

DR11B

GENERAL NPR INTERFACE
MD-11-DZDRB-F
DA11B-PDP11 INTERPROC LINK

EP-DZDRB-F-DL
COPYRIGHT © 71-78
FICHE 1 OF 1

JUN 1978
digital
MADE IN USA



.REM %

IDENTIFICATION

PRODUCT CODE: MAINDEC-11-DZDRB-F-D
 PRODUCT NAME: DA11B - PDP11 GENERAL MPA INTERFACE
 DA11B - PDP11 INTERPROCESSER INK

DATE RELEASED: MAY 1978
 MAINTAINER: DIAGNOSTIC ENGINEERING
 AUTHOR: LARRY CONDON, DICK JONES, STEVE POMFRET

REVISED BY: W.F. KELLICKER 29-JAN-1974
 ALAN BOSTICK 20-JUN-1976
 BARRY SUSSMAN 01-SEP-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.

NO RESPONSIBILITY IS ASSUED FOR THE USE OR RELIABILITY OF SOFTWARE ON EQUIPMENT THAT IS NOT SUPPLIED BY DIGITAL OR ITS AFFILIATED COMPANIES.

COPYRIGHT (C) 1971, 1978 BY DIGITAL EQUIPMENT CORPORATION

THE FOLLOWING ARE TRADEMARKS OF DIGITAL EQUIPMENT CORPORATION:

DIGITAL PDP UNIBUS MASSBUS

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 3 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.
NOTE: IF ALL SWITCHES ARE DOWN, IT IS ASSUMED THAT THE
BR LEVEL OF THE DR11B IS = 5. SEE OPERATIONAL SWITCH
SETTINGS, SECTION 5.1.2.

001

MD-11-DZDRB-F, DR11B/DA11B TEST MACY11 30(1046) 25-OCT-77 12:36 PAGE 3
DZDRBF.P11 25-OCT-77 12:35

SEQ 0003

(A) 200 = TEST OF LOGIC USING MAINTENANCE FEATURE M968 IN C04,004

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:

SWR= 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
OL* 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
SW02 TO SW00 = 4 ... BR LEVEL OF DR11B = 4
 = 5 ... BR LEVEL OF DR11B = 5
 = 6 ... BR LEVEL OF DR11B = 6

NOTE: IF SW02 TO SW00 = 0, THE BR LEVEL OF
THE DR11B IS ASSUMED TO BE = 5.

5.1.3

IF THE DIAGNOSTIC IS RUN ON A CPU WITHOUT A SWITCH
REGISTER THEN A SOFTWARE SWITCH REGISTER IS USED WHICH AL S
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: SWR=XXXXXXNEW= (XXXXXX IS THE OCTAL CONTENTS
OF THE SOFTWARE SWITCH REGISTER.)

F01

MO-11-DZDRB-F, DR11B/DA11B TEST MACY11 30(1046) 25-OCT-77 12:36 PAGE 5
DZDRBF.P11 25-OCT-77 12:35

SEQ 0005

3) AFTER THE 'NEW=' HAS BEEN TYPED THEN THE OPERATOR CAN DO ONE
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

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

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 (000000). 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

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
 ^BA16 SET & CLEARED
 XBA17 SET & CLEARED
 IE 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/L04
 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
 BAD0 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)
 FUNC? 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 (OR CNTRL AND CONT ON AN 11/34 OR 11/04), (ON SWITCHLESS PROCESSOR TOGGLE THE HALT/CONT SWITCH).
- 12.5 MASTER COMPUTER, PRESS CONTINUE (OR CNTRL AND CONT ON AN 11/34 OR 11/04), (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 DA11B MUST BE IN C04/D04 .

14. DA11B PROGRAM DISCRIPTION

- 14.1 THE SLAVE COMPUTER STARTS BY ENTERING A BACKGROUND TO WAIT FOR AN INTERUPT WITH THE INTERUPT ENABLED. THE FIRST INTERUPT THAT COMES SHOULD BE THE READY INTERUPT SET UP WHEN THE MASTER HIT THE START KEY. THE INTERUPT CAUSES THE SLAVE TO ENTER THE INTERUPT SERVICE ROUTINE.
- 14.2 THE INTERUPT SERVICE ROUTINE DETERMINS WHAT INTERUPT CAME UP AND IF IT SHOULD HAVE COME UP. IF THE INTERUPT 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 INTERUPT TO THE SLAVE. THIS IS A SIGNAL FOR THE SLAVE TO AC/EPT 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 DA11B 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 SINC WHEN AN

LO1

MO-11-DZDRB-F, DR11B/DA11B TEST MACY11 30(1046) 25-OCT-77 12:36 PAGE 11
DZDRBF.P11 25-OCT-77 12:35

SEQ 0011

```

448
449
450
451
452
453
454
455
456
457
458
459
460      000001
461      160000
462
463
464
465
466
467
468
469
470
471      000240
472      104400
473      104000
474      000001
475      000002
476      000004
477      000010
478      000020
479      000040
480      000100
481      000200
482      000400
483      001000
484      002000
485      004000
486      010000
487      020000
488      040000
489      100000
490
491      000004
492
493
494
495      000000
496      000030
497      000030 011366
498      000032 000340
499      000034 000034
500      000034 012154
501      000036 000340
502      000046 000046

```

```

*****
*****
TITLE MAINDEC-11-DZDRB-F
*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-DZQAC-C3), JAN 19, 1977.
*
$TN=1
$SWR=160000 ;:HALT ON ERROR, LOOP ON TEST, INHIBIT ERROR TYP0UT
*****
MODIFIED FOR SOFTWARE SWITCH REGISTER
INCLUDING DYNAMIC LOADING OF SWR

MODIFIED TO ELIMINATE BAOF ERROR WHEN
DMA TRANSFERS CROSS A 32K BOUNDARY
*****
*****
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

```

```

504          000052      000052          .=52
505 C00052      000000          0
506          000176      000176          .=176
507          000000      000000      SWREG: 0
508          000200      000200          .=200
509          000200      012706      014100      MOV      #BUFF,%6      ;SET UP STACK LIMIT
510          000204      005077      000606      CLR      @PSW
511          000210      023737      000042      000046      CMP      @#42,@#46      ;ARE WE IN ACT11 AUTO MODE?
512          000216      001404          BEQ      1$      ;SKIP TITLE IF YES
513          000220      012702      012715      MOV      @TITLE,%2      ;PRINT THE TITLE
514
515          000224      004767      012544          JSR
516          00023C      000167      000646      1$: JMP      SUSWR
517          001000          .=1000
518
519
520
521          ;*****
522          ; START OF BACK-TO-BACK DR11-B
523          ;*****
524
525          001000      000000      MSTART: HALT          ;MASTER START
526          001002      000167      006502      JMP      MS1
527          001006      000000      SSTART: HALT          ;SLAVE START
528          001010      000167      006554      JMP      SS1
529          001014      177570      SR: 177570
530          001016      177776      PSW: 177776
531          001020      172410      DRWC: 172410
532          001022      172412      DRBA: 172412
533          001024      172414      DRST: 172414
534          001026      172416      DRDB: 172416
535          001030      00012E      DRVS: 126
536          001032      000240      DRINL: 240
537          001034      000124      DRINV: 124
538          001036      052525      NPR1: 52525
539          001040      173000      DIOMEM: 173000
540          001042      014102      INBUF: XINBUF
541          001044      015104      CHKBUF: XCHKBU
542          001046      000000      BUFLN: HALT
543          001050      000000      LENCHK: HALT
544          001052      000000      BRWAIT: HALT
545          001054      000000      WCLN: HALT
546          001056      000000      ROYCHK: HALT
547          001060      177560      TKS: 177560
548          001062      177562      TKB: 177562
549          001064      177564      TPS: 177564
550          001066      177566      TPB: 177566
551          001070      000000      FNCCNT: HALT
552          001072      000000      INBUF1: HALT
553          001074      000000      PASCNT: J
554          001076      000000      TEMP: 0          ;NUMBER OF PASSES COMPLETED
555          001100      000000      TEMP2: 0          ;TEMPORARY STORAGE
556          ;
557          C01102      013746      000006      SUSWR: MOV      @#6,-(SP)      ;SAVE VECTORS
558          001106      013746      000004      MOV      @#4,-(SP)

```



```

560 001120 022777 177777 177666      CMP      #1,JSR      ;REFERENCE HARDWARE SWITCH REGISTER
561 001126 001402      BEQ      658:      ;
562 001130 000404      BR      668:      ;
563 001136 022626      648:    CMP      (SP)+,(SP)+ ;ADJUST STACK
564 001137 012767 000176 177652 658:    MOV      #SWREG,SR ;POINT TO SOFTWARE SWITCH REG
565 001142 012637 000004      668:    MOV      (SP)+,284 ;RESTORE VECTORS
566 001146 012637 000006      MOV      (SP)+,286 ;
567 001152 022767 000176 177634      CMP      #SWREG,SR ;IS SWREG USED
568 001160 001005      BNE      BEGIN    ;
569 001162 005737 000042      TST      284      ;ARE WE IN AUTO MODE?
570 001166 001002      BNE      BEGIN    ;IF SO, SKIP SWREG INPUT
571 001170 004767 011142      JSR      PC,CNTLU ;ALLOW SWREG TO BE LOADED
572 001174 012777 000340 177614 BEGIN:  MOV      #340,PSW ;PROC. AT LEVEL #7
573 001202 012767 001174 011036      MOV      #BEGIN,RETURN

```

```

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

```

```

581 001210 104400      SCOPE
582 001212 012767 001252 176564      MOV      #ERRA,BUSERR ;BUS ERROR VECTOR TO ERRA
583 001220 010700      MOV      %7,%0      ;PC TO R0
584 001222 005277 177572      INC      20A0C      ;ADDRESS DRWC
585 001226 010700      MOV      %7,%0      ;PC TO R0
586 001230 005277 177566      INC      20A0A      ;ADDRESS DRBA
587 001234 010700      MOV      %7,%0      ;PC TO R0
588 001236 005077 177562      CLR      20A08      ;ADDRESS DRST
589 001242 010700      MOV      %7,%0      ;PC TO R0
590 001244 005277 177556      INC      20A06      ;ADDRESS DR0B
591 001250 000401      BR      .+4        ;MADE IT - BRANCH OVER HALT
592 001252 104000      HLT
593 001254 012767 000006 176522  ERRA:  MOV      #6,BUSERR ;BUS ERROR, R0 HAS PC OF ERROR
594 001262 104400      SCOPE ;RESTORE #6 TO BUS ERROR VECTOR

```

```

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

```

```

601 001264 012767 000010 010750      MOV      #10,ICOUNT
602 001272 012777 177777 177520      MOV      #-1,DRWC ;ALL ONES TO DRWC
603 001300 004767 010760      JSR      %7,CKSWR
604 001304 000005      RESET
605 001306 005777 177506      TST      20A0C      ;INIT
606 001312 001401      BEQ      .+4        ;LOOKING FOR Z-BIT TO SET
607 001314 104000      HLT ;DID DRWC GET CLEARED?
608 001316 104400      SCOPE ;DRWC NOT CLEAR

```

```

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

```

```

610
611
612
613
614

```

```

616
617
618 001320 104400
619 001322 012777 177777 177472
620 001330 004767 010730
621 001334 000005
622 001336 005777 177460
623 001342 001401
624 001344 104000
625
626
627
628
629
630
631
632 001346 104400
633 001350 012767 004000 010664
634 001356 012777 177777 177434
635 001364 022777 177777 177426
636 001372 001401
637 001374 104000
638
639
640
641
642
643
644 001376 104400
645 001400 012777 177776 177414
646 001406 022777 177776 177406
647 001414 001401
648 001416 104000
649
650
651
652
653
654
655
656
657 001420 104400
658 001422 052777 000002 177374
659 001430 032777 000002 177366
660 001436 001001
661 001440 104000
662 001442 042777 000002 177354
663 001450 032777 000002 177346
664 001456 001401
665 001460 104000
666
667
668
669
670

```

```

SCOPE
MOV #-1,DRBA ;ALL ONES TO DRBA
JSR %7,CKSWR
RESET ;INIT
TST DRBA ;LOOKING FOR Z-BIT TO SET
BEQ .+4 ;DID DRBA GET CLEARED?
HLT ;DRBA NOT CLEAR

:*****
: TEST3 CAN ALL DRWC BITS BE SET?
:*****

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

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

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

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

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

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

```

```

672
673
674 001462 104400          SCOPE
675 001464 052777 000004 177332  BIS      #BIT2,DRST  :SET FNCT2
676 001472 032777 000004 177324  BIT      #BIT2,DRST  :TEST FNCT2
677 001500 001001          BNE      .+4      :IS IT SET?
678 001502 104000          HLT                      :FNCT2 IS CLEAR
679 001504 042777 000004 177312  BIC      #BIT2,DRST  :CLEAR FNCT2
680 001512 032777 000004 177304  BIT      #BIT2,DRST  :TEST FNCT2
681 001520 001401          BEQ      .+4      :WAS IT CLEAR?
682 001522 104000          HLT                      :FNCT2 WAS SET
683
684
685

```

```

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

```

```

686
687
688
689
690
691 001524 104400          SCOPE
692 001526 052777 000010 177270  BIS      #BIT3,DRST  :SET FNCT3
693 001534 032777 000010 177252  BIT      #BIT3,DRST  :TEST FNCT3
694 001542 001001          BNE      .+4      :IS IT SET?
695 001544 104000          HLT                      :FNCT3 IS CLEAR
696 001546 042777 000010 177250  BIC      #BIT3,DRST  :CLEAR FNCT3
697 001554 032777 000010 177242  BIT      #BIT3,DRST  :TEST FNCT3
698 001562 001401          BEQ      .+4      :WAS IT CLEAR?
699 001564 104000          HLT                      :FNCT3 WAS SET
700
701
702

```

```

: *****
: TES 11 TEST THAT XBA16 CAN BE SET AND CLEARED
: *****

```

```

703
704 001566 104400          SCOPE
705 001570 052777 000020 177226  BIS      #BIT4,DRST  :SET XBA16
706 001576 032777 000020 177220  BIT      #BIT4,DRST  :TEST XBA16
707 001604 001001          BNE      .+4      :IS IT SET?
708 001606 104000          HLT                      :XBA16 IS CLEAR
709 001610 042777 000020 177206  BIC      #BIT4,DRST  :CLEAR XBA16
710 001616 032777 000020 177200  BIT      #BIT4,DRST  :TEST XBA16
711 001624 001401          BEQ      .+4      :IS IT CLEAR
712 001626 104000          HLT                      :XBA16 WAS SET
713
714
715

```

```

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

```

```

716
717 001630 104400          SCOPE
718 001632 052777 000040 177164  BIS      #BIT5,DRST  :SET XBA17
719 001640 032777 000040 177156  BIT      #BIT5,DRST  :TEST XBA17
720 001646 001001          BNE      .+4      :IS IT SET?
721 001650 104000          HLT                      :XBA17 IS CLEAR
722 001652 042777 000070 177144  BIC      #BIT4,DRST  :CLEAR XBA17
723 001660 032777 000020 177136  BIT      #BIT4,DRST  :TEST XBA17
724 001666 001401          BEQ      .+4      :IS IT CLEAR?
725 001670 104000          HLT                      :XBA17 WAS SET
726

```



```

728
729
730 001672 104400
731 001674 052777 000100 177122
732 001702 032777 000100 177114
733 001710 001001
734 001712 104000
735 001714 042777 000100 177102
736 001722 032777 000100 177074
737 001730 001401
738 001732 104000
739
740
741
742
743 001734 104400
744 001736 052777 000400 177060
745 001744 032777 000400 177052
746 001752 001001
747 001754 104000
748 001756 042777 000400 177040
749 001764 032777 000400 177032
750 001772 001401
751 001774 104000
752
753
754
755
756 001776 104400
757 002000 052777 010000 177016
758 002006 032777 010000 177010
759 002014 001001
760 002016 104000
761 002020 042777 010000 176776
762 002026 032777 010000 176770
763 002034 001401
764 002036 104000
765
766
767
768
769 002040 104400
770 002042 052777 010576 176754
771
772 002050 032777 000002 176746
773 002056 001001
774 002060 104000
775 002062 032777 000004 176734
776 002070 001001
777 002072 104000
778 002074 032777 000010 176722
779 002102 001001
780 002104 104000
781 002106 032777 000020 176710
782 002114 001001

```

```

: TEST13 TEST THAT IE CAN BE SET AND CLEARED
: *****
: SCOPE
: BIS #BIT6,DRST ;SET IE
: BIT #BIT6,DRST ;TEST IE
: BNE .+4 ;IS IT SET?
: HLT ;IE IS CLEAR
: BIC #BIT6,DRST ;CLEAR IE
: BIT #BIT6,DRST ;TEST IE
: BEQ .+4 ;IS IT CLEAR?
: HLT ;IE WAS SET

: *****
: TEST14 TEST THAT CYCLE CAN BE SET AND CLEARED
: *****
: SCOPE
: BIS #BIT8,DRST ;SET CYCLE
: BIT #BIT8,DRST ;TEST CYCLE
: BNE .+4 ;IS IT SET?
: HLT ;CYCLE WAS CLEAR
: BIC #BIT8,DRST ;CLEAR CYCLE
: BIT #BIT8,DRST ;TEST CYCLE
: BEQ .+4 ;IS IT CLEAR?
: HLT ;CYCLE WAS SET

: *****
: TEST15 TEST THAT MAINT CAN BE SET AND CLEARED
: *****
: SCOPE
: BIS #BIT12,DRST ;SET MAINT
: BIT #BIT12,DRST ;TEST MAINT
: BNE .+4 ;IS IT SET?
: HLT ;MAINT WAS CLEAR
: BIC #BIT12,DRST ;CLEAR MAINT
: BIT #BIT12,DRST ;TEST MAINT
: BEQ .+4 ;IS MAINT CLEAR?
: HLT ;MAINT WAS SET

: *****
: TEST 16 TEST THAT ALL DRST R/W BITS CAN BE SET AND CLEARED
: *****
: SCOPE
: BIS #10576,DRST ;SET FOLLOWING: MAINT(12), CYCLE(08), IE(06), XBA17(05),
: ; XBA16(04),FNCT3(03),FUNCT2(02),FNCT1(01)
: BIT #BIT1,DRST ;TEST FNCT1
: BNE .+4 ;IS IT SET?
: HLT ;FNCT1 IS CLEAR
: BIT #BIT2,DRST ;TEST FNCT2
: BNE .+4 ;IS IT SET?
: HLT ;FNCT2 IS CLEAR
: BIT #BIT3,DRST ;TEST FNCT3
: BNE .+4 ;IS IT SET?
: HLT ;FNCT3 IS CLEAR
: BIT #BIT4,DRST ;TEST XBA16
: BNE .+4 ;IS IT SET?

```

```

784 002120 032777 000040 176676 BIT #BITS,DRST ;TEST XBA17
785 002126 001001 BNE .+4 ;IS IT SET?
786 002130 104000 HLT ;XBA17 IS CLEAR
787 002132 032777 000100 176664 BIT #BIT6,DRST ;TEST IE
788 002140 001001 BNE .+4 ;IS IT SET?
789 002142 104000 HLT ;IE IS CLEAR
790 002144 032777 000400 176652 BIT #BIT8,DRST ;TEST CYCLE
791 002152 001001 BNE .+4 ;IS CYCLE SET?
792 002154 104000 HLT ;CYCLE IS CLEAR
793 002156 032777 010000 176640 BIT #BIT12,DRST ;TEST MAINT
794 002164 001001 BNE .+4 ;IS MAINT SET?
795 002166 104000 HLT ;MAINT IS CLEAR
796 002170 042777 010576 176626 BIC #10576,DRST ;CLEAR ALL R/W BITS IN DRST
797 002176 032777 000002 176620 BIT #BIT1,DRST ;TEST FNCT1
798 002204 001401 BEQ .+4 ;IS FNCT1 CLEAR?
799 002206 104000 HLT ;FNCT1 IS SET
800 002210 032777 000004 176606 BIT #BIT2,DRST ;TEST FNCT2
801 002216 001401 BEQ .+4 ;IS FNCT2 CLEAR?
802 002220 104000 HLT ;FNCT2 IS SET
803 002222 032777 000010 176574 BIT #BIT3,DRST ;TEST FNCT3
804 002230 001401 BEQ .+4 ;IS FNCT3 CLEAR?
805 002232 104000 HLT ;FNCT3 IS SET
806 002234 032777 000020 176562 BIT #BIT4,DRST ;TEST XBA16
807 002242 001401 BEQ .+4 ;IS XBA16 CLEAR?
808 002244 104000 HLT ;XBA16 IS SET
809 002246 032777 000040 176550 BIT #BITS,DRST ;TEST XBA17
810 002254 001401 BEQ .+4 ;IS XBA17 CLEAR?
811 002256 104000 HLT ;XBA17 IS SET
812
813 002260 032777 000100 176536 BIT #BIT6,DRST ;TEST IE
814 002266 001401 BEQ .+4 ;IS IE CLEAR?
815 002270 104000 HLT ;IE IS SET
816 002272 032777 000400 176524 BIT #BIT8,DRST ;TEST CYCLE
817 002300 001401 BEQ .+4 ;IS CYCLE CLEAR?
818 002302 104000 HLT ;CYCLE IS SET
819 002304 032777 010000 176512 BIT #BIT12,DRST ;TEST MAINT
820 002312 001401 BEQ .+4 ;IS MAINT CLEAR?
821 002314 104000 HLT ;MAINT IS SET
822
823 ;*****
824 ;TEST17 ALL R/W BITS IN DRST CAN BE SET AND RESET TO ZERO, THAT READY
825 ;IS SET, NEX IS CLEAR, AND GO IS READ AS A 0.
826 ;*****
827 002316 104400 SCOPE
828 002320 012767 000010 007714 MOV #10,ICOUNT
829 002326 052777 010576 176470 BIS #10576,DRST ;SET FOLLOWING: MAINT(12),CYCLE(08),IE(06),XBA17(05),
830 ; XBA16(04),FNCT3(03),FNCT2(02),FNCT1(01)
831 002334 017701 176464 MOV DRST,%1 ;MOVE (DRST) TO R1
832 002340 052701 167201 BIS #167201,%1 ;SETS BITS IN R1 THAT WERE NOT SET IN DRST
833 002344 005201 INC %1 ;R1 SHOULD GO FROM -1 TO ZERO
834 002346 001401 BEQ .+4 ;WERE ALL DRST R/W BITS SET?
835 002350 104000 HLT ;NOT ALL BITS WERE SET
836 002352 004767 007706 JSR %7,CKSWR
837 002356 000005 RESET ;CLEAR ALL DRST R/W BITS
838 002360 017701 176440 MOV DRST,%1 ;MOVE (DRST) TO R1

```

```

840 002370 001401      BEQ      .+4      :SHOULD EQUAL ZERO
841 002372 104000      HLT      :RESET DIDN'T LEAVE DRST AS IT SHOULD HAVE
842
843
844
845
846 002374 104400      SCOPE
847 002376 012767 004000 007636  MOV      #4000,ICOUNT
848 002404 012777 752525 176406  MOV      #052525,@DRWC :ALT 0'S AND 1'S TO DRWC
849 002412 022777 052525 176400  CMP      #052525,@DRWC :LOOKING FOR Z-BIT TO SET
850 002420 001401      BEQ      .+4      :DOES DRWC HAVE THE CORRECT PATTERN?
851 002422 104000      HLT      :DRWC DOESN'T HAVE THE CORRECT PATTERN
852 002424 012777 125252 176366  MOV      #125252,@DRWC :ALT 1'S AND 0'S TO DRWC
853 002432 022777 125252 176360  CMP      #125252,@DRWC :LOOKING FOR Z-BIT TO SET
854 002440 001401      BEQ      .+4      :DOES DRWC HAVE THE CORRECT PATTERN?
855 002442 104000      HLT      :DRWC DOESN'T HAVE THE CORRECT PATTERN
856
857
858
859
860 002444 104400      SCOPE
861 002446 012777 052524 176346  MOV      #052524,@DRBA :ALT 0'S AND 1'S TO DRBA
862 002454 022777 052524 176340  CMP      #052524,@DRBA :LOOKING FOR Z-BIT TO SET
863 002462 001401      BEQ      .+4      :DOES DRBA HAVE THE CORRECT PATTERN?
864 002464 104000      HLT      :DRBA DOESN'T HAVE THE CORRECT PATTERN
865 002466 012777 125252 176326  MOV      #125252,@DRBA :ALT 1'S AND 0'S TO DRBA
866 002474 022777 125252 176320  CMP      #125252,@DRBA :LOOKING FOR Z-BIT TO SET
867 002502 001401      BEQ      .+4      :DOES DRBA HAVE THE CORRECT PATTERN?
868 002504 104000      HLT      :DRBA DOESN'T HAVE THE CORRECT PATTERN
869
870
871
872
873 002506 104400      SCOPE
874 002510 005067 007576  CLR      ICOUNT
875 002514 005001      CLR      %1      :SET-UP
876 002516 005077 176276  CLR      @DRWC   :SET-UP
877 002522 020177 176272  INCW:  CMP      %1,@DRWC :SEE IF THEY ARE EQUAL
878 002526 001401      BEQ      .+4      :ARE THEY EQUAL?
879 002530 104000      HLT      :THEY'RE NOT EQUAL
880 002532 005277 176262  INC      @DRWC   :GET NEXT NUMBER
881 002536 005201      INC      %1      :GET NEXT NUMBER
882 002540 001370      BNE      INCW    :DONE WITH TEST? IF NOT CONTINUE
883
884
885
886
887 002542 104400      SCOPE
888 002544 005001      CLR      %1      :SET-UP
889 002546 005077 176250  CLR      @DRBA   :SET-UP
890 002552 020177 176244  INCBA:  CMP      %1,@DRBA :SEE IF THEY ARE EQUAL
891 002556 001401      BEQ      .+4      :ARE THEY EQUAL?
892 002560 104000      HLT      :THEY'RE NOT EQUAL
893 002562 062777 000002 176232  ADD      #2,@DRBA :GET NEXT NUMBER
894 002570 062701 000002      ADD      #2,%1   :GET NEXT NUMBER

```

```

896
897
898
899
900
901 002576 104400
902 012600 012767 004000 007434
903 002606 012777 000340 176202
904 002614 032777 000200 176202
905 002622 001010
906 002624 004767 007434
907 002630 000005
908 002632 032777 000200 176164
909 002640 001001
910 002642 104000
911 002644 012777 002700 176162 P7INV:
912 002652 012737 000340 000126
913 002660 052777 000100 176136
914 002666 000240
915 002670 042777 000100 176126
916 002676 000405
917 002700 022626 P7ERR:
918 002702 042777 000100 176114
919 002710 104000
920
921
922
923
924 002712 104400
925 002714 012777 000300 176074
926 002722 032777 000200 176074
927 002730 001010
928 002732 004767 007326
929 002736 000005
930 002740 032777 000200 176056
931 002746 001001
932 002750 104000
933 002752 012777 003006 176054 P6INV:
934 002760 012737 000340 000126
935 002766 052777 000100 176030
936 002774 000240
937 002776 042777 000100 176020
938 003004 000405
939 003006 022626 P6ERR:
940 003010 042777 000100 176006
941 003016 104000
942
943
944
945
946
947 003020 104400
948 003022 012777 000240 175766
949 003030 032777 000200 175766
950 003036 001010

```

```

*****
TEST25 TEST THAT DR11-B DOES NOT INTERRUPT WITH PROC AT LEVEL 7
*****
SCOPE
MOV #4000,ICOUNT ;STATUS AT LEVEL 7
MOV #340,IPSW ;CHECK READY BIT
BIT #BIT7,DRST ;IS IT SET?
BNE P7INV ;IS IT SET?
JSR %7,CKSWR
RESET ;INIT TO SET READY
BIT #BIT7,DRST ;SEE IF READY IS SET NOW
BNE .+4 ;IS READY SET?
HLT ;READY CAN'T BE SET BY INIT
P7INV: MOV #P7ERR,DRINV ;SET UP INT VECTOR
MOV #340,IPSW
BIS #BIT6,DRST ;SET IE
NOP
BIC #BIT6,DRST ;DR11-B DIDN'T INTERRUPT - CLEAR IE
BR X1
P7ERR: CMP (%6)+,(%6)+ ;RESTORE STACK
BIC #BIT6,DRST ;CLEAR IE
HLT ;DR11-B INTERRUPTED, BUT IT SHOULDN'T HAVE
*****
TEST26 TEST THAT DR11-B DOES NOT INTERRUPT WITH PROC AT LEVEL 6
*****
X1: SCOPE
MOV #300,IPSW ;STATUS AT LEVEL 6
BIT #BIT7,DRST ;CHECK READY BIT
BNE P6INV ;IS IT SET?
JSR %7,CKSWR
RESET ;INIT TO SET READY
BIT #BIT7,DRST ;SEE IF READY IS SET NOW
BNE .+4 ;IS READY SET?
HLT ;READY CAN'T BE SET BY INIT
P6INV: MOV #P6ERR,DRINV ;SET UP INT VECTOR
MOV #340,IPSW
BIS #BIT6,DRST ;SET IE
NOP
BIC #BIT6,DRST ;DR11-B DIDN'T INTERRUPT - CLEAR IE
BR X2
P6ERR: CMP (%6)+,(%6)+ ;RESTORE STACK
BIC #BIT6,DRST ;CLEAR IE
HLT ;DR11-B INTERRUPTED, BUT IT SHOULDN'T HAVE
*****
TEST27 TEST THE DR11-B INTERRUPT WITH PROC AT LEVEL 5
*****
X2: SCOPE
MOV #240,IPSW ;STATUS AT LEVEL 5
BIT #BIT7,DRST ;CHECK READY BIT
BNE P5INV ;IS IT SET?

```

```

952 003044 000005          RESET          ;INIT TO SET READY
953 003046 032777 000200 175750  BIT          #BIT7,DRST  ;SEE IF READY IS SET NOW
954 003054 001001          BNE          .+4      ;IS IT SET?
955 003056 104000          HLT          ;RDY CAN'T BE SET BY INIT
956 003060 017767 175730 176010 PSINV: MOV        JSR,TEMP  ;GET SWITCH SETTINGS
957 003066 042767 177770 176002  BIC          #177770,TEMP ;ISOLATE BR LEVEL
958 003074 012777 003142 175732  MOV        #PSERR,DRINV ;SET UP INT VECTOR
959 003102 012737 000340 000126  MOV        #340,#126
960 003110 052777 000100 175706  BIS        #BIT6,DRST  ;SET IE
961 003116 000240          NOP
962 003120 042777 000100 175676  BIC        #BIT6,DRST  ;DR11-B DIDN'T INTERRUPT - CLEAR IE
963 003126 022767 000006 175742  CMP        #6,TEMP    ;BR LEVEL = 6 ?
964 003134 001001          BNE          IS      ;NO EVERYTHING IS OK
965 003136 104000          HLT          ;DR11-B WITH BR = 6 SHOULD HAVE INTERRUPTED
966 003140 000411          BR          X3
967 003142 022626          PSERR: CMP        (%6)+,(%6)+ ;RESTORE STACK
968 003144 042777 000100 175652  BIC        #BITS,DRST ;CLEAR IE
969 003152 022767 000006 175716  CMP        #6,TEMP    ;BR LEVEL = 6 ?
970 003160 001401          BEQ        X3        ;YES EVERYTHING IS OK
971 003162 104000          HLT          ;DR11-B WITH BR NOT = 6 SHOULD NOT HAVE INTERRUPTED
972
973
974 ;*****
975 ;*****
976 003164 104400          X3: SCOPE
977 003166 012777 000200 175622  MOV        #200,PSW   ;STATUS AT LEVEL 4
978 003174 032777 000200 175622  BIT        #BIT7,DRST ;CHECK READY BIT
979 003202 001010          BNE        P4INV    ;IS IT SET?
980 003204 004767 007054          JSR        %7,CKSWR
981 003210 000005          RESET
982 003212 032777 000200 175604  BIT        #BIT7,DRST ;INIT TO SET READY
983 003220 001001          BNE        .+4      ;SEE IF READY IS SET NOW
984 003222 104000          HLT          ;IS IT SET?
985 003224 017767 175564 175644 P4INV: MOV        JSR,TEMP  ;RDY CAN'T BE SET BY INIT
986 003232 042767 177770 175636  BIC        #177770,TEMP ;GET SWITCH SETTINGS
987 003240 012777 003314 175566  MOV        #P4INT,DRINV ;ISOLATE BR LEVEL
988 003246 012737 000340 000126  MOV        #340,#126  ;SET UP INT VECTOR
989 003254 052777 000100 175542  BIS        #BIT6,DRST  ;SET IE
990 003262 000240          NOP
991 003264 005077 175534          CLR        DRST     ;CLEAR IE
992 003270 012777 000126 175536  MOV        #126,DRINV ;RESTORE INTERRUPT VECTOR
993 003276 022767 000004 175572  CMP        #4,TEMP    ;BR LEVEL = 4 ?
994 003304 001401          BEQ        IS      ;YES EVERYTHING IS OK
995 003306 104000          HLT          ;DR11-B DIDN'T INTERRUPT
996 003310 000167 000026          IS: JMP        X4
997 003314 005077 175504          P4INT: CLR        DRST  ;CLEAR IE
998 003320 022626          CLP        (%6)+,(%6)+ ;REPOSITION THE STACK AFTER AN INTERRUPT
999 003322 012777 000126 175504  MOV        #126,DRINV ;RESTORE INTERRUPT VECTOR
1000 003330 022767 000004 175540  CMP        #4,TEMP    ;BR LEVEL = 4 ?
1001 003336 001001          BNE        X4        ;NO EVERYTHING IS OK
1002 003340 104000          HLT          ;DR11-B WITH BR = 4 SHOULD NOT HAVE INTERRUPTED
1003
1004
1005 ;*****
1006 ;*****

```



```

1008 003342 104400 X4: SCOPE
1009 003344 012777 000140 175444 MOV #140,DRST ;STATUS AT LEVEL 3
1010 003352 032777 000200 175444 BIT #BIT7,DRST ;CHECK READY BIT
1011 003360 001010 BNE P3INV ;IS IT SET?
1012 003362 004767 006676 JSP %7,CKSWR
1013 003366 000005 RESET ;INIT TO SET READY
1014 003370 032777 000200 175426 BIT #BIT7,DRST ;SEE IF READY IS SET NOW
1015 003376 001001 BNE .+4 ;IS IT SET
1016 003400 104000 HLT ;READY CAN'T BE SET BY INIT
1017 003402 012777 003452 175424 P3INV: MOV #P3INT,DRINV ;SET UP INT VECTOR
1018 003410 012737 000340 000126 MOV #340,DRST
1019 003416 052777 000100 175400 BIS #BIT6,DRST ;SET IE
1020 003424 000240 NOP
1021 003426 005077 175372 CLR DRST ;CLEAR IE
1022 003432 012777 000126 175374 MOV #126,DRINV ;RESTORE INTERRUPT VECTOR
1023 003440 005037 000126 CLR DRST ;RESTORE TRAP CATCHER
1024 003444 104000 HLT ;DR11-B DIDN'T INTERRUPT
1025 003446 000167 000020 JMP X5
1026 003452 005077 175346 P3INT: CLR DRST ;CLEAR IE
1027 003456 022626 CMP (%6)+,(%6)+ ;REPOSITION THE STACK AFTER AN INTERRUPT
1028 003460 012777 000126 175346 MOV #126,DRINV ;RESTORE INTERRUPT VECTOR
1029 003466 005037 000126 CLR DRST ;RESTORE TRAP CATCHER
1030
1031
1032
1033 ;*****
1034 ;*****
1035 003472 104400 X5: SCOPE
1036 003474 005077 175324 CLR DRST ;CLEAR FUNCTION BITS
1037 003500 032777 000016 175316 BIT #16,DRST ;CHECK FUNCTION BITS
1038 003506 001401 BEQ .+4 ;FUNCTION BITS CLEAR?
1039 003510 104000 HLT ;FUNCTION BITS NOT CLEAR
1040 003512 052777 000002 175304 BIS #BIT1,DRST ;SET FNCT1
1041 003520 032777 001000 175276 BIT #BIT9,DRST ;CHECK DSTAT C
1042 003526 001001 BNE .+4 ;IS IT SET?
1043 003530 104000 HLT ;DSTAT C IS CLEAR
1044 003532 032777 006000 175264 BIT #6000,DRST ;CHECK THAT DSTAT A AND DSTAT B ARE CLEAR
1045 003540 001401 BEQ .+4 ;ARE THEY CLEAR?
1046 003542 104000 HLT ;DSTAT A AND/OR DSTAT B IS SET
1047 003544 005077 175254 CLR DRST ;CLEAR DRST
1048 003550 052777 000004 175246 BIS #BIT2,DRST ;SET FNCT2
1049 003556 032777 002000 175240 BIT #BIT10,DRST ;CHECK DSTAT B
1050 003564 001001 BNE .+4 ;IS IT SET?
1051 003566 104000 HLT ;DSTAT B IS CLEAR
1052 003570 032777 005000 175226 BIT #5000,DRST ;CHECK THAT DSTAT A AND DSTAT C ARE CLEAR
1053 003576 001401 BEQ .+4 ;ARE THEY CLEAR?
1054 003600 104000 HLT ;DSTAT A AND/OR DSTAT B IS SET
1055 003602 005077 175216 CLR DRST ;CLEAR DRST
1056 003606 052777 000010 175210 BIS #BIT3,DRST ;SET FNCT3
1057 003614 032777 004000 175202 BIT #BIT11,DRST ;CHECK DSTAT A
1058 003622 001001 BNE .+4 ;IS IT SET?
1059 003624 104000 HLT ;DSTAT A IS CLEAR
1060 003626 032777 003000 175170 BIT #3000,DRST ;CHECK THAT DSTAT B AND DSTAT C ARE CLEAR
1061 003634 001401 BEQ .+4 ;ARE THEY CLEAR?
1062 003636 104000 HLT ;DSTAT B AND/OR DSTAT C IS SET

```

```

1064
1065
1066
1067 003644 104400
1068 003646 005777 175152
1069 003652 100027
1070 003654 032777 020000 175142
1071 003662 001401
1072 003664 104000
1073 003666 032777 040000 175130
1074 003674 001410
1075 003676 042777 040000 175120
1076 003704 032777 040000 175112
1077 003712 001401
1078 003714 104000
1079 003716 005077 175100
1080 003722 005777 175076
1081 003726 001401
1082 003730 104000
1083 003732 012777 177777 175060
1084 003740 012777 001036 175054
1085 003746 005077 175054
1086 003752 012767 052525 175056
1087 003760 012777 004024 175046
1088 003766 012777 000005 175034
1089 003774 005077 175016
1090 004000 012777 000101 175016
1091 004006 005067 000002
1092 004012 005227 000001
1093 004016 001375
1094 004020 104000
1095 004022 000424
1096 004024 004767 003204
1097 004030 005777 174764
1098 004034 001401
1099 004036 104000
1100 004040 022777 001040 174754
1101 004046 001401
1102 004050 104000
1103 004052 022777 052525 174746
1104 004060 001401
1105 004062 104000
1106 004064 004767 003054
1107 004070 022626
1108 004072 000403
1109 004074 005077 174724
1110 004100 000662
1111
1112
1113
1114 004102 104400
1115 004104 012777 177777 174706
1116 004112 012777 001036 174702
1117 004120 005067 174712
1118 004124 012777 052525 174674

```

```

;*****
; TEST 33 TEST FOR 1 DATI NPR TRANSFER (WITH M968 IN USER SLOTS)
;*****
SCOPE
TNPR1: TST @DRST ;CHECK ERROR BIT
        BPL NRRDY ;IS IT CLEAR?
        BIT @BIT13,@DRST ;CHECK ATTN
        BEQ .+4 ;IS ATTN CLEAR
        HLT ;ATTN IS SET
        BIT @BIT14,@DRST ;CHECK NEX
        BEQ N1413 ;IS NEX CLEAR?
        BIC @BIT14,@DRST ;TRY TO CLEAR NEX
        BIT @BIT14,@DRST ;CHECK AGAIN
        BEQ .+4 ;NEX STILL SET
        HLT ;NEX CAN'T BE CLEARED BY MOVING A 0 TO IT
N1413: CLR @DRBA ;TRY TO CLEAR BAOF
        TST @DRST ;CHECK ERROR BIT AGAIN
        BEQ .+4 ;IS IT CLEAR
        HLT ;ERROR CAUSED BY SOMETHING OTHER THAN NEX,ATTN, OR BAOF
NRRDY: MOV #-1,@DRWC ;SET UP FOR 1 TRANSFER
        MOV @NPR1,@DRBA ;TRANSFER FROM BUS ADDRESS IN NPR1
        CLR @DRDB ;GET READY TO RECEIVE DATA
        MOV @52525,NPR1 ;SET UP TRANSFER DATA
        MOV @INTB,@DRINV ;INTERRUPT VECTOR TO INTB
        MOV @5,@DRVS ;INTERRUPT PRIORITY TO LEVEL 5
        CLR @PSW ;LET THE DR11-B INTERRUPT
        MOV @101,@DRST ;IE AND DO TO DRST
        CLR @15+2 ;WAIT FOR NPR AND INTERRUPT
        INC @1 ;
        BNE @15 ;
        HLT ;NO DR11-B INTERRUPT
        BR T33CLR ;CLEAR IE
INTB: JSR %7,ERRCHK
        TST @DRWC ;TEST DRWC
        BEQ .+4 ;IS DRWC EQUAL TO ZERO?
        HLT ;DRWC NOT EQUAL TO ZERO
        CMP @NPR1+2,@DRBA ;COMPARE CORRECT DRBA WITH DRBA
        BEQ .+4 ;IS THE DRBA CORRECT?
        HLT ;DRBA IS WRONG
        CMP @52525,@DRDB ;CHECK FOR CORRECT DATA
        BEQ .+4 ;DATA GET TRANSFERRED?
        HLT ;BAD DATA IN DRDB
        JSR %7,NORMAL
        CMP (%6)+,(%6)+ ;RESTORE STACK
        BR TNPRO ;GO TO NEXT TEST (NPR OUT)
T33CLR: CLR @DRST ;CLEAR IE
        BR TNPR1 ;TRY TEST AGAIN
;*****
; TEST 34 TEST FOR 1 DATO NPR TRANSFER (WITH M968 IN USER SLOTS)
;*****
TNPRO: SCOPE
        MOV #-1,@DRWC ;SET UP FOR 1 TRANSFER
        MOV @NPR1,@DRBA ;TRANSFER TO BUS ADDRESS IN NPR1
        CLR NPR1 ;GET READY TO RECEIVE DATA
        MOV @52525,@DRDB ;SET UP TO TRANSFER DATA

```

```

1120 004140 016777 174666 174662      MOV      DRINL,DRVS      ; INTERRUPT STATUS TO LEVEL DRINL
1121 004146 005077 174644      CLR      @PSW           ; PROC STATUS TO ZERO
1122 004152 012777 000103 174644      MOV      #103,DRST     ; IE, FNCT1(C1 CONTROL), AND DO TO DRST
1123 004160 005067 000002      CLR      IS+2          ; WAIT FOR NPR AND INTER
1124 004164 005227 000001      INC      #1
1125 004170 001375      BNE      IS
1126 004172 104000      HLT
1127 004174 000424      BR       T34CLR        ; NO DR11-B INTERRUPT
1128 004176 004767 003032      JSR      %7,ERRCHK     ; CLEAR IE
1129 004202 005777 174612      TST      @DRWC         ; TEST DRWC
1130 004206 001401      BEQ      .+4           ; IS DRWC EQUAL TO ZERO?
1131 004210 104000      HLT
1132 004212 022777 001040 174602      CMP      @NPR1+2,@DRBA ; DRWC EQUAL TO ZERO
1133 004220 001401      BEQ      .+4           ; COMPARE CORRECT DRBA WITH DRBA
1134 004222 104000      HLT                   ; IS THE DRBA CORRECT?
1135                                     ; DRBA IS WRONG
1136 004224 026727 174606 052525      CMP      NPR1,#52525   ; CHECK FOR CORRECT DATA
1137 004232 001401      BEQ      .+4           ; CORRECT DATA TRANSFERRED?
1138 004234 104000      HLT                   ; BAD DATA
1139 004236 004767 002702      JSR      %7,NORMAL
1140 004242 022626      CMP      (%6)+,(%6)+   ; RESTORE STACK
1141 004244 000403      BR       T35           ; GO TO NEXT TEST
1142 004246 005077 174552      CLR      @DRST        ; CLEAR IE
1143 004252 000713      BR       TNPRO         ; TRY TEST AGAIN
1144
1145 ;*****
1146 ; TEST 35 STRING OF 10 DATI'S (WITH M968 IN USER SLOTS)
1147 ;*****
1148 004254 104400      T35: SCOPE
1149 004256 012767 000020 174562      MOV      #20,BUFLEN    ; LENGTH OF BUFFER=20
1150 004264 004767 002416      JSR      %7,LODBUF     ; LOAD THE BUFFER WITH INCREMENTING PATTERN
1151 004270 006267 174552      ASR      BUFLEN        ; BUFLEN=10
1152 004274 016767 174546 174552      MOV      BUFLEN,WLEN   ; PREPARE NUMBER FOR DRWC
1153 004302 005467 174546      NEG      WLEN          ; 2'S COMPLEMENT OF BUFLEN
1154 004306 016777 174542 174504      MOV      WLEN,@DRWC    ; SET UP DRWC
1155 004314 016777 174522 174500      MOV      INBUF,@DRBA   ; SET UP DRBA
1156 004322 012777 177777 174476      MOV      #-1,@DRDB    ; MAINT AIDE
1157 004330 012777 007010 174476      MOV      @INTA,@DRINV  ; INT VECTOR TO INTA
1158 004336 016777 174470 174464      MOV      DRINL,DRVS   ; INT VECTOR TO PRIORITY DRINL
1159 004344 005077 174446      CLR      @PSW         ; LET THE DR11-B INTERRUPT
1160 004350 012777 000101 174446      MOV      #101,DRST    ; IE AND DO TO DRST
1161 004356 000777      BR       .             ; WAIT FOR INTERRUPT
1162 004360 022777 000007 174440      CMP      #7,@DRDB     ; CHECK THAT WORD #10 OF INBUF IS IN DRBA
1163 004366 001401      BEQ      .+4           ; IS IT?
1164 004370 104000      HLT                   ; BAD DATA IN DRDB
1165
1166 ;*****
1167 ; TEST 36 STRING OF 10 DATO'S (WITH M968 IN USER SLOTS)
1168 ;*****
1169 004372 104400      SCOPE
1170 004374 012767 000020 174444      MOV      #20,BUFLEN    ; LENGTH OF BUFFER=20
1171 004402 004767 002300      JSR      %7,LODBUF     ; LOAD THE BUFFER WITH INCREMENTING PATTERN
1172 004406 006267 174434      ASR      BUFLEN        ; BUFLEN=10
1173 004412 016767 174430 174434      MOV      BUFLEN,WLEN   ; PREPARE NUMBER FOR DRWC
1174 004420 005467 174430      NEG      WLEN          ; 2'S COMPLEMENT OF BUFLEN

```

```

1176 004432 016777 174404 174362      MOV      INBUF,DRBA      ;SET UP DRBA
1177 004440 012777 052525 174360      MOV      #52525,DRDB    ;SET UP DRDB
1178 004446 012777 007010 174360      MOV      #INTA,DRINVS   ;INTERRUPT VECTOR TO INTA
1179 004454 016777 174352 174346      MOV      DRINL,DRVS     ;INTERRUPT VECTOR TO PRIORITY DRINL
1180 004462 005077 174330      CLR      DRPSW          ;LET THE DR11-B INTERRUPT
1181 004466 012777 000103 174330      MOV      #103,DRST     ;IE, FNCT1(C1 CONTROL), AND DO TO DRST
1182 004474 000777      BR                      ;WAIT FOR INTERRUPT
1183 004476 004767 002464      JSR      %7,DATOCK     ;CHECK INBUF
1184
1185
1186 ;*****
1187 ;TEST 37 STRING OF 200 DATI'S
1188 ;*****
1188 004502 104400      SCOPE
1189 004504 012767 007200 174334      MOV      #200,BUFLEN    ;LENGTH OF BUFFER=200
1190 004512 004767 002170      JSR      %7,LODBUF     ;LOAD THE BLFFER WITH INCREMENTING PATTERN
1191 004516 016767 174324 174330      MOV      BUFLN,WCLN     ;PREPARE NUMBER FOR DRWC
1192 004524 005467 174324      NEG      WCLN           ;2'S COMPLEMENT OF BUFLN
1193 004530 016777 174320 174262      MOV      WCLN,DRWC     ;SET UP DRWC
1194 004536 016777 174300 174256      MOV      INBUF,DRBA    ;SET UP DRBA
1195 004544 012777 177777 174254      MOV      #-1,DRDB     ;MAINT AIDE
1196 004552 017777 007010 174254      MOV      #INTA,DRINVS  ;INT VECTOR TO INTA
1197 004560 017777 174246 174242      MOV      DRINL,DRVS    ;INT VECTOR TO PRIORITY DRINL
1198 004566 005077 174224      CLR      DRPSW         ;LET THE DR11-B INTERRUPT
1199 004572 017777 000101 174224      MOV      #101,DRST    ;IE AND DO TO DRST
1200 004600 003777      BR                      ;WAIT FOR INTERRUPT
1201 004602 022777 000177 174216      CMP      #177,DRDB    ;CHECK THAT WORD #200 OF INBUF IS IN DRBA
1202 004610 001401      BEQ     .+4            ;IS IT?
1203 004612 104000      HLT                    ;BAD DATA IN DRDB
1204 ;*****
1205 ;TEST 40 STRING OR 200 DATO'S
1206 ;*****
1207 004614 104400      SCOPE
1208 004616 012767 000201 174222      MOV      #201,BUFLEN    ;LENGTH OF BUFFER=201
1209 004624 004767 002056      JSR      %7,LODBUF     ;LOAD THE BUFFER WITH INCREMENTING PATTERN
1210 004630 005367 174212      DEC      BUFLN         ;BUFLN=200
1211 004634 016767 174206 174212      MOV      BUFLN,WCLN    ;PREPARE NUMBER FOR DRWC
1212 004642 005467 174206      NEG      WCLN           ;2'S COMPLEMENT OF BUFLN
1213 004646 016777 174202 174144      MOV      WCLN,DRWC     ;SET UP DRWC
1214 004654 016777 174162 174140      MOV      INBUF,DRBA    ;SET UP DRBA
1215 004662 012777 052525 174136      MOV      #52525,DRDB   ;SET UP DRDB
1216 004670 012777 007010 174136      MOV      #INTA,DRINVS  ;INTERRUPT VECTOR TO INTA
1217 004676 016777 174130 174124      MOV      DRINL,DRVS    ;INTERRUPT VECTOR TO PRIORITY DRINL
1218 004704 005077 174106      CLR      DRPSW         ;LET THE DR11-B INTERRUPT
1219 004710 012777 000103 174106      MOV      #103,DRST    ;IE, FNCT1, AND DO TO DRST
1220 004716 000777      BR                      ;WAIT FOR INTERRUPT
1221 004720 004767 002242      JSR      %7,DATOCK     ;CHECK INBUF
1222
1223
1224 ;*****
1225 ;TEST 42 TEST THAT DOING F DATO TO THE DIODE MEMORY CAUSES NEX
1226 ;*****
1226 004724 104400      SCOPE
1227 004726 012777 177776 174064      MOV      #-2,DRWC     ;SET UP DRWC
1228 004734 016777 174100 174060      MOV      DIJMEM,DRBA   ;SET UP DRBA
1229 004742 012777 004776 174064      MOV      #NEXCHK,DRINVS ;INTERRUPT VECTOR TO NEXCHK
1230 004750 016777 174056 174052      MOV      DRINL,DRVS    ;INTERRUPT STATUS TO LEVEL DRINL

```

```

1232 004762 012777 000163 174034      MOV      #163,DRST      ; IE, FNCT1, XBA17, XBA16, AND GO TO DRST
1233 004770 005237 177560      INC      #177560      ; WAIT FOR INTERRUPT
1234 004774 104000      HLT      ; NO DR11-B INTERRUPT
1235 004776 042777 000100 174020 NEXCHK: BIC      #BIT6,DRST   ; CLEAR INTERRUPT ENABLE
1236 005004 005777 174014      TST      DRST         ; TEST DRST
1237 005010 001001      BNE     .+4          ; ERROR SET?
1238 005012 104000      HLT      ; ERROR NOT SET
1239 005014 105777 174004      TSTB    DRST         ; TEST FOR READY
1240 005020 001001      BNE     .+4          ; READY SET?
1241 005022 104000      HLT      ; READY ISN'T SET
1242 005024 032777 040000 173772      BIT      #BIT14,DRST  ; CHECK NEX
1243 005032 001001      BNE     .+4          ; NEX SET?
1244 005034 104000      HLT      ; NEX IS CLEAR
1245 005036 042777 040000 173760      BIC      #BIT14,DRST  ; CLEAR NEX
1246 005044 022626      CMP     (%6)+,(%6)+  ; RESTORE THE STACK
1247 005046 004767 002072      JSR     %7,NORMAL
1248
1249

```

```

; *****
; TEST 43 TEST THAT CROSSING A 32K BOUNDARY DOES NOT CAUSE
; A BAOF AND DOES NOT FORCE AN ERROR
; *****

```

```

1253 005052 104400      SCOPE
1254 005054 012767 000010 005160      MOV     #10,ICOUNT
1255 005062 012777 177760 173730      MOV     #-20,DRWC    ; SET UP DRWC
1256 005070 012777 177776 173724      MOV     #-2,DRBA     ; SET UP DRBA FOR PROC STATUS ADDRESS
1257 005076 012777 005132 173730      MOV     #BAOFCK,DRINV ; INTERRUPT VECTOR TO BAOFCK
1258 005104 016777 173722 173716      MOV     DRINL,DRVS   ; INTERRUPT STATUS TO LEVEL DRINL
1259 005112 005077 173700      CLR     DPSW         ; LET THE DR11-B INTERRUPT
1260 005116 012777 000163 173700      MOV     #163,DRST    ; IE, FNCT1, XBA17, XBA16, AND GO TO DRST
1261 005124 005237 177560      INC     #177560     ; WAIT FOR INTERRUPT
1262 005130 104000      HLT     ; NO DR11-B INTEPRUPT
1263 005132 042777 000100 173664 BAOFCK: BIC     #BIT6,DRST   ; CLEAR INTERRUPT ENABLE
1264 005140 022626      CMP     (%6)+,(%6)+  ; RESTORE THE STACK
1265 005142 005777 173656      TST     DRST         ; TEST DRST
1266 005146 100401      BMI     .+4          ; ERROR SET?
1267 005150 104000      HLT     ; ERROR NOT SET
1268 005152 105777 173646      TSTB    DRST         ; TEST FOR READY
1269 005156 100401      BMI     .+4          ; READY SET?
1270 005160 104000      HLT     ; READY ISN'T SET
1271 005162 042777 040000 173634      BIC     #BIT14,DRST  ; CLEAR NEX
1272 005170 032777 060000 173626      BIT     #60000,DRST  ; CHECK NEX AND ATTN
1273 005176 001401      BEQ     .+4          ; ARE THEY CLEAR?
1274 005200 104000      HLT     ; NEX AND/OR ATTN IS SET
1275 005202 005777 173616      TST     DRST         ; TEST FOR ERROR
1276 005206 100001      BPL     .+4          ; IS ERROR CLEAR?
1277 005210 104000      HLT     ; ERROR IS SET
1278 005212 005077 173604      CLR     DRBA        ; CLEAR BUS ADDRESS REGISTER
1279
1280 005216 004767 005042      JSR     %7,CKSWR
1281 005222 000005      RESET  ; INIT
1282 005224 004767 001714      JSR     %7,NORMAL
1283
1284
1285
1286

```

```

; *****
; TEST 44 TEST THAT RESET CLEARS DRDB
; *****

```



```

1288 005232 012767 000010 005002
1289 005240 012777 177777 173560
1290 005246 004767 005012
1291 005252 000005
1292 005254 005777 173546
1293 005260 001401
1294 005262 104000
1295
1296
1297
1298
1299 005264 104400
1300 005266 012767 004000 004746
1301 005274 012777 177777 173524
1302 005302 022777 177777 173516
1303 005310 001401
1304 005312 104000
1305
1306
1307
1308
1309 005314 104400
1310 005316 012777 052525 173502
1311 005324 022777 052525 173474
1312 005332 001401
1313 005334 104000
1314 005336 012777 125252 173462
1315 005344 022777 125252 173454
1316 005352 001401
1317 005354 104000
1318
1319
1320
1321
1322 005356 104400
1323 005360 005067 004656
1324 005364 005001
1325 005366 005077 173434
1326 005372 020177 173430
1327 005376 001401
1328 005400 104000
1329 005402 005277 173420
1330 005406 005201
1331 005410 001370
1332
1333
1334
1335
1336
1337 005412 104400
1338 005414 004767 004644
1339 005420 000005
1340 005422 032777 000200 173374
1341 005430 001001
1342 005432 104000

```

```

MOV #10,ICOUNT
MOV #-1,DZDRB ;ALL ONES TO DRBA
JSR %7,CKSWR
RESET ;INIT
TST DZDRB ;LOOKING FOR Z-BIT TO SET
BEQ .+4 ;DID DRDB GET CLEARED?
HLT ;DRDB NOT CLEAR

:*****
: TEST 45 TEST THAT ALL DRDB BITS CAN BE SET
:*****
SCOPE
MOV #4000,ICOUNT
MOV #-1,DZDRB ;SET ALL BITS IN DRDB
CMP #-1,DZDRB ;LOOKING FOR Z-BIT TO SET
BEQ .+4 ;SEE IF ALL BITS GOT SET
HLT ;ALL DRDB BITS AREN'T SET

:*****
: TEST 46 TEST THAT DRDB CAN HOLD ALTERNATE ONE'S AND ZERO'S
:*****
SCOPE
MOV #052525,DZDRB ;ALT 0'S AND 1'S TO DRDB
CMP #052525,DZDRB ;LOOKING FOR Z-BIT TO SET
BEQ .+4 ;DOES DRDB HAVE THE CORRECT PATTERN?
HLT ;DRDB IS WRONG
MOV #125252,DZDRB ;ALT 1'S AND 0'S TO DRDB
CMP #125252,DZDRB ;LOOKING FOR Z-BIT TO SET
BEQ .+4 ;DOES DRDB HAVE THE CORRECT PATTERN
HLT ;DRDB IS WRONG

:*****
: TEST 47 INCREMENTING PATTERN TO WRAP-AROUND IN DRDB
:*****
SCOPE
CLR ICOUNT
CLR %1 ;SET-UP
CLR DZDRB ;SET-UP
INCD8: CMP %1,DZDRB ;SEE IF THEY ARE EQUAL
BEQ .+4 ;ARE THEY EQUAL?
HLT ;THEY'RE NOT EQUAL
INC DZDRB ;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,CKSWR
RESET ;INIT
BIT #BIT7,DZDRST ;CHECK DRST
BNE .+4 ;IS READY SET?
HLT ;READY IS CLEAR

```

```

1344 005442 001401          BEQ      .+4          ;ARE THEY ALL CLEAR?
1345 005444 104000          HLT
1346
1347
1348 ;:*****
1349 ;:TEST 51 TEST THAT BA00 READS AS A ZERO WITH MAINT BOARD INSERTED
1350 ;:*****
1351 005446 104400          SCOPE
1352 005450 012767 004000 004564  MOV     #4000,ICOUNT
1353 005456 032777 000001 173336  BIT     #BIT0,DRBA      ;TEST BIT 0 OF DRBA
1354 005464 001401          BEQ      .+4          ;IS IT CLEAR?
1355 005466 104000          HLT                    ;BA00 IS SET
1356
1357
1358 ;:*****
1359 ;:TEST 52 TEST THAT GO CLEARS READY
1360 ;:*****
1361 005470 104400          SCOPE
1362 005472 012767 004000 004542  MOV     #4000,ICOUNT
1363 005500 012777 177500 173312  MOV     #-200,DRWC      ;SET-UP DRWC
1364 005506 016777 173330 173306  MOV     INBUF,DRBA      ;SET-UP DRBA
1365 005514 105777 173304          TSTB    DRST           ;CHECK READY
1366 005520 100401          BMI     .+4          ;IS READY SET?
1367 005522 104000          HLT                    ;READY IS CLEAR
1368 005524 012777 000011 173272  MOV     #11,DRST        ;FNCT3 (NON-BURST) AND GO TO DRST
1369 005532 105777 173266          TSTB    DRST           ;CHECK READY
1370 005536 100001          BPL     .+4          ;IS READY CLEAR?
1371 005540 104000          HLT                    ;READY IS STILL SET
1372 005542 005067 173310          CLR     RDYCHK          ;CLEAR READY CHECK
1373 005546 105777 173252  TSTRDY: TSTB    DRST           ;CHECK READY
1374 005552 100406          BMI     DONE           ;IF SET GO TO DONE
1375 005554 062767 000004 173274  ADD     #4,RDYCHK        ;CHECKING TIME FOR READY TO BE SET
1376 005562 100401          BMI     .+4          ;IF RDYCHK GETS NEGATIVE IT TOOK TOO LONG
1377 005564 000770          BR      TSTRDY         ;CHECK AGAIN
1378 005566 104000          HLT                    ;READY GOT CLEARED BUT NEVER SET AGAIN
1379 005570 000240          NOP                    ;GO TO NEXT TEST
1380
1381 ;:*****
1382 ;:TEST 55 TEST THAT GIVING A DO WITHOUT CLEARING A PREVIOUS ERROR
1383 ;:CAUSES ANOTHER INTERRUPT
1384 ;:FOR THIS TEST, THE NORMAL INTERRUPT MECHANISM SHOULD BE WORKING
1385 ;:*****
1386 005572 104400          SCOPE
1387 005574 012777 177760 173216  MOV     #-20,DRWC      ;SET-UP DRWC
1388 005602 016777 173232 173212  MOV     DIOMEM,DRBA     ;SET-UP DRBA
1389 005610 012777 005644 173216  MOV     ERRD0,DRINV     ;INTERRUPT VECTOR TO ERRD0
1390 005616 012777 000140 173204  MOV     #140,DRVS       ;INTERRUPT STATUS TO LEVEL 4
1391 005624 005077 173166          CLR     DPSW           ;LET THE DR11-B INTERRUPT
1392 005630 012777 000163 173166  MOV     #163,DRST       ;IE XBA17 XBA16 FNCT1 AND GO TO DRST
1393 005636 005277 173216          INC     DTKS           ;WAIT FOR INTERRUPT
1394 005642 104000          HLT                    ;NO DR11-B INTERRUPT
1395 005644 005777 173154  ERRD0: TST     DRST       ;TEST DRST
1396 005650 100401          BMI     .+4          ;ERROR SET?
1397 005652 104000          HLT                    ;ERROR IS CLEAR - SHOULD HAVE NEX
1398 005654 012777 005726 173152  MOV     #ERRD01,DRINV   ;INTERRUPT VECTOR TO ERRD01
1399 005662 005077 173134          CLR     DRBA           ;PREVENT CAUSING ANOTHER ERROR

```

```

1400 005674 012777 177777 173116      MOV      #-1,DRWC      ;SET-UP DRWC
1401 005702 005277 173116              INC      DRST        ;DO TO DRST
1402 005706 005067 000002              CLR      IS+2
1403 005712 005227 000001      15:      INC      #1
1404 005716 001375              BNE     IS
1405 005720 104000              HLT
1406 005722 162706 000004      ERRD01: SUB     #4,%6      ;NO DR11-B INTERRUPT
1407 005726 005777 173072              TST     DRST        ;FAKE AN INTERRUPT WITH STACK
1408 005732 100401              BMI     .+4         ;CHECK ERROR
1409 005734 104000              HLT              ;ERROR SET?
1410                                     ;ERROR IS CLEAR - SHOULD BE SET BECAUSE
1411 005736 062706 000010      ADD     #10,%6      ;PREVIOUS ERROR WAS NOT CLEARED
1412 005742 004767 001176      JSR     %7,NORMAL   ;REPOSITION THE STACK
1413
1414 ;*****
1415 ;TEST 56 STRING OF 200 DATI'S NON-BURST MODE
1416 ;*****
1417 005746 104400      SCOPE
1418 005750 012767 000200 173070      MOV     #200,BUFLEN ;LENGTH OF BUFFER=200
1419 005756 004767 000724              JSR     %7,LODBUF   ;LOAD THE BUFFER WITH INCREMENTING PATTERN
1420 005762 016767 173060 173064      MOV     BUFLN,WCLN  ;PREPARE NUMBER FOR DRWC
1421 005770 005467 173060      NEG     WCLN        ;2'S COMPLEMENT OF BUFLN
1422 005774 016777 173054 173016      MOV     WCLN,DRWC   ;SET-UP DRWC
1423 006002 016777 173034 173012      MOV     INBUF,DRBA  ;SET-UP DRBA
1424 006010 012777 177777 173010      MOV     #-1,DRDB   ;MAINT AIDE
1425 006016 012777 007010 173010      MOV     INTA,DRINV  ;INT VECTOR TO INTA
1426 006024 016777 173002 172776      MOV     DRINL,DRVS  ;INT VECTOR TO PRIORITY DRINL
1427 006032 005077 172760      CLR     DRPSW       ;LET THE DR11-B INTERRUPT
1428 006036 012777 000111 172760      MOV     #111,DRST  ;IE FNCT3, AND DO TO DRST
1429 006044 005267 173002              INC     BRWAIT      ;USE A WAIT OR BR. INSTRUCTION
1430 006050 032767 000001 172774      BIT     #BIT0,BRWAIT ;SEE WHICH ONE
1431 006056 001403              BEQ     DATINB     ;BIT 0 CLEAR=BR.
1432 006060 000001              WAIT
1433 006062 000240      NOP
1434 006064 000401      BR     .+4
1435 006066 000777      DATINB: BR
1436 006070 022777 000177 172730      CMP     #177,DRDB  ;CHECK THAT WORD #200 OF INBUF IS IN DRBA
1437 006076 001401              BEQ     .+4        ;IS IT?
1438 006100 104000              HLT              ;BAD DATA IN DRDB
1439
1440 ;*****
1441 ;TEST 57 STRING OF 200 DATO'S NON-BURST MODE
1442 ;*****
1443 006102 104400      SCOPE
1444 006104 012767 000201 172734      MOV     #201,BUFLEN ;LENGTH OF BUFFER=201
1445 006112 004767 000570              JSR     %7,LODBUF   ;LOAD THE BUFFER WITH INCREMENTING PATTERN
1446 006116 005367 172724              DEC     BUFLN       ;BUFLN=200
1447 006122 016767 172720 172724      MOV     BUFLN,WCLN  ;PREPARE NUMBER FOR DRWC
1448 006130 005467 172720      NEG     WCLN        ;2'S COMPLEMENT OF BUFLN

```

1450	006142	016777	172674	172652	MOV	INBUF,DRBA	:SET UP DRBA
1451	006150	012777	052525	172650	MOV	#52525,DRDB	:SET UP DRDB
1452	006156	012777	007010	172650	MOV	#INTA,DRINV	:INTERRUPT VECTOR TO INTA
1453	006164	016777	172642	172636	MOV	DRINL,DRVS	:INTERRUPT VECTOR TO PRIORITY DRINL
1454	006172	005077	172620		CLR	DRSW	:LET THE DR11-B INTERRUPT
1455	006176	012777	000113	172620	MOV	#113,DRST	:IE, FNCT3, FNCT1, AND DO TO DRST
1456	006204	005267	172642		INC	BRWAIT	:USE A WAIT OR BR. INSTRUCTION
1457	006210	032767	000001	172634	BIT	#BIT0,BRWAIT	:BIT 0 CLEAR=BR.
1458	006216	001403			BEQ	DATONB	

F03

MAINDEC-11-DZDRB-F MACY11 30(1046) 25-OCT-77 12:36 PAGE 31
DZDRBF.P11 25-OCT-77 12:35

SEQ 0031

G03

MAINDEC-11-DZDRB-F MACY11 30(1046) 25-OCT-77 12:36 PAGE 32
DZDRBF.P11 25-OCT-77 12:35

SEQ 0032

1461 006224 000401
1462 006226 000777
1463 006230 004767 000732
1464
1465
1466

DATONB: BR .+4
BR
JSR %7,DATOCK ;CHECK INBUF

::*****
: TEST 60 TEST THAT FUNCTION BITS INCREMENT WITH MAINT MODE TRANSFERS
:

1468	006234	104400			SCOPE		
1469	006236	012767	000010	172602	MOV	#10,BUFLEN	:SET-UP BUFLEN FOR LODBUF AND CHKBUFF
1470	006244	016777	172572	172550	MOV	INBUF,DRBA	:SET-UP DRBA
1471	006252	004767	000430		JSR	%7,LODBUF	:LOAD INBUF
1472	006256	004767	000462		JSR	%7,CHKBFF	:LOAD CHKBUFF
1473	006262	005077	172536		CLR	DRST	:INIT FOR STARTING
1474	006266	012767	000001	172574	MOV	#1,FNCCNT	:GET READY FOR CHECKING

MAINDEC-11-DZDRB-F MACY11 30(1046) 25-OCT-77 12:36 PAGE 34
DZDRBF.P11 25-OCT-77 12:35

SEQ 0034

1476 006302 016767 172534 172562 MOV INBUF,INBUF1 ;SAVE INBUF

1478	006316	016777	172510	172504	MOV	DRINL,DRVS	; INTERRUPT VECTOR PRIORITY TO DRINL
1479	006324	005077	172466		CLR	PSW	; LET THE DR11-B INTERRUPT
1480	006330	012777	177777	172462	MOV	#-1,DRWC	; SET-UP FOR 1 TRANSFER

```

1482 006344 000001      WAIT
1483 006346 000240      NOP
1484 006350 117701 172450  MOVB   %DRST,%1
1485 006354 042701 000600  BIC   %600,%1
1486 006350 006201      ASR   %1
1487 006362 126701 172502  CMPB  FNCNT,%1
1488 006366 001401      BEQ   .+4
1489 006370 104000      HLT
1490 006372 005267 172472  INC   FNCNT
1491 006376 022767 000010 172464  CMP   #10,FNCNT
1492 006404 001404      BEQ   MFCHK
1493 006406 062767 000002 172426  ADD   #2,INBUF
1494 006414 000735      BR    MFLOOP
1495 006416 012767 000007 172422 MFCHK: MOV  #7,BUFLEN
1496 006424 016767 172442 172410  MOV  INBUF1,INBUF
1497 006432 004767 000446      JSR  %7,DATCHK
1498
1499
1500      ; *****
1501      ; TEST 61 TEST FOR 10 MAINT MODE TRANSFERS
1502      ; *****
1503 006436 104400      SCOPE
1504 006440 012767 000010 172400  MOV  #10,BUFLEN
1505 006446 016767 172374 172400  MOV  BUFLEN,WLEN
1506 006454 005467 172374      NEG  WLEN
1507 006460 004767 000222      JSR  %7,LODBUF
1508 006470 004767 000254      JSR  %7,CHKBFF
1509 006476 016777 172360 172322  MOV  WLEN,%DRWC
1510 006504 012777 177777 172314  MOV  INBUF,%DRBA
1511 006512 012777 007010 172314  MOV  #-1,%DRDB
1512 006520 016777 172306 172302  MOV  #INTA,%DRINV
1513 006526 005077 172264      CLR  %PSW
1514 006532 012777 010101 172264  MOV  #10101,%DRST
1515 006540 000777      BR
1516 006542 004767 000336      JSR  %7,DATCHK
1517
1518
1519      ; *****
1520      ; TEST 62 TEST FOR 200 NPR TRANSFERS IN MAINT MODE
1521      ; *****
1522 006546 104400      SCOPE
1523 006550 012767 000200 172270  MOV  #200,BUFLEN
1524 006556 016767 172264 172270  MOV  BUFLEN,WLEN
1525 006564 005467 172264      NEG  WLEN
1526 006570 004767 000112      JSR  %7,LODBUF
1527 006574 004767 000144      JSR  %7,CHKBFF
1528 006600 016777 172250 172212  MOV  WLEN,%DRWC
1529 006606 016777 172230 172206  MOV  INBUF,%DRBA
1530 006614 012777 000001 172204  MOV  #1,%DRDB
1531 006622 012777 007010 172204  MOV  #INTA,%DRINV
1532 006630 016777 172176 172172  MOV  DRINL,%DRVS
1533 006636 005077 172154      CLR  %PSW
1534 006642 012777 010101 172154  MOV  #010101,%DRST
1535 006650 005267 172176      INC  BRWAIT
1536 006654 032767 000001 172170  BIT  #BIT0,BRWAIT
1537 006662 001403      BEQ  BRANCH

```

```

:WAIT FOR INTERRUPT
:FAKE-OUT RETURN ADDRESS CHANGING
:LOWER BYTE OF DRST TO R1
:GET RID OF READY AND CYCLE BECAUSE OF MAINT MODE
:MOVE IT RIGHT ONE PLACE
:CHECK AGAINST FNCNT
:SHOULD BE EQUAL
:FUNCTION BITS DIDN'T INCREMENT IN MAINT MODE
:GET READY FOR NEXT PASS
:ONLY 10 BECAUSE FNCT3-1 GO TO ZERO
:IF ITS EQUAL GO CHECK DATA
:FAKE-OUT INTA ROUTINE
:DO IT AGAIN
:SET UP FOR DATCHK (10 FNCT CHECKS, 7 TRANSFERS)
:RESTORE INBUF
:CHECK DATA

:*****
:TEST 61 TEST FOR 10 MAINT MODE TRANSFERS
:*****

SCOPE
:BUFLEN=10
:PREPARE NUMBER FOR DRWC
:2'S COMPLEMENT OF BUFLEN
:LOAD IN BUFFER WITH INCREMENTING PATTERN
:LOAD CHECK BUFFER WITH MODIFIED INCREMENTING PATTERN
:SET UP DRWC
:SET UP DRBA
:MAINT AIDE
:INTERRUPT VECTOR TO INTA
:INTERRUPT STATUS AT PRIORITY DRINL
:LET DR11-B INTERRUPT
:MAINT IE AND DO TO DRST
:WAIT FOR INTERRUPT

:*****
:TEST 62 TEST FOR 200 NPR TRANSFERS IN MAINT MODE
:*****

SCOPE
:LENGTH OF BUFFER = 200
:PREPARE NUMBER FOR DRWC
:2'S COMPLEMENT OF BUFLEN
:LOAD INBUF WITH INCREMENTING PATTERN
:LOAD CHKBUFF WITH MODIFIED INCREMENTED PATTERN
:SET UP DRWC
:SET UP DRBA
:MAINT AIDE
:INT VECTOR TO INTA
:INT VECTOR AT PRIORITY DRINL
:LET THE DR11-B INTERRUPT
:FOLLOWING TO DRST: MAINT(12),IE(06),00(00)
:USE A WAIT OR BR. INSTRUCTION
:SEE WHICH ONE
:BIT 0 CLEAR = BR.

```


1538	006666	000240			NOP			
1539	006670	000401			BR	.+4		
1540	006672	000777			BR	.+4		
1541	006674	004767	000204		BRANCH: JSR	%7,DATCHK		;CHECK THAT CORRECT DATA WAS TRANSFERRED
1542	006700	104400			SCOPE			
1543	006702	000167	000356		JMP	END		;DO IT ALL AGAIN.
1544	006706	016702	17213C		LODBUF: MOV	INBUF,%2		;MOVE STARTING ADDRESS OF INBUF TO R2
1545	006712	005067	172132		CLR	LENCHK		;CLEAR LENGTH CHECK
1546	006716	005022			CLR	(%2)+		;CLEAR STARTING ADDRESS OF INBUFF AND INC BY 2
1547	006720	005267	172124		LOADA: INC	LENCHK		;INC LENGTH CHECK BY 1
1548	006724	026767	172120	172114	CMP	LENCHK,BUFLEN		;CHECK FOR DONE
1549	006732	001403			BEQ	LDEXIT		;IS INBUF FILLED?
1550	006734	016722	172110		MOV	LENCHK,(%2)+		;LOAD NEXT BUFFER WORD
1551	006740	000767			BR	LOADA		;CONTINUE CHECKING
1552	006742	000207			LDEXIT: RTS	%7		;EXIT
1553	006744	016702	172074		CHKBFF: MOV	CHKBUFF,%2		;STARTING ADDRESS OF CHECK-BUFFER TO R2
1554	006750	005067	172074		CLR	LENCHK		;CLEAR LENGTH CHECK
1555	006754	005003			CLR	%3		;CLEAR R3
1556	006756	010322			CHKA: MOV	%3,(%2)+		;MOVE R3 TO CHKBUFF ADDRESS AND INC BY 2
1557	006760	010322			MOV	%3,(%2)+		;MOVE R3 TO NEXT CHKBUFF ADDRESS AND INC BY 2
1558	006762	062767	000032	172060	ADD	#2,LENCHK		;ADD 2 TO LENGTH CHECK
1559	006770	026767	172054	172050	CMP	LENCHK,BUFLEN		;CHECK FOR DONE
1560	006776	100003			BPL	.+10		;IS CHECK-BUFFER FILLED?
1561	007000	062703	000002		ADD	#2,%3		;NEXT NUMBER FOR BUFFER
1562	007004	000764			BR	CHKA		;CONTINUE FILLING
1563	007006	000207			RTS	%7		;EXIT
1564	007010	042777	000100	172006	INTA: BIC	#BIT6,%DRST		;CLEAR IE
1565	007016	005777	172002		TST	%DRST		;CHECKING FOR ERROR
1566	007022	100000			BPL	.+4		;ERROR SET?
1567	007024	104000			HLT			;ERROR BIT IS SET
1568	007026	105777	171772		TSTB	%DRST		;CHECKING READY BIT
1569	007032	100401			BMI	.+4		;IS READY SET
1570	007034	104000			HLT			;FALSE INTERRUPT - ERROR AND READY ARE CLEAR
1571	007036	005777	171756		TST	%DRWC		;TEST1 FOR DRWC=0
1572	007042	001401			BEQ	.+4		;WAS IT EQUAL?
1573	007044	104000			HLT			;DRWC NOT =0
1574	007046	016702	171774		MOV	BUFLEN,%2		;BUFFER LENGTH TO R2
1575	007052	066702	171770		ADD	BUFLEN,%2		;NUMBER OF TRANSFERS TIMES 2
1576	007056	066702	171760		ADD	INBUF,%2		;CORRECT DRBA
1577	007062	027702	171734		CMP	%DRBA,%2		;CHECKING DRBA
1578	007066	001401			BEQ	.+4		;IS DRBA CORRECT?
1579	007070	104000			HLT			;DRBA NOT CORRECT
1580	007072	062716	000002		ADD	#2,(%6)		;RETURN ADDRESS TO RETURN ADDRESS +2
1581	007076	004767	000042		JSR	%7,NORMAL		
1582	007102	000002			RTI			;EXIT
1583								
1584	007104	016702	171734		DATCHK: MOV	CHKBUFF,%2		;STARTING ADDRESS OF CHECK BUFFER TO R2
1585	007110	016703	171726		MOV	INBUF,%3		;STARTING ADDRESS OF IN BUFFER TO R3
1586	007114	005067	171730		CLR	LENCHK		;CLEAR LENGTH CHECK
1587	007120	005267	171724		COMPAR: INC	LENCHK		;MAKE A COMPARISON
1588	007124	022223			CMP	(%2)+,(%3)+		;IS THE DATA CORRECT?
1589	007126	001401			BEQ	.+4		;BRANCH IF OK
1590	007130	104000			HLT			;BAD DATA
1591	007132	026767	171712	171706	CMP	LENCHK,BUFLEN		;SEE IF THE BUFFER HAS BEEN CHECKED
1592	007140	001367			BNE	COMPAR		;BUFFER CHECKED?

```

1594 007144 012777 001030 171662 NORMAL: MOV #DRVS,DRINV ;RESTORE DR11-B INTERRUPT VECTOR
1595 007152 005077 171652 CLR DRVS ;RESTORE DR11-B INTERRUPT STATUS
1596 007156 012777 000340 171632 MOV #340,DRSW ;RESTORE PROC TO PRIORITY LEVEL 7
1597 007164 000207 RTS ;EXIT
1598 007166 012702 052525 DATOCK: MOV #52525,%2 ;DATO NUMBER TO R2
1599 007172 016703 171644 MOV INBUF,%3 ;STARTING ADDRESS OF IN BUFFER TO R3
1600 007176 005067 171646 CLR LENCHK ;CLEAR LENGTH CHECK
1601 007202 005267 171642 COMPRR: INC LENCHK ;MAKE A COMPARISON
1602 007206 020223 CMP %2,(%3)+ ;IS THE DATA CORRECT?
1603 007210 001401 BEQ .+4 ;BRANCH IF OK
1604 007212 104000 HLT ;BAD DATA
1605 007214 026767 171630 171624 CMP LENCHK,BUFLEN ;SEE IF THE BUFFER HAS BEEN CHECKED
1606 007222 001367 BNE COMPRR ;BUFFER CHECKED?
1607 007224 020223 CMP %2,(%3)+ ;CHECK END OF BUFFER + 1
1608 007226 001001 BNE .+4 ;SEE IF TOO MANY WORDS WERE TRANSFERRED
1609 007230 104000 HLT ;TOO MANY
1610 007232 000207 RTS ;EXIT
1611 007234 042777 000100 171562 ERRCHK: BIC #BIT6,DRST ;CLEAR IE
1612 007242 005777 171556 TST DRST ;CHECKING FOR ERROR
1613 007246 100001 BPL .+4 ;ERROR SET?
1614 007250 104000 HLT ;ERROR BIT IS SET
1615 007252 105777 171546 TSTB DRST ;CHECKING READY BIT
1616 007256 100401 BMI .+4 ;IS RDY SET
1617 007260 104000 HLT ;FALSE ENTRY - ERROR AND READY ARE CLEAR
1618 007262 000207 RTS ;EXIT
1619
1620 ;*****
1621 ; END OF PASS
1622 ;*****
1623 007264 012737 000207 177566 END: MOV #207,DRSW ;RING BELL
1624 007272 105737 177564 TSTB DRSW
1625 007276 100375 BPL .-4
1626 007300 005267 171570 INC PASCNT ;KEEP TRACK OF PASSES COMPLETED
1627 007304 012702 012677 MOV #SENPAS,%2 ;PRINT 'END PASS'
1628 007310 004767 003460 JSR %7,TTOUT
1629
1630
1631 007314 042777 000020 171474 END1: BIC #20,DRSW ;CLEAR T-BIT
1632 007322 013702 000042 MOV #42,%2
1633 007326 001405 BEQ TRTRAP
1634 007330 000005 RESET
1635 007332 004712 SENDAD: JSR %7,(2)
1636 007334 000240 NOP
1637 007336 000240 NOP
1638 007340 000240 NOP
1639
1640 ;*****
1641 ; ROUTINE TO CHECK FOR TRACE TRAP TO BE RUN WITH PROGRAM
1642 ;*****
1643 007342 004767 002716 TRTRAP: JSR %7,CKSWR ;CHECK FOR CONT G
1644 007346 032777 010000 171440 BIT #10000,JSR ;SHOULD WE RUN WITH TRACE TRAP
1645 007354 001417 BEQ YESTR ;YES
1646 007356 005767 000104 TST YESTR1 ;NO, HAVE WE RUN WITH TRACE TRAP ON?
1647 007362 001411 BEQ TRPA ;IF SO RESTORE PREVIOUS CONTENTS
1648 007364 016767 000076 170422 MOV YESTR1,14

```

```

1650 007400 042777 000020 171410      BIC      #20,@PSW      ;CLEAR TRACE TRAP
1651 007406 000167 171562      TRPA:    JMP      BEGIN      ;START OF TEST WITH TRACE OFF
1652 007412 000000                      TRPB:    0
1653
1654      ;*****
1655      ;      SAVE OLD CONTENTS, SET UP FOR TRACE TRAP
1656      ;*****
1657 007414 016767 170374 000044  YESTR:  MOV      14,YESTR1      ;SAVE ODT PC
1658 007422 016767 170370 000040      MOV      16,YESTR2      ;SAVE ODT STATUS
1659 007430 012767 007472 170356      MOV      #YESTRT,14      ;NEW TRAP VECTOR
1660 007436 005067 170354                      CLR      16              ;NEW CONDITION CODES
1661 007442 005077 171350                      CLR      @PSW
1662 007446 005167 177740                      COM      TRPB
1663 007452 100403                      BMI      .+10
1664 007454 052777 000020 171334      BIS      #20,@PSW      ;SET TRACE TRAP
1665 007462 000167 171506      JMP      BEGIN      ;START OF TEST WITH TRACE ON
1666
1667 007466 000000      YESTR1: 0              ;STORAGE FOR ODT PC
1668 007470 000000      YESTR2: 0              ;STORAGE FOR ODT STATUS
1669 007472 000002      YESRT:  RTI              ;RETURN TO PROGRAM FROM TRAP
1670 007474 000000                      HALT                    ;RTI FAILED
1671
1672
1673      ;*****
1674      ;      BUS TO BUS TEST (DR11-B TO DR11-B)
1675      ;*****
1676
1677
1678      000000      R0=%0
1679      000001      R1=%1
1680      000002      R2=%2
1681      000003      R3=%3
1682      000004      R4=%4
1683      000005      R5=%5
1684      000006      R6=%6
1685      000007      R7=%7
1686      000007      PC=%7
1687      000006      SP=%6
1688
1689      000001      GO=1
1690      000002      FNCT1=2          ;OUTPUT MODE
1691      000004      FNCT2=4          ;OUTPUT DIRECTION
1692      000010      FNCT3=10         ;OUTPUT INTER REQ
1693      000020      XBA16=20
1694      000040      XBA17=40
1695      000100      IE=100
1696      000200      READY=200
1697      000400      CYCLE=400
1698      001000      DSTATA=1000      ;INPUT MODE
1699      002000      DSTATB=2000      ;INPUT DIRECTION
1700      004000      DSTATC=4000      ;INPUT INTR REQ
1701      010000      MAINT=10000
1702      020000      ATTN=20000
1703      040000      NEX=40000
1704      100000      ERROR=100000

```

```

1706
1707
1708 007476 000004
1709 007500 000000
1710 007502 000000
1711 007504 000000
1712 007506 000000
1713
1714
1715
1716 007510 000240
1717 007512 005067 177770
1718 007516 012706 014100
1719 007522 004767 000256
1720 007526 005067 177752
1721 007532 012767 010164 177740
1722 007540 012767 000000 177734
1723 007546 012777 000340 171242
1724 007554 012777 000100 171242
1725 007562 005077 171230
1726 007566 000425
1727
1728
1729
1730
1731
1732
1733 007570 000240
1734 007572 012706 014100
1735 007576 004767 000202
1736 007602 005067 177676
1737 007606 012767 010406 177664
1738 007614 012767 010424 177660
1739 007622 012777 000340 171166
1740 007630 012777 000100 171166
1741 007636 005077 171154
1742
1743
1744
1745
1746
1747 007642 005200
1748 007644 005201
1749 007646 005202
1750 007650 005203
1751 007652 005204
1752 007654 005205
1753 007656 020005
1754 007660 001402
1755 007662 104000
1756 007664 000000
1757 007666 020104
1758 007670 001402
1759 007672 104000
1760 007674 000000

NWRDXF: 4 ; # OF WORDS TRANSFERRED UNDER FLAG CONTROL PRIOR TO NPR
NEXJOB: 0 ; HOLDS ADDRESS OF READY INTERRUPT ROUTINE
; HOLDS ADDRESS OF ERROR INTERRUPT ROUTINE
JBFLAG: 0 ; JOB FLAG
JBCNT: 0 ; JOB COUNT
; *****
; MASTER START
; *****
MS1: NOP
CLR JBCNT ; CLEAR JOB COUNT
MOV #BUFF,R6 ; SETUP STACK
JSR R7,SETVEC ; SET UP INTERRUPT VECTORS
CLR JBFLAG ; CLEAR JOB FLAG
MOV #JOB00,NEXJOB ; DO JOB00 FIRST
MOV #0,NEXJOB+2 ; NO ERROR RECOVERY
MOV #340,CPSW ; LOCK OUT INTERRUPTS
MOV #IE,ORST ; SET INTERRUPT ENABLE
CLR CPSW ; DROP PRIORITY TO ZERO
BR BACKGD ; WAIT FOR JOBS IN BACKGROUND

; *****
; SLAVE START
; *****
SS1: NOP
MOV #BUFF,R6
JSR R7,SETVEC ; SET UP INTERRUPT VECTORS
CLR JBFLAG
MOV #SJOB1,NEXJOB ; FOR READY INTERRUPT
MOV #SJOB2,NEXJOB+2 ; FOR ERROR INTERRUPT
MOV #340,CPSW ; RAISE CP PRIORITY TO 7
MOV #IE,ORST ; SET INTERRUPT ENABLE
CLR CPSW ; DROP CP PRIORITY TO 0 AND ENTER BACKGROUND

; *****
; BACKGROUND PROGRAM; WAITS FOR JBFLAG TO SET
; *****
BACKGD: INC R0
INC R1
INC R2
INC R3
INC R4
INC R5
CMP R0,R5
BEQ .+6
HLT ; BACKGROUND TEST FAILED
CMP R1,R4
BEQ .+6
HLT ; BACKGROUND TEST FAILED

```

```

1762 007700 001402 BEQ .+6
1763 007702 104000 HLT
1764 007704 000000 HALT
1765 007706 005767 177572 TST JBFLAG ;BACKGROUND TEST FAILED
1766 007712 001753 BEQ BACKGD ;ANY JOBS?
1767 007714 004567 000032 JSR RS,SAVALL ;BRANCH IF NONE
1768 007720 005067 177554 CLR NEXJOB ;YES & EXECUTE JOB WHOSE ADDRESS IS IN JBFLAG
1769 007724 005067 177552 CLR NEXJOB+2
1770 007730 016767 177550 170076 MOV JBFLAG,34
1771 007736 005067 177542 CLR JBFLAG
1772 007742 104400 TRAP ;TRAP THROUGH JBFLAG AT 34
1773 007744 004567 000016 JSR RS,RESALL
1774 007750 000734 BR BACKGD
1775
1776 ;*****
1777 ; SUBROUTINE TO PUSH ALL REGISTERS ONTO THE STACK
1778 ;*****
1779
1780 007752 010446 SAVALL: MOV R4,-(R6) ;RS WAS PUSHED BY JSR
1781 007754 010346 MOV R3,-(R6)
1782 007756 010246 MOV R2,-(R6)
1783 007760 010146 MOV R1,-(R6)
1784 007762 010046 MOV R0,-(R6)
1785 007764 000115 JMP (R5) ;RS HOLDS RETURN ADDRESS
1786
1787 ;*****
1788 ; SUBROUTINE TO POP ALL REGISTERS OFF THE STACK
1789 ;*****
1790
1791 007766 005726 RESALL: TST (R6)+
1792 007770 012600 MOV (R6)+,R0
1793 007772 012601 MOV (R6)+,R1
1794 007774 012602 MOV (R6)+,R2
1795 007776 012603 MOV (R6)+,R3
1796 010000 012604 MOV (R6)+,R4
1797 010002 000205 RTS R5
1798
1799 ;*****
1800 ; ROUTINE TO SET UP INTERRUPT VECTORS
1801 ;*****
1802
1803 010004 016700 171024 SETVEC: MOV DRINV,R0 ;R0 IS VECTOR ADDRESS
1804 010010 012720 010064 MOV DRINS,(R0)+ ;PUT SERVICE ADDRESS INTO VECTOR
1805 010014 016710 171012 MOV DRINL,(R0) ;PUT PRIORITY INTO VECTOR+2
1806 010020 012767 011366 170002 MOV @PRINT,30 ;SET UP EMT ADDRESS
1807 010026 016767 171000 167776 MOV DRINL,32 ;SET UP EMT PRIORITY LEVEL
1808 010034 005067 167774 CLR 34
1809 010040 016767 170766 167770 MOV DRINL,36 ;SET UP TRAP ADDRESS
1810 010046 005000 CLR R0 ;INITIALIZE REGISTERS
1811 010050 005001 CLR R1
1812 010052 005002 CLR R2
1813 010054 005003 CLR R3
1814 010056 005004 CLR R4
1815 010060 005005 CLR R5
1816 010062 000207 RTS R7

```

```
1818 : PRIMARY INTERRUPT SERVICE ROUTINE.  
1819 : SETS UP JBLFAG WITH ADDRESS OF JOB TO BE RUNSTARS  
1820 : *****  
1821 :  
1822 010064 005767 177414 DRINS: TST JBLFAG ;HAS THE PREVIOUS INTERRUPT BEEN SERVICED?  
1823 010070 001402 BEQ DRINO  
1824 010072 104000 HLT  
1825 010074 000000 HALT ;NO  
1826 010076 032777 004000 170720 DRINO: BIT #DSTATC,DRST ;CHECK FOR ERROR  
1827 010104 001411 BEQ DRIN3 ;BRANCH IF NO ERROR  
1828 010106 005767 177370 TST NEXJOB+2 ;IS THERE AN ERROR SERVICE ROUTINE?  
1829 010112 001002 BNE DRIN1 ;BRANCH IF THERE IS.  
1830 010114 104000 HLT  
1831 010116 000000 HALT ;ERROR INTERRUPT, NO ERROR SERVICE.  
1832 010120 016767 177356 177356 DRIN1: MOV NEXJOB+2,JBLFAG ;SET UP JOBFLAG WITH ADDRESS OF SERVICE ROUTINE  
1833 010126 000002 RTI  
1834 010130 105777 170670 DRIN3: TSTB DRST ;CHECK READY  
1835 010134 100402 BMI DRIN2 ;BRANCH IF SET  
1836 010136 104000 HLT  
1837 010140 000000 HALT ; INTERRUPT WITHOUT ERROR OR READY  
1838 010142 005767 177332 DRIN2: TST NEXJOB ;IS THERE A READY SERVICE ROUTINE  
1839 010146 001002 BNE .+6 ;BRANCH IF THERE IS.  
1840 010150 104000 HLT  
1841 010152 000000 HALT ;READY INTERRUPT, NO READY SERVICE  
1842 010154 016767 177320 177322 MOV NEXJOB,JBLFAG ;SET UP JOBFLAG WITH SERVICE ROUTINE ADDRESS  
1843 010162 000002 RTI  
1844 :  
1845 : *****  
1846 : MASTER'S INTERRUPT SERVICE ROUTINES  
1847 : ROUTINE A, SEGMENT 0  
1848 : FILL BUFFER AND TRANSMIT  
1849 : *****  
1850 010164 012700 011330 JOBA0: MOV #LISTA,RO ;RO IS XMIT LIST ADDRESS  
1851 010170 012701 011342 MOV #LISTA,R1 ;LISTA1 WILL BE REC LIST  
1852 010174 012021 MOV (RO)+,(R1)+ ;START WITH BUS ADDRESSES EQUAL  
1853 010176 012002 MOV (RO)+,R2 ;R2 HOLDS WORD COUNT OF XMIT  
1854 010200 010211 MOV R2,(R1) ;MAKE REC WORD COUNT THE SAME  
1855 010202 005402 NEG R2 ;MAKE WORD COUNT POSITIVE  
1856 010204 006302 ASL R2 ;TRANSFORM INTO BYTE COUNT  
1857 010206 060241 ADD R2,-(R1) ;ADD TO REC BUS ADDRESS  
1858 010210 005010 CLR (R0) ;CLEAR OFFSET IN XMIT LIST  
1859 010212 024040 CMP -(R0),-(R0) ;LEAVE R0=LISTA=XMIT LIST  
1860 :  
1861 010214 022767 000100 177264 CMP #100,JBCNT ;ENOUGH PASSES FOR BELL?  
1862 010222 003010 BGT JOBA0A ;BRANCH IF NOT ENOUGH  
1863 010224 105737 177564 JOBA0B: TSTB #177564 ;TTY READY?  
1864 010230 100375 BPL JOBA0B  
1865 010232 012737 000207 177566 MOV #207,#177566 ;RING BELL  
1866 010240 005067 177242 CLR JBCNT ;RESET JOB COUNT  
1867 010244 004767 000704 JOBA0A: JSR R7,SETBUF ;FILL UP XMIT BUFFER WITH SPECIAL BINARY COUNT  
1868 010250 012700 011330 MOV #LISTA,RO  
1869 010254 004767 000370 JSR R7,MXMIT ;TRANSMIT DATA TO SLAVE  
1870 010260 012767 010276 177212 MOV #JOBA1,NEXJOB ;JOBA1 IS NEXT  
1871 010266 012767 000000 177206 MOV #0,NEXJOB+2 ;NO ERROR RECOVERY  
1872 010274 000002 RTI ;RETURN TO BACKGROUND VIA TRAP
```

```

1874
1875
1876
1877
1878
1879 010276 012700 011342
1880 010302 004767 000574
1881 010306 012700 011342
1882 010312 004767 000406
1883 010316 012767 010334 177154
1884 010324 012767 000000 177150
1885 010332 000002
1886
1887
1888
1889
1890
1891
1892 010334 012700 011330
1893 010340 012701 011342
1894 010344 004767 000652
1895 010350 042777 000100 170446
1896 010356 012777 000100 170440
1897 010364 012767 010164 177106
1898 010372 012767 000000 177102
1899 010400 005607 177102
1900 010404 000002
1901
1902
1903
1904
1905
1906
1907
1908 010406 012767 000000 177064
1909 010414 012767 010424 177060
1910 010422 000002
1911
1912
1913
1914
1915
1916 010424 032777 004000 170372
1917 010432 001002
1918 010434 104000
1919 010436 000000
1920 010440 005001
1921 010442 016702 170356
1922 010446 012703 000010
1923 010452 012704 004000
1924 010456 016705 170344
1925 010462 012700 011356
1926 010466 004767 000122
1927 010472 012700 011354
1928 010476 016010 000004

;*****
; ROUTINE A, SEGMENT 1
; FLUSH A BUFFER AND RECEIVE DATA
;*****
JOB01: MOV #LISTA1,RO ;PUT REC LIST ADDRESS INTO RO
        JSR R7,FLUSH ;FLUSH BUFFER
        MOV #LISTA1,RO
        JSR R7,MREC ;RECEIVE DATA FROM SLAVE
        MOV #JOB02,NEXJOB ;JOB02 IS NEXT
        MOV #0,NEXJOB+2 ;NO ERROR RECOVERY
        RTI

;*****
; ROUTINE A, SEGMENT 2
; CHECKS TRANSMITTED DATA WITH RECEIVED
;*****
JOB02: MOV #LISTA,RO ;XMIT BUFFER LIST
        MOV #LISTA1,R1 ;REC BUFFER LIST
        JSR R7,BUFCHK ;COMPARE THE TWO BUFFERS
        BIC #IE,DRST ;GLITCH INTERRUPT
        MOV #IE,DRST
        MOV #JOB00,NEXJOB ;REPEAT JOB00
        MOV #0,NEXJOB+2
        INC JBCNT ;ADVANCE COUNT
        RTI

;*****
; SLAVE'S INTERRUPT SERVICE ROUTINES
;*****
JOB01: IGNORE FIRST READY INTERRUPT
;*****
SJ01: MOV #0,NEXJOB ;NO MORE READY INTERRUPTS
      MOV #SJ02,NEXJOB+2 ;UNTIL ATTN INTERRUPT
      RTI

;*****
;*****
;*****
JOB02: WAIT FOR COMMAND
;*****
;*****
SJ02: BIT #DSTATC,DRST ;TEST FOR INTER
      BNE SJ02A
      HLT
      SJ02A: CLR R1 ;ERROR OTHER THAN DSTATC
            MOV DRST,R2 ;SET UP FOR PARAMETERS
            MOV #FNCT3,R3 ;R2 IS STATUS ADDRESS
            MOV #DSTATC,R4 ;R3 IS FUNCTION BIT 3
            MOV DR0B,R5 ;R4 IS INTERRUPT BIT
            MOV #LISTB+2,RO ;R5 IS DATA BUFFER ADDRESS
            JSR R7,HNDSHK ;STORE PARAMETERS HERE STARTING WITH WORD COUNT
            MOV #LISTB,RO ;GET PARAMETERS
            MOV 4(RO),(RO) ;RO IS TOP OF LIST
            ;MOVE OFFSET TO TOP

```



```

1930 010506 012077 170310      MOV      (R0)+, @DRBA      ; SET UP BUS ADDRESS
1931 010512 012077 170302      MOV      (R0)+, @DRWC      ; SET UP WORD COUNT
1932 010516 005077 170302      CLR      @DRST            ; CLEAR ALL FUNCTION BITS
1933 010522 032777 002000 170274  BIT      @DSTATB, @DRST    ; WHICH DIRECTION
1934 010530 001405                BEQ      SJOB2C            ; BRANCH IF RECIEVE (LEAVE FNCT1 CLEAR FOR DATI'S)
1935 010532 032777 000400 170264  SJOB2B: BIT      #CYCLE, @DRST    ; WAIT FOR MASTER TO SET CYCLE
1936 010540 001774                BEQ      SJOB2B            ; BRANCH IF NOT SET
1937 010542 000412                BR       SJOB2D            ; GO DO THE COMAND
1938 010544 012700 011354  SJOB2C: MOV      #LISTB, R0
1939 010550 004767 000356                JSR      R7, BLUSH         ; BLUSH THE BUFFER
1940 010554 052777 000604 170242  BIS      #FNCT2, @DRST    ; SET FNCT2 FOR DATO'S
1941 010562 042777 000400 170234  BIC      #CYCLE, @DRST    ; CLEAR CYCLE
1942 010570 052777 000101 170226  SJOB2D: BIS      #IE!GO, @DRST ; EXECUTE COMMAND AND INTERRUPT WHEN DONE
1943 010576 012767 010406 176674  MOV      #SJOB1, NEXJOB    ; IGNORE READY INTERRUPT
1944 010604 012767 010424 176670  MOV      #SJOB2, NEXJOB+2 ; WAIT FOR ATTN INTERRUPT
1945 010612 000002                RTI
1946
1947
1948
1949
1950
1951 010614 011515                HNSH#K: MOV      (R5), (R5)    ; ECHO PARAMETER
1952 010616 011520                MOV      (R5), (R0)+      ; STORE PARAMETER
1953 010620 050312                BIS      R3, (R2)         ; REPLY WITH FNCT3
1954 010622 030412  HNSH1: BIT      R4, (R2)         ; WAIT FOR ATTN TO DROP
1955 010624 001376                BNE     HNSH1
1956 010626 040312                BIC      R3, (R2)         ; DROP FNCT3
1957 010630 005201                INC      R1                ; CHECK NUMBER
1958 010632 020167 176640                CMP      R1, NWRDXF
1959 010636 002401                BLT     HNSH2              ; BRANCH IF NOT DONE YET
1960 010640 000207                RTS
1961 010642 030412  HNSH2: BIT      R4, (R2)         ; WAIT FOR NEXT WORD
1962 010644 001776                BEQ     HNSH2              ; BRANCH IF ATTN CLEAR
1963 010646 000762                BR      HNSH#K            ; GET ANOTHER PARAMETER
1964
1965
1966
1967
1968
1969
1970 010650 005777 170150  MXMIT:  TST      @DRST            ; MAKE SURE ERROR IS CLEAR
1971 010654 100002                BPL     MXMIT1            ; AND READY IS SET
1972 010656 104000                HLT
1973 010660 000000                HALT                        ; ERROR IS SET
1974 010662 105777 170136  MXMIT1: TSTB     @DRST
1975 010666 100402                BMI     MXMIT2
1976 010670 104000                HLT
1977 010672 000000                HALT                        ; READY NOT SET
1978 010674 012777 000000 170122  MXMIT2: MOV      #0, @DRST    ; SET UP FUNCTION FOR DATI'S
1979 010702 004767 000102                JSR      R7, PRMXFR       ; TRANSFER PARAMETERS OF LIST WHOSE ADDRESS IS IN R0
1980 010706 042777 000400 170110  BIC      #CYCLE, @DRST    ; MAKE SURE CYCLE IS CLEAR
1981 010714 052777 000101 170102  BIS      #IE!GO, @DRST    ; EXECUTE COMMAND AND INTERRUPT WHEN DONE
1982 010722 000207                RTS      R7
1983
1984

```

```

1986 ; ENTER WITH TRANSFER LIST IN RO
1987 ; *****
1988 010724 005777 170074 MREC: TST @DRST ; MAKE SURE ERROR IS CLEAR
1989 010730 100002 BPL MREC1 ; AND READY SET.
1990 010732 104000 HLT ;
1991 010734 000000 HALT ; ERROR SET
1992 010736 105777 170062 MREC1: TSTB @DRST
1993 010742 100402 BMI MREC2
1994 010744 104000 HLT
1995 010746 000000 HALT ; READY CLEAR
1996 010750 012777 000004 170046 MREC2: MOV #FNCT2,@DRST ; SET UP FUNCTION FOR DATO'S
1997 010756 004767 000026 JSR R7,PRMXFR ; TRANSFER PARAMETERS OF LIST WHOSE ADDRESS IS IN RO
1998 010762 032777 002000 170034 MREC3: BIT #DSTATB,@DRST ; WAIT FOR SLAVE TO CLEAR DIRECTION
1999 010770 001374 BNE MREC3 ; BRANCH IF SET
2000 010772 042777 000400 170024 BIC #CYCLE,@DRST ; CLEAR CYCLE
2001 011000 052777 000101 170016 BIS #IE!GO,@DRST ; EXECUTE COMMAND AND INTERRUPT WHEN DONE.
2002 011006 000207 RTS R7
2003
2004 ; *****
2005 ; ROUTINE TO TRANSFER AND CHECK PARAMETERS UNDER FLAG CONTROL
2006 ; ENTER WITH RO POINTING TO TRANSFER LIST
2007 ; *****
2008
2009 011010 012077 170006 PRMXFR: MOV (RO)+,@DRBA ; FIRST WORD IN LIST IS ADDRESS
2010 011014 011077 170000 MOV (RO),@DRWC ; SECOND WORD IN LIST IS WORD COUNT
2011 011020 005001 CLR R1 ; R1 COUNTS PARAMETERS TRANSFERRED
2012 011022 016702 167776 MOV @DRST,R2 ; R2 IS THE STATUS ADDRESS
2013 011026 012703 000010 MOV #FNCT3,R3 ; R3 USED FOR FUNCTION BIT 3
2014 011032 012704 004000 MOV #DSTATC,R4 ; R4 USED FOR INTERRUPT BIT
2015 011036 016705 167764 MOV @DRDB,R5 ; R5 IS THE DATA BUFFER ADDRESS
2016 011042 011015 PRMXF1: MOV (RO),(R5) ; SET UP @DRDB WITH PARAMETER
2017 011044 050312 BIS R3,(R2) ; CALL SLAVE'S ATTN
2018 011046 030412 PRMXF2: BIT R4,(R2) ; WAIT FOR REPLY
2019 011050 001776 BEQ PRMXF2 ; BRANCH IF ATTN CLEAR
2020 011052 022015 CMP (RO)+,(R5) ; COMPARE PARAMETER SENT WITH SLAVE'S ECHO
2021 011054 001402 BEQ PRMXF3 ; BRANCH IF EQUAL
2022 011056 104000 HLT
2023 011060 000000 HALT ; PARAMETER DID NOT ECHO
2024 011062 040312 PRMXF3: BIC R3,(R2) ; DROP SLAVE'S ATTN
2025 011064 030412 PRMXF4: BIT R4,(R2) ; WAIT FOR REPLY
2026 011066 001376 BNE PRMXF4 ; BRANCH IF ATTN SET
2027 011070 005201 INC R1 ; ADVANCE PARAMETER COUNT
2028 011072 020167 176400 CMP R1,NWRDXF ; ALL PARAMETER XFERRED?
2029 011076 002761 BLT PRMXF1 ; BRANCH IF NOT DONE
2030 011100 000207 RTS R7 ; RETURN WHEN ALL PARAMETERS TRANSFERRED AND CHECKED.
2031
2032 ; *****
2033 ; ROUTINE TO CLEAR BUFFER
2034 ; ENTER WITH RO POINTING TO TRANSFER LIST
2035 ; *****
2036
2037 011102 005005 FLUSH: CLR R5 ; SET R5 TO ZIP
2038 011104 004767 000030 JSR R7,BSETUP ; SET UP REGISTERS
2039 011110 004767 000004 FLUSH1: JSR R7,BUFPUT ; STORE ITEM IN BUFFER
2040 011114 002775 BLT FLUSH1

```

```

2042
2043 011120 010521          BUFPUT: MOV    R5,(R1)+      ;PUT R5 INTO BUFFER
2044 011122 060503          ADD     R5,R3              ;INCLUDE IN CHECKSUM
2045 011124 005504          ADC     R4                 ;
2046 011126 005202          INC     R2                 ;ADVANCE WORD COUNT
2047 011130 000207          RTS      R7                ;RETURN WITH STATUS SET
2048
2049
2050 ;*****
2051 ;ROUTINE TO FILL BUFFER WITH ALL ONE'S
2052 ;ENTER WITH R0 POINTING TO TRANSFER LIST
2053 ;*****
2054 011132 012705 177777    FLUSH:  MOV    #-1,R5        ;SET R5 TO ALL ONE'S
2055 011136 000762          BR     FLUSH+2
2056
2057 ;REGISTER SETUP ROUTINE
2058 BSETUP: MOV    (R0)+,R1      ;R1 IS BUS ADDRESS
2059 011142 012002          MOV    (R0)+,R2          ;R2 IS WORD COUNT
2060 011144 005720          TST   (R0)+             ;SKIP OVER OFFSET
2061 011146 005003          CLR   R3                ;CLEAR LOW CHECKSUM
2062 011150 005004          CLR   R4                ;CLEAR HIGH CHECKSUM
2063 011152 000207          RTS      R7                ;RETURN
2064
2065 ;*****
2066 ;ROUTINE TO FILL BUFFER WITH FLOATING 1'S AND FLOATING 0'S PATTERN
2067 ;ENTER WITH R0 POINTING TO TRANSFER LIST
2068 ;*****
2069 011154 004767 177760    SETBUF: JSR    R7,BSETUP    ;SET UP REGISTERS
2070 011160 012705 000001    SETBF1: MOV    #1,R5        ;SET UP R5 WITH START PATTERN
2071 011164 004767 177730    SETBF2: JSR    R7,BUFPUT    ;NUMBER TO BUFFER
2072 011170 002011          BGE   BUFOUT
2073 011172 005105          COM   R5
2074 011174 004767 177720    JSR    R7,BUFPUT          ;COMPLEMENT OF NUMBER TO BUFFER
2075 011200 002005          BGE   BUFOUT
2076 011202 005105          COM   R5
2077 011204 000241          CLC
2078 011206 006105          ROL   R5
2079 011210 103763          BCS   SETBF1             ;IF CARRY SET, START PATTERN OVER
2080 011212 000764          BR    SETBF2
2081 011214 010320    BUFOUT: MOV    R3,(R0)+      ;MOVE LOW CHECK TO LIST
2082 011216 010420          MOV    R4,(R0)+          ;MOVE HIGH CHECK TO LIST
2083 011220 000207          RTS      R7                ;RETURN
2084
2085 ;*****
2086 ;ROUTINE TO COMPARE TWO BUFFERS
2087 ;BUFFER LIST #1 IS IN R1
2088 ;BUFFER LIST #2 IS IN R0. #2 CHECKED FOR PROPER CHECKSUM
2089 ;*****
2090
2091 011222 010046          BUFCHK: MOV    R0,-(R6)     ;SAVE LIST#2 ON STACK
2092 011224 010146          MOV    R1,-(R6)         ;SAVE LIST#1 ON STACK
2093 011226 004767 000036    JSR    R7,CHKSUM         ;VERIFY INTEGRITY OF CHECK BUFFER (#2)
2094 011232 012600          MOV    (R6)+,R0         ;RECOVER 1ST LIST POINTER
2095 011234 012001          MOV    (R0)+,R1         ;BUS ADDRESS #1
2096 011236 012002          MOV    (R0)+,R2         ;WORD COUNT #1

```

```

R098 011242 012003          MOV      (R0)+,R3          ;BUS ADDRESS #2
R099 011244 020220          CMP      R2,(R0)+        ;COMPARE WORD COUNTS
R100 011246 003402          BLE     BUFCK1          ;BRANCH IF EQUAL OR LESS THAN
R101 011250 104000          HLT                    ;BUFFER 2 IS LONGER THAN 1
R102 011252 000000          HALT
R103
R104 011254 022123          BUFCK1: CMP      (R1)+,(R3)+ ;COMPARE BUFFERS
R105 011256 001401          BEQ     BUFCK2          ;
R106 011260 104000          HLT                    ;DATA ERROR
R107 011262 005202          BUFCK2: INC     R2        ;ADVANCE COUNT
R108 011264 002773          BLT     BUFCK1          ;BRANCH IF NOT DONE
R109 011266 000207          RTS      R7             ;RETURN; STACK IS CLEAR
R110
R111
R112
R113
R114
R115 011270 004767 177644    CHKSUM: JSR     R7,BSETUP ;SET UP REGISTERS
R116 011274 004767 000016    CHKSM1: JSR     R7,GETBUF ;GET ITEM FROM BUFFER
R117 011300 002775          BLT     CHKSM1          ;BRANCH IF NOT DONE
R118 011302 0203c3          CMP     R3,(R0)+        ;COMPARE LOW ORDER CHECKS
R119 011304 001003          BNE     CHKSM2          ;
R120 011306 020420          CMP     R4,(R0)+        ;COMPARE HIGH ORDER CHECKS
R121 011310 001001          BNE     CHKSM2          ;
R122 011312 000207          RTS      R7             ;RETURN IF CHECKSUM OK.
R123 011314 104000          CHKSM2: HLT                    ;ORIGINAL BUFFER CHECKSUM DOES NOT AGREE WITH PRESENT
R124
R125 011316 012105          GETBUF: MOV     (R1)+,R5 ;GET ITEM OUT OF BUFFER
R126 011320 060503          ADD     R5,R3           ;ADD TO CHECKSUM
R127 011322 005504          ADC     R4              ;
R128 011324 005202          INC     R2             ;ADVANCE WORD COUNT
R129 011326 000207          RTS      R7
R130
R131 011330 014102          LISTA: XINBUF          ;START OF XMIT BUFFER
R132 011332 177605          -123.                 ;WORD COUNT
R133 011334 000000          0                     ;OFFSET
R134 011336 000000          0                     ;CHECKSUM LOW
R135 011340 000000          0                     ;CHECKSUM HIGH
R136
R137 011342 000000          LISTA1: 0              ;START OF REC BUFFER
R138 011344 000000          0                     ;WORD COUNT
R139 011346 000000          0                     ;OFFSET
R140 011350 000000          0                     ;CHECKSUM LOW
R141 011352 000000          0                     ;CHECKSUM HIGH
R142
R143 011354 014102          LISTB: XINBUF          ;SLAVE'S ECHO BUFFER
R144 011356 000000          0
R145 011360 000000          0
R146 011362 000000          0
R147 011364 000000          0
R148
R149
R150
R151
R152
;*****
; ENTERED WITH SYSTEM TRAP CALL(HLT)
; PRINT OUT THE ERROR PC AND STATUS REGISTER
;*****

```

```

2154
2155
2156 011366 004767 000672 PRINT: JSR %7,CKSWR
2157 011372 037727 167416 020000 BIT JSR,#20000 ;TEST FOR INHIBIT PRINT OUT
2158 011400 001067 BNE 1$ ;IF SO, BRANCH OVER PRINT
2159 011402 012667 000204 MOV (6)+,SAVPC ;PC OF FAILING ROUTINE
2160 011406 012667 000202 MOV (6)+,SAVC^ ;CC OF ERROR CONDITION
2161 011412 024646 CMP -(6),-(6) ;REPOSITION THE STACK
2162 011414 012777 000215 167444 MOV #215,@TPB ;CR
2163 011422 105777 167436 TSTB @TPS
2164 011426 100375 BPL .-4
2165 011430 012777 000212 167430 MOV #212,@TPB ;LINE FEED
2166 011436 105777 167422 TSTB @TPS
2167 011442 100375 BPL .-4
2168 011444 010267 000134 MOV %2,SAVR2 ;SAVE R2
2169 011450 010367 000132 MOV %3,SAVR3 ;SAVE R3
2170 011454 010467 000130 MOV %4,SAVR4 ;SAVE R4
2171 011460 016702 000126 MOV SAVPC,%2
2172 011464 004767 000126 JSR %7,PRTAB ;PRINT OCTAL NUMBER
2173 011470 012777 000240 167370 MOV #240,@TPB
2174 011476 105777 167362 TSTB @TPS ;SPACE BETWEEN WORDS
2175 011502 100375 BPL .-4
2176 011504 016702 000104 MOV SAVCC,%2
2177 011510 004767 000102 JSR %7,PRTAB ;PRINT OCTAL NUMBER
2178 011514 012777 000240 167344 MOV #240,@TPB ;PRINT SPACE
2179 011522 105777 167336 TSTB @TPS ;PRINTER DONE
2180 011526 100375 BPL .-4 ;BRANCH WHEN NOT DONE
2181 011530 017702 167270 MOV @DRST,%2 ;GET DR11B STATUS
2182 011534 004767 000056 JSR %7,PRTAB ;PRINT OCTAL NUMBER
2183 011540 016702 000040 MOV SAVR2,%2
2184 011544 016703 000036 MOV SAVR3,%3
2185 011550 016704 000034 MOV SAVR4,%4
2186 011554 004767 000504 JSR %7,CKSWR
2187 011560 005777 167230 1$: TST JSR ;CHECK SR FOR HALT SWITCH
2188 011564 100001 BPL .+4
2189 011566 000000 HALT
2190 011570 023737 000042 000046 CMP @#42,@#46 ;HALT ON ERROR UP IN SWR
2191 011576 001001 BNE .+4 ;ARE WE IN ACT11 AUTO MODE?
2192 011600 000000 HALT ;BRANCH ON NO
2193 011602 000002 RTI ;HALT ON ERROR IF IN ACT11 AUTO MODE
2194 011604 000000 SAVR2: 0 ;RETURN TO MAINLINE
2195 011606 000000 SAVR3: 0
2196 011610 000000 SAVR4: 0
2197 011612 000000 SAVPC: 0
2198 011614 000000 SAVCC: 0
2199
2200 011616 005067 000260 PRTAB: CLR BINCT
2201 011622 005067 000252 CLR WGTCT
2202 011626 012704 012106 MOV #LIST,%4 ;GET LIST ADDRESS
2203 011632 142777 000177 167224 BICB #177,@TPS ;CLR INT FLAG
2204 011640 012767 000005 000236 MOV #5,ASCNT
2205 011646 012767 000007 000220 MOV #7,SEVEN
2206 011654 012767 000001 000214 MOV #1,DECML
2207 011662 105777 167176 WAIT1: TSTB @TPS
2208 011666 100375 BPL WAIT1

```



```

2266 012142 022606          SCOPEB: CMP      (6)+,%6          ;REPOSITION THE STACK
2267 012144 012677 166646  MOV      (6)+,%PSW
2268 012150 000177 000072  JMP      @RETURN          ;SCOPE RETURN
2269
2270 ;*****
2271 ; SCOPE OR/AND ITERATION LOOP FOR EACH TEST 4000 TIMES
2272 ;*****
2273
2274
2275 012154 004767 000104  SCOPEC: JSR      %7,CKSWR
2276 012160 032777 040000 166626  BIT      #40000,@SR      ;TEST SR FOR SCOPE
2277 012166 001365          BNE      SCOPEB          ;YES SCOPE
2278 012170 005767 166700  TST      PASCNT          ;FIRST PASS (PASCNT=0) ?
2279 012174 001415          BEQ      SCOPEG          ;BR IF YES, INHIBIT ITERATIONS
2280 012176 004767 000062  JSR      %7,CKSWR
2281 012202 032777 004000 166604  BIT      #4000,@SR      ;TEST FOR ITERATION
2282 012210 001007          BNE      SCOPEG          ;INHIBIT ITERATION
2283 012212 026767 000026 000022  CMP      SCOPEF,%ICOUNT
2284 012220 001403          BEQ      SCOPEG          ;EXIT - DONE
2285 012222 005267 000016  INC      SCOPEF          ;INCREMENT COUNT
2286 012226 000745          BR       SCOPEB          ;LOOP SOME MORE
2287 012230 005067 000010  SCOPEG: CLR      SCOPEF          ;CLEAR COUNT
2288 012234 011667 000006  MOV      @%6,@RETURN      ;SAVE SCOPE RETURN POINTER
2289 012240 000002          RTI
2290 012242 004000          ICOUNT: 4000
2291 012244 000000          SCOPEF: 0
2292 012246 001174          RETURN: BEGIN          ;COUNT LOCATION FOR ITERATION LOOP
2293 ; FVEN          ;ADDRESS OF LAST TEST
2294 012250 000167 165724  JMP      200
2295
2296
2297 ;*****
2298 ; CHECK SWITCH REGISTER ROUTINE. CHECKS FOR ↑G TO ALLOW CHANGING
2299 ; OF LOC. 176.
2300 ;*****
2301 012254 000000          TEMPST: .WORD 0
2302 012256 000000          COUNT:  .WORD 0
2303 012260 000000          RDSW:   .WORD 0
2304 012262 000000          TIB:    .WORD 0
2305
2306 012264 022767 000176 166522  CKSWR:  CMP      #SWREG,SR      ;SOFTWARE SWITCH REGISTER PRESENT
2307 012272 001133          BNE      OUT
2308 012274 105777 166560  TSTB    @TKS
2309 012300 100130          BPL     OUT              ;YES, WAIT FOR
2310 012302 017767 166554 177752  MOV     @TKB,TIB          ;READY, GET CHARACTER
2311 012310 042767 177600 177744  BIC    #177600,TIB        ;AND STRIP OFF
2312 012316 022767 000007 177736  CMP    #7,TIB            ;THE GARBAGE
2313 012324 001116          BNE     OUT              ;IS IT A (↑G)
2314 012326 012702 012636  MOV     #SCNTG,%2
2315 012332 004767 000436  JSR    PC,TTOUT
2316 012336 012702 012650  CNTLU:  MOV     #SMSWR,%2
2317 012342 004767 000426  JSR    PC,TTOUT
2318 012346 017702 166442  MOV     @SR,%2
2319 012352 004767 177240  JSR    %7,PRTAB
2320 012356 012702 012660  MOV     #SMNEW,%2

```



```

2323 012366 005037 012254          $READ: CLR      @TEMPST
2323 012372 005067 177656          CLR      TEMPST
2324 012376 012767 000007 177652          MOV      #7,COUNT
2325 012404 004767 000154          1$: JSR      PC,TTIN          ;GO TO A CHARACTER
2325 012410 042767 177600 177614          BIC      @177600,TIB      ;STRIP OFF GARBAGE
2326 012416 122767 000025 177616          CMPB     #25,TIB          ;IS IT A ' '?
2326 012424 001001          BNE      2$              ;BRANCH IF NOT
2326 012426 000743          BR      CNTLU            ;START OVER
2330 012430 122767 000015 177624          2$: CMPB     #15,TIB          ;IS IT A <CR>?
2331 012436 001011          BNE      4$              ;BRANCH IF NOT
2332 012440 012702 012644          MOV      @SCRLF,%2
2333 012444 004767 000324          JSR      %7,TTOUT
2334 012450 022767 000007 177600          CMP      #7,COUNT          ;WAS IT FIRST CHARACTER
2335 012456 001036          BNE      7$              ;CHANGE SWR IF NOT FIRST ONE
2336 012460 000440          BR      OUT              ;GET OUT
2337 012462 122767 000060 177572          8$: CMPB     #60,TIB
2338 012470 003004          BGT      5$
2339 012472 122767 000067 177562          CMPB     #67,TIB
2340 012500 002005          BGE      6$
2341 012502 012702 012671          5$: MOV      @SQUEST,%2
2342 012506 004767 000262          JSR      PC,TTOUT
2343 012512 000745          BR      3$              ;START OVER IF NOT LEGAL CHARACTER
2344 012514 006367 177534          6$: ASL      TEMPST
2345 012520 006367 177530          ASL      TEMPST
2346 012524 006367 177524          ASL      TEMPST
2347 012530 142767 000060 177524          BICB     #60,TIB          ;GET NITTY-GRITTY
2348 012536 156767 177520 177510          BISB     TIB,TEMPST
2349 012544 005367 177506          DEC      COUNT          ;ONLY WANT 6 DIGITS
2350 012550 001754          BEQ      5$
2351 012552 000714          BR      1$
2352 012554 016777 177474 166232          7$: MOV      TEMPST,@SR      ;CHANGE SWITCH REGISTER CONTENTS
2353 012562 000207          OUT:    RTS      %7          ;RETURN TO PROGRAM
2354 ;*****
2355 ; TTY READ SUBROUTINE*****
2356 ;*****
2357 ;*****
2358 ;*****
2359 ;*****
2360 012564 005077 166270          TTIN:   CLR      @TKS
2361 012570 005077 166266          CLR      @TKB
2362 012574 005067 177462          CLR      TIB
2363 012600 005277 166254          INC      @TKS
2364 012604 105777 166250          TTIN1:  TSTB     @TKS
2365 012610 100375          BPL      TTIN1
2366 012612 017767 166244 177442          MOV      @TKB,TIB
2367 012620 105777 166240          TTIN2:  TSTB     @TPS
2368 012624 100375          BPL      TTIN2
2369 012626 116777 177430 166232          MOVB     TIB,@TPB
2370
2371 012634 000207          RTS      %7
2372 012636 057137 020107 000046          $CNTG:  .ASCIZ  '+IG &'
2373 012644 020137 000046          $CRLF:  .ASCIZ  '+ &'
2374 012650 051537 051127 020075          $MSWR:  .ASCIZ  '+SWR= &'
2375 012656 000046
2376 012660 020040 042516 036527          $MNEW:  .ASCIZ  ' NEW= &'
  
```

```

2378 012671 137 020077 023137 $QUEST: .ASCIZ '+? +&'
2379 012676 000
2380 012677 137 047105 020104 $ENPAS: .ASCIZ '+END PASS &'
2381 012704 040520 051523 020040
2382 012712 023040 000
2383 012715 137 046440
2384 012722 042116 041505 044501
2385 012730 026461 055104 00455
2386 012736 026502 026106 042040
2387 012744 030522 041061 046040
2388 012752 043517 041511 052040
2389 012760 051505 020124 020040
2390 012766 023137 000
2391 012772
2392
2393 012772 000000
2394
2395
2396
2397
2398
2399
2400
2401 012774 105712
2402 012776 001403
2403 013000 122712 000046
2404 013004 001005
2405 013006 042777 000100 166050
2406 013014 005002
2407 013016 000207
2408 013020 122712 000137
2409 013024 001411
2410 013026 122712 000041
2411 013032 001414
2412 013034 105777 166024
2413 013040 100375
2414 013042 122777 166020
2415 013046 000752
2416 013050 005202
2417 013052 010267 000020
2418 013056 012702 013072
2419 013062 000767
2420 013064 016702 000006
2421 013070 000741
2422
2423 013072 015 012 041
2424 013076
2425 013076 000000
2426 014100 014100
2427 014100 000000
2428 014102 014102
2429 015104
2430 015104 015104

```

```

.EVEN
OFL: 0 ;FIRST CHAR FLAG

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

TTOUT: TSTB (2) ;CHECK FOR NULL CHARACTER
BEQ 1$ ;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
TSTB @TPS
BPL 1$
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, '!'
.SAV: 0
BUFF: 0 ;FOR STACK POINTER 100 LOCATIONS
XINBUF: .=-+1000
XCHKBU: .=-+1000

```


STARS	448#	462	468	520	522	576	578	597	599	613	615	627	629	639	641
	652	654	669	671	686	688	701	703	714	716	727	729	740	742	753
	755	766	768	823	826	843	845	857	859	870	872	884	886	898	900
	921	923	944	946	973	975	1005	1007	1032	1034	1064	1066	1111	1113	1145
	1147	1166	1168	1185	1187	1204	1206	1223	1225	1249	1252	1284	1286	1296	1298
	1306	1308	1319	1321	1333	1336	1347	1349	1358	1360	1380	1384	1414	1416	1440
	1442	1465	1467	1499	1501	1518	1520	1620	1622	1639	1641	1654	1656	1673	1675
	1713	1715	1729	1731	1743	1745	1776	1778	1787	1789	1799	1801	1817	1820	1845
	1849	1874	1877	1887	1890	1902	1906	1912	1914	1947	1949	1965	1968	1984	1987
	2004	2007	2032	2035	2049	2052	2064	2067	2085	2089	2111	2114	2150	2153	2256
	2258	2270	2272	2297	2300	2354	2356	2396	2398						
. HEADE	448#	450													

. ABS. 015106 000

ERRORS DETECTED: 0

DZDRBF.BIN, DZDRBF.LST/CRF/SOL/NL: TOC=DZDRBF.P11

RUN-TIME: 4 6 .7 SECONDS

RUN-TIME RATIO: 50/11=4.4

CORE USED: 7K (13 PAGES)