BYTEMOVER

SOFTWARE SUPPLIED WITH THE BYTESAVER

CROMEMCO
ONE FIRST STREET
LOS ALTOS, CALIFORNIA
94022

BYTEMOVER 3.0 OPERATING INSTRUCTIONS

THE CROMEMCO BYTESAVER™ is supplied with BYTEMOVER™ software. This software is pre-programmed into the 2704 PROM that comes with the BYTESAVER.

The 2704 PROM containing the Bytemover software is normally inserted into PROM location 0 on the Bytesaver board. The Bytemover software can be used to program a PROM in any of the PROM locations on the Bytesaver board. The Bytemover software can also be used to transfer programs from PROM to RAM. The operation of the Bytemover software is controlled by the setting of the front panel sense switches on the Altair computer. To use the Bytemover software there must be a RAM board in the Altair beginning at location zero in memory; further, this RAM board must be unprotected for proper execution of the Bytemover software.

STEP-BY-STEP INSTRUCTIONS

1) Before using the Bytesaver you must install three jumper wires to set the location of the Bytesaver in memory space. This is shown in Figure 1. The assembled Bytesaver comes with A13, A14, and A15 each tied to the corresponding "Hi" pad to position the board at the very top of memory. In the following instructions it is assumed that this is the jumper connection used.

2) With the Altair 8800 power turned off, plug the Bytesaver board into the computer.

3) Be sure that the program power on the Bytesaver is turned OFF (program power switch in the down position.)

4) Turn on the Altair. Raise the reset switch, then raise the stop switch, and then raise the reset switch once again to initialize the Altair.

5) Raise address switches A15, A14, and A13. All other address switches should be down.

6) Raise the examine switch. You are now examining the contents of the first byte of PROM in PROM location zero of the Bytesaver memory board (memory location 340 000). If the PROM supplied with your Bytesaver is in this PROM location the data lights will read "061", the first byte of the Bytemover program.

EXAMPLE: Transfer the Bytemover program from PROM to RAM beginning at location zero in RAM.

1) Raise the reset switch.

2) Depress the unprotect switch (on the Altair front panel).

3) Raise A15, A14, and A13. Raise the examine switch. The data lights should read "061" octal.

4) Now set the sense switches for the task to be done, refering to Fig. 2.

A15 - Down to transfer from PROM to RAM
A14 - Down for the transfer of 1K bytes.
A13 - Down
A12 - Down All down since we are transferring from the same PROM that contains BYTEMOVER (PROM 0)
A11 - Down
A10 - Down
A9 - Down All down for storage to begin at location zero in RAM.
A8 - Down

5) Push the run switch. In less than one second the contents of PROM will be transferred to RAM. (Of course the contents of the PROM are unaffected by this operation.

6) Raise the STOP switch.

7) Raise the reset switch. Note that the data lights read "061".

EXAMPLE: Program a 2708 PROM inserted in PROM location 1. This PROM is to be programmed with the contents of the first 1K bytes of RAM beginning at location zero in memory. The Bytemover software is still in the PROM in PROM location zero on the Bytesaver board.

1) Raise the reset switch.

2) Depress the unprotect switch (on the Altair front panel)

3) Raise A15, A14, and A13. Raise the examine switch. The data lights should read "061" octal.

4) Raise the protect switch on the Bytesaver board (i.e. program power switch to the ON position). The protect light on the Altair front panel should go off when this switch is raised.

5) Now set the sense switches for the task to be done:

A15 - Up to program a PROM
A14 - Down (always down for PROM programming)
A13 - Down
A12 - Down To select the PROM 1K higher in memory than the PROM that contains BYTEMOVER
A11 - Up
A10 - Down
A9 - Down All down for transfer to begin at location zero in RAM.
A8 - Down

6) Push the RUN switch. Note that panel light A9 is blinking at a rate of about twice per second. When this light stops blinking the PROM programming is complete.

7) Raise the STOP switch.

8) Now note the INTE light on the Altair front panel. If this light is on, the BYTEMOVER VERIFIER has verified that the contents of the programmed PROM are indeed identical to the contents of the selected 1K bytes of RAM. If this light is off, the PROM has not programmed correctly; this could be due, for example, to a defective PROM.

EXAMPLE: Altair 8K BASIC can be stored in seven 2708 PROMs. Given that these seven PROMs are in PROM locations 1 through 7 on the BYTESAVER board, 8K BASIC can easily be transferred into RAM using the following procedure:

1) Raise the RESET switch.

2) Depress the unprotect switch (on the Altair front panel).

3) Raise A15, A14, and A13. Raise the examine switch. The data lights should read "061" octal.

4) Now set the sense switches for the task to be done:

A15 - Down to transfer from PROM to RAM.
A14 - Up for a 7K transfer
A13 - Down
A12 - Down
A11 - Up
} To begin transfer from the PROM 1K higher in memory than the BYTEMOVER program.
A10 - Down
A9 - Down
A8 - Down
} All down for storage to begin at location zero in RAM.

5) Push the RUN switch. In less than one second BASIC will be loaded into RAM (it sure beats paper tape!). Raise the STOP switch.

EXAMPLE: If you do not have BYTEMOVER in PROM, you can program a PROM with BYTEMOVER that is stored in RAM. The BYTEMOVER software (a listing of which is attached) must first be loaded into RAM beginning at location zero in memory. The BYTEMOVER software can then be burned into a PROM using the following procedure:

1) Raise the reset switch.

2) Depress the unprotect switch (on the Altair front panel).

3) Insert an erased PROM into PROM location 0 on the BYTESAVER board.

4) Examine location 000 240 in memory.

5) Raise the program power switch on the BYTESAVER board.

6) Set the sense switches with A15 and A14 and A13 up.

7) Push the RUN switch. When light A9 stops blinking the programming is complete. The INTE light will be on to verify correct programming.

8) Turn off PROM program power by depressing the switch on the BYTESAVER.

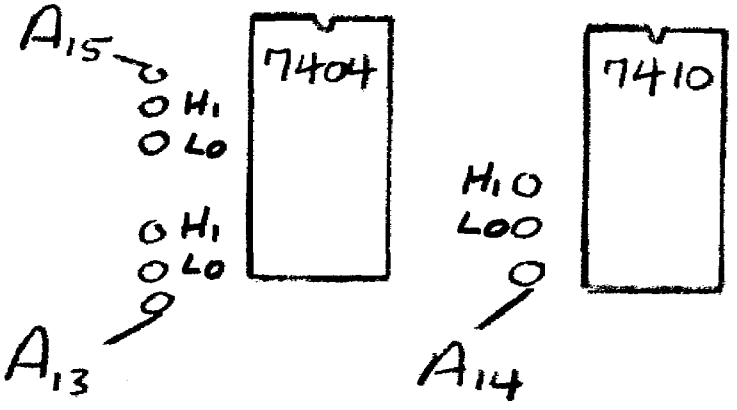
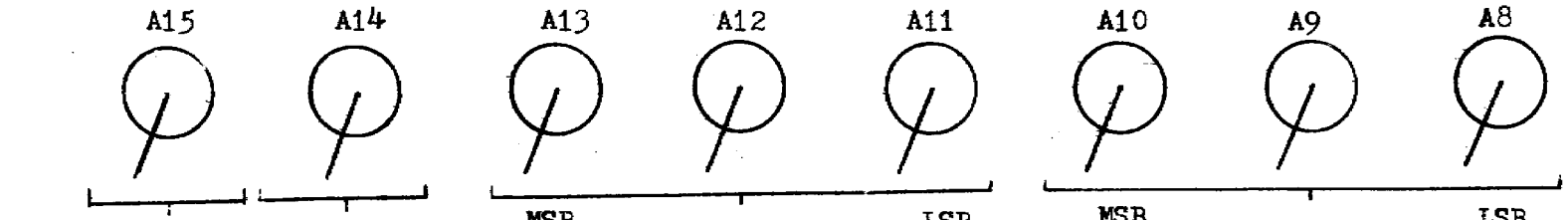


Fig. 1. How to set the Bytesaver address in memory. The built Bytesaver comes with A15, A14, and A13 connected to the corresponding "Hi" terminals so that memory address occurs when these three bits are high. Any or all of these address lines may be connected to the corresponding "Lo" terminal to move the memory board lower in memory. There are thus eight positions in memory that this board can be used.



UP - to program a PROM.

UP-for 7K transfer

DOWN - to move from PROM to RAM

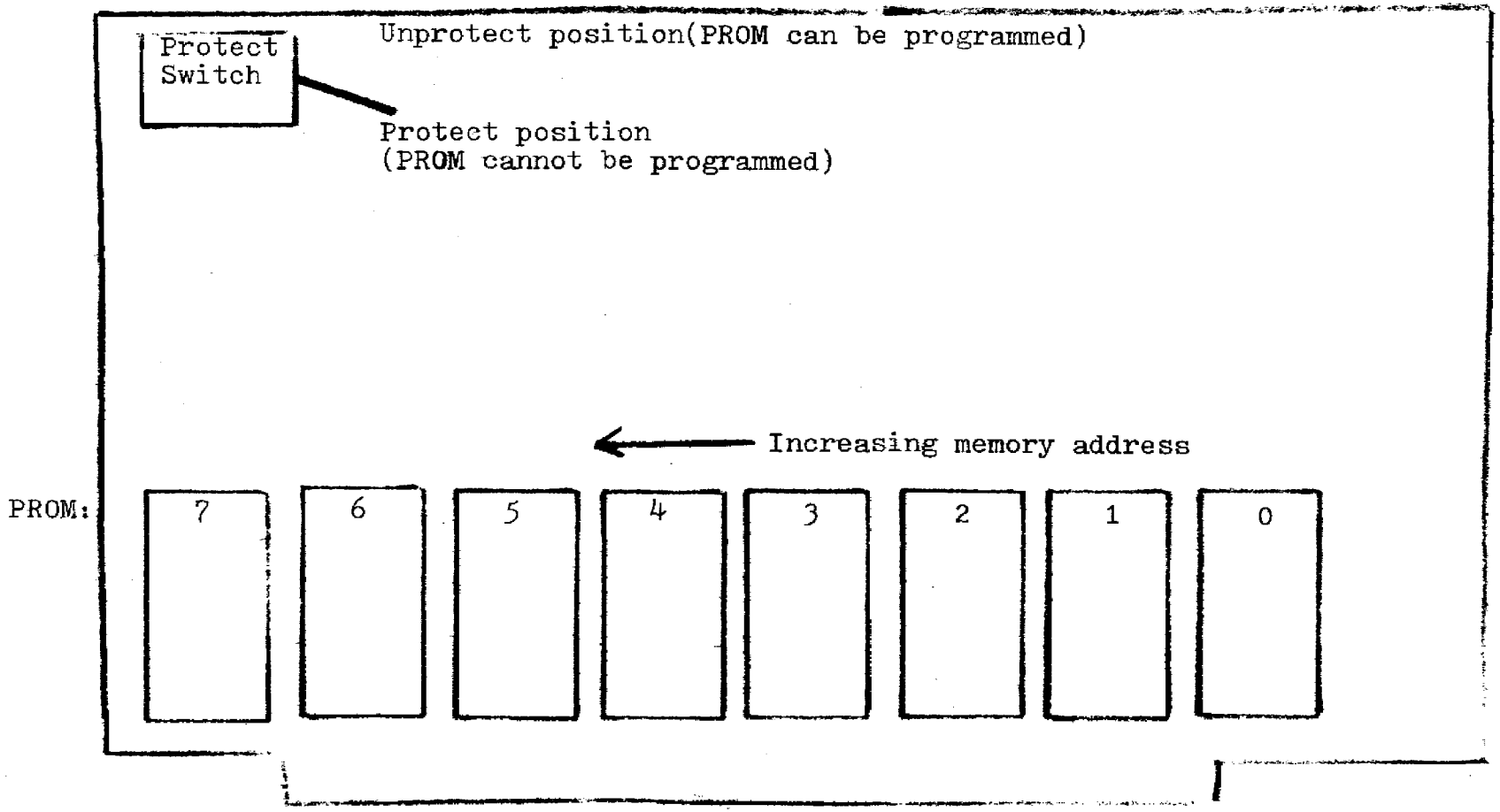
DOWN- for 1K transfer

PROM address location in increments of 1K from the PROM in which BYTEMOVER is stored

Selection of RAM address in 1K increments

FIGURE 2. FUNCTION OF THE SENSE SWITCHES IN THE BYTEMOVER PROGRAM.

Fig. 3. Bytesaver physical layout.



```
0000                    0000 * BYTEMOVER (T.M.) SOFTWARE FOR
0000                    0001 * CROMEMCO 8K BYTESAVER (T.M.)
0000                    0002 * VERSION 3.0
0000                    0003 * SELF-RELOCATING SOFTWARE LOCATABLE AT AN
0000                    0004 * 1024 BYTE (1K) BOUNDARY IN MEMORY
0000                    0009 * ROUTINE TO FIND ONESELF IN MEMORY
0000                    0010 SP EQU 6
0000                    0019 * DEFINE FIRST 4 BYTES IN MEMORY AS STACK
0000 31 00 00           0020  LXI SP, 0
0003                    0029 * SAVE FIRST FOUR BYTES IN REGISTERS
0003 C1                 0030  POP B
0004 D1                 0040  POP D
0005                    0049 * REPLACE BYTE 0 WITH A 'RETURN'
0005 3E C9              0050  MVI L, 0C9H
0007 00                 0051  NOP
0008 E5                 0060  PUSH H
0009 E5                 0070  PUSH H
000A 00                 0080  NOP
000B 00                 0081  NOP
000C 00                 0082  NOP
000D 31 04 00           0090  LXI SP, 4
0010 CD 00 00           0100  CALL 0
0013                    0101 * ROM LOCATION NOW IN BYTE 3
0013 31 02 00           0110  LXI SP, 2
0016 E1                 0120  POP H
0017                    0129 * RETURN BYTES 0-3
0017 31 04 00           0130  LXI SP,4
001A D5                 0140  PUSH D
001B C5                 0150  PUSH B
001C                    0159 * STORE ROM LOCATION IN SP
001C F9                 0160  SPHL
001D 0E 00              0170  MVI C, 0
001F 59                 0180  MOV E,C
0020 69                 0190  MOV L,C
0021                    0199 * INPUT SENSE SW COMMANDS
0021 DB FF              0200  IN 255
0023 57                 0210  MOV D,A
0024                    0219 * STRIP RAM ADDRESS
0024 E6 07              0220  ANI 7
0026 07                 0230  RLC
0027 07                 0240  RLC
0028                    0249 * STORE RAM ADDRESS IN BC
0028 47                 0250  MOV B,A
0029 7A                 0260  MOV A,D
002A                    0269 * STRIP ROM ADDRESS
002A E6 38              0270  ANI 56
002C 0F                 0280  RRC
002D 00                 0290  NOP
002E 67                 0300  MOV H,A
002F 39                 0310  DAD SP
```

```
0030 2E 00              0320  MVI L, 0
0032 7A                 0330  MØV A,D
0033 EB                 0340  XCHG
0034                    0341 * ADDRESS ØF RØM BEING PRØCESSED IN DE
0034                    0349 * BRANCH TØ TRANSFER ØP PRØGRAM RØUTINE
0034 E6 80              0350  ANI 128
0036 0F                 0360  RRC
0037 0F                 0370  RRC
0038 06 2D              0380  ADI 45
003A 21 00 00           0390  LXI H, 0
003D 6F                 0400  MØV L,A
003E 39                 0410  DAD SP
003F E9                 0420  PCHL
0040                    0500 * RØUTINE TØ TRANSFER RØM TØ RAM
0040 F9                 0510  SPHL
0041 21 0B 00           0520  LXI H, 11
0044 39                 0530  DAD SP
0045 EB                 0550  XCHG
0046 E9                 0560  SPHL   STACK CONTAINS RØM LØCATIØN
0047 EB                 0570  XCHG   H&L CØNTAIN LØØP ADDRESS
0048 11 00 00           0580  LXI D, 0
004B                    0588 * START ØF TRANSFER LØØP
004B                    0589 * INCREMENT RØM ADDRESS
004B 3B                 0590  DCX SP
004C                    0599 * MØVE DATA FRØM RØM TØ RAM
004C F1                 0600  PØP 6
004D 02                 0610  STAX B
004F                    0619 * INCREMENT RAM ADDRESS
004F 03                 0620  INX B
004F                    0629 * INCREMENT BYTE CØUNT
004F 13                 0630  INX D
0050 7A                 0640  MØV A,D
0051 E6 04              0650  ANI 4
0053 07                 0660  RLC
0054 07                 0670  RLC
0055 00                 0680  NØP
0056 85                 0690  ADD L
0057 6F                 0700  MØV L,A
0058 E9                 0710  PCHL
0059 00                 0716  NØP
005A 00                 0717  NØP
005B                    0719 *  JUMP TØ 00B1 FRØM TRANSFER RØUTINE
005B 3E 56              0720  MVI A, 56H
005D 85                 0725  ADD L
005E 6F                 0730  MØV L,A
005F E9                 0740  PCHL
0060                    1000 * RØUTINE TØ PRØGRAM RØM
0060 00                 1010  NØP
0061                    1019 * MØVE RAM ADDRESS INTØ HL
0061 69                 1020  MØV L,C
0062 7C                 1030  MØV A,H
0063 60                 1040  MØV H,B
0064                    1049 * MØVE RAM ADDRESS INTØ SP
0064 F9                 1050  SPHL
0065 57                 1060  MØV H,A
0066 3E 68              1070  MVI L, 107
0068                    1079 * INCREMENT RAM ADDRESS
0068 01 00 00           1080  LXI B, 0
006B                    1089 * INCREMENT RAM ADDRESS
```

```
006B 3B                 1090  DCX SP
006C                    1098 * USE STAX AND POP 6 (PSW)
006C                    1099 * TO MOVE DATA FROM ROM TO RAM
006C F1                 1100  POP 6
006D 12                 1110  STAX D
006E                    1119 * INCREMENT ROM ADDRESS
006E 13                 1120  INX D
006F                    1129 * INCREMENT BYTE COUNT
006F 03                 1130  INX B
0070                    1138 * B STORES TWO CONSTANTS
0070                    1139 * # COMPLETE PASSES & IN ROM CNT
0070 78                 1140  MOV A,B
0071                    1149 * # PASSES = 32 ?
0071 FE FC              1150  CPI 252
0073 3F                 1160  CMC
0074 1F                 1170  RAR
0075 1F                 1180  RAR
0076                    1198 * SET 64 TO 0 FOR TWO MINUTE TIMER VERSION
0076 E6 40              1200  ANI 64
0078                    1201 * A=64 IF COMPLETED 32 PASSES
0078 2E 7D              1205  MVI L, 7DH
007A 85                 1210  ADD L
007B 6F                 1220  MOV L,A
007C E9                 1225  PCHL
007D 2E 6B              1226  MVI L, 6BH
007F 78                 1230  MOV A,B
0080 E6 04              1240  ANI 4
0082                    1241 * A=4 IF END OF 1024 BYTE PASS
0082 07                 1250  RLC
0083 07                 1260  RLC
0084 07                 1270  RLC
0085 85                 1280  ADD L
0086 6F                 1290  MOV L,A
0087                    1291 * GO BACK TO 1090 UNLESS OVERFLOW
0087                    1292 * THEN GO TO 1380 FOR
0087                    1293 * ADDRESS SUBTRACTION
0087                    1294 * OR 2135 FOR OUTS
0087 E9                 1300  PCHL
0088 00                 1350  NOP
0089 00                 1360  NOP
008A 00                 1370  NOP
008B                    1378 * ANOTHER PROGRAM PASS TO BE DONE
008B                    1379 * ADJUST ROM AND RAM ADDRESSES
008B 7C                 1380  MOV A,H
008C 21 00 FC           1390  LXI H, 64512
008F                    1399 * SUBTRACT 1024 FROM ROM ADDRESS
008F 39                 1400  DAD SP
0090 F9                 1410  SPHL
0091 21 00 FC           1420  LXI H, 64512
0094                    1429 * SUBTRACT 1024 FROM RAM ADDRESS
0094 19                 1430  DAD D
0095 EB                 1440  XCHG
0096 67                 1450  MOV H,A
0097 2E 6B              1460  MVI L,107
0099 78                 1470  MOV A,B
009A E6 F8              1480  ANI 248
009C                    1489 * INCREMENT PASS COUNTER BY ONE
009C C6 08              1490  ADI 8
009E 47                 1495  MOV B,A
009F                    1499 * GO BACK TO 1090
009F E9                 1500  PCHL
```

```
00A0                    2000 * ROUTINE TO LOAD BYTEMOVER INTO ROM
00A0 DB FF              2010  IN 255
00A2 47                 2020  MOV B,A
00A3 E6 E0              2030  ANI 224
00A5 1E 00              2040  MVI E, 0
00A7 48                 2050  MOV C,E
00A8 57                 2060  MOV D,A
00A9 78                 2070  MOV A,B
00AA E6 1F              2080  ANI 31
00AC 47                 2090  MOV B,A
00AD 67                 2100  MOV H,A
00AE 2E 60              2110  MVI L, 96
00B0 E9                 2120  PCHL
00B1                    2121 * CHECK FOR 7K TRANSFER OF ROM TO RAM
00B1 C6 1A              2122  ADI 1AH
00B3 6F                 2123  MOV L,A
00B4 DB FF              2124  IN 255
00B6 E6 40              2125  ANI 64
00B8 0F                 2126  RRC
00B9 0F                 2127  RRC
00BA 85                 2128  ADD L
00BB 6F                 2129  MOV L,A
00BC E9                 2130  PCHL
00BD                    2133 * PROGRAMMER VERIFICATION ROUTINE
00BD                    2134 * PART 1
00BD 7C                 2135  MOV A,H
00BE 21 00 FC           2145  LXI H, 64512
00C1 39                 2155  DAD SP
00C2 F9                 2165  SPHL
00C3 2E CD              2175  MVI L, 0CDH
00C5 67                 2185  MOV H,A
00C6 E9                 2195  PCHL
00C7 00                 2205  NOP
00C8 00                 2210  NOP
00C9 00                 2215  NOP
00CA 00                 2220  NOP
00CB                    2229 * ROM TO RAM TRANSFER STOP ROUTINE
00CB FB                 2230  EI
00CC E9                 2240  PCHL
00CD                    2248 * PROGRAMMER VERIFICATION ROUTINE
00CD                    2249 * PART 2
00CD 7C                 2250  MOV A,H
00CE 21 00 FC           2260  LXI H,64512
00D1 19                 2270  DAD D
00D2 EB                 2280  XCHG
00D3 2E F1              2290  MVI L, 0F1H
00D5 67                 2300  MOV H,A
00D6 01 00 00           2310  LXI B, 0
00D9 E9                 2320  PCHL
00DA 00                 2625  NOP
00DB                    2629 * 7K TRANSFER COMPLETION CHECK
00DB D6 20              2630  SUI 904
00DD 6F                 2640  MOV L,A
00DE 7A                 2650  MOV A,D
00DF C6 04              2660  ADI 4
```

```
00E1 57          2670  MØV D,A
00E2 FE 38       2680  CPI 56
00E4 3F          2685  CMC
00E5 3E 00       2690  MVI A, 0
00E7 1F          2700  RAR
00E8 85          2710  ADD L
00E9 6F          2720  MØV L,A
00EA E9          2730  PCHL
00EB             2879 * RØM PRØGRAMMER STØP RØUTINE
00EB 00          2880  NØP
00EC 00          2881  NØP
00ED FB          2885  EI
00EE E9          2890  PCHL
00EF E9          2900  PCHL
00F0 E9          2906  PCHL
00F1             2918 * PRØGRAMMER VERIFICATIØN RØUTINE
00F1             2919 * PART 3
00F1 3B          2920  DCX SP
00F2 F1          2930  PØP 6
00F3 EB          2940  XCHG
00F4             2949 * CØMPARE FØR GREATER
00F4 BE          2950  CMP M
00F5 EB          2960  XCHG
00F6 17          2970  RAL
00F7 E6 01       3000  ANI 1
00F9 2F          3010  CMA
00FA 3C          3011  INR A
00FB 85          3015  ADD L
00FC 6F          3020  MØV L,A
00FD 3B          3030  DCX SP
00FE 3B          3040  DCX SP
00FF             3050 * CØMPARE FØR LESSER
00FF F1          3055  PØP 6
0100 2F          3056  CMA
0101 EB          3058  XCHG
0102 86          3059  ADD M
0103 EB          3060  XCHG
0104 C6 07       3061  ADI A,1
0106 3F          3065  CMC
0107 17          3070  RAL
0108 E6 01       3090  ANI 1
010A 2F          3100  CMA
010B 3C          3101  INR A
010C 85          3105  ADD L
010D 6F          3110  MØV L,A
010E 03          3130  INX B
010F 13          3140  INX D
0110 78          3150  MØV A,B
0111 E6 04       3180  ANI 4
0113 2F          3190  CMA
0114 3C          3191  INR A
0115 85          3195  ADD L
0116 6F          3200  MØV L,A
0117 E9          3210  PCHL
```

BYTEMOVER 3.0 OCTAL LISTING

BYTEMØVER VERSIØN 3.0

ØCTAL LISTING

```
061 000 000 301 321 056 311 000 345 345 000 000 000 061 004 000
315 000 000 061 002 000 341 061 004 000 325 305 371 016 000 131
151 333 377 127 346 007 007 007 107 172 346 070 017 000 147 071
056 000 172 353 346 200 017 017 306 055 041 000 000 157 071 351
371 041 013 000 071 353 371 353 021 000 000 073 361 002 003 023
172 346 004 007 007 000 205 157 351 000 000 076 126 205 157 351
000 151 174 140 371 147 056 153 001 000 000 073 361 022 023 003
170 376 374 077 037 037 346 100 056 175 205 157 351 056 153 170
346 004 007 007 007 205 157 351 000 000 000 174 041 000 374 071
371 041 000 374 031 353 147 056 153 170 346 370 306 010 107 351
333 377 107 346 340 036 000 113 127 170 346 037 107 147 056 140
351 306 032 157 333 377 346 100 017 017 205 157 351 174 041 000
374 071 371 056 315 147 351 000 000 000 000 373 351 174 041 000
374 031 353 056 361 147 001 000 000 351 000 326 220 157 172 306
004 127 376 070 077 076 000 037 205 157 351 000 000 373 351 351
351 073 361 353 276 353 027 346 001 057 074 205 157 073 073 361
057 353 206 353 306 007 077 027 346 001 057 074 205 157 003 023
170 346 004 057 074 205 157 351 000 000 000 000 000 000 000 000
```