

DR11-B,
DA11-B

DR11B/DA11-B
NPR DIA
CZDRBHO

AH-8644H-MC
 FICHE 1 OF 1

OCT 1983
COPYRIGHT © 71-83
MADE IN USA



.REM-%

IDENTIFICATION

PRODUCT CODE: AC-8643H-MC
PRODUCT NAME: CZDRBHO DR11-B/DA11-B NPR DIA
PRODUCT DATE: MAY 1983
MAINTAINER: DIAGNOSTIC ENGINEERING

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) 1971,1983 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 CABLES.

NOTE: WITH OR WITHOUT HARDWARE SWITCH REGISTER

2.2 STORAGE

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

3. LOADING PROCEDURE

3.1 METHOD

PROCEDURE FOR NORMAL BINARY TAPES SHOULD BE FOLLOWED.

4. DR11B STARTING PROCEDURE

4.1 CONTROL SWITCH SETTING

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

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 BUFLN.

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)

FUNCT BITS INC WITH MAINT MODE XFERS

10 MAINT MODE XFERS

200 MAINT MODE XFERS

10. LISTING

11. FLOW CHART(S)

12. DA11B STARTING PROCEDURE

- 12.1 THERE ARE TWO STARTING LOCATIONS; ONE FOR THE COMPUTER THAT WILL BE THE SLAVE AND ANOTHER FOR THE COMPUTER THAT WILL BE THE MASTER.
- 12.2 SLAVE COMPUTER, LOAD ADDRESS 1006 AND PRESS START. PROCESSOR WILL HALT.
- 12.3 MASTER COMPUTER, LOAD ADDRESS 1000 AND PRESS START. PROCESSOR WILL HALT.
- 12.4 SLAVE COMPUTER, PRESS CONTINUE (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 SINC 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		NOP=000240
470		SCOPE=TRAP
471		HLT=EMT
472		BIT0=000001
473		BIT1=000002
474		BIT2=000004
475		BIT3=000010
476		BIT4=000020
477		BIT5=000040
478		BIT6=000100
479		BIT7=000200
480		BIT8=000400
481		BIT9=001000
482		BIT10=002000
483		BIT11=004000
484		BIT12=010000
485		BIT13=020000
486		BIT14=040000
487		BIT15=100000
488		
489		BUSERR=000004

```

490
491
492
493
494
495 000030 000000
496 000032 000030
497 000034 011506
498 000034 000340
499 000036 000034
500 000036 012274
501 000046 000340
502 000046 000046
503 000052 007452
504 000052 000052
505 000176 000000
506 000176 000176
507 000200 000000
508 000204 012706 014210
509 000210 005077 000606
510 000216 023737 000042 000046
511 000220 001404
512 000220 012702 013035
513 000224 004767 012654
514 000230 000167 000646
515 001000 001000
516
517
518
519
520
521
522
523 001000 000000
524 001002 000167 006622
525 001006 000000
526 001010 000167 006674
527 001014 177570
528 001016 177776
529 001020 172410
530 001022 172412
531 001024 172414
532 001026 172416
533 001030 000126
534 001032 000240
535 001034 000124
536 001036 052525
537 001040 173000
538 001042 014212
539 001044 015214
540 001046 000000
541 001050 000000
542 001052 000000
543 001054 000000
544 001056 000000
545 001060 177560

;LOAD TRAP CATCHER INTO 0 THRU 777.

.=0
.=30
PRINT
340
.=34
SCOPEC
340
.=46
$ENDAD
.=52
0
.=176
SWREG: 0
.=200
MOV #BUFF,%6 ;SET UP STACK LIMIT
CLR @PSW
CMP @#42,@#46 ;ARE WE IN ACT11 AUTO MODE?
BEQ 1$ ;SKIP TITLE IF YES
MOV #TITLE,%2 ;PRINT THE TITLE

1$: JSR %7,TTOUT
JMP SUSWR
.=1000

*****
; START OF BACK-TO-BACK DR11-B
*****

MSTART: HALT ;MASTER START
JMP MS1
SSTART: HALT ;SLAVE START
JMP SS1

SR: 177570
PSW: 177776
DRWC: 172410
DRBA: 172412
DRST: 172414
DRDB: 172416
DRVS: 126
DRINL: 240
DRINV: 124
NPR1: 52525
DIOMEM: 173000
INBUF: XINBUF
CHKBUF: XCHKBUFF
BUFLN: HALT
LENCHK: HALT
BRWAIT: HALT
WCLEN: HALT
RDYCHK: HALT
TKS: 177560

```



```

546 001062 177562      TKB: 177562
547 001064 177564      TPS: 177564
548 001066 177566      TPB: 177566
549 001070 000000      FNCCNT: 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 #64$,@#4 ;SET UP FOR TIMEOUT
558 001120 022777 177777 177666      CMP #-1,@SR ;REFERENCE HARDWARE SWITCH REGISTER
559 001126 001402      BEQ 65$
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 011262      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 011156      MOV #BEGIN,RETURN
572
573
574 ;*****
575 ; TEST 0 CAN ALL DR11-B REG BE ADDRESSED WITHOUT ERROR?
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 ;BUS ERROR, R0 HAS PC OF ERROR
591 001254 012767 000006 176522      MOV #6,BUSERR ;RESTORE #6 TO BUS ERROR VECTOR
592 001262 104400      SCOPE
593
594
595 ;*****
596 ; TEST 1 DOES RESET CLEAR DRWC?
597 ;*****
598
599
600 001264 012767 000010 011070      MOV #10,ICOUNT
601 001272 012777 177777 177520      MOV #-1,@DRWC ;ALL ONES TO DRWC

```

602 001300 004767 011100
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 011050
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 011004
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
 659 001440 104000
 660 001442 042777 000002 177354
 661 001450 032777 000002 177346
 662 001456 001401
 663 001460 104000

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

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

672 001462 104400
 673 001464 052777 000004 177332
 674 001472 032777 000004 177324
 675 001500 001001
 676 001502 104000
 677 001504 042777 000004 177312
 678 001512 032777 000004 177304
 679 001520 001401
 680 001522 104000

SCOPE
 BIS #BIT2,ADRST ;SET FNCT2
 BIT #BIT2,ADRST ;TEST FNCT2
 BNE .+4 ;IS IT SET?
 HLT ;FNCT2 IS CLEAR
 BIC #BIT2,ADRST ;CLEAR FNCT2
 BIT #BIT2,ADRST ;TEST FNCT2
 BEQ .+4 ;WAS IT CLEAR?
 HLT ;FNCT2 WAS SET

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

689 001524 104400
 690 001526 052777 000010 177270
 691 001534 032777 000010 177262
 692 001542 001001
 693 001544 104000
 694 001546 042777 000010 177250
 695 001554 032777 000010 177242
 696 001562 001401
 697 001564 104000

SCOPE
 BIS #BIT3,ADRST ;SET FNCT3
 BIT #BIT3,ADRST ;TEST FNCT3
 BNE .+4 ;IS IT SET?
 HLT ;FNCT3 IS CLEAR
 BIC #BIT3,ADRST ;CLEAR FNCT3
 BIT #BIT3,ADRST ;TEST FNCT3
 BEQ .+4 ;WAS IT CLEAR?
 HLT ;FNCT3 WAS SET

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

702 001566 104400
 703 001570 052777 000020 177226
 704 001576 032777 000020 177220
 705 001604 001001
 706 001606 104000
 707 001610 042777 000020 177206
 708 001616 032777 000020 177200
 709 001624 001401
 710 001626 104000

SCOPE
 BIS #BIT4,ADRST ;SET XBA16
 BIT #BIT4,ADRST ;TEST XBA16
 BNE .+4 ;IS IT SET?
 HLT ;XBA16 IS CLEAR
 BIC #BIT4,ADRST ;CLEAR XBA16
 BIT #BIT4,ADRST ;TEST XBA16
 BEQ .+4 ;IS IT CLEAR
 HLT ;XBA16 WAS SET

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

711
 712
 713

```
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
:BIT5,ADRST ;SET XBA17
: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
:BIT6,ADRST ;SET IE
: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
:BIT8,ADRST ;SET CYCLE
: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
:BIT12,ADRST ;SET MAINT
: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), FUNCT2(02), FNCT1(01)
```


770	002050	032777	000002	176746	BIT	#BIT1,ADRST	:TEST FNCT1
771	002056	001001			BNE	+.4	:IS IT SET?
772	002060	104000			HLT		:FNCT1 IS CLEAR
773	002062	032777	000004	176734	BIT	#BIT2,ADRST	:TEST FNCT2
774	002070	001001			BNE	+.4	:IS IT SET?
775	002072	104000			HLT		:FNCT2 IS CLEAR
776	002074	032777	000010	176722	BIT	#BIT3,ADRST	:TEST FNCT3
777	002102	001001			BNE	+.4	:IS IT SET?
778	002104	104000			HLT		:FNCT3 IS CLEAR
779	002106	032777	000020	176710	BIT	#BIT4,ADRST	:TEST XBA16
780	002114	001001			BNE	+.4	:IS IT SET?
781	002116	104000			HLT		:XBA16 IS CLEAR
782	002120	032777	000040	176676	BIT	#BIT5,ADRST	:TEST XBA17
783	002126	001001			BNE	+.4	:IS IT SET?
784	002130	104000			HLT		:XBA17 IS CLEAR
785	002132	032777	000100	176664	BIT	#BIT6,ADRST	:TEST IE
786	002140	001001			BNE	+.4	:IS IT SET?
787	002142	104000			HLT		:IE IS CLEAR
788	002144	032777	000400	176652	BIT	#BIT8,ADRST	:TEST CYCLE
789	002152	001001			BNE	+.4	:IS CYCLE SET?
790	002154	104000			HLT		:CYCLE IS CLEAR
791	002156	032777	010000	176640	BIT	#BIT12,ADRST	:TEST MAINT
792	002164	001001			BNE	+.4	:IS MAINT SET?
793	002166	104000			HLT		:MAINT IS CLEAR
794	002170	042777	010576	176626	BIC	#10576,ADRST	:CLEAR ALL R/W BITS IN DRST
795	002176	032777	000002	176620	BIT	#BIT1,ADRST	:TEST FNCT1
796	002204	001401			BEQ	+.4	:IS FNCT1 CLEAR?
797	002206	104000			HLT		:FNCT1 IS SET
798	002210	032777	000004	176606	BIT	#BIT2,ADRST	:TEST FNCT2
799	002216	001401			BEQ	+.4	:IS FNCT2 CLEAR?
800	002220	104000			HLT		:FNCT2 IS SET
801	002222	032777	000010	176574	BIT	#BIT3,ADRST	:TEST FNCT3
802	002230	001401			BEQ	+.4	:IS FNCT3 CLEAR?
803	002232	104000			HLT		:FNCT3 IS SET
804	002234	032777	000020	176562	BIT	#BIT4,ADRST	:TEST XBA16
805	002242	001401			BEQ	+.4	:IS XBA16 CLEAR
806	002244	104000			HLT		:XBA16 IS SET
807	002246	032777	000040	176550	BIT	#BIT5,ADRST	: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,ADRST	:TEST IE
812	002266	001401			BEQ	+.4	:IS IE CLEAR?
813	002270	104000			HLT		:IE IS SET
814	002272	032777	000400	176524	BIT	#BIT8,ADRST	:TEST CYCLE
815	002300	001401			BEQ	+.4	:IS CYCLE CLEAR?
816	002302	104000			HLT		:CYCLE IS SET
817	002304	032777	010000	176512	BIT	#BIT12,ADRST	: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 010034      MOV    #10,ICOUNT
827 002326 052777 010576 176470      BIS    #10576,ADRST      ;SET FOLLOWING: MAINT(12),CYCLE(08),IE(06),XBA17(05),
828                                     ;XBA16(04),FNCT3(03),FNCT2(02),FNCT1(01)
829 002334 017701 176464      MOV    ADRST,%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 010026      JSR    %7,CKSWR
835 002356 000005              RESET                ;CLEAR ALL DRST R/W BITS
836 002360 017701 176440      MOV    ADRST,%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 007756      MOV    #4000,ICOUNT
846 002404 012777 052525 176406      MOV    #052525,ADRWC    ;ALT 0'S AND 1'S TO DRWC
847 002412 022777 052525 176400      CMP    #052525,ADRWC    ;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,ADRWC    ;ALT 1'S AND 0'S TO DRWC
851 002432 022777 125252 176360      CMP    #125252,ADRWC    ;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,ADRBA    ;ALT 0'S AND 1'S TO DRBA
860 002454 022777 052524 176340      CMP    #052524,ADRBA    ;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,ADRBA    ;ALT 1'S AND 0'S TO DRBA
864 002474 022777 125252 176320      CMP    #125252,ADRBA    ;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 007646      CLR    ICOUNT
873 002514 005001              CLR    %1          ;SET-UP
874 002516 005077 176276      CLR    ADRWC        ;SET-UP
875 002522 020177 176272      INCWC: CMP    %1,ADRWC    ;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    ADRWC        ;GET NEXT NUMBER
879 002536 005201              INC    %1          ;GET NEXT NUMBER
880 002540 001370              BNE    INCWC        ;DONE WITH TEST? IF NOT CONTINUE
881
```


Line	Address	Instruction	Comments
882		*****	
883		TEST23 INCREMENTING PATTERN TO WRAP-AROUND IN DRBA	
884		*****	
885	002542	104400	SCOPE
886	002544	005001	CLR X1 ;SET-UP
887	002546	005077	176250 CLR @DRBA ;SET-UP
888	002552	020177	176244 INCBA: CMP X1,@DRBA ;SEE IF THEY ARE EQUAL
889	002556	001401	BEQ .+4 ;ARE THEY EQUAL?
890	002560	104000	HLT ;THEY'RE NOT EQUAL
891	002562	062777	000002 176232 ADD #2,@DRBA ;GET NEXT NUMBER
892	002570	062701	000002 ADD #2,X1 ;GET NEXT NUMBER
893	002574	001366	BNE INCBA ;DONE WITH TEST? IF NOT CONTINUE
894			
895			
896			*****
897			TEST25 TEST THAT DR11-B DOES NOT INTERRUPT WITH PROC AT LEVEL 7
898			*****
899	002576	104400	SCOPE
900	002600	012767	004000 007554 MOV #4000,ICOUNT
901	002606	012777	000340 176202 MOV #340,@PSW ;STATUS AT LEVEL 7
902	002614	032777	000200 176202 BIT #BIT7,@DRST ;CHECK READY BIT
903	002622	001010	BNE P7INV ;IS IT SET
904	002624	004767	007554 JSR X7,CKSWR
905	002630	000005	RESET ;INIT TO SET READY
906	002632	032777	000200 176164 BIT #BIT7,@DRST ;SEE IF READY IS SET NOW
907	002640	001001	BNE .+4 ;IS READY SET?
908	002642	104000	HLT ;READY CAN'T BE SET BY INIT
909	002644	012777	002720 176162 P7INV: MOV #P7ERR,@DRINV ;SET UP INT VECTOR
910	002652	012737	000340 000126 MOV #340,@#126
911			*****
912	002660	012777	177777 176132 CHGG1: MOV #-1,@DRWC ;SET WORD COUNT TO -1
913	002666	016777	176150 176126 MOV INBUF,@DRBA ;SET BUSS ADDRESS LOC
914	002674	052777	000101 176122 BIS #101,@DRST ;SET I.E. AND GO BITS
915	002702	012704	000012 MOV #10,X4 ;ALLOW 11/60 AND 11/70 TIME TO INTERRUPT
916	002706	077401	ITIME1: SOB X4,ITIME1
917			*****
918			
919	002710	042777	000100 176106 BIC #BIT6,@DRST ;DR11-B DIDN'T INTERRUPT - CLEAR IE
920	002716	000405	BR X1
921	002720	022626	P7ERR: CMP (%6)+,(%6)+ ;RESTORE STACK
922	002722	042777	000100 176074 BIC #BIT6,@DRST ;CLEAR IE
923	002730	104000	HLT ;DR11-B INTERRUPTED, BUT IT SHOULDN'T HAVE
924			
925			*****
926			TEST26 TEST THAT DR11-B DOES NOT INTERRUPT WITH PROC AT LEVEL 6
927			*****
928	002732	104400	X1: SCOPE
929	002734	012777	000300 176054 MOV #300,@PSW ;STATUS AT LEVEL 6
930	002742	032777	000200 176054 BIT #BIT7,@DRST ;CHECK READY BIT
931	002750	001010	BNE P6INV ;IS IT SET?
932	002752	004767	007426 JSR X7,CKSWR
933	002756	000005	RESET ;INIT TO SET READY
934	002760	032777	000200 176036 BIT #BIT7,@DRST ;SEE IF READY IS SET NOW
935	002766	001001	BNE .+4 ;IS READY SET?
936	002770	104000	HLT ;READY CAN'T BE SET BY INIT
937	002772	012777	003046 176034 P6INV: MOV #P6ERR,@DRINV ;SET UP INT VECTOR

```
938 003000 012737 000340 000126      MOV      #340,a#126
939                                     :*****
940 003006 012777 177777 176004 CHGG2: MOV      #-1,aDRWC      ;SET WORD COUNT TO -1
941 003014 016777 176022 176000      MOV      INBUF,aDRBA      ;SET BUSS ADDRESS LOC
942 003022 052777 000101 175774      BIS      #101,aDRST      ;SET I.E. AND GO BITS
943 003030 012704 000012      MOV      #10, %4      ;ALLOW 11/60 AND 11/70 TIME TO INTERRUPT
944 003034 077401      ITIME2: SOB      %4,ITIME2
945                                     :*****
946 003036 042777 000100 175760      BIC      #BIT6,aDRST      ;DR11-B DIDN'T INTERRUPT - CLEAR IE
947 003044 000405      BR      X2
948 003046 022626 P6ERR: CMP      (%6)+,(%6)+      ;RESTORE STACK
949 003050 042777 000100 175746      BIC      #BIT6,aDRST      ;CLEAR IE
950 003056 104000      HLT
951                                     ;DR11-B INTERRUPTED, BUT IT SHOULDN'T HAVE
952
953                                     ::*****
954                                     :TEST27 TEST THE DR11-B INTERRUPT WITH PROC AT LEVEL 5
955                                     :*****
956 003060 104400 X2:      SCOPE
957 003062 012777      MOV      #240,aPSW      ;STATUS AT LEVEL 5
958 003070 032777 000240 175726      BIT      #BIT7,aDRST      ;CHECK READY BIT
959 003076 001010      BNE      P5INV      ;IS IT SET?
960 003100 004767 007300      JSR      %7,CKSWR
961 003104 000005      RESET
962 003106 032777 000200 175710      BIT      #BIT7,aDRST      ;INIT TO SET READY
963 003114 001001      BNE      .+4      ;SEE IF READY IS SET NOW
964 003116 104000      HLT      ;IS IT SET?
965 003120 017767 175670 175750 P5INV: MOV      aSR,TEMP      ;RDY CAN'T BE SET BY INIT
966 003126 042767 177770 175742      BIC      #177770,TEMP      ;GET SWITCH SETTINGS
967 003134 012777 003222 175672      MOV      #P5ERR,aDRINV      ;ISOLATE BR LEVEL
968 003142 012737 000340 000126      MOV      #340,a#126      ;SET UP INT VECTOR
969                                     :*****
970 003150 012777 177777 175642 CHGG3: MOV      #-1,aDRWC      ;SET WORD COUNT TO -1
971 003156 016777 175660 175636      MOV      INBUF,aDRBA      ;SET BUSS ADDRESS LOC
972 003164 052777 000101 175632      BIS      #101,aDRST      ;SET I.E. AND GO BITS
973 003172 012704 000012      MOV      #10, %4      ;ALLOW 11/60 AND 11/70 TIME TO INTERRUPT
974 003176 077401      ITIME3: SOB      %4,ITIME3
975                                     :*****
976 003200 042777 000100 175616      BIC      #BIT6,aDRST      ;DR11-B DIDN'T INTERRUPT - CLEAR IE
977 003206 022767 000006 175662      CMP      #6,TEMP      ;BR LEVEL = 6 ?
978 003214 001001      BNE      1$      ;NO, EVERYTHING IS OK
979 003216 104000      HLT      ;DR11-B WITH BR = 6 SHOULD HAVE INTERRUPTED
980 003220 000411      BR      X3
981 003222 022626 P5ERR: CMP      (%6)+,(%6)+      ;RESTORE STACK
982 003224 042777 000100 175572      BIC      #BIT6,aDRST      ;CLEAR IE
983 003232 022767 000006 175636      CMP      #6,TEMP      ;BR LEVEL = 6 ?
984 003240 001401      BEQ      X3      ;YES, EVERYTHING IS OK
985 003242 104000      HLT      ;DR11-B WITH BR NOT = 6 SHOULD NOT HAVE INTERRUPTED
986
987                                     ::*****
988                                     :TEST30 TEST THE DR11-B INTERRUPT WITH PROC AT LEVEL 4
989                                     :*****
990 003244 104400 X3:      SCOPE
991 003246 012777 000200 175542      MOV      #200,aPSW      ;STATUS AT LEVEL 4
992 003254 032777 000200 175542      BIT      #BIT7,aDRST      ;CHECK READY BIT
993 003262 001010      BNE      P4INV      ;IS IT SET?
```



```

994 003264 004767 007114 JSR %7,CKSWR
995 003270 000005 RESET
996 003272 032777 000200 175524 BIT #BIT7,ADRST
997 003300 001001 BNE .+4
998 003302 104000 HLT
999 003304 017767 175504 175564 P4INV: MOV @SR,TEMP
1000 003312 042767 177770 175556 BIC #177770,TEMP
1001 003320 012777 003414 175506 MOV #P4INT,ADRINV
1002 003326 012737 000340 000126 MOV #340,@#126
1003 *****
1004 003334 012777 177777 175456 CHGG4: MOV #-1,ADRWC
1005 003342 016777 175474 175452 MOV INBUF,ADRBA
1006 003350 052777 000101 175446 BIS #101,ADRST
1007 003356 012704 000012 MOV #10,%4
1008 003362 077401 ITIME4: SOB %4,ITIME4
1009 *****
1010 003364 005077 175434 CLR ADRST
1011 003370 012777 000126 175436 MOV #126,ADRINV
1012 003376 022767 000004 175472 CMP #4,TEMP
1013 003404 001401 BEQ 1$
1014 003406 104000 HLT
1015 003410 000167 000026 1$: JMP X4
1016 003414 005077 175404 P4INT: CLR ADRST
1017 003420 022626 CMP (%6)+,(%6)+
1018 003422 012777 000126 175404 MOV #126,ADRINV
1019 003430 022767 000004 175440 CMP #4,TEMP
1020 003436 001001 BNE X4
1021 003440 104000 HLT
1022
1023
1024
1025 *****
1026 ***** TEST30A TEST THAT DR11-B DOES INTERRUPT WITH PROC AT LEVEL 3 *****
1027 003442 104400 X4: SCOPE
1028 003444 012777 000140 175344 MOV #140,APSW
1029 003452 032777 000200 175344 BIT #BIT7,ADRST
1030 003460 001010 BNE P3INV
1031 003462 004767 006716 JSR %7,CKSWR
1032 003466 000005 RESET
1033 003470 032777 000200 175326 BIT #BIT7,ADRST
1034 003476 001001 BNE .+4
1035 003500 104000 HLT
1036 003502 012777 003572 175324 P3INV: MOV #P3INT,ADRINV
1037 003510 012737 000340 000126 MOV #340,@#126
1038 *****
1039 003516 012777 177777 175274 CHGG5: MOV #-1,ADRWC
1040 003524 016777 175312 175270 MOV INBUF,ADRBA
1041 003532 052777 000101 175264 BIS #101,ADRST
1042 003540 012704 000012 MOV #10,%4
1043 003544 077401 ITIME5: SOB %4,ITIME5
1044 *****
1045 003546 005077 175252 CLR ADRST
1046 003552 012777 000126 175254 MOV #126,ADRINV
1047 003560 005037 000126 CLR @#126
1048 003564 104000 HLT
1049 003566 000167 000020 JMP X5
  
```

:INIT TO SET READY
 :SEE IF READY IS SET NOW
 :IS IT SET
 :READY CAN'T BE SET BY INIT
 :GET SWITCH SETTINGS
 :ISOLATE BR LEVEL
 :SET UP INT VECTOR
 :SET WORD COUNT TO -1
 :SET BUSS ADDRESS LOC
 :SET I.E. AND GO BITS
 :ALLOW 11/60 AND 11/70 TIME TO INTERRUPT
 :CLEAR IE
 :RESTORE INTERRUPT VECTOR
 :BR LEVEL = 4 ?
 :YES, EVERYTHING IS OK
 :DR11-B DIDN'T INTERRUPT
 :CLEAR IE
 :REPOSITION THE STACK AFTER AN INTERRUPT
 :RESTORE INTERRUPT VECTOR
 :BR LEVEL = 4 ?
 :NO, EVERYTHING IS OK
 :DR11-B WITH BR = 4 SHOULD NOT HAVE INTERRUPTED

:STATUS AT LEVEL 3
 :CHECK READY BIT
 :IS IT SET?
 :INIT TO SET READY
 :SEE IF READY IS SET NOW
 :IS IT SET
 :READY CAN'T BE SET BY INIT
 :SET UP INT VECTOR
 :SET WORD COUNT TO -1
 :SET BUSS ADDRESS LOC
 :SET I.E. AND GO BITS
 :ALLOW 11/60 AND 11/70 TIME TO INTERRUPT
 :CLEAR IE
 :RESTORE INTERRUPT VECTOR
 :RESTORE TRAP CATCHER
 :DR11-B DIDN'T INTERRUPT

```
1050 003572 005077 175226 P3INT: CLR @DRST ;CLEAR IE
1051 003576 022626 CMP (%)+,(%)+ ;REPOSITION THE STACK AFTER AN INTERRUPT
1052 003600 012777 000126 175226 MOV #126,@DRINV ;RESTORE INTERRUPT VECTOR
1053 003606 005037 000126 CLR @#126 ;RESTORE TRAP CATCHER
```

```
::*****
::TEST31 TEST THAT FNCT BITS CONTROL DSTAT BITS (M968 MUST BE USED IN USER SLOTS)
::*****
```

```
1058 X5: SCOPE
1059 003612 104400 CLR @DRST ;CLEAR FUNCTION BITS
1060 003614 005077 175204 BIT #16,@DRST ;CHECK FUNCTION BITS
1061 003620 032777 000016 175176 BEQ .+4 ;FUNCTION BITS CLEAR?
1062 003626 001401 HLT ;FUNCTION BITS NOT CLEAR
1063 003630 104000 BIS #BIT1,@DRST ;SET FNCT1
1064 003632 052777 000002 175164 BIT #BIT9,@DRST ;CHECK DSTAT C
1065 003640 032777 001000 175156 BNE .+4 ;IS IT SET?
1066 003646 001001 HLT ;DSTAT C IS CLEAR
1067 003650 104000 BIT #6000,@DRST ;CHECK THAT DSTAT A AND DSTAT B ARE CLEAR
1068 003652 032777 006000 175144 BEQ .+4 ;ARE THEY CLEAR?
1069 003660 001401 HLT ;DSTAT A AND/OR DSTAT B IS SET
1070 003662 104000 CLR @DRST ;CLEAR DRST
1071 003664 005077 175134 BIS #BIT2,@DRST ;SET FNCT2
1072 003670 052777 000004 175126 BIT #BIT10,@DRST ;CHECK DSTAT B
1073 003676 032777 002000 175120 BNE .+4 ;IS IT SET?
1074 003704 001001 HLT ;DSTAT B IS CLEAR
1075 003706 104000 BIT #5000,@DRST ;CHECK THAT DSTAT A AND DSTAT C ARE CLEAR
1076 003710 032777 005000 175106 BEQ .+4 ;ARE THEY CLEAR?
1077 003716 001401 HLT ;DSTAT A AND/OR DSTAT B IS SET
1078 003720 104000 CLR @DRST ;CLEAR DRST
1079 003722 005077 175076 BIS #BIT3,@DRST ;SET FNCT3
1080 003726 052777 000010 175070 BIT #BIT11,@DRST ;CHECK DSTAT A
1081 003734 032777 004000 175062 BNE .+4 ;IS IT SET?
1082 003742 001001 HLT ;DSTAT A IS CLEAR
1083 003744 104000 BIT #3000,@DRST ;CHECK THAT DSTAT B AND DSTAT C ARE CLEAR
1084 003746 032777 003000 175050 BEQ .+4 ;ARE THEY CLEAR?
1085 003754 001401 HLT ;DSTAT B AND/OR DSTAT C IS SET
1086 003756 104000 CLR @DRST ;CLR DRST
1087 003760 005077 175040
```

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

```
1088 TNPR1: SCOPE
1089 003764 104400 TST @DRST ;CHECK ERROR BIT
1090 003766 005777 175032 BPL NPRRDY ;IS IT CLEAR?
1091 003772 100027 BIT #BIT13,@DRST ;CHECK ATTN
1092 003774 032777 020000 175022 BEQ .+4 ;IS ATTN CLEAR
1093 004002 001401 HLT ;ATTN IS SET
1094 004004 104000 BIT #BIT14,@DRST ;CHECK NEX
1095 004006 032777 040000 175010 BEQ N1413 ;IS NEX CLEAR?
1096 004014 001410 BIC #BIT14,@DRST ;TRY TO CLEAR NEX
1097 004016 042777 040000 175000 BIT #BIT14,@DRST ;CHECK AGAIN
1098 004024 032777 040000 174772 BEQ .+4 ;NEX STILL SET
1099 004032 001401 HLT ;NEX CAN'T BE CLEARED BY MOVING A 0 TO IT
1100 004034 104000 N1413: CLR @DRBA ;TRY TO CLEAR BAOF
1101 004036 005077 174760 TST @DRST ;CHECK ERROR BIT AGAIN
1102 004042 005777 174756 BEQ .+4 ;IS IT CLEAR
1103 004046 001401
```



```
1106 004050 104000 HLT ;ERROR CAUSED BY SOMETHING OTHER THAN NEX,ATTN, OR BAOF
1107 004052 012777 177777 174740 NPPRDY: MOV #-1,ADRWC ;SET UP FOR 1 TRANSFER
1108 004060 012777 001036 174734 MOV #NPR1,ADRBA ;TRANSFER FROM BUS ADDRESS IN NPR1
1109 004066 005077 174734 CLR ADRDB ;GET READY TO RECEIVE DATA
1110 004072 012767 052525 174736 MOV #52525,NPR1 ;SET UP TRANSFER DATA
1111 004100 012777 004144 174726 MOV #INTB,ADRINV ;INTERRUPT VECTOR TO INTB
1112 004106 012777 000005 174714 MOV #5,ADRVS ;INTERRUPT PRIORITY TO LEVEL 5
1113 004114 005077 174676 CLR APSW ;LET THE DR11-B INTERRUPT
1114 004120 012777 000101 174676 MOV #101,ADRST ;IE AND DO TO DRST
1115 004126 005067 000002 CLR I$+2 ;WAIT FOR NPR AND INTERRUPT
1116 004132 005227 000001 1$: INC #1
1117 004136 001375 BNE 1$
1118 004140 104000 HLT ;NO DR11-B INTERRUPT
1119 004142 000424 BR T33CLR ;CLEAR IE
1120 004144 004767 003204 INTB: JSR %7,ERRCHK
1121 004150 005777 174644 TST ADRWC ;TEST DRWC
1122 004154 001401 BEQ .+4 ;IS DRWC EQUAL TO ZERO?
1123 004156 104000 HLT ;DRWC NOT EQUAL TO ZERO
1124 004160 022777 001040 174634 CMP #NPR1+2,ADRBA ;COMPARE CORRECT DRBA WITH DRBA
1125 004166 001401 BEQ .+4 ;IS THE DRBA CORRECT?
1126 004170 104000 HLT ;DRBA IS WRONG
1127 004172 022777 052525 174626 CMP #52525,ADRDB ;CHECK FOR CORRECT DATA
1128 004200 001401 BEQ .+4 ;DATA GET TRANSFERRED?
1129 004202 104000 HLT ;BAD DATA IN DRDB
1130 004204 004767 003054 JSR %7,NORMAL
1131 004210 022626 CMP (%6)+,(%6)+ ;RESTORE STACK
1132 004212 000403 BR TNPRO ;GO TO NEXT TEST (NPR OUT)
1133 004214 005077 174604 T33CLR: CLR ADRST ;CLEAR IE
1134 004220 000662 BR TNPR1 ;TRY TEST AGAIN
1135 ;*****
1136 ; TEST 34 TEST FOR 1 DATO NPR TRANSFER (WITH M968 IN USER SLOTS)
1137 ;*****
1138 004222 104400 TNPRO: SCOPE
1139 004224 012777 177777 174566 MOV #-1,ADRWC ;SET UP FOR 1 TRANSFER
1140 004232 012777 001036 174562 MOV #NPR1,ADRBA ;TRANSFER TO BUS ADDRESS IN NPR1
1141 004240 005067 174572 CLR NPR1 ;GET READY TO RECEIVE DATA
1142 004244 012777 052525 174554 MOV #52525,ADRDB ;SET UP TO TRANSFER DATA
1143 004252 012777 004316 174554 MOV #INTC,ADRINV ;INTERRUPT VECTOR TO INTC
1144 004260 016777 174546 174542 MOV DRINL,ADRVS ;INTERRUPT STATUS TO LEVEL DRINL
1145 004266 005077 174524 CLR APSW ;PROC STATUS TO ZERO
1146 004272 012777 000103 174524 MOV #103,ADRST ;IE, FNCT1(C1 CONTROL), AND DO TO DRST
1147 004300 005067 000002 CLR I$+2 ;WAIT FOR NPR AND INTER
1148 004304 005227 000001 1$: INC #1
1149 004310 001375 BNE 1$
1150 004312 104000 HLT ;NO DR11-B INTERRUPT
1151 004314 000424 BR T34CLR ;CLEAR IE
1152 004316 004767 003032 INTC: JSR %7,ERRCHK
1153 004322 005777 174472 TST ADRWC ;TEST DRWC
1154 004326 001401 BEQ .+4 ;IS DRWC EQUAL TO ZERO?
1155 004330 104000 HLT ;DRWC EQUAL TO ZERO
1156 004332 022777 001040 174462 CMP #NPR1+2,ADRBA ;COMPARE CORRECT DRBA WITH DRBA
1157 004340 001401 BEQ .+4 ;IS THE DRBA CORRECT?
1158 004342 104000 HLT ;DRBA IS WRONG
1159 004344 026727 174466 052525 CMP NPR1,#52525 ;CHECK FOR CORRECT DATA
1161 004352 001401 BEQ .+4 ;CORRECT DATA TRANSFERRED?
```

```
1162 004354 104000 HLT ;BAD DATA
1163 004356 004767 002702 JSR %7,NORMAL
1164 004362 022626 CMP (%6)+,(%6)+ ;RESTORE STACK
1165 004364 000403 BR T35 ;GO TO NEXT TEST
1166 004366 005077 174432 T34CLR: CLR @DRST ;CLEAR IE
1167 004372 000713 BR TNPRO ;TRY TEST AGAIN
1168
1169 ::*****
1170 ::TEST 35 STRING OF 10 DATI'S (WITH M968 IN USER SLOTS)
1171 ::*****
1172 T35: SCOPE
1173 MOV #20,BUFLEN ;LENGTH OF BUFFER=20
1174 JSR %7,LODBUF ;LOAD THE BUFFER WITH INCREMENTING PATTERN
1175 ASR BUFLEN ;BUFLEN=10
1176 MOV BUFLEN,WLEN ;PREPARE NUMBER FOR DRWC
1177 NEG WLEN ;2'S COMPLEMENT OF BUFLEN
1178 MOV WLEN,@DRWC ;SET UP DRWC
1179 MOV INBUF,@DRBA ;SET UP DRBA
1180 MOV #-1,@DRDB ;MAINT AIDE
1181 MOV #INTA,@DRINV ;INT VECTOR TO INTA
1182 MOV DRINL,@DRVS ;INT VECTOR TO PRIORITY DRINL
1183 CLR @PSW ;LET THE DR11-B INTERRUPT
1184 MOV #101,@DRST ;IE AND DO TO DRST
1185 BR ;WAIT FOR INTERRUPT
1186 CMP #7,@DRDB ;CHECK THAT WORD #10 OF INBUF IS IN DRBA
1187 BEQ .+4 ;IS IT?
1188 HLT ;BAD DATA IN DRDB
1189
1190 ::*****
1191 ::TEST 36 STRING OF 10 DATO'S (WITH M968 IN USER SLOTS)
1192 ::*****
1193 SCOPE
1194 MOV #20,BUFLEN ;LENGTH OF BUFFER=20
1195 JSR %7,LODBUF ;LOAD THE BUFFER WITH INCREMENTING PATTERN
1196 ASR BUFLEN ;BUFLEN=10
1197 MOV BUFLEN,WLEN ;PREPARE NUMBER FOR DRWC
1198 NEG WLEN ;2'S COMPLEMENT OF BUFLEN
1199 MOV WLEN,@DRWC ;SET UP DRWC
1200 MOV INBUF,@DRBA ;SET UP DRBA
1201 MOV #52525,@DRDB ;SET UP DRDB
1202 MOV #INTA,@DRINV ;INTERRUPT VECTOR TO INTA
1203 MOV DRINL,@DRVS ;INTERRUPT VECTOR TO PRIORITY DRINL
1204 CLR @PSW ;LET THE DR11-B INTERRUPT
1205 MOV #103,@DRST ;IE, FNCT1(C1 CONTROL), AND DO TO DRST
1206 BR ;WAIT FOR INTERRUPT
1207 JSR %7,DATOCK ;CHECK INBUF
1208
1209 ::*****
1210 ::TEST 37 STRING OF 200 DATI'S
1211 ::*****
1212 SCOPE
1213 MOV #200,BUFLEN ;LENGTH OF BUFFER=200
1214 JSR %7,LODBUF ;LOAD THE BUFFER WITH INCREMENTING PATTERN
1215 MOV BUFLEN,WLEN ;PREPARE NUMBER FOR DRWC
1216 NEG WLEN ;2'S COMPLEMENT OF BUFLEN
1217 MOV WLEN,@DRWC ;SET UP DRWC
```



```
1218 004656 016777 174160 174136      MOV      INBUF,ADRBA      ;SET UP DRBA
1219 004664 012777 177777 174134      MOV      #-1,ADRDB      ;MAINT AIDE
1220 004672 012777 007130 174134      MOV      #INTA,ADRINV    ;INT VECTOR TO INTA
1221 004700 016777 174126 174122      MOV      DRINL,ADRVS     ;INT VECTOR TO PRIORITY DRINL
1222 004706 005077 174104      CLR      APSW           ;LET THE DR11-B INTERRUPT
1223 004712 012777 000101 174104      MOV      #101,ADRST      ;IE AND DO TO DRST
1224 004720 000777      BR      ;WAIT FOR INTERRUPT
1225 004722 022777 000177 174076      CMP      #177,ADRDB     ;CHECK THAT WORD #200 OF INBUF IS IN DRBA
1226 004730 001401      BEQ      .+4      ;IS IT?
1227 004732 104000      HLT      ;BAD DATA IN DRDB
1228                                     ;:*****
1229                                     ;:TEST 40 STRING OR 200 DATO'S
1230                                     ;:*****
1231 004734 104400      SCOPE
1232 004736 012767 000201 174102      MOV      #201,BUFLEN    ;LENGTH OF BUFFER=201
1233 004744 004767 002056      JSR      %7,LODBUF     ;LOAD THE BUFFER WITH INCREMENTING PATTERN
1234 004750 005367 174072      DEC      BUFLEN        ;BUFLEN=200
1235 004754 016767 174066 174072      MOV      BUFLEN,WCLN    ;PREPARE NUMBER FOR DRWC
1236 004762 005467 174066      NEG      WCLN          ;2'S COMPLEMENT OF BUFLEN
1237 004766 016777 174062 174024      MOV      WCLN,ADRWC     ;SET UP DRWC
1238 004774 016777 174042 174020      MOV      INBUF,ADRBA    ;SET UP DRBA
1239 005002 012777 052525 174016      MOV      #52525,ADRDB   ;SET UP DRDB
1240 005010 012777 007130 174016      MOV      #INTA,ADRINV    ;INTERRUPT VECTOR TO INTA
1241 005016 016777 174010 174004      MOV      DRINL,ADRVS     ;INTERRUPT VECTOR TO PRIORITY DRINL
1242 005024 005077 173766      CLR      APSW           ;LET THE DR11-B INTERRUPT
1243 005030 012777 000103 173766      MOV      #103,ADRST      ;IE, FNCT1, AND DO TO DRST
1244 005036 000777      BR      ;WAIT FOR INTERRUPT
1245 005040 004767 002242      JSR      %7,DATOCK      ;CHECK INBUF
1246
1247                                     ;:*****
1248                                     ;:TEST 42 TEST THAT DOING A DATO TO THE DIODE MEMORY CAUSES NEX
1249                                     ;:*****
1250 005044 104400      SCOPE
1251 005046 012777 177776 173744      MOV      #-2,ADRWC      ;SET UP DRWC
1252 005054 016777 173760 173740      MOV      DIOMEM,ADRBA    ;SET UP DRBA
1253 005062 012777 005116 173744      MOV      #NEXCHK,ADRINV  ;INTERRUPT VECTOR TO NEXCHK
1254 005070 016777 173736 173732      MOV      DRINL,ADRVS     ;INTERRUPT STATUS TO LEVEL DRINL
1255 005076 005077 173714      CLR      APSW           ;LET THE DR11-B INTERRUPT
1256 005102 012777 000163 173714      MOV      #163,ADRST      ;IE, FNCT1, XBA17, XBA16, AND GO TO DRST
1257 005110 005237 177560      INC      A#177560      ;WAIT FOR INTERRUPT
1258 005114 104000      HLT      ;NO DR11-B INTERRUPT
1259 005116 042777 000100 173700      NEXCHK: BIC      #BIT6,ADRST ;CLEAR INTERRUPT ENABLE
1260 005124 005777 173674      TST      ADRST         ;TEST DRST
1261 005130 001001      BNE      .+4      ;ERROR SET?
1262 005132 104000      HLT      ;ERROR NOT SET
1263 005134 105777 173664      TSTB     ADRST         ;TEST FOR READY
1264 005140 001001      BNE      .+4      ;READY SET?
1265 005142 104000      HLT      ;READY ISN'T SET
1266 005144 032777 040000 173652      BIT      #BIT14,ADRST   ;CHECK NEX
1267 005152 001001      BNE      .+4      ;NEX SET?
1268 005154 104000      HLT      ;NEX IS CLEAR
1269 005156 042777 040000 173640      BIC      #BIT14,ADRST   ;CLEAR NEX
1270 005164 022626      CMP      (%)+,(%)+ ;RESTORE THE STACK
1271 005166 004767 002072      JSR      %7,NORMAL
1272
1273                                     ;:*****
```

```

1274      :      TEST 43 TEST THAT CROSSING A 32K BOUNDRY DOES NOT CAUSE
1275      :      A BAOF AND DOES NOT FORCE AN ERROR
1276      :      :*****
1277      005172 104400      SCOPE
1278      005174 012767 000010 005160      MOV      #10,ICOUNT
1279      005202 012777 177760 173610      MOV      #-20,ADRWC      ;SET UP DRWC
1280      005210 012777 177776 173604      MOV      #-2,ADRBA      ;SET UP DRBA FOR PROC STATUS ADDRESS
1281      005216 012777 005252 173610      MOV      #BAOFCK,ADRINV      ;INTERRUPT VECTOR TO BAOFCK
1282      005224 016777 173602 173576      MOV      DRINL,ADRVS      ;INTERRUPT STATUS TO LEVEL DRINL
1283      005232 005077 173560      CLR      APSW      ;LET THE DR11-B INTERRUPT
1284      005236 012777 000163 173560      MOV      #163,ADRST      ;IE, FNCT1, XBA17, XBA16, AND GO TO DRST
1285      005244 005237 177560      INC      A#177560      ;WAIT FOR INTERRUPT
1286      005250 104000      HLT      ;NO DR11-B INTERRUPT
1287      005252 042777 000100 173544 BAOFCK: BIC      #BIT6,ADRST      ;CLEAR INTERRUPT ENABLE
1288      005260 022626      CMP      (%6)+, (%6)+      ;RESTORE THE STACK
1289      005262 005777 173536      TST      ADRST      ;TEST DRST
1290      005266 100401      BMI      .+4      ;ERROR SET?
1291      005270 104000      HLT      ;ERROR NOT SET
1292      005272 105777 173526      TSTB     ADRST      ;TEST FOR READY
1293      005276 100401      BMI      .+4      ;READY SET?
1294      005300 104000      HLT      ;READY ISN'T SET
1295      005302 042777 040000 173514      BIC      #BIT14,ADRST      ;CLEAR NEX
1296      005310 032777 060000 173506      BIT      #60000,ADRST      ;CHECK NEX AND ATTN
1297      005316 001401      BEQ      .+4      ;ARE THEY CLEAR?
1298      005320 104000      HLT      ;NEX AND/OR ATTN IS SET
1299      005322 005777 173476      TST      ADRST      ;TEST FOR ERROR
1300      005326 100001      BPL      .+4      ;IS ERROR CLEAR?
1301      005330 104000      HLT      ;ERROR IS SET
1302      005332 005077 173464      CLR      ADRBA      ;CLEAR BUS ADDRESS REGISTER
1303
1304      005336 004767 005042      JSR      %7,CKSWR
1305      005342 000005      RESET      ;INIT
1306      005344 004767 001714      JSR      %7,NORMAL
1307
1308      :      :*****
1309      :      TEST 44 TEST THAT RESET CLEARS DRDB
1310      :      :*****
1311      005350 104400      SCOPE
1312      005352 012767 000010 005002      MOV      #10,ICOUNT
1313      005360 012777 177777 173440      MOV      #-1,ADRDB      ;ALL ONES TO DRBA
1314      005366 004767 005012      JSR      %7,CKSWR
1315      005372 000005      RESET      ;INIT
1316      005374 005777 173426      TST      ADRDB      ;LOOKING FOR Z-BIT TO SET
1317      005400 001401      BEQ      .+4      ;DID DRDB GET CLEARED?
1318      005402 104000      HLT      ;DRDB NOT CLEAR
1319
1320      :      :*****
1321      :      TEST 45 TEST THAT ALL DRDB BITS CAN BE SET
1322      :      :*****
1323      005404 104400      SCOPE
1324      005406 012767 004000 004746      MOV      #4000,ICOUNT
1325      005414 012777 177777 173404      MOV      #-1,ADRDB      ;SET ALL BITS IN DRDB
1326      005422 022777 177777 173376      CMP      #-1,ADRDB      ;LOOKING FOR Z-BIT TO SET
1327      005430 001401      BEQ      .+4      ;SEE IF ALL BITS GOT SET
1328      005432 104000      HLT      ;ALL DRDB BITS AREN'T SET
1329

```



```
1330
1331
1332
1333 005434 104400
1334 005436 012777 052525 173362
1335 005444 022777 052525 173354
1336 005452 001401
1337 005454 104000
1338 005456 012777 125252 173342
1339 005464 022777 125252 173334
1340 005472 001401
1341 005474 104000
1342
1343
1344
1345
1346 005476 104400
1347 005500 005067 004656
1348 005504 005001
1349 005506 005077 173314
1350 005512 020177 173310
1351 005516 001401
1352 005520 104000
1353 005522 005277 173300
1354 005526 005201
1355 005530 001370
1356
1357
1358
1359
1360
1361 005532 104400
1362 005534 004767 004644
1363 005540 000005
1364 005542 032777 000200 173254
1365 005550 001001
1366 005552 104000
1367 005554 032777 177577 173242
1368 005562 001401
1369 005564 104000
1370
1371
1372
1373
1374 005566 104400
1375 005570 012767 004000 004564
1376 005576 032777 000001 173216
1377 005604 001401
1378 005606 104000
1379
1380
1381
1382
1383
1384
1385 005610 104400

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

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

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

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

*****
TEST 52 TEST THAT GO CLEARS READY
*****
SCOPE
```

```

1386 005612 012767 004000 004542      MOV      #4000,ICOUNT
1387 005620 012777 177600 173172      MOV      #-200,ADRWC      ;SET-UP DRWC
1388 005626 016777 173210 173166      MOV      INBUF,ADRBA      ;SET-UP DRBA
1389 005634 105777 173164              TSTB      ADRST      ;CHECK READY
1390 005640 100401              BMI      .+4      ;IS READY SET?
1391 005642 104000              HLT              ;READY IS CLEAR
1392 005644 012777 000011 173152      MOV      #11,ADRST      ;FNCT3 (NON-BURST) AND GO TO DRST
1393 005652 105777 173146              TSTB      ADRST      ;CHECK READY
1394 005656 100001              BPL      .+4      ;IS READY CLEAR?
1395 005660 104000              HLT              ;READY IS STILL SET
1396 005662 005067 173170              CLR      RDYCHK      ;CLEAR READY CHECK
1397 005666 105777 173132      TSTRDY: TSTB      ADRST      ;CHECK READY
1398 005672 100406              BMI      DONE      ;IF SET GO TO DONE
1399 005674 062767 000004 173154      ADD      #4,RDYCHK      ;CHECKING TIME FOR READY TO BE SET
1400 005702 100401              BMI      .+4      ;IF RDYCHK GETS NEGATIVE IT TOOK TOO LONG
1401 005704 000770              BR       TSTRDY      ;CHECK AGAIN
1402 005706 104000              HLT              ;READY GOT CLEARED BUT NEVER SET AGAIN
1403 005710 000240      DONE:  NOP      ;GO TO NEXT TEST
1404
1405      ;*****
1406      ;TEST 55 TEST THAT GIVING A DO WITHOUT CLEARING A PREVIOUS ERROR
1407      ;CAUSES ANOTHER INTERRUPT
1408      ;FOR THIS TEST, THE NORMAL INTERRUPT MECHANISM SHOULD BE WORKING
1409      ;*****
1409 005712 104400              SCOPE
1410 005714 012777 177760 173076      MOV      #-20,ADRWC      ;SET-UP DRWC
1411 005722 016777 173112 173072      MOV      DIOMEM,ADRBA      ;SET-UP DRBA
1412 005730 012777 005764 173076      MOV      #ERRDO,ADRINV      ;INTERRUPT VECTOR TO ERRDO
1413 005736 012777 000140 173064      MOV      #140,ADRVS      ;INTERRUPT STATUS TO LEVEL 4
1414 005744 005077 173046              CLR      APSW      ;LET THE DR11-B INTERRUPT
1415 005750 012777 000163 173046      MOV      #163,ADRST      ;IE, XBA17,XBA16,FNCT1 AND GO TO DRST
1416 005756 005277 173076              INC      ATKS      ;WAIT FOR INTERRUPT
1417 005762 104000              HLT              ;NO DR11-B INTERRUPT
1418 005764 005777 173034      ERRDO: TST      ADRST      ;TEST DRST
1419 005770 100401              BMI      .+4      ;ERROR SET?
1420 005772 104000              HLT              ;ERROR IS CLEAR - SHOULD HAVE NEX
1421 005774 012777 006046 173032      MOV      #ERRDO1,ADRINV      ;INTERRUPT VECTOR TO ERRDO1
1422 006002 005077 173014              CLR      ADRBA      ;PREVENT CAUSING ANOTHER ERROR
1423 006006 042777 000062 173010      BIC      #62,ADRST      ;CLEAR XBA17, XBA16, AND FNCT1
1424 006014 012777 177777 172776      MOV      #-1,ADRWC      ;SET-UP DRWC
1425 006022 005277 172776              INC      ADRST      ;DO TO DRST
1426 006026 005067 000002              CLR      1$+2
1427 006032 005227 000001      1$:      INC      #1
1428 006036 001375              BNE      1$
1429 006040 104000              HLT
1430 006042 162706 000004              SUB      #4,%6      ;NO DR11-B INTERRUPT
1431 006046 005777 172752      ERRDO1: TST      ADRST      ;FAKE AN INTERRUPT WITH STACK
1432 006052 100401              BMI      .+4      ;CHECK ERROR
1433 006054 104000              HLT              ;ERROR SET?
1434
1435              ADD      #10,%6      ;ERROR IS CLEAR - SHOULD BE SET BECAUSE
1436              JSR      %7,NORMAL      ;PREVIOUS ERROR WAS NOT CLEARED
1437              ;REPOSITION THE STACK
1438
1439      ;*****
1440      ;TEST 56 STRING OF 200 DATI'S NON-BURST MODE
1441      ;*****
1441 006066 104400              SCOPE

```


1442	006070	012767	000200	172750	MOV	#200,BUFLEN	:LENGTH OF BUFFER=200
1443	006076	004767	000724		JSR	%7,LODBUF	:LOAD THE BUFFER WITH INCREMENTING PATTERN
1444	006102	016767	172740	172744	MOV	BUFLEN,WCLLEN	:PREPARE NUMBER FOR DRWC
1445	006110	005467	172740		NEG	WCLLEN	:2'S COMPLEMENT OF BUFLEN
1446	006114	016777	172734	172676	MOV	WCLLEN,ADRWC	:SET-UP DRWC
1447	006122	016777	172714	172672	MOV	INBUF,ADRBA	:SET-UP DRBA
1448	006130	012777	177777	172670	MOV	#-1,ADRDB	:MAINT AIDE
1449	006136	012777	007130	172670	MOV	#INTA,ADRINV	:INT VECTOR TO INTA
1450	006144	016777	172662	172656	MOV	DRINL,ADRVS	:INT VECTOR TO PRIORITY DRINL
1451	006152	005077	172640		CLR	APSW	:LET THE DR11-B INTERRUPT
1452	006156	012777	000111	172640	MOV	#111,ADRST	:IE, FNCT3, AND DO TO DRST
1453	006164	005267	172662		INC	BRWAIT	:USE A WAIT OR BR. INSTRUCTION
1454	006170	032767	000001	172654	BIT	#BIT0,BRWAIT	:SEE WHICH ONE
1455	006176	001403			BEQ	DATINB	:BIT 0 CLEAR=BR.
1456	006200	000001			WAIT		:WAIT FOR INTERRUPT
1457	006202	000240			NOP		
1458	006204	000401			BR	+.4	
1459	006206	000777			DATINB: BR		
1460	006210	022777	000177	172610	CMP	#177,ADRDB	:CHECK THAT WORD #200 OF INBUF IS IN DRBA
1461	006216	001401			BEQ	+.4	:IS IT?
1462	006220	104000			HLT		:BAD DATA IN DRDB
1463							
1464					::*****		
1465					::	TEST 57 STRING OF 200 DATO'S NON-BURST MODE	
1466					::*****		
1467	006222	104400			SCOPE		
1468	006224	012767	000201	172614	MOV	#201,BUFLEN	:LENGTH OF BUFFER=201
1469	006232	004767	000570		JSR	%7,LODBUF	:LOAD THE BUFFER WITH INCREMENTING PATTERN
1470	006236	005367	172604		DEC	BUFLEN	:BUFLEN=200
1471	006242	016767	172600	172604	MOV	BUFLEN,WCLLEN	:PREPARE NUMBER FOR DRWC
1472	006250	005467	172600		NEG	WCLLEN	:2'S COMPLEMENT OF BUFLEN
1473	006254	016777	172574	172536	MOV	WCLLEN,ADRWC	:SET UP DRWC
1474							
1475	006262	016777	172554	172532	MOV	INBUF,ADRBA	:SET UP DRBA
1476	006270	012777	052525	172530	MOV	#52525,ADRDB	:SET UP DRDB
1477	006276	012777	007130	172530	MOV	#INTA,ADRINV	:INTERRUPT VECTOR TO INTA
1478	006304	016777	172522	172516	MOV	DRINL,ADRVS	:INTERRUPT VECTOR TO PRIORITY DRINL
1479	006312	005077	172500		CLR	APSW	:LET THE DR11-B INTERRUPT
1480	006316	012777	000113	172500	MOV	#113,ADRST	:IE, FNCT3, FNCT1, AND DO TO DRST
1481	006324	005267	172522		INC	BRWAIT	:USE A WAIT OR BR. INSTRUCTION
1482	006330	032767	000001	172514	BIT	#BIT0,BRWAIT	:BIT 0 CLEAR=BR.
1483	006336	001403			BEQ	DATONB	
1484	006340	000001			WAIT		:WAIT FOR INTERRUPT
1485							
1486	006342	000240			NOP		
1487							
1488	006344	000401			BR	+.4	
1489	006346	000777			DATONB: BR		
1490	006350	004767	000732		JSR	%7,DATOCK	:CHECK INBUF
1491							
1492					::*****		
1493					::	TEST 60 TEST THAT FUNCTION BITS INCREMENT WITH MAINT MODE TRANSFERS	
1494					::*****		
1495							
1496	006354	104400			SCOPE		
1497	006356	012767	000010	172462	MOV	#10,BUFLEN	:SET-UP BUFLEN FOR LODBUF AND CHKBUFF

```
1498 006364 016777 172452 172430      MOV      INBUF,ADRBA      ;SET-UP DRBA
1499 006372 004767 000430              JSR      %7,LODBUF      ;LOAD INBUF
1500 006376 004767 000462              JSR      %7,CHKBFF      ;LOAD CHKBUFF
1501 006402 005077 172416              CLR      @DRST          ;INIT FOR STARTING
1502 006406 012767 000001 172454      MOV      #1,FNCCNT      ;GET READY FOR CHECKING
1503 006414 012767 000001 172424      MOV      #1,BUFLEN      ;CHANGE IS NECESSARY FOR INTA ROUTINE
1504
1505 006422 016767 172414 172442      MOV      INBUF,INBUF1    ;SAVE INBUF
1506 006430 012777 007130 172376 MFLOOP: MOV      #INTA,@DRINV ;INTERRUPT VECTOR TO INTA
1507
1508 006436 016777 172370 172364      MOV      DRINL,@DRVS    ;INTERRUPT VECTOR PRIORITY TO DRINL
1509 006444 005077 172346              CLR      @PSW          ;LET THE DR11-B INTERRUPT
1510 006450 012777 177777 172342      MOV      #-1,@DRWC      ;SET-UP FOR 1 TRANSFER
1511 006456 052777 010101 172340      BIS      #10101,@DRST   ;MAINT, IE, AND DO TO DRST
1512
1513 006464 000001              WAIT                     ;WAIT FOR INTERRUPT
1514 006466 000240              NOP                     ;FAKE-OUT RETURN ADDRESS CHANGING
1515 006470 117701 172330      MOV      @DRST,%1        ;LOWER BYTE OF DRST TO R1
1516 006474 042701 000600      BIC      #600,%1        ;GET RID OF READY AND CYCLE BECAUSE OF MAINT MODE
1517 006500 006201              ASR      %1            ;MOVE IT RIGHT ONE PLACE
1518 006502 126701 172362      CMP      FNCCNT,%1      ;CHECK AGAINST FNCCNT
1519 006506 001401      BEQ      .+4            ;SHOULD BE EQUAL
1520 006510 104000      HLT                     ;FUNCTION BITS DIDN'T INCREMENT IN MAINT MODE
1521 006512 005267 172352      INC      FNCCNT          ;GET READY FOR NEXT PASS
1522 006516 022767 000010 172344      CMP      #10,FNCCNT     ;ONLY 10 BECAUSE FNCT3-1 GO TO ZERO
1523 006524 001404      BEQ      MFCHK          ;IF ITS EQUAL GO CHECK DATA
1524 006526 062767 000002 172306      ADD      #2,INBUF        ;FAKE-OUT INTA ROUTINE
1525 006534 000735      BR       MFLOOP        ;DO IT AGAIN
1526 006536 012767 000007 172302 MFCHK: MOV      #7,BUFLEN    ;SET UP FOR DATCHK (10 FNCT CHECKS, 7 TRANSFERS)
1527 006544 016767 172322 172270      MOV      INBUF1,INBUF   ;RESTORE INBUF
1528 006552 004767 000446      JSR      %7,DATCHK      ;CHECK DATA
1529
1530      ;*****
1531      ;TEST 61 TEST FOR 10 MAINT MODE TRANSFERS
1532      ;*****
1533 006556 104400      SCOPE
1534 006560 012767 000010 172260      MOV      #10,BUFLEN     ;BUFLEN=10
1535 006566 016767 172254 172260      MOV      BUFLEN,WLEN    ;PREPARE NUMBER FOR DRWC
1536 006574 005467 172254      NEG      WLEN           ;2'S COMPLEMENT OF BUFLEN
1537 006600 004767 000222      JSR      %7,LODBUF      ;LOAD IN BUFFER WITH INCREMENTING PATTERN
1538 006604 004767 000254      JSR      %7,CHKBFF      ;LOAD CHECK BUFFER WITH MODIFIED INCREMENTING PATTERN
1539 006610 016777 172240 172202      MOV      WLEN,@DRWC     ;SET UP DRWC
1540 006616 016777 172220 172176      MOV      INBUF,ADRBA    ;SET UP DRBA
1541 006624 012777 177777 172174      MOV      #-1,@DRDB      ;MAINT AIDE
1542 006632 012777 007130 172174      MOV      #INTA,@DRINV   ;INTERRUPT VECTOR TO INTA
1543 006640 016777 172166 172162      MOV      DRINL,@DRVS    ;INTERRUPT STATUS AT PRIORITY DRINL
1544 006646 005077 172144      CLR      @PSW          ;LET DR11-B INTERRUPT
1545 006652 012777 010101 172144      MOV      #10101,@DRST   ;MAINT, IE, AND DO TO DRST
1546 006660 000777      BR       ;WAIT FOR INTERRUPT
1547 006662 004767 000336      JSR      %7,DATCHK
1548
1549      ;*****
1550      ;TEST 62 TEST FOR 200 NPR TRANSFERS IN MAINT MODE
1551      ;*****
1552 006666 104400      SCOPE
1553 006670 012767 000200 172150      MOV      #200,BUFLEN    ;LENGTH OF BUFFER = 200
```


1554	006676	016767	172144	172150	MOV	BUFLN,WCLN	:PREPARE NUMBER FOR DRWC
1555	006704	005467	172144		NEG	WCLN	:2'S COMPLEMENT OF BUFLN
1556	006710	004767	000112		JSR	%7,LODBUF	:LOAD INBUF WITH INCREMENTING PATTERN
1557	006714	004767	000144		JSR	%7,CHKBFF	:LOAD CHKBUFF WITH MODIFIED INCREMENTED PATTERN
1558	006720	016777	172130	172072	MOV	WCLN,@DRWC	:SET UP DRWC
1559	006726	016777	172110	172066	MOV	INBUF,@DRBA	:SET UP DRBA
1560	006734	012777	000001	172064	MOV	#1,@DRDB	:MAINT AIDE
1561	006742	012777	007130	172064	MOV	#INTA,@DRINV	:INT VECTOR TO INTA
1562	006750	016777	172056	172052	MOV	DRINL,@DRVS	:INT VECTOR AT PRIORITY DRINL
1563	006756	005077	172034		CLR	@PSW	:LET THE DR11-B INTERRUPT
1564	006762	012777	010101	172034	MOV	#010101,@DRST	:FOLLOWING TO DRST: MAINT(12),IE(06),DO(00)
1565	006770	005267	172056		INC	BRWAIT	:USE A WAIT OR BR. INSTRUCTION
1566	006774	032767	000001	172050	BIT	#BIT0,BRWAIT	:SEE WHICH ONE
1567	007002	001403			BEQ	BRANCH	:BIT 0 CLEAR = BR.
1568	007004	000001			WAIT		
1569	007006	000240			NOP		
1570	007010	000401			BR	.+4	
1571	007012	000777			BR		
1572	007014	004767	000204		JSR	%7,DATCHK	:CHECK THAT CORRECT DATA WAS TRANSFERRED
1573	007020	104400			SCOPE		
1574	007022	000167	000356		JMP	END	:DO IT ALL AGAIN.
1575	007026	016702	172010		LODBUF: MOV	INBUF,%2	:MOVE STARTING ADDRESS OF INBUF TO R2
1576	007032	005067	172012		CLR	LENCHK	:CLEAR LENGTH CHECK
1577	007036	005022			CLR	(%2)+	:CLEAR STARTING ADDRESS OF INBUF AND INC BY 2
1578	007040	005267	172004		LOADA: INC	LENCHK	:INC LENGTH CHECK BY 1
1579	007044	026767	172000	171774	CMP	LENCHK,BUFLN	:CHECK FOR DONE
1580	007052	001403			BEQ	LDEXIT	:IS INBUF FILLED?
1581	007054	016722	171770		MOV	LENCHK,(%2)+	:LOAD NEXT BUFFER WORD
1582	007060	000767			BR	LOADA	:CONTINUE CHECKING
1583	007062	000207			LDEXIT: RTS	%7	:EXIT
1584	007064	016702	171754		CHKBFF: MOV	CHKBUFF,%2	:STARTING ADDRESS OF CHECK-BUFFER TO R2
1585	007070	005067	171754		CLR	LENCHK	:CLEAR LENGTH CHECK
1586	007074	005003			CLR	%3	:CLEAR R3
1587	007076	010322			CHKA: MOV	%3,(%2)+	:MOVE R3 TO CHKBUFF ADDRESS AND INC BY 2
1588	007100	010322			MOV	%3,(%2)+	:MOVE R3 TO NEXT CHKBUFF ADDRESS AND INC BY 2
1589	007102	062767	000002	171740	ADD	#2,LENCHK	:ADD 2 TO LENGTH CHECK
1590	007110	026767	171734	171730	CMP	LENCHK,BUFLN	:CHECK FOR DONE
1591	007116	100003			BPL	.+10	:IS CHECK-BUFFER FILLED?
1592	007120	062703	000002		ADD	#2,%3	:NEXT NUMBER FOR BUFFER
1593	007124	000764			BR	CHKA	:CONTINUE FILLING
1594	007126	000207			RTS	%7	:EXIT
1595	007130	042777	000100	171666	INTA: BIC	#BIT6,@DRST	:CLEAR IE
1596	007136	005777	171662		TST	@DRST	:CHECKING FOR ERROR
1597	007142	100001			BPL	.+4	:ERROR SET?
1598	007144	104000			HLT		:ERROR BIT IS SET
1599	007146	105777	171652		TSTB	@DRST	:CHECKING READY BIT
1600	007152	100401			BMI	.+4	:IS READY SET
1601	007154	104000			HLT		:FALSE INTERRUPT - ERROR AND READY ARE CLEAR
1602	007156	005777	171636		TST	@DRWC	:TEST1 FOR DRWC=0
1603	007162	001401			BEQ	.+4	:WAS IT EQUAL?
1604	007164	104000			HLT		:DRWC NOT =0
1605	007166	016702	171654		MOV	BUFLN,%2	:BUFFER LENGTH TO R2
1606	007172	066702	171650		ADD	BUFLN,%2	:NUMBER OF TRANSFERS TIMES 2
1607	007176	066702	171640		ADD	INBUF,%2	:CORRECT DRBA
1608	007202	027702	171614		CMP	@DRBA,%2	:CHECKING DRBA
1609	007206	001401			BEQ	.+4	:IS DRBA CORRECT?

```
1610 007210 104000 HLT ;DRBA NOT CORRECT
1611 007212 062716 000002 ADD #2,(%6) ;RETURN ADDRESS TO RETURN ADDRESS +2
1612 007216 004767 000042 JSR %7,NORMAL
1613 007222 000002 RTI ;EXIT
1614
1615 007224 016702 171614 DATCHK: MOV CHKBUFF,%2 ;STARTING ADDRESS OF CHECK BUFFER TO R2
1616 007230 016703 171606 MOV INBUF,%3 ;STARTING ADDRESS OF IN BUFFER TO R3
1617 007234 005067 171610 CLR LENCHK ;CLEAR LENGTH CHECK
1618 007240 005267 171604 COMPAR: INC LENCHK ;MAKE A COMPARISON
1619 007244 022223 CMP (%2)+,(%3)+ ;IS THE DATA CORRECT?
1620 007246 001401 BEQ .+4 ;BRANCH IF OK
1621 007250 104000 HLT ;BAD DATA
1622 007252 026767 171572 171566 CMP LENCHK,BUFLEN ;SEE IF THE BUFFER HAS BEEN CHECKED
1623 007260 001367 BNE COMPAR ;BUFFER CHECKED?
1624 007262 000207 RTS %7
1625 007264 012777 001030 171542 NORMAL: MOV #DRVS,@DRINV ;RESTORE DR11-B INTERRUPT VECTOR
1626 007272 005077 171532 CLR @DRVS ;RESTORE DR11-B INTERRUPT STATUS
1627 007276 012777 000340 171512 MOV #340,@PSW ;RESTORE PROC TO PRIORITY LEVEL 7
1628 007304 000207 RTS %7 ;EXIT
1629 007306 012702 052525 DATOCK: MOV #52525,%2 ;DATO NUMBER TO R2
1630 007312 016703 171524 MOV INBUF,%3 ;STARTING ADDRESS OF IN BUFFER TO R3
1631 007316 005067 171526 CLR LENCHK ;CLEAR LENGTH CHECK
1632 007322 005267 171522 COMPRR: INC LENCHK ;MAKE A COMPARISON
1633 007326 020223 CMP %2,(%3)+ ;IS THE DATA CORRECT?
1634 007330 001401 BEQ .+4 ;BRANCH IF OK
1635 007332 104000 HLT ;BAD DATA
1636 007334 026767 171510 171504 CMP LENCHK,BUFLEN ;SEE IF THE BUFFER HAS BEEN CHECKED
1637 007342 001367 BNE COMPRR ;BUFFER CHECKED?
1638 007344 020223 CMP %2,(%3)+ ;CHECK END OF BUFFER + 1
1639 007346 001001 BNE .+4 ;SEE IF TOO MANY WORDS WERE TRANSFERRED
1640 007350 104000 HLT ;TOO MANY
1641 007352 000207 RTS %7 ;EXIT
1642 007354 042777 000100 171442 ERRCHK: BIC #BIT6,@DRST ;CLEAR IE
1643 007362 005777 171436 TST @DRST ;CHECKING FOR ERROR
1644 007366 100001 BPL .+4 ;ERROR SET?
1645 007370 104000 HLT ;ERROR BIT IS SET
1646 007372 105777 171426 TSTB @DRST ;CHECKING READY BIT
1647 007376 100401 BMI .+4 ;IS RDY SET
1648 007400 104000 HLT ;FALSE ENTRY - ERROR AND READY ARE CLEAR
1649 007402 000207 RTS %7 ;EXIT
1650
1651 ;*****
1652 ; END OF PASS
1653 ;*****
1654 007404 012737 000207 177566 END: MOV #207,@#177566 ;RING BELL
1655 007412 105737 177564 TSTB @#177564
1656 007416 100375 BPL -4
1657 007420 005267 171450 INC PASCNT ;KEEP TRACK OF PASSES COMPLETED
1658 007424 012702 013017 MOV #SENPAS,%2 ;PRINT 'END PASS'
1659 007430 004767 003450 JSR %7,TTOUT
1660
1661 007434 042777 000020 171354 END1: BIC #20,@PSW ;CLEAR T-BIT
1662 007442 013702 000042 MOV @#42,%2
1663 007446 001405 BEQ TRTRAP
1664 007450 000005 RESET
```



```
1666 007452 004712          SENDAD: JSR      %7,(2)
1667 007454 000240          NOP
1668 007456 000240          NOP
1669 007460 000240          NOP
1670                                     ::*****
1671                                     ::ROUTINE TO CHECK FOR TRACE TRAP TO BE RUN WITH PROGRAM
1672                                     ::*****
1673
1674 007462 004767 002716 171320 TRTRAP: JSR      %7,CKSWR      ;CHECK FOR CONT G
1675 007466 032777 010000          BIT      #10000,@SR      ;SHOULD WE RUN WITH TRACE TRAP
1676 007474 001417          BEQ      YESTR      ;YES
1677 007476 005767 000104          TST      YESTR1      ;NO, HAVE WE RUN WITH TRACE TRAP ON?
1678 007502 001411          BEQ      TRPA      ;IF SO RESTORE PREVIOUS CONTENTS
1679 007504 016767 000076 170302          MOV      YESTR1,14
1680 007512 016767 000072 170276          MOV      YESTR2,16
1681 007520 042777 000020 171270          BIC      #20,@PSW      ;CLEAR TRACE TRAP
1682 007526 000167 171442          TRPA:  JMP      BEGIN      ;START OF TEST WITH TRACE OFF
1683 007532 000000          TRPB:  0
1684
1685                                     ::*****
1686                                     ::SAVE OLD CONTENTS, SET UP FOR TRACE TRAP
1687                                     ::*****
1688 007534 016767 170254 000044 YESTR:  MOV      14,YESTR1      ;SAVE ODT PC
1689 007542 016767 170250 000040          MOV      16,YESTR2      ;SAVE ODT STATUS
1690 007550 012767 007612 170236          MOV      #YESRT,14      ;NEW TRAP VECTOR
1691 007556 005067 170234          CLR      16      ;NEW CONDITION CODES
1692 007562 005077 171230          CLR      @PSW
1693 007566 005167 177740          COM      TRPB
1694 007572 100403          BMI      .+10
1695 007574 052777 000020 171214          BIS      #20,@PSW      ;SET TRACE TRAP
1696 007602 000167 171366          JMP      BEGIN      ;START OF TEST WITH TRACE ON
1697
1698 007606 000000          YESTR1: 0      ;STORAGE FOR ODT PC
1699 007610 000000          YESTR2: 0      ;STORAG FOR ODT STATUS
1700 007612 000002          YESRT:  RTI      ;RETURN TO PROGRAM FROM TRAP
1701 007614 000000          HALT      ;RTI FAILED
1702
1703                                     ::*****
1704                                     ::BUS TO BUS TEST (DR11-B TO DR11-B)
1705                                     ::*****
1706
1707
1708
1709          000000          R0=%0
1710          000001          R1=%1
1711          000002          R2=%2
1712          000003          R3=%3
1713          000004          R4=%4
1714          000005          R5=%5
1715          000006          R6=%6
1716          000007          R7=%7
1717          000007          PC=%7
1718          000006          SP=%6
1719
1720          000001          GO=1
1721          000002          FNCT1=2      ;OUTPUT MODE
```

```

1722      000004      FNCT2=4      :OUTPUT DIRECTION
1723      000010      FNCT3=10      :OUTPUT INTER REQ
1724      000020      XBA16=20
1725      000040      XBA17=40
1726      000100      IE=100
1727      000200      READY=200
1728      000400      CYCLE=400
1729      001000      DSTATA=1000      :INPUT MODE
1730      002000      DSTATB=2000      :INPUT DIRECTION
1731      004000      DSTATC=4000      :INPUT INTR REQ
1732      010000      MAINT=10000
1733      020000      ATTN=20000
1734      040000      NEX=40000
1735      100000      ERROR=100000
1736
1737
1738
1739      007616      000004      NWRDXF: 4      :# OF WORDS TRANSFERRED UNDER FLAG CONTROL PRIOR TO NPR
1740      007620      000000      NEXJOB: 0      :HOLDS ADDRESS OF READY INTERRUPT ROUTINE
1741      007622      000000      0      :HOLDS ADDRESS OF ERROR INTERRUPT ROUTINE
1742      007624      000000      JBFLAG: 0      :JOB FLAG
1743      007626      000000      JBCNT: 0      :JOB COUNT
1744      :*****
1745      :MASTER START
1746      :*****
1747      007630      000240      MS1:      NOP
1748      007632      005067      177770      CLR      JBCNT      :CLEAR JOB COUNT
1749      007636      012706      014210      MOV      #BUFF,R6      :SETUP STACK
1750      007642      004767      000256      JSR      R7,SETVEC      :SET UP INTERRUPT VECTORS
1751      007646      005067      177752      CLR      JBFLAG      :CLEAR JOB FLAG
1752      007652      012767      010304      177740      MOV      #JOB0,NEXJOB      :DO JOB0 FIRST
1753      007660      012767      000000      177734      MOV      #0,NEXJOB+2      :NO ERROR RECOVERY
1754      007666      012777      000340      171122      MOV      #340,@PSW      :LOCK OUT INTERRUPTS
1755      007674      012777      000100      171122      MOV      #IE,@DRST      :SET INTERRUPT ENABLE
1756      007702      005077      171110      CLR      @PSW      :DROP PRIOTIRY TO ZERO
1757      007706      000425      BR      BACKGD      :WAIT FOR JOBS IN BACKGROUND
1758
1759
1760      :*****
1761      :SLAVE START
1762      :*****
1763
1764      007710      000240      SS1:      NOP
1765      007712      012706      014210      MOV      #BUFF,R6
1766      007716      004767      000202      JSR      R7,SETVEC      :SET UP INTERRUPT VECTORS
1767      007722      005067      177676      CLR      JBFLAG
1768      007726      012767      010526      177664      MOV      #SJOB1,NEXJOB      :FOR READY INTERRUPT
1769      007734      012767      010544      177660      MOV      #SJOB2,NEXJOB+2      :FOR ERROR INTERRUPT
1770      007742      012777      000340      171046      MOV      #340,@PSW      :RAISE CP PRIORITY TO 7
1771      007750      012777      000100      171046      MOV      #IE,@DRST      :SET INTERRUPT ENABLE
1772      007756      005077      171034      CLR      @PSW      :DROP CP PRIORITY TO 0 AND ENTER BACKGROUND
1773
1774      :*****
1775      :BACKGROUND PROGRAM; WAITS FOR JBFLAG TO SET
1776      :*****
1777

```



```
1778 007762 005200      BACKGD: INC      R0
1779 007764 005201      INC      R1
1780 007766 005202      INC      R2
1781 007770 005203      INC      R3
1782 007772 005204      INC      R4
1783 007774 005205      INC      R5
1784 007776 020005      CMP      R0,R5
1785 010000 001402      BEQ      .+6
1786 010002 104000      HLT
1787 010004 000000      HALT      ;BACKGROUND TEST FAILED
1788 010006 020104      CMP      R1,R4
1789 010010 001402      BEQ      .+6
1790 010012 104000      HLT
1791 010014 000000      HALT      ;BACKGROUND TEST FAILED
1792 010016 020203      CMP      R2,R3
1793 010020 001402      BEQ      .+6
1794 010022 104000      HLT
1795 010024 000000      HALT      ;BACKGROUND TEST FAILED
1796 010026 005767 177572 TST      JBFLAG      ;ANY JOBS?
1797 010032 001753      BEQ      BACKGD      ;BRANCH IF NONE
1798 010034 004567 000032 JSR      R5,SAVALL      ;YES & EXECUTE JOB WHOSE ADDRESS IS IN JBFLAG
1799 010040 005067 177554 CLR      NEXJOB
1800 010044 005067 177552 CLR      NEXJOB+2
1801 010050 016767 177550 MOV      JBFLAG,34
1802 010056 005067 177542 CLR      JBFLAG
1803 010062 104400      TRAP      ;TRAP THROUGH JBFLAG AT 34
1804 010064 004567 000016 JSR      R5,RESALL
1805 010070 000734      BR      BACKGD
1806
1807      ;*****
1808      ; SUBROUTINE TO PUSH ALL REGISTERS ONTO THE STACK
1809      ;*****
1810
1811 010072 010446      SAVALL: MOV      R4,-(R6)      ;R5 WAS PUSHED BY JSR
1812 010074 010346      MOV      R3,-(R6)
1813 010076 010246      MOV      R2,-(R6)
1814 010100 010146      MOV      R1,-(R6)
1815 010102 010046      MOV      R0,-(R6)
1816 010104 000115      JMP      (R5)      ;R5 HOLDS RETURN ADDRESS
1817
1818      ;*****
1819      ; SUBROUTINE TO POP ALL REGISTERS OFF THE STACK
1820      ;*****
1821
1822 010106 005726      RESALL: TST      (R6)+
1823 010110 012600      MOV      (R6)+,R0
1824 010112 012601      MOV      (R6)+,R1
1825 010114 012602      MOV      (R6)+,R2
1826 010116 012603      MOV      (R6)+,R3
1827 010120 012604      MOV      (R6)+,R4
1828 010122 000205      RTS      R5
1829
1830      ;*****
1831      ; ROUTINE TO SET UP INTERRUPT VECTORS
1832      ;*****
1833
```

```

1834 010124 016700 170704      SETVEC: MOV      DRINV,R0      ;R0 IS VECTOR ADDRRESS
1835 010130 012720 010204      MOV      #DRINS,(R0)+    ;PUT SERVICE ADDRESS INTO VECTOR
1836 010134 016710 170672      MOV      DRINL,(R0)      ;PUT PRIORITY INTO VECTOR+2
1837 010140 012767 011506 167662 MOV      #PRINT,30      ;SET UP EMT ADDRESS
1838 010146 016767 170660 167656 MOV      DRINL,32      ;SET UP EMT PRIORITY LEVEL
1839 010154 005067 167654      CLR      34
1840 010160 016767 170646 167650 MOV      DRINL,36      ;SET UP TRAP ADDRESS
1841 010166 005000      CLR      R0      ;INITIALIZE REGISTERS
1842 010170 005001      CLR      R1
1843 010172 005002      CLR      R2
1844 010174 005003      CLR      R3
1845 010176 005004      CLR      R4
1846 010200 005005      CLR      R5
1847 010202 000207      RTS      R7
1848
1849      ;*****
1850      ;PRIMARY INTERRUPT SERVICE ROUTINE.
1851      ;SETS UP JBLFAG WITH ADDRESS OF JOB TO BE RUNSTARS
1852      ;*****
1853 010204 005767 177414      DRINS:  TST      JBFLAG      ;HAS THE PREVIOUS INTERRUPT BEEN SERVICED?
1854 010210 001402      BEQ      DRIN0
1855 010212 104000      HLT
1856 010214 000000      HALT
1857 010216 032777 004000 170600 DRIN0:  BIT      #DSTATC,ADRST    ;NO
1858 010224 001411      BEQ      DRIN3      ;CHECK FOR ERROR
1859 010226 005767 177370      TST      NEXJOB+2      ;BRANCH IF NO ERROR
1860 010232 001002      BNE      DRIN1      ;IS THERE AN ERROR SERVICE ROUTINE?
1861 010234 104000      HLT      ;BRANCH IF THERE IS.
1862 010236 000000      HALT
1863 010240 016767 177356 177356 DRIN1:  MOV      NEXJOB+2,JBFLAG    ;ERROR INTERRUPT, NO ERROR SERVICE.
1864 010246 000002      RTI      ;SET UP JOBFLAG WITH ADDRESS OF SERVICE ROUTINE
1865 010250 105777 170550      DRIN3:  TSTB     ADRST      ;CHECK READY
1866 010254 100402      BMI      DRIN2      ;BRANCH IF SET
1867 010256 104000      HLT
1868 010260 000000      HALT
1869 010262 005767 177332      DRIN2:  TST      NEXJOB      ;INTERRUPT WITHOUT ERROR OR READY
1870 010266 001002      BNE      .+6      ;IS THERE A READY SERVICE ROUTINE
1871 010270 104000      HLT      ;BRANCH IF THERE IS.
1872 010272 000000      HALT
1873 010274 016767 177320 177322 MOV      NEXJOB,JBFLAG    ;READY INTERRUPT, NO READY SERVICE
1874 010302 000002      RTI      ;SET UP JOBFLAG WITH SERVICE ROUTINE ADDRESS
1875
1876      ;*****
1877      ;MASTER'S INTERRUPT SERVICE ROUTINES
1878      ;ROUTINE A, SEGMENT 0
1879      ;FILL BUFFER AND TRANSMIT
1880      ;*****
1881 010304 012700 011450      JOBA0:  MOV      #LISTA,R0      ;R0 IS XMIT LIST ADDRESS
1882 010310 012701 011462      MOV      #LISTA1,R1     ;LISTA1 WILL BE REC LIST
1883 010314 012021      MOV      (R0)+,(R1)+    ;START WITH BUS ADDRESSES EQUAL
1884 010316 012002      MOV      (R0)+,R2      ;R2 HOLDS WORD COUNT OF XMIT
1885 010320 010211      MOV      R2,(R1)      ;MAKE REC WORD COUNT THE SAME
1886 010322 005402      NEG      R2      ;MAKE WORD COUNT POSITIVE
1887 010324 006302      ASL      R2      ;TRANSFORM INTO BYTE COUNT
1888 010326 060241      ADD      R2,-(R1)      ;ADD TO REC BUS ADDRESS
1889 010330 005010      CLR      (R0)      ;CLEAR OFFSET IN XMIT LIST

```



```
1890 010332 024040          CMP      -(R0),-(R0)      ;LEAVE R0=LISTA=XMIT LIST
1891
1892 010334 022767 000100 177264      CMP      #100,JBCNT      ;ENOUGH PASSES FOR BELL?
1893 010342 003010          BGT      JOBA0A      ;BRANCH IF NOT ENOUGH
1894 010344 105737 177564      JOBA0B: TSTB      @#177564      ;TTY READY?
1895 010350 100375          BPL      JOBA0B
1896 010352 012737 000207 177566      MOV      #207,@#177566      ;RING BELL
1897 010360 005067 177242          CLR      JBCNT      ;RESET JOB COUNT
1898 010364 004767 000704      JOBA0A: JSR      R7,SETBUF      ;FILL UP XMIT BUFFER WITH SPECIAL BINARY COUNT
1899 010370 012700 011450          MOV      #LISTA,R0
1900 010374 004767 000370          JSR      R7,MXMIT      ;TRANSMIT DATA TO SLAVE
1901 010400 012767 010416 177212      MOV      #JOBA1,NEXJOB      ;JOBA1 IS NEXT
1902 010406 012767 000000 177206      MOV      #0,NEXJOB+2      ;NO ERROR RECOVERY
1903 010414 000002          RTI      ;RETURN TO BACKGROUND VIA TRAP
1904
1905
1906
1907
1908
1909
1910 010416 012700 011462      JOBA1: MOV      #LISTA1,R0      ;PUT REC LIST ADDRESS INTO R0
1911 010422 004767 000574          JSR      R7,FLUSH      ;FLUSH BUFFER
1912 010426 012700 011462          MOV      #LISTA1,R0
1913 010432 004767 000406          JSR      R7,MREC      ;RECEIVE DATA FROM SLAVE
1914 010436 012767 010454 177154      MOV      #JOBA2,NEXJOB      ;JOBA2 IS NEXT
1915 010444 012767 000000 177150      MOV      #0,NEXJOB+2      ;NO ERROR RECOVERY
1916 010452 000002          RTI
1917
1918
1919
1920
1921
1922
1923 010454 012700 011450      JOBA2: MOV      #LISTA,R0      ;XMIT BUFFER LIST
1924 010460 012701 011462          MOV      #LISTA1,R1      ;REC BUFFER LIST
1925 010464 004767 000652          JSR      R7,BUFCHK      ;COMPARE THE TWO BUFFERS
1926 010470 042777 000100 170326      BIC      #IE,@DRST      ;GLITCH INTERRUPT
1927 010476 012777 000100 170320      MOV      #IE,@DRST
1928 010504 012767 010304 177106      MOV      #JOBA0,NEXJOB      ;REPEAT JOBA0
1929 010512 012767 000000 177102      MOV      #0,NEXJOB+2
1930 010520 005267 177102          INC      JBCNT      ;ADVANCE COUNT
1931 010524 000002          RTI
1932
1933
1934
1935
1936
1937
1938
1939 010526 012767 000000 177064      SJOB1: MOV      #0,NEXJOB      ;NO MORE READY INTERRUPTS
1940 010534 012767 010544 177060      MOV      #SJOB2,NEXJOB+2 ;UNTIL ATTN INTERRUPT
1941 010542 000002          RTI
1942
1943
1944
1945
```

```
*****
ROUTINE A, SEGMENT 1
FLUSH A BUFFER AND RECEIVE DATA
*****

ROUTINE A, SEGMENT 2
CHECKS TRANSMITTED DATA WITH RECEIVED
*****

SLAVE'S INTERRUPT SERVICE ROUTINES
JOB1: IGNORE FIRST READY INTERRUPT
*****

JOB2: WAIT FOR COMMAND
*****
```

```
1946
1947 010544 032777 004000 170252 SJOB2: BIT #DSTATC,ADRST ;TEST FOR INTER
1948 010552 001002 SJOB2A: BNE SJOB2A
1949 010554 104000 HLT
1950 010556 000000 SJOB2A: HALT ;ERROR OTHER THAN DSTATC
1951 010560 005001 SJOB2A: CLR R1 ;SET UP FOR PARAMETERS
1952 010562 016702 170236 MOV DRST,R2 ;R2 IS STATUS ADDRESS
1953 010566 012703 000010 MOV #FNCT3,R3 ;R3 IS FUNCTION BIT 3
1954 010572 012704 004000 MOV #DSTATC,R4 ;R4 IS INTERRUPT BIT
1955 010576 016705 170224 MOV DRDB,R5 ;R5 IS DATA BUFFER ADDRESS
1956 010602 012700 011476 MOV #LISTB+2,R0 ;STORE PARAMETERS HERE STARTING WITH WORD COUNT
1957 010606 004767 000122 JSR R7,HNDSHK ;GET PARAMETERS
1958 010612 012700 011474 MOV #LISTB,R0 ;R0 IS TOP OF LIST
1959 010616 016010 000004 MOV 4(R0),(R0) ;MOVE OFFSET TO TOP
1960 010622 066710 170214 ADD INBUF,(R0) ;TOP OF LIST IS BUFFER START + OFFSET
1961 010626 012077 170170 MOV (R0)+,ADRBA ;SET UP BUS ADDRESS
1962 010632 012077 170162 MOV (R0)+,ADRW ;SET UP WORD COUNT
1963 010636 005077 170162 CLR ADRST ;CLEAR ALL FUNCTION BITS
1964 010642 032777 002000 170154 BIT #DSTATB,ADRST ;WHICH DIRECTION
1965 010650 001405 SJOB2B: BEQ SJOB2C ;BRANCH IF RECIEVE (LEAVE FNCT1 CLEAR FOR DATI'S)
1966 010652 032777 000400 170144 SJOB2B: BIT #CYCLE,ADRST ;WAIT FOR MASTER TO SET CYCLE
1967 010660 001774 BEQ SJOB2B ;BRANCH IF NOT SET
1968 010662 000412 BR SJOB2D ;GO DO THE COMAND
1969 010664 012700 011474 SJOB2C: MOV #LISTB,R0
1970 010670 004767 000356 JSR R7,BLUSH ;BLUSH THE BUFFER
1971 010674 052777 000004 170122 BIS #FNCT2,ADRST ;SET FNCT2 FOR DATO'S
1972 010702 042777 000400 170114 BIC #CYCLE,ADRST ;CLEAR CYCLE
1973 010710 052777 000101 170106 SJOB2D: BIS #IE!GO,ADRST ;EXECUTE COMMAND AND INTERRUPT WHEN DONE
1974 010716 012767 010526 176674 MOV #SJOB1,NEXJOB ;IGNORE READY INTERRUPT
1975 010724 012767 010544 176670 MOV #SJOB2,NEXJOB+2 ;WAIT FOR ATTN INTERRUPT
1976 010732 000002 RTI
1977
1978 ;:*****
1979 ;: SLAVE ROUTINE TO ACCEPT PARAMETERS FROM MASTER
1980 ;:*****
1981
1982 010734 011515 HNDSHK: MOV (R5),(R5) ;ECHO PARAMETER
1983 010736 011520 MOV (R5),(R0)+ ;STORE PARAMETER
1984 010740 050312 BIS R3,(R2) ;REPLY WITH FNCT3
1985 010742 030412 HNDSH1: BIT R4,(R2) ;WAIT FOR ATTN TO DROP
1986 010744 001376 BNE HNDSH1
1987 010746 040312 BIC R3,(R2) ;DROP FNCT3
1988 010750 005201 INC R1 ;CHECK NUMBER
1989 010752 020167 176640 CMP R1,NWRDXF
1990 010756 002401 BLT HNDSH2 ;BRANCH IF NOT DONE YET
1991 010760 000207 RTS R7
1992 010762 030412 HNDSH2: BIT R4,(R2) ;WAIT FOR NEXT WORD
1993 010764 001776 BEQ HNDSH2 ;BRANCH IF ATTN CLEAR
1994 010766 000762 BR HNDSHK ;GET ANOTHER PARAMETER
1995
1996 ;:*****
1997 ;: MASTER TRANSMIT ROUTINE
1998 ;: ENTER WITH TRANSFER LIST IN R0
1999 ;:*****
2000
2001 010770 005777 170030 MXMIT: TST ADRST ;MAKE SURE ERROR IS CLEAR
```



```
2002 010774 100002          BPL      MXMIT1      ;AND READY IS SET
2003 010776 104000          HLT
2004 011000 000000          HALT      ;ERROR IS SET
2005 011002 105777 170016  MXMIT1: TSTB      @DRST
2006 011006 100402          BMI      MXMIT2
2007 011010 104000          HLT
2008 011012 000000          HALT      ;READY NOT SET
2009 011014 012777 000000 170002 MXMIT2: MOV      #0,@DRST      ;SET UP FUNCTION FOR DATI'S
2010 011022 004767 000102      JSR      R7,PRMXFR      ;TRANSFER PARAMETERS OF LIST WHOSE ADDRESS IS IN R0
2011 011026 042777 000400 167770 BIC      #CYCLE,@DRST      ;MAKE SURE CYCLE IS CLEAR
2012 011034 052777 000101 167762 BIS      #IE!GO,@DRST      ;EXECUTE COMMAND AND INTERRUPT WHEN DONE
2013 011042 000207          RTS      R7
2014
2015
2016
2017
2018
2019 011044 005777 167754  MREC:   TST      @DRST      ;MAKE SURE ERROR IS CLEAR
2020 011050 100002          BPL      MREC1      ;AND READY SET.
2021 011052 104000          HLT
2022 011054 000000          HALT      ;ERROR SET
2023 011056 105777 167742  MREC1:  TSTB      @DRST
2024 011062 100402          BMI      MREC2
2025 011064 104000          HLT
2026 011066 000000          HALT      ;READY CLEAR
2027 011070 012777 000004 167726 MREC2:  MOV      #FNCT2,@DRST      ;SET UP FUNCTION FOR DATO'S
2028 011076 004767 000026      JSR      R7,PRMXFR      ;TRANSFER PARAMETERS OF LIST WHOSE ADDRESS IS IN R0
2029 011102 032777 002000 167714 MREC3:  BIT      #DSTATB,@DRST      ;WAIT FOR SLAVE TO CLEAR DIRECTION
2030 011110 001374          BNE      MREC3      ;BRANCH IF SET
2031 011112 042777 000400 167704 BIC      #CYCLE,@DRST      ;CLEAR CYCLE
2032 011120 052777 000101 167676 BIS      #IE!GO,@DRST      ;EXECUTE COMMAND AND INTERRUPT WHEN DONE.
2033 011126 000207          RTS      R7
2034
2035
2036
2037
2038
2039
2040 011130 012077 167666  PRMXFR: MOV      (R0)+,@DRBA      ;FIRST WORD IN LIST IS ADDRESS
2041 011134 011077 167660      MOV      (R0),@DRWC      ;SECOND WORD IN LIST IS WORD COUNT
2042 011140 005001          CLR      R1      ;R1 COUNTS PARAMETERS TRANSFERRED
2043 011142 016702 167656      MOV      DRST,R2      ;R2 IS THE STATUS ADDRESS
2044 011146 012703 000010      MOV      #FNCT3,R3      ;R3 USED FOR FUNCTION BIT 3
2045 011152 012704 004000      MOV      #DSTATC,R4      ;R4 USED FOR INTERRUPT BIT
2046 011156 016705 167644      MOV      DRDB,R5      ;R5 IS THE DATA BUFFER ADDRESS
2047 011162 011015          PRMXF1: MOV      (R0),(R5)      ;SET UP DRDB WITH PARAMETER
2048 011164 050312          BIS      R3,(R2)      ;CALL SLAVE'S ATTN
2049 011166 030412          PRMXF2: BIT      R4,(R2)      ;WAIT FOR REPLY
2050 011170 001776          BEQ      PRMXF2      ;BRANCH IF ATTN CLEAR
2051 011172 022015          CMP      (R0)+,(R5)      ;COMPARE PARAMETER SENT WITH SLAVE'S ECHO
2052 011174 001402          BEQ      PRMXF3      ;BRANCH IF EQUAL
2053 011176 104000          HLT
2054 011200 000000          HALT      ;PARAMETER DID NOT ECHO
2055 011202 040312          PRMXF3: BIC      R3,(R2)      ;DROP SLAVE'S ATTN
2056 011204 030412          PRMXF4: BIT      R4,(R2)      ;WAIT FOR REPLY
2057 011206 001376          BNE      PRMXF4      ;BRANCH IF ATTN SET
```

```

2058 011210 005201          INC      R1          ;ADVANCE PARAMETER COUNT
2059 011212 020167 176400  CMP      R1,NWRDXF ;ALL PARAMETER XFERRED?
2060 011216 002761          BLT      PRMXF1    ;BRANCH IF NOT DONE
2061 011220 000207          RTS      R7          ;RETURN WHEN ALL PARAMETERS TRANSFERRED AND CHECKED.
2062
2063  ;*****
2064  ;ROUTINE TO CLEAR BUFFER
2065  ;ENTER WITH R0 POINTING TO TRANSFER LIST
2066  ;*****
2067
2068 011222 005005          FLUSH: CLR      R5          ;SET R5 TO ZIP
2069 011224 004767 000030  JSR      R7,BSETUP ;SET UP REGISTERS
2070 011230 004767 000004  FLUSH1: JSR      R7,BUFPUT ;STORE ITEM IN BUFFER
2071 011234 002775          BLT      FLUSH1    ;
2072 011236 000436          BR       BUFOUT    ;STOP WHEN BUFFER FULL
2073
2074 011240 010521          BUFPUT: MOV     R5,(R1)+ ;PUT R5 INTO BUFFER
2075 011242 060503          ADD      R5,R3      ;INCLUDE IN CHECKSUM
2076 011244 005504          ADC      R4        ;
2077 011246 005202          INC      R2        ;ADVANCE WORD COUNT
2078 011250 000207          RTS      R7        ;RETURN WITH STATUS SET
2079
2080  ;*****
2081  ;ROUTINE TO FILL BUFFER WITH ALL ONE'S
2082  ;ENTER WITH R0 POINTING TO TRANSFER LIST
2083  ;*****
2084 011252 012705 177777  BLUSH: MOV     #-1,R5 ;SET R5 TO ALL ONE'S
2085 011256 000762          BR       FLUSH+2    ;
2086
2087  ;REGISTER SETUP ROUTINE
2088 011260 012001          BSETUP: MOV     (R0)+,R1 ;R1 IS BUS ADDRESS
2089 011262 012002          MOV      (R0)+,R2 ;R2 IS WORD COUNT
2090 011264 005720          TST      (R0)+    ;SKIP OVER OFFSET
2091 011266 005003          CLR      R3        ;CLEAR LOW CHECKSUM
2092 011270 005004          CLR      R4        ;CLEAR HIGH CHECKSUM
2093 011272 000207          RTS      R7        ;RETURN
2094
2095  ;*****
2096  ;ROUTINE TO FILL BUFFER WITH FLOATING 1'S AND FLOATING 0'S PATTERN
2097  ;ENTER WITH R0 POINTING TO TRANSFER LIST
2098  ;*****
2099
2100 011274 004767 177760  SETBUF: JSR      R7,BSETUP ;SET UP REGISTERS
2101 011300 012705 000001  SETBF1: MOV     #1,R5 ;SET UP R5 WITH START PATTERN
2102 011304 004767 177730  SETBF2: JSR      R7,BUFPUT ;NUMBER TO BUFFER
2103 011310 002011          BGE      BUFOUT    ;
2104 011312 005105          COM      R5        ;
2105 011314 004767 177720  JSR      R7,BUFPUT ;COMPLEMENT OF NUMBER TO BUFFER
2106 011320 002005          BGE      BUFOUT    ;
2107 011322 005105          COM      R5        ;
2108 011324 000241          CLC          ;MOVE PATTERN ONE BIT
2109 011326 006105          ROL      R5        ;POSITION TO THE LEFT
2110 011330 103763          BCS      SETBF1    ;IF CARRY SET, START PATTERN OVER
2111 011332 000764          BR       SETBF2    ;
2112 011334 010320          BUFOUT: MOV     R3,(R0)+ ;MOVE LOW CHECK TO LIST
2113 011336 010420          MOV      R4,(R0)+ ;MOVE HIGH CHECK TO LIST

```



```

2114 011340 000207          RTS      R7          ;RETURN
2115
2116          ;*****
2117          ;ROUTINE TO COMPARE TWO BUFFERS
2118          ;BUFFER LIST #1 IS IN R1
2119          ;BUFFER LIST #2 IS IN R0.  #2 CHECKED FOR PROPER CHECKSUM
2120          ;*****
2121
2122 011342 010046          BUFCHK: MOV     R0,-(R6)      ;SAVE LIST#2 ON STACK
2123 011344 010146          MOV     R1,-(R6)      ;SAVE LIST#1 ON STACK
2124 011346 004767 000036  JSR     R7,CHKSUM    ;VERIFY INTEGRITY OF CHECK BUFFER (#2)
2125 011352 012600          MOV     (R6)+,R0      ;RECOVER 1ST LIST POINTER
2126 011354 012001          MOV     (R0)+,R1      ;BUS ADDRESS #1
2127 011356 012002          MOV     (R0)+,R2      ;WORD COUNT #1
2128 011360 012600          MOV     (R6)+,R0      ;RECOVER 2ND LIST POINTER
2129 011362 012003          MOV     (R0)+,R3      ;BUS ADDRESS #2
2130 011364 020220          CMP     R2,(R0)+      ;COMPARE WORD COUNTS
2131 011366 003402          BLE     BUFCK1        ;BRANCH IF EQUAL OR LESS THAN
2132 011370 104000          HLT
2133 011372 000000          HALT
2134
2135 011374 022123          BUFCK1: CMP     (R1)+,(R3)+  ;COMPARE BUFFERS
2136 011376 001401          BEQ     BUFCK2
2137 011400 104000          HLT                    ;DATA ERROR
2138 011402 005202          BUFCK2: INC     R2          ;ADVANCE COUNT
2139 011404 002773          BLT     BUFCK1        ;BRANCH IF NOT DONE
2140 011406 000207          RTS      R7          ;RETURN; STACK IS CLEAR
2141
2142          ;*****
2143          ;ROUTINE TO CHECK SUM OF BUFFER
2144          ;ENTER WITH R0 POINTING TO TRANSFER LIST
2145          ;*****
2146 011410 004767 177644  CHKSUM: JSR     R7,BSETUP    ;SET UP REGISTERS
2147 011414 004767 000016  CHKSM1: JSR     R7,GETBUF    ;GET ITEM FROM BUFFER
2148 011420 002775          BLT     CHKSM1        ;BRANCH IF NOT DONE
2149 011422 020320          CMP     R3,(R0)+      ;COMPARE LOW ORDER CHECKS
2150 011424 001003          BNE     CHKSM2
2151 011426 020420          CMP     R4,(R0)+      ;COMPARE HIGH ORDER CHECKS
2152 011430 001001          BNE     CHKSM2
2153 011432 000207          RTS      R7          ;RETURN IF CHECKSUM OK.
2154 011434 104000          CHKSM2: HLT          ;ORIGINAL BUFFER CHECKSUM DOES NOT AGREE WITH PRESENT
2155
2156 011436 012105          GETBUF: MOV     (R1)+,R5    ;GET ITEM OUT OF BUFFER
2157 011440 060503          ADD     R5,R3          ;ADD TO CHECKSUM
2158 011442 005504          ADC     R4           ;
2159 011444 005202          INC     R2          ;ADVANCE WORD COUNT
2160 011446 000207          RTS      R7
2161
2162 011450 014212          LISTA: XINBUF    ;START OF XMIT BUFFER
2163 011452 177605          -123.          ;WORD COUNT
2164 011454 000000          0              ;OFFSET
2165 011456 000000          0              ;CHECKSUM LOW
2166 011460 000000          0              ;CHECKSUM HIGH
2167
2168 011462 000000          LISTA1: 0          ;START OF REC BUFFER
2169 011464 000000          0              ;WORD COUNT

```

```
2170 011466 000000      0      :OFFSET
2171 011470 000000      0      :CHECKSUM LOW
2172 011472 000000      0      :CHECKSUM HIGH
2173
2174 011474 014212      LISTB: XINBUF      :SLAVE'S ECHO BUFFER
2175 011476 000000      0
2176 011500 000000      0
2177 011502 000000      0
2178 011504 000000      0
2179
2180
2181
2182
2183
2184
2185
2186
2187 011506 004767 000672      PRINT: JSR      %7,CKSWR
2188 011512 037727 167276 020000      BIT      @SR,#20000      :TEST FOR INHIBIT PRINT OUT
2189 011520 001067      BNE      1$      :IF SO, BRANCH OVER PRINT
2190 011522 012667 000204      MOV      (6)+,SAVPC      :PC OF FAILING ROUTINE
2191 011526 012667 000202      MOV      (6)+,SAVCC      :CC OF ERROR CONDITION
2192 011532 024646      CMP      -(6),-(6)      :REPOSITION THE STACK
2193 011534 012777 000215 167324      MOV      #215,@TPB      :CR
2194 011542 105777 167316      TSTB     @TPS
2195 011546 100375      BPL      .-4
2196 011550 012777 000212 167310      MOV      #212,@TPB      :LINE FEED
2197 011556 105777 167302      TSTB     @TPS
2198 011562 100375      BPL      .-4
2199 011564 010267 000134      MOV      %2,SAVR2      :SAVE R2
2200 011570 010367 000132      MOV      %3,SAVR3      :SAVE R3
2201 011574 010467 000130      MOV      %4,SAVR4      :SAVE R4
2202 011600 016702 000126      MOV      SAVPC,%2
2203 011604 004767 000126      JSR      %7,PRTAB      :PRINT OCTAL NUMBER
2204 011610 012777 000240 167250      MOV      #240,@TPB
2205 011616 105777 167242      TSTB     @TPS      :SPACE BETWEEN WORDS
2206 011622 100375      BPL      .-4
2207 011624 016702 000104      MOV      SAVCC,%2
2208 011630 004767 000102      JSR      %7,PRTAB      :PRINT OCTAL NUMBER
2209 011634 012777 000240 167224      MOV      #240,@TPB      :PRINT SPACE
2210 011642 105777 167216      TSTB     @TPS      :PRINTER DONE
2211 011646 100375      BPL      .-4      :BRANCH WHEN NOT DONE
2212 011650 017702 167150      MOV      @DRST,%2      :GET DR11B STATUS
2213 011654 004767 000056      JSR      %7,PRTAB      :PRINT OCTAL NUMBER
2214 011660 016702 000040      MOV      SAVR2,%2
2215 011664 016703 000036      MOV      SAVR3,%3
2216 011670 016704 000034      MOV      SAVR4,%4
2217 011674 004767 000504      JSR      %7,CKSWR
2218 011700 005777 167110      1$: TST      @SR      :CHECK SR FOR HALT SWITCH
2219 011704 100001      BPL      .+4
2220 011706 000000      HALT
2221 011710 023737 000042 000046      CMP      @#42,@#46      :HALT ON ERROR UP IN SWR
2222 011716 001001      BNE      .+4      :ARE WE IN ACT11 AUTO MODE?
2223 011720 000000      HALT      :BRANCH ON NO
2224 011722 000002      RTI      :HALT ON ERROR IF IN ACT11 AUTO MODE
2225 011724 000000      SAVR2: 0      :RETURN TO MAINLINE
```


2226	011726	000000		SAVR3:	0	
2227	011730	000000		SAVR4:	0	
2228	011732	000000		SAVPC:	0	
2229	011734	000000		SAVCC:	0	
2230						
2231	011736	005067	000260	PRTAB:	CLR	BINCT
2232	011742	005067	000252		CLR	WGTCT
2233	011746	012704	012226		MOV	#LIST,%4
2234	011752	142777	000177	167104	BICB	#177,@TPS
2235	011760	012767	000005	000236	MOV	#5,ASCNT
2236	011766	012767	000007	000220	MOV	#7,SEVEN
2237	011774	012767	000001	000214	MOV	#1,DECML
2238	012002	105777	167056	WAIT1:	TSTB	@TPS
2239	012006	100375			BPL	WAIT1
2240	012010	005702			TST	%2
2241	012012	100404			BMI	MINUS
2242	012014	012777	000260	167044	MOV	#260,@TPB
2243	012022	000403			BR	START
2244	012024	012777	000261	167034	MINUS:	MOV
2245	012032	016703	000156		START:	MOV
2246	012036	010267	000150			MOV
2247	012042	005167	000144			COM
2248	012046	046703	000140			BIC
2249	012052	001410				BEQ
2250	012054	066767	000136	000136	MKNUM:	ADD
2251	012062	005267	000134			INC
2252	012066	026703	000126			CMP
2253	012072	001370				BNE
2254	012074	062767	000260	000120	WRTOC:	ADD
2255	012102	016724	000114			MOV
2256	012106	066767	000102	000102		ADD
2257	012114	005067	000100			CLR
2258	012120	005067	000076			CLR
2259	012124	005367	000074			DEC
2260	012130	001410				BEQ
2261	012132	012703	000003			MOV
2262	012136	066767	000052	000050	MOADD:	ADD
2263	012144	005303				DEC
2264	012146	001373				BNE
2265	012150	000730				BR
2266	012152	012767	000005	000044	XLIST:	MOV
2267	012160	105777	166790		WAIT2:	TSTB
2268	012164	100375				BPL
2269	012166	014477	166674			MOV
2270	012172	005367	000026			DEC
2271	012176	001401				BEQ
2272	012200	000767				BR
2273	012202	105777	166656		HDFHM:	TSTB
2274	012206	100375				BPL
2275	012210	000207				RTS
2276	012212	000000			TOODLE:	0
2277	012214	000000			SEVEN:	0
2278	012216	000000			DECML:	0
2279	012220	000000			WGTCT:	0
2280	012222	000000			BINCT:	0
2281	012224	000000			ASCNT:	0

:GET LIST ADDRESS
 :CLR INT FLAG

 :NEG SIGN PRINT 1
 :POS SIGN PRINT 0

 :PUT MASK IN R3
 :GET READY TO DOODLE NUMBER IN TOODLE
 :COMPENSATES FOR COMPLEMENT DURING BIC
 :AND IN OCTAL CHARACTER
 :ZERO, WRITE 0 IN LIST
 :COUNT UP TO
 :AND RECORD
 :SAME BINARY WEIGHT
 :KEEP COUNTN
 :ADD ASCII PREFIX
 :WRITE ASCII CHAR IN LIST
 :EXPAND BINARY WEIGHT

 :5 CHAR IN LIST
 :SET X3 FOR ADD LOOP
 :MAKING SEVENTY BY SEVEN

 :NX SEVEN SET GET NX OCTAL
 :SEND 5 CHAR TO TTY

 :FINISH PRINTING GET NXT NUM

 :HEAD FOR HOME

2282 012226 000000
2283 012230 000000
2284 012232 000000
2285 012234 000000
2286 012236 000000

LIST: 0
0
0
0
0

::*****
: SCOPE LOOP ROUTINE ENTERED BY USER TRAP
:*****

2291
2292 012240 004767 000140
2293 012244 032777 040000 166542
2294 012252 001003
2295 012254 011667 000106
2296 012260 000002
2297 012262 022606
2298 012264 012677 166526
2299 012270 000177 000072

SCOPEA: JSR %7,CKSWR
BIT #40000,ASR
BNE SCOPEB ;SCOPE, BIT IS A ONE
MOV @%6,RETURN ;NO - SAVE PC FOR NEXT TIME
RTI ;RETURN IN SEQUENCE
SCOPEB: CMP (6)+,%6 ;REPOSITION THE STACK
MOV (6)+,@PSW
JMP @RETURN ;SCOPE RETURN

::*****
: SCOPE OR/AND ITERATION LOOP FOR EACH TEST 4000 TIMES
:*****

2306 012274 004767 000104
2307 012300 032777 040000 166506
2308 012306 001365
2309 012310 005767 166560
2310 012314 001415
2311 012316 004767 000062
2312 012322 032777 004000 166464
2313 012330 001007
2314 012332 026767 000026 000022
2315 012340 001403
2316 012342 005267 000016
2317 012346 000745
2318 012350 005067 000010
2319 012354 011667 000006
2320 012360 000002
2321 012362 004000
2322 012364 000000
2323 012366 001174

SCOPEC: JSR %7,CKSWR
BIT #40000,ASR ;TEST SR FOR SCOPE
BNE SCOPEB ;YES SCOPE
TST PASCNT ;FIRST PASS (PASCNT=0) ?
BEQ SCOPEG ;BR IF YES, INHIBIT ITERATIONS
JSR %7,CKSWR
BIT #4000,ASR ;TEST FOR ITERATION
BNE SCOPEG ;INHIBIT ITERATION
CMP SCOPEF,ICOUNT
BEQ SCOPEG ;EXIT - DONE
INC SCOPEF ;INCREMENT COUNT
BR SCOPEB ;LOOP SOME MORE
SCOPEG: CLR SCOPEF ;CLEAR COUNT
MOV @%6,RETURN ;SAVE SCOPE RETURN POINTER
RTI ;RETURN INLINE-NEXT TEST
ICOUNT: 4000
SCOPEF: 0 ;COUNT LOCATION FOR ITERATION LOOP
RETURN: BEGIN ;ADDRESS OF LAST TEST
.EVEN
JMP 200

::*****
: CHECK SWITCH REGISTER ROUTINE. CHECKS FOR ^G TO ALLOW CHANGING
: OF LOC. 176.
:*****

2332 012374 000000
2333 012376 000000
2334 012400 000000
2335 012402 000000

TEMPST: .WORD 0
COUNT: .WORD 0
RDSW: .WORD 0
TIB: .WORD 0

2336
2337 012404 022767 000176 166402

CKSWR: CMP #SWREG,SR ;SOFTWARE SWITCH REGISTER PRESENT

2338	012412	001133			BNE	OUT	
2339	012414	105777	166440		TSTB	@TKS	:YES, WAIT FOR
2340	012420	100130			BPL	OUT	:READY, GET CHARACTER
2341	012422	017767	166434	177752	MOV	@TKB, TIB	:AND STRIP OFF
2342	012430	042767	177600	177744	BIC	#177600, TIB	:THE GARBAGE
2343	012436	022767	000007	177736	CMP	#7, TIB	:IS IT A <^G>
2344	012444	001116			BNE	OUT	
2345	012446	012702	012756		MOV	#SCNTG, %2	
2346	012452	004767	000426		JSR	PC, TTOUT	
2347	012456	012702	012770		JSR	#SMSWR, %2	
2348	012462	004767	000416		JSR	PC, TTOUT	
2349	012466	017702	166322		MOV	@SR, %2	
2350	012472	004767	177240		JSR	%7, PRTAB	
2351	012476	012702	013000		MOV	#SMNEW, %2	
2352	012502	004767	000376		JSR	PC, TTOUT	
2353	012506	005037	012374		CLR	@TEMPST	
2354	012512	005067	177656		CLR	TEMPST	
2355	012516	012767	000007	177652	MOV	#7, COUNT	
2356	012524	004767	000154		JSR	PC, TTIN	:GO READ A CHARACTER
2357	012530	042767	177600	177644	BIC	#177600, TIB	:STRIP OFF GARBAGE
2358	012536	122767	000025	177636	CMPB	#25, TIB	:IS IT A ^U?
2359	012544	001001			BNE	2\$	:BRANCH IF NOT
2360	012546	000743			BR	CNTLU	:START OVER
2361	012550	122767	000015	177624	CMPB	#15, TIB	:IS IT A <CR>?
2362	012556	001011			BNE	4\$	:BRANCH IF NOT
2363	012560	012702	012764		MOV	#SCRLF, %2	
2364	012564	004767	000314		JSR	%7, TTOUT	
2365	012570	022767	000007	177600	CMP	#7, COUNT	:WAS IT FIRST CHARACTER
2366	012576	001036			BNE	7\$	:CHANGE SWR IF NOT FIRST ONE
2367	012600	000440			BR	OUT	:GET OUT
2368	012602	122767	000060	177572	CMPB	#60, TIB	
2369	012610	003004			BGT	5\$	
2370	012612	122767	000067	177562	CMPB	#67, TIB	
2371	012620	002005			BGE	6\$	
2372	012622	012702	013011		MOV	#SQUEST, %2	
2373	012626	004767	000252		JSR	PC, TTOUT	
2374	012632	000745			BR	3\$	:START OVER IF NOT LEGAL CHARACTER
2375	012634	006367	177534		ASL	TEMPST	
2376	012640	006367	177530		ASL	TEMPST	
2377	012644	006367	177524		ASL	TEMPST	
2378	012650	142767	000060	177524	BICB	#60, TIB	:GET NITTY-GRITTY
2379	012656	156767	177520	177510	BISB	TIB, TEMPST	
2380	012664	005367	177506		DEC	COUNT	:ONLY WANT 6 DIGITS
2381	012670	001754			BEQ	5\$	
2382	012672	000714			BR	1\$	
2383	012674	016777	177474	166112	MOV	TEMPST, @SR	:CHANGE SWITCH REGISTER CONTENTS
2384	012702	000207			OUT:	RTS	:RETURN TO PROGRAM
2385							::*****
2386							: TTY READ SUBROUTINE*****
2387							::*****
2388							
2389							
2390							
2391	012704	005077	166150		TTIN:	CLR	@TKS
2392	012710	005077	166146			CLR	@TKB
2393	012714	005067	177462			CLR	TIB

2394	012720	005277	166134			INC	@TKS	
2395	012724	105777	166130		TTIN1:	TSTB	@TKS	
2396	012730	100375				BPL	TTIN1	
2397	012732	017767	166124	177442		MOV	@TKB,TIB	
2398	012740	105777	166120		TTIN2:	TSTB	@TPS	
2399	012744	100375				BPL	TTIN2	
2400	012746	116777	177430	166112		MOVB	TIB,@TPB	
2401								
2402	012754	000207				RTS	%7	
2403	012756	057137	020107	000046	\$CNTG:	.ASCIZ	'^G &'	
2404	012764	020137	000046		\$CRLF:	.ASCIZ	'- &'	
2405	012770	051537	051127	020075	\$MSWR:	.ASCIZ	'-SWR= &'	
2406	012776	000046						
2407	013000	020040	042516	036527	\$MNEW:	.ASCIZ	' NEW= &'	
2408	013006	023040	000					
2409	013011	137	020077	023137	\$QUEST:	.ASCIZ	'_? _&'	
2410	013016	000						
2411	013017	137	047105	020104	\$ENPAS:	.ASCIZ	'_END PASS &'	
2412	013024	040520	051523	020040				
2413	013032	023040	000					
2414	013035	137	041440	042132	\$TITLE:	.ASCIZ	'_ CZDRBHO DR11-B/DA11-B NPR DIA _&'	
2415	013042	041122	030110	042040				
2416	013050	030522	026461	027502				
2417	013056	040504	030461	041055				
2418	013064	047040	051120	042040				
2419	013072	040511	020040	057440				
2420	013100	000046						
2421						.EVEN		
2422								
2423	013102	000000			OFL:	0		:FIRST CHAR FLAG
2424								
2425								
2426								
2427								
2428								
2429								
2430								
2431	013104	105712			TTOUT:	TSTB	(2)	:CHECK FOR NULL CHARACTER
2432	013106	001403				BEQ	1\$	:IF NOT, TYPE THE CHARACTER
2433	013110	122712	000046			CMPB	#'&,(2)	:CHECK FOR TERMINATOR
2434	013114	001005				BNE	.EMPTY	
2435	013116	042777	000100	165740	1\$:	BIC	#100,@TPS	
2436	013124	005002				CLR	%2	:CLEAR POINTER TO CHARACTER
2437	013126	000207				RTS	%7	:RETURN
2438	013130	122712	000137		.EMPTY:	CMPB	#'&,(2)	:CRLF CHAR?
2439	013134	001411				BEQ	.RET	
2440	013136	122712	000041			CMPB	#'!,(2)	:CHECK FOR RETURN TERMINATOR
2441	013142	001414				BEQ	.RET	
2442	013144	105777	165714		1\$:	TSTB	@TPS	
2443	013150	100375				BPL	1\$	
2444	013152	112277	165710			MOVB	(2)+,@TPB	:TYPE CHARACTER
2445	013156	000752				BR	TTOUT	
2446	013160	005202			.RET:	INC	%2	
2447	013162	010267	000020			MOV	%2, SAV	:SET UP NEW POINTER
2448	013166	012702	013202			MOV	#.RETR,%2	
2449	013172	000767				BR	.RET-6	

2450 013174 016702 000006
2451 013200 000741
2452
2453 013202 015 012 041
2454 013206 013206
2455 013206 000000
2456 014210 014210
2457 014210 000000
2458 014212 014212
2459 015214 015214
2460 015214 015214
2461 000001

.REST: MOV .SAV,%2
BR ITOUT
.RETR: .BYTE 15,12,'!
.EVEN
.SAV: 0
=+.1000
BUFF: 0
XINBUF: .
=+.1000
XCHKBU: .
.END

;FOR STACK POINTER 100 LOCATIONS

[illegible]

DATINB	006206	1455	1459#															
DATOCK	007306	1207	1245	1490	1629#													
DATONB	006346	1483	1489#															
DECML	012216	2237*	2250	2256*	2278#													
DIOMEM	001040	537#	1252	1411														
DONE	005710	1398	1403#															
DRBA	001022	530#	584*	617*	620	643*	644	859*	860	863*	864	887*	888	891*				
		913*	941*	971*	1005*	1040*	1103*	1108*	1124	1140*	1156	1179*	1200*	1218*				
		1238*	1252*	1280*	1302*	1376	1388*	1411*	1422*	1447*	1475*	1498*	1540*	1559*				
		1608	1961*	2040*														
DRDB	001026	532#	588*	1109*	1127	1142*	1180*	1186	1201*	1219*	1225	1239*	1313*	1316				
		1325*	1326	1334*	1335	1338*	1339	1349*	1350	1353*	1448*	1460	1476*	1541*				
		1560*	1955	2046														
DRINL	001032	534#	1144	1182	1203	1221	1241	1254	1282	1450	1478	1508	1543	1562				
		1836	1838	1840														
DRINS	010204	1835	1853#															
DRINV	001034	535#	909*	937*	967*	1001*	1011*	1018*	1036*	1046*	1052*	1111*	1143*	1181*				
		1202*	1220*	1240*	1253*	1281*	1412*	1421*	1449*	1477*	1506*	1542*	1561*	1625*				
		1834																
DRINO	010216	1854	1857#															
DRIN1	010240	1860	1863#															
DRIN2	010262	1866	1869#															
DRIN3	010250	1858	1865#															
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*	919*	922*	930	934	942*	946*	949*				
		958	962	972*	976*	982*	992	996	1006*	1010*	1016*	1029	1033	1041*				
		1045*	1050*	1060*	1061	1064*	1065	1068	1071*	1072*	1073	1076	1079*	1080*				
		1081	1084	1087*	1092	1094	1097	1099*	1100	1104	1114*	1133*	1146*	1166*				
		1184*	1205*	1223*	1243*	1256*	1259*	1260	1263	1266	1269*	1284*	1287*	1289				
		1292	1295*	1296	1299	1364	1367	1389	1392*	1393	1397	1415*	1418	1423*				
		1425*	1431	1452*	1480*	1501*	1511*	1515	1545*	1564*	1595*	1596	1599	1642*				
		1643	1646	1755*	1771*	1857	1865	1926*	1927*	1947	1952	1963*	1964	1966				
		1971*	1972*	1973*	2001	2005	2009*	2011*	2012*	2019	2023	2027*	2029	2031*				
		2032*	2043	2212														
DRVS	001030	533#	1112*	1144*	1182*	1203*	1221*	1241*	1254*	1282*	1413*	1450*	1478*	1508*				
		1543*	1562*	1625	1626*													
DRWC	001020	529#	582*	601*	604	632*	633	846*	847	850*	851	874*	875	878*				
		912*	940*	970*	1004*	1039*	1107*	1121	1139*	1153	1178*	1199*	1217*	1237*				
		1251*	1279*	1387*	1410*	1424*	1446*	1473*	1510*	1539*	1558*	1602	1962*	2041*				
		1729#																
DSTATB=	002000	1730#	1964	2029														
DSTATC=	004000	1731#	1857	1947	1954	2045												
END	007404	1574	1654#															
END1	007434	1662#																
ERRA	001252	580	590#															
ERRCHK	007354	1120	1152	1642#														
ERRDO	005764	1412	1418#															
ERRDO1	006046	1421	1431#															
ERROR =	100000	1735#																
FLUSH	011222	1911	2068#	2085														
FLUSH1	011230	2070#	2071															
FNCCNT	001070	549#	1502*	1518	1521*	1522												
FNCT1 =	000002	1721#																

FNCT2 = 000004	1722#	1971	2027											
FNCT3 = 000010	1723#	1953	2044											
GETBUF 011436	2147	2156#												
GO = 000001	1720#	1973	2012	2032										
HDFHM 012202	2271	2273#												
HLT = 104000	471#	590	606	622	635	646	659	663	676	680	693	697	706	
	710	719	723	732	736	745	749	758	762	772	775	778	781	
	784	787	790	793	797	800	803	806	809	813	816	819	833	
	839	849	853	862	866	877	890	908	923	936	950	964	979	
	985	998	1014	1021	1035	1048	1063	1067	1070	1075	1078	1083	1086	
	1096	1102	1106	1118	1123	1126	1129	1150	1155	1158	1162	1188	1227	
	1258	1262	1265	1268	1286	1291	1294	1298	1301	1318	1328	1337	1341	
	1352	1366	1369	1378	1391	1395	1402	1417	1420	1429	1433	1462	1520	
	1598	1601	1604	1610	1621	1635	1640	1645	1648	1786	1790	1794	1855	
	1861	1867	1871	1949	2003	2007	2021	2025	2053	2132	2137	2154		
	1957	1982#	1994											
HNDSHK 010734	1985#	1986												
HNDSH1 010742	1990	1992#	1993											
HNDSH2 010762	600*	631*	826*	845*	872*	900*	1278*	1312*	1324*	1347*	1375*	1386*	2314	
ICOUNT 012362	2321#													
	1726#	1755	1771	1926	1927	1973	2012	2032						
IE = 000100	538#	913	941	971	1005	1040	1179	1200	1218	1238	1388	1447	1475	
INBUF 001042	1498	1505	1524*	1527*	1540	1559	1575	1607	1616	1630	1960			
	550#	1505*												
INBUF1 001072	888#	893												
INCBA 002552	1350#	1355												
INCDB 005512	875#	880												
INCWC 002522	1181	1202	1220	1240	1449	1477	1506	1542	1561	1595#				
INTA 007130	1111	1120#												
INTB 004144	1143	1152#												
INTC 004316	916#													
ITIME1 002706	944#													
ITIME2 003034	974#													
ITIME3 003176	1008#													
ITIME4 003362	1043#													
ITIME5 003544	1743#	1748*	1892	1897*	1930*									
JBCNT 007626	1742#	1751*	1767*	1796	1801	1802*	1853	1863*	1873*					
JBFLAG 007624	1752	1881#	1928											
JOBA0 010304	1893	1898#												
JOBA0A 010364	1894#	1895												
JOBA0B 010344	1901	1910#												

CZDRBHO DR11-B/DA11-B NPR DIA
CZDRBH.P11 20-MAY-83 15:04

MACY11 30(1046) 20-MAY-83 15:09 PAGE 51
CROSS REFERENCE TABLE -- USER SYMBOLS

SEQ 0049

MOADD	012136	2262#	2264																	
MREC	011044	1913	2019#																	
MREC1	011056	2020	2023#																	
MREC2	011070	2024	2027#																	
MREC3	011102	2029#	2030																	
MSTART	001000	523#																		
MS1	007630	524	1747#																	
MXMIT	010770	1900	2001#																	
MXMIT1	011002	2002	2005#																	
MXMIT2	011014	2006	2009#																	
NEX =	040000	1734#																		
NEXCHK	005116	1253	1259#																	
NEXJOB	007620	1740#	1752*	1753*	1768*	1769*	1799*	1800*	1859	1863	1869	1873	1901*	1902*						
		1914*	1915*	1928*	1929*	1939*	1940*	1974*	1975*											
NOP =	000240	469#																		
NORMAL	007264	1130	1163	1271	1306	1436	1612	1625#												
NPRRDY	004052	1093	1107#																	
NPR1	001036	536#	1108	1110*	1124	1140	1141*	1156	1160											
NWRDXF	007616	1739#	1989	2059																
N1413	004036	1098	1103#																	
OFL	013102	2423#																		
OUT	012702	2338	2340	2344	2367	2384#														
PASCNT	001074	551#	1657*	2309																
PRINT	011506	495	1837	2187#																
PRMXFR	011130	2010	2028	2040#																
PRMXF1	011162	2047#	2060																	
PRMXF2	011166	2049#	2050																	
PRMXF3	011202	2052	2055#																	
PRMXF4	011204	2056#	2057																	
PRTAB	011736	2203	2208	2213	2231#	2350														
PSW	001016	508*	528#	570*	901*	929*	957*	991*	1028*	1113*	1145*	1183*	1204*	1222*						
		1242*	1255*	1283*	1414*	1451*	1479*	1509*	1544*	1563*	1627*	1662*	1681*	1692*						
		1695*	1754*	1756*	1770*	1772*	2298*													
P3INT	003572	1036	1050#																	
P3INV	003502	1030	1036#																	
P4INT	003414	1001	1016#																	
P4INV	003304	993	999#																	
P5ERR	003222	967	981#																	
P5INV	003120	959	965#																	
P6ERR	003046	937	948#																	
P6INV	002772	931	937#																	
P7ERR	002720	909	921#																	
P7INV	002644	903	909#																	
RDSW	012400	2334#																		
RDYCHK	001056	544#	1396*	1399*																
READY =	000200	1727#																		
RESALL	010106	1804	1822#																	
RETURN	012366	571*	2295*	2299	2319*	2323#														
SAVALL	010072	1798	1811#																	
SAVCC	011734	2191*	2207	2229#																
SAVPC	011732	2190*	2202	2228#																
SAVR2	011724	2199*	2214	2225#																
SAVR3	011726	2200*	2215	2226#																
SAVR4	011730	2201*	2216	2227#																
SCOPE =	104400	470#	579	592	607	616	630	642	655	672	689	702	715	728						
		741	754	767	825	844	858	871	885	899	928	956	990	1027						

MACY11 30(1046) 20-MAY-83 15:09 M 4 PAGE 53
CROSS REFERENCE TABLE -- USER SYMBOLS

[illegible]

1#	1#	461#	466	518	520	574	576	595	597	611	613	625	627	637	639
650	652	667	669	684	686	699	701	712	714	725	727	738	740	751	751
753	764	766	821	824	841	843	855	857	868	870	882	884	896	898	898
925	927	953	955	987	989	1024	1026	1056	1058	1088	1090	1135	1137	1169	1169
1171	1190	1192	1209	1211	1228	1230	1247	1249	1273	1276	1308	1310	1320	1322	1322
1330	1332	1343	1345	1357	1360	1371	1373	1382	1384	1404	1408	1438	1440	1464	1464
1466	1492	1494	1530	1532	1549	1551	1651	1653	1670	1672	1685	1687	1704	1706	1706
1744	1746	1760	1762	1774	1776	1807	1809	1818	1820	1830	1832	1848	1851	1876	1876
1880	1905	1908	1918	1921	1933	1937	1943	1945	1978	1980	1996	1999	2015	2018	2018
2035	2038	2063	2066	2080	2083	2095	2098	2116	2120	2142	2145	2181	2184	2287	2287
2289	2301	2303	2328	2331	2385	2387	2426	2428							

SWRSU	1#
TYPBIN	1#
TYPDEC	1#
TYPNAM	1#
TYPNUM	1#
TYPOCS	1#
TYPOCT	1#
TYPTXT	1#
\$\$ESCA	1#
\$\$NEWT	1#
\$\$\$SKIP	1#
.EQUAT	1#
.HEADE	1#
.KT11	1#
.SETUP	1#
.SWRHI	1#
.\$ACT1	1#
.\$APTB	1#
.\$APTH	1#
.\$APTY	1#
.\$ASTA	1#
.\$CATC	1#
.\$CMTA	1#
.\$DB2D	1#
.\$DB2O	1#
.\$DIV	1#
.\$EOP	1#
.\$ERRO	1#
.\$ERRT	1#
.\$MULT	1#
.\$POWE	1#

461#

.SRAND 1#
.SRDDE 1#
.SRDOC 1#
.SREAD 1#
.SR2AZ 1#
.\$SAVE 1#
.\$SB2D 1#
.\$SB2O 1#
.\$SCOP 1#
.\$SIZE 1#
.\$SUPR 1#
.\$TRAP 1#
.\$TYPB 1#
.\$TYPD 1#
.\$TYPE 1#
.\$TYPO 1#
.\$4OCA 1#
.1170 1#

. ABS. 015216 000

ERRORS DETECTED: 0

CZDRBH,CZDRBH/CRF/NL:TOC/SOL=SYSMAC.SML,CZDRBH.P11
RUN-TIME: 8 10 .7 SECONDS
RUN-TIME RATIO: 59/20=2.8
CORE USED: 35K (70 PAGES)