

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



Table with multiple columns and rows of data, including headers and numerical values. The text is small and difficult to read, but appears to be a structured data table.



.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 RELIABILTY 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 INTERUPT WITH THE INTERUPT ENABLED. THE FIRST INTERUPT THAT COMES SHOULD BE THE READY INTERUPT SET UP WHEN THE MASTER HIT THE START KEY. THE INTERUPT CAUSES THE SLAVE TO ENTER THE INTERUPT SERVICE ROUTINE.
- 14.2 THE INTERUPT SERVICE ROUTINE DETERMINS WHAT INTERUPT CAME UP AND IF IT SHOULD HAVE COME UP. IF THE INTERUPT WAS THE ONE EXPECTED THAN THE THAN THE PROGRAM GOES TO THE TO THE PROPER JOB ROUTINE, SJOBXX FOR SLAVE SERVICE AND JOBXX FOR THE MASTER ROUTINE.
- 14.3 THE NEXT THING THAT SHOULD HAPPEN IS THE MASTER SHOULD ISSUE AN INTERUPT TO THE SLAVE . THIS IS A SIGNAL FOR THE SLAVE TO 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
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489

000240
104400
104000
000001
000002
000004
000010
000020
000040
000100
000200
000400
001000
002000
004000
010000
020000
040000
100000

000004

```

:      MODIFIED FOR SOFTWARE SWITCH REGISTER
:      INCLUDING DYNAMIC LOADING OF SWR
:
:      MODIFIED TO ELIMINATE BAOF ERROR WHEN
:      DMA TRANSFERS CROSS A 32K BOUNDRY
:*****
:*****
:
NOP=000240
SCOPE=TRAP
HLT=EMT
BIT0=000001
BIT1=000002
BIT2=000004
BIT3=000010
BIT4=000020
BIT5=000040
BIT6=000100
BIT7=000200
BIT8=000400
BIT9=001000
BIT10=002000
BIT11=004000
BIT12=010000
BIT13=020000
BIT14=040000
BIT15=100000
  

BUSERR=000004
  
```



```

490
491
492
493      000000      . = 0
494      000030      . = 30
495 000030 011506    PRINT
496 000032 000340    340
497      000034      . = 34
498 000034 012274    SCOPEC
499 000036 000340    340
500      000046      . = 46
501 000046 007452    $ENDAD
502      000052      . = 52
503 000052 000000    0
504      000176      . = 176
505 000176 000000    SWREG: 0
506      000200      . = 200
507 000200 012706 014210  MOV #BUFF,%6 ;SET UP STACK LIMIT
508 000204 005077 000606  CLR @PSW
509 000210 023737 000042 000046  CMP @#42,@#46 ;ARE WE IN ACT11 AUTO MODE?
510 000216 001404      BEQ 1$ ;SKIP TITLE IF YES
511 000220 012702 013035  MOV # $TITLE,%2 ;PRINT THE TITLE
512
513 000224 004767 012654  JSR %7,TTOUT
514 000230 000167 000646  1$: JMP SUSWR
515      001000      . = 1000
516
517
518
519
520
521
522
523 001000 000000      MSTART: HALT ;MASTER START
524 001002 000167 006622  JMP MS1
525 001006 000000      SSTART: HALT ;SLAVE START
526 001010 000167 006674  JMP SS1
527 001014 177570      SR: 177570
528 001016 177776      PSW: 177776
529 001020 172410      DRWC: 172410
530 001022 172412      DRBA: 172412
531 001024 172414      DRST: 172414
532 001026 172416      DRDB: 172416
533 001030 000126      DRVS: 126
534 001032 000240      DRINL: 240
535 001034 000124      DRINV: 124
536 001036 052525      NPR1: 52525
537 001040 173000      DIOMEM: 173000
538 001042 014212      INBUF: XINBUF
539 001044 015214      CHKBUF: XCHKBU
540 001046 000000      BUFLN: HALT
541 001050 000000      LENCHK: HALT
542 001052 000000      BRWAIT: HALT
543 001054 000000      WCLN: HALT
544 001056 000000      RDYCHK: HALT
545 001060 177560      TKS: 177560

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

```

546	001062	177562			TKB:	177562		
547	001064	177564			TPS:	177564		
548	001066	177566			TPB:	177566		
549	001070	000000			FNCNT:	HALT		
550	001072	000000			INBUF1:	HALT		
551	001074	000000			PASCNT:	0		:NUMBER OF PASSES COMPLETED
552	001076	000000			TEMP:	0		:TEMPORARY STORAGE
553	001100	000000			TEMP2:	0		:TEMPORARY STORAGE
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
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599

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

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			

```
*****  
: TEST 1 DOES RESET CLEAR DRWC?  
:*****
```

600	001264	012767	000010	011070	MOV	#10,ICOUNT		
601	001272	012777	177777	177520	MOV	#+1,@DRWC		:ALL ONES TO DRWC

602	001300	004767	011100		JSR	%7,CKSWR	
603	001304	000005			RESET		:INIT
604	001306	005777	177506		TST	@DRWC	:LOOKING FOR Z-BIT TO SET
605	001312	001401			BEQ	+.4	:DID DRWC GET CLEARED?
606	001314	104000			HLT		:DRWC NOT CLEAR
607	001316	104400			SCOPE		

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

616	001320	104400			SCOPE		
617	001322	012777	177777	177472	MOV	#-1,@DRBA	:ALL ONES TO DRBA
618	001330	004767	011050		JSR	%7,CKSWR	
619	001334	000005			RESET		:INIT
620	001336	005777	177460		TST	@DRBA	:LOOKING FOR Z-BIT TO SET
621	001342	001401			BEQ	+.4	:DID DRBA GET CLEARED?
622	001344	104000			HLT		:DRBA NOT CLEAR

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

630	001346	104400			SCOPE		
631	001350	012767	004000	011004	MOV	#4000,ICOUNT	
632	001356	012777	177777	177434	MOV	#-1,@DRWC	:SET ALL BITS IN DRWC
633	001364	022777	177777	177426	CMP	#-1,@DRWC	:LOOKING FOR Z-BIT TO SET
634	001372	001401			BEQ	+.4	:SEE IF ALL BITS GOT SET
635	001374	104000			HLT		:ALL BITS AREN'T SET

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

642	001376	104400			SCOPE		
643	001400	012777	177776	177414	MOV	#-2,@DRBA	:SET BITS 15-01 IN DRBA
644	001406	022777	177776	177406	CMP	#-2,@DRBA	:LOOKING FOR Z-BIT TO SET
645	001414	001401			BEQ	+.4	:SEE IF BITS 15-01 GOT SET
646	001416	104000			HLT		:BITS 15-01 AREN'T SET

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

655	001420	104400			SCOPE		
656	001422	052777	000002	177374	BIS	#BIT1,@DRST	:SET FNCT1
657	001430	032777	000002	177366	BIT	#BIT1,@DRST	:TEST FNCT1

658	001436	001001			BNE	+.4	: IS IT SET?
659	001440	104000			HLT		: FNCT1 IS CLEAR
660	001442	042777	000002	177354	BIC	#BIT1,@DRST	: CLEAR FNCT1
661	001450	032777	000002	177346	BIT	#BIT1,@DRST	: TEST FNCT1
662	001456	001401			BEQ	+.4	: WAS IT CLEAR
663	001460	104000			HLT		: FNCT1 WAS SET

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

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

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

689	001524	104400			SCOPE		
690	001526	052777	000010	177270	BIS	#BIT3,@DRST	: SET FNCT3
691	001534	032777	000010	177262	BIT	#BIT3,@DRST	: TEST FNCT3
692	001542	001001			BNE	+.4	: IS IT SET?
693	001544	104000			HLT		: FNCT3 IS CLEAR
694	001546	042777	000010	177250	BIC	#BIT3,@DRST	: CLEAR FNCT3
695	001554	032777	000010	177242	BIT	#BIT3,@DRST	: TEST FNCT3
696	001562	001401			BEQ	+.4	: WAS IT CLEAR?
697	001564	104000			HLT		: FNCT3 WAS SET

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

702	001566	104400			SCOPE		
703	001570	052777	000020	177226	BIS	#BIT4,@DRST	: SET XBA16
704	001576	032777	000020	177220	BIT	#BIT4,@DRST	: TEST XBA16
705	001604	001001			BNE	+.4	: IS IT SET?
706	001606	104000			HLT		: XBA16 IS CLEAR
707	001610	042777	000020	177206	BIC	#BIT4,@DRST	: CLEAR XBA16
708	001616	032777	000020	177200	BIT	#BIT4,@DRST	: TEST XBA16
709	001624	001401			BEQ	+.4	: IS IT CLEAR
710	001626	104000			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,@DRST ;SET XBA17
:BIT5,@DRST ;TEST XBA17
:   +4 ;IS IT SET?
:HLT ;XBA17 IS CLEAR
:BIC4,@DRST ;CLEAR XBA17
:BIT4,@DRST ;TEST XBA17
:   +4 ;IS IT CLEAR?
:HLT ;XBA17 WAS SET
*****
:TEST13 TEST THAT IE CAN BE SET AND CLEARED
*****
:SCOPE
:BIT6,@DRST ;SET IE
:BIT6,@DRST ;TEST IE
:   +4 ;IS IT SET?
:HLT ;IE IS CLEAR
:BIC6,@DRST ;CLEAR IE
:BIT6,@DRST ;TEST IE
:   +4 ;IS IT CLEAR?
:HLT ;IE WAS SET
*****
:TEST14 TEST THAT CYCLE CAN BE SET AND CLEARED
*****
:SCOPE
:BIT8,@DRST ;SET CYCLE
:BIT8,@DRST ;TEST CYCLE
:   +4 ;IS IT SET?
:HLT ;CYCLE WAS CLEAR
:BIC8,@DRST ;CLEAR CYCLE
:BIT8,@DRST ;TEST CYCLE
:   +4 ;IS IT CLEAR?
:HLT ;CYCLE WAS SET
*****
:TEST15 TEST THAT MAINT CAN BE SET AND CLEARED
*****
:SCOPE
:BIT12,@DRST ;SET MAINT
:BIT12,@DRST ;TEST MAINT
:   +4 ;IS IT SET?
:HLT ;MAINT WAS CLEAR
:BIC12,@DRST ;CLEAR MAINT
:BIT12,@DRST ;TEST MAINT
:   +4 ;IS MAINT CLEAR?
:HLT ;MAINT WAS SET
*****
:TEST 16 TEST THAT ALL DRST R/W BITS CAN BE SET AND CLEARED
*****
:SCOPE
:BIS #10576,@DRST ;SET FOLLOWING: MAINT(12), CYCLE(08), IE(06), XBA17(05),
: ; XBA16(04),FNCT3(03),FUNCT2(02),FNCT1(01)
```

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,@DRST      ;SET FOLLOWING: MAINT(12),CYCLE(08),IE(06),XBA17(05),
828                                     ;XBA16(04),FNCT3(03),FNCT2(02),FNCT1(01)
829 002334 017701 176464      MOV      @DRST,%1         ;MOVE (DRST) TO R1
830 002340 052701 167201      BIS      #167201,%1       ;SETS BITS IN R1 THAT WERE NOT SET IN DRST
831 002344 005201                INC      %1               ;R1 SHOULD GO FROM -1 TO ZERO
832 002346 001401                BEQ      .+4              ;WERE ALL DRST R/W BITS SET?
833 002350 104000                HLT                                ;NOT ALL BITS WERE SET
834 002352 004767 010026      JSR      %7,CKSWR
835 002356 000005                RESET                       ;CLEAR ALL DRST R/W BITS
836 002360 017701 176440      MOV      @DRST,%1         ;MOVE (DRST) TO R1
837 002364 042701 127200      BIC      #127200,%1       ;CLEAR ALL BITS EXCEPT R/W BITS, NEX, AND GO
838 002370 001401                BEQ      .+4              ;SHOULD EQUAL ZERO
839 002372 104000                HLT                          ;RESET DIDN'T LEAVE DRST AS IT SHOULD HAVE
840
841                                     ;:*****
842                                     ;:TEST20 CAN DRWC HOLD ALTERNATE ONE'S AND ZERO'S
843                                     ;:*****
844 002374 104400                SCOPE
845 002376 012767 004000 007756      MOV      #4000,ICOUNT
846 002404 012777 052525 176406      MOV      #052525,@DRWC   ;ALT 0'S AND 1'S TO DRWC
847 002412 022777 052525 176400      CMP      #052525,@DRWC   ;LOOKING FOR Z-BIT TO SET
848 002420 001401                BEQ      .+4              ;DOES DRWC HAVE THE CORRECT PATTERN?
849 002422 104000                HLT                          ;DRWC DOESN'T HAVE THE CORRECT PATTERN
850 002424 012777 125252 176366      MOV      #125252,@DRWC   ;ALT 1'S AND 0'S TO DRWC
851 002432 022777 125252 176360      CMP      #125252,@DRWC   ;LOOKING FOR Z-BIT TO SET
852 002440 001401                BEQ      .+4              ;DOES DRWC HAVE THE CORRECT PATTERN?
853 002442 104000                HLT                          ;DRWC DOESN'T HAVE THE CORRECT PATTERN
854
855                                     ;:*****
856                                     ;:TEST21 CAN DRBA HOLD ALTERNATE ONE'S AND ZERO'S
857                                     ;:*****
858 002444 104400                SCOPE
859 002446 012777 052524 176346      MOV      #052524,@DRBA   ;ALT 0'S AND 1'S TO DRBA
860 002454 022777 052524 176340      CMP      #052524,@DRBA   ;LOOKING FOR Z-BIT TO SET
861 002462 001401                BEQ      .+4              ;DOES DRBA HAVE THE CORRECT PATTERN?
862 002464 104000                HLT                          ;DRBA DOESN'T HAVE THE CORRECT PATTERN
863 002466 012777 125252 176326      MOV      #125252,@DRBA   ;ALT 1'S AND 0'S TO DRBA
864 002474 022777 125252 176320      CMP      #125252,@DRBA   ;LOOKING FOR Z-BIT TO SET
865 002502 001401                BEQ      .+4              ;DOES DRBA HAVE THE CORRECT PATTERN?
866 002504 104000                HLT                          ;DRBA DOESN'T HAVE THE CORRECT PATTERN
867
868                                     ;:*****
869                                     ;:TEST22 INCREMENTING PATTERN TO WRAP-AROUND IN DRWC
870                                     ;:*****
871 002506 104400                SCOPE
872 002510 005067 007646      CLR      ICOUNT
873 002514 005001                CLR      %1               ;SET-UP
874 002516 005077 176276      CLR      @DRWC           ;SET-UP
875 002522 020177 176272      INCWC:  CMP      %1,@DRWC ;SEE IF THEY ARE EQUAL
876 002526 001401                BEQ      .+4              ;ARE THEY EQUAL?
877 002530 104000                HLT                          ;THEY'RE NOT EQUAL
878 002532 005277 176262      INC      @DRWC           ;GET NEXT NUMBER
879 002536 005201                INC      %1               ;GET NEXT NUMBER
880 002540 001370                BNE      INCWC           ;DONE WITH TEST? IF NOT CONTINUE
881

```

```
882
883
884
885 002542 104400
886 002544 005001
887 002546 005077 176250
888 002552 020177 176244
889 002556 001401
890 002560 104000
891 002562 062777 000002 176232
892 002570 062701 000002
893 002574 001366
894
895
896
897
898
899 002576 104400
900 002600 012767 004000 007554
901 002606 012777 000340 176202
902 002614 032777 000200 176202
903 002622 001010
904 002624 004767 007554
905 002630 000005
906 002632 032777 000200 176164
907 002640 001001
908 002642 104000
909 002644 012777 002720 176162 P7INV:
910 002652 012737 000340 000126
911
912 002660 012777 177777 176132 CHGG1:
913 002666 016777 176150 176126
914 002674 052777 000101 176122
915 002702 012704 000012
916 002706 077401
917
918
919 002710 042777 000100 176106
920 002716 000405
921 002720 022626
922 002722 042777 000100 176074 P7ERR:
923 002730 104000
924
925
926
927
928 002732 104400
929 002734 012777 000300 176054
930 002742 032777 000200 176054
931 002750 001010
932 002752 004767 007426
933 002756 000005
934 002760 032777 000200 176036
935 002766 001001
936 002770 104000
937 002772 012777 003046 176034 P6INV:
```

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

*****
TEST25 TEST THAT DR11-B DOES NOT INTERRUPT WITH PROC AT LEVEL 7
*****
SCOPE
MOV #4000,Icount
MOV #340,@PSW ;STATUS AT LEVEL 7
BIT #BIT7,@DRST ;CHECK READY BIT
BNE P7INV ;IS IT SET
JSR %7,CKSWR
RESET ;INIT TO SET READY
BIT #BIT7,@DRST ;SEE IF READY IS SET NOW
BNE .+4 ;IS READY SET?
HLT ;READY CAN'T BE SET BY INIT
P7INV: MOV #P7ERR,@DRINV ;SET UP INT VECTOR
MOV #340,@#126

*****
CHGG1: MOV #-1,@DRWC ;SET WORD COUNT TO -1
MOV INBUF,@DRBA ;SET BUSS ADDRESS LOC
BIS #101,@DRST ;SET I.E. AND GO BITS
MOV #10,%4 ;ALLOW 11/60 AND 11/70 TIME TO INTERRUPT
ITIME1: SOB %4,ITIME1

*****
BIC #BIT6,@DRST ;DR11-B DIDN'T INTERRUPT - CLEAR IE
BR X1
P7ERR: CMP (%6)+,(%6)+ ;RESTORE STACK
BIC #BIT6,@DRST ;CLEAR IE
HLT ;DR11-B INTERRUPTED, BUT IT SHOULDN'T HAVE

*****
TEST26 TEST THAT DR11-B DOES NOT INTERRUPT WITH PROC AT LEVEL 6
*****
X1: SCOPE
MOV #300,@PSW ;STATUS AT LEVEL 6
BIT #BIT7,@DRST ;CHECK READY BIT
BNE P6INV ;IS IT SET?
JSR %7,CKSWR
RESET ;INIT TO SET READY
BIT #BIT7,@DRST ;SEE IF READY IS SET NOW
BNE .+4 ;IS READY SET?
HLT ;READY CAN'T BE SET BY INIT
P6INV: MOV #P6ERR,@DRINV ;SET UP INT VECTOR
```



```

938 003000 012737 000340 000126          MOV      #340,@#126
939                                     :*****
940 003006 012777 177777 176004 CHGG2:  MOV      #-1,@DRWC      ;SET WORD COUNT TO -1
941 003014 016777 176022 176000          MOV      INBUF,@DRBA    ;SET BUSS ADDRESS LOC
942 003022 052777 000101 175774          BIS      #101,@DRST     ;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,@DRST    ;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,@DRST    ;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 000240 175726          MOV      #240,@PSW      ;STATUS AT LEVEL 5
958 003070 032777 000200 175726          BIT      #BIT7,@DRST    ;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,@DRST    ;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      @SR,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      #PSERR,@DRINV ;ISOLATE BR LEVEL
968 003142 012737 000340 000126          MOV      #340,@#126    ;SET UP INT VECTOR
969                                     :*****
970 003150 012777 177777 175642          CHGG3:  MOV      #-1,@DRWC      ;SET WORD COUNT TO -1
971 003156 016777 175660 175636          MOV      INBUF,@DRBA    ;SET BUSS ADDRESS LOC
972 003164 052777 000101 175632          BIS      #101,@DRST     ;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,@DRST    ;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          1$:     BR      X3
981 003222 022626          P5ERR:  CMP      (%6)+,(%6)+ ;RESTORE STACK
982 003224 042777 000100 175572          BIC      #BIT6,@DRST    ;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,@PSW      ;STATUS AT LEVEL 4
992 003254 032777 000200 175542          BIT      #BIT7,@DRST    ;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,@DRST ;INIT TO SET READY
997 003300 001001 BNE .+4 ;SEE IF READY IS SET NOW
998 003302 104000 HLT ;IS IT SET
999 003304 017767 175504 175564 P4INV: MOV @SR,TEMP ;READY CAN'T BE SET BY INIT
1000 003312 042767 177770 175556 BIC #177770,TEMP ;GET SWITCH SETTINGS
1001 003320 012777 003414 175506 MOV #P4INT,@DRINV ;ISOLATE BR LEVEL
1002 003326 012737 000340 000126 MOV #340,@#126 ;SET UP INT VECTOR
1003 *****
1004 003334 012777 177777 175456 CHGG4: MOV #-1,@DRWC ;SET WORD COUNT TO -1
1005 003342 016777 175474 175452 MOV INBUF,@DRBA ;SET BUSS ADDRESS LOC
1006 003350 052777 000101 175446 BIS #101,@DRST ;SET I.E. AND GO BITS
1007 003356 012704 000012 MOV #10,%4 ;ALLOW 11/60 AND 11/70 TIME TO INTERRUPT
1008 003362 077401 ITIME4: SOB %4,ITIME4
1009 *****
1010 003364 005077 175434 CLR @DRST ;CLEAR IE
1011 003370 012777 000126 175436 MOV #126,@DRINV ;RESTORE INTERRUPT VECTOR
1012 003376 022767 000004 175472 CMP #4,TEMP ;BR LEVEL = 4 ?
1013 003404 001401 BEQ 1$ ;YES, EVERYTHING IS OK
1014 003406 104000 HLT ;DR11-B DIDN'T INTERRUPT
1015 003410 000167 000026 1$: JMP X4
1016 003414 005077 175404 P4INT: CLR @DRST ;CLEAR IE
1017 003420 022626 CMP (%6)+,(%6)+ ;REPOSITION THE STACK AFTER AN INTERRUPT
1018 003422 012777 000126 175404 MOV #126,@DRINV ;RESTORE INTERRUPT VECTOR
1019 003430 022767 000004 175440 CMP #4,TEMP ;BR LEVEL = 4 ?
1020 003436 001001 BNE X4 ;NO, EVERYTHING IS OK
1021 003440 104000 HLT ;DR11-B WITH BR = 4 SHOULD NOT HAVE INTERRUPTED
1022
1023
1024
1025 *****
1026 *****
1027 003442 104400 X4: SCOPE
1028 003444 012777 000140 175344 MOV #140,@PSW ;STATUS AT LEVEL 3
1029 003452 032777 000200 175344 BIT #BIT7,@DRST ;CHECK READY BIT
1030 003460 001010 BNE P3INV ;IS IT SET?
1031 003462 004767 006716 JSR %7,CKSWR
1032 003466 000005 RESET ;INIT TO SET READY
1033 003470 032777 000200 175326 BIT #BIT7,@DRST ;SEE IF READY IS SET NOW
1034 003476 001001 BNE .+4 ;IS IT SET
1035 003500 104000 HLT ;READY CAN'T BE SET BY INIT
1036 003502 012777 003572 175324 P3INV: MOV #P3INT,@DRINV ;SET UP INT VECTOR
1037 003510 012737 000340 000126 MOV #340,@#126
1038 *****
1039 003516 012777 177777 175274 CHGG5: MOV #-1,@DRWC ;SET WORD COUNT TO -1
1040 003524 016777 175312 175270 MOV INBUF,@DRBA ;SET BUSS ADDRESS LOC
1041 003532 052777 000101 175264 BIS #101,@DRST ;SET I.E. AND GO BITS
1042 003540 012704 000012 MOV #10,%4 ;ALLOW 11/60 AND 11/70 TIME TO INTERRUPT
1043 003544 077401 ITIME5: SOB %4,ITIME5
1044 *****
1045 003546 005077 175252 CLR @DRST ;CLEAR IE
1046 003552 012777 000126 175254 MOV #126,@DRINV ;RESTORE INTERRUPT VECTOR
1047 003560 005037 000126 CLR @#126 ;RESTORE TRAP CATCHER
1048 003564 104000 HLT ;DR11-B DIDN'T INTERRUPT
1049 003566 000167 000020 JMP X5

```



```

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
1054
1055
1056
1057
1058
1059 003612 104400
1060 003614 005077 175204
1061 003620 032777 000016 175176
1062 003626 001401
1063 003630 104000
1064 003632 052777 000002 175164
1065 003640 032777 001000 175156
1066 003646 001001
1067 003650 104000
1068 003652 032777 006000 175144
1069 003660 001401
1070 003662 104000
1071 003664 005077 175134
1072 003670 052777 000004 175126
1073 003676 032777 002000 175120
1074 003704 001001
1075 003706 104000
1076 003710 032777 005000 175106
1077 003716 001401
1078 003720 104000
1079 003722 005077 175076
1080 003726 052777 000010 175070
1081 003734 032777 004000 175062
1082 003742 001001
1083 003744 104000
1084 003746 032777 003000 175050
1085 003754 001401
1086 003756 104000
1087 003760 005077 175040
1088
1089
1090
1091 003764 104400
1092 003766 005777 175032
1093 003772 100027
1094 003774 032777 020000 175022
1095 004002 001401
1096 004004 104000
1097 004006 032777 040000 175010
1098 004014 001410
1099 004016 042777 040000 175000
1100 004024 032777 040000 174772
1101 004032 001401
1102 004034 104000
1103 004036 005077 174760
1104 004042 005777 174756
1105 004046 001401

*****
: TEST31 TEST THAT FNCT BITS CONTROL DSTAT BITS (M968 MUST BE USED IN USER SLOTS)
*****
X5: SCOPE
CLR @DRST ;CLEAR FUNCTION BITS
BIT #16,@DRST ;CHECK FUNCTION BITS
BEQ .+4 ;FUNCTION BITS CLEAR?
HLT ;FUNCTION BITS NOT CLEAR
BIS #BIT1,@DRST ;SET FNCT1
BIT #BIT9,@DRST ;CHECK DSTAT C
BNE .+4 ;IS IT SET?
HLT ;DSTAT C IS CLEAR
BIT #6000,@DRST ;CHECK THAT DSTAT A AND DSTAT B ARE CLEAR
BEQ .+4 ;ARE THEY CLEAR?
HLT ;DSTAT A AND/OR DSTAT B IS SET
CLR @DRST ;CLEAR DRST
BIS #BIT2,@DRST ;SET FNCT2
BIT #BIT10,@DRST ;CHECK DSTAT B
BNE .+4 ;IS IT SET?
HLT ;DSTAT B IS CLEAR
BIT #5000,@DRST ;CHECK THAT DSTAT A AND DSTAT C ARE CLEAR
BEQ .+4 ;ARE THEY CLEAR?
HLT ;DSTAT A AND/OR DSTAT B IS SET
CLR @DRST ;CLEAR DRST
BIS #BIT3,@DRST ;SET FNCT3
BIT #BIT11,@DRST ;CHECK DSTAT A
BNE .+4 ;IS IT SET?
HLT ;DSTAT A IS CLEAR
BIT #3000,@DRST ;CHECK THAT DSTAT B AND DSTAT C ARE CLEAR
BEQ .+4 ;ARE THEY CLEAR?
HLT ;DSTAT B AND/OR DSTAT C IS SET
CLR @DRST ;CLR DRST
*****
: TEST 33 TEST FOR 1 DATI NPR TRANSFER (WITH M968 IN USER SLOTS)
*****
TNPR1: SCOPE
TST @DRST ;CHECK ERROR BIT
BPL NPRRDY ;IS IT CLEAR?
BIT #BIT13,@DRST ;CHECK ATTN
BEQ .+4 ;IS ATTN CLEAR
HLT ;ATTN IS SET
BIT #BIT14,@DRST ;CHECK NEX
BEQ N1413 ;IS NEX CLEAR?
BIC #BIT14,@DRST ;TRY TO CLEAR NEX
BIT #BIT14,@DRST ;CHECK AGAIN
BEQ .+4 ;NEX STILL SET
HLT ;NEX CAN'T BE CLEARED BY MOVING A 0 TO IT
N1413: CLR @DRBA ;TRY TO CLEAR BAOF
TST @DRST ;CHECK ERROR BIT AGAIN
BEQ .+4 ;IS IT CLEAR

```

```
1106 004050 104000 HLT ;ERROR CAUSED BY SOMETHING OTHER THAN NEX,ATTN, OR BAOF
1107 004052 012777 177777 174740 NPPRDY: MOV #-1,@DRWC ;SET UP FOR 1 TRANSFER
1108 004060 012777 001036 174734 MOV #NPR1,@DRBA ;TRANSFER FROM BUS ADDRESS IN NPR1
1109 004066 005077 174734 CLR @DRDB ;GET READY TO RECEIVE DATA
1110 004072 012767 052525 174736 MOV #52525,NPR1 ;SET UP TRANSFER DATA
1111 004100 012777 004144 174726 MOV #INTB,@DRINV ;INTERRUPT VECTOR TO INTB
1112 004106 012777 000005 174714 MOV #5,@DRVS ;INTERRUPT PRIORITY TO LEVEL 5
1113 004114 005077 174676 CLR @PSW ;LET THE DR11-B INTERRUPT
1114 004120 012777 000101 174676 MOV #101,@DRST ;IE AND DO TO DRST
1115 004126 005067 000002 CLR 1$+2 ;WAIT FOR NPR AND INTERRUPT
1116 004132 005227 000001 1$: 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 @DRWC ;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,@DRBA ;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,@DRDB ;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 @DRST ;CLEAR IE
1134 004220 000662 BR TNPR1 ;TRY TEST AGAIN
1135
1136 :*****
1137 : TEST 34 TEST FOR 1 DATO NPR TRANSFER (WITH M968 IN USER SLOTS)
1138 :*****
1138 004222 104400 TNPRO: SCOPE
1139 004224 012777 177777 174566 MOV #-1,@DRWC ;SET UP FOR 1 TRANSFER
1140 004232 012777 001036 174562 MOV #NPR1,@DRBA ;TRANSFER TO BUS ADDRESS IN NPR1
1141 004240 005067 174572 CLR NPR1 ;GET READY TO RECEIVE DATA
1142 004244 012777 052525 174554 MOV #52525,@DRDB ;SET UP TO TRANSFER DATA
1143 004252 012777 004316 174554 MOV #INTC,@DRINV ;INTERRUPT VECTOR TO INTC
1144 004260 016777 174546 174542 MOV DRINL,@DRVS ;INTERRUPT STATUS TO LEVEL DRINL
1145 004266 005077 174524 CLR @PSW ;PROC STATUS TO ZERO
1146 004272 012777 000103 174524 MOV #103,@DRST ;IE, FNCT1(C1 CONTROL), AND DO TO DRST
1147 004300 005067 000002 CLR 1$+2 ;WAIT FOR NPR AND INTER
1148 004304 005227 000001 1$: 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 @DRWC ;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,@DRBA ;COMPARE CORRECT DRBA WITH DRBA
1157 004340 001401 BEQ .+4 ;IS THE DRBA CORRECT?
1158 004342 104000 HLT ;DRBA IS WRONG
1159
1160 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          ::*****
1171          ::TEST 35 STRING OF 10 DATI'S (WITH M968 IN USER SLOTS)
1172          ::*****
1172 004374 104400          T35: SCOPE
1173 004376 012767 000020 174442  MOV                #20,BUFLEN ;LENGTH OF BUFFER=20
1174 004404 004767 002416          JSR                %7,LODBUF  ;LOAD THE BUFFER WITH INCREMENTING PATTERN
1175 004410 006267 174432          ASR                BUFLEN    ;BUFLEN=10
1176 004414 016767 174426 174432  MOV                BUFLEN,WLEN ;PREPARE NUMBER FOR DRWC
1177 004422 005467 174426          NEG                WLEN      ;2'S COMPLEMENT OF BUFLEN
1178 004426 016777 174422 174364  MOV                WLEN,@DRWC ;SET UP DRWC
1179 004434 016777 174402 174360  MOV                INBUF,@DRBA ;SET UP DRBA
1180 004442 012777 177777 174356  MOV                #-1,@DRDB  ;MAINT AIDE
1181 004450 012777 007130 174356  MOV                #INTA,@DRINV ;INT VECTOR TO INTA
1182 004456 016777 174350 174344  MOV                DRINL,@DRVS ;INT VECTOR TO PRIORITY DRINL
1183 004464 005077 174326          CLR                @PSW      ;LET THE DR11-B INTERRUPT
1184 004470 012777 000101 174326  MOV                #101,@DRST ;IE AND DO TO DRST
1185 004476 000777          BR                ;WAIT FOR INTERRUPT
1186 004500 022777 000007 174320  CMP                #7,@DRDB   ;CHECK THAT WORD #10 OF INBUF IS IN DRBA
1187 004506 001401          BEQ                .+4       ;IS IT?
1188 004510 104000          HLT                ;BAD DATA IN DRDB
1189
1190          ::*****
1191          ::TEST 36 STRING OF 10 DATO'S (WITH M968 IN USER SLOTS)
1192          ::*****
1193 004512 104400          SCOPE
1194 004514 012767 000020 174324  MOV                #20,BUFLEN ;LENGTH OF BUFFER=20
1195 004522 004767 002300          JSR                %7,LODBUF  ;LOAD THE BUFFER WITH INCREMENTING PATTERN
1196 004526 006267 174314          ASR                BUFLEN    ;BUFLEN=10
1197 004532 016767 174310 174314  MOV                BUFLEN,WLEN ;PREPARE NUMBER FOR DRWC
1198 004540 005467 174310          NEG                WLEN      ;2'S COMPLEMENT OF BUFLEN
1199 004544 016777 174304 174246  MOV                WLEN,@DRWC ;SET UP DRWC
1200 004552 016777 174264 174242  MOV                INBUF,@DRBA ;SET UP DRBA
1201 004560 012777 052525 174240  MOV                #52525,@DRDB ;SET UP DRDB
1202 004566 012777 007130 174240  MOV                #INTA,@DRINV ;INTERRUPT VECTOR TO INTA
1203 004574 016777 174232 174226  MOV                DRINL,@DRVS ;INTERRUPT VECTOR TO PRIORITY DRINL
1204 004602 005077 174210          CLR                @PSW      ;LET THE DR11-B INTERRUPT
1205 004606 012777 000103 174210  MOV                #103,@DRST ;IE, FNCT1(C1 CONTROL), AND DO TO DRST
1206 004614 000777          BR                ;WAIT FOR INTERRUPT
1207 004616 004767 002464          JSR                %7,DATOCK ;CHECK INBUF
1208
1209          ::*****
1210          ::TEST 37 STRING OF 200 DATI'S
1211          ::*****
1212 004622 104400          SCOPE
1213 004624 012767 000200 174214  MOV                #200,BUFLEN ;LENGTH OF BUFFER=200
1214 004632 004767 002170          JSR                %7,LODBUF  ;LOAD THE BUFFER WITH INCREMENTING PATTERN
1215 004636 016767 174204 174210  MOV                BUFLEN,WLEN ;PREPARE NUMBER FOR DRWC
1216 004644 005467 174204          NEG                WLEN      ;2'S COMPLEMENT OF BUFLEN
1217 004650 016777 174200 174142  MOV                WLEN,@DRWC ;SET UP DRWC

```

```
1218 004656 016777 174160 174136      MOV      INBUF,@DRBA      ;SET UP DRBA
1219 004664 012777 177777 174134      MOV      #-1,@DRDB      ;MAINT AIDE
1220 004672 012777 007130 174134      MOV      #INTA,@DRINV    ;INT VECTOR TO INTA
1221 004700 016777 174126 174122      MOV      DRINL,@DRVS     ;INT VECTOR TO PRIORITY DRINL
1222 004706 005077 174104                CLR      @PSW            ;LET THE DR11-B INTERRUPT
1223 004712 012777 000101 174104      MOV      #101,@DRST     ;IE AND DO TO DRST
1224 004720 000777                BR                      ;WAIT FOR INTERRUPT
1225 004722 022777 000177 174076      CMP      #177,@DRDB     ;CHECK THAT WORD #200 OF INBUF IS IN DRBA
1226 004730 001401                BEQ      .+4            ;IS IT?
1227 004732 104000                HLT                      ;BAD DATA IN DRDB
```

```
::*****
:
: TEST 40 STRING OR 200 DATO'S
:
:*****
```

```
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,WCLLEN  ;PREPARE NUMBER FOR DRWC
1236 004762 005467 174066                NEG      WCLLEN        ;2'S COMPLEMENT OF BUFLEN
1237 004766 016777 174062 174024      MOV      WCLLEN,@DRWC   ;SET UP DRWC
1238 004774 016777 174042 174020      MOV      INBUF,@DRBA   ;SET UP DRBA
1239 005002 012777 052525 174016      MOV      #52525,@DRDB  ;SET UP DRDB
1240 005010 012777 007130 174016      MOV      #INTA,@DRINV  ;INTERRUPT VECTOR TO INTA
1241 005016 016777 174010 174004      MOV      DRINL,@DRVS   ;INTERRUPT VECTOR TO PRIORITY DRINL
1242 005024 005077 173766                CLR      @PSW          ;LET THE DR11-B INTERRUPT
1243 005030 012777 000103 173766      MOV      #103,@DRST    ;IE, FNCT1, AND DO TO DRST
1244 005036 000777                BR                      ;WAIT FOR INTERRUPT
1245 005040 004767 002242                JSR      %7,DATOCK     ;CHECK INBUF
```

```
::*****
:
: TEST 42 TEST THAT DOING A DATO TO THE DIODE MEMORY CAUSES NEX
:
:*****
```

```
1250 005044 104400                SCOPE
1251 005046 012777 177776 173744      MOV      #-2,@DRWC     ;SET UP DRWC
1252 005054 016777 173760 173740      MOV      DIOMEM,@DRBA  ;SET UP DRBA
1253 005062 012777 005116 173744      MOV      #NEXCHK,@DRINV;INTERRUPT VECTOR TO NEXCHK
1254 005070 016777 173736 173732      MOV      DRINL,@DRVS   ;INTERRUPT STATUS TO LEVEL DRINL
1255 005076 005077 173714                CLR      @PSW          ;LET THE DR11-B INTERRUPT
1256 005102 012777 000163 173714      MOV      #163,@DRST    ;IE, FNCT1, XBA17, XBA16, AND GO TO DRST
1257 005110 005237 177560                INC      @#177560      ;WAIT FOR INTERRUPT
1258 005114 104000                HLT                      ;NO DR11-B INTERRUPT
1259 005116 042777 000100 173700      NEXCHK: BIC      #BIT6,@DRST ;CLEAR INTERRUPT ENABLE
1260 005124 005777 173674                TST      @DRST        ;TEST DRST
1261 005130 001001                BNE      .+4          ;ERROR SET?
1262 005132 104000                HLT                      ;ERROR NOT SET
1263 005134 105777 173664                TSTB    @DRST        ;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,@DRST  ;CHECK NEX
1267 005152 001001                BNE      .+4          ;NEX SET?
1268 005154 104000                HLT                      ;NEX IS CLEAR
1269 005156 042777 040000 173640      BIC      #BIT14,@DRST  ;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 @PSW ;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 @#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,@DRWC ;SET-UP DRWC
1388 005626 016777 173210 173166 MOV INBUF,@DRBA ;SET-UP DRBA
1389 005634 105777 173164 ;STB @DRST ;CHECK READY
1390 005640 100401 BMI .+4 ;IS READY SET?
1391 005642 104000 HLT ;READY IS CLEAR
1392 005644 012777 000011 173152 MOV #11,@DRST ;FNCT3 (NON-BURST) AND GO TO DRST
1393 005652 105777 173146 TSTB @DRST ;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 @DRST ;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 ;TEST 55 TEST THAT GIVING A DO WITHOUT CLEARING A PREVIOUS ERROR
1406 ;CAUSES ANOTHER INTERRUPT
1407 ;FOR THIS TEST, THE NORMAL INTERRUPT MECHANISM SHOULD BE WORKING
1408 *****
1409 SCOPE
1410 005712 104400 MOV #-20,@DRWC ;SET-UP DRWC
1411 005714 012777 177760 173076 MOV DIOMEM,@DRBA ;SET-UP DRBA
1412 005722 016777 173112 173072 MOV #ERRDO,@DRINV ;INTERRUPT VECTOR TO ERRDO
1413 005730 012777 005764 173076 MOV #140,@DRVS ;INTERRUPT STATUS TO LEVEL 4
1414 005744 005077 173046 CLR @PSW ;LET THE DR11-B INTERRUPT
1415 005750 012777 000163 173046 MOV #163,@DRST ;IE, XBA17,XBA16, FNCT1 AND GO TO DRST
1416 005756 005277 173076 INC @TKS ;WAIT FOR INTERUPT
1417 005762 104000 HLT ;NO DR11-B INTERRUPT
1418 005764 005777 173034 ERRDO: TST @DRST ;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,@DRINV ;INTERRUPT VECTOR TO ERRDO1
1422 006002 005077 173014 CLR @DRBA ;PREVENT CAUSING ANOTHER ERROR
1423 006006 042777 000062 173010 BIC #62,@DRST ;CLEAR XBA17, XBA16, AND FNCT1
1424 006014 012777 177777 172776 MOV #-1,@DRWC ;SET-UP DRWC
1425 006022 005277 172776 INC @DRST ;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 ;NO DR11-B INTERRUPT
1430 006042 162706 000004 SUB #4,%6 ;FAKE AN INTERRUPT WITH STACK
1431 006046 005777 172752 ERRDO1: TST @DRST ;CHECK ERROR
1432 006052 100401 BMI .+4 ;ERROR SET?
1433 006054 104000 HLT ;ERROR IS CLEAR - SHOULD BE SET BECAUSE
1434 ;PREVIOUS ERROR WAS NOT CLEARED
1435 006056 062706 000010 ADD #10,%6 ;REPOSITION THE STACK
1436 006062 004767 001176 JSR %7,NORMAL
1437 *****
1438 ;TEST 56 STRING OF 200 DATI'S NON-BURST MODE
1439 *****
1440 *****
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,@DRWC    ;SET-UP DRWC
1447 006122 016777 172714 172672      MOV      INBUF,@DRBA     ;SET-UP DRBA
1448 006130 012777 177777 172670      MOV      #-1,@DRDB      ;MAINT AIDE
1449 006136 012777 007130 172670      MOV      #INTA,@DRINV    ;INT VECTOR TO INTA
1450 006144 016777 172662 172656      MOV      DRINL,@DRVS     ;INT VECTOR TO PRIORITY DRINL
1451 006152 005077 172640              CLR      @PSW            ;LET THE DR11-B INTERRUPT
1452 006156 012777 000111 172640      MOV      #111,@DRST      ;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,@DRDB      ;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      SCOPE
1468 006222 104400              MOV      #201,BUFLEN     ;LENGTH OF BUFFER=201
1469 006224 012767 000201 172614      JSR      %7,LODBUF       ;LOAD THE BUFFER WITH INCREMENTING PATTERN
1470 006232 004767 000570              DEC      BUFLEN          ;BUFLEN=200
1471 006236 005367 172604              MOV      BUFLEN,WCLLEN   ;PREPARE NUMBER FOR DRWC
1472 006242 016767 172600 172604      NEG      WCLLEN          ;2'S COMPLEMENT OF BUFLEN
1473 006254 016777 172574 172536      MOV      WCLLEN,@DRWC    ;SET UP DRWC
1474
1475 006262 016777 172554 172532      MOV      INBUF,@DRBA     ;SET UP DRBA
1476 006270 012777 052525 172530      MOV      #52525,@DRDB    ;SET UP DRDB
1477 006276 012777 007130 172530      MOV      #INTA,@DRINV    ;INTERRUPT VECTOR TO INTA
1478 006304 016777 172522 172516      MOV      DRINL,@DRVS     ;INTERRUPT VECTOR TO PRIORITY DRINL
1479 006312 005077 172500              CLR      @PSW            ;LET THE DR11-B INTERRUPT
1480 006316 012777 000113 172500      MOV      #113,@DRST      ;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          ;WAIT FOR INTERRUPT
1484 006340 000001              WAIT
1485
1486 006342 000240              NOP
1487
1488 006344 000401              DATONB: BR      .+4
1489 006346 000777              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,@DRBA      ;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,FNCNT       ;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      MOVB    @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      CMPB    FNCNT,%1       ;CHECK AGAINST FNCNT
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     FNCNT          ;GET READY FOR NEXT PASS
1522 006516 022767 000010 172344      CMP     #10,FNCNT     ;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      SCOPE
1534 006556 104400              MOV      #10,BUFLEN    ;BUFLEN=10
1535 006560 012767 000010 172260      MOV      BUFLEN,WLEN   ;PREPARE NUMBER FOR DRWC
1536 006566 016767 172254 172260      NEG     WLEN           ;2'S COMPLEMENT OF BUFLEN
1537 006574 005467 172254              JSR     %7,LODBUF      ;LOAD IN BUFFER WITH INCREMENTING PATTERN
1538 006600 004767 000222              JSR     %7,CHKBFF      ;LOAD CHECK BUFFER WITH MODIFIED INCREMENTING PATTERN
1539 006604 004767 000254              MOV      WLEN,@DRWC    ;SET UP DRWC
1540 006610 016777 172240 172202      MOV      INBUF,@DRBA   ;SET UP DRBA
1541 006616 016777 172220 172176      MOV      #-1,@DRDB     ;MAINT AIDE
1542 006624 012777 177777 172174      MOV      #INTA,@DRINV  ;INTERRUPT VECTOR TO INTA
1543 006632 012777 007130 172174      MOV      DRINL,@DRVS   ;INTERRUPT STATUS AT PRIORITY DRINL
1544 006640 016777 172166 172162      CLR     @PSW           ;LET DR11-B INTERRUPT
1545 006646 005077 172144              MOV      #10101,@DRST ;MAINT, IE, AND DO TO DRST
1546 006652 012777 010101 172144      BR      ;WAIT FOR INTERRUPT
1547 006660 000777              JSR     %7,DATCHK
1548
1549      ;:*****
1550      ;: TEST 62 TEST FOR 200 NPR TRANSFERS IN MAINT MODE
1551      ;:*****
1552      SCOPE
1553 006666 104400              MOV      #200,BUFLEN   ;LENGTH OF BUFFER = 200
1554 006670 012767 000200 172150

```

1554	006676	016767	172144	172150	MOV	BUFLEN,WCLEN	:PREPARE NUMBER FOR DRWC
1555	006704	005467	172144		NEG	WCLEN	:2'S COMPLEMENT OF BUFLEN
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	WCLEN,@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 INBUFF AND INC BY 2
1578	007040	005267	172004		LOADA: INC	LENCHK	:INC LENGTH CHECK BY 1
1579	007044	026767	172000	171774	CMP	LENCHK,BUFLEN	: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,BUFLEN	: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	BUFLEN,%2	:BUFFER LENGTH TO R2
1606	007172	066702	171650		ADD	BUFLEN,%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		DATACHK: MOV	CHKBUF,%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		DATAOK: 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								
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	#\$ENPAS,%2		:PRINT 'END PASS'
1659	007430	004767	003450		JSR	%7,TTOUT		
1660								
1661								
1662	007434	042777	000020	171354	END1: BIC	#20,@PSW		:CLEAR T-BIT
1663	007442	013702	000042		MOV	@#42,%2		
1664	007446	001405			BEQ	TRTRAP		
1665	007450	000005			RESET			

1666 007452 004712
1667 007454 000240
1668 007456 000240
1669 007460 000240

SENDAD: JSR %7,(2)
NOP
NOP
NOP

: ROUTINE TO CHECK FOR TRACE TRAP TO BE RUN WITH PROGRAM
:*****

1674 007462 004767 002716
1675 007466 032777 010000 171320
1676 007474 001417
1677 007476 005767 000104
1678 007502 001411
1679 007504 016767 000076 170302
1680 007512 016767 000072 170276
1681 007520 042777 000020 171270
1682 007526 000167 171442
1683 007532 000000

TRTRAP: JSR %7,CKSWR ;CHECK FOR CONT G
BIT #10000,@SR ;SHOULD WE RUN WITH TRACE TRAP
BEQ YESTR ;YES
TST YESTR1 ;NO, HAVE WE RUN WITH TRACE TRAP ON?
BEQ TRPA ;IF SO RESTORE PREVIOUS CONTENTS
MOV YESTR1,14
MOV YESTR2,16
BIC #20,@PSW ;CLEAR TRACE TRAP
TRPA: JMP BEGIN ;START OF TEST WITH TRACE OFF
TRPB: 0

: SAVE OLD CONTENTS, SET UP FOR TRACE TRAP
:*****

1688 007534 016767 170254 000044
1689 007542 016767 170250 000040
1690 007550 012767 007612 170236
1691 007556 005067 170234
1692 007562 005077 171230
1693 007566 005167 177740
1694 007572 100403
1695 007574 052777 000020 171214
1696 007602 000167 171366

YESTR: MOV 14,YESTR1 ;SAVE ODT PC
MOV 16,YESTR2 ;SAVE ODT STATUS
MOV #YESRT,14 ;NEW TRAP VECTOR
CLR 16 ;NEW CONDITION CODES
CLR @PSW
COM TRPB
BMI .+10
BIS #20,@PSW ;SET TRACE TRAP
JMP BEGIN ;START OF TEST WITH TRACE ON

1698 007606 000000
1699 007610 000000
1700 007612 000002
1701 007614 000000

YESTR1: 0 ;STORAGE FOR ODT PC
YESTR2: 0 ;STORAG FOR ODT STATUS
YESRT: RTI ;RETURN TO PROGRAM FROM TRAP
HALT ;RTI FAILED

: BUS TO BUS TEST (DR11-B TO DR11-B)
:*****

1709 000000
1710 000001
1711 000002
1712 000003
1713 000004
1714 000005
1715 000006
1716 000007
1717 000007
1718 000006
1719
1720 000001
1721 000002

R0=%0
R1=%1
R2=%2
R3=%3
R4=%4
R5=%5
R6=%6
R7=%7
PC=%7
SP=%6

GO=1
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
1797 010032 001753          BEQ      BACKGD ;ANY JOBS?
1798 010034 004567 000032    JSR      R5,SAVALL ;BRANCH IF NONE
1799 010040 005067 177554    CLR      NEXJOB   ;YES & EXECUTE JOB WHOSE ADDRESS IS IN JBFLAG
1800 010044 005067 177552    CLR      NEXJOB+2
1801 010050 016767 177550 167756  MOV      JBFLAG,34
1802 010056 005067 177542    CLR      JBFLAG
1803 010062 104400          TRAP
1804 010064 004567 000016    JSR      R5,RESALL ;TRAP THROUGH JBFLAG AT 34
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 ADDRSS
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,@DRST  ;CHECK FOR ERROR
1858 010224 001411      BEQ      DRIN3              ;BRANCH IF NO ERROR
1859 010226 005767 177370      TST      NEXJOB+2           ;IS THERE AN ERROR SERVICE ROUTINE?
1860 010232 001002      BNE      DRIN1              ;BRANCH IF THERE IS.
1861 010234 104000      HLT
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     @DRST        ;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
```

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

```
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 FRM 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
```

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

```
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
```

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

```
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
```

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

1943
1944
1945


```

1946
1947 010544 032777 004000 170252 SJOB2: BIT #DSTATC,@DRST ;TEST FOR INTER
1948 010552 001002 SJOB2A: BNE SJOB2A
1949 010554 104000 HLT
1950 010556 000000 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)+,@DRBA ;SET UP BUS ADDRESS
1962 010632 012077 170162 MOV (R0)+,@DRWC ;SET UP WORD COUNT
1963 010636 005077 170162 CLR @DRST ;CLEAR ALL FUNCTION BITS
1964 010642 032777 002000 170154 BIT #DSTATB,@DRST ;WHICH DIRECTION
1965 010650 001405 BEQ SJOB2C ;BRANCH IF RECIEVE (LEAVE FNCT1 CLEAR FOR DATI'S)
1966 010652 032777 000400 170144 SJOB2B: BIT #CYCLE,@DRST ;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,@DRST ;SET FNCT2 FOR DATO'S
1972 010702 042777 000400 170114 BIC #CYCLE,@DRST ;CLEAR CYCLE
1973 010710 052777 000101 170106 SJOB2D: BIS #IE!GO,@DRST ;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 @DRST ;MAKE SURE ERROR IS CLEAR

```

```

2002 010774 100002          BPL      MXMIT1      :AND READY IS SET
2003 010776 104000          HLT
2004 011000 000000          HALT
2005 011002 105777 170016    MXMIT1: TSTB      @DRST      :ERROR IS SET
2006 011006 100402          BMI      MXMIT2
2007 011010 104000          HLT
2008 011012 000000          HALT
2009 011014 012777 000000 170002  MXMIT2: MOV      #0,@DRST      :READY NOT SET
2010 011022 004767 000102          JSR      R7,PRMXFR  :SET UP FUNCTION FOR DATI'S
2011 011026 042777 000400 167770    BIC      #CYCLE,@DRST :TRANSFER PARAMETERS OF LIST WHOSE ADDRESS IS IN RO
2012 011034 052777 000101 167762    BIS      #IE!GO,@DRST :MAKE SURE CYCLE IS CLEAR
2013 011042 000207          RTS      R7          :EXECUTE COMMAND AND INTERRUPT WHEN DONE
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
2023 011056 105777 167742    MREC1:  TSTB      @DRST      :ERROR SET
2024 011062 100402          BMI      MREC2
2025 011064 104000          HLT
2026 011066 000000          HALT
2027 011070 012777 000004 167726  MREC2:  MOV      #FNCT2,@DRST :READY CLEAR
2028 011076 004767 000026          JSR      R7,PRMXFR  :SET UP FUNCTION FOR DATO'S
2029 011102 032777 002000 167714  MREC3:  BIT      #DSTATB,@DRST :TRANSFER PARAMETERS OF LIST WHOSE ADDRESS IS IN RO
2030 011110 001374          BNE      MREC3        :WAIT FOR SLAVE TO CLEAR DIRECTION
2031 011112 042777 000400 167704    BIC      #CYCLE,@DRST :BRANCH IF SET
2032 011120 052777 000101 167676    BIS      #IE!GO,@DRST :CLEAR CYCLE
2033 011126 000207          RTS      R7          :EXECUTE COMMAND AND INTERRUPT WHEN DONE.
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 INTERUPT 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
2055 011202 040312          PRMXF3: BIC      R3,(R2)   :PARAMETER DID NOT ECHO
2056 011204 030412          PRMXF4: BIT      R4,(R2)   :DROP SLAVE'S ATTN
2057 011206 001376          BNE      PRMXF4      :WAIT FOR REPLY
                                :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
2109 011326 006105          ROL   R5
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
2160 011446 000207          RTS      R7        ;ADVANCE WORD COUNT
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
2171 011470 000000
2172 011472 000000
2173
2174 011474 014212
2175 011476 000000
2176 011500 000000
2177 011502 000000
2178 011504 000000
2179
2180
2181
2182
2183
2184
2185
2186

LISTB: XINBUF
0
0
0
0

:OFFSET
:CHECKSUM LOW
:CHECKSUM HIGH
:SLAVE'S ECHO BUFFER

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

2187 011506 004767 000672
2188 011512 037727 167276 020000
2189 011520 001067
2190 011522 012667 000204
2191 011526 012667 000202
2192 011532 024646
2193 011534 012777 000215 167324
2194 011542 105777 167316
2195 011546 100375
2196 011550 012777 000212 167310
2197 011556 105777 167302
2198 011562 100375
2199 011564 010267 000134
2200 011570 010367 000132
2201 011574 010467 000130
2202 011600 016702 000126
2203 011604 004767 000126
2204 011610 012777 000240 167250
2205 011616 105777 167242
2206 011622 100375
2207 011624 016702 000104
2208 011630 004767 000102
2209 011634 012777 000240 167224
2210 011642 105777 167216
2211 011646 100375
2212 011650 017702 167150
2213 011654 004767 000056
2214 011660 016702 000040
2215 011664 016703 000036
2216 011670 016704 000034
2217 011674 004767 000504
2218 011700 005777 167110
2219 011704 100001
2220 011706 000000
2221 011710 023737 000042 000046
2222 011716 001001
2223 011720 000000
2224 011722 000002
2225 011724 000000

PRINT: JSR %7,CKSWR
BIT @SR,#20000 :TEST FOR INHIBIT PRINT OUT
BNE 1\$:IF SO, BRANCH OVER PRINT
MOV (6)+,SAVPC :PC OF FAILING ROUTINE
MOV (6)+,SAVCC :CC OF ERROR CONDITION
CMP -(6),-(6) :REPOSITION THE STACK
MOV #215,@TPB :CR
TSTB @TPS
BPL .-4
MOV #212,@TPB :LINE FEED
TSTB @TPS
BPL .-4
MOV %2,SAVR2 :SAVE R2
MOV %3,SAVR3 :SAVE R3
MOV %4,SAVR4 :SAVE R4
MOV SAVPC,%2
JSR %7,PRTAB :PRINT OCTAL NUMBER
MOV #240,@TPB
TSTB @TPS :SPACE BETWEEN WORDS
BPL .-4
MOV SAVCC,%2
JSR %7,PRTAB :PRINT OCTAL NUMBER
MOV #240,@TPB :PRINT SPACE
TSTB @TPS :PRINTER DONE
BPL .-4 :BRANCH WHEN NOT DONE
MOV @DRST,%2 :GET DR11B STATUS
JSR %7,PRTAB :PRINT OCTAL NUMBER
MOV SAVR2,%2
MOV SAVR3,%3
MOV SAVR4,%4
JSR %7,CKSWR
1\$: TST @SR :CHECK SR FOR HALT SWITCH
BPL .+4
HALT :HALT ON ERROR UP IN SWR
CMP @#42,@#46 :ARE WE IN ACT11 AUTO MODE?
BNE .+4 :BRANCH ON NO
HALT :HALT ON ERROR IF IN ACT11 AUTO MODE
RTI :RETURN TO MAINLINE
SAVR2: 0

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	#261,@TPB
2245	012032	016703	000156		START:	MOV	SEVEN,%3
2246	012036	010267	000150			MOV	%2,TOODLE
2247	012042	005167	000144			COM	TOODLE
2248	012046	046703	000140			BIC	TOODLE,%3
2249	012052	001410				BEQ	WRTOC
2250	012054	066767	000136	000136	MKNUM:	ADD	DECML,WGTCT
2251	012062	005267	000134			INC	BINCT
2252	012066	026703	000126			CMP	WGTCT,%3
2253	012072	001370				BNE	MKNUM
2254	012074	062767	000260	000120	WRTOC:	ADD	#260,BINCT
2255	012102	016724	000114			MOV	BINCT,(4)+
2256	012106	066767	000102	000102		ADD	SEVEN,DECML
2257	012114	005067	000100			CLR	WGTCT
2258	012120	005067	000076			CLR	BINCT
2259	012124	005367	000074			DEC	ASCNT
2260	012130	001410				BEQ	XLIST
2261	012132	012703	000003			MOV	#3,%3
2262	012136	066767	000052	000050	MOADD:	ADD	SEVEN,SEVEN
2263	012144	005303				DEC	%3
2264	012146	001373				BNE	MOADD
2265	012150	000730				BR	START
2266	012152	012767	000005	000044	XLIST:	MOV	#5,ASCNT
2267	012160	105777	166790		WAIT2:	TSTB	@TPS
2268	012164	100375				BPL	WAIT2
2269	012166	014477	166674			MOV	-(4),@TPB
2270	012172	005367	000026			DEC	ASCNT
2271	012176	001401				BEQ	HDFHM
2272	012200	000767				BR	WAIT2
2273	012202	105777	166656		HDFHM:	TSTB	@TPS
2274	012206	100375				BPL	:-4
2275	012210	000207				RTS	%7
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,@SR ;SCOPE, BIT IS A ONE
BNE SCOPEB ;NO - SAVE PC FOR NEXT TIME
MOV @%6,RETURN ;RETURN IN SEQUENCE
RTI ;REPOSITION THE STACK
SCOPEB: CMP (6)+,%6
MOV (6)+,@PSW
JMP @RETURN ;SCOPE RETURN

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

2304
2305
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,@SR ;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,@SR ;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 ;COUNT LOCATION FOR ITERATION LOOP
SCOPEF: 0 ;ADDRESS OF LAST TEST
RETURN: BEGIN
.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 CNTLU: MOV #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 SREAD: CLR TEMPST
2355 012516 012767 000007 177652 MOV #7, COUNT
2356 012524 004767 000154 1$: JSR PC, TTIN :GO READ A CHARACTER
2357 012530 042767 177600 177644 BIC #177600, TIB :STRIP OFF GARBAGE
2358 012536 122767 000025 177636 CMPB #25, TIB :IS IT A ^U?
2359 012544 001001 BNE 2$ :BRANCH IF NOT
2360 012546 000743 3$: BR CNTLU :START OVER
2361 012550 122767 000015 177624 2$: 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 8$: BR OUT :GET OUT
2368 012602 122767 000060 177572 4$: 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 5$: MOV #SQUEST, %2
2373 012626 004767 000252 JSR PC, TTOUT
2374 012632 000745 BR 3$ :START OVER IF NOT LEGAL CHARACTER
2375 012634 006367 177534 6$: ASL TEMPST
2376 012640 006367 177530 ASL TEMPST
2377 012644 006367 177524 ASL TEMPST
2378 012650 142767 000060 177524 BICB #60, TIB :GET NITTY-GRITTY
2379 012656 156767 177520 177510 BISB TIB, TEMPST
2380 012664 005367 177506 DEC COUNT :ONLY WANT 6 DIGITS
2381 012670 001754 BEQ 5$
2382 012672 000714 BR 1$
2383 012674 016777 177474 166112 7$: MOV TEMPST, @SR :CHANGE SWITCH REGISTER CONTENTS
2384 012702 000207 OUT: RTS #7 :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		MOV	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	.REST	
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 TTOUT
.RETR: .BYTE 15,12, '!
.EVEN
.SAV: 0
=.+1000
BUFF: 0
XINBUF: .
=.+1000
XCHKBU: .
.END

;FOR STACK POINTER 100 LOCATIONS

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						

.SRAND 1#
.SRDDE 1#
.SRDOC 1#
.SREAD 1#
.SR2AZ 1#
.\$SAVE 1#
.\$SB2D 1#
.\$SB2O 1#
.\$SCOP 1#
.\$SIZE 1#
.\$SUPR 1#
.\$STRAP 1#
.\$STYPB 1#
.\$STYPD 1#
.\$STYPE 1#
.\$STYPO 1#
.\$40CA 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)