

DUP11

OFF LINE SDLC RECEIVER
CZDPCC0

AH-8581C-MC
COPYRIGHT © 75-78
FICHE 1 OF 1

DEC 1978
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 |

IDENTIFICATION

PRODUCT CODE: AC-8580C-MC
PRODUCT NAME: CZDPCCO DUP11 OFLNE SDLC RCVR
DATE: APRIL 1978
MAINTAINER: DIAGNOSTICS

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 DOCUMENT.

THE SOFTWARE DESCRIBED IN THIS DOCUMENT IS FURNISHED UNDER A LICENSE AND MAY ONLY BE USED OR COPIED IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE.

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

COPYRIGHT (C) 1975,1978 BY DIGITAL EQUIPMENT CORPORATION

1.0 ABSTRACT

THE FUNCTION OF THE DUP11 DIAGNOSTICS IS TO VERIFY THAT THE OPTION OPERATES ACCORDING TO SPECIFICATIONS. THE DIAGNOSTICS VERIFY THAT THERE ARE NO MALFUNCTIONS AND THAT ALL OPERATIONS OF THE DUP11 ARE CORRECT IN ITS ENVIRONMENT. PARAMETERS MAY BE SET TO ALERT DIAGNOSTICS AS TO THE DUP11 CONFIGURATION BY ANSWERING THE PARAMETER DIALOG (LOAD ADDRESS=200, START ADDRESS=1). ALL QUESTIONS SHOULD BE ANSWERED AND THEN EACH DIAGNOSTIC WILL 'OVERLAY' THESE PARAMETERS WHICH ARE STORED IN THE 'STATUS TABLE' (SEE SECTION 8.4). THE ALTERNATIVE TO THE PARAMETER DIALOG IS DEFAULT PARAMETERS (SEE SECTION 8.5).

THE DIAGNOSTICS WILL RUN UP TO EIGHT CONSECUTIVELY ADDRESSED AND CONSECUTIVELY VECTORED DUP11'S IN A CHAIN MODE, I.E., RUNNING THE DIAGNOSTIC COMPLETELY FOR ONE DEVICE BEFORE STARTING THE NEXT.

CZDPC TESTS ALL RECEIVER SDLC FUNCTIONS IN MAINTENANCE INTERNAL MODE, THAT IS, CLOCKING OF THE DEVICE IS DONE BY THE PROGRAM. THE DEVICE IS SET UP, A SPECIFIC NUMBER OF HALF-CLOCKS ARE DONE, AND A TEST IS MADE FOR A SIGNIFICANT EVENT.

IN CHECKING DATA, THE SOFTWARE EMULATES THE HARDWARE AND USES THE PROCESSOR CARRY BIT AFTER A ROTATE TO PROVIDE AN INPUT TO THE RECEIVER VIA THE MAINTENANCE INPUT DATA BIT. THE PROGRAM CAN THEN LOAD THE RECEIVER DATA BUFFER SERIALY, WITHOUT USING THE TRANSMITTER.

THE RECEIVER BCC IS CHECKED USING THE CRC.CCITT POLYNOMIAL IN THE SAME WAY AS DATA, WITH ONE EXCEPTION--THE BCC IS CALCULATED FIRST BY THE PROGRAM AND THEN COMPARED TO THE RECEIVER OUTPUT.

CZDPC CHECKS ALL MODEM CONTROL AND INTERRUPT LOGIC DEPENDING ON THE PARAMETER INFORMATION SUPPLIED THROUGH THE OVERLAY MAP. IN ADDITION, ALL EIA GATES, WITH THE EXCEPTION OF THE DATA GATES, ARE CHECKED.

CURRENTLY THERE ARE THREE OFF-LINE DIAGNOSTICS THAT ARE TO BE RUN IN SEQUENCE TO ENSURE THAT IF AN ERROR SHOULD OCCUR IT WILL BE DETECTED AT AN EARLY STAGE AND ESTABLISH THAT DIAGNOSIS OF THE ERROR WILL BE IMMEDIATE TO DISCOVERING THE PROBLEM.

NOTE: ADDITIONAL DIAGNOSTICS MAY BE ADDED IN THE FUTURE.

THE THREE DIAGNOSTICS ARE:

1. CZDPB [REV] BASIC AND OFFLINE TRANSMITTER TESTS
2. CZDPC [REV] OFFLINE RECEIVER AND MODEM CONTROL AND INTERRUPT TESTS
3. CZDPD [REV] OFFLINE SDLC AND DECMODE DATA AND FUNCTION TESTS

NOTE: THERE IS A FOURTH PROGRAM, TAPE CZDPE [REV] WHICH IS A QUICK-VERIFY TAPE THAT REQUIRES ANSWERING A DIALOG. ITS FUNCTION IS TO ENABLE THE OPERATOR TO QUICKLY DETERMINE IF THERE IS A PROBLEM WITH THE DEVICE. SEE THE DOCUMENTATION IN THAT LISTING FOR MORE INFORMATION.

2.0 REQUIREMENTS

2.1 EQUIPMENT

ANY PDP11 FAMILY CPU (WITH MINIMUM 8K MEMORY)
ASR 33 (OR EQUIVALENT)
DUPI1

2.2 STORAGE

PROGRAM WILL USE ALL 8K OF MEMORY EXCEPT WHERE ABS AND
BOOTSTRAP LOADER RESIDE. LOCATION 1500 THRU 1560 ARE
ESPECIALLY TO BE NOTED AND LEFT UNTOUCHED BY THE OPERATOR
AFTER THE DUPI1 PARAMETER DIALOG HAS BEEN EXECUTED OR AFTER
THE DEFAULT SETUP HAS BEEN DONE.

3.0 LOADING PROCEDURE

3.1 METHOD

ALL PROGRAMS ARE IN ABSOLUTE FORMAT AND ARE LOADED USING THE
ABSOLUTE LOADER. NOTE: IF THE DIAGNOSTICS ARE ON A MEDIA
SUCH AS DISK, MAGTAPE, DFCTAPE, OR CASSETTE FOLLOW
INSTRUCTIONS FOR THE MONITOR WHICH HAS BEEN PROVIDED ON THAT
SPECIFIC MEDIA.

ABSOLUTE LOADER STARTING ADDRESS = **500

| MEMORY | SIZE |
|--------|------|
| | (*)= |
| 8K | 37 |
| 12K | 57 |
| 16K | 77 |
| 20K | 117 |
| 24K | 137 |
| 28K | 157 |

3.1.1 PLACE ADDRESS OF ABS LOADER INTO SWITCH REGISTER. (ALSO PLACE
'HALT' SW UP)

3.1.2 DEPRESS 'LOAD ADDRESS' KEY ON CONSOLE AND RELEASE.

3.1.3 DEPRESS 'START KEY' ON CONSOLE AND RELEASE (PROGRAM SHOULD NOW
BE LOADING INTO CPU)

4.0 STARTING PROCEDURE

- A. SET SWITCH REGISTER TO 000200
- B. DEPRESS 'LOAD ADDRESS' KEY AND RELEASE
- C. SET SWR TO ZERO FOR DEFAULT PARAMETERS ESTABLISHED IN THE TAPE (SEE SECTION 8.5.3 FOR FULL EXPLANATION OF DEFAULT PARAMETERS) OR LEAVE SWR BIT 7=1 TO USE EXISTING PARAMETERS PREVIOUSLY SET UP BY THE DUP11 PARAMETER DIALOG OR A PREVIOUSLY RUN DUP11 DIAGNOSTIC. SET SWR=1 TO GO THROUGH THE PARAMETER DIALOG. (IT IS NOT NECESSARY TO INPUT NEW PARAMETERS FOR EACH TAPE.) (SECTION 7.2, 8.4 AND 8.5 MAY BE HELPFUL)
- D. DEPRESS 'START KEY' AND RELEASE. THE PROGRAM WILL TYPE MAINDEC NAME AND PROGRAM NAME (IF THIS WAS THE FIRST START UP OF THE PROGRAM) AND ALSO THE FOLLOWING:

'EXAMPLE'

'MAP OF DUP11 STATUS'

| | | |
|------|--------|----------------------------------|
| 1500 | 160050 | CSR OF FIRST DUP11 |
| 1502 | 000300 | VECTOR OF FIRST DUP11 |
| 1504 | 140026 | STATUS AND SYNC FOR FIRST DUP11 |
| 1506 | 160060 | CSR OF SECOND DUP11 |
| 1510 | 000310 | VECTOR OF SECOND DUP11 |
| 1512 | 140026 | STATUS AND SYNC FOR SECOND DUP11 |

THE ABOVE IS ONLY AN EXAMPLE! THIS WOULD INDICATE THE STATUS TABLE STARTING AT ADDRESS 1500 IN THE PROGRAM. THE STATUS TABLE MUST BE VERIFIED BY THE USER. FOR INFORMATION ON THE STATUS TABLE SEE SECTION 8.4 FOR HELP.

IT IS POSSIBLE FOR THE OPERATOR TO MANUALLY CHANGE (TOGGLE IN) THE INFORMATION IN THE MAP TO SUIT A SPECIFIC CONFIGURATION OF DEVICES, BUT THE RESPONSIBILITY FOR VERIFYING THAT INFORMATION RESTS WITH THE OPERATOR.

THE PROGRAM WILL TYPE 'R' AND PROCEED TO RUN THE DIAGNOSTIC

4.1 CONTROL SWITCH SETTINGS

| | | |
|-------|------|----------------------------------|
| SW 15 | SET: | HALT ON ERROR |
| SW 14 | SET: | LOOP ON CURRENT TEST |
| SW 13 | SET: | INHIBIT ERROR PRINT OUT |
| SW 12 | SET: | INHIBIT TYPE OUT/BELL ON ERROR. |
| SW 11 | SET: | INHIBIT ITERATIONS. (QUICK PASS) |
| SW 10 | SET: | ESCAPE TO NEXT TEST ON ERROR |
| SW 09 | SET: | LOOP WITH CURRENT DATA |
| SW 08 | SET: | CATCH ERROR AND LOOP ON IT |
| SW 07 | SET: | USE PREVIOUS STATUS TABLE. |
| SW 06 | SET: | RESERVED |
| SW 05 | SET: | RESERVED |

SW 04 SET: RESERVED
SW 03 SET: SELECT DUP11'S DESIRED ACTIVE
SW 02 SET: LOCK ON SELECTED TEST
SW 01 SET: RESTART PROGRAM AT SELECTED TEST
SW 00 SET: ENTER PARAMETERS USING MANUAL DIALOG

SWITCHES 8 THROUGH 15 ARE DYNAMIC AND SHOULD BE USED AS NEEDED IN THE DIAGNOSTIC. SWITCHES 0 THROUGH 3 ARE STATIC (ONLY ARE OPERABLE WHEN THE MONITOR PORTION OF THE TAPE IS RUNNING) AND SHOULD BE SET UP PRIOR TO STARTING OR RESTARTING THE DIAGNOSTIC.

4.1.2 SWITCH REGISTER RESTRICTIONS

SW 03 RESELECT DUP11'S DESIRED ACTIVE. PLEASE NOTE THAT A MESSAGE IS TYPED OUT FOR SETTING THE SWITCH REGISTER EQUAL TO DUP11'S ACTIVE. THIS MEANS IF THE SYSTEM HAS THREE DUP11S BITS 00, 01, 02 WILL BE SET IN LOC 'DUPACTV' FROM THE SWITCH REGISTER. USING THIS SWITCH(SW03) ALTERS THAT LOCATION. THEREFORE, IF THREE DUP11S ARE IN THE SYSTEM ***DO NOT*** SET SWITCHES GREATER THAN SW 02 IN THE UP POSITION. THIS WOULD BE A FATAL ERROR. DO NOT SELECT MORE ACTIVE DUP11S THAN HAS BEEN GIVEN INFORMATION ABOUT IN THE PARAMETER PROGRAM.

AS EXPLAINED IN SECTION 1.0, DEVICES SHOULD BE CONSECUTIVELY ADDRESSED, AND CAN BE SELECTED OR DESELECTED USING THIS SWITCH.

METHOD: A. LOAD ADDRESS 200
B. START WITH SW 03=1
C. PROGRAM WILL TYPE MESSAGE
D. SET THE BINARY NUMBER OF DUP11S DESIRED ACTIVE. EXAMPLE: 1=1 DUP11; 3=2 DUP11; 7=3 DUP11; 17=4 DUP11 37=5 DUP11 ETC. PRESS CONTINUE.
E. NUMBER (IF VALID) WILL BE IN DATA LIGHTS (EXCLUDING 11/05)
F. SET WITH ANY OTHER SWITCH SETTINGS DESIRED. PRESS CONTINUE.

SW 01 RESTART PROGRAM AT SELECTED TEST. IT IS STRONGLY SUGGESTED THAT AT LEAST ONE PASS HAS BEEN MADE BEFORE TRYING TO SELECT A TEST THAT IS NOT IN THE ORDER OF SEQUENCE. THE REASON FOR THIS IS THAT THE PROGRAM HAS TO CLEAR AREAS AND SET UP PARAMETERS IN THE MONITOR PORTION OF THE PROGRAM. IT IS POSSIBLE TO LD200, AND RAISE SW01, THEN START, PROVIDED PARAMETERS HAVE BEEN PREVIOUSLY SET UP AS DESCRIBED IN SECTION 4.0. ALSO, WHEN A TEST IS SELECTED, ALWAYS START AT THE VERY BEGINNING OF THAT TEST.

SW 09 LOOP ON CURRENT DATA. THIS SWITCH WILL ONLY WORK IF CALL 'SCOPI' IS IN THAT TEST. THE REASON IS THAT MOST

TESTS DEAL WITH BLOCKS OF DIFFERENT DATA TO BE SENT OR RECEIVED ALL AT ONCE, THUS KNOWN AS BLOCK DATA--ONE PATTERN CAN'T BE SINGLED OUT. (SEE SECTION 4.1.3.B.1)

4.1.3 SWITCH REGISTER PRIORITIES

A) ERROR SWITCHES

1. SW 12 DELETE PRINT OUT/BELL ON ERROR.
2. SW 13 DELETE ERROR PRINTOUT.
3. SW 15 HALT ON THE ERROR.
4. SW 08 GOTO BEGINNING OF THE TEST(ON ERROR).
5. SW 10 GOTO NEXT TEST(ON ERROR).

B) SCOPE SWITCHES

1. SW 09 - (IF ENABLED BY 'SCOPI') ON AN ERROR. IF AN ASTERISK '*' IS PRINTED IN FRONT OF THE TEST NUMBER (EX. *TEST NO. 10), SW09 IS INCORPORATED IN THAT TEST AND THEREFORE SW09 IS USUALLY THE BEST SWITCH FOR THE SCOPE LOOP (SW14=0, SW10=0, SW09=1, SW08=0).

IF SW09 IS NOT ENABELED AND THERE IS A *HARD* ERROR (CONSTANT ERROR) SW08 IS BEST. (SW14=0, SW10=0, SW09=0, SW08=1).

FOR INTERMITTENT ERRORS, SW14=1 WILL LOOP ON TEST REGARDLESS OF ERROR OR NO ERROR. (SW14=1, SW10=0, SW09=0, SW08=1,0)

2. SW 14 - LOOP ON TEST. WILL LOOP ON TEST UNTIL SWITCH IS LOWERED.
3. SW 11 - INHIBIT ITERATIONS (QUICK PASS). ALLOWS ONLY ONE PASS THROUGH A TEST.

4.2 STARTING ADDRESS

STARTING ADDRESS IS AT 000200. THERE ARE NO OTHER STARTING ADDRESSES FOR THE DUP11 DIAGNOSTICS.

NOTE: IF ADDRESS 000042 IS NON-ZERO THE PROGRAM ASSUMES IT IS UNDER ACT11 OR XXDP CONTROL AND WILL ACT ACCORDINGLY. AFTER *ALL* AVAILABLE DUP11'S ARE TESTED THE PROGRAM WILL RETURN TO 'XXDP' OR 'ACT-11'.

5.0 OPERATING PROCEDURE

WHEN THE PROGRAM IS INITIALLY STARTED MESSAGES AS DESCRIBED IN SECTION FOUR WILL BE PRINTED AND PROGRAM WILL BEGIN RUNNING THE DIAGNOSTIC.

5.1 PROGRAM AND/OR OPERATOR ACTION

THE TYPICAL APPROACH SHOULD BE

1. HALT ON ERROR (VIA SW 15=1) WHENEVER AN ERROR OCCURS.
2. CLEAR SW 15.
3. SET SW 14: (LOOP ON THIS TEST)
4. SET SW 13: (INHIBIT ERROR PRINT OUT)

THE TEST NUMBER AND PC WILL BE TYPED OUT AND POSSIBLY AN ERROR MESSAGE (THIS DEPENDS ON THE TEST), TO GIVE THE OPERATOR AN IDEA AS TO THE SOURCE OF THE PROBLEM. IF IT IS NECESSARY TO KNOW MORE INFORMATION CONCERNING THE ERROR REPORT, LOOK IN THE LISTING FOR THAT TEST NUMBER WHICH WAS TYPED OUT AND THEN NOTE THE PC OF THE ERROR REPORT. IN THIS WAY THE EXACT FUNCTIONING OF THE TEST CAN BE INTERPRETED SINCE THE ERROR PC IS THE HLT+2 LOCATION.

IN SOME TESTS, THERE IS A SUBROUTINE CALL THROUGH A REGISTER (E.G., JSR R1,FLAG). THE SUBROUTINE DOES THE DATA CHECKING FOR THE TEST AND WILL REPORT AN ERROR IF ONE OCCURS. THIS MEANS THAT THE FAILING TEST COULD BE IN ONE PART OF THE LISTING WHILE THE SUBROUTINE THAT FOUND THE ERROR IS IN ANOTHER PART. TO DETERMINE THE PC OF THE FAILING TEST, CHECK THE REGISTER USED BY THE SUBROUTINE. IT WILL CONTAIN THE RETURN ADDRESS OF THE FAILING TEST.

6.0 ERRORS

AS DESCRIBED PREVIOUSLY THERE WILL ALWAYS BE A TEST NUMBER AND PC TYPED OUT AT THE TIME OF AN ERROR (PROVIDING SW 13=0 AND SW 12=0). IN MOST CASES ADDITIONAL INFORMATION WILL BE SUPPLIED TO THE ERROR MESSAGE WHICH IS TO GIVE THE OPERATOR AN INDICATION OF THE ERROR.

6.1 ERROR RECOVERY

IF FOR SOME REASON THE DUP11 SHOULD 'HANG THE BUS' (GAIN CONTROL OF BUS SO THAT CONSOLE MANUAL FUNCTIONS ARE INHIBITED) AN INIT OR POWER DOWN/UP IS NECESSARY FOR OPERATOR TO REGAIN CONTROL OF CPU. IF THIS SHOULD HAPPEN LOOK IN LOCATION 'TSTNO' FOR THE NUMBER OF THE TEST THAT WAS RUNNING AT THE TIME OF THE CATASTROPHIC ERROR. THIS GIVES THE OPERATOR SOME IDEA AS TO WHAT THE DUP11 WAS DOING AT THE TIME OF THE ERROR.

7.0 RESTRICTIONS

7.1 STARTING RESTRICTIONS

SEE SECTION 4 (PLEASE). STATUS TABLE SHOULD BE VERIFIED REGARDLESS OF HOW THE PROGRAM WAS STARTED. ALSO, IT IS IMPORTANT TO USE THE LISTING ALONG WITH THE INFORMATION PRINTED ON THE TTY TO COMPLETELY ISOLATE PROBLEMS.

7.2 OPERATING RESTRICTIONS

DUP11 'PARAMETER DIALOG' MUST BE RUN ONLY ONCE PRIOR TO THE FIRST RUNNING OF ANY DUP11 DIAGNOSTIC IF 'DEFAULT PARAMETERS' ARE NOT USED. IF ONLY DUP11 DIAGNOSTICS WERE LOADED AFTER DUP11 PARAMETER SETUP, AND IF CORE MEMORY HAS NOT BEEN CHANGED, I.E., USE OF DIAGNOSTICS OTHER THAN DUP11 DIAGNOSTICS, AND IF THERE WERE NO DUP11 CONFIGURATION CHANGES, THE DUP11 PARAMETER SETUP NEED NEVER BE RUN AGAIN. HOWEVER, IF ANY OF THE ABOVE HAVE BEEN VIOLATED THE DUP11 PARAMETER SETUP MUST BE RUN AGAIN BEFORE RUNNING THE DIAGNOSTICS. UNDER NORMAL OPERATING CONDITIONS IT SHOULD NOT BE NECESSARY TO INPUT NEW PARAMETERS TO SUBSEQUENT DIAGNOSTICS, UNLESS A CHANGE IS REQUIRED.

NOTE: AN ALTERNATIVE TO THE ABOVE IS ATTEMPTING THE DEFAULT PARAMETERS WHEN THE PROGRAM IS INITIALLY STARTED WITH SWR-0.

7.3 HARDWARE CONFIGURATION RESTRICTIONS FOR THE PURPOSE OF RUNNING MULTIPLE DUP11'S IN CHAIN MODE.

1. CSR ADDRESSES MUST BE CONSECUTIVE.
2. VECTORS ARE CONSECUTIVE IF PARAMETER PROGRAM IS USED.
3. ALL JUMPERS ARE ASSUMED TO BE AS SETUP IN PARAMETER DIALOG.
4. PRIORITY LEVEL MUST BE THE SAME FOR ALL DEVICES.

8.0 MISCELLANEOUS

8.1 EXECUTION TIME

ALL DUP11 DEVICE DIAGNOSTICS WILL GIVE AN 'END PASS' MESSAGE (PROVIDING NO ERRORS AND SW12=0) WITHIN 4 MINS. THIS IS ASSUMING SW11=1 (DELETE ITERATIONS) IS SET TO GIVE THE FASTEST POSSIBLE EXECUTION. THE ACTUAL EXECUTION TIME DEPENDS GREATLY ON THE PDP11 CPU CONFIGURATION.

8.2 PASS COMPLETE

NOTE: *EVERY* TIME THE PROGRAM IS STARTED, THE TESTS WILL RUN AS IF SW11 (DELETE ITERATIONS) WAS UP (=1). THIS IS TO VERIFY NO *HARD* ERRORS AS SOON AS POSSIBLE. THEREFORE THE FIRST PASS--EACH TIME PROGRAM IS STARTED--WILL BE A 'QUICK PASS' UNTIL ALL DUP11'S IN SYSTEM ARE TESTED. WHEN THE DIAGNOSTIC HAS COMPLETED A PASS WITH THE NORMAL ITERATION COUNT (ICOUNT=50), THE FOLLOWING IS AN EXAMPLE OF THE PRINT OUT TO BE EXPECTED.

END PASS CZDPCC CSR:160050 VEC:300 PASSES:000001 ERRORS:000000

NOTE: THE NUMBERS FOR CSR AND VEC ARE NOT NECESSARILY THE VALUES FOR THE DEVICE. THEY ARE ONLY FOR THIS EXAMPLE.

8.3 KEY LOCATIONS

RETURN CONTAINS THE ADDRESS WHERE PROGRAM WILL RETURN WHEN ITERATION COUNT IS REACHED OR IF LOOP ON TEST IS ASSERTED.

NEXT CONTAINS THE ADDRESS OF THE NEXT TEST TO BE PERFORMED.

TSTNO CONTAINS THE NUMBER OF THE TEST NOW BEING PERFORMED.

RUN THE BIT IN 'RUN' ALWAYS POINTS ONE PAST THE DUP11 CURRENTLY BEING TESTED. EXAMPLE: (RUN) /0000000001000000 MEANS THAT DUP11 NO.05 IS THE DUP11 NOW RUNNING.

DUPCR00-DUPCR07 (1500)-(1560) THESE LOCATIONS CONTAIN THE INFORMATION NEEDED TO TEST UP TO 8 (DECIMAL) DUP11S SEQUENTIALY. THEY CONTAIN THE CSR, VECTOR AND STATUS CONCERNING THE CONFIGURATION OF EACH DUP11.

DUPACTV EACH BIT SET IN THIS LOCATION INDICATES THAT THE ASSOCIATED DUP11 WILL BE TESTED IN TURN. EXAMPLE: (DUPACTV) /0000000000011111 MEANS THAT DUP11 NO. 00,01,02,03,04 WILL BE TESTED. EXAMPLE: (DUPACTV) /0000000000010001 MEANS

THAT DUP11 NO. 00,04 WILL BE TESTED.

RXCSR CONTAINS THE RECEIVER CSR OF THE CURRENT DUP11 UNDER TEST.

8.4 MORE ON THAT 'STATUS TABLE' (1500-1560)

'MAP OF DUP11 STATUS'

| | |
|------|--------|
| 1500 | 160050 |
| 1502 | 000300 |
| 1504 | 140000 |

THE ABOVE INFORMATION WILL BE REPEATED FOR EACH OF UP TO 8 DUP11'S IN THE SYSTEM (THESE WILL FOLLOW UNDER THIS TABLE).
EXPLANATION:

| | | |
|------|--------|--|
| 1500 | 160050 | THIS IS THE SYSTEM CONTROL REGISTER FOR THE 1ST DUP11 IN THE SYSTEM. |
| 1502 | 000300 | THIS IS VECTOR 'A' FOR THE FIRST DUP11 IN THE SYSTEM. |
| 1504 | 140026 | THIS REPRESENTS SYNC AND SOFTWARE STATUS FOR THE FIRST DUP. |

THE BITS ARE AS FOLLOWS:

| | | |
|-----------|------|-----------------------------------|
| BIT 15 | SET: | OPTIONAL CLEAR JUMPER IN |
| BIT 14 | SET: | TURNAROUND CONNECTOR ON |
| BIT 13 | SET: | |
| BIT 12 | SET: | |
| BIT 11 | SET: | |
| BIT 10 | SET: | |
| BIT 09 | SET: | |
| BIT 08 | SET: | |
| BIT 07-00 | | SYNC CHARACTER FOR DECMODE TESTS. |

THE ABOVE IS REPEATED FOR EACH DUP11 IN THE SYSTEM. THE TABLE IS FILLED BY DEFAULT PARAMETERS OR BY THE MANUAL PARAMETER INPUT AS DESCRIBED PREVIOUSLY. ALSO, IF DESIRED BY THE USER - THE LOCATIONS MAY BE ALTERED BY HAND (TOGGLED IN) TO SUIT THE SPECIFIC CONFIGURATION, THUS MAKING EACH DEVICE MAP DIFFERENT. IT IS THE RESPONSIBILITY OF THE OPERATOR TO VERIFY THE DATA IN THE MAP.

8.5 METHOD OF DEVELOPING DEFAULT PARAMETERS

8.5.1 DEFAULT PARAMETER ASSUMPTIONS

TOO MUCH HARDWARE WOULD HAVE TO BE ANALYZED TO SIZE THE THE PARAMETERS. THE PROGRAM MUST ASSUME THE VARIATIONS. THE RESULT, IF NOT TO YOUR SPECIFIC CONFIGURATION, MAY BE ALTERED BY HAND (TOGGLE IN) AS DESIRED. IN THIS WAY 95% OF THE

PARAMETER SETUP WAS DONE BY THE PROGRAM AND 5% BY YOU.
THEREFORE:

- 1) ALL JUMPERS ARE ASSUMED TO BE IN THE FOLLOWING CONFIGURATION.

| | IN | OUT |
|-------------------------|----|-----|
| W1=SECONDARY REC ENABLE | X | |
| W2=SEC REC DISABLE | | X |
| W3-CLEAR OPTION | X | |
| W4=SEC TX ENABLE | X | |
| W5=DSC A CONTROL | | X |
| W6 A+B DS CONTROL | X | |
| W7-BUS GRANT CONTROL | X | |

- 2) THE H325 TURN AROUND CONNECTOR IS ASSUMED TO BE ON.
- 3) THE MANUFACTURING OPTION CSR OF 160050 AND VECTOR OF 770 ARE USED.
- 4) THE BR LEVEL IS ASSUMED TO BE 5.

IN ALL ADJUSTMENTS PLEASE REFER TO SECTION 8.4 FOR GREATER
DETAIL.

:*CZDPCCO /<377>/DUP11 OFLNE SDLC RCVR
:*COPYRIGHT(C) 1975,1978, DIGITAL EQUIPMENT CORP., MAYNARD, MASS. 01754
:-----

:STARTING PROCEDURE
:LOAD PROGRAM
:LOAD ADDRESS 000200
:PRESS START
:PROGRAM WILL TYPE 'CZDPCCO /<377>/DUP11 OFLNE SDLC RCVR ''
:PROGRAM WILL TYPE 'R' TO INDICATE THAT TESTING HAS STARTED
:AT THE END OF A PASS, PROGRAM WILL TYPE PASS COMPLETE MESSAGE
:AND THEN RESUME TESTING

:SWITCH REGISTER OPTIONS
:-----

| | | | |
|-----|--------|-------------|---|
| 550 | 100000 | SW15=100000 | : -1, HALT ON ERROR |
| 551 | 040000 | SW14=40000 | : -1, LOOP ON CURRENT TEST |
| 552 | 020000 | SW13=20000 | : -1, INHIBIT ERROR TYPEOUT |
| 553 | 010000 | SW12=10000 | : -1, DELETE TYPEOUT/BELL ON ERROR. |
| 554 | 004000 | SW11=4000 | : -1, INHIBIT ITERATIONS |
| 555 | 002000 | SW10=2000 | : -1, ESCAPE TO NEXT TEST ON ERROR |
| 556 | 001000 | SW09=1000 | : -1, LOOP WITH CURRENT DATA |
| 557 | 000400 | SW08=400 | : -1, LOOP ON ERROR |
| 558 | 000200 | SW07=200 | |
| 559 | 000100 | SW06=100 | |
| 560 | 000040 | SW05=40 | |
| 561 | 000020 | SW04=20 | |
| 562 | 000010 | SW03=10 | : SELECT DUP'S DESIRED ACTIVE |
| 563 | | | : NOTE: THIS MUST NOT EXCEED ORIGINAL COUNT |
| 564 | 000004 | SW02=4 | : LOCK ON TEST SELECT |
| 565 | 000002 | SW01=2 | : RESTART PROGRAM AT SELECTED TEST |
| 566 | 000001 | SW00=1 | : ENTER PARAMETERS |

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

:REGISTER DEFINITIONS

R0=%0 ;GENERAL REGISTER
R1=%1 ;GENERAL REGISTER
R2=%2 ;GENERAL REGISTER
R3=%3 ;GENERAL REGISTER
R4=%4 ;GENERAL REGISTER
R5=%5 ;GENERAL REGISTER
SP=%6 ;PROCESSOR STACK POINTER
PC=%7 ;PROGRAM COUNTER

:LOCATION EQUIVALENCIES

PS=177776 ;PROCESSOR STATUS WORD
STACK=1150 ;START OF PROCESSOR STACK

:INSTRUCTION DEFINITIONS

PUSH1SP=5746 ;DECREMENT PROCESSOR STACK 1 WORD
POP1SP=5726 ;INCREMENT PROCESSOR STACK 1 WORD
PUSHR0=10046 ;SAVE R0 ON STACK
POPR0=12600 ;RESTORE R0 FROM STACK
PUSH2SP=24646 ;DECREMENT STACK TWICE
POP2SP=22626 ;INCREMENT STACK TWICE
.EQUIV EMT,HLT ;BASIC DEFINITION OF ERROR CALL

BIT15=100000
BIT14=40000
BIT13=20000
BIT12=10000
BIT11=4000
BIT10=2000
BIT9=1000
BIT8=400
BIT7=200
BIT6=100
BIT5=40
BIT4=20
BIT3=10
BIT2=4
BIT1=2
BIT0=1

TRAPCATCHER FOR UNEXPECTED INTERRUPTS

```
617 ;:*****
618 ;-----
619 ; TRAPCATCHER FOR ILLEGAL INTERRUPTS
620 ; THE STANDARD 'TRAP CATCHER' IS PLACED
621 ; BETWEEN ADDRESS 0 TO ADDRESS 776.
622 ; IT LOOKS LIKE 'PC+2 HALT'.
623 ;-----
624 ;:*****
625
626      000000      .=0
627      ;STANDARD INTERRUPT VECTORS
628      ;-----
629
630      000024      .=24
631 000024 005050      .PFAIL      ;POWER FAIL HANDLER
632 000026 000340      340          ;SERVICE AT LEVEL 7
633 000030 004350      .HLT          ;ERROR HANDLER
634 000032 000340      340          ;SERVICE AT LEVEL 7
635 000034 004316      .TRPSRV      ;GENERAL HANDLER DISPATCH SERVICE
636 000036 000340      340          ;SERVICE AT LEVEL 7
637
638      000040      .=40
639 000042 000000      0          ;SAVE FOR ACT-11 OR DDP2
640 000044 000000      0          ;RETURN ADDRESS IF UNDER ACT-11 OR DDP2
641 000046 003104      0          ;SAVE FOR ACT-11 OR DDP2
642      000052      $ENDAD        ;FOR USE WITH ACT-11 OR DDP2
643 000052 000000      0          ;ACT-11 PROGRAM CHARACTERISTICS
644
645      000174      .=174
646 000174 000000      DISPREG:0    ;SOFTWARE DISPLAY REGISTER
647 000176 000000      SWREG: 0     ;SOFTWARE SWITCH REGISTER
648      000200      .=200
649 000200 000137 001562      JMP      .START      ;GO TO START OF PROGRAM
650
651
652      001000      .=1000
653 001000 005377 055103 050104      MTITLE: .ASCIZ <377><12>/CZDPCCO /<377>/DUP11 OFLNE SDLC RCVR /<377>
654      001200      .=1200
655      ;SWR AND LIGHTS
656      ;-----
657
658 001200 177570      DISPLAY:      177570      ;11/45 CONSOLE LIGHTS
659 001202 177570      SWR:          177570      ;INDIRECT POINTER TO SWITCH REGISTER
660
661      ;INDIRECT POINTERS TO TELETYPE VECTORS AND REGISTERS
662      ;-----
663
664 001204 177560      TKCSR:      177560      ;TELETYPE KEYBOARD CONTROL REGISTER
665 001206 177562      TKDBR:      177562      ;TELETYPE KEYBOARD DATA BUFFER
666 001210 177564      TPCSR:      177564      ;TELEPRINTER CONTROL REGISTER
667 001212 177566      TPDBR:      177566      ;TELEPRINTER DATA BUFFER
668
669      ;PROGRAM CONTROL PARAMETERS
670      ;-----
671
```


PROGRAM PARAMETERS, VARIABLES, AND TRAP CALLS.

| | | | | |
|-----|--------|--------|--------------------|--|
| 672 | 001214 | 000000 | RETURN: 0 | :SCOPE ADDRESS FOR LOOP ON TEST |
| 673 | 001216 | 000000 | NEXT: 0 | :ADDRESS OF NEXT TEST TO BE EXECUTED |
| 674 | 001220 | 000000 | LOCK: 0 | :ADDRESS FOR LOCK ON CURRENT DATA |
| 675 | 001222 | 000001 | ICOUNT: 1 | :NUMBER OF ITERATIONS THAT CURRENT TEST WILL BE EXECUTED |
| 676 | 001224 | 000000 | LPCNT: 0 | :NUMBER OF ITERATIONS COMPLETED |
| 677 | 001226 | 000000 | TSTNO: 0 | :NUMBER OF TEST IN PROGRESS |
| 678 | 001230 | 000000 | PASCNT: 0 | :NUMBER OF PASSES COMPLETED |
| 679 | 001232 | 000000 | ERRCNT: 0 | :TOTAL NUMBER OF ERRORS |
| 680 | 001234 | 000000 | LSTERR: 0 | :PC OF LAST ERROR CALL |
| 681 | | | | |
| 682 | | | :PROGRAM VARIABLES | |
| 683 | | | :----- | |
| 684 | | | | |
| 685 | 001236 | 000000 | TEMP1: 0 | :TEMPORARY STORAGE |
| 686 | 001240 | 000000 | TEMP2: 0 | :TEMPORARY STORAGE |
| 687 | 001242 | 000000 | TEMP3: 0 | :TEMPORARY STORAGE |
| 688 | 001244 | 000000 | TEMP4: 0 | :TEMPORARY STORAGE |
| 689 | 001246 | 000000 | TEMP5: 0 | :TEMPORARY STORAGE |
| 690 | 001250 | 000000 | SAVR0: 0 | :R0 STORAGE |
| 691 | 001252 | 000000 | SAVR1: 0 | :R1 STORAGE |
| 692 | 001254 | 000000 | SAVR2: 0 | :R2 STORAGE |
| 693 | 001256 | 000000 | SAVR3: 0 | :R3 STORAGE |
| 694 | 001260 | 000000 | SAVR4: 0 | :R4 STORAGE |
| 695 | 001262 | 000000 | SAVR5: 0 | :R5 STORAGE |
| 696 | 001264 | 000000 | SAVSP: 0 | :STACK POINTER STORAGE |
| 697 | 001266 | 000000 | SAVPC: 0 | :PROGRAM COUNTER STORAGE |
| 698 | | | | |
| 699 | 001270 | 000000 | SAVR0A: 0 | :R0 STORAGE |
| 700 | 001272 | 000000 | SAVR1A: 0 | :R1 STORAGE |
| 701 | 001274 | 000000 | SAVR2A: 0 | :R2 STORAGE |
| 702 | 001276 | 000000 | SAVR3A: 0 | :R3 STORAGE |
| 703 | 001300 | 000000 | SAVR4A: 0 | :R4 STORAGE |
| 704 | 001302 | 000000 | SAVR5A: 0 | :R5 STORAGE |
| 705 | 001304 | 000000 | SAVSPA: 0 | :STACK POINTER STORAGE |
| 706 | 001306 | 000000 | SAVPCA: 0 | :PROGRAM COUNTER STORAGE |
| 707 | | | | |
| 708 | 001310 | 000001 | DUPACTV: .BLKB 1 | :DUP11'S SELECTED ACTIVE. |
| 709 | 001311 | 000001 | DUPNUM: .BLKB 1 | :OCTAL NUMBER OF DUP11'S. |
| 710 | 001312 | 000001 | SAVACT: .BLKB 1 | :ORIGINAL ACTV. DEVICES. |
| 711 | 001313 | 000001 | SAVNUM: .BLKB 1 | :WORKABLE NUMBER. |
| 712 | 001314 | 000001 | RUN: .BLKB 1 | :POINTER ONE PAST RUNNING DEVICE. |
| 713 | | 001316 | .EVEN | |
| 714 | 001316 | 001500 | CREAM: DUP.MAP | :TABLE POINTER. |

PROGRAM PARAMETERS, VARIABLES, AND TRAP CALLS.

```
715
716                                     ;CONTROL REGISTER DEFINITIONS
717                                     ;-----
718                                     ;RXCSR BIT DEFINITIONS
719      100000      DSCA=BIT15      ;DATA SET CHANGE A
720      040000      RING=BIT14      ;RING
721      020000      CTS=BIT13      ;CLR TO SEND
722      010000      CARDET=BIT12    ;CARRIER DETECT
723      004000      RECACT=BIT11    ;REC ACTIVE
724      002000      SRD=BIT10      ;SEC REC DATA
725      001000      DSR=BIT9       ;DATA SET RDY
726      000400      STPSYN=BIT8    ;STRIP SYNC
727      000200      RXDONE=BIT7    ;REC DONE
728      000100      RINTEN=BIT6    ;REC INTR ENABLE
729      000040      DSINTE=BIT5    ;DSC INTR ENABLE
730      000020      RCVEN=BIT4    ;REC ENABLE
731      000010      STD=BIT3      ;SEC XMIT DATA
732      000004      RTS=BIT2      ;REQ TO SEND
733      000002      DTR=BIT1      ;DATA TERM RDY
734      000001      DSCB=BIT0     ;DATA SET CHANGE B
735                                     ;RXDBUF BIT DEFINITIONS
736      100000      RXDERR=BIT15    ;REC DATA ERROR
737      040000      OVRRUN=BIT14   ;OVERRUN ERROR
738      010000      CRCERR=BIT12   ;CRC ERROR
739      002000      RABORT=BIT10   ;REC ABOPT
740      001000      REOM=BIT9     ;REC END OF MESSAGE
741      000400      RSOM=BIT8     ;REC START OF MESSAGE
742                                     ;PARCSR BIT DEFINITIONS
743      100000      DECMOD=BIT15    ;DEC MODE (DDCMP)
744      001000      CRCEN=BIT9     ;CRC ENABLE
745      010000      PRISEC=BIT12   ;PRI/SEC SELECT
746                                     ;TXCSR BIT DEFINITIONS
747      100000      TXDLAT=BIT15   ;TX DATA LATE
748      040000      MTDATA=BIT14   ;MAINT DATA OUT
749      020000      CLK=BIT13     ;CLK
750      010000      MMODEB=BIT12   ;MAINT MODE B
751      004000      MMODEA=BIT11   ;MAINT MODE A
752      002000      BITW=BIT10    ;BIT WINDOW INPUT
753      001000      TXACT=BIT9    ;TX ACTIVE
754      000400      MRESET=BIT8   ;MASTER RESET
755      000200      TXDONE=BIT7   ;XMIT DONE
756      000100      TXINTE=BIT6   ;XMIT DONE INTR ENABLE
757      000020      SEND=BIT4     ;SEND
758      000010      HDXEN=BIT3    ;HDX/FDX
759                                     ;TXCSR WRD DEFINITIONS
760      000000      USER=0        ;USER MODE
761      014000      MMODE=14000   ;MAINT INT MODE
762      010000      MEXT=10000    ;MAINT EXT MODE
763      004000      SYSTST=4000   ;SYSTEM TEST MODE
764
765                                     ;TXDBUF BIT DEFINITIONS
766                                     ;-----
767      100000      RCRC7T=BIT15   ;
768      040000      RCRCIN=BIT14   ;
769      020000      TCRC7T=BIT13   ;
770      010000      TCRCIN=BIT12   ;
```

CZDPCC MACY11 30A(1052) 02-MAY-78 13:54 PAGE 19
CZDPCC.P11 02-MAY-78 13:47

PROGRAM PARAMETERS, VARIABLES, AND TRAP CALLS.

| | | | |
|-----|--------|--------------|----------------------------|
| 771 | 004000 | TIMER=BIT11 | :MAINTENANCE TIMER |
| 772 | 002000 | TABORT=BIT10 | :TRANSMIT ABORT |
| 773 | 001000 | TEOM=BIT9 | :TRANSMIT END OF MESSAGE |
| 774 | 000400 | TSOM=BIT8 | :TRANSMIT START OF MESSAGE |

775
776 :MISC. PROGRAM DEFINITIONS
777 :-----

| | | | |
|-----|--------|--------|-----------------|
| 778 | 001320 | 000000 | PRIRTY: .WORD 0 |
| 779 | 001322 | 000001 | TCNFLG: .BLKB 1 |
| 780 | 001323 | 000001 | OPCLRJ: .BLKB 1 |
| 781 | 001324 | 000000 | DATA: .WORD 0 |
| 782 | 001326 | 000000 | SHIFTS: .WORD 0 |
| 783 | 001330 | 000000 | MIND: .WORD 0 |
| 784 | 001332 | 000000 | FLAG: .WORD 0 |
| 785 | 001334 | 000001 | STJMFL: .BLKW 1 |
| 786 | 001336 | 000001 | SRJMFL: .BLKW 1 |

787
788

PROGRAM PARAMETERS, VARIABLES, AND TRAP CALLS.

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

001340 000
001341 000
001342 000
001343 000

000000

001344
001344 104400
001344 003160
001346 104401
001346 003312
001350 104402
001350 003336
001352 104403
001352 003412
001354 104404
001354 003516
001354 104405
001356 003536
001356 104406
001360 003736
001360 104407
001362 003776
001364 104410
001364 004030
001364 104411
001366 004034
001366 104412
001370 005006
001372 104413
001372 004242

```
      ;PROGRAM CONTROL FLAGS  
      -----  
INIFLG: .BYTE 0      ;PROGRAM INITIALIZATION FLAG  
ERRFLG: .BYTE 0      ;ERROR OCCURED FLAG  
LOKFLG: .BYTE 0      ;LOCK ON CURRENT TEST FLAG  
QV.FLG: .BYTE 0      ;QUICK VERIFY FLAG.  
                        ;ON FIRST PASS OF EACH DUP11 ITERATIONS  
                        ;WILL BE SUPPRESSED  
  
.EVEN  
$Y=0  
  
      ;DEFINITIONS FOR TRAP SUBROUTINE CALLS  
      ;POINTERS TO SUBROUTINES CAN BE FOUND  
      ;IN THE TABLE IMMEDIATLY FOLLOWING THE DEFINITIONS  
  
      ;*****  
      -----  
      .TRPTAB:  
SCOPE=TRAP+0      ;CALL TO SCOPE LOOP AND ITERATION HANDLER  
      .SCOPE  
SCOPE1=TRAP+1      ;CALL TO LOOP ON CURRENT DATA HANDLER  
      .SCOPE1  
TYPE=TRAP+2      ;CALL TO TELETYPE OUTPUT ROUTINE  
      .TYPE  
INSTR=TRAP+3      ;CALL TO ASCII STRING INPUT ROUTINE  
      .INSTR  
INSTER=TRAP+4      ;CALL TO INPUT ERROR HANDLER  
      .INSTER  
PARAM=TRAP+5      ;CALL TO NUMERICAL DATA INPUT ROUTINE  
      .PARAM  
SAV05=TRAP+6      ;CALL TO REGISTER SAVE ROUTINE  
      .SAV05  
RES05=TRAP+7      ;CALL TO REGISTER RESTORE ROUTINE  
      .RES05  
CONVRT=TRAP+10     ;CALL TO DATA OUTPUT ROUTINE  
      .CONVRT  
CNVRT=TRAP+11     ;CALL TO DATA OUTPUT ROUTINE WITHOUT CR/LF.  
      .CNVRT  
PKCLK=TRAP+12     ;CALL TO CLOCK ROUTINE  
      .PKCLK  
SETFLG=TRAP+13    ;CALL TO TELETYPE INPUT ROUTINE  
      .SETFLG  
  
      ;*****  
      -----
```

PROGRAM PARAMETERS, VARIABLES, AND TRAP CALLS.

```
836           ;DUP11 VECTOR AND REGISTER INDIRECT POINTERS
837
838 001374 000000  DUPRVC: 0           ;POINTER TO DUP11 RECEIVER INTERRUPT VECTOR
839 001376 000000  DUPRPS: 0           ;POINTER TO DUP11 RECEIVER INTERRUPT SERVICE PS
840 001400 000000  DUPTVC: 0           ;POINTER TO DUP11 TRANSMITTER INTERRUPT VECTOR
841 001402 000000  DUPTPS: 0          ;POINTER TO DUP11 TRANSMITTER INTERRUPT SERVICE PS
842 001404 000000  RXCSR: 0           ;POINTER TO DUP11 RECEIVER STATUS REGISTER
843 001406 000000  RXDBUF: 0          ;POINTER TO DUP11 RECEIVER DATA BUFFER
844 001410 000000  PARCSR: 0          ;POINTER TO DUP11 PARAMETER STATUS REGISTER
845 001412 000000  TXCSR: 0           ;POINTER TO DUP11 TRANSMITTER STATUS REGISTER
846 001414 000000  TXDBUF: 0          ;POINTER TO DUP11 TRANSMITTER DATA BUFFER
847 001416 000000  DUPSEC: 0          ;POINTER TO DUP11 SECONDARY REGISTER SELECT REGISTER
848 001420 000000  HUPPSR: 0          ;POINTER TO PARAMETER STATUS HIGH BYTE
849 001422 000000  HUPRBF: 0          ;POINTER TO RECEIVER BUFFER HIGH BYTE
850 001424 000000  HUPRCR: 0          ;POINTER TO RECEIVER CONTROL REG HIGH BYTE
851 001426 000000  HUPTBF: 0          ;POINTER TO TRANSMITTER BUFFER HIGH BYTE
852 001430 000000  HUPTCR: 0          ;POINTER TO TRANSMITTER CONTROL REG HIGH BYTE
853
854
855           ;DUP11 CONTROL INDICATORS FOR CURRENT DUP11 UNDER TEST
856           ;-----
857
858 001432 000      MASK.A: .BYTE 000      ;LAST CHAR TO TEST AND PARITY MASK
859
860 001433 010      CLK.A: .BYTE 8.         ;NUMBER OF CLOCKS NEEDED FOR ONE CHAR
861
862 001434 000000   L00.00: 000000      ;PARAMETERS
863
```

```
864                                     ;DUP11 STATUS TABLE AND ADDRESS ASSIGNMENTS
865                                     ;-----
866                                     . =1500
867 001500 001500 DUP.MAP:
868 001500 000001 DUPCR0: .BLKW 1 ;CONTROL STATUS REGISTER FOR DUP11 NUMBER 0
869 001502 000001 DUPTR0: .BLKW 1 ;VECTOR 'A' FOR DUP11 NUMBER 0
870 001504 000001 DUPO.A: .BLKW 1 ;PARAMETER FOR DUP11 NUMBER 0
871
872
873 001506 000001 DUPCR1: .BLKW 1 ;CONTROL STATUS REGISTER FOR DUP11 NUMBER 1
874 001510 000001 DUPTR1: .BLKW 1 ;VECTOR 'A' FOR DUP11 NUMBER 1
875 001512 000001 DUP1.A: .BLKW 1 ;PARAMETER FOR DUP11 NUMBER 1
876
877 001514 000001 DUPCR2: .BLKW 1 ;CONTROL STATUS REGISTER FOR DUP11 NUMBER 2
878 001516 000001 DUPTR2: .BLKW 1 ;VECTOR 'A' FOR DUP11 NUMBER 2
879 001520 000001 DUP2.A: .BLKW 1 ;PARAMETER FOR DUP11 NUMBER 2
880
881 001522 000001 DUPCR3: .BLKW 1 ;CONTROL STATUS REGISTER FOR DUP11 NUMBER 3
882 001524 000001 DUPTR3: .BLKW 1 ;VECTOR 'A' FOR DUP11 NUMBER 3
883 001526 000001 DUP3.A: .BLKW 1 ;PARAMETER FOR DUP11 NUMBER 3
884
885 001530 000001 DUPCR4: .BLKW 1 ;CONTROL STATUS REGISTER FOR DUP11 NUMBER 4
886 001532 000001 DUPTR4: .BLKW 1 ;VECTOR 'A' FOR DUP11 NUMBER 4
887 001534 000001 DUP4.A: .BLKW 1 ;PARAMETER FOR DUP11 NUMBER 4
888
889 001536 000001 DUPCR5: .BLKW 1 ;CONTROL STATUS REGISTER FOR DUP11 NUMBER 5
890 001540 000001 DUPTR5: .BLKW 1 ;VECTOR 'A' FOR DUP11 NUMBER 5
891 001542 000001 DUP5.A: .BLKW 1 ;PARAMETER FOR DUP11 NUMBER 5
892
893 001544 000001 DUPCR6: .BLKW 1 ;CONTROL STATUS REGISTER FOR DUP11 NUMBER 6
894 001546 000001 DUPTR6: .BLKW 1 ;VECTOR 'A' FOR DUP11 NUMBER 6
895 001550 000001 DUP6.A: .BLKW 1 ;PARAMETER FOR DUP11 NUMBER 6
896
897 001552 000001 DUPCR7: .BLKW 1 ;CONTROL STATUS REGISTER FOR DUP11 NUMBER 7
898 001554 000001 DUPTR7: .BLKW 1 ;VECTOR 'A' FOR DUP11 NUMBER 7
899 001556 000001 DUP7.A: .BLKW 1 ;PARAMETER FOR DUP11 NUMBER 7
900
901 001560 000000 DUP.END: 000000
902
903
904
905
906
```

PROGRAM PARAMETERS, VARIABLES, AND TRAP CALLS.

| | | | | | | | | | | | | | | | |
|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| 15 | 14 | 13 | 12 | 11 | 10 | 09 | 08 | 07 | 06 | 05 | 04 | 03 | 02 | 01 | 00 |
| I | I | I | I | I | I | I | I | I | I | I | I | I | I | I | I |
| I | C | O | N | T | R | O | L | | R | E | G | I | S | T | E |
| I | I | I | I | I | I | I | I | I | I | I | I | I | I | I | I |
| I | I | I | I | I | I | I | I | I | I | I | I | I | I | I | I |
| I | I | I | I | I | I | I | I | I | I | I | I | I | I | I | I |
| I | I | I | I | I | I | I | I | I | I | I | I | I | I | I | I |
| I | A | B | C | D | E | F | G | H | I | * | * | S | Y | N | C |
| I | I | I | I | I | I | I | I | I | I | I | I | I | I | I | I |

DEFINITIONS

- A- OPTIONAL CLEAR JUMPER IN-1
- B- TURNAROUND CONNECTOR ON 1
- C-
- D-

PROGRAM INITIALIZATION AND START UP.

```

926
927      ;PROGRAM INITIALIZATION
928      ;LOCK OUT INTERRUPTS
929      ;SET UP PROCESSOR STACK
930      ;SET UP POWER FAIL VECTOR
931      ;CLEAR PROGRAM CONTROL FLAGS AND COUNTS
932      ;TYPE TITLE MESSAGE
933
934 001562 012737 000340 177776 .START: MOV #340,PS ;LOCK OUT INTERRUPTS
935 001570 012706 001150 MOV #STACK,SP ;SET UP STACK
936 001574 012737 005050 000024 MOV #.PFAIL,@#24 ;SET UP POWER FAIL VECTOR
937 001602 113737 001311 001313 MOVB DUPNUM,SAVNUM ;SAVE NUMBER OF DEVICES IN SYSTEM
938 001610 005037 001230 CLR PASCNT ;CLEAR PASS COUNT
939 001614 105037 001341 CLRB ERRFLG ;CLEAR ERROR FLAG
940 001620 105037 001343 CLRB QV.FLG ;ZERO QUICK VERIFY FLAG
941 001624 012737 001500 001316 MOV #DUP.MAP,CREAM ;GET MAP POINTER.
942 001632 112737 000001 001314 MOVB #1,RUN ;POINT POINTER TO FIRST DEVICE.
943 001640 005037 001232 CLR ERRCNT ;CLEAR ERROR COUNT
944 001644 005037 001234 CLR LSTERR ;CLEAR LAST ERROR POINTER
945 001650 012737 000001 001226 MOV #1,TSTNO ;SET UP FOR TEST 1
946 001656 012737 001562 001214 MOV #.START,RETURN ;SET UP FOR POWER FAIL BEFORE
947 ;TESTING STARTS
948 001664 013746 000006 MOV @#6,-(SP) ;SAVE CURRENT VECTORS
949 001670 013746 000004 MOV @#4,-(SP)
950 001674 012737 001710 000004 MOV #12$,@#4 ;SETUP FOR TIMEOUT
951 001702 005777 177274 TST @SWR ;REFERENCE HARDWARE SWITCH REG
952 001706 000407 BR 13$ ;BR IF IT EXISTS
953 001710 012737 000176 001202 12$: MOV #SWREG,SWR ;POINT TO SOFT SWR
954 001716 012737 000174 001200 MOV #DISPREG,DISPLAY ;POINT TO SOFT DISPLAY REG
955 001724 022626 CMP (SP)+,(SP)+ ;ADJUST STACK
956 001726 012637 000004 13$: MOV (SP)+,@#4 ;RESTORE VECTORS
957 001732 012637 000006 MOV (SP)+,@#6
958 001736 105737 001340 TSTB INIFLG ;HAS INITIALIZATION BEEN PERFORMED
959 001742 001401 BEQ 11$
960 001744 000410 BR 6$
961 001746 022737 003104 000042 11$: CMP #SENDAD,@#42 ;IF ACT-11 AUTO MODE,
962 001754 001404 BEQ 6$ ;DON'T TYPE ID
963 001756 104402 001000 TYPE ,MTITLE ;TYPE TITLE MESSAGE
964 001762 105137 001340 COMB INIFLG ;IF NOT SET FLAG AND DO
965 001766 105777 177210 6$: TSTB @SWR ;BIT7=1??
966 001772 100002 BPL 10$
967 001774 000137 002520 JMP 1$
968 002000
969 002000 032777 000001 177174 10$: BIT #SW00,@SWR ;ENTER PARAMETERS
970 002006 001002 BNE .+6 ;YES
971 002010 000137 002300 JMP 21$ ;NO
972 002014 105137 001332 COMB FLAG
973 002020 112737 000001 001340 MOVB #1,INIFLG ;SET TO MANUAL ENTRY
974 002026 012700 001500 MOV #DUP.MAP,RO ;CLR MAP
975 002032 005020 CLR (RO)+
976 002034 020027 001560 68$: CMP RO,#DUP.END ;DONE WITH MAP?
977 002040 001374 BNE 68$ ;BR IF NO
978 002042 104403 INSTR ;OUTPUT MESSAGE & GET INPUT STRING
979 002044 005473 MCSR ;MESSAGE
980 002046 104405 PARAM ;CONVERT STRING
981 002050 160000 160000 ;LOW LIMIT

```


PROGRAM INITIALIZATION AND START UP.

| | | | | | | |
|------|--------|--------|--------|--------|---------|------------------------------------|
| 982 | 002052 | 175500 | | | 175500 | :HIGH LIMIT |
| 983 | 002054 | 001500 | | | DUPCRO | :STORE AT THIS LOCATION |
| 984 | 002056 | 001 | | | .BYTE 1 | :MASK |
| 985 | 002057 | 001 | | | .BYTE 1 | :HOW MANY TIMES + 2 |
| 986 | 002060 | 104403 | | | INSTR | :OUTPUT MESSAGE & GET INPUT STRING |
| 987 | 002062 | 005512 | | | MVEC | :MESSAGE |
| 988 | 002064 | 104405 | | | PARAM | :CONVERT STRING |
| 989 | 002066 | 000300 | | | 300 | :LOW LIMIT |
| 990 | 002070 | 000770 | | | 770 | :HIGH LIMIT |
| 991 | 002072 | 001502 | | | DUPTR0 | :STORE AT THIS LOCATION |
| 992 | 002074 | 001 | | | .BYTE 1 | :MASK |
| 993 | 002075 | 001 | | | .BYTE 1 | :HOW MANY TIMES + 2 |
| 994 | 002076 | 104403 | | | INSTR | :OUTPUT MESSAGE & GET INPUT STRING |
| 995 | 002100 | 005702 | | | MPAR | :MESSAGE |
| 996 | 002102 | 104405 | | | PARAM | :CONVERT STRING |
| 997 | 002104 | 000004 | | | 4 | :LOW LIMIT |
| 998 | 002106 | 000007 | | | 7 | :HIGH LIMIT |
| 999 | 002110 | 001240 | | | TEMP2 | :STORE AT THIS LOCATION |
| 1000 | 002112 | 000 | | | .BYTE 0 | :MASK |
| 1001 | 002113 | 001 | | | .BYTE 1 | :HOW MANY TIMES + 2 |
| 1002 | 002114 | 013737 | 001240 | 001320 | MOV | TEMP2,PRIRTY :SAVE PRIORITY |
| 1003 | 002122 | 104403 | | | INSTR | :OUTPUT MESSAGE & GET INPUT STRING |
| 1004 | 002124 | 005647 | | | MTOTAL | :MESSAGE |
| 1005 | 002126 | 104405 | | | PARAM | :CONVERT STRING |
| 1006 | 002130 | 000001 | | | 1 | :LOW LIMIT |
| 1007 | 002132 | 000010 | | | 8 | :HIGH LIMIT |
| 1008 | 002134 | 001236 | | | TEMP1 | :STORE AT THIS LOCATION |
| 1009 | 002136 | 000 | | | .BYTE 0 | :MASK |
| 1010 | 002137 | 001 | | | .BYTE 1 | :HOW MANY TIMES + 2 |
| 1011 | 002140 | 104403 | | | INSTR | :OUTPUT MESSAGE & GET INPUT STRING |
| 1012 | 002142 | 005525 | | | MJMPR | :MESSAGE |
| 1013 | 002144 | 104413 | | | SETFLG | :SET FLAG BASED UPON INPUT STRING |
| 1014 | 002146 | 001323 | | | OPCLRJ | :THIS FLAG |
| 1015 | 002150 | 104403 | | | INSTR | :OUTPUT MESSAGE & GET INPUT STRING |
| 1016 | 002152 | 005600 | | | MTCN | :MESSAGE |
| 1017 | 002154 | 104413 | | | SETFLG | :SET FLAG BASED UPON INPUT STRING |
| 1018 | 002156 | 001322 | | | TCNFLG | :THIS FLAG |
| 1019 | 002160 | 105737 | 001322 | | TSTB | TCNFLG |
| 1020 | 002164 | 001410 | | | BEQ | 71\$ |
| 1021 | 002166 | 104403 | | | INSTR | :OUTPUT MESSAGE & GET INPUT STRING |
| 1022 | 002170 | 005726 | | | MSTJM | :MESSAGE |
| 1023 | 002172 | 104413 | | | SETFLG | :SET FLAG BASED UPON INPUT STRING |
| 1024 | 002174 | 001334 | | | STJMFL | :THIS FLAG |
| 1025 | 002176 | 104403 | | | INSTR | :OUTPUT MESSAGE & GET INPUT STRING |
| 1026 | 002200 | 005761 | | | MSRJM | :MESSAGE |
| 1027 | 002202 | 104413 | | | SETFLG | :SET FLAG BASED UPON INPUT STRING |
| 1028 | 002204 | 001336 | | | SRJMFL | :THIS FLAG |
| 1029 | 002206 | 105737 | 001323 | | TSTB | OPCLRJ |
| 1030 | 002212 | 001403 | | | BEQ | 69\$ |
| 1031 | 002214 | 052737 | 100000 | 001504 | BIS | #BIT15,DUPO.A |
| 1032 | 002222 | 105737 | 001322 | | TSTB | TCNFLG |
| 1033 | 002226 | 001403 | | | BEQ | 70\$ |
| 1034 | 002230 | 052737 | 040000 | 001504 | BIS | #BIT14,DUPO.A |
| 1035 | 002236 | 112737 | 000001 | 001312 | MOVB | #1,SAVACT |
| 1036 | 002244 | 113737 | 001236 | 001311 | MOVB | TEMP1,DUPNUM |
| 1037 | 002252 | 113737 | 001236 | 001313 | MOVB | TEMP1,SAVNUM |

```

1038 002260 005337 001236      65$: DEC    TEMP1
1039 002264 001404              BEQ    64$
1040 002266 000261              SEC
1041 002270 106137 001312      ROLB  SAVACT
1042 002274 000771              BR    65$
1043 002276 113737 001312 001240 64$: MOVB  SAVACT,TEMP2 ;# OF TIMES
1044 002304 113737 001312 001310 MOVB  SAVACT,DUPACTV
1045 002312 000241              CLC
1046 002314 106037 001240      RORB  TEMP2
1047 002320 012700 001500      MOV  #DUPCRO,R0
1048 002324 012701 001506      MOV  #DUPCR1,R1
1049 002330 000241      67$: CLC
1050 002332 106037 001240      RORB  TEMP2
1051 002336 103051              BCC  66$
1052 002340 012011              MOV  (R0)+,(R1)
1053 002342 062721 000010      ADD  #10,(R1)+ ;CSR
1054 002346 012011              MOV  (R0)+,(R1)
1055 002350 062721 000010      ADD  #10,(R1)+ ;VECTOR
1056 002354 012021              MOV  (R0)+,(R1)+ ;PARAMETERS
1057 002356 000764              BR   67$
1058 002360 012700 001500      21$:  JV   #DUP.MAP,R0 ;SETUP TO CLEAR MAP
1059 002364 005020      20$: LLR  (R0)+ ;CLEAR
1060 002366 020027 001560      CMP  R0,#DUP.END ;CHECK FOR FINISH
1061 002372 001374              BNE  20$ ;BR IF MORE TO GO
1062 002374 012700 001500      MOV  #DUP.MAP,R0 ;SETUP TO DEFAULT
1063 002400 012710 160050      MOV  #160050,(R0) ;LOAD CSR
1064 002404 012760 000770 000002 MOV  #770,2(R0) ;LOAD VECTOR
1065 002412 012760 140026 000004 MOV  #140026,4(R0) ;LOAD PARAMETERS AND SYNC
1066 002420 112737 000005 001320 MOVB  #5,PRIORITY ;LOAD PRIORITY
1067 002426 012700 000001      MOV  #1,R0 ;SAVE CORE THIS WAY
1068 002432 110037 001310      MOVB R0,DUPACTV ;PRESET PROGRAM CONTROLS
1069 002436 110037 001311      MOVB R0,DUPNUM ;DITTO
1070 002442 110037 001312      MOVB R0,SAVACT ;DITTO
1071 002446 110037 001313      MOVB R0,SAVNUM ;DITTO
1072 002452 110037 001322      MOVB R0,TCNFLG ;DITTO
1073 002456 110037 001323      MOVB R0,OPCLRJ ;DITTO
1074 002462
1075 002462 104402 006014      66$: TYPE  ,XHEAD ;TYPE HEADER
1076 002466 012737 001500 001236 16$: MOV  #DUP.MAP,TEMP1 ;SET POINTER
1077 002474 017737 176536 001240 5$: MOV  @TEMP1,TEMP2 ;SET DATA
1078 002502 001406              BEQ  1$ ;ALL DONE WITH DATA
1079 002504 104410              CONVRT
1080 002506 006042              XSTATQ
1081 002510 062737 000002 001236      ADD  #2,TEMP1 ;UPDATE POINTER
1082 002516 000766              BR   5$
1083 002520 032777 000001 176454 1$: BIT  #SW00,@SWR
1084 002526 001405              BEQ  7$
1085 002530 005737 001332      TST  FLAG
1086 002534 001002              BNE  7$
1087 002536 000137 002000      JMP  10$
1088 002542 005037 001332      7$: CLR  FLAG
1089 002546 005737 000042      TST  @#42 ;IS PROGRAM RUNNING UNDER MONITOR
1090 002552 001030              BNE  3$ ;BR IF YES
1091 002554 032777 000010 176420      BIT  #SW03,@SWR ;SELECT SPECIFIC DEVICES??
1092 002562 001424              BEQ  3$ ;BR IF NO.
1093 002564 104402 005413      TYPE  ,MNEW ;TYPE THE MESSAGE.

```

```

1094 002570 005000          CLR      R0          ;ZERO DATA LIGHTS
1095 002572 000000          HALT          ;WAIT FOR USER TO TELL WHAT DEVICES TO RUN
1096 002574 127737 176402 001312  CMPB     @SWR,SAVACT ;IS THE NUMBER VALID?
1097 002602 101404          BLOS     2$          ;BR IF NUMBER IS OK.
1098 002604 104402 005254    TYPE     ,MERR3     ;TELL USER OF INVALID NUMBER.
1099 002610 000000          HALT          ;STOP EVERY THING.
1100 002612 000776          BR       .-2         ;RESTART THE PROGRAM AGAIN.
1101 002614 117737 176362 001310 2$:  MOVB    @SWR,DUPACTV ;GET NEW DEVICE PATTERN
1102 002622 113700 001310    MOVB    DUPACTV,R0  ;SHOW THE USER WHAT HE SELECTED.
1103 002626 042700 177400    BIC     #'C<377>,R0 ;USE ONLY LOW BYTE.
1104 002632 000000          HALT          ;CONTINUE DYNAMIC SWITCHES.
1105 002634 012700 000300    3$:  MOV     #300,R0    ;PREPARE TO CLEAR THE FLOATING
1106 002640 012701 000302    MOV     #302,R1    ;VECTOR AREA. 300-776
1107 002644 010120          4$:  MOV     R1,(R0)+   ;START PUTTING 'PC+2 - HALT'
1108 002646 005021          CLR     (R1)+      ;IN VECTOR AREA.
1109 002650 022021          CMP     (R0)+,(R1)+ ;POP POINTERS
1110 002652 022700 001000    CMP     #1000,R0   ;ALL DONE??
1111 002656 001372          BNE     4$         ;BR IF NO.
1112
1113                          ;TEST START AND RESTART
1114                          ;-----
1115
1116 002660 012737 000340 177776 .BEGIN: MOV     #340,PS    ;LOCK OUT INTERRUPTS
1117 002666 012706 001150    MOV     #STACK,SP ;SET UP STACK
1118 002672 005737 000042    TST     @#42      ;IS PROGRAM UNDER MONITOR CONTROL
1119 002676 001023          BNE     2$         ;BR IF YES
1120 002700 032777 000004 176274    BIT     #BIT2,@SWR ;CHECK FOR LOCK ON TEST
1121 002706 001411          BEQ     1$         ;BR IF NO LOCK DESIRED.
1122 002710 104402 005312    TYPE     ,MLOCK   ;TYPE LOCK SELECTED.
1123 002714 012737 000240 003174    MOV     #NOP,TTST ;ADJUST SCOPE ROUTINE.
1124 002722 012737 000240 003176    MOV     #NOP,TTST+2 ;SET UP TO LOCK
1125 002730 000406          BR       2$         ;CONTINUE ALONG.
1126 002732 013737 003306 003174 1$:  MOV     BRW,TTST   ;PREPARE NORMAL SCOPE ROUTINE
1127 002740 013737 003310 003176    MOV     BRX,TTST+2 ;LOCK NOT SELECTED, SET UP FOR NORMAL SCOPE LOOP
1128 002746 012737 006224 001214 2$:  MOV     #CYCLE,RETURN ;START AT 'CYCLE' FIND WHICH DEVICE TO TEST
1129 002754 104402 005202    TYPE     ,MR      ;TYPE R
1130 002760 000177 176230    JMP     @RETURN    ;START TESTING

```

```

1131                                     ;END OF PASS
1132                                     ;TYPE NAME OF TEST
1133                                     ;UPDATE PASS COUNT
1134                                     ;CHECK FOR EXIT TO ACT-11
1135                                     ;RESTART TEST
1136
1137 002764 005037 001234      .EOP: CLR      LSTERR      ;CLEAR LAST ERROR PC
1138 002770 105037 001341      CLR RB  ERRFLG      ;CLEAR ERROR FLAG
1139 002774 005237 001230      INC      PASCNT     ;UPDATE PASS COUNT
1140 003000 013777 001230 176172 MOV     PASCNT,@DISPLAY ;DISPLAY PASS COUNT
1141 003006 104402 005157      TYPE    ,MEPASS    ;TYPE END PASS
1142 003012 104402 005341      TYPE    ,MCSRX     ;TYPE CSR
1143 003016 104411 003130      CNVRT   ,XCSR      ;SHOW IT
1144 003022 104402 005347      TYPE    ,MVECX     ;TYPE VECTOR
1145 003026 104411 003136      CNVRT   ,XVEC      ;SHOW IT
1146 003032 104402 005355      TYPE    ,MPASSX    ;TYPE PASSES
1147 003036 104411 003144      CNVRT   ,XPASS     ;SHOW IT
1148 003042 104402 005366      TYPE    ,MERRX     ;TYPE ERRORS
1149 003046 104411 003152      CNVRT   ,XERR      ;SHOW IT
1150 003052 105337 001313      DECB   SAVNUM      ;ARE ALL DEVICES TESTED?
1151 003056 001017      BNE     RESTRT     ;BR IF NO.
1152 003060 112737 000377 001343 MOV B   #377,QV,FLG ;SET THE QUICK VERIFY FLAG.
1153 003066 113737 001311 001313 MOV B   DUPNUM,SAVNUM ;RESTORE THE COUNT
1154 003074 013701 000042      MOV     @#42,R1    ;CHECK FOR ACT-11 OR DDP
1155 003100 001406      BEQ    RESTRT     ;IF NOT, CONTINUE TESTING
1156 003102 000005      RESET                    ;STOP THE SHOW--CLEAR THE WORLD
1157 003104
1158 003104 004711      $ENDAD: JSR     PC,(R1)
1159 003106 000240      NOP
1160 003110 000240      NOP
1161 003112 000240      NOP
1162 003114 000240      NOP
1163 003116 012737 006224 001214 RESTRT: MOV     #CYCLE,RETURN
1164 003124 000137 006224      JMP     CYCLE
1165 003130 000001      XCSR:   1
1166 003132      006      002      .BYTE   6,2
1167 003134 001404      RXCSR
1168 003136 000001      XVEC:   1
1169 003140      003      002      .BYTE   3,2
1170 003142 001374      DUPRVC
1171 003144 000001      xPASS: 1
1172 003146      006      002      .BYTE   6,2
1173 003150 001230      PASCNT
1174 003152 000001      XERR:   1
1175 003154      006      002      .BYTE   6,2
1176 003156 001232      ERRCNT
1177
1178                                     ;SCOPE LOOP AND ITERATION HANDLER
1179
1180 003160 005037 001234      .SCOPE: CLR     LSTERR      ;CLEAR LAST ERROR PC
1181 003164 010016      MOV     R0,(SP)   ;SAVE R0 ON STACK
1182 003166 032777 040000 176006 BIT     #BIT14,@SWR ;LOOP ON TEST?
1183 003174 001407      TTST:  BEQ     1$    ;BR IF NO (IF LOCK SW01 = 1;THIS LOCATION = 240)
1184 003176 000437      BR     3$        ;GO TO 3$ (DITTO)
1185 003200 105777 176000      TST B  @TKCSR    ;KYBD DONE?
1186 003204 100034      BPL     3$        ;BR IF NO (LOCK: HIT A KEY ON TTY TO GO TO NEXT TEST)

```

```

1187 003206 017700 175774      MOV    @TKDBR,R0      ;CLR DONE BIT
1188 003212 000415      BR     2$            ;CONTINUE
1189 003214 032777 004000 175760 1$:  BIT    #SW11,@SWR    ;DELETE ITERATION (QUICK PASS)?
1190 003222 001011      BNE   2$            ;BR IF YES
1191 003224 105737 001343      TSTB  QV.FLG        ;HAS FIRST PASS BEEN COMPLETED?
1192 003230 001406      BEQ   2$            ;BR IF QUICK VERIFY
1193 003232 005237 001224      INC   LPCNT         ;UPDATE ITERATION COUNTER
1194 003236 023737 001224 001222  CMP    LPCNT,ICOUNT  ;ALL ITERATIONS DONE?
1195 003244 001014      BNE   3$            ;BR IF NOT YET
1196 003246 105037 001341      CLRB  ERRFLG        ;PREPARE FOR NEW TEST
1197 003252 005037 001224      CLR   LPCNT         ;START ICOUNT AT ZERO
1198 003256 005037 001220      CLR   LOCK          ;
1199 003262 012737 000050 001222  MOV    #50,ICOUNT    ;RESET ITERATIONS
1200 003270 013737 001216 001214  MOV    NEXT,RETURN   ;GET NEXT TEST
1201 003276 011600      MOV    (SP),R0      ;POP R0 OFF STACK
1202 003300 022626      POP2SP              ;FAKE AN RTI
1203 003302 000177 175706      JMP   @RETURN       ;GO DO THE TEST
1204 003306 001407      BRW:  1407
1205 003310 000437      BRX:  437
1206
1207      ;CHECK FOR FREEZE ON CURRENT DATA
1208      ;-----
1209
1210 003312 032777 001000 175662 .SCOP1: BIT    #SW09,@SWR    ;IS SW09-1(SET)?
1211 003320 001405      BEQ   1$            ;BR IF NOT SET.
1212 003322 005737 001220      TST   LOCK          ;
1213 003326 001402      BEQ   1$            ;
1214 003330 013716 001220      MOV    LOCK,(SP)    ;GOTO THE ADDRESS IN LOCK.
1215 003334 000002      1$:  RTI             ;GO BACK.
1216
1217      ;TELETYPE OUTPUT ROUTINE
1218      ;-----
1219
1220 003336 010546      .TYPE: MOV    R5,-(SP)    ;SAVE R5 ON THE STACK.
1221 003340 017605      MOV    @2(SP),R5    ;GET ADDRESS OF MESSAGE.
1222 003344 062766 000002 000002  ADD    #2,2(SP)     ;POP OVER ADDRESS.
1223 003352 032777 010000 175622 1$:  BIT    #SW12,@SWR    ;INHIBIT ALL PRINT OUT??
1224 003360 001012      BNE   3$            ;BR IF NO PRINT OUT WANTED (SW12=1)
1225 003362 105715      TSTB  (R5)          ;IS NUMBER MINUS? (MSB-1(BIT7))
1226 003364 100002      BPL   2$            ;BR IF NUMBER IS PLUS
1227 003366 104402 005136      TYPE  ,MCR LF       ;TYPE A CR/LF!
1228 003372 105777 175612      2$:  TSTB  @TPCSR     ;TTY READY?
1229 003376 100375      BPL   2$            ;BR IF NO.
1230 003400 112577 175606      MOVB  (R5)+,@TPDBR  ;PRINT CURRENT CHAR.
1231 003404 001362      BNE   1$            ;IF NOT ZERO KEEP PRINTING!
1232 003406 012605      3$:  MOV    (SP)+,R5   ;END OF OUTPUT. RESTORE R5
1233 003410 000002      RTI             ;GO HOME
1234
1235      ;-----
1236 003412 010346      .INSTR: MOV    R3,-(SP)  ;SAVE R3 ON STACK
1237 003414 010446      MOV    R4,-(SP)    ;SAVE R4 ON STACK
1238 003416 017637 000004 003434  MOV    @4(SP),MSG   ;
1239 003424 062766 000002 000004  ADD    #2,4(SP)     ;
1240 003432 104402      .INST1: TYPE
1241 003434 000000      .MSG:  0
1242 003436 012704 006160      MOV    #INBUF,R4

```

```

1243 003442 012703 000007      MOV      #7,R3
1244 003446 105777 175532      1$: TSTB   @TKCSR
1245 003452 100375          BPL      1$
1246 003454 117714 175526      MOVB    @TKDBR,(R4)
1247 003460 142714 000200      BICB    #200,(R4)
1248 003464 122427 000015      CMPB    (R4)+,#15
1249 003470 001417          BEQ     INSTR2
1250 003472 105777 175512      2$: TSTB   @TPCSR
1251 003476 100375          BPL      2$
1252 003500 017777 175502 175504      MOV     @TKDBR,@TPDBR
1253 003506 005303          DEC     R3
1254 003510 001356          BNE     1$
1255 003512 012604          MOV     (SP)+,R4
1256 003514 012603          MOV     (SP)+,R3
1257 003516 010346      .INSTE: MOV    R3,-(SP)
1258 003520 010446          MOV     R4,-(SP)
1259 003522 104402 005132          TYPE   ,MQM
1260 003526 000741          BR      .INST1
1261 003530 012604      INSTR2: MOV   (SP)+,R4      ;RESTORE R4
1262 003532 012603          MOV   (SP)+,R3      ;RESTORE R3
1263 003534 000002          RTI
1264
1265          ;CONVERT ASCII STRING TO OCTAL
1266          :-----
1267
1268 003536 010546      .PARAM: MOV   R5,-(SP)
1269 003540 010446          MOV   R4,-(SP)
1270 003542 016605 000004          MOV   4(SP),R5
1271 003546 012537 003726          MOV   (R5)+,LOLIM
1272 003552 012537 003730          MOV   (R5)+,HILIM
1273 003556 012537 003732          MOV   (R5)+,DEVADR
1274 003562 112537 003734          MOVB  (R5)+,LOBITS
1275 003566 112537 003735          MOVB  (R5)+,ADRCNT
1276 003572 010566 000004          MOV   R5,4(SP)
1277 003576 005005      PARAM1: CLR   R5
1278 003600 012704 006160          MOV   #INBUF,R4
1279 003604 122714 000015          CMPB  #15,(R4)
1280 003610 001420          BEQ   PARERR
1281 003612 121427 000060      1$: CMPB  (R4),#60
1282 003616 002415          BLT   PARERR
1283 003620 121427 000067          CMPB  (R4),#67
1284 003624 003012          BGT   PARERR
1285 003626 142714 000060          BICB  #60,(R4)
1286 003632 152405          BISB  (R4)+,R5
1287 003634 122714 000015          CMPB  #15,(R4)
1288 003640 001406          BEQ   LIMITS
1289 003642 006305          ASL   R5
1290 003644 006305          ASL   R5
1291 003646 006305          ASL   R5
1292 003650 000760          BR    1$
1293 003652 104404      PARERR: INSTER
1294 003654 000750          BR    PARAM1
1295
1296          ;TEST TO SEE IF NUMBER IS WITHIN LIMITS
1297          :-----
1298

```

END OF PASS ROUTINE

| | | | | | | |
|------|--------|--------|---------------|--------------|-------------|--|
| 1299 | 003656 | 020537 | 003730 | LIMITS: CMP | R5,HILIM | |
| 1300 | 003662 | 101373 | | BHI | PARERR | |
| 1301 | 003664 | 020537 | 003726 | CMP | R5,LOLIM | |
| 1302 | 003670 | 103770 | | BLO | PARERR | |
| 1303 | 003672 | 133705 | 003734 | BITB | LOBITS,R5 | |
| 1304 | 003676 | 001365 | | BNE | PARERR | |
| 1305 | | | | | | |
| 1306 | | | | | | :STORE NUMBER AT SPECIFIED ADDRESS |
| 1307 | | | | | | |
| 1308 | 003700 | 013704 | 003732 | MOV | DEVADR,R4 | |
| 1309 | 003704 | 010524 | | 1\$: MOV | R5,(R4)+ | |
| 1310 | 003706 | 062705 | 000002 | ADD | #2,R5 | |
| 1311 | 003712 | 105337 | 003735 | DECB | ADRCNT | |
| 1312 | 003716 | 001372 | | BNE | 1\$ | |
| 1313 | 003720 | 012604 | | MOV | (SP)+,R4 | |
| 1314 | 003722 | 012605 | | MOV | (SP)+,R5 | |
| 1315 | 003724 | 000002 | | RTI | | |
| 1316 | 003726 | 000000 | | LOLIM: | 0 | |
| 1317 | 003730 | 000000 | | HILIM: | 0 | |
| 1318 | 003732 | 000000 | | DEVADR: | 0 | |
| 1319 | 003734 | 000000 | | LOBITS: | 0 | |
| 1320 | | 003735 | | ADRCNT= | LOBITS+1 | |
| 1321 | | | | | | |
| 1322 | | | | | | :SAVE PC OF TEST THAT FAILED AND R0-R5 |
| 1323 | | | | | | ----- |
| 1324 | | | | | | |
| 1325 | 003736 | 016637 | 000004 001266 | .SAV05: MOV | 4(SP),SAVPC | :SAVE R7 (PC) |
| 1326 | | | | | | |
| 1327 | | | | | | :SAVE R0-R5 |
| 1328 | | | | | | |
| 1329 | 003744 | 010537 | 001262 | SV05: MOV | R5,SAVR5 | :SAVE R5 |
| 1330 | 003750 | 010437 | 001260 | MOV | R4,SAVR4 | :SAVE R4 |
| 1331 | 003754 | 010337 | 001256 | MOV | R3,SAVR3 | :SAVE R3 |
| 1332 | 003760 | 010237 | 001254 | MOV | R2,SAVR2 | :SAVE R2 |
| 1333 | 003764 | 010137 | 001252 | MOV | R1,SAVR1 | :SAVE R1 |
| 1334 | 003770 | 010037 | 001250 | MOV | R0,SAVR0 | :SAVE R0 |
| 1335 | 003774 | 000002 | | RTI | | :LEAVE. |
| 1336 | | | | | | |
| 1337 | | | | | | :RESTORE R0-R5 |
| 1338 | | | | | | |
| 1339 | 003776 | 013700 | 001250 | .RES05: MOV | SAVR0,R0 | :RESTORE R0 |
| 1340 | 004002 | 013701 | 001252 | MOV | SAVR1,R1 | :RESTORE R1 |
| 1341 | 004006 | 013702 | 001254 | MOV | SAVR2,R2 | :RESTORE R2 |
| 1342 | 004012 | 013703 | 001256 | MOV | SAVR3,R3 | :RESTORE R3 |
| 1343 | 004016 | 013704 | 001260 | MOV | SAVR4,R4 | :RESTORE R4 |
| 1344 | 004022 | 013705 | 001262 | MOV | SAVR5,R5 | :RESTORE R5 |
| 1345 | 004026 | 000002 | | RTI | | :LEAVE |
| 1346 | | | | | | |
| 1347 | | | | | | |
| 1348 | | | | | | :CONVERT OCTAL NUMBER TO ASCII AND OUTPUT TO TELEPRINTER |
| 1349 | | | | | | ----- |
| 1350 | | | | | | |
| 1351 | 004030 | 104402 | 005136 | .CONVR: TYPE | ,MCRLF | |
| 1352 | 004034 | 010046 | | .CNVRT: MOV | R0,-(SP) | |
| 1353 | 004036 | 010146 | | MOV | R1,-(SP) | |
| 1354 | 004040 | 010346 | | MOV | R3,-(SP) | |

| | | | | | | | |
|------|--------|--------|--------|--------|----------|--------------|-------------------------|
| 1355 | 004042 | 010446 | | | MOV | R4,-(SP) | |
| 1356 | 004044 | 010546 | | | MOV | R5,-(SP) | |
| 1357 | 004046 | 017601 | 000012 | | MOV | @12(SP),R1 | |
| 1358 | 004052 | 062766 | 000002 | 000012 | ADD | #2,12(SP) | |
| 1359 | 004060 | 012137 | 004234 | | MOV | (R1)+,WRDCNT | |
| 1360 | 004064 | 112137 | 004236 | | 1\$: | MOVB | (R1)+,CHRCNT |
| 1361 | 004070 | 112137 | 004237 | | | MOVB | (R1)+,SPACNT |
| 1362 | 004074 | 013137 | 004240 | | | MOV | @(R1)+,BINWRD |
| 1363 | 004100 | 013704 | 004240 | | 2\$: | MOV | BINWRD,R4 |
| 1364 | 004104 | 113705 | 004236 | | | MOVB | CHRCNT,R5 |
| 1365 | 004110 | 012700 | 006054 | | | MOV | #TEMP,R0 |
| 1366 | 004114 | 010403 | | | 3\$: | MOV | R4,R3 |
| 1367 | 004116 | 042703 | 177770 | | | BIC | #177770,R3 |
| 1368 | 004122 | 062703 | 000060 | | | ADD | #060,R3 |
| 1369 | 004126 | 110320 | | | | MOVB | R3,(R0)+ |
| 1370 | 004130 | 000241 | | | | CLC | |
| 1371 | 004132 | 006004 | | | | ROR | R4 |
| 1372 | 004134 | 000241 | | | | CLC | |
| 1373 | 004136 | 006004 | | | | ROR | R4 |
| 1374 | 004140 | 000241 | | | | CLC | |
| 1375 | 004142 | 006004 | | | | ROR | R4 |
| 1376 | 004144 | 005305 | | | | DEC | R5 |
| 1377 | 004146 | 001362 | | | | BNE | 3\$ |
| 1378 | 004150 | 012703 | 006116 | | | MOV | #MDATA,R3 |
| 1379 | 004154 | 114023 | | | 4\$: | MOVB | -(R0),(R3)+ |
| 1380 | 004156 | 105337 | 004236 | | | DECB | CHRCNT |
| 1381 | 004162 | 001374 | | | | BNE | 4\$ |
| 1382 | 004164 | 105737 | 004237 | | | TSTB | SPACNT |
| 1383 | 004170 | 001405 | | | | BEQ | 6\$ |
| 1384 | 004172 | 112723 | 000040 | | 5\$: | MOVB | #040,(R3)+ |
| 1385 | 004176 | 105337 | 004237 | | | DECB | SPACNT |
| 1386 | 004202 | 001373 | | | | BNE | 5\$ |
| 1387 | 004204 | 105013 | | | 6\$: | CLRB | (R3) |
| 1388 | 004206 | 104402 | 006116 | | | TYPE | ,MDATA |
| 1389 | 004212 | 005337 | 004234 | | | DEC | WRDCNT |
| 1390 | 004216 | 001322 | | | | BNE | 1\$ |
| 1391 | 004220 | 012605 | | | | MOV | (SP)+,R5 |
| 1392 | 004222 | 012604 | | | | MOV | (SP)+,R4 |
| 1393 | 004224 | 012603 | | | | MOV | (SP)+,R3 |
| 1394 | 004226 | 012601 | | | | MOV | (SP)+,R1 |
| 1395 | 004230 | 012600 | | | | MOV | (SP)+,R0 |
| 1396 | 004232 | 000002 | | | | RTI | |
| 1397 | 004234 | 000000 | | | | WRDCNT: | 0 |
| 1398 | 004236 | 000000 | | | | CHRCNT: | 0 |
| 1399 | | 004237 | | | | SPACNT= | CHRCNT+1 |
| 1400 | 004240 | 000000 | | | | BINWRD: | 0 |
| 1401 | | | | | | | |
| 1402 | | | | | | | |
| 1403 | | | | | | | |
| 1404 | | | | | | | |
| 1405 | | | | | | | |
| 1406 | | | | | | | |
| 1407 | | | | | | | |
| 1408 | 004242 | 017605 | 000000 | | .SETFLG: | MOV | @(SP),R5 |
| 1409 | 004246 | 042737 | 000040 | 006160 | | BIC | #40,INBUF |
| 1410 | 004254 | 122737 | 000116 | 006160 | | CMPB | #'N',INBUF ;IS IT 'N' ? |


```

1411 004262 001002      BNE      1$
1412 004264 105015      CLR      (R5)      ;000
1413 004266 000406      BR       2$
1414 004270 122737 000131 006160 1$:  CMP      #'Y',INBUF      ;IS IT 'Y' ?
1415 004276 001005      BNE      3$
1416 004300 112715 177777      MOV      #-1,(R5)      ;377
1417 004304 062716 000002      2$:  ADD      #2,(SP)
1418 004310 000002      RTI
1419 004312 104404      3$:  INSTER  ;RETRY
1420 004314 000752      BR       .SETFLG
1421
1422
1423      ;TRAP DISPATCH SERVICE
1424      ;ARGUMENT OF TRAP IS EXTRACTED
1425      ;AND USED AS OFFSET TO OBTAIN POINTER
1426      ;TO SELECTED SUBROUTINE
1427
1428 004316 011646      .TRPSR: MOV      (SP),-(SP)      ;GET PC OF RETURN
1429 004320 162716 000002      SUB      #2,(SP)      ;-PC OF TRAP
1430 004324 017616 000000      MOV      @ (SP), (SP)      ;GET TRP
1431 004330 006316      TRPOK: ASL      (SP)      ;MULTIPLY TRAP ARG BY 2
1432 004332 042716 177001      BIC      #177001,(SP)      ;CLEAR UNWANTED BITS
1433 004336 062716 001344      ADD      #.TRPTAB,(SP)      ;POINTER TO SUBROUTINE ADDRESS
1434 004342 017616 000000      MOV      @ (SP), (SP)      ;SUBROUTINE ADDRESS
1435 004346 000136      JMP      @ (SP)+      ;GO TO SUBROUTINE
1436
1437      ;ERROR HANDLER
1438      ;-----
1439
1440 004350 032777 010000 174624 .HLT:  BIT      #SW12,@SWR      ;BELL ON ERROR?
1441 004356 001406      BEQ      XBX      ;BR IF NO BELL
1442 004360 105777 174624      TSTB     @TPCSR      ;TTY READY.
1443 004364 100003      BPL      XBX      ;DON'T WAIT IF TTY NOT READY.
1444 004366 112777 000207 174616      MOV      #207,@TPDBR      ;PUSH A BELL AT THE TTY.
1445 004374 032777 020000 174600 XBX:  BIT      #SW13,@SWR      ;DELETE ERROR PRINT OUT?
1446 004402 001105      BNE      HALTS      ;BR IF NO PRINT OUT WANTED.
1447 004404 021637 001234      CMP      (SP),LSTERR      ;WAS THIS ERROR FOUND LAST TIME?
1448 004410 001404      BEQ      1$      ;BR IF YES
1449 004412 011637 001234      MOV      (SP),LSTERR      ;RECORD BEING HERE
1450 004416 105037 001341      CLR      ERRFLG      ;PREPARE HEADER
1451 004422 104406      1$:  SAVO5      ;SAVE ALL PROC REGISTERS
1452 004424 011605      MOV      (SP),R5      ;GET THE PC OF ERROR
1453 004426 162705 000002      SUB      #2,R5      ;GET ADDRESS OF TRAP CALL
1454 004432 011504      MOV      (R5),R4      ;GET HLT INSTRUCTION
1455 004434 006304      ASL      R4      ;MULT BY TWO
1456 004436 061504      ADD      (R5),R4      ;DOUBLE IT
1457 004440 006304      ASL      R4      ;MULT AGAIN
1458 004442 042704 177001      BIC      #177001,R4      ;CLEAR JUNK
1459 004446 062704 024274      ADD      #.ERRTAB,R4      ;GET POINTER
1460 004452 012437 004566      MOV      (R4)+,ERRMSG      ;GET ERROR MESSAGE
1461 004456 012437 004600      MOV      (R4)+,DATAHD      ;GET DATA HEADRER
1462 004462 011437 004612      MOV      (R4),DATABP      ;GET DATA TABLE
1463 004466 105737 001341      TSTB     ERRFLG      ;TYPE HEADREER
1464 004472 001403      BEQ      TYPMSG      ;BR IF YES
1465 004474 005737 004612      TST      DATABP      ;DOES DATA TABLE EXIST?
1466 004500 001040      BNE      TYPDAT      ;BR IF YES.
    
```

| | | | | | | | |
|------|--------|--------|--------|--------|---------------|---------------------|---------------------------------------|
| 1467 | 004502 | 104402 | 005136 | | TYPMSG: TYPE | ,MCRLF | |
| 1468 | 004506 | 104402 | 005136 | | TYPE | ,MCRLF | |
| 1469 | 004512 | 005737 | 001220 | | TST | LOCK | |
| 1470 | 004516 | 001402 | | | BEQ | 1\$ | |
| 1471 | 004520 | 104402 | 005411 | | TYPE | ,MASTEK | |
| 1472 | 004524 | 104402 | 005377 | | 1\$: TYPE | ,MTSTN | |
| 1473 | 004530 | 104411 | 005000 | | CONVRT | ,XTSTN | :SHOW IT |
| 1474 | 004534 | 104402 | 005466 | | TYPE | ,MERRPC | :TYPE PC. |
| 1475 | 004540 | 104411 | 004772 | | CONVRT | ,ERTABO | :SHOW IT |
| 1476 | 004544 | 104402 | 005136 | | TYPE | ,MCRLF | :GIVE A CR/LF |
| 1477 | 004550 | 112737 | 177777 | 001341 | MOV# | #-1,ERRFLG | :NO MORE HEADER UNLESS NO DATA TABLE. |
| 1478 | 004556 | 005737 | 004566 | | TST | ERRMSG | :IS THERE AN ERROR MESSAGE? |
| 1479 | 004562 | 001402 | | | BEQ | WRKO.FM | :BR IF NO. |
| 1480 | 004564 | 104402 | | | TYPE | | :TYPE |
| 1481 | 004566 | 000000 | | | ERRMSG: 0 | | : ERROR MESSAGE |
| 1482 | 004570 | | | | WRKO.FM: | | |
| 1483 | 004570 | 005737 | 004600 | | TST | DATAHD | :DATA HEADER? |
| 1484 | 004574 | 001402 | | | BEQ | TYPDAT | :BR IF NO |
| 1485 | 004576 | 104402 | | | TYPE | | :TYPE |
| 1486 | 004600 | 000000 | | | DATAHD: 0 | | : DATA HEADER |
| 1487 | 004602 | 005737 | 004612 | | TYPDAT: TST | DATABP | :DATA TABLE? |
| 1488 | 004606 | 001402 | | | BEQ | RESREG | :BR IF NO. |
| 1489 | 004610 | 104410 | | | CONVRT | | :SHOW |
| 1490 | 004612 | 000000 | | | DATABP: 0 | | : DATA TABLE |
| 1491 | 004614 | 104407 | | | RESREG: RES05 | | :RESTORE PROC REGISTERS |
| 1492 | 004616 | 022737 | 003104 | 000042 | HALTS: CMP | #SENDAD,@#42 | :IF ACT-11 AUTO MODE--HALT . |
| 1493 | 004624 | 001403 | | | BEQ | 1\$ | |
| 1494 | 004626 | 005777 | 174350 | | TST | @SWR | :HALT ON ERROR? |
| 1495 | 004632 | 100035 | | | BPL | EXITER | :BR IF NO HALT ON ERROR |
| 1496 | 004634 | 010046 | | | 1\$: PUSHRO | | :SAVE RO |
| 1497 | 004636 | 016600 | 000002 | | MOV | 2(SP),RO | :SHOW ERROR PC IN DATA LIGHTS |
| 1498 | 004642 | 013746 | 000004 | | MOV | 4,-(SP) | :SAVE OLD TRAP |
| 1499 | 004646 | 013746 | 000006 | | MOV | 6,-(SP) | |
| 1500 | 004652 | 012737 | 004710 | 000004 | MOV | #22\$,4 | :FORCE HALT IF TIME-OUT |
| 1501 | 004660 | 012737 | 000340 | 000006 | MOV | #340,6 | :WHEN REFERENCING TXCSR |
| 1502 | 004666 | 042777 | 014000 | 174516 | BIC | #SYSTST!MEXT,@TXCSR | |
| 1503 | 004674 | 000000 | | | HALT | | :HALT |
| 1504 | 004676 | 012637 | 000006 | | MOV | (SP)+,6 | :RESTORE TRAP |
| 1505 | 004702 | 012637 | 000004 | | MOV | (SP)+,4 | |
| 1506 | 004706 | 000406 | | | BR | 33\$ | |
| 1507 | 004710 | 000000 | | | 22\$: HALT | | :HALT |
| 1508 | 004712 | 022626 | | | CMP | (SP)+,(SP)+ | :POP STACK |
| 1509 | 004714 | 012637 | 000006 | | MOV | (SP)+,6 | :RESTORE TRAP |
| 1510 | 004720 | 012637 | 000004 | | MOV | (SP)+,4 | |
| 1511 | 004724 | 012600 | | | 33\$: POPRO | | :GET RO |
| 1512 | 004726 | 005237 | 001232 | | EXITER: INC | ERRCNT | :UPDATE ERROR COUNT |
| 1513 | 004732 | 032777 | 000400 | 174242 | BIT | #SW08,@SWR | :GOTO TOP OF TEST? |
| 1514 | 004740 | 001007 | | | BNE | 1\$ | :BR IF YES |
| 1515 | 004742 | 032777 | 002000 | 174232 | BIT | #SW10,@SWR | :GOTO NEXT TEST? |
| 1516 | 004750 | 001407 | | | BEQ | 2\$ | :BR IF NO |
| 1517 | 004752 | 013737 | 001216 | 001214 | MOV | NEXT,RETURN | :SET FOR NEXT TEST |
| 1518 | 004760 | 012706 | 001150 | | 1\$: MOV | #STACK,SP | :RESET SP |
| 1519 | 004764 | 000177 | 174224 | | JMP | @RETURN | :GOTO SPECIFIED TEST |
| 1520 | 004770 | 000002 | | | 2\$: RTI | | :RETURN |
| 1521 | 004772 | 000001 | | | ERTABO: 1 | | |
| 1522 | 004774 | 006 | 002 | | .BYTE | 6,2 | |

```

1523 004776 001266          SAVPC
1524 005000 000001          XTSTN: 1
1525 005002 003          002          .BYTE 3,2
1526 005004 001226          TSTNO
1527 005006 017600 000000  .PKCLK: MOV @ (SP),R0          ;GET THE # OF TICKS TO POKE
1528 005012 062716 000002  ADD #2,(SP)          ;POP OVER THE #
1529 005016
1530 005016 052777 020000 174366 1$: BIS #CLK,@TXCSR          ;POKE CLOCK UP
1531 005024 005300          DEC RO          ;ARE WE DONE?
1532 005026 001405          BEQ 2$          ;YES-GO TO 2$
1533 005030 042777 020000 174354  BIC #CLK,@TXCSR          ;POKE CLOCK DOWN
1534 005036 005300          DEC RO          ;ARE WE DONE?
1535 005040 001366          BNE 1$          ;NO-REPEAT
1536 005042 000002 2$: RTI          ;RETURN
1537
1538
1539          ;WAIT ROUTINE
1540 005044 000240  SMALL: NOP          ;STALL
1541 005046 000207          RTS PC          ;RETURN
1542
1543          ;POWER FAIL ROUTINE
1544
1545 005050 012737 005060 000024  .PFAIL: MOV #PWRUP,24          ;LOAD PFAIL VECTOR FOR POWER UP
1546 005056 000000          HALT          ;
1547 005060 000005  PWRUP: RESET          ;WAIT ITY TO COME UP
1548 005062 012706 001150          MOV #STACK,SP          ;REINIT STACK POINTER
1549 005066 012737 005050 000024  MOV #.PFAIL,24          ;LOAD PFAIL VECTOR FOR POWER DOWN
1550 005074 104402          TYPE
1551 005076 005141          MPOWER
1552 005100 000177 174110          JMP @RETURN
1553          ;CLRVEC,ROUTINE TO FILL COMMUNICATION VECTOR AREA WITH .+2,HALT
1554
1555 005104 012702 000300  CLRVEC: MOV #300,R2          ;R2 COMM VECTOR AREA ADRS
1556 005110 012701 000302  MOV #302,R1          ;INIT R1 WITH ADRS OF HALT
1557 005114 010122 1$: MOV R1,(R2)+          ;MOV .+2 TO PC
1558 005116 005022          CLR (R2)+          ;MOV HALT TO PC
1559 005120 022121          CMP (R1)+,(R1)+          ;INC TO NEXT VECTOR AREA
1560 005122 022701 000776  CMP #776,R1          ;END OF VECTOR AREA
1561 005126 001372          BNE 1$          ;NO
1562 005130 000207          RTS PC          ;RETURN
1563
1564
1565
1566 005132 020040 000077  MQM: .ASCIZ / ?/
(2) 005136 005015 000  MCRLF: .ASCIZ <15><12>
(2) 005141 377 053520 020122  MPOWER: .ASCIZ <377>/PWR FAILED. /
(2) 005157 015 042777 042116  MEPASS: .ASCIZ <15><377>/END PASS CZDPCC /
(2) 005202 051377 000  MR: .ASCIZ <377>/R/
(2) 005205 377 051120 043517  MERR2: .ASCIZ <377>/PROGRAM INDICATES NO DEVICES PRESENT./
(2) 005254 044777 051516 043125  MERR3: .ASCIZ <377>/INSUFFICIENT DATA!/
(2) 005300 052377 051505 020124  MTSTPC: .ASCIZ <377>/TEST PC-/
(2) 005312 046377 041517 020113  MLOCK: .ASCIZ <377>/LOCK ON SELECTED TEST/
(2) 005341 103 051123 020072  MCSRX: .ASCIZ /CSR: /
(2) 005347 126 041505 020072  MVECX: .ASCIZ /VEC: /
(2) 005355 120 051501 042523  MPASSX: .ASCIZ /PASSES: /
(2) 005366 051105 047522 051522  MERRX: .ASCIZ /ERRORS: /

```

```
(2) 005377 124 051505 020124 MTSTN: .ASCIZ /TEST NO: /
(2) 005411 052 000 MASTEK: .ASCIZ /*/
(2) 005413 377 042523 020124 MNEW: .ASCIZ <377>/SET SWITCH REG TO DUP11'S DESIRED ACTIVE./
(2) 005466 041520 020072 000 MERRPC: .ASCIZ /PC: /
(2) 005473 377 042522 020103 MCSR: .ASCIZ <377>/REC CSR ADRS /
(2) 005512 053377 041505 040440 MVEC: .ASCIZ <377>/VEC ADRS /
(2) 005525 377 051511 052040 MJMPR: .ASCIZ <377>/IS THE OPTIONAL CLR JMPR IN? (Y OR N) /
(2) 005600 044777 020123 044124 MTCN: .ASCIZ <377>/IS THE H325 CONNECTOR ON? (Y OR N) /
(2) 005647 377 020043 043117 MTOTAL: .ASCIZ <377>/# OF DUP'S (IN OCTAL) /
(2) 005702 050377 044522 051117 MPAR: .ASCIZ <377>/PRIORITY (4 TO 7) /
(2) 005726 051777 041505 052040 MSTJM: .ASCIZ <377>/SEC TX JMPR IN? (Y OR N) /
(2) 005761 377 042523 020103 MSRJM: .ASCIZ <377>/SEC RX JMPR IN? (Y OR N) /
(2) 006014 046777 050101 047440 XHEAD: .ASCIZ <377>/MAP OF DUP11 STATUS/<377>
(2) .EVEN
(2) 006042 000002 XSTATQ: 2
1567 006044 006 003 .BYTE 6,3
1568 006046 001236 TEMP1
1569 006050 006 002 .BYTE 6,2
1570 006052 001240 TEMP2
1571 .EVEN
1572
1573 006054 000000 TEMP: 0
1574 006116 .-.+40
1575 006116 000000 MDATA: 0
1576 006160 006160 .=.+40
1577 006160 000000 INBUF: 0
1578 006222 006222 .+.40
1579 006222 000001 TRP.PC: .BLKW 1
1580
```

```

1581
1582
1583          :ROUTINE USED TO "CYCLE" THROUGH UP TO EIGHT DUP11'S
1584          :THIS ROUTINE SETS UP THE CONTROL ADDRESS FOR THE DIAGNOSTIC
1585          :AND RUNS THE SPECIFIED DUP11'S.  THIS ROUTINE *MUST*
1586          :BE RUN FIRST BEFORE ENTERING THE DIAGNOSTIC FOR THE
1587          :SETUP NECESSARY.
1588          :
1589
1590 006224 105737 001310          CYCLE: TSTB   DUPACTV   ;ARE ANY DUP11'S TO BE TESTED?
1591 006230 001004                BNE     1$      ;BR IF OK.
1592 006232 104402 005205        TYPE    ,MERR2    ;NO DUP11'S SELECTED!!
1593 006236 000000                HALT                    ;STOP THE SHOW.
1594 006240 000776                BR      -2      ;DISQUALIFY CONT. SW.
1595 006242 133737 001314 001310 1$: BITB   RUN,DUPACTV ;IS THIS ONE 'ACTIVE'
1596 006250 001020                BNE     2$      ;BR IF GOOD ONE FOUND.
1597 006252 000241                CLC                      ;CLEAR PROC. CARRY BIT.
1598 006254 106137 001314        ROLB   RUN          ;UPDATE POINTER
1599 006260 105537 001314        ADCB   RUN          ;CATCH CARRY FROM RUN
1600 006254 062737 000006 001316  ADD    #6,CREAM ;UPDATE ADDRESS POINTER.
1601 006272 022737 001560 001316  CMP    #DUP.END,CREAM
1602 006300 001360                BNE     1$      ;KEEP GOING; NOT ALL TESTED FOR.
1603 006302 012737 001500 001316  MOV    #DUP.MAP,CREAM ;RESET ADDRESS POINTER.
1604 006310 000754                BR      1$      ;KEEP LOOKING FOR ACTIVE DUP11
1605 006312 000241                2$:  CLC                      ;CLEAR PROC. CARRY.
1606 006314 106137 001314        ROLB   RUN          ;UPDATE POINTER.
1607 006320 105537 001314        ADCB   RUN          ;CATCH CARRY.
1608 006324 013700 001316        MOV    CREAM,RO    ;GET ADDRESS POINTER.
1609 006330 062737 000006 001316  ADD    #6,CREAM ;UPDATE.
1610 006336 022737 001560 001316  CMP    #DUP.END,CREAM
1611
1612 006344 001003                BNE     3$      ;ALL DONE?
1613 006346 012737 001500 001316  MOV    #DUP.MAP,CREAM ;BR IF NO.
1614 006354 012037 001404                3$: MOV    (RO)+,RXCSR ;RESTORE POINTER.
1615 006360 012037 001374                MOV    (RO)+,DUPRVC ;LOAD SYSTEM CTRL. REG
1616 006364 012037 001434                MOV    (RO)+,LOO.00 ;LOAD VECTOR
1617 006370 012700 000002                MOV    #2,RO      ;GET PARAMETERS
1618 006374 013737 001404 001424        MOV    RXCSR,HUPRCR ;SAVE CORE THIS WAY!
1619 006402 005237 001424                INC    HUPRCR     ;GET CONTROL REG HIGH BYTE
1620 006406 013737 001424 001406        MOV    HUPRCR,RXDBUF ;GOT IT
1621 006414 005237 001406                INC    RXDBUF     ;GET RX CONTROL REG BUFFER
1622 006420 013737 001406 001416        MOV    RXDBUF,DUPSEC ;GOT IT
1623 006426 013737 001406 001410        MOV    RXDBUF,PARCSR ;GOT SECONDARY REG; SELECT REG
1624 006434 013737 001406 001422        MOV    RXDBUF,HUPRBF ;GOT PARAMETER STATUS REGISTER
1625 006442 005237 001422                INC    HUPRBF     ;GET RX BUFFER HIGH BYTE
1626 006446 013737 001422 001420        MOV    HUPRBF,HUPPSR ;GOT IT
1627 006454 013737 001420 001412        MOV    HUPPSR,TXCSR ;GOT PAR STATUS REG HIGH BYTE
1628 006462 005237 001412                INC    TXCSR      ;GET TX CONTROL REGISTER
1629 006466 013737 001412 001430        MOV    TXCSR,HUPTCR ;GOT IT
1630 006474 005237 001430                INC    HUPTCR     ;GET TX CONTROL REG HIGH BYTE
1631 006500 013737 001430 001414        MOV    HUPTCR,TXDBUF ;GOT IT
1632 006506 005237 001414                INC    TXDBUF     ;BET TX BUFFER
1633 006512 013737 001414 001426        MOV    TXDBUF,HUPTBF ;GOT IT
1634 006520 005237 001426                INC    HUPTBF     ;GET TX BUFFER HIGH BYTE
1635
1636 006524 013737 001374 001376        MOV    DUPRVC,DUPRPS ;GOT IT
          ;RX VECTOR

```

```

1637 006532 060037 001376          ADD    R0,DUPRPS      ;RX PRIORITY LEVEL
1638 006536 013737 001376 001400    MOV    DUPRPS,DUPTVC
1639 006544 060037 001400          ADD    R0,DUPTVC     ;TX VECTOR
1640 006550 013737 001400 001402    MOV    DUPTVC,DUPTPS
1641 006556 060037 001402          ADD    R0,DUPTPS     ;TX PRIORITY LEVEL
1642
1643
1644 006562 012700 001434          MOV    #L00.00,R0    ;LOAD STAUS 00-00
1645 006566 012701 001432          MOV    #MASK.A,R1   ;PREPARE MASK.
1646 006572 012702 001433          MOV    #CLK.A,R2    ;PREPARE CLOCKS
1647 006576 004737 006742          JSR    PC,FIX.00    ;GO AND CALCULATE CONFIGURATION.
1648 006602 005737 000042          TST    @#42
1649 006606 001050          BNE    4$
1650 006610 032777 000002 172364    BIT    #SW01,@SWR   ;IF SW01=1,GET STARTING TEST #
1651 006616 001444          BEQ    4$
1652 006620 104402 005136          7$:   TYPE    ,MCRLF
1653 006624 104403          INSTR  ;OUTPUT MESSAGE & GET INPUT STRING
1654 006626 005377          MTSTN  ;MESSAGE
1655 006630 104405          PARAM  ;CONVERT STRING
1656 006632 000001          1      ;LOW LIMIT
1657 006634 001000          1000   ;HIGH LIMIT
1658 006636 001226          TSTNO  ;STORE AT THIS LOCATION
1659 006640          000    .BYTE 0 ;MASK
1660 006641          001    .BYTE 1 ;HOW MANY TIMES + 2
1661 006642 012700 007356          MOV    #TST1,R0
1662 006646 022710 012737          5$:   CMP    #12737,(R0)
1663 006652 001017          BNE    6$
1664 006654 023760 001226 000002    CMP    TSTNO,2(R0)
1665 006662 001013          BNE    6$
1666 006664 022760 001226 000004    CMP    #TSTNO,4(R0)
1667 006672 001007          BNE    6$
1668 006674 010037 001214          MOV    R0,RETURN    ;SAVE PC
1669 006700 104402 005136          TYPE  ,MCRLF
1670 006704 104402 005202          TYPE  ,MR
1671 006710 000412          BR     8$
1672 006712 005720          6$:   TST    (R0)+
1673 006714 020027 022606          CMP    R0,#TLAST+10
1674 006720 001352          BNE    5$
1675 006722 104402 005132          TYPE  ,MQM
1676 006726 000734          BR     7$
1677
1678 006730 012737 007356 001214 4$:   MOV    #TST1,RETURN  ;PREPARE RETURN ADDRESS
1679 006736 000177 172252 8$:   JMP    @RETURN      ;GO START TESTING.
1680
1681 006742 011003          FIX.00: MOV    (R0),R3    ;GET PARAMETERS.
1682 006744 000207          5$:   RTS    PC        ;

```

```

1683
1684
1685
1686
1687                                     ;ROUTINE TO DETERMINE IF BIT STUFFING IS REQUIRED
1688                                     -----
1689
1690 006746 012737 000010 001326 STUFCK: MOV      #8.,SHIFTS
1691 006754 013700 001324          MOV      DATA,R0
1692 006760 012701 000200          MOV      #200,R1
1693 006764 006000          1$:    ROR      R0
1694 006766 005537 001330          ADC      MIND
1695 006772 022737 000005 001330    CMP      #5,MIND
1696 007000 003004          BGT      2$
1697 007002 005037 001330          CLR      MIND
1698 007006 005237 001326          INC      SHIFTS
1699 007012 006001          2$:    ROR      R1
1700 007014 103363          BCC      1$
1701 007016 000207          RTS      PC
1702
1703
1704                                     ;THIS ROUTINE POKES THE RECEIVER BIT WINDOW
1705                                     ;BASED ON THE INFORMATION IN DATA AND SHIFTS
1706                                     -----
1707
1708 007020 042777 002000 172364 RPOKE: BIC      #BITW,@TXCSR
1709 007026 006037 001324          ROR      DATA
1710 007032 103003          BCC      1$
1711 007034 052777 002000 172350    BIS      #BITW,@TXCSR
1712 007042 000240          1$:    NOP
1713 007044 104412 000002          PKCLK   ,2
1714 007050 005337 001326          DEC      SHIFTS
1715 007054 001361          BNE      RPOKE
1716 007056 000201          RTS      R1
1717
1718
1719 007060 012137 001244          RFLG:  MOV      (R1)+,TEMP4      ;GET THE # OF FLAGS
1720 007064 005037 001246          1$:    CLR      TEMP5              ;CLEAR ONES COUNT
1721 007070 042777 002000 172314    BIC      #BITW,@TXCSR          ;SET FIRST BIT
1722 007076 104412 000002          PKCLK   ,2                    ;PUSH OUT THE BIT
1723 007102 052777 002000 172302    2$:    BIS      #BITW,@TXCSR          ;LOAD THE BIT
1724 007110 104412 000002          PKCLK   ,2                    ;PUSH OUT THE BIT
1725 007114 005237 001246          INC      TEMP5                ;INCREMENT 1'S COUNTER
1726 007120 022737 000006 001246    CMP      #6,TEMP5              ;DID WE PUSH OUT 6 ONES
1727 007126 001365          BNE      2$                    ;NO-GO BACK
1728 007130 042777 002000 172254    BIC      #BITW,@TXCSR          ;SET THE LAST BIT
1729 007136 104412 000002          PKCLK   ,2                    ;PUSH OUT THE BIT
1730 007142 005337 001244          DEC      TEMP4                ;ARE WE DONE WITH FLAGS?
1731 007146 001346          BNE      1$                    ;BR IF NO
1732 007150 000201          RTS      R1
1733
1734
1735 007152 012137 001244          SFLG:  MOV      (R1)+,TEMP4      ;GET THE # OF FLAGS
1736 007156 005037 001246          1$:    CLR      TEMP5              ;CLEAR ONES COUNT
1737 007162 052777 002000 172222    2$:    BIS      #BITW,@TXCSR          ;LOAD THE BIT
1738 007170 104412 000002          PKCLK   ,2                    ;PUSH OUT THE BIT

```

```

1739 007174 005237 001246      INC      TEMP5      ;INCREMENT 1'S COUNTER
1740 007200 022737 000006 001246  CMP      #6,TEMP5   ;DID WE PUSH OUT 6 ONES
1741 007206 001365          BNE      2$         ;NO-GO BACK
1742 007210 042777 002000 172174  BIC      #BITW,@XCSR ;SET THE LAST BIT
1743 007216 104412 000002          PKCLK     .2        ;PUSH OUT THE BIT
1744 007222 005337 001244          DEC      TEMP4     ;ARE WE DONE WITH FLAGS?
1745 007226 001353          BNE      1$         ;BR IF NO
1746 007230 000201          RTS      R1
1747
1748 007232 012577 172136      SETVEC: MOV      (R5)+,@DUPRVC
1749 007236 012577 172136          MOV      (R5)+,@DUPTVC
1750 007242 112577 172130          MOVB     (R5)+,@DUPRPS
1751 007246 112577 172130          MOVB     (R5)+,@DUPTPS
1752 007252 000205          RTS      R5
1753 007254          NO.ATRAP:
1754 007254 104025          HLT      25
1755 007256 000002          RTI
1756
1757 007260          NO.BTRAP:
1758 007260 104026          HLT      26
1759 007262 000002          RTI
1760
1761          ;THIS ROUTINE PICKS UP THE ADDRESS OF
1762          ;THE JUMPER TABLE AND LOADS R1 WITH
1763          ;THE CORRECT DATA BASED ON THE STATE
1764          ;OF THE JUMPER AND CONNECTOR FLAGS.
1765          ;-----
1766
1767 007264 012100          JUMPER: MOV      (R1)+,R0      ;GET THE TABLE ADDRESS
1768 007266 105737 001322          TSTB     TCNFLG          ;TEST THE TURN AROUND CONNECTOR FLAG
1769 007272 001406          BEQ      2$             ;BRANCH IF CONNECTOR IS MISSING
1770 007274 105737 001323          TSTB     OPCLRJ          ;TEST CLEAR JUMPER FLAG
1771 007300 001403          BEQ      2$             ;BRANCH IF JUMPER IS MISSING
1772 007302 011005          MOV      (R0),R5        ;MOVE THE DATA TO R5, BOTH JUMPER
1773          ;AND CONNECTOR ARE THERE
1774 007304 000137 007330          JMP      5$
1775 007310 022020          2$:  CMP      (R0)+,(R0)+    ;POP POINTER
1776 007312 105737 001323          TSTB     OPCLRJ          ;TEST CLEAR JUMPER FLAG
1777 007316 001403          BEQ      3$             ;BRANCH IF MISSING
1778 007320 011005          MOV      (R0),R5        ;MOVE DATA- JUMPER IN, CONNECTOR OFF
1779 007322 000137 007330          JMP      5$
1780 007326 012005          3$:  MOV      (R0)+,R5        ;NO CONNECTOR OR JUMPER
1781 007330 000201          5$:  RTS      R1             ;RETURN
1782
1783 007332 012100          OJUMPER:MOV      (R1)+,R0    ;GET THE POINTER ADDRESS
1784 007334 105737 001322          TSTB     TCNFLG          ;CHECK FOR TURNAROUND CONNECTOR
1785 007340 001403          BEQ      4$             ;BR IF MISSING
1786 007342 011005          MOV      (R0),R5        ;MOVE THE INFO TO R5
1787 007344 000137 007354          JMP      6$             ;GO BACK
1788 007350 022020          4$:  CMP      (R0)+,(R0)+    ;POP POINTER
1789 007352 011005          MOV      (R0),R5        ;LOAD DATA TO R5
1790 007354 000201          6$:  RTS      R1             ;RETURN
1791
1792          ;***** TEST 1 *****
1793          ;*TEST TO PROVE THE INTERACTION OF RECEIVER ENABLE
1794          ;*WITH REC 'VER ACTIVE AND RECEIVE START

```


1795
1796
1797
1798
1799
1800
1801
1802
1803
1804
1805
1806
1807
1808
1809
1810
1811
1812
1813
1814
1815
1816
1817
1818
1819
1820
1821
1822
1823
1824
1825
1826
1827
1828
1829
1830
1831
1832
1833
1834
1835
1836
1837
1838
1839
1840
1841
1842
1843
1844
1845
1846
1847
1848
1849
1850

007356 012737 000001 001226
007364 012737 007640 001216
007372 052777 000400 172012
007400 004737 005044
007404 013703 001404
007410 052777 001000 171772
007416 052713 000020
007422 052777 014000 171762
007430 004137 007060
007434 000001
007436 032713 004000

007442 001401
007444 104006
007446 005037 001324
007452 012737 000011 001326
007460 004137 007020
007464 032713 000200
007470 001004
007472 012705 000200
007476 011304
007500 104003
007502 032777 000400 171676
007510 001006
007512 012705 000400
007516 013703 001406
007522 011304
007524 104003

007526 013703 001404
007532 032713 004000
007536 001001
007540 104014
007542 005037 001324
007546 012737 000011 001326
007554 004137 007020
007560 032713 000200
007564 001004
007566 012705 000200
007572 011304
007574 104003
007576 032777 000400 171602
007604 001406
007606 012705 000400
007612 013703 001406
007616 011304

```

;*OF MESSAGE IN PRIMARY MODE.
:*****
:
: TEST 1
:
:*****
:*****
TST1:  MOV #1,@TSTNO
      MOV #TST2,NEXT
      BIS #MRESET,@TXCSR ;RESET THE DEVICE
      JSR PC,SMALL ;WAIT FOR RESET TO FINISH
      MOV RXCSR,R3 ;GET THE RECEIVER CONTROL REGISTER
      BIS #CRCEN,@PARCSR ;TURN OFF CRC
      BIS #RCVEN,(R3) ;TURN ON THE RECEIVER
      BIS #MMODE,@TXCSR ;ENTER M/MODE
      JSR R1,RFLG ;PUSH OUT A FLAG
      BIT #REACT,(R3) ;RXACT SHOULD BE CLEAR WHEN READING FLAG CHAR.
      BEQ 1$ ;BRANCH IF BIT IS OFF
      YLT 6 ;BIT WAS SET - SHOULD BE CLEAR
1$:   CLR DATA ;SETUP FOR CHAR
      MOV #9,,SHIFTS ;POKE CLOCK SETUP
      JSR R1,RPOKE ;PUSH DATA INTO RECEIVER
      BIT #RXDONE,(R3) ;TEST DONE
      BNE 2$ ;BR IF SET
      MOV #RXDONE,R5 ;ERROR MSG SETUP
      MOV (R3),R4 ;SET UP FOUND
      HLT 3 ;RXDONE FAILED TO SET
2$:   BIT #RSOM,@RXDBUF ;RSOM SHOULD BE SET
      BNE 3$ ;BR IF SET
      MOV #RSOM,R5 ;SET EXPECTED
      MOV RXDBUF,R3 ;SET UP ERROR MSG
      MOV (R3),R4 ;SET FOUND
      HLT 3 ;RSOM FAILED TO SET
://////;
3$:   MOV RXCSR,R3
      BIT #REACT,(R3) ;SHOULD BE SET FOR START OF FRAME
      BNE 4$ ;BR IF SET
      HLT 14 ;FAILED TO SET
4$:   CLR DATA ;SETUP FOR CHAR
      MOV #9,,SHIFTS ;POKE CLOCK SETUP
      JSR R1,RPOKE ;PUSH DATA INTO RECEIVER
      BIT #RXDONE,(R3) ;RXDONE SHOULD BE SET
      BNE 5$ ;BR IF SET
      MOV #RXDONE,R5 ;ERROR MSG SETUP
      MOV (R3),R4 ;SET UP FOUND
      HLT 3 ;RXDONE FAILED TO SET
5$:   BIT #RSOM,@RXDBUF ;RSOM SHOULD BE CLEAR
      BEQ 10$ ;THIS IS SECOND DATA CHAR OF FRAME
      MOV #RSOM,R5 ;SET EXPECTED
      MOV RXDBUF,R3 ;SET UP ERROR MSG
      MOV (R3),R4 ;SET FOUND

```

TEST OF REC ENABLE,RXACT,AND RSOM IN PRIMARY MODE

```
1851 007620 104032          HLT      32          :RSOM FAILED TO CLEAR
1852  ;////////////////////////////////////
1853
1854 007622 013703 001404 10$:  MOV     RXCSR,R3      :SHOULD BE SET FOR START OF FRAME
1855 007626 032713 004000      BIT     #REACT,(R3)   :BR IF SET
1856 007632 001001              BNE     11$          :FAILED TO SET
1857 007634 104014              HLT     14
1858 007636 104400 11$:  SCOPE
1859
1860 ;***** TEST 2 *****
1861 ;*TEST TO PROVE THE INTERACTION OF RECEIVER ENABLE
1862 ;*WITH RECEIVER ACTIVE AND RECEIVE
1863 ;*START OF MESSAGE IN SECONDARY MODE.
1864 ;*****
1865
1866 ;:*****
1867 ;:
1868 ;: TEST 2
1869 ;:
1870 ;:*****
1871 ;:*****
1872 007640 012737 000002 001226 TST2: MOV     #2,@TSTNO
1873 007646 012737 010204 001216      MOV     #TST3,NEXT
1874 007654 052777 000400 171530      BIS     #MRESET,@TXCSR :RESET THE DEVICE
1875 007662 004737 005044              JSR     PC,SMALL      :WAIT FOR RESET TO FINISH
1876 007666 013703 001404              MOV     RXCSR,R3      :GET THE RECEIVER CONTROL REGISTER
1877 007672 052777 001000 171510      BIS     #RCEN,@PARCSR :TURN OFF CRC
1878 007700 052713 000020              BIS     #RCVEN,(R3)   :TURN ON THE RECEIVER
1879 007704 052777 010000 171476 BIS     #PRISEC,@PARCSR :ENTER SECONDARY MODE
1880 007712 052777 014000 171472      BIS     #MMODE,@TXCSR :ENTER M/MODE
1881 007720 004137 007060              JSR     R1,RFLG       :PUSH OUT A FLAG
1882 007724 000001
1883 007726 032713 004000      BIT     #REACT,(R3)   :RXACT SHOULD BE CLEAR WHEN READING FLAG CHAR.
1884
1885 007732 001403              BEQ     3$           :BR IF OFF
1886 007734 005005              CLR     R5           :SET EXPECTED
1887 007736 011304              MOV     (R3),R4      :SET FOUND
1888 007740 104003              HLT     3           :BIT WAS SET AND SHOULD BE CLEARED
1889 ;WE ARE IN SECONDARY MODE
1890 007742 032713 000200 3$:  BIT     #RXDONE,(R3)  :SHOULD BE CLEAR
1891 007746 001401              BEQ     4$           :BR IF CLEAR
1892 007750 104003              HLT     3           :DONE IS SET - SHOULD NOT BE BECAUSE
1893 ;WE ARE IN SECONDARY MODE AND HAVE NOT PUSHED DATA
1894 007752 005037 001324 001326 4$:  CLR     DATA        :CLEAR DATA CHAR
1895 007756 012737 000011              MOV     #9,SHIFTS    :LOAD THE # OF BITS TO PUSH
1896 007764 004137 007020              JSR     R1,RPOKE     :PUSH DATA TO RECEIVER
1897 007770 032713 004000      BIT     #REACT,(R3)  :RXACT SHOULD BE SET
1898 007774 001004              BNE     5$           :BR IF SET
1899 007776 012705 004000      MOV     #REACT,R5    :SET EXPECTED
1900 010002 011304              MOV     (R3),R4      :SET FOUND
1901 010004 104003              HLT     3           :ACTIVE FAILED TO SET
1902 010006 032713 000200 5$:  BIT     #RXDONE,(R3)  :RXDONE SHOULD BE CLEAR
1903 010012 001404              BEQ     6$           :BR IF CLEAR - THE CHAR WAS
1904 ;THE SECONDARY ADRS
1905 010014 012705 000200      MOV     #RXDONE,R5   :SET EXPECTED
1906 010020 011304              MOV     (R3),R4      :SET FOUND
```

TEST OF REC ENABLE,RXACT,AND RSOM IN SECONDARY MODE

```

1907 010022 104003          HLT      3          ;DONE WAS SET AND SHOULDN'T BE - THE FIRST
1908                                ;DATA WAS SECONDARY ADRS, NOT DATA TO THE BUFFER
1909 010024 032777 000400 171354 6$:  BIT      #RSOM,@RXDBUF ;CHECK START OF MSG
1910 010032 001401          BEQ      7$          ;BR IF CLEAR
1911 010034 104020          HLT      20          ;START OF MSG SHOULD BE CLEAR
1912 010036 112737 000252 001324 7$:  MOVB    #252,DATA ;LOAD DATA CHAR
1913 010044 012737 000010 001326      MOV     #8.,SHIFTS ;LOAD CLOCK COUNT
1914 010052 004137 007020      JSR     R1,RPOKE   ;PUSH OUT DATA
1915 010056 104412 000002      PKCLK   ,2         ;POKE ANOTHER FULL CLOCK
1916 010062 032713 000200      BIT     #RXDONE,(R3);TEST DONE
1917 010066 001004          BNE     12$         ;BR IF SET
1918 010070 012705 000200      MOV     #RXDONE,R5;SET EXPECTED
1919 010074 011304          MOV     (R3),R4    ;SET FOUND
1920 010076 104003          HLT      3          ;DONE FAILED TO SET AFTER PUSHING
1921                                ;OUT A CHAR IN SEC. MODE
1922                                ;//////////////////////////////////////;:++C
1923 010100 032777 000400 171300 12$:  BIT     #RSOM,@RXDBUF ;RSOM SHOULD BE SET
1924 010106 001001          BNE     14$         ;BR IF SET
1925 010110 104021          HLT      21          ;RSOM CLEAR
1926 010112 112737 000252 001324 14$:  MOVB    #252,DATA ;LOAD DATA CHAR
1927 010120 012737 000010 001326      MOV     #8.,SHIFTS ;LOAD CLOCK COUNT
1928 010126 004137 007020      JSR     R1,RPOKE   ;PUSH OUT DATA
1929 010132 104412 000002      PKCLK   ,2         ;POKE ANOTHER FULL CLOCK
1930 010136 032713 000200      BIT     #RXDONE,(R3);TEST DONE
1931 010142 001004          BNE     15$         ;BR IF SET
1932 010144 012705 000200      MOV     #RXDONE,R5;SET EXPECTED
1933 010150 011304          MOV     (R3),R4    ;SET FOUND
1934 010152 104003          HLT      3          ;DONE FAILED TO SET
1935 010154 032777 000400 171224 15$:  BIT     #RSOM,@RXDBUF ;RSOM SHOULD BE CLEAR
1936 010162 001401          BEQ     10$         ;BR IF CLEAR
1937 010164 104032          HLT     32         ;FAILED TO CLEAR
1938                                ;//////////////////////////////////////
1939 010166 013703 001404 10$:  MOV     RXCSR,R3   ;SHOULD BE SET FOR START OF FRAME
1940 010172 032713 004000      BIT     #REACT,(R3);BR IF SET
1941 010176 001001          BNE     11$         ;FAILED TO SET
1942 010200 104014          HLT     14
1943 010202 104400 11$:  SCOPE
1944
1945                                ;***** TEST 3 *****
1946                                ;*TEST TO PROVE THE INTERACTION OF REOM
1947                                ;*WITH DONE IN PRIMARY MODE
1948
1949                                ;*****
1950                                ;*****
1951                                ;*
1952                                ;* TEST 3
1953                                ;*
1954                                ;*****
1955                                ;*****
1956 010204 012737 000003 001226 TST3: MOV     #3,@TSTNO
1957 010212 012737 010472 001216      MOV     #TST4,NEXT
1958 010220 052777 000400 171164      BIS     #MRESET,@TXCSR ;RESET THE DEVICE
1959 010226 004737 005044          JSR     FC,SMALL   ;WAIT FOR RESET TO FINISH
1960 010232 052777 000020 171144      BIS     #RCVEN,@RXCSR ;TURN ON RECEIVER
1961 010240 052777 001000 171142      BIS     #CRCEN,@PARCSR ;TURN OFF CRC
1962 010246 005037 001236          CLR     TEMP1     ;BIT COUNTER

```

TEST OF REOM AND DONE IN PRIMARY MODE

```
1963 010252 052777 014000 171132 BIS #MMODE,@TXCSR ;ENTER MAINT MODE
1964 010260 004137 007060 JSR R1,RFLG ;PUSH OUT A FLAG
1965 010264 000001 1 JSR R1,RFLG ;ONE FLAG
1966 010266 012737 000125 001324 MOV #125,DATA ;CLEAR OUT CHAR
1967 010274 012737 000010 001326 MOV #8.,SHIFTS ;LOAD THE # OF CLOCKS
1968 010302 004137 007020 JSR R1,RPOKE ;PUSH OUT THE DATA
1969 010306 042777 002000 171076 BIC #BITW,@TXCSR ;CLEAR THE DATA WINDOW
1970 010314 104412 000002 PKCLK .2 ;PUSH OUT THE BIT
1971 010320 032777 000200 171056 BIT #RXDONE,@RXCSR ;CHECK FOR DONE
1972 010326 001001 BNE 5$ ;BR IF SET
1973 010330 104000 HLT ;DONE BIT ERROR
1974 010332 105777 171050 5$: TSTB @RXDBUF ;READ THE BUFFER
1975 010336 052777 002000 171046 64$: BIS #BITW,@TXCSR ;PUT A 1 IN THE WINDOW
1976 010344 104412 000002 PKCLK .2 ;PUSH IT OUT
1977 010350 005237 001236 INC TEMP1 ;INC THE # TO DO
1978 010354 022737 000006 001236 CMP #6,TEMP1 ;CHECK FOR FINISH
1979 010362 001365 BNE 64$ ;BR IF MORE TO GO
1980 010364 042777 002000 171020 BIC #BITW,@TXCSR ;CLEAR THE WINDOW
1981 010372 104412 000004 PKCLK, 4 ;PUSH OUT 2 BINTS
1982 010376 017737 171002 001240 MOV @RXCSR,TEMP2 ;GET THE CSR
1983 010404 032777 001000 170774 BIT #REOM,@RXDBUF ;TEST FOR END OF MSG
1984 010412 001001 BNE 1$ ;BR IF SET
1985 010414 104001 HLT 1 ;BIT FAILED TO SET
1986 010416 032737 000200 001240 1$: BIT #RXDONE,TEMP2 ;TEST DONE
1987 010424 001001 BNE 10$ ;BR IF SET
1988 010426 104002 HLT 2 ;DONE FAILED TO SET
1989 ;//////////////////////;:++C
1990 010430 004137 007060 10$: JSR R1,RFLG ;OUTPUT A FLAG CHAR
1991 010434 000001 1 MOV @RXCSR,TEMP2 ;GET CSR
1992 010436 017737 170742 001240 BIT #REOM,@RXDBUF ;REOM SHOULD BE CLEAR
1993 010444 032777 001000 170734 BEQ 11$ ;BR IF CLEAR
1994 010452 001401 HLT 30 ;REOM FAILED TO CLEAR
1995 010454 104030 000200 001240 11$: BIT #RXDONE,TEMP2 ;RXDONE SHOULD BE SET
1996 010456 032737 000200 001240 BNE 4$ ;BR IF SET
1997 010464 001001 HLT 2 ;FAILED TO SET
1998 010466 104002 ;//////////////////////
1999 4$: SCOPE ;SCOPE THIS TEST
2000 010470 104400
2001
2002 ;***** TEST 4 *****
2003 ;*TEST TO PROVE THE INTERACTION OF REOM
2004 ;*WITH DONE IN PRIMARY MODE
2005 ;*USING A COMMON ZERO BIT IN FLAGS.
2006
2007 ;:*****
2008 ;:*****
2009 ;: *
2010 ;: TEST 4
2011 ;: *
2012 ;:*****
2013 ;:*****
2014 010472 012737 000004 001226 TST4: MOV #4,@TSTNO
2015 010500 012737 011002 001216 MOV #TST5,NEXT
2016 010506 052777 000400 170676 BIS #MRESET,@TXCSR ;RESET THE DEVICE
2017 010514 004737 005044 JSR PC,SMALL ;WAIT FOR RESET TO FINISH
2018 010520 052777 000020 170656 BIS #RCVEN,@RXCSR ;TURN ON RECEIVER
```

```
2019 010526 052777 001000 170654      BIS      #CRCEN,@PARCSR  ;TURN OFF CRC
2020 010534 005037 001236      CLR      TEMP1        ;BIT COUNTER
2021 010540 052777 014000 170644      BIS      #MMODE,@TXCSR ;ENTER MAINT MODE
2022 010546 004137 007060      JSR      R1,RFLG      ;PUSH OUT A FLAG
2023 010552 000001      1        ;ONE FLAG
2024 010554 004137 007152      JSR      R1,SFLG      ;PUSH OUT SPECIAL FLAGS
2025 010560 000004      4        ;THE # TO DO
2026 010562 012737 000125 001324  MOV      #125,DATA     ;CLEAR OUT CHAR
2027 010570 012737 000010 001326  MOV      #8.,SHIFTS   ;LOAD THE # OF CLOCKS
2028 010576 004137 007020      JSR      R1,RPOKE     ;PUSH OUT THE DATA
2029 010602 042777 002000 170602      BIC      #BITW,@TXCSR ;CLEAR THE DATA WINDOW
2030 010610 104412 000002      PKCLK    ,2          ;PUSH OUT THE BIT
2031 010614 032777 000200 170562      BIT      #RXDONE,@RXCSR ;CHECK FOR DONE
2032 010622 001001      BNE      5$          ;BR IF SET
2033 010624 104000      HLT      ;DONE BIT ERROR
2034 010626 117737 170554 006054  5$:     MOV      @RXDBUF,TEMP ;GET DATA
2035 010634 122737 000125 006054  CMP      #125,TEMP    ;CHECK IT
2036 010642 001401      BEQ      64$        ;BR IF A MATCH
2037 010644 104010      HLT      10        ;DATA COMPARE ERROR
2038 010646 052777 002000 170536  64$:     BIS      #BITW,@TXCSR ;PUT A 1 IN THE WINDOW
2039 010654 104412 000002      PKCLK    ,2          ;PUSH IT OUT
2040 010660 005237 001236      INC      TEMP1       ;INC THE # TO DO
2041 010664 022737 000006 001236  CMP      #6,TEMP1    ;CHECK FOR FINISH
2042 010672 001365      BNE      64$        ;BR IF MORE TO GO
2043 010674 042777 002000 170510      BIC      #BITW,@TXCSR ;CLEAR THE WINDOW
2044 010702 104412 000004      PKCLK    ,4          ;PUSH OUT 2 BINTS
2045 010706 017737 170472 001240  MOV      @RXCSR,TEMP2 ;GET THE CSR
2046 010714 032777 001000 170464  BIT      #REOM,@RXDBUF ;TEST FOR END OF MSG
2047 010722 001001      BNE      1$         ;BR IF SET
2048 010724 104001      HLT      1          ;BIT FAILED TO SET
2049 010726 032737 000200 001240  1$:     BIT      #RXDONE,TEMP2 ;TEST DONE
2050 010734 001001      BNE      10$        ;BR IF SET
2051 010736 104002      HLT      2          ;DONE FAILED TO SET
2052  ;//////////////////////////////////////;:++C
2053 010740 004137 007060      10$:     JSR      R1,RFLG      ;OUTPUT A FLAG CHAR
2054 010744 000001      1
2055 010746 017737 170432 001240  MOV      @RXCSR,TEMP2 ;GET CSR
2056 010754 032777 001000 170424  BIT      #REOM,@RXDBUF ;REOM SHOULD BE CLEAR
2057 010762 001401      BEQ      11$        ;BR IF CLEAR
2058 010764 104030      HLT      30        ;REOM FAILED TO CLEAR
2059 010766 032737 000200 001240  11$:     BIT      #RXDONE,TEMP2 ;RXDONE SHOULD BE SET
2060 010774 001001      BNE      4$         ;BR IF SET
2061 010776 104002      HLT      2          ;FAILED TO SET
2062  ;//////////////////////////////////////
2063 011000 104400      4$:     SCOPE          ;SCOPE THIS TEST
2064
2065 ;***** TEST 5 *****
2066 ;*TEST TO PROVE THE INTERACTION OF REOM
2067 ;*WITH DONE IN SECONDARY MODE. TEST FOR REOM
2068 ;*AT THE WRONG ADDRESS, THEN AT THE CORRECT
2069 ;*SECONDARY ADDRESS.
2070 ;*****
2071 ;*****
2072 ;*
2073 ;* TEST 5
2074 ;*
```

TEST OF REOM AND DONE IN SECONDARY MODE

```

2075      ::*****
2076      ::*****
2077 011002 012737 000005 001226 TST5:  MOV    #5,@TSTNO
2078 011010 012737 011450 001216      MOV    #TST6,NEXT
2079 011016 052777 000400 170366      BIS    #MRESET,@TXCSR ;RESET THE DEVICE
2080 011024 004737 005044          JSR    PC,SMALL      ;WAIT FOR RESET TO FINISH
2081 011030 052777 000020 170346      BIS    #RCVEN,@RXCSR ;TURN ON RECEIVER
2082 011036 052777 001000 170344      BIS    #CRCEN,@PARCSR ;TURN OFF CRC
2083 011044 005037 001236          CLR    TEMP1        ;BIT COUNTER
2084 011050 052777 010000 170372 BIS    #PRISEC,@PARCSR ;ENTER SECONDARY MODE
2085 011056 052777 014000 170326      BIS    #MODE,@TXCSR  ;ENTER MAINT MODE
2086 011064 004137 007060          JSR    R1,RFLG      ;PUSH OUT A FLAG
2087 011070 000001          1                ;ONE FLAG
2088 011072 012737 000001 001324 MOV    #1,DATA      ;LOAD A CHAR
2089 011100 012737 000010 001326      MOV    #8.,SHIFTS   ;LOAD THE # OF CLOCKS
2090 011106 004137 007020          JSR    R1,RPOKE     ;PUSH OUT THE DATA
2091 011112 042777 002000 170272      BIC    #BITW,@TXCSR ;CLEAR THE DATA WINDOW
2092 011120 104412 000002          PKCLK ,2           ;PUSH OUT THE BIT
2093 011124 032777 000200 170252      BIT    #RXDONE,@RXCSR ;CHECK FOR DONE
2094 011132 001401          BEQ    5$          ;BR IF CLEAR
2095 011134 104000          HLT                    ;DONE BIT ERROR
2096 011136 105777 170244          5$:  TSTB   @RXDBUF      ;READ THE BUFFER
2097 011142 052777 002000 170242 64$:  BIS    #BITW,@TXCSR ;PUT A 1 IN THE WINDOW
2098 011150 104412 000002          PKCLK ,2           ;PUSH IT OUT
2099 011154 005237 001236          INC    TEMP1        ;INC THE # TO DO
2100 011160 022737 000006 001236      CMP    #6,TEMP1     ;CHECK FOR FINISH
2101 011166 001365          BNE    64$         ;BR IF MORE TO GO
2102 011170 042777 002000 170214      BIC    #BITW,@TXCSR ;CLEAR THE WINDOW
2103 011176 104412 000004          PKCLK ,4           ;PUSH OUT 2 BINTS
2104 011202 032777 001000 170176      BIT    #REOM,@RXDBUF ;TEST REC END OF MSG
2105 011210 001401          BEQ    2$          ;BR IF NOT SET
2106 011212 104004          HLT    4            ;BIT IS SET AND SHOULDN'T
2107      ;BE - THIS IS SECONDARY MODE
2108 011214 004137 007060          2$:  JSR    R1,RFLG      ;OUTPUT A FLAG
2109 011220 000001          1
2110 011222 005037 001324          CLR    DATA        ;CLEAR DATA
2111 011226 012737 000010 001326      MOV    #8.,SHIFTS   ;LOAD THE # OF CLOCKS
2112 011234 004137 007020          JSR    R1,RPOKE     ;PUSHOUT SEC. ADRS
2113 011240 012737 000252 001324      MOV    #252,DATA    ;LOAD DATA
2114 011246 012737 000010 001326      MOV    #8.,SHIFTS   ;LOAD # OF CLOCKS
2115 011254 004137 007020          JSR    R1,RPOKE     ;PUSHOUT DATA
2116 011260 042777 002000 170124      BIC    #BITW,@TXCSR ;CLEAR OUT DATA WINDOW
2117 011266 104412 000002          PKCLK ,2           ;PUSH OUT A BIT
2118 011272 032777 000200 170104      BIT    #RXDONE,@RXCSR ;CHECK DONE
2119 011300 001001          BNE    6$          ;BR IF SET
2120 011302 104002          HLT    2            ;DONE FAILED TO SET
2121 011304 005037 001236          6$:  CLR    TEMP1        ;CLEAR TO KEEP TRACK OF THE BITS
2122 011310 105777 170072          TSTB   @RXDBUF      ;READ THE BUFFER
2123 011314 052777 002000 170070 65$:  BIS    #BITW,@TXCSR ;SET THE WINDOW BIT
2124 011322 104412 000002          PKCLK ,2           ;PUSH IT OUT
2125 011326 005237 001236          INC    TEMP1        ;CHECK TO SEE IF
2126 011332 022737 000006 001236      CMP    #6,TEMP1     ;6 ARE DONE YET
2127 011340 001365          BNE    65$         ;BR IF NO
2128 011342 042777 002000 170042      BIC    #BITW,@TXCSR ;CLEAR THE WINDOW
2129 011350 104412 000004          PKCLK ,4           ;PUSH 2 BITS
2130 011354 017737 170024 001240      MOV    @RXCSR,TEMP2 ;READ THE CSR
  
```

TEST OF REOM AND DONE IN SECONDARY MODE

```

2131 011362 032777 001000 170016 BIT #REOM,@RXDBUF ;TEST END OF MSG
2132 011370 001001 BNE 3$ ;BR IF SET
2133 011372 104001 HLT 1 ;EOM FAILED TO SET IN SEC. MODE
2134 011374 032737 000200 001240 3$: BIT #RXDONE,TEMP2 ;TEST DONE
2135 011402 001001 BNE 10$ ;BR IF SET
2136 011404 104002 HLT 2 ;BIT FAILED TO SET
2137 ;//////////////////////////////////////;
2138 011406 004137 007060 10$: JSR R1,RFLG ;PUTPUT A FLAG
2139 011412 000001 1 MOV 1
2140 011414 017737 167764 001240 MOV @RXCSR,TEMP2 ;GET CSR
2141 011422 032777 001000 167756 BIT #REOM,@RXDBUF ;REOM SHOULD BE CLEAR
2142 011430 001401 BEQ 11$ ;BR IF CLEAR
2143 011432 104030 HLT 30 ;FAILED TO CLEAR
2144 011434 032737 000200 001240 11$: BIT #RXDONE,TEMP2 ;DONE SHOULD BE CLEAR
2145 011442 001401 BEQ 4$ ;BR IF CLEAR
2146 011444 104031 HLT 31 ;BIT FAILED TO CLEAR
2147 ;//////////////////////////////////////;
2148
2149 011446 104400 4$: SCOPE ;SCOPE THIS TEST
2150
2151 ;***** TEST 6 *****
2152 ;*TEST TO PROVE THE INTERACTION OF A3ORT
2153 ;*WITH A DONE AND RX ERROR IN PRIMARY MODE
2154
2155 ;:*****
2156 ;:*****
2157 ;: *
2158 ;: TEST 6
2159 ;: *
2160 ;:*****
2161 ;:*****
2162 011450 012737 000006 001226 TST6: MOV #6,@TSTNO
2163 011456 012737 011662 001216 MOV #TST7,NEXT
2164 011464 052777 000400 167720 BIS #MRESET,@TXCSR ;RESET THE DEVICE
2165 011472 004737 005044 JSR PC,SMALL ;WAIT FOR RESET TO FINISH
2166 011476 052777 001000 167704 BIS #CRCEN,@PARCSR ;TURN OFF CRC
2167 011504 052777 000020 167672 BIS #RCVEN,@RXCSR ;TURN ON RECEIVER
2168 011512 052777 014000 167672 BIS #MMODE,@TXCSR ;ENTER MAINT MODE
2169 011520 004137 007060 JSR R1,RFLG ;PUSH OUT A FLAG
2170 011524 000001 1 HLT ;ONE FLAG
2171 011526 005037 001324 CLR DATA ;CLEAR DATA
2172 011532 012737 000010 001326 MOV #8.,SHIFTS ;LOAD CLOCKS
2173 011540 004137 007020 JSR R1,RPOKE ;PUSH OUT DATA CHAR
2174 011544 052777 002000 167640 BIS #BITW,@TXCSR ;SET THE WINDOW
2175 011552 104412 000002 PKCLK. 2 ;PUSH OUT A BIT
2176 011556 105777 167624 TSTB @RXDBUF ;READ A CHAR
2177 011562 112737 000177 001324 MOVB #177,DATA ;LOAD A SECOND CHAR
2178 011570 012737 000010 001326 MOV #8.,SHIFTS ;LOAD CLOCKS
2179 011576 004137 007020 JSR R1,RPOKE ;PUSH OUT THE DATA
2180 011602 017737 167576 001236 MOV @RXCSR,TEMP1 ;READ THE CSR
2181 011610 017700 167572 MOV @RXDBUF,RO ;SAVE THE BUFFER
2182 011614 032700 002000 BIT #RABORT,RO ;TEST ABORT BIT
2183 011620 001001 BNE 1$ ;BR IF SET
2184 011622 104005 HLT 5 ;ABORT BIT FAILED TO SET
2185 011624 032737 004000 001236 1$: BIT #REACT,TEMP1 ;TEST REC ACT
2186 011632 001401 BEQ 2$ ;BR IF CLEAR

```

TEST OF ABORT,DONE,AND RXERR IN PRIMARY MODE

```

2187 011634 104006          HLT      6          ;REC ACT. FAILED TO CLEAR
2188 011636 032737 000200 001236 2$: BIT      #RXDONE,TEMP1 ;TEST DONE
2189 011644 001001          BNE      3$          ;BR IF SET
2190 011646 104002          HLT      2          ;DONE FAILED TO SET
2191 011650 032700 100000 3$: BIT      #RXDERR,R0    ;TEST ERROR BIT
2192 011654 001001          BNE      12$         ;BR IF SET
2193 011656 104007          HLT      7          ;ERROR BIT FAILED TO SET AFTER ABORT
2194 011660 104400          12$: SCOPE          ;SCOPE THIS TEST
2195          ;***** TEST 7 *****
2196          ;*TEST TO PROVE THE INTERACTION OF ABORT IN SECONDARY MODE
2197          ;*TEST FOR ABORT AT THE WRONG SECONDARY ADDRESS, THEN TEST
2198          ;*AT THE CORRECT SECONDARY ADDRESS
2199          ;*****
2200          ;*****
2201          ;*
2202          ; TEST 7
2203          ;*
2204          ;*****
2205          ;*****
2206 011662 012737 000007 001226 TST7: MOV      #7,@TSTNO
2207 011670 012737 012220 001216 MOV      #TST10,NEXT
2208 011676 052777 000400 167506 BIS      #MRESET,@TXCSR ;RESET THE DEVICE
2209 011704 004737 005044          JSR      PC,SMALL    ;WAIT FOR RESET TO FINISH
2210 011710 052777 001000 167472 BIS      #CRCEN,@PARCSR ;TURN OFF CRC
2211 011716 052777 000020 167460 BIS      #RCVEN,@RXCSR  ;TURN ON RECEIVER
2212 011724 052777 010000 167456 BIS      #PRISEC,@PARCSR ;ENTER SECONDARY MODE
2213 011732 052777 014000 167452 BIS      #MMODE,@TXCSR  ;ENTER MAINT MODE
2214 011740 004137 007060          JSR      R1,RFLG    ;PUSH OUT A FLAG
2215 011744 000001          1          ;ONE FLAG
2216 011746 012737 000001 001324 MOV      #1,DATA    ;LOAD WRONG ADRS
2217 011754 012737 000010 001326 MOV      #8.,SHIFTS ;LOAD CLOCKS
2218 011762 004137 007020          JSR      R1,RPOKE   ;PUSH OUT DATA CHAR
2219 011766 052777 002000 167416 BIS      #BITW,@TXCSR ;SET THE WINDOW
2220 011774 104412 000002          PKCLK, 2          ;PUSH OUT A BIT
2221 012000 105777 167402          TSTB    @RXDBUF    ;READ A CHAR
2222 012004 112737 000177 001324 MOVB    #177,DATA   ;LOAD A SECONC CHAR
2223 012012 012737 000010 001326 MOV      #8.,SHIFTS ;LOAD CLOCKS
2224 012020 004137 007020          JSR      R1,RPOKE   ;PUSH OUT THE DATA
2225 012024 032777 002000 167354 BIT      #RABORT,@RXDBUF ;TEST FOR ABORT
2226 012032 001401          BEQ     5$          ;BR IF NOT SET
2227 012034 104011          HLT     11         ;ABORT IS SET AND SHOULDN'T
2228          ;BE - THIS IS A SECONDARY STATION
2229 012036 004137 007060 5$: JSR      R1,RFLG    ;PUSH OUT A FLAG
2230 012042 000001          1          ;ONE FLAG
2231 012044 005037 001324          CLR     DATA      ;LOAD ADRS
2232 012050 012737 000010 001326 MOV      #8.,SHIFTS ;LOAD CLOCKS
2233 012056 004137 007020          JSR      R1,RPOKE   ;PUSH OUT THE ADRS
2234 012062 012737 000252 001324 MOV      #252,DATA   ;LOAD A CHAR
2235 012070 012737 000010 001326 MOV      #8.,SHIFTS ;LOAD CLOCKS
2236 012076 004137 007020          JSR      R1,RPOKE   ;PUSH OUT THE CHAR
2237 012102 052777 002000 167302 BIS      #BITW,@TXCSR ;SET THE WINDOW
2238 012110 104412 000002          PKCLK, 2          ;PUSH OUT A BIT
2239 012114 105777 167266          TSTB    @RXDBUF    ;READ THE CHAR
2240 012120 112737 000177 001324 MOVB    #177,DATA   ;LOAD THE ABORT
2241 012126 012737 000010 001326 MOV      #8.,SHIFTS ;LOAD CLOCKS
2242 012134 004137 007020          JSR      R1,RPOKE   ;PUSH OUT THE ABORT

```


TEST OF ABORT,DONE,AND RXERR IN SEC MODE

```
2243 012140 017737 167240 001240      MOV      @RXCSR,TEMP2      ;READ THE CSR
2244 012146 017700 167234              MOV      @RXDBUF,R0       ;SAVE THE BUFFER
2245 012152 032700 002000              BIT      #RABORT,R0      ;TEST THE ABORT BIT
2246 012156 001001                    BNE      6$              ;BR IF SET
2247 012160 104005                    HLT      5                ;BIT IS CLEAR - SHOULD BE SET
2248 012162 032737 004000 001240 6$:   BIT      #REACT,TEMP2    ;TEST ACTIVE
2249 012170 001401                    BEQ      7$              ;BR IF CLEAR
2250 012172 104006                    HLT      6                ;BIT FAILED TO CLEAR AFTER ABORT
2251 012174 032737 000200 001240 7$:   BIT      #RXDONE,TEMP2   ;TEST DONE
2252 012202 001001                    BNE      10$             ;BR IF SET
2253 012204 104002                    HLT      2                ;DONE FAILED TO SET
2254 012206 032700 100000 10$:        BIT      #RXDERR,R0      ;TEST FOR ERROR
2255 012212 001001                    BNE      12$             ;BR IF SET
2256 012214 104007                    HLT      7                ;ERROR WASN'T SET AFTER ABORT
2257
2258 012216 104400 12$:        SCOPE                ;SCOPE THIS TEST
2259                                     ;***** TEST 10 *****
2260                                     ;*TEST EFFECT OF REC ENABLE ON ABORT ;:++C
2261                                     ;*RXACTIVE, RXDONE, RXERROR.
2262
2263                                     ;:*****
2264                                     ;:*****
2265                                     ;:
2266                                     ;: TEST 10
2267                                     ;:
2268                                     ;:*****
2269                                     ;:*****
2270 012220 012737 000010 001226 TST10: MOV      #10,@TSTNO
2271 012226 012737 012462 001216      MOV      #TST11,NEXT
2272 012234 052777 000400 167150      BIS      #MRESET,@TXCSR ;RESET THE DEVICE
2273 012242 004737 005044              JSR      PC,SMALL        ;WAIT FOR RESET TO FINISH
2274 012246 052777 001000 167134      BIS      #CRCEN,@PARCSR ;TURN OFF CRC
2275 012254 052777 000020 167122      BIS      #RCVEN,@RXCSR  ;TURN ON RECEIVER
2276 012262 052777 014000 167122      BIS      #MMODE,@TXCSR  ;ENTER MAINT MODE
2277 012270 004137 007060              JSR      R1,RFLG        ;PUSH OUT A FLAG
2278 012274 000001                    1                ;ONE FLAG
2279 012276 005037 001324              CLR      DATA          ;CLEAR DATA
2280 012302 012737 000010 001326      MOV      #8.,SHIFTS     ;LOAD CLOCKS
2281 012310 004137 007020              JSR      R1,RPOKE       ;PUSH OUT DATA CHAR
2282 012314 052777 002000 167070      BIS      #BITW,@TXCSR   ;SET THE WINDOW
2283 012322 104412 000002              PKCLK, 2             ;PUSH OUT A BIT
2284 012326 105777 167054              TSTB    @RXDBUF        ;READ A CHAR
2285 012332 112737 000177 001324      MOVB    #177,DATA      ;LOAD A SECOND CHAR
2286 012340 012737 000010 001326      MOV      #8.,SHIFTS     ;LOAD CLOCKS
2287 012346 004137 007020              JSR      R1,RPOKE       ;PUSH OUT THE DATA
2288 012352 017737 167026 001236      MOV      @RXCSR,TEMP1   ;READ THE CSR
2289 012360 017700 167022              MOV      @RXDBUF,R0     ;SAVE THE BUFFER
2290 012364 032700 002000              BIT      #RABORT,R0     ;TEST ABORT BIT
2291 012370 001001                    BNE      20$            ;BR IF SET
2292 012372 104005                    HLT      5                ;ABORT BIT FAILED TO SET
2293 012374 042777 000020 167002 20$:   BIC      #RCVEN,@RXCSR  ;CLEAR REC ENABLE
2294 012402 017737 166776 001236      MOV      @RXCSR,TEMP1   ;SAVE STATUS
2295 012410 017700 166772              MOV      @RXDBUF,R0     ;SAVE REC BUFFER
2296 012414 032700 002000              BIT      #RABORT,R0     ;ABORT CLEAR?
2297 012420 001401                    BEQ      1$              ;IF YES,BR
2298 012422 104030                    HLT      30              ;ABORT FAILED TO CLEAR
```

TEST OF RCVEN, ABORT, RXACTIVE, RXDONE, RXERROR

```

2299 012424 032737 004000 001236 1$: BIT #REACT,TEMP1 ;TEST REC ACT
2300 012432 001401 BEQ 2$ ;BR IF CLEAR
2301 012434 104006 HLT 6 ;REC ACT. FAILED TO CLEAR
2302 012436 032737 000200 001236 2$: BIT #RXDONE,TEMP1 ;TEST DONE
2303 012444 001401 BEQ 3$ ;BR IF CLEAR
2304 012446 104031 HLT 31 ;DONE FAILED TO CLEAR
2305 012450 032700 100000 3$: BIT #RXDERR,RO ;TEST ERROR BIT
2306 012454 001401 BEQ 12$ ;BR IF CLEAR
2307 012456 104032 HLT 32 ;ERROR BIT FAILED TO CLEAR AFTER RCVEN
2308 ;CLEARED.
2309
2310 012460 104400 12$: SCOPE
2311 ;***** TEST 11 *****
2312 ;*TEST THE EFFECTS OF READIN, THE RXDBUF ;:++C
2313 ;*ON THE RXDONE, RXERROR, AND ABORT
2314
2315 ;:*****
2316 ;:*****
2317 ;*
2318 ; TEST 11
2319 ;*
2320 ;:*****
2321 ;:*****
2322 012462 012737 000011 001226 TST11: MOV #11,@TSTNO
2323 012470 012737 012710 001216 MOV #TST12,NEXT
2324 012476 052777 000400 166706 BIS #MRESET,@TXCSR ;RESET THE DEVICE
2325 012504 004737 005044 JSR PC,SMALL ;WAIT FOR RESET TO FINISH
2326 012510 052777 001000 166672 BIS #CRCEN,@PARCSR ;TURN OFF CRC
2327 012516 052777 000020 166660 BIS #RCVEN,@RXCSR ;TURN ON RECEIVER
2328 012524 052777 014000 166660 BIS #MODE,@TXCSR ;ENTER MAINT MODE
2329 012532 004137 007060 JSR R1,RFLG ;PUSH OUT A F_AG
2330 012536 000001 1 CLR 1 ;ONE FLAG
2331 012540 005037 001324 CLR DATA ;CLEAR DATA
2332 012544 012737 000010 001326 MOV #8.,SHIFTS ;LOAD CLOCKS
2333 012552 004137 007020 JSR R1,RPOKE ;PUSH OUT DATA CHAR
2334 012556 052777 002000 166626 BIS #BITW,@TXCSR ;SET THE WINDOW
2335 012564 104412 000002 PKCLK, 2 ;PUSH OUT A BIT
2336 012570 105777 166612 TSTB @RXDBUF ;READ A CHAR
2337 012574 112737 000177 001324 MOVB #177,DATA ;LOAD A SECOND CHAR
2338 012602 012737 000010 001326 MOV #8.,SHIFTS ;LOAD CLOCKS
2339 012610 004137 007020 JSR R1,RPOKE ;PUSH OUT THE DATA
2340 012614 017737 166564 001236 MOV @RXCSR,TEMP1 ;READ THE CSR
2341 012622 017700 166560 MOV @RXDBUF,RO ;SAVE THE BUFFER
2342 012626 032700 002000 BIT #RABORT,RO ;TEST ABORT BIT
2343 012632 001001 BNE 20$ ;BR IF SET
2344 012634 104005 HLT 5 ;ABORT BIT FAILED TO SET
2345 012636 005777 166544 20$: TST @RXDBUF ;READ RXDBUF
2346 012642 017737 166536 001236 MOV @RXCSR,TEMP1 ;SAVE STATUS
2347 012650 017700 166532 MOV @RXDBUF,RO ;SAVE REC BUFFER
2348 012654 032737 000200 001236 1$: BIT #RXDONE,TEMP1 ;TEST DONE
2349 012662 001401 BEQ 2$ ;BR IF CLEAR
2350 012664 104031 HLT 31 ;DONE FAILED TO CLEAR
2351 012666 032700 002000 2$: BIT #RABORT,RO ;ABORT CLEAR?
2352 012672 001401 BEQ 3$ ;IF YES,BR
2353 012674 104030 HLT 30 ;ABORT FAILED TO CLEAR
2354 012676 032700 100000 3$: BIT #RXDERR,RO ;TEST ERROR BIT

```

READ RXDBUF, AND TEST RXDONE, RXERROR, ABORT

```
2355 012702 001401      BEQ    12$      ;BR IF CLEAR
2356 012704 104032      HLT    32      ;ERROR BIT FAILED TO CLEAR AFTER RXDBUF
2357                                ;READ.
2358 012706 104400      12$:  SCOPE    ;SCOPE THIS TEST
2359                                ;***** TEST 12 *****
2360                                ;*DATA OVERRUN TEST IN PRIMARY MODE. TEST TO
2361                                ;*PROVE OVERRUN ERROR AND RX ERROR WILL OCCUR
2362                                ;*****
2363                                ;*****
2364                                ;*****
2365                                ;*
2366                                ; TEST 12
2367                                ;*
2368                                ;*****
2369                                ;*****
2370 012710 012737 000012 001226  TST12: MOV    #12,@TSTNO ;LOAD CHAR TO BE OUTPUT
2371 012716 012737 013062 001216  MOV    #TST13,NEXT ;CLOCK SETUP
2372 012724 052777 000400 166460  BIS    #MRESET,@TXCSR ;RESET THE DEVICE
2373 012732 004737 005044      JSR    PC,SMALL ;WAIT FOR RESET TO FINISH
2374 012736 013703 001404      MOV    RXCSR,R3 ;LOAD THE CONTROL REGISTER
2375 012742 052777 001000 166440  BIS    #CRCEN,@PARCSR ;TURN OFF CRC
2376 012750 052713 000020      BIS    #RCVEN,(R3) ;TURN ON THE RECEIVER
2377 012754 052777 014000 166430  BIS    #MODE,@TXCSR ;ENTER MAINTENANCE MODE
2378 012762 004137 007060      JSR    R1,RFLG ;PUSH OUT A FLAG
2379 012766 000001      1 ;ONE FLAG
2380 012770 012737 000252 001324  MOV    #252,DATA ;LOAD CHAR TO BE OUTPUT
2381 012776 012737 000010 001326  MOV    #8.,SHIFTS ;CLOCK SETUP
2382 013004 004137 007020      JSR    R1,RPOKE ;PUSH DATA INTO RECEIVER
2383 013010 012737 000070 001324  MOV    #70,DATA ;LOAD A SECOND CHARACTER
2384 013016 012737 000011 001326  MOV    #9.,SHIFTS ;SETUP #OF CLOCKS
2385 013024 004137 007020      JSR    R1,RPOKE ;PUSH THE SECOND CHARACTER INTO RX
2386 013030 017700 166352      MOV    @RXDBUF,R0 ;SAVE BUFFER
2387 013034 042700 037777      BIC    #^C<RXDERR.OVRRUN>,R0 ;CLEAR JUNK
2388 013040 022700 140000      CMP    #RXDERR!OVRRUN,R0 ;CHECK TO SEE IF BOTH ARE SET
2389 013044 001401      BEQ    2$      ;BR IF OK
2390 013046 104013      HLT    13      ;THEY DIDN'T MATCH
2391
2392 013050 032713 004000      2$:  BIT    #REACT,(R3) ;TEST REC. ACT
2393 013054 001001      BNE    3$      ;BR IF STILL SET
2394 013056 104014      HLT    14      ;REC. ACTIVE CLEARED AND SHOULD BE SET
2395
2396 013060 104400      3$:  SCOPE    ;SCOPE THIS TEST
2397
2398                                ;***** TEST 13 *****
2399                                ;*DATA OVERRUN TEST IN SECONDARY MODE. TEST TO PROVE
2400                                ;*THAT OVERRUN DOES NOT OCCUR IF THIS STATION IS NOT
2401                                ;*SELECTED. THEN SELECT THIS SECONDARY STATION AND
2402                                ;*PROVE OVERRUN ERROR AND RX ERROR WILL OCCUR
2403                                ;*****
2404                                ;*****
2405                                ;*****
2406                                ;*
2407                                ; TEST 13
2408                                ;*
2409                                ;*****
2410                                ;*****
```

```
2411 013062 012737 000013 001226 TST13: MOV #13,@TSTNO
2412 013070 012737 013320 001216 MOV #TST14,NEXT
2413 013076 052777 000400 166306 BIS #MRESET,@TXCSR ;RESET THE DEVICE
2414 013104 004737 005044 JSR PC,SMALL ;WAIT FOR RESET TO FINISH
2415 013110 013703 001404 MOV RXCSR,R3 ;LOAD THE CONTROL REGISTER
2416 013114 052777 001000 166266 BIS #CRCEN,@PARCSR ;TURN OFF CRC
2417 013122 052713 000020 BIS #RCVEN,(R3) ;TURN ON THE RECEIVER
2418 013126 052777 010000 166254 BIS #PRISEC,@PARCSR ;ENTER SECONDARY MODE
2419 013134 052777 014000 166250 BIS #MMODE,@TXCSR ;ENTER MAINTENANCE MODE
2420 013142 004137 007060 JSR R1,RFLG ;PUSH OUT A FLAG
2421 013146 000001 1 MOV #1,RFLG ;ONE FLAG
2422 013150 012737 000252 001324 MOV #252,DATA ;LOAD AN INCORRECT SEC. STA ADRS
2423 013156 012737 000031 001326 MOV #25,SHIFTS ;PUSH OUT 3 CHARS
2424 013164 004137 007020 JSR R1,RPOKE ;THRU THE RECEIVER
2425 013170 032777 040000 166210 BIT #OVRRUN,@RXDBUF ;TEST FOR OVERRUN
2426 013176 001401 1 BEQ 4$ ;BR IF NOT SET
2427
2428 013200 104015 HLT 15 ;OVERRUN IS SET AND SHOULDN'T BE
2429 ;THIS IS A SECONDARY STATION WHICH
2430 ;DID NOT GET ITS ADDRESS
2431 013202 004137 007060 4$: JSR R1,RFLG ;PUSH OUT A FLAG
2432 013206 000001 1 MOV #1,RFLG ;ONE FLAG
2433 013210 005037 001324 CLR DATA ;GET CORRECT SEC. STATION ADRS
2434 013214 012737 000010 001326 MOV #8,SHIFTS ;MOVE THE #OF CLOCKS TO PUSH
2435 013222 004137 007020 JSR R1,RPOKE ;PUSH OUT THE SEC. ADRS
2436 013226 012737 000252 001324 MOV #252,DATA ;PUSH OUT
2437 013234 012737 000010 001326 MOV #8,SHIFTS ;TWO
2438 013242 004137 007020 JSR R1,RPOKE ;DATA
2439 013246 012737 000070 001324 MOV #70,DATA ;CHARACTERS
2440 013254 012737 000011 001326 MOV #9,SHIFTS ;TO FORCE
2441 013262 004137 007020 JSR R1,RPOKE ;AN OVERRUN ERROR
2442 013266 017700 166114 MOV @RXDBUF,R0 ;SAVE
2443 013272 042700 037777 BI #C<RXDERR!OVRRUN>,R0 ;CLEAR JUNK
2444 013276 022700 140000 CMP #RXDERR!OVRRUN,R0 ;ARE THE ERROR BITS THERE
2445 013302 001401 1 BEQ 6$ ;BR IF YES
2446 013304 104013 1 HLT 13 ;ERROR BITS MISSING
2447
2448 013306 032713 004000 6$: BIT #REACT,(R3) ;TEST ACTIVE
2449 013312 001001 1 BNE 3$ ;BR IF SET
2450 013314 104005 1 HLT 5 ;ACTIVE DROPPED OUT
2451 013316 104400 3$: SCOPE ;SCOPE THIS TEST
2452
2453 ;***** TEST 14 *****
2454 ;*TEST OF SPECIFIC DATA PATTERNS THRU
2455 ;*THE RECEIVER IN PRIMARY MODE
2456 ;*****
2457 ;*****
2458 ;*
2459 ;* TEST 14
2460 ;*
2461 ;*****
2462 ;*****
2463 013320 012737 000014 001226 TST14: MOV #14,@TSTNO
2464 013326 012737 013570 001216 MOV #TST15,NEXT
2465 013334 052777 000400 166050 BIS #MRESET,@TXCSR ;RESET THE DEVICE
2466 013342 004737 005044 JSR PC,SMALL ;WAIT FOR RESET TO FINISH
```

```

2467 013346 012703 013556      MOV      #TBLA,R3      ;LOAD THE TABLE POINTER
2468 013352 012704 013556      MOV      #TBLA,R4      ;DITTO
2469 013356 012702 000004      MOV      #4,R2         ;LOAD THE # OF CHARS TO DO
2470 013362 052777 001000 166020  BIS      #CRCEN,@PARCSR
2471 013370 005077 166020      CLR      @TXDBUF       ;RESET TXDONE
2472 013374 052777 014000 166010  BIS      #MMODE,@TXCSR  ;ENTER M/MODE
2473 013402 052777 000020 165774  BIS      #RCVEN,@RXCSR  ;TURN ON THE RECEIVER
2474 013410 052777 000020 165774  BIS      #SEND,@TXCSR   ;TURN ON THE TRANSMITTER
2475 013416 052777 000400 165770  BIS      #TSOM,@TXDBUF  ;START A FLAG
2476 013424 104412 000010      PKCLK    ,8           ;START A FLAG
2477 013430 012377 165760      MOV      (R3)+,@TXDBUF ;LOAD FIRST DATA CHAR
2478 013434 005302                DEC      R2           ;LOWER THE # TO DO
2479 013436 104412 000020      PKCLK    ,16          ;PUSH OUT
2480 013442 012377 165746      1$:     MOV      (R3)+,@TXDBUF ;LOAD DATA
2481 013446 005302                DEC      R2           ;LOWER THE # TO DO
2482 013450 104412 000020      PKCLK    ,16          ;PUSH OUT DATA
2483 013454 105777 165724      TSTB    @RXCSR        ;CHECK FOR RX DONE
2484 013460 100401                BMI     2$           ;BR IF DONE SET
2485 013462 104000                HLT                    ;RX DONE FAILED TO SET
2486 013464 017737 165716 001324 2$:     MOV      @RXDBUF,DATA  ;GET THE BUFFER
2487 013472 121437 001324      CMPB    (R4),DATA     ;CHECK IT
2488 013476 001401                BEQ     3$           ;BR IF OK
2489 013500 104000                HLT                    ;DATA COMPARE ERROR
2490 013502 005724      3$:     TST      (R4)+
2491 013504 005702                TST      R2           ;CHECK FOR FINISH
2492 013506 001355                BNE     1$           ;BR IF MORE TO GO
2493 013510 104412 000020      PKCLK    ,16          ;PUSH OUT LAST CHAR
2494 013514 105777 165664      TSTB    @RXCSR        ;CHECK FOR DONE
2495 013520 100401                BMI     4$           ;BR IF DONE SET
2496 013522 104000                HLT                    ;DONE FAILED TO SET
2497 013524 017737 165656 001324 4$:     MOV      @RXDBUF,DATA  ;GET THE BUFFER
2498 013532 121437 001324      CMPB    (R4),DATA     ;CHECK IT
2499 013536 001401                BEQ     5$           ;BR IF A MATCH
2500 013540 104000                HLT                    ;DATA COMPARE ERROR
2501 013542      5$:
2502 013542 052777 000400 165642  BIS      #MRESET,@TXCSR ;RESET THE DEVICE
2503 013550 004737 005044      JSR     PC,SMALL      ;WAIT FOR RESET TO FINISH
2504 013554 104400      SCOPE
2505
2506      ;DATA TABLE
2507 013556 000125      TBLA:   .WORD    125
2508 013560 000252      .WORD    252
2509 013562 000000      .WORD    000
2510 013564 000377      .WORD    377
2511 013566 000000      .WORD    000
2512
2513      ;***** TEST 15 *****
2514      ;*TEST TO PROVE THAT THE DEVICE WILL
2515      ;*WORK WITH ALL POSSIBLE SECONDARY
2516      ;*STATION ADDRESSES.
2517      ;*****
2518
2519      ;*****
2520      ;*
2521      ;* TEST 15
2522      ;*

```

```
2523 :*****
2524 :*****
2525 013570 012737 000015 001226 TST15: MOV #15,@#TSTNO
2526 013576 012737 014064 001216 MOV #TST16,NEXT
2527 013604 012737 013632 001220 MOV #20$,LOCK ;SW09 SETUP
2528 013612 005005 CLR R5 ;CLEAR SEC ADRS HOLD
2529 013614 005037 001236 CLR TEMP1 ;CLEAR TEMP STORAGE
2530 013620 005037 001240 CLR TEMP2 ;DITTO
2531 013624 012737 011000 001236 MOV #CRCEN!PRISEC,TEMP1 ;LOAD MODE AND NO CRC
2532 013632 20$:
2533 013632 052777 000400 165552 BIS #MRESET,@TXCSR ;RESET THE DEVICE
2534 013640 004737 005044 JSR PC,SMALL ;WAIT FOR RESET TO FINISH
2535 013644 013777 001236 165536 MOV TEMP1,@PARCSR ;LOAD MODE AND NO CRC AND SEC STATION
2536 013652 052777 000020 165524 1$: BIS #RCVEN,@RXCSR ;TURN ON RECEIVER
2537 013660 052777 004020 165524 BIS #SYSTST!SEND,@TXCSR ;TURN ON TRANSMITTER AND CLOCK
2538 013666 105777 165520 2$: TSTB @TXCSR ;WAIT FOR
2539 013672 100375 BPL 2$ ;DONE
2540 013674 012777 000400 165512 MOV #TSOM,@TXDBUF ;LOAD START OF MSG
2541 013702 105777 165504 3$: TSTB @TXCSR ;WAIT FOR
2542 013706 100375 BPL 3$ ;DONE AGAIN
2543 013710 013777 001240 165476 MOV TEMP2,@TXDBUF ;LOAD SEC STATION ADRS AND
2544 013716 105777 165470 4$: TSTB @TXCSR ;WAIT FOR
2545 013722 100375 BPL 4$ ;DONE AGAIN
2546 013724 012777 000252 165462 MOV #252,@TXDBUF ;NOW LOAD DATA
2547 013732 032777 004000 165454 64$: BIT #TIMER,@TXDBUF ;CHECK THE TIMER BIT
2548 013740 001374 BNE 64$ ;BR IF SET
2549 013742 032777 004000 165444 65$: BIT #TIMER,@TXDBUF ;CHECK THE TIMER BIT
2550 013750 001774 BEQ 65$ ;BR IF CLEAR
2551 013752 105777 165426 5$: TSTB @RXCSR ;TEST FOR
2552 013756 100375 BPL 5$ ;RX DONE
2553 013760 005777 165420 TST @RXCSR ;TEST FOR ERROR
2554 013764 100001 BPL 6$ ;BR IF NO ERROR
2555 013766 104014 HLT 14 ;ERROR FOUND!
2556 013770 017704 165412 6$: MOV @RXDBUF,R4 ;GET THE BUFFER
2557 013774 032704 000400 BIT #RSOM,R4 ;CHECK FOR START OF MSG
2558 014000 001001 BNE 7$ ;BR IF SET
2559 014002 104000 HLT ;START OF MSG FAILED TO SET
2560 014004 122704 000252 7$: CMPB #252,R4 ;CHECK FOR DATA
2561 014010 001401 BEQ 10$ ;BR IF A MATCH
2562 014012 104012 HLT 12 ;FAILED TO RECEIVE DATA AS A SEC STATION
2563 014014 104401 10$: SCOPE1 ;SW09=1?
2564 014016 052777 000400 165366 BIS #MRESET,@TXCSR ;RESET THE DEVICE
2565 014024 004737 005044 JSR PC,SMALL ;WAIT FOR RESET TO FINISH
2566 014030 105205 INCB R5 ;GET NEXT SEC ADRS
2567 014032 001413 BEQ 11$ ;BR IF ALL DONE
2568 014034 110537 001236 MOV R5,TEMP1 ;LOAD THE ADRS FOR PAR CSR
2569 014040 110537 001240 MOV R5,TEMP2 ;DITTO FOR TXD BUF
2570 014044 052737 011000 001236 BIS #CRCEN!PRISEC,TEMP1 ;LOAD FOR PAR CSR
2571 014052 013777 001236 165330 MOV TEMP1,@PARCSR ;DO IT
2572 014060 000674 BR 1$ ;BR TO DO OVER
2573 014062 104400 11$: SCOPE ;SCOPE THIS TEST
2574
2575 :***** TEST 16 *****
2576 :*TEST OF SPECIFIC CHARACTER DATA PATTERNS
2577 :*USING BCC CHECK IN PRIMARY MODE.
2578 :*****
```

```

2579
2580
2581      :*****
2582      : TEST 16
2583      :*****
2584      :*****
2585      :*****
2586 014064 012737 000016 001226 TST16: MOV #16,@TSTNO
2587 014072 012737 014602 001216      MOV #TST17,NEXT
2588 014100 012704 014410      MOV #TBLB,R4 ;GET THE TABLE POINTER
2589 014104 012703 000004      MOV #4,R3 ;GET THE # TO DO
2590 014110 012701 014410      MOV #TBLB,R1 ;GET THE POINTER
2591 014114 012702 014410      MOV #TBLB,R2 ;DITTO
2592 014120 005037 001324      CLR DATA ;CLR DATA HOLD
2593 014124 012737 102010 014574 10$: MOV #CRC.CCITT,XPOLY ;LOAD THE POLYNOMIAL FOR CRC
2594 014132 012737 177777 014600      MOV #-1,CALBCC ;PRESET BCC FOR SDLC
2595 014140 013737 014600 014162      MOV CALBCC,12$ ;MOVE BCC
2596 014146 011437 014160      MOV (R4),11$ ;MOVE DATA
2597 014152 004537 014422      JSR R5,SIMBCC ;GO CALCULATE BCC
2598 014156 000010      8. ;BASED
2599 014160 000001      11$: .BLKW 1 ;ON THESE
2600 014162 000001      12$: .BLKW 1 ;PARAMETERS
2601 014164 052777 000400 165220      BIS #MRESET,@TXCSR ;RESET THE DEVICE
2602 014172 004737 005044      JSR PC,SMALL ;WAIT FOR RESET TO FINISH
2603 014176 052777 014000 165206      BIS #MODE,@TXCSR ;ENTER MAINT MODE
2604 014204 052777 000020 165172      BIS #RCVEN,@RXCSR ;TURN ON THE RECEIVER
2605 014212 052777 000020 165172      BIS #SEND,@TXCSR ;TURN ON TRANSMITTER
2606 014220 052777 000400 165166      BIS #TSOM,@TXDBUF ;TURN ON START OF MSG
2607 014226 104412 000012      PKCLK ,10. ;PUSH OUT 4 BITS
2608 014232 105777 165154      TSTB @TXCSR ;CHECK FOR DONE
2609 014236 100401      BMI 1$ ;BR IF SET
2610 014240 104000      HLT ;DONE FAILED TO SET
2611 014242      1$:
2612 014242 011177 165146      MOV (R1),@TXDBUF ;LOAD DATA
2613 014246 104412 000020      PKCLK ,16. ;PUSH OUT 8 BITS
2614 014252 052777 001000 165134      BIS #TEOM,@TXDBUF ;SET END OF MSG
2615 014260 104412 000020      PKCLK ,16. ;PUSH OUT 8 MORE BITS
2616 014264 105777 165114      TSTB @RXCSP ;CHECK FOR DONE
2617 014270 100401      BMI 2$ ;BR IF SET
2618 014272 104000      HLT ;DONE FAILED TO SET
2619 014274 017737 165106 001324 2$: MOV @RXDBUF,DATA ;READ THE BUFFER
2620 014302 121237 001324      CMPB (R2),DATA ;CHECK THE DATA
2621 014306 001401      BEQ 3$ ;BR IF A MATCH
2622 014310 104000      HLT ;DATA COMPARE ERROR
2623 014312 104412 000020      3$: PKCLK ,16. ;PUSH OUT 8 MORE BITS
2624 014316 105777 165062      TSTB @RXCSR ;CHECK FOR DONE
2625 014322 100401      BMI 4$ ;BR IF SET
2626 014324 104000      HLT ;DONE FAILED TO SET ON FIRST OCTET OF BCC
2627 014326 017737 165054 001324 4$: MOV @RXDBUF,DATA ;READ THE BUFFER
2628 014334 104412 000032      PKCLK ,26. ;FINISH POKING BITS
2629 014340 105777 165040      TSTB @RXCSR ;CHECK FOR DONE
2630 014344 100401      BMI 5$ ;BR IF SET
2631 014346 104000      HLT ;DONE FAILED TO SET ON SECOND OCTET OF BCC
2632 014350 117737 165032 001325 5$: MOVB @RXDBUF,DATA +1 ;GET IT
2633 014356 005137 014600      COM CALBCC ;INVERT SOFTWARE
2634 014362 023737 014600 001324      CMP CALBCC,DATA ;COMPARE
  
```

RECEIVER DATA TEST WITH CRC

```

2635 014370 001401      BEQ      6$      :BR IF A MATCH
2636 014372 104000      HLT                               :BCC COMPARE FAILURE
2637                                     :USE CRC DEBUGGING ROUTINE TO
2638                                     :REPAIR CRC.SEE FRONT OF LISTING
2639 014374 005721      6$:   TST      (R1)+   :POP POINTER
2640 014376 005722      TST      (R2)+   :DITTO
2641 014400 005724      TST      (R4)+   :DITTO
2642 014402 005303      DEC      R3      :DECREMENT THE # TO DO
2643 014404 001247      BNE      10$     :BR IF MORE TOGO
2644 014406 104400      SCOPE                               :SCOPE THIS TEST
2645
2646      ;DATA TABLE
2647 014410 000252      TBLB: .WORD    252
2648 014412 000000      .WORD    0
2649 014414 000125      .WORD   125
2650 014416 000377      .WORD   377
2651 014420 000000      .WORD    0
2652
2653
2654 014422 010046      SIMBCC: MOV     R0,-(SP)
2655 014424 010146      MOV     R1,-(SP)
2656 014426 010246      MOV     R2,-(SP)
2657 014430 012537 001236  MOV     (R5)+,TEMP1
2658 014434 012537 001240  MOV     (R5)+,TEMP2
2659 014440 012537 001242  MOV     (R5)+,TEMP3
2660 014444 005037 014576  1$:   CLR     BCCFBK
2661 014450 013700 001242  MOV     TEMP3,R0
2662 014454 006037 001240  ROR     TEMP2
2663 014460 005500      ADC     R0
2664 014462 032700 000001  BIT     #BIT0,R0
2665 014466 001402      BEQ     2$
2666 014470 005137 014576  COM     BCCFBK
2667 014474 013700 014574  2$:   MOV     XPOLY,R0
2668 014500 005100      COM     R0
2669 014502 040037 014576  BIC     R0,BCCFBK
2670 014506 000241      CLC
2671 014510 006037 001242  ROR     TEMP3
2672 014514 013700 014576  MOV     BCCFBK,R0
2673 014520 013701 001242  MOV     TEMP3,R1
2674 014524 010102      MOV     R1,R2
2675 014526 040100      BIC     R1,R0
2676 014530 043702 014576  BIC     BCCFBK,R2
2677 014534 050200      BIS     R2,R0
2678 014536 043737 014574 001242  BIC     XPOLY,TEMP3
2679 014544 050037 001242      BIS     R0,TEMP3
2680 014550 005337 001236      DEC     TEMP1
2681 014554 001333      BNE     1$
2682 014556 013737 001242 014600  MOV     TEMP3,CALBCC
2683 014564 012602      MOV     (SP)+,R2
2684 014566 012601      MOV     (SP)+,R1
2685 014570 012600      MOV     (SP)+,R0
2686 014572 000205      RTS
2687 014574 000000      XPOLY: 0
2688 014576 000000      BCCFBK: 0
2689 014600 000000      CALBCC: 0
2690      CRC16=120001

```


RECEIVER DATA TEST WITH CRC

CRC.CCITT=102010

2691 102010
2692
2693
2694
2695
2696
2697
2698
2699
2700
2701
2702
2703
2704
2705
2706 014602 012737 000017 001226
2707 014610 012737 015110 001216
2708 014616 052777 000400 164566
2709 014624 004737 005044
2710 014630 005037 001236
2711 014634 005037 001240
2712 014640 005037 001242
2713 014644 005077 164544
2714 014650 052777 014000 164534
2715 014656 052777 001000 164524
2716 014664 052777 000020 164520
2717 014672 052777 000020 164504
2718 014700 052777 001400 164506
2719 014706 104412 000002
2720 014712 005237 001236
2721 014716 022737 000021 001236
2722 014724 001407
2723 014726 104412 000002
2724 014732 032777 040000 164452
2725 014740 001764
2726 014742 104000
2727 014744 104412 000002 164434
2728 014750 032777 040000
2729 014756 001401
2730 014760 104000
2731 014762 012777 000125 164424
2732 014770 104412 000002
2733 014774 032777 040000 164410
2734 015002 001001
2735 015004 104000
2736 015006 005237 001240
2737 015012 022737 000006 001240
2738 015020 001363
2739 015022 104412 000002
2740 015026 032777 040000 164356
2741 015034 001401
2742 015036 104000
2743 015040 104412 000006
2744 015044 052777 001000 164340
2745 015052 104412 000016
2746 015056 105777 164322

***** TEST 17 *****
: THIS TEST PROVES THAT THE DEVICE WILL
: OUTPUT SIXTEEN SPACES IN FRONT OF A
: FLAG ,F STARTING FROM THE IDLE STATE.
: *****

: TEST 17
: *****

TST17: MOV #17,@TSTNO
MOV #TST20,NEXT
BIS #MRESET,@TXCSR ;RESET THE DEVICE
JSR PC,SMALL ;WAIT FOR RESET TO FINISH
CLR TEMP1 ;CLEAR FOR SOFTWARE STORAGE
CLR TEMP2
CLR TEMP3
CLR @TXDBUF ;RESET TXDONE
BIS #MMODE,@TXCSR ;ENTER MAINT MODE
BIS #CRCEN,@PARCSR ;TURN OFF CRC
BIS #SEND,@TXCSR ;TURN ON TRANSMITTER
BIS #RCVEN,@RXCSR ;TURN ON RECEIVER
BIS #TEOM!T5OM,@TXDBUF ;START SPECIAL SEQUENCE
PKCLK .2 ;SYNC UP DUP
1\$: INC TEMP1 ;UPDATE COUNT
CMP #17.,TEMP1 ;CHECK FOR FINISH
BEQ 2\$;BR IF YES
PKCLK .2 ;POKE CLOCK
BIT #MTDATA,@TXCSR ;CHECK THE DATA
BEQ 1\$;BR IF OK
HLT ;DATA SHOULD BE A 0, WAS A 1
2\$: PKCLK .2 ;POKE CLOCK
BIT #MTDATA,@TXCSR ;CHECK THE DATA
BEQ 3\$;BR IF OK
HLT ;DATA SHOULD BE 0, WAS A ONE
3\$: MOV #125,@TXDBUF ;LOAD DATA CHAR
4\$: PKCLK .2 ;POKE CLOCK
BIT #MTDATA,@TXCSR ;CHECK THE BIT
BNE 5\$;BR IF OK
HLT ;DATA SHOULD BE 1, WAS 0
5\$: INC TEMP2 ;UPDATE FOR FINISH
CMP #6,TEMP2 ;CHECK FOR DONE
BNE 4\$;BR IF MORE TO GO
PKCLK .2 ;POKE CLOCK
BIT #MTDATA,@TXCSR ;CHECK THE DATA
BEQ 6\$;BR IF OK
HLT ;DATA SHOULD BE 0, WAS A 1
6\$: PKCLK .6 ;START OUT DATA CHAR
BIS #TEOM,@TXCSR ;TURN OFF TRANSMITTER
PKCLK .14 ;FINISH
TSTB @RXCSR ;CHECK RECEIVER

```

2747 015062 100401          BMI      7$          ;BR IF OK
2748 015064 104000          HLT                    ;RECEIVER FAILED TO ACCEPT SPECIAL CHAR
2749 015066 117737 164314 001242 7$:  MOVB    @RXDBUF,TEMP3 ;GET THE CHAR
2750 015074 122737 000125 001242    CMPB    #125,TEMP3    ;CHECK IT
2751 015102 001401          BEQ     10$          ;
2752 015104 104000          HLT                    ;DATA FAILED TO MATCH AFTER
2753                                ;SPECIAL SPACES CHARACTER
2754 015106 104400          10$:  SCOPE          ;SCOPE THIS TEST
2755
2756
2757                                ;***** TEST 20 *****
2758                                ;*THIS TEST PROVES THE INTERACTION OF DTR
2759                                ;*WITH RING,DSR
2760                                ;*AND DATA SET CHANGE ONE AND DATA SET CHANGE TWO.
2761                                ;*SET THE BIT AND VERIFY THE OTHER BITS ARE SET. CLEAR
2762                                ;*THE BIT AND VERIFY CLEAR. REPEAT FOR MRESET.
2763                                ;*****
2764
2765                                ;*****
2766                                ;*
2767                                ;* TEST 20
2768                                ;*
2769                                ;*****
2770                                ;*****
2771 015110 012737 000020 001226  TST20:  MOV     #20,@TSTNO
2772 015116 012737 015556 001216    MOV     #TST21,NEXT
2773 015124 012737 015204 001220    MOV     #1$,LOCK
2774 015132 105737 001322          TSTB    TCNFLG
2775 015136 001002          BNE     .+6
2776 015140 000137 015544          10$:  JMP     6$
2777 015144 005077 164234          CLR     @RXCSR          ;CLEAR THE REGISTER
2778 015150 004137 007332          JSR     R1,OJUMPER      ;THIS CALL DETERMINES IF TURNAROUND CONNECTOR
2779 015154 015546          7$          ;AND OPTIONAL JUMPER ARE USED
2780                                ;AND LOADS R5 (EXPECTED) ACCORDINGLY.
2781 015156 052777 000400 164226    BIS     #MRESET,@TXCSR ;RESET THE DEVICE
2782 015164 004737 005044          JSR     PC,SMALL        ;WAIT FOR RESET TO FINISH
2783 015170 013703 001404          MOV     RXCSR,R3        ;LOAD THE RECEIVER CONTROL REGISTER TO R3.
2784 015174 005013          CLR     (R3)           ;CLEAR OUT EXTRA BITS
2785 015176 052777 010000 164206    BIS     #MEXT,@TXCSR    ;ENTER EXTERNAL MAINT. MODE
2786 015204 052713 000002          1$:  BIS     #DTR,(R3)     ;TURN ON DTR
2787 015210 012737 000110 015240    MOV     #110,68$        ;LOAD THE NUMBER
2788 015216 032777 004000 164170    66$:  BIT     #TIMER,@TXDBUF ;CHECK THE TIMER BIT
2789 015224 001374          BNF     66$            ;BR IF SET
2790 015226 032777 004000 164160    67$:  BIT     #TIMER,@TXDBUF ;CHECK THE BIT
2791 015234 001774          BEQ     67$            ;BR IF CLEAR
2792 015236 005327          DEC     (PC)+          ;DECREMENT THE NUMBER
2793 015240 000110          68$:  110            ;OF TIMES TO REPEAT
2794 015242 001365          BNE     66$            ;BR IF MORE TO GO
2795 015244 011304          MOV     (R3),R4        ;GET THE BITS FROM THE RXCSR
2796 015246 020504          CMP     R5,R4          ;R5=GOOD R4=?
2797 015250 001423          BEQ     2$             ;BRANCH IF THEY MATCH
2798 015252 104003          HLT     3              ;NO MATCH - SHOW OPR.
2799 015254 104401          SCOP1
2800 015256 012737 000005 015306    MOV     #5,73$         ;LOAD THE NUMBER
2801 015264 032777 004000 164122    71$:  BIT     #TIMER,@TXDBUF ;CHECK THE TIMER BIT
2802 015272 001374          BNE     71$           ;BR IF SET
    
```

MODEM CONTROL BITS DTR,RING,AND DSR INTERACTION TEST

```

2803 015274 032777 004000 164112 72$: BIT #TIMER,@TXDBUF ;CHECK THE BIT
2804 015302 001774 BEQ 72$ ;BR IF CLEAR
2805 015304 005327 DEC (PC)+ ;DECREMENT THE NUMBER
2806 015306 000005 73$: 5 ;OF TIMES TO REPEAT
2807 015310 001365 BNE 71$ ;BR IF MORE TO GO
2808 015312 032713 040000 BIT #RING,(R3)
2809 015316 001000 BNE 2$
2810 015320 012737 015332 001220 2$: MOV #3$,LOCK ;SW09 SETUP
2811 015326 042705 041002 BIC #RING!DSR!DTR,R5 ;CLEAR OUT UNWANTED BITS
2812 015332 005013 3$: CLR (R3) ;CLEAR OUT THE REGISTER
2813 015334 012737 000005 015364 MOV #5,78$ ;LOAD THE NUMBER
2814 015342 032777 004000 164044 76$: BIT #TIMER,@TXDBUF ;CHECK THE TIMER BIT
2815 015350 001374 BNE 76$ ;BR IF SET
2816 015352 032777 004000 164034 77$: BIT #TIMER,@TXDBUF ;CHECK THE BIT
2817 015360 001774 BEQ 77$ ;BR IF CLEAR
2818 015362 005327 DEC (PC)+ ;DECREMENT THE NUMBER
2819 015364 000005 78$: 5 ;OF TIMES TO REPEAT
2820 015366 001365 BNE 76$ ;BR IF MORE TO GO
2821 015370 011304 MOV (R3),R4 ;READ BACK THE REGISTER
2822 015372 020504 CMP R5,R4 ;R5=GOOD R4=?
2823 015374 001402 BEQ 4$ ;BRANCH IF ONLY THE DSC BITS ARE SET
2824 015376 104003 HLT 3 ;NO-GO TELL OPR
2825 015400 104401 SCOPE1
2826 015402 012737 015410 001220 4$: MOV #5$,LOCK ;SW09 SETUP
2827 015410 052713 000002 5$: BIS #DTR,(R3) ;TURN ON DTR
2828 015414 012737 000005 015444 MOV #5,83$ ;LOAD THE NUMBER
2829 015422 032777 004000 163764 81$: BIT #TIMER,@TXDBUF ;CHECK THE TIMER BIT
2830 015430 001374 BNE 81$ ;BR IF SET
2831 015432 032777 004000 163754 82$: BIT #TIMER,@TXDBUF ;CHECK THE BIT
2832 015440 001774 BEQ 82$ ;BR IF CLEAR
2833 015442 005327 DEC (PC)+ ;DECREMENT THE NUMBER
2834 015444 000005 83$: 5 ;OF TIMES TO REPEAT
2835 015446 001365 BNE 81$ ;BR IF MORE TO GO
2836 015450 005005 CLR R5 ;CLEAR OUT EXPECTED
2837 015452 005013 CLR (R3) ;CLEAR OUT THE REGISTER
2838 015454 052777 000400 163730 BIS #MRESET,@TXCSR ;RESET THE DEVICE
2839 015462 004737 005044 JSR PC,SMALL ;WAIT FOR RESET TO FINISH
2840 015466 052777 010000 163716 BIS #MEXT,@TXCSR ;TURN ON EXTERNAL MODE
2841 015474 012737 000005 015524 MOV #5,88$ ;LOAD THE NUMBER
2842 015502 032777 004000 163704 86$: BIT #TIMER,@TXDBUF ;CHECK THE TIMER BIT
2843 015510 001374 BNE 86$ ;BR IF SET
2844 015512 032777 004000 163674 87$: BIT #TIMER,@TXDBUF ;CHECK THE BIT
2845 015520 001774 BEQ 87$ ;BR IF CLEAR
2846 015522 005327 DEC (PC)+ ;DECREMENT THE NUMBER
2847 015524 000005 88$: 5 ;OF TIMES TO REPEAT
2848 015526 001365 BNE 86$ ;BR IF MORE TO GO
2849 015530 005713 TST (R3) ;STRIP DSCA & DSCB FROM CSR
2850 015532 011304 MOV (R3),R4 ;GET THE REGISTER
2851 015534 020504 CMP R5,R4 ;R5-GOOD,R4=?
2852 015536 001402 BIC 6$ ;BR IF OK
2853 015540 104003 HLT 3 ;REPORT THE ERROR
2854 015542 104401 SCOPE1 ;SW09=1?
2855
2856 015544 104400 6$: SCOPE ;SCOPE THE WHOLE TEST
2857 015546 141003 7$: .WORD 141003
2858 015550 141001 .WORD 141001

```

MODEM CONTROL BITS DTR,RING,AND DSR INTERACTION TEST

2859 015552 001002
2860 015554 000000

.WORD 1002
.WORD 0

2861
2862
2863
2864
2865
2866
2867
2868
2869

```
***** TEST 21 *****  
: *THIS TEST PROVES THE INTERACTION OF RTS  
: *WITH CTS,CARDET  
: *AND DATA SET CHANGE ONE AND DATA SET CHANGE TWO.  
: *SET THE BIT AND VERIFY THE OTHER BITS ARE SET. CLEAR  
: *THE BIT AND VERIFY CLEAR. REPEAT FOR MRESET.  
: *****
```

2870
2871
2872
2873
2874
2875

```
*****  
: *  
: TEST 21  
: *  
: *****
```

2876
2877 015556 012737 000021 001226
2878 015564 012737 016224 001216
2879 015572 012737 015652 001220
2880 015600 105737 001322
2881 015604 001002
2882 015606 000137 016212
2883 015612 005077 163566
2884 015616 004137 007332
2885 015622 016214
2886
2887 015624 052777 000400 163560
2888 015632 004737 005044
2889 015636 013703 001404
2890 015642 005013
2891 015644 052777 010000 163540
2892 015652 052713 000004
2893 015656 012737 000110 015706
2894 015664 032777 004000 163522
2895 015672 001374
2896 015674 032777 004000 163512
2897 015702 001774
2898 015704 005327
2899 015706 000110
2900 015710 001365
2901 015712 011304
2902 015714 020504
2903 015716 001423
2904 015720 104003
2905 015722 104401
2906 015724 012737 000005 015754
2907 015732 032777 004000 163454
2908 015740 001374
2909 015742 032777 004000 163444
2910 015750 001774
2911 015752 005327
2912 015754 000005
2913 015756 001365
2914 015760 032713 040000

```
TST21: MOV #21,@TSTNO  
MOV #TST22,NEXT  
MOV #1$,LOCK  
TSTB TCNFLG  
BNE .+6  
10$: JMP 6$  
CLR @RXCSR ;CLEAR THE REGISTER  
JSR R1,OJUMPER ;THIS CALL DETERMINES IF TURNAROUND CONNECTOR  
7$ ;AND OPTIONAL JUMPER ARE USED  
;AND LOADS R5 (EXPECTED) ACCORDINGLY.  
BIS #MRESET,@TXCSR ;RESET THE DEVICE  
JSR PC,SMALL ;WAIT FOR RESET TO FINISH  
MOV RXCSR,R3 ;LOAD THE RECEIVER CONTROL REGISTER TO R3.  
CLR (R3) ;CLEAR OUT EXTRA BITS  
BIS #MEXT,@TXCSR ;ENTER EXTERNAL MAINT. MODE  
1$: BIS #RTS,(R3) ;TURN ON RTS  
MOV #110,68$ ;LOAD THE NUMBER  
66$: BIT #TIMER,@TXDBUF ;CHECK THE TIMER BIT  
BNE 66$ ;BR IF SET  
67$: BIT #TIMER,@TXDBUF ;CHECK THE BIT  
BEQ 67$ ;BR IF CLEAR  
DEC (PC)+ ;DECREMENT THE NUMBER  
68$: 110 ;OF TIMES TO REPEAT  
BNE 66$ ;BR IF MORE TO GO  
MOV (R3),R4 ;GET THE BITS FROM THE RXCSR  
CMP R5,R4 ;R5=GOOD R4=?  
BEQ 2$ ;BRANCH IF THEY MATCH  
HLT 3 ;NO MATCH - SHOW OPR.  
SCOP1  
MOV #5,73$ ;LOAD THE NUMBER  
71$: BIT #TIMER,@TXDBUF ;CHECK THE TIMER BIT  
BNE 71$ ;BR IF SET  
72$: BIT #TIMER,@TXDBUF ;CHECK THE BIT  
BEQ 72$ ;BR IF CLEAR  
DEC (PC)+ ;DECREMENT THE NUMBER  
73$: 5 ;OF TIMES TO REPEAT  
BNE 71$ ;BR IF MORE TO GO  
BIT #RING,(R3)
```

MODEM CONTROL BITS RTS,CTS,AND CARDET INTERACTION TEST

```
2915 015764 001000      BNE      2$
2916 015766 012737 016000 001220 2$:  MOV     #3$,LOCK      ;SW09 SETUP
2917 015774 042705 030004      BIC     #CTS!CARDET!RTS,R5 ;CLEAR OUT UNWANTED BITS
2918 016000 005013      CLR     (R3)          ;CLEAR OUT THE REGISTER
2919 016002 012737 000005 016032 3$:  MOV     #5,78$       ;LOAD THE NUMBER
2920 016010 032777 004000 163376 76$: BIT     #TIMER,@TXDBUF ;CHECK THE TIMER BIT
2921 016016 001374      BNE     76$          ;BR IF SET
2922 016020 032777 004000 163366 77$: BIT     #TIMER,@TXDBUF ;CHECK THE BIT
2923 016026 001774      BEQ     77$          ;BR IF CLEAR
2924 016030 005327      DEC     (PC)+        ;DECREMENT THE NUMBER
2925 016032 000005      78$:  5              ;OF TIMES TO REPEAT
2926 016034 001365      BNE     76$          ;BR IF MORE TO GO
2927 016036 011304      MOV     (R3),R4      ;READ BACK THE REGISTER
2928 016040 020504      CMP     R5,R4        ;R5=GOOD,R4=?
2929 016042 001402      BEQ     4$           ;BRANCH IF ONLY THE DSC BITS ARE SET
2930 016044 104003      HLT     3            ;NO-GO TELL OPR
2931 016046 104401      SCOP1
2932 016050 012737 016056 001220 4$:  MOV     #5$,LOCK      ;SW09 SETUP
2933 016056 052713 000004 5$:  BIS     #RTS,(R3)    ;TURN ON RTS
2934 016062 012737 000005 016112 5$:  MOV     #5,83$       ;LOAD THE NUMBER
2935 016070 032777 004000 163316 81$: BIT     #TIMER,@TXDBUF ;CHECK THE TIMER BIT
2936 016076 001374      BNE     81$          ;BR IF SET
2937 016100 032777 004000 163306 82$: BIT     #TIMER,@TXDBUF ;CHECK THE BIT
2938 016106 001774      BEQ     82$          ;BR IF CLEAR
2939 016110 005327      DEC     (PC)+        ;DECREMENT THE NUMBER
2940 016112 000005      83$:  5              ;OF TIMES TO REPEAT
2941 016114 001365      BNE     81$          ;BR IF MORE TO GO
2942 016116 005005      CLR     R5           ;CLEAR OUT EXPECTED
2943 016120 005013      CLR     (R3)          ;CLEAR OUT THE REGISTER
2944 016122 052777 000400 163262 84$: BIS     #MRESET,@TXCSR ;RESET THE DEVICE
2945 016130 004737 005044 85$: JSR     PC,SMALL     ;WAIT FOR RESET TO FINISH
2946 016134 052777 010000 163250 85$: BIS     #MEXT,@TXCSR  ;TURN ON EXTERNAL MODE
2947 016142 012737 000005 016172 85$: MOV     #5,88$       ;LOAD THE NUMBER
2948 016150 032777 004000 163236 86$: BIT     #TIMER,@TXDBUF ;CHECK THE TIMER BIT
2949 016156 001374      BNE     86$          ;BR IF SET
2950 016160 032777 004000 163226 87$: BIT     #TIMER,@TXDBUF ;CHECK THE BIT
2951 016166 001774      BEQ     87$          ;BR IF CLEAR
2952 016170 005327      DEC     (PC)+        ;DECREMENT THE NUMBER
2953 016172 000005      88$:  5              ;OF TIMES TO REPEAT
2954 016174 001365      BNE     86$          ;BR IF MORE TO GO
2955 016176 005713      TST     (R3)          ;STRIP DSCA & DSCB FROM CSR
2956 016200 011304      MOV     (R3),R4      ;GET THE REGISTER
2957 016202 020504      CMP     R5,R4        ;R5=GOOD,R4=?
2958 016204 001402      BEQ     6$           ;BR IF OK
2959 016206 104003      HLT     3            ;REPORT THE ERROR
2960 016210 104401      SCOP1
2961
2962 016212 104400      6$:  SCOPE          ;SCOPE THE WHOLE TEST
2963 016214 130005      7$:  .WORD 130005
2964 016216 130001      .WORD 130001
2965 016220 000004      .WORD 4
2966 016222 000000      .WORD 0
2967
2968
2969
2970
```

```
***** TEST 22 *****  
*THIS TEST PROVES THE INTERACTION OF STD
```

MODEM CONTROL BITS STD AND SRD INTERACTION TEST

```
2971 ;*WITH STD,SRD
2972 ;*AND DATA SET CHANGE ONE AND DATA SET CHANGE TWO.
2973 ;*SET THE BIT AND VERIFY THE OTHER BITS ARE SET. CLEAR
2974 ;*THE BIT AND VERIFY CLEAR. REPEAT FOR MRESET.
2975 ;:*****
2976
2977 ;:*****
2978 ;*
2979 ;: TEST 22
2980 ;*
2981 ;:*****
2982 ;:*****
2983 016224 012737 000022 001226 TST22: MOV #22,@TSTNO
2984 016232 012737 016706 001216 MOV #TST23,NEXT
2985 016240 012737 016334 001220 MOV #1$,LOCK
2986 016246 105737 001322 TSTB TCNFLG
2987 016252 001002 BNE .+6
2988 016254 000137 016674 10$: JMP 6$
2989 016260 105737 001334 TSTB STJMFL
2990 016264 001773 BEQ 10$
2991 016266 105737 001336 TSTB SRJMFL
2992 016272 001770 BEQ 10$
2993 016274 005077 163104 CLR @RXCSR ;CLEAR THE REGISTER
2994 016300 004137 007332 JSR R1,OJUMPER ;THIS CALL DETERMINES IF TURNAROUND CONNECTOR
2995 016304 016676 7$ ;AND OPTIONAL JUMPER ARE USED
2996 ;AND LOADS R5 (EXPECTED) ACCORDINGLY.
2997 016306 052777 000400 163076 BIS #MRESET,@TXCSR ;RESET THE DEVICE
2998 016314 004737 005044 JSR PC,SMALL ;WAIT FOR RESET TO FINISH
2999 016320 013703 001404 MOV RXCSR,R3 ;LOAD THE RECEIVER CONTROL REGISTER TO R3.
3000 016324 005013 CLR (R3) ;CLEAR OUT EXTRA BITS
3001 016326 052777 010000 163056 BIS #MEXT,@TXCSR ;ENTER EXTERNAL MAINT. MODE
3002 016334 052713 000010 1$: BIS #STD,(R3) ;TURN ON STD
3003 016340 012737 000110 016370 MOV #110,68$ ;LOAD THE NUMBER
3004 016346 032777 004000 163040 66$: BIT #TIMER,@TXDBUF ;CHECK THE TIMER BIT
3005 016354 001374 BNE 66$ ;BR IF SET
3006 016356 032777 004000 163030 67$: BIT #TIMER,@TXDBUF ;CHECK THE BIT
3007 016364 001774 BEQ 67$ ;BR IF CLEAR
3008 016366 005327 DEC (PC)+ ;DECREMENT THE NUMBER
3009 016370 000110 68$: 110 ;OF TIMES TO REPEAT
3010 016372 001365 BNE 66$ ;BR IF MORE TO GO
3011 016374 011304 MOV (R3),R4 ;GET THE BITS FROM THE RXCSR
3012 016376 020504 CMP R5,R4 ;R5=GOOD R4=?
3013 016400 001423 BEQ 2$ ;BRANCH IF THEY MATCH
3014 016402 104003 HLT 3 ;NO MATCH - SHOW OPR.
3015 016404 104401 SCOP1
3016 016406 012737 000005 016436 MOV #5,73$ ;LOAD THE NUMBER
3017 016414 032777 004000 162772 71$: BIT #TIMER,@TXDBUF ;CHECK THE TIMER BIT
3018 016422 001374 BNE 71$ ;BR IF SET
3019 016424 032777 004000 162762 72$: BIT #TIMER,@TXDBUF ;CHECK THE BIT
3020 016432 001774 BEQ 72$ ;BR IF CLEAR
3021 016434 005327 DEC (PC)+ ;DECREMENT THE NUMBER
3022 016436 000005 73$: 5 ;OF TIMES TO REPEAT
3023 016440 001365 BNE 71$ ;BR IF MORE TO GO
3024 016442 032713 040000 BIT #RING,(R3)
3025 016446 001000 BNE 2$
3026 016450 012737 016462 001220 2$: MOV #3$,LOCK ;SW09 SETUP
```

MODEM CONTROL BITS STD AND SRD INTERACTION TEST

```
3027 016456 042705 002010          BIC      #STD!SRD,R5          ;CLEAR OUT UNWANTED BITS
3028 016462 005013          CLR      (R3)                ;CLEAR OUT THE REGISTER
3029 016464 012737 000005 016514    MOV      #5,78$              ;LOAD THE NUMBER
3030 016472 032777 004000 162714    BIT      #TIMER,@TXDBUF      ;CHECK THE TIMER BIT
3031 016500 001374          BNE      76$                  ;BR IF SET
3032 016502 032777 004000 162704    BIT      #TIMER,@TXDBUF      ;CHECK THE BIT
3033 016510 001774          BEQ      77$                  ;BR IF CLEAR
3034 016512 005327          DEC      (PC)+                ;DECREMENT THE NUMBER
3035 016514 000005          78$: 5                        ;OF TIMES TO REPEAT
3036 016516 001365          BNE      76$                  ;BR IF MORE TO GO
3037 016520 011304          MOV      (R3),R4              ;READ BACK THE REGISTER
3038 016522 020504          CMP      R5,R4                ;R5=GOOD R4=?
3039 016524 001402          BEQ      4$                    ;BRANCH IF ONLY THE DSC BITS ARE SET
3040 016526 104003          HLT      3                      ;NO-GO TELL OPR
3041 016530 104401          SCOPE1
3042 016532 012737 016540 001220    4$: MOV      #5$,LOCK          ;SW09 SETUP
3043 016540 052713 000010          5$: BIS      #STD,(R3)          ;TURN ON STD
3044 016544 012737 000005 016574    MOV      #5,83$              ;LOAD THE NUMBER
3045 016552 032777 004000 162634    81$: BIT      #TIMER,@TXDBUF      ;CHECK THE TIMER BIT
3046 016560 001374          BNE      81$                  ;BR IF SET
3047 016562 032777 004000 162624    82$: BIT      #TIMER,@TXDBUF      ;CHECK THE BIT
3048 016570 001774          BEQ      82$                  ;BR IF CLEAR
3049 016572 005327          DEC      (PC)+                ;DECREMENT THE NUMBER
3050 016574 000005          83$: 5                        ;OF TIMES TO REPEAT
3051 016576 001365          BNE      81$                  ;BR IF MORE TO GO
3052 016600 005005          CLR      R5                    ;CLEAR OUT EXPECTED
3053 016602 005013          CLR      (R3)                ;CLEAR OUT THE REGISTER
3054 016604 052777 000400 162600    BIS      #MRESET,@TXCSR      ;RESET THE DEVICE
3055 016612 004737 005044          JSR      PC,SMALL              ;WAIT FOR RESET TO FINISH
3056 016616 052777 010000 162566    BIS      #MEXT,@TXCSR        ;TURN ON EXTERNAL MODE
3057 016624 012737 000005 016654    MOV      #5,88$              ;LOAD THE NUMBER
3058 016632 032777 004000 162554    86$: BIT      #TIMER,@TXDBUF      ;CHECK THE TIMER BIT
3059 016640 001374          BNE      86$                  ;BR IF SET
3060 016642 032777 004000 162544    87$: BIT      #TIMER,@TXDBUF      ;CHECK THE BIT
3061 016650 001774          BEQ      87$                  ;BR IF CLEAR
3062 016652 005327          DEC      (PC)+                ;DECREMENT THE NUMBER
3063 016654 000005          88$: 5                        ;OF TIMES TO REPEAT
3064 016656 001365          BNE      86$                  ;BR IF MORE TO GO
3065 016660 005713          TST      (R3)                  ;STRIP DSCA & DSCB FROM CSR
3066 016662 011304          MOV      (R3),R4              ;GET THE REGISTER
3067 016664 020504          CMP      R5,R4                ;R5=GOOD,R4=?
3068 016666 001402          BEQ      6$                    ;BR IF OK
3069 016670 104003          HLT      3                      ;REPORT THE ERROR
3070 016672 104401          SCOPE1                          ;SW09=1?
3071
3072 016674 104400          6$: SCOPE                          ;SCOPE THE WHOLE TEST
3073 016676 002011          7$: .WORD 2011
3074 016700 002001          .WORD 2001
3075 016702 000010          .WORD 10
3076 016704 000000          .WORD 0
```

3077
3078
3079
3080
3081
3082

```
***** TEST 23 *****  
*THIS TEST PROVES THE INTERACTION OF DTR!RTS!STD  
*WITH RING,DSR,CTS,CARDET,STD,SRD  
*AND DATA SET CHANGE ONE AND DATA SET CHANGE TWO.
```

ALL MODEM CONTROL BITS INTERACTION TEST

3083
3084
3085
3086
3087
3088
3089
3090
3091
3092
3093
3094
3095
3096
3097
3098
3099
3100
3101
3102
3103
3104
3105
3106
3107
3108
3109
3110
3111
3112
3113
3114
3115
3116
3117
3118
3119
3120
3121
3122
3123
3124
3125
3126
3127
3128
3129
3130
3131
3132
3133
3134
3135
3136
3137
3138

016706 012737 000023 001226
016714 012737 017370 001216
016722 012737 017016 001220
016730 105737 001322
016734 001002
016736 000137 017356
016742 105737 001334
016746 001773
016750 105737 001336
016754 001770
016756 005077 162422
016762 004137 007332
016766 017360
016770 052777 000400 162414
016776 004737 005044
017002 013703 001404
017006 005013
017010 052777 010000 162374
017016 052713 000016
017022 012737 000110 017052
017030 032777 004000 162356
017036 001374
017040 032777 004000 162346
017046 001774
017050 005327
017052 000110
017054 001365
017056 011304
017060 020504
017062 001423
017064 104003
017066 104401
017070 012737 000005 017120
017076 032777 004000 162310
017104 001374
017106 032777 004000 162300
017114 001774
017116 005327
017120 000005
017122 001365
017124 032713 040000
017130 001000
017132 012737 017144 001220
017140 042705 073016
017144 005013

```

; *SET THE BIT AND VERIFY THE OTHER BITS ARE SET. CLEAR
; *THE BIT AND VERIFY CLEAR. REPEAT FOR MRESET.
; :*****
; :*****
; : TEST 23
; :*****
; :*****
TST23:  MOV    #23,@TSTNO
        MOV    #TST24,NEXT
        MOV    #1$,LOCK
        TSTB   TCNFLG
        BNE    .+6
10$:    JMP    6$
        TSTB   STJMFL
        BEQ    10$
        TSTB   SRJMFL
        BEQ    10$
        CLR    @RXCSR          ;CLEAR THE REGISTER
        JSR    R1,OJUMPER     ;THIS CALL DETERMINES IF TURNAROUND CUNNECTOR
        7$                  ;AND OPTIONAL JUMPER ARE USED
                               ;AND LOADS R5 (EXPECTED) ACCORDINGLY.
        BIS    #MRESET,@TXCSR ;RESET THE DEVICE
        JSR    PC,SMALL      ;WAIT FOR RESET TO FINISH
        MOV    RXCSR,R3     ;LOAD THE RECEIVER CONTROL REGISTER TO R3.
        CLR    (R3)         ;CLEAR OUT EXTRA BITS
        BIS    #MEXT,@TXCSR ;ENTER EXTERNAL MAINT. MODE
        1$:    BIS    #DTR!RTS!STD,(R3) ;TURN ON DTR!RTS!STD
        MOV    #110,68$     ;LOAD THE NUMBER
66$:    BIT    #TIMER,@TXDBUF ;CHECK THE TIMER BIT
        BNE    66$         ;BR IF SET
67$:    BIT    #TIMER,@TXDBUF ;CHECK THE BIT
        BEQ    67$         ;BR IF CLEAR
        DEC    (PC)+       ;DECREMENT THE NUMBER
68$:    110                ;OF TIMES TO REPEAT
        BNE    66$         ;BR IF MORE TO GO
        MOV    (R3),R4     ;GET THE BITS FROM THE RXCSR
        CMP    R5,R4      ;R5=GOOD R4=?
        BEQ    2$         ;BRANCH IF THEY MATCH
        HLT    3          ;NO MATCH - SHOW OPR.
        SCOP1
        MOV    #5,73$     ;LOAD THE NUMBER
71$:    BIT    #TIMER,@TXDBUF ;CHECK THE TIMER BIT
        BNE    71$         ;BR IF SET
72$:    BIT    #TIMER,@TXDBUF ;CHECK THE BIT
        BEQ    72$         ;BR IF CLEAR
        DEC    (PC)+       ;DECREMENT THE NUMBER
73$:    5                  ;OF TIMES TO REPEAT
        BNE    71$         ;BR IF MORE TO GO
        BNE    71$
        BIT    #RING,(R3)
        BNE    2$
        MOV    #3$,LOCK   ;SW09 SETUP
        BIC    #RING!CTS!CARDET!SRD!DSR!STD!RTS!DTR,R5 ;CLEAR OUT UNWANTED BITS
        3$:    CLR    (R3) ;CLEAR OUT THE REGISTER

```


ALL MODEM CONTROL BITS INTERACTION TEST

```
3139 017146 012737 000005 017176      MOV    #5,78$      ;LOAD THE NUMBER
3140 017154 032777 004000 162232 76$:   BIT    #TIMER,@TXDBUF ;CHECK THE TIMER BIT
3141 017162 001374          BNE    76$         ;BR IF SET
3142 017164 032777 004000 162222 77$:   BIT    #TIMER,@TXDBUF ;CHECK THE BIT
3143 017172 001774          BEQ    77$         ;BR IF CLEAR
3144 017174 005327          DEC    (PC)+       ;DECREMENT THE NUMBER
3145 017176 000005          5         ;OF TIMES TO REPEAT
3146 017200 001365          BNE    76$         ;BR IF MORE TO GO
3147 017202 011304          MOV    (R3),R4     ;READ BACK THE REGISTER
3148 017204 020504          CMP    R5,R4       ;R5=GOOD R4=?
3149 017206 001402          BEQ    4$          ;BRANCH IF ONLY THE DSC BITS ARE SET
3150 017210 104003          HLT    3           ;NO-GO TELL OPR
3151 017212 104401          SCOPI
3152 017214 012737 017222 001220 4$:   MOV    #5$,LOCK    ;SW09 SETUP
3153 017222 052713 000016 5$:   BIS    #DTR!RTS!STD,(R3) ;TURN ON DTR.RTS!STD
3154 017226 012737 000005 017256      MOV    #5,83$      ;LOAD THE NUMBER
3155 017234 032777 004000 162152 81$:   BIT    #TIMER,@TXDBUF ;CHECK THE TIMER BIT
3156 017242 001374          BNE    81$         ;BR IF SET
3157 017244 032777 004000 162142 82$:   BIT    #TIMER,@TXDBUF ;CHECK THE BIT
3158 017252 001774          BEQ    82$         ;BR IF CLEAR
3159 017254 005327          DEC    (PC)+       ;DECREMENT THE NUMBER
3160 017256 000005          5         ;OF TIMES TO REPEAT
3161 017260 001365          BNE    81$         ;BR IF MORE TO GO
3162 017262 005005          CLR    R5          ;CLEAR OUT EXPECTED
3163 017264 005013          CLR    (R3)        ;CLEAR OUT THE REGISTER
3164 017266 052777 000400 162116      BIS    #MRESET,@TXCSR ;RESET THE DEVICE
3165 017274 004737 005044          JSR    PC,SMALL    ;WAIT FOR RESET TO FINISH
3166 017300 052777 010000 162104      BIS    #MEXT,@TXCSR ;TURN ON EXTERNAL MODE
3167 017306 012737 000005 017336      MOV    #5,88$      ;LOAD THE NUMBER
3168 017314 032777 004000 162072 86$:   BIT    #TIMER,@TXDBUF ;CHECK THE TIMER BIT
3169 017322 001374          BNE    86$         ;BR IF SET
3170 017324 032777 004000 162062 87$:   BIT    #TIMER,@TXDBUF ;CHECK THE BIT
3171 017332 001774          BEQ    87$         ;BR IF CLEAR
3172 017334 005327          DEC    (PC)+       ;DECREMENT THE NUMBER
3173 017336 000005          5         ;OF TIMES TO REPEAT
3174 017340 001365          BNE    86$         ;BR IF MORE TO GO
3175 017342 005713          TST    (R3)        ;STRIP DSCA & DSCB FROM CSR
3176 017344 011304          MOV    (R3),R4     ;GET THE REGISTER
3177 017346 020504          CMP    R5,R4       ;R5=GOOD,R4=?
3178 017350 001402          BEQ    6$          ;BR IF OK
3179 017352 104003          HLT    3           ;REPORT THE ERROR
3180 017354 104401          SCOPI
3181
3182 017356 104400          6$:   SCOPE          ;SCOPE THE WHOLE TEST
3183 017360 173017          7$:   .WORD    173017
3184 017362 173001          .WORD    173001
3185 017364 001016          .WORD    1016
3186 017366 000000          .WORD    0
3187
3188          ;***** TEST 24 *****
3189          ;*TEST THAT SETTING TRANSMIT INTERRUPT
3190          ;*ENABLE AND TRANSMITTER DONE PRODUCE
3191          ;*AN INTERRUPT ON THE TRANSMITTER VECTOR.
3192          ;*****
3193
3194          ;*****
```

3195
3196
3197
3198
3199
3200 017370 012737 000024 001226
3201 017376 012737 017506 001216
3202 017404 012706 001150
3203 017410 000005
3204 017412 012737 000340 177776
3205 017420 052777 000400 161764
3206 017426 004737 005044
3207 017432 004537 007232
3208 017436 007254
3209 017440 017474
3210 017442 340 340
3211 017444 052777 004000 161740
3212 017452 005037 177776
3213 017456 052777 000100 161726
3214 017464 000240
3215 017466 000240
3216 017470 000240
3217 017472 104027
3218 017474 012706 001150
3219 017500 005077 161706
3220 017504 104400
3221
3222
3223
3224
3225
3226
3227
3228
3229
3230
3231
3232
3233
3234 017506 012737 000025 001226
3235 017514 012737 017700 001216
3236 017522 012737 000340 177776
3237 017530 052777 000400 161654
3238 017536 004737 005044
3239 017542 004537 007232
3240 017546 007254
3241 017550 017650
3242 017552 340 340
3243 017554 052777 004000 161630
3244 017562 052777 000100 161622
3245 017570 005037 177776
3246 017574 000240
3247 017576 000240
3248 017600 000240
3249 017602 104027
3250 017604 000411

```

:
: TEST 24
:
:*****
:*****
TST24: MOV #24,@TSTNO
MOV #TST25,NEXT
MOV #STACK,SP
RESET
MOV #340,PS ;LOCK OUT INTERRUPTS
BIS #MRESET,@TXCSR ;RESET THE DEVICE
JSR PC,SMALL ;WAIT FOR RESET TO FINISH
JSR R5,SETVEC ;SET UP THE VECTORS
NO.ATRAP ;VECTOR 'A'
1$ ;VECTOR B
.BYTE 340,340 ;PRIORITY
BIS #SYSTST,@TXCSR
CLR PS ;ZERO CPU PRIORITY
BIS #TXINTE,@TXCSR ;TURN ON TXINT ENABLE
NOP ;STALL
NOP ;DITTO
NOP ;DITTO
HLT 27 ;DUP FAILED TO INTERRUPT
1$: MOV #STACK,SP ;RESET THE STACK
CLR @TXCSR ;DISABLE DUP11
SCOPE ;SCOPE THIS TEST
:*****
:***** TEST 25 *****
:
: *TEST TO VERIFY THAT A TRANSMITTER DONE
: *INTERRUPT WILL ONLY OCCUR ONCE IF THE
: *TXCSR AND TXDBUF ARE *NOT* READ OR WRITTEN.
:*****
:*****
:
: TEST 25
:
:*****
:*****
TST25: MOV #25,@TSTNO
MOV #TST26,NEXT
MOV #340,PS ;LOCK OUT INTERRUPTS
BIS #MRESET,@TXCSR ;RESET THE DEVICE
JSR PC,SMALL ;WAIT FOR RESET TO FINISH
JSR R5,SETVEC ;SETUP FOR INTERRUPTS
NO.ATRAP ;RECEIVER
1$ ;TRANSMITTER
.BYTE 340,340 ;LEVEL
BIS #SYSTST,@TXCSR ;TURN ON CLOCK
BIS #TXINTE,@TXCSR ;TURN ON INT. ENABLE
CLR PS ;LOWER PROCESSOR STATUS
NOP ;STALL
NOP ;DITTO
NOP ;DITTO
HLT 27 ;DUP FAILED TO INTERRUPT
BR 5$ ;LEAVE TEST

```

ONLY ONE INTERRUPT PER TXDONE TEST

```

3251 017606 005037 177776      2$: CLR PS ;LOWER PROCESSOR STATUS
3252 017612 000240 ;NOP ;STALL
3253 017614 000240 ;NOP ;DITTO
3254 017616 000240 ;NOP ;DITTO
3255 017620 105777 161566      4$: TSTB @TXCSR ;CHECK THE DONE BIT
3256 017624 100401 ;BMI 5$ ;BR IF SET
3257 017626 104024 ;HLT 24 ;DONE IS CLEARED AND SHOULDN'T BE
3258 017630 012737 000340 177776 5$: MOV #340,PS ;RAISE PROCESSOR STATUS
3259 017636 005077 161550 ;CLR @TXCSR ;CLEAR OUT DUP
3260 017642 012706 001150 ;MOV #STACK,SP ;RESET STACK
3261 017646 104400 ;SCOPE ;SCOPE THIS TEST
3262
3263 017650 012716 017606      1$: MOV #2$, (SP) ;SET UP SECOND PART OF TEST
3264 017654 004537 007232 ;JSR R5,SETVEC ;SETUP FOR SECOND INTERRUPT TRY
3265 017660 007254 ;NO.ATRAP ;RECEIVER
3266 017662 017670 ;3$ ;TRANSMITTER
3267 017664 340 340 ;.BYTE 340,340 ;LEVEL
3268 017666 000002 ;RTI ;RETURN
3269 017670
3270 017670 104026 ;HLT 26 ;REPORT THE FACT YOU GOT HERE
3271 017672 012716 017620 ;MOV #4$, (SP) ;SETUP FOR END OF TEST
3272 017676 000002 ;RTI ;RETURN
3273
3274 ;***** TEST 26 *****
3275 ;*TEST THAT SETTING DATA SET INTERRUPT
3276 ;*ENABLE AND RECEIVING A DATA SET
3277 ;*CHANGE 1 OR DATA SET CHANGE 2
3278 ;*PRODUCES AN INTERRUPT TO THE
3279 ;*RECEIVER VECTOR
3280 ;*****
3281
3282 ;*****
3283 ;*
3284 ;* TEST 26
3285 ;*
3286 ;*****
3287 ;*****
3288 017700 012737 000026 001226 TST26: MOV #26,@TSTNO
3289 017706 012737 020216 001216 ;MOV #TST27,NEXT
3290 017714 105737 001322 ;TSTB TCNFLG
3291 017720 001002 ;BNE .+6
3292 017722 000137 020204 ;JMP 5$
3293 017726 012737 017774 001220 ;MOV #1$,LOCK ;SW09 SETUP
3294 017734 012737 000340 177776 ;MOV #340,PS ;LOCK OUT INTERRUPTS
3295 017742 052777 000400 161442 ;BIS #MRESET,@TXCSR ;RESET THE DEVICE
3296 017750 004737 005044 ;JSR PC,SMALL ;WAIT FOR RESET TO FINISH
3297 017754 052777 010000 161430 ;BIS #MEXT,@TXCSR ;ENTER MAINT EXTERNAL MODE
3298 017762 004537 007232 ;JSR R5,SETVEC ;SET UP VECTORS
3299 017766 020054 ;2$ ;RECEIVER
3300 017770 007260 ;NO.BTRAP ;TRANSMITTER
3301 017772 340 340 ;.BYTE 340,340 ;PRORITY AT 7
3302 017774 005037 177776      1$: CLR PS ;LOWER PS
3303 020000 052777 000040 161376 ;BIS #DSINTE,@RXCSR ;TURN ON INT. ENABLE
3304 020006 052777 000004 161370 ;BIS #RTS,@RXCSR ;TURN ON INT. BIT
3305 020014 012737 000005 020044 ;MOV #5,68$ ;LOAD THE NUMBER
3306 020022 032777 004000 161364 66$: BIT #TIMER,@TXDBUF ;CHECK THE TIMER BIT

```

RECEIVER DATA SET CHANGE BITS INTERRUPT TEST

```
3307 020030 001374      BNE      66$      ;BR IF SET
3308 020032 032777 004000 161354 67$:  BIT      #TIMER,@TXDBUF ;CHECK THE BIT
3309 020040 001774      BEQ      67$      ;BR IF CLEAR
3310 020042 005327      DEC      (PC)+    ;DECREMENT THE NUMBER
3311 020044 000005      5          ;OF TIMES TO REPEAT
3312 020046 001365      BNE      66$      ;BR IF MORE TO GO
3313 020050 104022      HLT      22      ;FAILED TO INTERRUPT
3314 020052 104401      SCOPI          ;SW09=1
3315 020054 012706 001150 2$:  MOV      #STACK,SP ;RESET THE STACK
3316 020060 005077 161320      CLR      @RXCSR   ;CLEAR OUT RECEIVER CONTRL REGISTER.
3317 020064 012737 020104 001220      MOV      #3$,LOCK ;SW09 SETUP
3318 020072 004537 007232      JSR      R5,SETVEC ;SET THE VECTORS
3319 020076 020172      4$          ;RECEIVER
3320 020100 007260      NO.BTRAP        ;TRANSMITTERS
3321 020102      340      340      .BYTE      340,340 ;PRIOR @7
3322 020104 012737 000005 001236 3$:  MOV      #5.,TEMP1 ;LOAD TEMP1
3323 020112 052777 000040 161264      BIS      #DSINTE,@RXCSR ;TURN ON INT. ENABLE
3324 020120 005037 177776      CLR      PS      ;LOWER CPU STATUS
3325 020124 052777 000002 161252      BIS      #DTR,@RXCSR ;PUSH OUT INT. BITS
3326 020132 012737 000156 020162      MOV      #110.,73$ ;LOAD THE NUMBER
3327 020140 032777 004000 161246 71$:  BIT      #TIMER,@TXDBUF ;CHECK THE TIMER BIT
3328 020146 001374      BNE      71$      ;BR IF SET
3329 020150 032777 004000 161236 72$:  BIT      #TIMER,@TXDBUF ;CHECK THE BIT
3330 020156 001774      BEQ      72$      ;BR IF CLEAR
3331 020160 005327      DEC      (PC)+    ;DECREMENT THE NUMBER
3332 020162 000156      110.        ;OF TIMES TO REPEAT
3333 020164 001365      BNE      71$      ;BR IF MORE TO GO
3334 020166 104022      HLT      22      ;FAILED TO INTERRUPT
3335 020170 104401      SCOPI          ;BIT09=1?
3336 020172      4$:
3337 020172 032777 100000 161204      BIT      #BIT15,@RXCSR ;CHECK FOR INTERRUPT FROM DSC1
3338 020200 001001      BNE      5$      ;IT CAME FROM DSC1
3339 020202 104023      HLT      23      ;BIT15 IS CLEARED - INTERRUPTED
3340      ;FROM THE WRONG DSC BIT.
3341 020204 005077 161174 5$:  CLR      @RXCSR   ;CLEAR OUT RECEIVER CONTROL REG
3342 020210 012706 001150      MOV      #STACK,SP ;RESET THE STACK
3343 020214 104400      SCOPE          ;SCOPE THIS TEST
3344      ;***** TEST 27 *****
3345      ;*TEST THAT SETTING RECEIVER INTERRUPT
3346      ;*ENABLE AND RECEIVER DONE CAUSES AN
3347      ;*INTERRUPT TO THE RECEIVER VECTOR
3348      ;*****
3349
3350      ;*****
3351      ;*
3352      ;* TEST 27
3353      ;*
3354      ;*****
3355      ;*****
3356 020216 012737 000027 001226 TST27: MOV      #27,@TSTNO
3357 020224 012737 020476 001216      MOV      #TST30,NEXT
3358 020232 012737 000340 177776      MOV      #340,PS   ;LOCK OUT INTERRUPTS
3359 020240 052777 000400 161144      BIS      #MRESET,@TXCSR ;RESET THE DEVICE
3360 020246 004737 005044      JSR      PC,SMALL  ;WAIT FOR RESET TO FINISH
3361 020252 052777 004000 161132      BIS      #SYSTST,@TXCSR ;ENTER SYSTEM TEST MODE
3362 020260 004537 007232      JSR      R5,SETVEC ;SET UP VECTORS
```

```

3363 020264 020460 3$ :RECEIVER
3364 020266 007260 NO.BTRAP :TRANSMITTER
3365 020270 340 340 .BYTE 340,340 :PRIORITY AT 7
3366 020277 005037 177776 CLR PS :LOWER PS
3367 020276 052777 000020 161100 BIS #RCVEN,@RXCSR :TURN ON RECEIVER
3368 020304 052777 000100 161072 BIS #RINTEN,@RXCSR :TURN ON INT. ENABLE
3369 020312 052777 000020 161072 BIS #SEND,@TXCSR :TURN ON TRANSMITTER
3370 020320 1$:
3371 020320 012737 000005 020350 MOV #5,68$ :LOAD THE NUMBER
3372 020326 032777 004000 161060 66$: BIT #TIMER,@TXDBUF :CHECK THE TIMER BIT
3373 020334 001374 BNE 66$ :BR IF SET
3374 020336 032777 004000 161050 67$: BIT #TIMER,@TXDBUF :CHECK THE BIT
3375 020344 001774 BEQ 67$ :BR IF CLEAR
3376 020346 005327 DEC (PC)+ :DECREMENT THE NUMBER
3377 020350 000005 68$: 5 :OF TIMES TO REPEAT
3378 020352 001365 BNE 66$ :BR IF MORE TO GO
3379 020354 032777 000200 161030 BIT #TXDONE,@TXCSR :TEST TXDONE
3380 020362 001001 BNE 2$ :BR IF SET
3381 020364 104024 HLT 24 :TXDONE FAILED TO SET
3382 020366 2$:
3383 020366 012777 000400 161020 MOV #400,@TXDBUF :LOAD TX BUFFER
3384 020374 105777 161012 TSTB @TXCSR :CHECK FOR
3385 020400 100375 BPL -4 :DONE
3386 020402 005077 161006 CLR @TXDBUF :CLEAR TX BUFFER
3387 020406 105777 161000 TSTB @TXCSR :AND CHECK
3388 020412 100375 BPL -4 :FOR DONE
3389 020414 012777 001000 160772 MOV #1000,@TXDBUF :LOAD END OF MSG
3390 020422 012737 000050 020452 MOV #40,73$ :LOAD THE NUMBER
3391 020430 032777 004000 160756 71$: BIT #TIMER,@TXDBUF :CHECK THE TIMER BIT
3392 020436 001374 BNE 71$ :BR IF SET
3393 020440 032777 004000 160746 72$: BIT #TIMER,@TXDBUF :CHECK THE BIT
3394 020446 001774 BEQ 72$ :BR IF CLEAR
3395 020450 005327 DEC (PC)+ :DECREMENT THE NUMBER
3396 020452 000050 73$: 40. :OF TIMES TO REPEAT
3397 020454 001365 BNE 71$ :BR IF MORE TO GO
3398 020456 104022 HLT 22 :RECEIVER FAILED TO INTERRUPT
3399 020460 012706 001150 3$: MOV #STACK,SP :RESET STACK
3400 020464 005077 160714 CLR @RXCSR :CLEAR OUT REGISTER
3401 020470 005077 160716 CLR @TXCSR :DITTO
3402 020474 104400 SCOPE :SCOPE THIS TEST

```

```

3403
3404 :***** TEST 30 *****
3405 :*TEST TO VERIFY THAT A RECEIVER DONE
3406 :*INTERRUPT WILL ONLY OCCUR ONCE IF THE
3407 :*RXCSR AND RXDBUF ARE NOT READ OR WRITTEN
3408 :*****
3409

```

```

3410 :*****
3411 :*
3412 :* TEST 30
3413 :*
3414 :*****
3415 :*****

```

```

3416 020476 012737 000030 001226 TST30: MOV #30,@TSTNO
3417 020504 012737 021056 001216 MOV #TST31,NEXT
3418 020512 012737 000340 177776 MOV #340,PS ;LOCK OUT INTERRUPTS

```

| | | | | | | | | |
|------|--------|--------|--------|--------|-------|----------|----------------|----------------------------------|
| 3419 | 020520 | 052777 | 000400 | 160664 | | BIS | #MRESET,@TXCSR | :RESET THE DEVICE |
| 3420 | 020526 | 004737 | 005044 | | | JSR | PC,SMALL | :WAIT FOR RESET TO FINISH |
| 3421 | 020532 | 052777 | 004000 | 160652 | | BIS | #SYSTST,@TXCSR | :ENTER SYSTST MODE |
| 3422 | 020540 | 004537 | 007232 | | | JSR | R5,SETVEC | :SETUP VECTORS |
| 3423 | 020544 | 021026 | | | | 6\$ | | :RECEIVER VECTOR |
| 3424 | 020546 | 007260 | | | | NO.BTRAP | | :TRANSMITTER VECTOR |
| 3425 | 020550 | 340 | 340 | | | .BYTE | 340,340 | :LEVEL |
| 3426 | 020552 | 052777 | 000020 | 160624 | | BIS | #RCVEN,@RXCSR | :TURN ON RECEIVER |
| 3427 | 020560 | 052777 | 000100 | 160616 | | BIS | #RINTEN,@RXCSR | :TURN ON INT. ENABLE |
| 3428 | 020566 | 052777 | 000020 | 160616 | | BIS | #SEND,@TXCSR | :TURN ON TRANSMITTER |
| 3429 | 020574 | | | | 1\$: | | | |
| 3430 | 020574 | 012737 | 000005 | 020624 | | MOV | #5,68\$ | :LOAD THE NUMBER |
| 3431 | 020602 | 032777 | 004000 | 160604 | 66\$: | BIT | #TIMER,@TXDBUF | :CHECK THE TIMER BIT |
| 3432 | 020610 | 001374 | | | | BNE | 66\$ | :BR IF SET |
| 3433 | 020612 | 032777 | 004000 | 160574 | 67\$: | BIT | #TIMER,@TXDBUF | :CHECK THE BIT |
| 3434 | 020620 | 001774 | | | | BEQ | 67\$ | :BR IF CLEAR |
| 3435 | 020622 | 005327 | | | | DEC | (PC)+ | :DECREMENT THE NUMBER |
| 3436 | 020624 | 000005 | | | 68\$: | 5 | | :OF TIMES TO REPEAT |
| 3437 | 020626 | 001365 | | | | BNE | 66\$ | :BR IF MORE TO GO |
| 3438 | 020630 | 032777 | 000200 | 160554 | | BIT | #TXDONE,@TXCSR | :TEST TXDONE |
| 3439 | 020636 | 001001 | | | | BNE | 2\$ | :WAIT |
| 3440 | 020640 | 104024 | | | | HLT | 24 | :TXDONE FAILED TO SET |
| 3441 | 020642 | 012777 | 000400 | 160544 | 2\$: | MOV | #TSOM,@TXDBUF | :LOAD TX BUFFER |
| 3442 | 020650 | 105777 | 160536 | | | TSTB | @TXCSR | :CHECK DONE |
| 3443 | 020654 | 100375 | | | | BPL | -.4 | :AND THEN |
| 3444 | 020656 | 005077 | 160532 | | | CLR | @TXDBUF | :LOAD BUFFER |
| 3445 | 020662 | 105777 | 160524 | | | TSTB | @TXCSR | :AND CHECK |
| 3446 | 020666 | 100375 | | | | BPL | -.4 | :DONE AGAIN, THEN |
| 3447 | 020670 | 012777 | 001000 | 160516 | | MOV | #TEOM,@TXDBUF | :SET END OF MSG |
| 3448 | 020676 | 005037 | 177776 | | | CLR | PS | :LOWER PS |
| 3449 | 020702 | | | | 10\$: | | | |
| 3450 | 020702 | 012737 | 000050 | 020732 | | MOV | #40.,73\$ | :LOAD THE NUMBER |
| 3451 | 020710 | 032777 | 004000 | 160476 | 71\$: | BIT | #TIMER,@TXDBUF | :CHECK THE TIMER BIT |
| 3452 | 020716 | 001374 | | | | BNE | 71\$ | :BR IF SET |
| 3453 | 020720 | 032777 | 004000 | 160466 | 72\$: | BIT | #TIMER,@TXDBUF | :CHECK THE BIT |
| 3454 | 020726 | 001774 | | | | BEQ | 72\$ | :BR IF CLEAR |
| 3455 | 020730 | 005327 | | | | DEC | (PC)+ | :DECREMENT THE NUMBER |
| 3456 | 020732 | 000050 | | | 73\$: | 40. | | :OF TIMES TO REPEAT |
| 3457 | 020734 | 001365 | | | | BNE | 71\$ | :BR IF MORE TO GO |
| 3458 | 020736 | 104022 | | | | HLT | 22 | :RECEIVER FAILED TO INTERRUPT |
| 3459 | 020740 | 000420 | | | | BR | 5\$ | :LEAVE |
| 3460 | 020742 | | | | 3\$: | | | |
| 3461 | 020742 | 032777 | 004000 | 160444 | 74\$: | BIT | #TIMER,@TXDBUF | :CHECK THE TIMER BIT |
| 3462 | 020750 | 001374 | | | | BNE | 74\$ | :BR IF SET |
| 3463 | 020752 | 032777 | 004000 | 160434 | 75\$: | BIT | #TIMER,@TXDBUF | :CHECK THE TIMER BIT |
| 3464 | 020760 | 001774 | | | | BEQ | 75\$ | :BR IF CLEAR |
| 3465 | 020762 | 012737 | 000340 | 177776 | | MOV | #340,PS | :RAISE PS |
| 3466 | 020770 | 033777 | 000200 | 160406 | | BIT | RXDONE,@RXCSR | :TEST RXDONE |
| 3467 | 020776 | 001001 | | | | BNE | 5\$ | :BR IF SET |
| 3468 | 021000 | 104024 | | | | HLT | 24 | :RXDONE IS NOT SET AND SHOULD BE |
| 3469 | 021002 | 012737 | 000340 | 177776 | 5\$: | MOV | #340,PS | :LOCKOUT INTERRUPTS |
| 3470 | 021010 | 005077 | 160370 | | | CLR | @RXCSR | :CLEAR OUT DEVICE |
| 3471 | 021014 | 005077 | 160366 | | | CLR | @RXDBUF | :DITTO |
| 3472 | 021020 | 012706 | 001150 | | | MOV | #STACK,SP | :RESET THE STACK |
| 3473 | 021024 | 104400 | | | | SCOPE | | :SCOPE THIS TEST |
| 3474 | 021026 | | | | 6\$: | | | |

ONLY ONE INTERRUPT PER RXDONE TEST

```
3475 021026 012716 020742      MOV    #3$, (SP)      ;2ND PART SETUP
3476 021032 004537 007232      JSR    R5,SETVEC     ;SETUP FOR SECOND INTERRUPT TRY
3477 021036 021046                7$      ;RECEIVER VECTOR
3478 021040 007260                NO.BTRAP             ;TRANSMITTER VECTOR
3479 021042      340      340      .BYTE  340,340      ;LEVEL
3480 021044 000002                RTI                ;RETURN
3481 021046                7$:
3482 021046 104023                HLT    23           ;REPORT THE FACT YOU GOT HERE
3483                                ;YOU TOOK A SECOND INTERRUPT AND SHOULDN'T H^AVE
3484 021050 012716 021002      MOV    #5$, (SP)     ;SETUP TO LEAVE TEST
3485 021054 000002                RTI                ;LEAVE
3486
3487                                ;***** TEST 31 *****
3488                                ;*TEST TO VERIFY THAT INTERRUPT VECTOR 'A'
3489                                ;*OCCURS BEFORE INTERRUPT VECTOR 'B' EVEN
3490                                ;*WHEN VECTOR 'B' IS ENABLED BEFORE
3491                                ;*VECTOR 'A'
3492                                ;*****
3493
3494                                ;*****
3495                                ;*
3496                                ;* TEST 31
3497                                ;*
3498                                ;*****
3499                                ;*****
3500 021056 012737 000031 001226  TST31: MOV    #31,@TSTNO
3501 021064 012737 021416 001216      MOV    #TST32,NEXT
3502 021072 012737 000340 177776      MOV    #340,PS      ;SET PRIORITY TO 7
3503 021100 052777 000400 160304      BIS    #MRESET,@TXCSR ;RESET THE DEVICE
3504 021106 004737 005044                JSR    PC,SMALL     ;WAIT FOR RESET TO FINISH
3505 021112 004537 007232                JSR    R5,SETVEC     ;SET UP THE VECTORS
3506 021116 021376                4$      ;RECEIVER VECTOR
3507 021120 007260                NO.BTRAP             ;TRANSMITTER VECTOR
3508 021122      340      340      .BYTE  340,340      ;LEVEL
3509 021124 052777 004000 160260      BIS    #SYSTST,@TXCSR ;ENTER SYSTEM TEST MODE
3510 021132 052777 000020 160244      BIS    #RCVEN,@RXCSR ;TURN ON RECEIVER
3511 021140 052777 000100 160236      BIS    #RINTEN,@RXCSR ;TURN ON REC. DONE INT. ENABLE
3512 021146 052777 001000 160234      BIS    #CRCEN,@PARCSR ;TURN OFF CRC
3513 021154 052777 000020 160230      BIS    #SEND,@TXCSR  ;TURN ON TRANSMITTER
3514 021162                1$:
3515 021162 012737 000005 021212      MOV    #5,68$       ;LOAD THE NUMBER
3516 021170 032777 004000 160216 66$: BIT    #TIMER,@TXDBUF ;CHECK THE TIMER BIT
3517 021176 001374                BNE    66$          ;BR IF SET
3518 021200 032777 004000 160206 67$: BIT    #TIMER,@TXDBUF ;CHECK THE BIT
3519 021206 001774                BEQ    67$          ;BR IF CLEAR
3520 021210 005327                DEC    (PC)+        ;DECREMENT THE NUMBER
3521 021212 000005                68$: 5             ;OF TIMES TO REPEAT
3522 021214 001365                BNE    66$          ;BR IF MORE TO GO
3523 021216 032777 000200 160166      BIT    #TXDONE,@TXCSR ;TEST TXDONE
3524 021224 001001                BNE    2$           ;BR IF SET
3525 021226 104024                HLT    24           ;TXDONE FAILED TO SET
3526 021230                2$:
3527 021230 012777 000400 160156      MOV    #400,@TXDBUF ;LOAD TX BUFFER
3528 021236 105777 160150                TSTB   @TXCSR       ;CHECK FOR
3529 021242 100375                BPL    -4           ;DONE
3530 021244 005077 160144                CLR    @TXDBUF      ;LOAD THE BUFFER
```

DUAL VECTORING---RECEIVER BEFORE TRANSMITTER---TEST

```

3531 021250 105777 160136      TSTB   @TXCSR      ;AND CHECK
3532 021254 100375              BPL     .-4        ;DONE AGAIN
3533 021256 012777 001000 160130  MOV     #1000,@TXDBUF ;LOAD TEOM
3534 021264 012737 000144 021314  MOV     #100.,73$   ;LOAD THE NUMBER
3535 021272 032777 004000 160114  71$:   BIT     #TIMER,@TXDBUF ;CHECK THE TIMER BIT
3536 021300 001374              BNE     71$       ;BR IF SET
3537 021302 032777 004000 160104  72$:   BIT     #TIMER,@TXDBUF ;CHECK THE BIT
3538 021310 001774              BEQ     72$       ;BR IF CLEAR
3539 021312 005327              DEC     (PC)+     ;DECREMENT THE NUMBER
3540 021314 000144              73$:   100.      ;OF TIMES TO REPEAT
3541 021316 001365              BNE     71$       ;BR IF MORE TO GO
3542 021320 105777 160060      TSTB   @RXCSR      ;CHECK DONE
3543 021324 100401              BMI     5$        ;BR IF SET
3544 021326 104024              HLT     24        ;DONE FAILED TO SET
3545 021330 032777 000200 160054  5$:   BIT     #TXDONE,@TXCSR ;TEST TXDONE
3546 021336 001001              BNE     3$        ;BR IF SET
3547 021340 104024              HLT     24        ;TXDONE NOT SET SHOULD BE
3548 021342 052777 000100 160042  3$:   BIS     #TXINTE,@TXCSR ;TURN ON TRANSMITTER INT. ENABLE
3549 021350 005037 177776              CLR     PS        ;LOWER PROCESSOR STATUS
3550 021354 032777 004000 160032  74$:   BIT     #TIMER,@TXDBUF ;CHECK THE TIMER BIT
3551 021362 001374              BNE     74$       ;BR IF SET
3552 021364 032777 004000 160022  75$:   BIT     #TIMER,@TXDBUF ;CHECK THE TIMER BIT
3553 021372 001774              BEQ     75$       ;BR IF CLEAR
3554 021374 104027              HLT     27        ;DUP FAILED TO INTERRUPT
3555 021376 012706 001150              MOV     #STACK,SP ;RESET THE STACK
3556 021402 052777 000400 160002  4$:   BIS     #MRESET,@TXCSR ;RESET THE DEVICE
3557 021410 004737 005044              JSR     PC,SMALL  ;WAIT FOR RESET TO FINISH
3558 021414 104400              SCOPE              ;SCOPE THIS TEST

```

```

3559
3560      ;***** TEST 32 *****
3561      ;*TEST TO VERIFY THAT SERVICING THE
3562      ;*TXDONE BIT RE-ARMS THE INTERRUPT
3563      ;*LOGIC IF INTERRUPT ENABLE IS SET.
3564      ;*****

```

```

3565
3566      ;*****
3567      ;*
3568      ;* TEST 32
3569      ;*
3570      ;*****
3571      ;*****

```

```

3572 021416 012737 000032 001226  TST32: MOV     #32,@TSTNO
3573 021424 012737 021724 001216  MOV     #TST33,NEXT
3574 021432 012737 000340 177776  MOV     #340,PS      ;LOCK OUT INTERRUPTS
3575 021440 052777 000400 157744  BIS     #MRESET,@TXCSR ;RESET THE DEVICE
3576 021446 004737 005044              JSR     PC,SMALL  ;WAIT FOR RESET TO FINISH
3577 021452 004537 007232              JSR     R5,SETVEC ;INTERRUPT VECTOR SETUP
3578 021456 007254              NO.ATRAP         ;RECEIVER VECTOR
3579 021460 021636              4$              ;TRANSMITTER VECTOR
3580 021462 340              .BYTE 340,340    ;LEVEL
3581 021464 052777 004120 157720  BIS     #SYSTST!SEND.TXINTE,@TXCSR ;TURN ON TRANSMITTER, CLOCK
3582              ;AND INTERRUPTS
3583 021472 052777 000400 157714  BIS     #TSOM,@TXDBUF ;LOAD START OF MSG
3584 021500 005037 177776              CLR     PS        ;LOWER PROCESSOR STATUS
3585 021504 012737 000005 021534  MOV     #5,68$     ;LOAD THE NUMBER
3586 021512 032777 004000 157674  66$:   BIT     #TIMER,@TXDBUF ;CHECK THE TIMER BIT

```



```
3587 021520 001374          BNE      66$          :BR IF SET
3588 021522 032777 004000 157664 67$:  BIT      #TIMER,@TXDBUF :CHECK THE BIT
3589 021530 001774          BEQ      67$          :BR IF CLEAR
3590 021532 005327          DEC      (PC)+        :DECREMENT THE NUMBER
3591 021534 000005          68$:  5              :OF TIMES TO REPEAT
3592 021536 001365          BNE      66$          :BR IF MORE TO GO
3593 021540 104027          HLT      27          :DUP FAILED TO INTERRUPT THE FIRST TIME
3594 021542 000427          BR       3$          :LEAVE THE TEST
3595 021544 005037 177776          1$:  CLR      PS          :LOWER PROCESSOR STATUS
3596 021550 012737 000005 021600  MOV      #5,73$       :LOAD THE NUMBER
3597 021556 032777 004000 157630 71$:  BIT      #TIMER,@TXDBUF :CHECK THE TIMER BIT
3598 021564 001374          BNE      71$          :BR IF SET
3599 021566 032777 004000 157620 72$:  BIT      #TIMER,@TXDBUF :CHECK THE BIT
3600 021574 001774          BEQ      72$          :BR IF CLEAR
3601 021576 005327          DEC      (PC)+        :DECREMENT THE NUMBER
3602 021600 000005          73$:  5              :OF TIMES TO REPEAT
3603 021602 001365          BNE      71$          :BR IF MORE TO GO
3604 021604 104027          HLT      27          :DUP FAILED TO INTERRUPT AFTER SERVICING DONE
3605 021606 000405          BR       3$          :LEAVE
3606 021610          2$:
3607 021610 005077 157576          CLR      @TXCSR       :SHUT DOWN THE DUP
3608 021614 012716 021622          MOV      #3$, (SP)    :SETUP TO END TEST
3609 021620 000002          RTI          :RETURN
3610 021622 012737 000340 177776 3$:  MOV      #340,PS      :RAISE PROCESSOR STATUS
3611 021630 012706 001150          MOV      #STACK,SP    :RESET STACK
3612 021634 104400          SCOPE        :SCOPE THIS TEST
3613 021636 032777 000200 157546 4$:  BIT      #TXDONE,@TXCSR :CLEAR DONE AND RE-ARM INTERRUPT
3614 021644 005077 157544          CLR      @TXDBUF      :LOAD BUFFER
3615 021650 012737 000005 021700  MOV      #5,78$       :LOAD THE NUMBER
3616 021656 032777 004000 157530 76$:  BIT      #TIMER,@TXDBUF :CHECK THE TIMER BIT
3617 021664 001374          BNE      76$          :BR IF SET
3618 021666 032777 004000 157520 77$:  BIT      #TIMER,@TXDBUF :CHECK THE BIT
3619 021674 001774          BEQ      77$          :BR IF CLEAR
3620 021676 005327          DEC      (PC)+        :DECREMENT THE NUMBER
3621 021700 000005          78$:  5              :OF TIMES TO REPEAT
3622 021702 001365          BNE      76$          :BR IF MORE TO GO
3623 021704 012716 021544          MOV      #1$, (SP)    :SETUP TO FINISH TEST
3624 021710 004537 007232          JSR      R5,SETVEC    :SETUP VECTORS FOR NEXT PART OF TEST
3625 021714 007254          NO.ATRAPP        :RECEIVER VECTOR
3626 021716 021610          2$          :TRANSMITTER VECTOR
3627 021720 340 340          .BYTE 340,340        :LEVEL
3628 021722 000002          RTI          :RETURN
```

```
3629
3630
3631 :***** TEST 33 *****
3632 :*TEST TO VERIFY THAT SERVICING THE
3633 :*RXDONE BIT RE-ARMS THE INTERRUPT
3634 :*LOGIC IF INTERRUPT ENABLE IS SET.
3635 :*****
3636
3637 :*****
3638 :*
3639 :* TEST 33
3640 :*
3641 :*****
3642 :*****
```

```

3643 021724 012737 000033 001226 TST33: MOV #33,@TSTNO
3644 021732 012737 022316 001216 MOV #TST34,NEXT
3645 021740 012737 000340 177776 MOV #340,PS ;LOCK OUT INTERRUPTS
3646 021746 052777 000400 157436 BIS #MRESET,@TXCSR ;RESET THE DEVICE
3647 021754 004737 005044 JSR PC,SMALL ;WAIT FOR RESET TO FINISH
3648 021760 052777 004000 157424 BIS #SYSTST,@TXCSR ;ENTER SYSTST MODE
3649 021766 004537 007232 JSR R5,SETVEC ;SETUP VECTORS
3650 021772 022234 4$ ;RECEIVER VECTOR
3651 021774 007260 NO.BTRAP ;TRANSMITTER VECTOR
3652 021776 340 340 .BYTE 340,340 ;LEVEL
3653 022000 052777 000020 157376 BIS #RCVEN,@RXCSR ;TURN ON RECEIVER
3654 022006 052777 000100 157370 BIS #RINTEN,@RXCSR ;TURN ON INT. ENABLE
3655 022014 052777 000020 157370 BIS #SEND,@TXCSR ;TURN ON TRANSMITTER
3656 022022 1$:
3657 022022 012737 000005 022052 MOV #5,68$ ;LOAD THE NUMBER
3658 022030 032777 004000 157356 66$: BIT #TIMER,@TXDBUF ;CHECK THE TIMER BIT
3659 022036 001374 BNE 66$ ;BR IF SET
3660 022040 032777 004000 157346 67$: BIT #TIMER,@TXDBUF ;CHECK THE BIT
3661 022046 001774 BEQ 67$ ;BR IF CLEAR
3662 022050 005327 DEC (PC)+ ;DECREMENT THE NUMBER
3663 022052 000005 68$: 5 ;OF TIMES TO REPEAT
3664 022054 001365 BNE 66$ ;BR IF MORE TO GO
3665 022056 032777 000200 157326 BIT #TXDONE,@TXCSR ;TEST TXDONE
3666 022064 001001 BNE 2$ ;BR IF SET
3667 022066 104024 HLT 24 ;TXDONE FAILED TO SET
3668 022070 005037 177776 2$: CLR PS ;LOWER PROCESSOR STATUS
3669 022074 012777 000400 157312 MOV #400,@TXDBUF ;LOAD TX BUFFER
3670 022102 105777 157304 TSTB @TXCSR
3671 022106 100375 BPL .-4
3672 022110 005077 157300 CLR @TXDBUF
3673 022114 105777 157272 TSTB @TXCSR
3674 022120 100375 BPL .-4
3675 022122 012777 001000 157264 MOV #1000,@TXDBUF
3676 022130 7$:
3677 022130 012737 000050 022160 MOV #40.,73$ ;LOAD THE NUMBER
3678 022136 032777 004000 157250 71$: BIT #TIMER,@TXDBUF ;CHECK THE TIMER BIT
3679 022144 001374 BNE 71$ ;BR IF SET
3680 022146 032777 004000 157240 72$: BIT #TIMER,@TXDBUF ;CHECK THE BIT
3681 022154 001774 BEQ 72$ ;BR IF CLEAR
3682 022156 005327 DEC (PC)+ ;DECREMENT THE NUMBER
3683 022160 000050 73$: 40. ;OF TIMES TO REPEAT
3684 022162 001365 BNE 71$ ;BR IF MORE TO GO
3685 022164 104022 HLT 22 ;RECEIVER FAILED TO INTERRUPT
3686 022166 000445 BR 6$ ;GET OUT OF TEST
3687 022170 005037 177776 3$: CLR PS ;LOWER STATUS
3688 022174 012737 000005 022224 MOV #5,78$ ;LOAD THE NUMBER
3689 022202 032777 004000 157204 76$: BIT #TIMER,@TXDBUF ;CHECK THE TIMER BIT
3690 022210 001374 BNE 76$ ;BR IF SET
3691 022212 032777 004000 157174 77$: BIT #TIMER,@TXDBUF ;CHECK THE BIT
3692 022220 001774 BEQ 77$ ;BR IF CLEAR
3693 022222 005327 DEC (PC)+ ;DECREMENT THE NUMBER
3694 022224 000005 78$: 5 ;OF TIMES TO REPEAT
3695 022226 001365 BNE 76$ ;BR IF MORE TO GO
3696 022230 104022 HLT 22 ;DUP FAILED TO INTFRRUPT AFTER SERVICING DONE
3697 022232 000423 BR 6$ ;LEAVE IT
3698

```

```

3699 022234 032777 000200 157142 4$: BIT #RXDONE,@RXCSR ;SERVICE THE DONE BIT
3700 022242 012716 022170 MOV #3$, (SP) ;SETUP FOR 2ND PART OF TEST
3701 022246 004537 007232 JSR R5,SETVEC ;SETUP NEW VECTORS
3702 022252 022262 5$ ;RECEIVER VECTOR
3703 022254 007260 NO.BTRAP ;TRANSMITTER VECTOR
3704 022256 340 340 .BYTE 340,340 ;LEVEL
3705 022260 000002 RTI ;GO FINISH TEST
3706 022262 5$:
3707 022262 052777 000400 157122 BIS #MRESET,@TXCSR ;RESET THE DEVICE
3708 022270 004737 005044 JSR PC,SMALL ;WAIT FOR RESET TO FINISH
3709 022274 012716 022302 MOV #6$, (SP) ;SETUP TO FINISH TEST
3710 022300 000002 RTI ;LEAVE
3711 022302 012737 000340 177776 6$: MOV #340,PS ;RAISE PROCESSOR STATUS
3712 022310 012706 001150 MOV #STACK,SP ;RESET THE STACK
3713 022314 104400 SCOPE ;SCOPE THIS TEST
3714 :***** TEST 34 *****
3715 :*TEST TO PROVE AN INTERRUPT REQUEST
3716 :*IS GENERATED WHEN AN ABORT IS RECEIVED.
3717 :*****
3718
3719 :*****
3720 :*
3721 : TEST 34
3722 :*
3723 :*****
3724 :*****
3725 022316 012737 000034 001226 TST34: MOV #34,@TSTNO
3726 022324 012737 022576 001216 MOV #TST35,NEXT
3727 022332 012737 000340 177776 MOV #340,PS ;LOCK OUT INTERRUPTS
3728 022340 052777 000400 157044 BIS #MRESET,@TXCSR ;RESET THE DEVICE
3729 022346 004737 005044 JSR PC,SMALL ;WAIT FOR RESET TO FINISH
3730 022352 052777 004000 157032 BIS #SYSTST,@TXCSR ;ENTER SYS TST MODE
3731 022360 004537 007232 JSR R5,SETVEC ;SET UP VECTORS
3732 022364 022550 2$ ;RECEIVER
3733 022366 007260 NO.BTRAP ;TRANSMITTER
3734 022370 340 340 .BYTE 340,340 ;LEVEL
3735 022372 005037 177776 CLR PS ;LOWER PROCESSOR STATUS
3736 022376 052777 000120 157000 BIS #RCVEN!RINTEN,@RXCSR ;TURN ON RECEIVER AND INTERRUPT ENABLE
3737 022404 052777 000020 157000 BIS #SEND,@TXCSR ;TURN ON TRANSMITTER
3738 022412 012737 000005 022442 MOV #5,68$ ;LOAD THE NUMBER
3739 022420 032777 004000 156766 66$: BIT #TIMER,@TXDBUF ;CHECK THE TIMER BIT
3740 022426 001374 BNE 66$ ;BR IF SET
3741 022430 032777 004000 156756 67$: BIT #TIMER,@TXDBUF ;CHECK THE BIT
3742 022436 001774 BEQ 67$ ;BR IF CLEAR
3743 022440 005327 DEC (PC)+ ;DECREMENT THE NUMBER
3744 022442 000005 68$: 5 ;OF TIMES TO REPEAT
3745 022444 001365 BNE 66$ ;BR IF MORE TO GO
3746 022446 105777 156740 TSTB @TXCSR ;TEST DONE
3747 022452 100401 BMI 1$ ;BR IF SET
3748 022454 104024 HLT 24 ;DONE FAILED TO SET
3749 022456 012777 000400 156730 1$: MOV #TSOM,@TXDBUF ;LOAD TX BUFFER
3750 022464 105777 156722 TSTB @TXCSR ;CHECK FOR DONE
3751 022470 100375 BPL -4 ;BR IF NOT SET
3752 022472 005077 156716 CLR @TXDBUF ;PUSH OUT A CHARACTER
3753 022476 105777 156710 TSTB @TXCSR ;WAIT FOR DONE
3754 022502 100375 BPL -4 ;BR IF NOT SET

```

```
3755 022504 052777 002000 156702      BIS      #TABORT,@TXDBUF ;SET ABORT
3756 022512 012737 000113 022542      MOV      #75.,73$      ;LOAD THE NUMBER
3757 022520 032777 004000 156666 71$:    BIT      #TIMER,@TXDBUF ;CHECK THE TIMER BIT
3758 022526 001374          BNE      71$          ;BR IF SET
3759 022530 032777 004000 156656 72$:    BIT      #TIMER,@TXDBUF ;CHECK THE BIT
3760 022536 001774          BEQ      72$          ;BR IF CLEAR
3761 022540 005327          DEC      (PC)+        ;DECREMENT THE NUMBER
3762 022542 000113          73$:    75.          ;OF TIMES TO REPEAT
3763 022544 001365          BNE      71$          ;BR IF MORE TO GO
3764 022546 104022          HLT      22          ;RECEIVER FAILED TO INTERRUPT
3765                                     ;ON AN ABORTED MESSAGE
3766 022550 012737 000340 177776 2$:    MOV      #340,PS      ;SET STATUS TO 7
3767 022556 012706 001150          MOV      #STACK,SP    ;RESET THE STACK
3768 022562 052777 000400 156622          BIS      #MRESET,@TXCSR ;RESET THE DEVICE
3769 022570 004737 005044          JSR      PC,SMALL     ;WAIT FOR RESET TO FINISH
3770 022574 104400          SCOPE
3771
3772                                     ;***** TEST 35 *****
3773                                     ;*THIS TEST IS AN AID FOR DEBUGGING CRC
3774                                     ;*ERRORS. A CHARACTER IS LOADED INTO THE
3775                                     ;*DUP AND PUSHED OUT BIT BY BIT WHILE
3776                                     ;*ALLOWING THE OPERATOR TO MONITOR THE CRC
3777                                     ;*CHARACTER AS IT IS GENERATED. THE DATA CHARACTER
3778                                     ;*CAN ALSO BE CHANGED BY THE OPERATOR.
3779                                     ;*PUT SW09=1 TO LOCK ON BITS. TO CONTINUE HIT
3780                                     ;*ANY KEY ON THE TTY. AFTER 16 TIMES PUT DOWN SW09 TO LEAVE
3781                                     ;*NOTE: REMEMBER--IN SDLC A ONE IS A LOGIC LOW IN THE
3782                                     ;*CRC GENERATOR.
3783                                     ;*****
3784                                     ;*****
3785                                     ;*
3786                                     ;* TEST 35
3787                                     ;*
3788                                     ;*****
3789                                     ;*****
3790 022576 012737 000035 001226 TST35: MOV      #35,@TSTNO
3791 022604 012737 002764 001216      MOV      #.EOP,NEXT
3792 022612 052777 000400 156572      BIS      #MRESET,@TXCSR ;RESET THE DEVICE
3793 022620 004737 005044          JSR      PC,SMALL     ;WAIT FOR RESET TO FINISH
3794 022624 012737 102010 014574      MOV      #CRC.CCITT,XPOLY ;LOAD THE POLYNOMIAL
3795 022632 012737 000125 023000      MOV      #125,3$      ;LOAD DATA TO SOFTWARE BCC-CHANGE CHARACTER HERE
3796 022640 013737 023000 001252      MOV      3$,SAVR1
3797 022646 012737 177777 014600      MOV      #-1,CALBCC    ;CLEAR FOR SOFTWARE BCC
3798 022654 013737 014600 023002      MOV      CALBCC,4$
3799 022662 005037 001242          CLR      TEMP3
3800 022666 005037 001244          CLR      TEMP4        ;CLEAR BIT COUNTER
3801 022672 005037 001246          CLR      TEMP5
3802 022676 005077 156512          CLR      @TXDBUF      ;RESET TXDONE
3803 022702 052777 014000 156502      BIS      #MMODE,@TXCSR ;ENTER MAINT MODE-PROGRAM CLOCKING
3804 022710 052777 000020 156466      BIS      #RCVEN,@RXCSR ;TURN ON RECEIVER
3805 022716 052777 000020 156466      BIS      #SEND,@TXCSR  ;TURN ON TRANSMITTER
3806 022724 012777 000400 156462      MOV      #TSOM,@TXDBUF ;
3807 022732 104412 000044          PKCLK   36.          ;PUSH OUT 2
3808 022736 013777 023000 156450      MOV      3$,@TXDBUF   ;LOAD DATA
3809 022744 104412 000020          PKCLK   ,16.         ;PUSH OUT ANOTHER
3810 022750 104412 000002 1$:    PKCLK   ,2           ;PUSH OUT A BIT
```

```

3811 022754 013737 001244 001254      MOV    TEMP4,SAVR2      ;SET UP TO TYPE
3812 022762 005237 001242              INC    TEMP3
3813 022766 005237 001244              INC    TEMP4            ;UPDATE BIT COUNTER
3814 022772 004537 014422      2$:   JSR    R5,SIMBCC    ;CALCULATE SOFTWARE BCC BASED ON THESE PARAMETERS
3815 022776 000001              1      ;SHIFTS
3816 023000 000000      3$:   .WORD 0          ;DATA
3817 023002 000000      4$:   .WORD 0          ;PREVIOUS BCC
3818 023004 004737 023102      JSR    PC,5$           ;CHECK TO SEE IF WE SHOULD WAIT FOR SCOPING
3819 023010 000241              CLC
3820 023012 106037 023000      RORB  3$              ;CLEAR FOR NEXT ROTATE
3821 023016 013737 014600 023002      MOV    CALBCC,4$      ;SET UP THE NEXT BIT
3822 023024 022737 000006 001244      CMP    #6,TEMP4       ;FOR THE SOFTWARE BCC
3823 023032 001002              BNE    .+6
3824 023034 005077 156354              CLR    @TXDBUF
3825 023040 022737 000014 001242      CMP    #12.,TEMP3
3826 023046 001003              BNE    12$
3827 023050 012777 001000 156336      MOV    #TEOM,@TXDBUF
3828 023056 022737 000020 001244      12$:  CMP    #16.,TEMP4    ;ALL DONE WITH THE CHARACTER?
3829                                ;INCREASE THE COMPARE NUMBER TO OUTPUT
3830                                ;CRC TO THE RECEIVER
3831 023064 001331              BNE    1$             ;BR IF MORE TO GO
3832 023066 052777 000400 156316      BIS    #MRESET,@TXCSR ;RESET THE DEVICE
3833 023074 004737 005044              JSR    PC,SMALL
3834 023100 104400              SCOPE                ;WAIT FOR RESET TO FINISH
3835                                ;SCOPE THIS TEST
3836 023102 032777 001000 156072      5$:   BIT    #SW09,@SWR   ;SW09=1?
3837 023110 001432              BEQ    6$             ;BR IF NO
3838 023112 013704 014600              MOV    CALBCC,R4     ;THE DATA CHARACTER IS ALWAYS
3839 023116 012737 000001 001256      MOV    #1,SAVR3      ;FOLLOWED BY A ZERO CHARACTER.TTHE
3840 023124 000241              CLC
3841 023126 006004              11$:  ROR    R4            ;DATA BIT IN CRC SHOWS WHICH BIT OF
3842 023130 006137 001256      ROL    SAVR3         ;THE TWO CHARACTERS IS BEING GENERATED.
3843 023134 103374              BCC    11$
3844 023136 105737 001246      TSTB  TEMP5
3845 023142 001006              BNE    10$
3846 023144 104402 023641              TYPE  .EM17          ;TYPE MSG
3847 023150 104402 023670              TYPE  .MH1           ;TYPE HEADER
3848 023154 105137 001246      COMB  TEMP5
3849 023160 104410      10$:  CONVRT
3850 023162 024536      DT1
3851 023164 105777 156014      7$:   TSTB  @TKCSR     ;CHECK TTY DONE--GO SCOPE THE CRC GENERATOR
3852                                ;NOTE: THE LSB IS RIGHT JUSTIFIED ON THE TTY OUTPUT
3853 023170 100375              BPL    7$            ;BR IF NOT YET
3854 023172 017701 156010      MOV    @TKDBR,R1    ;READ THE BUFFER
3855 023176 000207      6$:   RTS    PC        ;RETURN
3856
3857

```

```

3858 023200 051377 047505 020115 EM1: .ASCIZ <377>/REOM BIT /
(1) 023213 377 054122 047504 EM2: .ASCIZ <377>/RXDONE BIT /
(1) 023230 051377 040530 047502 EM3: .ASCIZ <377>/RXABORT BIT /
(1) 023246 051377 040530 052103 EM4: .ASCIZ <377>/RXACTIVE BIT /
(1) 023265 377 054122 042504 EM5: .ASCIZ <377>/RXDERR BIT /
(1) 023302 051377 043505 051511 EM6: .ASCIZ <377>/REGISTER ERROR /
(1) 023323 377 054122 042504 EM7: .ASCIZ <377>/RXDERR OR OVERRUN /
(1) 023347 377 053117 051105 EM10: .ASCIZ <377>/OVERRUN BIT /
(1) 023365 377 040504 040524 EM11: .ASCIZ <377>/DATA COMPARE ERROR /
(1) 023412 042377 052101 020101 EM12: .ASCIZ <377>/DATA COMPARE ERROR IN SECONDARY MODE /
(1) 023461 377 051103 020103 EM13: .ASCIZ <377>/CRC CALCULATION ERROR /
(1) 023511 377 051522 046517 EM14: .ASCIZ <377>/RSOM BIT /
(1) 023524 040506 046111 042105 DH1: .ASCIZ /FAILED TO CLEAR /
(1) 023545 106 044501 042514 DH2: .ASCIZ /FAILED TO SET /
(1) 023564 052777 042516 050130 DH3: .ASCIZ <377>/UNEXPECTED /
(1) 023601 377 054105 042520 DH6: .ASCIZ <377>/EXPECTED FOUND REGISTER /
(1) 023641 377 051103 020103 EM17: .ASCIZ <377>/CRC GENERATOR STATUS /
(1) 023670 042377 052101 020101 MH1: .ASCIZ <377>/DATA CHAR DATA BIT IN CRC GEN. CRC FOR THIS BIT /
(1) 023763 015 052012 040522 EM15: .ASCIZ <15><12>/TRANSMITTER /
(1) 024003 015 051012 041505 EM16: .ASCIZ <15><12>/RECEIVER /
(1) 024020 047504 042516 041040 EM23: .ASCIZ /DONE BIT /
(1) 024033 106 044501 042514 EM22: .ASCIZ /FAILED TO SET /
(1) 024052 052777 042516 050130 EM20: .ASCIZ <377>/UNEXPECTED RECEIVER INTERRUPT /
(1) 024113 377 047125 054105 EM21: .ASCIZ <377>/UNEXPECTED TRANSMITTER INTERRUPT /
(1) 024157 040 051522 046517 EM24: .ASCIZ / RSOM BIT /
(1) 024173 106 044501 042514 DH4: .ASCIZ /FAILED TO INTERRUPT. /
(1) 024220 047111 042524 052522 DH5: .ASCIZ /INTERUPTED UNEXPECTEDLY./
(1) 024251 040 042523 020124 DH7: .ASCIZ / SET PREMATURELY /
(1) 024274 .EVEN
  
```

```

(1)
(1)
(1)
(1) 024274 .ERRTAB:
(1) 024274 000000 0
(1) 024276 000000 0
(1) 024300 000000 0
(1) 024302 023200 EM1
(1) 024304 023545 DH2 :HALT 1
(1) 024306 000000 0
(1)
(1) 024310 023213 EM2
(1) 024312 023545 DH2 :HALT 2
(1) 024314 000000 0
(1)
(1) 024316 023302 EM6
(1) 024320 023601 DH6 :HALT 3
(1) 024322 024554 DT6
(1)
(1) 024324 023200 EM1
(1) 024326 023524 DH1 :HALT 4
(1) 024330 000000 0
(1)
(1) 024332 023230 EM3
(1) 024334 023545 DH2 :HALT 5
(1) 024336 000000 0
(1)
  
```

| | | | | | |
|-----|--------|--------|------|-------|----|
| (1) | 024340 | 023246 | EM4 | | |
| (1) | 024342 | 023524 | DH1 | ;HALT | 6 |
| (1) | 024344 | 000000 | 0 | | |
| (1) | | | | | |
| (1) | 024346 | 023265 | EM5 | | |
| (1) | 024350 | 023545 | DH2 | ;HALT | 7 |
| (1) | 024352 | 000000 | 0 | | |
| (1) | | | | | |
| (1) | 024354 | 023365 | EM11 | | |
| (1) | 024356 | 000000 | 0 | ;HALT | 10 |
| (1) | 024360 | 000000 | 0 | | |
| (1) | | | | | |
| (1) | 024362 | 023230 | EM3 | | |
| (1) | 024364 | 023524 | DH1 | ;HALT | 11 |
| (1) | 024366 | 000000 | 0 | | |
| (1) | | | | | |
| (1) | 024370 | 023412 | EM12 | | |
| (1) | 024372 | 000000 | 0 | ;HALT | 12 |
| (1) | 024374 | 000000 | 0 | | |
| (1) | | | | | |
| (1) | 024376 | 023323 | EM7 | | |
| (1) | 024400 | 023545 | DH2 | ;HALT | 13 |
| (1) | 024402 | 000000 | 0 | | |
| (1) | | | | | |
| (1) | 024404 | 023246 | EM4 | | |
| (1) | 024406 | 023545 | DH2 | ;HALT | 14 |
| (1) | 024410 | 000000 | 0 | | |
| (1) | | | | | |
| (1) | 024412 | 023347 | EM10 | | |
| (1) | 024414 | 023524 | DH1 | ;HALT | 15 |
| (1) | 024416 | 000000 | 0 | | |
| (1) | | | | | |
| (1) | 024420 | 023564 | DH3 | | |
| (1) | 024422 | 023302 | EM6 | ;HALT | 16 |
| (1) | 024424 | 000000 | 0 | | |
| (1) | | | | | |
| (1) | 024426 | 023461 | EM13 | | |
| (1) | 024430 | 000000 | 0 | ;HALT | 17 |
| (1) | 024432 | 000000 | 0 | | |
| (1) | | | | | |
| (1) | 024434 | 023511 | EM14 | | |
| (1) | 024436 | 023524 | DH1 | ;HALT | 20 |
| (1) | 024440 | 000000 | 0 | | |
| (1) | | | | | |
| (1) | 024442 | 023511 | EM14 | | |
| (1) | 024444 | 023545 | DH2 | ;HALT | 21 |
| (1) | 024446 | 000000 | 0 | | |
| (1) | | | | | |
| (1) | 024450 | 024003 | EM16 | | |
| (1) | 024452 | 024173 | DH4 | ;HALT | 22 |
| (1) | 024454 | 000000 | 0 | | |
| (1) | | | | | |
| (1) | 024456 | 024003 | EM16 | | |
| (1) | 024460 | 024220 | DH5 | ;HALT | 23 |
| (1) | 024462 | 000000 | 0 | | |
| (1) | | | | | |

| | | | | | |
|-----|--------|--------|-----|---------|----------|
| (1) | 024464 | 024020 | | EM23 | |
| (1) | 024466 | 024033 | | EM22 | ;HALT 24 |
| (1) | 024470 | 000000 | | 0 | |
| (1) | | | | | |
| (1) | 024472 | 024052 | | EM20 | |
| (1) | 024474 | 000000 | | 0 | ;HALT 25 |
| (1) | 024476 | 000000 | | 0 | |
| (1) | | | | | |
| (1) | 024500 | 024113 | | EM21 | |
| (1) | 024502 | 000000 | | 0 | ;HALT 26 |
| (1) | 024504 | 000000 | | 0 | |
| (1) | | | | | |
| (1) | 024506 | 023763 | | EM15 | |
| (1) | 024510 | 024173 | | DH4 | ;HALT 27 |
| (1) | 024512 | 000000 | | 0 | |
| (1) | | | | | |
| (1) | 024514 | 023200 | | EM1 | |
| (1) | 024516 | 023524 | | DH1 | |
| (1) | 024520 | 000000 | | 0 | ;HALT 30 |
| (1) | | | | | |
| (1) | 024522 | 023213 | | EM2 | |
| (1) | 024524 | 023524 | | DH1 | |
| (1) | 024526 | 000000 | | 0 | ;HALT 31 |
| (1) | | | | | |
| (1) | 024530 | 024157 | | EM24 | |
| (1) | 024532 | 023524 | | DH1 | |
| (1) | 024534 | 000000 | | 0 | ;HALT 32 |
| (1) | | | | | |
| (1) | 024536 | 000003 | | DT1: | 3 |
| (1) | 024540 | 006 | 021 | .BYTE | 6.17. |
| (1) | 024542 | 001252 | | SAVR1 | |
| (1) | 024544 | 006 | 017 | .BYTE | 6.15. |
| (1) | 024546 | 001254 | | SAVR2 | |
| (1) | 024550 | 006 | 002 | .BYTE | 6.2 |
| (1) | 024552 | 001256 | | SAVR3 | |
| (1) | | | | | |
| (1) | 024554 | 000003 | | DT6: | 3 |
| (1) | 024556 | 006 | 004 | .BYTE | 6.4 |
| (1) | 024560 | 001262 | | SAVR5 | |
| (1) | 024562 | 006 | 002 | .BYTE | 6.2 |
| (1) | 024564 | 001260 | | SAVR4 | |
| (1) | 024566 | 006 | 002 | .BYTE | 6.2 |
| (1) | 024570 | 001256 | | SAVR3 | |
| (1) | 024572 | 000001 | | CORMAX: | |
| | | | | .END | |

CZDPCC MACY11 30A(1052) 02-MAY-78 13:54 PAGE 83
 CZDPCC.P11 02-MAY-78 13:47 CROSS REFERENCE TABLE -- USER SYMBOLS

| | | | | | | | | | | |
|--------|----------|-------|-------|-------|-------|-------|------|------|------|------|
| DISPRE | 000174 | 646# | 954 | | | | | | | |
| DSCA | = 100000 | 719# | | | | | | | | |
| DSCB | = 000001 | 734# | | | | | | | | |
| DSINTE | = 000040 | 729# | 3303 | 3323 | | | | | | |
| DSR | = 001000 | 725# | 2811 | 3137 | | | | | | |
| DTR | = 000002 | 733# | 2786 | 2811 | 2827 | 3112 | 3137 | 3153 | 3325 | |
| DT1 | 024536 | 3850 | 3858# | | | | | | | |
| DT6 | 024554 | 3858# | | | | | | | | |
| DUPACT | 001310 | 708# | 1044* | 1068* | 1101* | 1102 | 1590 | 1595 | | |
| DUPCR0 | 001500 | 869# | 983 | 1047 | | | | | | |
| DUPCR1 | 001506 | 873# | 1048 | | | | | | | |
| DUPCR2 | 001514 | 877# | | | | | | | | |
| DUPCR3 | 001522 | 881# | | | | | | | | |
| DUPCR4 | 001530 | 885# | | | | | | | | |
| DUPCR5 | 001536 | 889# | | | | | | | | |
| DUPCR6 | 001544 | 893# | | | | | | | | |
| DUPCR7 | 001552 | 897# | | | | | | | | |
| DUPNUM | 001311 | 709# | 937 | 1036* | 1069* | 1153 | | | | |
| DUPRPS | 001376 | 839# | 1636* | 1637* | 1638 | 1750* | | | | |
| DUPRVC | 001374 | 838# | 1170 | 1615* | 1636 | 1748* | | | | |
| DUPSEC | 001416 | 847# | 1622* | | | | | | | |
| DUPTPS | 001402 | 841# | 1640* | 1641* | 1751* | | | | | |
| DUPTR0 | 001502 | 870# | 991 | | | | | | | |
| DUPTR1 | 001510 | 874# | | | | | | | | |
| DUPTR2 | 001516 | 878# | | | | | | | | |
| DUPTR3 | 001524 | 882# | | | | | | | | |
| DUPTR4 | 001532 | 886# | | | | | | | | |
| DUPTR5 | 001540 | 890# | | | | | | | | |
| DUPTR6 | 001546 | 894# | | | | | | | | |
| DUPTR7 | 001554 | 898# | | | | | | | | |
| DUPTVC | 001400 | 840# | 1638* | 1639* | 1640 | 1749* | | | | |
| DUP.EN | 001560 | 901# | 976 | 1060 | 1601 | 1610 | | | | |
| DUP.MA | 001500 | 714 | 868# | 941 | 974 | 1058 | 1062 | 1076 | 1603 | 1613 |
| DUP0.A | 001504 | 871# | 1031* | 1034* | | | | | | |
| DUP1.A | 001512 | 875# | | | | | | | | |
| DUP2.A | 001520 | 879# | | | | | | | | |
| DUP3.A | 001526 | 883# | | | | | | | | |
| DUP4.A | 001534 | 887# | | | | | | | | |
| DUP5.A | 001542 | 891# | | | | | | | | |
| DUP6.A | 001550 | 895# | | | | | | | | |
| DUP7.A | 001556 | 899# | | | | | | | | |
| EM1 | 023200 | 3858# | | | | | | | | |
| EM10 | 023347 | 3858# | | | | | | | | |
| EM11 | 023365 | 3858# | | | | | | | | |
| EM12 | 023412 | 3858# | | | | | | | | |
| EM13 | 023461 | 3858# | | | | | | | | |
| EM14 | 023511 | 3858# | | | | | | | | |
| EM15 | 023763 | 3858# | | | | | | | | |
| EM16 | 024003 | 3858# | | | | | | | | |
| EM17 | 023641 | 3846 | 3858# | | | | | | | |
| EM2 | 023213 | 3858# | | | | | | | | |
| EM20 | 024052 | 3858# | | | | | | | | |
| EM21 | 024113 | 3858# | | | | | | | | |
| EM22 | 024033 | 3858# | | | | | | | | |
| EM23 | 024020 | 3858# | | | | | | | | |
| EM24 | 024157 | 3858# | | | | | | | | |

| | | | | | | | | | | | | | |
|-----------------|-----------------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| \$RAYO = 177777 | 1# | 1792# | 1793 | 1796# | 1860# | 1861 | 1864# | 1945# | 1946 | 1949# | 2002# | 2003 | 2007# |
| | 2065# | 2066 | 2070# | 2151# | 2152 | 2155# | 2195# | 2196 | 2199# | 2259# | 2260 | 2263# | 2311# |
| | 2312 | 2315# | 2359# | 2360 | 2362# | 2398# | 2399 | 2403# | 2453# | 2454 | 2456# | 2513# | 2514 |
| | 2517# | 2575# | 2576 | 2578# | 2694# | 2695 | 2698# | 2757# | 2758 | 2763# | 2863# | 2864 | 2869# |
| | 2969# | 2970 | 2975# | 3079# | 3080 | 3085# | 3188# | 3189 | 3192# | 3222# | 3223 | 3226# | 3274# |
| | 3275 | 3280# | 3344# | 3345 | 3348# | 3404# | 3405 | 3408# | 3487# | 3488 | 3492# | 3560# | 3561 |
| | 3564# | 3631# | 3632 | 3635# | 3714# | 3715 | 3717# | 3772# | 3773 | 3783# | | | |
| \$Y 000014 | 800# | 809 | 811# | 813# | 815# | 817# | 819# | 821# | 823# | 825# | 827# | 829# | 831# |
| | 833# | | | | | | | | | | | | |
| . 024572 | 626# | 627 | 630# | 637# | 642# | 645# | 648# | 652# | 654# | 708# | 709# | 710# | 711# |
| | 712# | 713# | 779# | 780# | 785# | 786# | 867# | 869# | 870# | 871# | 873# | 874# | 875# |
| | 877# | 878# | 879# | 881# | 882# | 883# | 885# | 886# | 887# | 889# | 890# | 891# | 893# |
| | 894# | 895# | 897# | 898# | 899# | 970 | 1100 | 1574# | 1576# | 1578# | 1579# | 1594 | 2599# |
| | 2600# | 2775 | 2881 | 2987 | 3097 | 3291 | 3385 | 3388 | 3443 | 3446 | 3529 | 3532 | 3671 |
| | 3674 | 3751 | 3754 | 3823 | 3858# | | | | | | | | |
| .BEGIN 002660 | 1116# | | | | | | | | | | | | |
| .CNVRT 004034 | 828 | 1352# | | | | | | | | | | | |
| .CONVR 004030 | 826 | 1351# | | | | | | | | | | | |
| .EOP 002764 | 1137# | 3791 | | | | | | | | | | | |
| .ERRTA 024274 | 1459 | 3858# | | | | | | | | | | | |
| .HLT 004350 | 633 | 1440# | | | | | | | | | | | |
| .INSTE 003516 | 818 | 1257# | | | | | | | | | | | |
| .INSTR 003412 | 816 | 1236# | | | | | | | | | | | |
| .INST1 003432 | 1240# | 1260 | | | | | | | | | | | |
| .MSG 003434 | 1238* | 1241# | | | | | | | | | | | |
| .PARAM 003536 | 820 | 1268# | | | | | | | | | | | |
| .PFAIL 005050 | 631 | 936 | 1545# | 1549 | | | | | | | | | |
| .PKCLK 005006 | 830 | 1527# | | | | | | | | | | | |
| .RES05 003776 | 824 | 1339# | | | | | | | | | | | |
| .SAV05 003736 | 822 | 1325# | | | | | | | | | | | |
| .SCOPE 003160 | 810 | 1180# | | | | | | | | | | | |
| .SCOPI 003312 | 812 | 1210# | | | | | | | | | | | |
| .SETFL 004242 | 832 | 1408# | 1420 | | | | | | | | | | |
| .START 001562 | 649 | 934# | 946 | | | | | | | | | | |
| .TRPSR 004316 | 63 ^f | 1428# | | | | | | | | | | | |
| .TRPTA 001344 | 80-# | 1433 | | | | | | | | | | | |
| .TYPE 003336 | 8 ¹⁴ | 1220# | | | | | | | | | | | |

. ABS. 024572 000

ERRORS DETECTED: 0

DSKZ:CZDPCC,DSKZ:CZDPCC,SEQ DSKZ:DUP11.MAC,DSKZ:CZDPCC.P11
 RUN-TIME: 13 18 1 SECONDS
 RUN-TIME RATIO: 206/33-6.1
 CORE USED: 27K (53 PAGES)

DOCUMENT PAGES: 88