```
                                2313  ***     DAT     -       DATA BYTE OUTPUT TO Z-47
                                2314  *
                                2315  *       ENTRY:  (A) = BYTE TO OUTPUT
                                2316  *
                                2317  *       EXIT:   (A) = BYTE TO OUTPUT
                                2318  *               (D) = S.DTR
                                2319  *
                                2320  *       USE:    AF, D
                                2321
    006.023                     2322  DAT     EQU     *
    006.023  026 200            2323          MVI     D,S.DTR                 SET MATCH CONDITION TO DATA TRANSFER
    006.025  030 002            2324          JR      COM1                    REQUEST BIT


                                2326  ***     COM     - OUTPUT COMMAND BYTE TO Z-47
                                2327  *
                                2328  *       ENTRY:  (A) = COMMAND BYTE
                                2329  *
                                2330  *       EXIT:   (A) = COMMAND BYTE
                                2331  *               (D) = S.DON
                                2332  *
                                2333  *       USE:    AF, D
                                2334
    006.027                     2335  COM     EQU     *
    006.027  026 040            2336          MVI     D,S.DON                 SET MATCH CONDITION TO DONE BIT
    006.031  365                2337  COM1    PUSH    PSW
    006.032  315 170 006        2338  WTDON1  CALL    IN.                     READ CONTROLLER STATUS REGISTER
    006.035  242                2339          ANA     D                       GET MATCH BIT ONLY
    006.036  050 372            2340          JR      Z,WTDON1                IF NO MATCH, WAIT
    006.040  361                2341          POP     PSW
    006.041  315 146 006        2342          CALL    OUT1.                   OUTPUT THE BYTE TO THE DATA PORT
    006.044  311                2343          RET


    006.045                     2345          ORG     6045A
                                2346  **      HRNX - HORN EXTENSION ROUTINE
                                2347  *
                                2348  *       THIS IS AN EXTENSION TO *HORN* TO MAKE ROOM FOR A JUMP
                                2349
    006.045  056 011            2350  HRNX    MVI     L,#CTLFLG
    006.047  163                2351          MOV     M,E                     TURN OFF HORN
    006.050  321                2352          POP     D
    006.051  341                2353          POP     H
    006.052  311                2354          RET
```

```
                        2356  **      NOISE - DING BELL ON CONSOLE
                        2357  *
                        2358  *       THIS IS A MODIFICATION TO ALLOW THE H88/H89 TO USE THE CONSOLE BELL
                        2359
006.053 076 007         2360  NOISE   MVI     A,A,BEL
006.055 315 302 003     2361          CALL    WCC
006.060 303 140 002     2362          JMP     HORN            CONTINUE WITH NORMAL HORN DELAY



                        2364  **      OUT.    - OUTPUT BYTE TO Z-47
                        2365  *
                        2366  *       ENTRY:  (A) = OUTPUT BYTE
                        2367  *
                        2368  *       EXIT:   NONE
                        2369  *
                        2370  *       USE:    NONE
                        2371
006.063                 2372  OUT.    EQU     *
006.063 305             2373          PUSH    B
006.064 107             2374          MOV     B,A             SAVE THE OUTPUT DATA
006.065 072 120 041     2375          LDA     PRIM            GET PORT ADDRESS
006.070 117             2376  OUT.1   MOV     C,A             SET TO REG C
006.071 170             2377          MOV     A,B             GET OUTPUT BYTE DATA BACK
                        2378  *       OUT     (C),A           OUTPUT BYTE
006.072 355 171         2379          DB      355Q,171Q
006.074 301             2380          POP     B
006.075 311             2381          RET



006.100                 2383          ORG     6100A
                        2384  **      TYPMSG - TYPE MESSAGE TO CONSOLE
                        2385  *
                        2386  *       TYPMSG OUTPUTS AN ASCII MESSAGE FROM MEMORY TO THE CONSOLE
                        2387  *       UNTIL A NULL IS SENSED
                        2388  *
                        2389  *       ENTRY   (H,L) = ADDRESS OF MESSAGE
                        2390  *       EXIT    NONE
                        2391  *       USES    A,H,L,F
                        2392
                        2393
006.100 176             2394  TYPMSG  MOV     A,M             GET CHARACTER
006.101 267             2395          ORA     A               SEE IF A NULL
006.102 310             2396          RZ                      IF NULL, EXIT
                        2397
006.103 315 302 003     2398          CALL    WCC             ELSE OUTPUT CHARACTER TO CONSOLE
006.106 043             2399          INX     H               POINT TO NEXT CHARACTER
006.107 030 367         2400          JR      TYPMSG          OUTPUT IT
```

```
                          2402  **      MSG.PR - MESSAGE FOR MONITOR PROMPT
                          2403  *
                          2404  *       CRLF,' H: '
                          2405
                          2406
006.111  015 012 040     2407  MSG.PR  DB      A.CR,A.LF,'  H: ',0


                          2409  **      RDBLCK - INPUT A BLOCK FROM Z-47
                          2410  *
                          2411  *       RDBLCK  READS IN A BLOCK FROM THE DISK CONTROOLER
                          2412  *
                          2413  *       ENTRY:
                          2414  *               HL = LOAD ADDRESS
                          2415  *               B  = COUNT
                          2416  *               C  = SIDE/UNIT/SECTOR
                          2417  *
                          2418  *       EXIT:   NONE
                          2419  *
                          2420  *       USES:   ALL
                          2421
006.121                   2422  RDBLCK  EQU     *
006.121  076 007          2423  RD1     MVI     A,DC.REAR
006.123  315 027 006      2424          CALL    COM             SEND THE COMMAND
006.126  257              2425          XRA     A               FOR TRACK 0
006.127  315 023 006      2426          CALL    DAT             SEND IT TO DISK
006.132  171              2427          MOV     A,C             LOAD SIDE/UNIT/SECTOR
006.133  315 023 006      2428          CALL    DAT             SEND IT TO DISK
                          2429
006.136  315 067 001      2430  RD2     CALL    PIN             INPUT A BYTE FROM DISK
006.141  167              2431          MOV     M,A             STORE IN BUFFER
006.142  043              2432          INX     H               BUFFER TO NEXT ADDRESS
006.143  020 371          2433          DJNZ    RD2
006.145  311              2434          RET                     CONTINUE


                          2436  **      OUT1, - OUTPUT BYTE TO PORT (PRIM+1)
                          2437  *
                          2438  *       ENTRY:  (A) = OUTPUT PORT
                          2439  *
                          2440  *       EXIT:   NONE
                          2441  *
                          2442  *       USE:    NONE
                          2443
006.146                   2444  OUT1,   EQU     *
006.146  305              2445          PUSH    B
006.147  107              2446          MOV     B,A             SAVE THE OUTPUT DATA
006.150  072 120 041      2447          LDA     PRIM            GET PORT ADDRESS
006.153  074              2448          INR     A               SET TO (PRIM+1)
006.154  030 312          2449          JR      OUT.1           GO TO OUTPUT ROUTINE
```

```
                        2451  **      IN1.    - INPUT BYTE FROM (PRIM+1) PORT
                        2452  *
                        2453  *       ENTRY:  NONE
                        2454  *
                        2455  *       EXIT:   (A) = INPUT BYTE
                        2456  *
                        2457  *       USE:    A
                        2458
006.156                 2459  IN1.    EQU     *
006.156 305             2460          PUSH    B
006.157 072 120 041     2461          LDA     PRIM            GET PORT ADDRESS
006.162 074             2462          INR     A               SET TO (PRIM+1)
006.163 030 007         2463          JR      IN.1            GO TO INPUT ROUTINE


006.165                 2465          ORG     6165A
                        2466  **      MSG.GO - (G)O
                        2467  *
                        2468  *       'GO'
                        2469
006.165 157 040 000     2470  MSG.GO  DB      'o ',0


                        2472  **      IN.     - INPUT BYTE FROM PORT (PRIM)
                        2473  *
                        2474  *       ENTRY:  NONE
                        2475  *
                        2476  *       EXIT:   (A) = INPUT BYTE
                        2477  *
                        2478  *       USE:    A
                        2479
006.170                 2480  IN.     EQU     *
006.170 305             2481          PUSH    B
006.171 072 120 041     2482          LDA     PRIM            GET PORT ADDRESS
006.174 117             2483  IN.1    MOV     C,A             SET ADDR. TO REG C
                        2484  *       IN      A,(C)
006.175 355 170         2485          DB      355Q,170Q       INPUT BYTE
006.177 301             2486          POP     B
006.200 311             2487          RET


006.201                 2489          ORG     6201A
                        2490  **      MSG.SUB - (S)UBSTITUTE
                        2491  *
                        2492  *       'SUBSTITUTE'
                        2493
006.201 165 142 163     2494  MSG.SUB DB      'ubstitute ',0
```

```
                        2496  **      MSG.PC - (P)ROGRAM COUNTER
                        2497  *
                        2498  *       'PROGRAM COUNTER'
                        2499
 006.214 162 157 147    2500  MSG.PC  DB      'rogram Counter',0


                        2502  **      MSG.BT - (B)OOT
                        2503  *
                        2504  *       'BOOT'
                        2505
 006.234 157 157 164    2506  MSG.BT  DB      'oot',0
```

```
                        2509  ***   SPEED - ROTATIONAL SPEED TEST FOR 5.25 INCH DISK DRIVE
                        2510  *
                        2511  *     *SPEED* IS USED ONLY FOR GROSS ADJUSTMENT OF DRIVE ROTATIONAL
                        2512  *     SPEED IF THE FIRST READ/WRITE TEST OF THE UNIT FAILS DURING SET UP.
                        2513  *
                        2514  *              USE OF *SPEED* IS AS FOLLOWS:
                        2515  *
                        2516  *                      1.  ENTER 'GO AND THE ENTRY ADDRESS OF *SPEED*
                        2517  *                      2.  ADJUST DRIVE SPEED UNTIL DATA AT DISPLAYED
                        2518  *                          EQUALS 200
                        2519  *                              A.  IF SPEED < 200, TURN ADJUSTMENT CLOCKWISE
                        2520  *                              B.  IF SPEED > 200, TURN COUNTERCLOCKWISE
                        2521  *
                        2522  *     THE ABOVE TEST ADJUSTS SY0:.  TO ADJUST SY1:, USE HDOS


                        2524  **    LABLE EQUIVALENCES
                        2525  *
                        2526  *     I/O PORTS
   000,177              2527  OP.DC   EQU     177Q            DRIVE CONTROL OUTPUT PORT
   000,177              2528  IP.DS   EQU     177Q            DRIVE STATUS INPUT PORT

                        2530  *     MASKS
                        2531  *
   000.001              2532  DS.HOLE EQU     00000001B       DRIVE STATUS SECTOR/INDEX HOLE

                        2534  *     CONSTANTS
                        2535  *
   000.022              2536  ONDRO   EQU     022Q            TURN ON SY0:
```

```
006,240 041 371 006  2538  SPEED   LXI   H,MSG.SPD      OUTPUT SPEED MESSAGE
006,243 315 100 006  2539          CALL  TYPMSG
006,246 076 000      2540          MVI   A,0            SET FLAG AT IOWRK FOR 'WORKING' MESSAGE
006,250 062 002 040  2541          STA   IOWRK
006,253 076 022      2542          MVI   A,ONDRO        TURN ON DRIVE ZERO
006,255 323 177      2543          OUT   OP.DC
006,257 052 033 040  2544  SPEED1  LHLD  TICCNT         GET TICK COUNTER
006,262 174          2545          MOV   A,H            FORM TWO'S COMPLEMENT OF TICK COUNTER
006,263 057          2546          CMA
006,264 127          2547          MOV   D,A            (D,E) = NEGATIVE TICK COUNTER
006,265 175          2548          MOV   A,L
006,266 057          2549          CMA
006,267 074          2550          INR   A
006,270 137          2551          MOV   E,A
006,271 322 275 006  2552          JNC   SPEED2         IF NO CARRY FROM LSB
                     2553
006,274 024          2554          INR   D              ELSE, INCREMENT MSB
006,275 001 000 000  2555  SPEED2  LXI   B,0            ZERO REV COUNTERS
006,300 333 177      2556  SPEED3  IN    IP.DS          INPUT DISK STATUS
006,302 346 001      2557          ANI   DS.HOLE        MASK FOR SECTOR/INDEX PULSES
006,304 312 300 006  2558          JZ    SPEED3         IF NO HOLE PRESENT
                     2559
                     2560  *       HOLE PRESENT, WAIT FOR IT TO LEAVE
                     2561  *
006,307 333 177      2562  SPEED4  IN    IP.DS          GET DISK STATUS
006,311 346 001      2563          ANI   DS.HOLE        GET HOLE PULSES
006,313 302 307 006  2564          JNZ   SPEED4         WAIT UNTIL HOLE IS GONE AND WE HAVE MEDIA
                     2565
006,316 004          2566          INR   B              INCREMENT HOLE COUNTER
006,317 170          2567          MOV   A,B            TEST FOR FIVE REVOLUTIONS
006,320 376 070      2568          CPI   56
006,322 302 300 006  2569          JNZ   SPEED3         NOT FIVE, WAIT FOR MORE HOLES
                     2570
                     2571  *       HAVE FIVE REVS, DISPLAY DIFFERENCE OF TICK COUNTER AND EXPECTED TIME DIF
                     2572  *
006,325 052 033 040  2573          LHLD  TICCNT         GET CURRENT TICK VALUE
006,330 031          2574          DAD   D              SUBTRACT START VALUE
006,331 021 214 376  2575          LXI   D,377377A-500+1+200Q   SUBTRACT 500 FOR REVS, +200Q FOR OFFSET
006,334 031          2576          DAD   D              (H,L) = OFFSET RESULT
006,335 345          2577          PUSH  H              SAVE RESULT
006,336 041 062 007  2578          LXI   H,MSG.WRK      POINT TO 'WORKING' MESSAGE
006,341 072 002 040  2579          LDA   IOWRK          GET 'WORKING' FLAG
006,344 356 001      2580          XRI   1              INVERT LOWER BIT
006,346 062 002 040  2581          STA   IOWRK          SAVE NEW VALUE
006,351 302 357 006  2582          JNZ   SPEED5         IF TO DISPLAY 'WORKING'
                     2583
006,354 041 100 007  2584          LXI   H,MSG.HSS      POINT TO 'HOME', 'SPACES', AND SPEED MSG
006,357 315 100 006  2585  SPEED5  CALL  TYPMSG         OUTPUT MESSAGE
006,362 341          2586          POP   H              GET TEST RESULT
006,363 315 325 005  2587          CALL  TOA.           OUTPUT RESULT TO CONSOLE
006,366 303 257 006  2588          JMP   SPEED1         PERFORM ANOTHER SAMPLE
```

```
                                2590  **      MSG.SPD - SPEED TEST MESSAGE
                                2591  *
                                2592  *       .       Disk drive rotational speed test.
                                2593  *
                                2594  *
                                2595  *                       Drive speed = '
                                2596
006.371  033 105 012            2597  MSG.SPD DB      A.ESC,'E',A.LF
006.374  011 104 151            2598          DB      '       Disk drive rotational speed test.',A.CR,A.LF,A.LF
007.041  011 011 104            2599          DB      '               Drive speed = '
007.061  000                    2600          DB      0


                                2602  **      MSG.WRK - 'WORKING' MESSAGE FOR SPEED TEST
                                2603  *
                                2604  *       DISPLAYS 'WORKING' AT HOME POSITION AND RETURNS CURSOR TO SPEED =
                                2605
007.062  033 110                2606  MSG.WRK DB      A.ESC,'H'       CURSOR HOME
007.064  127 157 162            2607          DB      'Working'
007.073  033 131 043            2608          DB      A.ESC,'Y#>'     CURSOR ADDRESS OF SPEED = VALUE
007.077  000                    2609          DB      0               END MESSAGE


                                2611  **      MSG.HSS - BLANKS 'WORKING' MESSAGE
                                2612  *
                                2613
007.100  033 110                2614  MSG.HSS DB      A.ESC,'H'       CURSOR HOME
007.102  040 040 040            2615          DB      '   '           BLANKS
007.111  033 131 043            2616          DB      A.ESC,'Y#>'     CURSOR ADDRESS OF SPEED = VALUE
007.115  000                    2617          DB      0               END MESSAGE
```

```
                        2620  **      DYMEM - DYNAMIC MEMORY TEST
                        2621  *
                        2622  *       DYMEM TESTS THE DYNAMIC MEMORY IN THE H88/H89 BY PLACING
                        2623  *       A KNOWN PATTERN IN EACH DYNAMIC MEMORY CELL AND THEN
                        2624  *       PERFORMING A READ, INCREMENT, READ SEQUENCE WITH A DELAY
                        2625  *       BETWEEN EACH PASS OF THE TEST
                        2626  *
                        2627  *
                        2628  *       ENTRY   NONE
                        2629  *
                        2630  *       EXIT    ON RESET
                        2631  *
                        2632  *       USES    A,B,C,D,E,H,L,F,A',F',IX,IY
                        2633
                        2634
007.116  076 000        2635  DYMEM   MVI     A,0             MAKE SURE CLOCK AND SINGLE STEP ARE OFF
007.120  323 362        2636          OUT     H88.CTL
                        2637
                        2638  *       DETERMINE END OF MEMORY
                        2639  *
000.001                 2640          IF      .RAM.
                        2643          ELSE
007.122  041 000 040    2644  DYMEM1  LXI     H,START
                        2645          ENDIF
007.125  076 001        2646          MVI     A,1
007.127  066 000        2647  DYMEM2  MVI     M,0             SET RAM TO ZERO
007.131  064            2648          INR     M               SET MEMORY TO ONE
007.132  276            2649          CMP     M               SEE IF (A) = ((H,L))
007.133  040 003        2650          JR      NZ,DYMEM3       IF NOT EQUAL, THE END OF RAM HAS BEEN REACHED
                        2651
007.135  043            2652          INX     H               ELSE, POINT TO NEXT LOCATION IN RAM
007.136  030 367        2653          JR      DYMEM2
                        2654
                        2655
007.140  053            2656  DYMEM3  DCX     H               POINT TO LAST GOOD LOCATION
007.141  353            2657          XCHG                    PUT ENDING ADDRESS IN D,E
007.142  041 324 007    2658          LXI     H,MSG.RAM       OUTPUT ENDING ADDRESS
                        2659
                        2660  *       LD      IX,DY3.3        RETURN ADDRESS
007.145  335 041        2661          DB      MI.LDXA,MI.LDXB
007.147  153 007        2662          DW      DY3.3
                        2663
007.151  030 133        2664          JR      DYMSG
                        2665
007.153  172            2666  DY3.3   MOV     A,D             OUTPUT ADDRESS MSB
                        2667
                        2668  *       LD      IX,DY3.5        RETURN ADDRESS
007.154  335 041        2669          DB      MI.LDXA,MI.LDXB
007.156  163 007        2670          DW      DY3.5
                        2671
007.160  303 160 003    2672          JMP     DYBYT
                        2673
007.163  173            2674  DY3.5   MOV     A,E             LSB
                        2675
                        2676  *       LD      IX,DY3.7        RETURN ADDRESS
007.164  335 041        2677          DB      MI.LDXA,MI.LDXB
```

```
007.166  173 007        2678          DW      DY3.7
                        2679
007.170  303 160 003    2680          JMP     DYBYT
                        2681
007.173  023            2682  DY3.7   INX     D               (D,E) = LAST BYTE OF RAM + 1
                        2683
                        2684  *       TEST MEMORY
                        2685  *
007.174  006 001        2686          MVI     B,1             (B) = CONTENTS OF RAM AFTER SIZING
007.176  041 237 003    2687          LXI     H,MSG.PAS       OUTPUT PASS MESSAGE
                        2688
                        2689  *       LD      IX,DYMEM4       RETURN ADDRESS
007.201  335 041        2690          DB      MI.LDXA,MI.LDXB
007.203  207 007        2691          DW      DYMEM4
                        2692
007.205  030 077        2693          JR      DYMSG
                        2694
000.001                 2695          IF      .RAM.
                        2697          ELSE
007.207  041 000 040    2698  DYMEM4  LXI     H,START         POINT BACK TO BEGINNING OF RAM
                        2699          ENDIF
007.212  176            2700  DYMEM5  MOV     A,M             READ CURRENT CONTENTS
007.213  270            2701          CMP     B               SEE IF CORRECT CONTENTS STILL REMAIN
007.214  302 307 000    2702          JNZ     DYMEM9          FAILURE, SEE IF AT END OF RAM
                        2703
007.217  074            2704          INR     A
007.220  167            2705          MOV     M,A             INCREMENT RAM
007.221  276            2706          CMP     M               SEE IF WRITE WAS SUCCESSFUL
007.222  302 307 000    2707          JNZ     DYMEM9
                        2708
007.225  043            2709          INX     H
007.226  175            2710          MOV     A,L             GET LSB AND TEST FOR REACHING END OF RAM
007.227  273            2711          CMP     E
007.230  040 360        2712          JR      NZ,DYMEM5       IF LSB NOT EQUAL
                        2713
007.232  174            2714          MOV     A,H             CHECK LSB
007.233  272            2715          CMP     D
007.234  040 354        2716          JR      NZ,DYMEM5
                        2717
                        2718  *       HAVE REACHED END OF MEMORY!
                        2719  *       OUTPUT LAST VALUE TESTED
                        2720  *
007.236  046 003        2721          MVI     H,3             OUTPUT 3 BACKSPACES
007.240  076 010        2722          MVI     A,A.BKS
                        2723
007.242                 2724  DYME5.5 EQU     *
                        2725
                        2726  *       LD      IY,DY5.53       RETURN ADDRESS
007.242  375 041        2727          DB      MI.LDYA,MI.LDYB
007.244  251 007        2728          DW      DY5.53
                        2729
007.246  303 143 003    2730          JMP     DYASC
                        2731
007.251  045            2732  DY5.53  DCR     H
007.252  040 366        2733          JR      NZ,DYME5.5
                        2734
```

```
007.254  004              2735            INR     B               SHOW NEXT PASS VALUE
007.255  170              2736            MOV     A,B             VALUE TESTED
                          2737
                          2738  *         LD      IX,DYMEM6       RETURN ADDRESS
007.256  335 041          2739            DB      MI.LDXA,MI.LDXB
007.260  273 000          2740            DW      DYMEM6
                          2741
007.262  303 160 003      2742            JMP     DYBYT
                          2743
                          2744
                          2745  **        !!THE DYNAMIC RAM TEST CONTINUES ELSEWARE!!      **
                          2746  *         !!AND THEN RETURNS TO HERE!!!!!!!!!!!!!!!!!!      **
                          2747
                          2748
007.265  041 000 000      2749  DY10.5    LXI     H,0             DELAY AND DING BELL AGAIN
007.270  006 002          2750            MVI     B,2             2 LOOPS
007.272  045              2751  DYMEM11   DCR     H
007.273  040 375          2752            JR      NZ,DYMEM11
                          2753
007.275  055              2754            DCR     L
007.276  040 372          2755            JR      NZ,DYMEM11
                          2756
007.300  005              2757            DCR     B
007.301  040 367          2758            JR      NZ,DYMEM11
                          2759
007.303  303 360 003      2760            JMP     DYMEM10         AGAIN


                          2762  **        DYMSG - DYNAMIC RAM TEST MESSAGE OUTPUT ROUTINE
                          2763  *
                          2764  *         ENTRY   (H,L) = MESSAGE ADDRESS
                          2765  *                 (IX) = RETURN ADDRESS
                          2766  *
                          2767  *         EXIT    TO (IX)
                          2768  *
                          2769  *         USES    A,H,L,F,IY
                          2770
                          2771
007.306  176              2772  DYMSG     MOV     A,M             GET MESSAGE BYTE
                          2773
                          2774  *         LD      IY,DYMSG.5      RETURN ADDRESS
007.307  375 041          2775            DB      MI.LDYA,MI.LDYB
007.311  316 007          2776            DW      DYMSG.5
                          2777
007.313  303 143 003      2778            JMP     DYASC           OUTPUT ASCII
                          2779
007.316  267              2780  DYMSG.5   ORA     A               SEE IF NULL TO END STRING
007.317  043              2781            INX     H               POINT TO NEXT CHARACTER
007.320  040 364          2782            JR      NZ,DYMSG        IF NOT DONE YET
                          2783
                          2784  *         JP      (IX)            RETURN TO CALLER
007.322  335 351          2785            DB      MI.JIXA,MI.JIXB
```

```
                                2787  **      MSG.RAM - RAM TEST MESSAGE
                                2788  *
                                2789
    007.324  033 105            2790  MSG.RAM DB      A.ESC,'E'
    007.326  104 171 156        2791          DB      'Dynamic RAM test'
    007.346  015 012 012        2792          DB      A.CR,A.LF,A.LF
    007.351  011 040 114        2793          DB      '       LWA = '
    007.361  000                2794          DB      0


                                2796  **      MSG.EQ - EQUALS MESSAGE
                                2797  *
                                2798
    007.362  040 075 040        2799  MSG.EQ  DB      ' = '
    007.365  000                2800          DB      0
                                2801
    007.366  107 101 103        2802          DB      'GAC.'
```

```
                        2805  **      ENTRY POINT FOR FLOPPY DISK ROTATIONAL SPEED TEST
                        2806  *
    000.001             2807          IF      .RAM.
                        2808          ELSE
    000.000             2809          ERRNZ   10000A-6-*      MUST BE SIX BYTES BEFORE END
                        2810          ENDIF
                        2811
    007.372  303 240 006 2812  ESPEED  JMP     SPEED


                        2814  **      ENTRY POINT FOR DYNAMIC MEMORY TEST
                        2815  *
    000.001             2816          IF      .RAM.
                        2817          ELSE
    000.000             2818          ERRNZ   10000A-3-*      MUST BE THREE BYTES BEFORE END
                        2819          ENDIF
                        2820
    007.375  303 116 007 2821  EDYMEM  JMP     DYMEM
                        2822
    000.001             2823          IF      .RAM.
                        2824          ELSE
    000.000             2825          ERRNZ   *-10000A        MUST NOT EXCEED 2K BYTES
                        2826          ENDIF
                        2827
```

```
                    2830  **      THE FOLLOWING ARE CONTROL CELLS AND FLAGS USED BY THE KEYSET
                    2831  *       MONITOR.
                    2832
    040.000         2833          ORG     40000A          8192
    040.000         2834  START   DS      2               DUMP STARTING ADDRESS
    040.002         2835  IOWRK   DS      2               IN OR OUT INSTRUCTION
    040.004         2836  PRSRAM  EQU     *               FOLLOWING CELLS INITIALIZED FROM ROM
    040.004         2837          DS      1               RET
                    2838
    040.005         2839  REGI    DS      1               INDEX OF REGISTER UNDER DISPLAY
    040.006         2840  DSPROT  DS      1               PERIOD FLAG BYTE
    040.007         2841  DSPMOD  DS      1               DISPLAY MODE
                    2842
    040.010         2843  .MFLAG  DS      1               USER FLAG OPTIONS
                    2844  *                               SEE *UO.XXX* BITS DESCRIBED AT FRONT
                    2845
    040.011         2846  CTLFLG  DS      1               FRONT PANEL CONTROL BITS
    040.012         2847  REFIND  DS      1               REFRESH INDEX (0 TO 7)
    000.007         2848  PRSL    EQU     *-PRSRAM        END OF AREA INITIALIZED FROM ROM
                    2849
    040.013         2850  FPLEDS  EQU     *               FRONT PANEL LED PATTERNS
    040.013         2851  ALEDS   DS      1               ADDR 0
    040.014         2852          DS      1               ADDR 1
    040.015         2853          DS      1               ADDR 2
                    2854
    040.016         2855          DS      1               ADDR 3
    040.017         2856          DS      1               ADDR 4
    040.020         2857          DS      1               ADDR 5
                    2858
    040.021         2859  DLEDS   DS      1               DATA 0
    040.022         2860          DS      1               DATA 1
    040.023         2861          DS      1               DATA 2
                    2862
    040.024         2863  ABUSS   DS      2               ADDRESS BUSS
    040.026         2864  RCCA    DS      1               RCC SAVE AREA
    040.027         2865  CRCSUM  DS      2               CRC-16 CHECKSUM
    040.031         2866  TPERRX  DS      2               TAPE ERROR EXIT ADDRESS
    040.033         2867  TICCNT  DS      2               CLOCK TIC COUNTER
                    2868
    040.035         2869  REGPTR  DS      2               REGISETR CONTENTS POINTER
                    2870
    040.037         2871  UIVEC   DS      0               USER INTERRUPT VECTORS
    040.037         2872          DS      3               JUMP TO CLOCK PROCESSOR
    040.042         2873          DS      3               JUMP TO SINGLE STEP PROCESSOR
    040.045         2874          DS      3               JUMP TO I/O 3
    040.050         2875          DS      3               JUMP TO I/O 4
    040.053         2876          DS      3               JUMP TO I/O 5
    040.056         2877          DS      3               JUMP TO I/O 6
    040.061         2878          DS      3               JUMP TO I/O 7
                    2879
                    2880  **      H88/H89 RAM USAGE BEYOND THAT OF H8MTRF
                    2881  *
    040.064         2882  NMIRET  DS      2
                    2883  *
    041.120         2884          ORG     41120A
    041.120         2885  PRIM    DS      1               PRIMARY DEVICE ADDR. PORT
```

```
041,121  2886  TMFG   DS  1      TIMER INTERRUPT FLAG, =1 FOR Z47, =0 FOR H17
041,122  2887  MYCNT  DS  1      COUNTER FOR TIMER INTERRUPT
041,123  2888  AUTOR  DS  1      AUTO BOOT FLAG
041,124  2889  STK    DS  2      STACK POINTER FOR RE-BOOT
         2890
040,066  2891        ORG 40066A
040,066  2892  DATA  DS  1      OUTPUT 362Q DATA SAVE AREA
040,067  2893        END
ASSEMBLY COMPLETE
 2893 STATEMENTS
 0 ERRORS DETECTED
14204 BYTES FREE
```

```
MTR89 - H89 MONITOR  $09.01.00.                                    XREF V1.1
CROSS REFERENCE TABLE                                              PAGE  73

$MOVE   030252      340E    1343
$ZERO   031212      342E    1401
.       000004      533S     732S    910S   1084S   1849S   1850
.MFLAG  040010      732      780     853     859    1208    1324    1409    2843L
.RAM.   000001       42E      44     472     492     500     511     523     544     554     564     578     588
                     600      619     646     674     696     707     727     740     761     847     868     947    1049
                    1060     1076    1270    1278    1477    1497    1515    1544    1765    1780    2640    2695    2807
                    2816     2823
.UIVEC  040037      344E    1373    1410    1421
A.BEL   000007      156E     797     895    1118    1751    2100    2128    2154    2197    2360
A.BKS   000010      157E    2722
A.CR    000015      159E     936    1004    1025    1108    1564    1643    2071    2097    2125    2145    2247
                    2305    2407    2598    2792
A.DEL   000177      161E    1692    2021
A.ESC   000033      160E    2597    2606    2608    2614    2616    2790
A.LF    000012      158E     936     936    1028    1643    2249    2309    2407    2597    2598    2598    2792
                    2792
A.STX   000002      155E
A.SYN   000026      154E
ABUSS   040024      952    2863L
AC.DLY  000156      405E
AID.UNI 041061      343E    1145    1171    1233
ALARM   002136      863    1275E
ALARMB  002166     1276    1304L
ALEDS   040013     2851L
AS.ODD  000100      227E    1189
AS.IDD  000040      228E
AS.SIA  000020      229E
AS.SLW  000003      230E
ATB     004243     1258    1973L
AUTOB   041123     1256    1836    2888L
AUTOBO  001212     1040L   1976
B170    003040     1436    1439    1460E
B174    003012     1437    1440L
BH17    003030     1444    1453L
BOOT    004256      921    1988L
BOOTO   001336     1042    1109    1137L
BOOT2   002335     1411L   1418
BOOT5   001341     1113    1140L
BOOT6   001353     1138    1145L
BOOTA   037132      339E    1341
BOOTAL  000130      338E    1340
BOOTX   004274     1977    1993L
BR19.2  004103     1845L
BR96    004101     1844L
BRTAB   004101     1797    1842E   1850
BSEC    001325     1129E
BSMSG   001041      927L    1130
C.128   000200      259E
C.256   001000      258E
C.DSYN  000375      301E
CB.CLI  000100      167E     202     596     858    1949
CB.MTL  000040      166E     715     777     858    1086
CB.SPK  000200      168E     596     858    1284
CB.SSI  000020      165E     596     715     858    1068    1082    1949
CHAT2   002130     1255    1259L
CKAUTO  002113      879    1252E
CLK4    000270      784     807E
```

```
CLOCK    000201      510      513     766L
CLOCK17  034031      345E    1372    2045
COM      006027     1185     2335E   2424
COM1     006031     2324     2337L
CRCSUM   040027     2865L
CS.UNR   000200      222E
CS.WPD   000100      223E
CTLFLG   040011      533      770     775     780     859    1067    1070    1084    1289    1857    2350    2846L
CUI1     000165      733L     796     808
D.CON    040110      316L    1342
D.DAT    000171      212E
D.DECNT  040264      352E    1232
D.RAM    040240      319L    1399
D.RAML   000037      341E    1400
D.SDP    040206      348E    1349
D.STA    000170      211E     212    1423
D.VEC    040130      318L
DAT      006023     1187     2322E   2426    2428
DATA     040066     1837     1957    2892L
DC.BOOT  000000      234E
DC.CPY   000013      245E
DC.FRM0  000014      246E
DC.FRM1  000015      247E
DC.FRM2  000016      248E
DC.FRM3  000017      249E
DC.LSC   000003      237E
DC.RAD   000004      238E
DC.RAS   000002      236E    1184
DC.REA   000005      239E
DC.REAB  000007      241E    2423
DC.RST   000001      235E
DC.WRD   000011      243E
DC.WRDB  000012      244E
DC.WRI   000006      240E
DC.WRIB  000010      242E
DEV2     003025     1446L    1464
DEVICE   002301     1041     1106    1393E
DF.DI    000040      277E
DF.DS0   000002      273E
DF.DS1   000004      274E
DF.DS2   000010      275E
DF.HD    000001      267E
DF.MO    000020      276E
DF.SD    000010      270E
DF.ST    000100      278E
DF.TO    000002      268E
DF.WG    000001      272E
DF.WP    000004      269E
DF.WR    000200      279E
DLEDS    040021     2859L
DLY      000053      583L
DM.MR    000000      172E
DM.MW    000001      173E
DM.RR    000002      174E
DM.RW    000003      175E
DOD      003122     1549L
DP.DC    000177      265E    1325    1402
DS.HOLE  000001     2532E    2557    2563
```

DSPMOD	040007	2841L						
DSPROT	040006	2840L						
DY10.5	007265	1755	2749L					
DY3.3	007153	2662	2666L					
DY3.5	007163	2670	2674L					
DY3.7	007173	2678	2682L					
DY5.53	007251	2728	2732L					
DY9.3	003315	830	1717L					
DY9.4	003326	1722	1726L					
DY9.5	003335	1730	1734L					
DY9.8	003350	1739	1743L					
DYASC	003143	1577E	1611	1624	1634	1757	2730	2778
DYASC1	003144	1580L	1582					
DYBYT	003160	1597L	1724	1732	1749	2672	2680	2742
DYBYT.2	003202	1609	1613L					
DYBYT.4	003221	1622	1626L					
DYBYT.6	003235	1632	1636E					
DYME5.5	007242	2724E	2733					
DYMEM	007116	1830	2635L	2821				
DYMEM1	007122	2644L						
DYMEM10	003360	1747	1751L	2760				
DYMEM11	007272	2751L	2752	2755	2758			
DYMEM2	007127	2647L	2653					
DYMEM3	007140	2650	2656L					
DYMEM4	007207	821	2691	2698L				
DYMEM5	007212	2700L	2712	2716				
DYMEM6	000273	815L	2740					
DYMEM7	000276	816L	819					
DYMEM9	000307	825L	2702	2707				
DYMSG	007306	832	1741	2664	2693	2772L	2782	
DYMSG.5	007316	2776	2780L					
EDYMEM	007375	2821L						
EIXIT	034027	350E	1413	1415				
ERPTCNT	000012	351E	1231					
ERRMSG	001045	929L	1322					
ERROR	000322	654	801	852E	1326	1693		
ESPEED	007372	2812L						
FPLEDS	040013	2850E						
GO	001222	1030	1054L					
GO.	000063	596L	1054	1997				
GO88	001146	912	1015L					
GO88.1	001177	1018	1027L					
H17	002207	1339E	1453					
H17A	002237	1357L	1359					
H88.CTL	000362	130E	1791	1959	2636			
H88.SW	000362	134E	1253	1430	1798	1828		
H88B.CK	000002	131E	1790					
H88B.SS	000001	132E						
H88S.0	000014	140E	1462					
H88S.4	000003	142E	1443					
H88S.AT	000200	135E	1254					
H88S.BR	000100	136E	1799					
H88S.DV	000020	138E	1432					
H88S.M	000040	137E	1829					
HORN	002140	1283L	2362					
HRN0	002143	585	1286L					
HRN2	002160	1297L	1298					
HRNX	006045	1300	2350L					

```
IN.      006170    966    1205   2338   2480E
IN.1     006174   2463    2483L
IN1.     006156    969    2459E
INIT     000073    490     494    624L   628
INIT0    000000   488L
INIT0.0  000003   489L    1838
INITOX   004000    488    1790L
INITOX1  004050  1821L    1822   1824
INIT1    000107   637L     642
INIT2    000117   644L
INT1     000010   503E
INT2     000020   526E
INT3     000030   549L
INT4     000040   559L
INT5     000050   569L
INT6     000060   593L
INT7     000070   605L
INTXIT   000172   745L     778   1072
IOA      003062   1005    1026   1502L  2072
IOA1     005176   1502    2170L
IOA2     005204  2174L    2192   2201
IOA3     005242   2176    2179   2194L
IOA4     005260   2195    2205L
IOB      003066  1520L
IOB1     003070  1521L    1528   1560   1567
IOB1.5   003126   1539    1557L
IOB2     003135   1526    1564L
IOC      005301   2081    2116   2230L
IOWRK    040002   2070    2541   2579   2581   2835L
IP.DS    000177  2528E    2556   2562
IP.PAD   000360   124E
IP.TPC   000371   147E
IP.TPD   000370   149E
IROC     005150    994     998   1017   2067   2144L  2156
IROC1    005166   2149    2154L
LRA      003047  1482L
LRA.     003052    789     988   1022   1483L  1975   1991
MI.ANI   000346   184E
MI.EXAF  000010   191E    1114   1116   1122   1394   1433   1435   1438   1579   1585   2027   2040
                  2043
MI.HLT   000166   179E     795
MI.IN    000333   181E    1929   1938
MI.JIXA  000335   192E    1638   2785
MI.JIXB  000351   193E    1638   2785
MI.JIYA  000375   194E    1588
MI.JIYB  000351   195E    1588
MI.JMP   000303   186E    1411
MI.LDA   000072   183E
MI.LDXA  000335   187E     829   1721   1729   1738   1746   2661   2669   2677   2690   2739
MI.LDXB  000041   188E     829   1721   1729   1738   1746   2661   2669   2677   2690   2739
MI.LDYA  000375   189E    1608   1621   1631   1754   2727   2775
MI.LDYB  000041   190E    1608   1621   1631   1754   2727   2775
MI.LXID  000021   185E
MI.OUT   000323   182E    1926   1945
MI.RET   000311   180E    1778
MSG.BT   006234   1988    2506L
MSG.EQ   007362   1735    2799L
MSG.ERR  001047    826     936L
```

MTR89 - H89 MONITOR #09.01.00. XREF V1.1
CROSS REFERENCE TABLE PAGE 77

MSG.GO	006165	1015	2470L						
MSG.HSS	007100	2584	2614L						
MSG.PAS	003237	1643L	2687						
MSG.PC	006214	985	2500L						
MSG.PR	006111	1259	2407L						
MSG.RAM	007324	2658	2790L						
MSG.SPD	006371	2538	2597L						
MSG.SUB	006201	2065	2494L						
MSG.WRK	007062	2578	2606L						
MTR	000344	873E	1087						
MTR.15	000354	880L	1260						
MTR.2	000357	882L	897						
MTR.3	000371	886L	893						
MTR.4	001014	887	899L						
MTR1	000345	876E	877						
MTRA	001025	884	909E	923					
MTRAL	00000A	885	923E						
MYCNT	041122	1404	2032	2887L					
NB7	001316	1115	1122L						
NBOOT	001261	1105L	1992						
NBOOTO	001262	1106L	1133						
NMI	004116	705	1898L						
NMI0.5	004154	1916	1919	1924L					
NMI1	004173	1911	1936L						
NMI1.5	004206	1939	1945L						
NMI2	004212	1948L							
NMI2.2	004225	1955	1957L						
NMI2.5	004236	1922	1927	1930	1946	1963L			
NMI3	004237	1934	1943	1965L					
NMIENT	000146	705L							
NMIRET	040064	1899	1906	2882L					
NODEV	002171	1148	1207	1321E	1368	2022	2036		
NOISE	006053	1304	2360L						
ONDRO	000022	2536E	2542						
OP.CTL	000360	125E	771	1069	1083				
OP.DC	000177	2527E	2543						
OP.DIG	000360	126E							
OP.SEG	000361	127E							
OP.TPC	000371	148E	660						
OP.TPD	000370	150E							
OUT	006063	1180	2372E						
OUT.1	006070	2376L	2449						
OUT1	006146	2342	2444E						
PCA	001103	918	985L						
PCA1	001137	995	1003L						
PIN	001067	965E	968	1188	2430				
PRIM	041120	1424	1441	2375	2447	2461	2482	2885L	
PRSL	000007	489	2848E						
PRSRAM	040004	489	2836E	2848					
PRSROM	003371	1771E	1834						
R.ABORT	033366	346E	1363						
R.READ	034077	347E	1367						
R.SDP	002071	1230E	1348	1356					
R.SDP1	002110	1236	1238L						
RCC	003262	882	1107	1521	1684E	2144	2174	2230	2304
RCC1	003262	1686L	1688						
RCC2	003270	1690L							
RCCA	040026	2864L							

```
RCK      003260  1663E
RD1      006121  2423L
RD2      006136  2430L  2433
RDBLCK   006121  1198   1201  2422E
REFIND   040012  2847L
REGI     040005  1482   2839L
REGPTR   040035   719    861  1485  2869L
RESET    002003  1179L
RETRY    002063  1222L  2041
ROMBOOT  030000   311E
S.DON    000040   215E  2336
S.DTR    000200   216E   967  2323
S.ERR    000001   214E  1206
S.GRT    024000   308E
S.GRT1   025000   309E
S.INT    040343   321L
S.SOVR   041146   323L   325
S.VAL    040277   320L
SAE      001063   952L
SAVALL   000132   508    531   679L
SAVALLR  000151   709    712E  1859
SAVALLX  004105   694   1854E
SC.ACE   000350   404E  1580  1586  1686  1690  1707  1712  1796  1808  1811  1813  1815
                  2015  2019
SC.UART  000372   364E
SDF3     036073   349E  1238
SECOND   003007  1434   1438L
SECSCR   026000   310E
SINCR    004000   630E   632   633
SPEED    006240  2538L  2812
SPEED1   006257  2544L  2588
SPEED2   006275  2552   2555L
SPEED3   006300  2556L  2558  2569
SPEED4   006307  2562L  2564
SPEED5   006357  2582   2585L
SST1     001235   597   1070L
SSTEP    001225  1065E
STACK    042200   327E
STACKL   001032   325E
START    040000   633   2644  2698  2834L
START1   001265  1107L  1120
STK      041124  1167   1222  2889L
STPRTN   001244   534   1081E
SUBM     004370   915   2065L
SUBM1    005013  2075L  2088  2095
SUBM2    005027  2081L  2102
SUBM3    005042  2087L  2120
SUBM4    005046  2085   2090L
SUBM5    005053  2093L  2123
SUBM6    005062  2091   2097L
SUBM7    005075  2082   2104L
SUBM8    005077  2106L  2117
SUBM9    005115  2116L  2130
SYDD     040130   317E
TICCNT   040033   766    768  1293  1403  2028  2544  2573  2867L
TMFG     041121  1407   1455  2024  2032  2886L
TMOUT    004302  1420   2014E
TMOUT1   004345  2037L  2039
```

TMOUT2	004361	2030	2038	2043L									
TMOUT3	004363	2042	2045L										
TMOUT4	004321	2017	2024L										
TOA	005313	997	2075	2247L									
TOA.	005325	2252L	2587										
TOB	005343	2077	2253	2255	2270L								
TOB1	005353	2277L	2285										
TPERRX	040031	2866L											
TYPMSG	006100	880	986	1016	1131	1323	1989	2066	2394L	2400	2539	2585	
UC.2SB	000004	430E											
UC.5BW	000000	426E											
UC.6BW	000001	427E											
UC.7BW	000002	428E											
UC.8BW	000003	429E	1812										
UC.BI	000020	449E											
UC.CTS	000020	458E											
UC.DCS	000001	454E											
UC.DDR	000002	455E											
UC.DLA	000200	435E	1795										
UC.DR	000001	445E	1687	2016									
UC.DRL	000010	457E											
UC.DSR	000040	459E											
UC.DTR	000001	438E											
UC.EDA	000001	416E											
UC.EPS	000020	432E											
UC.FE	000010	448E											
UC.IID	000006	423E											
UC.IIP	000001	422E											
UC.LOO	000020	442E											
UC.MSI	000010	419E											
UC.OR	000002	446E											
UC.OU1	000004	440E											
UC.OU2	000010	441E											
UC.PE	000004	447E											
UC.PEN	000010	431E											
UC.RI	000100	460E											
UC.RLS	000200	461E											
UC.RSI	000004	418E											
UC.RTS	000002	439E											
UC.SB	000100	434E											
UC.SKP	000040	433E											
UC.TER	000004	456E											
UC.THE	000040	450E	1581	1708									
UC.TRE	000002	417E											
UC.TSE	000100	451E											
UCI.ER	000020	386E											
UCI.IE	000002	388E											
UCI.IR	000100	384E											
UCI.RE	000004	387E											
UCI.RO	000040	385E											
UCI.TE	000001	389E											
UDR	000000	361E											
UF.FCT	000100	294E											
UF.RDA	000001	291E											
UF.ROR	000002	292E											
UF.RPE	000004	293E											
UF.TBM	000200	295E											
UIVEC	040037	549	559	569	593	605	736	1088	2871L				

```
UMI.16X 000002  379E  659
UMI.1B  000100  369E  659
UMI.1X  000001  378E
UMI.2B  000300  371E
UMI.64X 000003  380E
UMI.HB  000200  370E
UMI.L5  000000  374E
UMI.L6  000004  375E
UMI.L7  000010  376E
UMI.L8  000014  377E  659
UMI.PA  000020  373E
UMI.PE  000040  372E
UNT.0   000000  253E
UNT.1   000040  254E
UNT.2   000100  255E
UNT.3   000140  256E
UO.CLK  000001  204E  734  1408
UO.DDU  000002  203E  855
UO.HLT  000200  201E  782
UO.NFR  000100  202E  855
UP.DP   000174  285E  1440
UP.FC   000175  286E
UP.SC   000176  288E
UP.SR   000176  289E
UP.ST   000175  287E
UR.DLL  000000  411E  1811
UR.DLM  000001  413E  1808
UR.IER  000001  415E  1815
UR.IIR  000002  421E
UR.LCR  000003  425E  1796  1813
UR.LSR  000005  444E  1580  1686  1707  2015
UR.MCR  000004  437E
UR.MSR  000006  453E
UR.RBR  000000  407E  1690  2019
UR.THR  000000  409E  1586  1712
USERFWA 042200  328E  1196  1209  1364  1374
USR     000001  362E
USR.FE  000040  393E
USR.OE  000020  394E
USR.PE  000010  395E
USR.RXR 000002  397E
USR.TXE 000004  396E
USR.TXR 000001  398E
W.RES   000002  218E  1179
WCC     003302  798   800   896   899   1027  1029  1119  1140  1532  1706L 2079  2093
                2101  2106  2129  2155  2181  2198  2205  2248  2250  2258  2282  2289  2308
                2310  2361  2398
WCC1    003303  1707L 1709
WCR     006003  1143  2304L 2306
WHD     036235  335E  1358
WNH     036271  336E  1357
WRONG   001307  1111  1117E 1125
WTDON1  006032  2338L 2340
Z47     001364  1164E 1425  1449
Z47A    001370  1169E 1224
```

22314 BYTES FREE

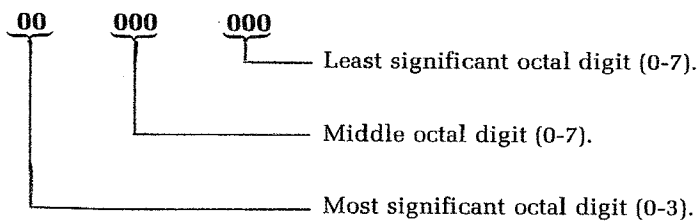
APPENDIX B

OCTAL DEFINITIONS

Binary numbers are converted to octal format for display. The following table shows binary to octal conversion.

BINARY NUMBER	OCTAL DIGIT
000	0
001	1
010	2
011	3
100	4
101	5
110	6
111	7

Each byte is displayed as two-and-one-half octal digits. The octal numbers lie in the range of 000 to 377 for binary numbers in the range 00000000 to 11111111, as shown below.



NOTE: As there are only eight bits in a byte, the most significant octal digit only represents two bits and is therefore displayed as 0 to 3. If the user should inadvertently enter the octal digits 4 to 7 into the most significant digit, the most significant bit is lost. Losing this bit converts 4 through 7 into the digits 0 through 3 respectively.

Also note that 16-bit numbers, such as memory addresses and certain register contents, are displayed as two eight-bit numbers. Therefore, the representation of 16-bit numbers is made up of **two** groups of three octal numbers in the range of 000 to 377. This representation of 16-bit binary numbers is known as offset octal or **split-octal**, and is used consistantly for displays of 16-bit numbers.

Split-octal must not be confused with octal. For example:

```
11 111 111   11 111 111      A 16-bit binary number
 |   |   |    |   |   |
 3   7   7    3   7   7      Split-octal representation (377  377)

 1 111 111 111 111 111       A 16-bit binary number
 |   |   |   |   |   |
 1   7   7   7   7   7       True Octal representation (177777)
```

The lower example shows true octal representation of a 16-bit binary number. True octal representation is never used in standard Zenith Data Systems software. Occassionally you will see split-octal numbers printed with a decimal point separating the upper and lower bytes. For example:

```
377.377
Hi Byte  Lo Byte
```

Note that 001.000 follows 000.377.