

DR11-B
DA11-B

DR11-B
DA11-B NPR DIA
CZDRBGO

AH 8644G-MC
FICHE 1 OF 1

SEP 1980
COPYRIGHT 71 80
MADE IN USA



.REM %

IDENTIFICATION

PRODUCT CODE:	AC-8643G-MC
PRODUCT NAME:	CZDRBG0 DR11-B/DA11-B NPR DIA
PRODUCT DATE:	MARCH 1980
MAINTAINER:	DIAGNOSTIC ENGINEERING
AUTHOR:	LARRY CONDON, DICK JONES, STEVE POMFRET

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

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

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

COPYRIGHT (C) 1971,1980 BY DIGITAL EQUIPMENT CORPORATION

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 BC08R TABLES.
NOTE: WITH OR WITHOUT HARDWARE SWITCH REGISTER
 - 2.2 STORAGE
 - 2.2.1 PROGRAM STORAGE - THE ROUTINE USES MEMORY FROM 0 TO 14000.
3. LOADING PROCEDURE
 - 3.1 METHOD
PROCEDURE FOR NORMAL BINARY TAPES SHOULD BE FOLLOWED.
4. DR11B STARTING PROCEDURE
 - 4.1 CONTROL SWITCH SETTING
STARTING AT SA 200 ALL SWITCHES SHOULD BE DOWN OR ZERO.
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.

4.2 STARTING ADDRESS OR ADDRESSES

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

4.3 PROGRAM AND/OR OPERATOR ACTION

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

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

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
OUT ON ERRORS AND CONTINUE IN TEST. ('END
PASS' TYPED AT COMPLETION OF A PASS)

5.1.2 SWITCH SETTINGS ARE

SW15 = 1 OR UP ... HALT ON ERROR
SW14 = 1 OR UP ... SCOPE LOOP
SW13 = 1 OR UP ... INHIBIT PRINTOUT
SW12 = 1 OR UP ... INHIBIT TRACE TRAP
SW11 = 1 OR UP ... INHIBIT ITERATIONS
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 ALLOWS
THE USER THE SAME SWITCH OPTIONS AS THE HARDWARE SWITCH REGISTER.
IF THE HARDWARE SWITCH REGISTER DOES NOT EXIST OR IF ONE DOES
AND IT CONTAINS ALL ONES (177777) THEN THE SOFTWARE SWITCH
REGISTER (LOC. 176) IS USED.

CONTROL:

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

DOING THE FOLLOWING:

- 1) TYPE CONTROL G <^G>; THIS WILL ALLOW THE TTY TO ENTER DATA INTO LOC. 176 AT SELECTED POINTS WITHIN THE PROGRAM.
- 2) THE MACHINE WILL THEN TYPE: SWR=XXXXXXNEW= (XXXXXX IS THE OCTAL CONTENTS OF THE SOFTWARE SWITCH REGISTER.)
- 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 WILL BE ALLOWED)

IF A <CR> IS THE FIRST KEY DEPRESSED THE SOFTWARE SWITCH REGISTER CONTENTS WILL NOT BE CHANGED.
 - B) IF A CONTROL U <^U> IS DEPRESSED THEN THE PROGRAM WILL SEND YOU BACK TO STEP 2.

5.2. SUBROUTINE ABSTRACTS

BEGIN SA 200

5.2.1 SCOPE

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

5.2.4 CHKBFF

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

5.2.5 INTA

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

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

5.2.6 DATCHK

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

5.2.7 NORMAL

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

5.2.8 DATOCK

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

5.2.9 ERRCHK

THIS ROUTINE CLEARS IE AND HALTS IF ERROR IS SET.

5.2.10 TRAPCATCHER

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

EACH VECTOR ENTRANCE ADDRESS IS LOADED WITH THE ADDRESS
OF THE NEXT LOCATION. THE NEXT LOCATION IS LOADED WITH
A HALT (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
THE LOCATION COUNTER WHEN THE TRAP OR INTERRUPT OCCURRED.

5.3 PROGRAM AND/OR OPERATOR ACTION

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

6. DR11B ERRORS

6.1 ERROR PRINTOUT

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

6.2 ERROR RECOVERY

DEPRESS CONTINUE TO RESTART SECTION

7. DR11B RESTRICTIONS

7.1 STARTING RESTRICTION

NONE

7.2 OPERATIONAL RESTRICTION

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

8. MISCELLANEOUS

8.1 EXECUTION TIME

ABOUT 2 MINUTES

9. PROGRAM DESCRIPTION

THE FOLLOWING IS A GENERAL LIST OF FUNCTIONS TESTED.

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

MAINT SET & CLEARED
ALL DRST R/W BITS CAN BE SET & CLEARED
ALL DRST R/W SET, RESET TO 0, RDY IS SET, NEX IS
CLEARED, GO IS 0
DRWC HOLD ALT. 0'S & 1'S AND ALT. 1'S & 0'S
DRBA HOLD ALT. 0'S & 1'S AND ALT. 1'S & 0'S
INC PATTERN TO WRAP-AROUND IN DRWC
INC PATTERN TO WRAP-AROUND IN DRBA
NO INT. AT LEVEL 7
NO INT. AT LEVEL 6
NO INT. AT LEVEL 5
DOES INT. AT LEVEL 4
NO MAINTBRD A02/B02
MAINTBRD C04/D04
FNCT BITS CONTROL DSTAT BITS
RESET 0'S DRDB
ALL DRDB BITS CAN BE SET
DRDB HOLD ALT. 1'S & 0'S AND ALT. 0'S & 1'S
INC TO WRAP-AROUND IN DRDB
RESET SETS ONLY RDY IN DRST
BA00 READS AS 0
1 DAT1 NPR
1 DAT0 NPR
BA0F 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)
FNCT 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 THE DA11B MUST BE IN C04/D04 .

14. DA11B PROGRAM DISCRIPTION

- 14.1 THE SLAVE COMPUTER STARTS BY ENTERING A BACKGROUND TO WAIT FOR AN INTERRUPT WITH THE INTERRUPT ENABLED. THE FIRST INTERRUPT THAT COMES SHOULD BE THE READY INTERRUPT SET UP WHEN THE MASTER HIT THE START KEY. THE INTERRUPT CAUSES THE SLAVE TO ENTER THE INTERRUPT SERVICE ROUTINE.
- 14.2 THE INTERRUPT SERVICE ROUTINE DETERMINS WHAT INTERRUPT CAME UP AND IF IT SHOULD HAVE COME UP. IF THE INTERRUPT WAS THE ONE EXPECTED THAN THE THAN THE PROGRAM GOES TO THE TO THE PROPER JOB ROUTINE, SJOBXX FOR SLAVE SERVICE AND JOBXX FOR THE MASTER ROUTINE.
- 14.3 THE NEXT THING THAT SHOULD HAPPEN IS THE MASTER SHOULD ISSUE AN INTERRUPT TO THE SLAVE . THIS IS A SIGNAL FOR THE SLAVE TO ACCEPT THE WORD COUNT, OFFSET, AND TWO CHECK SUM WORDS. THE SLAVE ACCEPTS A WORD AT A TIME FROM THE DATA BUFFER EACH TIME THE MASTER TOGGLES FUNCTION BIT 3. EACH TIME IT READS A WORD THE SLAVE SENDS THE WORD BACK TO THE MASTER FOR VERIFICATION.

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

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

15. DA11B ERRORS

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

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

16. ECO CHANGES

CHGG1 - CHANGES 1 THRU 5 REPLACED THE INTERRUPT ENABLE COMMAND
 WITH A DATA TRANSFER BEFORE ISSUING AN INTERRUPT.
 THIS WAS NECESSARY DUE TO HARDWARE CHANGES.

%

461	:	MODIFIED FOR SOFTWARE SWITCH REGISTER
462	:	INCLUDING DYNAMIC LOADING OF SWR
463	:	
464	:	MODIFIED TO ELIMINATE BAOF ERROR WHEN
465	:	DMA TRANSFERS CROSS A 32K BOUNDRY
466	:	*****
467	:	*****
468	:	
469	000240	NOP=000240
470	104400	SCOPE=TRAP
471	104000	HLT=EMT
472	000001	BIT0=000001
473	000002	BIT1=000002
474	000004	BIT2=000004
475	000010	BIT3=000010
476	000020	BIT4=000020
477	000040	BIT5=000040
478	000100	BIT6=000100
479	000200	BIT7=000200
480	000400	BIT8=000400
481	001000	BIT9=001000
482	002000	BIT10=002000
483	004000	BIT11=004000
484	010000	BIT12=010000
485	020000	BIT13=020000
486	040000	BIT14=040000
487	100000	BIT15=100000
488		
489	000004	BUSERR=000004

```

490
491          ;LOAD TRAP CATCHER INTO 0 THRU 777.
492
493          . = 0
494          . = 30
495          000030 011520 PRINT
496          000032 000340 340
497          . = 34
498          000034 012306 SCOPEC
499          000036 000340 340
500          . = 46
501          000046 007464 $ENDAD
502          . = 52
503          000052 000000 0
504          . = 176
505          000176 000000 SWREG: 0
506          . = 200
507          000200 012706 014222 MOV #BUFF,%6 ;SET UP STACK LIMIT
508          000204 005077 000606 CLR @PSW
509          000210 023737 000042 000046 CMP @#42,@#46 ;ARE WE IN ACT11 AUTO MODE?
510          000216 001404 BEQ 1$ ;SKIP TITLE IF YES
511          000220 012702 013047 MOV #TITLE,%2 ;PRINT THE TITLE
512
513          000224 004767 012666 JSR %7,TTOUT
514          000230 000167 000646 1$: JMP SUSWR
515          001000 . = 1000
516
517
518          ;*****
519          ; START OF BACK-TO-BACK DR11-B
520          ;*****
521
522
523          001000 000000 MSTART: HALT ;MASTER START
524          001002 000167 006634 JMP MS1
525          001006 000000 SSTART: HALT ;SLAVE START
526          001010 000167 006706 JMP SS1
527          001014 177570 SR: 177570
528          001016 177776 PSW: 177776
529          001020 172410 DRWC: 172410
530          001022 172412 DRBA: 172412
531          001024 172414 DRST: 172414
532          001026 172416 DRDB: 172416
533          001030 000126 DRVS: 126
534          001032 000240 DRINL: 240
535          001034 000124 DRINV: 124
536          001036 052525 NPR1: 52525
537          001040 173000 DIOMEM: 173000
538          001042 014224 INBUF: XINBUF
539          001044 015226 CHKBUF: XCHKBUFF
540          001046 000000 BUFLN: HALT
541          001050 000000 LENCHK: HALT
542          001052 000000 BRWAIT: HALT
543          001054 000000 WCLEN: HALT
544          001056 000000 RDYCHK: HALT
545          001060 177560 TKS: 177560

```

```

546 001062 177562          TKB: 177562
547 001064 177564          TPS: 177564
548 001066 177566          TPB: 177566
549 001070 000000          FNCNT: HALT
550 001072 000000          INBUF1: HALT
551 001074 000000          PASCNT: 0
552 001076 000000          TEMP: 0
553 001100 000000          TEMP2: 0
554
555 001102 013746 000006    SUSWR: MOV    @#6,-(SP)      ;SAVE VECTORS
556 001106 013746 000004    MOV    @#4,-(SP)
557 001112 012737 001132 000004    MOV    @#6$,@#4      ;SET UP FOR TIMEOUT
558 001120 022777 177777 177666    CMP    #-1,@SR      ;REFERENCE HARDWARE SWITCH REGISTER
559 001126 001402          BEQ    65$
560 001130 000404          BR     66$
561 001132 022626          64$: CMP    (SP)+,(SP)+      ;ADJUST STACK
562 001134 012767 000176 177652    65$: MOV    #SWREG,SR      ;POINT TO SOFTWARE SWITCH REG
563 001142 012637 000004          66$: MOV    (SP)+,@#4      ;RESTORE VECTORS
564 001146 012637 000006    MOV    (SP)+,@#6
565 001152 022767 000176 177634    CMP    #SWREG,SR      ;IS SWREG USED
566 001160 001005          BNE     BEGIN
567 001162 005737 000042          TST    @#42      ;ARE WE IN AUTO MODE?
568 001166 001002          BNE     BEGIN      ;IF SO, SKIP SWREG INPUT
569 001170 004767 011274          JSR    PC,CNTLU      ;ALLOW SWREG TO BE LOADED
570 001174 012777 000340 177614    BEGIN: MOV    #340,@PSW      ;PROC. AT LEVEL #7
571 001202 012767 001174 011170    MOV    #BEGIN,RETURN
572
573
574
575
576
577
578
579 001210 104400          SCOPE
580 001212 012767 001252 176564    MOV    #ERRA,BUSERR      ;BUS ERROR VECTOR TO ERRA
581 001220 010700          MOV    %7,%0      ;PC TO R0
582 001222 005277 177572          INC    @DRWC      ;ADDRESS DRWC
583 001226 010700          MOV    %7,%0      ;PC TO R0
584 001230 005277 177566          INC    @DRBA      ;ADDRESS DRBA
585 001234 010700          MOV    %7,%0      ;PC TO R0
586 001236 005077 177562          CLR    @DRST      ;ADDRESS DRST
587 001242 010700          MOV    %7,%0      ;PC TO R0
588 001244 005277 177556          INC    @DRDB      ;ADDRESS DRDB
589 001250 000401          BR     .+4      ;MADE IT - BRANCH OVER HALT
590 001252 104000          ERRA: HLT
591 001254 012767 000006 176522    MOV    #6,BUSERR      ;BUS ERROR, R0 HAS PC OF ERROR
592 001262 104400          SCOPE
593
594
595
596
597
598
599
600 001264 012767 000010 011102    MOV    #10,ICOUNT
601 001272 012777 177777 177520    MOV    #-1,@DRWC      ;ALL ONES TO DRWC

```


CZDRBG0 DR11-B/DA11-B NFR DIA
CZDRBG.P11 01-APR-80 11:25

MACY11 30A(1052) 01-APR-80 11:29 M 1
PAGE 13

SEQ 0012

602 001300 004767 011112
603 001304 000005
604 001306 005777 177506
605 001312 001401
606 001314 104000
607 001316 104400

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

```
*****
: TEST2 DOES RESET CLEAR DRBA?
*****
```

616 001320 104400
617 001322 012777 177777 177472
618 001330 004767 011062
619 001334 000005
620 001336 005777 177460
621 001342 001401
622 001344 104000

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

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

630 001346 104400
631 001350 012767 004000 011016
632 001356 012777 177777 177434
633 001364 022777 177777 177426
634 001372 001401
635 001374 104000

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

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

642 001376 104400
643 001400 012777 177776 177414
644 001406 022777 177776 177406
645 001414 001401
646 001416 104000

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

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

655 001420 104400
656 001422 052777 000002 177374
657 001430 032777 000002 177366

```
SCOPE
BIS      #BIT1,@DRST
BIT      #BIT1,@DRST
;SET FNCT1
;TEST FNCT1
```

```

658 001436 001001      BNE      .+4      ;IS IT SET?
659 001440 104000      HLT              ;FNCT1 IS CLEAR
660 001442 042777 000002 177354      BIC      #BIT1,@DRST ;CLEAR FNCT1
661 001450 032777 000002 177346      BIT      #BIT1,@DRST ;TEST FNCT1
662 001456 001401      BEQ      .+4      ;WAS IT CLEAR
663 001460 104000      HLT              ;FNCT1 WAS SET
664
665
666
667
668      ;*****
669      ;TEST7 TEST THAT FNCT2 CAN BE SET AND CLEARED
670      ;*****
671
672 001462 104400      SCOPE
673 001464 052777 000004 177332      BIS      #BIT2,@DRST ;SET FNCT2
674 001472 032777 000004 177324      BIT      #BIT2,@DRST ;TEST FNCT2
675 001500 001001      BNE      .+4      ;IS IT SET?
676 001502 104000      HLT              ;FNCT2 IS CLEAR
677 001504 042777 000004 177312      BIC      #BIT2,@DRST ;CLEAR FNCT2
678 001512 032777 000004 177304      BIT      #BIT2,@DRST ;TEST FNCT2
679 001520 001401      BEQ      .+4      ;WAS IT CLEAR?
680 001522 104000      HLT              ;FNCT2 WAS SET
681
682
683
684
685      ;*****
686      ;TEST10 TEST THAT FNCT3 CAN BE SET AND CLEARED
687      ;*****
688
689 001524 104400      SCOPE
690 001526 052777 000010 177270      BIS      #BIT3,@DRST ;SET FNCT3
691 001534 032777 000010 177262      BIT      #BIT3,@DRST ;TEST FNCT3
692 001542 001001      BNE      .+4      ;IS IT SET?
693 001544 104000      HLT              ;FNCT3 IS CLEAR
694 001546 042777 000010 177250      BIC      #BIT3,@DRST ;CLEAR FNCT3
695 001554 032777 000010 177242      BIT      #BIT3,@DRST ;TEST FNCT3
696 001562 001401      BEQ      .+4      ;WAS IT CLEAR?
697 001564 104000      HLT              ;FNCT3 WAS SET
698
699
700      ;*****
701      ;TEST11 TEST THAT XBA16 CAN BE SET AND CLEARED
702      ;*****
703 001566 104400      SCOPE
704 001570 052777 000020 177226      BIS      #BIT4,@DRST ;SET XBA16
705 001576 032777 000020 177220      BIT      #BIT4,@DRST ;TEST XBA16
706 001604 001001      BNE      .+4      ;IS IT SET?
707 001606 104000      HLT              ;XBA16 IS CLEAR
708 001610 042777 000020 177206      BIC      #BIT4,@DRST ;CLEAR XBA16
709 001616 032777 000020 177200      BIT      #BIT4,@DRST ;TEST XBA16
710 001624 001401      BEQ      .+4      ;IS IT CLEAR
711 001626 104000      HLT              ;XBA16 WAS SET
712
713      ;*****
714      ;TEST12 TEST THAT XBA17 CAN BE SET AND CLEARED

```

```

714
715 001630 104400
716 001632 052777 000040 177164
717 001640 032777 000040 177156
718 001646 001001
719 001650 104000
720 001652 042777 000020 177144
721 001660 032777 000020 177136
722 001666 001401
723 001670 104000
724
725
726
727
728 001672 104400
729 001674 052777 000100 177122
730 001702 032777 000100 177114
731 001710 001001
732 001712 104000
733 001714 042777 000100 177102
734 001722 032777 000100 177074
735 001730 001401
736 001732 104000
737
738
739
740
741 001734 104400
742 001736 052777 000400 177060
743 001744 032777 000400 177052
744 001752 001001
745 001754 104000
746 001756 042777 000400 177040
747 001764 032777 000400 177032
748 001772 001401
749 001774 104000
750
751
752
753
754 001776 104400
755 002000 052777 010000 177016
756 002006 032777 010000 177010
757 002014 001001
758 002016 104000
759 002020 042777 010000 176776
760 002026 032777 010000 176770
761 002034 001401
762 002036 104000
763
764
765
766
767 002040 104400
768 002042 052777 010576 176754
769

```

```

;*****
SCOPE
BIS #BIT5,ADRST ;SET XBA17
BIT #BIT5,ADRST ;TEST XBA17
BNE .+4 ;IS IT SET?
HLT ;XBA17 IS CLEAR
BIC #BIT4,ADRST ;CLEAR XBA17
BIT #BIT4,ADRST ;TEST XBA17
BEQ .+4 ;IS IT CLEAR?
HLT ;XBA17 WAS SET

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

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

;*****
TEST15 TEST THAT MAINT CAN BE SET AND CLEARED
;*****
SCOPE
BIS #BIT12,ADRST ;SET MAINT
BIT #BIT12,ADRST ;TEST MAINT
BNE .+4 ;IS IT SET?
HLT ;MAINT WAS CLEAR
BIC #BIT12,ADRST ;CLEAR MAINT
BIT #BIT12,ADRST ;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,ADRST ;SET FOLLOWING: MAINT(12), CYCLE(08), IE(06), XBA17(05),
; XBA16(04), FNCT3(03), FNCT2(02), FNCT1(01)

```

770	002050	032777	000002	176746	BIT	#BIT1,DRST	:TEST FNCT1
771	002056	001001			BNE	.+4	:IS IT SET?
772	002060	104000			HLT		:FNCT1 IS CLEAR
773	002062	032777	000004	176734	BIT	#BIT2,DRST	:TEST FNCT2
774	002070	001001			BNE	.+4	:IS IT SET?
775	002072	104000			HLT		:FNCT2 IS CLEAR
776	002074	032777	000010	176722	BIT	#BIT3,DRST	:TEST FNCT3
777	002102	001001			BNE	.+4	:IS IT SET?
778	002104	104000			HLT		:FNCT3 IS CLEAR
779	002106	032777	000020	176710	BIT	#BIT4,DRST	:TEST XBA16
780	002114	001001			BNE	.+4	:IS IT SET?
781	002116	104000			HLT		:XBA16 IS CLEAR
782	002120	032777	000040	176676	BIT	#BIT5,DRST	:TEST XBA17
783	002126	001001			BNE	.+4	:IS IT SET?
784	002130	104000			HLT		:XBA17 IS CLEAR
785	002132	032777	000100	176664	BIT	#BIT6,DRST	:TEST IE
786	002140	001001			BNE	.+4	:IS IT SET?
787	002142	104000			HLT		:IE IS CLEAR
788	002144	032777	000400	176652	BIT	#BIT8,DRST	:TEST CYCLE
789	002152	001001			BNE	.+4	:IS CYCLE SET?
790	002154	104000			HLT		:CYCLE IS CLEAR
791	002156	032777	010000	176640	BIT	#BIT12,DRST	:TEST MAINT
792	002164	001001			BNE	.+4	:IS MAINT SET?
793	002166	104000			HLT		:MAINT IS CLEAR
794	002170	042777	010576	176626	BIC	#10576,DRST	:CLEAR ALL R/W BITS IN DRST
795	002176	032777	000002	176620	BIT	#BIT1,DRST	:TEST FNCT1
796	002204	001401			BEQ	.+4	:IS FNCT1 CLEAR?
797	002206	104000			HLT		:FNCT1 IS SET
798	002210	032777	000004	176606	BIT	#BIT2,DRST	:TEST FNCT2
799	002216	001401			BEQ	.+4	:IS FNCT2 CLEAR?
800	002220	104000			HLT		:FNCT2 IS SET
801	002222	032777	000010	176574	BIT	#BIT3,DRST	:TEST FNCT3
802	002230	001401			BEQ	.+4	:IS FNCT3 CLEAR?
803	002232	104000			HLT		:FNCT3 IS SET
804	002234	032777	000020	176562	BIT	#BIT4,DRST	:TEST XBA16
805	002242	001401			BEQ	.+4	:IS XBA16 CLEAR
806	002244	104000			HLT		:XBA16 IS SET
807	002246	032777	000040	176550	BIT	#BIT5,DRST	:TEST XBA17
808	002254	001401			BEQ	.+4	:IS XBA17 CLEAR?
809	002256	104000			HLT		:XBA17 IS SET
810							
811	002260	032777	000100	176536	BIT	#BIT6,DRST	:TEST IE
812	002266	001401			BEQ	.+4	:IS IE CLEAR?
813	002270	104000			HLT		:IE IS SET
814	002272	032777	000400	176524	BIT	#BIT8,DRST	:TEST CYCLE
815	002300	001401			BEQ	.+4	:IS CYCLE CLEAR?
816	002302	104000			HLT		:CYCLE IS SET
817	002304	032777	010000	176512	BIT	#BIT12,DRST	:TEST MAINT
818	002312	001401			BEQ	.+4	:IS MAINT CLEAR?
819	002314	104000			HLT		:MAINT IS SET

 : TEST17 ALL R/W BITS IN DRST CAN BE SET AND RESET TO ZERO. THAT READY
 : IS SET, NEX IS CLEAR, AND GO IS READ AS A 0.
 : *****
 SCOPE

820
 821
 822
 823
 824
 825 002316 104400

826	002320	012767	000010	010046		MOV	#10,I COUNT		
827	002326	052777	010576	176470		BIS	#10576,@DRST	:	SET FOLLOWING: MAINT(12),CYCLE(08),IE(06),XBA17(05),
828								:	XBA16(04),FNCT3(03),FNCT2(02),FNCT1(01)
829	002334	017701	176464			MOV	@DRST,%1	:	MOVE (DRST) TO R1
830	002340	052701	167201			BIS	#167201,%1	:	SETS BITS IN R1 THAT WERE NOT SET IN DRST
831	002344	005201				INC	%1	:	R1 SHOULD GO FROM -1 TO ZERO
832	002346	001401				BEQ	.+4	:	WERE ALL DRST R/W BITS SET?
833	002350	104000				HLT		:	NOT ALL BITS WERE SET
834	002352	004767	010040			JSR	%7,CKSWR		
835	002356	000005				RESET		:	CLEAR ALL DRST R/W BITS
836	002360	017701	176440			MOV	@DRST,%1	:	MOVE (DRST) TO R1
837	002364	042701	127200			BIC	#127200,%1	:	CLEAR ALL BITS EXCEPT R/W BITS, NEX, AND GO
838	002370	001401				BEQ	.+4	:	SHOULD EQUAL ZERO
839	002372	104000				HLT		:	RESET DIDN'T LEAVE DRST AS IT SHOULD HAVE
840									
841								:*****	
842								: TEST20 CAN DRWC HOLD ALTERNATE ONE'S AND ZERO'S	
843								:*****	
844	002374	104400				SCOPE			
845	002376	012767	004000	007770		MOV	#4000,I COUNT		
846	002404	012777	052525	176406		MOV	#052525,@DRWC	:	ALT 0'S AND 1'S TO DRWC
847	002412	022777	052525	176400		CMP	#052525,@DRWC	:	LOOKING FOR Z-BIT TO SET
848	002420	001401				BEQ	.+4	:	DOES DRWC HAVE THE CORRECT PATTERN?
849	002422	104000				HLT		:	DRWC DOESN'T HAVE THE CORRECT PATTERN
850	002424	012777	125252	176366		MOV	#125252,@DRWC	:	ALT 1'S AND 0'S TO DRWC
851	002432	022777	125252	176360		CMP	#125252,@DRWC	:	LOOKING FOR Z-BIT TO SET
852	002440	001401				BEQ	.+4	:	DOES DRWC HAVE THE CORRECT PATTERN?
853	002442	104000				HLT		:	DRWC DOESN'T HAVE THE CORRECT PATTERN
854									
855								:*****	
856								: TEST21 CAN DRBA HOLD ALTERNATE ONE'S AND ZERO'S	
857								:*****	
858	002444	104400				SCOPE			
859	002446	012777	052524	176346		MOV	#052524,@DRBA	:	ALT 0'S AND 1'S TO DRBA
860	002454	022777	052524	176340		CMP	#052524,@DRBA	:	LOOKING FOR Z-BIT TO SET
861	002462	001401				BEQ	.+4	:	DOES DRBA HAVE THE CORRECT PATTERN?
862	002464	104000				HLT		:	DRBA DOESN'T HAVE THE CORRECT PATTERN
863	002466	012777	125252	176326		MOV	#125252,@DRBA	:	ALT 1'S AND 0'S TO DRBA
864	002474	022777	125252	176320		CMP	#125252,@DRBA	:	LOOKING FOR Z-BIT TO SET
865	002502	001401				BEQ	.+4	:	DOES DRBA HAVE THE CORRECT PATTERN?
866	002504	104000				HLT		:	DRBA DOESN'T HAVE THE CORRECT PATTERN
867									
868								:*****	
869								: TEST22 INCREMENTING PATTERN TO WRAP-AROUND IN DRWC	
870								:*****	
871	002506	104400				SCOPE			
872	002510	005067	007660			CLR	I COUNT		
873	002514	005001				CLR	%1	:	SET-UP
874	002516	005077	176276			CLR	@DRWC	:	SET-UP
875	002522	020177	176272		INCWC:	CMP	%1,@DRWC	:	SEE IF THEY ARE EQUAL
876	002526	001401				BEQ	.+4	:	ARE THEY EQUAL?
877	002530	104000				HLT		:	THEY'RE NOT EQUAL
878	002532	005277	176262			INC	@DRWC	:	GET NEXT NUMBER
879	002536	005							

```

882
883
884
885 002542 104400
886 002544 005001
887 002546 005077 176250
888 002552 020177 176244
889 002556 001401
890 002560 104000
891 002562 062777 000002 176232
892 002570 062701 000002
893 002574 001366
894
895
896
897
898
899 002576 104400
900 002600 012767 004000 007566
901 002606 012777 000340 176202
902 002614 032777 000200 176202
903 002622 001010
904 002624 004767 007566
905 002630 000005
906 002632 032777 000200 176164
907 002640 001001
908 002642 104000
909 002644 012777 002722 176162
910 002652 012737 000340 000126
911
912 002660 012777 177777 176132
913 002666 016777 176150 176126
914 002674 052777 000101 176122
915 002702 000240
916 002704 000240
917 002706 000240
918 002710 000240
919
920 002712 042777 000100 176104
921 002720 000405
922 002722 022626
923 002724 042777 000100 176072
924 002732 104000
925
926
927
928
929 002734 104400
930 002736 012777 000300 176052
931 002744 032777 000200 176052
932 002752 001010
933 002754 004767 007436
934 002760 000005
935 002762 032777 000200 176034
936 002770 001001
937 002772 104000

;*****
; TEST23 INCREMENTING PATTERN TO WRAP-AROUND IN DRBA
;*****
SCOPE
CLR X1 ;SET-UP
CLR @DRBA ;SET-UP
INCB: CMP X1,@DRBA ;SEE IF THEY ARE EQUAL
BEQ .+4 ;ARE THEY EQUAL?
HLT ;THEY'RE NOT EQUAL
ADD #2,@DRBA ;GET NEXT NUMBER
ADD #2,X1 ;GET NEXT NUMBER
BNE INCB ;DONE WITH TEST? IF NOT CONTINUE

;*****
; TEST25 TEST THAT DR11-B DOES NOT INTERRUPT WITH PROC AT LEVEL 7
;*****
SCOPE
MOV #4000,ICOUNT
MOV #340,@PSW ;STATUS AT LEVEL 7
BIT #BIT7,@DRST ;CHECK READY BIT
BNE P7INV ;IS IT SET
JSR X7,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,@#126

;*****
CHGG1: MOV #-1,@DRWC ;SET WORD COUNT TO -1
MOV INBUF,@DRBA ;SET BUSS ADDRESS LOC
BIS #101,@DRST ;SET I.E. AND GO BITS
NOP ;ALLOW 11/60 TIME TO INTERRUPT
NOP
NOP
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,@PSW ;STATUS AT LEVEL 6
BIT #BIT7,@DRST ;CHECK READY BIT
BNE P6INV ;IS IT SET?
JSR X7,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

```

```

938 002774 012777 003052 176032 P6INV: MOV #P6ERR,@DRINV ;SET UP INT VECTOR
939 003002 012737 000340 000126 MOV #340,@#126
940
941 003010 012777 177777 176002 CHGG2: MOV #-1,@DRWC ;SET WORD COUNT TO -1
942 003016 016777 176020 175776 MOV INBUF,@DRBA ;SET BUSS ADDRESS LOC
943 003024 052777 000101 175772 BIS #101,@DRST ;SET I.E. AND GO BITS
944 003032 000240 NOP ;ALLOW 11/60 TIME TO INTERRUPT
945 003034 000240 NOP
946 003036 000240 NOP
947 003040 000240 NOP
948
949 003042 042777 000100 175754 ;***** BIC #BIT6,@DRST ;DR11-B DIDN'T INTERRUPT - CLEAR IE
950 003050 000405 BR X2
951 003052 022626 P6ERR: CMP (%6)+,(%6)+ ;RESTORE STACK
952 003054 042777 000100 175742 BIC #BIT6,@DRST ;CLEAR IE
953 003062 104000 HLT ;DR11-B INTERRUPTED, BUT IT SHOULDN'T HAVE
954
955
956 ;*****
957 ; TEST27 TEST THE DR11-B INTERRUPT WITH PROC AT LEVEL 5
958 ;*****
959 003064 104400 X2: SCOPE
960 003066 012777 000240 175722 MOV #240,@PSW ;STATUS AT LEVEL 5
961 003074 032777 000200 175722 BIT #BIT7,@DRST ;CHECK READY BIT
962 003102 001010 BNE P5INV ;IS IT SET?
963 003104 004767 007306 JSR %7,CKSWR
964 003110 000005 RESET ;INIT TO SET READY
965 003112 032777 000200 175704 BIT #BIT7,@DRST ;SEE IF READY IS SET NOW
966 003120 001001 BNE .+4 ;IS IT SET?
967 003122 104000 HLT ;RDY CAN'T BE SET BY INIT
968 003124 017767 175664 175744 P5INV: MOV @SR,TEMP ;GET SWITCH SETTINGS
969 003132 042767 177770 175736 BIC #177770,TEMP ;ISOLATE BR LEVEL
970 003140 012777 003230 175666 MOV #P5ERR,@DRINV ;SET UP INT VECTOR
971 003146 012737 000340 000126 MOV #340,@#126
972
973 003154 012777 177777 175636 CHGG3: MOV #-1,@DRWC ;SET WORD COUNT TO -1
974 003162 016777 175654 175632 MOV INBUF,@DRBA ;SET BUSS ADDRESS LOC
975 003170 052777 000101 175626 BIS #101,@DRST ;SET I.E. AND GO BITS
976 003176 000240 NOP ;ALLOW 11/60 TIME TO INTERRUPT
977 003200 000240 NOP
978 003202 000240 NOP
979 003204 000240 NOP
980
981 003206 042777 000100 175610 ;***** BIC #BIT6,@DRST ;DR11-B DIDN'T INTERRUPT - CLEAR IE
982 003214 022767 000006 175654 CMP #6,TEMP ;BR LEVEL = 6 ?
983 003222 001001 BNE 1$ ;NO, EVERYTHING IS OK
984 003224 104000 HLT ;DR11-B WITH BR = 6 SHOULD HAVE INTERRUPTED
985 003226 000411 1$: BR X3
986 003230 022626 P5ERR: CMP (%6)+,(%6)+ ;RESTORE STACK
987 003232 042777 000100 175564 BIC #BIT6,@DRST ;CLEAR IE
988 003240 022767 000006 175630 CMP #6,TEMP ;BR LEVEL = 6 ?
989 003246 001401 BEQ X3 ;YES, EVERYTHING IS OK
990 003250 104000 HLT ;DR11-B WITH BR NOT - 6 SHOULD NOT HAVE INTERRUPTED
991
992 ;*****
993 ; TEST30 TEST THE DR11-B INTERRUPT WITH PROC AT LEVEL 4

```

```

994
995 003252 104400
996 003254 012777 000200 175534
997 003262 032777 000200 175534
998 003270 001010
999 003272 004767 007120
1000 003276 000005
1001 003300 032777 000200 175516
1002 003306 001701
1003 003310 104000
1004 003312 017767 175476 175556
1005 003320 042767 177770 175550
1006 003326 012777 003424 175500
1007 003334 012737 000340 000126
1008
1009 003342 012777 177777 175450
1010 003350 016777 175466 175444
1011 003356 052777 000101 175440
1012 003364 000240
1013 003366 000240
1014 003370 000240
1015 003372 000240
1016
1017 003374 005077 175424
1018 003400 012777 000126 175426
1019 003406 022767 000004 175462
1020 003414 001401
1021 003416 104000
1022 003420 000167 000026
1023 003424 005077 175374
1024 003430 022626
1025 003432 012777 000126 175374
1026 003440 022767 000004 175430
1027 003446 001001
1028 003450 104000
1029
1030
1031
1032
1033
1034 003452 104400
1035 003454 012777 000140 175334
1036 003462 032777 000200 175334
1037 003470 001010
1038 003472 004767 006720
1039 003476 000005
1040 003500 032777 000200 175316
1041 003506 001001
1042 003510 104000
1043 003512 012777 003604 175314
1044 003520 012737 000340 000126
1045
1046 003526 012777 177777 175264
1047 003534 016777 175302 175260
1048 003542 052777 000101 175254
1049 003550 000240

*****
X3: SCOPE
MOV #200,@PSW ;STATUS AT LEVEL 4
BIT #BIT7,@DRST ;CHECK READY BIT
BNE P4INV ;IS IT SET?
JSR %7,CKSWR
RESET ;INIT TO SET READY
BIT #BIT7,@DRST ;SEE IF READY IS SET NOW
BNE .+4 ;IS IT SET
HLT ;READY CAN'T BE SET BY INIT
P4INV: MOV @SR,TEMP ;GET SWITCH SETTINGS
BIC #177770,TEMP ;ISOLATE BR LEVEL
MOV #P4INT,@DRINV ;SET UP INT VECTOR
MOV #340,@#126
*****
CHGG4: MOV #-1,@DRWC ;SET WORD COUNT TO -1
MOV INBUF,@DRBA ;SET BUSS ADDRESS LOC
BIS #101,@DRST ;SET I.E. AND GO BITS
NOP ;ALLOW 11/60 TIME TO INTERRUPT
NOP
NOP
*****
CLR @DRST ;CLEAR IE
MOV #126,@DRINV ;RESTORE INTERRUPT VECTOR
CMP #4,TEMP ;BR LEVEL = 4 ?
BEQ 1$ ;YES, EVERYTHING IS OK
HLT ;DR11-B DIDN'T INTERRUPT
1$: JMP X4
P4INT: CLR @DRST ;CLEAR IE
CMP (%6)+,(%6)+ ;REPOSITION THE STACK AFTER AN INTERRUPT
MOV #126,@DRINV ;RESTORE INTERRUPT VECTOR
CMP #4,TEMP ;BR LEVEL = 4 ?
BNE X4 ;NO, EVERYTHING IS OK
HLT ;DR11-B WITH BR = 4 SHOULD NOT HAVE INTERRUPTED

*****
TEST30A TEST THAT DR11-B DOES INTERRUPT WITH PROC AT LEVEL 3
*****
X4: SCOPE
MOV #140,@PSW ;STATUS AT LEVEL 3
BIT #BIT7,@DRST ;CHECK READY BIT
BNE P3INV ;IS IT SET?
JSR %7,CKSWR
RESET ;INIT TO SET READY
BIT #BIT7,@DRST ;SEE IF READY IS SET NOW
BNE .+4 ;IS IT SET
HLT ;READY CAN'T BE SET BY INIT
P3INV: MOV #P3INT,@DRINV ;SET UP INT VECTOR
MOV #340,@#126
*****
CHGG5: MOV #-1,@DRWC ;SET WORD COUNT TO -1
MOV INBUF,@DRBA ;SET BUSS ADDRESS LOC
BIS #101,@DRST ;SET I.E. AND GO BITS
NOP ;ALLOW 11/60 TIME TO INTERRUPT

```



```

1050 003552 000240      NOP
1051 003554 000240      NOP
1052 003556 000240      NOP
1053      ;*****
1054 003560 005077 175240      CLR @DRST      ;CLEAR IE
1055 003564 012777 000126 175242      MOV #126,@DRINV ;RESTORE INTERRUPT VECTOR
1056 003572 005037 000126      CLR @#126      ;RESTORE TRAP CATCHER
1057 003576 104000      HLT      ;DR11-B DIDN'T INTERRUPT
1058 003600 000167 000020      JMP X5
1059 003604 005077 175214      P3INT: CLR @DRST      ;CLEAR IE
1060 003610 022626      CMP (#6)+, (#6)+ ;REPOSITION THE STACK AFTER AN INTERRUPT
1061 003612 012777 000126 175214      MOV #126,@DRINV ;RESTORE INTERRUPT VECTOR
1062 003620 005037 000126      CLR @#126      ;RESTORE TRAP CATCHER
1063
1064
1065      ;*****
1066      ; TEST31 TEST THAT FNCT BITS CONTROL DSTAT BITS (M968 MUST BE USED IN USER SLOTS)
1067      ;*****
1068 003624 104400      X5: SCOPE
1069 003626 005077 175172      CLR @DRST      ;CLEAR FUNCTION BITS
1070 003632 032777 000016 175164      BIT #16,@DRST ;CHECK FUNCTION BITS
1071 003640 001401      BEQ .+4      ;FUNCTION BITS CLEAR?
1072 003642 104000      HLT      ;FUNCTION BITS NOT CLEAR
1073 003644 052777 000002 175152      BIS #BIT1,@DRST ;SET FNCT1
1074 003652 032777 001000 175144      BIT #BIT9,@DRST ;CHECK DSTAT C
1075 003660 001001      BNE .+4      ;IS IT SET?
1076 003662 104000      HLT      ;DSTAT C IS CLEAR
1077 003664 032777 006000 175132      BIT #6000,@DRST ;CHECK THAT DSTAT A AND DSTAT B ARE CLEAR
1078 003672 001401      BEQ .+4      ;ARE THEY CLEAR?
1079 003674 104000      HLT      ;DSTAT A AND/OR DSTAT B IS SET
1080 003676 005077 175122      CLR @DRST      ;CLEAR DRST
1081 003702 052777 000004 175114      BIS #BIT2,@DRST ;SET FNCT2
1082 003710 032777 002000 175106      BIT #BIT10,@DRST ;CHECK DSTAT B
1083 003716 001001      BNE .+4      ;IS IT SET?
1084 003720 104000      HLT      ;DSTAT B IS CLEAR
1085 003722 032777 005000 175074      BIT #5000,@DRST ;CHECK THAT DSTAT A AND DSTAT C ARE CLEAR
1086 003730 001401      BEQ .+4      ;ARE THEY CLEAR?
1087 003732 104000      HLT      ;DSTAT A AND/OR DSTAT B IS SET
1088 003734 005077 175064      CLR @DRST      ;CLEAR DRST
1089 003740 052777 000010 175056      BIS #BIT3,@DRST ;SET FNCT3
1090 003746 032777 004000 175050      BIT #BIT11,@DRST ;CHECK DSTAT A
1091 003754 001001      BNE .+4      ;IS IT SET?
1092 003756 104000      HLT      ;DSTAT A IS CLEAR
1093 003760 032777 003000 175036      BIT #3000,@DRST ;CHECK THAT DSTAT B AND DSTAT C ARE CLEAR
1094 003766 001401      BEQ .+4      ;ARE THEY CLEAR?
1095 003770 104000      HLT      ;DSTAT B AND/OR DSTAT C IS SET
1096 003772 005077 175026      CLR @DRST      ;CLR DRST
1097
1098      ;*****
1099      ; TEST 33 TEST FOR 1 DATI NPR TRANSFER (WITH M968 IN USER SLOTS)
1100      ;*****
1100 003776 104400      SCOPE
1101 004000 005777 175020      TNPR1: TST @DRST ;CHECK ERROR BIT
1102 004004 100027      BPL NPRRDY ;IS IT CLEAR?
1103 004006 032777 020000 175010      BIT #BIT13,@DRST ;CHECK ATTN
1104 004014 001401      BEQ .+4      ;IS ATTN CLEAR
1105 004016 104000      HLT      ;ATTN IS SET

```

```

1106 004020 032777 040000 174776 BIT #BIT14,@DRST ;CHECK NEX
1107 004026 001410 BEQ N1413 ;IS NEX CLEAR?
1108 004030 042777 040000 174766 BIC #BIT14,@DRST ;TRY TO CLEAR NEX
1109 004036 032777 040000 174760 BIT #BIT14,@DRST ;CHECK AGAIN
1110 004044 001401 BEQ .+4 ;NEX STILL SET
1111 004046 104000 HLT ;NEX CAN'T BE CLEARED BY MOVING A 0 TO IT
1112 004050 005077 174746 N1413: CLR @DRBA ;TRY TO CLEAR BAOF
1113 004054 005777 174744 TST @DRST ;CHECK ERROR BIT AGAIN
1114 004060 001401 BEQ .+4 ;IS IT CLEAR
1115 004062 104000 HLT ;ERROR CAUSED BY SOMETHING OTHER THAN NEX,ATTN, OR BAOF
1116 004064 012777 177777 174726 NPRRDY: MOV #-1,@DRWC ;SET UP FOR 1 TRANSFER
1117 004072 012777 001036 174722 MOV #NPR1,@DRBA ;TRANSFER FROM BUS ADDRESS IN NPR1
1118 004100 005077 174722 CLR @DRDB ;GET READY TO RECEIVE DATA
1119 004104 012767 052525 174724 MOV #52525,NPR1 ;SET UP TRANSFER DATA
1120 004112 012777 004156 174714 MOV #INTB,@DRINV ;INTERRUPT VECTOR TO INTB
1121 004120 012777 000005 174702 MOV #5,@DRVS ;INTERRUPT PRIORITY TO LEVEL 5
1122 004126 005077 174664 CLR @PSW ;LET THE DR11-B INTERRUPT
1123 004132 012777 000101 174664 MOV #101,@DRST ;IE AND DO TO DRST
1124 004140 005067 000002 CLR 1$+2 ;WAIT FOR NPR AND INTERRUPT
1125 004144 005227 000001 1$: INC #1
1126 004150 001375 BNE 1$
1127 004152 104000 HLT ;NO DR11-B INTERRUPT
1128 004154 000424 BR T33CLR ;CLEAR IE
1129 004156 004767 003204 INTB: JSR %7,ERRCHK
1130 004162 005777 174632 TST @DRWC ;TEST DRWC
1131 004166 001401 BEQ .+4 ;IS DRWC EQUAL TO ZERO?
1132 004170 104000 HLT ;DRWC NOT EQUAL TO ZERO
1133 004172 022777 001040 174622 CMP #NPR1+2,@DRBA ;COMPARE CORRECT DRBA WITH DRBA
1134 004200 001401 BEQ .+4 ;IS THE DRBA CORRECT?
1135 004202 104000 HLT ;DRBA IS WRONG
1136 004204 022777 052525 174614 CMP #52525,@DRDB ;CHECK FOR CORRECT DATA
1137 004212 001401 BEQ .+4 ;DATA GET TRANSFERRED?
1138 004214 104000 HLT ;BAD DATA IN DRDB
1139 004216 004767 003054 JSR %7,NORMAL
1140 004222 022626 CMP (%6)+,(%6)+ ;RESTORE STACK
1141 004224 000403 BR TNPRO ;GO TO NEXT TEST (NPR OUT)
1142 004226 005077 174572 T33CLR: CLR @DRST ;CLEAR IE
1143 004232 000662 BR TNPR1 ;TRY TEST AGAIN
1144
1145 ;*****
1146 ; TEST 34 TEST FOR 1 DATO NPR TRANSFER (WITH M968 IN USER SLOTS)
1147 ;*****
1147 004234 104400 TNPRO: SCOPE
1148 004236 012777 177777 174554 MOV #-1,@DRWC ;SET UP FOR 1 TRANSFER
1149 004244 012777 001036 174550 MOV #NPR1,@DRBA ;TRANSFER TO BUS ADDRESS IN NPR1
1150 004252 005067 174560 CLR NPR1 ;GET READY TO RECEIVE DATA
1151 004256 012777 052525 174542 MOV #52525,@DRDB ;SET UP TO TRANSFER DATA
1152 004264 012777 004330 174542 MOV #INTC,@DRINV ;INTERRUPT VECTOR TO INTC
1153 004272 016777 174534 174530 MOV DRINL,@DRVS ;INTERRUPT STATUS TO LEVEL DRINL
1154 004300 005077 174512 CLR @PSW ;PROC STATUS TO ZERO
1155 004304 012777 000103 174512 MOV #103,@DRST ;IE, FNCT1(C1 CONTROL), AND DO TO DRST
1156 004312 005067 000002 CLR 1$+2 ;WAIT FOR NPR AND INTER
1157 004316 005227 000001 1$: INC #1
1158 004322 001375 BNE 1$
1159 004324 104000 HLT ;NO DR11-B INTERRUPT
1160 004326 000424 BR T34CLR ;CLEAR IE
1161 004330 004767 003032 INTC: JSR %7,ERRCHK

```

1162	004334	005777	174460		TST	@DRWC	:TEST DRWC
1163	004340	001401			BEQ	.+4	:IS DRWC EQUAL TO ZERO?
1164	004342	104000			HLT		:DRWC EQUAL TO ZERO
1165	004344	022777	001040	174450	CMP	#NPR1+2,@DRBA	:COMPARE CORRECT DRBA WITH DRBA
1166	004352	001401			BEQ	.+4	:IS THE DRBA CORRECT?
1167	004354	104000			HLT		:DRBA IS WRONG
1168							
1169	004356	026727	174454	052525	CMP	NPR1,#52525	:CHECK FOR CORRECT DATA
1170	004364	001401			BEQ	.+4	:CORRECT DATA TRANSFERRED?
1171	004366	104000			HLT		:BAD DATA
1172	004370	004767	002702		JSR	%7,NORMAL	
1173	004374	022626			CMP	(%6)+,(%6)+	:RESTORE STACK
1174	004376	000403			BR	T35	:GO TO NEXT TEST
1175	004400	005077	174420		T34CLR: CLR	@DRST	:CLEAR IE
1176	004404	000713			BR	TNPRO	:TRY TEST AGAIN
1177							
1178							
1179							
1180							
1181	004406	104400					
1182	004410	012767	000020	174430	T35: SCOPE		
1183	004416	004767	002416		MOV	#20,BUFLEN	:LENGTH OF BUFFER=20
1184	004422	006267	174420		JSR	%7,LODBUF	:LOAD THE BUFFER WITH INCREMENTING PATTERN
1185	004426	016767	174414	174420	ASR	BUFLEN	:BUFLEN=10
1186	004434	005467	174414		MOV	BUFLEN,WLEN	:PREPARE NUMBER FOR DRWC
1187	004440	016777	174410	174352	NEG	WLEN	:2'S COMPLEMENT OF BUFLEN
1188	004446	016777	174370	174346	MOV	WLEN,@DRWC	:SET UP DRWC
1189	004454	012777	177777	174344	MOV	INBUF,@DRBA	:SET UP DRBA
1190	004462	012777	007142	174344	MOV	#-1,@DRDB	:MAINT AIDE
1191	004470	016777	174336	174332	MOV	#INTA,@DRINV	:INT VECTOR TO INTA
1192	004476	005077	174314		MOV	DRINL,@DRVS	:INT VECTOR TO PRIORITY DRINL
1193	004502	012777	000101	174314	CLR	@PSW	:LET THE DR11-B INTERRUPT
1194	004510	000777			MCV	#101,@DRST	:IE AND DO TO DRST
1195	004512	022777	000007	174306	BR	.	:WAIT FOR INTERRUPT
1196	004520	001401			CMP	#7,@DRDB	:CHECK THAT WORD #10 OF INBUF IS IN DRBA
1197	004522	104000			BEQ	.+4	:IS IT?
1198					HLT		:BAD DATA IN DRDB
1199							
1200							
1201							
1202	004524	104400					
1203	004526	012767	000020	174312	SCOPE		
1204	004534	004767	002300		MOV	#20,BUFLEN	:LENGTH OF BUFFER=20
1205	004540	006267	174302		JSR	%7,LODBUF	:LOAD THE BUFFER WITH INCREMENTING PATTERN
1206	004544	016767	174276	174302	ASR	BUFLEN	:BUFLEN=10
1207	004552	005467	174276		MOV	BUFLEN,WLEN	:PREPARE NUMBER FOR DRWC
1208	004556	016777	174272	174234	NEG	WLEN	:2'S COMPLEMENT OF BUFLEN
1209	004564	016777	174252	174230	MOV	WLEN,@DRWC	:SET UP DRWC
1210	004572	012777	052525	174226	MOV	INBUF,@DRBA	:SET UP DRBA
1211	004600	012777	007142	174226	MOV	#52525,@DRDB	:SET UP DRDB
1212	004606	016777	174220	174214	MOV	#INTA,@DRINV	:INTERRUPT VECTOR TO INTA
1213	004614	005077	174176		MOV	DRINL,@DRVS	:INTERRUPT VECTOR TO PRIORITY DRINL
1214	004620	012777	000103	174176	CLR	@PSW	:LET THE DR11-B INTERRUPT
1215	004626	000777			MOV	#103,@DRST	:IE, FNCT1(C1 CONTROL), AND DO TO DRST
1216	004630	004767	002464		BR	.	:WAIT FOR INTERRUPT
1217					JSR	%7,DATOCK	:CHECK INBUF

```

1218
1219
1220
1221 004634 104400
1222 004636 012767 000200 174202
1223 004644 004767 002170
1224 004650 016767 174172 174176
1225 004656 005467 174172
1226 004662 016777 174166 174130
1227 004670 016777 174146 174124
1228 004676 012777 177777 174122
1229 004704 012777 007142 174122
1230 004712 016777 174114 174110
1231 004720 005077 174072
1232 004724 012777 000101 174072
1233 004732 000777
1234 004734 022777 000177 174064
1235 004742 001401
1236 004744 104000
1237
1238
1239
1240 004746 104400
1241 004750 012767 000201 174070
1242 004756 004767 002056
1243 004762 005367 174060
1244 004766 016767 174054 174060
1245 004774 005467 174054
1246 005000 016777 174050 174012
1247 005006 016777 174030 174006
1248 005014 012777 052525 174004
1249 005022 012777 007142 174004
1250 005030 016777 173776 173772
1251 005036 005077 173754
1252 005042 012777 000103 173754
1253 005050 000777
1254 005052 004767 002242
1255
1256
1257
1258
1259 005056 104400
1260 005060 012777 177776 173732
1261 005066 016777 173746 173726
1262 005074 012777 005130 173732
1263 005102 016777 173724 173720
1264 005110 005077 173702
1265 005114 012777 000163 173702
1266 005122 005237 177560
1267 005126 104000
1268 005130 042777 000100 173666 NEXCHK:
1269 005136 005777 173662
1270 005142 001001
1271 005144 104000
1272 005146 105777 173652
1273 005152 001001

*****
TEST 37 STRING OF 200 DATI'S
*****
SCOPE
MOV #200,BUFLEN ;LENGTH OF BUFFER=200
JSR %7,LODBUF ;LOAD THE BUFFER WITH INCREMENTING PATTERN
MOV BUFLEN,WLEN ;PREPARE NUMBER FOR DRWC
NEG WLEN ;2'S COMPLEMENT OF BUFLEN
MOV WLEN,@DRWC ;SET UP DRWC
MOV INBUF,@DRBA ;SET UP DRBA
MOV #-1,@DRDB ;MAINT AIDE
MOV #INTA,@DRINV ;INT VECTOR TO INTA
MOV DRINL,@DRVS ;INT VECTOR TO PRIORITY DRINL
CLR @PSW ;LET THE DR11-B INTERRUPT
MOV #101,@DRST ;IE AND DO TO DRST
BR ;WAIT FOR INTERRUPT
CMP #177,@DRDB ;CHECK THAT WORD #200 OF INBUF IS IN DRBA
BEQ .+4 ;IS IT?
HLT ;BAD DATA IN DRDB
*****
TEST 40 STRING OR 200 DATO'S
*****
SCOPE
MOV #201,BUFLEN ;LENGTH OF BUFFER=201
JSR %7,LODBUF ;LOAD THE BUFFER WITH INCREMENTING PATTERN
DEC BUFLEN ;BUFLEN=200
MOV BUFLEN,WLEN ;PREPARE NUMBER FOR DRWC
NEG WLEN ;2'S COMPLEMENT OF BUFLEN
MOV WLEN,@DRWC ;SET UP DRWC
MOV INBUF,@DRBA ;SET UP DRBA
MOV #52525,@DRDB ;SET UP DRDB
MOV #INTA,@DRINV ;INTERRUPT VECTOR TO INTA
MOV DRINL,@DRVS ;INTERRUPT VECTOR TO PRIORITY DRINL
CLR @PSW ;LET THE DR11-B INTERRUPT
MOV #103,@DRST ;IE, FNCT1, AND DO TO DRST
BR ;WAIT FOR INTERRUPT
JSR %7,DATOCK ;CHECK INBUF
*****
TEST 42 TEST THAT DOING A DATO TO THE DIODE MEMORY CAUSES NEX
*****
SCOPE
MOV #-2,@DRWC ;SET UP DRWC
MOV DIOMEM,@DRBA ;SET UP DRBA
MOV #NEXCHK,@DRINV ;INTERRUPT VECTOR TO NEXCHK
MOV DRINL,@DRVS ;INTERRUPT STATUS TO LEVEL DRINL
CLR @PSW ;LET THE DR11-B INTERRUPT
MOV #163,@DRST ;IE, FNCT1, XBA17, XBA16, AND GO TO DRST
INC @#177560 ;WAIT FOR INTERRUPT
HLT ;NO DR11-B INTERRUPT
NEXCHK: BIC #BIT6,@DRST ;CLEAR INTERRUPT ENABLE
TST @DRST ;TEST DRST
BNE .+4 ;ERROR SET?
HLT ;ERROR NOT SET
TSTB @DRST ;TEST FOR READY
BNE .+4 ;READY SET?

```



```

1274 005154 104000 HLT ;READY ISN'T SET
1275 005156 032777 040000 173640 BIT #BIT14,DRST ;CHECK NEX
1276 005164 001001 BNE .+4 ;NEX SET?
1277 005166 104000 HLT ;NEX IS CLEAR
1278 005170 042777 040000 173626 BIC #BIT14,DRST ;CLEAR NEX
1279 005176 022626 CMP (%)+, (%)+ ;RESTORE THE STACK
1280 005200 004767 002072 JSR %7,NORMAL
1281
1282
1283 ;*****
1284 ; TEST 43 TEST THAT CROSSING A 32K BOUNDARY DOES NOT CAUSE
1285 ; A BAOF AND DOES NOT FORCE AN ERROR
1286 ;*****
1286 005204 104400 SCOPE
1287 005206 012767 000010 005160 MOV #10,ICOUNT
1288 005214 012777 177760 173576 MOV #-20,DRWC ;SET UP DRWC
1289 005222 012777 177776 173572 MOV #-2,DRBA ;SET UP DRBA FOR PROC STATUS ADDRESS
1290 005230 012777 005264 173576 MOV #BAOFCK,DRINV ;INTERRUPT VECTOR TO BAOFCK
1291 005236 016777 173570 173564 MOV DRINL,DRVS ;INTERRUPT STATUS TO LEVEL DRINL
1292 005244 005077 173546 CLR @PSW ;LET THE DR11-B INTERRUPT
1293 005250 012777 000163 173546 MOV #163,DRST ;IE, FNCT1, XBA17, XBA16, AND GO TO DRST
1294 005256 005237 177560 INC @#177560 ;WAIT FOR INTERRUPT
1295 005262 104000 HLT ;NO DR11-B INTERRUPT
1296 005264 042777 000100 173532 BAOFCK: BIC #BIT6,DRST ;CLEAR INTERRUPT ENABLE
1297 005272 022626 CMP (%)+, (%)+ ;RESTORE THE STACK
1298 005274 005777 173524 TST @DRST ;TEST DRST
1299 005300 100401 BMI .+4 ;ERROR SET?
1300 005302 104000 HLT ;ERROR NOT SET
1301 005304 105777 173514 TSTB @DRST ;TEST FOR READY
1302 005310 100401 BMI .+4 ;READY SET?
1303 005312 104000 HLT ;READY ISN'T SET
1304 005314 042777 040000 173502 BIC #BIT14,DRST ;CLEAR NEX
1305 005322 032777 060000 173474 BIT #60000,DRST ;CHECK NEX AND ATTN
1306 005330 001401 BEQ .+4 ;ARE THEY CLEAR?
1307 005332 104000 HLT ;NEX AND/OR ATTN IS SET
1308 005334 005777 173464 TST @DRST ;TEST FOR ERROR
1309 005340 100001 BPL .+4 ;IS ERROR CLEAR?
1310 005342 104000 HLT ;ERROR IS SET
1311 005344 005077 173452 CLR @DRBA ;CLEAR BUS ADDRESS REGISTER
1312
1313 005350 004767 005042 JSR %7,CKSWR
1314 005354 000005 RESET ;INIT
1315 005356 004767 001714 JSR %7,NORMAL
1316
1317 ;*****
1318 ; TEST 44 TEST THAT RESET CLEARS DRDB
1319 ;*****
1320 005362 104400 SCOPE
1321 005364 012767 000010 005002 MOV #10,ICOUNT
1322 005372 012777 177777 173426 MOV #-1,DRDB ;ALL ONES TO DRDB
1323 005400 004767 005012 JSR %7,CKSWR
1324 005404 000005 RESET ;INIT
1325 005406 005777 173414 TST @DRDB ;LOOKING FOR Z-BIT TO SET
1326 005412 001401 BEQ .+4 ;DID DRDB GET CLEARED?
1327 005414 104000 HLT ;DRDB NOT CLEAP
1328
1329 ;*****

```

```

1330      ;***** TEST 45 TEST THAT ALL DRDB BITS CAN BE SET *****
1331      ;*****
1332      SCOPE
1333      MOV     #4000,ICOUNT
1334      MOV     #-1,@DRDB      ;SET ALL BITS IN DRDB
1335      CMP     #-1,@DRDB      ;LOOKING FOR Z-BIT TO SET
1336      BEQ     .+4            ;SEE IF ALL BITS GOT SET
1337      HLT                     ;ALL DRDB BITS AREN'T SET
1338
1339      ;*****
1340      ;***** TEST 46 TEST THAT DRDB CAN HOLD ALTERNATE ONE'S AND ZERO'S *****
1341      ;*****
1342      SCOPE
1343      MOV     #052525,@DRDB   ;ALT 0'S AND 1'S TO DRDB
1344      CMP     #052525,@DRDB   ;LOOKING FOR Z-BIT TO SET
1345      BEQ     .+4            ;DOES DRDB HAVE THE CORRECT PATTERN?
1346      HLT                     ;DRDB IS WRONG
1347      MOV     #125252,@DRDB   ;ALT 1'S AND 0'S TO DRDB
1348      CMP     #125252,@DRDB   ;LOOKING FOR Z-BIT TO SET
1349      BEQ     .+4            ;DOES DRDB HAVE THE CORRECT PATTERN
1350      HLT                     ;DRDB IS WRONG
1351
1352      ;*****
1353      ;***** TEST 47 INCREMENTING PATTERN TO WRAP-AROUND IN DRDB *****
1354      ;*****
1355      SCOPE
1356      CLR     ICOUNT
1357      CLR     %1              ;SET-UP
1358      CLR     @DRDB          ;SET-UP
1359      INCD8:  CMP     %1,@DRDB ;SEE IF THEY ARE EQUAL
1360      BEQ     .+4            ;ARE THEY EQUAL?
1361      HLT                     ;THEY'RE NOT EQUAL
1362      INC     @DRDB          ;GET NEXT NUMBER
1363      INC     %1              ;GET NEXT NUMBER
1364      BNE     INCD8          ;DONE WITH TEST? IF NOT CONTINUE
1365
1366      ;*****
1367      ;***** TEST 50 TEST THAT RESET SETS READY AND CLEARS ALL OTHER *****
1368      ;***** DRST BITS (WITH M968 INSERTED) *****
1369      ;*****
1370      SCOPE
1371      JSR     %7,CKSWR
1372      RESET
1373      BIT     #BIT7,@DRST     ;INIT
1374      BNE     .+4            ;CHECK DRST
1375      HLT                     ;IS READY SET?
1376      BIT     #177577,@DRST   ;READY IS CLEAR
1377      BEQ     .+4            ;CHECK DRST
1378      HLT                     ;ARE THEY ALL CLEAR?
1379      ;A BIT OTHER THAN READY IS SET IN THE DRST
1380
1381      ;*****
1382      ;***** TEST 51 TEST THAT BA00 READS AS A ZERO WITH MAINT BOARD INSERTED *****
1383      ;*****
1384      SCOPE
1385      MOV     #4000,ICOUNT
1386      BIT     #BIT0,@DRBA     ;TEST BIT 0 OF DRBA
  
```

```

1386 005616 001401      BEQ      .+4      ;IS IT CLEAR?
1387 005620 104000      HLT              ;BA00 IS SET
1388
1389
1390
1391
1392
1393
1394 005622 104400      ;*****
1395 005624 012767 004000 004542      ;TEST 52 TEST THAT GO CLEARS READY
1396 005632 012777 177600 173160      ;*****
1397 005640 016777 173176 173154      SCOPE
1398 005646 105777 173152      MOV      #4000,ICOUNT
1399 005652 100401      MOV      #-200,@DRWC      ;SET-UP DRWC
1400 005654 104000      MOV      INBUF,@DRBA      ;SET-UP DRBA
1401 005656 012777 000011 173140      TSTB     @DRST      ;CHECK READY
1402 005664 105777 173134      BMI      .+4      ;IS READY SET?
1403 005670 100001      HLT              ;READY IS CLEAR
1404 005672 104000      MOV      #11,@DRST      ;FNCT3 (NON-BURST) AND GO TO DRST
1405 005674 005067 173156      TSTB     @DRST      ;CHECK READY
1406 005700 105777 173120      BPL      .+4      ;IS READY CLEAR?
1407 005704 100406      HLT              ;READY IS STILL SET
1408 005706 062767 000004 173142      CLR      RDYCHK      ;CLEAR READY CHECK
1409 005714 100401      TSTRDY: TSTB     @DRST      ;CHECK READY
1410 005716 000770      BMI      DONE      ;IF SET GO TO DONE
1411 005720 104000      ADD      #4,RDYCHK      ;CHECKING TIME FOR READY TO BE SET
1412 005722 000240      BMI      .+4      ;IF RDYCHK GETS NEGATIVE IT TOOK TOO LONG
1413      BR      TSTRDY      ;CHECK AGAIN
1414      HLT              ;READY GOT CLEARED BUT NEVER SET AGAIN
1415      DONE: NOP      ;GO TO NEXT TEST
1416
1417
1418
1419
1420
1421
1422
1423
1424
1425
1426
1427
1428
1429
1430
1431
1432
1433
1434
1435
1436
1437
1438
1439
1440
1441
005724 104400      ;*****
005726 012777 177760 173064      ;TEST 55 TEST THAT GIVING A DO WITHOUT CLEARING A PREVIOUS ERROR
005734 016777 173100 173060      ;CAUSES ANOTHER INTERRUPT
005742 012777 005776 173064      ;FOR THIS TEST, THE NORMAL INTERRUPT MECHANISM SHOULD BE WORKING
005750 012777 000140 173052      ;*****
005756 005077 173034      SCOPE
005762 012777 000163 173034      MOV      #-20,@DRWC      ;SET-UP DRWC
005770 005277 173064      MOV      DIOMEM,@DRBA      ;SET-UP DRBA
005774 104000      MOV      #ERRDO,@DRINV      ;INTERRUPT VECTOR TO ERRDO
005776 005777 173022      MOV      #140,@DRVS      ;INTERRUPT STATUS TO LEVEL 4
006002 100401      CLR      @PSW      ;LET THE DR11-B INTERRUPT
006004 104000      MOV      #163,@DRST      ;IE, XBA17,XBA16,FNCT1 AND GO TO DRST
006006 012777 006060 173020      INC      @TKS      ;WAIT FOR INTERRUPT
006014 005077 173002      HLT              ;NO DR11-B INTERRUPT
006020 042777 000062 172776      ERRDO: TST      @DRST      ;TEST DRST
006026 012777 177777 172764      BMI      .+4      ;ERROR SET?
006034 005277 172764      HLT              ;ERROR IS CLEAR - SHOULD HAVE NEX
006040 005067 000002      MOV      #ERRDO1,@DRINV      ;INTERRUPT VECTOR TO ERRDO1
006044 005227 000001      CLR      @DRBA      ;PREVENT CAUSING ANOTHER ERROR
006050 001375      BIC      #62,@DRST      ;CLEAR XBA17, XBA16, AND FNCT1
006052 104000      MOV      #-1,@DRWC      ;SET-UP DRWC
006054 162706 000004      INC      @DRST      ;DO TO DRST
006060 005777 172740      CLR      1$+2
006064 100401      1$: INC      #1
      BNE      1$
      HLT
      SUB      #4,26
      ERRDO1: TST      @DRST
      BMI      .+4
;NO DR11-B INTERRUPT
;FAKE AN INTERRUPT WITH STACK
;CHECK ERROR
;ERROR SET?

```

```

1442 006066 104000          HLT          ;ERROR IS CLEAR - SHOULD BE SET BECAUSE
1443                                     ;PREVIOUS ERROR WAS NOT CLEARED
1444 006070 062706 000010    ADD          #10,%6      ;REPOSITION THE STACK
1445 006074 004767 001176    JSR          %7,NORMAL
1446
1447                                     ;*****
1448                                     ;TEST 56 STRING OF 200 DATI'S NON-BURST MODE
1449                                     ;*****
1450 006100 104400          SCOPE
1451 006102 012767 000200 172736    MOV          #200,BUFLEN      ;LENGTH OF BUFFER=200
1452 006110 004767 000724          JSR          %7,LODBUF      ;LOAD THE BUFFER WITH INCREMENTING PATTERN
1453 006114 016767 172726 172732    MOV          BUFLEN,WCLEN    ;PREPARE NUMBER FOR DRWC
1454 006122 005467 172726          NEG          WCLEN          ;2'S COMPLEMENT OF BUFLEN
1455 006126 016777 172722 172664    MOV          WCLEN,%DRWC    ;SET-UP DRWC
1456 006134 016777 172702 172660    MOV          INBUF,%DRBA    ;SET-UP DRBA
1457 006142 012777 177777 172656    MOV          #-1,%DRDB    ;MAINT AIDE
1458 006150 012777 007142 172656    MOV          %INTA,%DRINV   ;INT VECTOR TO INTA
1459 006156 016777 172650 172644    MOV          DRINL,%DRVS   ;INT VECTOR TO PRIORITY DRINL
1460 006164 005077 172626          CLR          %PSW          ;LET THE DR11-B INTERRUPT
1461 006170 012777 000111 172626    MOV          #111,%DRST   ;IE, FNCT3, AND DO TO DRST
1462 006176 005267 172650          INC          BRWAIT        ;USE A WAIT OR BR. INSTRUCTION
1463 006202 032767 000001 172642    BIT          %BIT0,BWAIT  ;SEE WHICH ONE
1464 006210 001403          BEQ          DATINB      ;BIT 0 CLEAR=BR.
1465 006212 000001          WAIT          ;WAIT FOR INTERRUPT
1466 006214 000240          NOP
1467 006216 000401          BR          .+4
1468 006220 000777          DATINB: BR          .
1469 006222 022777 000177 172576    CMP          #177,%DRDB    ;CHECK THAT WORD #200 OF INBUF IS IN DRBA
1470 006230 001401          BEQ          .+4          ;IS IT?
1471 006232 104000          HLT          ;BAD DATA IN DRDB
1472
1473                                     ;*****
1474                                     ;TEST 57 STRING OF 200 DATO'S NON-BURST MODE
1475                                     ;*****
1476 006234 104400          SCOPE
1477 006236 012767 000201 172602    MOV          #201,BUFLEN    ;LENGTH OF BUFFER=201
1478 006244 004767 000570          JSR          %7,LODBUF      ;LOAD THE BUFFER WITH INCREMENTING PATTERN
1479 006250 005367 172572          DEC          BUFLEN        ;BUFLEN=200
1480 006254 016767 172566 172572    MOV          BUFLEN,WCLEN    ;PREPARE NUMBER FOR DRWC
1481 006262 005467 172566          NEG          WCLEN          ;2'S COMPLEMENT OF BUFLEN
1482 006266 016777 172562 172524    MOV          WCLEN,%DRWC    ;SET UP DRWC
1483
1484 006274 016777 172542 172520    MOV          INBUF,%DRBA    ;SET UP DRBA
1485 006302 012777 052525 172516    MOV          #52525,%DRDB   ;SET UP DRDB
1486 006310 012777 007142 172516    MOV          %INTA,%DRINV   ;INTERRUPT VECTOR TO INTA
1487 006316 016777 172510 172504    MOV          DRINL,%DRVS   ;INTERRUPT VECTOR TO PRIORITY DRINL
1488 006324 005077 172466          CLR          %PSW          ;LET THE DR11-B INTERRUPT
1489 006330 012777 000113 172466    MOV          #113,%DRST     ;IE, FNCT3, FNCT1, AND DO TO DRST
1490 006336 005267 172510          INC          BRWAIT        ;USE A WAIT OR BR. INSTRUCTION
1491 006342 032767 000001 172502    BIT          %BIT0,BWAIT  ;BIT 0 CLEAR=BR.
1492 006350 001403          BEQ          DATONB
1493 006352 000001          WAIT          ;WAIT FOR INTERRUPT
1494
1495 006354 000240          NOP
1496
1497 006356 000401          BR          .+4

```

```

1498 006360 000777
1499 006362 004767 000732
1500
1501
1502
1503
1504
1505 006366 104400
1506 006370 012767 000010 172450
1507 006376 016777 172440 172416
1508 006404 004767 000430
1509 006410 004767 000462
1510 006414 005077 172404
1511 006420 012767 000001 172442
1512 006426 012767 000001 172412
1513
1514 006434 016767 172402 172430
1515 006442 012777 007142 172364
1516
1517 006450 016777 172356 172352
1518 006456 005077 172334
1519 006462 012777 177777 172330
1520 006470 052777 010101 172326
1521
1522 006476 000001
1523 006500 000240
1524 006502 117701 172316
1525 006506 042701 000600
1526 006512 006201
1527 006514 126701 172350
1528 006520 001401
1529 006522 104000
1530 006524 005267 172340
1531 006530 022767 000010 172332
1532 006536 001404
1533 006540 062767 000002 172274
1534 006546 000735
1535 006550 012767 000007 172270
1536 006556 016767 172310 172256
1537 006564 004767 000446
1538
1539
1540
1541
1542 006570 104400
1543 006572 012767 000010 172246
1544 006600 016767 172242 172246
1545 006606 005467 172242
1546 006612 004767 000222
1547 006616 004767 000254
1548 006622 016777 172226 172170
1549 006630 016777 172206 172164
1550 006636 012777 177777 172162
1551 006644 012777 007142 172162
1552 006652 016777 172154 172150
1553 006660 005077 172132

DATONB: BR
JSR %7,DATOCK ;CHECK INBUF

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

SCOPE
MOV #10,BUFLEN ;SET-UP BUFLEN FOR LODBUF AND CHKBUFF
MOV INBUF,ADRBA ;SET-UP DRBA
JSR %7,LODBUF ;LOAD INBUF
JSR %7,CHKBFF ;LOAD CHKBUFF
CLR @DRST ;INIT FOR STARTING
MOV #1,FNCNT ;GET READY FOR CHECKING
MOV #1,BUFLEN ;CHANGE IS NECESSARY FOR INTA ROUTINE

MFLOOP: MOV INBUF,INBUF1 ;SAVE INBUF
MOV #INTA,ADRINV ;INTERRUPT VECTOR TO INTA

MOV DRINL,ADRVS ;INTERRUPT VECTOR PRIORITY TO DRINL
CLR @PSW ;LET THE DR11-B INTERRUPT
MOV #-1,@DRWC ;SET-UP FOR 1 TRANSFER
BIS #10101,@DRST ;MAINT, IE, AND DO TO DRST

WAIT ;WAIT FOR INTERRUPT
NOP ;FAKE-OUT RETURN ADDRESS CHANGING
MOVB @DRST,%1 ;LOWER BYTE OF DRST TO R1
BIC #600,%1 ;GET RID OF READY AND CYCLE BECAUSE OF MAINT MODE
ASR %1 ;MOVE IT RIGHT ONE PLACE
CMPB FNCNT,%1 ;CHECK AGAINST FNCNT
BEQ .+4 ;SHOULD BE EQUAL
HLT ;FUNCTION BITS DIDN'T INCREMENT IN MAINT MODE
INC FNCNT ;GET READY FOR NEXT PASS
CMP #10,FNCNT ;ONLY 10 BECAUSE FNCT3-1 GO TO ZERO
BEQ MFCHK ;IF ITS EQUAL GO CHECK DATA
ADD #2,INBUF ;FAKE-OUT INTA ROUTINE
BR MFLOOP ;DO IT AGAIN

MFCHK: MOV #7,BUFLEN ;SET UP FOR DATCHK (10 FNCT CHECKS, 7 TRANSFERS)
MOV INBUF1,INBUF ;RESTORE INBUF
JSR %7,DATCHK ;CHECK DATA

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

SCOPE
MOV #10,BUFLEN ;BUFLEN=10
MOV BUFLEN,WLEN ;PREPARE NUMBER FOR DRWC
NEG WLEN ;2'S COMPLEMENT OF BUFLEN
JSR %7,LODBUF ;LOAD IN BUFFER WITH INCREMENTING PATTERN
JSR %7,CHKBFF ;LOAD CHECK BUFFER WITH MODIFIED INCREMENTING PATTERN
MOV WLEN,@DRWC ;SET UP DRWC
MOV INBUF,ADRBA ;SET UP DRBA
MOV #-1,@DRDB ;MAINT AIDE
MOV #INTA,ADRINV ;INTERRUPT VECTOR TO INTA
MOV DRINL,ADRVS ;INTERRUPT STATUS AT PRIORITY DRINL
CLR @PSW ;LET DR11-B INTERRUPT

```

1554	006664	012777	010101	172132	MOV	#10101,ADRST	;MAINT, IE, AND DO TO DRST
1555	006672	000777			BR		;WAIT FOR INTERRUPT
1556	006674	004767	000336		JSR	%7,DATCHK	
1557							
1558							
1559							
1560							
1561	006700	104400					
1562	006702	012767	000200	172136	MOV	#200,BUFLEN	;LENGTH OF BUFFER = 200
1563	006710	016767	172132	172136	MOV	BUFLEN,WLEN	;PREPARE NUMBER FOR DRWC
1564	006716	005467	172132		NEG	WLEN	;2'S COMPLEMENT OF BUFLEN
1565	006722	004767	000112		JSR	%7,LODBUF	;LOAD INBUF WITH INCREMENTING PATTERN
1566	006726	004767	000144		JSR	%7,CHKBFF	;LOAD CHKBUFF WITH MODIFIED INCREMENTED PATTERN
1567	006732	016777	172116	172060	MOV	WLEN,ADRWC	;SET UP DRWC
1568	006740	016777	172076	172054	MOV	INBUF,ADRBA	;SET UP DRBA
1569	006746	012777	000001	172052	MOV	#1,ADRDB	;MAINT AIDE
1570	006754	012777	007142	172052	MOV	#INTA,ADRINV	;INT VECTOR TO INTA
1571	006762	016777	172044	172040	MOV	DRINL,ADRVS	;INT VECTOR AT PRIORITY DRINL
1572	006770	005077	172022		CLR	APSW	;LET THE DR11-B INTERRUPT
1573	006774	012777	010101	172022	MOV	#010101,ADRST	;FOLLOWING TO DRST: MAINT(12),IE(06),DO(00)
1574	007002	005267	172044		INC	BRWAIT	;USE A WAIT OR BR. INSTRUCTION
1575	007006	032767	000001	172036	BIT	#BIT0,BRWAIT	;SEE WHICH ONE
1576	007014	001403			BEQ	BRANCH	;BIT 0 CLEAR = BR.
1577	007016	000001			WAIT		
1578	007020	000240			NOP		
1579	007022	000401			BR	.+4	
1580	007024	000777			BRANCH: BR		
1581	007026	004767	000204		JSR	%7,DATCHK	;CHECK THAT CORRECT DATA WAS TRANSFERRED
1582	007032	104400			SCOPE		
1583	007034	000167	000356		JMP	END	;DO IT ALL AGAIN.
1584	007040	016702	171776		LODBUF: MOV	INBUF,%2	;MOVE STARTING ADDRESS OF INBUF TO R2
1585	007044	005067	172000		CLR	LENCHK	;CLEAR LENGTH CHECK
1586	007050	005022			CLR	(%2)+	;CLEAR STARTING ADDRESS OF INBUF AND INC BY 2
1587	007052	005267	171772		LOADA	INC	;INC LENGTH CHECK BY 1
1588	007056	026767	171766	171762	CMP	LENCHK,BUFLEN	;CHECK FOR DONE
1589	007064	001403			BEQ	LDEXIT	;IS INBUF FILLED?
1590	007066	016722	171756		MOV	LENCHK,(%2)+	;LOAD NEXT BUFFER WORD
1591	007072	000767			BR	LOADA	;CONTINUE CHECKING
1592	007074	000207			LDEXIT: RTS	%7	;EXIT
1593	007076	016702	171742		CHKBFF: MOV	CHKBUFF,%2	;STARTING ADDRESS OF CHECK-BUFFER TO R2
1594	007102	005067	171742		CLR	LENCHK	;CLEAR LENGTH CHECK
1595	007106	005003			CLR	%3	;CLEAR R3
1596	007110	010322			CHKA: MOV	%3,(%2)+	;MOVE R3 TO CHKBUFF ADDRESS AND INC BY 2
1597	007112	010322			MOV	%5,(%2)+	;MOVE R3 TO NEXT CHKBUFF ADDRESS AND INC BY 2
1598	007114	062767	000002	171726	ADD	#2,LENCHK	;ADD 2 TO LENGTH CHECK
1599	007122	026767	171722	171716	CMP	LENCHK,BUFLEN	;CHECK FOR DONE
1600	007130	100003			BPL	.+10	;IS CHECK-BUFFER FILLED?
1601	007132	062703	000002		ADD	#2,%3	;NEXT NUMBER FOR BUFFER
1602	007136	000764			BR	CHKA	;CONTINUE FILLING
1603	007140	000207			RTS	%7	;EXIT
1604	007142	042777	000100	171654	INTA: BIC	#BIT6,ADRST	;CLEAR IE
1605	007150	005777	171650		TST	ADRST	;CHECKING FOR ERROR
1606	007154	100001			BPL	.+4	;ERROR SET?
1607	007156	104000			HLT		;ERROR BIT IS SET
1608	007160	105777	171640		TSTB	ADRST	;CHECKING READY BIT
1609	007164	100401			BMI	.+4	;IS READY SET

1610	007166	104000			HLT			;FALSE INTERRUPT - ERROR AND READY ARE CLEAR
1611	007170	005777	171624		TST	@DRWC		;TEST1 FOR DRWC=0
1612	007174	001401			BEQ	+.4		;WAS IT EQUAL?
1613	007176	104000			HLT			;DRWC NOT =0
1614	007200	016702	171642		MOV	BUFLEN,%2		;BUFFER LENGTH TO R2
1615	007204	066702	171636		ADD	BUFLEN,%2		;NUMBER OF TRANSFERS TIMES 2
1616	007210	066702	171626		ADD	INBUF,%2		;CORRECT DRBA
1617	007214	027702	171602		CMP	@DRBA,%2		;CHECKING DRBA
1618	007220	001401			BEQ	+.4		;IS DRBA CORRECT?
1619	007222	104000			HLT			;DRBA NOT CORRECT
1620	007224	062716	000002		ADD	#2,(%6)		;RETURN ADDRESS TO RETURN ADDRESS +2
1621	007230	004767	000042		JSR	%7,NORMAL		
1622	007234	000002			RTI			;EXIT
1623								
1624	007236	016702	171602		CATCHK: MOV	CHKBUF,%2		;STARTING ADDRESS OF CHECK BUFFER TO R2
1625	007242	016703	171574		MOV	INBUF,%3		;STARTING ADDRESS OF IN BUFFER TO R3
1626	007246	005067	171576		CLR	LENCHK		;CLEAR LENGTH CHECK
1627	007252	005267	171572		COMPAR: INC	LENCHK		;MAKE A COMPARISON
1628	007256	022223			CMP	(%2)+,(%3)+		;IS THE DATA CORRECT?
1629	007260	001401			BEQ	+.4		;BRANCH IF OK
1630	007262	104000			HLT			;BAD DATA
1631	007264	026767	171560	171554	CMP	LENCHK,BUFLEN		;SEE IF THE BUFFER HAS BEEN CHECKED
1632	007272	001367			BNE	COMPAR		;BUFFER CHECKED?
1633	007274	000207			RTS	%7		
1634	007276	012777	001030	171530	NORMAL: MOV	@DRVS,@DRINV		;RESTORE DR11-B INTERRUPT VECTOR
1635	007304	005077	171520		CLR	@DRVS		;RESTORE DR11-B INTERRUPT STATUS
1636	007310	012777	000340	171500	MOV	#340,@PSW		;RESTORE PROC TO PRIORITY LEVEL 7
1637	007316	000207			RTS	%7		;EXIT
1638	007320	012702	052525		DATOCK: MOV	#52525,%2		;DATO NUMBER TO R2
1639	007324	016703	171512		MOV	INBUF,%3		;STARTING ADDRESS OF IN BUFFER TO R3
1640	007330	005067	171514		CLR	LENCHK		;CLEAR LENGTH CHECK
1641	007334	005267	171510		COMPRR: INC	LENCHK		;MAKE A COMPARISON
1642	007340	020223			CMP	%2,(%3)+		;IS THE DATA CORRECT?
1643	007342	001401			BEQ	+.4		;BRANCH IF OK
1644	007344	104000			HLT			;BAD DATA
1645	007346	026767	171476	171472	CMP	LENCHK,BUFLEN		;SEE IF THE BUFFER HAS BEEN CHECKED
1646	007354	001367			BNE	COMPRR		;BUFFER CHECKED?
1647	007356	020223			CMP	%2,(%3)+		;CHECK END OF BUFFER + 1
1648	007360	001001			BNE	+.4		;SEE IF TOO MANY WORDS WERE TRANSFERRED
1649	007362	104000			HLT			;TOO MANY
1650	007364	000207			RTS	%7		;EXIT
1651	007366	042777	000100	171430	ERRCHK: BIC	#BIT6,@DRST		;CLEAR IE
1652	007374	005777	171424		TST	@DRST		;CHECKING FOR ERROR
1653	007400	100001			BPL	+.4		;ERROR SET?
1654	007402	104000			HLT			;ERROR BIT IS SET
1655	007404	105777	171414		TSTB	@DRST		;CHECKING READY BIT
1656	007410	100401			BMI	+.4		;IS RDY SET
1657	007412	104000			HLT			;FALSE ENTRY - ERROR AND READY ARE CLEAR
1658	007414	000207			RTS	%7		;EXIT
1659								
1660					;;*****			
1661					END OF PASS			
1662					;;*****			
1663	007416	012777	000207	177566	END: MOV	#207,@177566		;RING BELL
1664	007424	105757	177564		TSTB	@177564		
1665	007430	100375			BPL	-.4		

```

1666 007432 005267 171436      INC      PASCNT      ;KEEP TRACK OF PASSES COMPLETED
1667 007436 012702 013031      MOV      #SENPA5,%2 ;PRINT 'END PASS'
1668 007442 004767 003450      JSR      %7,TTOUT
1669
1670
1671 007446 042777 000020 171342 END1: BIC      #20,@PSW      ;CLEAR T-BIT
1672 007454 013702 000042      MOV      @#42,%2
1673 007460 001405      BEQ      TRTRAP
1674 007462 000005      RESET
1675 007464 004712      SENDAD: JSR      %7,(2)
1676 007466 000240      NOP
1677 007470 000240      NOP
1678 007472 000240      NOP
1679
1680      ;*****
1681      ;ROUTINE TO CHECK FOR TRACE TRAP TO BE RUN WITH PROGRAM
1682      ;*****
1683 007474 004767 002716      TRTRAP: JSR      %7,CKSWR      ;CHECK FOR CONT G
1684 007500 032777 010000 171306      BIT      #10000,@SR      ;SHOULD WE RUN WITH TRACE TRAP
1685 007506 001417      BEQ      YESTR      ;YES
1686 007510 005767 000104      TST      YESTR1      ;NO, HAVE WE RUN WITH TRACE TRAP ON?
1687 007514 001411      BEQ      TRPA      ;IF SO RESTORE PREVIOUS CONTENTS
1688 007516 016767 000076 170270      MOV      YESTR1,14
1689 007524 016767 000072 170264      MOV      YESTR2,16
1690 007532 042777 000020 171256      BIC      #20,@PSW      ;CLEAR TRACE TRAP
1691 007540 000167 171430      TRPA:   JMP      BEGIN      ;START OF TEST WITH TRACE OFF
1692 007544 000000      TRPB:   0
1693
1694      ;*****
1695      ;SAVE OLD CONTENTS, SET UP FOR TRACE TRAP
1696      ;*****
1697 007546 016767 170242 000044      YESTR:  MOV      14,YESTR1      ;SAVE ODT PC
1698 007554 016767 170236 000040      MOV      16,YESTR2      ;SAVE ODT STATUS
1699 007562 012767 007624 170224      MOV      #YESRT,14      ;NEW TRAP VECTOR
1700 007570 005067 170222      CLR      16      ;NEW CONDITION CODES
1701 007574 005077 171216      CLR      @PSW
1702 007600 005167 177740      COM      TRPB
1703 007604 100403      BMI      .+10
1704 007606 052777 000020 171202      BIS      #20,@PSW      ;SET TRACE TRAP
1705 007614 000167 171354      JMP      BEGIN      ;START OF TEST WITH TRACE ON
1706
1707 007620 000000      YESTR1: 0      ;STORAGE FOR ODT PC
1708 007622 000000      YESTR2: 0      ;STORAG FOR ODT STATUS
1709 007624 000002      YESRT: RTI      ;RETURN TO PROGRAM FROM TRAP
1710 007626 000000      HALT      ;RTI FAILED
1711
1712
1713      ;*****
1714      ;BUS TO BUS TEST (DR11-B TO DR11-B)
1715      ;*****
1716
1717
1718      000000      R0=%0
1719      000001      R1=%1
1720      000002      R2=%2
1721      000003      R3=%3

```

1722		000004		R4=%4	
1723		000005		R5=%5	
1724		000006		R6=%6	
1725		000007		R7=%7	
1726		000007		PC=%7	
1727		000006		SP=%6	
1728					
1729		000001		GO=1	
1730		000002		FNCT1=2	; OUTPUT MODE
1731		000004		FNCT2=4	; OUTPUT DIRECTION
1732		000010		FNCT3=10	; OUTPUT INTER REQ
1733		000020		XBA16=20	
1734		000040		XBA17=40	
1735		000100		IE=100	
1736		000200		READY=200	
1737		000400		CYCLE=400	
1738		001000		DSTAT=1000	; INPUT MODE
1739		002000		DSTATB=2000	; INPUT DIRECTION
1740		004000		DSTATC=4000	; INPUT INTR REQ
1741		010000		MAINT=10000	
1742		020000		ATTN=20000	
1743		040000		NEX=40000	
1744		100000		ERROR=100000	
1745					
1746					
1747					
1748	007630	000004		NWRDXF: 4	; # OF WORDS TRANSFERRED UNDER FLAG CONTROL PRIOR TO NPR
1749	007632	000000		NEXJOB: 0	; HOLDS ADDRESS OF READY INTERRUPT ROUTINE
1750	007634	000000		0	; HOLDS ADDRESS OF ERROR INTERRUPT ROUTINE
1751	007636	000000		JBFLAG: 0	; JOB FLAG
1752	007640	000000		JBCNT: 0	; JOB COUNT
1753				*****	
1754				: MASTER START	
1755				*****	
1756	007642	000240		MS1: NOP	
1757	007644	005067	177770	CLR JBCNT	; CLEAR JOB COUNT
1758	007650	012706	014222	MOV #BUFF,R6	; SETUP STACK
1759	007654	004767	000256	JSR R7,SETVEC	; SET UP INTERRUPT VECTORS
1760	007660	005067	177752	CLR JBFLAG	; CLEAR JOB FLAG
1761	007664	012767	010316 177740	MOV #JOB0,NEXJOB	; DO JOB0 FIRST
1762	007672	012767	000000 177734	MOV #0,NEXJOB+2	; NO ERROR RECOVERY
1763	007700	012777	000340 171110	MOV #340,@PSW	; LOCK OUT INTERRUPTS
1764	007706	012777	000100 171110	MOV #IE,@DRST	; SET INTERRUPT ENABLE
1765	007714	005077	171076	CLR @PSW	; DROP PRIOTIRY TO ZERO
1766	007720	000425		BR BACKGD	; WAIT FOR JOBS IN BACKGROUND
1767					
1768					
1769				*****	
1770				: SLAVE START	
1771				*****	
1772					
1773	007722	000240		SS1: NOP	
1774	007724	012706	014222	MOV #BUFF,R6	
1775	007730	004767	000202	JSR R7,SETVEC	; SET UP INTERRUPT VECTORS
1776	007734	005067	177676	CLR JBFLAG	
1777	007740	012767	010540 177664	MOV #SJOB1,NEXJOB	; FOR READY INTERRUPT

```

1778 007746 012767 010556 177660      MOV    #SJOB2,NEXJOB+2 ;FOR ERROR INTERRUPT
1779 007754 012777 000340 171034      MOV    #340,@PSW      ;RAISE CP PRIORITY TO 7
1780 007762 012777 000100 171034      MOV    #IE,@DRST     ;SET INTERRUPT ENABLE
1781 007770 005077 171022                CLR    @PSW        ;DROP CP PRIORITY TO 0 AND ENTER BACKGROUND
1782
1783      ;*****
1784      ; BACKGROUND PROGRAM; WAITS FOR JBFLAG TO SET
1785      ;*****
1786
1787 007774 005200      BACKGD: INC    R0
1788 007776 005201      INC    R1
1789 010000 005202      INC    R2
1790 010002 005203      INC    R3
1791 010004 005204      INC    R4
1792 010006 005205      INC    R5
1793 010010 020005      CMP    R0,R5
1794 010012 001402      BEQ    .+6
1795 010014 104000      HLT
1796 010016 000000      HALT          ;BACKGROUND TEST FAILED
1797 010020 020104      CMP    R1,R4
1798 010022 001402      BEQ    .+6
1799 010024 104000      HLT
1800 010026 000000      HALT          ;BACKGROUND TEST FAILED
1801 010030 020203      CMP    R2,R3
1802 010032 001402      BEQ    .+6
1803 010034 104000      HLT
1804 010036 000000      HALT          ;BACKGROUND TEST FAILED
1805 010040 005767 177572      TST    JBFLAG      ;ANY JOBS?
1806 010044 001753      BEQ    BACKGD      ;BRANCH IF NONE
1807 010046 004567 000032      JSR    R5,SAVALL      ;YES & EXECUTE JOB WHOSE ADDRESS IS IN JBFLAG
1808 010052 005067 177554      CLR    NEXJOB
1809 010056 005067 177552      CLR    NEXJOB+2
1810 010062 016767 177550 167744      MOV    JBFLAG,34
1811 010070 005067 177542      CLR    JBFLAG
1812 010074 104400      TRAP
1813 010076 004567 000016      JSR    R5,RESALL      ;TRAP THROUGH JBFLAG AT 34
1814 010102 000734      BR     BACKGD
1815
1816      ;*****
1817      ; SUBROUTINE TO PUSH ALL REGISTERS ONTO THE STACK
1818      ;*****
1819
1820 010104 010446      SAVALL: MOV    R4,-(R6)      ;R5 WAS PUSHED BY JSR
1821 010106 010346      MOV    R3,-(R6)
1822 010110 010246      MOV    R2,-(R6)
1823 010112 010146      MOV    R1,-(R6)
1824 010114 010046      MOV    R0,-(R6)
1825 010116 000115      JMP     (R5)          ;R5 HOLDS RETURN ADDRESS
1826
1827      ;*****
1828      ; SUBROUTINE TO POP ALL REGISTERS OFF THE STACK
1829      ;*****
1830
1831 010120 005726      RESALL: TST    (R6)+
1832 010122 012600      MOV    (R6)+,R0
1833 010124 012601      MOV    (R6)+,R1
  
```

```

1834 010126 012602      MOV      (R6)+,R2
1835 010130 012603      MOV      (R6)+,R3
1836 010132 012604      MOV      (R6)+,R4
1837 010134 000205      RTS       R5
1838
1839
1840      ;*****
1841      ;ROUTINE 10 SET UP INTERRUPT VECTORS
1842      ;*****
1843 010136 016700 170672  SETVEC: MOV      DRINV,R0      ;R0 IS VECTOR ADDRESS
1844 010142 012720 010216      MOV      #DRINS,(R0)+ ;PUT SERVICE ADDRESS INTO VECTOR
1845 010146 016710 170660      MOV      DRINL,(R0)   ;PUT PRIORITY INTO VECTOR+2
1846 010152 012767 011520 167650      MOV      #PRINT,30    ;SET UP EMT ADDRESS
1847 010160 016767 170646 167644      MOV      DRINL,32    ;SET UP EMT PRIORITY LEVEL
1848 010166 005067 167642      CLR       34
1849 010172 016767 170634 167636      MOV      DRINL,36    ;SET UP TRAP ADDRESS
1850 010200 005000      CLR       R0      ;INITIALIZE REGISTERS
1851 010202 005001      CLR       R1
1852 010204 005002      CLR       R2
1853 010206 005003      CLR       R3
1854 010210 005004      CLR       R4
1855 010212 005005      CLR       R5
1856 010214 000207      RTS       R7
1857
1858      ;*****
1859      ;PRIMARY INTERRUPT SERVICE ROUTINE.
1860      ;SETS UP JBFLAG WITH ADDRESS OF JOB TO BE RUNSTARS
1861      ;*****
1862 010216 005767 177414      DRINS:  TST      JBFLAG      ;HAS THE PREVIOUS INTERRUPT BEEN SERVICED?
1863 010222 001402      BEQ      DRINO
1864 010224 104000      HLT
1865 010226 000000      HALT
1866 010230 032777 004000 170566      DRINO:  BIT      #DSTATC,@DRST ;NO
1867 010236 001411      BEQ      DRIN3      ;CHECK FOR ERROR
1868 010240 005767 177370      TST      NEXJOB+2    ;BRANCH IF NO ERROR
1869 010244 001002      BNE      DRIN1      ;IS THERE AN ERROR SERVICE ROUTINE?
1870 010246 104000      HLT                ;BRANCH IF THERE IS.
1871 010250 000000      HALT
1872 010252 016767 177356 177356      DRIN1:  MOV      NEXJOB+2,JBFLAG ;ERROR INTERRUPT, NO ERROR SERVICE.
1873 010260 000002      RTI                ;SET UP JOBFLAG WITH ADDRESS OF SERVICE ROUTINE
1874 010262 105777 170536      DRIN3:  TSTB     @DRST      ;CHECK READY
1875 010266 100402      BMI      DRIN2      ;BRANCH IF SET
1876 010270 104000      HLT
1877 010272 000000      HALT
1878 010274 005767 177332      DRIN2:  TST      NEXJOB      ;INTERUPT WITHOUT ERROR OR READY
1879 010300 001002      BNE      .+6        ;IS THERE A READY SERVICE ROUTINE
1880 010302 104000      HLT                ;BRANCH IF THERE IS.
1881 010304 000000      HALT
1882 010306 016767 177320 177322      MOV      NEXJOB,JBFLAG ;READY INTERRUPT, NO READY SERVICE
1883 010314 000002      RTI                ;SET UP JOBFLAG WITH SERVICE ROUTINE ADDRESS
1884
1885      ;*****
1886      ;MASTER'S INTERRUPT SERVICE ROUTINES
1887      ;ROUTINE A, SEGMENT 0
1888      ;FILL BUFFER AND TRANSMIT
1889      ;*****

```

```

1890 010316 012700 011462      JOBA0: MOV    #LISTA,R0      ;R0 IS XMIT LIST ADDRESS
1891 010322 012701 011474      MOV    #LISTA1,R1     ;LISTA1 WILL BE REC LIST
1892 010326 012021              MOV    (R0)+,(R1)+      ;START WITH BUS ADDRESSES EQUAL
1893 010330 012002              MOV    (R0)+,R2         ;R2 HOLDS WORD COUNT OF XMIT
1894 010332 010211              MOV    R2,(R1)          ;MAKE REC WORD COUNT THE SAME
1895 010334 005402              NEG     R2              ;MAKE WORD COUNT POSITIVE
1896 010336 006302              ASL     R2              ;TRANSFORM INTO BYTE COUNT
1897 010340 060241              ADD     R2,-(R1)        ;ADD TO REC BUS ADDRESS
1898 010342 005010              CLR     (R0)          ;CLEAR OFFSET IN XMIT LIST
1899 010344 024040              CMP     -(R0),-(R0)    ;LEAVE R0=LISTA=XMIT LIST
1900
1901 010346 022767 000100 177264      CMP     #100,JBCNT      ;ENOUGH PASSES FOR BELL?
1902 010354 003010              BGT     JOBA0A         ;BRANCH IF NOT ENOUGH
1903 010356 105737 177564      JOBA0B: TSTB    @#177564    ;TTY READY?
1904 010362 100375              BPL     JOBA0B
1905 010364 012737 000207 177566      MOV     #207,@#177566  ;RING BELL
1906 010372 005067 177242              CLR     JBCNT         ;RESET JOB COUNT
1907 010376 004767 000704      JOBA0A: JSR     R7,SETBUF  ;FILL UP XMIT BUFFER WITH SPECIAL BINARY COUNT
1908 010402 012700 011462              MOV     #LISTA,R0
1909 010406 004767 000370              JSR     R7,MXMIT       ;TRANSMIT DATA TO SLAVE
1910 010412 012767 010430 177212      MOV     #JOBA1,NEXJOB  ;JOBA1 IS NEXT
1911 010420 012767 000000 177206      MOV     #0,NEXJOB+2    ;NO ERROR RECOVERY
1912 010426 000002              RTI              ;RETURN TO BACKGROUND VIA TRAP
1913
1914      ;*****
1915      ;ROUTINE A, SEGMENT 1
1916      ;FLUSH A BUFFER AND RECEIVE DATA
1917      ;*****
1918
1919 010430 012700 011474      JOBA1: MOV     #LISTA1,R0  ;PUT REC LIST ADDRESS INTO R0
1920 010434 004767 000574              JSR     R7,FLUSH       ;FLUSH BUFFER
1921 010440 012700 011474              MOV     #LISTA1,R0
1922 010444 004767 000406              JSR     R7,MREC        ;RECEIVE DATA FROM SLAVE
1923 010450 012767 010466 177154      MOV     #JOBA2,NEXJOB  ;JOBA2 IS NEXT
1924 010456 012767 000000 177150      MOV     #0,NEXJOB+2    ;NO ERROR RECOVERY
1925 010464 000002              RTI
1926
1927      ;*****
1928      ;ROUTINE A, SEGMENT 2
1929      ;CHECKS TRANSMITTED DATA WITH RECEIVED
1930      ;*****
1931
1932 010466 012700 011462      JOBA2: MOV     #LISTA,R0  ;XMIT BUFFER LIST
1933 010472 012701 011474      MOV     #LISTA1,R1     ;REC BUFFER LIST
1934 010476 004767 000652              JSR     R7,BUFCHK      ;COMPARE THE TWO BUFFERS
1935 010502 042777 000100 170314      BIC     #IE,@DRST      ;GLITCH INTERRUPT
1936 010510 012777 000100 170306      MOV     #IE,@DRST
1937 010516 012767 010316 177106      MOV     #JOBA0,NEXJOB  ;REPEAT JOBA0
1938 010524 012767 000000 177102      MOV     #0,NEXJOB+2
1939 010532 005267 177102      INC     JBCNT          ;ADVANCE COUNT
1940 010536 000002              RTI
1941
1942      ;*****
1943      ;SLAVE'S INTERRUPT SERVICE ROUTINES
1944      ;
1945      ;JOB1: IGNORE FIRST READY INTERRUPT

```



```

1946                                     ;:*****
1947                                     ;:*****
1948 010540 012767 000000 177064 SJOB1: MOV    #0,NEXJOB      ;NO MORE READY INTERRUPTS
1949 010546 012767 010556 177060      MOV    #SJOB2,NEXJOB+2 ;UNTIL ATTN INTERRUPT
1950 010554 000002      RTI
1951
1952                                     ;:*****
1953                                     ;:      JOB2: WAIT FOR COMMAND
1954                                     ;:*****
1955
1956 010556 032777 004000 170240 SJOB2: BIT     #DSTATC,@DRST  ;TEST FOR INTER
1957 010564 001002      BNE     SJOB2A
1958 010566 104000      HLT
1959 010570 000000      HALT
1960 010572 005001      SJOB2A: CLR     R1                ;ERROR OTHER THAN DSTATC
1961 010574 016702 170224      MOV     DRST,R2          ;SET UP FOR PARAMETERS
1962 010600 012703 000010      MOV     #FNCT3,R3        ;R2 IS STATUS ADDRESS
1963 010604 012704 004000      MOV     #DSTATC,R4       ;R3 IS FUNCTION BIT 3
1964 010610 016705 170212      MOV     DRDB,R5          ;R4 IS INTERRUPT BIT
1965 010614 012700 011510      MOV     #LISTB+2,R0      ;R5 IS DATA BUFFER ADDRESS
1966 010620 004767 000122      JSR     R7,HNDSHK        ;STORE PARAMETERS HERE STARTING WITH WORD COUNT
1967 010624 012700 011506      MOV     #LISTB,R0        ;GET PARAMETERS
1968 010630 016010 000004      MOV     4(R0),(R0)       ;R0 IS TOP OF LIST
1969 010634 066710 170202      ADD     INBUF,(R0)       ;MOVE OFFSET TO TOP
1970 010640 012077 170156      MOV     (R0)+,@DRBA     ;TOP OF LIST IS BUFFER START + OFFSET
1971 010644 012077 170150      MOV     (R0)+,@DRWC     ;SET UP BUS ADDRESS
1972 010650 005077 170150      CLR     @DRST          ;SET UP WORD COUNT
1973 010654 032777 002000 170142      BIT     #DSTATB,@DRST ;CLEAR ALL FUNCTION BITS
1974 010662 001405      BEQ     SJOB2C                ;WHICH DIRECTION
1975 010664 032777 000400 170132 SJOB2B: BIT     #CYCLE,@DRST ;BRANCH IF RECIEVE (LEAVE FNCT1 CLEAR FOR DATI'S)
1976 010672 001774      BEQ     SJOB2B                ;WAIT FOR MASTER TO SET CYCLE
1977 010674 000412      BR      SJOB2D                ;BRANCH IF NOT SET
1978 010676 012700 011506      SJOB2C: MOV     #LISTB,R0 ;GO DO THE COMAND
1979 010702 004767 000356      JSR     R7,BLUSH        ;BLUSH THE BUFFER
1980 010706 052777 000004 170110      BIS     #FNCT2,@DRST ;SET FNCT2 FOR DATO'S
1981 010714 042777 000400 170102      BIC     #CYCLE,@DRST ;SET FNCT2 FOR DATO'S
1982 010722 052777 000101 170074 SJOB2D: BIS     #IE!GO,@DRST ;CLEAR CYCLE
1983 010730 012767 010540 176674      MOV     #SJOB1,NEXJOB ;EXECUTE COMMAND AND INTERRUPT WHEN DONE
1984 010736 012767 010556 176670      MOV     #SJOB2,NEXJOB+2 ;IGNORE READY INTERRUPT
1985 010744 000002      RTI
1986
1987                                     ;:*****
1988                                     ;:      SLAVE ROUTINE TO ACCEPT PARAMETERS FROM MASTER
1989                                     ;:*****
1990
1991 010746 011515      HNDSHK: MOV     (R5),(R5)        ;ECHO PARAMETER
1992 010750 011520      MOV     (R5),(R0)+            ;STORE PARAMETER
1993 010752 050312      BIS     R3,(R2)              ;REPLY WITH FNCT3
1994 010754 030412      HNDSH1: BIT     R4,(R2)        ;WAIT FOR ATTN TO DROP
1995 010756 001376      BNE     HNDSH1
1996 010760 040312      BIC     R3,(R2)              ;DROP FNCT3
1997 010762 005201      INC     R1                  ;CHECK NUMBER
1998 010764 020167 176640      CMP     R1,NWRDXF
1999 010770 002401      BLT     HNDSH2                ;BRANCH IF NOT DONE YET
2000 010772 000207      RTS     R7
2001 010774 030412      HNDSH2: BIT     R4,(R2)        ;WAIT FOR NEXT WORD

```

Address	BEQ	BR	WDSH2	WDSHK	Comments
2002	010776	001776			;BRANCH IF ATTN CLEAR
2003	011000	000762			;GET ANOTHER PARAMETER
2004					
2005					
2006					*****
2007					MASTER TRANSMIT ROUTINE
2008					ENTER WITH TRANSFER LIST IN R0
2009					*****
2010	011002	005777	170016		MXMIT: TST @DRST ;MAKE SURE ERROR IS CLEAR
2011	011006	100002			BPL MXMIT1 ;AND READY IS SET
2012	011010	104000			HLT
2013	011012	000000			HALT ;ERROR IS SET
2014	011014	105777	170004		MXMIT1: TSTB @DRST
2015	011020	100402			BMI MXMIT2
2016	011022	104000			HLT
2017	011024	000000			HALT ;READY NOT SET
2018	011026	012777	000000	167770	MXMIT2: MOV #0,@DRST ;SET UP FUNCTION FOR DATI'S
2019	011034	004767	000102		JSR R7,PRMXFR ;TRANSFER PARAMETERS OF LIST WHOSE ADDRESS IS IN R0
2020	011040	042777	000400	167756	BIC #CYCLE,@DRST ;MAKE SURE CYCLE IS CLEAR
2021	011046	052777	000101	167750	BIS #IE!GO,@DRST ;EXECUTE COMMAND AND INTERRUPT WHEN DONE
2022	011054	000207			RTS R7
2023					
2024					*****
2025					MASTER RECEIVE ROUTINE
2026					ENTER WITH TRANSFER LIST IN R0
2027					*****
2028	011056	005777	167742		MREC: TST @DRST ;MAKE SURE ERROR IS CLEAR
2029	011062	100002			BPL MREC1 ;AND READY SET.
2030	011064	104000			HLT
2031	011066	000000			HALT ;ERROR SET
2032	011070	105777	167730		MREC1: TSTB @DRST
2033	011074	100402			BMI MREC2
2034	011076	104000			HLT
2035	011100	000000			HALT ;READY CLEAR
2036	011102	012777	000004	167714	MREC2: MOV #FNCT2,@DRST ;SET UP FUNCTION FOR DATO'S
2037	011110	004767	000026		JSR R7,PRMXFR ;TRANSFER PARAMETERS OF LIST WHOSE ADDRESS IS IN R0
2038	011114	032777	002000	167702	MREC3: BIT #DSTATB,@DRST ;WAIT FOR SLAVE TO CLEAR DIRECTION
2039	011122	001374			BNE MREC3 ;BRANCH IF SET
2040	011124	042777	000400	167672	BIC #CYCLE,@DRST ;CLEAR CYCLE
2041	011132	052777	000101	167664	BIS #IE!GO,@DRST ;EXECUTE COMMAND AND INTERRUPT WHEN DONE.
2042	011140	000207			RTS R7
2043					
2044					*****
2045					ROUTINE TO TRANSFER AND CHECK PARAMETERS UNDER FLAG CONTROL
2046					ENTER WITH R0 POINTING TO TRANSFER LIST
2047					*****
2048					
2049	011142	012077	167654		PRMXFR: MOV (R0)+,@DRBA ;FIRST WORD IN LIST IS ADDRESS
2050	011146	011077	167646		MOV (R0),@DRWC ;SECOND WORD IN LIST IS WORD COUNT
2051	011152	005001			CLR R1 ;R1 COUNTS PARAMETERS TRANSFERRED
2052	011154	016702	167644		MOV DRST,R2 ;R2 IS THE STATUS ADDRESS
2053	011160	012703	000010		MOV #FNCT3,R3 ;R3 USED FOR FUNCTION BIT 3
2054	011164	012704	004000		MOV #DSTATC,R4 ;R4 USED FOR INTERRUPT BIT
2055	011170	016705	167632		MOV DRDB,R5 ;R5 IS THE DATA BUFFER ADDRESS
2056	011174	011015			PRMXF1: MOV (R0),(R5) ;SET UP DRDB WITH PARAMETER
2057	011176	050312			BIS R3,(R2) ;CALL SLAVE'S ATTN

2058	011200	030412		PRMXF2: BIT	R4,(R2)	;WAIT FOR REPLY
2059	011202	001776		BEQ	PRMXF2	;BRANCH IF ATTN CLEAR
2060	011204	022015		CMP	(R0)+(R5)	;COMPARE PARAMETER SENT WITH SLAVE'S ECHO
2061	011206	001402		BEQ	PRMXF3	;BRANCH IF EQUAL
2062	011210	104000		HLT		
2063	011212	000000		PRMXF3: BIC	R3,(R2)	;PARAMETER DID NOT ECHO
2064	011214	040312		PRMXF4: BIT	R4,(R2)	;DROP SLAVE'S ATTN
2065	011216	030412		BNE	PRMXF4	;WAIT FOR REPLY
2066	011220	001376		INC	R1	;BRANCH IF ATTN SET
2067	011222	005201		CMP	R1,MURDXF	;ADVANCE PARAMETER COUNT
2068	011224	020167	176400	BLT	PRMXF1	;ALL PARAMETER XFERRED?
2069	011230	002761		RTS	R7	;BRANCH IF NOT DONE
2070	011232	000207				;RETURN WHEN ALL PARAMETERS TRANSFERRED AND CHECKED.
2071						
2072						
2073						
2074						
2075						
2076						
2077	011234	005005		FLUSH: CLR	R5	;SET R5 TO ZIP
2078	011236	004767	000030	JSR	R7,BSETUP	;SET UP REGISTERS
2079	011242	004767	000004	FLUSH1: JSR	R7,BUFPUT	;STORE ITEM IN BUFFER
2080	011246	002775		BLT	FLUSH1	
2081	011250	000436		BR	BUFOUT	;STOP WHEN BUFFER FULL
2082						
2083	011252	010521		BUFPUT: MOV	R5,(R1)+	;PUT R5 INTO BUFFER
2084	011254	060503		ADD	R5,R3	;INCLUDE IN CHECKSUM
2085	011256	005504		ADC	R4	
2086	011260	005202		INC	R2	;ADVANCE WORD COUNT
2087	011262	000207		RTS	R7	;RETURN WITH STATUS SET
2088						
2089						
2090						
2091						
2092						
2093	011264	012705	177777	BLUSH: MOV	#-1,R5	;SET R5 TO ALL ONE'S
2094	011270	000762		BR	FLUSH+2	
2095						
2096						
2097	011272	012001				
2098	011274	012002				
2099	011276	005720				
2100	011300	005003				
2101	011302	005004				
2102	011304	000207				
2103						
2104						
2105						
2106						
2107						
2108						
2109	011306	004767	177760	SETBUF: JSR	R7,BSETUP	;SET UP REGISTERS
2110	011312	012705	000001	SETBF1: MOV	#1,R5	;SET UP R5 WITH START PATTERN
2111	011316	004767	177730	SETBF2: JSR	R7,BUFPUT	;NUMBER TO BUFFER
2112	011322	002011		BGE	BUFOUT	
2113	011324	005105		COM	R5	

2114 011326 004767 177720
 2115 011332 002005
 2116 011334 005105
 2117 011336 000241
 2118 011340 006105
 2119 011342 103763
 2120 011344 000764
 2121 011346 010320
 2122 011350 010420
 2123 011352 000207

```

JSR      R7,BUFPUT      ;COMPLEMENT OF NUMBER TO BUFFER
BGE      BUFOUT
COM      R5              ;MOVE PATTERN ONE BIT
CLC                      ;POSITION TO THE LEFT
ROL      R5
BCS      SETBF1          ;IF CARRY SET, START PATTERN OVER
BR       SETBF2
BUFOUT:  MOV      R3,(R0)+ ;MOVE LOW CHECK TO LIST
         MOV      R4,(R0)+ ;MOVE HIGH CHECK TO LIST
         RTS      R7      ;RETURN
  
```

```

;*****
;ROUTINE TO COMPARE TWO BUFFERS
;BUFFER LIST #1 IS IN R1
;BUFFER LIST #2 IS IN R0. #2 CHECKED FOR PROPER CHECKSUM
;*****
  
```

2130
 2131 011354 010046
 2132 011356 010146
 2133 011360 004767 000036
 2134 011364 012600
 2135 011366 012001
 2136 011370 012002
 2137 011372 012600
 2138 011374 012003
 2139 011376 020220
 2140 011400 003402
 2141 011402 104000
 2142 011404 000000
 2143
 2144 011406 022123
 2145 011410 001401
 2146 011412 104000
 2147 011414 005202
 2148 011416 002773
 2149 011420 000207

```

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

```

BUFCK1:  CMP      (R1)+,(R3)+ ;COMPARE BUFFERS
         BEQ      BUFCK2
         HLT                      ;DATA ERROR
BUFCK2:  INC      R2           ;ADVANCE COUNT
         BLT      BUFCK1
         RTS      R7         ;RETURN; STACK IS CLEAR
  
```

```

;*****
;ROUTINE TO CHECK SUM OF BUFFER
;ENTER WITH R0 POINTING TO TRANSFER LIST
;*****
  
```

2150
 2151
 2152
 2153
 2154
 2155 011422 004767 177644
 2156 011426 004767 000016
 2157 011432 002775
 2158 011434 020320
 2159 011436 001003
 2160 011440 020420
 2161 011442 001001
 2162 011444 000207
 2163 011446 104000
 2164
 2165 011450 012105
 2166 011452 060503
 2167 011454 005504
 2168 011456 005202
 2169 011460 000207

```

CHKSUM:  JSR      R7,BSETUP ;SET UP REGISTERS
CHKSM1:  JSR      R7,GETBUF ;GET ITEM FROM BUFFER
         BLT      CHKSM1    ;BRANCH IF NOT DONE
         CMP      R3,(R0)+ ;COMPARE LOW ORDER CHECKS
         BNE      CHKSM2
         CMP      R4,(R0)+ ;COMPARE HIGH ORDER CHECKS
         BNE      CHKSM2
         RTS      R7      ;RETURN IF CHECKSUM OK.
CHKSM2:  HLT                      ;ORIGINAL BUFFER CHECKSUM DOES NOT AGREE WITH PRESENT
GETBUF:  MOV      (R1)+,R5 ;GET ITEM OUT OF BUFFER
         ADD      R5,R3    ;ADD TO CHECKSUM
         ADC      R4
         INC      R2      ;ADVANCE WORD COUNT
         RTS      R7
  
```

2170						
2171	011462	014224				
2172	011464	177605				
2173	011466	000000				
2174	011470	000000				
2175	011472	000000				
2176						
2177	011474	000000				
2178	011476	000000				
2179	011500	000000				
2180	011502	000000				
2181	011504	000000				
2182						
2183	011506	014224				
2184	011510	000000				
2185	011512	000000				
2186	011514	000000				
2187	011516	000000				
2188						
2189						
2190						
2191						
2192						
2193						
2194						
2195						
2196	011520	004767	000672			
2197	011524	037727	167264	020000		
2198	011532	001067				
2199	011534	012667	000204			
2200	011540	012667	000202			
2201	011544	024646				
2202	011546	012777	000215	167312		
2203	011554	105777	167304			
2204	011560	100375				
2205	011562	012777	000212	167276		
2206	011570	105777	167270			
2207	011574	100375				
2208	011576	010267	000134			
2209	011602	010367	000132			
2210	011606	010467	000130			
2211	011612	016702	000126			
2212	011616	004767	000126			
2213	011622	012777	000240	167236		
2214	011630	105777	167230			
2215	011634	100375				
2216	011636	016702	000104			
2217	011642	004767	000102			
2218	011646	012777	000240	167212		
2219	011654	105777	167204			
2220	011660	100375				
2221	011662	017702	167136			
2222	011666	004767	000056			
2223	011672	016702	000040			
2224	011676	016703	000036			
2225	011702	016704	000034			

LISTA:	XINBUF	:START OF XMIT BUFFER
	-123.	:WORD COUNT
	0	:OFFSET
	0	:CHECKSUM LOW
	0	:CHECKSUM HIGH
LISTA1:	0	:START OF REC BUFFER
	0	:WORD COUNT
	0	:OFFSET
	0	:CHECKSUM LOW
	0	:CHECKSUM HIGH
LISTB:	XINBUF	:SLAVE'S ECHO BUFFER
	0	
	0	
	0	
	0	


```

*****
: ENTERED WITH SYSTEM TRAP CALL(HLT)
: PRINT OUT THE ERROR PC AND STATUS REGISTER
*****

```


PRINT:	JSR	%7,CKSWR	
	BIT	@SR,#20000	:TEST FOR INHIBIT PRINT OUT
	BNE	1\$:IF SO, BRANCH OVER PRINT
	MOV	(6)+,SAVPC	:PC OF FAILING ROUTINE
	MOV	(6)+,SAVCC	:CC OF ERROR CONDITION
	CMP	-(6),-(6)	:REPOSITION THE STACK
	MOV	#215,@TPB	:CR
	TSTB	@TPS	
	BPL	.-4	
	MOV	#212,@TPB	:LINE FEED
	TSTB	@TPS	
	BPL	.-4	
	MOV	%2,SAVR2	:SAVE R2
	MOV	%3,SAVR3	:SAVE R3
	MOV	%4,SAVR4	:SAVE R4
	MOV	SAVPC,%2	
	JSR	%7,PRTAB	:PRINT OCTAL NUMBER
	MOV	#240,@TPB	
	TSTB	@TPS	:SPACE BETWEEN WORDS
	BPL	.-4	
	MOV	SAVCC,%2	
	JSR	%7,PRTAB	:PRINT OCTAL NUMBER
	MOV	#240,@TPB	:PRINT SPACE
	TSTB	@TPS	:PRINTER DONE
	BPL	.-4	:BRANCH WHEN NOT DONE
	MOV	@DRST,%2	:GET DR11 STATUS
	JSR	%7,PRTAB	:PRINT OCTAL NUMBER
	MOV	SAVR2,%2	
	MOV	SAVR3,%3	
	MOV	SAVR4,%4	

2226	011706	004767	000504		JSR	%7,CKSWR	
2227	011712	005777	167076	18:	TST	@SR	;CHECK SR FOR HALT SWITCH
2228	011716	100001			BPL	.+4	
2229	011720	000000			HALT		;HALT ON ERROR UP IN SWR
2230	011722	023737	000042	003046	CMP	@#42,@#46	;ARE WE IN ACT11 AUTO MODE?
2231	011730	001001			BNE	.+4	;BRANCH ON NO
2232	011732	000000			HALT		;HALT ON ERROR IF IN ACT11 AUTO MODE
2233	011734	000002			RTI		;RETURN TO MAINLINE
2234	011736	000000		SAVR2:	0		
2235	011740	000000		SAVR3:	0		
2236	011742	000000		SAVR4:	0		
2237	011744	000000		SAVPC:	0		
2238	011746	000000		SAVCC:	0		
2239							
2240	011750	005067	000260		CLR	BINCT	
2241	011754	005067	000252		CLR	WGTCT	
2242	011760	012704	012240		MOV	#LIST,%4	;GET LIST ADDRESS
2243	011764	142777	000177	167072	BICB	#177,@TPS	;CLR INT FLAG
2244	011772	012767	000005	000236	MOV	#5,ASCNT	
2245	012000	012767	000007	000220	MOV	#7,SEVEN	
2246	012006	012767	000001	000214	MOV	#1,DECM	
2247	012014	105777	167044		WAIT1:	TSTB	
2248	012020	100375			BPL	WAIT1	
2249	012022	005702			TST	%2	
2250	012024	100404			BMI	MINUS	;NEG SIGN PRINT 1
2251	012026	012777	000260	167032	MOV	#260,@TPB	;POS SIGN PRINT 0
2252	012034	000403			BR	START	
2253	012036	012777	000261	167022	MINUS:	MOV	#261,@TPB
2254	012044	016703	000156		START:	MOV	SEVEN,%3
2255	012050	010267	000150			MOV	%2,TOODLE
2256	012054	005167	000144			COM	TOODLE
2257	012060	046703	000140			BIC	TOODLE,%3
2258	012064	001410				BEQ	WRTOC
2259	012066	066767	000136	000136	MKNUM:	ADD	DECM, WGTCT
2260	012074	005267	000134			INC	BINCT
2261	012100	026703	000126			CMP	WGTCT,%3
2262	012104	001370				BNE	MKNUM
2263	012106	062767	000260	000120	WRTOC:	ADD	#260,BINCT
2264	012114	016724	000114			MOV	BINCT,(4)+
2265	012120	066767	000102	000102		ADD	SEVEN,DECM
2266	012126	005067	000100			CLR	WGTCT
2267	012132	005067	000076			CLR	BINCT
2268	012136	005367	000074			DEC	ASCNT
2269	012142	001410				BEQ	XLIST
2270	012144	012703	000003			MOV	#3,%3
2271	012150	066767	000052	000050	MOADD:	ADD	SEVEN,SEVEN
2272	012156	005303				DEC	%3
2273	012160	001373				BNE	MOADD
2274	012162	000730				BR	START
2275	012164	012767	000005	000044	XLIST:	MOV	#5,ASCNT
2276	012172	105777	166666		WAIT2:	TSTB	
2277	012176	100375				BPL	WAIT2
2278	012200	014477	166662			MOV	-(4),@TPB
2279	012204	005367	000026			DEC	ASCNT
2280	012210	001401				BEQ	HDFHM
2281	012212	000767				BR	WAIT2
							;FINISH PRINTING GET NXT NUM


```

2282 012214 105777 166644      HDFHM: TSTB @TPS
2283 012220 100375              BPL      -4
2284 012222 000207              RTS      %7      ;HEAD FOR HOME
2285 012224 000000      TOODLE: 0
2286 012226 000000      SEVEN: 0
2287 012230 000000      DECML: 0
2288 012232 000000      WGTCT: 0
2289 012234 000000      BINCT: 0
2290 012236 000000      ASCNT: 0
2291 012240 000000      LIST: 0
2292 012242 000000              0
2293 012244 000000              0
2294 012246 000000              0
2295 012250 000000              0
2296
2297      ;*****
2298      ;      SCOPE LOOP ROUTINE ENTERED BY USER TRAP
2299      ;*****
2300
2301 012252 004767 000140      SCOPEA: JSR      %7,CKSWR
2302 012256 032777 040000 166530      BIT      #40000,@SR
2303 012264 001003              BNE      SCOPEB      ;SCOPE, BIT IS A ONE
2304 012266 011667 000106      MOV      @%6,RETURN      ;NO - SAVE PC FOR NEXT TIME
2305 012272 000002              RTI      ;RETURN IN SEQUENCE
2306 012274 022606      SCOPEB: CMP      (6)+,%6      ;REPOSITION THE STACK
2307 012276 012677 166514      MOV      (6)+,@PSW
2308 012302 000177 000072      JMP      @RETURN      ;SCOPE RETURN
2309
2310      ;*****
2311      ;      SCOPE OR/AND ITERATION LOOP FOR EACH TEST 4000 TIMES
2312      ;*****
2313
2314
2315 012306 004767 000104      SCOPEC: JSR      %7,CKSWR
2316 012312 032777 040000 166474      BIT      #40000,@SR      ;TEST SR FOR SCOPE
2317 012320 001365              BNE      SCOPEB      ;YES SCOPE
2318 012322 005767 166546      TST      PASCNT      ;FIRST PASS (PASCNT=0) ?
2319 012326 001415              BEQ      SCOPEG      ;BR IF YES, INHIBIT ITERATIONS
2320 012330 004767 000062      JSR      %7,CKSWR
2321 012334 032777 004000 166452      BIT      #4000,@SR      ;TEST FOR ITERATION
2322 012342 001007              BNE      SCOPEG      ;INHIBIT ITERATION
2323 012344 026767 000026 000022      CMP      SCOPEF,ICOUNT
2324 012352 001403              BEQ      SCOPEG      ;EXIT - DONE
2325 012354 005267 000016      INC      SCOPEF      ;INCREMENT COUNT
2326 012360 000745              BR      SCOPEB      ;LOOP SOME MORE
2327 012362 005067 000010      SCOPEG: CLR      SCOPEF      ;CLEAR COUNT
2328 012366 011667 000006      MOV      @%6,RETURN      ;SAVE SCOPE RETURN POINTER
2329 012372 000002              RTI      ;RETURN INLINE-NEXT TEST
2330 012374 004000      ICOUNT: 4000
2331 012376 000000      SCOPEF: 0      ;COUNT LOCATION FOR ITERATION LOOP
2332 012400 001174      RETURN: BEGIN      ;ADDRESS OF LAST TEST
2333              .EVEN
2334 012402 000167 165572      JMP      200
2335
2336
2337      ;*****

```

```

2338      :      CHECK SWITCH REGISTER ROUTINE. CHECKS FOR ^G TO ALLOW CHANGING
2339      :      OF LOC. 176.
2340      :      *****
2341      :      TEMPST: .WORD 0
2342      :      COUNT: .WORD 0
2343      :      RDSW: .WORD 0
2344      :      TIB: .WORD 0
2345
2346      012406 000000 000176 166370 CKSWR: CMP #SWREG,SR ;SOFTWARE SWITCH REGISTER PRESENT
2347      012424 001133 BNE OUT
2348      012426 105777 166426 TSTB @TKS ;YES, WAIT FOR
2349      012432 100130 BPL OUT ;READY, GET CHARACTER
2350      012434 017767 166422 177752 MOV @TKB,TIB ;AND STRIP OFF
2351      012442 042767 177600 177744 BIC #177600,TIB ;THE GARBAGE
2352      012450 022767 000007 177736 CMP #7,TIB ;IS IT A ^G>
2353      012456 001116 BNE OUT
2354      012460 012702 012770 MOV #SCNTG,%2
2355      012464 004767 000426 JSR PC,TTOUT
2356      012470 012702 013002 CNTLU: MOV #SMSWR,%2
2357      012474 004767 000416 JSR PC,TTOUT
2358      012500 017702 166310 MOV @SR,%2
2359      012504 004767 177240 JSR %7,PRTAB
2360      012510 012702 013012 MOV #SMNEW,%2
2361      012514 004767 000376 JSR PC,TTOUT
2362      012520 005037 012406 SREAD: CLR @TEMPST
2363      012524 005067 177656 CLR TEMPST
2364      012530 012767 000007 177652 MOV #7,COUNT
2365      012536 004767 000154 1$: JSR PC,TTIN ;GO READ A CHARACTER
2366      012542 042767 177600 177644 BIC #177600,TIB ;STRIP OFF GARBAGE
2367      012550 122767 000025 177636 CMPB #25,TIB ;IS IT A ^U?
2368      012556 001001 BNE 2$ ;BRANCH IF NOT
2369      012560 000743 3$: BR CNTLU ;START OVER
2370      012562 122767 000015 177624 2$: CMPB #15,TIB ;IS IT A <CR>?
2371      012570 001011 BNE 4$ ;BRANCH IF NOT
2372      012572 012702 012776 MOV #SCRLF,%2
2373      012576 004767 000314 JSR %7,TTOUT
2374      012602 022767 000007 177600 CMP #7,COUNT ;WAS IT FIRST CHARACTER
2375      012610 001036 BNE 7$ ;CHANGE SWR IF NOT FIRST ONE
2376      012612 000440 8$: BR OUT ;GET OUT
2377      012614 122767 000060 177572 4$: CMPB #60,TIB
2378      012622 003004 BGT 5$
2379      012624 122767 000067 177562 CMPB #67,TIB
2380      012632 002005 BGE 6$
2381      012634 012702 013023 5$: MOV #SQUEST,%2
2382      012640 004767 000252 JSR PC,TTOUT
2383      012644 000745 BR 3$ ;START OVER IF NOT LEGAL CHARACTER
2384      012646 006367 177534 6$: ASL TEMPST
2385      012652 006367 177530 ASL TEMPST
2386      012656 006367 177524 ASL TEMPST
2387      012662 142767 000060 177524 BICB #60,TIB ;GET NITTY-GRITTY
2388      012670 156767 177520 177510 BISB TIB,TEMPST
2389      012676 005367 177506 DEC COUNT ;ONLY WANT 6 DIGITS
2390      012702 001754 BEQ 5$
2391      012704 000714 BR 1$
2392      012706 016777 177474 166100 7$: MOV TEMPST,@SR ;CHANGE SWITCH REGISTER CONTENTS
2393      012714 000207 OUT: RTS ;RETURN TO PROGRAM

```

```

2394
2395
2396
2397
2398
2399
2400 012716 005077 166136
2401 012722 005077 166134
2402 012726 005067 177462
2403 012732 005277 166122
2404 012736 105777 166116
2405 012742 100375
2406 012744 017767 166112 177442
2407 012752 105777 166106
2408 012756 100375
2409 012760 116777 177430 166100
2410
2411 012766 000207
2412 012770 057137 020107 000046
2413 012776 020137 000046
2414 013002 051537 051127 020075
2415 013010 000046
2416 013012 020040 042516 036527
2417 013020 023040 000
2418 013023 137 020077 023137
2419 013030 000
2420 013031 137 047105 020104
2421 013036 040520 051523 020040
2422 013044 023040 000
2423 013047 137 041440 042132
2424 013054 041122 030107 042040
2425 013062 030522 026461 027502
2426 013070 040504 030461 041055
2427 013076 047040 051120 042040
2428 013104 040511 020040 057440
2429 013112 000046
2430
2431
2432 013114 000000
2433
2434
2435
2436
2437
2438
2439
2440 013116 105712
2441 013120 001403
2442 013122 122712 000046
2443 013126 001005
2444 013130 042777 000100 165726
2445 013136 005002
2446 013140 000207
2447 013142 122712 000137
2448 013146 001411
2449 013150 122712 000041

;*****
; TTY READ SUBROUTINE *****
;*****

TTIN: CLR @TKS
      CLR @TKB
      CLR TIB
      INC @TKS
TTIN1: TSTB @TKS
      BPL TTIN1
      MOV @TKB,TIB
TTIN2: TSTB @TPS
      BPL TTIN2
      MOVB TIB,@TPB
      RTS %7
$CNTG: .ASCIZ ' ^G &'
$CRLF: .ASCIZ ' - &'
$MSWR: .ASCIZ ' -SWR= &'
$MNEW: .ASCIZ ' NEW= &'
$QUEST: .ASCIZ ' _? _&'
$ENPAS: .ASCIZ ' _END PASS &'
$TITLE: .ASCIZ ' _ CZDRBG DR11-B/DA11-B NPR DIA _&'

.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
      RTS %7 ;CLEAR POINTER TO CHARACTER
      .EMPTY: CMPB #' ,(2) ;RETURN
      BEQ .RET ;CRLF CHAR?
      CMPB #' ,(2) ;CHECK FOR RETURN TERMINATOR

```

2450	013154	001414			BEO	.REST	
2451	013156	105777	165702		1\$: TSTB	@TPS	
2452	013162	100375			BPL	1\$	
2453	013164	112277	165676		MOVB	(2)+, @TPB	;TYPE CHARACTER
2454	013170	000752			BR	TTOUT	
2455	013172	005202			.RET: INC	%2	
2456	013174	010267	000020		MOV	%2, .SAV	;SET JP NEW POINTER
2457	013200	012702	013214		MOV	#.RETR, %2	
2458	013204	000767			BR	.RET-6	
2459	013206	016702	000006		.REST: MOV	.SAV, %2	
2460	013212	000741			BR	TTOUT	
2461							
2462	013214	015	012	041	.RETR: .BYTE	15, 12, '	
2463		013220				.EVEN	
2464	013220	000000			.SAV: 0		
2465		014222				.=. +1000	
2466	014222	000000			BUFF: 0		;FOR STACK POINTER 100 LOCATIONS
2467	014224	014224			XINBUF: .		
2468		015226				.=. +1000	
2469	015226	015226			XCHKBU: .		
2470		000001				.END	

SEQ 0046

[illegible]

DATINB	006220	1464	1468#											
DATGCK	007320	1216	1254	1499	1638#									
DATONB	006360	1492	1498#											
DECMIL	012230	2246*	2259	2265*	2287#									
DICMEM	001040	537#	1261	1420										
DONE	005722	1407	1412#											
DRBA	001022	530#	584*	617*	620	643*	644	859*	860	863*	864	887*	888	891*
		913*	942*	974*	1010*	1047*	1112*	1117*	1133	1149*	1165	1188*	1209*	1227*
		1247*	1261*	1289*	1311*	1385	1397*	1420*	1431*	1456*	1484*	1507*	1549*	1568*
		1617	1970*	2049*										
DRDB	001026	532#	588*	1118*	1136	1151*	1189*	1195	1210*	1228*	1234	1248*	1322*	1325
		1334*	1335	1343*	1344	1347*	1348	1358*	1359	1362*	1457*	1469	1485*	1550*
		1569*	1964	2055										
DRINL	001032	534#	1153	1191	1212	1230	1250	1263	1291	1459	1487	1517	1552	1571
		1845	1847	1849										
DRINS	010216	1844	1862#											
DRINV	001034	535#	909*	938*	970*	1006*	1018*	1025*	1043*	1055*	1061*	1120*	1152*	1190*
		1211*	1229*	1249*	1262*	1290*	1421*	1430*	1458*	1486*	1515*	1551*	1570*	1634*
		1843												
DRINO	010230	1863	1866#											
DRIN1	010252	1869	1872#											
DRIN2	010274	1875	1878#											
DRIN3	010262	1867	1874#											
DRST	001024	531#	586*	656*	657	660*	661	673*	674	677*	678	690*	691	694*
		695	703*	704	707*	708	716*	717	720*	721	729*	730	733*	734
		742*	743	746*	747	755*	756	759*	760	768*	770	773	776	779
		782	785	788	791	794*	795	798	801	804	807	811	814	817
		827*	829	836	902	906	914*	920*	923*	931	935	943*	949*	952*
		961	965	975*	981*	987*	997	1001	1011*	1017*	1023*	1036	1040	1048*
		1054*	1059*	1069*	1070	1073*	1074	1077	1080*	1081*	1082	1085	1088*	1089*
		1090	1093	1096*	1101	1103	1106	1108*	1109	1113	1123*	1142*	1155*	1175*
		1193*	1214*	1232*	1252*	1265*	1268*	1269	1272	1275	1278*	1293*	1296*	1298
		1301	1304*	1305	1308	1373	1376	1398	1401*	1402	1406	1424*	1427	1432*
		1434*	1440	1461*	1489*	1510*	1520*	1524	1554*	1573*	1604*	1605	1608	1651*
		1652	1655	1764*	1780*	1866	1874	1935*	1936*	1956	1961	1972*	1973	1975
		1980*	1981*	1982*	2010	2014	2018*	2020*	2021*	2028	2032	2036*	2038	2040*
		2041*	2052	2221										
DRVS	001030	533#	1121*	1153*	1191*	1212*	1230*	1250*	1263*	1291*	1422*	1459*	1487*	1517*
		1552*	1571*	1634	1635*									
DRWC	001020	529#	582*	601*	604	632*	633	846*	847	850*	851	874*	875	878*
		912*	941*	973*	1009*	1046*	1116*	1130	1148*	1162	1187*	1208*	1226*	1246*
		1260*	1288*	1396*	1419*	1433*	1455*	1482*	1519*	1548*	1567*	1611	1971*	2050*
		1738#												
DSTATB=	002000	1739#	1973	2038										
DSTATC=	004000	1740#	1866	1956	1963	2054								
END	007416	1583	1663#											
END1	007446	1671#												
ERRA	001252	580	590#											
ERRCHK	007366	1129	1161	1651#										
ERRDO	005776	1421	1427#											
ERRDO1	006060	1430	1440#											
ERROR =	100000	1744#												
FLUSH	011234	1920	2077#	2094										
FLUSH1	011242	2079#	2080											
FNCCNT	001070	549#	1511*	1527	1530*	1531								
FNCT1 =	000002	1730#												

[illegible]

[illegible]

[illegible]

MACY11 52A(1052) 01-APP-80 11:29 M 4 PAGE 53
CROSS REFERENCE TABLE -- USER SYMBOLS

[illegible]

MACY11 30A(1052) 01-APR-80 11:29 N 4 PAGE 55
CROSS REFERENCE TABLE -- MACRO NAMES

[illegible]

CZDRBG DR11-B/DA11-B *PR DIA MACY11 30A(1052) 01-APR-80 11:29 ^{B 5} PAGE 56
CZDRBG.P11 01-APR-80 11:25 CROSS REFERENCE TABLE -- MACRO NAMES

SEQ 0053

.SRAND 1#
.SRDE 1#
.SRDOC 1#
.SREAD 1#
.SR2AZ 1#
.SSAVE 1#
.SSB2D 1#
.SSB2O 1#
.SSCOP 1#
.SSIZE 1#
.SSUPR 1#
.STRAP 1#
.STYPB 1#
.STYPD 1#
.STYPE 1#
.STYPO 1#
.S4OCA 1#
.1170 1#

. ABS. 015230 000

ERRORS DETECTED: 0

CZDRBG.BIN,CZDRBG.LST/CRF/SOL/NL:TOC=CZDRBG.SML,CZDRBG.P11
RUN-TIME: 30 39 2 SECONDS
RUN-TIME RATIO: 202/72=2.7
CORE USED: 31K (61 PAGES)