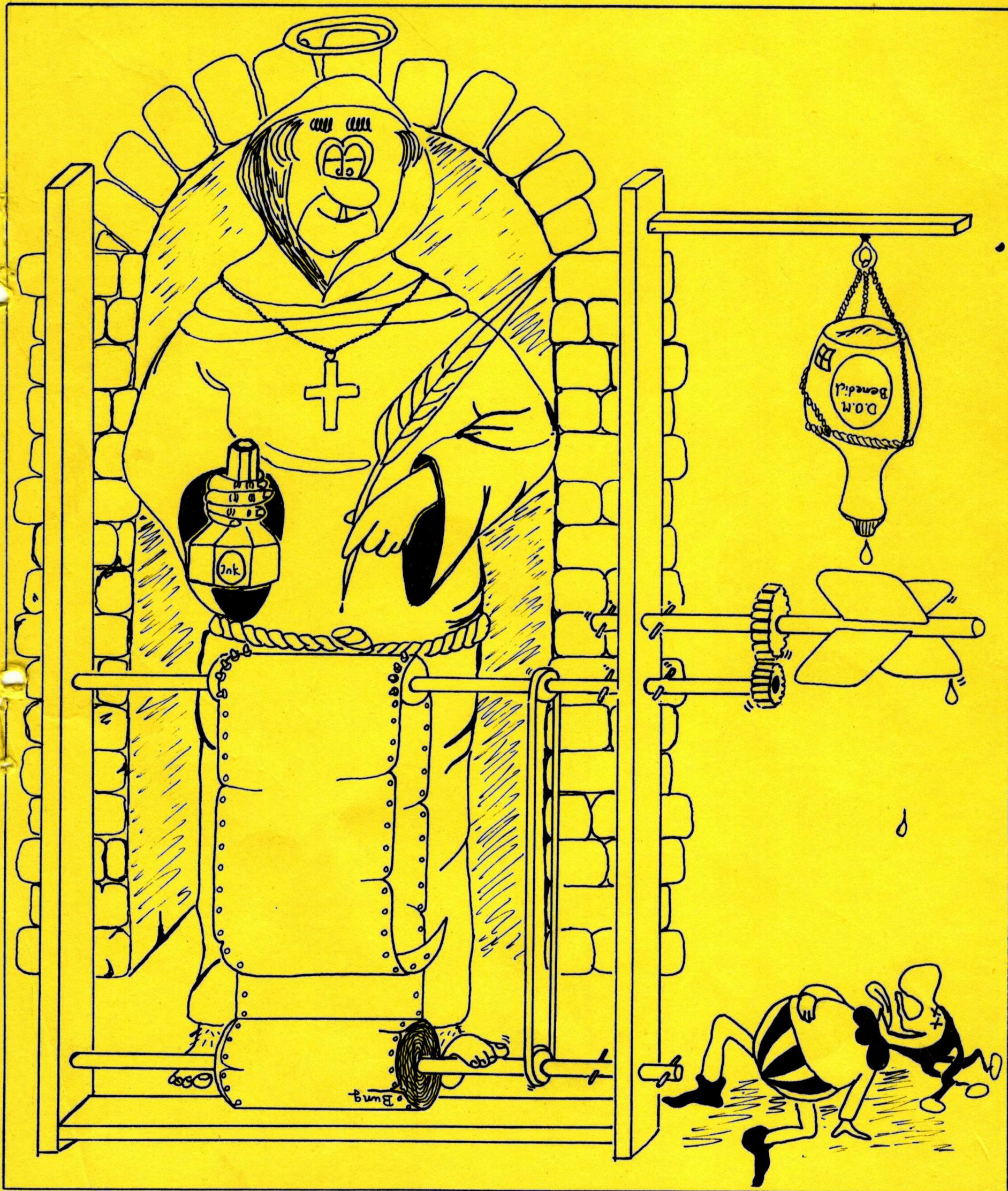


Tidskrift för Privatdataklubben PD68 Årgång 3



\*\*\*\*\*

REDAKTIONEN

STYRELSEN

MPU-larenredaktionen består av

\* PD68 nuvarandestyrelse.

Roger Holmstrand (RH)  
Rapsodivägen 168  
142 00 TRÅNGSUND 08/7713852  
(FLEX2, FLEX9, Papperstape, KCS )

\* Tommy Bland (TBl) Ordförande  
\* Norrholmsvägen 106  
\* 132 00 SALTSJÖ-B00

Carl Axel Bruno (CAB)  
Timotejvägen 116  
191 77 SOLLENTUNA 08/7543788  
(D2-kit, FLEX2, KCS)

\* Nils Ohlsson (NO) V. Ordförande  
\* Sandfjärdsgatan 68, 12tr  
\* 121 69 JOHANNESHÖV

Lars Landqvist  
Skagersvägen 20 1tr  
121 42 JOHANNESHÖV 08/912625  
(SMOKE 5", KCS)

\* Åke Holm (ÅH) Suppleant  
\* Långsjövägen 15B  
\* 135 54 TYRESÖ

Bernt Svensson (BS)  
co Petzäll, Kocksg 43, 1tr  
116 29 STOCKHOLM 08/448864

\* Anders Karlsson Sekreterare  
\* Fasanstigen 40  
\* 197 00 BRO

Lennart Uusitalo (LU)  
Skrakvägen 20  
141 72 HUDDINGE  
08/973559

\* Bo Ljungblad (BLj) Kassör  
\* Brännkyrkagatan 117  
\* 117 28 STOCKHOLM

\*\*\*\*\*

ANNONSER

Helsida 200:-  
Halvsida 100:-  
Kvartsida 50:-  
Bilagor 200:-

\* Lars Bergstedt Suppleant  
\* Forvägen 31  
\* 145 51 NORSBORG

Priserna gäller tryckfärdigt material. Radannonser för medlemmar är gratis om det inte gäller firma. Tidningens upplaga är för närvarande 250 ex.

\* Gunnar Lovius V. Sekreterare  
\* Ragnebergsvägen 21  
\* 136 67 HANDEN

För material till tidningen ring eller skriv gärna till någon i redaktionen ovan eller sänd detta till, PD68

BOX 98 122 21 ENSKEDE 1

\*\*\*\*\*

MEDLEMSINTRÄDE

\* Vill DU bli medlem i PD68 ?  
\* Sätt in 75:- på postgironr  
\* PG960468-7 samt namn adress o  
\* tele nr så kommer medlemskap  
\* på posten. OBS GLÖM EJ NAMN!!!

\*\*\*\*\*

EN UPPMANING TILL MEDLEMMARNA!!! VAR INTE BLYGA MED ATT SKRIVA. MED SÅ MÅNGA I KLUBBEN BORDE DET REGNA IN MATERIAL TILL REDAKTIONEN. MEN TYVÄR ÄR DET INTE SÅ, MEN SKÄRPNING, ALLT ÄRVÄLKOMMET STORT SOM smått. MED HÄLSNING REDAKTIONEN

PS. DET ÄR FÖR OSS EN FÖRDEL OM NI KAN SÄNDA MATERIALET PÅ DISK ELLER TAPE. VI RETURNERAR SEDAN DISKEN, TAPEN.

\*\*\*\*\*

Som nog alla vet är PD68 en ideell förening. Det betyder att själva existensen av klubben är grundat på frivilligt och oavlönat arbete. Tag tidningen till exempel. Principen för hur MPU-laren skall ges ut är enkel: Den ges ut när tillräckligt material finns för att det skall vara ide att trycka den.

Varifrån får då tidningens redaktion detta hett eftertraktade material? Är det männe från vår månghövdade medlemsskara som artiklar, uppslag, ideer och program väljer in i en aldrig sinande flod? Ni gissar tyvärr alldeles rätt. Det är inte därifrån som spaltfyllnaden kommer. Nej det är i stället styrelsen och redaktionen som sliter sitt hår för att MPU-laren skall ha något mellan fram och baksidan.

Flera av oss tycker det är roligt och stimulerande att skriva. Det är inte där skon klämmer. Utan ibland får vi lite mycket att göra med annat. Klubbaktiviteten läggs åt sidan för ett tag. Konsekvensen blir att vi får titta förgäves efter MPU-laren i brevlådan. En annan följd av denna inavel kan vara att det vi skriver om inte är det flertalet finner intressant. Men hur skall vi kunna veta detta om ingen säger hur han/hon vill ha det.

Bland annat detta är ett av skälen till att vi går ut med en enkät. Genom den försöker vi ta reda på vad PD68 medlemmen vill få ut av sitt engagemang i klubben. Fyll därför i de frågor som du tycker är relevanta. Har du ingen egen dator hoppar du över de frågorna. Ett frankerat returkuvert har vi bifogat för att du inte skall kunna säga: "Äh... jag som är allergisk mot frimärksklister..." Fatta pennan! Vi räknar med dig!

En annan grej som man saknar är berättelser från medlemmarna om vad de har för sig med datorn. Tag radioamatörerna som exempel. Jag vet flera som har M6800/09 datorn i någon radioapplikation. Att många är intresserade vet jag också av de förfrågningar som vi får. Men varför skriver aldrig någon och berättar om sina system? (Hörde ni i Västerås?..)

Annat att berätta om kan vara en inköpt byggsats, disk, printer etc. Information om bra programvara efterlyses också. Till firmorna i branchen riktar vi bönen och uppmaningen: Skicka information om både hård och mjukvara. Många är intresserade av att läsa "news-flashar" typ pressmeddelanden från TSC lättfattligt tolkade.

För att MPU-laren skall bli en bättre tidning, krävs mera insatser från flera. Ställ inte så höga krav på det egna materialet. Är tanken god kan vi alltid fixa till formen. Avslutningsvis - lyft krattan från tangentbordet, fatta pennan och skriv till PD68. Än har ingen programlista behövts läggas åt sidan för att ge plats åt redaktionellt material. Vi ser fram emot den dagen. Bergis!!

# Othello<sup>TM</sup>

The Game that takes a minute to learn —  
and a lifetime to master

FOR TWO PLAYERS

1. The players decide who is to be white and who is to be black. The same colours are kept throughout the game.
2. Each player places two discs of his colour on the board as shown in Figure 1.

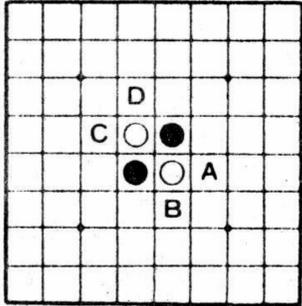


Figure 1

3. Black makes the first move.
4. On every move a disc must be placed next to an opponent's disc, either sideways, lengthways or diagonally. The disc placed **must** trap an opponent's disc (or discs) between the one placed and one already on the board in any direction.

In Figure 1 A, B, C, D, represent the moves that black can make.

If black decides to place his disc in position A, he has captured the white disc E shown in Figure 2 and turns it over so that it is black side up.

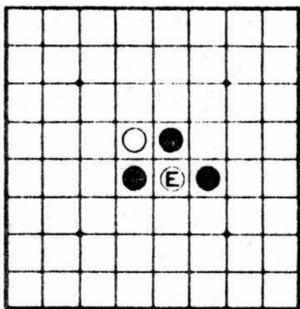


Figure 2

5. Now it is white's move. Figure 3 shows the moves that are available (A, B, C).

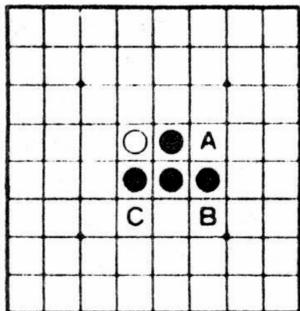


Figure 3

If white decides to place his disc in position B he has captured the black disc F (Figure 4) and turns it white side up (see Figure 5).

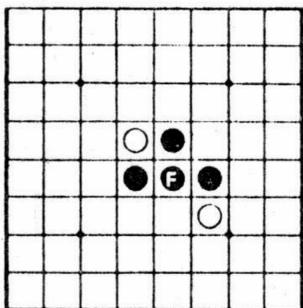


Figure 4

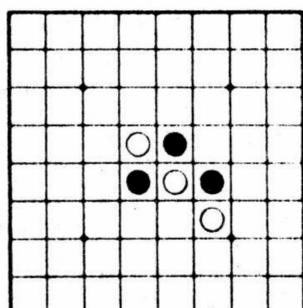


Figure 5

6. Players continue to take turns as illustrated above.

NOTE:- A The number of discs that can be captured in a turn in any or multiple directions is unlimited, eg. in Figure 6 black places his disc in position A and captures all the discs shown in Figure 7.

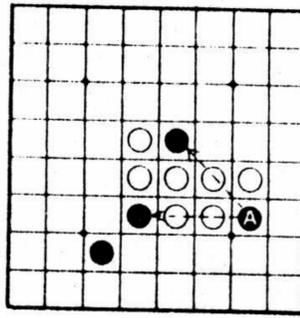


Figure 6

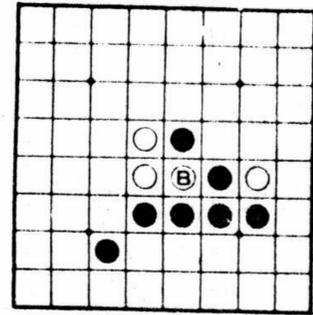


Figure 7

B A disc or discs can only be captured as a direct result of a move, eg. in Figure 7 white disc B is **not** turned over because it is not captured directly on that move.

C Remember there is no limit to the number of discs or the number of directions in which pieces can be captured on any one move. (See Figure 8) black places disc in position A, and captures all the pieces bisected by the arrows, thus arriving at the situation shown in Figure 9 and capturing pieces in eight directions in one move, which is the maximum possible.

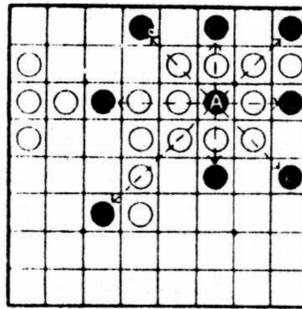


Figure 8

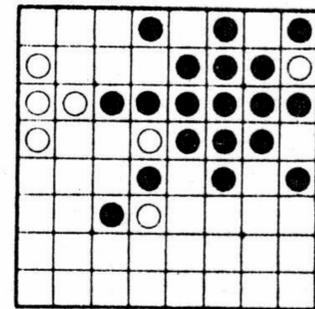


Figure 9

D If it is impossible to capture one of your opponent's men when it is your move, you miss that turn and cannot move again until you can lay a disc that captures at least one of your opponent's discs. Until you can lay such a disc your opponent continues to play, turning over as many of your discs as possible.

7. Play continues as above until:
  - The board is filled;
  - One player has no pieces of his colour left on the board;
  - No further moves are possible (as in Note D) for either player.

8. At this time the discs are counted and the player with most of his colour on the board is declared the winner.

9. There is a slight advantage to going first. Therefore, the more experienced player may give this advantage to the less experienced player. After one game, the winner may wish to let the loser go first. These rules should be established before starting a series of games.

10. When a skillful player is playing against an unskilled player, the skilled player may take on a handicap by setting up the board to give his opponent a four corner advantage, as white (the skilled player) is doing in Figure G. If the difference in skill is not so great, the more skilled player may give only one, two or three corner advantages.

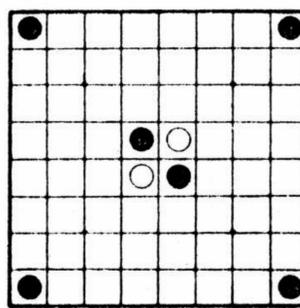


Figure G

```

$
50 REM FRÅN TIDNINGEN "ROM" APRIL 1978
60 REM TBL REV C 1980-10-26
100 PRINT "OTHELLO 1.5"
280 PRINT
290 PRINT "VILL DU HA INSTRUKTIONER (1=JA 2=NEJ)";
300 INPUT R
310 IF R=2 THEN 400
340 PRINT "FÖR ATT SPELA MATA IN X,Y KOORDINATERNA"
350 PRINT "AV POSITIONEN DU VILL FLYTTA. OM DU"
360 PRINT "GER 99,9 SKRIVS BRÅDET UT."
370 PRINT "X ÄR NEDÅT OCH Y HORIZONTELLT"
380 PRINT
390 PRINT "OM DU GER 22,9 PRINTAS ALLA DINA TILLGÅNGLIGA DRAG"
395 PRINT "OM DU GER 0,0 AVSLUTAS SPELET."
400 PRINT
405 REM
410 PRINT
420 DIM T(10,10),M(20,4),D(10,10)
430 DIM W(10,10),F(20,2),Z(7,2)
440 DIM G(4,8),O(8)
445 DIM V(10)
450 REM INITIERA BESLUTSMATRISEN
460 GOSUB 1670
470 REM INITIERA BORDET
480 GOSUB 1930
490 F9=0
500 PRINT "VEM BÖRJAR (1=DATORN 2 =MOTSPELAREN)";
510 INPUT R
580 GOSUB 2100
590 IF R=2 THEN 640
600 T(5,4)=1
610 T(5,5)=1
620 PRINT "DATORN DRAR :4,3"
630 PRINT "DATORN VÄNDER :4,4"
640 PRINT
641 REM
642 PRINT
650 U1=2
660 U2=1
670 GOSUB 5140
680 IF P2=1 THEN 730
690 PRINT "DU HAR INGA GILTIGA DRAG"
700 IF F9=1 THEN 1430
710 F9=1
720 GOTO 1080
730 IF Q9=2 THEN 750
740 GOSUB 2100
750 PRINT "DU FLYTTAR";
760 INPUT M1,M2
770 IF M1<>99 THEN 800
780 GOSUB 2100
790 GOTO 750
800 IF M1<>22 THEN 890
810 PRINT "DINA DRAG ANTAL VÄNDNINGAR"
820 FOR I=1 TO X
830 M1=M(I,1)-1
840 M2=M(I,2)-1
850 PRINT " ";M1;" ";M2;" ";M(I,3)

```

## TYPES OF PLAY

1. One complete game is played as above.
2. A given number of games (say three) is played. The winner of each game enters as his score the number of chips more than his opponent's showing at the end of the game (ie. white finishes with 40 discs, black with 24, white scores 16). The player with most points at the end of the series is declared the winner.
3. Scoring as in 2. Set a given number of points (say 100). The first player to reach or pass this total at the end of a game is declared the winner.

## DETAILS TO BE CAREFULLY NOTED

1. It must be emphasised that discs are only turned as a direct result of a move, and must be trapped in a direct line (horizontal, vertical, or diagonal). See Figures A and B.

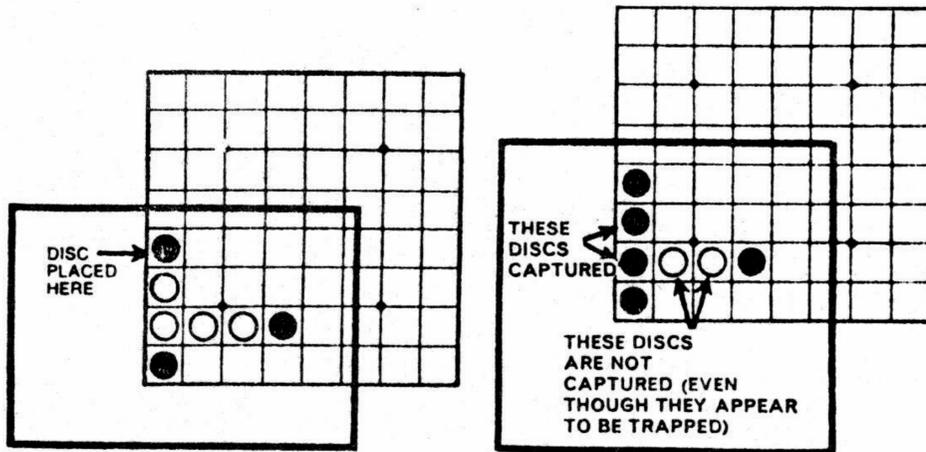


Figure A

Figure B

2. Discs that are to be turned over must be in a continuous line (you may not skip over an empty square or different colour disc). See Figures C and D.

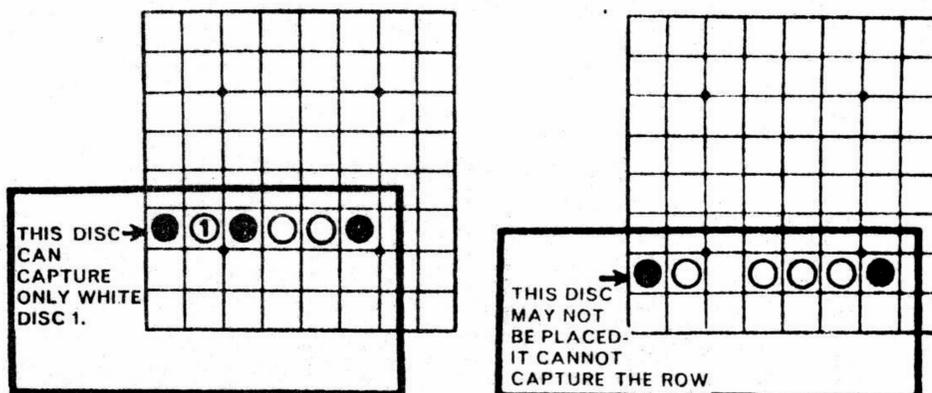


Figure C

Figure D

3. At no time can you place a disc that does not capture at least one opposing disc. When this situation occurs you miss your turn, and continue to do so until you can make a capture on your move.
4. Once placed, a disc is NEVER moved to another square, it may be turned over several times during a game, but it is NEVER moved.
5. If one player runs out of discs, he draws discs from his opponent's stock. The opponent CANNOT REFUSE to hand over the discs.
6. If an incorrect move is made it may only be corrected BEFORE the opponent's next move.

## SOME HINTS ON TACTICS

Even among experienced players of Othello there is controversy regarding tactics and strategy. In fact it may be fun not to read the following hints until you have played a few times, and formed your own ideas. In any case, the following deals primarily with board positions and their advantages; but remember each new opponent means a new strategy, and a whole new challenge!

1. A disc in a corner cannot be captured since it cannot be trapped between two discs. Note, however, that it can be a permanent end for three directions. Thus a corner is very valuable and it is a great advantage to occupy a corner square.
2. The effectiveness of a corner can however, be diminished by the opponent manoeuvring a solid defence against it. While a corner is undoubtedly a strong position, it is not necessarily true that gaining one or more corner squares assures winning the game.
3. The squares directly adjoining the corners are also important. By taking these positions it is possible to defeat the strength of the corners.
4. Discs placed along the outside edges are also valuable, because they can be captured from only two directions (left and right of the disc itself) yet can act as the outside end for five directions (see Figure E. White disc A is unable to capture the black disc which has been permanently locked into an edge position by previous moves.

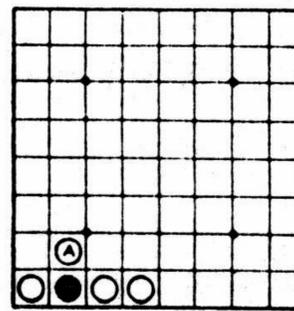


Figure E

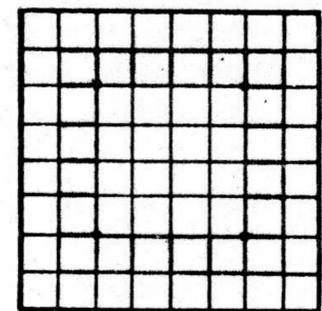


Figure F

5. The next rows in from the four edge rows (see Figure F) could be considered a "danger zone", since a disc in these rows could be your opponent's bridge to a corner or an outside edge position. The dots on the board indicate this "danger zone".
6. While the move that captures as many discs as possible may seem the most attractive, sometimes the less dramatic move is wiser.
7. A solid block of one colour, especially backed up against a corner or a side, is often the key to winning the game.
8. On the other hand, aggressively infiltrating a solid block of one colour can often change the run of the game.

```

860 NEXT I
870 PRINT
880 GOTO 750
890 M1=M1+1
900 M2=M2+1
905 IF M1=1 THEN 1430
910 F9=0
940 U1=2
950 U2=1
960 GOSUB 2500
970 IF P1=1 THEN 1000
980 PRINT "OTILLÄTET DRAG...FÖRSÖK IGEN"
990 GOTO 640
1000 T(M1,M2)=2
1010 FOR I=1 TO L
1020 T(F(I,1),F(I,2))=2
1030 F(I,1)=F(I,1)-1
1040 F(I,2)=F(I,2)-1
1050 PRINT "    VÄNDNINGAR :";F(I,1);", ";F(I,2)
1060 NEXT I
1070 REM *** KOLLA SLUT AV SPEL
1080 GOSUB 2410
1090 IF P1=1 THEN 1430
1110 GOSUB 3700
1130 U1=1
1140 U2=2
1150 GOSUB 5140
1160 IF P2=1 THEN 1220
1170 PRINT "DATORN HAR INGA TILLÄTNA DRAG"
1180 IF F9=1 THEN 1430
1190 F9=1
1200 GOTO 640
1220 GOSUB 5390
1230 F9=0
1260 U1=1
1270 U2=2
1280 GOSUB 2500
1290 T(M1,M2)=1
1300 M1=M1-1
1310 M2=M2-1
1320 PRINT "DATORN FLYTTAR : ";M1;",";M2
1330 FOR I=1 TO L
1340 T(F(I,1),F(I,2))=1
1350 F(I,1)=F(I,1)-1
1360 F(I,2)=F(I,2)-1
1370 PRINT "    VÄNDNINGAR : ";F(I,1);", ";F(I,2)
1380 NEXT I
1390 REM ***SLUT PÅ SPE ?
1400 GOSUB 2410
1410 IF P1=0 THEN 640
1420 REM SLUT PÅ SPE
1430 C1=0
1440 C2=0
1450 FOR I=2 TO 9
1460 FOR J=2 TO 9
1470 IF T(I,J)=1 THEN 1500
1480 IF T(I,J)=2 THEN 1520
1490 GOTO 1530

```

```

1500 C1=C1+1
1510 GOTO 1530
1520 C2=C2+1
1530 NEXT J
1540 NEXT I
1550 PRINT "***** SPE SLUT *****"
1560 PRINT
1570 PRINT "DATORN HAR ";C1;" PÅSER, DU HAR ";C2;" PÅSER"
1590 GOSUB 2100
1600 PRINT "SPELA IGEN (1=JA 2=NEJ) ";
1610 INPUT R
1620 IF R=1 THEN 480
1630 GOTO 6400
1650 REM ***SUBROUTINER
1670 REM ***INITIERA BESLUTSMATRISEN
1680 FOR I=1 TO 10
1690 FOR J=1 TO 10
1700 READ W(I,J)
1710 NEXT J
1720 NEXT I
1730 FOR I=1 TO 4
1740 FOR J=1 TO 8
1750 READ G(I,J)
1760 NEXT J
1770 NEXT I
1780 RETURN
1790 DATA 00,00,00,00,00,00,00,00,00,00
1800 DATA 00,10,00,09,09,09,09,00,10,00
1810 DATA 00,00,00,09,05,05,09,00,00,00
1820 DATA 00,09,09,07,07,07,07,09,09,00
1830 DATA 00,09,05,07,06,06,07,05,09,00
1840 DATA 00,09,05,07,06,06,07,05,09,00
1850 DATA 00,09,09,07,07,07,07,09,09,00
1860 DATA 00,00,00,09,05,05,09,00,00,00
1870 DATA 00,10,00,09,09,09,09,00,10,00
1880 DATA 00,00,00,00,00,00,00,00,00,00
1890 DATA 00,00,02,02,02,02,01,03
1900 DATA 00,00,02,02,02,01,03,03
1910 DATA 00,00,02,02,01,01,03,03
1920 DATA 00,00,02,01,01,01,03,03
1930 REM *** SÄTT UPP BRÄDET
1940 FOR I=2 TO 9
1950 FOR J=2 TO 9
1960 T(I,J)=0
1970 NEXT J
1980 NEXT I
1990 FOR I=1 TO 10
2000 T(I,1)=9
2010 T(I,10)=9
2020 T(1,I)=9
2030 T(10,I)=9
2040 NEXT I
2050 T(5,5)=2
2060 T(6,6)=2
2070 T(6,5)=1
2080 T(5,6)=1
2090 RETURN
2100 REM SKRIV UT BRÄDET

```

```

2120 PRINT ".=BLANK, 1=DATORN, 2=DU"
2130 PRINT " 12345678"
2140 FOR I=2 TO 9
2150 I1=I-1
2160 PRINT I1;" ";
2170 FOR J=2 TO 9
2172 Y6=T(I,J)
2174 IF Y6=0 THEN PRINT ".";
2176 IF Y6=1 THEN PRINT "1";
2178 IF Y6=2 THEN PRINT "2";
2190 NEXT J
2200 PRINT
2210 NEXT I
2215 PRINT " 12345678"
2217 PRINT
2220 RETURN
2410 REM *** SLUT PÅ SPE
2420 P1=0
2430 FOR I=2 TO 9
2440 FOR J=2 TO 9
2450 IF T(I,J)=0 THEN 2490
2460 NEXT J
2470 NEXT I
2480 P1=1
2490 RETURN
2500 REM **AR DET ETT TILLÄTET DRAG
2510 P1=0
2520 L=0
2530 IF M1<2 THEN 2560
2540 IF M2<2 THEN 2560
2550 GOTO 2580
2560 P1=0
2570 RETURN
2580 IF M1>9 THEN 2610
2590 IF M2>9 THEN 2610
2600 GOTO 2630
2610 P1=0
2620 RETURN
2630 IF T(M1,M2)=0 THEN 2660
2640 P1=0
2650 RETURN
2660 IF T(M1+1,M2)<>U2 THEN 2770
2670 Q=0
2680 FOR I=M1+1 TO 9
2690 IF T(I,M2)=U1 THEN 2760
2700 IF T(I,M2)=0 THEN 2770
2710 Q=Q+1
2720 Z(Q,1)=I
2730 Z(Q,2)=M2
2740 NEXT I
2750 GOTO 2770
2760 GOSUB 3630
2770 IF T(M1+1,M2+1)<>U2 THEN 2890
2780 Q=0
2790 FOR I=1 TO 7
2800 IF T(M1+I,M2+I)=9 THEN 2890
2810 IF T(M1+I,M2+I)=0 THEN 2890
2820 IF T(M1+I,M2+I)=U1 THEN 2880

```

```

2830 Q=Q+1
2840 Z(Q,1)=M1+I
2850 Z(Q,2)=M2+I
2860 NEXT I
2870 GOTO 2890
2880 GOSUB 3630
2890 IF T(M1,M2+1)<>U2 THEN 3000
2900 Q=0
2910 FOR I=M2+1 TO 9
2920 IF T(M1,I)=U1 THEN 2990
2930 IF T(M1,I)=0 THEN 3000
2940 Q=Q+1
2950 Z(Q,1)=M1
2960 Z(Q,2)=I
2970 NEXT I
2980 GOTO 3000
2990 GOSUB 3630
3000 IF T(M1-1,M2-1)<>U2 THEN 3120
3010 Q=0
3020 FOR I=1 TO 7
3030 IF T(M1-I,M2-1)=9 THEN 3120
3040 IF T(M1-I,M2-1)=0 THEN 3120
3050 IF T(M1-I,M2-1)=U1 THEN 3110
3060 Q=Q+1
3070 Z(Q,1)=M1-I
3080 Z(Q,2)=M2-I
3090 NEXT I
3100 GOTO 3120
3110 GOSUB 3630
3120 IF T(M1-1,M2)<>U2 THEN 3240
3130 Q=0
3140 FOR I=1 TO 7
3150 IF T(M1-I,M2)=9 THEN 3240
3160 IF T(M1-I,M2)=0 THEN 3240
3170 IF T(M1-I,M2)=U1 THEN 3230
3180 Q=Q+1
3190 Z(I,1)=M1-I
3200 Z(I,2)=M2
3210 NEXT I
3220 GOTO 3240
3230 GOSUB 3630
3240 IF T(M1-1,M2+1)<>U2 THEN 3360
3250 Q=0
3260 FOR I=1 TO 7
3270 IF T(M1-I,M2+1)=9 THEN 3360
3280 IF T(M1-I,M2+1)=0 THEN 3360
3290 IF T(M1-I,M2+1)=U1 THEN 3350
3300 Q=Q+1
3310 Z(Q,1)=M1-I
3320 Z(Q,2)=M2+I
3330 NEXT I
3340 GOTO 3360
3350 GOSUB 3630
3360 IF T(M1,M2-1)<>U2 THEN 3480
3370 Q=0
3380 FOR I=1 TO 7
3390 IF T(M1,M2-I)=9 THEN 3480
3400 IF T(M1,M2-I)=0 THEN 3480

```

```

3410 IF T(M1,M2-I)=U1 THEN 3470
3420 Q=Q+1
3430 Z(Q,1)=M1
3440 Z(Q,2)=M2-I
3450 NEXT I
3460 GOTO 3480
3470 GOSUB 3630
3480 IF T(M1+1,M2-1)<>U2 THEN 3600
3490 Q=0
3500 FOR I=1 TO 7
3510 IF T(M1+I,M2-I)=9 THEN 3600
3520 IF T(M1+I,M2-I)=0 THEN 3600
3530 IF T(M1+I,M2-I)=U1 THEN 3590
3540 Q=Q+1
3550 Z(Q,1)=M1+I
3560 Z(Q,2)=M2-I
3570 NEXT I
3580 GOTO 3600
3590 GOSUB 3630
3600 IF L=0 THEN 3620
3610 P1=1
3620 RETURN
3630 REM **SHIFT RUTIN
3640 FOR Q2=1 TO Q
3650 L=L+1
3660 F(L,1)=Z(Q2,1)
3670 F(L,2)=Z(Q2,2)
3680 NEXT Q2
3690 RETURN
3700 REM **BERÄKNA BESLUTS MATRISEN
3710 FOR I=1 TO 10
3720 FOR J=1 TO 10
3730 D(I,J)=W(I,J)
3740 NEXT J
3750 NEXT I
3760 IF T(2,2)<>1 THEN 3840
3770 D(2,3)=9
3780 D(3,2)=9
3790 D(3,3)=8
3800 D(4,4)=8
3810 D(5,5)=8
3820 D(6,6)=8
3830 D(7,7)=8
3840 IF T(2,9)<>1 THEN 3920
3850 D(2,8)=9
3860 D(3,9)=9
3870 D(3,8)=8
3880 D(4,7)=8
3890 D(5,6)=8
3900 D(6,5)=8
3910 D(7,4)=8
3920 IF T(9,2)<>1 THEN 4000
3930 D(8,2)=9
3940 D(9,3)=9
3950 D(8,3)=8
3960 D(7,4)=8
3970 D(6,5)=8
3980 D(5,6)=8

```

```

3990 D(4,7)=8
4000 IF T(9,9)<>1 THEN 4080
4010 D(9,8)=9
4020 D(8,9)=9
4030 D(8,8)=8
4040 D(7,7)=8
4050 D(6,6)=8
4060 D(5,5)=8
4070 D(4,4)=8
4080 IF T(2,2)<>2 THEN 4120
4090 D(2,3)=1
4100 D(3,2)=1
4110 D(3,3)=1
4120 IF T(2,9)<>2 THEN 4160
4130 D(2,8)=1
4140 D(3,8)=1
4150 D(3,9)=1
4160 IF T(9,9)<>2 THEN 4200
4170 D(8,9)=1
4180 D(8,7)=1
4190 D(9,8)=1
4200 IF T(9,2)<>2 THEN 4240
4210 D(8,2)=1
4220 D(8,3)=1
4230 D(9,3)=1
4240 IF T(2,4)<>1 THEN 4310
4250 IF T(3,3)<>2 THEN 4280
4260 IF T(2,2)=1 THEN 4280
4270 D(4,2)=1
4280 FOR I=3 TO 7
4290 D(I,4)=8
4300 NEXT I
4310 IF T(2,5)<>1 THEN 4350
4320 FOR I=3 TO 7
4330 D(I,5)=8
4340 NEXT I
4350 IF T(2,6)<>1 THEN 4390
4360 FOR I=3 TO 7
4370 D(I,6)=8
4380 NEXT I
4390 IF T(2,7)<>1 THEN 4460
4400 IF T(3,8)<>2 THEN 4430
4410 IF T(2,9)=1 THEN 4430
4420 D(4,9)=1
4430 FOR I=3 TO 7
4440 D(I,7)=8
4450 NEXT I
4460 IF T(4,2)<>1 THEN 4530
4470 IF T(3,3)<>2 THEN 4500
4480 IF T(2,2)=1 THEN 4500
4490 D(2,4)=1
4500 FOR I=3 TO 7
4510 D(4,I)=8
4520 NEXT I
4530 IF T(5,2)<>1 THEN 4570
4540 FOR I=3 TO 7
4550 D(5,I)=8
4560 NEXT I

```

```

4570 IF T(6,2)<>1 THEN 4610
4580 FOR I=3 TO 7
4590 D(6,I)=8
4600 NEXT I
4610 IF T(7,2)<>1 THEN 4680
4620 IF T(8,3)<>2 THEN 4650
4630 IF T(9,2)=1 THEN 4650
4640 D(9,4)=1
4650 FOR I=3 TO 7
4660 D(7,I)=8
4670 NEXT I
4680 IF T(9,4)<>1 THEN 4750
4690 IF T(8,3)<>2 THEN 4720
4700 IF T(9,2)=1 THEN 4720
4710 D(7,2)=1
4720 FOR I=1 TO 5
4730 D(9-I,4)=8
4740 NEXT I
4750 IF T(9,5)<>1 THEN 4790
4760 FOR I=1 TO 5
4770 D(9-I,5)=8
4780 NEXT I
4790 IF T(9,6)<>1 THEN 4830
4800 FOR I=1 TO 5
4810 D(9-I,6)=8
4820 NEXT I
4830 IF T(9,7)<>1 THEN 4900
4840 IF T(8,8)<>2 THEN 4870
4850 IF T(9,9)=1 THEN 4870
4860 D(7,9)=1
4870 FOR I=1 TO 5
4880 D(9-I,7)=8
4890 NEXT I
4900 IF T(7,9)<>1 THEN 4970
4910 IF T(8,8)<>2 THEN 4940
4920 IF T(9,9)=1 THEN 4940
4930 D(9,7)=1
4940 FOR I=1 TO 5
4950 D(7,9-I)=8
4960 NEXT I
4970 IF T(6,9)<>1 THEN 5010
4980 FOR I=1 TO 5
4990 D(6,9-I)=8
5000 NEXT I
5010 IF T(5,9)<>1 THEN 5050
5020 FOR I=1 TO 5
5030 D(5,9-I)=8
5040 NEXT I
5050 IF T(4,9)<>1 THEN 5120
5060 IF T(3,3)<>2 THEN 5090
5070 IF T(2,9)=1 THEN 5090
5080 D(2,7)=1
5090 FOR I=1 TO 5
5100 D(4,9-I)=8
5110 NEXT I
5120 GOSUB 5770
5130 RETURN
5140 REM FUNDERA UT TILLÄTNA DRAG...DATORN

```

```

5150 X=0
5160 P2=0
5170 REM KOLLA HUR MANGA AV SPELARENS PJASER PA BRÄDET
5180 Q7=0
5190 FOR M1=2 TO 9
5200 FOR M2=2 TO 9
5210 IF T(M1,M2)<>2 THEN 5230
5220 Q7=Q7+1
5230 NEXT M2
5240 NEXT M1
5250 FOR M1=2 TO 9
5260 FOR M2=2 TO 9
5270 GOSUB 2500
5280 IF P1=0 THEN 5360
5290 P2=1
5300 X=X+1
5310 M(X,1)=M1
5320 M(X,2)=M2
5330 M(X,3)=L
5340 IF L<>Q7 THEN 5360
5350 D(M1,M2)=11
5360 NEXT M2
5370 NEXT M1
5380 RETURN
5390 REM HITTA BASTA DRAG
5400 FOR I=1 TO X
5410 M(I,4)=D(M(I,1),M(I,2))
5420 NEXT I
5430 K2=4
5440 GOSUB 5570
5450 IF X=1 THEN 5540
5460 IF M(1,4)<>M(2,4) THEN 5540
5470 FOR I=1 TO X-1
5480 IF M(I,4)<>M(I+1,4) THEN 5510
5490 NEXT I
5500 GOTO 5520
5510 X=1
5520 K2=3
5530 GOSUB 5570
5540 M1=M(1,1)
5550 M2=M(1,2)
5560 RETURN
5570 REM ***SORTERA DRAGEN
5580 F=1
5590 FOR J=1 TO X
5600 IF F=0 THEN 5760
5610 F=0
5620 FOR I=1 TO X-1
5630 IF M(I,K2)>M(I+1,K2) THEN 5740
5640 F=1
5650 FOR K=1 TO 4
5660 V(K)=M(I,K)
5670 NEXT K
5680 FOR K=1 TO 4
5690 M(I,K)=M(I+1,K)
5700 NEXT K
5710 FOR K=1 TO 4
5720 M(I+1,K)=V(K)

```

```

5730 NEXT K
5740 NEXT I
5750 NEXT J
5760 RETURN
5770 FOR I=1 TO 8
5780 O(I)=T(2,I+1)
5790 NEXT I
5800 GOSUB 6270
5810 IF O1=1 THEN 5830
5820 D(2,3)=9
5830 FOR I=1 TO 8
5840 O(I)=T(2,10-I)
5850 NEXT I
5860 GOSUB 6270
5870 IF O1=1 THEN 5890
5880 D(2,8)=9
5890 FOR I=1 TO 8
5900 O(I)=T(I+1,2)
5910 NEXT I
5920 GOSUB 6270
5930 IF O1=1 THEN 5950
5940 D(3,2)=9
5950 FOR I=1 TO 8
5960 O(I)=T(10-I,2)
5970 NEXT I
5980 GOSUB 6270
5990 IF O1=1 THEN 6010
6000 D(8,2)=9
6010 FOR I=1 TO 8
6020 O(I)=T(9,I+1)
6030 NEXT I
6040 GOSUB 6270
6050 IF O1=1 THEN 6070
6060 D(9,3)=9
6070 FOR I=1 TO 8
6080 O(I)=T(9,10-I)
6090 NEXT I
6100 GOSUB 6270
6110 IF O1=1 THEN 6130
6120 D(9,8)=9
6130 FOR I=1 TO 8
6140 O(I)=T(I+1,9)
6150 NEXT I
6160 GOSUB 6270
6170 IF O1=1 THEN 6190
6180 D(3,9)=9
6190 FOR I=1 TO 8
6200 O(I)=T(10-I,9)
6210 NEXT I
6220 GOSUB 6270
6230 IF O1=1 THEN 6250
6240 D(8,9)=9
6250 RETURN
6260 REM KOLLA MÖNSTER G MOT ARRAY O
6270 FOR O2=1 TO 4
6280 FOR O3=1 TO 8
6290 IF G(O2,O3)=3 THEN 6340
6300 IF G(O2,O3)=O(O3) THEN 6340
6310 GOTO 6370
6320 IF O(O3)<>2 THEN 6340
6330 GOTO 6370
6340 NEXT O3
6350 O1=0
6360 RETURN
6370 NEXT O2
6380 O1=1
6390 RETURN
6400 END

```

+GET EKO

READY

RUN

OTHELLO 1.5

VILL DU HA INSTRUKTIONER (1=JA 2=NEJ)? 1  
FÖR ATT SPELA MATA IN X,Y KOORDINATERNA  
AV POSITIONEN DU VILL FLYTTA. OM DU  
GER 99,9 SKRIVS BRÅDET UT.  
X ÄR NEDÅT OCH Y HORIZONTELLT

OM DU GER 22,9 PRINTAS ALLA DINA TILLGÅNGLIGA DRAG  
OM DU GER 0,0 AVSLUTAS SPELET.

VEM BÖRJAR (1=DATORN 2 =MOTSPELAREN)? 1

.=BLANK, 1=DATORN, 2=DU

```

12345678
1 .....
2 .....
3 .....
4 ...21...
5 ...12...
6 .....
7 .....
8 .....
12345678

```

DATORN FLYTTAR : 2 , 3  
VÄNDNINGAR : 3 , 3

DATORN DRAR : 4, 3  
DATORN VÄNDER : 4, 4

.=BLANK, 1=DATORN, 2=DU

```

12345678
1 .....
2 ..1.....
3 ..1.....
4 ..121...
5 ...12...
6 .....
7 .....
8 .....
12345678

```

.=BLANK, 1=DATORN, 2=DU

```

12345678
1 .....
2 .....
3 .....
4 ..111...
5 ...12...
6 .....
7 .....
8 .....
12345678

```

DU FLYTTAR? 0,0

\*\*\*\*\* SPE SLUT \*\*\*\*\*

DATORN HAR 5 PÅSER, DU HAR 2 PÅSER

.=BLANK, 1=DATORN, 2=DU

```

12345678
1 .....
2 ..1.....
3 ..1.....
4 ..121...
5 ...12...
6 .....
7 .....
8 .....
12345678

```

DU FLYTTAR? 22,9	DINA DRAG	ANTAL	VÄNDNINGAR
	3 , 3	1	
	3 , 5	1	
	5 , 3	1	

DU FLYTTAR? 3,3  
VÄNDNINGAR : 4 , 4

SPELA IGEN (1=JA 2=NEJ) ? 2

READY

NAM PROMPG

```
* 2716 PROM PROGRAMMER
*
* WRITTEN BY BO MOLANDER 80-09-10
*
* R E M A R K S :
* SOFTWARE TIMING LOOPS FOR A 1MHZ SYSTEM.
*
*
*+++++
*
* THE FOLLOWING COMMANDS ARE ACCEPTED:
*
* PROG,XXXX,YYYY,ZZZ
*   PROGRAM PROM FROM MEMORY LOC XXXX
*   THRU YYYY FROM PROM OFFSET ZZZ
*
* READ,XXXX,YYYY,ZZZ
*   READ PROM TO MEMORY
*
* VRF,XXXX,YYYY,ZZZ
*   VERIFY PROM AGAINST MEMORY
*
* CHCK,NNN,ZZZ
*   BLANK CHECK NNN BYTES FROM OFFSET ZZZ
*
* STAT
*   DISPLAY STATUS
*
* SHWM,XXXX,YYYY
*   SHOW MEMORY FROM XXXX THRU YYYY
*
* SHWP,NNN,ZZZ
*   SHOW PROM NNN BYTES FROM OFFSET ZZZ
*
* EXIT
*   EXIT TO MBUG CONTROL
*
*+++++
```

\* HARDWARE EQU'S

\* A 6821 LOCATED AT \$FCED

FCED	DATA	EQU	\$FCED	DATA TO/FROM PROM
FCED	DATDIR	EQU	DATA	DIRECTION (CRA-2)
FCE1	CTRLA	EQU	DATA+1	CONTROL REG A
FCE2	ADDR	EQU	\$FCE2	ADDRESS TO PROM
FCE2	ADRDIR	EQU	ADDR	OUTPUT ONLY
FCE3	CTRLB	EQU	ADDR+1	CONTROL REG B

\* SOFTWARE EQU'S

\* ALL EXBUG(TM) COMPATIBLE

\* EXCEPT 'MBUGE' (\$F564 FOR EXBUG)

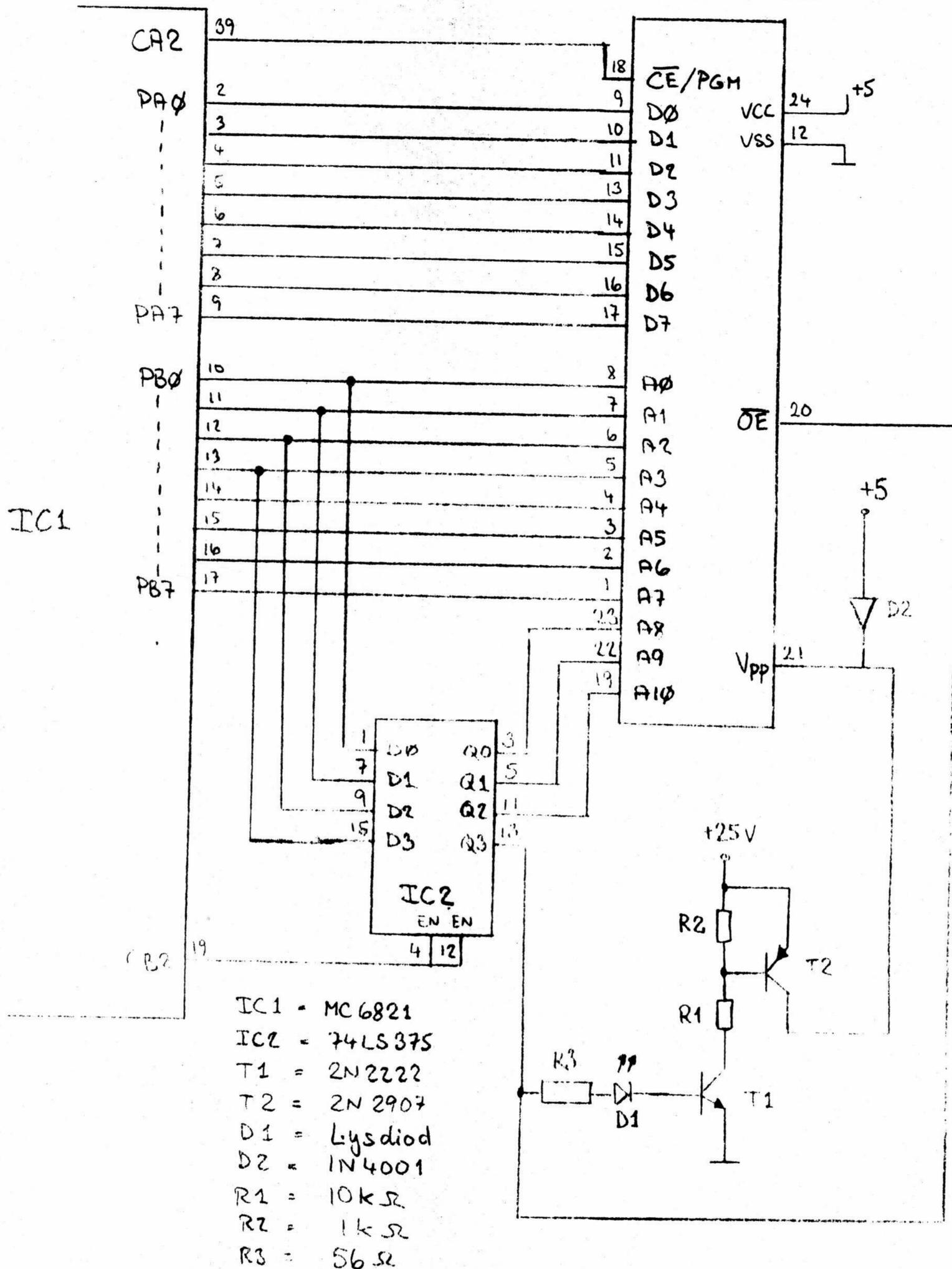
\* AND 'IFBRK' (DOESN'T EXIST)

F003	MBUGE	EQU	\$F003	MONITOR SOFT ENTRY
F006	CBCDHX	EQU	\$F006	CONVERT ASCII TO HEX
F015	INCH	EQU	\$F015	INPUT 1 CHAR
F018	OUTCH	EQU	\$F018	OUTPUT 1 CHAR
F01B	OUT2H	EQU	\$F01B	OUTPUT 2 HEX CHAR
F01E	OUT4HS	EQU	\$F01E	OUTPUT 4 HEX + SPACE
F021	PCRLF	EQU	\$F021	OUTPUT CR + LF
F024	PDATA	EQU	\$F024	PRINT STRING W/ CR,LF
F027	PDATA1	EQU	\$F027	PRINT STRING
F02A	PSPACE	EQU	\$F02A	OUTPUT 1 SPACE

FD2D	IFBRK	EQU	\$FD2D	LOOK FOR AW, AX
FF10	PC	EQU	\$FF10	MONITOR PSEUDO-PC
FF80	STACK	EQU	\$FF80	MBUG STACK
0DDA	NL	EQU	\$DDDA	NEW LINE (CR, LF)
000D	CR	EQU	13	

PIA

PROM 2716



```

1000                                ORG    $1000
                                OPT    0,NOG
                                1000    START EQU    *          COLD START HERE
                                * INIT VARIABLES
1000 8E FF80                        LDS    #STACK
1003 CE FF80                        LDX    #SADR
1006 6F 00                        CLRV   CLR    X
1008 08                            INX
1009 8C FFC6                        CPX    #TEMP
100C 26 F8                            BNE   CLRV
100E 86 08                        LDAA   #8
1010 B7 FF88                        STAA  SIZE    SIZE=2K
1013 B7 FF84                        STAA  NBYTE
                                * READY FOR COMMANDS
1016 CE 14AD                        LDX    #MREADY
1019 BD F024                        JSR    PDATA

                                * INIT PIA
101C 7F FCE3                        CLR    CTRLB
101F 86 FF                            LDAA  #$FF
1021 B7 FCE2                        STAA  ADRDIR  ALL OUTPUTS

                                1024    RESTAR EQU    *          WARM START HERE
1024 8E FF80                        LDS    #STACK  INIT SP
1027 86 3C                        LDAA  #%00111100  CRB-2 = 1
1029 B7 FCE3                        STAA  CTRLB  ACCESS TO ADDR
                                * CB2 OUTPUT AND HIGH
102C 7F FCE2                        CLR    ADDR    MSW=0
102F 84 34                            ANDA  #%00110100
1031 B7 FCE3                        STAA  CTRLB  LATCH
1034 86 30                            LDAA  #%00110000
1036 B7 FCE1                        STAA  CTRLA  CA2 OUTPUT, LOW
1039 7F FFC5                        CLR    BURN   BURN INACTIVE

                                * GET INPUT
103C CE 1403                        INPUT  LDX    #MPROMT
103F BD F024                        JSR    PDATA
1042 CE FFC8                        LDX    #INBUF  INPUT BUFFER
1045 BD F015                        GETCH  JSR    INCH
1048 81 18                            CMPA  #$18
104A 27 F0                            BEQ   INPUT
104C A7 00                            STAA  X
104E 8D 45                            BSR   DELIM  CHECK DELIMETER
1050 27 0E                            BEQ   SRCH
1052 08                            INX
1053 8C FFC0                        CPX    #INBUF+5
1056 26 ED                            BNE   GETCH

                                * ERROR
1058 CE 1406                        MERR  LDX    #MWHAT
105B BD F024                        JSR    PDATA
105E 20 C4                            BRA   RESTAR

1060 16                            SRCH  TAB    SAVE DELIMETER
1061 CE 147D                        LDX    #COMTAB
1064 B6 FFC8                        SRCH1  LDAA  INBUF  FIRST CHAR
1067 A1 00                            CMPA  0,X
1069 26 15                            BNE   SRCH2
106B B6 FFC9                        LDAA  INBUF+1
106E A1 01                            CMPA  1,X
1070 26 0E                            BNE   SRCH2
1072 B6 FFCA                        LDAA  INBUF+2
1075 A1 02                            CMPA  2,X
1077 26 07                            BNE   SRCH2

```

```

1079 B6 FFCB      LDAA  INBUF+3
107C A1 03      CMPA  3,X
107E 27 0D      BEQ   FIND      INPUT OK!
1080 08          SRCH2  INX      SKIP TO NEXT ENTRY
1081 08          INX
1082 08          INX
1083 08          INX
1084 08          INX      SKIP ADDRESS
1085 08          INX
1086 8C 14AD     CPX   #ENDTAB
1089 26 D9      BNE   SRCH1
108B 20 CB      BRA   MERR      ERROR!
108D EE 04      FIND  LDX   4,X   GET ADDRESS
108F AD 00      JSR   X        GO DO IT
1091 29 C5      BVS   MERR     ERROR!!!
1093 20 8F      BRA   RESTAR

1095 81 2C      DELIM CMPA  #' ,
1097 27 06          BEQ  DEL9
1099 81 20          CMPA #'
109B 27 02          BEQ  DEL9
109D 81 0D          CMPA #CR
109F 39          DEL9  RTS

10A0 CE 1024     EXIT  LDX   #RESTAR
10A3 FF FF10     STX   PC      FIX RESTART FROM MBUG
10A6 7E F003     JMP   MBUGE

10A9          PROG  EQU   *      BURN PROM
10A9 BD 13BA     JSR   IN3     READ ARGUMENTS
10AC 25 AA      BCS   MERR     ERROR
10AE BD 141F     JSR   ADRCHK  ADDR BOUNDARIES
10B1 24 03      BCC   PROG05  OK!
10B3 7E 1138     JMP   PSIZE

* CHECK UNPROGRAMMED
10B6 BD F021     PROG05 JSR   PCRLF
10B9 FE FFBA     LDX   CURADR  START ADDRESS
10BC BD 125D     JSR   PIAIN   SETUP PIA FOR INPUT
10BF BD 135C     BCHCK JSR   INBYTE
10C2 81 FF      CMPA  #$FF
10C4 26 08      BNE   PROG10  ALREADY PROGRAMMED
10C6 BC FFB2     CPX   EADR    ALL BYTES DONE?
10C9 27 10      BEQ   PROG11
10CB 08          INX
10CC 20 F1      BRA   BCHCK
10CE CE 1511     PROG10 LDX   #MCHG
10D1 BD F024     JSR   PDATA
10D4 BD F015     JSR   INCH
10D7 81 59      CMPA  #'Y
10D9 26 5C      BNE   PROG90
10DB 86 08      PROG11 LDAA  #%00001000
10DD B7 FFC5     STAA  BURN
10E0 BD 1364     JSR   SETADR  VPP=25V; 0E="H"
10E3 CE 04E2     LDX   #1250
10E6 09          W10MS  DEX      WAIT 10MS
10E7 26 FD      BNE   W10MS
10E9 FE FFB6     LDX   OFST
10EC FF FFBC     STX   CUROF  RESTORE POINTERS
10EF FE FFBA     LDX   CURADR

10F2 86 30      LDAA  #%00110000

```

```

10F4 B7 FCE1      STAA  CTRLA      ACCESS TO DATDIR
10F7 C6 FF        LDAB  #$FF
10F9 F7 FCE0      STAB  DATDIR     ALL OUTPUTS
10FC 8A 04        ORAA  %#00000100
10FE B7 FCE1      STAA  CTRLA
1101 BD 1364 PROG1 JSR   SETADR     OUTPUT PROM ADDR
1104 A6 00        LDAA  X          MEMORY BYTE
1106 B7 FCE0      STAA  DATA
1109 CE 186A      LDX   #6250
110C 86 3C        LDAA  %#00111100
110E B7 FCE1      STAA  CTRLA     CA2 HIGH = BURN!
                * NOW WAIT 50MS
1111 09          W50MS  DEX
1112 26 FD        BNE   W50MS
                * PROGRAMMING DONE!
1114 84 F7        ANDA  %#11110111
1116 B7 FCE1      STAA  CTRLA     CA2 LOW
1119 FE FFBA      LDX   CURADR
111C BC FFB2      CPX   EADR       ALL BYTES DONE?
111F 27 06        BEQ   PROG20
1121 08          INX
1122 FF FFBA      STX   CURADR     UPDATE
1125 20 DA        BRA   PROG1     NEXT BYTE.

1127 86 07        PROG20 LDAA  #7         BELL
1129 BD FD18      JSR   OUTCH
112C 7F FFC5      CLR   BURN
112F BD 1364      JSR   SETADR     VPP=5V; OE="L"
1132 C6 0D        LDAB  #CR        DUMMY FOR IN3
1134 20 22        BRA   VERF       GO VERIFY PROGRAMMING
1136 0B          READ99 SEV
1137 39          PROG90 RTS

1138 CE 150B PSIZE LDX   #MSIZE
113B 7E F024      JMP   PDATA     OUT & RTS

                113E      READ  EQU   *          READ PROM TO MEM
113E BD 13BA      JSR   IN3       READ ARGUMENTS
1141 25 F3        BCS   READ99
                * X POINTS TO FIRST ADDRESS
1143 BD 141F      JSR   ADRCHK
1146 25 F0        BCS   PSIZE     ERROR
1148 BD 125D      JSR   PIAIN
                * ACCESS TO DATA, CA2 LOW (CE)

114B BD 135C READ1 JSR   INBYTE    GET ONE BYTE
114E A7 00        STAA  X         WRITE TO MEMORY
1150 BC FFB2      CPX   EADR     DONE?
1153 27 E2        BEQ   PROG90
1155 08          INX
1156 20 F3        BRA   READ1

                1158      VERF  EQU   *          VERIFY PROM
1158 BD 13BA      JSR   IN3
115B 25 D9        BCS   READ99    C SET ON ERROR
115D BD 141F      JSR   ADRCHK
1160 25 D6        BCS   PSIZE
1162 86 FF        LDAA  #$FF
1164 B7 FFC3      STAA  N
1167 B7 FFC4      STAA  NN

```

```

116A BD 125D      JSR   PIAIN      CE LOW
116D BD 135C VERF1 JSR   INBYTE
1170 A1 00      CMPA  X
1172 27 08      BEQ   VERF3
1174 8D 0E      BSR   VERFE      CALL PRINT MISMATCH
1176 BD F02D     JSR   IFBRK      CHECK IF BREAK
1179 26 01      BNE   VERF3
117B 39          VERF90 RTS
117C BC FFB2 VERF3 CPX   EADR
117F 27 FA      BEQ   VERF90
1181 08          INX
1182 20 E9      BRA   VERF1

1184 B7 FFC2 VERFE STAA  SAVEA      PROM BYTE
1187 FF FFC0     STX   SAVEX      MEM. POINTER
118A C6 03      LDAB  #3          4 ITEMS PER LINE
118C 7C FFC3     INC   N
118F F4 FFC3     ANDB  N
1192 26 03      BNE   VERFE1
1194 BD F021     JSR   PCRLF
1197 C6 1F VERFE1 LDAB  %#00011111
1199 7C FFC4     INC   NN
119C F4 FFC4     ANDB  NN
119F 26 11      BNE   VERFE2

* PRINT HEADING EVERY 64 BYTES
11A1 BD F021     JSR   PCRLF
11A4 C6 04      LDAB  #4          ITEMS/LINE
11A6 CE 14FE VERFE3 LDX   #MVERFE    TEXT
11A9 BD F027     JSR   PDATA1     PRINT TEXT
11AC 5A          DECB
11AD 26 F7      BNE   VERFE3
11AF BD F021     JSR   PCRLF
11B2 FE FFBC VERFE2 LDX   CUROF      OFFSET POINTER
11B5 09          DEX          ADJUST
11B6 FF FFC6     STX   TEMP
11B9 CE FFC6     LDX   #TEMP
11BC BD F01E     JSR   OUT4HS     PRINT OFFSET
11BF CE FFC2     LDX   #SAVEA     POINT TO PROM BYTE
11C2 BD 1473     JSR   OUT2HS     PRINT IT
11C5 FE FFC0     LDX   SAVEX      MEMORY POINTER
11C8 BD 1479     JSR   O2H2SP     PRINT MEM-BYTE
11CB 09          DEX
11CC 39          RTS          OUT2H INC' BY 1

11CD          CHCK  EQU   *          CHECK PROM IS BLANK
11CD BD 138B     JSR   IN2          READ TWO ARGUMENTS
11D0 25 2B      BCS   CHCK99
11D2 BD 1413     JSR   SIZCHK
11D5 24 03      BCC   CHCK1
11D7 7E 1138     JMP   PSIZE       ERROR
11DA C6 FF      CHCK1 LDAB  #$FF
11DC F7 FFC3     STAB  N
11DF F7 FFC4     STAB  NN
11E2 BD 125D     JSR   PIAIN
11E5 CE 0000     LDX   #0          COUNTER
11E8 BD 135C CHCK11 JSR   INBYTE      READ A DATA BYTE
11EB 81 FF      CMPA  #$FF
11ED 27 02      BEQ   CHCK13
11EF 8D 0E      BSR   CHCKE
11F1 BD F02D CHCK13 JSR   IFBRK
11F4 27 06      BEQ   CHCK90     BREAK
11F6 08          INX

```

```

11F7 BC FFB4          CPX    NBYTE
11FA 26 EC           BNE    CHCK11
11FC 39             CHCK90 RTS
11FD 0B             CHCK99 SEV
11FE 39             RTS
* DISPLAY THE ERRONOUS BYTE
11FF B7 FFC2 CHCKE STAA  SAVEA
1202 C6 03          LDAB  #3
1204 7C FFC3          INC   N
1207 F4 FFC3          ANDB N
120A 26 03          BNE  CHCK2
120C BD F021         JSR  PCRLF
120F C6 1F CHCK2 LDAB  %#00011111
1211 7C FFC4          INC  NN
1214 F4 FFC4          ANDB NN
1217 26 14          BNE  CHCK3
* PRINT HEADING
1219 FF FFBE          STX  SLASK    SAVE POINTER
121C BD F021         JSR  PCRLF
121F C6 04          LDAB  #4
1221 CE 151A CHCK4 LDX  #MCHCK
1224 BD F027         JSR  PDATA1
1227 5A             DECB
1228 26 F7          BNE  CHCK4
122A BD F021         JSR  PCRLF
122D FE FFBC CHCK3 LDX  CUROF    OFFSET
1230 09             DEX
POINT TO CURRENT BYTE
1231 FF FFC6          STX  TEMP    POINT TO OFST
1234 CE FFC6          LDX  #TEMP
1237 BD F01E         JSR  OUT4HS  PRINT OFFSET
123A CE FFC2          LDX  #SAVEA
123D BD 1479         JSR  02H25P
1240 FE FFBE          LDX  SLASK    RESTORE X
1243 39             RTS          NEXT BYTE

1244 STAT EQU * DISPLAY STATUS
1244 C1 0D          CMPB #CR
1246 26 B5          BNE  CHCK99  ERROR RETURN
1248 CE 14CC        LDX  #MSTAT
124B BD F024        JSR  PDATA
124E CE FFBD        LDX  #SADR
1251 BD F01E        JSR  OUT4HS
1254 BD F01E        JSR  OUT4HS
1257 BD F01E        JSR  OUT4HS  PRINT
125A 7E F01E        JMP  OUT4HS

* SUBROUTINE TO SETUP PIA
125D 36 PIAIN PSHA
125E 86 38          LDAA #38
1260 B7 FCE1        STAA CTRLA
1263 7F FCE0        CLR  DATDIR  ALL INPUTS
1266 86 34          LDAA #34
1268 B7 FCE1        STAA CTRLA
126B 32          PULA
126C 39          RTS

126D SHWP EQU * SHOW PROM CONTENTS
126D BD 138B        JSR  IN2     READ TWO ARGUMENTS
1270 25 8B          BCS  CHCK99
1272 BD 1413        JSR  SIZCHK

```

```

* ON RETURN, TEMP=ZZZ+NNN-1
1275 24 03          BCC  SHWP1
1277 7E 1138       JMP  PSIZE  WRONG SIZE
127A 8D E1  SHWP1  BSR  PIAIN
127C F6 FFC7       LDAB TEMP+1
127F C4 F8         ANDB  $$F8  MASK TO FULL LINE
1281 F7 FFC7       STAB  TEMP+1  END PROM ADDRESS
1284 F6 FFBD       LDAB  CUROF+1  OFST LSB
1287 C4 F8         ANDB  $$F8  MASK TO FULL LINE
1289 F7 FFBD       STAB  CUROF+1
128C BD F021  SHWP8 JSR  PCRLF
128F BD F02D  SHWP10 JSR  IFBRK
1292 27 4E        BEQ   SHWP90
1294 BD F021       JSR  PCRLF
1297 CE FFBC       LDX  #CUROF
129A BD F01E       JSR  OUT4HS  PRINT OFFSET
129D FE FFBC       LDX  CUROF
12A0 FF FFC0       STX  SAVEX
12A3 C6 08        LDAB  #8  BYTES/LINE
12A5 BD 135C  SHWP12 JSR  INBYTE  READ PROM, INC CUROF
12A8 B7 FFC2       STAA SAVEA
12AB CE FFC2       LDX  #SAVEA  POINT TO PROM BYTE
12AE BD 1473       JSR  OUT2HS
12B1 5A          DECB
12B2 26 F1        BNE  SHWP12
12B4 FE FFC0       LDX  SAVEX
12B7 FF FFBC       STX  CUROF  RESTORE FOR ASCII PR

* PRINT ASCII
12BA BD F02A       JSR  PSPACE
12BD BD F02A       JSR  PSPACE
12C0 C6 08        LDAB  #8
12C2 BD 135C  SHWP14 JSR  INBYTE  READ, INC CUROF
12C5 8D 1C        BSR  OUTASC
12C7 5A          DECB
12C8 26 F8        BNE  SHWP14

* CHECK IF DONE
12CA B6 FFC6       LDAA  TEMP
12CD F6 FFC7       LDAB  TEMP+1

* SAVEX CONTAINS OFFSET B E F O R E PRINT
12D0 FD FFC1       SUBB  SAVEX+1
12D3 B2 FFC0       SBCA  SAVEX
12D6 26 03        BNE  SHWP15  MSB NOT ZERO
12D8 5D          TSTB  TEST LSB
12D9 27 07        BEQ   SHWP90  DONE
12DB 7D FFBD  SHWP15 TST  CUROF+1
12DE 26 AF        BNE  SHWP10  END OF LINE
12E0 20 AA        BRA  SHWP8   END OF BLOCK
12E2 39          SHWP90 RTS

12E3 84 7F  OUTASC ANDA  $$7F
12E5 81 20          CMPA  #'
12E7 2D 04          BLT  ASC1
12E9 81 61          CMPA  $$61
12EB 2D 02          BLT  ASC3
12ED 86 2E  ASC1  LDAA  #'  NON-PRINT CHAR
12EF 7E F018  ASC3  JMP  OUTCH  OUT & RTS

12F2  SHWM  EQU  *  SHOW MEMORY CONTENTS
12F2  BD 13BA          JSR  IN3
12F5 25 63          BCS  SHWM99
12F7  BD 125D          JSR  PIAIN

```

```

12FA F6 FFBB          LDAB  CURADR+1
12FD C4 F8            ANDB  ##F8
12FF F7 FFBB          STAB  CURADR+1 MASK TO FULL LINE
1302 B6 FFBB          LDAA  EADR
1305 F6 FFBB          LDAB  EADR+1
1308 C4 F8            ANDB  ##F8
130A B7 FFC6          STAA  TEMP
130D F7 FFC7          STAB  TEMP+1
1310 BD F021 SHWM8    JSR   PCRLF
1313 BD F02D SHWM10  JSR   IFBRK
1316 27 43            BEQ   SHWM90
1318 BD F021          JSR   PCRLF
131B CE FFBA          LDX   #CURADR
131E BD F01E          JSR   OUT4HS   PRINT ADDRESS
1321 FE FFBA          LDX   CURADR
1324 C6 08            LDAB  #8
1326 BD 1473 SHWM12  JSR   OUT2HS   PRINT DATA
1329 5A              DECB
132A 26 FA            BNE   SHWM12
                * PRINT ASCII
132C BD F02A          JSR   PSPACE
132F BD F02A          JSR   PSPACE
1332 C6 08            LDAB  #8
1334 FE FFBA          LDX   CURADR
1337 A6 00 SHWM14  LDAA  X
1339 8D A8            BSR   OUTASC
133B 08              INX
133C 5A              DECB
133D 26 F8            BNE   SHWM14
                * CHECK IF DONE
133F B6 FFC6          LDAA  TEMP
1342 F6 FFC7          LDAB  TEMP+1   END ADDRESS
1345 F0 FFBB          SUBB  CURADR+1
1348 B2 FFBA          SBCA  CURADR
134B 26 03            BNE   SHWM15   MSB NOT ZERO
134D 5D              TSTB   TEST LSB
134E 27 0B            BEQ   SHWM90   DONE!
1350 FF FFBA SHWM15  STX   CURADR   UPDATE
1353 7D FFBB          TST   CURADR+1
1356 26 BB            BNE   SHWM10   END OF LINE
1358 20 B6            BRA   SHWM8    END OF BLOCK
135A 0B SHWM99  SEV
135B 39 SHWM90  RTS

                135C   INBYTE EQU *   GET ONE BYTE
135C 37              PSHB
135D 8D 05            BSR   SETADR
135F 33              PULB
1360 B6 FCE0          LDAA  DATA   READ A BYTE
1363 39              RTS

                1364   SETADR EQU *   SETUP ADDR AT PROM
                *   AND INC PROM ADDRESS
1364 B6 FFBC          LDAA  CUROF   PROM MSB ADDRESS
1367 BA FFC5          ORAA  BURN    BURN?
136A B7 FCE2          STAA  ADDR    -> PIA PORT B
136D C6 3C            LDAB  #%00111100
136F F7 FCE3          STAB  CTRLB   CB2 HIGH
1372 C4 F7            ANDB  #%11110111
1374 F7 FCE3          STAB  CTRLB   LATCH MSB
1377 B6 FFBD          LDAA  CUROF+1 PROM LSB
137A B7 FCE2          STAA  ADDR    -> PIA PORT B

```

```

* INCREMENT PROM ADDRESS
137D FF FFBE STX SLASK
1380 FE FFBC LDX CUROF CURRENT OFFSET
1383 08 INX
1384 FF FFBC STX CUROF
1387 FE FFBE LDX SLASK
138A 39 RTS

138B IN2 EQU * READ 2 ARGUMENTS
138B 17 TBA
138C BD 1095 JSR DELIM CHECK DELIMETER
138F 26 2F BNE ERRETN
1391 81 0D CMPA #CR
1393 27 1E BEQ RETN2
1395 BD 1445 JSR INADDR
1398 BD 1095 JSR DELIM
139B 26 23 BNE ERRETN
139D 8C 0000 CPX #0
13A0 27 1E BEQ ERRETN ZERO INPUT
13A2 FF FFB4 STX NBYTE OK!!
13A5 81 0D CMPA #CR
13A7 27 0A BEQ RETN2
13A9 BD 1445 JSR INADDR
13AC 81 0D CMPA #CR
13AE 26 10 BNE ERRETN
13B0 FF FFB6 STX OFST OK!!
13B3 FE FFB6 RETN2 LDX OFST
13B6 FF FFBC STX CUROF
13B9 39 RTS

13BA IN3 EQU * READ ARGUMENTS
13BA 17 TBA
13BB BD 1095 JSR DELIM
13BE 27 02 BEQ IN31 OK
13C0 0D ERRETN SEC ERROR
13C1 39 RTS
13C2 81 0D IN31 CMPA #CR
13C4 27 28 BEQ RETN NO CHANGE
13C6 BD 1445 JSR INADDR
13C9 BD 1095 JSR DELIM
13CC 26 F2 BNE ERRETN
13CE FF FF80 STX SADR OK!!
13D1 81 0D CMPA #CR
13D3 27 19 BEQ RETN
13D5 BD 1445 JSR INADDR
13D8 BD 1095 JSR DELIM
13DB 26 E3 BNE ERRETN
13DD FF FFB2 STX EADR OK!!
13E0 81 0D CMPA #CR
13E2 27 0A BEQ RETN
13E4 BD 1445 JSR INADDR
13E7 81 0D CMPA #CR
13E9 26 D5 BNE ERRETN
13EB FF FFB6 STX OFST OK!!
13EE B6 FFB2 RETN LDAA EADR
13F1 F6 FFB3 LDAB EADR+1
13F4 F0 FFB1 SUBB SADR+1
13F7 B2 FF80 SBCA SADR
13FA 25 16 BCS RETN1

* CHANGE NNN TO MEMORY SIZE
13FC CB 01 ADDB #1

```

13FE	89	00		ADCA	#0	
1400	B7	FFB4		STAA	NBYTE	
1403	F7	FFB5		STAB	NBYTE+1	
1406	FE	FFB6		LDX	OFST	
1409	FF	FFBC		STX	CUR0F	CURRENT OFFSET
140C	FE	FFB0		LDX	SADR	
140F	FF	FFBA		STX	CURADR	CURRENT ADDRESS
1412	39		RETN1	RTS		
	1413		SIZCHK	EQU	*	CHECK SIZE
1413	B6	FFB4		LDAA	NBYTE	
1416	F6	FFB5		LDAB	NBYTE+1	
1419	00	01		SUBB	#1	ADJUST
141B	82	00		SBCA	#0	
141D	20	11		BRA	SIZ2	ADD TO OFST
	141F		ADRCHK	EQU	*	CHECK BOUNDARIES
141F	B6	FFB2		LDAA	EADR	
1422	F6	FFB3		LDAB	EADR+1	
1425	F0	FFB1		SUBB	SADR+1	
1428	B2	FFB0		SBCA	SADR	
142B	B1	FFB8		CMPA	SIZE	HIBYTE
142E	2C	13		BGE	ADRERR	TO LARGE
1430	FB	FFB7	SIZ2	ADDB	OFST+1	
1433	B9	FFB6		ADCA	OFST	
* CHECK (EADR-SADR)+OFST <= \$800						
1436	B1	FFB8		CMPA	SIZE	
1439	2C	08		BGE	ADRERR	
143B	B7	FFC6		STAA	TEMP	
143E	F7	FFC7		STAB	TEMP+1	SAVE FOR SHWP
1441	0C			CLC		
1442	39			RTS		
1443	0D		ADRERR	SEC		
1444	39			RTS		
	1445		INADDR	EQU	*	INPUT HEX ADDR TO X
1445	CE	FFC6		LDX	#TEMP	
1448	6F	00		CLR	X	
144A	6F	01		CLR	1,X	CLEAR INPUT
144C	5F			CLRB		ZERO COUNT
144D	BD	F015	INAD10	JSR	INCH	READ A CHAR
1450	BD	F006		JSR	CBCDHX	CONVERT TO HEX
1453	2B	14		BMI	INAD70	NOT HEX
1455	37			PSHB		
1456	C6	04		LDAB	#4	
1458	68	01	INAD20	ASL	1,X	
145A	69	00		ROL	X	
145C	5A			DECB		
145D	26	F9		BNE	INAD20	
145F	33			PULB		
1460	AA	01		ORAA	1,X	
1462	A7	01		STAA	1,X	
1464	5C			INCB		
1465	C1	05		CMPB	#5	
1467	26	E4		BNE	INAD10	4 OR LESS CHAR
1469	81	18	INAD70	CMPA	#\$18	CTRL X?
146B	27	03		BEQ	INAD90	
146D	EE	00		LDX	X	CHAR READ
146F	39			RTS		
1470	7E	1024	INAD90	JMP	RESTAR	BREAK!!

```

      1473      OUT2HS EQU      *          2 HEX PLUS SPACE
1473 BD F01B      JSR      OUT2H
1476 7E F02A SP      JMP      PSPACE

1479 8D F8      02H2SP BSR      OUT2HS
147B 20 F9      BRA      SP

```

```

      147D      COMTAB EQU      *          COMPARE COMMANDS HERE
147D 50          FCC      /PROG/
1481 10A9      FDB      PROG
1483 52          FCC      /READ/
1487 113E      FDB      READ
1489 56          FCC      /VERF/
148D 1158      FDB      VERF
148F 43          FCC      /CHCK/
1493 11CD      FDB      CHCK
1495 53          FCC      /STAT/
1499 1244      FDB      STAT
149B 53          FCC      /SHWP/
149F 126D      FDB      SHWP
14A1 53          FCC      /SHWM/
14A5 12F2      FDB      SHWM
14A7 45          FCC      /EXIT/
14AB 10AD      FDB      EXIT
      14AD      ENDTAB EQU      *
      *

```

```

14AD 0DDA      MREADY FDB      NL
14AF 50          FCC      /PROM PROGRAMMER 1.0/
14C2 04          FCB      4
14C3 50      MPROMT FCC      /P /
14C5 04          FCB      4
14C6 57      MWHAT  FCC      /WHAT?/
14CB 04          FCB      4
14CC 32      MSTAT  FCC      /2716 ERASABLE PROM 2K X 8/
14E5 0DDA      FDB      NL
14E7 58          FCC      /XXXX YYYY NNN ZZZ/
14FA 0DDA      FDB      NL,4
14FE 4F      MVERFE FCC      /OFST PR MM /
150A 04          FCB      4
150B 53      MSIZE  FCC      /SIZE?/
1510 04          FCB      4
1511 43      MCHG   FCC      /CHANGE? /
1519 04          FCB      4
151A 4F      MCHCK  FCC      /OFST PR /
1523 04          FCB      4

```

```

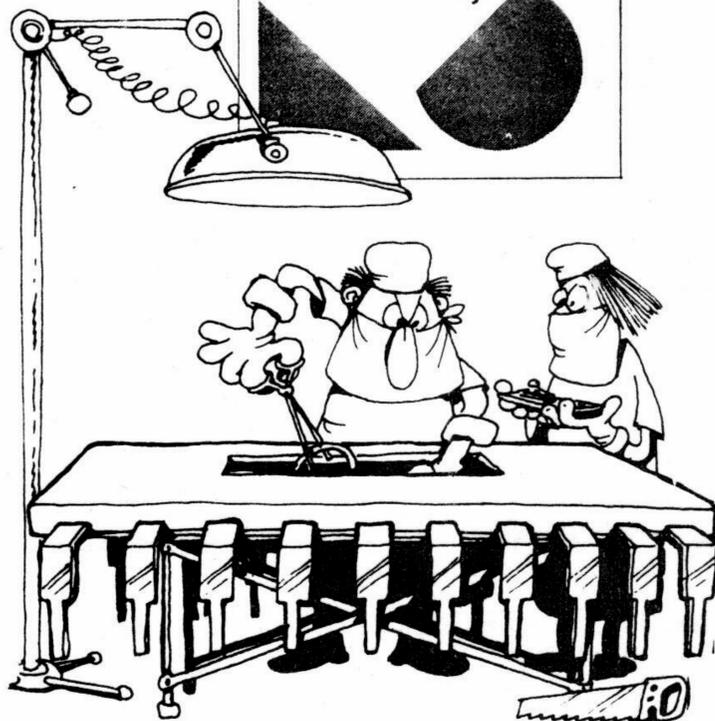
FFB0          ORG      $FFB0      STORAGE IN SCRATCH
FFB0 0002      SADR   RMB      2      START ADDRESS
FFB2 0002      EADR   RMB      2      END ADDRESS
FFB4 0002      NBYTE  RMB      2      NUMBER OF BYTES
FFB6 0002      OFST   RMB      2      OFFSET IN PROM
FFB8 0002      SIZE   RMB      2      MAX NUMBER OF BYTES
FFBA 0002      CURADR RMB      2      CURRENT MEMORY ADDRESS
FFBC 0002      CUROF  RMB      2      CURRENT PROM OFFSET
FFBE 0002      SLASK  RMB      2
FFC0 0002      SAVEX  RMB      2      TEMP FOR X

```



**MICROPROCESSOR  
INTERFACING  
TECHNIQUES**

THIRD EDITION  
AUSTIN LESEA  
RODNEY ZAKS



## MPU-laren HAR PROVAT SKRIVARE

En skrivare är en väsentlig del av datorsystemet och tyvärr också en ganska dyr sådan. Den som inte har någon skrivare märker förr eller senare att han eller hon måste anskaffa en. Skrivare finns i en mängd utföranden för olika användningsområden. Beroende på om man behöver skönskrift eller snabbskrift kan man välja bland typerna skönskrivare eller matris skrivare. Den senare typen arbetar vanligtvis med sju nålar och tecknen blir då i matris 5x7. Vissa matris skrivare har nio nålar och de två extra används då för underslängarna på de små bokstäverna g j p q och y.

För PD-68 medlemmarna har vi provat två olika matris skrivare vilka kostar runt 5.000 kr. De är Axiom IMP2 och Epson MX-80. Eftersom det vid provningen fanns tillgång till en Texas 810 har även denna tagits upp som jämförelseobjekt. Texas 810 kostar dock 3 ggr mer än de övriga och för denna prisskillnad får man endast en snabbare utskrift.

Axiom IMP2 har sju nålar och skriver i 5x7 matris. Den kan återge grafik med god upplösning eftersom varje nål kan styras. I fig 1 finns ett skriftprov från IMP2. Den skriver med 80, 96 eller 132 tecken per rad. Som standard finns parallellsnitt (Centronics), V24-snitt och strömloop. Överföringshastighet är inställbart mellan 110 och 1200 baud. Naturligtvis finns både Å, Ä och Ö med. I tabellen finns de uppmätta tiderna för olika typer av utskrifter. IMP2 skriver åt båda hållen. Skrivhuvudet går dock alltid en hel rad eftersom mekaniken är så konstruerad.

Epson MX-80 har nio nålar och skriver även den i 5x7 matris med undersläng på små bokstäver. Som standard finns fyra olika teckenuppsättningar, engelska, tyska, franska och svenska. Omkoppling sker med en DIP-omkopplare inuti printern. MX-80 har som standard Centronics-parallellsnitt med kan kompletteras för V24 eller IEEE-488-bus. MX-80 skriver även den åt båda hållen men skrivhuvudet flyttar sig bara så mycket som behövs. I fig 2 finns utskriftsprov och i tabellen finns skrivtiderna. Grafiken hos MX-80 är av samma konstruktion som hos teletextsystemet, dvs varje teckenposition motsvarar 2 x 3 punkter och varje punkt motsvarar en bit i ASCII-koden. MX-80 kan även skriva med fetstil och komprimerad text.

Både Epson och Axiom kan programmeras för tabulering och varierande radmatning.

Texas 810 är först och främst avsedd för snabba utskrifter. Den har inga underslängar på små bokstäver, och inga grafikmöjligheter.

Fyra olika utskriftsprov har gjorts. För att få samma förutsättningar vid mätningarna så har fyra OUT-filer skapats på en flexskiva. Dessa har sedan LIST-ats till printerutgången. Tiden är uppmätt från det att return-tangenten är nedtryckt till dess att de tre plustecknen (Flex identifiering) har kommit upp på bildskärmen igen, dvs den tid som datorn är upptagen.

Den första textfilen är en katalog av en flexskiva med 93 filer. Den andra är en utskrift av etiketter. Dessa båda är av typen lite text per rad. Den tredje textfilen är en textprocessad utskrift av ett protokoll från ett styrelsemöte. Denna textfil representerar mycket text per rad och blandat stora och små

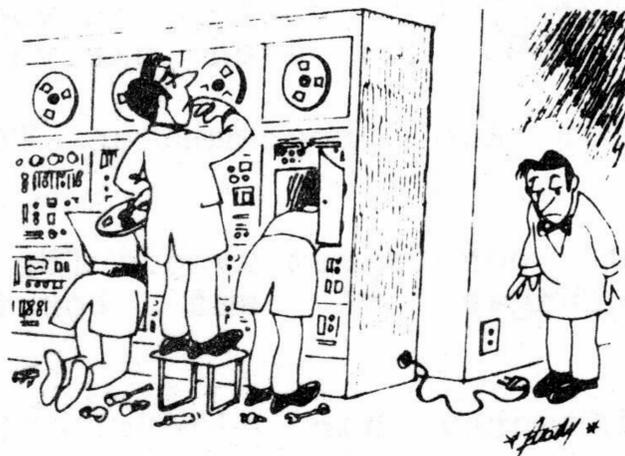
bokstäver. Den fjärde textfilen slutligen är en assemblerlista på 752 rader. Skrivtiden är som synes i stort densamma för IMP2 och MX-80 vid medelstora utskrifter (protokoll). Vid utskrift av lite text per rad är MX-80 nästan dubbelt så snabb som IMP2. Långa assemblerlistor blir något snabbare utskrivna med IMP2 eftersom skrivhuvudet går kontinuerligt. MX-80 stannar ett kort ögonblick efter varje rad och "tänker" innan den fortsätter. IMP2 har en stor teckenbuffert vilket gör att den skriver en stund efter det att datorn är ledig igen. Mellan 10 och 30 sekunder efter det att plustecknen har kommit tillbaka fortsätter IMP2 att skriva.

Vilken printer man skall välja av dessa beror på vilken sorts utskrift man gör mest. Om man vill ha en snygg skrift och inte har några krav på högupplösande grafik så kan man tryggt välja Epson. Vill man å andra sidan kunna göra grafik och skriva ut långa listor lite fortare väljer man IMP2. Båda printrarna har s.k. traktormatning och är alltså avsedda för papper på löpande bana. Till IMP2 medföljer dock hållare för pappersrulle.

### Tabell

#### TIDER FÖR OLIKA UTSKRIFTER:

Textfil	Texas 810 (2400 baud)	Axiom IMP2 (1200 baud)	Epson MX-80 (1200 baud)
Katalog	25 sek	1m 40s/2m 5 s	1m 8 s
Etiketter	1m 16 s	6m 2s/6m 34s	3m 54 s
Protokoll	1m 7s	3m 30s/3m 40s	3m 33s
Assemblerlista	5m 23s	13m 50s/14m 5s	16m 54s



# Epson MX-80 Print Sample

8.25 CPI Double Emphasized

ABCDEFGHIJKLMNOPQRSTUVWXYZ  
abcdefghijklmnopqrstuvwxyz  
1234567890: - ! # \$ % & ' ( ) \* = + < > ?  
EPSON MAKES MORE PRINT MECHANISMS  
Epson makes more print mechanisms

16.5 CPI Standard

ABCDEFGHIJKLMNOPQRSTUVWXYZABCDEFGHIJKLMNOPQRSTUVWXYZ  
abcdefghijklmnopqrstuvwxyzabcdefghijklmnopqrstuvwxyz  
12345678901234567890! # \$ % & ' ( ) \* = + < > ? ! # \$ % & ' ( ) \* = + < > ?  
EPSON MAKES MORE PRINT MECHANISMS THAN ANYONE ELSE IN THE WORLD.  
Epson makes more print mechanisms than anyone else in the world.

10 CPI Standard

ABCDEFGHIJKLMNOPQRSTUVWXYZABCDEFGHIJKLMNOPQRSTUVWXYZ  
abcdefghijklmnopqrstuvwxyzabcdefghijklmnopqrstuvwxyz  
12345678901234567890! # \$ % & ' ( ) \* = + < > ? ! # \$ % & ' ( ) \* = + < > ?  
EPSON MAKES MORE PRINT MECHANISMS THAN ANYONE ELSE IN THE WORLD.  
Epson makes more print mechanisms than anyone else in the world.

10 CPI Emphasized

ABCDEFGHIJKLMNOPQRSTUVWXYZABCDEFGHIJKLMNOPQRSTUVWXYZ  
abcdefghijklmnopqrstuvwxyzabcdefghijklmnopqrstuvwxyz  
12345678901234567890! # \$ % & ' ( ) \* = + < > ? ! # \$ % & ' ( ) \* = + < > ?  
EPSON MAKES MORE PRINT MECHANISMS THAN ANYONE ELSE IN THE WORLD.  
Epson makes more print mechanisms than anyone else in the world.

5 CPI Double Emphasized

ABCDEFGHIJKLMNOPQRSTUVWXYZ  
abcdefghijklmnopqrstuvwxyz  
1234567890: - ! # \$ % & ' ( ) \* = + < > ?  
EPSON MAKES MORE PRINT MECHANISMS  
Epson makes more print mechanisms

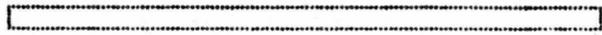
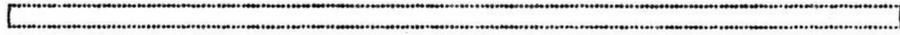
## The IMP 2 MiniPrinter from AXIOM

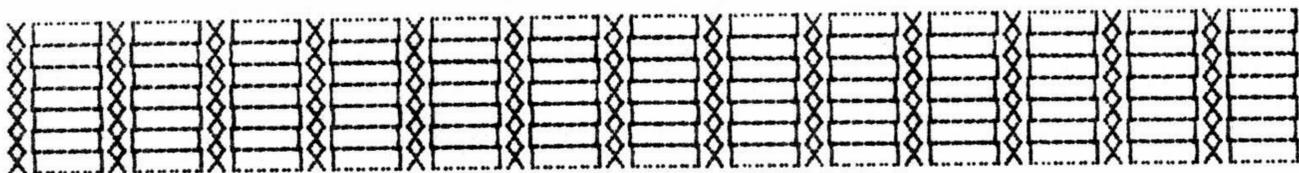
- 7x7 Dot Matrix; Plain paper, 3 Copies
- 8 Switch selectable Forms Lengths, and choice of Perforation Skip 80, 96, or 132 Columns, Software Selectable
- 60 Lines per minute Bidirectional
- 96 Character ASCII plus special characters by software

Double Sized Characters mixed on any line (This is 96 column)

Industry Standard Parallel & Serial & 20 mA LOOP Inputs All Included - many other features

FULL GRAPHICS included-

1980   
1981 



## TEXAS 810

## 4.0 Medlemsförmåner.

Bosse Ljungblad berättade att vi eventuellt kunde få en starkt rabatterad prenumeration av Modern Elektronik med Dator Nytt. Bosse kollar med Tore Rösnäs på ME. Bosse tyckte vidare att pengar kunde användas till aktivitetsbefrämjande ändamål. Exempelvis pristävlingar, kurser etc. Hur mycket pengar som kan spenderas när MPU-larens framställningskostnader undantagits, beror på klubb datorprojektet. Inget beslut togs därför i ämnet. Gratislitteratur kan vi få från olika datafirmor antydde Bosse. Vilket givetvis renderade honom uppdraget att kolla vad som finns av värde för medlemmarna.

## Axiom IMP2

## 4.0 Medlemsförmåner.

Bosse Ljungblad berättade att vi eventuellt kunde få en starkt rabatterad prenumeration av Modern Elektronik med Dator Nytt. Bosse kollar med Tore Rösnäs på ME. Bosse tyckte vidare att pengar kunde användas till aktivitetsbefrämjande ändamål. Exempelvis pristävlingar, kurser etc. Hur mycket pengar som kan spenderas när MPU-larens framställningskostnader undantagits, beror på klubb datorprojektet. Inget beslut togs därför i ämnet. Gratislitteratur kan vi få från olika datafirmor antydde Bosse. Vilket givetvis renderade honom uppdraget att kolla vad som finns av värde för medlemmarna.

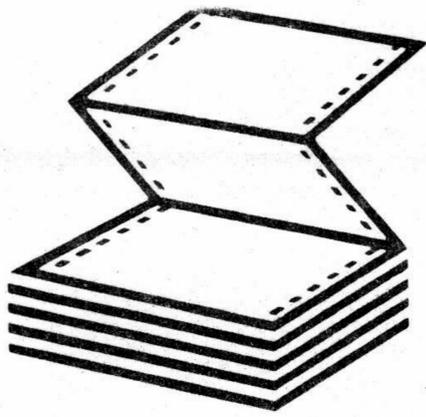
fig. 1

## Epson MX-80

## 4.0 Medlemsförmåner.

Bosse Ljungblad berättade att vi eventuellt kunde få en starkt rabatterad prenumeration av Modern Elektronik med Dator Nytt. Bosse kollar med Tore Rösnäs på ME. Bosse tyckte vidare att pengar kunde användas till aktivitetsbefrämjande ändamål. Exempelvis pristävlingar, kurser etc. Hur mycket pengar som kan spenderas när MPU-larens framställningskostnader undantagits, beror på klubb datorprojektet. Inget beslut togs därför i ämnet. Gratislitteratur kan vi få från olika datafirmor antydde Bosse. Vilket givetvis renderade honom uppdraget att kolla vad som finns av värde för medlemmarna.

fig. 2



# Datablankett Service AB

Slipgatan 9

Box 9043

102 71 STOCKHOLM. Tel. 08-69 08 45

## Tabulatorpapper

Vi har stort lager på dom mest aktuella typerna. Ni kan få dom otryckta eller linjerade.

Vi kan också erbjuda tab.-papper med Er egna logotype för en ringa merkostnad.

## Datablanketter

Vi hjälper Er gärna att rationalisera Ert behov av blanketter, order, följesedlar, fakturor m m.

## Telex, Printerrullar, Hållremsor

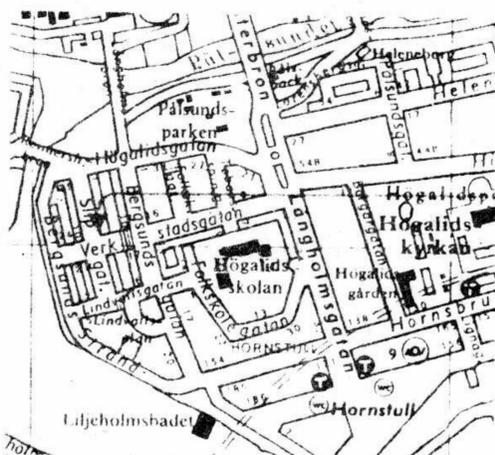
Från enkel bana upp till 10 blad, med karbon eller självkopierande.

## Trycksaker

Vi står till tjänst med att utforma och trycka Era trycksaker.

## Färgband

## Data-etiketter



Vill Du besöka oss är adressen:

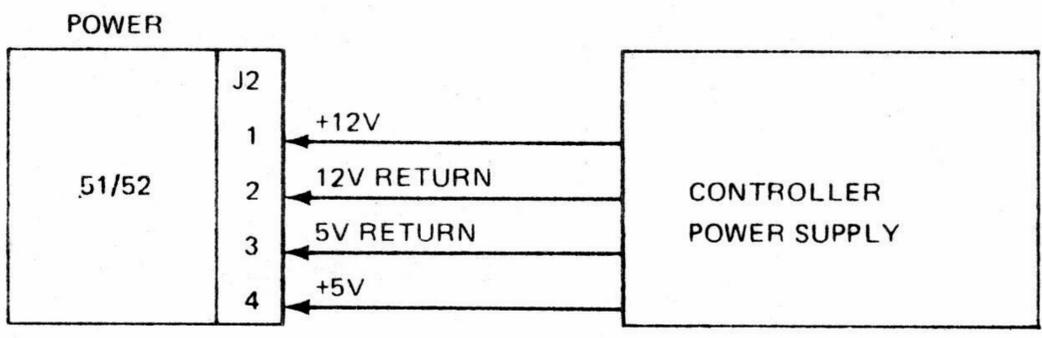
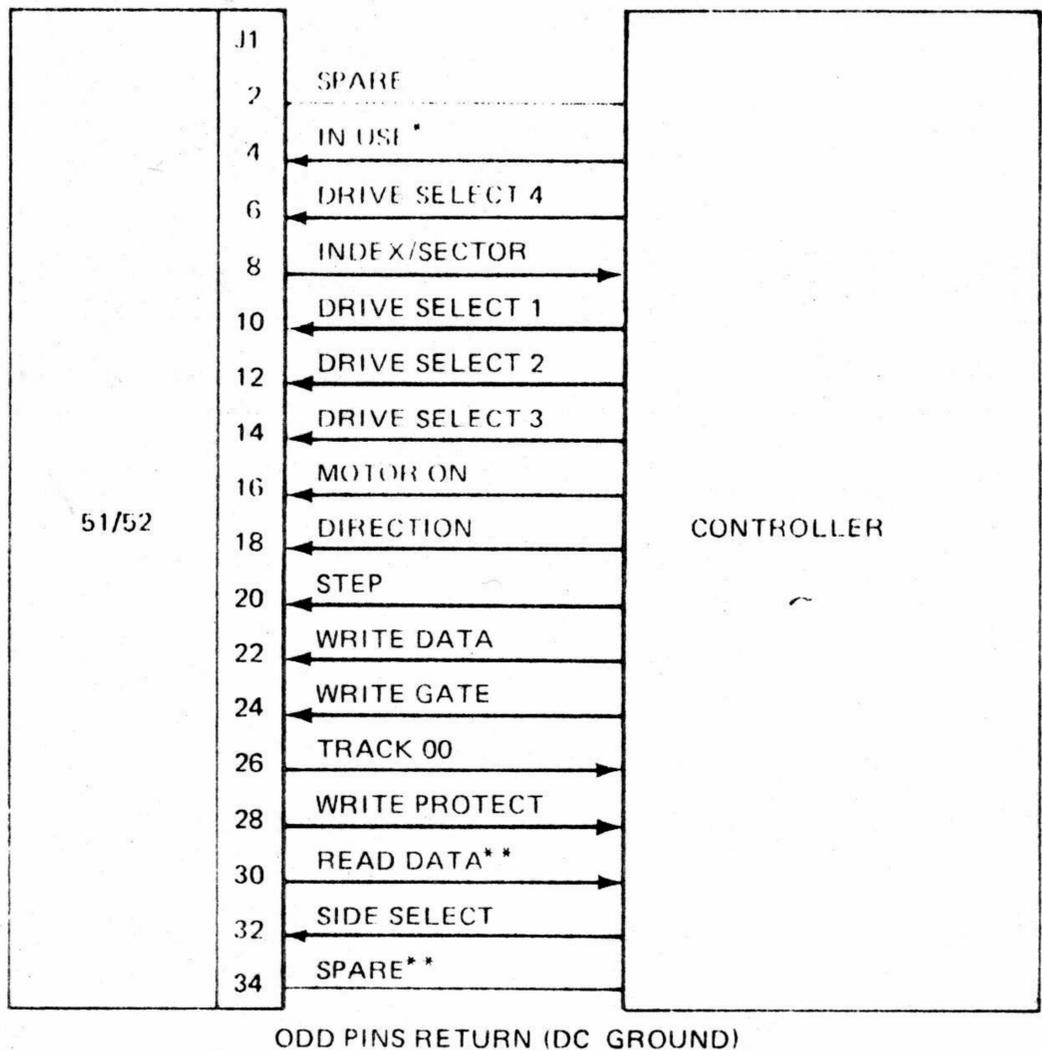
**DATABLANKETTSERVICE AB**

**SLIPGATAN 9**

Postadressen är:

**BOX 9043 — 10271 STOCKHOLM**

**08-69 08 45**



\* In use may be configured as Door Lock or Activity Light.  
 \*\* With the optional Data Separator installed, Pin 30 is Separate Clock and Pin 34 is Separate Data.

Figure 3-11 Interface Signals 51/52

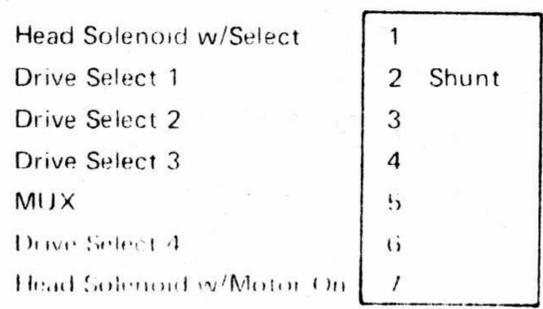
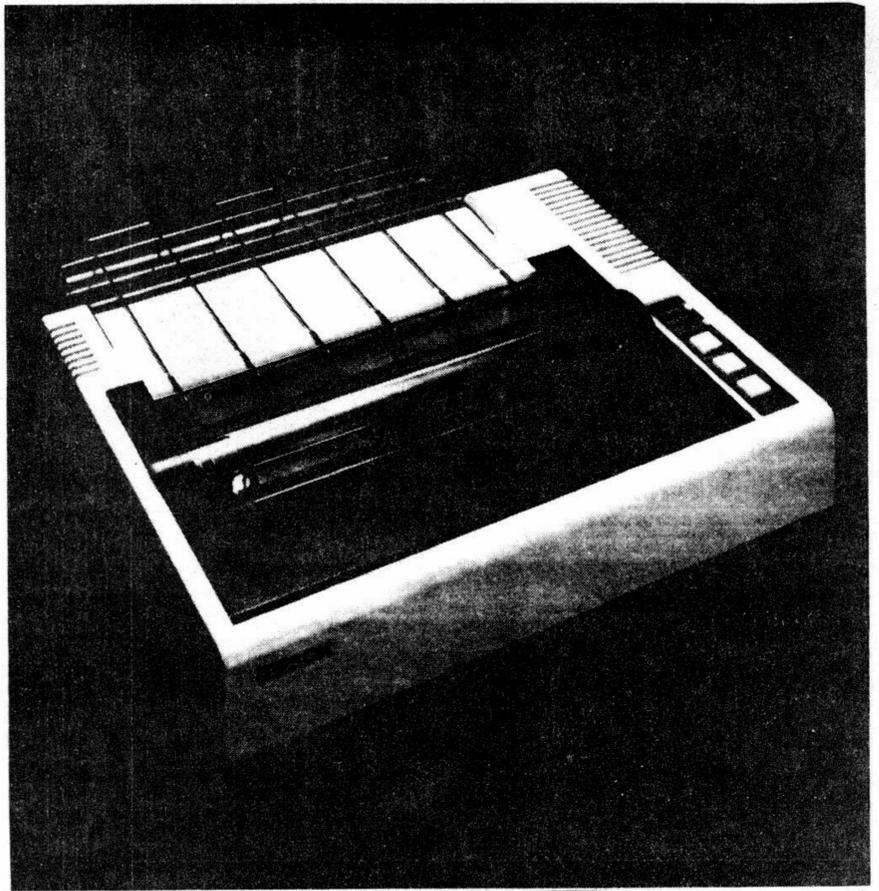


Figure 3-12 Shunt Configuration

# Epson

# MX-80



## Specifications

Print method ..... Serial impact dot matrix  
Print Rate ..... 80 CPS  
Print Direction ..... Bidirectional  
Number of Pins in  
  Head ..... 9  
Matrix ..... 9x9  
Line Spacings ..... 1/8", 1/6", 7/72", plus programmable  
Throughput at 10 CPI. Logical seeking function—  
  105 LPM, 20 character line;  
  73 LPM, 40 character line;  
  46 LPM, 80 character line.

### PRINTING CHARACTERISTICS

Character Set ..... Full 96-character ASCII with descenders  
Graphics Characters .. 64 block characters  
Printing Modes ..... Standard.  
  Double (advance paper 1/206th and  
  repeat line).  
  Emphasized (shift right and double  
  strike).  
  Double Emphasized (combination of  
  above).

PRINTING SIZES	Characters per inch	Max. Characters per line
Normal .....	10	80
Normal Expanded .....	5	40
Compressed .....	16.5	132
Compressed Expanded ..	8.25	66

### FORMS HANDLING

Line Feed ..... Programmable length 1 to 85/72nds  
Form Feed ..... Programmable length to 66 lines  
Horizontal Tab ..... To 112 positions  
Vertical Tab ..... To 64 positions

### MEDIA HANDLING

Paper Feed ..... Adjustable tractor-type pin feed  
Paper Width Range ... 4" to 10"  
Number of Parts ..... 3  
Paper Path ..... Rear

### INTERFACES

Standard ..... Centronics-style 8-bit Parallel  
Optional ..... RS232, IEEE488  
Buffer Size ..... 1 line

### SWITCHES/LIGHTS/DETECTORS

Indicators ..... Power Light; Printer Ready; Paper  
  Out; On Line  
Switches ..... Power On/Off; On Line; Form Feed;  
  Line Feed  
Detectors ..... Internal buzzer (bell) responds to  
  Paper Out and error conditions with a  
  periodic 3-second tone for 30 seconds.

### RELIABILITY

Print Head Life  
  Expectancy ..... 50 to 100 x 10<sup>6</sup> characters  
MCBF (Excluding Print  
  Head) ..... 5 million lines

### INKED RIBBON

Color ..... Black  
Type ..... Cartridge  
Life Expectancy ..... 3 Million characters

### ENVIRONMENTAL CONDITIONS

Operating Temperature  
  Range ..... 41°F to 95°F  
Operating Humidity .. 10 to 80% non-condensing

### POWER REQUIREMENT

Voltage ..... 115V, 60Hz  
Current ..... <1 amp  
Power Consumption .. 100 VA maximum

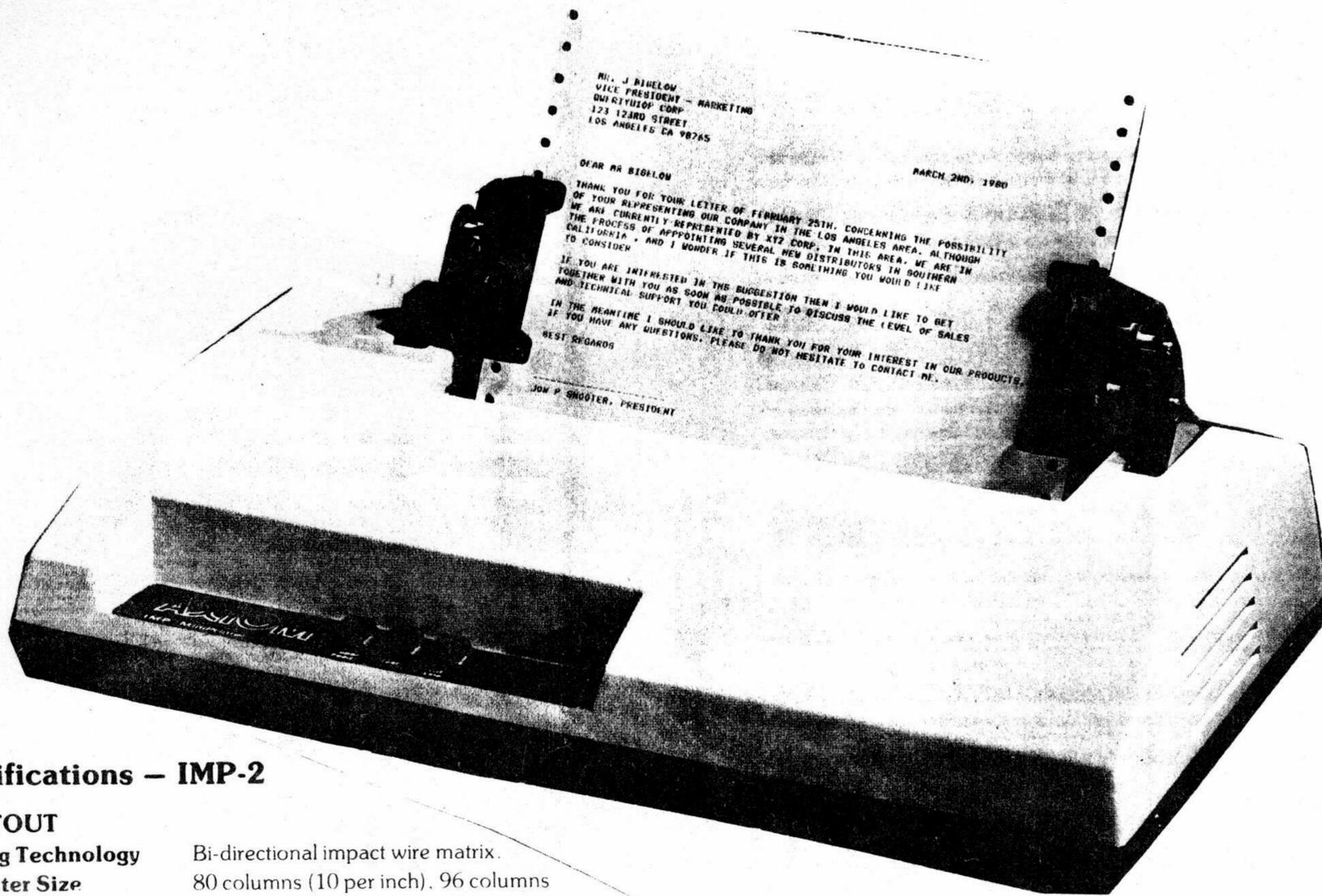
### SELF TEST MODE

Depressing Line Feed Switch while turning power ON engages  
self-test which prints all characters in ROM.

### PHYSICAL CHARACTERISTICS

Height ..... 4.2"  
Width ..... 14.7"  
Depth ..... 12.0"  
Weight ..... 12 lbs.

Specifications subject to change without notice.



## Specifications – IMP-2

### PRINTOUT

#### Printing Technology

Bi-directional impact wire matrix.

#### Character Size

80 columns (10 per inch), 96 columns (12 per inch), 132 columns (16.5 per inch), plus expanded printout in each size.

#### Vertical Line Spacing

6 or 12 lines per inch, switch/software selectable, allows subscripts, superscripts and graphics.

#### Printing Speed

Up to 60 lpm throughput

#### Standard Character Set

96 character ASCII.

#### Optional Character Set

User's 96 character set on 7 x 7 dot matrix may be inserted in place of standard character set. External character set also available (C128 option).

### PAPER HANDLING

#### Tractor Feed

Adjustable tractors for all forms widths between 1.5 inches (40 mm) and 9.5 inches (240 mm).

#### Paper Roll

Continuous printing on standard teleprinter or similar paper roll.

#### Single Sheet

Simple loading of letterhead or any single sheet up to 8.5 inches (216 mm) wide.

#### Copies

Original plus two copies.

#### Paper Advance

Single line and high speed advance of all types of paper.

### FORMS CONTROL

#### Switch Selectable

8 common pre-programmed forms lengths are switch-selectable (3.5", 4", 5.5", 8", 8.5", 11", 12", 14").

#### Software Selectable

Forms length can be user-programmed in half line increments up to 127.5 lines.

#### Top of Form

Set by front panel push button.

#### Form Feed

Paper advances to next top of form on receipt of software or front panel command

#### Auto Pagination

Switch selection of automatic 1 inch skip at end of form.

### GRAPHICS

#### Graphics Type

Dot addressable by 7 X N bit horizontal raster scan. Prints up to 3300 dots per second.

#### Resolution

Horizontal - 50 or 68 dots per inch.  
Vertical - 72 dots per inch.

### INPUT BUFFER

#### Type

Multi-line asynchronous

#### Standard Size

512 characters.

40 MPU-laren Nr3 1980

### IMP 2 DATA INPUT

(Both Parallel and Serial inputs are provided as standard)

#### PARALLEL INPUT

##### Connector Type

36-pin ribbon connector "Centronics" type

##### Signal Levels

Parallel ASCII. All low power TTL compatible. Standard - 7 bits ASCII. With expanded character set - 8 bits (bit 8 = Ø for ASCII printout without expanded character set).

##### Byte Length

Min 500 nsec.

##### Strobe (STR)

##### Acknowledge (ACK)

TTL pulse 12.5 µsec wide after each data entry is accepted.

##### Busy (BSY)

TTL high level whenever input not ready for data.

#### SERIAL INPUT

##### Connector Type

25-pin connector DB-25S.

##### Signal Specifications

Serial RS232C & 20mA current loop built in as standard.

##### Baud Rates

Standard - Normally used rates between 50 and 1200 baud. Switch selectable.

Available High Speed Option - 15 switch-selectable rates to 9600 baud.

##### Byte Length

7 or 8 bits.

##### Parity

Standard - None tested

High Speed Option - Selectable, odd, even or none.

##### Buffer Over-Run (Double buffered characters)

Standard - Pin 20 (DTR) goes to -V low when buffer is not ready for next character.

### PHYSICAL

#### Dimensions

17.5 inches (445 mm) wide by 8.75 inches (222 mm) deep by 6.0 inches (152 mm) high (includes tractors).

#### Weight

14 pounds (6.36 kg).

#### Electrical

105 V to 125 V AC or 200 V to 260 V AC, 50, 60 Hz, 100 VA maximum power. Forced air cooling.

#### Operator Controls

On/Off switch, Paper Advance (2 speeds), Paper Release, On/Off Line, Test, Top of Form.

## E N K Ä T - - - M O D E M

Undertecknad är i färd med att utveckla ett s.k. korthållsmodem för hobbydatorer. Det är alltså ett modem enligt CCITT för tvåtrådsförbindelser över 50 m. Av diverse orsaker så kommer modemmet att uppfylla de krav som televerket ställer på 300 bauds modem. Det kommer dock inte att lämnas in till verket för godkännande p.g.a. kostnadsskäl. Modemet får det mekaniska utförandet av ett kretskort 114 x 165 mm med en 22-polig kortkontakt för anslutning till 6800-databus. På kortet finns en ACIA, baudrategenerator, modemkrets samt en uppsättning aktiva filter.

Tanken är också den att låta tillverka färdiga kretskort för PD68-medlemmar. Denna konstruktion kommer inte att publiceras i Radio & Television. Förutsättningen för kretskortstillverkning är att minst 25 medlemmar anmäler sitt intresse för detta modem. Kompletta satser till modemmet kommer att säljas av undertecknad till självkostnadspris. Modemet innehåller en del specialkretsar, vilka för närvarande har lång leveranstid. Det är därför värdefullt om alla intresserade meddelar mig så snart som möjligt, helst före den 15 november. En komplett sats beräknas kosta 510 kr exkl. moms. Priset kan variera något beroende på vilka kretsar som kan levereras (plast el. keramik).

*Ake Holm*

---

Sänd in kupongen till Åke Holm, Långsjövägen 15 B. 135 54 TYRESÖ

Ja, jag är intresserad av korthållsmodemet.

- Sänd en komplett byggsats för ca 510 kr exkl. moms.
- Sänd kretskort, special (MOS)-kretsar och 1%-motstånd
- Sänd enbart kretskort (ca 55 kr exkl. moms)

Namn: \_\_\_\_\_

Adress: \_\_\_\_\_

Postnr: \_\_\_\_\_ Postadress: \_\_\_\_\_

Telefon: arb: \_\_\_\_\_ - \_\_\_\_\_ bost: \_\_\_\_\_ - \_\_\_\_\_