

# RL11,RLV11

CONTROLLER TEST PART 2  
CZRLBB0

AH-E041B-MC

COPYRIGHT 77-78

FICHE 1 OF 1

JAN 1979

**digital**

MADE IN USA

|      |      |      |      |      |      |      |      |      |      |      |      |
|------|------|------|------|------|------|------|------|------|------|------|------|
| 1    | 2    | 3    | 4    | 5    | 6    | 7    | 8    | 9    | 10   | 11   | 12   |
| 13   | 14   | 15   | 16   | 17   | 18   | 19   | 20   | 21   | 22   | 23   | 24   |
| 25   | 26   | 27   | 28   | 29   | 30   | 31   | 32   | 33   | 34   | 35   | 36   |
| 37   | 38   | 39   | 40   | 41   | 42   | 43   | 44   | 45   | 46   | 47   | 48   |
| 49   | 50   | 51   | 52   | 53   | 54   | 55   | 56   | 57   | 58   | 59   | 60   |
| 61   | 62   | 63   | 64   | 65   | 66   | 67   | 68   | 69   | 70   | 71   | 72   |
| 73   | 74   | 75   | 76   | 77   | 78   | 79   | 80   | 81   | 82   | 83   | 84   |
| 85   | 86   | 87   | 88   | 89   | 90   | 91   | 92   | 93   | 94   | 95   | 96   |
| 97   | 98   | 99   | 100  | 101  | 102  | 103  | 104  | 105  | 106  | 107  | 108  |
| 109  | 110  | 111  | 112  | 113  | 114  | 115  | 116  | 117  | 118  | 119  | 120  |
| 121  | 122  | 123  | 124  | 125  | 126  | 127  | 128  | 129  | 130  | 131  | 132  |
| 133  | 134  | 135  | 136  | 137  | 138  | 139  | 140  | 141  | 142  | 143  | 144  |
| 145  | 146  | 147  | 148  | 149  | 150  | 151  | 152  | 153  | 154  | 155  | 156  |
| 157  | 158  | 159  | 160  | 161  | 162  | 163  | 164  | 165  | 166  | 167  | 168  |
| 169  | 170  | 171  | 172  | 173  | 174  | 175  | 176  | 177  | 178  | 179  | 180  |
| 181  | 182  | 183  | 184  | 185  | 186  | 187  | 188  | 189  | 190  | 191  | 192  |
| 193  | 194  | 195  | 196  | 197  | 198  | 199  | 200  | 201  | 202  | 203  | 204  |
| 205  | 206  | 207  | 208  | 209  | 210  | 211  | 212  | 213  | 214  | 215  | 216  |
| 217  | 218  | 219  | 220  | 221  | 222  | 223  | 224  | 225  | 226  | 227  | 228  |
| 229  | 230  | 231  | 232  | 233  | 234  | 235  | 236  | 237  | 238  | 239  | 240  |
| 241  | 242  | 243  | 244  | 245  | 246  | 247  | 248  | 249  | 250  | 251  | 252  |
| 253  | 254  | 255  | 256  | 257  | 258  | 259  | 260  | 261  | 262  | 263  | 264  |
| 265  | 266  | 267  | 268  | 269  | 270  | 271  | 272  | 273  | 274  | 275  | 276  |
| 277  | 278  | 279  | 280  | 281  | 282  | 283  | 284  | 285  | 286  | 287  | 288  |
| 289  | 290  | 291  | 292  | 293  | 294  | 295  | 296  | 297  | 298  | 299  | 300  |
| 301  | 302  | 303  | 304  | 305  | 306  | 307  | 308  | 309  | 310  | 311  | 312  |
| 313  | 314  | 315  | 316  | 317  | 318  | 319  | 320  | 321  | 322  | 323  | 324  |
| 325  | 326  | 327  | 328  | 329  | 330  | 331  | 332  | 333  | 334  | 335  | 336  |
| 337  | 338  | 339  | 340  | 341  | 342  | 343  | 344  | 345  | 346  | 347  | 348  |
| 349  | 350  | 351  | 352  | 353  | 354  | 355  | 356  | 357  | 358  | 359  | 360  |
| 361  | 362  | 363  | 364  | 365  | 366  | 367  | 368  | 369  | 370  | 371  | 372  |
| 373  | 374  | 375  | 376  | 377  | 378  | 379  | 380  | 381  | 382  | 383  | 384  |
| 385  | 386  | 387  | 388  | 389  | 390  | 391  | 392  | 393  | 394  | 395  | 396  |
| 397  | 398  | 399  | 400  | 401  | 402  | 403  | 404  | 405  | 406  | 407  | 408  |
| 409  | 410  | 411  | 412  | 413  | 414  | 415  | 416  | 417  | 418  | 419  | 420  |
| 421  | 422  | 423  | 424  | 425  | 426  | 427  | 428  | 429  | 430  | 431  | 432  |
| 433  | 434  | 435  | 436  | 437  | 438  | 439  | 440  | 441  | 442  | 443  | 444  |
| 445  | 446  | 447  | 448  | 449  | 450  | 451  | 452  | 453  | 454  | 455  | 456  |
| 457  | 458  | 459  | 460  | 461  | 462  | 463  | 464  | 465  | 466  | 467  | 468  |
| 469  | 470  | 471  | 472  | 473  | 474  | 475  | 476  | 477  | 478  | 479  | 480  |
| 481  | 482  | 483  | 484  | 485  | 486  | 487  | 488  | 489  | 490  | 491  | 492  |
| 493  | 494  | 495  | 496  | 497  | 498  | 499  | 500  | 501  | 502  | 503  | 504  |
| 505  | 506  | 507  | 508  | 509  | 510  | 511  | 512  | 513  | 514  | 515  | 516  |
| 517  | 518  | 519  | 520  | 521  | 522  | 523  | 524  | 525  | 526  | 527  | 528  |
| 529  | 530  | 531  | 532  | 533  | 534  | 535  | 536  | 537  | 538  | 539  | 540  |
| 541  | 542  | 543  | 544  | 545  | 546  | 547  | 548  | 549  | 550  | 551  | 552  |
| 553  | 554  | 555  | 556  | 557  | 558  | 559  | 560  | 561  | 562  | 563  | 564  |
| 565  | 566  | 567  | 568  | 569  | 570  | 571  | 572  | 573  | 574  | 575  | 576  |
| 577  | 578  | 579  | 580  | 581  | 582  | 583  | 584  | 585  | 586  | 587  | 588  |
| 589  | 590  | 591  | 592  | 593  | 594  | 595  | 596  | 597  | 598  | 599  | 600  |
| 601  | 602  | 603  | 604  | 605  | 606  | 607  | 608  | 609  | 610  | 611  | 612  |
| 613  | 614  | 615  | 616  | 617  | 618  | 619  | 620  | 621  | 622  | 623  | 624  |
| 625  | 626  | 627  | 628  | 629  | 630  | 631  | 632  | 633  | 634  | 635  | 636  |
| 637  | 638  | 639  | 640  | 641  | 642  | 643  | 644  | 645  | 646  | 647  | 648  |
| 649  | 650  | 651  | 652  | 653  | 654  | 655  | 656  | 657  | 658  | 659  | 660  |
| 661  | 662  | 663  | 664  | 665  | 666  | 667  | 668  | 669  | 670  | 671  | 672  |
| 673  | 674  | 675  | 676  | 677  | 678  | 679  | 680  | 681  | 682  | 683  | 684  |
| 685  | 686  | 687  | 688  | 689  | 690  | 691  | 692  | 693  | 694  | 695  | 696  |
| 697  | 698  | 699  | 700  | 701  | 702  | 703  | 704  | 705  | 706  | 707  | 708  |
| 709  | 710  | 711  | 712  | 713  | 714  | 715  | 716  | 717  | 718  | 719  | 720  |
| 721  | 722  | 723  | 724  | 725  | 726  | 727  | 728  | 729  | 730  | 731  | 732  |
| 733  | 734  | 735  | 736  | 737  | 738  | 739  | 740  | 741  | 742  | 743  | 744  |
| 745  | 746  | 747  | 748  | 749  | 750  | 751  | 752  | 753  | 754  | 755  | 756  |
| 757  | 758  | 759  | 760  | 761  | 762  | 763  | 764  | 765  | 766  | 767  | 768  |
| 769  | 770  | 771  | 772  | 773  | 774  | 775  | 776  | 777  | 778  | 779  | 780  |
| 781  | 782  | 783  | 784  | 785  | 786  | 787  | 788  | 789  | 790  | 791  | 792  |
| 793  | 794  | 795  | 796  | 797  | 798  | 799  | 800  | 801  | 802  | 803  | 804  |
| 805  | 806  | 807  | 808  | 809  | 810  | 811  | 812  | 813  | 814  | 815  | 816  |
| 817  | 818  | 819  | 820  | 821  | 822  | 823  | 824  | 825  | 826  | 827  | 828  |
| 829  | 830  | 831  | 832  | 833  | 834  | 835  | 836  | 837  | 838  | 839  | 840  |
| 841  | 842  | 843  | 844  | 845  | 846  | 847  | 848  | 849  | 850  | 851  | 852  |
| 853  | 854  | 855  | 856  | 857  | 858  | 859  | 860  | 861  | 862  | 863  | 864  |
| 865  | 866  | 867  | 868  | 869  | 870  | 871  | 872  | 873  | 874  | 875  | 876  |
| 877  | 878  | 879  | 880  | 881  | 882  | 883  | 884  | 885  | 886  | 887  | 888  |
| 889  | 890  | 891  | 892  | 893  | 894  | 895  | 896  | 897  | 898  | 899  | 900  |
| 901  | 902  | 903  | 904  | 905  | 906  | 907  | 908  | 909  | 910  | 911  | 912  |
| 913  | 914  | 915  | 916  | 917  | 918  | 919  | 920  | 921  | 922  | 923  | 924  |
| 925  | 926  | 927  | 928  | 929  | 930  | 931  | 932  | 933  | 934  | 935  | 936  |
| 937  | 938  | 939  | 940  | 941  | 942  | 943  | 944  | 945  | 946  | 947  | 948  |
| 949  | 950  | 951  | 952  | 953  | 954  | 955  | 956  | 957  | 958  | 959  | 960  |
| 961  | 962  | 963  | 964  | 965  | 966  | 967  | 968  | 969  | 970  | 971  | 972  |
| 973  | 974  | 975  | 976  | 977  | 978  | 979  | 980  | 981  | 982  | 983  | 984  |
| 985  | 986  | 987  | 988  | 989  | 990  | 991  | 992  | 993  | 994  | 995  | 996  |
| 997  | 998  | 999  | 1000 | 1001 | 1002 | 1003 | 1004 | 1005 | 1006 | 1007 | 1008 |
| 1009 | 1010 | 1011 | 1012 | 1013 | 1014 | 1015 | 1016 | 1017 | 1018 | 1019 | 1020 |
| 1021 | 1022 | 1023 | 1024 | 1025 | 1026 | 1027 | 1028 | 1029 | 1030 | 1031 | 1032 |
| 1033 | 1034 | 1035 | 1036 | 1037 | 1038 | 1039 | 1040 | 1041 | 1042 | 1043 | 1044 |
| 1045 | 1046 | 1047 | 1048 | 1049 | 1050 | 1051 | 1052 | 1053 | 1054 | 1055 | 1056 |
| 1057 | 1058 | 1059 | 1060 | 1061 | 1062 | 1063 | 1064 | 1065 | 1066 | 1067 | 1068 |
| 1069 | 1070 | 1071 | 1072 | 1073 | 1074 | 1075 | 1076 | 1077 | 1078 | 1079 | 1080 |
| 1081 | 1082 | 1083 | 1084 | 1085 | 1086 | 1087 | 1088 | 1089 | 1090 | 1091 | 1092 |
| 1093 | 1094 | 1095 | 1096 | 1097 | 1098 | 1099 | 1100 | 1101 | 1102 | 1103 | 1104 |
| 1105 | 1106 | 1107 | 1108 | 1109 | 1110 | 1111 | 1112 | 1113 | 1114 | 1115 | 1116 |
| 1117 | 1118 | 1119 | 1120 | 1121 | 1122 | 1123 | 1124 | 1125 | 1126 | 1127 | 1128 |
| 1129 | 1130 | 1131 | 1132 | 1133 | 1134 | 1135 | 1136 | 1137 | 1138 | 1139 | 1140 |
| 1141 | 1142 | 114  |      |      |      |      |      |      |      |      |      |

IDENTIFICATION

B 1

SEQ 0001

PRODUCT CODE: AC-E040B-MC  
PRODUCT NAME: CZRLBB0 RL11/RLV11 CONTROLLER TEST PART 2  
DATE CREATED: 11-OCT-78  
MAINTAINER: DIAGNOSTIC ENGINEERING  
AUTHOR: D. DEKNIS

THE INFORMATION IN THIS DOCUMENT IS SUBJECT TO CHANGE WITHOUT NOTICE AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT CORPORATION. DIGITAL EQUIPMENT CORPORATION ASSUMES NO RESPONSIBILITY FOR ANY ERRORS THAT MAY APPEAR IN THIS MANUAL.

THE SOFTWARE DESCRIBED IN THIS DOCUMENT IS FURNISHED TO THE PURCHASER UNDER A LICENSE FOR USE ON A SINGLE COMPUTER SYSTEM AND CAN BE COPIED (WITH INCLUSION OF DIGITAL'S COPYRIGHT NOTICE) ONLY FOR USE IN SUCH SYSTEM, EXCEPT AS MAY OTHERWISE BE PROVIDED IN WRITING BY DIGITAL.

DIGITAL EQUIPMENT CORPORATION ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS SOFTWARE ON EQUIPMENT THAT IS NOT SUPPLIED BY DIGITAL.

COPYRIGHT (C) 1977, 1978, DIGITAL EQUIPMENT CORPORATION

TABLE OF CONTENTS

|       |                                    |
|-------|------------------------------------|
| 1.0   | GENERAL INFORMATION                |
| 1.1   | PROGRAM ABSTRACT                   |
| 1.2   | SYSTEM REQUIREMENTS                |
| 1.3   | RELATED DOCUMENTS AND STANDARDS    |
| 1.4   | DIAGNOSTIC HIERARCHY PREREQUISITES |
| 1.5   | ASSUMPTIONS                        |
| 2.0   | OPERATING INSTRUCTIONS             |
| 2.1   | HOW TO RUN THIS DIAGNOSTIC         |
| 2.1.1 | THE SIX STEPS OF EXECUTION         |
| 2.1.2 | SAMPLE RUN-THROUGH                 |
| 2.2   | HOW TO CREATE A CHAINABLE FILE     |
| 2.3   | DETAILS OF COMMANDS AND SYNTAX     |
| 2.3.1 | TABLE OF COMMAND VALIDITY          |
| 2.3.2 | COMMAND SYNTAX                     |
| 2.4   | EXTENDED P-TABLE DIALOGUE          |
| 2.5   | HARDWARE PARAMETERS                |
| 2.6   | SOFTWARE PARAMETERS                |
| 3.0   | ERROR INFORMATION                  |
| 4.0   | PERFORMANCE AND PROGRESS REPORTS   |
| 5.0   | DEVICE INFORMATION TABLES          |
| 6.0   | TEST SUMMARIES                     |

## 1.0 GENERAL INFORMATION

### 1.1 PROGRAM ABSTRACT

#### 1.1.1 STRUCTURE OF PROGRAM

-----

THIS DIAGNOSTIC OCCUPIES 14.5K WORDS OF MEMORY AND IS COMPATIBLE WITH BOTH XXDP AND ACT. IT CAN BE RUN STANDALONE UNDER XXDP, AND CAN BE CHAINED UNDER XXDP, ACT AND APT IN ACT MODE (SEE "CREATE CORE IMAGE" COMMAND BELOW FOR DETAILS OF CHAINING PROCEDURE). IT IS A SINGLE PROGRAM FROM THE STANDPOINT OF THE DIAGNOSTIC USER, BUT WE HAVE INCORPORATED INTO IT A CONTROL MODULE WHICH WILL LATER BE RELEASED INDEPENDENTLY AS A DIAGNOSTIC SUPERVISOR.

WHEN THIS DIAGNOSTIC IS STARTED AT ADDRESS 200, CONTROL GOES FIRST TO THE SUPERVISOR PORTION, WHICH WILL ASK CERTAIN "HARD CORE" QUESTIONS ABOUT THE ENVIRONMENT. THEN IT WILL ENTER COMMAND MODE, INDICATED BY A PROMPT CHARACTER (DS B>). AT COMMAND MODE THE OPERATOR MAY ENTER ANY OF SEVERAL COMMANDS AS DESCRIBED BELOW.

THE SUPERVISOR CODING FOLLOWS IMMEDIATELY THE DIAGNOSTIC TEST CODING, BUT THE SUPERVISOR LISTING HAS BEEN SUPPRESSED FOR GENERAL DISTRIBUTION. A LIMITED DISTRIBUTION HAS BEEN MADE TO FIELD SERVICE OF THE SUPERVISOR ASSEMBLY LISTING, AND IT MAY BE CONSULTED IN EVENT OF A SOFTWARE PROBLEM.

#### 1.1.2 DIAGNOSTIC INFORMATION

THE RL11/RLV11 CONTROLLER TEST (PART 2) IS A PDP-11 (LSI-11) BASED PROGRAM THAT WILL TEST THE CONTROLLER. IT COMPLIMENTS PART 1 BY EXTENDING THE TEST COVERAGE TO INCLUDE WRITE DATA, READ DATA, WRITE CHECK AND READ DATA WITHOUT HEADER COMPARE. IT IS AIMED AT FULLY TESTING THE CONTROLLER IN THESE AREAS, BUT BY DEFAULT ALSO EXERCISES THE DRIVE. THE TEST COVERAGE OF THE PROGRAM IS EXTREMELY HIGH.

### 1.2 SYSTEM REQUIREMENTS

#### 1.2.1 HARDWARE REQUIREMENTS

-----

PDP-11/LSI-11 PROCESSOR WITH 16K OR MORE OF MEMORY  
CONSOLE DEVICE (LA30, LA36, VT50, ETC.)  
RL11/RLV11 CONTROLLER(S)  
1 - 8 RLO1 DRIVES  
1 - 8 RLO1K CARTRIDGES WITH BAD SECTOR FILE  
KW11P, KW11L (OPTIONAL)  
LINEPRINTER (OPTIONAL)

#### 1.2.2 SOFTWARE REQUIREMENTS

-----

CZRLBB RL11/RLV11 CTLR 2

(FORMERLY MD-11-DZRLB-A)

### 1.3 RELATED DOCUMENTS AND STANDARDS

RL01 USERS MANUAL (EK-RL01-UG-PRE)  
XXDP USERS MANUAL

### 1.4 DIAGNOSTIC HIERARCHY PREREQUISITES

THE RL01 SUBSYSTEM SHOULD HAVE SUCCESSFULLY RUN THE FOLLOWING PROGRAMS:

CZRLAB0 RL11/RLV11 RL01 CONTROLLER TEST (PART 1)

### 1.5 ASSUMPTIONS

THE HARDWARE OTHER THAN THE RL01 SUBSYSTEM IS ASSUMED TO WORK PROPERLY. FALSE ERRORS MAY BE REPORTED IF THE PROCESSOR, ETC., DO NOT FUNCTION PROPERLY.

### 2.0 OPERATING INSTRUCTIONS

#### 2.1 HOW TO RUN THIS DIAGNOSTIC

##### 2.1.1 THE SIX STEPS OF EXECUTION

THIS DIAGNOSTIC SHOULD BE LOADED AND STARTED USING NORMAL XXDP PROCEDURES. THE START COMMAND SHOULD NOT SPECIFY AN ADDRESS, BECAUSE THE DIAGNOSTIC HAS THE PROPER TRANSFER ADDRESS CODED INTO IT.

WHEN THIS DIAGNOSTIC IS STARTED, THE FOLLOWING STEPS WILL OCCUR:

\*\*\*\*\*  
\* STEP 1 \*  
\*\*\*\*\*

A SHORT SERIES OF "HARDWARE QUESTIONS" WILL BE ASKED:

| QUESTION         | MEANING                                |
|------------------|----------------------------------------|
| L-CLK (L) N ?    | IS THERE AN L-CLOCK?                   |
| P-CLK (L) N ?    | IS THERE A P-CLOCK?                    |
| 50HZ (L) N ?     | IS THE POWER 50 CYCLES (AS IN EUROPE)? |
| LSI (L) N ?      | IS MACHINE AN LSI?                     |
| LPT (L) N ?      | IS THERE A LINE PRINTER?               |
| MEM (K) (D) 16 ? | HOW MANY K OF MEMORY ARE THERE?        |

THE DEFAULTS (SHOWN AFTER EACH QUESTION) CAN BE SELECTED BY HITTING CARRIAGE RETURN. IT IS POSSIBLE THAT NOT ALL OF THE QUESTIONS WILL BE ASKED: FOR EXAMPLE, IF YOU SAY "YES" TO THE L-CLOCK QUESTION, THE P-CLOCK QUESTION WILL NOT BE ASKED.

IF NEITHER P OR L CLOCK ARE ANSWERED YES THE OPERATOR WILL BE ASKED TO TYPE TWO CHARACTERS 4 SECONDS APART.

\*\*\*\*\*  
\* STEP 2 \*  
\*\*\*\*\*

WHEN YOU HAVE ANSWERED ALL THE HARDCORE QUESTIONS, THE DIAGNOSTIC WILL ISSUE THE PROMPT 'DS-B>'. FROM THIS POINT UNTIL THE TIME WHEN YOU RESTART XXDP, YOU WILL BE TALKING TO THE DIAGNOSTIC, NOT XXDP. WE WILL REFER TO THE PRESENCE OF THIS PROMPT AS BEING IN DIAGNOSTIC COMMAND MODE, AS OPPOSED TO XXDP COMMAND MODE.

AT THIS POINT YOU WILL ENTER A 'START' COMMAND. THIS IS NOT THE SAME AS THE XXDP 'START' COMMAND, WHICH YOU ALREADY ISSUED IN RESPONSE TO THE XXDP DOT PROMPT. THIS 'START' COMMAND CAN TAKE A NUMBER OF SWITCHES AND FLAGS (ALL OPTIONAL) AND THE DETAILS OF THESE ARE SET FORTH IN '2.3 DETAILS OF COMMANDS AND SYNTAX'. HOWEVER, IN ORDER TO USE THE PROGRAM, ALL YOU NEED TO SAY IS SOMETHING LIKE THIS:

STA/PASS:1/FLAGS:HOE

THINGS TO NOTE HERE:

1. ONLY THE FIRST THREE CHARACTERS OF THIS OR ANY COMMAND AT THE 'DS-B>' LEVEL NEED TO BE TYPED.
2. THE 'PASS' SWITCH SPECIFIES HOW MANY PASSES YOU DESIRE. A PASS CONSISTS OF RUNNING THE FULL DIAGNOSTIC AGAINST ALL UNITS BEING TESTED (THIS WILL BE EXPLAINED SHORTLY). ONE PASS IS SPECIFIED IN THE ABOVE EXAMPLE.
3. THE 'FLAGS' SWITCH MAY SPECIFY ANY OF A NUMBER OF FLAGS, BUT THE MAIN USEFUL ONES ARE:

|     |                        |
|-----|------------------------|
| LOE | LOOP ONE ERROR         |
| HOE | HALT ON ERROR          |
| IER | INHIBIT ERROR PRINTOUT |

THE HOE FLAG IS SPECIFIED IN THE ABOVE EXAMPLE (WE'LL SEE WHY SHORTLY).

\*\*\*\*\*  
\* STEP 3 \*  
\*\*\*\*\*

WHEN YOU HAVE TYPED IN A 'START' COMMAND, THE DIAGNOSTIC WILL COME BACK WITH THE QUESTION 'N UNITS?' TO WHICH YOU SHOULD RESPOND BY TYPING IN THE NUMBER OF DEVICES YOU WISH TO TEST.

A WORD OF WARNING HERE: THE NUMBER OF UNITS DEPENDS ON THE TARGET DEVICE OF THE DIAGNOSTIC. FOR EXAMPLE, IF THE DIAGNOSTIC IS DIRECTED

AT A DISK DRIVE, THEN THE NUMBER OF UNITS WOULD BE THE NUMBER OF DRIVES TO BE TESTED. WHEREAS IF THE DIAGNOSTIC WAS DIRECTED AT THE DISK CONTROLLER, THEN THE NUMBER OF UNITS WOULD BE THE NUMBER OF CONTROLLERS. THE TARGET DEVICE OF A DIAGNOSTIC CAN ALWAYS BE DETERMINED BY INSPECTING THE 'HEADER' STATEMENT NEAR THE BEGINNING OF THE SOURCE CODE. ONE OF THE OPERANDS OF THIS 'HEADER' STATEMENT SHOULD BE THE DEVICE TYPE OF THE DIAGNOSTIC.

\*\*\*\*\*  
\* STEP 4 \*  
\*\*\*\*\*

WHEN YOU HAVE TYPED IN THE NUMBER OF UNITS TO BE TESTED, THE DIAGNOSTIC WILL ASK YOU THE 'HARDWARE QUESTIONS'. THE ANSWERS TO THESE QUESTIONS ARE USED TO BUILD TABLES IN CORE, CALLED 'HARDWARE P-TABLES'. ONE HARDWARE P-TABLE WILL BE BUILT FOR EACH UNIT TO BE TESTED.

THERE ARE SEVERAL HARDWARE QUESTIONS AND THE ENTIRE SERIES WILL BE POSED N TIMES, WHERE N IS THE NUMBER OF UNITS.

THIS REPRESENTS A NEW PHILOSOPHY IN DIAGNOSTIC ENGINEERING. DIAGNOSTICS IN THE FUTURE WILL NOT BE WRITTEN TO AUTOSIZE OR ASSUME STANDARD ADDRESSES: INSTEAD, THEY WILL ASK THE OPERATOR FOR ALL THE INFORMATION THEY NEED TO TEST THE DEVICE.

\*\*\*\*\*  
\* STEP 5 \*  
\*\*\*\*\*

AFTER YOU HAVE ANSWERED ALL THE HARDWARE QUESTIONS (SEC 2.5) FOR ALL THE UNITS, YOU WILL BE ASKED 'CHANGE SW?' IF YOU WANT TO BE ASKED THE SOFTWARE QUESTIONS THAT DETERMINE THE BEHAVIOR OF THIS PROGRAM, TYPE 'Y'. IF YOU WANT TO TAKE ALL THE DEFAULTS TO THESE QUESTIONS, TYPE 'N'. IF YOU TYPE 'Y' YOU WILL BE ASKED THE SOFTWARE QUESTIONS (SEC 2.6), AND THE ANSWERS WILL BE PUT INTO THE SOFTWARE P-TABLE IN THE PROGRAM. THE SERIES OF QUESTIONS WILL BE ASKED JUST ONCE, REGARDLESS OF THE NUMBER OF UNITS TO BE TESTED.

\*\*\*\*\*  
\* STEP 6 \*  
\*\*\*\*\*

AFTER YOU HAVE ANSWERED THE SOFTWARE QUESTIONS, THE DIAGNOSTIC WILL BEGIN TO EXECUTE THE HARDWARE TEST CODE. THERE ARE SEVERAL THINGS THAT CAN HAPPEN NEXT, DEPENDING ON WHETHER A HARDWARE ERROR IS ENCOUNTERED AND ALSO ON WHAT SWITCH VALUES YOU SELECTED ON THE START COMMAND. CONSIDER THE POSSIBILITIES:

1. IF NO ERROR IS ENCOUNTERED, THEN THE DIAGNOSTIC WILL SIMPLY EXECUTE THE DESIRED NUMBER OF PASSES AND RETURN TO COMMAND MODE (PROMPT DS-B>).

2. IF AN ERROR IS ENCOUNTERED, THEN ONE OF THREE THINGS HAPPENS, DEPENDING ON THE SETTINGS OF THE HOE AND LOE FLAGS.

HOE SET: THE ERROR WILL BE REPORTED ON THE CONSOLE AND THE DIAGNOSTIC WILL RETURN TO COMMAND MODE.

LOE SET: THE DIAGNOSTIC WILL LOOP ENLESSLY ON THE BLOCK OF CODE THAT DETECTED THE ERROR.

NEITHER HOE NOR LOE SET: THE ERROR WILL BE REPORTED ON THE CONSOLE AND NORMAL EXECUTION WILL RESUME AS IF NO ERROR HAD OCCURED.

#### 2.1.2 SAMPLE RUN-THROUGH

LET'S SEE HOW ALL THIS WORKS IN A REAL SITUATION. RECALL THAT WE ENTERED THE COMMAND 'STA/PASS.1/FLAGS:HOE'. THIS WOULD BE A VERY TYPICAL WAY TO RUN THE DIAGNOSTIC. IF NO ERRORS ARE ENCOUNTERED, THE SINGLE REQUESTED PASS WILL BE EXECUTED AND THE PROMPT WILL BE REISSUED.

IF AN ERROR IS ENCOUNTERED, THE ERROR WILL BE REPORTED AND THE PROMPT WILL BE REISSUED (BECAUSE THE HOE FLAG IS SET). AT THIS POINT THERE ARE FOUR DIFFERENT WAYS YOU CAN GET THE PROGRAM GOING AGAIN:

1. ISSUE ANOTHER 'START' COMMAND (THUS GOING THRU ALL OF STEPS 2, 3, 4, 5, AND 6 AGAIN)
2. ISSUE A 'RESTART' COMMAND (SAME AS START COMMAND EXCEPT THAT THE HARDWARE QUESTIONS ARE NOT ASKED)
3. ISSUE A 'CONTINUE' COMMAND (EXECUTION WILL RESUME AT THE BEGINNING OF THE PARTICULAR HARDWARE TEST (MOST DIAGNOSTICS CONSIST OF A NUMBER OF THESE) THAT IT WAS IN WHEN THE ERROR HALT OCCURED. NO QUESTIONS ASKED).
4. ISSUE A 'PROCEED' COMMAND: EXECUTION WILL RESUME AT THE INSTRUCTION FOLLOWING THE ERROR REPORT (THIS IS A SPECIAL COMMAND AND CAN BE ISSUED ONLY AT A HALT ON ERROR).

THE MCS' TYPICAL THING TO DO HERE IS TO ISSUE THE PROCEED, BUT WITH DIFFERENT FLAG SETTINGS. PROBABLY YOU WOULD WANT TO SAY

PRO/FLAGS:IER:LOE:HOE-0

THIS WILL DO THE FOLLOWING:

1. TURN ON THE IER (INHIBIT ERROR PRINTOUT) FLAG
2. TURN ON THE LOE FLAG
3. TURN OFF THE HOE FLAG
4. RESUME EXECUTION AT INSTRUCTION AFTER ERROR REPORT

THE DIAGNOSTIC WILL NOW LOOP ON THE BLOCK OF CODE THAT DETECTED AND REPORTED THE ERROR, BUT NO ERROR PRINTOUT WILL OCCUR. THUS YOU CAN STUDY THE ERROR OR SCOPE IT OR WHATEVER.

WHEN YOU'VE SEEN ENOUGH, YOU MAY HIT CONTROL/C. THIS WILL TAKE YOU OUT OF THE LOOP AND PUT YOU BACK INTO COMMAND MODE. YOU NOW HAVE THREE CHOICES:

1. START
2. RESTART
3. CONTINUE

LET'S SAY YOU'VE REPAIRED THE DEFECT FOUND ABOVE AND WANT TO FINISH RUNNING THE DIAGNOSTIC. YOU WOULD TYPE

CON/FLAGS:HOE:IER=0:LOE=0

THIS WILL RESTORE THE FLAGS TO THEIR ORIGINAL VALUES AND RESUME EXECUTION AT THE BEGINNING OF THE HARDWARE TEST YOU WERE IN. IF THE ERROR DOES NOT RECUR, THE EXECUTION WILL FLOW RIGHT ON THRU TO THE NEXT ERROR OR TO END OF PASS.

IF AT END OF PASS YOU WANT TO RUN THE DIAGNOSTIC AGAIN, YOU HAVE TWO CHOICES:

1. START
2. RESTART

YOU WOULD CHOOSE ONE, DEPENDING ON WHETHER YOU WANTED TO ANSWER THE HARDWARE QUESTIONS AGAIN.

THE FULL PRINT-OUT FROM THE ABOVE DIALOGUE MIGHT LOOK LIKE THIS:

|                                                 | BY<br>WHOM<br>ENTERED: |
|-------------------------------------------------|------------------------|
| .R DZRKXX                                       | O                      |
| DZRKXX                                          | D                      |
| L-CLK (L) N ?                                   | D.O                    |
| 50HZ (L) N ?                                    | D                      |
| LSI (L) N ?                                     | D                      |
| LPT (L) N ?                                     | D                      |
| MEM (K) (D) 16 ?                                | D                      |
| DS-B>STA/PASS:1/FLAGS:HOE                       | D.O                    |
| # UNITS (D) ? 2                                 | D.O                    |
| UNIT 1                                          | D                      |
| CSR (O) ?                                       | D.O                    |
| VECTOR (O) ?                                    | D.O                    |
| BR LEVEL (O) ?                                  | D.O                    |
| DRIVE (O) ? 0                                   | D.O                    |
| UNIT 2                                          | D                      |
| CSR (O) ?                                       | D.O                    |
| VECTOR (O) ?                                    | D.O                    |
| BR LEVEL (O) ?                                  | D.O                    |
| DRIVE (O) ? 1                                   | D.O                    |
| CHANGE SW (L) ? N                               | D.O                    |
| DZRKXX HARD ERR 00004 TST 003 SUB 002 PC:004130 | D                      |
| ERR HLT                                         | D                      |
| DS-B>PRO/FLAGS:IER:LOE:HOE 0                    | D.O                    |

\*\*\*\*\*  
 AT THIS POINT THE DIAGNOSTIC IS LOOPING ON THE  
 ERROR WITHOUT PRINTING ANYTHING. YOU CAN SCOPE  
 THE ERROR UNTIL YOU HAVE LOCATED IT, THEN ^C OUT  
 \*\*\*\*\*

|                              |     |
|------------------------------|-----|
| ^C                           | O   |
| DS-B>CON/FLAGS:HOE:IER:LOE=0 | D.O |
| CHANGE SW (L) ? N            | D.O |
| DZRKXX EOP 1                 | D   |
| DS-B>RESTART/PASS:1          | D.O |
| CHANGE SW (L) ? N            | D.O |

-----  
 -----  
 -----  
 -----  
 -----

## 2.2 HOW TO CREATE A CHAINABLE FILE

THE DIAGNOSTIC AS RECEIVED FROM RELEASE ENGINEERING CANNOT BE RUN IN CHAIN MODE. THAT IS WHY IT BEARS THE EXTENSION 'BIN' INSTEAD OF 'BIC'. THERE IS A WAY, HOWEVER, TO CREATE A CHAINABLE PROGRAM FROM WHAT YOU'VE GOT.

IT CONSISTS OF RUNNING THE PROGRAM WITH THE SPECIAL COMMAND 'CCI' ISSUED WHERE YOU WOULD NORMALLY ISSUE A START COMMAND (TO THE PROMPT DS-B>). THIS COMMAND CAUSES THE DIAGNOSTIC TO GO THRU ALL THE QUESTIONS AND ANSWERS AND THEN TO HALT, JUST WHERE IT WOULD ORDINARILY BEGIN EXECUTION OF THE HARDWARE TEST CODE. AT THIS POINT YOU CAN DUMP THE PROGRAM AS IT SITS IN CORE TO THE LOAD MEDIUM, WITH THE NEW EXTENSION 'BIC'.

HERE IS A SAMPLE DIALOGUE TO ACCOMPLISH THIS:

```
.R UPD2
RESTART: XXXXXX
*CLR
*LOAD DIAG.BIN
XFER:200 CORE:0.60602
*START 200
L-CLK (L) N ?
-----
-----
-----
```

```
DS-B>CCI
# UNITS (D) ? 4
-----
-----
-----
```

```
CHANGE SW (L) ? N
PTAB END: 60632
```

```
*****
*AT THIS POINT THE MACHINE HALTS AND*
*YOU MUST RESTART AT ADDRESS XXXXXX*
*****
```

```
*HICORE 60632
CORE: 0.60632
*DUMP DK0: DIAG.BIC
```

THE RESULT OF DOING THIS IS THAT YOU CAN NOW BUILD AN XXDP CHAIN FILE CONTAINING THE XXDP COMMAND

```
.R DIAG.BIC
```

AND THE DIAGNOSTIC WILL EXECUTE WITHOUT MANUAL INTERVENTION, USING THE ANSWERS THAT YOU GAVE IT WHEN YOU DID THE CCI COMMAND.

## 2.3 DETAILS OF COMMANDS AND SYNTAX

### 2.3.1 TABLE OF COMMAND VALIDITY

THERE ARE FOUR WAYS OF ENTERING DIAGNOSTIC COMMAND MODE, AND DIFFERENT SUBSETS OF THE DIAG COMMAND SET ARE AVAILABLE WITH EACH:

| HOW ENTERED                                           | LEGAL COMMANDS                                                                 |
|-------------------------------------------------------|--------------------------------------------------------------------------------|
| 1. OPERATOR ENTERED 'RUN DIAG'                        | START<br>PRINT<br>DISPLAY<br>FLAGS<br>ZFLAGS                                   |
| 2. DIAGNOSTIC HAS FINISHED ALL ITS REQUESTED PASSED   | START<br>RESTART<br>PRINT<br>DISPLAY<br>FLAGS<br>ZFLAGS                        |
| 3. OPERATOR INTERRUPTED THE DIAGNOSTIC WITH CTRL/C    | START<br>RESTART<br>CONTINUE<br>PRINT<br>DISPLAY<br>FLAGS<br>ZFLAGS            |
| 4. AN ERROR WAS ENCOUNTERED WITH THE HOE FLAG SET SET | START<br>RESTART<br>CONTINUE<br>PROCEED<br>PRINT<br>DISPLAY<br>FLAGS<br>ZFLAGS |

### 2.3.2 COMMAND SYNTAX

```
*****
S*(RT)/TESTS:TEST-LIST/PASS:PASS-CNT/FLAGS:FLAG-LIST/EOP:EOP-1NCR
*****
```

THE DIAGNOSTIC IN CORE IS EXECUTED IN ACCORDANCE WITH THE SWITCHES SPECIFIED. THE MESSAGE '# UNITS?' IS PRINTED. THE START COMMAND MAY BE ISSUED WHEN DIAGNOSTIC COMMAND MODE HAS BEEN ENTERED VIA ONE OF THE FOLLOWING: A) OPERATOR TYPED 'RUN DIAGNOSTIC' B) DIAGNOSTIC FINISHED EXECUTING C) ERROR WAS ENCOUNTERED WITH HOE FLAG SET D) OPERATOR ENTERED CONTROL/C.

AFTER THE OPERA OR RESPONDS TO '"# UNI'S?'"', THE HARDWARE DIALOGUE IS INITIATED. WHEN IT IS COMPLETED, THE QUESTIONS '"CHANGE SW?'" IS ISSUED, AND THE ANSWERS, IF GIVEN, BECOME THE NEW DEFAULTS. THEREFORE IT IS NECESSARY TO RELOAD THE PROGRAM IN ORDER TO RETURN TO THE LOAD DEFAULTS.

THE SWITCH ARGUMENTS ARE AS FOLLOWS:

'TEST-LIST' IS A SEQUENCE OF DECIMAL NUMBERS (1:2 ETC.) OR RANGES OF DECIMAL NUMBERS (1-5:8-10 ETC.) THAT SPECIFY THE TESTS TO BE EXECUTED. THE NUMBERS ARE SEPARATED BY COLONS. THE NUMBERS RANGE FROM 1 TO THE LARGEST TEST NUMBER IN THE DIAGNOSTIC. THEY MAY BE SPECIFIED IN ANY ORDER. TESTS WILL BE EXECUTED IN NUMERICAL ORDER REGARDLESS OF THE ORDER OF SPECIFICATION. THE DEFAULT IS TO EXECUTE ALL TESTS.

'PASS-CNT' IS A DECIMAL NUMBER INDICATING THE DESIRED NUMBER OF PASSES. A PASS IS DEFINED AS THE EXECUTION OF THE FULL DIAGNOSTIC (ALL SELECTED TESTS) AGAINST ALL UNITS SUBMITTED. THE DEFAULT IS NON-ENDING EXECUTION..B 'FLAG-LIST' IS A SEQUENCE OF ELEMENTS OF THE FORM <FLAG>, <FLAG=1>, OR <FLAG=0>, SEPARATED BY COLONS, WHERE <FLAG> HAS ONE OF THE FOLLOWING VALUES:

|     |                                                                                                                                                         |
|-----|---------------------------------------------------------------------------------------------------------------------------------------------------------|
| HOE | HALT ON ERROR, CAUSING COMMAND MODE TO BE ENTERED WHEN AN ERROR IS ENCOUNTERED                                                                          |
| LOE | LOOP ON ERROR, CAUSING THE DIAGNOSTIC TO LOOP CONTINUOUSLY WITHIN THE SMALLEST DEFINED BLOCK OF CODING (SEGMENT, SUBTEST, OR TEST) CONTAINING THE ERROR |
| IER | INHIBIT ERROR REPORTING                                                                                                                                 |
| IBE | INHIBIT BASIC ERROR REPORTS                                                                                                                             |
| IXE | INHIBIT EXTENDED ERROR REPORTS                                                                                                                          |
| PRI | DIRECT ALL MESSAGES TO A LINE PRINTER                                                                                                                   |
| PNT | PRINT NUMBER OF TES BEING EXECUTED                                                                                                                      |
| BOE | BELL ON ERROR                                                                                                                                           |
| UAM | RUN IN UNATTENDED MODE, BYPASSING MANUAL INTERVENTION TESTS                                                                                             |
| ISR | INHIBIT STATISTICAL REPORTS                                                                                                                             |
| IDU | INHIBIT DROPPING OF UNITS BY DIAGNOSTIC                                                                                                                 |

THE FLAGS NAMED OR EQUATED TO 1 ARE SET, THOSE EQUATED TO 0 ARE CLEARED. A FLAG NOT SPECIFIED IS CLEARED. IF THE FLAGS SWITCH IS NOT GIVEN ALL FLAGS ARE CLEARED.

'EOP-INCR' IS A DECIMAL NUMBER INDICATING HOW OFTEN (IN TERMS OF PASSES) IT IS DESIRED THAT THE END OF PASS MESSAGE BE PRINTED. THE DEFAULT IS AT THE END OF EVERY PASS.

\*\*\*\*\*  
RES(TART)/TEST:TEST-LIST/PASS:PASS-CNT/FLAGS:FLAG-LIST/EOP:EOP-INCR/UNITS:UNIT-LIST  
\*\*\*\*\*

THE DIAGNOSTIC IN CORE IS EXECUTED IN ACCORDANCE WITH THE SWITCHES SPECIFIED. HOWEVER, NEW P-TABLES ARE NOT BUILT. INSTEAD, THE ONES IN CORE ARE USED.

THE QUESTION 'CHANGE SW?' IS ASKED, AND THE ANSWERS IF GIVEN BECOME THE NEW DEFAULTS. THE COMMAND MAY BE ISSUED WHEN COMMAND MODE HAS BEEN ENTERED VIA A) DIAGNOSTIC IS FINISHED B) HALT ON ERROR C) CONTROL/C.

THE SWITCH ARGUMENTS ARE AS IN THE START COMMAND EXCEPT:

1. 'UNIT-LIST' IS A SEQUENCE OF LOGICAL UNIT NUMBERS RANGING FROM 1 THRU N (N - NUMBER OF UNITS BEING TESTED) SPECIFYING WHICH UNITS ARE TO BE TESTED. THE LOGICAL UNIT NUMBER DESIGNATES THE POSITION OF THE P-TABLE IN CORE, ACCORDING TO THE ORDER IN WHICH THEY WERE BUILT. THE UNITS SPECIFIED MUST NOT HAVE BEEN DROPPED BY THE OPERATOR DROP COMMAND. THE UNIT-LIST DEFAULTS TO 'ALL THAT HAVE NOT BEEN DROPPED BY OPERATOR COMMAND'. THE EFFECT OF THE UNIT-LIST LASTS UNTIL THE NEXT START (WHERE IT IS AUTOMATICALLY RESET TO 'ALL') OR THE NEXT RESTART.

2. ALL UNSPECIFIED FLAG SETTINGS ARE UNCHANGED.

\*\*\*\*\*  
CON(TINUE)/PASS:<PASS-CNT/FLAGS:<FLAG-LIST>  
\*\*\*\*\*

COMMAND MODE MUST HAVE BEEN ENTERED DUE TO A HALT ON ERROR OR A CONTROL/C. THE EFFECT OF THE COMMAND IS TO GO TO THE BEGINNING OF THE TEST THAT WAS BEING EXECUTED WHEN THE HALT OR CONTROL/C TOOK PLACE. SOFTWARE DIALOGUE MAY OPTIONALLY BE REEXECUTED. HARDWARE PARAMETERS MAY NOT BE CHANGED.

THE SWITCH ARGUMENTS ARE AS IN THE START COMMAND EXCEPT:

1. DEFAULT FOR PASS-CNT IS THE UNSATISFIED PASS-CNT FROM THE PREVIOUS START OR RESTART
2. UNSPECIFIED FLAG SETTINGS ARE UNCHANGED

\*\*\*\*\*  
PROCEED)/FLAGS:<FLAG-LIST>  
\*\*\*\*\*

COMMAND MODE MUST HAVE BEEN ENTERED VIA A HALT ON ERROR. THE EFFECT OF THE COMMAND IS TO BEGIN EXECUTION AT THE LOCATION FOLLOWING THE ERROR CALL. NEITHER HARDWARE NOR SOFTWARE PARAMETERS MAY BE ALTERED.

THE SWITCH ARGUMENTS ARE THE SAME AS THE START COMMAND EXCEPT:

1. UNSPECIFIED FLAG SETTINGS ARE UNCHANGED

\*\*\*\*\*  
CCI/TEST:TEST-LIST/PASS:PASS-CNT/FLAGS:FLAG-LIST/EOP:EOP-INCR  
\*\*\*\*\*

THE DIAGNOSTIC EXECUTES THRU ALL OPERATOR DIALOGUE AND HALTS AT THE HARDWARE TEST CODE. NOW THE OPERATOR CAN DUMP THE CORE IMAGE TO THE MEDIUM WITH A BIC EXTENSION.

THE BIC FILE MUST BE HANDLED DIFFERENTLY DEPENDING ON WHETHER IT IS RUN MANUALLY OR IN CHAIN MODE. IF RUN MANUALLY IT CAN BE INVOKED EITHER WITH A 'START' (IN WHICH CASE IT WILL BEHAVE LIKE THE BIN FILE: THE PRE-GENERATED ANSWERS TO OPERATOR QUESTIONS WILL BE IGNORED) OR WITH A 'RESTART' (IN WHICH CASE THE PRE-GENERATED OPERATOR ANSWERS WILL BE USED).

IF RUN IN CHAIN MODE, AUTOMATIC EXECUTION WILL COMMENCE IMMEDIATELY FROM THE XXDP COMMAND '.R DIAG'. THE COMMAND PROMPT 'DS-B>' WILL NOT BE ISSUED.

ANY SWITCHES SPECIFIED ON THE CCI COMMAND WILL CARRY OVER WHEN THE BIC FILE IS RUN IN CHAIN MODE (EXCEPT THAT UAM IS ALWAYS SET THERE) BUT WILL NOT CARRY OVER WHEN IT IS RUN MANUALLY.

TO DO A CCI ON A FULL SIZED DIAGNOSTIC (14.5K WORDS), A MACHINE SIZE LARGER THAN 16K IS REQUIRED. THE EXACT SIZE NEEDED DEPENDS ON WHICH UTILITY IS USED TO EXECUTE THE DIAGNOSTIC AT CCI TIME.

\*\*\*\*\*  
DRO(P)/UNITS:UNIT-LIST  
\*\*\*\*\*

THE UNITS SPECIFIED ARE DROPPED FROM TESTING UNTIL THEY ARE ADDED BACK OR UNTIL A START COMMAND IS GIVEN. A DROP CANNOT BE FOLLOWED BY A PROCEED.

THERE IS ALSO A 'DROP' MACRO INTERNAL TO THE DIAGNOSTIC, WHICH GIVES THE FACILITY OF AUTO-DROPPING. THE DURATION OF A PROGRAM DROP, HOWEVER, IS ONLY UNTIL THE NEXT START OR RESTART.

\*\*\*\*\*  
ADD/UNITS:UNIT-LIST  
\*\*\*\*\*

THE UNITS SPECIFIED ARE ADDED BACK (THEY MUST HAVE BEEN PREVIOUSLY DROPPED BY THE DROP COMMAND) TO THE TEST SEQUENCE. AN ADD CANNOT BE FOLLOWED BY A PROCEED.

\*\*\*\*\*  
PRI(NT)  
\*\*\*\*\*

ALL STATISTICS TABLES ACCUMULATED BY THE DIAGNOSTIC ARE PRINTED. THE ISR (INHIBIT STATISTICAL REPORTING) FLAG IS CLEARED.

\*\*\*\*\*  
DIS(PLAY)/UNITS:<UNIT-LIST>  
\*\*\*\*\*

THE HARDWARE P-TABLES FOR ALL UNITS UNDER TEST ARE PRINTED OUT IN THE FORMAT IN WHICH THEY WERE ENTERED. ANY UNITS THAT WERE DROPPED BY THE OPERATOR 'DROP' COMMAND ARE SO DESIGNATED.

\*\*\*\*\*  
FLA(GS)  
\*\*\*\*\*

THE CURRENT SETTINGS OF ALL FLAGS ARE PRINTED.

\*\*\*\*\*  
ZFL(AGS)  
\*\*\*\*\*

ALL FLAGS ARE CLEARED.

## 2.4 EXTENDED P-TABLE DIALOGUE

THE FULL CAPABILITY OF THE HARDWARE DIALOGUE IS REVEALED BY THE FOLLOWING DISCUSSION OF WHAT HAPPENS INTERNALLY.

AS SOON AS THE QUESTION '# UNITS?' IS ANSWERED (WITH THE NUMBER N, SAY) SPACE IN CORE IS ALLOCATED FOR N P-TABLES. ALL OF THE P-TABLES ARE OF THE SAME FORMAT, AND THERE IS A ONE-TO-ONE CORRESPONDENCE BETWEEN THE HARDWARE PARAMETER QUESTIONS AND THE SLOTS IN THE P-TABLE FORMAT.

ON THE FIRST TRIP THRU THE QUESTIONS, ALL OF THE SLOTS IN ALL OF THE P-TABLES ARE FILLED. IF THE OPERATOR TYPES IN LESS THAN N EXPLICIT VALUES IN RESPONSE TO A PARTICULAR QUESTION, THESE VALUES ARE PLACED IN THE P-TABLES (ONE VALUE GOING INTO THE PROPER SLOT OF EACH P-TABLE BEGINNING WITH THE FIRST P-TABLE) UNTIL THE STRING OF VALUES IS EXHAUSTED. THE LAST VALUE THAT SLOT IN THE REMAINING P-TABLES.

ON SUBSEQUENT TRIPS THRU THE QUESTIONS, THE SAME PROCESS IS CARRIED OUT, EXCEPT THAT THE EARLIEST P-TABLE NOT TO HAVE RECEIVED AN EXPLICIT VALUE IN ANY OF ITS SLOTS NOW ASSUMES THE ROLE THAT TABLE NUMBER ONE PLAYED IN THE FIRST TRIP.

THE SERIES OF QUESTIONS IS REISSUED UNTIL AT LEAST ONE QUESTION HAS RECEIVED N EXPLICIT VALUES FROM THE OPERATOR.

IN GIVING A STRING OF VALUES, COMMAS WITHOUT INTERVENING VALUES MAY BE USED TO INDICATE A REPETITION OF THE LAST NAMED VALUE.

A STRING OF VALUES MAY BE GIVEN AS A RANGE (6-10 FOR EXAMPLE). IF THE VALUES REPRESENT PURE NUMERICAL DATA, THIS SAMPLE RANGE TRANSLATES TO THE STRING 6,7,8,9,10 (AN INCREMENT OF 1). IF THE VALUES ARE ADDRESSES, THE SAMPLE RANGE TRANSLATES TO THE STRING 6,8,10 (AN INCREMENT OF 2).

NOW LET US SEE HOW WE COULD USE THESE CAPABILITIES TO CONSTRUCT A SET OF P-TABLES. ASSUME THAT WE HAVE 64 UNITS, AND THAT THERE ARE THREE HARDWARE PARAMETERS FOR EACH (THREE SLOTS IN THE P-TABLE, THREE HARDWARE QUESTIONS IN THE DIALOGUE). LET THE DESIRED VALUE FOR THE FIRST PARAMETER BE THE NUMBER 75 FOR ALL 64 TABLES. LET THE DESIRED VALUE FOR THE SECOND PARAMETER BE EQUAL TO THE UNIT NUMBER (1,2,3,...,64) EXCEPT FOR UNIT 50, WHICH SHOULD RECEIVE THE VALUE 49. LET THE DESIRED VALUE FOR THE THIRD PARAMETER BE THE NUMBER 76 FOR THE FIRST 20 UNITS AND THE NUMBER 77 FOR THE LAST 44 UNITS.

THE FOLLOWING DIALOGUE WOULD ACCOMPLISH THIS GOAL:

# UNITS (D) ? 64

UNIT 1

<QUESTION 1> ? 75

<QUESTION 2> ? 1-20

<QUESTION 3> ? 76

UNIT 21

<QUESTION 1> ?

<QUESTION 2> ? 21-49,,51-64

<QUESTION 3> ? 77

THE FIRST TIME THE SERIES IS ASKED, SLOT ONE RECEIVES A 75 IN ALL 64 TABLES. SLOT TWO RECEIVES THE VALUES 1,2,3,...,20 IN TABLES 1 THRU 20 AND A CONSTANT 20 IN TABLES 21 THRU 64. SLOT THREE RECEIVES A CONSTANT 76 IN ALL 64 TABLES.

THE SECOND TIME THRU THE SERIES, TABLES 21 THRU THE END ARE GOING TO BE AFFECTED (NOTE THAT THIS PIECE OF INFORMATION IS PRINTED OUT FOR THE OPERATOR IN THE FORM 'UNIT XX' AT THE BEGINNING OF EACH SERIES). QUESTION 1 IS RESPONDED TO BY A <CR>, SO SLOT ONE STAYS A CONSTANT 75 IN TABLES 21 THRU 64, SINCE NO NEW EXPLICIT VALUES ARE TYPED IN. SLOT TWO GETS THE VALUES 21,22,23,...,49 IN TABLES 21 THRU 49, AND GETS A 49 IN SLOT 50, AND GETS THE VALUES 51,52,53,...,64 IN TABLES 51 THRU 64. SLOT THREE GETS THE VALUE 77 IN TABLES 21 THRU 64.

THE DIALOGUE IS TERMINATED WHEN THE SOFTWARE RECOGNIZES THAT 64 EXPLICIT VALUES HAVE BEEN GIVEN FOR AT LEAST ONE QUESTION (NAMELY QUESTION 2).

## 2.5 HARDWARE PARAMETERS

THE FOLLOWING QUESTIONS WILL BE ASKED ON A START COMMAND. THE VALUE LOCATED TO THE LEFT OF THE QUESTION MARK IS THE DEFAULT VALUE THAT WILL BE TAKEN ON A CARRIAGE RETURN RESPONSE.

RL11 (L) Y?

ANSWER YES(Y) IF YOU HAVE AN RL11 CONTROLLER, NO(N) IF YOU HAVE AN RLV'1 CONTROLLER.

BUS ADDRESS (O) 174400?

ANSWER WITH THE BUS ADDRESS OF THE CONTROLLER.

VECTOR (O) 330?

ANSWER WITH THE INTERRUPT VECTOR OF THE CONTROLLER.

BR LEVEL (O) 5?

ANSWER WITH THE INTERRUPT PRIORITY OF THE CONTROLLER.

DRIVE (O) 0?

ANSWER WITH THE DRIVE(S) CONNECTED TO THE CONTROLLER.

## 2.6 SOFTWARE PARAMETERS

THE FOLLOWING QUESTIONS ARE ASKED IF REQUESTED ON A START, RESTART, OR CONTINUE. THEY ALLOW FLEXIBILITY IN THE WAY THE PROGRAM BEHAVES. THE SOFTWARE PARAMETERS GIVE THE PROGRAM FLEXIBILITY IN THE WAY IT RUNS. THE PARAMETERS CAN BE MODIFIED ON A START, RESTART, OR CONTINUE BY ANSWERING (Y)ES TO THE FOLLOWING QUESTION:

CHANGE S.W. ?

A YES ANSWER WILL ASK THE FOLLOWING SOFTWARE PARAMETER QUESTIONS, WITH THE PRESENT DEFAULT VALUE PRINTED TO THE LEFT OF THE QUESTION MARK. (THE LAST ANSWER GIVEN IS THE DEFAULT) THE DEFAULT IS TAKEN ON A <CR>. CONTROL Z (^Z) WILL DEFAULT ALL REMAINING QUESTIONS AND START THE TEST.

DROP ON ERROR LIMIT (L) Y?

TO ALLOW THE UNIT TO BE DROPPED ONCE A PREDETERMINED NUMBER OF ERRORS ARE ENCOUNTERED.

ANSWER Y OR N

ERROR LIMIT (D) 10?

NUMBER OF ERRORS ALLOWED BEFORE DROPPING UNIT.

ANSWER 1 TO 65K

AUTOSIZE (L) N?

TO CHECK TO SEE IF UNIT SPECIFIED ACTUALLY EXISTS BEFORE TESTING IT (VIA DRIVE READY), IF NOT UNIT WILL NOT BE TESTED.

ANSWER Y OR N

COMPARE DATA ON DCK (L) N?

WHEN A DATA CHECK IS ENCOUNTERED AND DATA IS KNOWN, ALLOW AN INCORE COMPARISON OF DATA.

ANSWER Y OR N

# OF WORDS IN ERROR REPORTED (D) 3?

NUMBER OF MISCOMPARES TO BE PRINTED ON CONSOLE DEVICE.

ANSWER 0 - 128

### 3.0 ERROR INFORMATION

ALL ERROR INFORMATION IS PRINTED ON THE CONSOLE DEVICE. ERROR REPORTS ARE AIMED AT BEING SELF EXPLANATORY. THE GENERAL FORMAT IS:

DZRL? XXX ERR YYYYY TST ZZZ SUB PPP PC: RRRRRR

WHERE:

? IS PROGRAM LETTER  
XXX IS SFT - SOFT ERROR  
HRD - HARD ERROR

DV FAT - DEVICE FATAL ERROR  
SYS FAT - SYSTEM FATAL ERROR  
YYYYY IS THE ERROR NUMBER  
ZZZ IS THE TEST NUMBER  
PPP IS THE SUBTEST NUMBER  
RRRRR IS THE PROGRAM LISTING LOCATION

ERRORS GIVE THE REGISTER CONTENTS BEFORE AND AFTER THE ERROR  
ALONG WITH A ONE LINE DESCRIPTION AND RELEVANT DATA.

EXAMPLE:

ONE LINE DESCRIPTION  
(OPTIONAL SECOND LINE)  
(OPTIONAL THIRD LINE)  
BEFORE COMMAND: CS:XXXXXX BA:XXXXXX DA:XXXXXX MP:XXXXXX  
TIME OF ERROR: CS:XXXXXX BA:XXXXXX DA:XXXXXX MP:XXXXXX XXXXXX XXXXXX

### 3.2 ERROR HALTS

ERROR HALTS ARE SUPPORTED PER DESCRIBED IN THE PREVIOUS SECTION WITH  
/FLAG:HOE. THERE ARE NO OTHER HALTS.

### 4.0 PERFORMANCE AND PROGRESS REPORTS

#### 4.1 PERFORMANCE REPORTS

THIS PROGRAM WILL NOT GIVE ANY PERFORMANCE REPORTS.

#### 4.2 PROGRESS REPORTS

THIS PROGRAM WILL NOT GIVE ANY PROGRESS REPORTS.

### 5.0 DEVICE INFORMATION TABLES

THE RL11/RLV11 CONTROLLER HAS THE FOLLOWING FOUR(4) REGISTERS FOR  
CONTROL OF THE SUBSYSTEM.

RLCS - CONTROL AND STATUS REGISTER (XXXXX0)  
-----

BIT 15 - COMPOSITE ERROR  
BIT 14 - RIVE ERROR  
BIT 13 - NON EXISTANT MEMORY ERROR  
BIT 12 - HEADER NOT FOUND (WITH BIT 10 SET)  
- DATA LATE (WITH BIT 10 CLEAR)  
BIT 11 - HEADER CRC (WITH BIT 10 SET)  
- DATA CRC (WITH BIT 10 CLEAR)  
BIT 10 - OPERATION INCOMPLETE  
BIT 9/8 - DRIVE SELECT (0-3)

BIT 7 - CONTROLLER READY  
BIT 6 - INTERRUPT ENABLE  
BIT 5 - EXTENDED BUS ADDRESS (BIT 17)  
BIT 4 - EXTENDED BUS ADDRESS (BIT 16)  
BIT 3-1 - FUNCTION CODE  
    0 - NOP (PDP-11) MAINT (LSI-11)  
    1 - WRITE CHECK  
    2 - GET DRIVE STATUS  
    3 - SEEK  
    4 - READ HEADER  
    5 - WRITE DATA  
    6 - READ DATA  
    7 - READ WITHOUT HEADER COMPARE

BIT 0 - DRIVE READY

RLBA - BUS ADDRESS REGISTER (XXXXX2)  
-----

BITS 15-1 BUS ADDRESS OF DATA TRANSFER  
BIT 0 SHOULD BE 0

RLDA - DISK ADDRESS REGISTER (XXXXX4)  
-----

FOR READ/WRITE FUNCTIONS  
-----

BIT 15 - MUST BE ZERO(0)  
BIT 14-7 - CYLINDER ADDRESS FOR TRANSFER  
BIT 6 - SURFACE FOR TRANSFER  
BIT 5-0 - SECTOR FOR TRANSFER (0-47)

FOR SEEK FUNCTION  
-----

BIT 15 - MUST BE ZERO(0)  
BIT 14-7 - DIFFERENCE TO NEW CYLINDER  
BIT 6-5 - MUST BE ZERO(0)  
BIT 4 - SURFACE  
BIT 3 - MUST BE ZERO  
BIT 2 - SEEK DIRECTION( 1 - IN / 0 - OUT )  
BIT 1 - MUST BE ZERO  
BIT 0 - MUST BE ONE(1)

FOR GET STATUS FUNCTION  
-----

BIT 15-4 - IGNORED SHOULD BE ZERO  
BIT 3 - DRIVE RESET  
BIT 2 - MUST BE ZERO  
BIT 1 - MUST BE ONE  
BIT 0 - MUST BE ONE

RLMP - MULTIPURPOSE REGISTER

-----  
FOR READ/WRITE FUNCTION  
-----

BIT 15 - 0 - WORD COUNT(TWO'S COMPLIMENT)

-----  
FOR READ HEADER FUNCTION  
-----

BIT 15-0 - DISK HEADER OF SECTOR (FIRST READ)  
          - ZERO WORD (SECOND READ)  
          - HEADER CRC (THIRD READ)

-----  
FOR GET STATUS FUNCTION  
-----

HAS DRIVE STATUS

BIT 15 - WRITE DATA ERROR  
BIT 14 - CURRENT HEAD ERROR(CHE)  
BIT 13 - WRITE LOCK STATUS(WL)  
BIT 12 - SEEK TIME OUT(SKTO)  
BIT 11 - SPIN ERROR(SPE)  
BIT 10 - WRITE GATE ERROR(WGE)  
BIT 9 - VOLUME CHECK(VC)  
BIT 8 - DRIVE SELECT ERROR(DSE)  
BIT 7 - RESERVED(0)  
BIT 6 - SURFACE  
BIT 5 - COVER OPEN  
BIT 4 - HEADS HOME  
BIT 3 - BRUSHES HOME  
BIT 2-0 - STATE BITS  
          0 - LOAD STATE  
          1 - SPIN UP  
          2 - BRUSH CYCLE  
          3 - LOAD HEADS  
          4 - SEEK - TRACK COUNTING  
          5 - SEEK - LINEAR MODE  
          6 - UNLOAD HEADS  
          7 - SPIN DOWN

6.0 TEST SUMMARIES

TEST 01 - WRITE NPR INTEGRITY

THIS TEST WILL VERIFY THAT THE WRITE FUNCTION WILL NOT CAUSE  
A BUS TRAP THEREFORE VERIFYING THE NPR LOGIC BETWEEN THE  
CONTROLLER AND PROCESSOR.

TEST 02 - WRITE FUNCTION

THIS TEST WILL VERIFY THAT THE WRITE FUNCTION WILL RESET CONTROLLER READY AND POST NO ERRORS.

TEST 03 - WRITE FUNCTION INTERRUPT

THIS TEST WILL VERIFY THAT THE WRITE FUNCTION WILL GENERATE AN INTERRUPT ON COMPLETION.

TEST 04 - PROPER INCREMENT OF RLBA ON WRITE

THIS TEST WILL VERIFY THAT THE BUS ADDRESS REGISTER INCREMENTS PROPERLY ON A WRITE FUNCTION.

TEST 05 - PROPER INCREMENT OF RLDA ON WRITE

THIS TEST WILL VERIFY THAT THE DISK ADDRESS REGISTER INCREMENTS PROPERLY ON A WRITE FUNCTION.

TEST 06 - FORCE HEADER NOT FOUND WITH WRITE

THIS TEST WILL FORCE A HEADER NOT FOUND ERROR ON A WRITE. THE RLDA IS SET UP TO LOOK FOR SECTOR 40. A WRITE IS THEN ISSUED. THE HEADER NOT FOUND ERROR SHOULD THEN SET.

TEST 07 - FORCE INTERRUPT WITH HNF

THIS TEST WILL FORCE A HEADER NOT FOUND ERROR UNDER INTERRUPT CONTROL.

TEST 08 - CHECK OPI TIME WITH HNF

THIS TEST WILL TIME THE SETTING OF HNF (OPI) FROM ISSUANCE. THIS IS DONE BY ISSUING A WRITE TO SECTOR 40. THE TIME OF OPI SHOULD BE AROUND 200 MILLISECONDS.

TEST 09 - MULTIPLE SECTOR TRANSFER ON WRITE

THIS TEST THE ABILITY FOR THE WRITE FUNCTION TO WRITE MORE THAN ONE SECTOR. WE SET UP FOR A TWO SECTOR WRITE.

TEST 10 - CHECK DIRECTION OF WRITE NPR

THIS TEST WILL VERIFY THAT THE NPR DIRECTION OF A WRITE FUNCTION IS FROM MEMORY TO THE CONTROLLER. THIS IS DONE BY WRITING A PATTERN IN MEMORY AND ISSUING A WRITE, THEN CHECKING MEMORY TO VERIFY THAT IT DID NOT GET DISTURBED.

TEST 11 - CHECK FULL INCREMENT OF RLBA

THIS TEST WILL CHECK THAT THE RLBA CAN INCREMENT OF THE FULL 16 BIT RANGE. THIS IS DONE BY ISSUING A ONE WORD WRITE TO CHECK EACH BIT TOGGLE FROM 1-0 AND 0-1. THIS IS DONE FROM 0 TO 177776 REGARDLESS OF MEMORY SIZE.

TEST 12 - BA BIT 16 INCREMENT

THIS TEST WILL CHECK THAT BUS ADDRESS BIT 16 WILL SET WHEN THE RLBA IS 177776. AND THAT THE RLBA GOES TO 0.

TEST 13 - BA BIT 17 INCREMENT

THIS TEST WILL CHECK THAT BUS ADDRESS BIT 17 WILL SET WHEN BIT 16 AND THE RLBA ARE SET. THE RLBA AND BIT 16 ARE CHECKED TO GO TO ZERO.

TEST 14 - READ NPR INTEGRITY

THIS TEST WILL VERIFY THAT THE READ FUNCTION WILL NOT CAUSE A BUS TRAP THEREFORE VERIFYING \* - NPR LOGIC BETWEEN THE CONTROLLER AND PROCESSOR.

TEST 15 - READ FUNCTION

THIS TEST WILL VERIFY THAT THE READ FUNCTION WILL RESET CONTROLLER READY AND POST NO ERRORS.

TEST 16 - READ FUNCTION INTERRUPT

THIS TEST WILL VERIFY THAT THE READ FUNCTION WILL GENERATE AN INTERRUPT ON COMPLETION.

TEST 17 - CHECK DIRECTION OF READ NPR

THIS TEST WILL VERIFY THAT THE NPR DIRECTION OF A READ FUNCTION IS FROM CONTROLLER TO THE MEMORY. THIS IS DONE BY WRITING A PATTERN IN MEMORY AND ISSUING A READ, THEN CHECKING MEMORY TO VERIFY THAT IT DID NOT GET DISTURBED.

TEST 18 - PROPER INCREMENT OF RLBA ON READ

THIS TEST WILL VERIFY THAT THE BUS ADDRESS REGISTER INCREMENTS PROPERLY ON A READ FUNCTION.

TEST 19 - PROPER INCREMENT OF RLDA ON READ

THIS TEST WILL VERIFY THAT THE DISK ADDRESS REGISTER INCREMENTS PROPERLY ON A READ FUNCTION.

TEST 20 - FORCE HEADER NOT FOUND WITH READ

THIS TEST WILL FORCE A HEADER NOT FOUND ERROR ON A READ. THE RLDA IS SET UP TO LOOK FOR SECTOR 40, A READ IS THEN ISSUED. THE HEADER NOT FOUND ERROR SHOULD THEN SET.

TEST 21 - FORCE INTERRUPT WITH HNF

THIS TEST WILL FORCE A HEADER NOT FOUND ERROR UNDER INTERRUPT CONTROL.

TEST 22 - CHECK HEADER COMPARE LOGIC

THIS TEST WILL EXTENSIVELY CHECK THE CYLINDER AND HEAD BITS OF THE HEADER WORD TO COMPARE CORRECTLY. THIS IS DONE BY WALKING AND GROWING 0'S AND 1'S THRU THE PROPER RLDA BITS AND ISSUING READ TO SEE IF ALL BIT POSITIONS CAN COMPARE.

TEST 23 - MULTIPLE SECTOR TRANSFER ON READ

THIS TEST THE ABILITY FOR THE READ FUNCTION TO WRITE MORE THAN ONE SECTOR. WE SET UP FOR A TWO SECTOR READ.

TEST 24 - FORCE HNF AT END OF TRACK

THIS TEST WILL CHECK THE ABILITY TO DETECT HEADER NOT FOUND AT THE END OF A TRACK. THIS DONE BY SETTING UP FOR A TWO SECTOR READ AT SECTOR 39.

TEST 25 - FORCE NON-EXISTANT MEMORY ERROR

THIS TEST WILL CHECK THAT THE NON-EXISTANT MEMORY ERROR (NXM) CAN SET. WE WILL ISSUE A READ TO THE MAXIMUM ADDRESS AND EXPECT A NXM ERROR. (THIS TEST WILL NOT BE DONE ON A 128K MACHINE.)

TEST 26 - FORCE NXM UNDER INTERRUPT

THIS TEST WILL ATTEMPT TO FORCE AN INTERRUPT VIA NXM. (THIS TEST WILL NOT BE DONE ON A 128K MACHINE.)

TEST 27 - CHECK READ WRITE LOOP

THIS TEST WILL WRITE A PATTERN TO SECTOR 0 AND TRY TO RECOVER IT WITH A WRITE.

TEST 28 - CHECK OF SILO LINES

THIS TEST WILL CHECK THAT WE CAN WRITE AND READ UNIQUE BIT PATTERNS VERIFY THAT THE LINES ON THE SILO ARE NOT STUCK OR TIED TOGETHER. THIS IS DONE WITH WALKING AND GROWING 0'S AND 1'S.

TEST 29 - CHECK THROUGHPUT OF SILO

THIS TEST WILL ATTEMPT TO CHECK THAT THE FALL THROUGH OF THE SILO IS WORKING CORRECTLY. WE WRITE A SECTOR OF 128 UNIQUE PATTERNS AND READ IT BACK CHECKING THAT EACH LOCATION IS UNIQUE AND CORRECT.

TEST 30 - CHECK ZERO FILL ON WRITE

THIS TEST WILL CHECK THE ABILITY OF THE CONTROLLER TO FILL THE REMAINING SECTOR WITH ZEROS ON A WRITE. WE WRITE A SECTOR WITH FROM 1 TO 127 WORDS, READ IT BACK AND VERIFY THAT THE NON WRITTEN WORDS ARE ZERO.

TEST 31 - CHECK SECTOR BITS ON HEADER COMPARE

THIS TEST WILL CHECK THAT THE SECTOR BITS CAN COMPARE CORRECTLY. THIS IS DONE BY WRITING THE SECTORS ADDRESS INTO THE SECTOR FOR A FULL TRACK. EACH SECTOR IS READ TO VERIFY THE SECTOR HAS THE CORRECT DATA, IF NOT THEN THE SECTOR BITS ARE NOT COMPARING CORRECTLY.

TEST 32 - WRITE CHECK NPR INTEGRITY

THIS TEST WILL CHECK THAT THE WRITE CHECK WILL FUNCTION WITHOUT CAUSING A BUS TRAP. TEST IS SET UP TO HANDLE BUS TRAPS.

TEST 33 - WRITE CHECK FUNCTION

THIS TEST WILL CHECK THAT A WRITE CHECK FUNCTION WILL COMPLETE WITH THE SPECIFIED TIME WITHOUT POSTING ERRORS.

TEST 34 - WRITE CHECK FUNCTION INTERRUPT

THIS TEST WILL CHECK THAT AN INTERRUPT CAN BE GENERATED FROM ISSUING A WRITE CHECK.

TEST 35 - PROPER INCREMENT OF RLBA ON WRITE CHECK

THIS TEST WILL CHECK THAT THE RLBA INCREMENTS PROPERLY DURING A

WRITE CHECK.

TEST 36 - PROPER INCREMENT OF RLDA ON WRITE CHECK

THIS TEST WILL CHECK THAT THE RLDA INCREMENTS PROPERLY DURING A WRITE CHECK.

TEST 37 - MULTIPLE SECTOR WRITE CHECK

THIS TEST WILL CHECK THAT WE CAN WRITE CHECK MORE THAN ONE SECTOR AT A TIME.

TEST 38 - FORCE DCK WITH WRITE CHECK

THIS TEST WILL CHECK THAT WE CAN DETECT A DCK DURING A WRITE CHECK. THIS IS DONE BY MODIFYING MEMORY BETWEEN A WRITE AND A WRITE CHECK.

TEST 39 - FORCE DCK WITH WRITE CHECK INTERRUPT

THIS TEST WILL CHECK THAT A DCK DURING A WRITE CHECK WILL CAUSE AN INTERRUPT TO OCCUR.

TEST 40 - CHECK ZERO FILL ON WRITE WITH WRITE CHECK

THIS TEST WILL VERIFY THAT WE CAN SUCCESSFULLY WRITE CHECK ALL WORD COUNTS FROM 1 - 127.

TEST 41 - 42 - EXTENDED CHECK OF WRITE CHECK

THESE TESTS VERIFY THAT WE CAN WRITE CHECK SUCCESSFULLY ALL PATTERNS. PATTERNS USED ARE WALKING 1'S, 0'S, GROWING 1'S, 0'S.

TEST 43 - READ WITHOUT HEADER COMPARE

THIS TEST VERIFIES THAT THE FUNCTION READ WITHOUT HEADER COMPARE (7) RESETS THE CONTROLLER READY AND POSTS NO ERRORS. THE DISK ADDRESS IS SET TO ALL ONES.

TEST 44 - READ WITHOUT HEADER COMPARE INTERRUPT

THIS TEST WILL VERIFY THAT THE FUNCTION READ WITHOUT HEADER COMPARE (7) CAN GENERATE AN INTERRUPT ON COMPLETION.

TEST 45 - CHECK RD W/O HDR CMP READS

THIS TEST CHECKS THAT THE FUNCTION CAN ACTUALLY RECOVER DATA. WE WRITE A PATTERN IN MEMORY AND CHECK THAT THE FUNCTION CAN OVERLAY IT WITH DATA.

TEST 46 - CHECK RLBA INCREMENT WITH RD W/O HDR CMP

THIS TEST CHECKS THAT THE RLBA CAN INCREMENT PROPERLY ON THE  
FUNCTION.

TEST 47 - CHECK RLDA DOES INCREMENT

THIS TEST CHECKS THAT THE RLDA DOES INCREMENT WITH THE  
FUNCTION READ WITHOUT HEADER COMPARE.

.MAIN. MACY11 30A(1052) 22-NOV-78 15:35  
 CZRLBB.P11 22-NOV-78 15:28 TABLE OF CONTENTS

SEQ 0028

|      |                                                              |
|------|--------------------------------------------------------------|
| 88   | GLOBAL DATA                                                  |
| 154  | LIST TO CHECK HEADER COMPARE LOGIC                           |
| 221  | BUFFER FOR READ/WRITE                                        |
| 227  | GLOBAL TEXT                                                  |
| 334  | GLOBAL ERRORS                                                |
| 580  | INITIALIZATION CODE                                          |
| 723  | GLOBAL SUBROUTINES                                           |
| 757  | ROUTINE TO CHECK FOR CONTROLLER ERRORS                       |
| 819  | LOAD RLCS                                                    |
| 1056 | **TEST 1** - WRITE NPR INTEGRITY                             |
| 1107 | **TEST 2** - WRITE FUNCTION                                  |
| 1163 | **TEST 3** - WRITE FUNCTION INTERRUPT                        |
| 1205 | **TEST 4** - PROPER INCREMENT OF RLBA ON WRITE               |
| 1248 | **TEST 5** - PROPER INCREMENT OF RLDA ON WRITE               |
| 1291 | **TEST 6** - FORCE HEADER NOT FOUND WITH WRITE               |
| 1334 | **TEST 7** - FORCE HEADER NOT FOUND WITH WRITE INTERRUPT     |
| 1390 | **TEST 8** - CHECK OPI TIME WITH HDR NT FND                  |
| 1453 | **TEST 9** - MULTIPLE SECTOR TRANSFER ON WRITE               |
| 1506 | **TEST 10** - CHECK DIRECTION OF WRITE NPR                   |
| 1564 | **TEST 11** - CHECK FULL RLBA INCREMENT                      |
| 1614 | **TEST 12** - BA BIT 16 INCREMENT                            |
| 1670 | **TEST 13** - BA BIT 17 INCREMENT                            |
| 1726 | **TEST 14** - TEST READ NPR INTEGRITY                        |
| 1769 | **TEST 15** - READ FUNCTION                                  |
| 1803 | **TEST 16** - READ FUNCTION INTERRUPT                        |
| 1843 | **TEST 17** - CHECK READ NPR DIRECTION                       |
| 1905 | **TEST 18** - PROPER INCREMENT OF RLBA ON READ               |
| 1945 | **TEST 19** - PROPER INCREMENT OF RLDA ON READ               |
| 1987 | **TEST 20** - FORCE HEADER NOT FOUND WITH READ               |
| 2026 | **TEST 21** - FORCE HEADER NOT FOUND WITH READ INTERRUPT     |
| 2075 | **TEST 22** - CHECK HEADER COMPARE LOGIC                     |
| 2206 | **TEST 23** - CHECK MULTIPLE SECTORS ON READ                 |
| 2265 | **TEST 24** - FORCE HDR NT FND AT END OF TRACK               |
| 2301 | **TEST 25** - FORCE NON-EXISTANT MEMORY ERROR                |
| 2344 | **TEST 26** - FORCE NON-EXISTANT MEMORY ERROR INTERRUPT      |
| 2391 | **TEST 27** - CHECK READ WRITE LOOP                          |
| 2477 | **TEST 28** - CHECK SILO LINES                               |
| 2574 | **TEST 29** - CHECK THROUGHPUT OF SILO                       |
| 2670 | **TEST 30** - CHECK ZERO FILL ON WRITE                       |
| 2775 | **TEST 31** - CHECK SECTOR BITS OF HEADER COMPARE            |
| 2887 | **TEST 32** - WRITE CHECK NPR INTEGRITY                      |
| 2970 | **TEST 33** - WRITE CHECK FUNCTION                           |
| 3035 | **TEST 34** - WRITE CHECK FUNCTION INTERRUPT                 |
| 3106 | **TEST 35** - PROPER INCREMENT OF RLBA ON WRITE CHECK        |
| 3179 | **TEST 36** - PROPER INCREMENT OF RLDA ON WRITE CHECK        |
| 3252 | **TEST 37** - MULTIPLE SECTOR WRITE CHECK                    |
| 3338 | **TEST 38** - FORCE DCK WITH WRITE CHECK                     |
| 3411 | **TEST 39** - FORCE DCK WITH WRITE CHECK INTERRUPT           |
| 3495 | **TEST 40** - CHECK ZERO FILL ON WRITE WITH WRITE CHECK      |
| 3572 | **TEST 41** - EXTENDED CHECK OF WRITE CHECK FUNCTION         |
| 3653 | **TEST 42** - EXTENDED CHECK OF WRITE CHECK FUNCTION         |
| 3734 | **TEST 43** - READ WITHOUT HEADER COMPARE FUNCTION           |
| 3764 | **TEST 44** - READ WITHOUT HEADER COMPARE FUNCTION INTERRUPT |
| 3800 | **TEST 45** - CHECK RD W/O HDR CMP ACTUALLY READS            |
| 3862 | **TEST 46** - CHECK RLBA INCREMENT WITH RD W/O HDR CMP       |
| 3908 | **TEST 47** - CHECK RLDA DOES INCREMENT WITH RD W/O HDR CMP  |

.MAIN. MACY11 30A(1052) 22-NOV-78 15:35  
CZRLBB.SUP 23-OCT-78 09:52 TABLE OF CONTENTS

D 3

SEQ 0029

4015 DIAGNOSTIC SUPERVISOR -- LOW CORE SET UP

```

1      .ENABLE AMA
2      .ENABLE ABS
3      .NLIST ME,CND,MD
4
5
6
7
18
19
20      002000      .=2000
21
22      002000      SVC
23      000000      SVCINS=0
24      000000      SVCTAG=0
25
26      002000      POINTER BGNSW,BGNSFT,BGNDU
27
28      002000      BGNMOD MDHEDR
29
30      002000      HEADER CZRLB,B,0,60,60,4,RL01
(4) 002000      103      .ASCII /C/
(4) 002001      132      .ASCII /Z/
(4) 002002      122      .ASCII /R/
(4) 002003      114      .ASCII /L/
(4) 002004      102      .ASCII /B/
(6) 002005      000      .BYTE 0
(6) 002006      000      .BYTE 0
(5) 002007      000      .BYTE 0
(4) 002010      102      .ASCII /B/
(4) 002011      060      .ASCII /O/
(4) 002012      000000      .WORD 0
(4) 002014      000004      .WORD 4
(4) 002016      037606      .WORD L$HARD
(4) 002020      037732      .WORD L$SOFT
(4) 002022      016752      .WORD L$HW
(4) 002024      016766      .WORD L$SW
(4) 002026      040514      .WORD L$LAST
(4) 002030      000000      .WORD 0
(4) 002032      000000      .WORD 0
(4) 002034      000000      .WORD 0
(4) 002036      000000      .WORD 0
(4) 002040      017002      .WORD L$DISPATCH
(4) 002042      000000      .WORD 0
(4) 002044      000000      .WORD 0
(4) 002046      000000      .WORD 0
(4) 002050      002      .BYTE C$REVISION
(3) 002051      002      .BYTE C$EDIT
(4) 002052      000060      .WORD 60
(4) 002054      000060      .WORD 60
(4) 002056      000000      .WORD 0
(5) 002060      000000      .WORD 0
(4) 002062      000000      .WORD 0
(4) 002064      002114      .WORD L$DVTYP
(4) 002066      000000      .WORD 0
(4) 002070      002112      .WORD L$DR
(4) 002072      002112      .WORD L$DRST
(4) 002074      000000      .WORD 0
  
```

```

(4) 002076 020104 .WORD LSDU
(5) 002100 000014 .WORD 14
(4) 002102 000000 .WORD 0
(4) 002104 017140 .WORD L$INIT
(4) 002106 020010 .WORD L$CLEAN
31
32 002110 ENDMOD
33
34
35 002110 DEVREG
(5) 002110 000000 .WORD 0
(2) 002112 000001 .BLKW
36
37 002114 DEV TYP <RL01>
(3) 002114 046122 030460 000 .ASCIZ /RL01/
(2) 002122 002122 .EVEN
38 002122 BGNMOD GLBEQAT
39 002122 EQUALS
40 000001 DRDY=BIT0 ;DRIVE READY (RLCS)
41 000100 INTEN=BIT6 ;INTERRUPT ENABLE (RLCS)
42 100000 ERR=BIT15 ;RL11 ERROR (RLCS)
43 040000 DERR=BIT14 ;RL01 DRIVE ERROR (RLCS)
44 002000 OPI=BIT10 ;OPERATION INCOMPLETE (RLCS)
45 000200 CRDY=BIT7 ;CONTROLLER READY (RLCS)
46 000040 BA17=BIT5 ;EXTENDED ADDRESS BIT 17 (RLCS)
47 000020 BA16=BIT4 ;EXTENDED ADDRESS BIT 16 (RLCS)
48 020000 NXM=BIT13 ;NON-EXISTANT MEMORY (RLCS)
49 000000 DS0=0 ;DRIVE SELECT 0 (RLCS)
50 000400 DS1=BIT8 ;DRIVE SELECT 1 (RLCS)
51 001000 DS2=BIT9 ;DRIVE SELECT 2 (RLCS)
52 001400 DS3=BIT8!BIT9 ;DRIVE SELECT 3 (RLCS)
53 000000 NOOP0=0 ;FUNCTION-NOOP(0)
54 000002 WRCHK=BIT1 ;WRITE CHECK FUNCTION
55 000004 GSTAT=BIT2 ;GET STATUS FUNCTION
56 000006 SEEK=BIT2!BIT1 ;SEEK FUNCTION
57 000010 RDHDR=BIT3 ;READ HEADER FUNCTION
58 000012 WRITE=BIT3.BIT1 ;WRITE DATA FUNCTION
59 000014 READ=BIT3!BIT2 ;READ DATA FUNCTION
60 000016 RDNHD=BIT3!BIT2.BIT1 ;READ W/O HEADER VERIFICATION
61 000202 GODRVR=BIT1!BIT7 ;CRDY AND DRDY
62 000010 DRST=BIT3 ;DRIVE RESET (RLDA)
63 000002 GSBIT=BIT1 ;GET STATUS BIT (RLDA)
64 000001 MK=BIT0 ;MARKER BIT (RLDA)
65 000004 SIGN=BIT2 ;SIGN BIT (RLDA)
66 000100 RHHS=BIT6 ;HEAD SELECT IN READ HEADER
67 000100 STHS=BIT6 ;HEAD SELECT IN STATUS BACK
68 000020 DAHS=BIT4 ;HEAD SELECT IN SEEK
69 ;OFFSET FOR HARDWARE P-TABLE
70
71 000000 CSR=0
72 000002 VECT=2
73 000004 PRIOR=4
74 000006 DRBT=6
75 000010 CNT=10
76
77 ;OFFSET FOR SOFTWARE P-TABLE

```

```

78
79      000000      DLT=0
80      000002      ELT=2
81      000004      SIZE=4
82      000006      DMPCK=6
83      000010      DLMT=10
84
85      002122      BGNMOD      ENDMOD
86      002122      GLBDAT
87
88      .SBTTL      GLOBAL DATA
89
90      002122      000000      CHECK: .WORD 0
91      002124      000000      T.CRC: .WORD 0
92      002126      000000      WHY: .WORD 0
93      002130      000000      CDCNT: .WORD 0
94      002132      000004      ERRVEC: .WORD 4
95      002134      000000      DRIVE: .WORD 0
96      002136      000000      UUT: .WORD 0
97      002140      000000      UNITST: .WORD 0
98      002142      000000      TRPFLG: .WORD 0
99      002144      000000      INTFLG: .WORD 0
100     002146      000000      LDCSR: .WORD 0
101     002150      000077      SECMSK: .WORD 77
102     002152      120001      XPOLY: .WORD 120001
103     002154      000000      BCCFBK: .WORD 0
104     002156      000000      CALBCC: .WORD 0
105     002160      000000      TMP0: .WORD 0
106     002162      000000      TMP1: .WORD 0
107     002164      000000      TMP2: .WORD 0
108     002166      000000      GDDAT: .WORD 0
109     002170      000000      BDDAT: .WORD 0
110     002172      000000      TEMP2: .WORD 0
111     002174      000000      TEMP3: .WORD 0
112     002176      000000      TEMP4: .WORD 0
113     002200      000000      FIRST: .WORD 0
114     002202      177700      CYLSK: .WORD 177700
115     002204      000050      MXSEC1: .WORD 40
116     002206      000047      MAXSEC: .WORD 39
117     002210      000000      DWORD: .WORD 0
118     002212      077600      MAXCYL: .WORD 77600
119     002214      000000      SVHD: .WORD 0
120     002216      000000      B.CS: .WORD 0
121     002220      000000      B.BA: .WORD 0
122     002222      000000      B.DA: .WORD 0
123     002224      000000      B.MP: .WORD 0
124     002226      000000      E.CS: .WORD 0
125     002230      000000      E.BA: .WORD 0
126     002232      000000      E.DA: .WORD 0
127     002234      000000      E.MP: .WORD 0
128     002236      000000      E.MP1: .WORD 0
129     002240      000000      E.MP2: .WORD 0
130     002242      000000      RLCS: .WORD 0
131     002244      000000      RLBA: .WORD 0
132     002246      000000      RLDA: .WORD 0
133     002250      000000      RLMP: .WORD 0

```

; INTERRUPT OCCURANCE FLAG

; LOCATION TO FORM RLCS

; MASK OUT SECTOR

; POLYNOMIAL FOR CRC 16

; LOCATION USED BY 'SIMBCC'

; LOCATION USED BY 'SIMBCC'

; LOCATION USED BY 'SIMBCC'

; LOCATION USED BY 'SIMBCC'

; LOCATION USED BY 'SIMBCC'

; FIRST SECTOR READ

; MASK CYLINDER AND HEAD SELECT

; MAX SECTOR ADDRESS +1

; MAX SECTOR ADDRESS

; DIFFERENCE WORD (SEEK)

; MAXIMUM CYLINDER ADDRESS

; SAVE CURRENT HEAD SELECT

; CS - BEFORE OPERATION

; BA - BEFORE OPERATION

; DA - BEFORE OPERATION

; MF - BEFORE OPERATION

; CS - AT OCCURANCE OF ERROR

; BA - AT OCCURANCE OF ERROR

; DA - AT OCCURANCE OF ERROR

; MP - AT OCCURANCE OF ERROR

|     |        |        |          |       |      |  |                         |
|-----|--------|--------|----------|-------|------|--|-------------------------|
| 134 | 002252 | 000000 | BCSR:    | .WORD | 0    |  | :CSR FROM P TABLE       |
| 135 | 002254 | 000000 | BVEC:    | .WORD | 0    |  | :VECTOR FROM P TABLE    |
| 136 | 002256 | 000000 | BPRIOR:  | .WORD | 0    |  | :BR LEVEVL FROM P TABLE |
| 137 | 002260 | 000000 | FNDFNC:  | .WORD | 0    |  |                         |
| 138 | 002262 | 000000 | XMEM:    | .WORD | 0    |  |                         |
| 139 | 002264 | 000000 | TRYFNC:  | .WORD | 0    |  | :                       |
| 140 | 002266 | 000000 | ERFLG:   | .WORD | 0    |  |                         |
| 141 | 002270 | 001212 | LOPIMX:  | .WORD | 650. |  |                         |
| 142 | 002272 | 000233 | LOPIMN:  | .WORD | 155. |  |                         |
| 143 | 002274 | 000620 | UOPIMX:  | .WORD | 400. |  |                         |
| 144 | 002276 | 000240 | UOPIMN:  | .WORD | 160. |  |                         |
| 145 | 002300 | 000000 | OPIMN:   | .WORD | 0    |  |                         |
| 146 | 002302 | 000000 | OPIMX:   | .WORD | 0    |  |                         |
| 147 | 002304 | 000000 | PWRFLG:  | .WORD | 0    |  |                         |
| 148 | 002306 | 000000 | T.CNTRL: | .WORD | 0    |  |                         |
| 149 | 002310 | 000000 | DERFLG:  | .WORD | 0    |  |                         |
| 150 | 002312 | 000000 | ERPOINT: | .WORD | 0    |  |                         |
| 151 | 002314 | 000074 | ERCOUNT: | .BLKW | 00.  |  |                         |

|     |        |        |         |                                    |       |  |         |
|-----|--------|--------|---------|------------------------------------|-------|--|---------|
| 152 |        |        |         |                                    |       |  |         |
| 153 |        |        |         |                                    |       |  |         |
| 154 |        |        | .SBTTL  | LIST TO CHECK HEADER COMPARE LOGIC |       |  |         |
| 155 | 002504 | 000000 | HDRTAB: | .WORD                              | 0     |  | :WALK 1 |
| 156 | 002506 | 000001 |         | .WORD                              | BIT0  |  |         |
| 157 | 002510 | 000002 |         | .WORD                              | BIT1  |  |         |
| 158 | 002512 | 000004 |         | .WORD                              | BIT2  |  |         |
| 159 | 002514 | 000010 |         | .WORD                              | BIT3  |  |         |
| 160 | 002516 | 000020 |         | .WORD                              | BIT4  |  |         |
| 161 | 002520 | 000040 |         | .WORD                              | BIT5  |  |         |
| 162 | 002522 | 000100 |         | .WORD                              | BIT6  |  |         |
| 163 | 002524 | 000200 |         | .WORD                              | BIT7  |  |         |
| 164 | 002526 | 000400 |         | .WORD                              | BIT8  |  |         |
| 165 | 002530 | 001000 |         | .WORD                              | BIT9  |  |         |
| 166 | 002532 | 002000 |         | .WORD                              | BIT10 |  |         |
| 167 | 002534 | 004000 |         | .WORD                              | BIT11 |  |         |
| 168 | 002536 | 010000 |         | .WORD                              | BIT12 |  |         |
| 169 | 002540 | 020000 |         | .WORD                              | BIT13 |  |         |
| 170 | 002542 | 040000 |         | .WORD                              | BIT14 |  |         |
| 171 | 002544 | 000003 |         | .WORD                              | 3     |  | :GROW 1 |
| 172 | 002546 | 000007 |         | .WORD                              | 7     |  |         |
| 173 | 002550 | 000017 |         | .WORD                              | 17    |  |         |
| 174 | 002552 | 000037 |         | .WORD                              | 37    |  |         |
| 175 | 002554 | 000137 |         | .WORD                              | 137   |  |         |
| 176 | 002556 | 000337 |         | .WORD                              | 337   |  |         |
| 177 | 002560 | 000737 |         | .WORD                              | 737   |  |         |
| 178 | 002562 | 001737 |         | .WORD                              | 1737  |  |         |
| 179 | 002564 | 003737 |         | .WORD                              | 3737  |  |         |
| 180 | 002566 | 007737 |         | .WORD                              | 7737  |  |         |
| 181 | 002570 | 017737 |         | .WORD                              | 17737 |  |         |
| 182 | 002572 | 037737 |         | .WORD                              | 37737 |  |         |
| 183 | 002574 | 077737 |         | .WORD                              | 77737 |  |         |
| 184 | 002576 | 077736 |         | .WORD                              | 77736 |  | :GROW 0 |
| 185 | 002600 | 077734 |         | .WORD                              | 77734 |  |         |
| 186 | 002602 | 077730 |         | .WORD                              | 77730 |  |         |
| 187 | 002604 | 077720 |         | .WORD                              | 77720 |  |         |
| 188 | 002606 | 077700 |         | .WORD                              | 77700 |  |         |
| 189 | 002610 | 077600 |         | .WORD                              | 77600 |  |         |

```

190 002612 077400 .WORD 77400
191 002614 077000 .WORD 77000
192 002616 076000 .WORD 76000
193 002620 074000 .WORD 74000
194 002622 070000 .WORD 70000
195 002624 060000 .WORD 60000
196 002626 077735 .WORD 77735 ;WALK 0
197 002630 077733 .WORD 77733
198 002632 077727 .WORD 77727
199 002634 077717 .WORD 77717
200 002636 077637 .WORD 77637
201 002640 077537 .WORD 77537
202 002642 077337 .WORD 77337
203 002644 076737 .WORD 76737
204 002646 075737 .WORD 75737
205 002650 073737 .WORD 73737
206 002652 067737 .WORD 67737
207 002654 057737 .WORD 57737
208 002656 037737 .WORD 37737
209 002660 000000 HDREND: .WORD 0
210
211
212 002662 000001 000002 000004 DATPAT: .WORD 1,2,4,10,20,40,100,200,400,1000,2000,4000,10000,20000,40000,100000
    002670 000010 000020 000040
    002676 000100 000200 000400
    002704 001000 002000 004000
    002712 010000 020000 040000
    002720 100000
213 002722 177777 177776 177775 .WORD 177777,177776,177775,177773,177767,177757,177737,177677
    002730 177773 177767 177757
    002736 177737 177677
214 002742 177577 177377 176777 .WORD 177577,177377,176777,175777,173777,167777,157777,137777
    002750 175777 173777 167777
    002756 157777 137777
215 002762 077777 177774 177770 .WORD 77777,177774,177770,177760,177740,177700,177600,177400
    002770 177760 177740 177700
    002776 177600 177400
216 003002 177000 176000 174000 .WORD 177000,176000,174000,170000,160000,140000,3,7,17,37,77
    003010 170000 160000 140000
    003016 000003 000007 000017
    003024 000037 000077
217 003030 000177 000377 000777 .WORD 177,377,777,1777,3777,7777,17777,37777,0
    003036 001777 003777 007777
    003044 017777 037777 000000
218
219
220
221 .SBTTL BUFFER FOR READ/WRITE
222 003052 002000 BUF: .BLKW 1024.
223
224
225 007052 ENDMOD
226
227 .SBTTL GLOBAL TEXT
228 007052 BEGINMOD GLBTXT
232 007052 047516 041440 047112 NDRS: .ASCII /NO CONTROLLER/
  
```

233 007070 047516 042040 044522 NORDY: .ASCIZ /NO DRIVE/  
234 007101 103 035123 000040 ARLCS: .ASCIZ /CS: /  
235 007106 041040 035101 000040 ARLBA: .ASCIZ /BA: /  
236 007114 042040 035101 000040 ARLDA: .ASCIZ /DA: /  
237 007122 046440 035120 000040 ARLMP: .ASCIZ /MP: /  
238 007130 042502 047506 042522 BEREG: .ASCIZ /BEFORE COMMAND: /  
239 007151 124 046511 020105 AFREG: .ASCIZ /TIME OF ERROR: /  
240 007172 047503 052116 047522 CRTIM: .ASCIZ /CONTROLLER TIMED OUT/  
241 007217 104 044522 042526 DRTIM: .ASCIZ /DRIVE READY TIMED OUT/  
242 007245 040 051104 000126 DEMES: .ASCIZ / DRV/  
243 007252 047040 046530 000 NXMMES: .ASCIZ / NXM/  
244 007257 040 050117 000111 OPIES: .ASCIZ / OPI/  
245 007264 044040 051103 000103 HCRCMES: .ASCIZ / HCRC/  
246 007272 044040 043116 000 HNFMES: .ASCIZ / HNF/  
247 007277 040 041504 000113 DCKMES: .ASCIZ / DCK/  
248 007304 042040 052114 000 DLTMES: .ASCIZ / DLT/  
249 007311 015 000 LF: .ASCIZ <15>  
250 007313 015 000012 MSCRLF: .ASCIZ <15><12>  
251 007316 041440 046517 000120 COMP: .ASCIZ / COMP/  
252 007324 047506 041522 042105 OPIERR: .ASCIZ /FORCED OPI(GET STATUS) CAUSED OTHER ERRORS/  
253 007377 116 047517 020120 NOPMES: .ASCIZ /NOOP OPERATION-FLAG MODE/  
254 007430 047516 050117 047440 NOPINT: .ASCIZ /NOOP OPERATION-INTR. MODE/  
255 007462 051127 052111 020105 WCKMES: .ASCIZ /WRITE CHECK OPERATION-FLAG MODE/  
256 007522 051127 052111 020105 WCKINT: .ASCIZ /WRITE CHECK OPERATION-INTR. MODE/  
257 007563 122 040505 020104 RHDMES: .ASCIZ /READ HEADER OPERATION-FLAG MODE/  
258 007623 122 040505 020104 RHDINT: .ASCIZ /READ HEADER OPERATION-INTR. MODE/  
259 007664 042523 045505 047440 SEKMES: .ASCIZ /SEEK OPERATION-FLAG MODE/  
260 007715 123 042505 020113 SEKINT: .ASCIZ /SEEK OPERATION-INTR. MODE/  
261 007747 107 052105 051440 GSTMES: .ASCIZ /GET STATUS OPERATION-FLAG MODE/  
262 010006 042507 020124 052123 GSTINT: .ASCIZ /GET STATUS OPERATION-INTR MODE/  
263 010045 122 040505 020104 RDDMES: .ASCIZ /READ OPERATION-FLAG MODE/  
264 010076 042522 042101 047440 RDDINT: .ASCIZ /READ OPERATION-INTR MODE/  
265 010127 127 044522 042524 WRTMES: .ASCIZ /WRITE OPERATION-FLAG MODE/  
266 010161 127 044522 042524 WRTINT: .ASCIZ /WRITE OPERATION-INTR MODE/  
267 010213 122 040505 020104 RDNMES: .ASCIZ %READ W/O HEADER - FLAG MODE%  
268 010247 122 040505 020104 RDNINT: .ASCIZ %READ W/O HEADER - INTR MODE%  
269 010303 103 047101 052047 SKHOME: .ASCIZ /CAN'T SEEK TO TRACK 0/  
270 010331 127 044522 042524 WRLOCK: .ASCIZ /WRITE LOCK ERROR/  
271 010352 046122 051503 041440 EM1: .ASCIZ /RLCS CONTAINED FOLLOWING ERROR(S): /  
272 010417 000170 EM100: .BLKB 120.  
273 010607 116 020117 047111 EM4: .ASCIZ /NO INTERRUPT ON READ OPERATION/  
274 010646 042522 042101 047440 EM5: .ASCIZ /READ OPERATION DID NOT WRITE MEMORY/  
275 010712 046122 040502 042040 EM6: .ASCIZ /RLBA DID NOT INCREMENT PROPERLY DURING READ/  
276 010766 042523 052103 051117 EM7: .ASCIZ /SECTOR DID NOT INCREMENT PROPERLY AFTER READ/  
277 011043 110 040505 042504 EM10: .ASCIZ /HEADER NOT FOUND COULD NOT BE FORCED/  
278 011110 051127 047117 020107 EM11: .ASCIZ /WRONG CYLINDER ON SEEK/  
279 011137 110 040505 042504 EM12: .ASCIZ /HEADER NOT FOUND WOULD NOT SET/  
280 011176 051104 053111 020105 EM13: .ASCIZ /DRIVE READY WOULD NOT SET/  
281 011230 044504 045523 040440 EM14: .ASCIZ /DISK ADDRESS INCORRECT AFTER MULTIPLE SECTOR READ/  
282 011312 051104 053111 020105 EM16: .ASCIZ /DRIVE ERROR ON WRITE OPERATION/  
283 011351 116 020117 047111 EM17: .ASCIZ /NO INTERRUPT ON WRITE OPERATION/  
284 011411 122 041114 020101 EM20: .ASCIZ /RLBA DID NOT INCREMENT PROPERLY DURING WRITE/  
285 011466 042523 052103 051117 EM21: .ASCIZ /SECTOR DID NOT INCREMENT PROPERLY AFTER WRITE/  
286 011544 044504 045523 040440 EM22: .ASCIZ /DISK ADDRESS (RLDA) INCORRECT AFTER MULTIPLE SECTOR WRITE/  
287 011636 042110 020122 047516 EM23: .ASCIZ /;DR NOT FND COULD NOT BE FORCED AT END OF TRACK/  
288 011716 047516 026516 054105 EM24: .ASCIZ /NON-EXISTANT MEMORY ERROR COULD NOT BE FORCED/

```
289 011774 040504 040524 041440 EM25: .ASCIIZ %DATA COMPARISON ERROR - READ/WRITE ERROR%
290
291 012045 127 044522 042524 EM26: .ASCIIZ /WRITE OPERATION MODIFIED MEMORY/
292 012105 105 051122 051117 EM27: .ASCIIZ /ERROR ON PARTIAL SECTOR WRITE - ZERO FILL CHECK/
293 012165 122 041114 020101 EM30: .ASCIIZ /RLBA DID NOT INCREMENT PROPERLY/
294 012225 102 020101 044502 EM31: .ASCIIZ /BA BIT 16 DID NOT SET ON INCREMENT/
295 012270 040502 041040 052111 EM32: .ASCIIZ /BA BIT 17 SET ON BA16 INCREMENT TEST/
296 012335 122 041114 020101 EM33: .ASCIIZ /RLBA DID NOT INCREMENT WITH BA16/
297 012376 040502 041040 052111 EM34: .ASCIIZ /BA BIT 17 DID NOT SET ON INCREMENT/
298 012441 102 020101 044502 EM35: .ASCIIZ /BA BIT 16 DID NOT CLEAR ON INCREMENT/
299 012506 046122 040502 042040 EM36: .ASCIIZ /RLBA DID NOT INCREMENT WITH BA17/
300 012547 122 040505 024104 EM40: .ASCIIZ /READ(FUNCTION 7) DID NOT INTERRUPT/
301 012612 042522 042101 043050 EM41: .ASCIIZ /READ(FUNCTION 7) ERROR - BAD DATA/
302 012654 042522 042101 043050 EM42: .ASCIIZ /READ(FUNCTION 7) ERROR AT END OF TRACK/
303 012723 116 020117 047111 EM43: .ASCIIZ /NO INTERRUPT WITH HDR NT FND FORCED/
304 012767 116 020117 047111 EM44: .ASCIIZ /NO INTERRUPT WITH NXM FORCED/
305 013024 051105 047522 020122 EM45: .ASCIIZ %ERROR ON BIT BANG OF SILO%
306 013056 044523 047514 047440 EM47: .ASCIIZ /SILO OPERATION FAILURE/
307 013105 110 040505 042504 EM50: .ASCIIZ /HEADER COMPARE FAILURE - SECTOR/
308 013145 127 044522 042524 EM51: .ASCIIZ /WRITE NPR CAUSED BUS TRAP/
309 013177 122 040505 020104 EM52: .ASCIIZ /READ NPR CAUSED BUS TRAP/
310 013230 042522 042101 053440 EM55: .ASCIIZ ?READ W/O HDR CMP OPERATION DID NOT WRITE MEMORY?
311 013310 046122 040502 042040 EM53: .ASCIIZ ?RLBA DID NOT INCREMENT PROPERLY DURING READ W/O HDR CMP?
312 013400 046122 040504 042040 EM54: .ASCIIZ ?RLDA DID NOT INCREMENT AFTER READ W/O HDR CMP?
313 013456 050117 020111 044524 EM56: .ASCIIZ /OPI TIMING ERROR/
314 013477 127 044522 042524 EM57: .ASCIIZ /WRITE CHECK NPR CAUSED BUS TRAP/
315 013537 127 044522 042524 EM60: .ASCIIZ /WRITE CHECK DID NOT INTERRUPT/
316 013575 122 041114 020101 EM61: .ASCIIZ /RLBA DID NOT INCREMENT PROPERLY DURING WRCHK/
317 013652 046122 040504 042040 EM62: .ASCIIZ /RLDA DID NOT INCREMENT PROPERLY DURING WRCHK/
318 013727 122 042114 020101 EM63: .ASCIIZ /RLDA DID NOT INCREMENT PROPERLY AFTER A MULTIPLE SECTOR WRITE CHK/
319 014031 127 044522 042524 EM64: .ASCIIZ /WRITE CHECK OF PARTIAL SECTOR WRITE FAILURE/
320 014105 103 047101 047040 EM65: .ASCIIZ /CAN NOT FORCE DCK ON WRITE CHECK/
321 014146 040503 020116 047516 EM66: .ASCIIZ /CAN NOT FORCE INTERRUPT WITH DCK ON WRCHK/
322 014220 051127 052111 020105 EM70: .ASCIIZ /WRITE CHECK FAILURE/
323
324 .EVEN
325
326
330 014244 ENDMOD
331
332 014244 BGNMOD GLBERR
333
334 .SBTTL GLOBAL ERRORS
335 014244 BGNMSG ERRO
336
337 014244 004737 015256 JSR PC,LINE1
338 014250 004737 015312 JSR PC,LINE2
339
340
341 014254 004537 020126 JSR R5,CKERLT ;INCREMENT ERROR AND CHECK LIMIT
342
343 014260 ENDMMSG
(3) 014260 L10000:
(3) 014260 104023 EMT C$MSG
344
345 014262 BGNMSG ERR1
```

```

346
347 014262 004737 015256          JSR      PC,LINE1
348
349
350 014266 004537 020126          JSR      R5,CKERLT          ;INCREMENT ERROR AND CHECK LIMIT
351
352 014272          ENDMSG
(3) 014272          L10001:
(3) 014272 104023          EMT      C$MSG
353
354 014274          BGNMSG  ERR2
355
356 014274 004737 015256          JSR      PC,LINE1
357 014300          PRINTB  #FRMT4,GDDAT,BDDAT
(9) 014300 013746 002170          MOV      BDDAT,-(SP)
(8) 014304 013746 002166          MOV      GDDAT,-(SP)
(7) 014310 012746 015733          MOV      #FRMT4,-(SP)
(6) 014314 012746 000003          MOV      #3,-(SP)
(3) 014320 010600          MOV      SP,R0
(4) 014322 104014          EMT      C$PNTB
(4) 014324 062706 000010          ADD      #10,SP
358
359
360 014330 004537 020126          JSR      R5,CKERLT          ;INCREMENT ERROR AND CHECK LIMIT
361
362 014334          ENDMSG
(3) 014334          L10002:
(3) 014334 104023          EMT      C$MSG
363
364 014336          BGNMSG  ERR3
365
366 014336 004737 015256          JSR      PC,LINE1
367 014342 004737 015312          JSR      PC,LINE2
368 014346          PRINTB  #FRMT5,TMP0,BDDAT,GDDAT
(10) 014346 013746 002166          MOV      GDDAT,-(SP)
(9) 014352 013746 002170          MOV      BDDAT,-(SP)
(8) 014356 013746 002160          MOV      TMP0,-(SP)
(7) 014362 012746 015771          MOV      #FRMT5,-(SP)
(6) 014366 012746 000004          MOV      #4,-(SP)
(3) 014372 010600          MOV      SP,R0
(4) 014374 104014          EMT      C$PNTB
(4) 014376 062706 000012          ADD      #12,SP
369
370
371 014402 004537 020126          JSR      R5,CKERLT          ;INCREMENT ERROR AND CHECK LIMIT
372
373 014406          ENDMSG
(3) 014406          L10003:
(3) 014406 104023          EMT      C$MSG
374
375 014410          BGNMSG  ERR4
376
377 014410 004737 015256          JSR      PC,LINE1
378 014414 004737 015312          JSR      PC,LINE2
379 014420          PRINTB  #FRMT4,GDDAT,BDDAT
(9) 014420 013746 002170          MOV      BDDAT,-(SP)

```

```

(8) 014424 013746 002166      MOV      GDDAT,-(SP)
(7) 014430 012746 015733      MOV      #FRMT4,-(SP)
(6) 014434 012746 000003      MOV      #3,-(SP)
(3) 014440 010600              MOV      SP,R0
(4) 014442 104014              EMT      C$PNTB
(4) 014444 062706 000010      ADD      #10,SP
380
381
382 014450 004537 020126      JSR      R5,CKERLT          ;INCREMENT ERROR AND CHECK LIMIT
383
384 014454              ENDMSG
(3) 014454              L10004:
(3) 014454 104023          EMT      C$MSG
385
386 014456              BGNMSG  ERR5
387
388 014456 004737 015256      JSR      PC,LINE1
389 014462              PRINTB  #FRMT3,RESTMS
(8) 014462 013746 020440      MOV      RESTMS,-(SP)
(7) 014466 012746 015726      MOV      #FRMT3,-(SP)
(6) 014472 012746 000002      MOV      #2,-(SP)
(3) 014476 010600              MOV      SP,R0
(4) 014500 104014              EMT      C$PNTB
(4) 014502 062706 000006      ADD      #6,SP
390
391
392 014506 004537 020126      JSR      R5,CKERLT          ;INCREMENT ERROR AND CHECK LIMIT
393
394 014512              ENDMSG
(3) 014512              L10005:
(3) 014512 104023          EMT      C$MSG
395
396 014514              BGNMSG  ERR6
397
398 014514 004737 015256      JSR      PC,LINE1
399 014520 004737 015534      JSR      PC,LINE3
400 014524 004737 015312      JSR      PC,LINE2
401
402
403 014530              PRINTB  #FRMT99
(7) 014530 012746 016667      MOV      #FRMT99,-(SP)
(6) 014534 012746 000001      MOV      #1,-(SP)
(3) 014540 010600              MOV      SP,R0
(4) 014542 104014              EMT      C$PNTB
(4) 014544 062706 000004      ADD      #4,SP
404 014550 004537 020126      JSR      R5,CKERLT          ;INCREMENT ERROR AND CHECK LIMIT
405
406 014554              ENDMSG
(3) 014554              L10006:
(3) 014554 104023          EMT      C$MSG
407
408 014556              BGNMSG  ERR7
409
410
411
412 014556 004537 020126      JSR      R5,CKERLT          ;INCREMENT ERROR AND CHECK LIMIT

```

```

413
414 014562
(3) 014562
(3) 014562 104023
415
416
417
418 014564
419
420 014564 004737 015256 JSR PC,LINE1
421 014570 004737 015312 JSR PC,LINE2
422 014574 PRINTB #FRMT6,TMP1,GDDAT,BDDAT
(10) 014574 013746 002170 MOV BDDAT,-(SP)
(9) 014600 013746 002166 MOV GDDAT,-(SP)
(8) 014604 013746 002162 MOV TMP1,-(SP)
(7) 014610 012746 016042 MOV #FRMT6,-(SP)
(6) 014614 012746 000004 MOV #4,-(SP)
(3) 014620 010600 MOV SP,R0
(4) 014622 104014 EMT C$PNTB
(4) 014624 062706 000012 ADD #12,SP
423
424
425 014630 004537 020126 JSR R5,CKERLT ;INCREMENT ERROR AND CHECK LIMIT
426
427 014634
(3) 014634
(3) 014634 104023
428
429 014636
430
431 014636 004737 015236 JSR PC,LINE1
432 014642 004737 015312 JSR PC,LINE2
433 014646 PRINTB #FRMT4,TMP0,R2
(9) 014646 010246 MOV R2,-(SP)
(8) 014650 013746 002160 MOV TMP0,-(SP)
(7) 014654 012746 015733 MOV #FRMT4,-(SP)
(6) 014660 012746 000003 MOV #3,-(SP)
(3) 014664 010600 MOV SP,R0
(4) 014666 104014 EMT C$PNTB
(4) 014670 062706 000010 ADD #10,SP
434
435
436 014674 004537 020126 JSR R5,CKERLT ;INCREMENT ERROR AND CHECK LIMIT
437
438 014700
(3) 014700
(3) 014700 104023
439
440 014702
441
442 014702 004737 015256 JSR PC,LINE1
443 014706 004737 015312 JSR PC,LINE2
444 014712 PRINTB #FRMT7,TMP1,GDDAT,BDDAT
(10) 014712 013746 002170 MOV BDDAT,-(SP)
(9) 014716 013746 002166 MOV GDDAT,-(SP)
(8) 014722 013746 002162 MOV TMP1,-(SP)

```

```

(7) 014726 012746 016117      MOV    #FRMT7,-(SP)
(6) 014732 012746 000004      MOV    #4,-(SP)
(3) 014736 010600              MOV    SP,R0
(4) 014740 104014              EMT    C$PNTB
(4) 014742 062706 000012      ADD     #12,SP
445
446
447 014746 004537 020126      JSR     R5,CKERLT          ;INCREMENT ERROR AND CHECK LIMIT
448
449 014752              ENDMSG
(3) 014752              L10012:
(3) 014752 104023          EMT     C$MSG
450
451 014754              BGNMSG  ERR11
452
453 014754 004737 015256      JSR     PC,LINE1
454 014760 004737 015312      JSR     PC,LINE2
455 014764              PRINTB  #FRMT8,TMP0,GDDAT,BDDAT
(10) 014764 013746 002170      MOV     BDDAT,-(SP)
(9) 014770 013746 002166      MOV     GDDAT,-(SP)
(8) 014774 013746 002160      MOV     TMP0,-(SP)
(7) 015000 012746 016171      MOV     #FRMT8,-(SP)
(6) 015004 012746 000004      MOV     #4,-(SP)
(3) 015010 010600              MOV     SP,R0
(4) 015012 104014              EMT    C$PNTB
(4) 015014 062706 000012      ADD     #12,SP
456
457
458 015020 004537 020126      JSR     R5,CKERLT          ;INCREMENT ERROR AND CHECK LIMIT
459
460 015024              ENDMSG
(3) 015024              L10013:
(3) 015024 104023          EMT     C$MSG
461
462 015026              BGNMSG  ERR12
463
464 015026 004737 015256      JSR     PC,LINE1
465 015032 004737 015312      JSR     PC,LINE2
466 015036              PRINTB  #FRMT9,TMP1,R3,GDDAT,BDDAT
(11) 015036 013746 002170      MOV     BDDAT,-(SP)
(10) 015042 013746 002166      MOV     GDDAT,-(SP)
(9) 015046 010346              MOV     R3,-(SP)
(8) 015050 013746 002162      MOV     TMP1,-(SP)
(7) 015054 012746 016312      MOV     #FRMT9,-(SP)
(6) 015060 012746 000005      MOV     #5,-(SP)
(3) 015064 010600              MOV     SP,R0
(4) 015066 104014              EMT    C$PNTB
(4) 015070 062706 000014      ADD     #14,SP
467
468
469 015074 004537 020126      JSR     R5,CKERLT          ;INCREMENT ERROR AND CHECK LIMIT
470
471 015100              ENDMSG
(3) 015100              L10014:
(3) 015100 104023          EMT     C$MSG
472

```

```

473 015102          BGNMSG  ERR13
474
475 015102 004737 015256      JSR      PC,LINE1
476 015106          PRINTB   #FRMT10,OPIMN,OPIMX,BDDAT
(10) 015106 013746 002170      MOV      BDDAT,-(SP)
(9) 015112 013746 002302      MOV      OPIMX,-(SP)
(8) 015116 013746 002300      MOV      OPIMN,-(SP)
(7) 015122 012746 016415      MOV      #FRMT10,-(SP)
(6) 015126 012746 000004      MOV      #4,-(SP)
(3) 015132 010600          MOV      SP,R0
(4) 015134 104014          EMT      C$PNTB
(4) 015136 062706 000012      ADD      #12,SP
477
478
479 015142 004537 020126      JSR      R5,CKERLT          ;INCREMENT ERROR AND CHECK LIMIT
480
481 015146          ENDMSG
(3) 015146          L10015:
(3) 015146 104023          EMT      C$MSG
482
483 015150          BGNMSG  ERR14
484
485 015150 004737 015256      JSR      PC,LINE1
486 015154 004737 015312      JSR      PC,LINE2
487 015160          PRINTB   #FRMT14,TMP1,#BUF
(9) 015160 012746 003052      MOV      #BUF,-(SP)
(8) 015164 013746 002162      MOV      TMP1,-(SP)
(7) 015170 012746 016241      MOV      #FRMT14,-(SP)
(6) 015174 012746 000003      MOV      #3,-(SP)
(3) 015200 010600          MOV      SP,R0
(4) 015202 104014          EMT      C$PNTB
(4) 015204 062706 000010      ADD      #10,SP
488
489
490 015210 004537 020126      JSR      R5,CKERLT          ;INCREMENT ERROR AND CHECK LIMIT
491
492 015214          ENDMSG
(3) 015214          L10016:
(3) 015214 104023          EMT      C$MSG
493
494 015216          BGNMSG  ERR15
495
496 015216 004737 015256      JSR      PC,LINE1
497 015222 004737 015312      JSR      PC,LINE2
498 015226          PRINTB   #FRMT15,R2
(8) 015226 010246          MOV      R2,-(SP)
(7) 015230 012746 016723      MOV      #FRMT15,-(SP)
(6) 015234 012746 000002      MOV      #2,-(SP)
(3) 015240 010600          MOV      SP,R0
(4) 015242 104014          EMT      C$PNTB
(4) 015244 062706 000006      ADD      #6,SP
499 015250 004537 020126      JSR      R5,CKERLT
500
501 015254          ENDMSG
(3) 015254          L10017:
(3) 015254 104023          EMT      C$MSG

```

```

502
503 015256 LINE1: PRINTB #FRMT1,RLCS,<B,DRIVE+1>
(9) 015256 005046 CLR -(SP)
(9) 015260 153716 002135 BISB DRIVE+1,(SP)
(8) 015264 013746 002242 MOV RLCS, -(SP)
(7) 015270 012746 015606 MOV #FRMT1, -(SP)
(6) 015274 012746 000003 MOV #3, -(SP)
(3) 015300 010600 MOV SP,R0
(4) 015302 104014 EMT C$PNTB
(4) 015304 062706 000010 ADD #10,SP
504 015310 000207 RTS PC
505
506 015312 LINE2: PRINTB #FRMT2,#BEREG,#ARLCS,B.CS,#ARLBA,B.BA
(12) 015312 013746 002220 MOV B.BA, -(SP)
(11) 015316 012746 007106 MOV #ARLBA, -(SP)
(10) 015322 013746 002216 MOV B.CS, -(SP)
(9) 015326 012746 007101 MOV #ARLCS, -(SP)
(8) 015332 012746 007130 MOV #BEREG, -(SP)
(7) 015336 012746 015645 MOV #FRMT2, -(SP)
(6) 015342 012746 000006 MOV #6, -(SP)
(3) 015346 010600 MOV SP,R0
(4) 015350 104014 EMT C$PNTB
(4) 015352 062706 000016 ADD #16,SP
507 015356 PRINTB #FRMT2A,#ARLDA,B.DA,#ARLMP,B.MP
(11) 015356 013746 002224 MOV B.MP, -(SP)
(10) 015362 012746 007122 MOV #ARLMP, -(SP)
(9) 015366 013746 002222 MOV B.DA, -(SP)
(8) 015372 012746 007114 MOV #ARLDA, -(SP)
(7) 015376 012746 015664 MOV #FRMT2A, -(SP)
(6) 015402 012746 000005 MOV #5, -(SP)
(3) 015406 010600 MOV SP,R0
(4) 015410 104014 EMT C$PNTB
(4) 015412 062706 000014 ADD #14,SP
508 015416 PRINTB #FRMT2,#AFREG,#ARLCS,E.CS,#ARLBA,E.BA
(12) 015416 013746 002230 MOV E.BA, -(SP)
(11) 015422 012746 007106 MOV #ARLBA, -(SP)
(10) 015426 013746 002226 MOV E.CS, -(SP)
(9) 015432 012746 007101 MOV #ARLCS, -(SP)
(8) 015436 012746 007151 MOV #AFREG, -(SP)
(7) 015442 012746 015645 MOV #FRMT2, -(SP)
(6) 015446 012746 000006 MOV #6, -(SP)
(3) 015452 010600 MOV SP,R0
(4) 015454 104014 EMT C$PNTB
(4) 015456 062706 000016 ADD #16,SP
509 015462 PRINTB #FRMT2B,#ARLDA,E.DA,#ARLMP,E.MP,E.MP1,E.MP2
(13) 015462 013746 002240 MOV E.MP2, -(SP)
(12) 015466 013746 002236 MOV E.MP1, -(SP)
(11) 015472 013746 002234 MOV E.MP, -(SP)
(10) 015476 012746 007122 MOV #ARLMP, -(SP)
(9) 015502 013746 002232 MOV E.DA, -(SP)
(8) 015506 012746 007114 MOV #ARLDA, -(SP)
(7) 015512 012746 015677 MOV #FRMT2B, -(SP)
(6) 015516 012746 000007 MOV #7, -(SP)
(3) 015522 010600 MOV SP,R0
(4) 015524 104014 EMT C$PNTB
(4) 015526 062706 000020 ADD #20,SP

```

```

510 015532 000207          RTS      PC
511
512 015534          LINE3: PRINTB  #FRMT3,#EM1
(8) 015534 012746 010352      MOV      #EM1,-(SP)
(7) 015540 012746 015726      MOV      #FRMT3,-(SP)
(6) 015544 012746 000002      MOV      #2,-(SP)
(3) 015550 010600          MOV      SP,R0
(4) 015552 104014          EMT      C$PNTB
(4) 015554 062706 000006      ADD      #6,SP
513 015560          PRINTB  #FRMT3,#EM100
(8) 015560 012746 010417      MOV      #EM100,-(SP)
(7) 015564 012746 015726      MOV      #FRMT3,-(SP)
(6) 015570 012746 000002      MOV      #2,-(SP)
(3) 015574 010600          MOV      SP,R0
(4) 015576 104014          EMT      C$PNTB
(4) 015600 062706 000006      ADD      #6,SP
514 015604 000207          RTS      PC
515
516
520
521 015606 040445 047503 052116 FRMT1: .ASCIIZ  /%ACONTROLLER: %06%A DRIVE: %01/
522 015645      045 022516 022524 FRMT2: .ASCIIZ  /%N%T%T%06%T%06/
523 015664 052045 047445 022466 FRMT2A: .ASCIIZ /%T%06%T%06/
524 015677      045 022524 033117 FRMT2B: .ASCIIZ /%T%06%T%06%A %06%A %06/
525 015726 047045 052045      000 FRMT3: .ASCIIZ  /%N%T/
526 015733      045 022516 042501 FRMT4: .ASCIIZ  /%N%AE%P'D: %06%A REC'D: %06%N/
527 015771      045 022516 046101 FRMT5: .ASCIIZ  /%N%ALAST: %06%A PRES: %06%A EXP'D: %06%N/
528 016042 047045 040445 052502 FRMT6: .ASCIIZ  /%N%ABUS ADR: %06%A EXP'D: %06%A REC'D: %06%N/
529 016117      045 022516 053501 FRMT7: .ASCIIZ  /%N%AWORD: %D3%A EXP'D: %06%A REC'D: %06%N/
530 016171      045 022516 042101 FRMT8: .ASCIIZ  /%N%ADA: %06%A REC'D: %06%A EXP'D: %06%N/
531 016241      045 022516 053501 FRMT14: .ASCIIZ /%N%AWORDS WRITTEN: %D3%A BUS ADDR: %06%N/
532 016312 047045 040445 047527 FRMT9: .ASCIIZ  /%N%AWORDS WRITTEN: %D3%A BUS ADDR: %06%A EXP'D: %06%A REC'D: %06%N/
533 016415      045 022516 051101 FRMT10: .ASCII  /%N%ARANGE %D3%A - %D3%A MILLISECONDS WAS %D6%N/
534 016474 040445 040515 044530 .ASCIIZ  /%AMAXIMUM TIMEOUT OF PROGRAM IS 3 SECONDS%N/
535 016550 047045 040445 051105 FRMT11: .ASCIIZ /%N%AE%ROR LIMIT EXCEEDED - DROPPED%N/
536 016615      045 042101 044522 FRMT98: .ASCII  /%ADRI%VE DID NOT RECOVER FROM POWER FAILURE/
537 016667      045 000116          FRMT99: .ASCIIZ  /%N/
538 016672 047045 052045 040445 FRMT13: .ASCIIZ  /%N%T%A - WILL NOT TEST%N/
539 016723      045 022516 050101 FRMT15: .ASCIIZ /%N%APATTERN WAS: %06/
540
541          .EVEN
542
546
547 016750          BGNMOD      ENDMOD
548 016750          HPTCODE
549
550 016750          BGNHW
(3) 016750 000005          .WORD      L10020-L$HW/2
551 016752 174400          .WORD      174400          ;CSR
552 016754 000160          .WORD      160          ;VECTOR
553 016756 000240          .WORD      240          ;PRIORITY
554 016760 000000          .WORD      0          ;DRIVE (BITS 8,9,10)
555 016762 000001          .WORD      1          ;RL11=1 RLV11=0
556
557 016764          ENDMHW
(3) 016764          L10020:

```

```

558
559 016764          ENDMOD
560
561 016764          BGNMOD  SPTCODE
562
563 016764          BGNSW
(3) 016764 000005      .WORD  L10021-L$SW/2
564
565 016766 000000      DROP:  .WORD  0
566 016770 000012      MERLMT: .WORD  10.
567 016772 000000      T.SIZE: .WORD  0
568 016774 000000      T.DMP:  .WORD  0
569 016776 000000      T.LMT:  .WORD  0
570
571 017000          ENDSW
(3) 017000          I10021:
572
573 017000          ENDMOD
574
575 017000          BGNMOD  DSPCODE
576
577 017000          DISPATCH  47
(4) 017000 000057      .WORD  47
(6) 017002 021632      .WORD  T1
(6) 017004 022102      .WORD  T2
(6) 017006 022246      .WORD  T3
(6) 017010 022376      .WORD  T4
(6) 017012 022532      .WORD  T5
(6) 017014 022664      .WORD  T6
(6) 017016 023022      .WORD  T7
(6) 017020 023220      .WORD  T8
(6) 017022 023522      .WORD  T9
(6) 017024 023712      .WORD  T10
(6) 017026 024110      .WORD  T11
(6) 017030 024262      .WORD  T12
(6) 017032 024460      .WORD  T13
(6) 017034 024660      .WORD  T14
(6) 017036 025030      .WORD  T15
(6) 017040 025132      .WORD  T16
(6) 017042 025256      .WORD  T17
(6) 017044 025452      .WORD  T18
(6) 017046 025606      .WORD  T19
(6) 017050 025740      .WORD  T20
(6) 017052 026060      .WORD  T21
(6) 017054 026240      .WORD  T22
(6) 017056 027016      .WORD  T23
(6) 017060 027212      .WORD  T24
(6) 017062 027356      .WORD  T25
(6) 017064 027472      .WORD  T26
(6) 017066 027650      .WORD  T27
(6) 017070 030250      .WORD  T28
(6) 017072 030672      .WORD  T29
(6) 017074 031320      .WORD  T30
(6) 017076 032000      .WORD  T31
(6) 017100 032432      .WORD  T32
(6) 017102 033042      .WORD  T33

```

```

(6) 017104 033274 .WORD T34
(6) 017106 033564 .WORD T35
(6) 017110 034060 .WORD T36
(6) 017112 034352 .WORD T37
(6) 017114 034744 .WORD T38
(6) 017116 035244 .WORD T39
(6) 017120 035604 .WORD T40
(6) 017122 036116 .WORD T41
(6) 017124 036402 .WORD T42
(6) 017126 036672 .WORD T43
(6) 017130 036762 .WORD T44
(6) 017132 037114 .WORD T45
(6) 017134 037312 .WORD T46
(6) 017136 037450 .WORD T47
578 017140 ENDMOD
579
580 .SBTTL INITIALIZATION CODE
581
582 017140 BGNMOD INITCODE
583
584 017140 BGNINIT
585
586 017140 SETPRI #PRI07
(3) 017140 012700 000340 MOV #PRI07,R0
(3) 017144 104041 EMT C$SPRI
587
588 017146 READEF #EF.PWR
(3) 017146 012700 000034 MOV #EF.PWR,R0
(3) 017152 104050 EMT C$REFG
589 017154 BNCOMPLETE NOPWR
(2) 017154 103004 BCC NOPWR
590 017156 013737 002012 0C2304 MOV L$UNIT,PWRFLG
591 017164 000473 BR CONT
592 017166 NOPWR: READEF #EF.RESTART
(3) 017166 012700 000037 MOV #EF.RESTART,R0
(3) 017172 104050 EMT C$REFG
593 017174 BCOMPLETE START1
(2) 017174 103404 BCS START1
594 017176 READEF #EF.START
(3) 017176 012700 000040 MOV #EF.START,R0
(3) 017202 104050 EMT C$REFG
595 017204 BNCOMPLETE CONTINUE
(2) 017204 103010 BCC CONTINUE
596 017206 012700 0023*4 START1: MOV #ERCOUNT,R0
597 017212 012701 000100 MOV #64,R1
598 017216 005020 1$: CLR (R0)+
599 017220 005301 DEC R1
600 017222 001375 BNE 1$
601
602 017224 000407 BR START
603
604 017226 CONTINUE: READEF #EF.CONTINUE
(3) 017226 012700 000036 MOV #EF.CONTINUE,R0
(3) 017232 104050 EMT C$REFG
605 017234 BCOMPLETE CONT
(2) 017234 103447 BCS CONT

```

```

606 017236 005737 002136      NXT:  TST      UUT          ;DONE WITH ALL UNITS
607 017242 001011              BNE      XXX          ;NO
608 017244 012737 177777 002140 START:  MOV      #-1,UNITST
609 017252 013737 002012 002136      MOV      L$UNIT,UUT
610 017260 012737 002312 002312      MOV      #ERCOUNT-2,ERPOINT
611
612 017266 005237 002140      XXX:    INC      UNITST
613 017272 062737 000002 002312      ADD      #2,ERPOINT
614 017300 005337 002136      DEC      UUT
615 017304      REST:  GPHARD  UNITST,RO
(3) 017304 013700 002140      MOV      UNITST,RO
(3) 017310 104042      EMT      C$GPHRD
616 017312      BCOMPLETE 2$
(2) 017312 103406      BCS      2$
617 017314 005737 002304      TST      PWRFLG
618 017320 001746      BEQ      NXT
619 017322 005337 002304      DEC      PWRFLG
620 017326 000743      BR       NXT
621 017330 012037 002252      2$:    MOV      (RO)+,BCSR          ;GET BUS ADDRESS
622 017334 012037 002254      MOV      (RO)+,BVEC          ;GET VECTOR
623 017340 012037 002256      MOV      (RO)+,BPRIOR          ;GET PRIORITY
624 017344 012037 002134      MOV      (RO)+,DRIVE          ;GET DRIVE
625 017350 012037 002306      MOV      (RO)+,T.CNTRL          ;GET CONTROLLER TYPE
626
627 017354 013700 002252      (CONT:  MOV      BCSR,RO          ;CREATE REGISTERS
628 017360 010037 002242      MOV      RO,RLCS
629 017364 062700 000002      ADD      #2,RO
630 017370 010037 002244      MOV      RO,RLBA
631 017374 062700 000002      ADD      #2,RO
632 017400 010037 002246      MOV      RO,RLDA
633 017404 062700 000002      ADD      #2,RO
634 017410 010037 002250      MOV      RO,RLMP
635
636 017414 005737 002304      TST      PWRFLG
637 017420 001064      BNE      5$
638 017422 005737 016772      TST      T.SIZE          ;DO WE WANT TO CHECK UNITS??
639 017426 001461      BEQ      5$          ;NO
640 017430 005037 002142      CLR      TRPFLG          ;CLEAR OUT TRAP INDICATOR
641 017434      SETVEC  ERRVEC,#TRPHAN,#340 ;SETUP TO CATCH TIMEOUT
(7) 017434 012746 000340      MOV      #340,-(SP)
(6) 017440 012746 021350      MOV      #TRPHAN,-(SP)
(5) 017444 013746 002132      MOV      ERRVEC,-(SP)
(4) 017450 012746 000003      MOV      #3,-(SP)
(3) 017454 104037      EMT      C$SVEC
(2) 017456 062706 000010      ADD      #10,SP
642 017462 005777 162554      TST      @RLCS          ;ACCESS CONTROLLER
643 017466      CLRVEC  ERRVEC
(3) 017466 013700 002132      MOV      ERRVEC,RO
(3) 017472 104036      EMT      C$CVEC
644 017474 005737 002142      TST      TRPFLG          ;DID TRAP OCCUR??
645 017500 001404      BEQ      7$          ;NO, CHECK DRIVE
646 017502 012737 007052 002126      MOV      #NORES,WHY
647 017510 000415      BR       8$
648
649 017512 012777 000200 162522 7$:    MOV      #200,@RLCS          ;NOW CHECK DRIVE FOR READY
650 017520 053777 002134 162514      BIS      DRIVE,@RLCS
  
```

```
651 017526 032777 000001 162506      BIT      #1, @RLCS
652 017534 001016                      BNE      5$
653 017536 012737 007070 002126      MOV      #NORDY, WHY
654 017544                      PRINTB  #FRMT13, WHY
(8) 017544 013746 002126      MOV      WHY, -(SP)
(7) 017550 012746 016672      MOV      #FRMT13, -(SP)
(6) 017554 012746 000002      MOV      #2, -(SP)
(3) 017560 010600      MOV      SP, R0
(4) 017562 104014      EMT      C$PNTB
(4) 017564 062706 000006      ADD      #6, SP
655 017570 000444      BR       6$
656
657 017572 005737 002304      5$:      TST      PWRFLG           ;POWER UP
658 017576 001451                      BEQ      END             ;NO
659 017600 012777 000200 162434      MOV      #200, @RLCS
660 017606 053777 002134 162426      BIS      DRIVE, @RLCS
661 017614 012701 000074      MOV      #60, R1
662 017620                      3$:      WAITMS  #10.
(3) 017620 012700 000012      MOV      #10, R0
(3) 017624 104026      EMT      C$WTM
663 017626 032777 000001 162406      BIT      #1, @RLCS
664 017634 001032                      BNE      END
665 017636 005301      DEC      R1
666 017640 001367                      BNE      3$
667
668 017642                      PRINTF  #FRMT99
(7) 017642 012746 016667      MOV      #FRMT99, -(SP)
(6) 017646 012746 000001      MOV      #1, -(SP)
(3) 017652 010600      MOV      SP, R0
(4) 017654 104017      EMT      C$PNTF
(4) 017656 062706 000004      ADD      #4, SP
669 017662                      PRINTF  #FRMT98
(7) 017662 012746 016615      MOV      #FRMT98, -(SP)
(6) 017666 012746 000001      MOV      #1, -(SP)
(3) 017672 010600      MOV      SP, R0
(4) 017674 104017      EMT      C$PNTF
(4) 017676 062706 000004      ADD      #4, SP
670 017702 004737 015256      6$:      JSR      PC, LINE1
671 017706                      DODU    UNITST
(3) 017706 013700 002140      MOV      UNITST, R0
(3) 017712 104053      EMT      C$DODU
672 017714                      DOCLN
(3) 017714 104044      EMT      C$DCLN
673 017716 000137 017236      JMP      NXT
674
675 017722 013737 002276 002300      END:    MOV      UOPIMN, OPIMN
676 017730 013737 002274 002302      MOV      UOPIMX, OPIMX
677 017736 005737 002306      TST      T, CNTRLR
678 017742 001006                      BNE      1$
679 017744 013737 002272 002300      MOV      LOPIMN, OPIMN
680 017752 013737 002270 002302      MOV      LOPIMX, OPIMX
681 017760                      1$:
682 017760                      SETVEC  BVEC, #INTSRV, #340
(7) 017760 012746 000340      MOV      #340, -(SP)
(6) 017764 012746 020110      MOV      #INTSRV, -(SP)
(5) 017770 013746 002254      MOV      BVEC, -(SP)
```

RL1??  
YES. THEN KEEP LIMITS SET

```

(4) 017774 012746 000003      MOV    #3,-(SP)
(3) 020000 104037              EMT    C$SVEC
(2) 020002 062706 000010      ADD    #10,SP
683
684
685
686 020006                      L10022: ENDINIT
(3) 020006                      EMT    C$INIT
(3) 020006 104011
687
688 020010                      ENDMOD
689
690 020010                      BGNMOD  CLNCODE
691
692 020010                      BGNCLN
693
694
695 020010                      SETVEC  ERRVEC,#TRPHAN,#340
(7) 020010 012746 000340      MOV    #340,-(SP)
(6) 020014 012746 021350      MOV    #TRPHAN,-(SP)
(5) 020020 013746 002132      MOV    ERRVEC,-(SP)
(4) 020024 012746 000003      MOV    #3,-(SP)
(3) 020030 104037              EMT    C$SVEC
(2) 020032 062706 000010      ADD    #10,SP
696 020036 032777 000200 162176 1$: BIT    #CRDY,@RLCS
697 020044 001774              BEQ    1$
698
699 020046 042777 000100 162166      BIC    #INTEN,@RLCS
700
701 020054                      CLRVEC  RVEC
(3) 020054 013700 002254      MOV    BVEC,R0
(3) 020060 104036              EMT    C$CVEC
702 020062 005737 002304      TST    PWRFLG
703 020066 001402              BEQ    2$
704 020070 005337 002304      DEC    PWRFLG
705 020074                      2$: CLRVEC  ERRVEC
(3) 020074 013700 002132      MOV    ERRVEC,R0
(3) 020100 104036              EMT    C$CVEC
706
707
708
709 020102                      L10023: ENDCLN
(3) 020102                      EMT    C$CLEAN
(3) 020102 104012
710
711 020104                      ENDMOD
712
713 020104                      BGNMOD  DRPCODE
714
715 020104                      BGNCLN
716
717 020104 000240              NOP
718
719 020106                      10024: ENDDU
(3) 020106                      EMT    C$DU
(3) 020106 104055

```

```

720
721 020110      ENDMOD
722
723      .SBTTL  GLOBAL SUBROUTINES
724
725 020110      BGNMOD  GLBSUB
726
727 020110      BGNSRV
728 020110 005237 002144  INTSRV: INC      INTFLG      ;SET INTERRUPT OCCURANCE FLAG
729
730 020114      ENDSRV
731 (3) 020114
732 (2) 020114 000002  L10025:
733      RTI
734
735      ;ROUTINE USED IN TIMING OPI
736 020116 005237 002144  TMSRV: INC      INTFLG
737 020122      ABORTWAIT
738 (3) 020122 104021  EMT      C$4BRT
739 020124 000002      RTI
740
741 020126 000240  CKERLT: NOP
742 020130      INLOOP
743 (3) 020130 104020  EMT      C$INLP
744 020132      BCOMPLETE 99$
745 (2) 020132 103427  BCS      99$
746
747 020134 005737 016766  TST      DROP
748 020140 001424      BEQ      99$
749 020142 005277 162144  INC      @ERPOINT
750 020146 027737 162140 016770  CMP      @ERPOINT,MERLMT
751 020154 002416      BLT      99$
752
753 020156      PRINTF #FRMT11
754 (7) 020156 012746 016550  MOV      #FRMT11,-(SP)
755 (6) 020162 012746 000001  MOV      #1,-(SP)
756 (3) 020166 010600      MOV      SP,R0
757 (4) 020170 104017      EMT      C$PNTF
758 (4) 020172 062706 000004  ADD      #4,SP
759 020176 004737 015256  JSR      PC,LINE1
760 020202      DODU
761 (3) 020202 013700 002140  UNITST ;DROP THIS UNIT
762 (3) 020206 104053      MOV      UNITST,R0
763 020210      EMT      C$DODU
764 (3) 020210 104044      DOCLN
765      EMT      C$DCLN
766
767 020212      99$:
768 020212 000205      RTS      R5
769
770      .SBTTL  ROUTINE TO CHECK FOR CONTROLLER ERRORS
771
772      ;*****
773      ;*THIS ROUTINE WILL CHECK RLCS FOR ERRORS AND PRINT THEM
774      ;*ACCORDINGLY. IT WILL MERGE THE ERROR PRINTOUT WITH THE TEST
775      ;*ERROR MESSAGE.
  
```

```
763
764
765
766
767
768
769
770
771 020214 005037 002124 CHERR: CLR T.CRC
772 020220 032737 176000 002226 BIT #176000,E.CS ;ANY ERROR BITS SET?
773 020226 001001 BNE 2$ ;YES,FIND OUT WHICH
774 020230 000205 RTS R5 ;NO EXIT
775 020232 012701 010417 2$: MOV #EM100,R1 ;GET START OF STRING
776 020236 005737 002226 TST E.CS ;IS COMPOSITE ERROR SET?(BETTER BE)
777 020242 100003 BPL 99$ ;IT'S NOT SOMETHING IS WRONG
778 020244 004537 020752 JSR R5,FIX ;YES, PUT 'COMP' IN STRING
779 020250 007316 COMP ;'COMP'
780 020252 032737 040000 002226 99$: BIT #DERR,E.CS ;DRIVE ERROR SET?
781 020260 001405 BEQ 3$ ;NO, CONTINUE
782 020262 005237 002310 INC DERFLG
783 020266 004537 020752 JSR R5,FIX ;YES, PUT 'DRV' INTO STRING
784 020272 007245 DEMES ;'DRV'
785 020274 032737 020000 002226 3$: BIT #NXM,E.CS ;NON-EXISTENT MEMORY ERROR?
786 020302 001403 BEQ 4$ ;NO, CONTINUE
787 020304 004537 020752 JSR R5,FIX ;YES, PUT 'NXM' INTO STRING
788 020310 007252 NXMMES ;'NXM'
789 020312 032737 002000 002226 4$: BIT #OPI,E.CS ;IS OPI SET?
790 020320 001422 BEQ 6$ ;NO, GO CHECK BITS 11 & 12
791 020322 004537 020752 JSR R5,FIX ;PUT 'OPI' INTO STRING
792 020326 007257 OPIMES ;'OPI'
793 020330 032737 004000 002226 BIT #BIT11,E.CS ;HEADERCRC ERROR?
794 020336 001403 BEQ 5$ ;NO, GO CHECK HEADER NOT FOUND
795 020340 004537 020752 JSR R5,FIX ;GO PUT 'HCRC' IN STRING
796 020344 007264 HCRCMES ;'HCRC'
797 020346 032737 010000 002226 5$: BIT #BIT12,E.CS ;HEADER NOT FOUND?
798 020354 001424 BEQ 8$ ;NO, GO PUT 'CRLF' IN STRING
799 020356 004537 020752 JSR R5,FIX ;PUT 'HNF' IN STRING
800 020362 007272 HNFMES ;'HNF'
801 020364 000420 BR 8$ ;PUT 'CRLF' IN STRING
802 020366 032737 004000 002226 6$: BIT #BIT11,E.CS ;DATA CRC ERROR?
803 020374 001405 BEQ 7$ ;NO, GO CHECK DATA LATE
804 020376 005237 002124 INC T.CRC
805 020402 004537 020752 JSR R5,FIX ;PUT 'DCK' IN STRING
806 020406 007277 DCKMES ;'DCK'
807 020410 032737 010000 002226 7$: BIT #BIT12,E.CS ;DATA LATE ERROR?
808 020416 001403 BEQ 8$ ;NO, GO PUT IN 'CRLF'
809 020420 004537 020752 JSR R5,FIX ;PUT 'DLT' IN STRING
810 020424 007304 DLTMES ;'DLT'
811 020426 004537 020752 8$: JSR R5,FIX ;PUT 'CRLF' INTO STRING
812 020432 007313 MSCRLF ;'CRLF'
813 020434 004537 020752 JSR R5,FIX ;MOVE HEADER
814 020440 000000 RESTMS: .WORD 0 ;HEADER FROM TEST
815 020442 105011 CLR R1 ;PUT TERMINATOR IN
816 020444 ERRDF 300,LF,ERR6
(3) 020444 104462 TRAP T.CRCODE
(5) 020446 000454 .WORD 300
```

```

(5) 020450 007311          .WORD  LF
(5) 020452 014514          .WORD  ERR6
817 020454 000205          R'S    R5          ;EXIT ROUTINE
818
819          .SBITL LOAD RLCS
820          *****
821          * ROUTINE TO LOAD RLCS WITH FUNCTION TO BE PERFORMED
822          * CALL: JSR R5,LDFUNC          ;LOAD THE FUNCTION IN NEXT WORD
823          *          .WORD          ;BITS TO BE LOADED, FUNCTION
824          *          ;AND INTR ENABLE ONLY
825          *
826          *
827          *
828 020456 032777 040000 161556 LDFUNC: BIT #BIT14,@RLCS ;DRIVE ERROR SET
829 020464 001426          BEQ 5$
830 020466 017737 161554 002222 MOV @RLDA,B.DA
831 020474 012777 000013 161544 MOV #13,@RLDA
832 020502 012737 000200 002216 MOV #200,B.CS
833 020510 053737 002134 002216 BIS DRIVE,B.CS
834 020516 013777 002216 161516 MOV B.CS,@RLCS
835 020524 032777 000200 161510 6$: BIT #200,@RLCS
836 020532 001774          BEQ 6$
837 020534 013777 002222 161504 MOV B.DA,@RLDA
838 020542 012537 002146 5$: MOV (R5)+,LDCSR ;GET BITS TO LOAD
839 020546 010346          MOV R3,-(SP) ;SAVE R3
840 020550 042737 177661 002146 BIC #177661,LDCSR ;CLEAR ALL BUT FUNC & INTR EN
841 020556 013737 002146 002260 MOV LDCSR,FNDFNC ;SAVE FUNCTION
842 020564 042737 000100 002260 BIC #INTEN,FNDFNC ;ONLY FUNCTION
843 020572 012703 020712          MOV #HDRLST,R3 ;GET HEADER LIST
844 020576 006237 002260          ASR FNDFNC ;ALIGN TO LEFT
845 020602 001404          BEQ 2$ ;IF EQUAL TO ZERO, SET R3
846 020604 022323          CMP (R3)+,(R3)+ ;BUMP R3 BY 4
847 020606 005337 002260 1$: DEC FNDFNC ;DEC FUNCTION
848 020612 001374          BNE 1$ ;FOUND IT? NO-GO BACK
849 020614 032737 000100 002146 2$: BIT #INTEN,LDCSR ;YES, DO WE WANT FLAG OR INTR?
850 020622 001401          BEQ 3$ ;FLAG BRANCH
851 020624 005723          TST (R3)+ ;INTR POINT TO THAT ONE
852 020626 011303          MOV (R3),R3 ;SET HEADER
853 020630 010337 020440          MOV R3,RESTMS ;SET UP HEADER
854 020634 010337 002264          MOV R3,TRYFNC ;SAVE HEADER FOR LATER
855 020640 053737 002262 002146 BIS XMEM,LDCSR ;LOAD E.A. BITS
856 020646 005037 002262          CLR XMEM ;CLEAR OUT THE BITS
857 020652 053737 002134 002146 BIS DRIVE,LDCSR ;SELECT DRIVE
858 020660 052737 000200 002146 BIS #200,LDCSR
859 020666 013777 002146 161346 MOV LDCSR,@RLCS ;LOAD FUNCTION
860 020674 004537 020764          JSR R5,BEFORE ;READ REGISTERS
861 020700 042777 000200 161334 4$: BIC #200,@RLCS ;ISSUE COMMAND
862 020706 012603          MOV (SP)+,R3 ;RESTORE R3
863 020710 000205          RTS R5 ;EXIT
864
865
866
867 020712 007377          HDRLST: NOPMES
868 020714 007430          NOPINT
869 020716 007462          WCKMES
870 020720 007522          WCKINT

```

```

871 020722 007747          GSTMES
872 020724 010006          GSTINT
873 020726 007664          SEKMES
874 020730 007715          SEKINT
875 020732 007563          RHDMES
876 020734 007623          RHDINT
877 020736 010127          WRTMES
878 020740 010161          WRTINT
879 020742 010045          RDDMES
880 020744 010076          RDDINT
881 020746 010213          RDNMES
882 020750 010247          RDNINT
883
884 *****
885 :ROUTINE TO MOVE ASCII STRINGS
886 :USES REGISTERS R1 - WHERE STRING IS BEING BUILT
887 :
888 :      CALL      JSR      R5, FIX
889 :      .WORD
890 :      :ADDRESS OF STRING TO MOVE
891 020752 012504          FIX:  MOV      (R5)+, R4      :GET ADDRESS AND MOVE RETURN
892 020754 112421          1$:   MOVB     (R4)+, (R1)+    :GET BYTE AND UPDATE
893 020756 001376          BNE      1$                  :WATCH 0 BYTE TERMINATOR
894 020760 105741          TSTB     -(R1)                :BACK UP OVER ZERO BYTE
895 020762 000205          RTS      R5                  :EXIT
896
897
898 :ROUTINE TO READ REGISTERS PRIOR TO OPERATION
899 :CALL: JSR R5,BEFORE
900
901 020764 017737 161252 002216 BEFORE: MOV      @RLCS,B.CS      :READ CS
902 020772 017737 161246 002220      MOV      @RLBA,B.BA      :BA
903 021000 017737 161242 002222      MOV      @RLDA,B.DA      :DA
904 021006 017737 161236 002224      MOV      @RLMP,B.MP      :MP
905 021014 000205          RTS      R5
906
907 :ROUTINE TO READ REGISTERS AT TIME OF ERROR
908 :CALL: JSR R5,AFTER
909
910 021016 017737 161220 002226 AFTER:  MOV      @RLCS,E.CS      :READ CS
911 021024 017737 161214 002230      MOV      @RLBA,E.BA      :BA
912 021032 017737 161210 002232      MOV      @RLDA,E.DA      :DA
913 021040 017737 161204 002234      MOV      @RLMP,E.MP      :MP
914 021046 017737 161176 002236      MOV      @RLMP,E.MP1     :MP
915 021054 017737 161170 002240      MOV      @RLMP,E.MP2     :MP
916 021062 000205          RTS      R5
917
918
919 021064 010046          SIMBCC: MOV      R0,-(SP)      :SAVE R0
920 021066 010146          MOV      R1,-(SP)      :SAVE R1
921 021070 010246          MOV      R2,-(SP)      :SAVE R2
922 021072 012537 002172          MOV      (R5)+,TEMP2    :GET NUMBER OF BITS
923 021076 012537 002174          MOV      (R5)+,TEMP3    :GET DATA FOR CRC CALCULATION
924 021102 012537 002176          MOV      (R5)+,TEMP4    :GET STARTING CRC
925 021106 005037 002154          1$:   CLR      B0,FBK
926 021112 013700 002176          MOV      TEMP4,R0      :GET PRESENT R

```

```

927 021116 006037 002174      ROR    TEMP3      ;ROTATE NEW DATA
928 021122 005500      ADC     R0          ;MERGE NEW WITH OLD
929 021124 032700 000001      BIT     #1,R0      ;BIT 0 SET
930 021130 001402      BEQ     2$          ;IF NOT CONTINUE
931 021132 005137 002154      COM     BCCFBK      ;
932 021136 013700 002152      2$: MOV     XPOLY,R0      ;GET CRC POLYNOMIAL (CRC-16)
933 021142 005100      COM     R0          ;COMPLIMENT POLYNOMIAL
934 021144 040037 002154      BIC     R0,BCCFBK
935 021150 000241      CLC              ;CLEAR CARRY
936 021152 006037 002176      ROR     TEMP4
937 021156 013700 002154      MOV     BCCFBK,R0
938 021162 013701 002176      MOV     TEMP4,R1
939 021166 010102      MOV     R1,R2
940 021170 040100      BIC     R1,R0
941 021172 043702 002154      BIC     BCCFBK,R2
942 021176 050200      BIS     R2,R0
943 021200 043737 002152 002176 BIC     XPOLY,TEMP4
944 021206 050037 002176      BIS     R0,TEMP4
945 021212 005337 002172      DEC     TEMP2
946 021216 001333      BNE     1$
947
948 021220 013737 002176 002156 MOV     TEMP4,CALBCC
949 021226 012602      MOV     (SP)+,R2
950 021230 012601      MOV     (SP)+,R1
951 021232 012600      MOV     (SP)+,R0
952 021234 000205      RTS     R5          ;RETURN
953
954
955      ;ROUTINE TO WAIT FOR DRIVE READY
956
957
958
959
960 021236 012701 000144      WTD RDY: MOV     #100.,R1
961 021242 032777 000001 160772 1$: BIT     #DRDY,@RLCS
962 021250 001011      BNE     2$
963
964 021252      WAITUS  #20.
(3) 021252 012700 000024      MOV     #20.,R0
(3) 021256 104027      EMT     C$WTU
965 021260 005301      DEC     R1
966 021262 001367      BNE     1$
967
968 021264      ERRDF  200.,DRTIM,ERR5
(3) 021264 104462      TRAP    T$ERCODE
(5) 021266 000310      .WORD   200
(5) 021270 007217      .WORD   DRTIM
(5) 021272 014456      .WORD   ERR5
969
970 021274 000205      2$:  RTS     R5
971
972      ;ROUTINE TO WAIT FOR CONTROLLER
973
974 021276 012701 000620      WTC RDY: MOV     #400.,R1
975 021302 032777 000200 160732 1$: BIT     #CDY,@RLCS
976 021310 001014      BNF     2$

```

```

977
978 021312          WAITJS #20.
(3) 021312 012700 000024  MOV #20.,R0
(3) 021316 104027  EMT C$WTU
979 021320 005301  DEC R1
980 021322 001367  BNE 1$
981 021324 004537 021016  JSR R5,AFTER
982
983 021330          ERRDF 100.,CRTIM,ERR5
(3) 021330 104462  TRAP T$ERCODE
(5) 021332 000144  .WORD 100
(5) 021334 007172  .WORD CRTIM
(5) 021336 014456  .WORD ERR5
984 021340 000205  RTS R5
985
986 021342 004537 021016  2$: JSR R5,AFTER
987 021346 000205  RTS R5
988
989
990 021350 005237 002142  TRPHAN: INC TRPFLG
991 021354 000002  RTI
992
993 021356          HDHOME:
994
995 021356          BGNSEG          ;%%START OF SEGMENT%%
(3) 021356 104004  EMT C$BSEG
996          ;ISSUE DRIVE RESET
997
998 021360 012737 000001 002266  MOV #1,ERFLG          ;SET ERROR FLAG
999 021366 012777 000013 160652  MOV #DRST!MK.GSBIT,0RLDA
1000 021374 004537 020456  JSR R5,LDFUNC          ;LOAD THE FUNCTION IN NEXT WORD
1001 021400 000004  GSTAT
1002 021402 004537 021276  JSR R5,WTCRDY
1003 021406          ESCAPE SEG          ;CHECK FOR FL:LOE, ELSE EXIT SEG
(3) 021406 104010  EMT C$ESCAPE
(3) 021410 000216  .WORD 10000$-.
1004 021412 004537 020214  JSR R5,CHERR          ;CHECK CNTLR FOR ERRORS
1005 021416          ESCAPE SEG          ;CHECK FOR FL:LOE, ELSE EXIT SEG
(3) 021416 104010  EMT C$ESCAPE
(3) 021420 000206  .WORD 10000$-.
1006
1007
1008 021422 004537 020456  JSR R5,LDFUNC          ;LOAD THE FUNCTION IN NEXT WORD
1009 021426 000010  RDHDR
1010 021430          ESCAPE SEG          ;CHECK FOR FL:LOE, ELSE EXIT SEG
(3) 021430 104010  EMT C$ESCAPE
(3) 021432 000174  .WORD 10000$-.
1011 021434 004537 021276  JSR R5,WTCRDY
1012 021440          ESCAPE SEG          ;CHECK FOR FL:LOE, ELSE EXIT SEG
(3) 021440 104010  EMT C$ESCAPE
(3) 021442 000164  .WORD 10000$-.
1013
1014 021444 004537 020214  JSR R5,CHERR          ;CHECK CNTLR FOR ERRORS
1015 021450          ESCAPE SEG          ;CHECK FOR FL:LOE, ELSE EXIT SEG
(3) 021450 104010  EMT C$ESCAPE
(3) 021452 000154  .WORD 10000$-.

```

```

1016
1017 021454 013737 002234 002160      MOV      E,MP,TMPO      ;GET HEADER
1018 021462 042737 000077 002160      BIC      #77,TMPO
1019 021470 001424                BEQ      99$           ;SEEK IS NOT NECESSARY
1020 021472 042737 000100 002160      BIC      #100,TMPO
1021 021500 012777 000001 160540      MOV      #MK,@RLDA      ;SET TO SEEK
1022 021506 053777 002160 160532      BIS      TMPO,@RLDA      ;SET IN DIFFERENCE
1023
1024 021514 004537 020456                JSR      R5,LDFUNC      ;LOAD THE FUNCTION IN NEXT WORD
1025 021520 000006                SEEK
1026 021522 004537 021276                JSR      R5,WTCRDY
1027 021526                ESCAPE SEG      ;CHECK FOR FL:LOE, ELSE EXIT SEG
(3) 021526 104010                EMT      C$ESCAPE
(3) 021530 000076                .WORD    10000$-.
1028
1029 021532 004537 020214                JSR      R5,CHERR      ;CHECK CNTLR FOR ERRORS
1030 021536                ESCAPE SEG      ;CHECK FOR FL:LOE, ELSE EXIT SEG
(3) 021536 104010                EMT      C$ESCAPE
(3) 021540 000066                .WORD    10000$-.
1031
1032 021542 004537 020456      99$:      JSR      R5,LDFUNC      ;LOAD THE FUNCTION IN NEXT WORD
1033 021546 000010                RDHDR
1034 021550 004537 021276                JSR      R5,WTCRDY
1035 021554                ESCAPE SEG      ;CHECK FOR FL:LOE, ELSE EXIT SEG
(3) 021554 104010                EMT      C$ESCAPE
(3) 021556 000050                .WORD    10000$-.
1036 021560 004537 020214                JSR      R5,CHERR
1037 021564                ESCAPE SEG
(3) 021564 104010                EMT      C$ESCAPE
(3) 021566 000040                .WORD    10000$-.
1038
1039 021570 013737 002234 002160      MOV      E,MP,TMPO      ;GET HEADER
1040 021576 043737 002150 002160      BIC      SECMSK,TMPO      ;IGNORE SECTOR
1041 021604 001404                BEQ      1$           ;ON ZERO
1042
1043 021606                ERRDF 400.,SKHOME,ERRO      ;CAN'T SEEK TO TRACK 0
(3) 021606 104462                TRAP   T$ERCODE
(5) 021610 000620                .WORD    400
(5) 021612 010303                .WORD    SKHOME
(5) 021614 014244                .WORD    ERRO
1044
1045 021616                1$:      ESCAPE SEG      ;CHECK FOR FL:LOE, ELSE EXIT SEG
(3) 021616 104010                EMT      C$ESCAPE
(3) 021620 000006                .WORD    10000$-.
1046
1047 021622 005037 002266                CLR      ERFLG      ;INDICATE SUCCESS BACK TO MAIN PROGRAM
1048
1049
1050 021626                10000$:      ENDSEG      ;%%END OF SEGMENT%%
(3) 021626                EMT      C$ESEG
(3) 021626 104005
1051 021630 000207                RTS      PC
1052
1053
1054 021632                ENDMOD
1055

```

```

1056      .SBTTL  **TEST 1** - WRITE NPR INTEGRITY
1057
1058      021632      BGNSTST                      ;**START OF TEST**
1059
1060      021632      STARS
1061      (2)          ;:*****
1062      1061          ;:CHECK THAT NPR WILL NOT INTERFERE WITH THE OPERATION OF THE
1063      021632      ;:UNIBUS. WE SET UP LOCATION 4 TO HANDLE THE TRAP IF IT HAPPENS.
1064      (2)          ;:*****
1065
1066      021632      004737      021356      JSR      PC,HDHOME      ;HEADS OVER TRACK 0
1067      021636      CKERFG      ;HEADS GO HOME OKAY
1068      (4) 021644      104032      EMT      C$EXIT
1069      (4) 021646      000232      .WORD    L10026-.
1070
1071      021650      BGNSEG      ;%%START OF SEGMENT%%
1072      (3) 021650      104004      EMT      C$BSEG
1073
1074      021652      1$: SETVEC      ERRVEC,#TRPHAN,#340      ;SET UP FOR TRAP
1075      (7) 021652      012746      000340      MOV      #340,-(SP)
1076      (6) 021656      012746      021350      MOV      #TRPHAN,-(SP)
1077      (5) 021662      013746      002132      MOV      ERRVEC,-(SP)
1078      (4) 021666      012746      000003      MOV      #3,-(SP)
1079      (3) 021672      104037      EMT      C$SVEC
1080      (2) 021674      062706      000010      ADD      #10,SP
1081      021700      005037      002142      CLR      TRPFLG      ;CLEAR TRAP OCCURANCE
1082      021704      012777      003052      160332      MOV      #BUF,@RLBA      ;BUS ADDRESS
1083      021712      005077      160330      CLR      @RLDA      ;LOAD DISK ADDRESS
1084      021716      012777      177777      160324      MOV      #-1,@RLMP      ;WORD COUNT OF 1
1085      021724      005037      002166      CLR      GDDAT      ;SET UP CSR TO LOAD
1086      021730      013737      002134      002166      MOV      DRIVE,GDDAT      ;SET IN DRIVE
1087      021736      052737      000012      002166      BIS      #WRITE,GDDAT      ;SET IN FUNCTION
1088      021744      004537      020764      JSR      R5,BEFORE      ;LOAD FOR ERROR PRINTOUT
1089      021750      013737      002166      002216      MOV      GDDAT,B.CS      ;SET IN COMMAND
1090      021756      052737      000201      002216      BIS      #201,B.CS      ;LOAD CRDY
1091      021764      042737      002000      002216      BIC      #OPI,B.CS      ;CLEAR (BIT 10)
1092      021772      013777      002166      160242      MOV      GDDAT,@RLCS      ;ISSUE WRITE
1093      022000      012701      000144      MOV      #100.,R1      ;WAIT FOR CRDY
1094      022004      032777      000200      160230      5$: BIT      #CRDY,@RLCS      ;NPR DONE
1095      022012      001013      BNE      6$      ;YES, 6$
1096      022014      WAITUS      #20.      ;WAIT A WHILE
1097      (3) 022014      012700      000024      MOV      #20.,R0
1098      (3) 022020      104027      EMT      C$WTU
1099      022022      005301      DEC      R1      ;A WHILE UP
1100      022024      001367      BNE      5$      ;NO, GO BACK
1101
1102      022026      004537      021016      JSR      R5,AFTER
1103      022032      ERRDF      0.,CRTIM,ERR5      ;CONTROLLER TIMED OUT
1104      (3) 022032      104462      TRAP      T$ERCODE
1105      (5) 022034      000000      .WORD    0
1106      (5) 022036      007172      .WORD    CRTIM
1107      (5) 022040      014456      .WORD    ERR5
1108
1109      022042      6$: CLRVEC      ERRVEC      ;CLEAR VECTOR
1110      (3) 022042      013700      002132      MOV      ERRVEC,R0

```

```

(3) 022046 104036 EMT C$CVEC
1094 022050 ESCAPE SEG ;CHECK FOR FL:LOE, ELSE EXIT SEG
(3) 022050 104010 EMT C$ESCAPE
(3) 022052 000024 .WORD 10001$-.

1095
1096 022054 005737 002142 TST TRPFLG ;DID TRAP OCCUR?
1097 022060 001406 BEQ 7$ ;NO
1098 022062 004537 021016 JSR R5,AFTER
1099 022066 ERRSF 1,EM51,ERRO ;TRAP ON WRITE
(3) 022066 104461 TRAP T$ERCODE
(5) 022070 000001 .WORD 1
(5) 022072 013145 .WORD EM51
(5) 022074 014244 .WORD ERRO
1100 022076 7$:

1101
1102
1103 022076 ENDSEG ;%%END OF SEGMENT%%
(3) 022076 10001$:
(3) 022076 104005 EMT C$FSEG

1104
1105 022100 ENDTST ;**END OF TEST**
(3) 022100 L10026:
(3) 022100 104001 EMT C$ETST

1106
1107 .SBITL **TEST 2** - WRITE FUNCTION
1108
1109 022102 BGNST ;**START OF TEST**
1110
1111
1112
1113 022102 STARS
(2) ;*****
1114 ;CHECK OF WRITE LOGIC UNDER FLAG MODE, WE WILL FIRST ISSUE A
1115 ;READ HEADER SO THAT WE DON'T WRITE ON THE BAD SECTOR
1116 ;FILE TRACK. WE WILL WRITE A FULL SECTOR (128 WORDS) FROM
1117 ;MEMORY (BUF). WE CHECK THAT NO ERRORS OCCUR. IF WE
1118 ;HAVE A DRIVE ERROR WE WILL DO A 'GET STATUS' TO SEE
1119 ;IF WRITE PROTECT IS SET IF IT IS WE WILL ABORT THE
1120 ;TEST. AN ERROR ON THE WRITE WILL LOOP ON JUST THE
1121 ;WRITE PORTION. LOOP ON TEST WILL READ HEADER, SEEK (IF
1122 ;NECESSARY) AND WRITE.
1123 022102 STARS
(2) ;*****
1124
1125
1126 022102 004737 021356 JSR PC,HDHOME ;HEADS OVER TRACK 0
1127 022106 CKERFG ;HEADS GO HOME OKAY
(4) 022114 104032 EMT C$EXIT
(4) 022116 000126 .WORD L10027-.

1128
1129 022120 BGNSEG ;%%START OF SEGMENT%%
(3) 022120 104004 EMT C$BSEG

1130
1131 022122 3$:
1132 022122 005077 160120 CLR @R,DA ;SET DISK ADDRESS
1133 022126 012777 177600 160114 MOV #-128,@RLMP ;WORD COUNT
  
```

```

1134 022134 012777 003052 160102      MOV    #BUF,@RLBA      ;BUS ADDRESS
1135 022142 004537 020456              JSR    R5,LDFUNC      ;LOAD THE FUNCTION IN NEXT WORD
1136 022146 000012                      WRITE                ;WRITE
1137
1138 022150 004537 021276              JSR    R5,WTCRDY      ;WAIT FOR CONTROLLER READY
1139 022154                      ESCAPE  SEG                ;CHECK FOR FL:LOE, ELSE EXIT SEG
(3) 022154 104010                      EMT    C$ESCAPE
(3) 022156 000064                      .WORD  10000$-.
1140
1141
1142 022160 032777 040000 160054      BIT    #DERR,@RLCS    ;DRIVE ERROR SET?
1143 022166 001425                      BEQ    4$              ;BRANCH IF NOT
1144
1145 022170 012777 000003 160050      MOV    #MK.GSBIT,@RLDA ;SET GET STATUS OF DRIVE
1146 022176 004537 020456              JSR    R5,LDFUNC      ;LOAD THE FUNCTION IN NEXT WORD
1147 022202 000004                      GSTAT                ;GET STATUS
1148 022204 004537 021276              JSR    R5,WTCRDY      ;WAIT FOR CONTROLLER READY
1149 022210                      ESCAPE  SEG                ;CHECK FOR FL:LOE, ELSE EXIT SEG
(3) 022210 104010                      EMT    C$ESCAPE
(3) 022212 000030                      .WORD  10000$-.
1150
1151 022214 013737 002234 002166      MOV    E.MP,GDDAT     ;READ DRIVE STATUS
1152 022222 032737 020000 002166      BIT    #BIT13,GDDAT   ;WRITE LOCK ERROR?
1153 022230 001404                      BEQ    4$              ;NO, BRANCH
1154
1155
1156 022232                      ERRSF   3.,WRLOCK,ERRO      ;WRITE LOCK ERROR
(3) 022232 104461                      TRAP    T$ERCODE
(5) 022234 000003                      .WORD  3
(5) 022236 010331                      .WORD  WRLOCK
(5) 022240 014244                      .WORD  ERRO
1157 022242                      4$:
1158
1159
1160 022242                      ENDSEG                ;%%END OF SEGMENT%%
(3) 022242                      10000$:
(3) 022242 104005                      EMT    C$FSEG
1161 022244                      ENDTST                ;**END OF TEST**
(3) 022244                      L10027:
(3) 022244 104001                      EMT    C$ETST
1162
1163                      .SBTTL  **TEST 3** - WRITE FUNCTION INTERRUPT
1164
1165 022246                      BGTST                ;**START OF TEST**
1166
1167 022246                      STARS
(2)                      ;*****
1168                      ;CHECK OF WRITE LOGIC UNDER INTERRUPT MODE, WE WILL ISSUE A
1169                      ;READ HEADER SO THAT WE DON'T WRITE ON THE BAD SECTOR FILE
1170                      ;TRACK. WE WILL WRITE A FULL SECTOR (128 WORDS) FROM MEMORY (BUF).
1171                      ;WE CHECK THAT NO ERRORS OCCUR. WE DO NOT CHECK RLDA OR RLBA
1172                      ;INCREMENT AT THIS TIME.
1173 022246                      STARS
(2)                      ;*****
1174
1175

```

```

ASSEMBLY ROUTINES MACY11 30A(1052) 22-NOV-78 15:35 PAGE 1-29 H 5
CZRLBB.P11 22-NOV-78 15:28 **TEST 3** - WRITE FUNCTION INTERRUPT SEQ 0059

1176 022246 004737 021356 JSR PC,HDHOME ;HEADS OVER TRACK 0
1177 022252 CKFRFG ;HEADS GO HOME OKAY
(4) 022260 104032 EMT C$EXIT
(4) 022262 000112 .WORD L10030-.
1178
1179 022264 BGNSEG ;%%START OF SEGMENT%%
(3) 022264 104004 EMT C$BSEG
1180
1181
1182 022266 005037 002144 CLR INTFLG ;CLEAR INTERRUPT OCCURANCE FLAG
1183 022272 005077 157750 CLR @RLDA
1184 022276 012777 177600 157744 MOV #-128,@RLMP ;SET UP WORD COUNT
1185 022304 012777 003052 157732 MOV #BUF,@RLBA ;SET UP BUS ADDRESS
1186
1187 022312 SETPRI #PRI00 ;PRIORITY TO 0
(3) 022312 012700 000000 MOV #PRI00,R0
(3) 022316 104041 EMT C$SPRI
1188 022320 004537 020456 JSR R5,LDFUNC ;LOAD THE FUNCTION IN NEXT WORD
1189 022324 000112 WRITE.INTEN ;WRITE UNDER INTERRUPT
1190 022326 004537 021276 JSR R5,WTCRDY ;WAIT FOR INTERRUPT
1191 022332 ESCAPE SEG ;CHECK FOR FL:LOE, ELSE EXIT SEG
(3) 022332 104010 EMT C$ESCAPE
(3) 022334 000036 .WORD 10000$-.
1192
1193 022336 SETPRI #PRI07 ;SET PRIORITY TO 7
(3) 022336 012700 000340 MOV #PRI07,R0
(3) 022342 104041 EMT C$SPRI
1194 022344 005737 002144 TST INTFLG ;DID INTERRUPT OCCUR?
1195 022350 001004 BNE 2$ ;YES-BRANCH NO-REPORT
1196
1197 022352 ERRDF 4,EM17,ERR0 ;WRITE DID NOT INTERRUPT
(3) 022352 104462 TRAP T$ERCODE
(5) 022354 000004 .WORD 4
(5) 022356 011351 .WORD EM17
(5) 022360 014244 .WORD ERR0
1198 2$: 022362 ESCAPE SEG ;CHECK FOR FL:LOE, ELSE EXIT SEG
(3) 022362 104010 EMT C$ESCAPE
(3) 022364 000006 .WORD 10000$-.
1199
1200 022366 004537 020214 JSR R5,CHERR ;CHECK CNTLR FOR ERRORS
1201
1202 022372 ENDSEG ;%%END OF SEGMENT%%
(3) 022372 10000$: EMT C$ESEG
(3) 022372 104005 ENDTST L10030: ;**END OF TEST**
1203 022374 (3) 022374 104001 EMT C$ETST
1204
1205 .SBTTL **TEST 4** - PROPER INCREMENT OF RLBA ON WRITE
1206
1207 022376 BGNST ;**START OF TEST**
1208
1209
1210 022376 STARS
(2) ;*****
1211 ;CHECK THAT THE RLBA WILL INCREMENT PROPERLY AFTER THE

```

```

1212
1213
1214
1215 022376
1216
1217
1218 022376 004737 021356 JSR PC,MDHOME ;HEADS OVER TRACK 0
1219 022402 ZKERFG ;HEADS GO HOME OKAY
(4) 022410 104032 EMT C$EXIT
(4) 022412 000116 .WORD L10031-.
1220
1221 022414 BGNSEG ;%%START OF SEGMENT%%
(3) 022414 104004 EMT C$BSEG
1222
1223 022416 3$: CLR @RLDA
1224 022416 005077 157624 MOV #BUF,@RLBA ;SET UP BUS ADDRESS
1225 022422 012777 003052 157614 MOV #-128,@RLMP ;WORD COUNT
1226 022430 012777 177600 157612 MOV #BUF,GDDAT ;FORM EXPECTED BUS ADDRESS
1227 022436 012737 003052 002166 MOV #256,GDDAT ;AFTER WRITE
1228 022444 062737 000400 002166 ADD
1229
1230 022452 004537 020456 JSR R5,LDFUNC ;LOAD THE FUNCTION IN NEXT WORD
1231 022456 000012 WRITE ;WRITE
1232 022460 004537 021276 JSR R5,WTCRDY ;WAIT FOR CONTROLLER READY
1233 022464 ESCAPE SEG ;CHECK FOR FL:LOE, ELSE EXIT SEG
(3) 022464 104010 EMT C$ESCAPE
(3) 022466 000040 .WORD 10000$-.
1234
1235 022470 004537 020214 JSR R5,CHERR ;CHECK CNTLR FOR ERRORS
1236 022474 ESCAPE SEG ;CHECK FOR FL:LOE, ELSE EXIT SEG
(3) 022474 104010 EMT C$ESCAPE
(3) 022476 000030 .WORD 10000$-.
1237 022500 017737 157540 002170 MOV @RLBA,BDDAT ;READ 'RLBA' FOR PRESENT ADDRESS
1238 022506 023737 002170 002166 CMP BDDAT,GDDAT ;DID 'BA' INCREMENT PROPERLY?
1239 022514 001404 BEQ 2$ ;YES, CONTINUE
1240
1241 022516 ERROF 5,EM20,ERR4 ;BA DID NOT INCREMENT
(3) 022516 104462 TRAP 1$ERRCODE
(5) 022520 000005 .WORD 5
(5) 022522 011411 .WORD EM20
(5) 022524 014410 .WORD ERR4
1242
1243 022526 2$:
1244
1245 022526 ENDSEG ;%%END OF SEGMENT%%
(3) 022526 10000$: EMT C$ESEG
(3) 022526 10400$
1246 022530 ENDTST ;**END OF TEST**
(3) 022530 L10031: EMT C$ETST
(3) 022530 104001
1247
1248 .SBTTL **TEST 5** - PROPER INCREMENT OF RLDA ON WRITE
1249
1250 022532 BGNIST ;**START OF TEST**
1251
  
```

1252 022532

(2)

1253

1254

1255

1256

1257 022532

(2)

1258

1259

1260 022532 004737 021356

1261 022536

(4) 022544 104032

(4) 022546 000114

1262

1263 022550

(3) 022550 104004

1264

1265 022552

1266 022552 005037 002166

1267 022556 013777 002166 157462

1268 022564 005237 002166

1269 022570 012777 177600 157452

1270 022576 012777 003052 157440

1271

1272 022604 004537 020456

1273 022610 000012

1274 022612 004537 021276

1275 022616

(3) 022616 104010

(3) 022620 000040

1276

1277 022622 004537 020214

1278 022626

(3) 022626 104010

(3) 022630 000030

1279

1280 022632 013737 002232 002170

1281 022640 023737 002166 002170

1282 022646 001404

1283

1284 022650

(3) 022650 104462

(5) 022652 000006

(5) 022654 011466

(5) 022656 014410

1285

1286 022660

1287

1288 022660

(3) 022660

(3) 022660 104005

1289 022662

(3) 022662

(3) 022662 104001

1290

STARS

:\*\*\*\*\*

:CHECK THAT THE SECTOR INCREMENTS AFTER THE WRITE WAS FINISHED.

:WE RANDOMLY PICK A SECTOR (OTHER THAN LAST TRACK) AND ISSUE

:A FULL SECTOR WRITE THE RLDA SHOULD REFLECT AN INCREMENT

:OF THE SECOTR. 'GDDAT' WAS THE EXPECTED RLDA.

STARS

:\*\*\*\*\*

JSR PC,HDHOME ;HEADS OVER TRACK 0

CKERFG ;HEADS GO HOME OKAY

EMT C\$EXIT

.WORD L10032-

BUNSEG

EMT C\$BSEG

;%%START OF SEGMENT%%

3\$:

CLR GDDAT

MOV GDDAT,@RLDA

;SETUP DISK ADDRESS

INC GDDAT

;CREATE EXPECTED SECTOR

MOV #-128,@RLMP

;WORD COUNT

MOV #BUF,@RLBA

;SETUP BUS ADDRESS

JSR R5,LDFUNC

;LOAD THE FUNCTION IN NEXT WORD

WRITE

;WRITE

JSR R5,WTCRDY

;WAIT FOR CONTROLLER READY

ESCAPE SEG

;CHECK FOR FL:LOE, ELSE EXIT SEG

EMT C\$ESCAPE

.WORD 10000\$-

JSR R5,CHERR

;CHECK CNTLR FOR ERRORS

ESCAPE SEG

;CHECK FOR FL:LOE, ELSE EXIT SEG

EMT C\$ESCAPE

.WORD 10000\$-

MOV E,DA,BDDAT

;READ DISK ADDRESS

CMP GDDAT,BDDAT

;DID SECTOR INCREMENT PROPERLY

BEQ 2\$

;YES, BRANCH NO, REPORT ERROR

ERRDF 6,EM21,ERR4

;DA DID NOT INCREMENT

TRAP T\$ERRCODE

.WORD 6

.WORD EM21

.WORD ERR4

2\$:

ENDSEG

;%%END OF SEGMENT%%

10000\$:

EMT C\$ESEG

ENDTST

;\*\*END OF TEST\*\*

L10032:

EMT C\$TST

```

ASSEMBLY ROUTINES          MACY11 30A(1052) 22-NOV-78 15:35 PAGE 1-32 K 5
CZRLBB.P11                22-NOV-78 15:28  **TEST 6** - FORCE HEADER NOT FOUND WITH WRITE
                                                                    SEQ 0062

1291      .SBTTL  **TEST 6** - FORCE HEADER NOT FOUND WITH WRITE
1292
1293 022664  BGNTST                                     ;**START OF TEST**
1294
1295 022664  STARS
1296  (2)      ;*****
1297          ;FORCE HEADER NOT FOUND ERROR TO OCCUR. THIS IS DONE
1298          ;BY SETTING SECTOR 40 OF THE RLDA AND ISSUING A
1299          ;WRITE. SECTOR 40 DOES NOT EXIST ON THE RL01 PACK
1300          ;THEREFORE HDR NT FOUND SHOULD SET.
1301  (2)      STARS
1302          ;*****
1302 022664 004737 021356      JSR      PC,HDHOME          ;HEADS OVER TRACK 0
1303 022670      CKERFG          ;HEADS GO HOME OKAY
1304  (4) 022676 104032      EMT      C$EXIT
1305  (4) 022700 000120      .WORD    !10033-.
1306
1305 022702      BGNSEG          ;%%START OF SEGMENT%%
1306  (3) 022702 104004      EM      C$BSEG
1307
1308 022704 012777 000050 157334  MOV     #40.,@RLDA      ;INSURE NOT TO FIND HEADER BY
1309 022712 012777 003052 157324  MOV     #BUF,@RLBA     ;SETTING SECTOR 40 OF CYL. ADDR.
1310 022720 012777 177777 157322  MOV     #-1,@RLMP     ;WORD COUNT
1311
1312 022726 004537 020456      JSR      R5,LDFUNC          ;LOAD THE FUNCTION IN NEXT WORD
1313 022732 000012      WRITE      ;WRITE
1314 022734 004537 021276      JSR      R5,WTCRDY         ;WAIT FOR CONTROLLER READY
1315 022740      ESCAPE SEG      ;CHECK FOR FL:LOE, ELSE EXIT SEG
1316  (3) 022740 104010      EMT      C$ESCAPE
1317  (3) 022742 000054      .WORD    10000$-.
1318
1317 022744 013737 002226 002160  MOV     E.CS,TMPO      ;GET RLCS
1318 022752 042737 001777 002160  BIC     #1777,TMPO     ;SAVE ERROR BITS
1319 022760 022737 112000 002160  CMP     #BIT15.BIT12:BIT10,TMPO ;HDR NOT FOUND SET.
1320 022766 001402      BEQ      1$                      ;YES, CONTINUE
1321
1322 022770 004537 020214      JSR      R5,CHERR
1323 022774      CKLOOP
1324  (3) 022774 104006      EMT      C$CLP1
1325
1325 022776 022737 112000 002160  CMP     #BIT15.BIT12:BIT10,TMPO
1326 023004 001404      BEQ      2$
1327 023006      ERRDF 23.,EM10,ERRO
1328  (3) 023006 104462      TRAP     T$ERCODE
1329  (5) 023010 000027      .WORD    23
1330  (5) 023012 011043      .WORD    EM10
1331  (5) 023014 014244      .WORD    ERRO
1332
1328      ;WHEN FORCED
1329 023016      2$:
1330
1331 023016      ENDSEG          ;%%END OF SEGMENT%%
1332  (3) 023016      10000$:
1333  (3) 023016 104005      EMT      C$ESEG
1334 023020      ENDTST          ;**END OF TEST**

```

```

ASSEMBLY ROUTINES          MACY11 30A(1052) 22-NOV-78 15:35 PAGE 1-33 L 5
CZRLBB.P11 22-NOV-78 15:28 **TEST 6** - FORCE HEADER NOT FOUND WITH WRITE          SEQ 0063

(3) 023020          L10033:          EMT          C$ETST
(3) 023020 104001
1333
1334          .SBTTL **TEST 7** - FORCE HEADER NOT FOUND WITH WRITE INTERRUPT
1335
1336 023022          BGNTST          ;**START OF TEST**
1337
1338
1339 023022          STARS
(2)          :*****
1340          :TEST THAT HEADER NOT FOUND ERROR WILL GENERATE AN INTERRUPT
1341          :ON OCCURANCE. HEADER NOT FOUND WILL BE FORCED BY SETTING
1342          :SECTOR 40 OF RLDA AND ISSUING A WRITE
1343 023022          STARS
(2)          :*****
1344
1345
1346 023022 00 737 021356          JSR          PC,HDHOME          ;HEADS OVER TRACK 0
1347 023026          CKERFG          ;HEADS GO HOME OKAY
(4) 023034 104032          EMT          C$EXIT
(4) 023036 000160          .WORD          L10034-.
1348
1349 023040          BGNSEG          ;%%START OF SEGMENT%%
(3) 023040 104004          EMT          C$BSEG
1350
1351 023042          SETPRI          #PRI00
(3) 023042 012700 000000          MOV          #PRI00,R0
(3) 023046 104041          EMT          C$SPRI
1352 023050 005037 002144          CLR          INTFLG          ;CLEAR INTERRUPT OCCURANCE FLAG
1353 023054 012777 000050 157164          MOV          #40.,@RLDA          ;INSURE NOT TO FIND HEADER BY
1354 023062 012777 003052 157154          MOV          #BUF,@RLBA          ;SETTING SECTOR 40 OF CYL. ADDR.
1355 023070 012777 177777 157152          MOV          #-1,@RLMP          ;WORD COUNT
1356
1357 023076 004537 020456          JSR          R5,DFUNC          ;LOAD THE FUNCTION IN NEXT WORD
1358 023102 000112          WRITE.INTEN          .WRITE
1359 023104 004537 021276          JSR          R5,WTCRDY          ;WAIT FOR CONTROLLER READY
1360 023110          CKLOOP
(3) 023110 104006          EMT          C$CLP1
1361 023112          SETPRI          #PRI07
(3) 023112 012700 000340          MOV          #PRI07,R0
(3) 023116 104041          EMT          C$SPRI
1362
1363 023120 005737 002144          TST          INTFLG          ;DID INTERRUPT OCCUR
1364 023124 001004          BNE          2$          ;YES OKAY
1365
1366 023126          ERRDF          24.,EM43,ERRO          ;NO INTERRUPT FROM OPI
(3) 023126 104462          TRAP          T$ERCODE
(5) 023130 000030          .WORD          24
(5) 023132 012723          .WORD          EM43
(5) 023134 014244          .WORD          ERRO
1367
1368 023136          2$:          ESCAPE          SEG          ;CHECK FOR FL:LOE, ELSE EXIT SEG
(3) 023136          EMT          C$ESCAPE
(3) 023140 000054          .WORD          10000$-.
1369
1370

```

```

ASSEMBLY ROUTINES          MACY11 30A(1052) 22-NOV-78 15:35 PAGE 1-34 M 5
CZRLBB.P11 22-NOV-78 15:28 **TEST 7** - FORCE HEADER NOT FOUND WITH WRITE INTERRUPT          SEQ 0064

1371 023142 013737 002226 002160      MOV      E,CS, TMP0      ;GET RLCS
1372 023150 042737 001777 002160      BIC      #1777, TMP0    ;SAVE ERROR BITS
1373 023156 022737 112000 002160      CMP      #BIT15.BIT12.BIT10, TMP0 ;WDR NOT FOUND SET.
1374 023164 001402                      BEQ      1$              ;YES, CONTINUE
1375
1376 023166 004537 020214                JSR      R5,CHERR
1377 023172                      1$:    CKLOOP
(3) 023172 104006                      EMT      C$CLP1
1378
1379 023174 022737 112000 002160      CMP      #BIT15!BIT12!BIT10, TMP0
1380 023202 001404                      BEQ      3$
1381 023204                      ERRDF  25.,EM10,ERR0
(3) 023204 104462                      TRAP    T$ERCODE
(5) 023206 000031                      .WORD   25
(5) 023210 011043                      .WORD   EM10
(5) 023212 014244                      .WORD   ERR0
1382
1383 023214                      3$:
1384
1385 023214                      ENDSEG                      ;%%END OF SEGMENT%%
(3) 023214                      10000$:
(3) 023214 104005                      EMT      C$ESEG
1386 023216                      ENDTST                      ;**END OF TEST**
(3) 023216                      L10034:
(3) 023216 104001                      EMT      C$ETST
1387
1388
1389
1390                      .SBTTL  **TEST 8** - CHECK OPI TIME WITH HDR NT FND
1391
1392 023220                      BGNST                      ;**START OF TEST**
1393
1394 023220                      STARS
(2)
1395                      :*****
1396                      :CHECK OPI TIME IT SHOULD BE AROUND 200 MILLISECONDS (ON UNIBUS)
1397                      :CHECK THIS BY TIMING OPI ON A FORCED HEADER NOT FOUND
1398 023220                      :ISSUE WRITE WITH SECTOR 40 SET IN THE DISK ADDRESS
(2)
1399                      :*****
1400 023220 004737 021356                JSR      PC,HDHOME      ;HEADS OVER TRACK 0
1401 023224                      CKERFG                      ;HEADS GO HOME OKAY
(4) 023232 104032                      EMT      C$EXIT
(4) 023234 000264                      .WORD   L10035-.
1402
1403 023236                      BGNSEG                      ;%%START OF SEGMENT%%
(3) 023236 104004                      EMT      C$BSEG
1404
1405 023240                      CLRVEC                      ;CLEAR PRESENT INTERRUPT VECTOR
(3) 023240 013700 002254                MOV      BVEC,R0
(3) 023244 104036                      EMT      C$CVEC
1406 023246                      SETVEC  BVEC,#TIMSRV,#340      ;SET INTR. VEC. WITH ABORT WAIT
(7) 023246 012746 000340                MOV      #340,-(SP)
(6) 023252 012746 020116                MOV      #TIMSRV,-(SP)
(5) 023256 013746 002254                MOV      B,FC,-(SP)
(4) 023262 012746 000003                MOV      #3,-(SP)

```

```

(3) 023266 104037 EMT C$SVEC
(2) 023270 062706 000010 ADD #10,SP
1407 023274 SETPRI #PRI00
(3) 023274 012700 000000 MOV #PRI00,R0
(3) 023300 104041 EMT C$SPRI
1408 023302 005037 002144 CLR INTFLG ;CLEAR INTERRUPT FLAG
1409 023306 012777 000050 156732 MOV #40.,@RLDA ;SET UP FOR HDR NT FND
1410 023314 012777 003052 156722 MOV #BUF,@RLBA ;BUS ADDRESS
1411 023322 012777 177777 156720 MOV #-1,@RLMP ;WORD COUNT
1412
1413 023330 004537 020456 JSR R5,LDFUNC ;LOAD THE FUNCTION IN NEXT WORD
1414 023334 000112 WRITE!INTEN
1415
1416 023336 013700 002302 MOV OPIMX,R0
1417 023342 006300 ASL R0
1418 023344 006300 ASL R0
1419 023346 006300 ASL R0
1420 023350 063700 002302 ADD OPIMX,R0
1421 023354 063700 002302 ADD OPIMX,R0
1422 023360 WAITUS R0 ;WAIT MAX MILLISECONDS
(3) 023360 104027 EMT C$WTJ
1423 023362 010037 002170 MOV R0,BDDAT ;SETUP FOR WORST CASE
1424 023366 005737 002144 TST INTFLG ;DID INTERRUPT OCCUR
1425 023372 001427 BEQ 4$ ;NO, REPORT ERROR
1426
1427 023374 GETTIM BDDAT ;GET TIME EXPIRED
(3) 023374 104052 EMT C$GTIM
(3) 023376 010037 002170 MOV R0,BDDAT
1428 023402 005000 CLR R0 ;DIVIDE
1429 023404 162737 000012 002170 1$: SUB #10.,BDDAT ;ANSWER
1430 023412 100402 BMI 3$ ;BY 10 TO GET
1431 023414 005200 INC R0 ;RIGHT ANSWER
1432 023416 000772 BR 1$
1433 023420 010037 002170 3$: MOV R0,BDDAT
1434
1435 ;CHECK THAT OPI TIME IS WITHIN LIMITS
1436
1437 023424 2$: SETPRI #PRI07
(3) 023424 012700 000340 MOV #PRI07,R0
(3) 023430 104041 EMT C$SPRI
1438 023432 023737 002302 002170 CMP OPIMX,BDDAT ;IS IT WITHIN LIMITS
1439 023440 002404 BLT 4$ ;NO, REPORT ERROR
1440
1441 023442 023737 002300 002170 CMP OPIMN,BDDAT ;WITHIN LIMITS
1442 023450 003404 BLE 5$ ;YES
1443
1444 023452 4$: ERPDF 974.,EM56,ERR13 ;OPI TIMING INCORRECT
(3) 023452 104462 TRAP T$ERCODE
(5) 023454 001716 .WORD 974.
(5) 023456 013456 .WORD EM56
(5) 023460 015102 .WORD ERR13
1445
1446 023462 5$: CLVEC BVEC ;CLEAR PRESENT VECTOR
(3) 023462 013700 002254 MOV BVEC,R0
(3) 023466 104036 EMT C$VEC
1447 023470 SETVEC BVEC,#INTSRV,#34 ;SET IN OLD VECTOR

```

```

(7) 023470 012746 000340      MOV    #340,-(SP)
(6) 023474 012746 020110      MOV    #INTSRV,-(SP)
(5) 023500 013746 002254      MOV    BVEC,-(SP)
(4) 023504 012746 000003      MOV    #3,-(SP)
(3) 023510 104037              EMT    C$SVEC
(2) 023512 062706 000010      ADD     #10,SP
1448
1449 023516                      ENDSEG                      ;%%END OF SEGMENT%%
(3) 023516                      10000$:
(3) 023516 104005              EMT    C$ESEG
1450
1451 023520                      ENDTST                      ;**END OF TEST**
(3) 023520                      L10035:
(3) 023520 104001              EMT    C$ETST
1452
1453 .SBTTL  **TEST 9** - MULTIPLE SECTOR TRANSFER ON WRITE
1454
1455 023522                      BGNST                      ;**START OF TEST**
1456
1457 023522                      STARS
(2)                               ;:*****
1458                               ;:CHECK FOR MULTIPLE SECTOR TRANSFER ON WRITE. THIS TEST CHECKS
1459                               ;:THAT TWO SECTORS CAN BE SUCCESSFULLY WRITTEN. WE LOAD
1460                               ;:A WORD COUNT OF 129 WORDS (ONE SECTOR + 1 WORD) STARTING AT
1461                               ;:SECTOR 0 THRU SECTOR 37 AND VERIFY THAT THE RLDA DOES
1462                               ;:A DOUBLE INCREMENT EACH TIME.
1463 023522                      STARS
(2)                               ;:*****
1464
1465
1466
1467 023522 004737 021356      JSR     PC,HDHOME      ;HEADS OVER TRACK 0
1468 023526                      CKERFG      ;HEADS GO HOME OKAY
(4) 023534 104032              EMT    C$EXIT
(4) 023536 000152              .WORD    L10036-.
1469
1470 023540 005037 002160      CLR     TMP0      ;CLEAR TEMP LOCATIONS
1471 023544 005037 002162      CLR     TMP1
1472
1473 023550                      BGNSEG                      ;%%START OF SEGMENT%%
(3) 023550 104004              EMT    C$BSEG
1474
1475
1476 023552 013737 002162 002166 1$: MOV    TMP1,GDDAT      ;GET CYLINDER
1477 023560 053737 002160 002166      BIS    TMP0,GDDAT      ;GET SECTOR
1478 023566 013777 002166 156452      MOV    GDDAT,@RLDA      ;SET DISK ADDRESS-SECTOR 0
1479 023574 062737 000002 002166      ADD     #2,GDDAT      ;SET EXPECTED + 2
1480 023602 012777 003052 156434      MOV    #BUF,@RLBA      ;SET BUS ADDRESS
1481 023610 012777 177577 156432      MOV    #-129,@RLMP      ;WORD COUNT-SECTOR+1 WORD
1482
1483 023616 004537 020456      JSR     R5,LDFUNC      ;LOAD THE FUNCTION IN NEXT WORD
1484 023622 000012              WRITE      ;WRITE
1485 023624 004537 021276      JSR     R5,WTCRDY      ;WAIT FOR CONTROLLER READY?
1486 023630              ESCAPE SEG      ;CHECK FOR FL:LOE, ELSE EXIT SEG
(3) 023630 104010              EMT    C$ESCAPE
(3) 023632 000054              .WORD    10000$-.

```

```
1487
1488 023634 004537 020214      JSR    R5,CHERR      ;CHECK CNTLR FOR ERRORS
1489 023640      ESCAPE SEG      ;CHECK FOR FL:LOE, ELSE EXIT SEG
(3) 023640 104010      EMT    C$ESCAPE
(3) 023642 000044      .WORD  10000$-.
1490
1491 023644 013737 002232 002170  MOV    E,DA,BDDAT      ;READ DISK ADDRESS
1492 023652 023737 002170 002166  CMP    BDDAT,GDDAT      ;IS DISK ADDRESS CORRECT
1493 023660 001404      BEQ    2$      ;YES, BRANCH NO, REPORT ERROR
1494
1495 023662      ERRDF  7,EM22,ERR4      ;DISK ADDRESS NOT CORRECT
(3) 023662 104462      TRAP  T$ERRCODE
(5) 023664 000007      .WORD  7
(5) 023666 011544      .WORD  EM22
(5) 023670 014410      .WORD  ERR4
1496
1497 023672      2$:
1498
1499 023672 005237 002160      INC    TMP0      ;NEXT SECTOR
1500 023676 022737 000046 002160  CMP    #46,TMP0      ;AT END?
1501 023704 001322      BNE    1$      ;NO, GO BACK
1502
1503 023706      ENDSEG      ;%%END OF SEGMENT%%
(3) 023706      10000$:
(3) 023706 104005      EMT    C$ESEG
1504 023710      ENDTST      ;**END OF TEST**
(3) 023710      L10036:
(3) 023710 104001      EMT    C$ETST
1505
1506      .SBTTL  **TEST 10** - CHECK DIRECTION OF WRITE NPR
1507
1508 023712      BGNST      ;**START OF TEST**
1509
1510 023712      STARS
(2)      ;*****
1511      ;VERIFY THAT A WRITE IS WRITING NOT READING. WE WRITE A
1512      ;KNOWN PATTERN IN 'BUF' (128 WORD), WF THEN ISSUE A WRITE.
1513      ;ONCE THE WRITE IS FINISHED WE CHECK THAT 'BUF' IS INTACT.
1514      ;THIS IS DONE TO PROVE THAT THE NPR IS GOING THE RIGHT
1515      ;WAY.
1516 023712      STARS
(2)      ;*****
1517
1518
1519 023712 004737 021356      JSR    PC,HDHOME      ;HEADS OVER TRACK 0
1520 023716      CKERFG      ;HEADS GO HOME OKAY
(4) 023724 104032      EMT    C$EXIT
(4) 023726 000160      .WORD  L10037-.
1521
1522 023730      BGNSEG      ;%%START OF SEGMENT%%
(3) 023730 104004      EMT    C$BSEG
1523
1524 023732      2$:
1525 023732 012702 003052      MOV    #BUF,R2      ;WRITE BUFFER FOR WRITE OPERATION
1526 023736 012701 000200      MOV    #28,R1      ;ONE SECTOR'S WORTH
1527 023742 012722 125252      MOV    #125252,(R2)+      ;WRITE BUFFER
```

```

1528 023746 005301      DEC      R1      ;DONE?
1529 023750 001374      BNE      3$      ;NO, GO BACK
1530
1531 023752 005077 156270      CLR      @RLDA      ;LOAD DISK ADDRESS
1532 023756 012777 177600 156264      MOV      #-128.,@RLMP      ;WORD COUNT
1533 023764 012777 003052 156252      MOV      #BUF,@RLBA      ;BUS ADDRESS
1534 023772 004537 020456      JSR      R5,LDFUNC      ;LOAD THE FUNCTION IN NEXT WORD
1535 023776 000012      WRITE      ;WRITE SOME DATA
1536 024000 004537 021276      JSR      R5,WTCRDY      ;WAIT FOR IT TO FINISH
1537 024004      ESCAPE      SEG      ;CHECK FOR FL:LOE, ELSE EXIT SEG
(3) 024004 104010      EMT      C$ESCAPE
(3) 024006 000076      .WORD      10000$-.
1538
1539 024010 004537 020214      JSR      R5,CHERR      ;CHECK CNTLR FOR ERRORS
1540 024014      ESCAPE      SEG      ;CHECK FOR FL:LOE, ELSE EXIT SEG
(3) 024014 104010      EMT      C$ESCAPE
(3) 024016 000066      .WORD      10000$-.
1541
1542 024020 012702 003052      MOV      #BUF,R2      ;SET UP TO CHECK BUFFER
1543 024024 012701 000200      MOV      #128.,R1      ;CHECK 128 WORDS
1544
1545 024030      BGNSEG      ;%%START OF SEGMENT%%
(3) 024030 104004      EMT      C$BSEG
1546
1547 024032 012737 125252 002166      MOV      #125252,GDDAT      ;DATA SHOULD BE 125252
1548 024040 011237 002170 4$:      MOV      (R2),BDDAT      ;LOAD DATA INTO BDDAT
1549 024044 023737 002166 002170      CMP      GDDAT,BDDAT      ;IS IT OKAY?
1550 024052 001406      BEQ      5$      ;YES, CONTINUE
1551
1552 024054 010237 002162      MOV      R2,TMP1 ;LOAD MEMORY LOCATION OF FAILURE
1553 024060      ERRDF      8.,EM26,ERR8
(3) 024060 104462      TRAP      T$ERRCODE
(5) 024062 000010      .WORD      8
(5) 024064 012045      .WORD      EM26
(5) 024066 014564      .WORD      ERR8
1554
1555 024070 5$:      ESCAPE      SEG      ;CHECK FOR FL:LOE, ELSE EXIT SEG
(3) 024070 104010      EMT      C$ESCAPE
(3) 024072 000010      .WORD      10001$-.
1556 024074 005722 6$:      TST      (R2)+      ;NEXT:
1557 024076 005301      DEC      R1      ;DONE?
1558 024100 001357      BNE      4$      ;NO, GO BACK
1559
1560 024102      ENDSEG      ;%%END OF SEGMENT%%
(3) 024102 10001$:      EMT      C$ESEG
(3) 024102 104005      ENDSEG      ;%%END OF SEGMENT%%
1561 024104 10000$:      EMT      C$LSEG
(3) 024104 104005      ENDTST      ;**END OF TEST**
(3) 024106 L10037:      EMT      C$ETST
(3) 024106 104001
1563
1564 .SBTTL **TEST 11** - CHECK FULL RLBA INCREMENT
1565
1566 024110 BGNTST      ;**START OF TEST**
  
```

```

1567
1568 024110 . STARS
(2) :*****
1569 :TEST THAT THE RLBA WILL INCREMENT, WE DO NOT DO A FULL 16
1570 :BIT INCREMENT WE CHECK THAT EACH BIT WILL TOGGLE 0 TO 1
1571 :AND 1 TO 0. WE DO CHECK ALL BITS EVEN IF ALL MEMORY
1572 :IS NOT AVAILABLE. (WE IGNORE NON-EXISTANT MEMORY ERRORS).
1573 :WE USE THE SAME DISK ADDRESS (RANDOM) AND A 1 WORD TRANSFER.
1574 024110 STARS
(2) :*****
1575
1576
1577 024110 004737 021356 JSR PC,HDHOME ;HEADS OVER TRACK 0
1578 024114 CKERFG ;HEADS GO HOME OKAY
(4) 024122 104032 EMT C$EXIT
(4) 024124 000134 .WORD L10040-.
1579
1580
1581 024126 005037 002162 CLR TMP1 ;CLEAR LOCATION
1582
1583 024132 BGNSEG ;%%START OF SEGMENT%%
(3) 024132 104004 EMT C$BSEG
1584
1585 024134 3$: MOV #-1,@RLMP ;ONLY ONE (1) WORD
1586 024134 012777 177777 156106 CLR @RLDA ;LOAD DISK ADDRESS
1587 024142 005077 156100 MOV TMP1,@RLBA ;BUS ADDRESS
1588 024146 013777 002162 156070
1589
1590 024154 004537 020456 JSR R5,LDFUNC ;LOAD THE FUNCTION IN NEXT WORD
1591 024160 000012 WRITE
1592 024162 004537 021276 JSR R5,WTCRDY ;WAIT FOR WRITE TO FINISH
1593 024166 ESCAPE SEG ;CHECK FOR FL:LOE, ELSE EXIT SEG
(3) 024166 104010 EMT C$ESCAPE
(3) 024170 000066 .WORD 10000$-.
1594
1595 024172 013737 002162 002166 4$: MOV TMP1,GDDAT ;SET UP EXPECTED RLBA
1596 024200 062737 000002 002166 ADD #2,GDDAT ;PREVIOUS RLBA+2
1597 024206 013737 002230 002170 MOV E.BA,BDDAT ;READ RLBA
1598 024214 023737 002166 002170 CMP GDDAT,BDDAT ;WAS IT UPDATED PROPERLY?
1599 024222 001404 BEQ 5$ ;YES, CONTINUE
1600
1601 024224 ERRDF 9.,EM30,ERR4 ;BA INCREMENT ERROR
(3) 024224 104462 TRAP T$ERRCODE
(5) 024226 000011 .WORD 9
(5) 024230 012165 .WORD EM30
(5) 024232 014410 .WORD ERR4
1602 024234 5$: ESCAPE SEG ;CHECK FOR FL:LOE, ELSE EXIT SEG
(3) 024234 104010 EMT C$ESCAPE
(3) 024236 000020 .WORD 10000$-.
1603
1604 024240 006337 002162 ASL TMP1 ;NEXT PATTERN TO TEST RLBA
1605 024244 103404 BCS 6$ ;DONE?
1606 024246 052737 000002 002162 BIS #BIT1,TMP1 ;NO, SET IN BIT 1
1607 024254 000727 BR 3$ ;GO CHECK NEXT.
1608
1609 024256 6$: ;END TEST

```

```

1610
1611 024256          ENDSEG                      ;%%END OF SEGMENT%%
(3) 024256          10000$: EMT C$ESEG
(3) 024256 104005
1612 024260          ENDIST                      ;**END OF TEST**
(3) 024260          L10040: EMT C$ETST
(3) 024260 104001
1613
1614 .SBTTL **TEST 12** - BA BIT 16 INCREMENT
1615
1616 024262          BGNST                        ;**START OF TEST**
1617
1618 024262          STARS
(2) :*****
1619 :CHECK THAT BA BIT 16 WILL INCREMENT. WE WILL LOAD THE
1620 :RLBA WITH 177776 AND ISSUE A ONE WORD WRITE WE THEN
1621 :CHECK BA BIT 16 TO SET, BA 17 TO STAY A 0 AND THE RLBA
1622 :TO GO TO ZERO
1623 024262          STARS
(2) :*****
1624
1625
1626 024262 004737 021356 JSR PC,HDHOME          ;HEADS OVER TRACK 0
1627 024266          CKERFG                      ;HEADS GO HOME OKAY
(4) 024274 104032          EMT C$EXIT
(4) 024276 000160          .WORD L10041-.
1628
1629 024300          BGNSEG                      ;%%START OF SEGMENT%%
(3) 024300 104004          EMT C$BSEG
1630
1631 024302          2$:
1632 024302 012777 177776 155734 MOV #177776,@RLBA ;SET MAX BA TO INC. BA16
1633 024310 005037 002262 CLR XMEM              ;WE DON'T WANT TO LOAD ANY EA
1634 024314 012777 177777 155726 MOV #-1,@RLMP   ;ONE WORD TRANSFER
1635 024322 005077 155720 CLR @RLDA
1636 024326 004537 020456 JSR R5,LDFUNC          ;LOAD THE FUNCTION IN NEXT WORD
1637 024332 000012 WRITE
1638 024334 004537 021276 JSR R5,WTCRDY          ;WAIT FOR WRITE TO FINISH
1639 024340          ESCAPE SEG                  ;CHECK FOR FL:LOE, ELSE EXIT SEG
(3) 024340 104010          EMT C$ESCAPE
(3) 024342 000112 .WORD 10000$-.
1640 024344 032737 020000 002226 BIT #NXM,E.CS ;NON-EXISTANT MEMORY ERROR?
1641 024352 001002 BNE 3$                      ;YES, CONTINUE
1642
1643 024354 004537 020214          3$: JSR R5,CHERR ;CHECK CNTLR FOR ERRORS
1644 024360          ESCAPE SEG                  ;CHECK FOR FL:LOE, ELSE EXIT SEG
(3) 024360 104010          EMT C$ESCAPE
(3) 024362 000072 .WORD 10000$-.
1645
1646 024364 032737 000020 002226 BIT #BA16,E.CS ;DID BA16 SET?
1647 024372 001004 BNE 4$                      ;YES, CONTINUE
1648
1649 024374          ERRDF 10.,EM31,ERRC ;BA 16 DID NOT INCREMENT
(3) 024374 104462 TRAP T$ERCODE
(5) 024376 000012 .WORD 10
(5) 024400 012225 .WORD EM31
  
```

```

(5) 024402 014244 .WORD ERRO
1650
1651 024404 4$: CKLOOP
(3) 024404 104006 EMT C$CLP1
1652
1653 024406 032737 000040 002226 BIT #BA17,E.CS ;DID BA17 SET ALSO?
1654 024414 001404 BEQ 5$ ;NO, GOOD CONTINUE
1655
1656 024416 ERRDF 11.,EM32,ERRO ;BA 17 GOT CARRIED AWAY
(3) 024416 104402 TRAP T$ERCODE
(5) 024420 000013 .WORD 11
(5) 024422 012270 .WORD EM32
(5) 024424 014244 .WORD ERRO
1657
1658 024426 5$: CKLOOP
(3) 024426 104006 EMT C$CLP1
1659
1660 024430 005037 002166 CLR GDDAT ;CHECK THAT BA15-BA0 IS CLEAR
1661 024434 013737 002230 002170 MOV E,BA,BDDAT ;READ BA
1662 024442 001404 BEQ 6$ ;IS BA ZERO?
1663 024444 ERRDF 12.,EM33,ERR4 ;BA SHOULD BE ZERO
(3) 024444 104462 TRAP T$ERCODE
(5) 024446 000014 .WORD 12
(5) 024450 012335 .WORD EM33
(5) 024452 014410 .WORD ERR4
1664
1665 024454 6$: ;
1666
1667 024454 ENDSEG ;%%END OF SEGMENT%%
(3) 024454 10000$:
(3) 024454 104005 EMT C$ESEG
1668 024456 ENDTST ;**END OF TEST**
(3) 024456 L10041:
(3) 024456 104001 EMT C$E1ST
1669
1670 .SBTTL **TEST 13** - BA BIT 17 INCREMENT
1671
1672 024460 BGN1ST ;**START OF TEST**
1673
1674 024460 STARS
(2) ;*****
1675 ;CHECK THAT BA BIT 17 WILL INCREMENT. WE WILL LOAD THE
1676 ;RLBA WITH 177776 AND BA 16 SET, WE WILL ISSUE A ONE WORD
1677 ;WRITE. WE THEN CHECK BA17 TO SET, BA16 TO CLEAR AND
1678 ;BA15 - BAO TO CLEAR.
1679 024460 STARS
(2) ;*****
1680
1681
1682
1683 024460 004737 021356 JSR PC,HDHOME ;HEADS OVER TRACK 0
1684 024464 CKERFG ;HEADS GO HOME OKAY
(4) 024472 104032 EMT C$EXIT
(4) 024474 000162 .WORD L10042-.
1685
1686 024476 BGNSEG ;%%START OF SEGMENT%%
  
```

```

(3) 024476 104004 EMT CSBSEG
1687
1688 024500 2$:
1689 024500 012777 177776 155536 MOV #177776,@RLBA ;SET MAX BA TO INC. BA16
1690 024506 012737 000020 002262 MOV #BA16,XMEM ;SET BA16 IN RLCS
1691 024514 012777 177777 155526 MOV #-1,@RLMP ;ONE WORD TRANSFER
1692 024522 005077 155520 CLR @RLDA
1693 024526 004537 020456 JSR R5,LDFUNC ;LOAD THE FUNCTION IN NEXT WORD
1694 024532 000012 WRITE
1695 024534 004537 021276 JSR R5,WTCRDY ;WAIT FOR WRITE TO FINISH
1696 024540 ESCAPE SEG ;CHECK FOR FL:LOE, ELSE EXIT SEG
(3) 024540 104010 EMT C$ESCAPE
(3) 024542 000112 .WORD 10000$-
1697 024544 032737 020000 002226 BIT #NXM,E.CS ;NON-EXISTANT MEMORY ERROR?
1698 024552 001002 BNE 3$ ;YES, CONTINUE
1699
1700 024554 004537 020214 JSR R5,CHERR ;CHECK CNTLR FOR ERRORS
1701 024560 3$: ESCAPE SEG ;CHECK FOR FL:LOE, ELSE EXIT SEG
(3) 024560 104010 EMT C$ESCAPE
(3) 024562 000072 .WORD 10000$-
1702
1703 024564 032737 000040 002226 BIT #BA17,E.CS ;DID BA17SET?
1704 024572 001004 BNE 4$ ;YES, CONTINUE
1705
1706 024574 ERRDF 13.,EM34,ERRO ;BA 17 DID NOT SET
(3) 024574 104462 TRAP T$ERCODE
(5) 024576 000015 .WORD 13
(5) 024600 012376 .WORD EM34
(5) 024602 014244 .WORD ERRO
1707
1708 024604 4$: CKLOOP
(3) 024604 104006 EMT C$CLP1
1709
1710 024606 032737 000020 002226 BIT #BA16,E.CS ;DID BA16 SET ALSO?
1711 024614 001404 BEQ 5$ ;NO, GOOD CONTINUE
1712
1713 024616 ERRDF 14.,EM35,ERRO ;BA 16 DIDN'T KNOW WHEN TO QUIT.
(3) 024616 104462 TRAP T$ERCODE
(5) 024620 000016 .WORD 14
(5) 024622 012441 .WORD EM35
(5) 024624 014244 .WORD ERRO
1714 024626 5$: CKLOOP
(3) 024626 104006 EMT C$CLP1
1715
1716 024630 005037 002166 CLR GDDAT ;CHECK THAT BA15-BA0 IS CLEAR
1717 024634 013737 002230 002170 MOV E.BA,BDDAT ;READ BA
1718 024642 001404 BEQ 6$ ;IS BA ZERO?
1719 024644 ERRDF 15.,EM36,ERR4 ;BA SHOULD BE ZERO
(3) 024644 104462 TRAP T$ERCODE
(5) 024646 000017 .WORD 15
(5) 024650 012506 .WORD EM36
(5) 024652 014410 .WORD ERR4
1720
1721 024654 6$:
1722
1723 024654 ENDSEG ;%%END OF SEGMENT%%
  
```

```

(3) 024654 10000$:
(3) 024654 104005 EMT C$ESEG
1724 024656 ENDTST ;**END OF TEST**
(3) 024656 L10042:
(3) 024656 104001 EMT C$ETST
1725
1726 .SBTTL **TEST 14** - TEST READ NPR INTEGRITY
1727
1728 024660 BGNSTST ;**START OF TEST**
1729
1730
1731
1732
1733 024660 STARS
(2) ;*****
1734 ;CHECK THAT NPR WILL NOT INTERFERE WITH THE OPERATION OF THE UNIBUS
1735 ;WE SETUP LOCATION 4 TO HANDLE THE TRAP IF IT HAPPENS
1736 024660 STARS
(2) ;*****
1737
1738
1739 024660 004737 021356 JSR PC,HDHOME ;HEADS OVER TRACK 0
1740 024664 CKERFG ;HEADS GO HOME OKAY
(4) 024672 104032 EMT C$EXIT
(4) 024674 000132 .WORD L10043-.
1741
1742 024676 BGNSEG ;%%START OF SEGMENT%%
(3) 024676 104004 EMT C$BSEG
1743
1744
1745 024700 1$: SETVEC ERRVEC,#TRPHAN,#340 ;SET UP VECTOR
(7) 024700 MOV #340,-(SP)
(6) 024704 012746 000340 MOV #TRPHAN,-(SP)
(5) 024710 013746 002132 MOV ERRVEC,-(SP)
(4) 024714 012746 000003 MOV #3,-(SP)
(3) 024720 104037 EMT C$SVEC
(2) 024722 062706 000010 ADD #10,SP
1746 024726 005037 002142 CLR TRPFLG ;CLEAR TRAP FLAG
1747 024732 012777 003052 155304 MOV #BUF,@RLBA ;LOAD BA
1748 024740 005077 155302 CLR @RLDA ;LOAD DA
1749 024744 012777 177777 155276 MOV #-1,@RLMP ;LOAD WC
1750 024752 004537 020456 JSR R5,LDFUNC ;LOAD THE FUNCTION IN NEXT WORD
1751 024756 000014 READ
1752 024760 004537 021276 JSR R5,WTCRDY
1753 024764 CLRVEC ERRVEC ;CLEAR OUT VECTOR
(3) 024764 013700 002132 MOV ERRVEC,R0
(3) 024770 104036 EMT C$CVEC
1754 024772 ESCAPE SEG ;CHECK FOR FL:LOE, ELSE EXIT SEG
(3) 024772 104010 EMT C$ESCAPE
(3) 024774 000030 .WORD 10000$-
1755 024776 004537 020214 JSR R5,CHERR ;CHECK CNTLR FOR ERRORS
1756 025002 ESCAPE SEG ;CHECK FOR FL:LOE, ELSE EXIT SEG
(3) 025002 104010 EMT C$ESCAPE
(3) 025004 000020 .WORD 10000$-
1757
1758 025006 005737 002142 TST TRPFLG ;DID TRAP OCCUR?

```

```

1759 025012 001404      BEQ      7$      ;NO, OKAY
1760 025014      ERRDF    17,EM52,ERRO ;YES, PRINT ERROR
(3) 025014 104462      TRAP    T$ERCODE
(5) 025016 000021      .WORD    17
(5) 025020 013177      .WORD    EM52
(5) 025022 014244      .WORD    ERRO
1761 025024      7$:
1762
1763
1764 025024      ENDSEG      ;%%END OF SEGMENT%%
(3) 025024      10000$:
(3) 025024 104005      EMT      C$ESEG
1765
1766
1767 025026      ENDTST      ;**END OF TEST**
(3) 025026      L10043:
(3) 025026 104001      EMT      C$ETST
1768
1769      .SBTTL  **TEST 15** - READ FUNCTION
1770
1771 025030      BGNST      ;**START OF TEST**
1772
1773 025030      STARS
(2)      ;*****
1774      ;CHECK OF THE READ FUNCTION. WE WILL FIRST DO A READ
1775      ;HEADER TO FIND OUT WHERE WE ARE AND THEN ISSUE
1776      ;A FULL SECTOR READ, WAIT FOR READY AND CHECK FOR
1777      ;ANY ERRORS
1778 025030      STARS
(2)      ;*****
1779
1780
1781 025030 004737 021356      JSR      PC,HDHOME      ;HEADS OVER TRACK 0
1782 025034      CKERFG      ;HEADS GO HOME OKAY
(4) 025042 104032      EMT      C$EXIT
(4) 025044 000064      .WORD    L10044-.
1783
1784 025046      BGNSEG      ;%%START OF SEGMENT%%
(3) 025046 104004      EMT      C$BSEG
1785
1786 025050 012737 001750 002160      MOV      #1000.,TMP0
1787 025056 005077 155164      CLR      @RLDA      ;LOAD DISK ADDRESS
1788 025062 012777 177600 155160      MOV      #-128.,@RLMP      ;SET WORD LENGTH
1789 025070 012777 003052 155146      MOV      #BUF,@RLBA      ;SET BUS ADDRESS
1790
1791 025076 004537 020456      JSR      R5,LDFUNC      ;LOAD THE FUNCTION IN NEXT WORD
1792 025102 000014      READ      ;READ
1793 025104 004537 021276      JSR      R5,WTCRDY      ;WAIT FOR CONTROLLER READY
1794 025110      ESCAPE      SEG      ;CHECK FOR FL:LOE, ELSE EXIT SEG
(3) 025110 104010      EMT      C$ESCAPE
(3) 025112 000014      .WORD    10000$-.
1795
1796 025114 004537 020214      JSR      R5,CHERR      ;CHECK CNTLR FOR ERRORS
1797
1798 025120 005337 002160      DEC      T1,0
1799 025124 001354      BNE      1$
  
```

```

1800 025126          ENDSEG                      ;%%END OF SEGMENT%%
(3) 025126          10000$: EMT C$ESEG
(3) 025126 104005    ENDTST                      ;**END OF TEST**
1801 025130          L10044: EMT C$ETST
(3) 025130 104001
1802
1803 .SBTTL **TEST 16** - READ FUNCTION INTERRUPT
1804
1805 025132          BGNST                      ;**START OF TEST**
1806
1807 025132          STARS
(2) ;*****
1808 ;CHECK OF THE READ FUNCTION UNDER INTERRUPT CONTROL, WE WILL
1809 ;ISSUE A READ HEADER TO GET POSITION AND THEN READ
1810 ;A FULL SECTOR WAITING FOR THE INTERRUPT. CHECK FOR
1811 ;ERRORS ON INTERRUPT.
1812 025132          STARS
(2) ;*****
1813
1814
1815 025132 004737 021356 JSR PC,HDHOME          ;HEADS OVER TRACK 0
1816 025136          CKERFG                      ;HEADS GO HOME OKAY
(4) 025144 104032    EMT C$EXIT
(4) 025146 000106    .WORD L10045-.
1817
1818 025150          BGNSEG                      ;%%START OF SEGMENT%%
(3) 025150 104004    EMT C$BSEG
1819
1820 025152 005037 002144 CLR INTFLG             ;CLEAR INTERRUPT INDICATOR
1821 025156 005077 155064 CLR @RLDA             ;SET DISK ADDRESS
1822 025162 012777 177600 155060 MOV #-128.,@RLMP ;SET UP WORD COUNT
1823 025170 012777 003052 155046 MOV #BUF,@RLBA ;SET UP BUS ADDRESS
1824
1825 025176          SETPRI #PRI00              ;PRIORITY TO 0
(3) 025176 012700 000000 MOV #PRI00,R0
(3) 025202 104041    EMT C$SPRI
1826 025204 004537 020456 JSR P5,LDFUNC        ;LOAD THE FUNCTION IN NEXT WORD
1827 025210 000114    READ.INTEN              ;READ UNDER INTERRUPT
1828 025212 004537 021276 JSR R5,WTCRDY        ;WAIT FOR INTERRUPT
1829 025216          CKLOOP
(3) 025216 104C06    EMT C$CLP1
1830 025220          SETPRI #PRI07              ;PRIORITY TO 7
(3) 025220 012700 000340 MOV #PRI07,R0
(3) 025224 104041    EMT C$SPRI
1831
1832 025226 005737 002144 TST INTFLG            ;DID INTERRUPT OCCUR?
1833 025232 001004    BNE 1$                  ;YES-BRANCH NO-REPORT
1834
1835 025234          ERRDF 19.,EM4,ERR0         ;READ DID NOT INTERRUPT
(3) 025234 104462    TRAP T$ERCODE
(5) 025236 000023    .WORD 19
(5) 025240 010607    .WORD EM4
(5) 025242 014244    .WORD ERR0
1836 025244          1$: CKLOOP
(3) 025244 104006    EMT C$CLP1

```

```

1837
1838 025246 004537 020214      JSR      R5,CHERR      ;CHECK CNTLR FOR ERRORS
1839
1840 025252      ENDSEG      ;%%END OF SEGMENT%%
(3) 025252      10000$:
(3) 025252 104005      EMT      C$ESEG
1841 025254      ENDTST      ;**END OF TEST**
(3) 025254      L10045:
(3) 025254 104001      EMT      C$ETST
1842
1843      .SBTTL  **TEST 17** - CHECK READ NPR DIRECTION
1844
1845 025256      BGNSTST      ;**START OF TEST**
1846
1847 025256      STARS
(2)
1848      ;*****
1849      ;CHECK THAT THE READ FUNCTION ACTUALLY READS (INTO MEMORY)
1850      ;WE WILL WRITE A PATTERN INTO MEMORY AND THEN ISSUE
1851      ;A READ TO OVERLAY THAT PATTERN. AFTER THE READ
1852      ;WE CHECK TO SEE IF THE WRITTEN PATTERN HAS CHANGED.
1853      ;IF NOT WE ISSUE IT AGAIN AT THE SAME SECTION AFTER
1854      ;HAVING MODIFIED OUR PATTERN IN MEMORY (SINCE THERE IS
1855      ;ONE CHANCE THAT THE DISK COULD HAVE OUR PATTERN). AFTER
1856      ;THE SECOND READ WE CHECK THE BUFFER AGAIN. IF IT'S
1857      ;NO CHANGED WE REPORT AN ERROR
1858      STARS
(2)
1859      ;*****
1860 025256 004737 021356      JSR      PC,HDHOME      ;HEADS OVER TRACK 0
1861 025262      CKERFG      ;HEADS GO HOME OKAY
(4) 025270 104032      EMT      C$EXIT
(4) 025272 000156      .WORD    L10046-.
1862
1863 025274      BGNSEG      ;%%START OF SEGMENT%%
(3) 025274 104004      EMT      C$BSEG
1864
1865 025276 012737 123456 002160      MOV      #123456,TMP0      ;SET PATTERN TO WRITE
1866 025304 005037 002162      CLR      TMP1      ;CLEAR PASS INDICATOR
1867 025310 012700 003052      1$:      MOV      #BUF,R0      ;SET UP BUFFER BEGINNING
1868 025314 012701 000200      MOV      #128,R1
1869 025320 013720 002160      2$:      MOV      TMP0,(R0)+      ;WRITE BUFFER
1870 025324 005301      DEC      R1      ;DONE??
1871 025326 001374      BNE      2$      ;NO, GO BACK
1872 025330 005077 154712      CLR      @RLDA      ;LOAD DISK ADDRESS
1873 025334 012777 177600 154706      MOV      #-128,@RLMP      ;SET WORD COUNT
1874 025342 012777 003052 154674      MOV      #BUF,@RLBA      ;LOAD BUS ADDRESS
1875 025350 012737 003052 002166      MOV      #BUF,GDDAT      ;FOR ERROR PRINT
1876
1877 025356 004537 020456      JSR      R5,LDFUNC      ;LOAD THE FUNCTION IN NEXT WORD
1878 025362 000014      READ      ;READ
1879 025364 004537 021276      JSR      R5,WTCRDY      ;WAIT FOR CONTROLLER READY
1880 025370      ESCAPE      SEG      ;CHECK FOR FL:LUE, ELSE EXIT SEG
(3) 025370 104010      EMT      C$ESCAPE
(3) 025372 000054      .WORD    10000$-.
1881

```

```

ASSEMBLY ROUTINES MACY11 30A(1052) 22-NOV-78 15:35 PAGE 1-47 M 6
CZRLBB.P11 22-NOV-78 15:28 **TEST 17** - CHECK READ NPR DIRECTION SEQ 0077

1882 025374 004537 020214 JSR R5,CHERR ;CHECK CNTLR FOR ERRORS
1883 025400 ESCAPE SEG ;CHECK FOR FL:LOE, ELSE EXIT SEG
(3) 025400 104010 EMT C$ESCAPE
(3) 025402 000044 .WORD 10000$-.

1884
1885 025404 012702 003052 MOV #BUF,R2 ;SET TO START COMPARING DATA
1886 025410 022237 002160 4$: CMP (R2)+,TMP0 ;DID DATA CHANGE?
1887 025414 001014 BNE 6$ ;YES, CHECK FOR END
1888
1889
1890 ;DATA DIDN'T CHANGE, CHECK
1891 025416 005737 002162 TST TMP1 ;IF 1ST OR 2ND TIME?
1892 025422 001005 BNE $ ;2ND-REPORT 1ST-TRY AGAIN
1893
1894 025424 005237 002162 INC TMP1 ;INC PASS COUNT
1895 025430 005137 002160 COM TMP0 ;COMPLIMENT PATTERN
1896 025434 000725 BR 1$ ;GO DO IT AGAIN
1897
1898 025436 5$: ERRDF 20,EM5,ERR9 ;READ DID NOT MODIFY MEMORY
(3) 025436 104462 TRAP T$ERRCODE
(5) 025440 000024 .WORD 20
(5) 025442 010646 .WORD EM5
(5) 025444 014636 .WORD ERR9

1899
1900 025446 6$:
1901
1902 025446 ENDSEG ;%%END OF SEGMENT%%
(3) 025446 10000$: EMT C$ESEG
(3) 025446 104005
1903 025450 ENDTST ;**END OF TEST**
(3) 025450 L10046: EMT C$ETST
(3) 025450 104001
1904
1905 .SBTTL **TEST 18** - PROPER INCREMENT OF RLBA ON READ
1906
1907 025452 BGNST ;**START OF TEST**
1908
1909 025452 STARS
(2) ;*****
1910 ;CHECK THAT THE RLBA WILL INCREMENT WITH THE READ
1911 ;THE RLBA SHOULD CONTAIN 'BUF +256.' AFTER A FULL SECTOR
1912 ;READ.
1913 025452 STARS
(2) ;*****
1914
1915
1916 025452 004737 021356 JSR PC,HDHOME ;HEADS OVER TRACK 0
1917 025456 CKERFG ;HEADS GO HOME OKAY
(4) 025464 104032 EMT C$EXIT
(4) 025466 000116 .WORD L10047-.

1918
1919 025470 BGNSEG ;%%START OF SEGMENT%%
(3) 025470 104004 EMT C$BSEG
1920
1921 025472 005077 154550 CLR @R1,DA ;SET UP DISK ADDRESS
1922 025476 012777 003052 154540 MOV #BUF,@RLBA ;SET UP BUS ADDRESS

```

```

ASSEMBLY ROUTINES          MACY11 30A(1052) 22-NOV-78 15:35 PAGE 1-48 N 6
CZRLBB.P11 22-NOV-78 15:28 **TEST 18** - PROPER INCREMENT OF RLBA ON READ          SEQ 0078

1923 025504 012777 177600 154536      MOV      #-128.,@RLMP      ;WORD COUNT
1924 025512 012737 003052 002166      MOV      #BJF,GDDAT      ;FORM EXPECTED BUS ADDRESS
1925 025520 062737 000400 002166      ADD      #256.,GDDAT      ;AFTER READ
1926
1927 025526 004537 020456              JSR      R5,LDFUNC      ;LOAD THE FUNCTION IN NEXT WORD
1928 025532 000014                    READ      ;READ
1929 025534 004537 021276              JSR      R5,WTCRDY      ;WAIT FOR CONTROLLER READY
1930 025540                    ESCAPE SEG      ;CHECK FOR FL:LOE, ELSE EXIT SEG
(3) 025540 104010                    EMT      C$ESCAPE
(3) 025542 000040                    .WORD    10000$-.
1931
1932 025544 004537 020214              JSR      R5,CHERR      ;CHECK CNTLR FOR ERRORS
1933 025550                    ESCAPE SFG      ;CHECK FOR FL:LOE, ELSE EXIT SEG
(3) 025550 104010                    EMT      C$ESCAPE
(3) 025552 000030                    .WORD    10000$-.
1934 025554 013737 002230 002170      MOV      E.BA,BDDAT      ;READ 'RLBA' FOR PRESENT ADDRESS
1935 025562 023737 002170 002166      CMP      BDDAT,GDDAT      ;DID 'BA' INCREMENT PROPERLY?
1936 025570 001404                    BEQ      1$      ;YES, CONTINUE
1937
1938 025572                    ERRDF 21.,EM6,ERR4 ;BA DID NOT INCREMENT PROPERLY
(3) 025572 104462                    TRAP    T$ERRCODE
(5) 025574 000025                    .WORD    21
(5) 025576 010712                    .WORD    EM6
(5) 025600 014410                    .WORD    ERR4
1939
1940 025602                    1$:
1941
1942 025602                    ENDSEG      ;%%END OF SEGMENT%%
(3) 025602                    10000$:
(3) 025602 104005                    EMT      C$ESEU
1943 025604                    ENDTST      ;**END OF TEST**
(3) 025604 L10047:
(3) 025604 104001                    EMT      C$ETST
1944
1945                    .SBTTL **TEST 19** - PROPER INCREMENT OF RLDA ON READ
1946
1947 025606                    BGN1ST      ;**START OF TEST**
1948
1949 025606                    STARS
(2)
1950                    ;:*****
1951                    ;CHECK THAT THE RLDA INCREMENTS BY ONE AFTER A
1952                    ;FULL SECTOR READ, WE FIRST READ A HEADER TO FIND
1953                    ;OUT WHERE WE ARE, THEN ISSUE A READ AFTER
1954                    ;THE READ THE RLDA SHOULD BE RLDA (START) + 1
1955                    STARS
(2)
1956                    ;:*****
1956 025606 004737 021356              JSR      PC,HDHOME      ;HEADS OVER TRACK 0
1957 025612                    CKERFG      ;HEADS GO HOME OKAY
(4) 025620 104032                    EMT      C$EXIT
(4) 025622 000114                    .WORD    L10050-.
1958
1959 025624                    BGNSEG      ;%%START OF SEGMENT%%
(3) 025624 104014                    EMT      S$SEI.
1960
1961

```

```

ASSEMBLY ROUTINES          MACY11 30A(1052) 22-NOV-78 15:35 PAGE 1-49      B 7
CZRLBB.P11 22-NOV-78 15:28  **TEST 19** - PROPER INCREMENT OF RLDA ON READ      SEQ 0079

1962 025626 005037 002166      CLR      GDDAT
1963 025632 013777 002166 154406  MOV      GDDAT,@RLDA      ;SETUP DISK ADDRESS
1964 025640 005237 002166      INC      GDDAT      ;CREATE EXPECTED SECTOR
1965 025644 012777 177600 154376  MOV      #-128,@RLMP      ;WORD COUNT
1966 025652 012777 003052 154364  MOV      #BUF,@RLBA      ;SETUP BUS ADDRESS
1967
1968 025660 004537 020456      JSR      R5,LDFUNC      ;LOAD THE FUNCTION IN NEXT WORD
1969 025664 000014      READ      ;READ
1970 025666 004537 021276      JSR      R5,WTCRDY      ;WAIT FOR CONTROLLER READY
1971 025672      ESCAPE SEG      ;CHECK FOR FL:LOE, ELSE EXIT SEG
(3) 025672 104010      EMT      C$ESCAPE
(3) 025674 000040      .WORD    10000$-.
1972
1973 025676 004537 020214      JSR      R5,CHERR      ;CHECK CNTLR FOR ERRORS
1974 025702      ESCAPE SEG      ;CHECK FOR FL:LOE, ELSE EXIT SEG
(3) 025702 104010      EMT      C$ESCAPE
(3) 025704 000030      .WORD    10000$-.
1975
1976 025706 013737 002232 002170  MOV      E.DA,BDDAT      ;READ DISK ADDRESS
1977 025714 023737 002166 002170  CMP      GDDAT,BDDAT      ;DID SECTOR INCREMENT PROPERLY
1978 025722 001404      BEQ      1$      ;YES, BRANCH NO, REPORT ERROR
1979
1980 025724      ERRDF 22,EM7,ERR4      ;DISK ADDRESS DID NOT INCREMENT
(3) 025724 104462      TRAP   T$ERRCODE
(5) 025726 000026      .WORD    22
(5) 025730 010766      .WORD    EM7
(5) 025732 014410      .WORD    ERR4
1981
1982 025734      1$:
1983
1984 025734      ENDSEG      ;%%END OF SEGMENT%%
(3) 025734      10000$:
(3) 025734 104005      EMT      C$ESEG
1985 025736      ENDTST      ;**END OF TEST**
(3) 025736      L10050:
(3) 025736 104001      EMT      C$ETST
1986
1987      .SBTTL  **TEST 20** - FORCE HEADER NOT FOUND WITH READ
1988
1989 025740      BGNST      ;**START OF TEST**
1990
1991 025740      STARS
(2)      ;*****
1992      ;FORCE HEADER NOT FOUND ERROR TO OCCUR. THIS IS DONE
1993      ;BY SETTING SECTOR 40 OF THE RLDA AND ISSUING A
1994      ;READ. SECTOR 40 DOES NOT EXIST ON THE RL01 PACK
1995      ;THEREFORE HDR NT FOUND SHOULD SET.
1996 025740      STARS
(2)      ;*****
1997
1998 025740 004737 021356      JSR      PC,HDHOME      ;HEADS OVER TRACK 0
1999 025744      CKERFG      ;HEADS GO HOME OKAY
(4) 025752 104032      EMT      C$EXIT
(4) 025754 000102      .WORD    L10051-.
2000
2001 025756      BGNSEG      ;%%START OF SEGMENT%%

```

```

(3) 025756 104004 EMT C$BSEG
2002
2003
2004 025760 012777 000050 154260 MOV #40, @RLDA ;INSURE NOT TO FIND HEADER BY
2005 025766 012777 003052 154250 MOV #BUF, @RLBA ;SETTING SECTOR 40 OF CYL. ADDR.
2006 025774 012777 177777 154246 MOV #-1, @RLMP ;WORD COUNT
2007
2008 026002 004537 020456 JSR R5, LDFUNC ;LOAD THE FUNCTION IN NEXT WORD
2009 026006 000014 READ ;READ
2010 026010 004537 021276 JSR R5, WTCRDY ;WAIT FOR CONTROLLER READY
2011 026014 ESCAPE SEG ;CHECK FOR FL:LOE, ELSE EXIT SEG
(3) 026014 104010 EMT C$ESCAPE
(3) 026016 000036 .WORD 10000$.
2012
2013 026020 013737 002226 002160 MOV E, CS, TMP0 ;GET RLCS
2014 026026 042737 001777 002160 BIC #1777, TMP0 ;SAVE ERROR BITS
2015 026034 022737 112000 002160 CMP #BIT15!BIT12!BIT10, TMP0 ;HDR NOT FOUND SET.
2016 026042 001404 BEQ 1$ ;YES, CONTINUE
2017
2018 026044 ERRDF 23, EM10, ERRO ;HEADER NOT FOUND WOULD NOT SET
(3) 026044 104462 TRAP T$ERCODE
(5) 026046 000027 .WORD 23
(5) 026050 011043 .WORD EM10
(5) 026052 014244 .WORD ERRO
2019
2020 026054 1$:
2021 ;
2022
2023 026054 ENDSEG ;%%END OF SEGMENT%%
(3) 026054 10000$:
(3) 026054 104005 EMT C$ESEG
2024 026056 ENDTST ;**END OF TEST**
(3) 026056 L10051:
(3) 026056 104001 EMT C$E*ST
2025
2026 .SBTTL **TEST 21** - FORCE HEADER NOT FOUND WITH READ INTERRUPT
2027
2028 026060 BGNST ;**START OF TEST**
2029
2030
2031 026060 STARS
(2) ;*****
2032 ;TEST THAT HEADER NOT FOUND ERROR WILL GENERATE AN INTERRUPT
2033 ;ON OCCURANCE. HEADER NOT FOUND WILL BE FORCED BY SETTING
2034 ;SECTOR 40 OF RLDA AND ISSUING A READ
2035 026060 STARS
(2) ;*****
2036
2037
2038 026060 004737 021356 JSR PC, HDHOME ;HEADS OVER TRACK 0
2039 026064 CKERFG ;HEADS GO HOME OKAY
(4) 026072 104032 EMT C$EXIT
(4) 026074 000142 .WORD L10052-.
2040
2041 026076 BGNSEG ;%%START OF SEGMENT%%
(3) 026076 104004 EMT C$BSEG
  
```

```

2042
2043 026100          SETPRI #PRI00
      (3) 026100 012700 000000      MOV #PRI00,R0
      (3) 026104 104041          EMT C$SPRI
2044 026106 005037 002144          CLR INTFLG          ;CLEAR INTERRUPT OCCURANCE FLAG
2045 026112 012777 000050 154126      MOV #40.,@RLDA      ;INSURE NOT TO FIND HEADER BY
2046 026120 012777 003052 154116      MOV #BUF,@RLBA      ;SETTING SECTOR 40 OF CYL. ADDR.
2047 026126 012777 177777 154114      MOV #-1,@RLMP      ;WORD COUNT
2048
2049 026134 004537 020456          JSR R5,LDFUNC          ;LOAD THE FUNCTION IN NEXT WORD
2050 026140 000114          READ!INTEN          ;READ
2051 026142 004537 021276          JSR R5,WTCRDY          ;WAIT FOR CONTROLLER READY
2052 026146          CKLOOP
      (3) 026146 104006          EMT C$CLP1
2053 026150          SETPRI #PRI07
      (3) 026150 012700 000340      MOV #PRI07,R0
      (3) 026154 104041          EMT C$SPRI
2054
2055 026156 005737 002144          TST INTFLG          ;DID INTERRUPT OCCUR
2056 026162 001004          BNE 2$          ;YES
2057
2058 026164          ERRDF 24.,EM43,ERRO          ;HNF DID NOT INTERRUPT
      (3) 026164 104462          TRAP T$ERCODE
      (5) 026166 000030          .WORD 24
      (5) 026170 012723          .WORD EM43
      (5) 026172 014244          .WORD ERRO
2059
2060 026174          2$: ESCAPE SEG          ;CHECK FOR FL:LOE, ELSE EXIT SEG
      (3) 026174 104010          EMT C$ESCAPE
      (3) 026176 000036          .WORD 10000$-.
2061
2062
2063 026200 013737 002226 002160      MOV E.CS,TMP0          ;GET RLCS
2064 026206 042737 001777 002160      BIC #1777,TMP0          ;SAVE ERROR BITS
2065 026214 022737 112000 002160      CMP #BIT15.BIT12.BIT10,TMP0 ;WDR NOT FOUND SET.
2066 026222 001404          BEQ 1$          ;YES, CONTINUE
2067
2068 026224          ERRDF 25.,EM10,ERRO
      (3) 026224 104462          TRAP T$ERCODE
      (5) 026226 000031          .WORD 25
      (5) 026230 011043          .WORD EM10
      (5) 026232 014244          .WORD ERRO
2069
2070 026234          1$: ;WHEN FORCED
2071
2072 026234          ENDSEG          ;%%END OF SEGMENT%%
      (3) 026234 10000$:
      (3) 026234 1040C5          EMT C$ESEG
2073 026236          ENDTST          ;**END OF TEST**
      (3) 026236 L10052:
      (3) 026236 104001          EMT C$ETST
2074
2075          .SBTTL **TEST 22** - CHECK HEADER COMPARE LOGIC
2076
2077 026240          BGNTEST          ;**START OF TEST**
2078

```

```

2079 026240      STARS
      (2)      :*****
2080      :CHECK THE HEADER COMPARE LOGIC WORKS.  UP TO THIS POINT WE
2081      :KNOW THAT THE LOGIC FUNCTIONS PROPERLY BUT NOW WE WILL
2082      :CHECK ALL THE BITS IN THE HEADER WORD.  FOUR PATTERNS
2083      :ARE USED A WALKING 1, GROWING 1, WALKING 0, GROWING 0.  A SEEK
2084      :IS ISSUED BEFORE EACH READ TO INSURE WE ARE ON THE PROPER
2085      :TRACK.  ONCE WE ARE ON THE RIGHT TRACKWE LOAD THE RLDA
2086      :AND ISSUE THE READ.  UPON COMPLETION WE WILLCHECK FOR ERRORS
2087      :WE THEN LOAD THE COMPLIMENT PATTERN INTO THE RLDA
2088      :EXPECTING A HEADER NOT FOUND TO SET
2089 026240      STARS
      (2)      :*****
2090
2091
2092 026240 004737 021356      JSR      PC,HDHOME      :HEADS OVER TRACK 0
2093 026244      CKERFG      :HEADS GO HOME OKAY
      (4) 026252 104032      EMT      C$EXIT
      (4) 026254 000540      .WORD    L10053-.
2094
2095 026256      BGNSEG      :%%START OF SEGMENT%%
      (3) 026256 104004      EMT      C$BSEG
2096
2097 026260      SETPRI      #PRI07      :PRIORITY TO 7
      (3) 026260 012700 000340      MOV      #PRI07,R0
      (3) 026264 104041      EMT      C$SPRI
2098 026266 012703 002504      MOV      #HDRTAB,R3      :GET LIST START
2099
2100 026272      BGNSEG      :%%START OF SEGMENT%%
      (3) 026272 104004      EMT      C$BSEG
2101 026274      1$:
2102 026274 004537 020456      JSR      R5,LDFUNC      :LOAD THE FUNCTION IN NEXT WORD
2103 026300 000010      RDHDR      :READ HEADER
2104 026302 004537 021276      JSR      R5,WTCRDY      :WAIT FOR CONTROLLRE READY
2105 026306      ESCAPE      SEG      :CHECK FOR FL:LOE, ELSE EXIT SEG
      (3) 026306 104010      EMT      C$ESCAPE
      (3) 026310 000500      .WORD    10001$-.
2106
2107 026312 004537 020214      JSR      R5,CHERR      :CHECK CNTLR FOR ERRORS
2108 026316      ESCAPE      SEG      :CHECK FOR FL:LOE, ELSE EXIT SEG
      (3) 026316 104010      EMT      C$ESCAPE
      (3) 026320 000470      .WORD    10001$-.
2109 026322 013737 002234 002162      MOV      E.MP,TMP1      :READ AND SAVE HEADER
2110
2111 026330 042737 000177 002162      BIC      #177,TMP1      :CLEAR OUT SECTOR AND H.S.
2112 026336 012777 000001 153702      MOV      #1,@RLDA      :SETUP MARKER FOR SEEK
2113 026344 011337 002164      MOV      (R3),TMP2      :GET HEADER PATTERN
2114 026350 042737 000177 002164      BIC      #177,TMP2      :CLEAR OUT SECTOR AND H.S.
2115 026356 163737 002162 002164      SUB      TMP1,TMP2      :CALCULATE DIFFERENCE TO SEEK
2116 026364 103404      BCS      2$      :BRANCH FOR SEEK OUT
2117 026366 052777 000004 153652      BIS      #SIGN,@RLDA      :SEEK TOWARDS SPINDLE
2118 026374 000402      BR      3$      :GO PUT IN DIFFERENCE WORD
2119 026376 005437 002164      NFG      TMP2      :WE HAVE TO NEGATE DIFFERENCE
2120 026402 053777 002164 153636      BIS      TMP2,@RLDA      :SET IN DIFFERENCE WORD
2121 026410 032713 000100      BIT      #RHS,(R3)      :DO WE WANT HEAD SELECT AS 0?
2122 026414 001403      BFG      4$      :YES, SKIP OVER SETTING H.S.

```

|      |        |        |        |        |             |              |                                  |
|------|--------|--------|--------|--------|-------------|--------------|----------------------------------|
| 2123 | 026416 | 052777 | 000020 | 153622 | BIS         | #DAHS,@RLDA  | ;SET HEAD SELECT TO ONE          |
| 2124 | 026424 | 004537 | 020456 |        | JSR         | R5,LDFUNC    | ;LOAD THE FUNCTION IN NEXT WORD  |
| 2125 | 026430 | 000006 |        |        | SEEK        |              | ;SEEK                            |
| 2126 |        |        |        |        |             |              |                                  |
| 2127 |        |        |        |        |             |              |                                  |
| 2128 | 026432 | 004537 | 021276 |        | JSR         | R5,WTCRDY    | ;WAIT FOR CONTROLLER READY       |
| 2129 | 026436 |        |        |        | ESCAPE      | SEG          | ;CHECK FOR FL:LOE, ELSE EXIT SEG |
| (3)  | 026436 | 104010 |        |        | EMT         | C\$ESCAPE    |                                  |
| (3)  | 026440 | 000350 |        |        | .WORD       | 10001\$-     |                                  |
| 2130 |        |        |        |        |             |              |                                  |
| 2131 | 026442 | 004537 | 020214 |        | JSR         | R5,CHERR     | ;CHECK CNTLR FOR ERRORS          |
| 2132 | 026446 |        |        |        | ESCAPE      | SEG          | ;CHECK FOR FL:LOE, ELSE EXIT SEG |
| (3)  | 026446 | 104010 |        |        | EMT         | C\$ESCAPE    |                                  |
| (3)  | 026450 | 000340 |        |        | .WORD       | 10001\$-     |                                  |
| 2133 |        |        |        |        |             |              |                                  |
| 2134 | 026452 | 004537 | 021236 |        | JSR         | R5,WTCRDY    | ;WAIT FOR DRIVE READY            |
| 2135 | 026456 |        |        |        | ESCAPE      | SEG          | ;CHECK FOR FL:LOE, ELSE EXIT SEG |
| (3)  | 026456 | 104010 |        |        | EMT         | C\$ESCAPE    |                                  |
| (3)  | 026460 | 000330 |        |        | .WORD       | 10001\$-     |                                  |
| 2136 | 026462 | 004537 | 020456 |        | JSR         | R5,LDFUNC    | ;LOAD THE FUNCTION IN NEXT WORD  |
| 2137 | 026466 | 000010 |        |        | RDHDR       |              | ;READ HEADER (VERIFY SEEK)       |
| 2138 | 026470 | 004537 | 021276 |        | JSR         | R5,WTCRDY    | ;WAIT FOR CONTROLLER READY       |
| 2139 | 026474 |        |        |        | ESCAPE      | SEG          | ;CHECK FOR FL:LOE, ELSE EXIT SEG |
| (3)  | 026474 | 104010 |        |        | EMT         | C\$ESCAPE    |                                  |
| (3)  | 026476 | 000312 |        |        | .WORD       | 10001\$-     |                                  |
| 2140 |        |        |        |        |             |              |                                  |
| 2141 | 026500 | 004537 | 020214 |        | JSR         | R5,CHERR     | ;CHECK CNTLR FOR ERRORS          |
| 2142 | 026504 |        |        |        | ESCAPE      | SEG          | ;CHECK FOR FL:LOE, ELSE EXIT SEG |
| (3)  | 026504 | 104010 |        |        | EMT         | C\$ESCAPE    |                                  |
| (3)  | 026506 | 000302 |        |        | .WORD       | 10001\$-     |                                  |
| 2143 |        |        |        |        |             |              |                                  |
| 2144 | 026510 | 013737 | 002234 | 002170 | MOV         | E.MP,BDDAT   | ;READ HEADER                     |
| 2145 | 026516 | 043737 | 002150 | 002170 | BIC         | SECMSK,BDDAT | ;SAVE CYLINDER FOR COMPARE       |
| 2146 | 026524 | 011337 | 002166 |        | MOV         | (R3),GDDAT   | ;GET EXPECTED HEADER             |
| 2147 | 026530 | 043737 | 002150 | 002166 | BIC         | SECMSK,GDDAT | ;SAVE CYLINDER FOR COMPARE       |
| 2148 | 026536 | 023737 | 002166 | 002170 | CMF         | GDDAT,BDDAT  | ;SEEK END UP OKAY                |
| 2149 | 026544 | 001404 |        |        | BEQ         | \$           | ;YES, CONTINUE                   |
| 2150 |        |        |        |        |             |              |                                  |
| 2151 | 026546 |        |        |        | ERRDF       | 27,EM11,ERR4 | ;SEEK INCORRECT                  |
| (3)  | 026546 | 104462 |        |        | TRAP        | T\$ERCODE    |                                  |
| (5)  | 026550 | 000033 |        |        | .WORD       | 27           |                                  |
| (5)  | 026552 | 011110 |        |        | .WORD       | EM11         |                                  |
| (5)  | 026554 | 014410 |        |        | .WORD       | ERR4         |                                  |
| 2152 |        |        |        |        |             |              |                                  |
| 2153 | 026556 |        |        |        | 5\$: ESCAPE | SEG          | ;CHECK FOR FL:LOE, ELSE EXIT SEG |
| (3)  | 026556 | 104010 |        |        | EMT         | C\$ESCAPE    |                                  |
| (3)  | 026560 | 000230 |        |        | .WORD       | 10001\$-     |                                  |
| 2154 |        |        |        |        |             |              |                                  |
| 2155 | 026562 | 011377 | 153460 |        | MOV         | (R3),@RLDA   | ;SET UP DISK ADDRESS             |
| 2156 | 026566 | 042777 | 000077 | 153452 | BIC         | #77,@RLDA    |                                  |
| 2157 | 026574 | 012777 | 177777 | 153446 | MOV         | #-1,@RLMP    | ;WORD COUNT                      |
| 2158 | 026602 | 012777 | 003052 | 153434 | MOV         | #BUF,@RLBA   | ;BUS ADDRESS                     |
| 2159 |        |        |        |        |             |              |                                  |
| 2160 | 026610 | 004537 | 020456 |        | JSR         | R5,LDFUNC    | ;LOAD THE FUNCTION IN NEXT WORD  |
| 2161 | 026614 | 000014 |        |        | READ        |              | ;READ                            |
| 2162 | 026616 | 004537 | 021276 |        | JSR         | R5,WTCRDY    | ;WAIT FOR CONTROLLER READY       |

```

2163 026622          ESCAPE SEG          ;CHECK FOR FL:LOE, ELSE EXIT SEG
(3) 026622 104010    EMT C$ESCAPE
(3) 026624 000164    .WORD 10001$-.
2164
2165 026626 004537 020214 JSR R5,CHERR          ;CHECK CNTLR FOR ERRORS
2166 026632          ESCAPE SEG          ;CHECK FOR FL:LOE, ELSE EXIT SEG
(3) 026632 104010    EMT C$ESCAPE
(3) 026634 000154    .WORD 10001$-.
2167
2168 026636 011377 153404 MOV (R3),@RLDA          ;SET UP DISK ADDRESS AS
2169 026642 005177 153400 COM @RLDA          ;COMPLIMENT TO CAUSE HDR NT FND
2170 026646 012777 177777 153374 MOV #-1,@RLMP          ;WORD COUNT
2171 026654 012777 003052 153362 MOV #BUF,@RLBA          ;BUS ADDRESS
2172
2173 026662 004537 020456 JSR R5,LDFUNC          ;LOAD THE FUNCTION IN NEXT WORD
2174 026666 000014 READ          ;READ
2175 026670 004537 021276 JSR R5,WTCRDY          ;WAIT FOR CONTROLLER READY
2176 026674          ESCAPE SEG          ;CHECK FOR FL:LOE, ELSE EXIT SFG
(3) 026674 104010    EMT C$ESCAPE
(3) 026676 000112    .WORD 10001$-.
2177 026700 013737 002226 002160 MOV E,CS,TMP0          ;GET CS
2178 026706 042737 001777 002160 BIC #1777,TMP0          ;SAVE ERROR BITS
2179 026714 022737 112000 002160 CMP #BIT15:BIT12:BIT10,TMP0 ;DID HEADER NOT FOUND SET
2180 026722 001402 BEQ 8$          ;YES, CONTINUE
2181 026724 004537 020214 JSR R5,CHERR
2182 026730          8$: CKLOOP
(3) 026730 104006    EMT C$CLP1
2183
2184 026732 022737 112000 002160 CMP #BIT15:BIT12:BIT10,TMP0
2185 026740 001413 BEQ 6$
2186
2187 026742 011337 002166 MOV (R3),GDDAT          ;SET UP DATA FOR ERROR
2188 026746 013737 002166 002170 MOV GDDAT,BDDAT          ;PRINT OUT
2189 026754 005137 002170 COM BDDAT
2190
2191 026760 ERRDF 28,EM12,ERR4 ;HDR NOT FOUND WOULD NOT SET
(3) 026760 104462 TRAP T$ERRCODE
(5) 026762 000034 .WORD 28
(5) 026764 011137 .WORD EM12
(5) 026766 014410 .WORD ERR4
2192
2193 026770          6$: ESCAPE SEG          ;CHECK FOR FL:LOE, ELSE EXIT SEG
(3) 026770 104010    EMT C$ESCAPE
(3) 026772 000016    .WORD 10001$-.
2194
2195 026774 005723 TST (R3)+          ;GET NEXT PATTERN
2196 026776 020327 002660 CMP R3,#HDREND          ;AT END?
2197 027002 001402 BEQ 7$          ;YES, EXIT TEST
2198 027004 000137 026274 JMP 1$          ;NO, GO BACK
2199
2200          7$:
2201 027010          10001$: ENDSEG          ;%%END OF SEGMENT%%
(3) 027010          EMT C$ESEG
(3) 027010 104005
2202
2203 027012          ENDSEG          ;%%END OF SEGMENT%%

```

```

(3) 027012 10000$:
(3) 027012 104005 EMT C$ESEG
2204 027014 ENDTST ;**END OF TEST**
(3) 027014 L10053:
(3) 027014 104001 EMT C$ETST
2205
2206 .SBTTL **TEST 23** - CHECK MULTIPLE SECTORS ON READ
2207
2208 027016 BGNST ;**START OF TEST**
2209
2210 027016 STARS
(2) ;*****
2211 ;VERIFY THAT MULTIPLE SECTORS CAN BE READ, WE WILL CHECK
2212 ;THAT THE RLDA INCREMENTS PROPERLY.
2213 027016 STARS
(2) ;*****
2214
2215
2216 027016 004737 021356 JSR PC,HDHOME ;HEADS OVER TRACK 0
2217 027022 CKERFG ;HEADS GO HOME OKAY
(4) 027030 104032 EMT C$EXIT
(4) 027032 000156 .WORD L10054-.
2218
2219
2220 027034 005037 002160 CLR TMP0 ;CLEAR LOCATIONS
2221 027040 005037 002162 CLR TMP1
2222
2223 027044 BGNSEG ;%%START OF SEGMENT%%
(3) 027044 104004 EMT C$BSEG
2224
2225 027046 1$:
2226 027046 013737 002162 002166 MOV TMP1,GDDAT ;GET CYLINDER
2227 027054 053737 002160 002166 BIS TMP0,GDDAT ;GET SECTOR
2228 027062 013777 002166 153156 MOV GDDAT,RLDA ;SET DISK ADDRESS-SECTOR 0
2229 027070 062737 000002 002166 ADD #2,GDDAT ;SET EXPECTED + 2
2230 027076 012777 003052 153140 MOV #BUF,RLBA ;SET BUS ADDRESS
2231 027104 012777 177577 153136 MOV #-129,RLMP ;WORD COUNT-SECTOR+1 WORD
2232
2233 027112 004537 020456 JSR RS,LDFUNC ;LOAD THE FUNCTION IN NEXT WORD
2234 027116 000014 READ ;READ
2235 027120 004537 021276 JSR RS,WTCRDY ;WAIT FOR CONTROLLER READY?
2236 027124 ESCAPE SEG ;CHECK FOR FL:LOE, ELSE EXIT SEG
(3) 027124 104010 EMT C$ESCAPE
(3) 027126 000060 .WORD 10000$-.
2237
2238 027130 004537 020214 JSR RS,CHERR ;CHECK CNTLR FOR ERRORS
2239 027134 ESCAPE SEG ;CHECK FOR FL:LOE, ELSE EXIT SEG
(3) 027134 104010 EMT C$ESCAPE
(3) 027136 000050 .WORD 10000$-.
2240
2241 027140 013737 002232 002170 MOV E,DA,BUDAT ;READ DISK ADDRESS
2242 027146 023737 002170 002166 CMP BDDAT,GDDAT ;IS DISK ADDRESS CORRECT
2243 027154 001404 BFO 2$ ;YES, BRANCH NO, REPORT ERROR
2244
2245 027156 ERROF 2$,EM14,ERR4 ;DA DID NOT INCREMENT
(3) 027156 104462 TRAP T$ERCODE
  
```

```

(5) 027160 000035      .WORD 29
(5) 027162 011230      .WORD EM14
(5) 027164 014410      .WORD ERR4
2246
2247 027166          2$:  ESCAPE SEG          ;CHECK FOR FL:LOE, ELSE EXIT SEG
(3) 027166 104010      EMT C$ESCAPE
(3) 027170 000016      .WORD 10000$-.
2248
2249 027172 005237 002160 INC TMP0          ;NEXT SECTOR?
2250 027176 022737 000046 002160 CMP #46,TMP0 ;DONE?
2251 027204 001320      BNE 1$             ;NO, GO BACK
2252
2253
2254 027206          ENDSEG                  ;%%END OF SEGMENT%%
(3) 027206          10000$:
(3) 027206 104005      EMT C$ESEG
2255 027210          LNDTST                  ;**END OF TEST**
(3) 027210          L10054:
(3) 027210 104001      EMT C$ETST
2256 027212          STARS
(2)
2257          ;*****
2258          ;CHECK THAT WE CAN FORCE A HEADER NOT FOUND AT THE
2259          ;END OF A TRACK DOING A MULTIPLE SECTOR READ. WE
2260          ;SET UP TO READ TWO SECTORS STARTING AT SECTOR 39
2261          ;WE SHOULD TRANSFER 128 WORDS THEN ABORT WITH A
2262 027212          ;HEADER NOT FOUND FOR SECTOR 40
(2)          STARS
2263          ;*****
2264
2265          .SBTTL **TEST 24** - FORCE HDR NT FND AT END OF TRACK
2266
2267 027212          BGNTST                  ;**START OF TEST**
2268
2269
2270 027212 004737 021356 JSR PC,HDHOME      ;HEADS OVER TRACK 0
2271 027216          CKERFG                ;HEADS GO HOME OKAY
(4) 027224 104032      EMT C$EXIT
(4) 027226 000126      .WORD L10055-.
2272
2273 027230          BGNSEG                  ;%%START OF SEGMENT%%
(3) 027230 104004      EMT C$BSEG
2274
2275 027232 012737 000047 002166 MOV #39,GDDAT ;CREATE LAST SECTOR
2276 027240 013777 002166 153000 MOV GDDAT,@RLDA ;LOAD DISK ADDRESS
2277 027246 012777 177577 152774 MOV #-129,@RLMP ;WORD COUNT
2278 027254 012777 003052 152762 MOV #BUF,@RLBA ;BUS ADDRESS
2279 027262 004537 020456 JSR R5,LDFUNC ;LOAD THE FUNCTION IN NEXT WORD
2280 027266 000014      READ                ;READ
2281 027270 004537 021276 JSR R5,WTCRDY ;WAIT FOR CONTROLLER READY
2282 027274          ESCAPE SEG            ;CHECK FOR FL:LOE, ELSE EXIT SEG
(3) 027274 104010      EMT C$ESCAPE
(3) 027276 000054      .WORD 10000$-.
2283
2284 027300 013737 002226 002170 MOV E,CS,BDDAT ;READ CS
2285 027306 042737 001777 002170 BIC #1777,BDDAT ;SAVE ERROR BITS
  
```

```

ASSEMBLY ROUTINES MACY11 30A(1052) 22-NOV-78 15:35 PAGE 1-57
CZRLBB.P11 22-NOV-78 15:28 **TEST 24** - FORCE HDR NT FND AT END OF TRACK
2286 027314 022737 112000 002170 CMP #112000,BDDAT ;HDR NOT FOUND SET?
2287 027322 001402 BEQ 4$ ;YES, CONTINUE
2288 027324 004537 020214 JSR R5,CHERR
2289 027330 4$: CKLOOP
(3) 027330 104006 EMT C$CLP1
2290
2291 027332 022737 112000 002170 CMP #112000,BDDAT
2292 027340 001404 BEQ 1$
2293
2294 027342 ERRDF 30,EM23,ERRO ;HEADER NOT FOUND DID NOT SET
(3) 027342 104462 TRAP T$ERCODE
(5) 027344 000036 .WORD 30
(5) 027346 011636 .WORD EM23
(5) 027350 014244 .WORD ERRO
2295
2296 027352 1$:
2297
2298 027352 ENDSEG ;%%END OF SEGMENT%%
(3) 027352 10000$:
(3) 027352 104005 EMT C$ESFG
2299 027354 ENDTST ;**END OF TEST**
(3) 027354 L'0055:
(3) 027354 104001 EMT C$ETST
2300
2301 .SBTTL **TEST 25** - FORCE NON-EXISTANT MEMORY ERROR
2302
2303 027356 BGNSTST ;**START OF TEST**
2304
2305
2306 027356 STARS
(2) ;*****
2307 ;FORCE A NON-EXISTANT MEMORY ERROR,
2308 ;WE SET THE R.LBA TO EQUAL THE
2309 ;LAST ADDRESS IN MEMORY AND ISSUE A READ. THE
2310 ;READ SHOULD ABORT AFTER ONE WORD TRANSFERRED
2311 027356 STARS
(2) ;*****
2312
2313
2314 027356 004737 021356 JSR PC,HDHOME ;HEADS OVER TRACK 0
2315 027362 CKERFG ;HEADS GO HOME OKAY
(4) 027370 104032 EMT C$EXIT
(4) 027372 000076 .WORD L10056-.
2316
2317 027374 BGNSEG ;%%START OF SEGMENT%%
(3) 027374 104004 EMT C$BSEG
2318
2319
2320
2321
2322 027376 012777 177774 152640 MOV #177774,@RLBA ;LEAD BA
2323 027404 012737 000060 002262 MOV #BA16!BA17,XMEM ;SET EA BIT
2324 027412 005077 152630 CLR @RLDA ;LOAD DISK AVAILABLE
2325 027416 012777 177600 152624 MOV #-128,@RLMP ;WORD COUNT
2326 027424 004537 020456 JSR R5,LDFUNC ;LOAD THE FUNCTION IN NEXT WORD
2327 027430 000014 READ ;READ

```

SEQ 0087

```

ASSEMBLY ROUTINES MACY11 30A(1052) 22-NOV-78 15:35 PAGE 1-58 K 7
CZRLBB.P11 22-NOV-78 15:28 **TEST 25** - FORCE NON-EXISTANT MEMORY ERROR SEQ 0088

2328 027432 004537 021276 JSR R5,WTCRDY ;WAIT FOR CONTROLLER
2329 027436 ESCAPE SEG ;CHECK FOR FL:LOE, ELSE EXIT SEG
(3) 027436 104010 EMT C$ESCAPE
(3) 027440 000026 .WORD 10000$-.

2330
2331 027442 0 737 020000 002226 BIT #NXM,E.CS ;DID NXM SET?
2332 027450 ( 004 BNE 3$ ;YES, CONTINUE
2333
2334 027452 ERRDF 31.,EM24,ERRO ;NXM DID NOT SET
(3) 027452 104462 TRAP T$ERCODE
(5) 027454 000037 .WORD 31
(5) 027456 011716 .WORD EM24
(5) 027460 014244 .WORD ERRO

2335
2336 027462 3$: ESCAPE SEG ;CHECK FOR FL:LOE, ELSE EXIT SEG
(3) 027462 104010 EMT C$ESCAPE
(3) 027464 000002 .WORD 10000$-.

2337
2338
2339
2340
2341 027466 ENDSEG ;%%END OF SEGMENT%%
(3) 027466 10000$:
(3) 027466 104005 EMT C$ESEG
2342 027470 ENDTST ;**END OF TEST**
(3) 027470 L10056:
(3) 027470 104001 EMT C$ETST

2343
2344 .SBTTL **TEST 26** - FORCE NON-EXISTANT MEMORY ERROR INTERRUPT
2345
2346 027472 BGNST ;**START OF TEST**
2347 027472 STARS
(2)
2348 ;*****
2349 ;CHECK THAT WE CAN FORCE AN INTERRUPT WITH A
2350 027472 ;NON-EXISTANT MEMORY ERROR.
(2) STARS
2351 ;*****
2352
2353 027472 004737 021356 JSR PC,HDHOME ;HEADS OVER TRACK 0
2354 027476 CKERFG ;HEADS GO HOME OKAY
(4) 027504 104032 EMT C$EXIT
(4) 027506 000140 .WORD L10057-.

2355
2356 027510 BGNSEG ;%%START OF SEGMENT%%
(3) 027510 104004 EMT C$BSEG

2357
2358 027512 005037 002144 CLR INTFLG ;CLEAR INTERRUPT OCCURANCE FLAG
2359
2360
2361
2362 027516 SETPRI #PRI00
(3) 027516 012700 000000 MOV #PRI00,R0
(3) 027522 104041 EMT C$SPRI
2363 027524 012777 177774 152512 MOV #177774,@RLBA ;PRELOAD BA
2364 027532 012737 000060 002262 MOV #BA16.BA17,XMEM ;SET EA BITS

```

```

ASSEMBLY ROUTINES MACY11 30A(1052) 22-NOV-78 15:35 PAGE 1-59 L 7
CZRLBB.P11 22-NOV-78 15:28 **TEST 26** - FORCE NON-EXISTANT MEMORY ERROR INTERRUPT SEQ 0089

2365 027540 005077 152502 CLR @RLDA ;LOAD DA
2366 027544 012777 177777 152476 MOV #-1,@RLMP ;WORD COUNT
2367 027552 004537 020456 JSR R5,LDFUNC ;LOAD THE FUNCTION IN NEXT WORD
2368 027556 000114 READ!INTEN ;READ
2369 027560 004537 021276 JSR R5,WTCRDY ;WAIT FOR CONTROLLER
2370 027564 SETPRI #PRI07 ;PRIORITY TO 7
(3) 027564 012700 000340 MOV #PRI07,R0
(3) 027570 104041 EMT C$SPRI
2371 027572 ESCAPE SEG ;CHECK FOR FL:LOE, ELSE EXIT SEG
(3) 027572 104010 EMT C$ESCAPE
(3) 027574 000050 .WORD 10000$-.
2372
2373 027576 005737 002144 TST INTFLG ;INTERRUPT OCCUR?
2374 027602 001004 BNE 4$ ;YES OKAY
2375
2376 027604 ERRDF 32,EM44,ERRO ;NO INTERRUPT W/NXM
(3) 027604 104462 TRAP T$ERCODE
(5) 027606 000040 .WORD 32
(5) 027610 012767 .WORD EM44
(5) 027612 014244 .WORD ERRO
2377
2378 027614 4$: ESCAPE SEG ;CHECK FOR FL:LOE, ELSE EXIT SEG
(3) 027614 104010 EMT C$ESCAPE
(3) 027616 000026 .WORD 10000$-.
2379
2380 027620 032737 020000 002226 BIT #NXM,E.CS ;DID NXM SET?
2381 027626 001004 BNE 3$ ;YES, CONTINUE
2382
2383 027630 ERRDF 33,EM24,ERRO ;NO NXM
(3) 027630 104462 TRAP T$ERCODE
(5) 027632 000041 .WORD 33
(5) 027634 011716 .WORD EM24
(5) 027636 014244 .WORD ERRO
2384
2385 027640 3$: ESCAPE SEG ;CHECK FOR FL:LOE, ELSE EXIT SEG
(3) 027640 104010 EMT C$ESCAPE
(3) 027642 000002 .WORD 10000$-.
2386
2387
2388 027644 ENDSEG ;%%END OF SEGMENT%%
(3) 027644 10000$: EMT C$ESEG
(3) 027644 104005
2389 027646 ENDTST ;**END OF TEST**
(3) 027646 L10057: EMT C$ETST
(3) 027646 104001
2390
2391 .SBTTL **TEST 27** - CHECK READ WRITE LOOP
2392
2393 027650 BGNTST ;**START OF TEST**
2394
2395 027650 STARS
(2)
2396 ;*****
2397 ;VERIFY THAT THE WRITE ACTUALLY WRITES. AT THIS
2398 ;TIME WE KNOW THAT THE WRITE FUNCTION GOES THRU
2399 ;THE MOTIONS BUT WE DON'T KNOW THAT THE DATA
;ACTUALLY GETS RECORDED ON THE PLATTER.

```

```

2400 027650          STARS
      (2)          ;:*****
2401
2402
2403 027650 004737 021356      JSR      PC,HDHOME      ;HEADS OVER TRACK 0
2404 027654          CKERFG          ;HEADS GO HOME OKAY
      (4) 027662 104032          EMT      C$EXIT
      (4) 027664 000362          .WORD   L10060-.
2405
2406 027666          BGNSEG          ;%%START OF SEGMENT%%
      (3) 027666 104004          EMT      C$BSEG
2407
2408 027670 012700 003052      MOV      #BUF,R0          ;SET UP WRITE BUFFER
2409 027674 012701 000200      MOV      #128.,R1         ;128 WORDS/ONE SECTOR
2410 027700 012720 125252      3$:    MOV      #125252,(R0)+ ;WRITE PATTERN TO BUFFER
2411 027704 005301          DEC      R1                 ;DONE?
2412 027706 001374          BNE      3$                 ;NO, BRANCH BACK
2413 027710 005077 152332      CLR      @RLDA          ;DISK ADDRESS
2414 027714 012777 177600 152326 MOV      #-128.,@RLMP      ;WORD COUNT
2415 027722 012777 003052 152314 MOV      #BUF,@RLBA      ;BUS ADDRESS
2416 027730 004537 020456      JSR      R5,LDFUNC        ;LOAD THE FUNCTION IN NEXT WORD
2417 027734 000012          WRITE     ;WRITE THE PATTERN
2418 027736 004537 021276      JSR      R5,WTCRDY       ;WAIT FOR CONTROLLER READY
2419 027742          ESCAPE SEG          ;CHECK FOR FL:LOE, ELSE EXIT SEG
      (3) 027742 104010          EMT      C$ESCAPE
      (3) 027744 000300          .WORD   10000$-.
2420
2421 027746 004537 020214      JSR      R5,CHERR          ;CHECK CNTLR FOR ERRORS
2422 027752          ESCAPE SEG          ;CHECK FOR FL:LOE, ELSE EXIT SEG
      (3) 027752 104010          EMT      C$ESCAPE
      (3) 027754 000270          .WORD   10000$-.
2423 027756          BGNSEG          ;%%START OF SEGMENT%%
      (3) 027756 104004          EMT      C$BSEG
2424 027760 012700 003052      MOV      #BUF,R0          ;CLEAR OUT BUFFER BEFORE
2425 027764 012701 000200      MOV      #128.,R1         ;READING
2426 027770 005020          CLR      (R0)+              ;CLEAR BUFFER
2427 027772 005301          DEC      R1                 ;DONE?
2428 027774 001375          BNE      4$                 ;NO, BRANCH BACK
2429
2430 027776 005077 152244      CLR      @RLDA          ;LOAD DISK ADDRESS
2431 030002 012777 177600 152240 MOV      #-128.,@RLMP      ;WORD COUNT/ONE SECTION
2432 030010 012777 003052 152226 MOV      #BUF,@RLBA      ;LOAD BUS ADDRESS
2433 030016 004537 020456      JSR      R5,LDFUNC        ;LOAD THE FUNCTION IN NEXT WORD
2434 030022 000014          READ          ;GO READ
2435 030024 004537 021276      JSR      R5,WTCRDY       ;WAIT FOR CONTROLLER READY
2436 030030          ESCAPE SEG          ;CHECK FOR FL:LOE, ELSE EXIT SEG
      (3) 030030 104010          EMT      C$ESCAPE
      (3) 030032 000210          .WORD   10001$-.
2437
2438 030034 004537 020214      JSR      R5,CHERR          ;CHECK CNTLR FOR ERRORS
2439 030040 005737 002124      TST      T.CRC          ;WAS ERROR A DCK??
2440 030044 001003          BNE      8$                 ;YES,SEE IF WE A DUMP
2441 030046          ESCAPE SEG          ;CHECK FOR FL:LOE, ELSE EXIT SEG
      (3) 030046 104010          EMT      C$ESCAPE
      (3) 030050 000172          .WORD   10001$-.
2442 030052 000404          BR      99$

```

;SKIP AROUND

```

2443 030054 005737 016774      8$:  TST      T.DMP      ;DO WE STILL WANT TO CHECK IT
2444 030060 001772              BEQ      10$          ;NO
2445 030062              CKLOOP          ;YES, CHECK FOR LOOP FIRST
(3) 030062 104006              EMT      C$CLP1
2446
2447 030064 005037 002130      99$:  CLR      CDCNT      ;CLEAR NUMBER WE'RE TO PRINT
2448 030070 005037 002122          CLR      CHECK      ;ALLOW HEADER ON FIRST PRINT
2449 030074 012702 003052          MOV      #BUF,R2      ;COMPARE BUFFER TO CHECK WRITE
2450 030100 012701 000200          MOV      #128,R1      ;128 WORDS
2451 030104 012737 125252 002166  MOV      #125252,GDDAT ;SET UP EXPECTED
2452 030112 011237 002170      5$:  MOV      (R2),BDDAT ;GET DATA
2453 030116 023737 002166 002170  CMP      GDDAT,BDDAT ;IS DATA OKAY
2454 030124 001442              BEQ      6$          ;YES, CONTINUE
2455 030126 010237 002162          MOV      R2,TMP1      ;LOAD BAD MEM LOCATION
2456 030132 023737 002130 016776  CMP      CDCNT,T.LMT ;CHECKED ENOUGH??
2457 030140 001002              BNE      333$         ;NO
2458 030142              ESCAPE SEG      ;CHECK FOR FL:LOE, ELSE EXIT SEG
(3) 030142 104010              EMT      C$ESCAPE
(3) 030144 000076              .WORD    10001$-
2459 030146 005237 002130      333$: INC      CDCNT      ;ACCOUNT FOR IT
2460
2461 030152 005737 002122          TST      CHECK      ;HEADER OR JUST DATA
2462 030156 001007              BNE      9$          ;JUST DATA
2463 030160              ERRDF  34,EM25,ERR8 ;BAD DATA
(3) 030160 104462              TRAP     T$ERRCODE
(5) 030162 000042              .WORD    34
(5) 030164 011774              .WORD    EM25
(5) 030166 014564              .WORD    ERR8
2464 030170 005237 002122          INC      CHECK      ;ACCOUNT FOR PRINT OF HEADER
2465 030174 000416              BR       6$
2466
2467 030176      9$:  PRINTB #FRMT6,TMP1,GDDAT,BDDAT
(10) 030176 013746 002170          MOV      BDDAT,-(SP)
(9) 030202 013746 002166          MOV      GDDAT,-(SP)
(8) 030206 013746 002162          MOV      TMP1,-(SP)
(7) 030212 012746 016042          MOV      #FRMT6,-(SP)
(6) 030216 012746 000004          MOV      #4,-(SP)
(3) 030222 010600          MOV      SP,R0
(4) 030224 104014          EMT      C$PNTB
(4) 030226 062706 000012          ADD      #12,SP
2468
2469 030232      6$:  CKLOOP
(3) 030232 104006              EMT      C$CLP1
2470 030234 005722      7$:  TST      (R2)+      ;BUMP BUFFER POINTER
2471 030236 005301              DEC      R1          ;DONE?
2472 030240 001324              BNE      5$          ;NO, GO BACK
2473 030242              ENDSEG      ;%%END OF SEGMENT%%
(3) 030242      10001$: EMT      C$SEG
(3) 030242 104005              ENDSEG      ;%%END OF SEGMENT%%
2474 030244      10000$: EMT      C$SEG
(3) 030244 104005              ENDSEG      ;**END OF TEST**
2475 030246
(3) 030246
(3) 030246

```

```

2477 .SBTTL **TEST 28** - CHECK SILO LINES
2478
2479 030250 BCNTST ;**START OF TEST**
2480
2481
2482
2483 030250 STARS
(2) ;*****
2484 ;TEST THAT LINES IN / TO SILO ARE GOOD, THAT IS THAT EACH LINE IS
2485 ;GOOD AND CAN BE AT EITHER A 1 OR A 0 STATE INDEPENDENTLY OF EACH
2486 ;OTHER BIT POSITION THIS IS DONE BY WRITING PATTERNS OF FLOATING 1,
2487 ;FLOATING 0, WALKING 0, WALKING 1
2488 030250 STARS
(2) ;*****
2489
2490
2491 030250 004737 021356 JSR PC,HDHOME ;HEADS OVER TRACK 0
2492 030254 CKERFG ;HEADS GO HOME OKAY
(4) 030262 104032 EMT C$EXIT
(4) 030264 000404 .WORD L10061-.
2493
2494 030266 012703 002662 MOV #DATPAT,R3
2495
2496
2497 030272 BGNSEG ;%%START OF SEGMENT%%
(3) 030272 104004 EMT C$BSEG
2498 030274 012700 003052 6$: MOV #BUF,R0 ;WRITE PATTERN INTO MEMORY
2499 030300 012701 000200 MOV #128,R1 ;128 WORDS
2500 030304 011320 2$: MOV (R3),(R0)+ ;WRITE THE PATTERN
2501 030306 005301 DEC R1 ;DONE?
2502 030310 001375 BNE 2$ ;NO GO BACK
2503
2504 030312 012777 003052 151724 MOV #BUF,@RLBA ;SETUP TO WRITE PATTERN ONTO DISK
2505 030320 005077 151722 CLR @RLDA ;LOAD DA
2506 030324 012777 177600 151716 MOV #-128,@RLMP ;WORD COUNT
2507 030332 004537 020456 JSR R5,LDFUNC ;LOAD THE FUNCTION IN NEXT WORD
2508 030336 000012 WRITE
2509 030340 004537 021276 JSR R5,WTCRDY
2510 030344 ESCAPE SEG ;CHECK FOR FL:LOE, ELSE EXIT SEG
(3) 030344 104010 EMT C$ESCAPE
(3) 030346 000320 .WORD 10000$-.
2511 030350 004537 020214 JSR R5,CHERR ;CHECK CNTLR FOR ERRORS
2512 030354 ESCAPE SEG ;CHECK FOR FL:LOE, ELSE EXIT SEG
(3) 030354 104010 EMT C$ESCAPE
(3) 030356 000310 .WORD 10000$-.
2513 030360 BGNSEG ;%%START OF SEGMENT%%
(3) 030360 104004 EMT C$BSEG
2514 030362 012700 003052 MOV #BUF,R0 ;CLEAR MEMORY BEFORE READING IT BACK
2515 030366 012701 000200 MOV #128,R1 ;128 WORDS
2516 030372 005020 3$: CLR (R0)+ ;CLEAR
2517 030374 005301 DEC R1 ;EONE
2518 030376 001375 BNE 3$ ;NO
2519
2520 030400 012777 003052 151636 MOV #BUF,@RLBA ;SETUP TO READ IT BACK
2521 030406 012777 177600 151634 MOV #-128,@RLMP ;128 WORDS
2522 030414 005077 151626 CLR @RLDA ;SECTOR ZERO

```

|      |        |        |        |        |        |                            |                                  |
|------|--------|--------|--------|--------|--------|----------------------------|----------------------------------|
| 2523 | 030420 | 004537 | 020456 |        | JSR    | R5, LDFUNC                 | :LOAD THE FUNCTION IN NEXT WORD  |
| 2524 | 030424 | 000014 |        |        | READ   |                            |                                  |
| 2525 | 030426 | 004537 | 021276 |        | JSR    | R5, WTCRDY                 |                                  |
| 2526 | 030432 |        |        |        | ESCAPE | SEG                        | :CHECK FOR FL:LOE, ELSE EXIT SEG |
| (3)  | 030432 | 104010 |        |        | EMT    | C\$ESCAPE                  |                                  |
| (3)  | 030434 | 000224 |        |        | .WORD  | 10001\$-                   |                                  |
| 2527 | 030436 | 004537 | 020214 |        | JSR    | R5, CHERR                  | :CHECK CNTLR FOR ERRORS          |
| 2528 | 030442 | 005737 | 002124 |        | TST    | T.CRC                      | :WAS ERROR A DCK??               |
| 2529 | 030446 | 001003 |        |        | BNE    | 8\$                        | :YES, SEE IF WE A DUMP           |
| 2530 | 030450 |        |        | 0\$:   | ESCAPE | SEG                        | :CHECK FOR FL:LOE, ELSE EXIT SEG |
| (3)  | 030450 | 104010 |        |        | EMT    | C\$ESCAPE                  |                                  |
| (3)  | 030452 | 000206 |        |        | .WORD  | 10001\$-                   |                                  |
| 2531 | 030454 | 000404 |        |        | BR     | 99\$                       | :SKIP AROUND                     |
| 2532 | 030456 | 005737 | 016774 | 8\$:   | TST    | T.DMP                      | :DO WE STILL WANT TO CHECK IT    |
| 2533 | 030462 | 001772 |        |        | BEQ    | 10\$                       | :NO                              |
| 2534 | 030464 |        |        |        | CKLOOP |                            | :YES, CHECK FOR LOOP FIRST       |
| (3)  | 030464 | 104006 |        |        | EMT    | C\$CLP1                    |                                  |
| 2535 |        |        |        |        |        |                            |                                  |
| 2536 | 030466 | 005037 | 002130 | 99\$:  | CLR    | CDCNT                      | :CLEAR NUMBER WE'RE TO PRINT     |
| 2537 | 030472 | 005037 | 002122 |        | CLR    | CHECK                      | :ALLOW HEADER ON FIRST PRINT     |
| 2538 | 030476 | 011337 | 002166 |        | MOV    | (R3), GDDAT                | :COMPARE WHAT WE READ BACK       |
| 2539 | 030502 | 012737 | 003052 | 002164 | MOV    | #BUF, TMP2                 | :BUFFER START                    |
| 2540 | 030510 | 012737 | 000001 | 002162 | MOV    | #1, TMP1                   | :START WITH FIRST                |
| 2541 |        |        |        |        |        |                            |                                  |
| 2542 | 030516 | 017737 | 151442 | 002170 | 5\$:   | MOV                        | @TMP2, BDDAT                     |
| 2543 | 030524 | 023737 | 002166 | 002170 | CMP    | GDDAT, BDDAT               | :GOOD?                           |
| 2544 | 030532 | 001440 |        |        | BEQ    | 4\$                        | :YES, BRANCH                     |
| 2545 |        |        |        |        |        |                            |                                  |
| 2546 | 030534 | 023737 | 002130 | 016776 | CMP    | CDCNT, T.LMT               | :CHECKED ENOUGH??                |
| 2547 | 030542 | 001002 |        |        | BNE    | 333\$                      | :NO                              |
| 2548 | 030544 |        |        |        | ESCAPE | SEG                        | :CHECK FOR FL:LOE, ELSE EXIT SEG |
| (3)  | 030544 | 104010 |        |        | EMT    | C\$ESCAPE                  |                                  |
| (3)  | 030546 | 000112 |        |        | .WORD  | 10001\$-                   |                                  |
| 2549 | 030550 | 005237 | 002130 | 333\$: | INC    | CDCNT                      | :ACCOUNT FOR IT                  |
| 2550 |        |        |        |        |        |                            |                                  |
| 2551 | 030554 | 005737 | 002122 |        | TST    | CHECK                      | :HEADER OR JUST DATA             |
| 2552 | 030560 | 001007 |        |        | BNE    | 9\$                        | :JUST DATA                       |
| 2553 | 030562 |        |        |        | ERRDF  | 35, EM45, ERR10            | :BAD DATA BACK                   |
| (3)  | 030562 | 104462 |        |        | TRAP   | T\$ERCODE                  |                                  |
| (5)  | 030564 | 000043 |        |        | .WORD  | 35                         |                                  |
| (5)  | 030566 | 013024 |        |        | .WORD  | EM45                       |                                  |
| (5)  | 030570 | 014702 |        |        | .WORD  | ERR10                      |                                  |
| 2554 |        |        |        |        |        |                            |                                  |
| 2555 | 030572 | 005237 | 002122 |        | INC    | CHECK                      | :ACCOUNT FOR PRINT OF HEADER     |
| 2556 | 030576 | 000416 |        |        | BR     | 4\$                        |                                  |
| 2557 |        |        |        |        |        |                            |                                  |
| 2558 | 030600 |        |        | 9\$:   | PRINTB | #FRMT7, TMP1, GDDAT, BDDAT |                                  |
| (10) | 030600 | 013746 | 002170 |        | MOV    | BDDAT, -(SP)               |                                  |
| (9)  | 030604 | 013746 | 002166 |        | MOV    | GDDAT, -(SP)               |                                  |
| (8)  | 030610 | 013746 | 002162 |        | MOV    | TMP1, -(SP)                |                                  |
| (7)  | 030614 | 012746 | 016117 |        | MOV    | #FRMT7, -(SP)              |                                  |
| (6)  | 030620 | 012746 | 000004 |        | MOV    | #4, -(SP)                  |                                  |
| (3)  | 030624 | 010600 |        |        | MOV    | SP, R0                     |                                  |
| (4)  | 030626 | 104014 |        |        | EMT    | C\$PNTB                    |                                  |
| (4)  | 030630 | 062706 | 000012 |        | ADD    | #2, SP                     |                                  |
| 2559 | 030634 |        |        | 4\$:   | CKLOOP |                            |                                  |

```

(3) 030634 104006 EMT C$CLP1
2560
2561 030636 062737 000002 002164 ADD #2,TMP2 ;NEXT LOCATION
2562 030644 005237 002162 INC TMP1 ;NEXT WORD
2563 030650 023727 002162 000201 CMP TMP1,#129. ;DONE
2564 030656 001317 BNE 5$ ;NO, GO BACK
2565
2566 030660 ENDSEG ;%%END OF SEGMENT%%
(3) 030660 10001$:
(3) 030660 104005 EMT C$ESEG
2567
2568 030662 005723 TST (R3)+ ;DONE ALL PATTERNS
2569 030664 001203 BNE 6$ ;NO, GO BACK
2570
2571 030666 ENDSEG ;%%END OF SEGMENT%%
(3) 030666 10000$:
(3) 030666 104005 EMT C$ESEG
2572 030670 ENDTST ;**END OF TEST**
(3) 030670 L10061:
(3) 030670 104001 EMT C$ETST
2573
2574 .SBTTL **TEST 29** - CHECK THROUGHPUT OF SILO
2575
2576 030672 BGNSTST ;**START OF TEST**
2577
2578
2579
2580 030672 STARS
(2)
2581 ;*****
2582 ;TEST THAT THE SILO OPERATES CORRECTLY, WE WILL WRITE A PATTERN THAT CONTAINS
2583 ;A UNIQUE PATTERN IN EACH LOCATION. WE EXPECT IT BACK IN PROPER
2584 ;ORDER, WE DO A ONE SECTOR TRANSFER
2585 STARS
(2)
2586 ;*****
2587 030672 004737 021356 JSR PC,HDHOME ;HEADS OVER TRACK 0
2588 030676 CKERFG ;HEADS GO HOME OKAY
(4) 030704 104032 EMT C$EXIT
(4) 030706 000410 .WORD L10062-.
2589
2590 030710 BGNSEG ;%%START OF SEGMENT%%
(3) 030710 104004 EMT C$BSEG
2591
2592
2593 030712 012700 000001 MOV #1,R0 ;INITIAL 1
2594 030716 012701 000200 MOV #128.,R1 ;128 WORDS
2595 030722 012702 003052 MOV #BUF,R2 ;BUFFER
2596 030726 010022 2$: MOV R0,(R2)+ ;WRITE A WORD
2597 030730 005200 INC R0 ;NEXT PATTERN (1-128)
2598 030732 005301 DEC R1 ;DONE
2599 030734 001374 BNE 2$ ;NO
2600
2601 030736 012777 003052 151300 MOV #BUF,@RLBA ;SETUP TO WRITE
2602 030744 012777 177600 151276 MOV #-128.,@RLMP ;128 WORDS
2603 030752 005077 151270 CLR @RLDA ;DISK ADDRESS 0
  
```

```

2604 030756 004537 020456      JSR      R5,LDFUNC      ;LOAD THE FUNCTION IN NEXT WORD
2605 030762 000012              WRITE
2606 030764 004537 021276      JSR      R5,WTCRDY
2607 030770              ESCAPE SEG      ;CHECK FOR FL:LOE, ELSE EXIT SEG
(3) 030770 104010              EMT      C$ESCAPE
(3) 030772 000322              .WORD    10000$-.
2608
2609 030774 004537 020214      JSR      R5,CHERR      ;CHECK CNTLR FOR ERRORS
2610 031000              ESCAPE SEG      ;CHECK FOR FL:LOE, ELSE EXIT SEG
(3) 031000 104010              EMT      C$ESCAPE
(3) 031002 000312              .WORD    10000$-.
2611 031004              BGNSEG
(3) 031004 104004              EMT      C$BSEG      ;%%START OF SEGMENT%%
2612 031006 012700 003052      MOV      #BUF,R0      ;CLEAR BUFFER
2613 031012 012701 000200      MOV      #128.,R1      ;128 IN LENGTH
2614 031016 005020              CLR      (R0)+      ;CLEAR
2615 031020 005301              DEC      R1      ;DOWN COUNT
2616 031022 001375              BNE      3$      ;DONE?
2617
2618 031024 012777 003052 151212  MOV      #BUF,@RLBA      ;BUS ADDRESS
2619 031032 012777 177600 151210  MOV      #-128.,@RLMP      ;WORD COUNT
2620 031040 005077 151202              CLR      @RLDA      ;DISK ADDRESS
2621 031044 004537 020456      JSR      R5,LDFUNC      ;LOAD THE FUNCTION IN NEXT WORD
2622 031050 000014
2623 031052 004537 021276      JSR      R5,WTCRDY
2624 031056              ESCAPE SEG      ;CHECK FOR FL:LOE, ELSE EXIT SEG
(3) 031056 104010              EMT      C$ESCAPE
(3) 031060 000232              .WORD    10001$-.
2625
2626 031062 004537 020214      JSR      R5,CHERR      ;CHECK CNTLR FOR ERRORS
2627 031066 005737 002124      TST      T.CRC      ;WAS ERROR A DCK??
2628 031072 001003              BNE      8$      ;YES,SEE IF WE A DUMP
2629 031074              ESCAPE SEG      ;CHECK FOR FL:LOE, ELSE EXIT SEG
(3) 031074 104010              EMT      C$ESCAPE
(3) 031076 000214              .WORD    10001$-.
2630 031100 000404              BR      99$      ;SKIP AROUND
2631 031102 005737 016774      TST      T.DMP      ;DO WF STILL WANT TO CHECK IT
2632 031106 001772              BEQ      10$      ;NO
2633 031110              CKLOOP
(3) 031110 104006              EMT      C$CLP1      ;YES, CHECK FOR LOOP FIRST
2634
2635 031112 005037 002130      CLR      CDCNT      ;CLEAR NUMBER WE'RE TO PRINT
2636 031116 005037 002122      CLR      CHECK      ;ALLOW HEADER ON FIRST PRINT
2637 031122 012737 000001 002166  MOV      #1,GDDAT      ;START GOOD AT 1
2638 031130 012737 003052 002164  MOV      #BUF,TMP2      ;START OF BUFFER
2639 031136 012737 000001 002162  MOV      #1,TMP1      ;FIRST WORD
2640
2641 031144 017737 151014 002170 4$:  MOV      @TMP2,BDDAT      ;GET WORD
2642 031152 023737 002170 002166  CMP      BDDAT,GDDAT      ;CORRECT?
2643 031160 001440              BEQ      6$      ;YES
2644
2645 031162 023737 002130 016776  CMP      CDCNT,T.LMT      ;CHECKED ENOUGH??
2646 031170 001002              BNE      33$      ;NO
2647 031172              ESCAPE SEG      ;CHECK FOR FL:LOE, ELSE EXIT SEG
(3) 031172 104010              EMT      C$ESCAPE
(3) 031174 000116              .WORD    10001$-.

```

```

2648 031176 005237 002130 333$: INC CDCNT ;ACCOUNT FOR IT
2649
2650 031202 005737 002122 TST CHECK ;HEADER OR JUST DATA
2651 031206 001007 BNE 9$ ;JUST DATA
2652 031210 ERRDF 36,EM47,ERR10 ;BAD DATA
(3) 031210 104462 TRAP T$ERCODE
(5) 031212 000044 .WORD 36
(5) 031214 013056 .WORD EM47
(5) 031216 014702 .WORD ERR10
2653 031220 005237 002122 INC CHECK ;ACCOUNT FOR PRINT OF HEADER
2654 031224 000416 BR 6$
2655
2656 031226 9$: PRINTB #FRMT7,TMP1,GDDAT,BDDAT
(10) 031226 013746 002170 MOV BDDAT,-(SP)
(9) 031232 013746 002166 MOV GDDAT,-(SP)
(8) 031236 013746 002162 MOV TMP1,-(SP)
(7) 031242 012746 016117 MOV #FRMT7,-(SP)
(6) 031246 012746 000004 MOV #4,-(SP)
(3) 031252 010600 MOV SP,R0
(4) 031254 104014 EMT C$PNTB
(4) 031256 062706 000012 ADD #12,SP
2657 031262 6$: CKLOOP
(3) 031262 104006 EMT C$CLP1
2658
2659 031264 062737 000002 002164 ADD #2,TMP2 ;NEXT
2660 031272 005237 002162 INC TMP1 ;NEXT
2661 031276 005237 002166 INC GDDAT ;NEXT
2662 031302 023727 002162 000201 CMP TMP1,#129. ;DONE?
2663 031310 001315 BNE 4$
2664
2665 031312 ENDSEG ;%%END OF SEGMENT%%
(3) 031312 10001$: EMT C$ESEG
(3) 031312 104005
2666
2667 031314 ENDSEG ;%%END OF SEGMENT%%
(3) 031314 10000$: EMT C$ESEG
(3) 031314 104005
2668 031316 ENDTST ;**END OF TEST**
(3) 031316 L10062: EMT C$ETST
(3) 031316 104001
2669
2670 .SBTTL **TEST 30** - CHECK ZERO FILL ON WRITE
2671
2672 031320 BCNTST ;**START OF TEST**
2673
2674
2675
2676 031320 STARS
(2) ;:*****
2677 ;:WHEN WRITING PARTIAL SECTORS (LESS THAN 128 WORDS) THE
2678 ;:CONTROLLER WILL FILL IN THE REMAINING PORTION OF
2679 ;:THE SECTOR WITH ZERO WORDS. CHECK THIS FEATURE
2680 ;:WITH WORD COUNTS FROM 1 TO 127
2681 031320 STARS
(2) ;:*****
2682

```

|      |        |        |        |        |        |               |                                       |
|------|--------|--------|--------|--------|--------|---------------|---------------------------------------|
| 2683 | 031320 | 004737 | 021356 |        | JSR    | PC,HDHOME     | :HEADS OVER TRACK 0                   |
| 2684 | 031324 |        |        |        | CKERFG |               | :HEADS GO HOME OKAY                   |
| (4)  | 031332 | 104032 |        |        | EMT    | C\$EXIT       |                                       |
| (4)  | 031334 | 000442 |        |        | .WORD  | L10063-       |                                       |
| 2685 |        |        |        |        |        |               |                                       |
| 2686 | 031336 |        |        |        | BGNSEG |               | :%%START OF SEGMENT%%                 |
| (3)  | 031336 | 104004 |        |        | EMT    | C\$BSEG       |                                       |
| 2687 |        |        |        |        |        |               |                                       |
| 2688 | 031340 | 012737 | 000001 | 002162 | MOV    | #1,TMP1       | :START WITH 1 WORD WRITE              |
| 2689 | 031346 | 012700 | 003052 | 35\$:  | MOV    | #BUF,R0       | :WRITE BUFFER WITH 52525, WE'LL       |
| 2690 | 031352 | 012701 | 000200 |        | MOV    | #128,R1       | :WRITE 128 WORDS ALL THOUGH WE'RE     |
| 2691 | 031356 | 012720 | 052525 | 3\$:   | MOV    | #52525,(R0)+  | :ONLY GOING TO TRANSFER < 128         |
| 2692 | 031362 | 005301 |        |        | DEC    | R1            | :DONE WITH BUFFER?                    |
| 2693 | 031364 | 001374 |        |        | BNE    | 3\$           | :NO, GO BACK                          |
| 2694 | 031366 | 013700 | 002162 | 33\$:  | MOV    | TMP1,R0       | :GET TRANSFER WORD COUNT              |
| 2695 | 031372 | 005400 |        |        | NEG    | R0            | :NEGATE FOR RLMP                      |
| 2696 | 031374 | 010077 | 150650 |        | MOV    | R0,@RLMP      | :STORE WORD COUNT AWAY                |
| 2697 | 031400 | 012777 | 003052 | 150636 | MOV    | #BUF,@RLBA    | :SET UP RLBA                          |
| 2698 | 031406 | 005077 | 150634 |        | CLR    | @RLDA         |                                       |
| 2699 | 031412 | 004537 | 020456 |        | JSR    | R5,LDFUNC     | :LOAD THE FUNCTION IN NEXT WORD       |
| 2700 | 031416 | 000012 |        |        | WRITE  |               | :WRITE IT                             |
| 2701 | 031420 | 004537 | 021276 |        | JSR    | R5,WTCRDY     | :WAIT FOR WRITE TO FINISH             |
| 2702 | 031424 |        |        |        | ESCAPE | SEG           | :CHECK FOR FL:LOE, ELSE EXIT SEG      |
| (3)  | 031424 | 104010 |        |        | EMT    | C\$ESCAPE     |                                       |
| (3)  | 031426 | 000346 |        |        | .WORD  | 10000\$-      |                                       |
| 2703 |        |        |        |        |        |               |                                       |
| 2704 | 031430 | 004537 | 020214 |        | JSR    | R5,CHERR      | :CHECK CNTLR FOR ERRORS               |
| 2705 | 031434 |        |        |        | ESCAPE | SEG           | :CHECK FOR FL:LOE, ELSE EXIT SEG      |
| (3)  | 031434 | 104010 |        |        | EMT    | C\$ESCAPE     |                                       |
| (3)  | 031436 | 000336 |        |        | .WORD  | 10000\$-      |                                       |
| 2706 | 031440 |        |        |        | BGNSEG |               | :%%START OF SEGMENT%%                 |
| (3)  | 031440 | 104004 |        |        | EMT    | C\$BSEG       |                                       |
| 2707 | 031442 | 012700 | 003052 |        | MOV    | #BUF,R0       | :WE'RE GOING TO OVERLAY BUFFER BEFORE |
| 2708 | 031446 | 012701 | 000200 |        | MOV    | #128,R1       | :READING IT BACK.                     |
| 2709 | 031452 | 012720 | 125252 | 18\$:  | MOV    | #125252,(R0)+ | :OVERLAY IT WITH COMPLIMENT           |
| 2710 | 031456 | 005301 |        |        | DEC    | R1            | :DONE?                                |
| 2711 | 031460 | 001374 |        |        | BNE    | 18\$          | :NO, KEEP GOING                       |
| 2712 |        |        |        |        |        |               |                                       |
| 2713 | 031462 | 012777 | 003052 | 150554 | MOV    | #BUF,@RLBA    | :SET UP TO READ                       |
| 2714 | 031470 | 012777 | 177600 | 150552 | MOV    | #-128,@RLMP   | :128 WORDS TO CHECK ZERO FILL         |
| 2715 | 031476 | 005077 | 150544 |        | CLR    | @RLDA         | :SECTOR                               |
| 2716 | 031502 | 004537 | 020456 |        | JSR    | R5,LDFUNC     | :LOAD THE FUNCTION IN NEXT WORD       |
| 2717 | 031506 | 000014 |        |        | READ   |               |                                       |
| 2718 | 031510 | 004537 | 021276 |        | JSR    | R5,WTCRDY     | :WAIT TIL WE FINISH THE READ          |
| 2719 | 031514 |        |        |        | ESCAPE | SEG           | :CHECK FOR FL:LOE, ELSE EXIT SEG      |
| (3)  | 031514 | 104010 |        |        | EMT    | C\$ESCAPE     |                                       |
| (3)  | 031516 | 000234 |        |        | .WORD  | 10001\$-      |                                       |
| 2720 |        |        |        |        |        |               |                                       |
| 2721 | 031520 | 004537 | 020214 |        | JSR    | R5,CHERR      | :CHECK CNTLR FOR ERRORS               |
| 2722 | 031524 | 005737 | 002124 |        | IST    | T.CRC         | :WAS ERROR A DCK??                    |
| 2723 | 031530 | 001003 |        |        | BNE    | 8\$           | :YES,SEE IF WE A DUMP                 |
| 2724 | 031532 |        |        | 10\$:  | ESCAPE | SEG           | :CHECK FOR FL:LOE, ELSE EXIT SEG      |
| (3)  | 031532 | 104010 |        |        | EMT    | C\$ESCAPE     |                                       |
| (3)  | 031534 | 000216 |        |        | .WORD  | 10001\$-      |                                       |
| 2725 | 031536 | 000404 |        |        | BR     | 92\$          | :SKIP AROUND                          |
| 2726 | 031540 | 005737 | 016774 | 8\$:   | IST    | T.DMP         | :DO WE STILL WANT TO CHECK IT         |

```

2727 031544 001772      BEQ      10$      ;NO
2728 031546      CKLOOP      ;YES, CHECK FOR LOOP FIRST
(3) 031546 104006      EMT      C$CLP1
2729 031550 005037 002130 99$: CLR      CDCNT      ;CLEAR NUMBER WE'RE TO PRINT
2730 031554 005037 002122 CLR      CHECK      ;ALLOW HEADER ON FIRST PRINT
2731 031560 013702 002162 MOV      TMP1,R2      ;WORDS WRITTEN IN R2
2732 031564 012701 000200 MOV      #128.,R1      ;CHECK 128 WORDS
2733
2734 031570 012703 003052 MOV      #BUF,R3      ;SET UP BUFFER BEGINNING
2735 031574 005037 002164 CLR      TMP2      ;ZERO WORD COUNT
2736 031600 012737 052525 002166 MOV      #52525,GDDAT ;SET UP EXPECTED
2737 031606 011337 002170 4$: MOV      (R3),BDDAT ;GET WORD
2738 031612 023737 002170 002166 CMP      BDDAT,GDDAT ;IS WORD CORRECT?
2739 031620 001441      BEQ      12$      ;YES, GO CHECK COUNTS AND REPEAT
2740
2741 031622 023737 002130 016776 CMP      CDCNT,T.LMT      ;CHECKED ENOUGH??
2742 031630 001002      BNE      333$      ;NO
2743 031632      ESCAPE      SEG      ;CHECK FOR FL:LOE, ELSE EXIT SEG
(3) 031632 104010      EMT      C$ESCAPE
(3) 031634 000116      .WORD 10001$-.
2744 031636 005237 002130 333$: INC      CDCNT      ;ACCOUNT FOR IT
2745
2746 031642 005737 002122 TST      CHECK      ;HEADER OR JUST DATA
2747 031646 001007      BNE      9$      ;JUST DATA
2748 031650      ERRDF      37.,EM27,ERR12
(3) 031650 104462      TRAP      T$ERCODE
(5) 031652 000045      .WORD 37
(5) 031654 012105      .WORD EM27
(5) 031656 015026      .WORD ERR12
2749 031660 005237 002122 INC      CHECK      ;ACCOUNT FOR PRINT OF HEADER
2750 031664 000417      BR      12$
2751
2752 031666      9$: PRINTB #FRMT9,TMP1,R3,GDDAT,BDDAT
(11) 031666 013746 002170 MOV      BDDAT,-(SP)
(10) 031672 013746 002166 MOV      GDDAT,-(SP)
(9) 031676 010346      MOV      R3,-(SP)
(8) 031700 013746 002162 MOV      TMP1,-(SP)
(7) 031704 012746 016312 MOV      #FRMT9,-(SP)
(6) 031710 012746 000005 MOV      #5,-(SP)
(3) 031714 010600      MOV      SP,R0
(4) 031716 104014      EMT      C$PNTB
(4) 031720 062706 000014 ADD      #14,SP
2753 031724      12$: CKLOOP
(3) 031724 104006      EMT      C$CLP1
2754 031726 005723      6$: TST      (R3)+
2755 031730 005237 002164 INC      TMP2
2756 031734 005301      DEC      R1      ;DONE ALL WORDS?
2757 031736 001405      BEQ      7$      ;EXIT TEST
2758 031740 005302      DEC      R2      ;DONE CHECKING NON-ZERO WORDS
2759 031742 003321      BGT      4$      ;NO, BRANCH BACK
2760 031744 005037 002166 CLR      GDDAT      ;YES, SET EXP'D AS ZERO
2761 031750 000716      BR      4$      ;BRANCH BACK
2762
2763 031752      7$:      ;EXIT TEST
2764 031752      10001$: ENDSEG      ;%%END OF SEGMENT%%
(3) 031752

```

```

(3) 031752 104005          EMT      C$ESEG
2765
2766 031754 005237 002162      INC      TMP1
2767 031760 023727 002162 000200    CMP      TMP1,#128.
2768 031766 001402          BEQ      34$
2769 031770 000137 031346          JMP      35$
2770 031774          34$:
2771
2772 031774          ENDSEG          ;%%END OF SEGMENT%%
(3) 031774          *0000$:
(3) 031774 104005          EMT      C$ESEG
2773 031776          ENDTST          ;**END OF TEST**
(3) 031776          L10063:
(3) 031776 104001          EMT      C$ETST
2774
2775          .SBTTL  **TEST 31** - CHECK SECTOR BITS OF HEADER COMPARE
2776
2777 032000          BGNST          ;**START OF TEST**
2778
2779
2780 032000          STARS
(2)          ;*****
2781          ;TEST THAT ALL SECTOR BITS OF HEADER WORD CAN COMPARE
2782          ;UNIQUELY. WE TESTED THE HEADER COMPARE LOGIC EARLIER
2783          ;BUT THAT WAS NOT AN EXTENSIVE TEST OF THE SECTOR BITS.
2784          ;THE TEST PROCEDURE IS TO WRITE EACH SECTOR OF TRACK
2785          ;0 WITH THE SECTOR ADDRESS, THEN GO BACK AND READ
2786          ;EACH SECTOR. IF ANY SECTOR HAS ANY DATA THEN THAT
2787          ;WHICH WAS EXPECTED THEN WE HAVE AN ERROR
2788          ;ERROR PRINT OUT WILL GIVE SECTOR, EXPECTED AND RECEIVED
2789 032000          STARS
(2)          ;*****
2790
2791
2792
2793
2794
2795
2796
2797 032000 004737 021356          JSR      PC,HDHOME          ;HEADS OVER TRACK 0
2798 032004          CKERFG          ;HEADS GO HOME OKAY
(4) 032012 104032          EMT      C$EXIT
(4) 032014 000414          .WORD    L10064-.
2799
2800 032016          BGNSEG          ;%%START OF SEGMENT%%
(3) 032016 104004          EMT      C$BSEG
2801
2802 032020 005037 002160          1$:    CLR      TMP0          ;CLEAR
2803
2804 032024          BGNSEG          ;%%START OF SEGMENT%%
(3) 032024 104004          EMT      C$BSEG
2805
2806 032026 012702 003052          199$:  MOV      #BUF,R2          ;WRITE A PATTERN FOR THE WRITE
2807 032032 012701 000200          MOV      #128,R1          ;ONE SECTOR'S WORTH
2808 032036 013722 002160          2$:    MOV      T:PO,(R2)+      ;WRITE IT
2809 032042 005301          DEC      R1          ;DONE.

```

```

2810 032044 001374      BNE      2$          ;IF NOT, GO BACK
2811
2812 032046 012777 177600 150174      MOV      #-128.,@RLMP      ;ONE SECTOR WORD COUNT
2813 032054 012777 003052 150162      MOV      #BUF,@RLBA      ;WRITE FROM BUF
2814 032062 013777 002160 150156      MOV      TMP0,@RLDA      ;SECTOR
2815 032070 004537 020456      JSR      R5,LDFUNC      ;LOAD THE FUNCTION IN NEXT WORD
2816 032074 000012      WRITE
2817 032076 004537 021276      JSR      R5,WTCRDY      ;WAIT FOR WRITE TO FINISH
2818 032102      ESCAPE SEG      ;CHECK FOR FL:LOE, ELSE EXIT SEG
(3) 032102 104010      EMT      C$ESCAPE
(3) 032104 000320      .WORD    10001$-.
2819 032106 005237 002160      INC      TMP0      ;NEXT SECTOR
2820 032112 023727 002160 000050      CMP      TMP0,#40.      ;ALL DONE?
2821 032120 001342      BNE      199$      ;NO GO BACK
2822 032122 005037 002160      CLR      TMP0      ;CLEAR
2823
2824 032126      BGNSEG      ;%%START OF SEGMENT%%
(3) 032126 104004      EMT      C$BSEG
2825
2826 032130 012702 003052      98$: MOV      #BUF,R2      ;CLEAR THE BUFFER FIRST
2827 032134 012701 000200      MOV      #128.,R1      ;128 WORDS
2828 032140 005022      3$: CLR      (R2)+
2829 032142 005301      DEC      R1
2830 032144 001375      BNE      3$
2831
2832 032146 013777 002160 150072      MOV      TMP0,@RLDA      ;GET SECTOR
2833 032154 012777 003052 150062      MOV      #BUF,@RLBA      ;SETUP BUS ADDRESS
2834
2835 032162 012777 177600 150060      MOV      #-128.,@RLMP      ;READ A SECTOR
2836 032170 004537 020456      JSR      R5,LDFUNC      ;LOAD THE FUNCTION IN NEXT WORD
2837 032174 000014      READ
2838 032176 004537 021276      JSR      R5,WTCRDY
2839 032202      ESCAPE SEG      ;CHECK FOR FL:LOE, ELSE EXIT SEG
(3) 032202 104010      EMT      C$ESCAPE
(3) 032204 000216      .WORD    10002$-.
2840
2841 032206 004537 020214      JSR      R5,CHERR      ;CHECK CNTLR FOR ERRORS
2842 032212 005737 002124      TST      T.CRC      ;WAS ERROR A DCK??
2843 032216 001003      BNE      8$      ;YES,SEE IF WE A DUMP
2844 032220      10$: ESCAPE SEG      ;CHECK FOR FL:LOE, ELSE EXIT SEG
(3) 032220 104010      EMT      C$ESCAPE
(3) 032222 000200      .WORD    10002$-.
2845 032224 000404      BR      99$      ;SKIP AROUND
2846 032226 005737 016774      8$: TST      T.DMP      ;DO WE STILL WANT TO CHECK IT
2847 032232 001772      BEQ      10$      ;NO
2848 032234      CKLOOP      ;YES, CHECK FOR LOOP FIRST
(3) 032234 104006      EMT      C$CLP1
2849
2850      ;CHECK NOW TO SEE IF WE READ THE RIGHT SECTOR
2851
2852 032236 005037 002130      99$: CLR      CDCNT      ;CLEAR NUMBER WE'RE TO PRINT
2853 032242 005037 002122      CLR      CHECK      ;ALLOW HEADER ON FIRST PRINT
2854 032246 013737 002160 002166      MOV      TMP0,GDDAT      ;EXPECTED DATA
2855 032254 012702 003052      MOV      #BUF,R2      ;BUFFER
2856 032260 012701 000200      MOV      #-128.,R1      ;WORD COUNT
2857 032264 012237 002170      5$: MOV      (R2)+,BDDAT
  
```

```

ASSEMBLY ROUTINES MACY11 30A(1052) 22-NOV-78 15:35 PAGE 1-71 K 8
CZRLBB.P11 22-NOV-78 15:28 **TEST 31** - CHECK SECTOR BITS OF HEADER COMPARE SEQ 0101

2858 032270 023737 002170 002166      CMP      BDDAT,GDDAT
2859 032276 001440                      BEQ      6$
2860
2861 032300 023737 002130 016776      CMP      CDCNT,T.LMT      ;CHECKED ENOUGH??
2862 032306 001002                      BNE      333$      ;NO
2863 032310                      ESCAPE SEG      ;CHECK FOR FL:LOE, ELSE EXIT SEG
(3) 032310 104010                      EMT      C$ESCAPE
(3) 032312 000110                      .WORD    10002$-
2864 032314 005237 002130      333$: INC      CDCNT      ;ACCOUNT FOR IT
2865
2866 032320 005737 002122      TST      CHECK      ;HEADER OR JUST DATA
2867 032324 001007                      BNE      9$      ;JUST DATA
2868 032326                      ERRDF  38.,EM50,ERR11 ;
(3) 032326 104462                      TRAP    T$ERCODE
(5) 032330 000046                      .WORD    38
(5) 032332 013105                      .WORD    EM50
(5) 032334 014754                      .WORD    FRR11
2869 032336 005237 002122      INC      CHECK      ;ACCOUNT FOR PRINT OF HEADER
2870 032342 000416                      BR       6$
2871
2872 032344      9$: PRINTB #FRMT8,TMP0,GDDAT,BDDAT
(10) 032344 013746 002170      MOV      BDDAT,-(SP)
(9) 032350 013746 002166      MOV      GDDAT,-(SP)
(8) 032354 013746 002160      MOV      TMP0,-(SP)
(7) 032360 012746 016171      MOV      #FRMT8,-(SP)
(6) 032364 012746 000004      MOV      #4,-(SP)
(3) 032370 010600      MOV      SP,R0
(4) 032372 104014      EMT      C$PNTB
(4) 032374 062706 000012      ADD      #12,SP
2873 032400      6$: CKLOOP
(3) 032400 104006      EMT      C$CLP1
2874
2875 032402 005301      DEC      R1      ;ALL OF SECTOR CHECKED?
2876 032404 001327      BNE      5$      ;GO BACK IF NOT
2877 032406 005237 002160      INC      TMP0      ;NEXT SECTOR
2878 032412 023727 002160 000050      CMP      TMP0,#40.      ;DONE?
2879 032420 001243      BNE      98$      ;NO, GO BACK
2880
2881 032422      ENDSEG      ;%%END OF SEGMENT%%
(3) 032422      10002$: EMT      C$ESEG
(3) 032422 104005
2882
2883 032424      ENDSEG      ;%%END OF SEGMENT%%
(3) 032424      10001$: EMT      C$ESEG
(3) 032424 104005
2884 032426      ENDSEG      ;%%END OF SEGMENT%%
(3) 032426      10000$: EMT      C$ESEG
(3) 032426 104005
2885 032430      ENDTST      ;**END OF TEST**
(3) 032430      L10064: EMT      C$ETST
(3) 032430 104001
2886
2887      .SBTTL  **TEST 32** - WRITE CHECK NPR INTEGRITY
2888
2889 032432      BGN1ST      ;**START OF TEST**
2890

```

```

2891 032432 STARS
(2) :*****
2892 :CHECK THAT NPR WILL NOT INTERFERE WITH THE OPERATION OF THE
2893 :UNIBUS. WE SET UP LOCATION 4 TO HANDLE THE TRAP IF IT HAPPENS.
2894 032432 STARS
(2) :*****
2895 :*****
2896
2897 032432 004737 021356 JSR PC,HDHOME ;HEADS OVER TRACK 0
2898 032436 CKERFG ;HEADS GC HOME OKAY
(4) 032444 104032 EMT C$EXIT
(4) 032446 000372 .WORD L10065-.
2899
2900 032450 BGNSEG ;%%START OF SEGMENT%%
(3) 032450 104004 EMT C$BSEG
2901
2902 032452 012700 003052 MOV #BUF,R0 ;SETUP AND WRITE
2903 032456 012701 000200 MOV #128,R1 ;128 WORDS
2904 032462 012720 125252 299$: MOV #125252,(R0)+ ;WRITE
2905 032466 005301 DEC R1 ;DONE??
2906 032470 001374 BNE 299$
2907
2908 032472 012777 003052 147544 MOV #BUF,@RLBA ;LOAD BUS ADDRESS
2909 032500 012777 177600 147542 MOV #-128,@RLMP ;WORD COUNT
2910 032506 005077 147534 CLR @RLDA ;CLEAR DISK ADDRESS
2911 032512 004537 020456 JSR R5,LDFUNC ;LOAD THE FUNCTION IN NEXT WORD
2912 032516 000012 WRITE
2913 032520 004537 021276 JSR R5,WTCRDY ;WAIT FOR CONTROLLER READY
2914 032524 ESCAPE SEG ;CHECK FOR FL:LOE, ELSE EXIT SEG
(3) 032524 104010 EMT C$ESCAPE
(3) 032526 000310 .WORD 10000$-.
2915 032530 004537 020214 JSR R5,CHERR ;CHECK CNTLR FOR ERRORS
2916 032534 ESCAPE SEG ;CHECK FOR FL:LOE, ELSE EXIT SEG
(3) 032534 104010 EMT C$ESCAPE
(3) 032536 000300 .WORD 10000$-.
2917
2918
2919 ;VERIFY WRITE WITH READ BEFORE WRCHK
2920
2921 032540 005077 147502 CLR @RLDA
2922 032544 012777 003052 147472 MOV #BUF,@RLBA
2923 032552 012777 177600 147470 MOV #-128,@RLMP
2924 032560 004537 020456 JSR R5,LDFUNC ;LOAD THE FUNCTION IN NEXT WORD
2925 032564 000014 READ
2926 032566 004537 021276 JSR R5,WTCRDY
2927 032572 ESCAPE SEG ;CHECK FOR FL:LOE, ELSE EXIT SEG
(3) 032572 104010 EMT C$ESCAPE
(3) 032574 000242 .WORD 10000$-.
2928 032576 004537 020214 JSR R5,CHERR ;CHECK CNTLR FOR ERRORS
2929 032602 ESCAPE SEG ;CHECK FOR FL:LOE, ELSE EXIT SEG
(3) 032602 104010 EMT C$ESCAPE
(3) 032604 000232 .WORD 10000$-.
2930
2931 032606 BGNSEG ;%%START OF SEGMENT%%
(3) 032606 104004 EMT C$BSEG
2932

```

```

2933 032610          1$:  SETVEC  ERRVEC,#TRPHAN,#340      ;SET UP FOR TRAP
      (7) 032610 012746 000340      MOV      #340,-(SP)
      (6) 032614 012746 021350      MOV      #TRPHAN,-(SP)
      (5) 032620 013746 002132      MOV      ERRVEC,-(SP)
      (4) 032624 012746 000003      MOV      #3,-(SP)
      (3) 032630 104037          EMT      C$SVEC
      (2) 032632 062706 000010      ADD      #10,SP
2934 032636 005037 002142      CLR      TRPFLG          ;CLEAR TRAP OCCURANCE
2935 032642 012777 003052 147374      MOV      #BUF,@RLBA      ;BUS ADDRESS
2936 032650 005077 147372      CLR      @RLDA      ;LOAD DISK ADDRESS
2937 032654 012777 177600 147366      MOV      #-128,@RLMP      ;WORD COUNT OF 128
2938 032662 005037 002166      CLR      GDDAT      ;SET UP CSR TO LOAD
2939 032666 013737 002134 002166      MOV      DRIVE,GDDAT      ;SET IN DRIVE
2940 032674 052737 000002 002166      BIS      #WRCHK,GDDAT      ;SET IN FUNCTION
2941 032702 004537 020764      JSR      R5,BEFORE      ;LOAD FOR ERROR PRINTOUT
2942 032706 013737 002166 002216      MOV      GDDAT,B.CS      ;SET IN COMMAND
2943 032714 052737 000201 002216      BIS      #201,B.CS      ;LOAD CRDY
2944 032722 042737 002000 002216      BIC      #OPI,B.CS      ;CLEAR (BIT 10)
2945 032730 013777 002166 147304      MOV      GDDAT,@RLCS      ;ISSUE WRITE CHECK
2946 032736 012701 000144      MOV      #100,R1      ;WAIT FOR CRDY
2947 032742 032777 000200 147272 5$: BIT      #CRDY,@RLCS      ;NPR DONE
2948 032750 001013          BNE      6$      ;YES, 6$
2949 032752          WAITUS  #20.      ;WAIT A WHILE
      (3) 032752 012700 000024      MOV      #20,R0
      (3) 032756 104027          EMT      C$WTU
2950 032760 005301          DEC      R1
2951 032762 001367          BNE      5$      ;A WHILE UP
2952          ;NO, GO BACK
2953 032764 004537 021016      JSR      R5,AFTER
2954 032770          ERRDF  0,CRTIM,ERR5      ;CONTROLLER TIMED OUT
      (3) 032770 104462          TRAP      T$ERCODE
      (5) 032772 000000          .WORD  0
      (5) 032774 007172          .WORD  CRTIM
      (5) 032776 014456          .WORD  ERR5
2955 033000          6$:  CLRVEC  ERRVEC          ;CLEAR VECTOR
      (3) 033000 013700 002132      MOV      ERRVEC,R0
      (3) 033004 104036          EMT      C$CVEC
2956 033006          ESCAPE  SEG      ;CHECK FOR FL:LOE, ELSE EXIT SEG
      (3) 033006 104010          EMT      C$ESCAPE
      (3) 033010 000024          .WORD  10001$-
2957
2958 033012 005737 002142      TST      TRPFLG          ;DID TRAP OCCUR?
2959 033016 001406          BEQ      7$      ;NO
2960 033020 004537 021016      JSR      R5,AFTER
2961 033024          ERRSF  1,EM57,ERR0      ;TRAP ON WRITE
      (3) 033024 104461          TRAP      T$ERCODE
      (5) 033026 000001          .WORD  1
      (5) 033030 013477          .WORD  EM57
      (5) 033032 014244          .WORD  ERR0
2962 033034          7$:
2963
2964
2965 033034          10001$:  ENDSEG      ;%%END OF SEGMENT%%
      (3) 033034          EMT      C$FSEG
      (3) 033034 104005          ENDSEG      ;%%END OF SEGMENT%%
2966 033036

```

```

(3) 033036 104005 10000$: EMT C$ESEG
2967 (3) 033036 104005
2968 033040 ENDTST ;**END OF TEST**
(3) 033040 L10065: EMT C$ETST
(3) 033040 104001
2969 .SBTTL **TEST 33** - WRITE CHECK FUNCTION
2970
2971 BGNTST ;**START OF TEST**
2972 033042
2973
2974 033042 STARS
(2) ;*****
2975 ;CHECK OF WRITE CHECK LOGIC UNDER FLAG MODE
2976 ; WE WILL WRITE CHECK A FULL SECTOR (128 WORDS) FROM
2977 ;MEMORY (BUF). WE CHECK THAT NO ERRORS OCCUR.
2978 033042 STARS
(2) ;*****
2979
2980
2981 033042 004737 021356 JSR PC,HDHOME ;HEADS OVER TRACK 0
2982 033046 CKERFG ;HEADS GO HOME OKAY
(4) 033054 104032 EMT C$EXIT
(4) 033056 000214 .WORD L10066-.
2983
2984 033060 BGNSEG ;%%START OF SEGMENT%%
(3) 033060 104004 EMT C$BSEG
2985
2986 033062 012700 003052 MOV #BUF,R0 ;SETUP AND WRITE
2987 033066 012701 000200 MOV #128,R1 ;128 WORDS
2988 033072 012720 125252 299$: MOV #125252,(R0)+ ;WRITE
2989 033076 005301 DEC R1 ;DONE??
2990 033100 001374 BNE 299$
2991
2992 033102 012777 003052 147134 MOV #BUF,@RLBA ;LOAD BUS ADDRESS
2993 033110 012777 177600 147132 MOV #-128,@RLMP ;WORD COUNT
2994 033116 005077 147124 CLR @RLDA ;CLEAR DISK ADDRESS
2995 033122 004537 020456 JSR R5,LDFUNC ;LOAD THE FUNCTION IN NEXT WORD
2996 033126 000012 WRITE
2997 033130 004537 021276 JSR R5,WTCRDY ;WAIT FOR CONTROLLER READY
2998 033134 ESCAPE SEG ;CHECK FOR FL:LOE, ELSE EXIT SEG
(3) 033134 104010 EMT C$ESCAPE
(3) 033136 000132 .WORD 10000$-.
2999 033140 004537 020214 JSR R5,CHERR ;CHECK CNTLR FOR ERRORS
3000 033144 ESCAPE SEG ;CHECK FOR FL:LOE, ELSE EXIT SEG
(3) 033144 104010 EMT C$ESCAPE
(3) 033146 000122 .WORD 10000$-.
3001 033150 BGNSEG ;%%START OF SEGMENT%%
(3) 033150 104004 EMT C$BSEG
3002
3003 ;VERIFY WRITE WITH READ BEFORE WRCHK
3004
3005 033152 005077 147070 CLR @RLDA
3006 033156 012777 003052 147060 MOV #BUF,@RLBA
3007 033164 012777 177600 147056 MOV #-128,@RLMP
3008 033172 004537 020456 JSR R5,LDFUNC

```

```

3009 033176 000014      READ
3010 033200 004537 021276 JSR    R5,WTCRDY
3011 033204      ESCAPE SEG          ;CHECK FOR FL:LOE, ELSE EXIT SEG
    (3) 033204 104010 EMT    C$ESCAPE
    (3) 033206 000060 .WORD 10001$-
3012 033210 004537 020214 JSR    R5,CHERR          ;CHECK CNTLR FOR ERRORS
3013 033214      ESCAPE SEG          ;CHECK FOR FL:LOE, ELSE EXIT SEG
    (3) 033214 104010 EMT    C$ESCAPE
    (3) 033216 000050 .WORD 10001$-
3014
3015 033220      BGNSEG          ;%%START OF SEGMENT%%
    (3) 033220 104004 EMT    C$BSEG
3016
3017 033222      3$:
3018 033222 005077 147020 CLR    @RLDA
3019 033226 012777 177600 MOV    #-128,@RLMP          ;WORD COUNT
3020 033234 012777 003052 MOV    #BUF,@RLBA          ;BUS ADDRESS
3021 033242 004537 020456 JSR    R5,LDFUNC          ;LOAD THE FUNCTION IN NEXT WORD
3022 033246 000002      WRCHK          ;WRITE CHECK
3023
3024 033250 004537 021276 JSR    R5,WTCRDY          ;WAIT FOR CONTROLLER READY
3025 033254      ESCAPE SEG          ;CHECK FOR FL:LOE, ELSE EXIT SEG
    (3) 033254 104010 EMT    C$ESCAPE
    (3) 033256 000006 .WORD 10002$-
3026
3027
3028 033260 004537 020214 JSR    R5,CHERR          ;CHECK CNTLR FOR ERRORS
3029
3030 033264      ENDSEG          ;%%END OF SEGMENT%%
    (3) 033264      10002$:
    (3) 033264 104005 EMT    C$ESEG
3031 033266      10001$:
    (3) 033266 104005 EMT    C$ESEG          ;%%END OF SEGMENT%%
3032 033270      10000$:
    (3) 033270 104005 EMT    C$ESEG          ;%%END OF SEGMENT%%
3033 033272      ENDTST          ;**END OF TEST**
    (3) 033272      L10066:
    (3) 033272 104001 EMT    C$ETST
3034
3035      .SBTTL  **TEST 34** - WRITE CHECK FUNCTION INTERRUPT
3036
3037 033274      BGNST          ;**START OF TEST**
3038
3039 033274      STARS
    (2) ;*****
3040      ;CHECK OF WRITE CHECK LOGIC UNDER INTERRUPT MODE
3041      ;WE WILL WRITE CHECK A FULL SECTOR (128 WORDS) FROM MEMORY (BUF).
3042      ;WE CHECK THAT NO ERRORS OCCUR. WE DO NOT CHECK RLDA OR RLBA
3043      ;INCREMENT AT THIS TIME.
3044 033274      STARS
    (2) ;*****
3045
3046
3047 033274 004737 021356 JSR    PC,HDHOME          ;HEADS OVER TRACK 0
  
```

```

3048 033300          CKERFG          ;HEADS GO HOME OKAY
      (4) 033306 104032 EMT          C$EXIT
      (4) 033310 000252 .WORD        L10067-.
3049
3050 033312          BGNSEG          ;%%START OF SEGMENT%%
      (3) 033312 104004 EMT          C$BSEG
3051
3052 033314 012700 003052 MOV        #BUF,R0      ;SETUP AND WRITE
3053 033320 012701 000200 MOV        #128,R1      ;128 WORDS
3054 033324 012720 125252 299$: MOV      #125252,(R0)+ ;WRITE
3055 033330 005301 DEC        R1          ;DONE??
3056 033332 001374 BNE        299$
3057
3058 033334 012777 003052 146702 MOV      #BUF,@RLBA    ;LOAD BUS ADDRESS
3059 033342 012777 177600 146700 MOV      #-128,@RLMP   ;WORD COUNT
3060 033350 005077 146672 CLR        @RLDA      ;CLEAR DISK ADDRESS
3061 033354 004537 020456 JSR        R5,LDFUNC    ;LOAD THE FUNCTION IN NEXT WORD
3062 033360 000012 WRITE
3063 033362 004537 021276 JSR        R5,WTCRDY    ;WAIT FOR CONTROLLER READY
3064 033366 ESCAPE SEG          ;CHECK FOR FL:LOE, ELSE EXIT SEG
      (3) 033366 104010 EMT          C$ESCAPE
      (3) 033370 000170 .WORD        10000$-.
3065 033372 004537 020214 JSR        R5,CHERR      ;CHECK CNTLR FOR ERRORS
3066 033376 ESCAPE SEG          ;CHECK FOR FL:LOE, ELSE EXIT SEG
      (3) 033376 104010 EMT          C$ESCAPE
      (3) 033400 000160 .WORD        10000$-.
3067          ;VERIFY WRITE WITH READ BEFORE WRCHK
3068
3069 033402 005077 146640 CLR        @RLDA
3070 033406 012777 003052 146630 MOV      #BUF,@RLBA
3071 033414 012777 177600 146626 MOV      #-128,@RLMP
3072 033422 004537 020456 JSR        R5,LDFUNC    ;LOAD THE FUNCTION IN NEXT WORD
3073 033426 000014 READ
3074 033430 004537 021276 JSR        R5,WTCRDY
3075 033434 ESCAPE SEG          ;CHECK FOR FL:LOE, ELSE EXIT SEG
      (3) 033434 104010 EMT          C$ESCAPE
      (3) 033436 000122 .WORD        10000$-.
3076 033440 004537 020214 JSR        R5,CHERR      ;CHECK CNTLR FOR ERRORS
3077 033444 ESCAPE SEG          ;CHECK FOR FL:LOE, ELSE EXIT SEG
      (3) 033444 104010 EMT          C$ESCAPE
      (3) 033446 000112 .WORD        10000$-.
3078
3079 033450          BGNSEG          ;%%START OF SEGMENT%%
      (3) 033450 104004 EMT          C$BSEG
3080
3081
3082 033452 005037 002144 CLR        INTFLG    ;CLEAR INTERRUPT OCCURANCE FLAG
3083 033456 005077 146564 CLR        @RLDA
3084 033462 012777 177600 146560 MOV      #-128,@RLMP   ;SET UP WORD COUNT
3085 033470 012777 003052 146546 MOV      #BUF,@RLBA    ;SET UP BUS ADDRESS
3086
3087 033476 SETPRI  #PRI00          ;PRIORITY TO 0
      (3) 033476 012700 000000 MOV      #PRI00,R0
      (3) 033502 104041 EMT          C$SPRI
3088 033504 004537 020456 JSR        R5,LDFUNC    ;LOAD THE FUNCTION IN NEXT WORD
3089 033510 000102 WRCHK.INTEN ;WRITE CHECK UNDER INTERRUPT

```

D 9

ASSEMBLY ROUTINES MACY11 30A(1052) 22-NOV-78 15:35 PAGE 1-77  
 CZRLBB.P11 22-NOV-78 15:28 \*\*TEST 34\*\* - WRITE CHECK FUNCTION INTERRUPT SEQ 0107

```

3090 033512 004537 021276      JSR      R5,WTCRDY      ;WAIT FOR INTERRUPT
3091 033516                      ESCAPE  SEG      ;CHECK FOR FL:LOE, ELSE EXIT SEG
(3) 033516 104010              EMT      C$ESCAPE
(3) 033520 000036              .WORD    10001$-.

3092
3093 033522                      SETPRI  #PRI07      ;SET PRIORITY TO 7
(3) 033522 012700 000340      MOV      #PRI07,R0
(3) 033526 104041              EMT      C$SPRI
3094 033530 005737 002144      TST      INTFLG      ;DID INTERRUPT OCCUR?
3095 033534 001004              BNE      2$          ;YES-BRANCH NO-REPORT
3096
3097 033536                      ERRDF   4.,EM60,ERRO    ;WRITE DID NOT INTERRUPT
(3) 033536 104462              TRAP     T$ERCODE
(5) 033540 000004              .WORD    4
(5) 033542 013537              .WORD    EM60
(5) 033544 014244              .WORD    ERRO

3098 033546                      2$:      ESCAPE  SEG      ;CHECK FOR FL:LOE, ELSE EXIT SEG
(3) 033546 104010              EMT      C$ESCAPE
(3) 033550 000006              .WORD    10001$-.

3099
3100 033552 004537 020214      JSR      R5,CHERR      ;CHECK CNTLR FOR ERRORS
3101
3102 033556                      ENDSEG      ;%%END OF SEGMENT%%
(3) 033556 10001$:
(3) 033556 104005              EMT      C$ESEG
3103 033560                      ENDSEG      ;%%END OF SEGMENT%%
(3) 033560 10000$:
(3) 033560 104005              EMT      C$ESEG
3104 033562                      ENDTST      ;**END OF TEST**
(3) 033562 L10067:
(3) 033562 104001              EMT      C$ETST
3105
3106                      .SBTTL  **TEST 35** - PROPER INCREMENT OF RLBA ON WRITE CHECK
3107
3108 033564                      BGNTST      ;**START OF TEST**
3109
3110
3111 033564                      STARS
(2)
3112                      ;:*****
3113                      ;CHECK THAT THE RLBA WILL INCREMENT PROPERLY AFTER THE
3114                      ;WRITE CHECK WAS FINISHED THE RLBA SHOULD BE 128 WORDS (256 BYTES)
3115                      ;CREATER. STARTING RLBA IS 'BUF', ENDING SHOULD BE 'BUF + 256.'
3116 033564                      ;WE WILL MONITOR ALL ERRORS AND REPORT THEM ACCORDINGLY
(2)
3117                      STARS
3118                      ;:*****
3119 033564 004737 021356      JSR      PC,HDHOME    ;HEADS OVER TRACK 0
3120 033570                      CKERFG    ;HEADS GO HOME OKAY
(4) 033576 104032              EMT      C$EXIT
(4) 033600 000256              .WORD    L10070-.

3121
3122 033602                      BGNSEG      ;%%START OF SEGMENT%%
(3) 033602 104004              EMT      C$BSEG
3123
3124 033604 012700 003052      MOV      #BUF,R0      ;SETUP AND WRITE

```

E 9

ASSEMBLY ROUTINES MACY11 30A(1052) 22-NOV-78 15:35 PAGE 1-78  
 CZRLBB.P11 22-NOV-78 15:28 \*\*TEST 35\*\* - PROPER INCREMENT OF RLBA ON WRITE CHECK SEQ 0108

```

3125 033610 012701 000200      MOV      #128.,R1      ;128 WORDS
3126 033614 012720 125252      MOV      #125252,(R0)+ ;WRITE
3127 033620 005301              DEC      R1              ;DONE??
3128 033622 001374              BNE      299$
3129
3130 033624 012777 003052 146412 MOV      #BUF,@RLBA      ;LOAD BUS ADDRESS
3131 033632 012777 177600 146410 MOV      #-128.,@RLMP    ;WORD COUNT
3132 033640 005077 146402      CLR      @RLDA      ;CLEAR DISK ADDRESS
3133 033644 004537 020456      JSR      R5,LDFUNC    ;LOAD THE FUNCTION IN NEXT WORD
3134 033650 000012              WRITE
3135 033652 004537 021276      JSR      R5,WTCRDY    ;WAIT FOR CONTROLLER READY
3136 033656      ESCAPE SEG    ;CHECK FOR FL:LOE, ELSE EXIT SEG
(3) 033656 104010      EMT      C$ESCAPE
(3) 033660 000174      .WORD    10000$-.
3137 033662 004537 020214      JSR      R5,CHERR      ;CHECK CNTLR FOR ERRORS
3138 033666      ESCAPE SEG    ;CHECK FOR FL:LOE, ELSE EXIT SEG
(3) 033666 104010      EMT      C$ESCAPE
(3) 033670 000164      .WORD    10000$-.
3139      ;VERIFY WRITE WITH READ BEFORE WRCHK
3140
3141 033672 005077 146350      CLR      @RLDA
3142 033676 012777 003052 146340 MOV      #BUF,@RLBA
3143 033704 012777 177600 146336 MOV      #-128.,@RLMP
3144 033712 004537 020456      JSR      R5,LDFUNC    ;LOAD THE FUNCTION IN NEXT WORD
3145 033716 000014      READ
3146 033720 004537 021276      JSR      R5,WTCRDY
3147 033724      ESCAPE SEG    ;CHECK FOR FL:LOE, ELSE EXIT SEG
(3) 033724 104010      EMT      C$ESCAPE
(3) 033726 000126      .WORD    10000$-.
3148 033730 004537 020214      JSR      R5,CHERR      ;CHECK CNTLR FOR ERRORS
3149 033734      ESCAPE SEG    ;CHECK FOR FL:LOE, ELSE EXIT SEG
(3) 033734 104010      EMT      C$ESCAPE
(3) 033736 000116      .WORD    10000$-.
3150
3151 033740      BGNSEG
(3) 033740 104004      EMT      C$BSEG      ;%%START OF SEGMENT%%
3152
3153 033742      3$:
3154 033742 005077 146300      CLR      @RLDA
3155 033746 012777 003052 146270 MOV      #BUF,@RLBA      ;SET UP BUS ADDRESS
3156 033754 012777 177600 146266 MOV      #-128.,@RLMP    ;WORD COUNT
3157 033762 012737 003052 002166 MOV      #BUF,GDDAT      ;FORM EXPECTED BUS ADDRESS
3158 033770 062737 000400 002166 ADD      #256.,GDDAT    ;AFTER WRITE
3159
3160 033776 004537 020456      JSR      R5,LDFUNC    ;LOAD THE FUNCTION IN NEXT WORD
3161 034002 000002      WRCHK    ;WRITE CHECK
3162 034004 004537 021276      JSR      R5,WTCRDY    ;WAIT FOR CONTROLLER READY
3163 034010      ESCAPE SEG    ;CHECK FOR FL:LOE, ELSE EXIT SEG
(3) 034010 104010      EMT      C$ESCAPE
(3) 034012 000040      .WORD    10001$-.
3164
3165 034014 004537 020214      JSR      R5,CHERR      ;CHECK CNTLR FOR ERRORS
3166 034020      ESCAPE SEG    ;CHECK FOR FL:LOE, ELSE EXIT SEG
(3) 034020 104010      EMT      C$ESCAPE
(3) 034022 000030      .WORD    10001$-.
3167 034024 017737 146214 002170 MOV      @RLBA,BDDAT    ;READ 'RLBA' FOR PRESENT ADDRESS

```

```

ASSEMBLY ROUTINES MACY11 30A(1052) 22-NOV-78 15:35 PAGE 1-79 F 9
CZRLBB.P11 22-NOV-78 15:28 **TEST 35** - PROPER INCREMENT OF RLBA ON WRITE CHECK SEQ 0109

3168 034032 023737 002170 002166 CMP BDDAT,GDDAT ;DID 'BA' INCREMENT PROPERLY?
3169 034040 001404 BEQ 2$ ;YES, CONTINUE
3170
3171 034042 ERRDF 5,EM61,ERR4 ;BA DID NOT INCREMENT
(3) 034042 104462 TRAP T$ERRCODE
(5) 034044 000005 .WORD 5
(5) 034046 013575 .WORD EM61
(5) 034050 014410 .WORD ERR4
3172
3173 034052 2$:
3174
3175 034052 ENDSEG ;%%END OF SEGMENT%%
(3) 034052 10001$:
(3) 034052 104005 EMT C$ESEG
3176 034054 10000$:
(3) 034054 104005 EMT C$ESEG ;%%END OF SEGMENT%%
(3) 034054 104005 EMT C$ESEG
3177 034056 ENDTST ;**END OF TEST**
(3) 034056 L10070:
(3) 034056 104001 EMT C$ETST
3178
3179 .SBTTL **TEST 36** - PROPER INCREMENT OF RLDA ON WRITE CHECK
3180
3181 034060 BGNST ;**START OF TEST**
3182
3183 034060 STARS
(2) ;*****
3184 ;CHECK THAT THE SECTOR INCREMENTS AFTER THE WRITE CHECK WAS FINISHED.
3185 ;A FULL SECTOR WRITE CHECK THE RLDA SHOULD REFLECT AN INCREMENT
3186 ;OF THE SECTOR. 'GDDAT' WAS THE EXPECTED RLDA.
3187 034060 STARS
(2) ;*****
3188
3189
3190 034060 004737 021356 JSR PC,HDHOME ;HEADS OVER TRACK 0
3191 034064 CKERFG ;HEADS GO HOME OKAY
(4) 034072 104032 EMT C$EXIT
(4) 034074 000254 .WORD L10071-.
3192
3193 034076 BGNSEG ;%%START OF SEGMENT%%
(3) 034076 104004 EMT C$BSFG
3194
3195 034100 012700 003052 MOV #BUF,R0 ;SETUP AND WRITE
3196 034104 012701 000200 MOV #128,R1 ;128 WORDS
3197 034110 012720 125252 299$: MOV #125252,(R0)+ ;WRITE
3198 034114 005301 DEC R1 ;DONE??
3199 034116 001374 BNE 299$
3200
3201 034120 012777 003052 146116 MOV #BUF,@RLBA ;LOAD BUS ADDRESS
3202 034126 012777 177600 146114 MOV #-128,@RLMP ;WORD COUNT
3203 034134 005077 146106 CLR @RLDA ;CLEAR DISK ADDRESS
3204 034140 004537 020456 JSR R5,LDFUNC ;LOAD THE FUNCTION IN NEXT WORD
3205 034144 000012 WRITE
3206 034146 004537 021276 JSR R5,WTCRDY ;WAIT FOR CONTROLLER READY
3207 034152 ESCAPE SEC ;CHECK FOR FL:LOE, ELSE EXIT SEG
(3) 034152 104010 EMT C$ESCAPE

```

|      |        |        |        |        |                                      |             |                                  |
|------|--------|--------|--------|--------|--------------------------------------|-------------|----------------------------------|
| (3)  | 034154 | 000172 |        |        | .WORD                                | 10000\$-.   |                                  |
| 3208 | 034156 | 004537 | 020214 |        | JSR                                  | R5,CHERR    | ;CHECK CNTLR FOR ERRORS          |
| 3209 | 034162 |        |        |        | ESCAPE                               | SEG         | ;CHECK FOR FL:LOE, ELSE EXIT SEG |
| (3)  | 034162 | 104010 |        |        | EMT                                  | C\$ESCAPE   |                                  |
| (3)  | 034164 | 000162 |        |        | .WORD                                | 10000\$-.   |                                  |
| 3210 |        |        |        |        | ;VERIFY WRITE WITH READ BEFORE WRCHK |             |                                  |
| 3211 |        |        |        |        |                                      |             |                                  |
| 3212 | 034166 | 005077 | 146054 |        | CLR                                  | @RLDA       |                                  |
| 3213 | 034172 | 012777 | 003052 | 146044 | MOV                                  | #BUF,@RLBA  |                                  |
| 3214 | 034200 | 012777 | 177600 | 146042 | MOV                                  | #-128,@RLMP |                                  |
| 3215 | 034206 | 004537 | 020456 |        | JSR                                  | R5,LDFUNC   | ;LOAD THE FUNCTION IN NEXT WORD  |
| 3216 | 034212 | 000014 |        |        | READ                                 |             |                                  |
| 3217 | 034214 | 004537 | 021276 |        | JSR                                  | R5,WTCRDY   |                                  |
| 3218 | 034220 |        |        |        | ESCAPE                               | SEG         | ;CHECK FOR FL:LOE, ELSE EXIT SEG |
| (3)  | 034220 | 104010 |        |        | EMT                                  | C\$ESCAPE   |                                  |
| (3)  | 034222 | 000124 |        |        | .WORD                                | 10000\$-.   |                                  |
| 3219 | 034224 | 004537 | 020214 |        | JSR                                  | R5,CHERR    | ;CHECK CNTLR FOR ERRORS          |
| 3220 | 034230 |        |        |        | ESCAPE                               | SEG         | ;CHECK FOR FL:LOE, ELSE EXIT SEG |
| (3)  | 034230 | 104010 |        |        | EMT                                  | C\$ESCAPE   |                                  |
| (3)  | 034232 | 000114 |        |        | .WORD                                | 10000\$-.   |                                  |
| 3221 |        |        |        |        |                                      |             |                                  |
| 3222 | 034234 |        |        |        | BGNSEG                               |             | ;%%START OF SEGMENT%%            |
| (3)  | 034234 | 104004 |        |        | EMT                                  | C\$BSEG     |                                  |
| 3223 |        |        |        |        |                                      |             |                                  |
| 3224 | 034236 |        |        |        |                                      |             |                                  |
| 3225 | 034236 | 005037 | 002166 |        | CLR                                  | GDDAT       |                                  |
| 3226 | 034242 | 013777 | 002166 | 145776 | MOV                                  | GDDAT,@RLDA | ;SETUP DISK ADDRESS              |
| 3227 | 034250 | 005237 | 002166 |        | INC                                  | GDDAT       | ;CREATE EXPECTED SECTOR          |
| 3228 | 034254 | 012777 | 177600 | 145766 | MOV                                  | #-128,@RLMP | ;WORD COUNT                      |
| 3229 | 034262 | 012777 | 003052 | 145754 | MOV                                  | #BUF,@RLBA  | ;SETUP BUS ADDRESS               |
| 3230 |        |        |        |        |                                      |             |                                  |
| 3231 | 034270 | 004537 | 020456 |        | JSR                                  | R5,LDFUNC   | ;LOAD THE FUNCTION IN NEXT WORD  |
| 3232 | 034274 | 000002 |        |        | WRCHK                                |             | ;WRITE CHECK                     |
| 3233 | 034276 | 004537 | 021276 |        | JSR                                  | R5,WTCRDY   | ;WAIT FOR CONTROLLER READY       |
| 3234 | 034302 |        |        |        | ESCAPE                               | SEG         | ;CHECK FOR FL:LOE, ELSE EXIT SEG |
| (3)  | 034302 | 104010 |        |        | EMT                                  | C\$ESCAPE   |                                  |
| (3)  | 034304 | 000040 |        |        | .WORD                                | 10001\$-.   |                                  |
| 3235 |        |        |        |        |                                      |             |                                  |
| 3236 | 034306 | 004537 | 020214 |        | JSR                                  | R5,CHERR    | ;CHECK CNTLR FOR ERRORS          |
| 3237 | 034312 |        |        |        | ESCAPE                               | SEG         | ;CHECK FOR FL:LOE, ELSE EXIT SEG |
| (3)  | 034312 | 104010 |        |        | EMT                                  | C\$ESCAPE   |                                  |
| (3)  | 034314 | 000030 |        |        | .WORD                                | 10001\$-.   |                                  |
| 3238 |        |        |        |        |                                      |             |                                  |
| 3239 | 034316 | 013737 | 002232 | 002170 | MOV                                  | E.DA,BDDAT  | ;READ DISK ADDRESS               |
| 3240 | 034324 | 023737 | 002166 | 002170 | CMP                                  | GDDAT,BDDAT | ;DID SECTOR INCREMENT PROPERLY   |
| 3241 | 034332 | 001404 |        |        | BEQ                                  | 2\$         | ;YES, BRANCH NO, REPORT ERROR    |
| 3242 |        |        |        |        |                                      |             |                                  |
| 3243 | 034334 |        |        |        | ERRDF                                | 6,EM62,ERR4 | ;DA DID NOT INCREMENT            |
| (3)  | 034334 | 104462 |        |        | TRAP                                 | T\$RLODF    |                                  |
| (5)  | 034336 | 000006 |        |        | .WORD                                | 6           |                                  |
| (5)  | 034340 | 013652 |        |        | .WORD                                | EM62        |                                  |
| (5)  | 034342 | 014410 |        |        | .WORD                                | ERR4        |                                  |
| 3244 |        |        |        |        |                                      |             |                                  |
| 3245 | 034344 |        |        |        |                                      |             |                                  |
| 3246 |        |        |        |        |                                      |             |                                  |
| 3247 | 034344 |        |        |        | ENDSEG                               |             | ;%%END OF SEGMENT%%              |

3\$:

2\$:

```

ASSEMBLY ROUTINES          MACY11 30A(1052) 22-NOV-78 15:35 PAGE 1-81 H 9
CZRLBB.P11 22-NOV-78 15:28 **TEST 36** - PROPER INCREMENT OF RLDA ON WRITE CHECK                      SEQ 0111

(3) 034344 10001$: EMT C$ESEG ;%%END OF SEGMENT%%
(3) 034344 104005 ENDSEG
(3) 034346 10000$: EMT C$ESEG
(3) 034346 104005 ENDTST ;**END OF TEST**
(3) 034350 L10071: EMT C$ETST
(3) 034350 104001

3250
3251
3252 .SBTTL **TEST 37** - MULTIPLE SECTOR WRITE CHECK
3253 BGN1ST ;**START OF TEST**
3254 034352
3255
3256 034352
(2) STARS
3257 ;:*****
3258 ;CHECK FOR MULTIPLE SECTOR WRITE CHECK. THIS TEST CHECKS
3259 ;THAT TWO SECTORS CAN BE SUCCESSFULLY CHECKED. WE LOAD
3260 ;A WORD COUNT OF 129 WORDS (ONE SECTOR + 1 WORD) STARTING AT
3261 ;SECTOR 0 THRU SECTOR 37 AND VERIFY THAT THE RLDA DOES
3262 ;A DOUBLE INCREMENT EACH TIME.
3263 STARS
3264 ;:*****
3265
3266 034352 004737 021356 JSR PC,HDHOME ;HEADS OVER TRACK 0
3267 034356 CKERFG ;HEADS GO HOME OKAY
(4) 034364 104032 EMT C$EXIT
(4) 034366 000354 .WORD L10072-.

3268
3269 034370 BGNSEG ;%%START OF SEGMENT%%
(3) 034370 104004 EMT C$BSEG

3270
3271 034372 012737 000000 002160 MOV #0,TMP0
3272 034400 012737 000000 002162 MOV #0,TMP1
3273 034406 012700 003052 MOV #BUF,R0 ;SETUP AND WRITE
3274 034412 012701 000201 MOV #129,R1 ;129 WORDS
3275 034416 012720 125252 299$: MOV #125252,(R0)+ ;WRITE
3276 034422 005301 DEC R1 ;DONE??
3277 034424 001374 BNE 299$

3278
3279 034426 012777 003052 145610 1$: MOV #BUF,@RLBA ;LOAD BUS ADDRESS
3280 034434 012777 177577 145606 MOV #-129,@RLMP ;WORD COUNT
3281 034442 013737 002162 002166 MOV TMP1,GDDAT
3282 034450 053737 002160 002166 BIS TMP0,GDDAT
3283 034456 013777 002166 145552 MOV GDDAT,@RLDA
3284 034464 004537 020456 JSR R5,LDFUNC ;LOAD THE FUNCTION IN NEXT WORD
3285 034470 000012 WRITE
3286 034472 004537 021276 JSR R5,WTCRDY ;WAIT FOR CONTROLLER READY
3287 034476 ESCAPE SEG ;CHECK FOR FL:LOE, ELSE EXIT SEG
(3) 034476 104010 EMT C$ESCAPE
(3) 034500 000240 .WORD 10000$-.
3288 034502, 004537 020214 JSR R5,CHERR ;CHECK CNTLR FOR ERRORS
3289 034506 ESCAPE SEG ;CHECK FOR FL:LOE, ELSE EXIT SEG
(3) 034506 104010 EMT C$ESCAPE

```

```

(3) 034510 000230 .WORD 10000$-.
3290
3291 ;VERIFY WRITE WITH READ BEFORE WRCHK
3292
3293 034512 013737 002162 002166 MOV TMP1,GDDAT
3294 034520 053737 002160 002166 BIS TMP0,GDDAT
3295 034526 013777 002166 145512 MOV GDDAT,@RLDA
3296 034534 012777 003052 145502 MOV #BUF,@RLBA
3297 034542 012777 177577 145500 MOV #-129,@RLMP
3298 034550 004537 020456 JSR R5,LDFUNC ;LOAD THE FUNCTION IN NEXT WORD
3299 034554 000014 READ
3300 034556 004537 021276 JSR R5,WTCRDY
3301 034562 ESCAPE SEG ;CHECK FOR FL:LOE, ELSE EXIT SEG
(3) 034562 104010 EMT C$ESCAPE
(3) 034564 000154 .WORD 10000$-.
3302 034566 004537 020214 JSR R5,CHERR ;CHECK CNTLR FOR ERRORS
3303 034572 ESCAPE SEG ;CHECK FOR FL:LOE, ELSE EXIT SEG
(3) 034572 104010 EMT C$ESCAPE
(3) 034574 000144 .WORD 10000$-.
3304
3305 034576 BGNSEG ;%%START OF SEGMENT%%
(3) 034576 104004 EMT C$BSEG
3306
3307
3308 034600 013737 002162 002166 MOV TMP1,GDDAT ;GET CYLINDER
3309 034606 053737 002160 002166 BIS TMP0,GDDAT ;GET SECTOR
3310 034614 013777 002166 145424 MOV GDDAT,@RLDA ;SET DISK ADDRESS-SECTOR 0
3311 034622 062737 000002 002166 ADD #2,GDDAT ;SET EXPECTED + 2
3312 034630 012777 003052 145406 MOV #BUF,@RLBA ;SET BUS ADDRESS
3313 034636 012777 177577 145404 MOV #-129,@RLMP ;WORD COUNT-SECTOR+1 WORD
3314
3315 034644 004537 020456 JSR R5,LDFUNC ;LOAD THE FUNCTION IN NEXT WORD
3316 034650 000002 WRCHK ;WRITE CHECK
3317 034652 004537 021276 JSR R5,WTCRDY ;WAIT FOR CONTROLLER READY?
3318 034656 ESCAPE SEG ;CHECK FOR FL:LOE, ELSE EXIT SEG
(3) 034656 104010 EMT C$ESCAPE
(3) 034660 000042 .WORD 10001$-.
3319
3320 034662 004537 020214 JSR R5,CHERR ;CHECK CNTLR FOR ERRORS
3321 034666 ESCAPE SEG ;CHECK FOR FL:LOE, ELSE EXIT SEG
(3) 034666 104010 EMT C$ESCAPE
(3) 034670 000032 .WORD 10001$-.
3322
3323 034672 013737 002232 002170 MOV E.DA,BDDAT ;READ DISK ADDRESS
3324 034700 023737 002170 002166 CMP BDDAT,GDDAT ;IS DISK ADDRESS CORRECT
3325 034706 001404 BEQ 2$ ;YES, BRANCH NO, REPORT ERROR
3326
3327 034710 ERRDF 7,EM63,ERR4 ;DISK ADDRESS NOT CORRECT
(3) 034710 104462 TRAP T$LRCODE
(5) 034712 000007 .WORD 7
(5) 034714 013727 .WORD EM63
(5) 034716 014410 .WORD ERR4
3328
3329 034720 2$: CKLOOP
(3) 034720 104006 EMT C$CLP1
3330

```

```

3331 034722          ENDSEG                      ;%%END OF SEGMENT%%
(3) 034722          10001$: EMT C$ESEG
(3) 034722 104005
3332
3333 034724 005237 002160          INC TMP0          ;NEXT SECTOR
3334 034730 022737 000046 002160      CMP #46,TMP0      ;AT END?
3335 034736 001233          BNE 1$          ;NO, GO BACK
3336 034740          ENDSEG                      ;%%END OF SEGMENT%%
(3) 034740          10000$: EMT C$ESEG
(3) 034740 104005
3337 034742          ENDTST                      ;**END OF TEST**
(3) 034742          L10072:
(3) 034742 104001          EMT C$ETST
3338          .SBTTL **TEST 38** - FORCE DCK WITH WRITE CHECK
3339
3340 034744          BGNST                      ;**START OF TEST**
3341
3342 034744          STARS
(2)          ;:*****
3343          ;FORCE A DCK WITH WRITE CHECK. THIS IS DONE BY WRITING
3344          ;A SECTOR AND CHANGING A WORD IN MEMORY BEFORE WRITE CHECK
3345          ;IS ISSUED..
3346 034744          STARS
(2)          ;:*****
3347
3348 034744 004737 021356          JSR PC,HDHOME          ;HEADS OVER TRACK 0
3349 034750          CKERFG                      ;HEADS GO HOME OKAY
(4) 034756 104032          EMT C$EXIT
(4) 034760 000262          .WORD L10073-.
3350
3351 034762          BGNSEG                      ;%%START OF SEGMENT%%
(3) 034762 104004          EMT C$BSEG
3352
3353 034764 012700 003052          MOV #BUF,R0          ;SETUP AND WRITE
3354 034770 012701 000200          MOV #128,R1          ;128 WORDS
3355 034774 012720 125252          299$: MOV #125252,(R0)+ ;WRITE
3356 035000 005301          DEC R1          ;DONE??
3357 035002 001374          BNE 299$
3358
3359 035004 012777 003052 145232      MOV #BUF,@RLBA          ;LOAD BUS ADDRESS
3360 035012 012777 177600 145230      MOV #-128,@RLMP          ;WORD COUNT
3361 035020 005077 145222          CLR @RLDA          ;CLEAR DISK ADDRESS
3362 035024 004537 020456          JSR R5,LDFUNC          ;LOAD THE FUNCTION IN NEXT WORD
3363 035030 000012          WRITE
3364 035032 004537 021276          JSR R5,WTCRDY          ;WAIT FOR CONTROLLER READY
3365 035036          ESCAPE SEG          ;CHECK FOR FL:LOE, ELSE EXIT SEG
(3) 035036 104010          EMT C$ESCAPE
(3) 035040 000200          .WORD 10000$-.
3366 035042 004537 020214          JSR R5,CHERR          ;CHECK CNTLR FOR ERRORS
3367 035046          ESCAPE SEG          ;CHECK FOR FL:LOE, ELSE EXIT SEG
(3) 035046 104010          EMT C$ESCAPE
(3) 035050 000170          .WORD 10000$-.
3368          ;VERIFY WRITE WITH READ BEFORE WRCHK
3369
3370 035052 005077 145170          CLR @RLDA
3371 035056 012777 003052 145160      MOV #BUF,@RLBA
  
```

```

3372 035064 012777 177600 145156      MOV    #-128.,@RLMP
3373 035072 004537 020456      JSR    R5,LDFUNC          ;LOAD THE FUNCTION IN NEXT WORD
3374 035076 000014      READ
3375 035100 004537 021276      JSR    R5,WTCRDY
3376 035104      ESCAPE SEG          ;CHECK FOR FL:LOE, ELSE EXIT SEG
(3) 035104 104010      EMT    C$ESCAPE
(3) 035106 000132      .WORD 10000$-
3377 035110 004537 020214      JSR    R5,CHERR          ;CHECK CNTLR FOR ERRORS
3378 035114      ESCAPE SEG          ;CHECK FOR FL:LOE, ELSE EXIT SEG
(3) 035114 104010      EMT    C$ESCAPE
(3) 035116 000122      .WORD 10000$-
3379
3380 035120      BGNSEG          ;%%START OF SEGMENT%%
(3) 035120 104004      EMT    C$BSEG
3381
3382
3383 035122 005037 003052      CLR    3UF
3384 035126 005077 145114      CLR    @RLDA
3385 035132 012777 003052 145104      MOV    #3UF,@RLBA      ;SETTING SECTOR 40 OF CYL. ADDR.
3386 035140 012777 177600 145102      MOV    #-128.,@RLMP    ;WORD COUNT
3387
3388 035146 004537 020456      JSR    R5,LDFUNC          ;LOAD THE FUNCTION IN NEXT WORD
3389 035152 000002      WRCHK          ;WRITE CHECK
3390 035154 004537 021276      JSR    R5,WTCRDY          ;WAIT FOR CONTROLLER READY
3391 035160      ESCAPE SEG          ;CHECK FOR FL:LOE, ELSE EXIT SEG
(3) 035160 104010      EMT    C$ESCAPE
(3) 035162 000054      .WORD 10001$-
3392
3393 035164 013737 002226 002160      MOV    E.CS,TMP0      ;GET RLCS
3394 035172 042737 001777 002160      BIC    #1777,TMP0      ;SAVE ERROR BITS
3395 035200 022737 104000 002160      CMP    #BIT15:BIT11,TMP0 ;DCK SET.
3396 035206 001402      BEQ    1$          ;YES, CONTINUE
3397 035210 004537 020214      JSR    R5,CHERR
3398 035214      CKLOOP
(3) 035214 104006      EMT    C$CLP1
3399
3400 035216 022737 104000 002160      CMP    #BIT15:BIT11,TMP0
3401 035224 001404      BEQ    2$
3402
3403 035226      ERRDF 23.,EM65,ERRO
(3) 035226 104462      TRAP  T$ERCODE
(5) 035230 000027      .WORD 23
(5) 035232 014105      .WORD EM65
(5) 035234 014244      .WORD ERRO
3404
3405 035236      2$:          ;WHEN FORCED
3406
3407 035236      ENDSEG          ;%%END OF SEGMENT%%
(3) 035236      10001$:
(3) 035236 104005      EMT    C$ESEG
3408 035240      ENDSEG          ;%%END OF SEGMENT%%
(3) 035240      10000$:
(3) 035240 104005      EMT    C$ESEG
3409 035242      ENDTST          ;**END OF TEST**
(3) 035242      110073:
(3) 035242 104001      EMT    C$ETST

```

```

3410
3411 .SBTTL **TEST 39** - FORCE DCK WITH WRITE CHECK INTERRUPT
3412
3413 035244 BGNTST ;**START OF TEST**
3414
3415
3416 035244 STARS
3417 (2) ;*****
3418 035244 ;:FORCE A DCK INTERRUPT MODE
3419 (2) STARS
3420 ;:*****
3421 035244 004737 021356 .HDHOME ;HEADS OVER TRACK 0
3422 035250 ;HEADS GO HOME OKAY
3423 (4) 035256 104032 L EMT C$EXIT
3424 (4) 035260 000322 .WORD 110074-.
3425
3426 035262 BGNSEG ;%%START OF SEGMENT%%
3427 (3) 035262 104004 EMT C$BSEG
3428
3429 035264 012700 003052 MOV #BUF,R0 ;SETUP AND WRITE
3430 035270 012701 000200 MOV #128,R1 ;128 WORDS
3431 035274 012720 125252 299$: MOV #125252,(R0)+ ;WRITE
3432 035300 005301 DEC R1 ;DONE??
3433 035302 001374 BNE 299$
3434
3435 035304 012777 003052 144732 MOV #BUF,@RLBA ;LOAD BUS ADDRESS
3436 035312 012777 177600 144730 MOV #-128,@RLMP ;WORD COUNT
3437 035320 005077 144722 CLR @RLDA ;CLEAR DISK ADDRESS
3438 035324 004537 020456 JSR R5,LDFUNC ;LOAD THE FUNCTION IN NEXT WORD
3439 035330 000012 WRITE
3440 035332 004537 021276 JSR R5,WTCRDY ;WAIT FOR CONTROLLER READY
3441 035336 ESCAPE SEG ;CHECK FOR FL:LOE, ELSE EXIT SEG
3442 (3) 035336 104010 EMT C$ESCAPE
3443 (3) 035340 000240 .WORD 10000$-.
3444 035342 004537 020214 JSR R5,CHERR ;CHECK CNTLR FOR ERRORS
3445 035346 ESCAPE SEG ;CHECK FOR FL:LOE, ELSE EXIT SEG
3446 (3) 035346 104010 EMT C$ESCAPE
3447 (3) 035350 000230 .WORD 10000$-.
3448 ;VERIFY WRITE WITH READ BEFORE WPCHK
3449
3450 035352 005077 144670 CLR @RLDA
3451 035356 012777 003052 144660 MOV #BUF,@RLBA
3452 035364 012777 177600 144656 MOV #-128,@RLMP
3453 035372 004537 020456 JSR R5,LDFUNC ;LOAD THE FUNCTION IN NEXT WORD
3454 035376 000014 READ
3455 035400 004537 021276 JSR R5,WTCRDY
3456 035404 ESCAPE SEG ;CHECK FOR FL:LOE, ELSE EXIT SEG
3457 (3) 035404 104010 EMT C$ESCAPE
3458 (3) 035406 000172 .WORD 10000$-.
3459 035410 004537 020214 JSR R5,CHERR ;CHECK CNTLR FOR ERRORS
3460 035414 ESCAPE SEG ;CHECK FOR FL:LOE, ELSE EXIT SEG
3461 (3) 035414 104010 EMT C$ESCAPE
3462 (3) 035416 000162 .WORD 10000$-.
  
```

```

3453 035420          BGNSEG          ;%%START OF SEGMENT%%
(3) 035420 104004    EMT          C$BSEG
3454
3455 035422          SETPRI #PRI00
(3) 035422 012700 000000 MOV #PRI00,RO
(3) 035426 104041    EMT          C$SPRI
3456 035430 005037 002144 CLR INTFLG          ;CLEAR INTERRUPT OCCURANCE FLAG
3457 035434 005037 003052 CLR BUF
3458 035440 005077 144602 CLR @RLDA
3459 035444 012777 003052 144572 MOV #BUF,@RLBA          ;SETTING SECTOR 40 OF CYL. ADDR.
3460 035452 012777 177600 144570 MOV #-128.,@RLMP          ;WORD COUNT
3461
3462 035460 004537 020456 JSR R5,LDFUNC          ;LOAD THE FUNCTION IN NEXT WORD
3463 035464 000102 WRCHK.INTEN          ;WRITE CHECK
3464 035466 004537 021276 JSR R5,WTCRDY          ;WAIT FOR CONTROLLER READY
3465 035472
(3) 035472 104006    EMT          C$CLP1
3466 035474          SETPRI #PRI07
(3) 035474 012700 000340 MOV #PRI07,RO
(3) 035500 104041    EMT          C$SPRI
3467
3468 035502 005737 002144 TST INTFLG          ;DID INTERRUPT OCCUR
3469 035506 001004    BNE 2$          ;YES OKAY
3470
3471 035510          ERRDF 24.,EM66,ERRO          ;NO INTERRUPT FROM DCK
(3) 035510 104462    TRAP T$ERCODE
(5) 035512 000030    .WORD 24
(5) 035514 014146    .WORD EM66
(5) 035516 014244    .WORD ERRO
3472
3473 035520          2$: ESCAPE SEG          ;CHECK FOR FL:LOE, ELSE EXIT SEG
(3) 035520 104010    EMT          C$ESCAPE
(3) 035522 000054    .WORD 10001$-.
3474
3475
3476 035524 013737 002226 002160 MOV E.CS,TMP0          ;GET RLCS
3477 035532 042737 001777 002160 BIC #1777,TMP0          ;SAVE ERROR BITS
3478 035540 022737 104000 002160 CMP #BIT15:BIT11,TMP0          ;DCK SET.
3479 035546 001402    BEQ 1$          ;YES, CONTINUE
3480
3481 035550 004537 020214          JSR R5,CHERR
3482 035554          1$: CKLOOP
(3) 035554 104006    EMT          C$CLP1
3483
3484 035556 022737 104000 002160 CMP #BIT15:BIT11,TMP0
3485 035564 001404    BEQ 3$
3486 035566          ERRDF 25.,EM65,ERRO
(3) 035566 104462    TRAP T$ERCODE
(5) 035570 000031    .WORD 25
(5) 035572 014105    .WORD EM65
(5) 035574 014244    .WORD ERRO
3487
3488 035576          3$:          ;WHEN FORCED
3489
3490 035576          ENDSEG          ;%%END OF SEGMENT%%
(3) 035576          10001$:

```

```

ASSEMBLY ROUTINES MACY11 30A(1052) 22-NOV-78 15:35 PAGE 1-87 N 9
CZRLBB.P11 22-NOV-78 15:28 **TEST 39** - FORCE DCK WITH WRITE CHECK INTERRUPT SEQ 0117

(3) 035576 104005 EMT C$ESEG
3491 035600 ENDSEG ;%%END OF SEGMENT%%
(3) 035600 10000$: EMT C$ESEG
3492 035602 ENDTST ;**END OF TEST**
(3) 035602 L10074: EMT C$ETST
3493 035602 104001
3494
3495 .SBTTL **TEST 40** - CHECK ZERO FILL ON WRITE WITH WRITE CHECK
3496 BGNSTST ;**START OF TEST**
3497 035604
3498
3499
3500
3501 035604 STARS
(2) ;*****
3502 ;WHEN WRITING PARTIAL SECTORS (LESS THAN 128 WORDS) THE
3503 ;CONTROLLER WILL FILL IN THE REMAINING PORTION OF
3504 ;THE SECTOR WITH ZERO WORDS. CHECK THIS FEATURE CAN BE WRITE CHECKED
3505 ;WITH WORD COUNTS FROM 1 TO 127
3506 035604 STARS
(2) ;*****
3507
3508 035604 004737 021356 JSR PC,HDHOME ;HEADS OVER TRACK 0
3509 035610 CKERFG ;HEADS GO HOME OKAY
(4) 035616 104032 EMT C$EXIT
(4) 035620 000274 .WORD L10075-.
3510
3511 035622 BGNSEG ;%%START OF SEGMENT%%
(3) 035622 104004 EMT C$BSEG
3512
3513 035624 012737 000001 002162 MOV #1,TMP1 ;START WITH 1 WORD WRITE
3514 035632 012700 003052 33$: MOV #BUF,R0 ;WRITE BUFFER WITH 52525, WE'LL
3515 035636 012701 000200 MOV #128,R1 ;WRITE 128 WORDS ALL THOUGH WE'RE
3516 035642 012720 052525 3$: MOV #52525,(R0)+ ;ONLY GOING TO TRANSFER < 128
3517 035646 005301 DEC R1 ;DONE WITH BUFFER?
3518 035650 001374 BNE 3$ ;NO, GO BACK
3519 035652 013700 002162 MOV TMP1,R0 ;GET TRANSFER WORD COUNT
3520 035656 005400 NEG R0 ;NEGATE FOR RLMP
3521 035660 010077 144364 MOV R0,@RLMP ;STORE WORD COUNT AWAY
3522 035664 012777 003052 144352 MOV #BUF,@RLBA ;SET UP RLBA
3523 035672 005077 144350 CLR @RLDA
3524 035676 004537 020456 JSR R5,LDFUNC ;LOAD THE FUNCTION IN NEXT WORD
3525 035702 000012 WRITE ;WRITE IT
3526 035704 004537 021276 JSR R5,WTCRDY ;WAIT FOR WRITE TO FINISH
3527 035710 ESCAPE SEG ;CHECK FOR FL:LOE, ELSE EXIT SEG
(3) 035710 104010 EMT C$ESCAPE
(3) 035712 000200 .WORD 10000$-.
3528
3529 035714 004537 020214 JSR R5,CHERR ;CHECK CNLRL FOR ERRORS
3530 035720 ESCAPE SEG ;CHECK FOR FL:LOE, ELSE EXIT SEG
(3) 035720 104010 EMT C$ESCAPE
(3) 035722 000170 .WORD 10000$-.
3531
3532
3533

```

|      |        |        |        |          |        |               |                                  |
|------|--------|--------|--------|----------|--------|---------------|----------------------------------|
| 3533 | 035724 | 005077 | 144316 |          | CLR    | ARLDA         |                                  |
| 3534 | 035730 | 012777 | 003052 | 144306   | MOV    | ABUF,ARLBA    |                                  |
| 3535 | 035736 | 013700 | 002162 |          | MOV    | TMP1,RO       |                                  |
| 3536 | 035742 | 005400 |        |          | NEG    | RO            |                                  |
| 3537 | 035744 | 010077 | 144300 |          | MOV    | RO,ARLMP      |                                  |
| 3538 | 035750 | 004537 | 020456 |          | JSR    | R5,LDFUNC     | ;LOAD THE FUNCTION IN NEXT WORD  |
| 3539 | 035754 | 000014 |        |          | READ   |               |                                  |
| 3540 | 035756 | 004537 | 021276 |          | JSR    | R5,WTCRDY     |                                  |
| 3541 | 035762 |        |        |          | ESCAPE | SEG           | ;CHECK FOR FL:LOE, ELSE EXIT SEG |
| (3)  | 035762 | 104010 |        |          | EMT    | C\$ESCAPE     |                                  |
| (3)  | 035764 | 000126 |        |          | .WORD  | 10000\$-      |                                  |
| 3542 | 035766 | 004537 | 020214 |          | JSR    | R5,CHERR      | ;CHECK CNTLR FOR ERRORS          |
| 3543 | 035772 |        |        |          | ESCAPE | SEG           | ;CHECK FOR FL:LOE, ELSE EXIT SEG |
| (3)  | 035772 | 104010 |        |          | EMT    | C\$ESCAPE     |                                  |
| (3)  | 035774 | 000116 |        |          | .WORD  | 10000\$-      |                                  |
| 3544 |        |        |        |          |        |               |                                  |
| 3545 | 035776 |        |        |          | BGNSEG |               | ;%%START OF SEGMENT%%            |
| (3)  | 035776 | 104004 |        |          | EMT    | C\$BSEG       |                                  |
| 3546 | 036000 | 012777 | 003052 | 144236   | MOV    | ABUF,ARLBA    | ;SET UP TO READ                  |
| 3547 | 036006 | 013700 | 002162 |          | MOV    | TMP1,RO       |                                  |
| 3548 | 036012 | 005400 |        |          | NEG    | RO            |                                  |
| 3549 | 036014 | 010077 | 144230 |          | MOV    | RO,ARLMP      |                                  |
| 3550 | 036020 | 005077 | 144222 |          | CLR    | ARLDA         | ;SECTOR                          |
| 3551 | 036024 | 004537 | 020456 |          | JSR    | R5,LDFUNC     | ;LOAD THE FUNCTION IN NEXT WORD  |
| 3552 | 036030 | 000002 |        |          | WRCHK  |               |                                  |
| 3553 | 036032 | 004537 | 021276 |          | JSR    | R5,WTCRDY     | ;WAIT TIL WE FINISH THE WRCHK    |
| 3554 | 036036 |        |        |          | ESCAPE | SEG           | ;CHECK FOR FL:LOE, ELSE EXIT SEG |
| (3)  | 036036 | 104010 |        |          | EMT    | C\$ESCAPE     |                                  |
| (3)  | 036040 | 000034 |        |          | .WORD  | 10001\$-      |                                  |
| 3555 |        |        |        |          |        |               |                                  |
| 3556 | 036042 | 004537 | 020214 |          | JSR    | R5,CHERR      | ;CHECK CNTLR FOR ERRORS          |
| 3557 | 036046 | 005737 | 002124 |          | TST    | T.CRC         | ;WAS ERROR A DCK??               |
| 3558 | 036052 | 001003 |        |          | BNE    | 8\$           | ;YES, GIVE MOR INFO              |
| 3559 | 036054 |        |        | 10\$:    | ESCAPE | SEG           | ;CHECK FOR FL:LOE, ELSE EXIT SEG |
| (3)  | 036054 | 104010 |        |          | EMT    | C\$ESCAPE     |                                  |
| (3)  | 036056 | 000016 |        |          | .WORD  | 10001\$-      |                                  |
| 3560 | 036060 | 000405 |        |          | BR     | 99\$          | ;SKIP AROUND                     |
| 3561 | 036062 |        |        | 8\$:     | (KLOOP |               | ;YES, CHECK FOR LOOP FIRST       |
| (3)  | 036062 | 104006 |        |          | EMT    | C\$CLP1       |                                  |
| 3562 | 036064 |        |        |          | ERRDF  | 37,EM64,ERR14 |                                  |
| (3)  | 036064 | 104462 |        |          | TRAP   | T\$ERCODE     |                                  |
| (5)  | 036066 | 000045 |        |          | .WORD  | 37            |                                  |
| (5)  | 036070 | 014031 |        |          | .WORD  | EM64          |                                  |
| (5)  | 036072 | 015150 |        |          | .WORD  | ERR14         |                                  |
| 3563 | 036074 |        |        | 99\$:    |        |               | ;EXIT TEST                       |
| 3564 | 036074 |        |        |          | ENDSEG |               | ;%%END OF SEGMENT%%              |
| (3)  | 036074 |        |        | 10001\$: |        |               |                                  |
| (3)  | 036074 | 104005 |        |          | EMT    | C\$ESEG       |                                  |
| 3565 |        |        |        |          |        |               |                                  |
| 3566 | 036076 | 005237 | 002162 |          | INC    | TMP1          |                                  |
| 3567 | 036102 | 023727 | 002162 | 000200   | CMP    | TMP1,#128.    |                                  |
| 3568 | 036110 | 001250 |        |          | BNE    | 33\$          |                                  |
| 3569 |        |        |        |          |        |               |                                  |
| 3570 | 036112 |        |        |          | ENDSEG |               | ;%%END OF SEGMENT%%              |
| (3)  | 036112 |        |        | 10000\$: |        |               |                                  |
| (3)  | 036112 | 104005 |        |          | EMT    | C\$ESEG       |                                  |

```

3571 036114      ENDTST                      ;**END OF TEST**
(3) 036114      L10075:
(3) 036114 104001      EMT      C$ETST
3572      .SBTTL  **TEST 41** - EXTENDED CHECK OF WRITE CHECK FUNCTION
3573
3574 036116      BGNST                      ;**START OF TEST**
3575
3576 036116      STARS
(2)      ;*****
3577      ;CHECK OF WRITE CHECK LOGIC UNDER FLAG MODE
3578      ;THIS TEST IS DONE WITH ALL BIT PATTERNS
3579      ; WE WILL WRITE CHECK A FULL SECTOR (128 WORDS) FROM
3580      ;MEMORY (BUF). WE CHECK THAT NO ERRORS OCCUR.
3581 036116      STARS
(2)      ;*****
3582
3583
3584 036116 004737 021356      JSR      PC,HDHOME      ;HEADS OVER TRACK 0
3585 036122      CKERFG      ;HEADS GO HOME OKAY
(4) 036130 104032      EMT      C$EXIT
(4) 036132 000246      .WORD      L10076-.
3586
3587 036134 012703 002504      MOV      #HDRTAB,R3
3588
3589 036140      BGNSEG                      ;%%START OF SEGMENT%%
(3) 036140 104004      EMT      C$BSEG
3590
3591 036142 012700 003052      298$: MOV      #BUF,R0      ;SETUP AND WRITE
3592 036146 012701 000200      MOV      #128.,R1      ;128 WORDS
3593 036152 011302      MOV      (R3),R2
3594 036154 010220      299$: MOV      R2,(R0)+      ;WRITE
3595 036156 005301      DEC      R1      ;DONE??
3596 036160 001375      BNE      299$
3597
3598 036162 012777 003052 144054      MOV      #BUF,@RLBA      ;LOAD BUS ADDRESS
3599 036170 012777 177600 144052      MOV      #-128.,@RLMP      ;WORD COUNT
3600 036176 005077 144044      CLR      @RLDA      ;CLEAR DISK ADDRESS
3601 036202 004537 020456      JSR      R5,LDFUNC      ;LOAD THE FUNCTION IN NEXT WORD
3602 036206 000012      WRITE
3603 036210 004537 021276      JSR      R5,WTCRDY      ;WAIT FOR CONTROLLER READY
3604 036214      ESCAPE SEG      ;CHECK FOR FL:LOE, ELSE EXIT SEG
(3) 036214 104010      EMT      C$ESCAPE
(3) 036216 000160      .WORD      10000$-.
3605 036220 004537 020214      JSR      R5,CHERR      ;CHECK CNTLR FOR ERRORS
3606 036224      ESCAPE SEG      ;CHECK FOR FL:LOE, ELSE EXIT SEG
(3) 036224 104010      EMT      C$ESCAPE
(3) 036226 000150      .WORD      10000$-.
3607 036230      BGNSEG                      ;%%START OF SEGMENT%%
(3) 036230 104004      EMT      C$BSEG
3608
3609      ;VERIFY WRITE WITH READ BEFORE WRCHK
3610
3611 036232 005077 144010      CLR      @RLDA
3612 036236 012777 003052 144000      MOV      #BUF,@RLBA
3613 036244 012777 177600 143776      MOV      #-128.,@RLMP
3614 036252 004537 020456      JSR      R5,LDFUNC      ;LOAD THE FUNCTION IN NEXT WORD

```

```

3615 036256 000014      READ
3616 036260 004537 021276 JSR    R5,WTCRDY
3617 036264      ESCAPE SEG          ;CHECK FOR FL:LOE, ELSE EXIT SEG
(3) 036264 104010 EMT    C$ESCAPE
(3) 036266 000076 .WORD 10001$-
3618 036270 004537 020214 JSR    R5,CHERR          ;CHECK CNTLR FOR ERRORS
3619 036274      ESCAPE SEG          ;CHECK FOR FL:LOE, ELSE EXIT SEG
(3) 036274 104010 EMT    C$ESCAPE
(3) 036276 000066 .WORD 10001$-
3620
3621 036300      BGNSEG                ;%%START OF SEGMENT%%
(3) 036300 104004 EMT    C$BSEG
3622
3623 036302      3$:
3624 036302 005077 143740 CLR    @RLDA
3625 036306 012777 177600 143734 MOV    #-128.,@RLMP          ;WORD COUNT
3626 036314 012777 003052 143722 MOV    #BUF,@RLBA          ;BUS ADDRESS
3627 036322 004537 020456 JSR    R5,LDFUNC          ;LOAD THE FUNCTION IN NEXT WORD
3628 036326 000002      WRCHK          ;WRITE CHECK
3629
3630 036330 004537 021276 JSR    R5,WTCRDY          ;WAIT FOR CONTROLLER READY
3631 036334      ESCAPE SEG          ;CHECK FOR FL:LOE, ELSE EXIT SEG
(3) 036334 104010 EMT    C$ESCAPE
(3) 036336 000024 .WORD 10002$-
3632
3633
3634 036340 004537 020214 JSR    R5,CHERR          ;CHECK CNTLR FOR ERRORS
3635 036344 005737 002124 TST    T.CRC          ;WRITE CHECK ERROR??
3636 036350 001404      BEQ    4$          ;NO
3637
3638 036352      ERRHRD 410.,ERR15,EM70
(3) 036352 104463 TRAP    T$ERCODE
(5) 036354 000632 .WORD 410
(5) 036356 015216 .WORD ERR15
(5) 036360 014220 .WORD EM70
3639
3640 036362      4$:
3641
3642 036362      ENDSEG                ;%%END OF SEGMENT%%
(3) 036362 10002$: EMT    C$ESEG
(3) 036362 104005      ENDSEG                ;%%END OF SEGMENT%%
3643 036364 10001$: EMT    C$ESEG
(3) 036364 104005
3644
3645 036366 005723      TST    (R3)+
3646 036370 020327 002660 CMP    R3,#HDREND
3647 036374 001262      BNE    298$
3648
3649 036376      ENDSEG                ;%%END OF SEGMENT%%
(3) 036376 10000$: EMT    C$ESEG
(3) 036376 104005
3650 036400      ENDTST
(3) 036400 L10076: EMT    C$ETST
(3) 036400 104001
3651

```

```

3652
3653      .SBTTL  **TEST 42** - EXTENDED CHECK OF WRITE CHECK FUNCTION
3654
3655 036402      BGNSTST                      ;**START OF TEST**
3656
3657 036402      STARS
(2)      ;*****
3658      ;CHECK OF WRITE CHECK LOGIC UNDER FLAG MODE
3659      ;TEST IS DONE WITH ALL BIT PATTERNS
3660      ; WE WILL WRITE CHECK A FULL SECTOR (128 WORDS) FROM
3661      ;MEMORY (BUF). WE CHECK THAT NO ERRORS OCCUR.
3662 036402      STARS
(2)      ;*****
3663
3664
3665 036402 004737 021356      JSR      PC,HDHOME      ;HEADS OVER TRACK 0
3666 036406      CKERFG      ;HEADS GO HOME OKAY
(4) 036414 104032      EMT      C$EXIT
(4) 036416 000252      .WORD    L10077-
3667
3668 036420 012703 002504      MOV      #HDIRTAB,R3
3669
3670 036424      BGNSEG                      ;%%START OF SEGMENT%%
(3) 036424 104004      EMT      C$BSEG
3671
3672 036426 012700 003052      298$: MOV      #BUF,R0      ;SETUP AND WRITE
3673 036432 012701 000200      MOV      #128,R1      ;128 WORDS
3674 036436 011302      MOV      (R3),R2      ;GET PATTERN
3675 036440 052702 100000      BIS      #BIT15,R2
3676 036444 010220      299$: MOV      R2,(R0)+
3677 036446 005301      DEC      R1
3678 036450 001375      BNE      299$      ;DONE??
3679
3680 036452 012777 003052 143564      MOV      #BUF,@RLBA      ;LOAD BUS ADDRESS
3681 036460 012777 177600 143562      MOV      #-128,@RLMP      ;WORD COUNT
3682 036466 005077 143554      CLR      @RLDA      ;CLEAR DISK ADDRESS
3683 036472 004537 020456      JSR      R5,LDFUNC      ;LOAD THE FUNCTION IN NEXT WORD
3684 036476 000012      WRITE
3685 036500 004537 021276      JSR      R5,WTCRDY      ;WAIT FOR CONTROLLER READY
3686 036504      ESCAPE SEG      ;CHECK FOR FI:LOE, ELSE EXIT SEG
(3) 036504 104010      EMT      C$ESCAPE
(3) 036506 000160      .WORD    10000$-
3687 036510 004537 020214      JSR      R5,CHERR      ;CHECK CNTLR FOR ERRORS
3688 036514      ESCAPE SEG      ;CHECK FOR FL:LOE, ELSE EXIT SEG
(3) 036514 104010      EMT      C$ESCAPE
(3) 036516 000150      .WORD    10000$-
3689 036520      BGNSEG                      ;%%START OF SEGMENT%%
(3) 036520 104004      EMT      C$BSEG
3690
3691      ;VERIFY WRITE WITH READ BEFORE WRCHK
3692
3693 036522 005077 143520      CLR      @RLDA
3694 036526 012777 003052 143510      MOV      #BUF,@RLBA
3695 036534 012777 177600 143506      MOV      #-128,@RLMP
3696 036542 004537 020456      JSR      R5,LDFUNC      ;LOAD THE FUNCTION IN NEXT WORD
3697 036546 000014      READ
  
```

ASSEMBLY ROUTINES

CZRLBB.P11

22-NOV-78

MACY11

15:28

30A(1052)

22-NOV-78

15:35

PAGE 1-92

\*\*TEST 42\*\* - EXTENDED CHECK OF WRITE CHECK FUNCTION

SEQ 0122

```

3698 036550 004537 021276      JSR      R5,WTCRDY
3699 036554      ESCAPE SEG      ;CHECK FOR FL:LOE, ELSE EXIT SEG
(3) 036554 104010      EMT      C$ESCAPE
(3) 036556 000076      .WORD    10001$-
3700 036560 004537 020214      JSR      R5,CHERR      ;CHECK CNTLR FOR ERRORS
3701 036564      ESCAPE SEG      ;CHECK FOR FL:LOE, ELSE EXIT SEG
(3) 036564 104010      EMT      C$ESCAPE
(3) 036566 000066      .WORD    10001$-
3702
3703 036570      BGNSEG      ;%%START OF SEGMENT%%
(3) 036570 104004      EMT      C$BSEG
3704
3705 036572      3$:
3706 036572 005077 143450      C R      @RLDA
3707 036576 012777 177600      MOV      #-128.,@RLMP      ;WORD COUNT
3708 036604 012777 003052      MOV      #BUF,@RLBA      ;BUS ADDRESS
3709 036612 004537 020456      JSR      R5,LDFUNC      ;LOAD THE FUNCTION IN NEXT WORD
3710 036616 000002      WRCHK      ;WRITE CHECK
3711
3712 036620 004537 021276      JSR      R5,WTCRDY      ;WAIT FOR CONTROLLER READY
3713 036624      ESCAPE SEG      ;CHECK FOR FL:LOE, ELSE EXIT SEG
(3) 036624 104010      EMT      C$ESCAPE
(3) 036626 000024      .WORD    10002$-
3714
3715
3716 036630 004537 020214      JSR      R5,CHERR      ;CHECK CNTLR FOR ERRORS
3717 036634 005737 002124      TST      T.CRC
3718 036640 001404      BEQ      4$
3719
3720 036642      ERRHRD 410.,ERR15,EM70
(3) 036642 104463      TRAP      T$ERCODE
(5) 036644 000632      .WORD    410
(5) 036646 015216      .WORD    ERR15
(5) 036650 014220      .WORD    EM70
3721
3722 036652      4$:
3723
3724
3725 036652      10002$:      ENDSEG      ;%%END OF SEGMENT%%
(3) 036652      EMT      C$ESEG
(3) 036652 104005      ENDSEG      ;%%END OF SEGMENT%%
3726 036654      10001$:
(3) 036654 104005      EMT      C$ESEG
3727
3728 036656 005723      TST      (R3)+
3729 036660 020327 002660      CMP      R3,#HDREND
3730 036664 001260      BNE      298$
3731
3732 036666      10000$:      ENDSEG      ;%%END OF SEGMENT%%
(3) 036666      EMT      C$ESEG
(3) 036666 104005      ENDTST
3733 036670      L10077:      ;**END OF TEST**
(3) 036670      EMT      C$TST
(3) 036670 104001      .SBTTL      **TEST 43** - READ WITHOUT HEADER (COMPARE FUNCTION)
3734

```

```

3735
3736 036672          STARS
(2)                ;*****
3737                ;TEST THAT READ WITHOUT HEADER VERIFICATION WORKS. THIS FUNCTION SHOULD
3738                ;READ AT THE NEXT SECTOR ENCOUNTERED. SET THE RLDA TO 0
3739                ;AND ISSUE THE FUNCTION IN FLAG MODE. UPON COMPLETION CHECK
3740                ;FOR ERRORS
3741 036672          STARS
(2)                ;*****
3742 036672          BGNTST                                ;**START OF TEST**
3743
3744
3745 036672 004737 021356      JSR    PC,HDHOME          ;HEADS OVER TRACK 0
3746 036676          CKERFG                                ;HEADS GO HOME OKAY
(4) 036704 104032          EMT    C$EXIT
(4) 036706 000052          .WORD  L10100-.
3747
3748 036710          BGNSEG                                ;%%START OF SEGMENT%%
(3) 036710 104004          EMT    C$BSEG
3749
3750
3751 036712 012777 177600 143330  MOV    #-128.,@RLMP      ;SET UP WORD COUNT
3752 036720 012777 003052 143316  MOV    #BUF,@RLBA      ;SETUP BUS ADDRESS
3753 036726 012777 177777 143312  MOV    #-1,@RLDA      ;HEADER SHOULDN'T MATTER
3754 036734 004537 020456          JSR    R5,LDFUNC        ;LOAD THE FUNCTION IN NEXT WORD
3755 036740 000016          RDNHD                          ;READ DATA WITHOUT HEADER VERIFY
3756 036742 004537 021276          JSR    R5,WTCRDY        ;WAIT FOR IT TO FINISH
3757 036746          ESCAPE SEG                            ;CHECK FOR FL:LOE, ELSE EXIT SEG
(3) 036746 104010          EMT    C$ESCAPE
(3) 036750 000006          .WORD  10000$-.
3758
3759 036752 004537 020214          JSR    R5,CHERR          ;CHECK CNTLR FOR ERRORS
3760
3761 036756          ENDSEG                                ;%%END OF SEGMENT%%
(3) 036756          10000$:
(3) 036756 104005          EMT    C$ESEG
3762 036760          ENDTST                                ;**END OF TEST**
(3) 036760          L10100:
(3) 036760 104001          EMT    C$ETST
3763
3764          .SBTTL **TEST 44** - READ WITHOUT HEADER COMPARE FUNCTION INTERRUPT
3765
3766 036762          BGNTST                                ;**START OF TEST**
3767
3768 036762          STARS
(2)                ;*****
3769                ;TEST THAT READ WITHOUT HEADER VERIFICATION WORKS IN
3770                ;INTERRUPT MODE.
3771 036762          STARS
(2)                ;*****
3772
3773 036762 004737 021356      JSR    PC,HDHOME          ;HEADS OVER TRACK 0
3774 036766          CKERFG                                ;HEADS GO HOME OKAY
(4) 036774 104032          EMT    C$EXIT
(4) 036776 000114          .WORD  L10101-.
3775

```

```

3776 037000          BGNSEG          ;%%START OF SEGMENT%%
(3) 037000 104004    EMT      C$BSEG
3777
3778 037002 005037 002144    CLR      INTFLG ;CLEAR INTERRUPT OCCURANCE FLAG
3779 037006 012777 177600 143234    MOV      #-128.,@RLMP ;SET UP WORD COUNT FOR ONE SECTOR
3780 C37014 012777 003052 143222    MOV      #BUF,@RLBA ;SETUP BUFFER ADDRESS
3781 037022 012777 177777 143216    MOV      #-1,@RLDA ;DISK ADDRESS IS A DON'T CARE
3782 037030
(3) 037030 012700 000000    SETPRI   #PRI00
(3) 037034 104041    MOV      #PRI00,R0
3783 037036 004537 020456    EMT      C$SPRI
3784 037042 000116    JSR      R5,LDFUNC ;LOAD THE FUNCTION IN NEXT WORD
3785 037044 004537 021276    RDNHD:INTEN ;INTERRUPT ENABLED
3786 037050    JSR      R5,WTCRDY ;WAIT FOR INTERRUPT
(3) 037050 012700 000340    SETPRI   #PRI07
(3) 037054 104041    MOV      #PRI07,R0
3787 037056    EMT      C$SPRI
(3) 037056 104010    ESCAPE   SEG ;CHECK FOR FL:LOE, ELSE EXIT SEG
(3) 037060 000030    EMT      C$ESCAPE
3788    .WORD      10000$-.
3789 037062 005737 002144    'ST      INTFLG ;DID IT INTERRUPT
3790 037066 001004    BNE      1$ ;IF INTERRUPT GO TO 1$
3791
3792 037070    ERRDF      40.,EM40,ERRO ;NO INTERRUPT
(3) 037070 104462    TRAP      T$ERCODE
(5) 037072 000050    .WORD      40
(5) 037074 012547    .WORD      EM40
(5) 037076 014244    .WORD      ERRO
3793 037100    ' $:    ESCAPE   SEG ;CHECK FOR FL:LOE, ELSE EXIT SEG
(3) 037100 104010    EMT      C$ESCAPE
(3) 037102 000006    .WORD      10000$-.
3794
3795 037104 004537 020214    JSR      R5,CHERR ;CHECK CNTLR FOR ERRORS
3796
3797 037110    ENDSEG ;%%END OF SEGMENT%%
(3) 037110 10000$:
(3) 037110 104005    EMT      C$ESEG
3798 037112    ENDTST ;**END OF TEST**
(3) 037112 L10101:
(3) 037112 104001    EMT      C$ETST
3799
3800    .SBTTL **TEST 45** - CHECK RD W/O HDR CMP ACTUALLY READS
3801
3802 037114    BCNTST ;**START OF TEST**
3803
3804 037114    STARS
(2)
3805    ;*****
3806    ;CHECK THAT THE READ W/O HDR CMP FUNCTION ACTUALLY READS (INTO MEMORY)
3807    ;WE WILL WRITE A PATTERN INTO MEMORY AND THEN ISSUE
3808    ;A READ TO OVERLAY THAT PATTERN. AFTER THE READ
3809    ;WE CHECK TO SEE IF THE WRITTEN PATTERN HAS CHANGED.
3810    ;IF NOT WE ISSUE IT AGAIN AT THE SAME SECTION AFTER
3811    ;HAVING MODIFIED OUR PATTERN IN MEMORY (SINCE THERE IS
3812    ;ONE CHANCE THAT THE DISK COULD HAVE OUR PATTERN). AFTER
3813    ;THE SECOND READ WE CHECK THE BUFFER AGAIN. IF IT'S
    ;NOT CHANGED WE REPORT AN ERROR
  
```

```

3814 037114          STARS
      (2)          ;:*****
3815
3816
3817 037114 004737 021356      JSR      PC,HDHOME      ;HEADS OVER TRACK 0
3818 037120          CKERFG      ;HEADS GO HOME OKAY
      (4) 037126 104032      EMT      C$EXIT
      (4) 037130 000160      .WORD    L10102-.
3819
3820 037132          BGNSEG          ;%%START OF SEGMENT%%
      (3) 037132 104004      EMT      C$BSEG
3821
3822 037134 012737 024350 002160      MOV      #24350,TMP0      ;SET PATTERN TO WRITE
3823 037142 005037 002162          CLR      TMP1      ;CLEAR PASS INDICATOR
3824 037146 012700 003052      1$:      MOV      #BUF,R0      ;SET UP BUFFER BEGINNING
3825 037152 012701 000200          MOV      #128,R1
3826 037156 013720 002160      2$:      MOV      TMP0,(R0)+      ;WRITE BUFFER
3827 037162 005301          DEC      R1      ;DONE??
3828 037164 001374          BNE      2$      ;NO, GO BACK
3829 037166 012777 000050 143052      MOV      #40,@RLDA      ;LOAD DISK ADDRESS TO NONSENSE
3830 037174 012777 177600 143046      MOV      #-128,@RLMP      ;SET WORD COUNT
3831 037202 012777 003052 143034      MOV      #BUF,@RLBA      ;LOAD BUS ADDRESS
3832 037210 012737 003052 002166      MOV      #BUF,GDDAT      ;FOR ERROR PRINT
3833
3834 037216 004537 020456          JSR      R5,LDFUNC      ;LOAD THE FUNCTION IN NEXT WORD
3835 037222 000016          RDNHD      ;READ W/O HDR CMP
3836 037224 004537 021276          JSR      R5,WTCRDY      ;WAIT FOR CONTROLLER READY
3837 037230          ESCAPE      SEG      ;CHECK FOR FL:LOE, ELSE EXIT SEG
      (3) 037230 104010      EMT      C$ESCAPE
      (3) 037232 000054      .WORD    10000$-.
3838
3839 037234 004537 020214          JSR      R5,CHERR
3840 037240          ESCAPE      SEG      ;CHECK CNTLR FOR ERRORS
      (3) 037240 104010      EMT      C$ESCAPE      ;CHECK FOR FL:LOE, ELSE EXIT SEG
      (3) 037242 000044      .WORD    10000$-.
3841
3842 037244 012702 003052          MOV      #BUF,R2      ;SET TO START COMPARING DATA
3843 037250 022237 002160      4$:      CMP      (R2)+,TMP0      ;DID DATA CHANGE?
3844 037254 001014          BNE      6$      ;YES, CHECK FOR END
3845
3846
3847          ;DATA DIDN'T CHANGE, CHECK
3848 037256 005737 002162          TST      TMP1      ;IF 1ST OR 2ND TIME?
3849 037262 001005          BNE      5$      ;2ND-REPORT 1ST-TRY AGAIN
3850
3851 037264 005237 002162          INC      TMP1      ;INC PASS COUNT
3852 037270 005137 002160          COM      TMP0      ;COMPLIMENT PATTERN
3853 037274 000724          BR      1$      ;GO DO IT AGAIN
3854
3855          5$:      ERRDF      20,EM55,ERR9
      (3) 037276 104462      TRAP      T$ERRCODE
      (5) 037300 000024      .WORD    20
      (5) 037302 013230      .WORD    EM55
      (5) 037304 014636      .WORD    ERR9
3856
3857          6$:
  
```

```

3858
3859 037306          10000$: ENDSEG          ;%%END OF SEGMENT%%
(3) 037306
(3) 037306 104005    EMT      C$ESEG
3860 037310          END1ST          ;**END OF TEST**
(3) 037310          L10102:
(3) 037310 104001    EMT      C$E1
3861
3862          .SBTTL  **TEST 46** - CHECK RLBA INCREMENT WITH RD W/O HDR CMP
3863
3864 037312          BGN1ST          ;**START OF TEST**
3865
3866 037312          STARS
(2)          ;*****
3867          ;CHECK THAT THE RLBA WILL INCREMENT WITH THE READ W/O HDR CMP
3868          ;THE RLBA SHOULD CONTAIN 'BUF +256.' AFTER A FULL SECTOR
3869          ;READ.
3870 037312          STARS
(2)          ;*****
3871
3872
3873 037312 004737 021356 JSR      PC,HDHOME      ;HEADS OVER TRACK 0
3874 037316          CKERFG          ;HEADS GO HOME OKAY
(4) 037324 104032    EMT      C$EXIT
(4) 037326 000120    .WORD     L10103-.
3875
3876 037330          BGNSEG          ;%%START OF SEGMENT%%
(3) 037330 104004    EMT      C$BSEG
3877
3878 037332 012777 000050 142706 MOV     #40.,@RLDA
3879 037340 012777 003052 142676 MOV     #BUF,@RLBA      ;SET UP BUS ADDRESS
3880 037346 012777 177600 142674 MOV     #-128.,@RLMP    ;WORD COUNT
3881 037354 012737 003052 002166 MOV     #BUF,GDDAT      ;FORM EXPECTED BUS ADDRESS
3882 037362 062737 000400 002166 ADD      #256.,GDDAT    ;AFTER READ
3883
3884 037370 004537 020456 JSR      R5,LDFUNC      ;LOAD THE FUNCTION IN NEXT WORD
3885 037374 000016 RDNHD          ;READ W/O HDR CMP
3886 037376 004537 021276 JSR      R5,WTCRDY      ;WAIT FOR CONTROLLER READY
3887 037402          ESCAPE SEG      ;CHECK FOR FL:LOE, ELSE EXIT SEG
(3) 037402 104010    EMT      C$ESCAPE
(3) 037404 000040    .WORD     10000$-.
3888
3889 037406 004537 020214 JSR      R5,CHERR      ;CHECK CNTLR FOR ERRORS
3890 037412          ESCAPE SEG      ;CHECK FOR FL:LOE, ELSE EXIT SEG
(3) 037412 104010    EMT      C$ESCAPE
(3) 037414 000030    .WORD     10000$-.
3891 037416 013737 002230 002170 MOV     E.BA,BDDAT      ;READ 'RLBA' FOR PRESENT ADDRESS
3892 037424 023737 002170 002166 CMP      BDDAT,GDDAT    ;DID 'BA' INCREMENT PROPERLY?
3893 037432 001404 BEQ      1$          ;YES, CONTINUE
3894
3895 037434          ERRDF 21.,EM53,ERR4
(3) 037434 104462 TRAP     T$ERCODE
(5) 037436 000025 .WORD     21
(5) 037440 013310 .WORD     EM53
(5) 037442 014410 .WORD     ERR4
3896

```

```

ASSEMBLY ROUTINES          MACY11 30A(1052) 22-NOV-78 15:35 PAGE 1-97 K 10
CZRLBB.P11 22-NOV-78 15:28 **TEST 46** - CHECK RLBA INCREMENT WITH RD W/O HDR CMP
                                                                    SEQ 0127

3897 037444          1$:
3898
3899 037444          ENDSEG                                ;%%END OF SEGMENT%%
(3) 037444          10000$:
(3) 037444 104005    EMT      C$ESEG
3900 037446          ENDTST                                ;**END OF TEST**
(3) 037446          L10103:
(3) 037446 104001    EMT      C$ETST
3901
3902
3903
3904
3905
3906
3907
3908
3909
3910 037450          .SBTTL  **TEST 47** - CHECK RLDA DOES INCREMENT WITH RD W/O HDR CMP
3911 BGNTST                                                ;**START OF TEST**
3912 037450          STARS
(2) ;*****
3913 ;CHECK THAT THE RLDA DOES INCREMENT BY ONE AFTER A
3914 ;FULL SECTOR READ W/O HDR CMP
3915 ;AFTER THE READ THE RLDA SHOULD STILL BE THE INITIAL RLDA + 1
3916 037450          STARS
(2) ;*****
3917
3918 037450 004737 021356 JSR      PC,HDHOME          ;HEADS OVER TRACK 0
3919 037454          CKERFG          ;HEADS GO HOME OKAY
(4) 037462 104032    EMT      C$EXIT
(4) 037464 000116    .WORD     L10104-.
3920
3921 037466          BGNSEG                                ;%%START OF SEGMENT%%
(3) 037466 104004    EMT      C$BSEG
3922
3923
3924 037470 012737 000050 002166 MOV     #40.,GDDAT          ;DA TO NONSENSE
3925 037476 013777 002166 142542 MOV     GDDAT,@RLDA        ;SETUP DISK ADDRESS
3926 037504 005237 002166          INC     GDDAT
3927 037510 012777 177600 142532 MOV     #-128.,@RLMP        ;WORD COUNT
3928 037516 012777 003052 142520 MOV     #BUF,@RLBA        ;SETUP BUS ADDRESS
3929
3930 037524 004537 020456 JSR      R5,LDFUNC          ;LOAD THE FUNCTION IN NEXT WORD
3931 037530 000016          RDNHD          ;READ WITHOUT HEADER COMPARE
3932 037532 004537 021276 JSR      R5,WTCRDY        ;WAIT FOR CONTROLLER READY
3933 037536          ESCAPE SEG          ;CHECK FOR FL:LOE, ELSE EXIT SEG
(3) 037536 104010    EMT      C$ESCAPE
(3) 037540 000040    .WORD     10000$-.
3934
3935 037542 004537 020214 JSR      R5,CHERR          ;CHECK CNTLR FOR ERRORS
3936 037546          ESCAPE SEG          ;CHECK FOR FL:LOE, ELSE EXIT SEG
(3) 037546 104010    EMT      C$ESCAPE
(3) 037550 000030    .WORD     10000$-.
3937
3938 037552 013737 002232 002170 MOV     E,DA,BDDAT        ;READ DISK ADDRESS
3939 037560 023737 002166 002170 CMP      GDDAT,BDDAT        ;DID SECTOR INCREMENT PROPERLY

```

```

3940 037566 001404      BEQ      1$              ;YES, BRANCH NO, REPORT ERROR
3941
3942 037570      ERRDF    22.,EM54,ERR4
      (3) 037570 104462      TRAP    T$ERCODE
      (5) 037572 000026      .WORD   22
      (5) 037574 013400      .WORD   EM54
      (5) 037576 014410      .WORD   ERR4
3943
3944 037600      1$:
3945
3946 037600      ENDSEG                      ;%%END OF SEGMENT%%
      (3) 037600
      (3) 037600 104005      10000$:
3947 037602      EMT      C$ESEG
      (3) 037602
      (3) 037602 104007      ENDTST?
      L10104:                  ;**END OF TEST**
                               EMT      C$ETST
3948
3949
3950
3951
3952 037604      BGVMOD  HRDPRM
3953
3954 037604      BGNHRD
      (3) 037604 000025      .WORD L10105-L$HARD/2
3955
3956 037606      GPRML    CNTYPE,CNT,1,YES
      (4) 037606 004130      .WORD   T$CODE
      (4) 037610 037660      .WORD   CNTYPE
      (4) 037612 000001      .WORD   1
3957 037614      GPRMA    CSRMSG,CSR,0,160000,177776,YES
      (4) 037614 000031      .WORD   T$CODE
      (4) 037616 037665      .WORD   CSRMSG
      (4) 037620 160000      .WORD   T$LLOLIM
      (4) 037622 177776      .WORD   T$HILIM
3958 037624      GPRMA    VECMSG,VECT,0,0,776,YES
      (4) 037624 001031      .WORD   T$CODE
      (4) 037626 037712      .WORD   VECMSG
      (4) 037630 000000      .WORD   T$LLOLIM
      (4) 037632 000776      .WORD   T$HILIM
3959 037634      GPRMD    BRMSG,PRIOR,0,340,0,7,YES
      (4) 037634 002032      .WORD   T$CODE
      (4) 037636 037701      .WORD   BRMSG
      (4) 037640 000340      .WORD   340
      (4) 037642 000000      .WORD   T$LLOLIM
      (4) 037644 000007      .WORD   T$HILIM
3960 037646      CPRMD    DRMSG,DRBT,0,03400,0,7,YES
      (4) 037646 003032      .WORD   T$CODE
      (4) 037650 037721      .WORD   DRMSG
      (4) 037652 003400      .WORD   03400
      (4) 037654 000000      .WORD   T$LLOLIM
      (4) 037656 000007      .WORD   T$HILIM
3961
3962 037660      ENDHRD
      (2)
      (3) 037660      10105:
3963
  
```

```

3964 037660 046122 030461 000 CNTYPE: .ASCIZ /RL11/
3965 037665 102 051525 040440 CSRMSC: .ASCIZ /BUS ADDRESS/
      037672 042104 042522 051523
      037700 000
3966 037701 102 020122 042514 BRMSG: .ASCIZ /BR LEVEL/
      037706 042526 000114
3967 037712 042526 052103 051117 VECMSG: .ASCIZ /VECTOR/
      037720 000
3968 037721 104 044522 042526 DRMSG: .ASCIZ /DRIVE/
      037726 000
3969 037730 .EVEN
3970
3971 037730 ENDMOD
3972
3973
3974 037730 BGNMOD SF TPRM
3975
3976 037730 BGNSFT
      (3) 037730 000025 .WORD L10106-L$SOFT/2
3977
3978 037732 GPRML DMSG,DLT,1,YES
      (4) 037732 000130 .WORD T$CODE
      (4) 037734 040004 .WORD DMSG
      (4) 037736 000001 .WORD 1
3979 037740 XFERF 1$
      (5) 037740 006044 .WORD T$CODE
3980 037742 GPRMD EMSG,ELT,D,177777,0,177777,YES
      (4) 037742 001052 .WORD T$CODE
      (4) 037744 040122 .WORD EMSG
      (4) 037746 177777 .WORD 177777
      (4) 037750 000000 .WORD T$LOLIM
      (4) 037752 177777 .WORD T$HILIM
3981 037754 1$: GPRML SMSG,SIZE,1,YES
      (4) 037754 002130 .WORD T$CODE
      (4) 037756 040030 .WORD SMSG
      (4) 037760 000001 .WORD 1
3982 037762 GPRML CMSG,DMPCK,1,YES
      (4) 037762 003130 .WORD T$CODE
      (4) 037764 040041 .WORD CMSG
      (4) 037766 000001 .WORD 1
3983 037770 XFERF 2$
      (5) 037770 006044 .WORD T$CODE
3984 037772 GPRMD LMSG,DLMT,D,177777,1,128.,YES
      (4) 037772 004052 .WORD T$CODE
      (4) 037774 040065 .WORD LMSG
      (4) 037776 177777 .WORD 177777
      (4) 040000 000001 .WORD T$LOLIM
      (4) 040002 000200 .WORD T$HILIM
3985 040004 2$:
3986
3987
3988 040004 ENDSFT
      (2) .EVEN
      (3) 040004 L10106:
3989
3990 040004 051104 050117 047440 DMSG: .ASCIZ /DROP ON ERROR LIMIT/
  
```

```

040012 020116 051105 047522
040020 020122 044514 044515
040026 000124
3991 040030 052501 047524 044523 SMSG: .ASCIZ /AUTOSIZE/
040036 042532 000
3992 040041 103 046517 040520 CMSG: .ASCIZ /COMPARE DATA ON DCK/
040046 042522 042040 052101
040054 020101 047117 042040
040062 045503 000
3993 040065 043 047440 020106 LMSG: .ASCIZ /# OF WORDS IN ERROR REPORTED/
040072 047527 042122 020123
040100 047111 042440 051122
040106 051117 051040 050105
040114 051117 042524 000104
3994 040122 051105 047522 020122 EMSG: .ASCIZ /ERROR LIMIT/
040130 044514 044515 000124
3995
3996 040136 ENDMOD
3997
3998
3999
4000 040514 .=40514
4001
4002 ;AREA RESERVED AS PATCH AREA FOR DIAGNOSTICS.
4003 ;.-40514 WAS SELECTED AS 'LASTAD' TO PROVIDE APT TO LSI-11 COMPATIBILITY.
4004 ;BIT 7 OF 'LASTAD' MUST BE CLEARED TO ACHIEVE A VALID MAILBOX ADDRESS
4005 ;WHEN RUNNING ON THE LSI-11 UNDER APT.
4006
4007 040514 LASTAD
(2) .EVEN
(3) 040514 L$LAST::
4008
4009
4010
4011
4012
4013

```

```
4015  
14886 071310 000000 .SBTTL DIAGNOSTIC SUPERVISOR -- LOW CORE SET UP  
14887 071312 000000 .WORD 0 :SPACE FOR USER POOL POINTER  
14888 071314 000000 .WORD 0 :SIZE  
14889 071316 000000 .WORD 0 :CHECKSUM (NOT CURRENTLY USED)  
14890 071322 .END.SUPV-.+2 :SIZE OF H.W. PTAB. ALLOCATION  
14891 000200 .END 200
```

|                  |                 |                 |                 |                  |
|------------------|-----------------|-----------------|-----------------|------------------|
| ABOFLA 041040 G  | BIT9 = 001000 G | CONTIN 017226   | C\$POIN= 000040 | DSAAL 057572     |
| ABOPAS 040756 G  | BLD.HW 046202   | CRDY = 000200   | C\$QIO = 000377 | DSAAM 057602     |
| ABO.FM 043320    | BLOCK 063614    | CRLF = 060172   | C\$RDBU= 000007 | EF.CON= 000036 G |
| AFREG 007151     | BPRIOR 002256   | CRTIM 007172    | C\$REFG= 00005C | EF.NEW= 000035 G |
| AFSI 040546 G    | BRMSG 037701    | CSR = 000000    | C\$REQT= 000045 | EF.PWR= 000034 G |
| AFTER 021016     | BUF 003052      | CSRMSG 037665   | C\$RESE= 000033 | EF.RES= 000037 G |
| ALLOC 061460     | BVEC 002254     | CURR.S 040522 G | C\$REVI= 000002 | EF.STA= 000040 G |
| APT.ER 042450    | B\$AAB 047604   | CURR.T 040524 G | C\$RPT = 000025 | EF01 = 000001 G  |
| ARLBA 007106     | B\$AAF 047516   | CYLSK 002202    | C\$SEFG= 000047 | EF02 = 000002 G  |
| ARLCS 007101     | B.BA 002220     | C\$AAD 053062   | C\$SPRI= 000041 | EF03 = 000003 G  |
| ARLDA 007114     | B.CS 002216     | C\$AAE 053074   | C\$SVEC= 000037 | EF04 = 000004 G  |
| ARLMP 007122     | B.DA 002222     | C\$AAK 054072   | C\$TPRI= 000013 | EF05 = 000005 G  |
| ASSEMB= 000010   | B.MP 002224     | C\$AAL 054236   | C\$UNBU= 000031 | EF06 = 000006 G  |
| ASAAV 045316     | CALBCC 002156   | C\$ABRT= 000021 | C\$WTM = 000026 | EF07 = 000007 G  |
| ASAAW 045332     | CALLPC= 000022  | C\$ADR = 000020 | C\$WTU = 000027 | EF08 = 000010 G  |
| ASAAZ 045344     | CALLPS= 000024  | C\$AU = 000054  | DAHS = 000020   | EF09 = 000011 G  |
| ASAAZ 045352     | CALLSP= 000026  | C\$BRK = 000022 | DATPAT 002662   | EF10 = 000012 G  |
| ASAAZ 045366     | CALLTC= 000030  | C\$BSEG= 000004 | DCKMES 007277   | EF11 = 000013 G  |
| ASABA 045376     | CAL.CL 066202   | C\$BSUB= 000002 | DECMES 060004   | EF12 = 000014 G  |
| BA16 = 000020    | CAL.TI 066240 G | C\$BUFG= 000030 | DEMES 007245    | EF13 = 000015 G  |
| BA17 = 000040    | CDCNT 002130    | C\$CEFG= 000046 | DERFLG 002310   | EF14 = 000016 G  |
| BCCFBK 002154    | CHECK 002122    | C\$CLEA= 000012 | DERR = 040000   | EF15 = 000017 G  |
| BCSR 002252      | CHERR 020214    | C\$CLP1= 000006 | DEV.CO 040526 G | EF16 = 000020 G  |
| BDDAT 002170     | CHKLUP 047620   | C\$CVEC= 000036 | DIAGMC= 000000  | ELT = 000002     |
| BEFORE 020764    | CHKSTR 062022   | C\$DCLN= 000044 | DIAG.T 041046 G | EMSG 040122      |
| BEREG 007130     | CHKTTY 060110   | C\$DODU= 000053 | DLMT = 000010   | EMT.TR 041044 G  |
| BGN.SU= 040514   | CHK.MA 045760   | C\$DRPT= 000024 | DLT = 000000    | EM1 010352       |
| BINMSG 057770    | CHK.PC 053110   | C\$DU = 000055  | DLTMS 007304    | EM10 011043      |
| BIT0 = 000001 G  | CHK.SW 042150   | C\$EDIT= 000002 | DMPCK = 000006  | EM100 010417     |
| BIT00 = 000001 G | CHRCNT 061342   | C\$ERDF= 000002 | DMSG 040004     | EM11 011110      |
| BIT01 = 000002 G | CH.FLA 045466   | C\$ERHR= 000003 | DPDVD 070456 G  | EM12 011137      |
| BIT02 = 000004 G | CH.PAS 045504   | C\$ERSF= 000001 | DPMUL 070344 G  | EM13 011176      |
| BIT03 = 000010 G | CKERLT 020126   | C\$ERSO= 000004 | DRBT = 000006   | EM14 011230      |
| BIT04 = 000020 G | CLEAR. 047102   | C\$ESCA= 000010 | DRDY = 000001   | EM16 011312      |
| BIT05 = 000040 G | CLKACC 040754 G | C\$ESEG= 000005 | DRIVE 002134    | EM17 011351      |
| BIT06 = 000100 G | CLKBFR 066204   | C\$ESUB= 000003 | DRMSG 037721    | EM20 011411      |
| BIT07 = 000200 G | CLKCNT 040752 G | C\$ETST= 000001 | DROP 016766     | EM21 011466      |
| BIT08 = 000400 G | CLKJUM 066610 G | C\$EXIT= 000032 | DRPCOD 020104 G | EM22 011544      |
| BIT09 = 001000 G | CLKRES 067612 G | C\$GMAN= 000043 | DRST = 000010   | EM23 011636      |
| BIT1 = 000002 G  | CLKSER 067746 G | C\$GPHR= 000042 | DRTIM 007217    | EM24 011716      |
| BIT10 = 002000 G | CLKSON 041012 G | C\$GPRI= 000040 | DSPCOD 017000 G | EM25 011774      |
| BIT11 = 004000 G | CLK.SE 045562   | C\$GTIM= 000052 | DS0 = 000000    | EM26 012045      |
| BIT12 = 010000 G | CLNCOD 020010 G | C\$INIT= 000011 | DS1 = 000400    | EM27 012105      |
| BIT13 = 020000 G | CLR.MA 046036   | C\$INLP= 000020 | DS2 = 001000    | EM30 012165      |
| BIT14 = 040000 G | CMSG 040041     | C\$KWOF= 000035 | DS3 = 001400    | EM31 012225      |
| BIT15 = 100000 G | CNT = 000010    | C\$KWON= 000034 | DUNIT. 040762 G | EM32 012270      |
| BIT2 = 000004 G  | CNTYPE 037660   | C\$LOOP= 000100 | DVC.FT 054042   | EM33 012335      |
| BIT3 = 000010 G  | CNVT 064260     | C\$MANI= 000051 | DWORD 002210    | EM34 012376      |
| BIT4 = 000020 G  | COMMAN 040564 G | C\$MSG = 000023 | DSAAG 054746    | EM35 012441      |
| BIT5 = 000040 G  | COMMTA 064074   | C\$PNTB= 000014 | DSAAG 054764    | EM36 012506      |
| BIT6 = 000100 G  | COMP 007316     | C\$PNTF= 000017 | DSAAI 057532    | EM4 010607       |
| BIT7 = 000200 G  | CONT 017354     | C\$PNTS 000016  | DSAAJ 057536    | EM40 012547      |
| BIT8 = 000400 G  | CONTCL 067672 G | C\$PNTX= 000015 | DSAAK 057554    | EM41 012612      |

|         |        |          |        |   |           |        |          |        |         |        |   |
|---------|--------|----------|--------|---|-----------|--------|----------|--------|---------|--------|---|
| EM42    | 012654 | ERR5     | 014456 | G | F\$MOD =  | 000000 | HDREND   | 002660 | LPBFR   | 040622 | G |
| EM43    | 012723 | ERR6     | 014514 | G | F\$MSG =  | 000011 | HDRLST   | 020712 | LPCNTR  | 040620 | G |
| EM44    | 012767 | ERR7     | 014556 | G | F\$PWR =  | 000017 | HDRTAB   | 002504 | LPT.AD  | 045144 |   |
| EM45    | 013024 | ERR8     | 014564 | G | F\$RPT =  | 000012 | HERTZ.   | 045126 | LPT.RE  | 045140 |   |
| EM47    | 013056 | ERR9     | 014636 | G | F\$SEG =  | 000003 | HAFMES   | 007272 | LSI.RE  | 045134 |   |
| EM5     | 010646 | ESC.PC   | 053106 |   | F\$SOFT=  | 000005 | HOLDSP=  | 000020 | LUP     | 066106 |   |
| EM50    | 013105 | EV.COU   | 040520 | G | F\$SRV =  | 000010 | HPTCOD   | 016750 | LUP.AD  | 053112 |   |
| EM51    | 013145 | E.BA     | 002230 |   | F\$SUB =  | 000002 | HRDPRM   | 037604 | L\$APT  | 002036 | G |
| EM52    | 013177 | E.CS     | 002226 |   | F\$SW =   | 000014 | HW.ADR   | 040550 | L\$AUT  | 002074 | G |
| EM53    | 013310 | E.DA     | 002232 |   | F\$TEST=  | 000001 | H\$AAB   | 064606 | L\$CCP  | 002106 | G |
| EM54    | 013400 | E.MP     | 002234 |   | GARBAG    | 061344 | ININIT   | 040772 | L\$CLEA | 020010 | G |
| EM55    | 013230 | E.MP1    | 002236 |   | GDDAT     | 002166 | INITCO   | 017140 | L\$CO   | 002032 | G |
| EM56    | 013456 | E.MP2    | 002240 |   | GETCHR    | 060050 | INITIA   | 060020 | L\$DEPO | 002011 | G |
| EM57    | 013477 | FILL     | 060640 |   | GETCMN    | 063434 | INIT.M   | 046104 | L\$DESC | 002102 | G |
| EM6     | 010712 | FILL.C   | 000204 | G | GETPAR    | 055126 | INIT.R   | 040606 | L\$DEVP | 002064 | G |
| EM60    | 013537 | FIRST    | 002200 |   | GETSWI    | 062430 | INPUTA   | 060746 | L\$DISP | 017002 | G |
| EM61    | 013575 | FIX      | 020752 |   | GET.TW    | 062200 | INTEN =  | 000100 | L\$DR   | 002112 | G |
| EM62    | 013652 | FLAGS    | 040560 | G | GLBDAT    | 002122 | INTFLG   | 002144 | L\$DRCT | 002070 | G |
| EM63    | 013727 | FLAGS1   | 040562 | G | GLBEQA    | 002122 | INTFOR   | 054244 | L\$DRS  | 002072 | G |
| EM64    | 014031 | FLAGTA   | 064012 |   | GLBERR    | 014244 | INTSRV   | 020110 | L\$DRST | 002112 | G |
| EM65    | 014105 | FLAG.I   | 045546 |   | GLBSUB    | 020110 | INVAL.   | 045052 | L\$DTP  | 002040 | G |
| EM66    | 014146 | FLA.SE   | 063760 |   | GLBTXT    | 007052 | INVINT   | 054102 | L\$DU   | 020104 | G |
| EM7     | 010766 | FLG.MA   | 045506 |   | GODRVR=   | 000202 | INV.SW   | 042104 | L\$DUT  | 002076 | G |
| EM70    | 014220 | FNDFNC   | 002260 |   | G\$BIT =  | 000002 | IN.SUF   | 047054 | L\$DVTY | 002114 | G |
| END     | 017722 | FORM.T   | 054410 |   | G\$STAT = | 000004 | I\$AU =  | 000041 | L\$EF   | 002056 | G |
| END.OF  | 047070 | FREE     | 061716 |   | G\$TINT   | 010006 | I\$CLN = | 000041 | L\$EFLG | 002034 | G |
| END.SU- | 071322 | FRMT1    | 015606 |   | G\$TMES   | 007747 | I\$DU =  | 000041 | L\$EXP1 | 002042 | G |
| ENVIRO  | 040566 | FRMT10   | 016415 |   | G\$EXCP=  | 000400 | I\$HRD = | 000041 | L\$EXP2 | 002044 | G |
| EOP.CH  | 067770 | FRMT11   | 016550 |   | G\$HILI=  | 000002 | I\$INIT= | 000041 | L\$EXP3 | 002046 | G |
| EOP.FM  | 043334 | FRMT13   | 016672 |   | G\$LOLI=  | 000001 | I\$MOD = | 000041 | L\$HARD | 037606 | G |
| EOP.IN  | 045500 | FRMT14   | 016241 |   | G\$NO =   | 000000 | I\$MSG = | 000041 | L\$HPCP | 002016 | G |
| ERCOUN  | 002314 | FRMT15   | 016723 |   | G\$OFFS=  | 000400 | I\$PWR = | 000041 | L\$HPTP | 002022 | G |
| ERFLG   | 002266 | FRMT2    | 015645 |   | G\$OF SI- | 000376 | I\$RPT = | 000041 | L\$HW   | 016752 | G |
| ERPOIN  | 002312 | FRMT2A   | 015664 |   | G\$PRMA=  | 000001 | I\$SEG = | 000041 | L\$ICP  | 002104 | G |
| ERR     | 100000 | FRMT2B   | 015677 |   | G\$PRMD=  | 000002 | I\$SFT = | 000041 | L\$INIT | 017140 | G |
| ERRFOR  | 054314 | FRMT3    | 015726 |   | G\$PRML=  | 000000 | I\$SRV = | 000041 | L\$LADP | 002026 | G |
| ERRHAN  | 053114 | FRMT4    | 015733 |   | G\$RADA=  | 000140 | I\$SUB = | 000041 | L\$LAST | 040514 | G |
| ERRVEC  | 002132 | FRMT5    | 015771 |   | G\$RADB=  | 000000 | I\$TST = | 000041 | L\$MREV | 002050 | G |
| ERR.HR  | 054052 | FRMT6    | 016042 |   | G\$RADF=  | 000040 | J\$JMP = | 000167 | L\$NAME | 002000 | G |
| ERR.NU  | 040516 | FRMT7    | 016117 |   | G\$RADL=  | 000120 | KBPTR    | 040624 | L\$REPP | 002066 | G |
| ERR.SF  | 054056 | FRMT8    | 016171 |   | G\$RADO=  | 000020 | KBUF     | 040626 | L\$REV  | 002010 | G |
| ERR0    | 014244 | FRMT9    | 016312 |   | G\$RADT=  | 000100 | LDCSR    | 002146 | L\$SOFT | 037732 | G |
| ERR1    | 014262 | FRMT98   | 016615 |   | G\$XFER=  | 000004 | LDFUNC   | 020456 | L\$SPC  | 002062 | G |
| ERR1FO  | 054400 | FRMT99   | 016667 |   | G\$YES =  | 000010 | LF       | 007311 | L\$SPCP | 002020 | G |
| ERR10   | 014702 | F\$AU -  | 000015 |   | HCORED    | 045256 | LINE.F   | 041042 | L\$SPTP | 002024 | G |
| ERR11   | 014754 | F\$BGN = | 000040 |   | HCOREQ    | 045166 | LINE1    | 015256 | L\$STA  | 002030 | G |
| ERR12   | 015026 | F\$CLEA= | 000007 |   | HCORET    | 041002 | LINE2    | 015312 | L\$SW   | 016766 | G |
| ERR13   | 015102 | F\$DU =  | 000016 |   | HCRCME    | 007264 | LINE3    | 015534 | L\$TML  | 002014 | G |
| ERR14   | 015150 | F\$END = | 000041 |   | HC.ADR    | 040552 | LMSG     | 040065 | L\$TIMU | 002054 | G |
| ERR15   | 015216 | F\$HARD= | 000004 |   | HC.DEF    | 040544 | LOAD.F   | 045502 | L\$TIM1 | 002052 | G |
| ERR2    | 014274 | F\$HW -  | 000013 |   | HC.DIA    | 040542 | LOGMSG   | 060012 | L\$STI  | 002100 | G |
| ERR3    | 014336 | F\$INIT- | 000006 |   | HDHOME    | 021356 | LOFIMN   | 002272 | L\$UNIT | 002012 | G |
| ERR4    | 014410 | F\$JMP   | 000050 |   |           |        | LOPIMX   | 002270 | L.CLK.  | 045112 |   |

|        |        |        |          |         |          |          |          |          |          |          |        |  |
|--------|--------|--------|----------|---------|----------|----------|----------|----------|----------|----------|--------|--|
| L10000 | 014260 | L10065 | 033040   | NUM.LA  | 054602   | READ.P   | 066210   | G        | SWCHAN   | 045300   |        |  |
| L10001 | 014272 | L10066 | 033272   | NUM.NO  | 040556   | REGBAC   | 070700   | G        | SWITCH   | 064152   |        |  |
| L10002 | 014334 | L10067 | 033562   | NUM.UN  | 041164   | REGSAV   | 070664   | G        | SW.ADR   | 040554   | G      |  |
| L10003 | 014406 | L10070 | 034056   | NUNITS  | 047572   | REQN.P   | 040570   | G        | SW.PTA   | 045264   |        |  |
| L10004 | 014454 | L10071 | 034350   | NXM     | = 020000 | REQN.T   | 045462   |          | SYS.FT   | 054032   |        |  |
| L10005 | 014512 | L10072 | 034742   | NXMMES  | 007252   | REST     | 017304   |          | SSL SYM= | 010000   |        |  |
| L10006 | 014554 | L10073 | 035242   | NXT     | 017236   | RESTMS   | 020440   |          | TEMP2    | 002172   |        |  |
| L10007 | 014562 | L10074 | 035602   | NXTFOR  | 064252   | RE.SET   | 042252   |          | TEMP3    | 002174   |        |  |
| L10010 | 014634 | L10075 | 036114   | OCTMSG  | 057776   | RHDINT   | 007623   |          | TEMP4    | 002176   |        |  |
| L10011 | 014700 | L10076 | 036400   | OPI     | = 002000 | RHDMES   | 007563   |          | TERMI    | 066176   |        |  |
| L10012 | 014752 | L10077 | 036670   | OPIERR  | 007324   | RHHS     | = 000100 |          | TERMLI   | 064000   |        |  |
| L10013 | 015024 | L10100 | 036760   | OPIMES  | 007257   | RLBA     | 002244   |          | TERMTA   | 057762   |        |  |
| L10014 | 015100 | L10101 | 037112   | OPIMN   | 002300   | RLCS     | 002242   |          | TEST.M   | 045420   |        |  |
| L10015 | 015146 | L10102 | 037310   | OPIMX   | 002302   | RLDA     | 002246   |          | TIMFLG   | 040750   | G      |  |
| L10016 | 015214 | L10103 | 037446   | OSAPTS= | 000000   | RLMP     | 002250   |          | TIMSRV   | 020116   |        |  |
| L10017 | 015254 | L10104 | 037602   | OSAU    | = 000000 | RSTACK   | 070140   | G        | TIM.CO   | 040602   | G      |  |
| L10020 | 016764 | L10105 | 037660   | OSBGNR= | 000000   | SAVEDO=  | 042450   |          | TIM.OP   | 054406   |        |  |
| L10021 | 017000 | L10106 | 040004   | OSBGNS= | 000001   | SEARCH   | 062146   |          | TMP0     | 002160   |        |  |
| L10022 | 020006 | MAJ.IN | 040576   | G       | OSDU     | = 000001 | SECMSK   | 002150   | TMP1     | 002162   |        |  |
| L10023 | 020102 | MAJ.LO | 066206   |         | OSGNSW=  | 000001   | SEEK     | = 000006 | TMP2     | 002164   |        |  |
| L10024 | 020106 | MAJ.US | 040600   | G       | OSPOIN=  | 000001   | SEGSTA   | 041014   | G        | TOO.MA   | 057742 |  |
| L10025 | 020114 | MAN.TI | 001244   |         | PARSES   | 063506   | SEKINT   | 007715   |          | TRPFLG   | 002142 |  |
| L10026 | 022100 | MAP16  | 070714   | G       | PAP.LA   | 057474   | SEKMES   | 007664   |          | TRPHAN   | 021350 |  |
| L10027 | 022244 | MASK.B | 047616   |         | PASS.C   | 040530   | SET.MA   | 045672   |          | TRYFNC   | 002264 |  |
| L10030 | 022374 | MASK.W | 047614   |         | PRINTC   | 061320   | SFTPRM   | 037730   | G        | TST.AB   | 047730 |  |
| L10031 | 022530 | MAXCYL | 002212   |         | PRINTF   | 064626   | SHIFT    | 070776   | G        | TST.TO   | 042132 |  |
| L10032 | 022662 | MAXSEC | 002206   |         | PRIOR    | = 000004 | SIGN     | = 000004 |          | TYPEC    | 060336 |  |
| L10033 | 023020 | MDHEDR | 002000   | G       | PRI00    | = 000000 | SIMBCC   | 021064   |          | TYPEPC   | 054232 |  |
| L10034 | 023216 | MEM.SI | 045154   |         | PRI01    | = 000040 | SIZE     | = 000004 |          | TYPFLA   | 063654 |  |
| L10035 | 023520 | MERLMT | 016770   |         | PRI02    | = 000100 | SKHOME   | 010303   |          | TYPLIN   | 060234 |  |
| L10036 | 023710 | MIN.IN | 040572   | G       | PRI03    | = 000140 | SMG      | 040030   |          | TYPNUM   | 057616 |  |
| L10037 | 024106 | MIN.US | 040574   | G       | PRI04    | = 000200 | SPEC.U   | 045406   |          | TYPSTR   | 060254 |  |
| L10040 | 024260 | MK     | = 000001 |         | PRI05    | = 000240 | SPTCOD   | 016764   | G        | TYP.ER   | 054062 |  |
| L10041 | 024456 | MODR   | 070256   | G       | PRI06    | = 000300 | SPV.SE   | 000400   |          | TY.UNI   | 047074 |  |
| L10042 | 024656 | MSCRLF | 007313   |         | PRI07    | = 000340 | START    | 017244   |          | T\$ARGC= | 000004 |  |
| L10043 | 025026 | MSG.AD | 040540   | G       | PRNTST   | 061210   | STARTC   | 067666   | G        | T\$CODE= | 004052 |  |
| L10044 | 025130 | MSG.TY | 040514   | G       | PRO.CM   | 045460   | START1   | 017206   |          | T\$ERCO= | 000062 |  |
| L10045 | 025254 | MUL    | 070212   | G       | PTAB.S   | 041000   | STHS     | = 000100 |          | T\$ERN=  | 000026 |  |
| L10046 | 025450 | MXSEC1 | 002204   |         | PUTCHR   | 060024   | STRCHR   | 060700   |          | T\$EXCP= | 000000 |  |
| L10047 | 025604 | NEWPRI | 067736   | G       | PWRFLG   | 002304   | STRT.T   | 045464   |          | T\$FLAG= | 000040 |  |
| L10050 | 025736 | NEXTAR | 064176   |         | PWR.FA   | 071150   | ST.SET   | 042316   |          | T\$HILI= | 000200 |  |
| L10051 | 026056 | NOOPC  | = 000000 |         | PWR.FL   | 040604   | SUNIT.   | 045470   |          | T\$LOLI= | 000001 |  |
| L10052 | 026236 | NOPINT | 007430   |         | PWR.MS   | 071276   | SUPERV   | 043352   |          | T\$LSYM= | 010000 |  |
| L10053 | 027014 | NOPMES | 007377   |         | PWR.SA   | 071272   | SUPFLA   | 040760   | G        | T\$NEST= | 177777 |  |
| L10054 | 027210 | NOPWR  | 017166   |         | PWR.UP   | 071274   | SUPV.T   | 041132   | G        | T\$NSK0= | 000000 |  |
| L10055 | 027354 | NORDY  | 007070   |         | P.CLK.   | 045120   | SUP.PR   | 042070   |          | T\$NSK1= | 000005 |  |
| L10056 | 027470 | NORES  | 007052   |         | RDDINT   | 010076   | SVCGBL=  | 000000   |          | T\$NSK2= | 000003 |  |
| L10057 | 027646 | NO.CLK | 045102   |         | RDDMES   | 010045   | SVCHAN   | 050006   |          | T\$NSK3= | 000003 |  |
| L10060 | 030246 | NO.FLA | 063772   |         | RDHDR    | = 000010 | SVCINS=  | 000000   |          | T\$SAVL= | 177777 |  |
| L10061 | 030670 | NO.LPT | 061310   |         | RDNHD    | = 000016 | SVC SUB= | 177777   |          | T\$SEGL= | 177777 |  |
| L10062 | 031316 | NO.PTA | 045306   |         | RDNINT   | 010247   | SVCTAG=  | 000000   |          | T\$SEK0= | 010000 |  |
| L10063 | 031776 | NR     | = 000000 |         | RDNMES   | 010213   | SVCTST=  | 177777   |          | T\$SEK1= | 010001 |  |
| L10064 | 032430 | NUMBIN | 054434   |         | READ     | = 000014 | SVHD     | 002214   |          | T\$SEK2= | 010002 |  |

|                 |     |          |        |          |         |          |          |          |
|-----------------|-----|----------|--------|----------|---------|----------|----------|----------|
| TSSUBN= 000000  | T11 | 024110 G | T34    | 033274 G | USER.P  | 040774 G | XEQ.OP   | 047310   |
| T\$TAGL= 177777 | T12 | 024262 G | T35    | 033564 G | USER.T  | 040776 G | XEQ.PR   | 042510   |
| T\$TAGN= 010107 | T13 | 024460 G | T36    | 034060 G | UUT     | 002136   | XEQ.TE   | 047354   |
| T\$TEMP= 000000 | T14 | 024660 G | T37    | 034352 G | VALID.  | 041234   | XMEM     | 002262   |
| T\$TEST= 000057 | T15 | 025030 G | T38    | 034744 G | VAL.LA  | 042054   | XPOLY    | 002152   |
| T\$TSTM= 177777 | T16 | 025132 G | T39    | 035244 G | VAL.SW  | 045520   | XTIME    | 066676 G |
| T\$TSTS= 000001 | T17 | 025256 G | T4     | 022376 G | VECMSG  | 037712   | XTIMEN   | 067522   |
| T\$SCLE= 010023 | T18 | 025452 G | T40    | 035604 G | VECT =  | 000002   | XTIMST   | 066720   |
| T\$SDU = 010024 | T19 | 025606 G | T41    | 036116 G | WCKINT  | 007522   | XXDP.D   | 045064   |
| T\$SHAR= 010105 | T2  | 022102 G | T42    | 036402 G | WCKMES  | 007462   | XXX      | 017266   |
| T\$SHW = 010020 | T20 | 025740 G | T43    | 036672 G | WHY     | 002126   | X\$ALWA= | 000000   |
| T\$SINI= 010022 | T21 | 026060 G | T44    | 036762 G | WIDTH   | 055002   | X\$FALS= | 000040   |
| T\$MSG= 010017  | T22 | 026240 G | T45    | 037114 G | WRCHK = | 000002   | X\$OFFS= | 000400   |
| T\$SEGE= 010000 | T23 | 027016 G | T46    | 037312 G | WRITE = | 000012   | X\$TRUE= | 000020   |
| T\$SOF= 010106  | T24 | 027212 G | T47    | 037450 G | WRLOCK  | 010331   | \$BREG   | 045560   |
| T\$SRV= 010025  | T25 | 027356 G | T5     | 022532 G | WRTINT  | 010161   | \$ENDAD  | 067776 G |
| T\$SW = 010021  | T26 | 027472 G | T6     | 022664 G | WRTMES  | 010127   | \$SAV2   | 071042 G |
| T\$TES= 010104  | T27 | 027650 G | T7     | 023022 G | WTRDY   | 021276   | \$SAV3   | 071056 G |
| T.CNTL 002306   | T28 | 030250 G | T8     | 023220 G | WTRDY   | 021236   | \$SAV4   | 071074 G |
| T.CRC 002124    | T29 | 030672 G | T9     | 023522 G | XEQDIA  | 070024 G | \$SAV5   | 071114 G |
| T.DMP 016774    | T3  | 022246 G | UNITST | 002140   | XEQSUB  | 070012 G | .        | = 071320 |
| T.LMT 016776    | T30 | 031320 G | UNIT.D | 040532 G | XEQ.CL  | 047534   |          |          |
| T.SIZE 016772   | T31 | 032000 G | UNI.MA | 045410   | XEQ.CM  | 045044   |          |          |
| T1 021632 G     | T32 | 032432 G | UOPIMN | 002276   | XEQ.IN  | 047216   |          |          |
| T10 023712 G    | T33 | 033042 G | UOPIMX | 002274   | XEQ.LA  | 043306   |          |          |

. ABS. 071320 000

ERRORS DETECTED: 0

DSKZ: CZRLBB, DSKZ: CZRLBB= CZRLBB/ML, CZRLBB.P11, CZRLBB.SUP  
 RUN-TIME: 66 62 1 SECONDS  
 RUN-TIME RATIO: 323/130-2.4  
 CORE USED: 16K (31 PAGES)