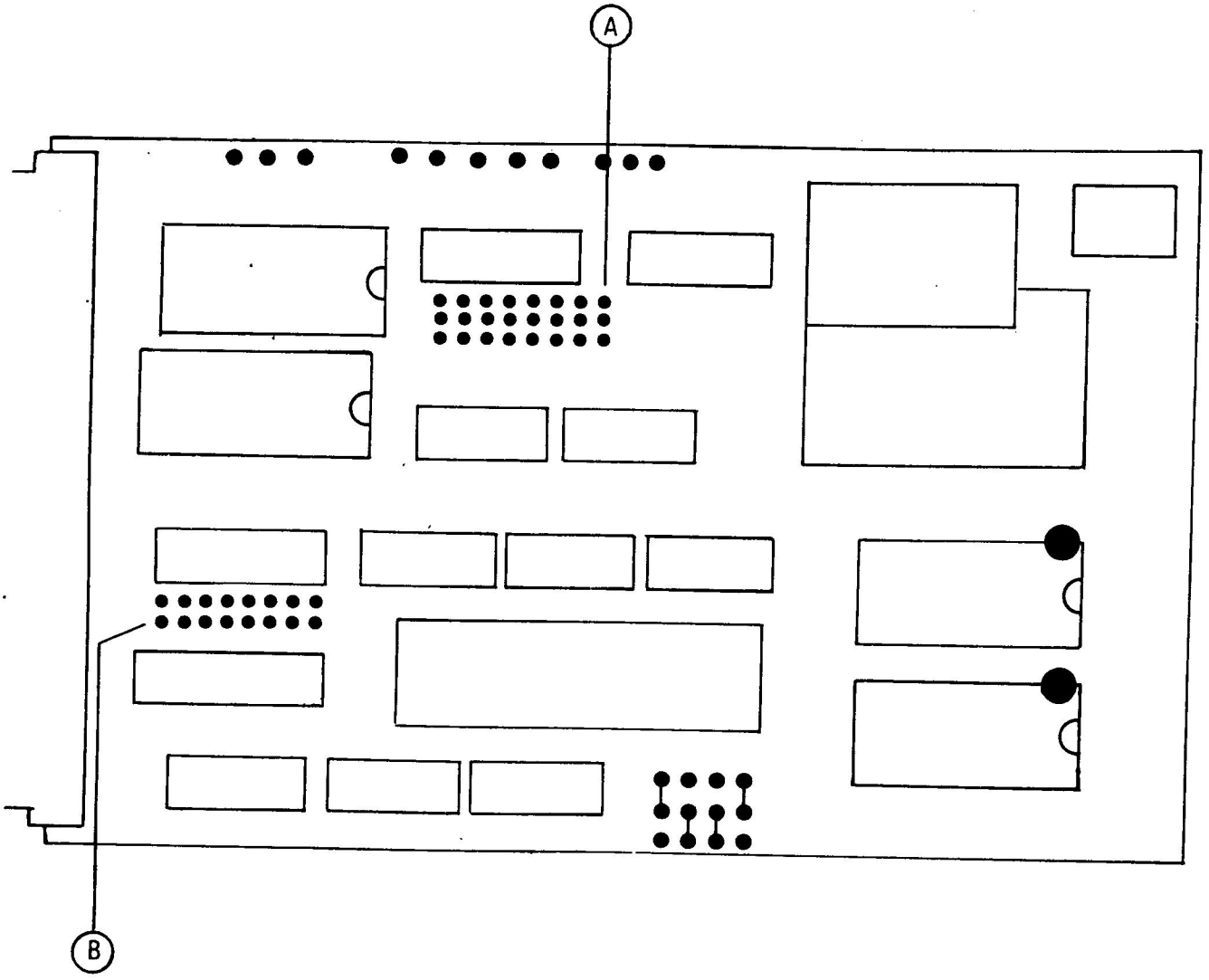
A B C D

OBS

Programvaran som medföljer kräver adresserna 24-27 och därför är kortet byglat

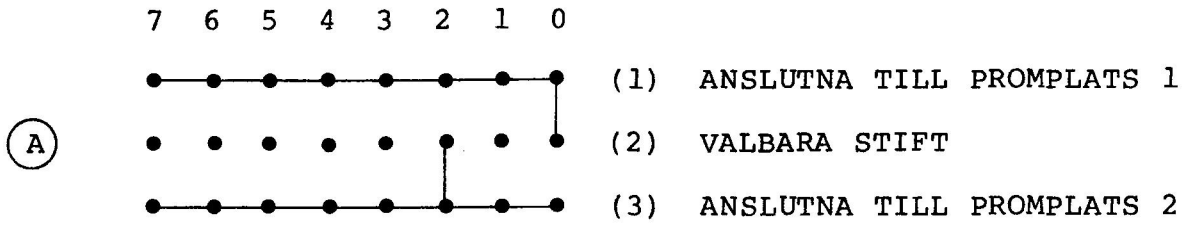
1
2 = (0 0 0 1 1 0 X X)
3

Adressbygling av ABC-PROM/PROG



Adressbyglingen sker i två steg. Med hjälp av stiftraden B välj först ett 8 kbyte-block. Därefter väljs med stiftrad A inom vilket 1 kbyte-block adressering skall ske.

VAL AV 1kb BLOCK



Inom det i (B) valda 8k-blocket väljs nu inom vilket 1k-block som resp. promplats skall ha som startadress.

Precis som vid (B) avser

STIFT	ADRESSOMRÅDE	
0	0-1023	(0H-3FFH)
1	1024-2047	(400H-7FFH)
2	2048-3071	(800H-BFFH)
3	3072-4095	(C00H-FFFH)

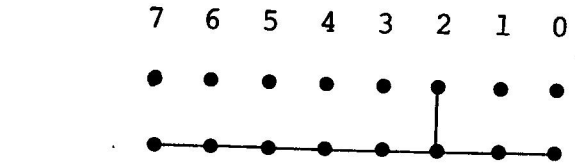
osv

I ovanstående exempel är alltså promplats 1 byglad till startadress 0 och plats 2 till startadress 2048.

Tillsammans med 8k-byglingen tidigare erhålls då adresserna

promplats 1	16384+0	= 16384 (4000H)
promplats 2	16384+2048	= 18432 (4800H)

VAL AV 8K-BLOCK



(B)

Observera att bägge promplatserna måste ligga inom samma 8K-block i och med detta sätt att bygla.

Anslutning av stift		ger adressområdet	
0	0- 8 kb	0- 8191	(0H
1	8-16 kb	8192-16383	(2000H)
2	16-24 kb	16384-24575	(4000H
3	24-32 kb	24576-32767	(6000H
4	32-40 kb	32768-40959	(8000H
5	40-48 kb	40960-49151	(A000H
6	48-56 kb	49152-57343	(C000H
7	56-64 kb	57344-65535	(E000H

I ovanstående exempel är stift 2 anslutet, dvs det valda 8k-blocket ligger inom adresserna (16384-24575). M.a.o. direkt efter ABC 80:s 16k-BASIC.

MINNESKARTA ABC 80

Hexadec. adress	Decimal adress	
FF80H	65408	128 bytes lediga för POKE
FD00H	64768	Systemvariabler
FC00H	64512	CASBUF 1 DOSBUF 7
FB00H	64256	CASBUF 0 DOSBUF 6
FA00H	64000	DOSBUF 5
F900H	63744	DOSBUF 4
F800H	63488	DOSBUF 3
F700H	63232	DOSBUF 2
F600H	62976	DOSBUF 1
F500H	62720	DOSBUF 0
		STACK
C000H	49152	16 KB Arbetsminne
8000H	32768	16 KB Extern minne
7000H	31744	1 KB Bildminne
7800H	30720	1 KB ROM (Printeroption)
7400H	29696	1 KB (Ledigt)
7000H	28672	1 KB ROM (IEC-option)
6000H	24576	4 KB ROM (Flexskiv-option)
4000H	16384	8 KB (Ledigt)
0H	0	16 KB BASIC

BILAGA 1

Exempel på I/O-rutin

```
22
23
24
25
26              Rutin för att hämta in ett
27              Basic-program som är prommat
28              i ASCII-format (.BAS)
29
30              Detta program fungerar som
31              drivrutin för enheten "ROM:".
32
33              Innan enheten "ROM:" kan an-
34              vändas så måste man göra anrop
35              till rutinen "INIT"
36
37
38
39
40
41              I N I T
42              Rutin som definierar enheten
43              ROM:
44
45
46
47
48      INIT:
49              LD      HL,(-502)
50              LD      (32760),HL
51              LD      HL,'RO'
52              LD      (32762),HL
53              LD      A,'M'
54              LD      (32764),A
55              LD      HL,ROM
56              LD      (32765),HL
57              LD      HL,32760
58              LD      (-502),HL
59              RET
60
61
62
63              RUTINER FÖR ENHETEN ROM:
64
65      ROM:    JP      OPEN
66              JP      OPEN
67              JP      CLOSE
68              JP      INPUT
69              JP      FEL
70              JP      FEL
71              JP      FEL
72              JP      FEL
73              JP      FEL
```

```
74
75
76    FEL:    RST     10H
77            DEFB    88H     FEL 8
78
79    OPEN:
80            LD      HL,BASIC
81            LD      (IX+10),L
82            LD      (IX+11),H
83    CLOSE:
84            AND     A
85            RET
86
87
88    INPUT:
89            LD      E,(IX+10)
90            LD      D,(IX+11)
91    INP1:   LD      A(DE)   ;HÄMTA TECKEN FRÅN PROM
92            CP      3
93            JR      Z,EOF; SLUTMÄRKE?
94            LD      (HL),A  ; LAGRA DÄR ABC80 VILL HA DET
95            INC     HL
96            INC     DE
97            CP      13      ; SLUT PÅ RADEN?
98            JR      NZ,INP1 ; NEJ FORTSÄTT
99            LD      (IX+10),E
100           LD      (IX+11),D
101           AND     A
102           RET
102
104   EOF:    XOR     A
105           SCF
106           RET
107
108
109           BASICPROGRAMMET BÖRJAR EFTER
110           LÄGET "BASIC"
111           DET FINNS ETT RETURN TECKEN
112           MELLAN VARJE RAD
113           EFTER SISTA RADEN FINNS DET
114           ETX
115
116   BASIC:
117           DEFM    '1PRINT "HEJ PÅ DEJ'
118           DEFB    13
119           DEFM    '2X=X+1'
120           DEFB    13
121           DEFM    '3IFX<1000GOTO1'
122           DEFB    13
123           DEFM    '4;"KLAR ! ! !"'
124           DEFB    13
125           DEFB    3
126           END
```

BILAGA 2

```
1
2
3               AUTOSTART AV PROMMADE PROGRAM
4
5
6               ORG     6000H   ;DOSPROMMETS PLATS
7               DEFS    4BH
8               JP      AUTO    ;HOPPET MÅSTE VARA PÅ ADR 604BH
9
10    AUTO:
11              CALL    INIT    ;LÄGG IN DEVICET
12              XOR     A
13              LD      (OFE07H),A  } ← File
14              LD      (IY+0EH),1  }
15              EI
16
17              LD      HL,CMD
18              JP      244 Dec
19
20    CMD:      DEFM    'RUN ROM:'
21              DEFB    13 Dec
```

60 YB

```
0000                    ;*************************************
0000                    ;
0000                    ; Rutin för att "hämta in" (ställa
0000                    ; om pekare) för ett program prommat
0000                    ; i BAC-format
0000                    ;                          ol 1981-08-27
0000                    ;
0000                    ;*************************************
0000                    ;
0000                    INIT:
0000 2A0AFE                     LD      HL,(-502)
0003 22F87F                     LD      (32760),HL
0006 21524F                     LD      HL,'RO'
0009 22FA7F                     LD      (32762),HL
000C 3E4D                       LD      A,'M'
000E 32FC7F                     LD      (32764),A
0011 211E00                     LD      HL,ROM_
0014 22FD7F                     LD      (32765),HL
0017 21F87F                     LD      HL,32760
001A 220AFE                     LD      (-502),HL
001D C9                         RET
001E                    ;
001E                    ;***    RUTINER FÖR ENHETEN ROM:  *****
001E                    ;
001E C33B00             ROM_    JP      OPEN
0021 C33B00                     JP      OPEN
0024 C34100                     JP      CLOSE
0027 C34300                     JP      INPUT
002A C33900                     JP      FEL
002D C33900                     JP      FEL
0030 C33900                     JP      FEL
0033 C33900                     JP      FEL
0036 C33900                     JP      FEL
0039                    ;
0039 D7                 FEL:    RST     10H
003A 88                         DEFB    88H     ; FEL 8
003B                    ;
003B                    OPEN:
003B 2100A8                     LD      HL,43008 ;START PÅ PROM
003E 221CFE                     LD      (65052),HL ;TILL EOFA
0041                    CLOSE:
0041 A7                         AND     A
0042 C9                         RET
0043                    ;
0043                    ;***    DUMMY INPUT-RUTIN         *****
0043                    ;
0043                    ;       STÄLLER OM EOFA OCH HEAP
0043                    ;       SAMT GER FILSLUT
0043                    ;
0043                    INPUT
0043 2100C0                     LD      HL,49152 ;BÖRJAN PÅ RAM
0046 221EFE                     LD      (65054),HL
0049 2220FE                     LD      (65056),HL
004C AF                         XOR     A       ;FILSLUT
004D 37                         SCF
004E C9                         RET
0000                            END
```