

Table of Contents

1- 6	*****		
1- 7	*		*
1- 8	*	INFOSPHERE, INC	*
1- 9	*		*
1- 10	*	Z80/SPHERE RESIDENT DEBUGGER	*
1- 11	*		*
1- 12	*	PCP VERSION 1.0	*
1- 13	*		*
1- 14	*****		
1- 15			
2- 1	DEBUG	- CONFIGURATION EQUATES	
3- 1	DEBUG	- VECTOR PATCHES	
4- 1	DEBUG	- SID INITIALIZATION AND CONSOLE PRIMITIVES	
5- 1	DEBUG	- I/O SUPPORT ROUTINES	
6- 1	DEBUG	- INITIALIZATION AND COMMAND LOOP	
7- 1	DEBUG	- COMMAND TABLE	
8- 1	DEBUG	- HEX LOADER	
9- 1	DEBUG	- ASCII DUMP	
10- 1	DEBUG	- HEX ARITHMETIC & HEX DUMP	
11- 1	DEBUG	- EDIT MEMORY	
12- 1	DEBUG	- VARIOUS COMMANDS	
13- 1	DEBUG	- MOVE, VERIFY, AND CHECKSUM COMMANDS	
14- 1	DEBUG	- MEMORY TEST COMMAND	
15- 1	DEBUG	- EXCHANGE AND FILL MEMORY COMMANDS	
16- 1	DEBUG	- BREAKPOINT COMMANDS	

```
1 .TITLE DEBUG PROM FOR Z80/SPHERE
2 .IDENT /DEBUG/
3 .ENABL LC
4 .NLIST TTM
5
6 .SBTTL *****
7 .SBTTL *
8 .SBTTL * INFOSPHERE, INC *
9 .SBTTL *
10 .SBTTL * Z80/SPHERE RESIDENT DEBUGGER *
11 .SBTTL *
12 .SBTTL * PCP VERSION 1.0 *
13 .SBTTL *
14 .SBTTL *****
15 .SBTTL
16 ;
17 ; This program serves as a resident debugger for testing the initial
18 ; versions of Z80/Sphere on the PCP. It resides in ROM starting at
19 ; location 0000H. The Z80 restart and interrupt vectors are redirected
20 ; to locations starting at C000H, which is the beginning of the 16K byte
21 ; block of dynamic RAM on the PCP. Test versions of Sphere are down-loaded
22 ; into this area using the debugger's 'L' command.
23 ;
24 ; Summary of commands:
25 ;
26 ; A Ascii dump
27 ; B Binary arithmetic
28 ; D Hex dump
29 ; E Edit memory (in hex)
30 ; F Find 1 byte pattern
31 ; G Goto and execute program
32 ; H Hunt for 2 byte pattern
33 ; I Initialize or Input from port (depending on arguments)
34 ; L Intel format hex loader
35 ; M Move block of memory
36 ; O Output to port
37 ; Q Compute checksum of memory block
38 ; T Test memory block
39 ; U Compare 2 blocks of memory
40 ; X Exchange 2 blocks of memory
41 ; Z Fill memory block with a value
42 ; @ Set/reset breakpoint
```

DEBUG - CONFIGURATION EQUATES

```

1          .SBTTL  DEBUG - CONFIGURATION EQUATES
2          ;
3          ; Port definitions
4          ;
5          - 0009      CTC1 ==      Z09          ; CTC Baud rate ch 1
6          - 000A      CTC2 ==      Z0A          ; CTC Baud rate ch 2
7          - 0004      CNDAT ==     Z04          ; SIO A data port
8          - 0005      CNCMD ==     Z05          ; SIO A command/status port
9          - 0006      AUXDAT ==     Z06          ; SIO B data port
10         - 0007      AUXCMD ==     Z07          ; SIO B command/status port
11         ;
12         ; Serial I/O constants
13         ;
14         - 0005      WR5  ==      Z05          ; SIO write register 5
15         - 0004      WR4  ==      Z04          ; SIO write register 4
16         - 0003      WR3  ==      Z03          ; SIO write register 3
17         - 00C0      MASK5 ==     ZC0          ; SIO WR5 mask
18         - 0044      MASK4 ==     Z44          ; SIO WR4 mask
19         - 00E0      MASK3 ==     ZE0          ; SIO WR3 mask
20         - 0001      RXBF  ==     Z01          ; SIO Reciever ready mask
21         - 0004      TXBE  ==     Z04          ; SIO Transmitter mask
22         - 0018      RESET ==     Z18          ; SIO Reset mask
23         ;
24         ; Character Constants
25         ;
26         - 0008      BSP   ==     Z08          ; Backspace
27         - 000A      ALF   ==     Z0A          ; Line feed
28         - 000D      ACR   ==     Z0D          ; Carriage return
29         - 001B      ESC   ==     Z1B          ; Escape
30         - 0020      ASP   ==     Z20          ; Space
31         ;
32         ; Memory configuration and work storage definition.
33         ;
34         - E000      WORK  ==     ZE000        ; Start of work area
35         - E040      STAK  ==     WORK+64.     ; Debugger stack area
36         - E060      BSTAK ==     STAK+32.     ; Breakpoint stack area
37         - E060      IBUF  ==     BSTAK        ; Command buffer
38         0020      BUFL  =      32.           ; Command buffer length
39         - E080      IFLAG ==     IBUF+BUFL    ; Initializer flag
40         - E081      FLGP  ==     IFLAG+1      ; Pause flag
41         - E082      FLGBP ==     FLGP+1       ; Breakpoint flag
42         - E083      TCNTR ==     FLGBP+1      ; Memory test counter
43         - E084      DECNB ==     TCNTR+1      ;
44         - E085      YTEMP ==     DECNB+1      ;
45         - E086      BSTOR ==     YTEMP+1      ; Breakpoint storage area

```

DEBUG - VECTOR PATCHES

```
1          .SBTTL  DEBUG - VECTOR PATCHES
2          ;
3      0000          .ASECT  PROM
4          ;
5          ; Cold start: Start the debugger.
6          ;
7      0000          . = 0
8      0000 C3 0159  COLD: JP      START          ; Start the debugger
9          .BLKB  5
10         ;
11         ; Restart vectors: ---> C008 through C038
12         ;
13      0008 C3 C008  RST1: JP      ZC008
14          .BLKB  5
15         ;
16      0010 C3 C010  RST2: JP      ZC010
17          .BLKB  5
18         ;
19      0018 C3 C018  RST3: JP      ZC018
20          .BLKB  5
21         ;
22      0020 C3 C020  RST4: JP      ZC020
23          .BLKB  5
24         ;
25      0028 C3 C028  RST5: JP      ZC028
26          .BLKB  5
27         ;
28      0030 C3 C030  RST6: JP      ZC030
29          .BLKB  5
30         ;
31      0038 C3 C038  RST7: JP      ZC038
32         ;
33         ; NMI vector: ---> ZC066
34         ;
35      0066          . = Z66
36      0066 C3 C066  NMI:  JP      ZC066
```

DEBUG - SIO INITIALIZATION AND CONSOLE PRIMITIVES

```

1          .SBTTL  DEBUG  - SIO INITIALIZATION AND CONSOLE PRIMITIVES
2          ;
3          ; Initialize both channels of SIO, and both CTC baud rate channels.
4          ;
5 0069 21 0079  INITS:  LD      HL,30$      ; Point to initialization table
6 006C 7E      10$: LD      A,(HL)      ;
7 006D E7      OR      A              ; Exit if no more
8 006E 28 08   JR      Z,20$          ;
9 0070 46      LD      B,(HL)        ; Get byte count
10 0071 23     INC     HL             ;
11 0072 4E     LD      C,(HL)        ; Get port address
12 0073 23     INC     HL             ;
13 0074 B3ED   OTIR                    ; Block move data out
14 0076 18 F4   JR      10$          ; Until no more
15 0078 C9     20$: RET
16          ;
17 0079 02 09 47 0D 30$: .BYTE 2,CTC1,Z47,X0D
18 007D 02 0A 47 0D .BYTE 2,CTC2,Z47,X0D
19 0081 08 05 04 44 05 .BYTE 8,CNCMD,4,Z44,5,ZEA,3,XC1,1,0
20 008B 08 07 04 44 05 .BYTE 8,AUXCMD,4,Z44,5,ZEA,3,XC1,1,0
21 0095 00     .BYTE 0
22          ;
23          ; Console primitives:
24          ;
25          ;
26 0096 DB 07   QCI: IN      A,(AUXCMD)    ; See if reciever char ready
27 0098 E6 01   AND     RXBF
28 009A C9     RET
29          ;
30 009B CD 0096 XCI: CALL   QCI          ; Read in char (wait if need be)
31 009E 28 FB   JR      Z,XCI
32 00A0 DB 06   IN      A,(AUXDAT)
33 00A2 C9     RET
34          ;
35 00A3 F5     XCD: PUSH   AF
36 00A4 DB 07   10$: IN      A,(AUXCMD)    ; Send out character
37 00A6 E6 04   AND     TXBE
38 00A8 28 FA   JR      Z,10$
39 00AA F1     POP     AF
40 00AB B3 06   OUT     (AUXDAT),A
41 00AD C9     RET

```

```

1          .SBTTL  DEBUG - I/O SUPPORT ROUTINES
2          ;
3 00AE CD 0096  KEYBC:  CALL  QCI          ; Keyboard break (ESC) check
4 00B1 C8      RET      Z
5 00B2 CD 009B  CALL  XCI
6 00B5 FE 1B   CP      Z1B
7 00B7 C0      RET      NZ
8 00B8 C3 0159 JP      START
9          ;
10 00BB 3A E081 PAUSE:  LD      A,(FLGP)      ; Debugger pause
11 00BE FE 20   CP      ASP
12 00C0 28 07  JR      Z,KEYB
13 00C2 CD 00AE CALL  KEYBC
14 00C5 C8      RET      Z
15 00C6 FE 20   CP      ASP
16 00C8 C0      RET      NZ
17 00C9 CD 00AE KEYB: CALL  KEYBC
18 00CC 28 FB   JR      Z,KEYB
19 00CE 32 E081 LD      (FLGP),A
20 00D1 C9      RET
21          ;
22 00D2 3E 0D  CRLF: LD      A,ACR          ; Output <CR><LF>
23 00D4 CD 00A3 CALL  XCD
24 00D7 3E 0A  LD      A,ALF
25 00D9 C3 00A3 JP      XCD
26          ;
27 00DC FE 30  HEXC: CP      '0          ; Hex check
28 00DE D8      RET      C
29 00DF FE 3A  CP      <'9+1>
30 00E1 3F      CCF
31 00E2 D0      RET      NC
32 00E3 FE 41  CP      'A
33 00E5 D8      RET      C
34 00E6 FE 47  CP      <'F+1>
35 00E8 3F      CCF
36 00E9 C9      RET
37          ;
38 00EA 29  AHEX: ADD  HL,HL          ; Ascii to binary
39 00EB 29      ADD  HL,HL
40 00EC 29      ADD  HL,HL
41 00ED 29      ADD  HL,HL
42 00EE D6 30  SUB  48.
43 00F0 FE 0A  CP      10.
44 00F2 38 02  JR      C,10$
45 00F4 D6 07  SUB  7.
46 00F6 85  10$: ADD  A,L
47 00F7 6F      LD      L,A
48 00F8 C9      RET
49          ;
50 00F9 46  STRNG:  LD      B,(HL)      ; Print string
51 00FA 23  10$: INC  HL
52 00FB 7E      LD      A,(HL)
53 00FC CD 00A3 CALL  XCD
54 00FF 10 F9  DJNZ  10$

```

DEBUG - I/O SUPPORT ROUTINES

```

55 0101 C9          RET
56                ;
57 0102 FE 03      DLMT: CP      3          ; Dump limit
58 0104 38 03      JR          C,LENG
59 0106 21 FFFF    LD          HL,-1
60 0109 EB          LENG: EX      DE,HL
61 010A 7B          LD          A,E
62 010B 95          SUB          L
63 010C 5F          LD          E,A
64 010D 7A          LD          A,D
65 010E 9C          SBC          A,H
66 010F 57          LD          D,A
67 0110 C9          RET
68                ;
69                ;
70 0111 21 011A    ERR:  LD      HL,MSGER      ; Output 'ERROR'
71 0114 CD 00F9    CALL     STRNG
72 0117 C3 0159    JP       START
73                ;
74 011A 05 45 52 52 4F MSGER: .ASCIB /ERROR/
75                ;
76 0120 CD 0138    HLSP2: CALL   HLOUT      ; Output HL and 2 spaces
77 0123 18 0B      JR          SP2
78                ;
79 0125 CD 00D2    ADRO: CALL  CRLF      ; Output address
80 0128 CD 0138    CALL     HLOUT
81 012B 3E 3A      LD          A,':
82 012D CD 00A3    CALL     XCO
83 0130 CD 0133    SP2: CALL  SP1
84 0133 3E 20      SP1: LD     A,ASP
85 0135 C3 00A3    JP       XCO
86                ;
87 0138 7C          HLOUT: LD     A,H      ; Output H,L registers
88 0139 CD 0140    CALL     OUTH
89 013C 7D          LD          A,L
90 013D 18 01      JR          OUTH
91                ;
92 013F 7E          OUTH: LD     A,(HL)   ; Output hex format
93 0140 F5          OUTH: PUSH  AF
94 0141 0F          RRCA
95 0142 0F          RRCA
96 0143 0F          RRCA
97 0144 0F          RRCA
98 0145 CD 0149    CALL     BIASC
99 0148 F1          POP     AF
100 0149 E6 0F     BIASC: AND     15.      ; Binary to ascii
101 014B FE 0A      CP      10.
102 014D 3B 02      JR      C,NUMB
103 014F C6 07      ADD     A,7
104 0151 C6 30      NUMB: ADD  A,'0
105 0153 C3 00A3    JP     XCO

```

DEBUG - INITIALIZATION AND COMMAND LOOP

```

1          .SBTTL  DEBUG - INITIALIZATION AND COMMAND LOOP
2          ;
3 0156 32 E080  INITC:  LD      (IFLAG),A      ; Re-initialization
4 0159 F3      START:  DI                      ; --- Program start ---
5 015A 31 E040      LD      SP,STAK          ; Init stack
6 015D 3A E080      LD      A,(IFLAG)         ; Check Init flag
7 0160 FE C9      CP      ZC9              ; Warm start?
8 0162 28 1B      JR      Z,PRMPT          ; Br if so
9 0164 21 E060  INIT: LD      HL,IBUF         ; Else initialize by ...
10 0167 06 00      LD      B,0              ;
11 0169 36 00  INIT2: LD      (HL),0          ; ... clearing command buffer
12 016B 23          INC      HL              ;
13 016C 10 FB      DJNZ   INIT2             ;
14 016E 3E C9      LD      A,ZC9            ; ... setting IFLAG for warm starts
15 0170 32 E080      LD      (IFLAG),A         ;
16 0173 CD 0069      CALL   INITS              ;
17 0176 CD 00D2      CALL   CRLF              ;
18 0179 21 025B      LD      HL,CMMSG         ; ... and print sign-on message
19 017C CD 00F9      CALL   STRNG              ;
20 017F 21 0159  PRMPT: LD      HL,START        ; give a prompt
21 0182 E5          PUSH   HL              ;
22 0183 CD 00D2      CALL   CRLF              ;
23 0186 21 0271      LD      HL,PR1            ;
24 0189 CD 00F9      CALL   STRNG              ;
25 018C CD 009B  CMND: CALL   XCI              ; Read in command
26 018F FE 08      CP      BSP              ; ignore backspace
27 0191 26 F9      JR      Z,CMND          ;
28 0193 FE 14      CP      20.              ;
29 0195 38 F5      JR      C,CMND          ;
30 0197 CD 00A3      CALL   XCO              ;
31 019A CD 01F6      CALL   COMM              ; Valid command?
32 019D 28 11      JR      Z,BS1            ; Br if not
33 019F 21 E060      LD      HL,IBUF         ; Else put in buffer
34 01A2 01 0020      LD      BC,BUFL          ;
35 01A5 77          LD      (HL),A          ;
36 01A6 23          INC      HL              ;
37 01A7 CD 01C2      CALL   READ              ; Read in parameters
38 01AA FE 0D      CP      ACR              ; No more parameters?
39 01AC 28 5C      JR      Z,DECOD          ; Then go decode command
40 01AE 18 DC      JR      CMND              ;
41          ;
42 01B0 3E 08  BS1:  LD      A,BSP              ; Backspace on bad command
43 01B2 CD 00A3      CALL   XCO              ;
44 01B5 CD 01EC      CALL   BSPO              ;
45 01B8 18 D2      JR      CMND              ;
46          ;
47 01BA 2B      BLANK: DEC      HL              ;
48 01BB BE          CP      (HL)              ;
49 01BC 23          INC      HL              ;
50 01BD 20 1C      JR      NZ,FULL          ;
51 01BF CD 01EC  BS2:  CALL   BSPO              ;
52 01C2 CD 009B  READ: CALL   XCI              ; Load command buffer
53 01C5 CD 00A3      CALL   XCO              ;
54 01C8 77          LD      (HL),A          ;

```


DEBUG - INITIALIZATION AND COMMAND LOOP

```

55 01C9 FE 0D          CP      ACR          ;
56 01CB C8           RET      Z            ;
57 01CC FE 08          CP      BSP          ;
58 01CE 28 12         JR      Z,ERASE       ;
59 01D0 FE 20          CP      ASP          ;
60 01D2 28 E6         JR      Z,BLANK      ;
61 01D4 38 EC         JR      C,READ       ;
62 01D6 CD 00DC       CALL   HEXC          ;
63 01D9 38 E4         JR      C,BS2        ;
64 01DB 7D           FULL: LD   A,L          ;
65 01DC B9           CP      C            ;
66 01DD 28 E0         JR      Z,BS2        ;
67 01DF 23           INC     HL           ;
68 01E0 18 E0         JR      READ         ;
69                   ;
70 01E2 CD 01EC       ERASE: CALL  BSPO          ;
71 01E5 2R           DEC     HL           ;
72 01E6 3E 60         LD     A,"LB IBUF    ;
73 01E8 RD           CP      L            ;
74 01E9 20 D7         JR      NZ,READ      ;
75 01EB C9           RET                    ;
76                   ;
77 01EC 3E 20         BSPO: LD   A,ASP      ; Back Space out
78 01EE CD 00A3       CALL   XCD          ;
79 01F1 3E 08         LD     A,BSP        ;
80 01F3 C3 00A3       JP     XCD          ;
81                   ;
82 01F6 21 0279       COMM: LD   HL,<CTBLE-3> ; Command search
83 01F9 11 0003       LD     DE,3
84 01FC 47           LD     B,A
85 01FD 19           COM1: ADD  HL,DE
86 01FE 7E           LD     A,(HL)
87 01FF B7           OR     A
88 0200 C8           RET     Z
89 0201 B8           CP     B
90 0202 20 F9         JR     NZ,COM1
91 0204 23           INC    HL
92 0205 5E           LD     E,(HL)
93 0206 23           INC    HL
94 0207 56           LD     D,(HL)
95 0208 B7           OR     A
96 0209 C9           RET
97                   ;
98                   .ENABL  LSB
99 020A 3A E060       DECOD: LD   A,(IBUF) ;
100 020D CD 01F6      CALL   COMM
101 0210 21 E060      LD     HL,IBUF
102 0213 06 04       LD     B,4
103 0215 EB         EX     DE,HL
104 0216 13         104: INC  DE
105 0217 1A         LD     A,(DE)
106 0218 FE 0D       CP     13
107 021A 28 1A       JR     Z,SETUP
108 021C FE 20       CP     ASP

```

DEBUG - INITIALIZATION AND COMMAND LOOP

```

109 021E 28 0D          JR    Z,COUNT
110 0220 1B           DEC    DE
111 0221 3E 04        LD    A,4
112 0223 B8           CP    B
113 0224 28 07        JR    Z,COUNT
114 0226 13           INC    DE
115 0227 1A           LD    A,(DE)
116 0228 CD 00EA      CALL  AHX
117 022B 18 E9        JR    10$
118                   ;
119 022D 05           COUNT: DEC    B
120 022E 28 06        JR    Z,SETUP
121 0230 E5           PUSH  HL
122 0231 21 0000      LD    HL,0
123 0234 18 E0        JR    10$
124                   .DSABL LSB
125                   ;
126 0236 78           SETUP: LD    A,B ;
127 0237 01 0000      LD    BC,0
128 023A 11 0000      LD    DE,0
129 023D FE 04        CP    4
130 023F 28 0E        JR    Z,AD0
131 0241 FE 03        CP    3
132 0243 28 0E        JR    Z,AD1
133 0245 FE 02        CP    2
134 0247 28 03        JR    Z,AD2
135 0249 E5           PUSH  HL
136 024A C1           POP   BC
137 024B E1           POP   HL
138 024C D1          AD2: POP   DE
139 024D 23           INC   HL
140 024E C9           RET
141                   ;
142 024F E5           AD0: PUSH HL
143 0250 21 0000      LD    HL,0
144 0253 E5           AD1: PUSH HL
145 0254 06 10        LD    B,Z10
146 0256 09           ADD  HL,BC
147 0257 06 00        LD    B,0
148 0259 D1           POP   DE
149 025A C9           RET
150                   ;
151                   ; Sign on message
152                   ;
153 025B 15 50 43 50 2D MSG: .ASCIB /PCP-SPHERE DEBUGGER/<ACR><ALF>
154                   ;
155                   ; Prompt
156 0271 0A 43 6F 6D 6D PR1: .ASCIB /Command?/<ACR><ALF>

```

```
1          .SBTTL  DEBUG  -  COMMAND  TABLE
2          ;
3          ; Table generation macro
4          ;
5          .MACRO      TBGEN  $CMD,$ADDRS
6          .BYTE      $CMD
7          .WORD      $ADDRS
8          .ENDM
9          ;
10         ; Command table
11         ;
12         027C      CTBLE:      TBGEN  'A,ASCII
13         027F      TBGEN      'B,HEXAR
14         0282      TBGEN      'D,DUMP
15         0285      TBGEN      'E,EDIT
16         0288      TBGEN      'F,LOOK
17         028B      TBGEN      'G,EXCT
18         028E      TBGEN      'H,LOOK
19         0291      TBGEN      'I,ICMD
20         0294      TBGEN      'L,LOAD
21         0297      TBGEN      'K,MOVE
22         029A      TBGEN      'O,OUTP
23         029D      TBGEN      'Q,CKSUM
24         02A0      TBGEN      'T,TEST
25         02A3      TBGEN      'V,VRFY
26         02A6      TBGEN      'X,XCHG
27         02A9      TBGEN      'Z,ZERO
28         02AC      TBGEN      '@,BKPT
29         ;
30         02AF 0000      .WORD  0
31         02B1 0000      .WORD  0
```

```

1          .SBTTL  DEBUG  -  HEX  LOADER
2          ;
3          ; Intel format hex loader:
4          ;
5 02B3 CD 009B    LDR: CALL  XCI          ; Get character
6 02B6 FE 3A      CP      ':'          ; See if record mark
7 02B8 20 F9      JR      NZ,LDR      ; Br if not and keep trying
8 02BA CD 02DB    CALL    40$          ; Read two chars from console
9 02BD 47         LD      B,A          ; Put length in B (for DJNZ)
10 02BE CD 02DB   CALL    40$          ; Get MSB of load byte
11 02C1 67         LD      H,A          ; Store
12 02C2 CD 02DB   CALL    40$          ; Get LSB of load byte
13 02C5 6F         LD      L,A          ; Store address
14 02C6 CD 02DB   CALL    40$          ; Get record type
15 02C9 20 0C     JR      NZ,20$      ; If not type 0, all done
16          ;
17 02CB CD 02DB    10$: CALL    40$          ; Get data from tape
18 02CE 77         LD      (HL),A        ; Store data in memory
19 02CF 23         INC     HL          ; Bump memory address
20 02D0 10 F9     DJNZ    10$          ; Dec B and loop until done
21 02D2 CD 02DB   CALL    40$          ; Eat checksum
22 02D5 18 DC     JR      LDR          ; Then get next frame
23          ;
24 02D7 CD 02DB    20$: CALL    40$          ; Eat checksum
25 02DA C9         RET              ; And return
26          ;
27 02DB 0E 00     40$: LD      C,0          ; Init temporary storage
28 02DD CD 02E5   CALL    50$          ; Get next hex digit
29 02E0 07         RLCA           ; Shift into upper bits
30 02E1 07         RLCA
31 02E2 07         RLCA
32 02E3 07         RLCA
33 02E4 4F         LD      C,A          ; Store temporary
34 02E5 CD 009B    50$: CALL    XCI          ; Get next char
35 02E8 D6 30     SUB     '0          ; De-asciify
36 02EA FE 0A     CP      10.          ; Check for 0-9
37 02EC FA 02F1   JP      M,60$          ; Skip if in range
38 02EF D6 07     SUB     7          ; Else de-hexify
39 02F1 B1        60$: OR      C          ; Add in high order bits
40 02F2 C9         RET              ; Then return without further checking
41          ;
42          ;
43          ;
44 02F3 CD 02B3    LOAD: CALL  LDR          ;
45 02F6 C3 0159   JP      START          ;

```

```

1          .SBTTL  DEBUG - ASCII DUMP
2          ;
3          ;
4  02F9 2B      ASCII: DEC   HL      ; Ascii dump <start><finish>
5  02FA F5      PUSH   AF
6  02FB 7D      LD     A,L
7  02FC B4      OR     H
8  02FD 4F      LD     C,A
9  02FE F1      POP    AF
10 02FF FE 03   CP     3
11 0301 38 01   JR     C,5%
12 0303 4F      LD     C,A
13 0304 23      5%:  INC   HL
14 0305 CD 0102 CALL  DLMT
15 0308 CD 0125 10%: CALL  ADRO
16 030B 06 30   LD     B,48.
17 030D 0C      INC   C
18 030E 0D      DEC   C
19 030F 20 05   JR     NZ,20%
20 0311 11 FFFF LD     DE,-1
21 0314 06 10   LD     B,16.
22 0316 7A      20%: LD     A,D
23 0317 B3      OR     E
24 0318 C8      RET   Z
25 0319 7E      LD     A,(HL)
26 031A E6 7F   AND   Z7F
27 031C 1F      DEC   DE
28 031D 0C      INC   C
29 031E 0D      DEC   C
30 031F 28 1B   JR     Z,50%
31 0321 FE 08   CP     BSP
32 0323 28 08   JR     Z,30%
33 0325 FE 0D   CP     ACR
34 0327 28 0D   JR     Z,40%
35 0329 FE 20   CP     ASP
36 032B 30 02   JR     NC,31%
37 032D 3E 2E   30%: LD     A,'.
38 032F CD 00A3 31%: CALL  XCD
39 0332 23      70%: INC   HL
40 0333 10 E1   DJNZ  20%
41 0335 2B      DEC   HL
42 0336 23      40%: INC   HL
43 0337 CD 00B5 CALL  PAUSE
44 033A 18 CC   JR     10%
45          ;
46 033C FE 20   50%: CP     ASP
47 033E 38 15   JR     C,60%
48 0340 FE 08   CP     BSP
49 0342 28 11   JR     Z,60%
50 0344 CD 00A3 CALL  XCD
51 0347 7E      LD     A,(HL)
52 0348 07      RLCA
53 0349 30 0F   JR     NC,90%
54 034B 3E 2E   LD     A,'.

```

DEBUG PROM FOR Z80/SPHERE 23-Sep-83 16:16

Cymric Computer Systems MaxMZ8-20DEC2

Licensee: INFOSPHERE, INC. 4730 Southwest Macadam, Portland, OR 97201

Page: 9-2 (556)

DEBUG - ASCII DUMP

55	034D	CD 00A3		CALL	XCO
56	0350	CD 0133	80%	CALL	SP1
57	0353	18 DD		JR	70%
58			;		
59	0355	CD 013F	60%	CALL	OUTM
60	0358	18 F6		JR	80%
61			;		
62	035A	CD 0130	90%	CALL	SP2
63	035D	18 B3		JR	70%

DEBUG - HEX ARITHMETIC & HEX DUMP

```

1          .SBTTL  DEBUG - HEX ARITHMETIC & HEX DUMP
2
3 035F FE 03      HEXAR:  CP      3          ; Binary arithmetic (B)
4 0361 38 29      JR      C,HEX2
5 0363 EB        EX      DE,HL
6 0364 11 D8F0    DECML:  LD      DE,-10000.    ; Decimal output
7 0367 CD 037F    CALL    DIVD
8 036A 11 FC18    LD      DE,-1000.
9 036D CD 037F    CALL    DIVD
10 0370 11 FF9C   LD      DE,-100.
11 0373 CD 037F   CALL    DIVD
12 0376 11 FFF6   DEC2:  LD      DE,-10.
13 0379 CD 037F   CALL    DIVD
14 037C 11 FFFF   LD      DE,-1.
15 037F 3E 2F     DIVD:  LD      A,<'0-1>
16 0381 E5       PUSH   HL
17 0382 C1       10$:  POP    BC
18 0383 3C       INC    A
19 0384 E5       PUSH   HL
20 0385 19       ADD    HL,DE
21 0386 38 FA    JR      C,10$
22 0388 E1       POP    HL
23 0389 C3 00A3  JF     XCO
24
25 038C 2E       ;
26 038D E5       HEX2:  DEC    HL
27 038E 21 03DA  PUSH   HL
28 0391 CD 00F9  LD     HL,YMSG
29 0394 CD 00D2  CALL   STRNG
30 0397 E1       CALL   CRLF
31 0398 E5       POP    HL
32 0399 CD 0109  PUSH   HL
33 039C CD 0133  CALL   LENG
34 039F CD 0120  CALL   SP1
35 03A2 EB       CALL   HLSP2
36 03A3 E3       EX     DE,HL
37 03A4 CD 0120  EX     (SP),HL
38 03A7 EB       CALL   HLSP2
39 03A8 E5       EX     DE,HL
40 03A9 CD 0109  PUSH   HL
41 03AC C1       CALL   LENG
42 03AD E5       POP    BC
43 03AE 09       PUSH   HL
44 03AF CD 0120  ADD    HL,BC
45 03B2 EB       CALL   HLSP2
46 03B3 CD 0120  EX     DE,HL
47 03B6 22 E085  CALL   HLSP2
48 03B9 E1       LD     (YTEMP),HL
49 03BA E3       POP    HL
50 03BB CD 0138  EX     (SP),HL
51 03BE E3       CALL   HLOUT
52 03BF E5       EX     (SP),HL
53 03C0 2A E085  PUSH   HL
54 03C3 E3       LD     HL,(YTEMP)
                    EX     (SP),HL

```

55	03C4	D5		PUSH	DE				
56	03C5	E5		PUSH	HL				
57	03C6	C5		PUSH	BC				
58	03C7	CD 00D2		CALL	CRLF				
59	03CA	06 06		LD	B,6				
60	03CC	C5		PUSH	BC				
61	03CD	C1	HEX1:	POP	BC				
62	03CE	05		DEC	B				
63	03CF	C8		RET	Z				
64	03D0	E1		POP	HL				
65	03D1	C5		PUSH	BC				
66	03D2	CD 0364		CALL	DECML				
67	03D5	CD 0133		CALL	SP1				
68	03D8	18 F3		JR	HEX1				
69									
70	03DA	1E 20 20 41 20	YMSG:	.ASCIB	/ A	B	A+B	A-B	B-A/
71									
72	03F9	CD 0102	DUMP:	CALL	DLMT				; Dump hex (D)
73	03FC	CD 0125	5#:	CALL	ADRO				
74	03FF	06 10		LD	B,16.				
75	0401	CD 013F	10#:	CALL	OUTM				
76	0404	23		INC	HL				
77	0405	1B		DEC	DE				
78	0406	7A		LD	A,D				
79	0407	B3		OR	E				
80	0408	C8		RET	Z				
81	0409	CD 0133		CALL	SP1				
82	040C	10 F3		DJNZ	10#				
83	040E	CD 00BB		CALL	PAUSE				
84	0411	18 E9		JR	5#				

DEBUG - EDIT MEMORY

```

1          .SBTTL  DEBUG - EDIT MEMORY
2          ;
3 0413 1B   EDO: DEC  DE
4 0414 EB   EDIT: EX  DE,HL      ; Edit memory (E)
5 0415 CD 0125 CALL  ADRO
6 0418 EB   EX      DE,HL
7 0419 EB   20$: EX  DE,HL
8 041A CD 013F CALL  OUTH
9 041D EB   EX      DE,HL
10 041E 3E 2D LD    A,'-
11 0420 CD 00A3 CALL  XCO
12 0423 21 0000 LD    HL,0
13 0426 06 01  LD    B,1
14 0428 CD 009B 30$: CALL XCI
15 042B 28 FB   JR    Z,30$
16 042D 4F     LD    C,A
17 042E CD 00DC CALL  HEXC
18 0431 79     LD    A,C
19 0432 30 19   JR    NC,40$
20 0434 FE 20   CP    ASP
21 0436 28 1E   JR    Z,50$
22 0438 FE 0D   CP    ACR
23 043A 28 1F   JR    Z,60$
24 043C FE 08   CP    BSP
25 043E 28 E3   JR    Z,EDO
26 0440 FE 1B   CP    ESC
27 0442 20 E4   JR    NZ,30$
28 0444 05     80$: DEC  B
29 0445 28 02   JR    Z,90$
30 0447 7D     LD    A,L
31 0448 12     LD    (DE),A
32 0449 13     90$: INC  DE
33 044A C3 0130 JP    SP2
34          ;
35 044D CD 00A3 40$: CALL XCD
36 0450 47     LD    B,A
37 0451 CD 00EA CALL  AHX
38 0454 18 D2   JR    30$
39          ;
40 0456 CD 0444 50$: CALL 80$
41 0459 18 BE   JR    20$
42          ;
43 045B CD 0444 60$: CALL 80$
44 045E 18 B4   JR    EDIT
    
```

```

1          .SBTTL  DEBUG - VARIOUS COMMANDS
2          ;
3 0460 CD 0109   LOOK: CALL  LENG          ; 1 and 2 byte searches (F,H)
4 0463 7E       10$: LD    A,(HL)
5 0464 B9       CP      C
6 0465 CC 046F   CALL    Z,20$
7 0468 7A       LD      A,D
8 0469 B3       OR      E
9 046A C8       RET     Z
10 046B 23      INC     HL
11 046C 1B      DEC     DE
12 046D 18 F4   JR      10$
13         ;
14 046F 3A E060  20$: LD    A,(IBUF)
15 0472 FE 46   CP      'F
16 0474 20 11   JR      NZ,30$
17 0476 CD 0125   CALL   ADRO          ; Find 1 byte (F)
18 0479 CD 013F   CALL   OUTM
19 047C CD 0130   CALL   SP2
20 047F 23      INC     HL
21 0480 CD 013F   CALL   OUTM
22 0483 2B      DEC     HL
23 0484 C3 00BB  JP      PAUSE
24         ;
25 0487 23      30$: INC   HL          ; Hunt for 2 bytes (H)
26 0488 7E       LD      A,(HL)
27 0489 BB       CP      B
28 048A 2B      DEC     HL
29 048B C0       RET     NZ
30 048C 2B      DEC     HL
31 048D CD 0125   CALL   ADRO
32 0490 CD 013F   CALL   OUTM
33 0493 CD 0130   CALL   SP2
34 0496 23      INC     HL
35 0497 23      INC     HL
36 0498 CD 013F   CALL   OUTM
37 049B 2B      DEC     HL
38 049C CD 013F   CALL   OUTM
39 049F C3 00BB  JP      PAUSE
40         ;
41 04A2 D5      EXCT: PUSH DE          ; Execute (G)
42 04A3 C9      RET
43         ;
44 04A4 FE 04   ICMD: CP      4          ; I command
45 04A6 CA 0156   JP      Z,INITC        ; Initialize or (I)
46 04A9 CD 00D2   CALL   CRLF
47 04AC 4D      LD      C,L          ; Input from port (I)
48 04AD 78ED    IN      A,(C)
49 04AF CD 0140   CALL   OUTH
50 04B2 C3 0159   JP      START
51         ;
52 04B5 4B      OUTP: LD     C,E          ; Output to port (O)
53 04B6 69ED    OUT     (C),L
54 04B8 C3 0159   JP      START
  
```

DEBUG - MOVE, VERIFY, AND CHECKSUM COMMANDS

```

1          .SBTTL  DEBUG - MOVE, VERIFY, AND CHECKSUM COMMANDS
2          ;
3  04BB FE 02      MOVE: CP      2          ; Move memory (M)
4  04BD D2 0111    JP      NC,ERR
5  04C0 E5        PUSH     HL
6  04C1 CD 0109    CALL    LENG
7  04C4 B5        PUSH     DE
8  04C5 EB        EX      DE,HL
9  04C6 C5        PUSH     BC
10 04C7 E1        POP      HL
11 04C8 CD 0109    CALL    LENG
12 04CB 30 0C     JR      NC,GTR
13 04CD D1        POP      DE
14 04CE F1        POP      AF
15 04CF 7E        MV1: LD     A,(HL)
16 04D0 02        LD      (BC),A
17 04D1 03        INC     BC
18 04D2 23        INC     HL
19 04D3 1B        DEC     DE
20 04D4 7A        LD      A,D
21 04D5 B3        OR      E
22 04D6 20 F7     JR      NZ,MV1
23 04D8 C9        RET
24          ;
25 04D9 D1        GTR: POP     DE
26 04DA B5        PUSH     DE
27 04DB E1        POP     HL
28 04DC 09        ADD     HL,BC
29 04DD C1        POP     BC
30 04DE 2B        DEC     HL
31 04DF 0B        DEC     BC
32 04E0 0A        MV2: LD     A,(BC)
33 04E1 77        LD      (HL),A
34 04E2 0B        DEC     BC
35 04E3 2B        DEC     HL
36 04E4 1B        DEC     DE
37 04E5 7A        LD      A,D
38 04E6 B3        OR      E
39 04E7 20 F7     JR      NZ,MV2
40 04E9 C9        RET
41          ;
42 04EA FE 02      VRFY: CP      2          ; Verify memory (V)
43 04EC B2 0111    JP      NC,ERR
44 04EF CD 0109    CALL    LENG
45 04F2 0A        10$: LD     A,(BC)
46 04F3 BE        CP      (HL)
47 04F4 C4 04FF    CALL    NZ,20$
48 04F7 23        INC     HL
49 04F8 03        INC     BC
50 04F9 1B        DEC     DE
51 04FA 7A        LD      A,D
52 04FB B3        OR      E
53 04FC 20 F4     JR      NZ,10$
54 04FE C9        RET

```

DEBUG - MOVE, VERIFY, AND CHECKSUM COMMANDS

```

55      ;
56 04FF CD 0125      20$: CALL  ADRO
57 0502 CD 013F      CALL   OUTM
58 0505 CD 0130      CALL   SP2
59 0508 0A          LD    A,(BC)
60 0509 CD 0140      CALL   OUTH
61 050C C3 00BB      JP    PAUSE
62      ;
63      ;
64 050F CD 0109      CKSUM: CALL  LENG      ; Checksum of memory (Q)
65 0512 0E 00          LD    C,0
66 0514 79          5$:  LD    A,C
67 0515 86          ADD   A,(HL)
68 0516 4F          LD    C,A
69 0517 23          INC   HL
70 0518 1B          DEC   DE
71 0519 7A          LD    A,D
72 051A B3          OR    E
73 051B 20 F7       JR    NZ,5$
74 051D CD 00D2      CALL   CRLF
75 0520 79          LD    A,C
76 0521 C3 0140      JP    OUTH
    
```

DEBUG - MEMORY TEST COMMAND

```

1          .SBTTL  DEBUG - MEMORY TEST COMMAND
2          ;
3 0524 CD 0109      TEST: CALL  LENG          ; Memory test (T)
4 0527 01 5A5A      LD      BC,Z5A5A
5 052A CD 056F      CYCL: CALL  RNDM
6 052D C5           PUSH  BC
7 052E E5           PUSH  HL
8 052F D5           PUSH  DE
9 0530 CD 056F      TLOP: CALL  RNDM
10 0533 70          LD      (HL),B
11 0534 23          INC   HL
12 0535 1B          DEC   DE
13 0536 7A          LD      A,D
14 0537 B3          OR    E
15 0538 20 F6       JR    NZ,TLOP
16 053A CD 00BB     CALL  PAUSE
17 053D D1          POP   DE
18 053E E1          POP   HL
19 053F C1          POP   BC
20 0540 E5          PUSH  HL
21 0541 D5          PUSH  DE
22 0542 CD 056F     RLOP: CALL  RNDM
23 0545 7E          LD      A,(HL)
24 0546 B8          CP    B
25 0547 C4 055D     CALL  NZ,MERR
26 054A 23          INC   HL
27 054B 1B          DEC   DE
28 054C 7A          LD      A,D
29 054D B3          OR    E
30 054E 20 F2       JR    NZ,RLOP
31 0550 21 E0B3     LD      HL,TCNTR
32 0553 34          INC   (HL)
33 0554 C5          PUSH  BC
34 0555 CC 057E     CALL  Z,TESTN
35 0558 C1          POP   BC
36 0559 D1          POP   DE
37 055A E1          POP   HL
38 055B 18 CD       JR    CYCL
39          ;
40 055D F5          MERR: PUSH  AF          ; Error found
41 055E CD 00BB     CALL  PAUSE
42 0561 CD 0125     CALL  ADRO
43 0564 78          LD      A,B
44 0565 CD 0140     CALL  OUTH
45 0568 CD 0130     CALL  SP2
46 056B F1          POP   AF
47 056C C3 0140     JP    OUTH
48          ;
49 056F 78          RNDM: LD      A,B          ; Pseudo-random number generator
50 0570 E6 B4       AND   XB4
51 0572 A7          AND   A
52 0573 EA 0577     JP    PE,10%
53 0576 37          SCF
54 0577 79          10%: LD      A,C

```

```
55 0578 17          RLA
56 0579 4F          LD      C,A
57 057A 78          LD      A,B
58 057B 17          RLA
59 057C 47          LD      B,A
60 057D C9          RET
61                ;
62 057E 3A E084      TESTN: LD      A,(DECNB) ; Test counter
63 0581 C6 01        ADD     A,1
64 0583 27          DAA
65 0584 32 E084      LD      (DECNB),A
66 0587 CD 0140      CALL   OUTH
67 058A 21 0593      LD      HL,NMSG
68 058D CD 00F9      CALL   STRNG
69 0590 C3 00D2      JP      CRLF
70                ;
71 0593 06 20 54 45 53 NMSG: .ASCIB / TESTS/
```

DEBUG - EXCHANGE AND FILL MEMORY COMMANDS

```
1          .SBTTL  DEBUG - EXCHANGE AND FILL MEMORY COMMANDS
2          ;
3 059A FE 02      XCHG: CP      2          ; Exchange memory (X)
4 059C D2 0111    JP      NC,ERR
5 059F CD 0109    CALL     LENG
6 05A2 0A        10#: LD      A,(BC)
7 05A3 F5        PUSH    AF
8 05A4 7E        LD      A,(HL)
9 05A5 02        LD      (BC),A
10 05A6 F1       POP     AF
11 05A7 77       LD      (HL),A
12 05A8 23       INC     HL
13 05A9 03       INC     BC
14 05AA 1B       DEC     DE
15 05AB 7A       LD      A,D
16 05AC B3       OR      E
17 05AD 20 F3    JR      NZ,10#
18 05AF C9       RET
19          ;
20          ;
21 05B0 CD 0109    ZERG: CALL  LENG          ; Fill memory (Z)
22 05B3 71       10#: LD      (HL),C
23 05B4 23       INC     HL
24 05B5 1B       DEC     DE
25 05B6 7A       LD      A,D
26 05B7 B3       OR      E
27 05B8 20 F9    JR      NZ,10#
28 05BA C9       RET
```

```

1          .SBTTL  DEBUG - BREAKPOINT COMMANDS
2          ;
3          0006   BRKH =   ^HB BREAK
4          0008   BRKL =   ^LB BREAK
5          ;
6          .ENABL  LSB
7          05BB FE 04   BKPT: CP      4           ; Set breakpoint (@)
8          05BD 28 27   JR          Z,30$
9          05BF D5     PUSH        DE
10         05C0 3A E082  LD          A,(FLGBF)
11         05C3 B7     OR          A
12         05C4 28 03   JR          Z,20$
13         05C6 CD 05F2 CALL        RESBP
14         05C9 E1     20$: POP      HL
15         05CA 22 E086 LD          (BSTOR),HL
16         05CD 7C     LD          A,H
17         05CE B5     OR          L
18         05CF 28 15   JR          Z,30$
19         05D1 11 E082 LD          DE,FLGBF
20         05D4 12     LD          (DE),A
21         05D5 13     INC        DE
22         05D6 7E     LD          A,(HL)
23         05D7 12     LD          (DE),A
24         05D8 36 CD   LD          (HL),XCD
25         05DA 23     INC        HL
26         05DB 13     INC        DE
27         05DC 7E     LD          A,(HL)
28         05DD 12     LD          (DE),A
29         05DE 36 08  LD          (HL),BRKL
30         05E0 23     INC        HL
31         05E1 13     INC        DE
32         05E2 7E     LD          A,(HL)
33         05E3 12     LD          (DE),A
34         05E4 36 06  LD          (HL),BRKH
35         05E6 21 064E 30$: LD      HL,MSGBP
36         05E9 CD 00F9 CALL        STRNG
37         05EC 2A E086 LD          HL,(BSTOR)
38         05EF C3 0138 JP          HLOUT
39         ;
40         05F2 2A E086 RESBP:   LD      HL,(BSTOR)
41         05F5 11 E082 LD          DE,FLGBF
42         05F8 AF     XOR        A
43         05F9 12     LD          (DE),A
44         05FA 13     INC        DE
45         05FB 06 03  LD          B,3
46         05FD 1A     40$: LD      A,(DE)
47         05FE 77     LD          (HL),A
48         05FF 23     INC        HL
49         0600 13     INC        DE
50         0601 10 FA  DJNZ       40$
51         0603 AF     XOR        A
52         0604 12     LD          (DE),A
53         0605 13     INC        DE
54         0606 12     LD          (DE),A

```


DEBUG - BREAKPOINT COMMANDS

```

55 0607 C9          RET
56                ;
57 0608 F3          BREAK:    DI          ;
58 0609 D9          EXX
59 060A 22 E05A     LD      (BSTAK-6),HL
60 060D E1          POP      HL
61 060E E1          POP      HL
62 060F 22 E05E     LD      (BSTAK-2),HL
63 0612 21 0000     LD      HL,0
64 0615 F5          PUSH     AF
65 0616 39          ADD     HL,SP
66 0617 F1          POP      AF
67 0618 22FD E05C   LD      (BSTAK-4),IY
68 061C 31 E05A     LD      SP,<BSTAK-6>
69 061F 08          EX      AF,AF'
70 0620 D5          PUSH     DE
71 0621 C5          PUSH     BC
72 0622 F5          PUSH     AF
73 0623 E5          PUSH     HL
74 0624 D9          EXX
75 0625 08          EX      AF,AF'
76 0626 E5DD        PUSH     IX
77 0628 E5          PUSH     HL
78 0629 D5          PUSH     DE
79 062A C5          PUSH     BC
80 062B F5          PUSH     AF
81 062C CD 05E6     CALL    30$
82 062F 21 0658     LD      HL,MSGBK
83 0632 CD 00F9     CALL    STRNG
84 0635 0E 02       LD      C,2
85 0637 CD 00D2     50$: CALL    CRLF
86 063A 06 06       LD      B,6
87 063C E1          60$: POP      HL
88 063D CD 0138     CALL    HLOUT
89 0640 CD 0133     CALL    SP1
90 0643 10 F7       DJNZ   60$
91 0645 0D          DEC     C
92 0646 20 EF       JR     NZ,50$
93 0648 CD 05F2     CALL    RESBP
94 064B C3 0159     JP     START
95                .DSABL  LSB
96                ;
97 064E 09 0D 42 72 65 MSGBP: .ASCIB <ACR>/Break at/
98 0658 1F 0D 0D 41 46 MSGBK: .ASCIB <ACR><ACR>/AF BC DE HL IX-Y STAK/
99 0001                .END

```

Symbol Table: ACR to ZERO

ACR=	D-		GTR	4D9	PROM	SP1	133	PROM
ADRO	125	PROM	HEXAR	35F	PROM	SP2	130	PROM
ADO	24F	PROM	HEXC	DC	PROM	STAK=	E040-	
AD1	253	PROM	HEX1	3CD	PROM	START	159	PROM
AD2	24C	PROM	HEX2	38C	PROM	STRNG	F9	PROM
AHEX	EA	PROM	HLOUT	138	PROM	TCNTR=	E083-	
ALF=	A-		HLSP2	120	PROM	TEST	524	PROM
ASCI	2F9	PROM	IBUF=	E060-		TESTN	57E	PROM
ASP=	20-		ICMD	4A4	PROM	TLOP	530	PROM
AUXCMD=	7-		IFLAG=	E080-		TXBE=	4-	
AUXDAT=	6-		INIT	164	PROM	URFY	4EA	PROM
BIASC	149	PROM	INITC	156	PROM	WORK=	E000-	
BKPT	5BB	PROM	INITS	69	PROM	WR3=	3-	
BLANK	1BA	PROM	INIT2	169	PROM	WR4=	4-	
BREAK	608	PROM	KEYB	C9	PROM	WR5=	5-	
BRKH=	6		KEYBC	AE	PROM	XCHG	59A	PROM
BRKL=	8		LDR	283	PROM	XCI	9B	PROM
BSP=	8-		LENG	109	PROM	XCO	A3	PROM
BSPO	1EC	PROM	LOAD	2F3	PROM	YMSG	3DA	PROM
BSTAK=	E060-		LOOK	460	PROM	YTEMP=	E085-	
BSTOR=	E086-		MASK3=	E0-		ZERO	5B0	PROM
BS1	1B0	PROM	MASK4=	44-				
BS2	1BF	PROM	MASK5=	C0-				
BUFL=	20		MERR	55D	PROM			
CKSUM	50F	PROM	MOVE	48B	PROM			
CMND	18C	PROM	MSGBK	658	PROM			
CMSG	25B	PROM	MSGBP	64E	PROM			
CNCMD=	5-		MSGER	11A	PROM			
CNDAT=	4-		MV1	4CF	PROM			
COLD	0	PROM	MV2	4E0	PROM			
COMM	1FA	PROM	NMI	66	PROM			
COM1	1FD	PROM	NMSG	593	PROM			
COUNT	22D	PROM	NUMB	151	PROM			
CRLF	D2	PROM	OUTH	140	PROM			
CTELE	27C	PROM	OUTM	13F	PROM			
CTC1=	9-		OUTP	4B5	PROM			
CTC2=	A-		PAUSE	BB	PROM			
CYCL	52A	PROM	PRMPT	17F	PROM			
DECML	364	PROM	PR1	271	PROM			
DECNB=	E084-		QCI	96	PROM			
DECOB	20A	PROM	READ	1C2	PROM			
DEC2	376	PROM	RESBP	5F2	PROM			
DIVD	37F	PROM	RESET=	18-				
DLMT	102	PROM	RLOP	542	PROM			
DUMP	3F9	PROM	RNDM	56F	PROM			
EDIT	414	PROM	RST1	8	PROM			
EDO	413	PROM	RST2	10	PROM			
ERASE	1E2	PROM	RST3	18	PROM			
ERR	111	PROM	RST4	20	PROM			
ESC=	1B-		RST5	28	PROM			
EXCT	4A2	PROM	RST6	30	PROM			
FLGBP=	E082-		RST7	38	PROM			
FLGP=	E081-		RXBF=	1-				
FULL	1DB	PROM	SETUP	236	PROM			

DEBUG PROM FOR Z80/SPHERE 23-Sep-83 16:16
Cymric Computer Systems MaxMZ8-20DEC2
Licensee: INFOSPHERE, INC. 4730 Southwest Macadam, Portland, OR 97201

Program Sections

Sector	Bytes	Words	Attributes				
<.ABS.>	0	0	OVR	ABS	GBL	RW	LD
<◇>	0	0	CON	REL	LCL	[RO]	LD
<PROM>	678	828	[CON]	ABS	[LCL]	RW	LD

Total Free Tables Linked Blocks Errors

2578 1488 1042 48 2 0
,PROM,PROM-PROM