

DR11B/DA11

GENERAL INTERFACE
MD-11-DZDRB-E

EP-DZDRB-E-DL-A
COPYRIGHT © 71-77
FICHE 1 OF 1

AUG 1977
digital
MADE IN USA

Frame 1	Frame 2	Frame 3
Frame 4	Frame 5	Frame 6
Frame 7	Frame 8	Frame 9
Frame 10	Frame 11	Frame 12
Frame 13	Frame 14	Frame 15
Frame 16	Frame 17	Frame 18
Frame 19	Frame 20	Frame 21
Frame 22	Frame 23	Frame 24
Frame 25	Frame 26	Frame 27
Frame 28	Frame 29	Frame 30
Frame 31	Frame 32	Frame 33
Frame 34	Frame 35	Frame 36

.REM %

IDENTIFICATION

PRODUCT CODE: MAINDEC-11-DZDRB-E-D
PRODUCT NAME: DR11B - PDP11 GENERAL NPR INTERFACE
DA11B - PDP11 INTERPROCESSER LINK
DATE RELEASED: JUNE, 1977
MAINTAINER: DIAGNOSTIC ENGINEERING
AUTHOR: LARRY CONDON, DICK JONES, STEVE POMFRET
REVISED BY: W.F.KELICKER 29-JAN-1974
ALAN BOSTICK 20-JUN-1976

COPYRIGHT (C) DIGITAL EQUIPMENT CORPORATION
1971, 1977

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.

1. ABSTRACT

- 1.1 THIS IS A LOGIC TEST OF THE "NPR GENERAL INTERFACE" -DR11B.
THERE IS A SPECIAL MAINTENANCE FEATURE THAT ALLOWS TESTING
OF NPRS WITHOUT A CUSTOMERS DEVICE ATTACHED.
1.2 THERE IS A SECOND TEST INCLUDED FOR EXERCISING THE DA11B
INTERPROCESSER LINK. THE DR11B TEST SHOULD BE RUN IN
IN EACH COMPUTER BEFOUR TESTING THE DA11B.

2. REQUIREMENTS

2.1 EQUIPMENT

2.1.1 FOR THE DR11B

PDP-11 STANDARD COMPUTER

DR11B

NOTE: WITH OR WITHOUT HARDWARE SWITCH REGISTER

2.1.2 FOR THE DA11B

2 PDP-11 STANDARD COMPUTERS

1 DA11B CONSISTING OF 2 M7229 MODULES AND 2
BCOBR CABLES.

NOTE: WITH OR WITHOUT HARDWARE SWITCH REGISTER

2.2 STORAGE

- 2.2.1 PROGRAM STORAGE - THE ROUTINE USES MEMORY
FROM 0 TO 14000.

3. LOADING PROCEDURE

3.1 METHOD

PROCEDURE FOR NORMAL BINARY TAPES SHOULD BE FOLLOWED.

4. DR11B STARTING PROCEDURE

4.1 CONTROL SWITCH SETTING

STARTING AT SA 200 ALL SWITCHES SHOULD BE DOWN OR ZERO.

4.2 STARTING ADDRESS OR ADDRESSES

(A) 200 = TEST OF LOGIC USING MAINTENANCE FEATURE M968 IN C04,D04

4.3 PROGRAM AND/OR OPERATOR ACTION

LOAD PROGRAM INTO MEMORY.
 LOAD STARTING ADDRESS
 PRESS START.
 THE PROGRAM WILL LOOP. 'END PASS' WILL BE
 TYPED AT THE END OF THE PROGRAM.

NOTE: IF SOFTWARE SWITCH REGISTER IS SELECTED THEN THE
 FOLLOWING WILL BE PRINTED:

SMR= XXXXXX NEW=
 (REFER TO SECTION 5.1 FOR OPERATOR OPTIONS)

5. DR11B OPERATING PROCEDURE
 5.1 OPERATIONAL SWITCH SETTINGS

5.1.1 AT SA 200 .. THE INSTRUCTION AND LOGIC TEST.
 WITH ALL SWITCHES DOWN THE PROGRAM WILL PRINT
 OUT ON ERRORS AND CONTINUE IN TEST. ('END
 PASS' TYPED AT COMPLETION OF A PASS)

5.1.2 SWITCH SETTINGS ARE

SW15 = 1 OR UP ... HALT ON ERROR
 SW14 = 1 OR UP ... SCOPE LOOP
 SW13 = 1 OR UP ... INHIBIT PRINTOUT
 SW12 = 1 OR UP ... INHIBIT TRACE TRAP
 SW11 = 1 OR UP ... INHIBIT ITERATIONS

5.1.3 IF THE DIAGNOSTIC IS RUN ON A CPU WITHOUT A SWITCH
 REGISTER THEN A SOFTWARE SWITCH REGISTER IS USED WHICH ALLOWS
 THE USER THE SAME SWITCH OPTIONS AS THE HARDWARE SWITCH REGISTER.
 IF THE HARDWARE SWITCH REGISTER DOES NOT EXIST OR IF ONE DOES
 AND IT CONTAINS ALL ONES (177777) THEN THE SOFTWARE SWITCH
 REGISTER (LOC. 176) IS USED.

CONTROL:

THIS PROGRAM ALSO SUPPORTS THE DYNAMIC LOADING OF THE SOFTWARE SWITCH
 REGISTER (LOC. 176) FROM THE TTY. THIS CAN BE ACCOMPLISHED BY
 DOING THE FOLLOWING:

- 1) TYPE CONTROL G (<↑G>); THIS WILL ALLOW THE TTY TO ENTER DATA INTO
 LOC. 176 AT SELECTED POINTS WITHIN THE PROGRAM.
- 2) THE MACHINE WILL THEN TYPE: SMR=XXXXXXNEW= (XXXXXX IS THE OCTAL CONTENTS
 OF THE SOFTWARE SWITCH REGISTER.)
- 3) AFTER THE ''NEW=''
 OF THE FOLLOWING AT THE TTY:
 - A) TYPE A NUMBER TO BE LOADED INTO LOC. 176 FOLLOWED BY A <CR>.
 (ONLY NUMBERS BETWEEN 0-7 WILL BE ACCEPTED AND ONLY 6 NUMBERS
 WILL BE ALLOWED)

IF A <CR> IS THE FIRST KEY DEPRESSED THE SOFTWARE SWITCH REGISTER CONTENTS WILL NOT BE CHANGED.

B) IF A CONTROL U (<U>) IS DEPRESSED THEN THE PROGRAM WILL SEND YOU BACK TO STEP 2.

5.2. SUBROUTINE ABSTRACTS

BEGIN SA 200

5.2.1 SCOPE

THIS SUBROUTINE CALL IS PLACED BETWEEN EACH SUBTEST IN THE INSTRUCTION SECTION. IT RECORDS THE STARTING ADDRESS OF EACH SUB-TEST AS IT IS BEING ENTERED. IF A SCOPE LOOP IS REQUESTED, IT WILL JUMP TO THE START OF THE SUBTEST THAT THE SCOPE LOOP IS REQUESTED FOR. IF SCOPE LOOP IS NOT REQUESTED, THERE WILL BE EITHER A FIXED OR RANDOM NUMBER OF ITERATIONS ON THAT SUBTEST BEFORE THE NEXT SUBTEST IS ENTERED. SWITCH 11 ON A 1 INHIBITS ITERATION OF SUBTESTS.
NOTE: SUPPORTS IG ROUTINE FOR DYNAMIC LOADING OF SOFTWARE SWITCH REGISTER

5.2.2 HALT

IS A ROUTINE THAT PRINTS-OUT AN ADDRESS THAT TAGS THE FAILING SUBTEST, THE CP STATUS REGISTER AND THE DR11B STATUS REGISTER AT THE TIME OF FAILURE.
NOTE: SUPPORTS IG ROUTINE FOR DYNAMIC LOADING OF SOFTWARE SWITCH REGISTER

5.2.3 LODBUF

THE INBUF BUFFER IS LOADED WITH AN INCREMENTING PATTERN (0,1,2,3,...) BEGINNING AT THE STARTING ADDRESS OF INBUF. THE NUMBER OF WORDS LOADED IS DETERMINED BY THE CONTENTS OF BUFLN.

5.2.4 CHKBFF

THE CHKBUFF BUFFER IS LOADED WITH A MODIFIED INCREMENTING PATTERN (0,0,2,2,4,4,6,6,...) BEGINNING AT THE STARTING ADDRESS OF CHKBUFF. THE NUMBER OF WORDS LOADED IS DETERMINED BY THE CONTENTS OF BUFLN. THIS BUFFER IS LOADED ONLY FOR TESTS WHICH USE THE MAINTENANCE MODE OF THE DR11-B WHICH HAS A SPECIAL ALTERNATING DATI-DATO SEQUENCE OF OPERATION.

5.2.5 INTA

THE IE BIT IS CLEARED IN THE DRST THEN THE DRST IS CHECKED FOR THE ABSENCE OF AN ERROR AND THE PRESENCE OF READY. THE DRWC IS CHECKED TO SEE THAT IT IS EQUAL TO ZERO. THE CORRECT CONTENTS

OF THE DRBA ARE CALCULATED AND CHECKED. THERE IS A JSR TO THE NORMAL SUB-ROUTINE BEFORE THIS ROUTINE IS EXITED. THE PROGRAM WILL HALT IF ERROR IS SET, READY IS CLEAR, OR READY AND ERROR ARE CLEAR.

5.2.6 DATCHK

THIS ROUTINE IS ENTERED TO CHECK INBUF AFTER A MAINTENANCE MODE OPERATION. THE CONTENTS OF INBUF AND THE CONTENTS OF CHKBUF ARE CHECKED TO SEE THAT THEY ARE THE SAME. THE NUMBER OF COMPARISONS MADE IS DETERMINED BY THE CONTENTS OF BUFLN.

5.2.7 NORMAL

THE ROUTINE IS ENTERED FROM INTA AND FROM SOME TESTS WHICH DON'T USE INTA. THE NUMBER OF THE DRINV+2 IS PUT INTO DRINV AND THE DRVS IS CLEARED. IF THE DR11-B INTERRUPTS UNDER THESE CONDITIONS THE PDP-11 WILL HALT AT DRVS. THE PROCESSOR STATUS WORD IS RESTORED TO LEVEL 7 AND THE ROUTINE IS EXITED.

5.2.8 DATOCK

AFTER A STRING OF DATO'S HAS BEEN COMPLETED THIS ROUTINE CHECKS THAT THE CORRECT DATA PATTERN (52525) WAS TRANSFERRED TO INBUF. THE NUMBER OF COMPARISONS MADE IS DETERMINED BY THE CONTENTS OF BUFLN. AN ADDITIONAL CHECK IS MADE ON BUFLN+2 TO INSURE THAT TOO MANY WORDS WEREN'T TRANSFERRED.

5.2.9 ERRCHK

THIS ROUTINE CLEARS IE AND HALTS IF ERROR IS SET.

5.2.10 TRAPCATCHER

THIS IS A SERIES OF INSTRUCTIONS STARTING AT LOCATION 0 DESIGNED TO DETECT AND ISOLATE UNEXPECTED TRAPS AND INTERRUPTS TO THE TRAP AND INTERRUPT VECTOR AREA OF MEMORY.

EACH VECTOR ENTRANCE ADDRESS IS LOADED WITH THE ADDRESS OF THE NEXT LOCATION. THE NEXT LOCATION IS LOADED WITH A HALT (00000). THUS AN ILLEGAL TRAP OR INTERRUPT WILL CAUSE A HALT AT THE TRAP LOCATION PLUS TWO.

IF A HALT OCCURS IN THE TRAP OR INTERRUPT AREA, EXAMINE REGISTER SIX, IT WILL CONTAIN THE CURRENT STACK ADDRESS. THE CONTENTS OF THE CURRENT STACK ADDRESS IS THE VALUE OF THE LOCATION COUNTER WHEN THE TRAP OR INTERRUPT OCCURRED.

5.3 PROGRAM AND/OR OPERATOR ACTION

5.3.1 LOADING AND STARTING AT 200 WITH ALL SWITCHES DOWN IS THE INSTRUCTION AND LOGIC TEST. IF AN ERROR IS DETECTED HERE, THERE WILL BE A PRINTOUT. WHEN AN ERROR IS DETECTED AND IT IS NECESSARY TO SCOPE ON IT, PLACE SW15 UP TO HALT ON ERROR, THEN SW14 UP TO LOOP ON ERROR, THEN SW13 UP TO DELETE PRINTOUTS.

6. DR11B ERRORS

6.1 ERROR PRINTOUT

THE PC OF THE FAILING TEST AND THE CP STATUS WILL BE PRINTED.

6.2 ERROR RECOVERY

DEPRESS CONTINUE TO RESTART SECTION

7. DR11B RESTRICTIONS

7.1 STARTING RESTRICTION

NONE

7.2 OPERATIONAL RESTRICTION

M968 MUST BE IN SLOTS C04/D04 - FOR DIAGNOSTIC TESTING SHOULD BE IN A02/B02 FOR NORMAL USER OPERATION.

8. MISCELLANEOUS

8.1 EXECUTION TIME

ABOUT 2 MINUTES

9. PROGRAM DESCRIPTION

THE FOLLOWING IS A GENERAL LIST OF FUNCTIONS TESTED.

CAN ALL REG BE ADDRESSED WITHOUT ERROR
RESET CLEAR DRWC
RESET CLEAR DRBA
CAN ALL DRWC BITS BE SET
CAN 15-1 IN DRBA BE SET
FNCT1 SET & CLEARED
FNCT2 SET & CLEARED
FNCT3 SET & CLEARED
XBA16 SET & CLEARED
XBA17 SET & CLEARED
IE SET & CLEARED
CYCLE SET & CLEARED

MAINT SET & CLEARED
 ALL DRST R/W BITS CAN BE SET & CLEARED
 ALL DRST R/W SET, RESET TO 0, RDY IS SET, NEX IS
 CLEARED, GO IS 0
 DRWC HOLD ALT. 0'S & 1'S AND ALT. 1'S & 0'S
 DRBA HOLD ALT. 0'S & 1'S AND ALT. 1'S & 0'S
 INC PATTERN TO WRAP-AROUND IN DRWC
 INC PATTERN TO WRAP-AROUND IN DRBA
 NO INT. AT LEVEL 7
 NO INT. AT LEVEL 6
 NO INT. AT LEVEL 5
 DOES INT. AT LEVEL 4
 NO MAINTBRD A02/B02
 MAINTBRD C04/D04
 FNCT BITS CONTROL DSTAT BITS
 RESET 0'S DRDB
 ALL DRDB BITS CAN BE SET
 DRDB HOLD ALT. 1'S & 0'S AND ALT. 0'S & 1'S
 INC TO WRAP-AROUND IN DRDB
 RESET SETS ONLY RDY IN DRST
 BADD READS AS 0
 1 DAT1 NPR
 1 DAT0 NPR
 BAOF FORCES ERROR & IS CLEARED BY CLEARING DRBA OR RESET
 GO CLEARS RDY
 DAT0 TO DIODE MEM CAUSES NEX
 DO WITHOUT CLEARING PREVIOUS ERROR CAUSES ANOTHER INT.
 10 DAT1'S (BURST)
 10 DAT0'S (BURST)
 200 DAT1'S (BURST)
 200 DAT0'S (BURST)
 200 DAT1'S (NON-BURST)
 200 DAT0'S (NON-BURST)
 FUNCT BITS INC WITH MAINT MODE XFERS
 10 MAINT MODE XFERS
 200 MAINT MODE XFERS

10. LISTING

11. FLOW CHART(S)

12. DA11B STARTING PROCEDURE

- 12.1 THERE ARE TWO STARTING LOCATIONS: ONE FOR THE COMPUTER THAT WILL BE THE SLAVE AND ANOTHER FOR THE COMPUTER THAT WILL BE THE MASTER.
- 12.2 SLAVE COMPUTER, LOAD ADDRESS 1006 AND PRESS START. PROCESSOR WILL HALT.
- 12.3 MASTER COMPUTER, LOAD ADDRESS 1000 AND PRESS START. PROCESSOR WILL HALT.

12.4 SLAVE COMPUTER, PRESS CONTINUE (ON SWITCHLESS PROCESSOR, TOGGLE THE HALT/CONT SWITCH).

12.5 MASTER COMPUTER, PRESS CONTINUE (ON SWITCHLESS PROCESSOR, TOGGLE THE HALT/CONT SWITCH).

13. DA11B OPERATING PROCEDURE

13.1 THE PROGRAM WILL LOOP AFTER STARTING AND PRINT OUT ANY ERRORS. THE PROGRAM WILL HALT AFTER NON RECOVERABLE ERRORS.

13.2 THE MAINT MODULE MUST BE IN A02/B02 AND THE THE DA11B MUST BE IN C04/D04 .

14. DA11B PROGRAM DISCRIPTION

14.1 THE SLAVE COMPUTER STARTS BY ENTERING A BACKGROUND TO WAIT FOR AN INTERRUPT WITH THE INTERRUPT ENABLED. THE FIRST INTERRUPT THAT COMES SHOULD BE THE READY INTERRUPT SET UP WHEN THE MASTER HIT THE START KEY. THE INTERRUPT CAUSES THE SLAVE TO ENTER THE INTERRUPT SERVICE ROUTINE.

14.2 THE INTERRUPT SERVICE ROUTINE DETERMINS WHAT INTERRUPT CAME UP AND IF IT SHOULD HAVE COME UP. IF THE INTERRUPT WAS THE ONE EXPECTED THAN THE THAN THE PROGRAM GOES TO THE TO THE PROPER JOB ROUTINE, SJOBXX FOR SLAVE SERVICE AND JOBXX FOR THE MASTER ROUTINE.

14.3 THE NEXT THING THAT SHOULD HAPPEN IS THE MASTER SHOULD ISSUE AN INTERRUPT TO THE SLAVE . THIS IS A SIGNAL FOR THE SLAVE TO ACCEPT THE WORD COUNT, OFFSET AND TWO CHECK SUM WORDS. THE SLAVE ACCEPTS A WORD AT A TIME FROM THE DATA BUFFER EACH TIME THE MASTER TOGGLES FUNCTION BIT 3. EACH TIME IT READS A WORD THE SLAVE SENDS THE WORD BACK TO THE MASTER FOR VERIFICATION.

14.4 AFTER THE SLAVE HAS RECIEVED ALL THE PARAMETERS IT SETS ITS DIRECTION BIT TO THE OPPOSIT DIRECTION AS THE MASTER AND STARTS THE NPR TRANSFER.

14.5 THE MASTER SETS UP THE TYPE OF TRANSFERS AND CHECKS THE DATA WHEN IT COMES BACK FROM THE SLAVE.

15. DA11B ERRORS

15.1 THE PC OF THE FAILING TEST, THE CP STATUS AND THE DR11B STATUS REGISTER WILL BE PRINTED AFTER AN ERROR.

15.2 THERE IS NO ERROR RECOVERY FOR THE DA11B TEST BECAUSE THE OTHER COMPUTER WILL GET OUT OF SYNC WHEN AN ERROR OCCURS.

%

434
435
436
437
438
439
440
441
442
443
444
445
446 000001
447 160000
448
449
450
451
452
453
454
455 000240
456 104400
457 104000
458 000001
459 000002
460 000004
461 000010
462 000020
463 000040
464 000100
465 000200
466 000400
467 001000
468 002000
469 004000
470 010000
471 020000
472 040000
473 100000
474
475 000004
476
477
478
479 000000
480 000030
481 000030 011176
482 000032 000340
483 000034 000034
484 000034 011764
485 000036 000340
486 000046 000046
487 000046 007142
488 000052 000052
489 000052 000000

```

*****
*****
TITLE MAINDEC-11-DZDRB-E
*COPYRIGHT (C) 1971, 1977
*DIGITAL EQUIPMENT CORP.
*MAYNARD, MASS. 01754
*
*PROGRAM BY POMFRET, JONES, CONDON
*
*THIS PROGRAM WAS ASSEMBLED USING THE PDP-11 MAINDEC SYSMAC
*PACKAGE (MAINDEC-11-DZGAC-C3), JAN 19, 1977.
*
$TN=1
SSWR=160000 ;:HALT ON ERROR, LOOP ON TEST, INHIBIT ERROR TYP0UT
*****
REVISD BY ALAN BOSTICK JUNE 1976
MODIFIED FOR SOFTWARE SWITCH REGISTER
INCLUDING DYNAMIC LOADING OF SWR
*****
*****
NOP=000240
SCOPE=TRAP
HLT=EMT
BIT0=000001
BIT1=000002
BIT2=000004
BIT3=000010
BIT4=000020
BIT5=000040
BIT6=000100
BIT7=000200
BIT8=000400
BIT9=001000
BIT10=002000
BIT11=004000
BIT12=010000
BIT13=020000
BIT14=040000
BIT15=100000

BUSERR=000004

;LOAD TRAP CATCHER INTO 0 THRU 777.

.=0
.=30
PRINT
340
.=34
SCOPEC
340
.=46
$ENDAD
.=52
0

```

```

490      000176      000176      . = 176
491      000176      000000      SWREG: 0
492      000200      000200      . = 200
493      000200      012706      013710      MOV      #BUFF,%6      ;SET UP STACK LIMIT
494      000204      005077      000606      CLR      #PSW
495      000210      023737      000042      000046      CMP      @#42,@#46      ;ARE WE IN ACT11 AUTO MODE?
496      000216      001404      BEQ      1$      ;SKIP TITLE IF YES
497      000220      012702      012525      MOV      #TITLE,%2      ;PRINT THE TITLE
498
499      000224      004767      012354      JSR      X7,TTOUT
500      000230      000167      000642      1$:     JMP      SUSWR
501      001000      . = 1000
502
503
504      ;*****
505      ; START OF BACK-TO-BACK DR11-B
506      ;*****
507
508
509      001000      000000      MSTART: HALT      ;MASTER START
510      001002      000167      006312      JMP      MS1
511      001006      000000      SSTART: HALT      ;SLAVE START
512      001010      000167      006364      JMP      SS1
513      001014      177570      SR:      177570
514      001016      177776      PSW:     177776
515      001020      172410      DRWC:    172410
516      001022      172412      DRBA:    172412
517      001024      172414      DRST:    172414
518      001026      172416      DRDB:    172416
519      001030      000126      DRVS:    126
520      001032      000240      DRINL:   240
521      001034      000124      DRINV:   124
522      001036      052525      NPR1:    52525
523      001040      173000      DIOMEM:  173000
524      001042      013712      INBUF:   XINBUF
525      001044      014714      CHKBUF:  XCHKBU
526      001046      000000      BUFLN:   HALT
527      001050      000000      LENCHK:  HALT
528      001052      000000      BRWAIT:  HALT
529      001054      000000      WCLN:    HALT
530      001056      000000      RDYCHK:  HALT
531      001060      177560      TKS:     177560
532      001062      177562      TKB:     177562
533      001064      177564      TPS:     177564
534      001066      177566      TPB:     177566
535      001070      000000      FNCCNT:  HALT
536      001072      000000      INBUF1:  HALT
537      001074      000000      PASCNT:  0      ;NUMBER OF PASSES COMPLETED
538
539      001076      013746      000006      SUSWR:   MOV      @#6,-(SP)      ;SAVE VECTORS
540      001102      013746      000004      MOV      @#4,-(SP)
541      001106      012737      001126      000004      MOV      @#4$,@#4      ;SET UP FOR TIMEOUT
542      001114      022777      177777      177672      CMP      #-1,@SR      ;REFERENCE HARDWARE SWITCH REGISTER
543      001122      001402      BEQ      65$
544      001124      000404      BR      65$
545      001126      022626      64$:     CMP      (SP)+,(SP)+      ;ADJUST STACK

```

```

546 001130 012767 000176 177656 655: MOV #SWREG,SR ;POINT TO SOFTWARE SWITCH REG
547 001136 012637 000004 665: MOV (SP)+,R#4 ;RESTORE VECTORS
548 001142 012637 000006 MOV (SP)+,R#6
549 001146 022767 000176 177640 CMP #SWREG,SR ;IS SWREG USED
550 001154 001005 BNE BEGIN
551 001156 005737 000042 TST R#42 ;ARE WE IN AUTO MODE?
552 001162 001002 BNE BEGIN ;IF SO, SKIP SWREG INPUT
553 001164 004767 010756 JSR PC,CNTLU ;ALLOW SWREG TO BE LOADED
554 001170 012777 000340 177620 BEGIN: MOV #340,R#PSW ;PROC. AT LEVEL #7
555 001176 012767 001170 010652 MOV #BEGIN,RETURN

```

```

;*****
;TEST 0 CAN ALL DR11-B REG BE ADDRESSED WITHOUT ERROR?
;*****

```

```

563 001204 104400 SCOPE
564 001206 012767 001246 176570 MOV #ERRA,BUSERR ;BUS ERROR VECTOR TO ERRA
565 001214 010700 MOV %7,%0 ;PC TO R0
566 001216 005277 177576 INC @DRWC ;ADDRESS DRWC
567 001222 010700 MOV %7,%0 ;PC TO R0
568 001224 005277 177572 INC @DRBA ;ADDRESS DRBA
569 001230 010700 MOV %7,%0 ;PC TO R0
570 001232 005077 177566 CLR @DRST ;ADDRESS DRST
571 001236 010700 MOV %7,%0 ;PC TO R0
572 001240 005277 177562 INC @DRDB ;ADDRESS DRDB
573 001244 000401 BR .+4 ;MADE IT - BRANCH OVER HALT
574 001246 104000 HLT ;BUS ERROR, R0 HAS PC OF ERROR
575 001250 012767 000006 176526 ERRAS: MOV #6,BUSERR ;RESTORE #6 TO BUS ERROR VECTOR
576 001256 104400 SCOPE

```

```

;*****
;TEST 1 DOES RESET CLEAR DRWC?
;*****

```

```

584 001260 012767 000010 010564 MOV #10,ICOUNT
585 001266 012777 177777 177524 MOV #-1,@DRWC ;ALL ONES TO DRWC
586 001274 004767 010574 JSR %7,CKSWR
587 001300 000005 RESET ;INIT
588 001302 005777 177512 TST @DRWC ;LOOKING FOR Z-BIT TO SET
589 001306 001401 BEQ .+4 ;DID DRWC GET CLEARED?
590 001310 104000 HLT ;DRWC NOT CLEAR
591 001312 104400 SCOPE

```

```

;*****
;TEST2 DOES RESET CLEAR DRBA?
;*****

```

```

600 001314 104400 SCOPE
601 001316 012777 177777 177476 MOV #-1,@DRBA ;ALL ONES TO DRBA

```

602 001324 004767 010544
 603 001330 000005
 604 001332 005777 177464
 605 001336 001401
 606 001340 104000
 607
 608
 609

JSR %7,CKSWR
 RESET
 TST @DRBA
 BEQ .+4
 HLT
 ;INIT
 ;LOOKING FOR Z-BIT TO SET
 ;DID DRBA GET CLEARED?
 ;DRBA NOT CLEAR

;;*****
 ;TEST3 CAN ALL DRMC BITS BE SET?
 ;*****

614 001342 104400
 615 001344 012767 004000 010500
 616 001352 012777 177777 177440
 617 001360 022777 177777 177432
 618 001366 001401
 619 001370 104000
 620
 621
 622
 623
 624
 625

SCOPE
 MOV #4000,ICOUNT
 MOV #-1,@DRMC
 CMP #-1,@DRMC
 BEQ .+4
 HLT
 ;SET ALL BITS IN DRMC
 ;LOOKING FOR Z-BIT TO SET
 ;SEE IF ALL BITS GOT SET
 ;ALL BITS AREN'T SET

;;*****
 ;TEST4 CAN BITS 15-01 IN DRBA BE SET?
 ;*****

626 001372 104400
 627 001374 012777 177776 177420
 628 001402 022777 177776 177412
 629 001410 001401
 630 001412 104000
 631
 632
 633
 634
 635
 636
 637
 638

SCOPE
 MOV #-2,@DRBA
 CMP #-2,@DRBA
 BEQ .+4
 HLT
 ;SET BITS 15-01 IN DRBA
 ;LOOKING FOR Z-BIT TO SET
 ;SEE IF BITS 15-01 GOT SET
 ;BITS 15-01 AREN'T SET

;;*****
 ;TEST6 TEST THAT FNCT1 CAN BE SET AND CLEARED
 ;*****

639 001414 104400
 640 001416 052777 000002 177400
 641 001424 032777 000002 177372
 642 001432 001001
 643 001434 104000
 644 001436 042777 000002 177360
 645 001444 032777 000002 177352
 646 001452 001401
 647 001454 104000
 648
 649
 650
 651
 652
 653
 654
 655

SCOPE
 BIS #BIT1,@DRST
 BIT #BIT1,@DRST
 BNE .+4
 HLT
 BIC #BIT1,@DRST
 BIT #BIT1,@DRST
 BEQ .+4
 HLT
 ;SET FNCT1
 ;TEST FNCT1
 ;IS IT SET?
 ;FNCT1 IS CLEAR
 ;CLEAR FNCT1
 ;TEST FNCT1
 ;WAS IT CLEAR
 ;FNCT1 WAS SET

;;*****
 ;TEST7 TEST THAT FNCT2 CAN BE SET AND CLEARED
 ;*****

656 001456 104400
 657 001460 052777 000004 177336

SCOPE
 BIS #BIT2,@DRST ;SET FNCT2

658	001466	032777	000004	177330	BIT	#BIT2,ORST	:TEST FNCT2
659	001474	001001			BNE	.+4	:IS IT SET?
660	001476	104000			HLT		:FNCT2 IS CLEAR
661	001500	042777	000004	177316	BIC	#BIT2,ORST	:CLEAR FNCT2
662	001506	032777	000004	177310	BIT	#BIT2,ORST	:TEST FNCT2
663	001514	001401			BEQ	.+4	:WAS IT CLEAR?
664	001516	104000			HLT		:FNCT2 WAS SET

```

:*****
:TEST10 TEST THAT FNCT3 CAN BE SET AND CLEARED
:*****

```

673	001520	104400			SCOPE		
674	001522	052777	000010	177274	BIS	#BIT3,ORST	:SET FNCT3
675	001530	032777	000010	177266	BIT	#BIT3,ORST	:TEST FNCT3
676	001536	001001			BNE	.+4	:IS IT SET?
677	001540	104000			HLT		:FNCT3 IS CLEAR
678	001542	042777	000010	177254	BIC	#BIT3,ORST	:CLEAR FNCT3
679	001550	032777	000010	177246	BIT	#BIT3,ORST	:TEST FNCT3
680	001556	001401			BEQ	.+4	:WAS IT CLEAR?
681	001560	104000			HLT		:FNCT3 WAS SET

```

:*****
:TEST11 TEST THAT XBA16 CAN BE SET AND CLEARED
:*****

```

686	001562	104400			SCOPE		
687	001564	052777	000020	177232	BIS	#BIT4,ORST	:SET XBA16
688	001572	032777	000020	177224	BIT	#BIT4,ORST	:TEST XBA16
689	001600	001001			BNE	.+4	:IS IT SET?
690	001602	104000			HLT		:XBA16 IS CLEAR
691	001604	042777	000020	177212	BIC	#BIT4,ORST	:CLEAR XBA16
692	001612	032777	000020	177204	BIT	#BIT4,ORST	:TEST XBA16
693	001620	001401			BEQ	.+4	:IS IT CLEAR
694	001622	104000			HLT		:XBA16 WAS SET

```

:*****
:TEST12 TEST THAT XBA17 CAN BE SET AND CLEARED
:*****

```

699	001624	104400			SCOPE		
700	001626	052777	000040	177170	BIS	#BIT5,ORST	:SET XBA17
701	001634	032777	000040	177162	BIT	#BIT5,ORST	:TEST XBA17
702	001642	001001			BNE	.+4	:IS IT SET?
703	001644	104000			HLT		:XBA17 IS CLEAR
704	001646	042777	000020	177150	BIC	#BIT4,ORST	:CLEAR XBA17
705	001654	032777	000020	177142	BIT	#BIT4,ORST	:TEST XBA17
706	001662	001401			BEQ	.+4	:IS IT CLEAR?
707	001664	104000			HLT		:XBA17 WAS SET

```

:*****
:TEST13 TEST THAT IE CAN BE SET AND CLEARED
:*****

```

712	001666	104400			SCOPE		
713	001670	052777	000100	177126	BIS	#BIT6,ORST	:SET IE

```

714 001676 032777 000100 177120 BIT #BIT6,DRST ;TEST IE
715 001704 001001 BNE .+4 ;IS IT SET?
716 001706 104000 HLT ;IE IS CLEAR
717 001710 042777 000100 177106 BIC #BIT6,DRST ;CLEAR IE
718 001716 032777 000100 177100 BIT #BIT6,DRST ;TEST IE
719 001724 001401 BEQ .+4 ;IS IT CLEAR?
720 001726 104000 HLT ;IE WAS SET
721
722 ;*****
723 ;TEST14 TEST THAT CYCLE CAN BE SET AND CLEARED
724 ;*****
725 001730 104400 SCOPE
726 001732 052777 000400 177064 BIS #BIT8,DRST ;SET CYCLE
727 001740 032777 000400 177056 BIT #BIT8,DRST ;TEST CYCLE
728 001746 001001 BNE .+4 ;IS IT SET?
729 001750 104000 HLT ;CYCLE WAS CLEAR
730 001752 042777 000400 177044 BIC #BIT8,DRST ;CLEAR CYCLE
731 001760 032777 000400 177036 BIT #BIT8,DRST ;TEST CYCLE
732 001766 001401 BEQ .+4 ;IS IT CLEAR?
733 001770 104000 HLT ;CYCLE WAS SET
734
735 ;*****
736 ;TEST15 TEST THAT MAINT CAN BE SET AND CLEARED
737 ;*****
738 001772 104400 SCOPE
739 001774 052777 010000 177022 BIS #BIT12,DRST ;SET MAINT
740 002002 032777 010000 177014 BIT #BIT12,DRST ;TEST MAINT
741 002010 001001 BNE .+4 ;IS IT SET?
742 002012 104000 HLT ;MAINT WAS CLEAR
743 002014 042777 010000 177002 BIC #BIT12,DRST ;CLEAR MAINT
744 002022 032777 010000 176774 BIT #BIT12,DRST ;TEST MAINT
745 002030 001401 BEQ .+4 ;IS MAINT CLEAR?
746 002032 104000 HLT ;MAINT WAS SET
747
748 ;*****
749 ;TEST 16 TEST THAT ALL DRST R/W BITS CAN BE SET AND CLEARED
750 ;*****
751 002034 104400 SCOPE
752 002036 052777 010576 176760 BIS #10576,DRST ;SET FOLLOWING: MAINT(12), CYCLE(08), IE(06), XBA17(05),
753 ; XBA16(04),FNCT3(03),FUNCT2(02),FNCT1(01)
754 002044 032777 000002 176752 BIT #BIT1,DRST ;TEST FNCT1
755 002052 001001 BNE .+4 ;IS IT SET?
756 002054 104000 HLT ;FNCT1 IS CLEAR
757 002056 032777 000004 176740 BIT #BIT2,DRST ;TEST FNCT2
758 002064 001001 BNE .+4 ;IS IT SET?
759 002066 104000 HLT ;FNCT2 IS CLEAR
760 002070 032777 000010 176726 BIT #BIT3,DRST ;TEST FNCT3
761 002076 001001 BNE .+4 ;IS IT SET?
762 002100 104000 HLT ;FNCT3 IS CLEAR
763 002102 032777 000020 176714 BIT #BIT4,DRST ;TEST XBA16
764 002110 001001 BNE .+4 ;IS IT SET?
765 002112 104000 HLT ;XBA16 IS CLEAR
766 002114 032777 000040 176702 BIT #BIT5,DRST ;TEST XBA17
767 002122 001001 BNE .+4 ;IS IT SET?
768 002124 104000 HLT ;XBA17 IS CLEAR
769 002126 032777 000100 176670 BIT #BIT6,DRST ;TEST IE

```



```

770 002134 001001      BNE      .+4      ; IS IT SET?
771 002136 104000      HLT      ; IE IS CLEAR
772 002140 032777 000400 176656 BIT      #BIT8,DRST ; TEST CYCLE
773 002146 001001      BNE      .+4      ; IS CYCLE SET?
774 002150 104000      HLT      ; CYCLE IS CLEAR
775 002152 032777 010000 176644 BIT      #BIT12,DRST ; TEST MAINT
776 002160 001001      BNE      .+4      ; IS MAINT SET?
777 002162 104000      HLT      ; MAINT IS CLEAR
778 002164 042777 010576 176632 BIC      #10576,DRST ; CLEAR ALL R/W BITS IN DRST
779 002172 032777 000002 176624 BIT      #BIT1,DRST ; TEST FNCT1
780 002200 001401      BEQ      .+4      ; IS FNCT1 CLEAR?
781 002202 104000      HLT      ; FNCT1 IS SET
782 002204 032777 000004 176612 BIT      #BIT2,DRST ; TEST FNCT2
783 002212 001401      BEQ      .+4      ; IS FNCT2 CLEAR?
784 002214 104000      HLT      ; FNCT2 IS SET
785 002216 032777 000010 176600 BIT      #BIT3,DRST ; TEST FNCT3
786 002224 001401      BEQ      .+4      ; IS FNCT3 CLEAR?
787 002226 104000      HLT      ; FNCT3 IS SET
788 002230 032777 000020 176566 BIT      #BIT4,DRST ; TEST XBA16
789 002236 001401      BEQ      .+4      ; IS XBA16 CLEAR?
790 002240 104000      HLT      ; XBA16 IS SET
791 002242 032777 000040 176554 BIT      #BIT5,DRST ; TEST XBA17
792 002250 001401      BEQ      .+4      ; IS XBA17 CLEAR?
793 002252 104000      HLT      ; XBA17 IS SET
794
795 002254 032777 000100 176542 BIT      #BIT6,DRST ; TEST IE
796 002262 001401      BEQ      .+4      ; IS IE CLEAR?
797 002264 104000      HLT      ; IE IS SET
798 002266 032777 000400 176530 BIT      #BIT8,DRST ; TEST CYCLE
799 002274 001401      BEQ      .+4      ; IS CYCLE CLEAR?
800 002276 104000      HLT      ; CYCLE IS SET
801 002300 032777 010000 176516 BIT      #BIT12,DRST ; TEST MAINT
802 002306 001401      BEQ      .+4      ; IS MAINT CLEAR?
803 002310 104000      HLT      ; MAINT IS SET
804
805 ; *****
806 ; TEST17 ALL R/W BITS IN DRST CAN BE SET AND RESET TO ZERO, THAT READY
807 ; IS SET, NEX IS CLEAR, AND GO IS READ AS A 0.
808 ; *****
809 002312 104400      SCOPE
810 002314 012767 000010 007530 MOV      #10,ICOUNT
811 002322 052777 010576 176474 BIS      #10576,DRST ; SET FOLLOWING: MAINT(12),CYCLE(08),IE(06),XBA17(05),
812 ; XBA16(04),FNCT3(03),FNCT2(02),FNCT1(01)
813 002330 017701 176470 MOV      DRST,%1 ; MOVE (DRST) TO R1
814 002334 052701 167201 BIS      #167201,%1 ; SETS BITS IN R1 THAT WERE NOT SET IN DRST
815 002340 005201      INC      %1 ; R1 SHOULD GO FROM -1 TO ZERO
816 002342 001401      BEQ      .+4      ; WERE ALL DRST R/W BITS SET?
817 002344 104000      HLT      ; NOT ALL BITS WERE SET
818 002346 004767 007522 JSR      %7,CKSWR
819 002352 000005      RESET ; CLEAR ALL DRST R/W BITS
820 002354 017701 176444 MOV      DRST,%1 ; MOVE (DRST) TO R1
821 002360 042701 127200 BIC      #127200,%1 ; CLEAR ALL BITS EXCEPT R/W BITS, NEX, AND GO
822 002364 001401      BEQ      .+4      ; SHOULD EQUAL ZERO
823 002366 104000      HLT      ; RESET DIDN'T LEAVE DRST AS IT SHOULD HAVE
824
825 ; *****

```

```

826 ;
827 ; *****
828 002370 104400 SCOPE
829 002372 012767 004000 007452 MOV #4000,ICOUNT
830 002400 012777 052525 176412 MOV #052525,DRWC ;ALT 0'S AND 1'S TO DRWC
831 002406 022777 052525 176404 CMP #052525,DRWC ;LOOKING FOR Z-BIT TO SET
832 002414 001401 BEQ .+4 ;DOES DRWC HAVE THE CORRECT PATTERN?
833 002416 104000 HLT ;DRWC DOESN'T HAVE THE CORRECT PATTERN
834 002420 012777 125252 176372 MOV #125252,DRWC ;ALT 1'S AND 0'S TO DRWC
835 002426 022777 125252 176364 CMP #125252,DRWC ;LOOKING FOR Z-BIT TO SET
836 002434 001401 BEQ .+4 ;DOES DRWC HAVE THE CORRECT PATTERN?
837 002436 104000 HLT ;DRWC DOESN'T HAVE THE CORRECT PATTERN
838 ;
839 ; *****
840 ; TEST21 CAN DRBA HOLD ALTERNATE ONE'S AND ZERO'S
841 ; *****
842 002440 104400 SCOPE
843 002442 012777 052524 176352 MOV #052524,DRBA ;ALT 0'S AND 1'S TO DRBA
844 002450 022777 052524 176344 CMP #052524,DRBA ;LOOKING FOR Z-BIT TO SET
845 002456 001401 BEQ .+4 ;DOES DRBA HAVE THE CORRECT PATTERN?
846 002460 104000 HLT ;DRBA DOESN'T HAVE THE CORRECT PATTERN
847 002462 012777 125252 176332 MOV #125252,DRBA ;ALT 1'S AND 0'S TO DRBA
848 002470 022777 125252 176324 CMP #125252,DRBA ;LOOKING FOR Z-BIT TO SET
849 002476 001401 BEQ .+4 ;DOES DRBA HAVE THE CORRECT PATTERN?
850 002500 104000 HLT ;DRBA DOESN'T HAVE THE CORRECT PATTERN
851 ;
852 ; *****
853 ; TEST22 INCREMENTING PATTERN TO WRAP-AROUND IN DRWC
854 ; *****
855 002502 104400 SCOPE
856 002504 005067 007342 CLR ICOUNT
857 002510 005001 CLR %1 ;SET-UP
858 002512 005077 176302 CLR DRWC ;SET-UP
859 002516 020177 176276 INCMC: CMP %1,DRWC ;SEE IF THEY ARE EQUAL
860 002522 001401 BEQ .+4 ;ARE THEY EQUAL?
861 002524 104000 HLT ;THEY'RE NOT EQUAL
862 002526 005277 176266 INC DRWC ;GET NEXT NUMBER
863 002532 005201 INC %1 ;GET NEXT NUMBER
864 002534 001370 BNE INCMC ;DONE WITH TEST? IF NOT CONTINUE
865 ;
866 ; *****
867 ; TEST23 INCREMENTING PATTERN TO WRAP-AROUND IN DRBA
868 ; *****
869 002536 104400 SCOPE
870 002540 005001 CLR %1 ;SET-UP
871 002542 005077 176254 CLR DRBA ;SET-UP
872 002546 020177 176250 INCBA: CMP %1,DRBA ;SEE IF THEY ARE EQUAL
873 002552 001401 BEQ .+4 ;ARE THEY EQUAL?
874 002554 104000 HLT ;THEY'RE NOT EQUAL
875 002556 062777 000002 176236 ADD #2,DRBA ;GET NEXT NUMBER
876 002564 062701 000002 ADD #2,%1 ;GET NEXT NUMBER
877 002570 001366 BNE INCBA ;DONE WITH TEST? IF NOT CONTINUE
878 ;
879 ; *****
880 ; TEST25 TEST THAT DR11-B DOES NOT INTERRUPT WITH PROC AT LEVEL 7
881 ; *****

```

```

882 ;*****
883 002572 104400 SCOPE
884 002574 012767 004000 007250 MOV #4000,ICOUNT
885 002502 012777 000340 176206 MOV #340,APSW ;STATUS AT LEVEL 7
886 002610 032777 000200 176206 BIT #BIT7,DRST ;CHECK READY BIT
887 002616 001010 BNE P7INV ;IS IT SET
888 002620 004767 007250 JSR %7,CKSWR
889 002624 000005 RESET ;INIT TO SET READY
890 002626 032777 000200 176170 BIT #BIT7,DRST ;SEE IF READY IS SET NOW
891 002634 001001 BNE .+4 ;IS READY SET?
892 002636 104000 HLT ;READY CAN'T BE SET BY INIT
893 002640 012777 002666 176166 P7INV: MOV #P7ERR,DRINV ;SET UP INT VECTOR
894 002646 052777 000100 176150 BIS #BIT6,DRST ;SET IE
895 002654 000240 NOP
896 002656 042777 000100 176140 BIC #BIT6,DRST ;DR11-B DIDN'T INTERRUPT - CLEAR IE
897 002664 000401 BR .+4
898 002666 104000 P7ERR: HLT ;DR11-B INTERRUPTED, BUT IT SHOULDN'T HAVE
899
900 ;*****
901 ; TEST26 TEST THAT DR11-B DOES NOT INTERRUPT WITH PROC AT LEVEL 6
902 ;*****
903 002670 104400 SCOPE
904 002672 012777 000300 176116 MOV #300,APSW ;STATUS AT LEVEL 6
905 002700 032777 000200 176116 BIT #BIT7,DRST ;CHECK READY BIT
906 002706 001010 BNE P6INV ;IS IT SET?
907 002710 004767 007160 JSR %7,CKSWR
908 002714 000005 RESET ;INIT TO SET READY
909 002716 032777 000200 176100 BIT #BIT7,DRST ;SEE IF READY IS SET NOW
910 002724 001001 BNE .+4 ;IS READY SET?
911 002726 104000 HLT ;READY CAN'T BE SET BY INIT
912 002730 012777 002756 176076 P6INV: MOV #P6ERR,DRINV ;SET UP INT VECTOR
913 002736 052777 000100 176060 BIS #BIT6,DRST ;SET IE
914 002744 000240 NOP
915 002746 042777 000100 176050 BIC #BIT6,DRST ;DR11-B DIDN'T INTERRUPT - CLEAR IE
916 002754 000401 BR .+4
917 002756 104000 P6ERR: HLT ;DR11-B INTERRUPTED, BUT IT SHOULDN'T HAVE
918
919 ;*****
920 ; TEST27 TEST THAT DR11-B DOES NOT INTERRUPT WITH PROC AT LEVEL 5
921 ;*****
922 ;*****
923 002760 104400 SCOPE
924 002762 012777 000240 176026 MOV #240,APSW ;STATUS AT LEVEL 5
925 002770 032777 000200 176026 BIT #BIT7,DRST ;CHECK READY BIT
926 002776 001010 BNE P5INV ;IS IT SET?
927 003000 004767 007070 JSR %7,CKSWR
928 003004 000005 RESET ;INIT TO SET READY
929 003006 032777 000200 176010 BIT #BIT7,DRST ;SEE IF READY IS SET NOW
930 003014 001001 BNE .+4 ;IS IT SET?
931 003016 104000 HLT ;RDY CAN'T BE SET BY INIT
932 003020 012777 003046 176006 P5INV: MOV #P5ERR,DRINV ;SET UP INT VECTOR
933 003026 052777 000100 175770 BIS #BIT6,DRST ;SET IE
934 003034 000240 NOP
935 003036 042777 000100 175760 BIC #BIT6,DRST ;DR11-B DIDN'T INTERRUPT - CLEAR IE
936 003044 000401 BR .+4
937 003046 104000 P5ERR: HLT ;DR11-B INTERRUPTED, BUT IT SHOULDN'T HAVE

```

```

938
939
940
941
942 003050 104400
943 003052 012777 000200 175736
944 003060 032777 000200 175736
945 003066 001010
946 003070 004767 007000
947 003074 000005
948 003076 032777 000200 175720
949 003104 001001
950 003106 104000
951 003110 012777 003130 175716 P4INV:
952 003116 052777 000100 175700
953 003124 000240
954 003126 104000
955 003130 005077 175670 P4INT:
956 003134 022626
957 003136 012777 000126 175670
958
959
960
961
962
963 003144 104400
964 003146 005077 175652
965 003152 032777 000016 175644
966 003160 001401
967 003162 104000
968 003164 052777 000002 175632
969 003172 032777 001000 175624
970 003200 001001
971 003202 104000
972 003204 032777 006000 175612
973 003212 001401
974 003214 104000
975 003216 005077 175602
976 003222 052777 000004 175574
977 003230 032777 002000 175566
978 003236 001001
979 003240 104000
980 003242 032777 005000 175554
981 003250 001401
982 003252 104000
983 003254 005077 175544
984 003260 052777 000010 175536
985 003266 032777 004000 175530
986 003274 001001
987 003276 104000
988 003300 032777 003000 175516
989 003306 001401
990 003310 104000
991 003312 005077 175506
992
993

```

```

;*****
; TEST30 TEST THAT DR11-B DOES INTERRUPT WITH PROC AT LEVEL 4
;*****
SCOPE
MOV #200,DRST ;STATUS AT LEVEL 4
BIT #BIT7,DRST ;CHECK READY BIT
BNE P4INV ;IS IT SET?
JSR #7,CKSWR
RESET ;INIT TO SET READY
BIT #BIT7,DRST ;SEE IF READY IS SET NOW
BNE .+4 ;IS IT SET
HLT ;READY CAN'T BE SET BY INIT
P4INV: MOV #P4INT,DRINV ;SET UP INT VECTOR
BIS #BIT6,DRST ;SET IE
NOP
HLT ;DR11-B DIDN'T INTERRUPT
P4INT: CLR DRST ;CLEAR IE
CMP (%6)+,(%6)+ ;REPOSITION THE STACK AFTER AN INTERRUPT
MOV #126,DRINV ;RESTORE INTERRUPT VECTOR

;*****
; TEST31 TEST THAT FNCT BITS CONTROL DSTAT BITS (M968 MUST BE USED IN USER SLOTS)
;*****
SCOPE
CLR DRST ;CLEAR FUNCTION BITS
BIT #16,DRST ;CHECK FUNCTION BITS
BEQ .+4 ;FUNCTION BITS CLEAR?
HLT ;FUNCTION BITS NOT CLEAR
BIS #BIT1,DRST ;SET FNCT1
BIT #BIT9,DRST ;CHECK DSTAT C
BNE .+4 ;IS IT SET?
HLT ;DSTAT C IS CLEAR
BIT #6000,DRST ;CHECK THAT DSTAT A AND DSTAT B ARE CLEAR
BEQ .+4 ;ARE THEY CLEAR?
HLT ;DSTAT A AND/OR DSTAT B IS SET
CLR DRST ;CLEAR DRST
BIS #BIT2,DRST ;SET FNCT2
BIT #BIT10,DRST ;CHECK DSTAT B
BNE .+4 ;IS IT SET?
HLT ;DSTAT B IS CLEAR
BIT #5000,DRST ;CHECK THAT DSTAT A AND DSTAT C ARE CLEAR
BEQ .+4 ;ARE THEY CLEAR?
HLT ;DSTAT A AND/OR DSTAT B IS SET
CLR DRST ;CLEAR DRST
BIS #BIT3,DRST ;SET FNCT3
BIT #BIT11,DRST ;CHECK DSTAT A
BNE .+4 ;IS IT SET?
HLT ;DSTAT A IS CLEAR
BIT #3000,DRST ;CHECK THAT DSTAT B AND DSTAT C ARE CLEAR
BEQ .+4 ;ARE THEY CLEAR?
HLT ;DSTAT B AND/OR DSTAT C IS SET
CLR DRST ;CLR DRST

;*****
; TEST 33 TEST FOR 1 DATI NPR TRANSFER (WITH M968 IN USER SLOTS)
;*****

```

```

994                                     ;;*****
995 003316 104400                       ;;
996 003320 005777 175500                TNPR1: SCOPE
997 003324 100027                       TST      2DRST      ;CHECK ERROR BIT
998 003326 032777 020000 175470        BPL      NRRDY     ;IS IT CLEAR?
999 003334 001401                       BIT      #BIT13,2DRST ;CHECK ATTN
1000 003336 104000                      BEQ      .+4       ;IS ATTN CLEAR
1001 003340 032777 040000 175456        HLT                                     ;ATTN IS SET
1002 003346 001410                      BIT      #BIT14,2DRST ;CHECK NEX
1003 003350 042777 040000 175446        BEQ      N1413    ;IS NEX CLEAR?
1004 003356 032777 040000 175440        BIC      #BIT14,2DRST ;TRY TO CLEAR NEX
1005 003364 001401                      BIT      #BIT14,2DRST ;CHECK AGAIN
1006 003366 104000                      BEQ      .+4       ;NEX STILL SET
1007 003370 005077 175426                N1413: CLR      2DRBA  ;NEX CAN'T BE CLEARED BY MOVING A 0 TO IT
1008 003374 005777 175424                TST      2DRST     ;TRY TO CLEAR BAOF
1009 003400 001401                      BEQ      .+4       ;CHECK ERROR BIT AGAIN
1010 003402 104000                      HLT                                     ;IS IT CLEAR
1011 003404 012777 177777 175406        NRRDY: MOV      #-1,2DRWC ;ERROR CAUSED BY SOMETHING OTHER THAN NEX,ATTN, OR BAOF
1012 003412 012777 001036 175402        MOV      #NPR1,2DRBA ;SET UP FOR 1 TRANSFER
1013 003420 005077 175402                CLR      2DRDB     ;TRANSFER FROM BUS ADDRESS IN NPR1
1014 003424 012767 052525 175404        MOV      #52525,NPR1 ;GET READY TO RECEIVE DATA
1015 003432 012777 003476 175374        MOV      #INTB,2DRINV ;SET UP TRANSFER DATA
1016 003440 012777 000005 175362        MOV      #5,2DRVS  ;INTERRUPT VECTOR TO INTB
1017 003446 005077 175344                CLR      2PSW      ;INTERRUPT PRIORITY TO LEVEL 5
1018 003452 012777 000101 175344        MOV      #101,2DRST ;LET THE DR11-B INTERRUPT
1019 003460 005067 000002                CLR      1$+2     ;IE AND DO TO DRST
1020 003464 005227 000001                1$: INC      #1    ;WAIT FOR NPR AND INTERRUPT
1021 003470 001375                      BNE      1$
1022 003472 104000                      HLT                                     ;NO DR11-B INTERRUPT
1023 003474 000424                      BR       T33CLR    ;CLEAR IE
1024 003476 004767 003342                INTB: JSR      %7,ERRCHK
1025 003502 005777 175312                TST      2DRWC     ;TEST DRWC
1026 003506 001401                      BEQ      .+4       ;IS DRWC EQUAL TO ZERO?
1027 003510 104000                      HLT                                     ;DRWC NOT EQUAL TO ZERO
1028 003512 022777 001040 175302        CMP      #NPR1+2,2DRBA ;COMPARE CORRECT DRBA WITH DRBA
1029 003520 001401                      BEQ      .+4       ;IS THE DRBA CORRECT?
1030 003522 104000                      HLT                                     ;DRBA IS WRONG
1031 003524 022777 052525 175274        CMP      #52525,2DRDB ;CHECK FOR CORRECT DATA
1032 003532 001401                      BEQ      .+4       ;DATA GET TRANSFERRED?
1033 003534 104000                      HLT                                     ;BAD DATA IN DRDB
1034 003536 004767 003212                JSR      %7,NORMAL ;RESTORE STACK
1035 003542 022626                      CMP      (%6)+,(%6)+ ;GO TO NEXT TEST (NPR OUT)
1036 003544 000403                      BR       TNPRO
1037 003546 005077 175252                T33CLR: CLR     2DRST ;CLEAR IE
1038 003552 000662                      BR       TNPR1    ;TRY TEST AGAIN
1039                                     ;;*****
1040                                     ;; TEST 34 TEST FOR 1 DATO NPR TRANSFER (WITH M968 IN USER SLOTS)
1041                                     ;;*****
1042 003554 104400                       TNPRO: SCOPE
1043 003556 012777 177777 175234        MOV      #-1,2DRWC ;SET UP FOR 1 TRANSFER
1044 003564 012777 001036 175230        MOV      #NPR1,2DRBA ;TRANSFER TO BUS ADDRESS IN NPR1
1045 003572 005067 175240                CLR      NPR1     ;GET READY TO RECEIVE DATA
1046 003576 012777 052525 175222        MOV      #52525,2DRDB ;SET UP TO TRANSFER DATA
1047 003604 012777 003650 175222        MOV      #INTC,2DRINV ;INTERRUPT VECTOR TO INTC
1048 003612 016777 175214 175210        MOV      DRINL,2DRVS ;INTERRUPT STATUS TO LEVEL DRINL
1049 003620 005077 175172                CLR      2PSW     ;PROC STATUS TO ZERO

```

```

1050 003624 012777 000103 175172      MOV      #103,DRST      ;IE, FNCT1(C1 CONTROL), AND DO TO DRST
1051 003632 005067 000002          CLR      1$+2          ;WAIT FOR NPR AND INTER
1052 003636 005227 000001      IS:      INC      #1
1053 003642 001375          BNE      1$
1054 003644 104000          HLT
1055 003646 000424          BR       T34CLR        ;NO DR11-B INTERRUPT
1056 003650 004767 003170      INTC:   JSR      %7,ERRCHK ;CLEAR IE
1057 003654 005777 175140          TST      DRWC          ;TEST DRWC
1058 003660 001401          BEQ      .+4           ;IS DRWC EQUAL TO ZERO?
1059 003662 104000          HLT          ;DRWC EQUAL TO ZERO
1060 003664 022777 001040 175130      CMP      #NPR1+2,DRBA ;COMPARE CORRECT DRBA WITH DRBA
1061 003672 001401          BEQ      .+4           ;IS THE DRBA CORRECT?
1062 003674 104000          HLT          ;DRBA IS WRONG
1063
1064 003676 026727 175134 052525      CMP      NPR1,#52525   ;CHECK FOR CORRECT DATA
1065 003704 001401          BEQ      .+4           ;CORRECT DATA TRANSFERRED?
1066 003706 104000          HLT          ;BAD DATA
1067 003710 004767 003040          JSR      %7,NORMAL
1068 003714 022626          CMP      (%6)+,(%6)+  ;RESTORE STACK
1069 003716 000403          BR       T35          ;GO TO NEXT TEST
1070 003720 005077 175100      T34CLR: CLR      DRST   ;CLEAR IE
1071 003724 000713          BR       TNPRO        ;TRY TEST AGAIN
1072
1073      ;*****
1074      ;TEST 35 STRING OF 10 DATI'S (WITH M968 IN USER SLOTS)
1075      ;*****
1076 003726 104400      T35:   SCOPE
1077 003730 012767 000020 175110      MOV      #20,BUFLEN   ;LENGTH OF BUFFER=20
1078 003736 004767 002554          JSR      %7,LODBUF   ;LOAD THE BUFFER WITH INCREMENTING PATTERN
1079 003742 006267 175100          ASR      BUFLEN      ;BUFLEN=10
1080 003746 016767 175074 175100      MOV      BUFLEN,WLEN  ;PREPARE NUMBER FOR DRWC
1081 003754 005467 175074          NEG      WLEN        ;2'S COMPLEMENT OF BUFLEN
1082 003760 016777 175070 175032      MOV      WLEN,DRWC   ;SET UP DRWC
1083 003766 016777 175050 175026      MOV      INBUF,DRBA  ;SET UP DRBA
1084 003774 012777 177777 175024      MOV      #-1,DRDB   ;MAINT AIDE
1085 004002 012777 006620 175024      MOV      INTA,DRINV  ;INT VECTOR TO INTA
1086 004010 016777 175016 175012      MOV      DRINL,DRVS  ;INT VECTOR TO PRIORITY DRINL
1087 004016 005077 174774          CLR      DPSW        ;LET THE DR11-B INTERRUPT
1088 004022 012777 000101 174774      MOV      #101,DRST  ;IE AND DO TO DRST
1089 004030 000777          BR
1090 004032 022777 000007 174766      CMP      #7,DRDB     ;CHECK THAT WORD #10 OF INBUF IS IN DRBA
1091 004040 001401          BEQ      .+4         ;IS IT?
1092 004042 104000          HLT          ;BAD DATA IN DRDB
1093
1094      ;*****
1095      ;TEST 36 STRING OF 10 DATO'S (WITH M968 IN USER SLOTS)
1096      ;*****
1097 004044 104400      SCOPE
1098 004046 012767 000020 174772      MOV      #20,BUFLEN   ;LENGTH OF BUFFER=20
1099 004054 004767 002436          JSR      %7,LODBUF   ;LOAD THE BUFFER WITH INCREMENTING PATTERN
1100 004060 006267 174762          ASR      BUFLEN      ;BUFLEN=10
1101 004064 016767 174756 174762      MOV      BUFLEN,WLEN  ;PREPARE NUMBER FOR DRWC
1102 004072 005467 174756          NEG      WLEN        ;2'S COMPLEMENT OF BUFLEN
1103 004076 016777 174752 174714      MOV      WLEN,DRWC   ;SET UP DRWC
1104 004104 016777 174732 174710      MOV      INBUF,DRBA  ;SET UP DRBA
1105 004112 012777 052525 174706      MOV      #52525,DRDB ;SET UP DRDB

```

```

1106 004120 012777 006620 174706      MOV      #INTA,DRINV      ; INTERRUPT VECTOR TO INTA
1107 004126 016777 174700 174674      MOV      DRINL,DRVS      ; INTERRUPT VECTOR TO PRIORITY DRINL
1108 004134 005077 174656                CLR      DPSW            ; LET THE DR11-B INTERRUPT
1109 004140 012777 000103 174656      MOV      #103,DRST       ; IE, FNCT1(C1 CONTROL), AND DO TO DRST
1110 004146 000777                BR                          ; WAIT FOR INTERRUPT
1111 004150 004767 002622      JSR      %7,DATOCK       ; CHECK INBUF
1112
1113      ; *****
1114      ; TEST 37 STRING OF 200 DATI'S
1115      ; *****
1116 004154 104400                SCOPE
1117 004156 012767 000200 174662      MOV      #200,BUFLEN     ; LENGTH OF BUFFER=200
1118 004164 004767 002326                JSR      %7,LOADBUF      ; LOAD THE BUFFER WITH INCREMENTING PATTERN
1119 004170 016767 174652 174656      MOV      BUFLN,WCLN      ; PREPARE NUMBER FOR DRWC
1120 004176 005467 174652                NEG      WCLN            ; 2'S COMPLEMENT OF BUFLN
1121 004202 016777 174646 174610      MOV      WCLN,DRWC       ; SET UP DRWC
1122 004210 016777 174626 174604      MOV      INBUF,DRBA      ; SET UP DRBA
1123 004216 012777 177777 174602      MOV      #-1,DRDB        ; MAINT AIDE
1124 004224 012777 006620 174602      MOV      #INTA,DRINV     ; INT VECTOR TO INTA
1125 004232 016777 174574 174570      MOV      DRINL,DRVS      ; INT VECTOR TO PRIORITY DRINL
1126 004240 005077 174552                CLR      DPSW            ; LET THE DR11-B INTERRUPT
1127 004244 012777 000101 174552      MOV      #101,DRST       ; IE AND DO TO DRST
1128 004252 000777                BR                          ; WAIT FOR INTERRUPT
1129 004254 022777 000177 174544      CMP      #177,DRDB       ; CHECK THAT WORD #200 OF INBUF IS IN DRBA
1130 004262 001401                BEQ      .+4             ; IS IT?
1131 004264 104000                HLT                       ; BAD DATA IN DRDB
1132      ; *****
1133      ; TEST 40 STRING OR 200 DATO'S
1134      ; *****
1135 004266 104400                SCOPE
1136 004270 012767 000201 174550      MOV      #201,BUFLEN     ; LENGTH OF BUFFER=201
1137 004276 004767 002214                JSR      %7,LOADBUF      ; LOAD THE BUFFER WITH INCREMENTING PATTERN
1138 004302 005367 174540                DEC      BUFLN           ; BUFLN=200
1139 004306 016767 174534 174540      MOV      BUFLN,WCLN      ; PREPARE NUMBER FOR DRWC
1140 004314 005467 174534                NEG      WCLN            ; 2'S COMPLEMENT OF BUFLN
1141 004320 016777 174530 174472      MOV      WCLN,DRWC       ; SET UP DRWC
1142 004326 016777 174510 174466      MOV      INBUF,DRBA      ; SET UP DRBA
1143 004334 012777 052525 174464      MOV      #52525,DRDB     ; SET UP DRDB
1144 004342 012777 006620 174464      MOV      #INTA,DRINV     ; INTERRUPT VECTOR TO INTA
1145 004350 016777 174456 174452      MOV      DRINL,DRVS      ; INTERRUPT VECTOR TO PRIORITY DRINL
1146 004356 005077 174434                CLR      DPSW            ; LET THE DR11-B INTERRUPT
1147 004362 012777 000103 174434      MOV      #103,DRST       ; IE, FNCT1, AND DO TO DRST
1148 004370 000777                BR                          ; WAIT FOR INTERRUPT
1149 004372 004767 002400      JSR      %7,DATOCK       ; CHECK INBUF
1150
1151      ; *****
1152      ; TEST 42 TEST THAT DOING A DATO TO THE DIODE MEMORY CAUSES NEX
1153      ; *****
1154 004376 104400                SCOPE
1155 004400 012777 177776 174412      MOV      #-2,DRWC        ; SET UP DRWC
1156 004406 016777 174426 174406      MOV      DIOMEM,DRBA     ; SET UP DRBA
1157 004414 012777 004450 174412      MOV      #NEXCHK,DRINV   ; INTERRUPT VECTOR TO NEXCHK
1158 004422 016777 174404 174400      MOV      DRINL,DRVS      ; INTERRUPT STATUS TO LEVEL DRINL
1159 004430 005077 174362                CLR      DPSW            ; LET THE DR11-B INTERRUPT
1160 004434 012777 000163 174362      MOV      #163,DRST       ; IE, FNCT1, XBA17, XBA16, AND GO TO DRST
1161 004442 005237 177560      INC      @#177560        ; WAIT FOR INTERRUPT

```

```

1162 004446 104000          HLT
1163 004450 042777 000100 174346 NEXCHK: BIC      #BIT6,DRST  ;NO DR11-B INTERRUPT
1164 004456 005777 174342      TST      DRST    ;CLEAR INTERRUPT ENABLE
1165 004462 001001          BNE      .+4     ;TEST DRST
1166 004464 104000          HLT          ;ERROR SET?
1167 004466 105777 174332      TSTB     DRST    ;ERROR NOT SET
1168 004472 001001          BNE      .+4     ;TEST FOR READY
1169 004474 104000          HLT          ;READY SET?
1170 004476 032777 040000 174320      BIT      #BIT14,DRST ;READY ISN'T SET
1171 004504 001001          BNE      .+4     ;CHECK NEX
1172 004506 104000          HLT          ;NEX SET?
1173 004510 042777 040000 174306      BIC      #BIT14,DRST ;NEX IS CLEAR
1174 004516 022626          CMP      (%6)+,(%6)+ ;CLEAR MEX
1175 004520 004767 002230      JSR     %7,NORMAL ;RESTORE THE STACK
1176
1177
1178 ;*****
1179 ;TEST 43 TEST THAT BAOF FORCES ERROR AND READY AND THAT BAOF IS
1180 ;CLEARED BY CLEARING THE DRBA OR A RESET
1181 ;*****
1181 004524 104400          SCOPE
1182 004526 012767 000010 005316      MOV     #10,ICOUNT
1183 004534 012777 177760 174256      MOV     @-20,DRWC   ;SET UP DRWC
1184 004542 012777 177776 174252      MOV     @-2,DRBA   ;SET UP DRBA FOR PROC STATUS ADDRESS
1185 004550 012777 004604 174256      MOV     @BAOFCK,DRINV ;INTERRUPT VECTOR TO BAOFCK
1186 004556 016777 174250 174244      MOV     DRINL,DRVS  ;INTERRUPT STATUS TO LEVEL DRINL
1187 004564 005077 174226          CLR     @PSW        ;LET THE DR11-B INTERRUPT
1188 004570 012777 000163 174226      MOV     #163,DRST  ;I.E. FNCT1, XBA17, XBA16, AND GO TO DRST
1189 004576 005237 177560          INC     @#177560   ;WAIT FOR INTERRUPT
1190 004602 104000          HLT          ;NO DR11-B INTERRUPT
1191 004604 042777 000100 174212      BAOFCK: BIC     #BIT6,DRST ;CLEAR INTERRUPT ENABLE
1192 004612 022626          CMP     (%6)+,(%6)+ ;RESTORE THE STACK
1193 004614 005777 174204          TST     DRST      ;TEST DRST
1194 004620 100401          BMI     .+4       ;ERROR SET?
1195 004622 104000          HLT          ;ERROR NOT SET
1196 004624 105777 174174          TSTB    DRST      ;TEST FOR READY
1197 004630 100401          BMI     .+4       ;READY SET?
1198 004632 104000          HLT          ;READY ISN'T SET
1199 004634 042777 040000 174162      BIC     #BIT14,DRST ;CLEAR NEX
1200 004642 032777 060000 174154      BIT     #60000,DRST ;CHECK NEX AND ATTN
1201 004650 001401          BEQ     .+4       ;ARE THEY CLEAR?
1202 004652 104000          HLT          ;NEX AND/OR ATTN IS SET
1203 004654 005777 174144          TST     DRST      ;TEST FOR ERROR
1204 004660 100401          BMI     .+4       ;IS ERROR SET?
1205 004662 104000          HLT          ;ERROR IS CLEAR
1206 004664 005077 174132          CLR     DRBA      ;CLEAR BAOF
1207 004670 005777 174130          TST     DRST      ;CHECK ERROR
1208 004674 100001          BPL     .+4       ;SHOULD BE CLEAR
1209 004676 104000          HL      ;CLEARING DRBA DIDN'T CLEAR BAOF
1210
1211 004700 012777 177776 174114      MOV     @-2,DRBA   ;SET UP DRBA FOR PROC STATUS ADDRESS
1212 004706 012777 004742 174120      MOV     @BAOCK1,DRINV ;INTERRUPT VECTOR TO BAOCK1
1213 004714 016777 174112 174106      MOV     DRINL,DRVS  ;INTERRUPT STATUS TO LEVEL DRINL
1214 004722 005077 174070          CLR     @PSW        ;LET THE DR11-B INTERRUPT
1215 004726 012777 000163 174070      MOV     #163,DRST  ;I.E., XBA17, XBA16, FNCT1, AND GO TO DRST
1216 004734 005237 177560          INC     @#177560   ;WAIT FOR INTERRUPT
1217 004740 104000          HLT          ;NO DR11-B INTERRUPT

```



```

1218 004742 042777 000100 174054 BROCK1: BIC #BIT6,DRST ;CLEAR INTERRUPT ENABLE
1219 004750 022526 CMP (%6)+,(%6)+ ;RESTORE THE STACK
1220 004752 005777 174046 TST DRST ;TEST DRST
1221 004756 100401 BMI .+4 ;ERROR SET?
1222 004760 104000 HLT ;ERROR NOT SET
1223 004762 105777 174036 TSTB DRST ;TEST FOR READY
1224 004766 100401 BMI .+4 ;READY SET?
1225 004770 104000 HLT ;READY ISN'T SET
1226 004772 042777 040000 174024 BIC #BIT14,DRST ;CLEAR NEX
1227 005000 032777 060000 174016 BIT #60000,DRST ;CHECK NEX AND ATTN
1228 005006 001401 BEQ .+4 ;ARE THEY CLEAR?
1229 005010 104000 HLT ;NEX AND/OR ATTN IS SET
1230 005012 005777 174006 TST DRST ;TEST FOR ERROR
1231 005016 100401 BMI .+4 ;IS ERROR SET?
1232 005020 104000 HLT ;ERROR IS CLEAR
1233 005022 004767 005046 JSR %7,CKSWR
1234 005026 000005 RESET ;INIT
1235 005030 005777 173770 TST DRST ;CHECK ERROR
1236 005034 100001 BPL .+4 ;SHOULD BE CLEAR
1237 005036 104000 HLT ;RESET DIDN'T CLEAR BROF
1238 005040 004767 001710 JSR %7,NORMAL
1239
1240 ;*****
1241 ; TEST 44 TEST THAT RESET CLEARS DRDB
1242 ;*****
1243 005044 104400 SCOPE
1244 005046 012767 000010 004776 MOV #10,ICOUNT
1245 005054 012777 177777 173744 MOV #-1,DRDB ;ALL ONES TO DRDB
1246 005062 004767 005006 JSR %7,CKSWR
1247 005066 000005 RESET ;INIT
1248 005070 005777 173732 TST DRDB ;LOOKING FOR Z-BIT TO SET
1249 005074 001401 BEQ .+4 ;DID DRDB GET CLEARED?
1250 005076 104000 HLT ;DRDB NOT CLEAR
1251
1252 ;*****
1253 ; TEST 45 TEST THAT ALL DRDB BITS CAN BE SET
1254 ;*****
1255 005100 104400 SCOPE
1256 005102 012767 004000 004742 MOV #4000,ICOUNT
1257 005110 012777 177777 173710 MOV #-1,DRDB ;SET ALL BITS IN DRDB
1258 005116 022777 177777 173702 CMP #-1,DRDB ;LOOKING FOR Z-BIT TO SET
1259 005124 001401 BEQ .+4 ;SEE IF ALL BITS GOT SET
1260 005126 104000 HLT ;ALL DRDB BITS AREN'T SET
1261
1262 ;*****
1263 ; TEST 46 TEST THAT DRDB CAN HOLD ALTERNATE ONE'S AND ZERO'S
1264 ;*****
1265 005130 104400 SCOPE
1266 005132 012777 052525 173666 MOV #052525,DRDB ;ALT 0'S AND 1'S TO DRDB
1267 005140 022777 052525 173660 CMP #052525,DRDB ;LOOKING FOR Z-BIT TO SET
1268 005146 001401 BEQ .+4 ;DOES DRDB HAVE THE CORRECT PATTERN?
1269 005150 104000 HLT ;DRDB IS WRONG
1270 005152 012777 125252 173646 MOV #125252,DRDB ;ALT 1'S AND 0'S TO DRDB
1271 005160 022777 125252 173640 CMP #125252,DRDB ;LOOKING FOR Z-BIT TO SET
1272 005166 001401 BEQ .+4 ;DOES DRDB HAVE THE CORRECT PATTERN?
1273 005170 104000 HLT ;DRDB IS WRONG

```

```

1274
1275
1276
1277
1278 005172 104400
1279 005174 005067 004652
1280 005200 005001
1281 005202 005077 173620
1282 005206 020177 173614
1283 005212 001401
1284 005214 104000
1285 005216 005277 173604
1286 005222 005201
1287 005224 001370
1288
1289
1290
1291
1292
1293 005226 104400
1294 005230 004767 004640
1295 005234 000005
1296 005236 032777 000200 173560
1297 005244 001001
1298 005246 104000
1299 005250 032777 177577 173546
1300 005256 001401
1301 005260 104000
1302
1303
1304
1305
1306 005262 104400
1307 005264 012767 004000 004560
1308 005272 032777 000001 173522
1309 005300 001401
1310 005302 104000
1311
1312
1313
1314
1315
1316
1317 005304 104400
1318 005306 012767 004000 004536
1319 005314 012777 177600 173476
1320 005322 016777 173514 173472
1321 005330 105777 173470
1322 005334 100401
1323 005336 104000
1324 005340 012777 000011 173456
1325 005346 105777 173452
1326 005352 100001
1327 005354 104000
1328 005356 005067 173474
1329 005362 105777 173436

```

```

;*****
; TEST 47 INCREMENTING PATTERN TO WRAP-AROUND IN DRDB
;*****
SCOPE
CLR ICOUNT
CLR %1 ;SET-UP
CLR @DRDB ;SET-UP
INCD8: CMP %1,@DRDB ;SEE IF THEY ARE EQUAL
BEQ .+4 ;ARE THEY EQUAL?
HLT ;THEY'RE NOT EQUAL
INC @DRDB ;GET NEXT NUMBER
INC %1 ;GET NEXT NUMBER
BNE INCD8 ;DONE WITH TEST? IF NOT CONTINUE

;*****
; TEST 50 TEST THAT RESET SETS READY AND CLEARS ALL OTHER
; DRST BITS (WITH M968 INSERTED)
;*****
SCOPE
JSR %7,CKSMR
RESET ;INIT
BIT #BIT7,@DRST ;CHECK DRST
BNE .+4 ;IS READY SET?
HLT ;READY IS CLEAR
BIT #177577,@DRST ;CHECK DRST
BEQ .+4 ;ARE THEY ALL CLEAR?
HLT ;A BIT OTHER THAN READY IS SET IN THE DRST

;*****
; TEST 51 TEST THAT BA00 READS AS A ZERO WITH MAINT BOARD INSERTED
;*****
SCOPE
MOV #4000,ICOUNT
BIT #BIT0,@DRBA ;TEST BIT 0 OF DRBA
BEQ .+4 ;IS IT CLEAR?
HLT ;BA00 IS SET

;*****
; TEST 52 TEST THAT GO CLEARS READY
;*****
SCOPE
MOV #4000,ICOUNT
MOV #-200,@DRWC ;SET-UP DRWC
MOV INBUF,@DRBA ;SET-UP DRBA
TSTB @DRST ;CHECK READY
BMI .+4 ;IS READY SET?
HLT ;READY IS CLEAR
MOV #11,@DRST ;FNCT3 (NON-BURST) AND GO TO DRST
TSTB @DRST ;CHECK READY
BPL .+4 ;IS READY CLEAR?
HLT ;READY IS STILL SET
CLR RDYCHK ;CLEAR READY CHECK
TSTRDY: TSTB @DRST ;CHECK READY

```

```
1330 005366 100406 BMI DONE ; IF SET GO TO DONE
1331 005370 062767 000004 173460 ADD #4, RDYCHK ; CHECKING TIME FOR READY TO BE SET
1332 005376 100401 BMI .+4 ; IF RDYCHK GETS NEGATIVE IT TOOK TOO LONG
1333 005400 000770 BR TSTRDY ; CHECK AGAIN
1334 005402 104000 HLT ; READY GOT CLEARED BUT NEVER SET AGAIN
1335 005404 000240 NOP ; GO TO NEXT TEST
1336 ; *****
1337 ; TEST 55 TEST THAT GIVING A DO WITHOUT CLEARING A PREVIOUS ERROR
1338 ; CAUSES ANOTHER INTERRUPT
1339 ; *****
1340 005406 104400 SCOPE
1341 005410 012777 177760 173402 MOV #20, DRWC ; SET-UP DRWC
1342 005416 016777 173416 173376 MOV DIOMEM, DRBA ; SET-UP DRBA
1343 005424 012777 005460 173402 MOV #ERRDO, DRINV ; INTERRUPT VECTOR TO ERRDO
1344 005432 012777 000200 173370 MOV #200, DRVS ; INTERRUPT STATUS TO LEVEL 4
1345 005440 005077 173352 CLR #PSW ; LET THE DR11-B INTERRUPT
1346 005444 012777 000163 173352 MOV #163, DRST ; IE, XBA17, XBA16, FNCT1 AND GO TO DRST
1347 005452 005277 173402 INC #TKS ; WAIT FOR INTERRUPT
1348 005456 104000 HLT ; NO DR11-B INTERRUPT
1349 005460 005777 173340 ERRDO: TST DRST ; TEST DRST
1350 005464 100401 BMI .+4 ; ERROR SET?
1351 005466 104000 HLT ; ERROR IS CLEAR - SHOULD HAVE NEX
1352 005470 012777 005536 173336 MOV #ERRD01, DRINV ; INTERRUPT VECTOR TO ERRD01
1353 005476 005077 173320 CLR DRBA ; PREVENT CAUSING ANOTHER ERROR
1354 005502 042777 000062 173314 BIC #62, DRST ; CLEAR XBA17, XBA16, AND FNCT1
1355 005510 012777 177777 173302 MOV #1, DRWC ; SET-UP DRWC
1356 005516 005277 173302 INC DRST ; DO TO DRST
1357 005522 005067 000002 CLR #S+2
1358 005526 005227 000001 IS: INC #1
1359 005532 001375 BNE IS
1360 005534 104000 HLT ; NO DR11-B INTERRUPT
1361 005536 005777 173262 ERRD01: TST DRST ; CHECK ERROR
1362 005542 100401 BMI .+4 ; ERROR SET?
1363 005544 104000 HLT ; ERROR IS CLEAR - SHOULD BE SET BECAUSE
; PREVIOUS ERROR WAS NOT CLEARED
1364 ; REPOSITION THE STACK
1365 005546 062706 000010 ADD #10, %6
1366 005552 004767 001176 JSR %7, NORMAL
1367 ; *****
1368 ; TEST 56 STRING OF 200 DATI'S NON-BURST MODE
1369 ; *****
1370 ; *****
1371 005556 104400 SCOPE
1372 005560 012767 000200 173260 MOV #200, BUFLN ; LENGTH OF BUFFER=200
1373 005566 004767 000724 JSR %7, LODBUF ; LOAD THE BUFFER WITH INCREMENTING PATTERN
1374 005572 016767 173250 173254 MOV BUFLN, WCLN ; PREPARE NUMBER FOR DRWC
1375 005600 005467 173250 NEG WCLN ; 2'S COMPLEMENT OF BUFLN
1376 005604 016777 173244 173206 MOV WCLN, DRWC ; SET-UP DRWC
1377 005612 016777 173224 173202 MOV INBUF, DRBA ; SET-UP DRBA
1378 005620 012777 177777 173200 MOV #1, DRDB ; MAINT AIDE
1379 005626 012777 006620 173200 MOV #INTA, DRINV ; INT VECTOR TO INTA
1380 005634 016777 173172 173166 MOV DRINL, DRVS ; INT VECTOR TO PRIORITY DRINL
1381 005642 005077 173150 CLR #PSW ; LET THE DR11-B INTERRUPT
1382 005646 012777 000111 173150 MOV #111, DRST ; IE, FNCT3, AND DO TO DRST
1383 005654 005267 173172 INC BRWAIT ; USE A WAIT OR BR. INSTRUCTION
1384 005660 032767 000001 173164 BIT #BIT0, BRWAIT ; SEE WHICH ONE
1385 005666 001403 BEQ DATINB ; BIT 0 CLEAR=BR.
```

```

1386 005670 000001          WAIT          ;WAIT FOR INTERRUPT
1387 005672 000240          NOP
1388 005674 000401          BR          .+4
1389 005676 000777          DATINB: BR          .+4
1390 005700 022777 000177 173120  CMP          #177,DRDB      ;CHECK THAT WORD #200 OF INBUF IS IN DRBA
1391 005706 001401          BEQ          .+4          ;IS IT?
1392 005710 104000          HLT          ;BAD DATA IN DRDB
1393
1394
1395 ;*****
1396 ;TEST 57 STRING OF 200 DATO'S NON-BURST MODE
1397 ;*****
1397 005712 104400          SCOPE
1398 005714 012767 000201 173124  MOV          #201,BUFLEN  ;LENGTH OF BUFFER=201
1399 005722 004767 000570          JSR          X7,LOOBUF    ;LOAD THE BUFFER WITH INCREMENTING PATTERN
1400 005726 005367 173114          DEC          BUFLEN      ;BUFLEN=200
1401 005732 016767 173110 173114  MOV          BUFLEN,MLEN  ;PREPARE NUMBER FOR DRMC
1402 005740 005467 173110          NEG          MLEN        ;2'S COMPLEMENT OF BUFLEN
1403 005744 016777 173104 173046  MOV          MLEN,DRMC    ;SET UP DRMC
1404 005752 016777 173064 173042  MOV          INBUF,DRBA   ;SET UP DRBA
1405 005760 012777 052525 173040  MOV          #52525,DRDB  ;SET UP DRDB
1406 005766 012777 006620 173040  MOV          #INTA,DRINV  ;INTERRUPT VECTOR TO INTA
1407 005774 016777 173032 173026  MOV          DRINL,DRVS   ;INTERRUPT VECTOR TO PRIORITY DRINL
1408 006002 005077 173010          CLR          #PSW        ;LET THE DR11-B INTERRUPT
1409 006006 012777 000113 173010  MOV          #113,DRST    ;IE, FNCT3, FNCT1, AND DO TO DRST
1410 006014 005267 173032          INC          BRWAIT      ;USE A WAIT OR BR. INSTRUCTION
1411 006020 032767 000001 173024  BIT          #BIT0,BRWAIT ;BIT 0 CLEAR=BR.
1412 006026 001403          BEQ          DATONB
1413 006030 000001          WAIT
1414 006032 000240          NOP
1415 006034 000401          BR          .+4
1416 006036 000777          DATONB: BR          .+4
1417 006040 004767 000732          JSR          X7,DATOCK   ;CHECK INBUF
1418
1419 ;*****
1420 ;TEST 60 TEST THAT FUNCTION BITS INCREMENT WITH MAINT MODE TRANSFERS
1421 ;*****
1422 006044 104400          SCOPE
1423 006046 012767 000010 172772  MOV          #10,BUFLEN  ;SET-UP BUFLEN FOR LOOBUF AND CHKBUFF
1424 006054 016777 172762 172740  MOV          INBUF,DRBA   ;SET-UP DRBA
1425 006062 004767 000430          JSR          X7,LOOBUF    ;LOAD INBUF
1426 006066 004767 000462          JSR          X7,CHKBFF    ;LOAD CHKBUFF
1427 006072 005077 172726          CLR          DRST        ;INIT FOR STARTING
1428 006076 012767 000001 172764  MOV          #1,FNCNT     ;GET READY FOR CHECKING
1429 006104 012767 000001 172734  MOV          #1,BUFLEN    ;CHANGE IS NECESSARY FOR INTA ROUTINE
1430 006112 016767 172724 172752  MOV          INBUF,INBUF1 ;SAVE INBUF
1431 006120 012777 006620 172706  MFLoop: MOV          #INTA,DRINV ;INTERRUPT VECTOR TO INTA
1432 006126 016777 172700 172674  MOV          DRINL,DRVS   ;INTERRUPT VECTOR PRIORITY TO DRINL
1433 006134 005077 172656          CLR          #PSW        ;LET THE DR11-B INTERRUPT
1434 006140 012777 177777 172652  MOV          #-1,DRMC     ;SET-UP FOR 1 TRANSFER
1435 006146 052777 010101 172650  BIS          #10101,DRST  ;MAINT, IE, AND DO TO DRST
1436 006154 000001          WAIT
1437 006156 000240          NOP
1438 006160 117701 172640          MOVB         DRST,%1     ;FAKE-OUT RETURN ADDRESS CHANGING
1439 006164 042701 000600          BIC          #600,%1     ;LOWER BYTE OF DRST TO R1
1440 006170 006201          ASR          %1          ;GET RID OF READY AND CYCLE BECAUSE OF MAINT MODE
1441 006172 126701 172672          CMPB         FNCNT,%1   ;MOVE IT RIGHT ONE PLACE
                           ;CHECK AGAINST FNCNT

```

```

1442 006176 001401 BEQ .+4 ; SHOULD BE EQUAL
1443 006200 104000 HLT ; FUNCTION BITS DIDN'T INCREMENT IN MAINT MODE
1444 006202 005267 172662 INC FNCNT ; GET READY FOR NEXT PASS
1445 006206 022767 000010 172654 CMP #10,FNCNT ; ONLY 10 BECAUSE FNCT3-1 GO TO ZERO
1446 006214 001404 BEQ MFCHK ; IF ITS EQUAL GO CHECK DATA
1447 006216 062767 000002 172616 ADD #2,INBUF ; FAKE-OUT INTA ROUTINE
1448 006224 000735 BR MFLOOP ; DO IT AGAIN
1449 006226 012767 000007 172612 MFCHK: MOV #7,BUFLEN ; SET UP FOR DATCHK (10 FNCT CHECKS, 7 TRANSFERS)
1450 006234 016767 172632 172600 MOV INBUF1,INBUF ; RESTORE INBUF
1451 006242 004767 000446 JSR X7,DATCHK ; CHECK DATA
1452
1453 ;*****
1454 ; TEST 61 TEST FOR 10 MAINT MODE TRANSFERS
1455 ;*****
1456 006246 104400 SCOPE
1457 006250 012767 000010 172570 MOV #10,BUFLEN ; BUFLEN=10
1458 006256 016767 172564 172570 MOV BUFLEN,WLEN ; PREPARE NUMBER FOR DRWC
1459 006264 005467 172564 NEG WLEN ; 2'S COMPLEMENT OF BUFLEN
1460 006270 004767 000222 JSR X7,LOOBUF ; LOAD IN BUFFER WITH INCREMENTING PATTERN
1461 006274 004767 000254 JSR X7,CHKBFF ; LOAD CHECK BUFFER WITH MODIFIED INCREMENTING PATTEFN
1462 006300 016777 172550 172512 MOV WLEN,DRWC ; SET UP DRWC
1463 006306 016777 172530 172506 MOV INBUF,DRBA ; SET UP DRBA
1464 006314 012777 177777 172504 MOV #1,DRDB ; MAINT AIDE
1465 006322 012777 006620 172504 MOV #INTA,DRINV ; INTERRUPT VECTOR TO INTA
1466 006330 016777 172476 172472 MOV DRINL,DRVS ; INTERRUPT STATUS AT PRIORITY DRINL
1467 006336 005077 172454 CLR #PSW ; LET DR11-B INTERRUPT
1468 006342 012777 010101 172454 MOV #10101,DRST ; MAINT, IE, AND DO TO DRST
1469 006350 000777 BR ; WAIT FOR INTERRUPT
1470 006352 004767 000336 JSR X7,DATCHK
1471
1472 ;*****
1473 ; TEST 62 TEST FOR 200 NPR TRANSFERS IN MAINT MODE
1474 ;*****
1475 006356 104400 SCOPE
1476 006360 012767 000200 172460 MOV #200,BUFLEN ; LENGTH OF BUFFER = 200
1477 006366 016767 172454 172460 MOV BUFLEN,WLEN ; PREPARE NUMBER FOR DRWC
1478 006374 005467 172454 NEG WLEN ; 2'S COMPLEMENT OF BUFLEN
1479 006400 004767 000112 JSR X7,LOOBUF ; LOAD INBUF WITH INCREMENTING PATTERN
1480 006404 004767 000144 JSR X7,CHKBFF ; LOAD CHKBUFF WITH MODIFIED INCREMENTED PATTERN
1481 006410 016777 172440 172402 MOV WLEN,DRWC ; SET UP DRWC
1482 006416 016777 172420 172376 MOV INBUF,DRBA ; SET UP DRBA
1483 006424 012777 000001 172374 MOV #1,DRDB ; MAINT AIDE
1484 006432 012777 006620 172374 MOV #INTA,DRINV ; INT VECTOR TO INTA
1485 006440 016777 172366 172362 MOV DRINL,DRVS ; INT VECTOR AT PRIORITY DRINL
1486 006446 005077 172344 CLR #PSW ; LET THE DR11-B INTERRUPT
1487 006452 012777 010101 172344 MOV #010101,DRST ; FOLLOWING TO DRST: MAINT(12),IE(06),DO(00)
1488 006460 005267 172366 BRWAIT ; USE A WAIT OR BR. INSTRUCTION
1489 006464 032767 000001 172360 BIT #BIT0,BRWAIT ; SEE WHICH ONE
1490 006472 001403 BEQ BRANCH ; BIT 0 CLEAR = BR.
1491 006474 000001 WAIT
1492 006476 000240 NOP
1493 006500 000401 BR .+4
1494 006502 000777 BRANCH: BR .+4
1495 006504 004767 000204 JSR X7,DATCHK ; CHECK THAT CORRECT DATA WAS TRANSFERRED
1496 006510 104400 SCOPE
1497 006512 000167 000356 JMP END ; DO IT ALL AGAIN.

```

1498	006516	016702	172320		LODBUF:	MOV	INBUF,%2	:MOVE STARTING ADDRESS OF INBUF TO R2
1499	006522	005067	172322			CLR	LENCHK	:CLEAR LENGTH CHECK
1500	006526	005022				CLR	(%2)+	:CLEAR STARTING ADDRESS OF INBUFF AND INC BY 2
1501	006530	005267	172314		LOADA:	INC	LENCHK	:INC LENGTH CHECK BY 1
1502	006534	026767	172310	172304		CMP	LENCHK,BUFLEN	:CHECK FOR DONE
1503	006542	001403				BEQ	LDEXIT	:IS INBUF FILLED?
1504	006544	016722	172300			MOV	LENCHK,(%2)+	:LOAD NEXT BUFFER WORD
1505	006550	000767				BR	LOADA	:CONTINUE CHECKING
1506	006552	000207			LDEXIT:	RTS	%7	:EXIT
1507	006554	016702	172264		CHKBFF:	MOV	CHKBUFF,%2	:STARTING ADDRESS OF CHECK-BUFFER TO R2
1508	006560	005067	172264			CLR	LENCHK	:CLEAR LENGTH CHECK
1509	006564	005003				CLR	%3	:CLEAR R3
1510	006566	010322			CHKA:	MOV	%3,(%2)+	:MOVE R3 TO CHKBUFF ADDRESS AND INC BY 2
1511	006570	010322				MOV	%3,(%2)+	:MOVE R3 TO NEXT CHKBUFF ADDRESS AND INC BY 2
1512	006572	062767	000002	172250		ADD	#2,LENCHK	:ADD 2 TO LENGTH CHECK
1513	006600	026767	172244	172240		CMP	LENCHK,BUFLEN	:CHECK FOR DONE
1514	006606	100003				BPL	+.10	:IS CHECK-BUFFER FILLED?
1515	006610	062703	000002			ADD	#2,%3	:NEXT NUMBER FOR BUFFER
1516	006614	000764				BR	CHKA	:CONTINUE FILLING
1517	006616	000207				RTS	%7	:EXIT
1518	006620	042777	000100	172176	INTA:	BIC	#BIT6,%DRST	:CLEAR IE
1519	006626	005777	172172			TST	%DRST	:CHECKING FOR ERROR
1520	006632	100001				BPL	+.4	:ERROR SET?
1521	006634	104000				HLT		:ERROR BIT IS SET
1522	006636	105777	172162			TSTB	%DRST	:CHECKING READY BIT
1523	006642	100401				BMI	+.4	:IS READY SET
1524	006644	104000				HLT		:FALSE INTERRUPT - ERROR AND READY ARE CLEAR
1525	006646	005777	172146			TST	%DRWC	:TEST1 FOR DRWC=0
1526	006652	001401				BEQ	+.4	:WAS IT EQUAL?
1527	006654	104000				HLT		:DRWC NOT =0
1528	006656	016702	172164			MOV	BUFLEN,%2	:BUFFER LENGTH TO R2
1529	006662	066702	172160			ADD	BUFLEN,%2	:NUMBER OF TRANSFERS TIMES 2
1530	006666	066702	172150			ADD	INBUF,%2	:CORRECT DRBA
1531	006672	027702	172124			CMP	%DRBA,%2	:CHECKING DRBA
1532	006676	001401				BEQ	+.4	:IS DRBA CORRECT?
1533	006700	104000				HLT		:DRBA NOT CORRECT
1534	006702	062716	000002			ADD	#2,(%6)	:RETURN ADDRESS TO RETURN ADDRESS +2
1535	006706	004767	000042			JSR	%7,NORMAL	
1536	006712	000002				RTI		:EXIT
1537								
1538	006714	016702	172124		DATCHK:	MOV	CHKBUFF,%2	:STARTING ADDRESS OF CHECK BUFFER TO R2
1539	006720	016703	172116			MOV	INBUF,%3	:STARTING ADDRESS OF IN BUFFER TO R3
1540	006724	005067	172120			CLR	LENCHK	:CLEAR LENGTH CHECK
1541	006730	005267	172114		COMPAR:	INC	LENCHK	:MAKE A COMPARISON
1542	006734	022223				CMP	(%2)+,(%3)+	:IS THE DATA CORRECT?
1543	006736	001401				BEQ	+.4	:BRANCH IF OK
1544	006740	104000				HLT		:BAD DATA
1545	006742	026767	172102	172076		CMP	LENCHK,BUFLEN	:SEE IF THE BUFFER HAS BEEN CHECKED
1546	006750	001367				BNE	COMPAR	:BUFFER CHECKED?
1547	006752	000207				RTS	%7	
1548	006754	012777	001030	172052	NORMAL:	MOV	#DRVS,%DRINV	:RESTORE DR11-B INTERRUPT VECTOR
1549	006762	005077	172042			CLR	%DRVS	:RESTORE DR11-B INTERRUPT STATUS
1550	006766	012777	000340	172022		MOV	#340,%PSW	:RESTORE PROC TO PRIORITY LEVEL 7
1551	006774	000207				RTS	%7	:EXIT
1552	006776	012702	052525		DATOCK:	MOV	#52525,%2	:DATO NUMBER TO R2
1553	007002	016703	172034			MOV	INBUF,%3	:STARTING ADDRESS OF IN BUFFER TO R3

```

1554 007006 005067 172036          CLR      LENCHK      ;CLEAR LENGTH CHECK
1555 007012 005267 172032  COMP RR: INC      LENCHK      ;MAKE A COMPARISON
1556 007016 020223          CMP      %2,(%3)+   ;IS THE DATA CORRECT?
1557 007020 001401          BEQ      .+4         ;BRANCH IF OK
1558 007022 104000          HLT      BAD DATA
1559 007024 026767 172020 172014  CMP      LENCHK,BUFLEN ;SEE IF THE BUFFER HAS BEEN CHECKED
1560 007032 001367          BNE      COMP RR    ;BUFFER CHECKED?
1561 007034 020223          CMP      %2,(%3)+   ;CHECK END OF BUFFER + 1
1562 007036 001001          BNE      .+4         ;SEE IF TOO MANY WORDS WERE TRANSFERRED
1563 007040 104000          HLT      TOO MANY
1564 007042 000207          RTS      EXIT
1565 007044 042777 000100 171752  ERRCHK: BIC      %BIT6,%DRST ;CLEAR IE
1566 007052 005777 171746          TST      %DRST      ;CHECKING FOR ERROR
1567 007056 100001          BPL      .+4         ;ERROR SET?
1568 007060 104000          HLT      ERROR BIT IS SET
1569 007062 105777 171736          TSTB    %DRST      ;CHECKING READY BIT
1570 007066 100401          BMI      .+4         ;IS RDY SET
1571 007070 104000          HLT      FALSE ENTRY - ERROR AND READY ARE CLEAR
1572 007072 000207          RTS      %7         ;EXIT
1573
1574
1575          ;*****
1576          ;      END OF PASS
1577          ;*****
1577 007074 012737 000207 177566  END:   MOV      %207,%177566 ;RING BELL
1578 007102 105737 177564          TSTB    %177564
1579 007106 100375          BPL      -4
1580 007110 005267 171760          INC      PASCNT     ;KEEP TRACK OF PASSES COMPLETED
1581 007114 012702 012507          MOV      %SENPAS,%2 ;PRINT 'END PASS'
1582 007120 004767 003460          JSR      %7,%TOUT
1583
1584
1585 007124 042777 000020 171664  END1:  BIC      %20,%PSW    ;CLEAR T-BIT
1586 007132 013702 000042          MOV      %42,%2
1587 007136 001405          BEQ      TRTRAP
1588 007140 000005          RESET
1589 007142 004712          SENDAD: JSR      %7,(2)
1590 007144 000240          NOP
1591 007146 000240          NOP
1592 007150 000240          NOP
1593
1594          ;*****
1595          ;      ROUTINE TO CHECK FOR TRACE TRAP TO BE RUN WITH PROGRAM
1596          ;*****
1597 007152 004767 002716          TRTRAP: JSR      %7,CKSWR ;CHECK FOR CONT G
1598 007156 032777 010000 171630  BIT      %10000,%SR ;SHOULD WE RUN WITH TRACE TRAP
1599 007164 001417          BEQ      YESTR      ;YES
1600 007166 005767 000104          TST      YESTR1     ;NO, HAVE WE RUN WITH TRACE TRAP ON?
1601 007172 001411          BEQ      TRPA       ;IF SO RESTORE PREVIOUS CONTENTS
1602 007174 016767 000076 170612  MOV      YESTR1,14
1603 007202 016767 000072 170606  MOV      YESTR2,16
1604 007210 042777 000020 171600  BIC      %20,%PSW    ;CLEAR TRACE TRAP
1605 007216 000167 171746          TRPA:   JMP      BEGIN ;START OF TEST WITH TRACE OFF
1606 007222 000000          TRPB:   0
1607
1608          ;*****
1609          ;      SAVE OLD CONTENTS, SET UP FOR TRACE TRAP

```

```

1610
1611 007224 016767 170564 000044 YESTR: MOV 14,YESTR1 ;SAVE ODT PC
1612 007232 016767 170560 000040 MOV 16,YESTR2 ;SAVE ODT STATUS
1613 007240 012767 007302 170546 MOV #YESRT,14 ;NEW TRAP VECTOR
1614 007246 005067 170544 CLR 16 ;NEW CONDITION CODES
1615 007252 005077 171540 CLR @PSW
1616 007256 005167 177740 COM TRPB
1617 007262 100403 BMI .+10
1618 007264 052777 000020 171524 BIS #20,@PSW ;SET TRACE TRAP
1619 007272 000167 171672 JMP BEGIN ;START OF TEST WITH TRACE ON
1620
1621 007276 000000 YESTR1: 0 ;STORAGE FOR ODT PC
1622 007300 000000 YESTR2: 0 ;STORAG FOR ODT STATUS
1623 007302 000002 YESRT: RTI ;RETURN TO PROGRAM FROM TRAP
1624 007304 000000 HALT ;RTI FAILED
1625
1626
1627 ;*****
1628 ; BUS TO BUS TEST (DR11-B TO DR11-B)
1629 ;*****
1630
1631
1632 000000 R0=%0
1633 000001 R1=%1
1634 000002 R2=%2
1635 000003 R3=%3
1636 000004 R4=%4
1637 000005 R5=%5
1638 000006 R6=%6
1639 000007 R7=%7
1640 000007 PC=%7
1641 000006 SP=%6
1642
1643 000001 GC=1
1644 000002 FNCT1=2 ;OUTPUT MODE
1645 000004 FNCT2=4 ;OUTPUT DIRECTION
1646 000010 FNCT3=10 ;OUTPUT INTER REQ
1647 000020 XBA16=20
1648 000040 XBA17=40
1649 000100 IE=100
1650 000200 READY=200
1651 000400 CYCLE=400
1652 001000 DSTATA=1000 ;INPUT MODE
1653 002000 DSTATB=2000 ;INPUT DIRECTION
1654 004000 DSTATC=4000 ;INPUT INTR REQ
1655 010000 MAINT=10000
1656 020000 ATTN=20000
1657 040000 NEX=40000
1658 100000 ERROR=100000
1659
1660
1661
1662 007306 000004 NWRDXF: 4 ;# OF WORDS TRANSFERRED UNDER FLAG CONTROL PRIOR TO NPR
1663 007310 000000 NEXJOB: 0 ;HOLDS ADDRESS OF READY INTERRUPT ROUTINE
1664 007312 000000 ;HOLDS ADDRESS OF ERROR INTERRUPT ROUTINE
1665 007314 000000 JBFLAG: 0 ;JOB FLAG

```



```

1666 007316 000000 JBCNT: 0 ;JOB COUNT
1667 ;*****
1668 ;MASTER START
1669 ;*****
1670 007320 000240 MS1: NOP
1671 007322 005067 177770 CLR JBCNT ;CLEAR JOB COUNT
1672 007326 012706 013710 MOV #BUFF,R6 ;SETUP STACK
1673 007332 004767 000256 JSR R7,SETVEC ;SET UP INTERRUPT VECTORS
1674 007336 005067 177752 CLR JBFLAG ;CLEAR JOB FLAG
1675 007342 012767 007774 177740 MOV #JOB0,NEXJOB ;DO JOB00 FIRST
1676 007350 012767 000000 177734 MOV #0,NEXJOB+2 ;NO ERROR RECOVERY
1677 007356 012777 000340 171432 MOV #340,PSW ;LOCK OUT INTERRUPTS
1678 007364 012777 000100 171432 MOV #IE,DRST ;SET INTERRUPT ENABLE
1679 007372 005077 171420 CLR PSM ;DROP PRIOTIRY TO ZERO
1680 007376 000425 BR BACKGD ;WAIT FOR JOBS IN BACKGROUND
1681
1682
1683 ;*****
1684 ; SLAVE START
1685 ;*****
1686
1687 007400 000240 SS1: NOP
1688 007402 012706 013710 MOV #BUFF,R6
1689 007406 004767 000202 JSR R7,SETVEC ;SET UP INTERRUPT VECTORS
1690 007412 005067 177676 CLR JBFLAG
1691 007416 012767 010216 177664 MOV #SJOB1,NEXJOB ;FOR READY INTERRUPT
1692 007424 012767 010234 177660 MOV #SJOB2,NEXJOB+2 ;FOR ERROR INTERRUPT
1693 007432 012777 000340 171356 MOV #340,PSW ;RAISE CP PRIORITY TO 7
1694 007440 012777 000100 171356 MOV #IE,DRST ;SET INTERRUPT ENABLE
1695 007446 005077 171344 CLR PSM ;DROP CP PRIORITY TO 0 AND ENTER BACKGROUND
1696
1697 ;*****
1698 ; BACKGROUND PROGRAM; WAITS FOR JBFLAG TO SET
1699 ;*****
1700
1701 007452 005200 BACKGD: INC R0
1702 007454 005201 INC R1
1703 007456 005202 INC R2
1704 007460 005203 INC R3
1705 007462 005204 INC R4
1706 007464 005205 INC R5
1707 007466 020005 CMP R0,R5
1708 007470 001402 BEQ .+6
1709 007472 104000 HLT
1710 007474 000000 HALT ;BACKGROUND TEST FAILED
1711 007476 020104 CMP R1,R4
1712 007500 001402 BEQ .+6
1713 007502 104000 HLT
1714 007504 000000 HALT ;BACKGROUND TEST FAILED
1715 007506 020203 CMP R2,R3
1716 007510 001402 BEQ .+6
1717 007512 104000 HLT
1718 007514 000000 HALT ;BACKGROUND TEST FAILED
1719 007516 005767 177572 TST JBFLAG ;ANY JOBS?
1720 007522 001753 BEQ BACKGD ;BRANCH IF NONE
1721 007524 004567 000032 JSR R5,SAVALL ;YES & EXECUTE JOB WHOSE ADDRESS IS IN JBFLAG

```

```

1722 007530 005067 177554          CLR      NEXJOB
1723 007534 005067 177552          CLR      NEXJOB+2
1724 007540 016767 177550 170266    MOV      JBFLAG,34
1725 007546 005067 177542          CLR      JBFLAG
1726 007552 104400          TRAP
1727 007554 004567 000016          JSR      R5,RESALL ;TRAP THROUGH JBFLAG AT 34
1728 007560 000734          BR       BACKGD
1729
1730 ;*****
1731 ; SUBROUTINE TO PUSH ALL REGISTERS ONTO THE STACK
1732 ;*****
1733
1734 007562 010446    SAVALL: MOV      R4,-(R6) ;R5 WAS PUSHED BY JSR
1735 007564 010346    MOV      R3,-(R6)
1736 007566 010246    MOV      R2,-(R6)
1737 007570 010146    MOV      R1,-(R6)
1738 007572 010046    MOV      R0,-(R6)
1739 007574 000115    JMP      (R5) ;R5 HOLDS RETURN ADDRESS
1740
1741 ;*****
1742 ; SUBROUTINE TO POP ALL REGISTERS OFF THE STACK
1743 ;*****
1744
1745 007576 005726    RESALL: TST      (R6)+
1746 007600 012600    MOV      (R6)+,R0
1747 007602 012601    MOV      (R6)+,R1
1748 007604 012602    MOV      (R6)+,R2
1749 007606 012603    MOV      (R6)+,R3
1750 007610 012604    MOV      (R6)+,R4
1751 007612 000205    RTS      R5
1752
1753 ;*****
1754 ; ROUTINE TO SET UP INTERRUPT VECTORS
1755 ;*****
1756
1757 007614 016700 171214    SETVEC: MOV      DRINV,R0 ;R0 IS VECTOR ADDRESS
1758 007620 012720 007674    MOV      #DRINS,(R0)+ ;PUT SERVICE ADDRESS INTO VECTOR
1759 007624 016710 171202    MOV      DRINL,(R0) ;PUT PRIORITY INTO VECTOR+2
1760 007630 012767 011176 170172    MOV      #PRINT,30 ;SET UP EMT ADDRESS
1761 007636 016767 171170 170166    MOV      DRINL,32 ;SET UP EMT PRIORITY LEVEL
1762 007644 005067 170164          CLR      34
1763 007650 016767 171156 170160    MOV      DRINL,36 ;SET UP TRAP ADDRESS
1764 007656 005000          CLR      R0 ;INITIALIZE REGISTERS
1765 007660 005001          CLR      R1
1766 007662 005002          CLR      R2
1767 007664 005003          CLR      R3
1768 007666 005004          CLR      R4
1769 007670 005005          CLR      R5
1770 007672 000207          RTS      R7
1771
1772 ;*****
1773 ; PRIMARY INTERRUPT SERVICE ROUTINE.
1774 ; SETS UP JBFLAG WITH ADDRESS OF JOB TO BE RUNSTARS
1775 ;*****
1776 007674 005767 177414    DRINS: TST      JBFLAG ;HAS THE PREVIOUS INTERRUPT BEEN SERVICED?
1777 007700 001402          BEQ     DRINO

```

```

1778 007702 104000 HLT
1779 007704 000000 HALT
1780 007706 032777 004000 171110 DRINO: BIT #DSTATC,DRST ;NO
1781 007714 001411 BEQ DRIN3 ;CHECK FOR ERROR
1782 007716 005767 177370 TST NEXJOB+2 ;BRANCH IF NO ERROR
1783 007722 001002 BNE DRIN1 ;IS THERE AN ERROR SERVICE ROUTINE?
1784 007724 104000 HLT ;BRANCH IF THERE IS.
1785 007726 000000 HALT ;ERROR INTERRUPT, NO ERROR SERVICE.
1786 007730 016767 177356 177356 DRIN1: MOV NEXJOB+2,JBFLAG ;SET UP JOBFLAG WITH ADDRESS OF SERVICE ROUTINE
1787 007736 000002 RTI
1788 007740 105777 171060 DRIN3: TSTB DRST ;CHECK READY
1789 007744 100402 BMI DRIN2 ;BRANCH IF SET
1790 007746 104000 HLT
1791 007750 000000 HALT ;INTERRUPT WITHOUT ERROR OR READY
1792 007752 005767 177332 DRIN2: TST NEXJOB ;IS THERE A READY SERVICE ROUTINE
1793 007756 001002 BNE .+6 ;BRANCH IF THERE IS.
1794 007760 104000 HLT
1795 007762 000000 HALT ;READY INTERRUPT, NO READY SERVICE
1796 007764 016767 177320 177322 MOV NEXJOB,JBFLAG ;SET UP JOBFLAG WITH SERVICE ROUTINE ADDRESS
1797 007772 000002 RTI

```

```

1798
1799
1800 ;*****
1801 ; MASTER'S INTERRUPT SERVICE ROUTINES
1802 ; ROUTINE A, SEGMENT 0
1803 ; FILL BUFFER AND TRANSMIT
1804 ;*****
1804 007774 012700 011140 JOBA0: MOV #LISTA,RO ;RO IS XMIT LIST ADDRESS
1805 010000 012701 011152 MOV #LISTA,R1 ;LISTA1 WILL BE REC LIST
1806 010004 012021 MOV (RO)+,(R1)+ ;START WITH BUS ADDRESSES EQUAL
1807 010006 012002 MOV (RO)+,R2 ;R2 HOLDS WORD COUNT OF XMIT
1808 010010 010211 MOV R2,(R1) ;MAKE REC WORD COUNT THE SAME
1809 010012 005402 NEG R2 ;MAKE WORD COUNT POSITIVE
1810 010014 006302 ASL R2 ;TRANSFORM INTO BYTE COUNT
1811 010016 060241 ADD R2,-(R1) ;ADD TO REC BUS ADDRESS
1812 010020 005010 CLR (RO) ;CLEAR OFFSET IN XMIT LIST
1813 010022 024040 CMP -(RO),-(RO) ;LEAVE RO=LISTA=XMIT LIST
1814

```

```

1815 010024 022767 000100 177264 CMP #100,JBcnt ;ENOUGH PASSES FOR BELL?
1816 010032 003010 BGT JOBA0A ;BRANCH IF NOT ENOUGH
1817 010034 105737 177564 JOBA0B: TSTB @#177564 ;TTY READY?
1818 010040 100375 BPL JOBA0B
1819 010042 012737 000207 177566 MOV #207,@#177566 ;RING BELL
1820 010050 005067 177242 CLR JBCNT ;RESET JOB COUNT
1821 010054 004767 000704 JOBA0A: JSR R7,SETBUF ;FILL UP XMIT BUFFER WITH SPECIAL BINARY COUNT
1822 010060 012700 011140 MOV #LISTA,RO
1823 010064 004767 000370 JSR R7, XMIT ;TRANSMIT DATA TO SLAVE
1824 010070 012767 010106 177212 MOV #JOB1,NEXJOB ;JOB1 IS NEXT
1825 010076 012767 000000 177206 MOV #0,NEXJOB+2 ;NO ERROR RECOVERY
1826 010104 000002 RTI ;RETURN TO BACKGROUND VIA TRAP

```

```

1827
1828 ;*****
1829 ; ROUTINE A, SEGMENT 1
1830 ; FLUSH A BUFFER AND RECEIVE DATA
1831 ;*****
1832
1833 010106 012700 011152 JOBA1: MOV #LISTA1,RO ;PUT REC LIST ADDRESS INTO RO

```

```

1834 010112 004767 000574 JSR R7,FLUSH ;FLUSH BUFFER
1835 010116 012700 011152 MOV #LISTA1,R0
1836 010122 004767 000406 JSR R7,MREC ;RECEIVE DATA FROM SLAVE
1837 010126 012767 010198 MOV #JOB2,NEXJOB ;JOB2 IS NEXT
1838 010134 012767 000000 177154 MOV #0,NEXJOB+2 ;NO ERROR RECOVERY
1839 010142 000002 RTI
1840
1841 ;*****
1842 ;ROUTINE A, SEGMENT 2
1843 ;CHECKS TRANSMITTED DATA WITH RECEIVED
1844 ;*****
1845
1846 010144 012700 011140 JOBA2: MOV #LISTA,R0 ;XMIT BUFFER LIST
1847 010150 012701 011152 MOV #LISTA1,R1 ;REC BUFFER LIST
1848 010154 004767 000652 JSR R7,BUFCHK ;COMPARE THE TWO BUFFERS
1849 010160 042777 000100 170636 BIC #IE,DRST ;GLITCH INTERRUPT
1850 010166 012777 000100 170630 MOV #IE,DRST
1851 010174 012767 007774 177106 MOV #JOB0,NEXJOB ;REPEAT JOB0
1852 010202 012767 000000 177102 MOV #0,NEXJOB+2
1853 010210 005267 177102 INC JBCNT ;ADVANCE COUNT
1854 010214 000002 RTI
1855
1856 ;*****
1857 ;SLAVE'S INTERRUPT SERVICE ROUTINES
1858 ;
1859 ;JOB1: IGNORE FIRST READY INTERRUPT
1860 ;*****
1861
1862 010216 012767 000000 177064 SJOB1: MOV #0,NEXJOB ;NO MORE READY INTERRUPTS
1863 010224 012767 010234 177060 MOV #SJOB2,NEXJOB+2 ;UNTIL ATTN INTERRUPT
1864 010232 000002 RTI
1865
1866 ;*****
1867 ;JOB2: WAIT FOR COMMAND
1868 ;*****
1869
1870 010234 032777 004000 170562 SJOB2: BIT #DSTATC,DRST ;TEST FOR INTER
1871 010242 001002 BNE SJOB2A
1872 010244 104000 HLT
1873 010246 000000 HALT ;ERROR OTHER THAN DSTATC
1874 010250 005001 SJOB2A: CLR R1 ;SET UP FOR PARAMETERS
1875 010252 016702 170546 MOV DRST,R2 ;R2 IS STATUS ADDRESS
1876 010256 012703 000010 MOV #FNCT3,R3 ;R3 IS FUNCTION BIT 3
1877 010262 012704 004000 MOV #DSTATC,R4 ;R4 IS INTERRUPT BIT
1878 010266 016705 170534 MOV DRDB,R5 ;R5 IS DATA BUFFER ADDRESS
1879 010272 012700 011166 MOV #LISTB+2,R0 ;STORE PARAMETERS HERE STARTING WITH WORD COUNT
1880 010276 004767 000122 JSR R7,HNDCHK ;GET PARAMETERS
1881 010302 012700 011164 MOV #LISTB,R0 ;R0 IS TOP OF LIST
1882 010306 016010 000004 MOV 4(R0),(R0) ;MOVE OFFSET TO TOP
1883 010312 066710 170524 ADD INBUF,(R0) ;TOP OF LIST IS BUFFER START + OFFSET
1884 010316 012077 170500 MOV (R0)+,DRBA ;SET UP BUS ADDRESS
1885 010322 012077 170472 MOV (R0)+,DRWC ;SET UP WORD COUNT
1886 010326 005077 170472 CLR DRST ;CLEAR ALL FUNCTION BITS
1887 010332 032777 002000 170464 BIT #DSTATB,DRST ;WHICH DIRECTION
1888 010340 001405 BEQ SJOB2C ;BRANCH IF RECIEVE (LEAVE FNCT1 CLEAR FOR DATI'S)
1889 010342 032777 000400 170454 SJOB2B: BIT #CYCLE,DRST ;WAIT FOR MASTER TO SET CYCLE

```

```

1890 010350 001774          BEQ      SJ082B      ;BRANCH IF NOT SET
1891 010352 000412          BR       SJ082D      ;GO DO THE COMAND
1892 010354 012700 011164  SJ082C: MOV      R1,STB,R0
1893 010360 004767 000356  JSR      R7,BLUSH     ;BLUSH THE BUFFER
1894 010364 052777 000004 170432  BIS      #FNCT2,DRST   ;SET FNCT2 FOR DATO'S
1895 010372 042777 000400 170424  BIC      #CYCLE,DRST   ;CLEAR CYCLE
1896 010400 052777 000101 170416  SJ082D: BIS      #IE!GO,DRST ;EXECUTE COMMAND AND INTERRUPT WHEN DONE
1897 010406 012767 010216 176674  MOV      #SJ081,NEXJOB ;IGNORE READY INTERRUPT
1898 010414 012767 010234 176670  MOV      #SJ082,NEXJOB+2 ;WAIT FOR ATTN INTERRUPT
1899 010422 000002          RTI
1900
1901 ;*****
1902 ; SLAVE ROUTINE TO ACCEPT PARAMETERS FROM MASTER
1903 ;*****
1904
1905 010424 011515          HNSHK: MOV      (R5),(R5) ;ECHO PARAMETER
1906 010426 011520          MOV      (R5),(R0)+ ;STORE PARAMETER
1907 010430 050312          BIS      R3,(R2)     ;REPLY WITH FNCT3
1908 010432 030412          HNSH1: BIT      R4,(R2) ;WAIT FOR ATTN TO DROP
1909 010434 001376          BNE      HNSH1
1910 010436 040312          BIC      R3,(R2)     ;DROP FNCT3
1911 010440 005201          INC      R1          ;CHECK NUMBER
1912 010442 020167 176640  CMP      R1,NWRDXF
1913 010446 002401          BLT      HNSH2       ;BRANCH IF NOT DONE YET
1914 010450 000207          RTS      R7
1915 010452 030412          HNSH2: BIT      R4,(R2) ;WAIT FOR NEXT WORD
1916 010454 001776          BEQ      HNSH2       ;BRANCH IF ATTN CLEAR
1917 010456 000762          BR       HNSHK       ;GET ANOTHER PARAMETER
1918
1919 ;*****
1920 ; MASTER TRANSMIT ROUTINE
1921 ; ENTER WITH TRANSFER LIST IN R0
1922 ;*****
1923
1924 010460 005777 170340  MXMIT: TST      DRST   ;MAKE SURE ERROR IS CLEAR
1925 010464 100002          BPL      MXMIT1     ;AND READY IS SET
1926 010466 104000          HLT
1927 010470 000000          HALT              ;ERROR IS SET
1928 010472 105777 170326  MXMIT1: TSTB    DRST   ;
1929 010476 100402          BMI      MXMIT2     ;
1930 010500 104000          HLT
1931 010502 000000          HALT
1932 010504 012777 000000 170312  MXMIT2: MOV      #0,DRST ;READY NOT SET
1933 010512 004767 000102  JSR      R7,PRMXFR  ;SET UP FUNCTION FOR DATI'S
1934 010516 042777 000400 170300  BIC      #CYCLE,DRST ;TRANSFER PARAMETERS OF LIST WHOSE ADDRESS IS IN R0
1935 010524 052777 000101 170272  BIS      #IE!GO,DRST ;MAKE SURE CYCLE IS CLEAR
1936 010532 000207          RTS      R7         ;EXECUTE COMMAND AND INTERRUPT WHEN DONE
1937
1938 ;*****
1939 ; MASTER RECEIVE ROUTINE
1940 ; ENTER WITH TRANSFER LIST IN R0
1941 ;*****
1942 010534 005777 170264  MREC: TST      DRST   ;MAKE SURE ERROR IS CLEAR
1943 010540 100002          BPL      MREC1     ;AND READY SET.
1944 010542 104000          HLT
1945 010544 000000          HALT              ;ERROR SET

```

```

1946 010546 105777 170252 MREC1: TSTB @DRST
1947 010552 100402 BMI MREC2
1948 010554 104000 HLT
1949 010556 000000 HALT
1950 010560 012777 000004 170236 MREC2: MOV #FNCT2,@DRST ;READY CLEAR
1951 010566 004767 000026 JSR R7,PRMFR ;SET UP FUNCTION FOR DATO'S
1952 010572 032777 002000 170224 MREC3: BIT @DSTATB,@DRST ;TRANSFER PARAMETERS OF LIST WHOSE ADDRESS IS IN RO
1953 010600 001374 BNE MREC3 ;WAIT FOR SLAVE TO CLEAR DIRECTION
1954 010602 042777 000400 170214 BIC #CYCLE,@DRST ;BRANCH IF SET
1955 010610 052777 000101 170206 BIS #IE!GO,@DRST ;CLEAR CYCLE
1956 010616 000207 RTS R7 ;EXECUTE COMMAND AND INTERRUPT WHEN DONE.
1957
1958 ;*****
1959 ; ROUTINE TO TRANSFER AND CHECK PARAMETERS UNDER FLAG CONTROL
1960 ; ENTER WITH RO POINTING TO TRANSFER LIST
1961 ;*****
1962
1963 010620 012077 170176 PRMFR: MOV (RO)+,@DRBA ;FIRST WORD IN LIST IS ADDRESS
1964 010624 011077 170170 MOV (RO),@DRWC ;SECOND WORD IN LIST IS WORD COUNT
1965 010630 005001 CLR R1 ;R1 COUNTS PARAMETERS TRANSFERRED
1966 010632 016702 170166 MOV DRST,R2 ;R2 IS THE STATUS ADDRESS
1967 010636 012703 000010 MOV #FNCT3,R3 ;R3 USED FOR FUNCTION BIT 3
1968 010642 012704 004000 MOV @DSTAT,R4 ;R4 USED FOR INTERRUPT BIT
1969 010646 016705 170154 MOV DRDB,R5 ;R5 IS THE DATA BUFFER ADDRESS
1970 010652 011015 PRMXF1: MOV (RO),(R5) ;SET UP DRDB WITH PARAMETER
1971 010654 050312 BIS R3,(R2) ;CALL SLAVE'S ATTN
1972 010656 030412 PRMXF2: BIT R4,(R2) ;WAIT FOR REPLY
1973 010660 001776 BEQ PRMXF2 ;BRANCH IF ATTN CLEAR
1974 010662 022015 CMP (RO)+,(R5) ;COMPARE PARAMETER SENT WITH SLAVE'S ECHO
1975 010664 001402 BEQ PRMXF3 ;BRANCH IF EQUAL
1976 010666 104000 HLT
1977 010670 000000 HALT
1978 010672 040312 PRMXF3: BIC R3,(R2) ;PARAMETER DID NOT ECHO
1979 010674 030412 PRMXF4: BIT R4,(R2) ;DROP SLAVE'S ATTN
1980 010676 001376 BNE PRMXF4 ;WAIT FOR REPLY
1981 010700 005201 INC R1 ;BRANCH IF ATTN SET
1982 010702 020167 176400 CMP R1,NRDXF ;ADVANCE PARAMETER COUNT
1983 010706 002761 BLT PRMXF1 ;ALL PARAMETER XFERRED?
1984 010710 000207 RTS R7 ;BRANCH IF NOT DONE
1985 ;*****
1986 ; ROUTINE TO CLEAR BUFFER
1987 ; ENTER WITH RO POINTING TO TRANSFER LIST
1988 ;*****
1989
1990
1991 010712 005005 FLUSH: CLR R5 ;SET R5 TO ZIP
1992 010714 004767 000030 JSR R7,BSETUP ;SET UP REGISTERS
1993 010720 004767 000004 FLUSH1: JSR R7,BUFPUT ;STORE ITEM IN BUFFER
1994 010724 002775 BLT FLUSH1
1995 010726 000436 BR BUFOUT ;STOP WHEN BUFFER FULL
1996
1997 010730 010521 BUFPUT: MOV R5,(R1)+ ;PUT R5 INTO BUFFER
1998 010732 060503 ADD R5,R3 ;INCLUDE IN CHECKSUM
1999 010734 005504 ADC R4
2000 010736 005202 INC R2 ;ADVANCE WORD COUNT
2001 010740 000207 RTS R7 ;RETURN WITH STATUS SET

```

```

2002
2003
2004
2005
2006
2007 010742 012705 177777
2008 010746 000762
2009
2010
2011 010750 012001
2012 010752 012002
2013 010754 005720
2014 010756 005003
2015 010760 005004
2016 010762 000207
2017
2018
2019
2020
2021
2022
2023 010764 004767 177760
2024 010770 012705 000001
2025 010774 004767 177730
2026 011000 002011
2027 011002 005105
2028 011004 004767 177720
2029 011010 002005
2030 011012 005105
2031 011014 000241
2032 011016 006105
2033 011020 103763
2034 011022 000764
2035 011024 010320
2036 011026 010420
2037 011030 000207
2038
2039
2040
2041
2042
2043
2044
2045 011032 010046
2046 011034 010146
2047 011036 004767 000036
2048 011042 012600
2049 011044 012001
2050 011046 012002
2051 011050 012600
2052 011052 012003
2053 011054 020220
2054 011056 003402
2055 011060 104000
2056 011062 000000
2057

```

```

;*****
; ROUTINE TO FILL BUFFER WITH ALL ONE'S
; ENTER WITH RO POINTING TO TRANSFER LIST
;*****
BLUSH: MOV     #1,R5           ;SET R5 TO ALL ONE'S
        BR      FLUSH+2

;REGISTER SETUP ROUTINE
BSETUP: MOV     (RO)+,R1       ;R1 IS BUS ADDRESS
        MOV     (RO)+,R2       ;R2 IS WORD COUNT
        TST     (RO)+         ;SKIP OVER OFFSET
        CLR     R3            ;CLEAR LOW CHECKSUM
        CLR     R4            ;CLEAR HIGH CHECKSUM
        RTS     R7            ;RETURN

;*****
; ROUTINE TO FILL BUFFER WITH FLOATING 1'S AND FLOATING 0'S PATTERN
; ENTER WITH RO POINTING TO TRANSFER LIST
;*****
SETBUF: JSR     R7,BSETUP      ;SET UP REGISTERS
SETBF1: MOV     #1,R5         ;SET UP R5 WITH START PATTERN
SETBF2: JSR     R7,BUFPUT     ;NUMBER TO BUFFER
        BGE     BUFOUT
        COM     R5
        JSR     R7,BUFPUT     ;COMPLEMENT OF NUMBER TO BUFFER
        BGE     BUFOUT
        COM     R5
        CLC
        ROL     R5
        BCS     SETBF1       ;IF CARRY SET, START PATTERN OVER
        BR      SETBF2
BUFOUT: MOV     R3,(RO)+      ;MOVE LOW CHECK TO LIST
        MOV     R4,(RO)+      ;MOVE HIGH CHECK TO LIST
        RTS     R7            ;RETURN

;*****
; ROUTINE TO COMPARE TWO BUFFERS
; BUFFER LIST #1 IS IN R1
; BUFFER LIST #2 IS IN R0. #2 CHECKED FOR PROPER CHECKSUM
;*****
BUFCHK: MOV     R0,-(R6)      ;SAVE LIST#2 ON STACK
        MOV     R1,-(R6)      ;SAVE LIST#1 ON STACK
        JSR     R7,CHKSUM     ;VERIFY INTEGRITY OF CHECK BUFFER (#2)
        MOV     (R6)+,R0      ;RECOVER 1ST LIST POINTER
        MOV     (RO)+,R1      ;BUS ADDRESS #1
        MOV     (RO)+,R2      ;WORD COUNT #1
        MOV     (R6)+,R0      ;RECOVER 2ND LIST POINTER
        MOV     (RO)+,R3      ;BUS ADDRESS #2
        CMP     R2,(RO)+      ;COMPARE WORD COUNTS
        BLE     BUFCK1        ;BRANCH IF EQUAL OR LESS THAN
        HLT
        HALT
        ;BUFFER 2 IS LONGER THAN 1

```

```
2058 011064 022123      BUFCK1: CMP      (R1)+ (R3)+ ;COMPARE BUFFERS
2059 011066 001401      BEQ      BUFCK2          ;DATA ERROR
2060 011070 109000      BLT      ;ADVANCE COUNT
2061 011072 005202      BUFCK2: INC      R2      ;ADVANCE COUNT
2062 011074 002773      BLT      BUFCK1        ;BRANCH IF NOT DONE
2063 011076 000207      RTS      R7            ;RETURN; STACK IS CLEAR
2064
2065 ;*****
2066 ;ROUTINE TO CHECK SUM OF BUFFER
2067 ;ENTER WITH R0 POINTING TO TRANSFER LIST
2068 ;*****
2069 011100 004767 177644  CHKSUM: JSR      R7,BSETUP ;SET UP REGISTERS
2070 011104 004767 000016  CHKSM1: JSR      R7,GETBUF ;GET ITEM FROM BUFFER
2071 011110 002775      BLT      CHKSM1        ;BRANCH IF NOT DONE
2072 011112 020320      CMP      R3,(R0)+     ;COMPARE LOW ORDER CHECKS
2073 011114 001003      BNE      CHKSM2        ;COMPARE HIGH ORDER CHECKS
2074 011116 020420      CMP      R4,(R0)+     ;COMPARE HIGH ORDER CHECKS
2075 011120 001001      BNE      CHKSM2
2076 011122 000207      RTS      R7            ;RETURN IF CHECKSUM OK.
2077 011124 104000      CHKSM2: HLT          ;ORIGINAL BUFFER CHECKSUM DOES NOT AGREE WITH PRESENT
2078
2079 011126 012105      GETBUF: MOV      (R1)+,R5 ;GET ITEM OUT OF BUFFER
2080 011130 060503      ADD      R5,R3        ;ADD TO CHECKSUM
2081 011132 005504      ADC      R4            ;ADVANCE WORD COUNT
2082 011134 005202      INC      R2
2083 011136 000207      RTS      R7
2084
2085 011140 013712      LISTA: XINBUF        ;START OF XMIT BUFFER
2086 011142 177605      -123.                ;WORD COUNT
2087 011144 000000      0                    ;OFFSET
2088 011146 000000      0                    ;CHECKSUM LOW
2089 011150 000000      0                    ;CHECKSUM HIGH
2090
2091 011152 000000      LISTA1: 0             ;START OF REC BUFFER
2092 011154 000000      0                    ;WORD COUNT
2093 011156 000000      0                    ;OFFSET
2094 011160 000000      0                    ;CHECKSUM LOW
2095 011162 000000      0                    ;CHECKSUM HIGH
2096
2097 011164 013712      LISTB: XINBUF        ;SLAVE'S ECHO BUFFER
2098 011166 000000      0
2099 011170 000000      0
2100 011172 000000      0
2101 011174 000000      0
2102
2103 ;*****
2104 ;ENTERED WITH SYSTEM TRAP CALL(HLT)
2105 ;PRINT OUT THE ERROR PC AND STATUS REGISTER
2106 ;*****
2107
2108
2109
2110 011176 004767 000672  PRINT: JSR      %7,CKSWR
2111 011202 037727 167606 020000  BIT      JSR,#20000 ;TEST FOR INHIBIT PRINT OUT
2112 011210 001067      BNE      1$          ;IF SO, BRANCH OVER PRINT
2113 011212 012667 000204  MOV      (6)+,SAVPC ;PC OF FAILING ROUTINE
```


2114	011216	012667	000202		MOV	(6)+,SAVCC		:CC OF ERROR CONDITION
2115	011222	024646			CMP	-(6),-(6)		:REPOSITION THE STACK
2116	011224	012777	000215	167634	MOV	#215,@TPB		:CR
2117	011232	105777	167626		TSTB	@TPS		
2118	011236	100375			BPL	.-4		
2119	011240	012777	000212	167620	MOV	#212,@TPB		:LINE FEED
2120	011246	105777	167612		TSTB	@TPS		
2121	011252	100375			BPL	.-4		
2122	011254	010267	000134		MOV	x2,SAVR2		:SAVE R2
2123	011260	010367	000132		MOV	x3,SAVR3		:SAVE R3
2124	011264	010467	000130		MOV	x4,SAVR4		:SAVE R4
2125	011270	016702	000126		MOV	SAVPC,x2		
2126	011274	004767	000126		JSR	x7,PRTAB		:PRINT OCTAL NUMBER
2127	011300	012777	000240	167560	MOV	#240,@TPB		
2128	011306	105777	167552		TSTB	@TPS		:SPACE BETWEEN WORDS
2129	011312	100375			BPL	.-4		
2130	011314	016702	000104		MOV	SAVCC,x2		
2131	011320	004767	000102		JSR	x7,PRTAB		:PRINT OCTAL NUMBER
2132	011324	012777	000240	167534	MOV	#240,@TPB		:PRINT SPACE
2133	011332	105777	167526		TSTB	@TPS		:PRINTER DONE
2134	011336	100375			BPL	.-4		:BRANCH WHEN NOT DONE
2135	011340	017702	167460		MOV	@DRST,x2		:GET DR118 STATUS
2136	011344	004767	000056		JSR	x7,PRTAB		:PRINT OCTAL NUMBER
2137	011350	016702	000040		MOV	SAVR2,x2		
2138	011354	016703	000036		MOV	SAVR3,x3		
2139	011360	016704	000034		MOV	SAVR4,x4		
2140	011364	004767	000504		JSR	x7,CKSWR		
2141	011370	005777	167420	15:	TST	@SR		:CHECK SR FOR HALT SWITCH
2142	011374	100001			BPL	.-4		
2143	011376	000000			HALT			:HALT ON ERROR UP IN SWR
2144	011400	023737	000042	000046	CMP	@#42,@#46		:ARE WE IN ACT11 AUTO MODE?
2145	011406	001001			BNE	.-4		:BRANCH ON NO
2146	011410	000000			HALT			:HALT ON ERROR IF IN ACT11 AUTO MODE
2147	011412	000002			RTI			:RETURN TO MAINLINE
2148	011414	000000			SAVR2:	0		
2149	011416	000000			SAVR3:	0		
2150	011420	000000			SAVR4:	0		
2151	011422	000000			SAVPC:	0		
2152	011424	000000			SAVCC:	0		
2153								
2154	011426	005067	000260		PRTAB:	CLR	BINCT	
2155	011432	005067	000252		CLR	WGTCT		
2156	011436	012704	011716		MOV	#LIST,x4		:GET LIST ADDRESS
2157	011442	142777	000177	167414	BICB	#177,@TPS		:CLR INT FLAG
2158	011450	012767	000005	000236	MOV	#5,ASCNT		
2159	011456	012767	000007	000220	MOV	#7,SEVEN		
2160	011464	012767	000001	000214	MOV	#1,DECM1		
2161	011472	105777	167366		WAIT1:	TSTB	@TPS	
2162	011476	100375			BPL	WAIT1		
2163	011500	005702			TST	x2		
2164	011502	100404			BMI	MINUS		:NEG SIGN PRINT 1
2165	011504	012777	000260	167354	MOV	#260,@TPB		:POS SIGN PRINT 0
2166	011512	000403			BR	START		
2167	011514	012777	000261	167344	MINUS:	MOV	#261,@TPB	
2168	011522	016703	000156		START:	MOV	SEVEN,x3	:PUT MASK IN R3
2169	011526	010267	000150		MOV	x2,TOODLE		:GET READY TO DOODLE NUMBER IN TOODLE

```

2170 011532 005167 000144 COM TOODLE ;COMPENSATES FOR COMPLEMENT DURING BIC
2171 011536 046703 000140 BIC TOODLE,X3 ;AND IN OCTAL CHARACTER
2172 011542 001410 BEQ WRTOC ;ZERO, WRITE 0 IN LIST
2173 011544 066767 000136 000136 MKNUM: ADD DECML,WGTCT ;COUNT UP TO
2174 011552 005267 000134 INC BINCT ;AND RECORD
2175 011556 026703 000126 CMP WGTCT,X3 ;SAME BINARY WEIGHT
2176 011562 001370 BNE MKNUM ;KEEP COUNTN
2177 011564 062767 000260 000120 WRTOC: ADD #260,BINCT ;ADD ASCII PREFIX
2178 011572 016724 000114 MOV BINCT,(4)+ ;WRITE ASCII CHAR IN LIST
2179 011576 066767 000102 000102 ADD SEVEN,DECML ;EXPAND BINARY WEIGHT
2180 011604 005067 000100 CLR WGTCT
2181 011610 005067 000076 CLR BINCT
2182 011614 005367 000074 DEC ASCNT
2183 011620 001410 BEQ XLIST ;5 CHAR IN LIST
2184 011622 012703 000003 MOV #3,X3 ;SET X3 FOR ADD LOOP
2185 011626 066767 000052 000050 MOADD: ADD SEVEN,SEVEN ;MAKING SEVENTY BY SEVEN
2186 011634 005303 DEC X3
2187 011636 001373 BNE MOADD
2188 011640 000730 BR START ;NX SEVEN SET GET NX OCTAL
2189 011642 012767 000005 000044 XLIST: MOV #5,ASCNT ;SEND 5 CHAR TO TTY
2190 011650 105777 167210 WAIT2: TSTB #TPS
2191 011654 100375 BPL WAIT2
2192 011656 014477 167204 MOV -(4),#TPB
2193 011662 005367 000026 DEC ASCNT
2194 011666 001401 BEQ HDFHM ;FINISH PRINTING GET NXT NUM
2195 011670 000767 BR WAIT2
2196 011672 105777 167166 HDFHM: TSTB #TPS
2197 011676 100375 BPL #-4
2198 011700 000207 RTS %7 ;HEAD FOR HOME
2199 011702 000000 TOODLE: 0
2200 011704 000000 SEVEN: 0
2201 011706 000000 DECML: 0
2202 011710 000000 WGTCT: 0
2203 011712 000000 BINCT: 0
2204 011714 000000 ASCNT: 0
2205 011716 000000 LIST: 0
2206 011720 000000 0
2207 011722 000000 0
2208 011724 000000 0
2209 011726 000000 0
2210 ;*****
2211 ; SCOPE LOOP ROUTINE ENTERED BY USER TRAP
2212 ;*****
2213
2214
2215 011730 004767 000140 SCOPEA: JSR %7,CKSWR
2216 011734 032777 040000 167052 BIT #40000,#JSR
2217 011742 001003 BNE SCOPEB ;SCOPE BIT IS A ONE
2218 011744 011667 000106 MOV #%6,RETURN ;NO - SAVE PC FOR NEXT TIME
2219 011750 000002 RTI ;RETURN IN SEQUENCE
2220 011752 022606 SCOPEB: CMP (6)+,%6 ;REPOSITION THE STACK
2221 011754 012677 167036 MOV (6)+,#PSW
2222 011760 000177 000072 JMP #RETURN ;SCOPE RETURN
2223
2224 ;*****
2225 ; SCOPE OR/AND ITERATION LOOP FOR EACH TEST 4000 TIMES

```

```

2226 ;*****
2227
2228
2229 011764 004767 000104 SCOPEC: JSR %7,CKSWR
2230 011770 032777 040000 167016 BIT #4000,JSR ;TEST SR FOR SCOPE
2231 011776 001365 BNE SCOPEB ;YES SCOPE
2232 012000 005767 167070 TST PASCNT ;FIRST PASS (PASCNT=0) ?
2233 012004 001415 BEQ SCOPEG ;BR IF YES, INHIBIT ITERATIONS
2234 012006 004767 000062 JSR %7,CKSWR
2235 012012 032777 004000 166774 BIT #4000,JSR ;TEST FOR ITERATION
2236 012020 001007 BNE SCOPEB ;INHIBIT ITERATION
2237 012022 026767 000026 000022 CMP SCOPEF,ICOUNT
2238 012030 001403 BEQ SCOPEG ;EXIT - DONE
2239 012032 005267 000016 INC SCOPEF ;INCREMENT COUNT
2240 012036 000745 BR SCOPEB ;LOOP SOME MORE
2241 012040 005067 000010 SCOPEG: CLR SCOPEF ;CLEAR COUNT
2242 012044 011667 000006 MOV %2,%6,RETURN ;SAVE SCOPE RETURN POINTER
2243 012050 000002 RTI ;RETURN INLINE-NEXT TEST
2244 012052 004000 ICOUNT: 4000
2245 012054 000000 SCOPEF: 0 ;COUNT LOCATION FOR ITERATION LOOP
2246 012056 001170 RETURN: BEGIN ;ADDRESS OF LAST TEST
2247 .EVEN
2248 012060 000167 166114 JMP 200

```

```

2251 ;*****
2252 ;CHECK SWITCH REGISTER ROUTINE. CHECKS FOR ↑G TO ALLOW CHANGING
2253 ;OF LOC. 176.
2254 ;*****
2255 012064 000000 TEMPST: .WORD 0
2256 012066 000000 COUNT: .WORD 0
2257 012070 000000 RDSW: .WORD 0
2258 012072 000000 TIB: .WORD 0
2259
2260 012074 022767 000176 166712 CKSWR: CMP #SWREG,SR ;SOFTWARE SWITCH REGISTER PRESENT
2261 012102 001133 BNE OUT
2262 012104 105777 166750 TSTB %TKS ;YES, WAIT FOR
2263 012110 100130 BPL OUT ;READY, GET CHARACTER
2264 012112 017767 166744 177752 MOV %TKB,TIB ;AND STRIP OFF
2265 012120 042767 177600 177744 BIC #177600,TIB ;THE GARBAGE
2266 012126 022767 000007 177736 CMP #7,TIB ;IS IT A <↑G>
2267 012134 001116 BNE OUT
2268 012136 012702 012446 MOV #SCNTG,%2
2269 012142 004767 000436 JSR PC,TTOUT
2270 012146 012702 012460 CNTLU: MOV #MSWR,%2
2271 012152 004767 000426 JSR PC,TTOUT
2272 012156 017702 166632 MOV %SR,%2
2273 012162 004767 177240 JSR %7,PRTAB
2274 012166 012702 012470 MOV #SNEW,%2
2275 012172 004767 000406 JSR PC,TTOUT
2276 012176 005037 012064 CLR %TEMPST
2277 012202 005067 177656 SREAD: CLR TEMPST
2278 012206 012767 000007 177652 MOV #7,COUNT
2279 012214 004767 000154 IS: JSR PC,TTIN ;GO READ A CHARACTER
2280 012220 042767 177600 177644 BIC #177600,TIB ;STRIP OFF GARBAGE
2281 012226 122767 000025 177636 CMPB #25,TIB ;IS IT A ↑U?

```

```

2282 012234 001001      BNE      25      ;BRANCH IF NOT
2283 012236 000743      BR CNTLU      ;START OVER
2284 012240 122767 000015 177624 25:  CMPB     #15,TIB ;IS IT A <CR>?
2285 012246 001011      BNE      45      ;BRANCH IF NOT
2286 012250 012702 012454      MOV     #SCRLF,%2
2287 012254 004767 000324      JSR     %7,TTOUT
2288 012260 022767 000007 177600      CMP     #7,COUNT ;WAS IT FIRST CHARACTER
2289 012266 001036      BNE      75      ;CHANGE SWR IF NOT FIRST ONE
2290 012270 000440      BR      OUT      ;GET OUT
2291 012272 122767 000060 177572 45:  CMPB     #60,TIB
2292 012300 003004      BGT     55
2293 012302 122767 000067 177562      CMPB     #67,TIB
2294 012310 002005      BGE     65
2295 012312 012702 012501 55:  MOV     #SQUEST,%2
2296 012316 004767 000262      JSR     PC,TTOUT
2297 012322 000745      BR      35      ;START OVER IF NOT LEGAL CHARACTER
2298 012324 006367 177534 65:  ASL     TEMPST
2299 012330 006367 177530      ASL     TEMPST
2300 012334 006367 177524      ASL     TEMPST
2301 012340 142767 000060 177524      BICB     #60,TIB ;GET NITTY-GRITTY
2302 012346 156767 177520 177510      BISB     TIB,TEMPST
2303 012354 005367 177506      DEC     COUNT ;ONLY WANT 6 DIGITS
2304 012360 001754      BEQ     55
2305 012362 000714      BR      15
2306 012364 016777 177474 166422 75:  MOV     TEMPST,%SR ;CHANGE SWITCH REGISTER CONTENTS
2307 012372 000207      OUT:    RTS     %7 ;RETURN TO PROGRAM
2308 ;*****
2309 ; TTY READ SUBROUTINE*****
2310 ;*****
2311
2312
2313
2314 012374 005077 166460      TTIN:   CLR     @TKS
2315 012400 005077 166456      CLR     @TKB
2316 012404 005067 177462      CLR     TIB
2317 012410 005277 166444      INC     @TKS
2318 012414 105777 166440      TTIN1: TSTB   @TKS
2319 012420 100375      BPL     TTIN1
2320 012422 017767 166434 177442      MOV     @TKB,TIB
2321 012430 105777 166430      TTIN2: TSTB   @TPS
2322 012434 100375      BPL     TTIN2
2323 012436 116777 177430 166422      MOVB   TIB,@TPB
2324
2325 012444 000207      RTS     %7
2326 012446 057137 020107 000046  SCNTG:  .ASCIZ '+IG &'
2327 012454 020137 000046      SCRLF:  .ASCIZ '+ &'
2328 012460 051537 051127 020075  SMSWR:  .ASCIZ '+SWR= &'
2329 012466 000046
2330 012470 020040 042516 036527  SMNEW:  .ASCIZ ' NEW= &'
2331 012476 023040 000
2332 012501 137 020077 023137  SQUEST: .ASCIZ '+? +&'
2333 012506 000
2334 012507 137 047105 020104  SENPAS: .ASCIZ '+END PASS &'
2335 012514 040520 051523 020040
2336 012522 023040 000
2337 012525 137 046440 044501  $TITLE: .ASCIZ '+ MAINDEC-11-DZDRB-E, DR11B LOGIC TEST +&'

```

```

2338 012532 042116 041505 030455
2339 012540 026461 055104 051104
2340 012546 026502 026106 042040
2341 012554 030522 041061 046040
2342 012562 043517 041511 052040
2343 012570 051505 020124 020040
2344 012576 023137 000
2345 012602 012602
2346
2347 012602 000000
2348
2349
2350
2351
2352
2353
2354
2355 012604 105712
2356 012606 001403
2357 012610 122712 000046
2358 012614 001005
2359 012616 042777 000100 166240
2360 012624 005002
2361 012626 000207
2362 012630 122712 000137
2363 012634 001411
2364 012636 122712 000041
2365 012642 001414
2366 012644 105777 166214
2367 012650 100375
2368 012652 112277 166210
2369 012656 000752
2370 012660 005202
2371 012662 010267 000020
2372 012666 012702 012702
2373 012672 000767
2374 012674 016702 000006
2375 012700 000741
2376
2377 012702 015 012 041
2378 012706 012706
2379 012706 000000
2380 013710 013710
2381 013710 000000
2382 013712 013712
2383 014714 014714
2384 014714 014714
2385 000001

```

```

.EVEN
OFL: 0 ;FIRST CHAR FLAG

;*****
; TTY ASCII OUTPUT ROUTINE
;*****

TTOUT: TSTB (2) ;CHECK FOR NULL CHARACTER
        BEQ 15 ;IF NOT, TYPE THE CHARACTER
        CMPB #'&, (2) ;CHECK FOR TERMINATOR
        BNE .EMPTY
        BIC #100, @TPS
        CLR %2 ;CLEAR POINTER TO CHARACTER
        RTS %7 ;RETURN
        .EMPTY: CMPB #'&, (2) ;CRLF CHAR?
        BEQ .RET
        CMPB #'!', (2) ;CHECK FOR RETURN TERMINATOR
        BEQ .REST
        IS: TSTB @TPS
        BPL 15
        MOVB (2)+, @TPB ;TYPE CHARACTER
        BR TTOUT
        .RET: INC %2
        MOV %2, .SAV ;SET UP NEW POINTER
        MOV #.RETR, %2
        BR .RET-6
        .REST: MOV .SAV, %2
        BR TTOUT

        .RETR: .BYTE 15, 12, '!'
        .EVEN
        .SAV: 0
        .=. +1000
        BUFF: 0 ;FOR STACK POINTER 100 LOCATIONS
        XINBUF: .
        .=. +1000
        XCHKBU: .END

```

ASCNT	011714	DATOCK	006776	INTC	003650	PRTAB	011426	TKS	001060
ATTN =	020000	DATONB	006036	JBCNT	007316	PSW	001016	TNPRO	003554
BACKGD	007452	DECM	011706	JFLAG	007314	P4INT	003130	TNPR1	003320
BAOCK1	004742	DIONEM	001040	JOBAD	007774	P4INV	003110	TOODLE	011702
BAOFCK	004604	DONE	005404	JOBADR	010054	P5ERR	003046	TPB	001066
BEGIN	001170	DRBA	001022	JOBASB	010034	P5INV	003020	TPS	001064
BINCT	011712	DRDB	001026	JOBAB	010106	P6ERR	002756	TRPA	007216
BIT0 =	000001	DRINL	001032	JOBAB	010144	P6INV	002730	TRPB	007222
BIT1 =	000002	DRINS	007674	LDEXIT	006552	P7ERR	002666	TRTRAP	007152
BIT10 =	002000	DRINV	001034	LENCHK	001050	P7INV	002640	TSTRDY	005362
BIT11 =	004000	DRINO	007706	LIST	011716	RDSW	012070	TTIN	012374
BIT12 =	010000	DRIN1	007730	LISTA	011140	RDYCHK	001056	TTIN1	012414
BIT13 =	020000	DRIN2	007752	LISTA1	011152	READY =	000200	TTIN2	012430
BIT14 =	040000	DRIN3	007740	LISTB	011164	RESALL	007576	TTOUT	012604
BIT15 =	100000	DRST	001024	LOADA	006530	RETURN	012056	T33CLR	003546
BIT2 =	000004	DRVS	001030	LODBUF	006516	R6 =	%000006	T34CLR	003720
BIT3 =	000010	DRMC	001020	MAINT =	010000	R7 =	%000007	T35	003726
BIT4 =	000020	DSTATA=	001000	MFCHK	006226	SAVALL	007562	WAIT1	011472
BIT5 =	000040	DSTATB=	002000	MFLOOP	006120	SAVCC	011424	WAIT2	011650
BIT6 =	000100	DSTATC=	004000	MINUS	011514	SAVPC	011422	WLEN	001054
BIT7 =	000200	END	007074	MNUM	011544	SAVR2	011414	WGTCT	011710
BIT8 =	000400	END1	007124	MOADD	011626	SAVR3	011416	WTOC	011564
BIT9 =	001000	ERRA	001246	MREC	010534	SAVR4	011420	XBA16 =	000020
BLUSH	010742	ERRCHK	007044	MREC1	010546	SCOPE =	104400	XBA17 =	000040
BRANCH	006502	ERRDO	005460	MREC2	010560	SCOPEA	011730	XCHKBU	014714
BRWAIT	001052	ERRD01	005536	MREC3	010572	SCOPEB	011752	XINBUF	013712
BSETUP	010750	ERROR =	100000	MSTART	001000	SCOPEC	011764	XLIST	011642
BUFCHK	011032	FLUSH	010712	MS1	007320	SCOPEF	012054	YESRT	007302
BUFCK1	011064	FLUSH1	010720	MXMIT	010460	SCOPEG	012040	YESTR	007224
BUFCK2	011072	FNCNT	001070	MXMIT1	010472	SETBF1	010770	YESTR1	007276
BUFF	013710	FNCT1 =	000002	MXMIT2	010504	SETBF2	010774	YESTR2	007300
BUFLN	001046	FNCT2 =	000004	NEX =	040000	SETBUF	010764	SCNTG	012446
BUFOU	011024	FNCT3 =	000010	NEXCHK	004450	SETVEC	007614	SCRLF	012454
BUFPUT	010730	GETBUF	011126	NEXJOB	007310	SEVEN	011704	SENDAD	007142
BUSERR=	000004	GO =	000001	NOP =	000240	SJOB1	010216	SENPAS	012507
CHKA	006566	HDFHM	011672	NORMAL	006754	SJOB2	010234	SHD =	000003
CHKBFF	006554	HLT =	104000	NPRDY	003404	SJOB2A	010250	SMNEW	012470
CHKBUF	001044	HNSHK	010424	NPR1	001036	SJOB2B	010342	SMSR	012460
CHKSM1	011104	HNSH1	010432	NWRDXF	007306	SJOB2C	010354	SQUEST	012501
CHKSM2	011124	HNSH2	010452	N1413	003370	SJOB2D	010400	SREAD	012202
CHKSUM	011100	ICOUNT	012052	OFL	012602	SR	001014	SSWR =	160000
CKSMR	012074	IE =	000100	OUT	012372	SSTART	001006	STITLE	012525
CNTLU	012146	INBUF	001042	PASCNT	001074	SS1	007400	STN =	000001
COMPAR	006730	INBUF1	001072	PRINT	011176	START	011522	.EMPTY	012630
COMPR	007012	INCBA	002546	PRMXFR	010620	SUSWR	001076	.REST	012674
COUNT	012066	INCDB	005206	PRMXF1	010652	SWREG	000176	.RET	012660
CYCLE =	000400	INCWC	002516	PRMXF2	010656	TEMPST	012064	.RETR	012702
DATCHK	006714	INTA	006620	PRMXF3	010672	TIB	012072	.SAV	012706
DATING	005676	INTB	003476	PRMXF4	010674	TKB	001062		

. ABS. 014716 000

ERRORS DETECTED: 0
DEFAULT GLOBALS GENERATED: 0

H04

MAINDEC-11-DZDRBE-E MACY11 27(1006) 26-MAY-77 15:39 PAGE 48
DZDRBE.P11 26-MAY-77 15:37 SYMBOL TABLE

DZDRBE,DSKZ:DZDRBE/SOL=DSKZ:SYSMAC.SML,DSKM:DZDRBE.P11
RUN-TIME: 9 11 .2 SECONDS
RUN-TIME RATIO: 420/21=19.8
CORE USED: 31K (62 PAGES)