

# AR11

DIAGNOSTIC TEST I  
MD-11-DZARA-B

EP-DZARA-B-DL  
COPYRIGHT © 74-76  
FICHE 1 OF 1

MAY 1978  
**digital**  
MADE IN USA

This microfiche contains 98 frames of technical data, organized into 14 rows and 7 columns. Each frame displays a different set of diagnostic test results for the MD-11-DZARA-B system. The data is presented in a structured format, likely including test parameters, results, and error codes. The frames are arranged in a grid that is 14 rows high and 7 columns wide, with a small tab at the bottom center of the grid.



IDENTIFICATION

SEQ 0001

PRODUCT CODE:           MAINDEC-11-0ZARA-R-D  
PRODUCT NAME:           AR11 DIAGNOSTIC TEST I  
DATE:                    MAY     21, 1976  
MAINTAINER:             DIAGNOSTIC GROUP

COPYRIGHT (C) 1974, 1976 DIGITAL EQUIPMENT CORP., MAYNARD, MASS.

THIS SOFTWARE IS FURNISHED TO PURCHASER UNDER A LICENSE FOR USE ON A SINGLE COMPUTER SYSTEM AND CAN BE COPIED (WITH INCLUSION OF DEC'S COPYRIGHT NOTICE) ONLY FOR USE IN SUCH SYSTEM, EXCEPT AS MAY OTHERWISE BE PROVIDED IN WRITING BY DEC.

THE INFORMATION IN THIS DOCUMENT IS SUBJECT TO CHANGE WITHOUT NOTICE AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT CORPORATION.

DEC ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DEC.



1. ABSTRACT  
-----

THIS PROGRAM IS A LOGIC TEST OF THE AR-11 LAB SYSTEM OPTION. ALL FUNCTIONS OF THE OPTION WILL BE TESTED. DUE TO THE FLEXIBILITY OF THE OPTION, THE OPERATOR MAY BE REQUIRED TO SUPPLY OPTION CHARACTERISTICS.

2. REQUIREMENTS  
-----

2.1 EQUIPMENT

PDP-11 FAMILY COMPUTER WITH 8K WORDS OF MEMORY  
AR11 HEX OPTION MODULE INSTALLED  
TELETYPE

2.2 STORAGE

THIS PROGRAM USES LESS THAN 8K OF MEMORY.

3. LOADING PROCEDURE  
-----

PROCEDURE FOR NORMAL BINARY TAPES SHOULD BE FOLLOWED.

4. STARTING PROCEDURE  
-----

4.1 CONTROL SWITCH SETTINGS

THIS PROGRAM HAS BEEN MODIFIED TO RUN WITH OR WITHOUT A HARDWARE SWITCH REGISTER.

STANDARD PDP-11 FORMAT

SW 15 = 1	HALT ON ERROR
SW 14 = 1	LOOP ON TEST
SW 13 = 1	INHIBIT ERROR TYPINGS
SW 12 = 1	STORAGE SCOPE CONNECTED
SW 11 = 1	INHIBIT INTERACTIONS
SW 10 = 1	ALL 1'S A TO D TEST JUMPER INSTALLED
SW 09 = 1	LOOP ON ERROR
SW 08 = 1	LOOP ON TEST IN SWR <710>

REFER TO 9. FOR SOFTWARE SWITCH REGISTER CONTROL

4.2 STARTING ADDRESS OR ADDRESSES

200 IS THE STARTING ADDRESS OF THE LOGIC TEST.  
204 IS THE RESTART ADDRESS OF THE LOGIC TEST.



## 5. OPERATING PROCEDURE -----

THE OPERATOR MUST INSERT THE CORRECT INFORMATION IN THE SWITCH REGISTER WHEN REQUESTED BY THE PROGRAM OR AN ERROR WILL OCCUR. ONCE STARTED THE TEST WILL RUN IN ITS NORMAL MANNER WITHOUT OPERATOR INTERVENTION OR SWITCH SELECTION.

## 6. ERRORS -----

THIS PROGRAM USES THE DIAGNOSTIC "SYSMAC" PACKAGE FOR ERROR REPORTING AND TYPEOUT. REFER TO THE "ERROR POINTER TABLE" FOR TYPE AND DESCRIPTION OF ERRORS.

## 7. RESTRICTIONS -----

EXTERNAL A TO D START INPUT MUST NOT BE CONNECTED.

## 8. MISCELLANEOUS -----

### 8.1 EXECUTION TIME

THE LOGIC TEST WILL TAKE APPROXIMATELY 60 SECONDS FOR COMPLETION AND WILL TYPE "END PASS".

### 8.2 DEVICE ADDRESS PROGRAM LOCATIONS

LOCATION SBASE CONTAINS THE AR11 BASE DEVICE ADDRESS <170400>  
LOCATION SVECT1 CONTAINS THE AR11 BASE INTERRUPT VECTOR <340>  
LOCATION ADDR1 CONTAINS THE AR11 A TO D DR LEVEL <300><6>  
LOCATION CKR1 CONTAINS THE AR11 CLOCK DR LEVEL <300><6>  
LOCATION VCR1 CONTAINS THE AR11 SCOPE DR LEVEL <200><4>  
LOCATION SFILLS CONTAINS THE TTY FILLER CHARACTER COUNT  
LOCATION SNULL CONTAINS THE TTY FILLER CHARACTER

NOTE: IF THESE LOCATIONS ARE CHANGED, THE OPERATOR MUST START THE TEST AGAIN AT LOC. 200. THE PROGRAM WILL USE THE BASE ADDRESS AND VECTOR AND UPDATE THE ACTUAL PROGRAM VALUES.

### 8.3 MULTIPLE AR-11 TESTING

A PROVISION IS MADE FOR TESTING SEQUENTIAL AR-11. STARTING AT BUS ADDRESS/VECTOR DEFINED BY SBASE AND SVECT1. THE HEADER TYPEOUT WILL INFORM THE OPERATOR OF THE NUMBER OF AR-11'S FOUND.

### 8.4 XXDP/ACT/APT NOTES

THIS PROGRAM IS A CHAINABLE PROGRAM UNDER XXDP/ACT. THE APT HOOKS HAVE BEEN INSTALLED BUT NOT TESTED.



## 9. SOFTWARE SWITCH REGISTER OPERATION

-----  
THE PROGRAM SUPPORTS NON-SWITCH REGISTER CPU TYPES.  
A CHANGE IN SWR VALUE IS ACCOMPLISHED BY TYPING A "CTRL G".  
THE RESPONSE WILL BE "SWR = " AND WAIT FOR A NEW VALUE.  
THE OPERATOR NOW INPUTS THE NEW VALUE AND TERMINATES WITH A "CR".

## 10. TABLE OF CONTENTS

-----  
ATTACHED.



13	BASIC DEFINITIONS
18	OPERATIONAL SWITCH SETTINGS
19	TRAP CATCHER
(1)	STARTING ADDRESS(ES)
23	ACT11 HOOKS
25	APT PARAMETER BLOCK
26	COMMON TAGS
(2)	APT MAILBOX-ETABLE
(1)	ERROR POINTER TABLE
230	T1 TEST EACH BUS ADDRESS FOR TIMEOUT
250	T2 TEST THAT THE PRESET BUFFER CAN HOLD 00
258	T3 TEST THE COUNTER PRESET BUFFER CAN HOLD 0377
266	T4 TEST THAT PRESET BUFFER CAN HOLD 0125
274	T5 TEST THAT PRESET BUFFER CAN HOLD 0252
283	T6 TEST THAT PRESET BUFFER CAN HOLD A COUNT PATTERN
295	T7 TEST INIT TO CLEAR COUNT PRESET BUFFER WHEN IT IS --1
303	T10 TEST THAT THE COUNTER CAN HOLD 00
311	T11 TEST THE COUNTER CAN HOLD 0377
321	T12 TEST THAT COUNTER CAN HOLD 0125
329	T13 TEST THAT COUNTER CAN HOLD 0252
337	T14 TEST THAT COUNTER CAN HOLD A COUNT PATTERN
349	T15 TEST INIT TO CLEAR COUNTER WHEN IT IS --1
359	T16 TEST ENABLE COUNTER (BIT 0) CAN BE SET
367	T17 TEST RATE SELECT (BIT 1) MAY BE SET
375	T20 TEST THAT RATE SELECT (BIT 2) MAY BE SET
383	T21 TEST THAT RATE SELECT (BIT 3) MAY BE SET
392	T22 TEST CLOCK INTERRUPT ENABLE (BIT 6) CAN BE SET
400	T23 TEST MODE (BIT 8) CAN BE SET
400	T24 TEST EXT INTERRUPT ENABLE (BIT 14) CAN BE SET
416	T25 TEST THAT CLK DONE (BIT 7) CAN BE SET
423	T26 TEST THAT CLK EXT INPUT (BIT 15) CAN BE SET
431	T27 TEST THAT THE EXT FLAG DOES NOT SET FROM THE OUTSIDE SOURCE
441	T30 MAINT. COUNT THE COUNTER REGISTER AT RATE 1MHZ
461	T31 TEST THAT OVERFLOW SET CLK DONE (BIT 7)
473	T32 MAINT. COUNT THE COUNTER REGISTER AT RATE 0100KHZ
493	T33 MAINT. COUNT THE COUNTER REGISTER AT RATE 010KHZ
513	T34 MAINT. COUNT THE COUNTER REGISTER AT RATE 01KHZ
530	T35 MAINT. COUNT THE COUNTER REGISTER AT RATE 0100HZ
540	T36 TEST THAT RESET CLEARS RATE SELECT AND MODE BITS
557	T37 TEST THAT RESET CLEARS CLK INTERRUPT ENABLE
566	T40 TEST THAT RESET CLEARS CLK FLAGS
575	T41 TEST THAT RESET CLEARS COUNTER ENABLE
587	T42 TEST CLOCK TO COUNT UP AT 1 MHZ
602	T43 TEST CLOCK TO COUNT UP AT 100KHZ
617	T44 TEST CLOCK TO COUNT UP AT 10 KHZ
628	T45 TEST CLOCK TO COUNT UP AT 1KHZ
637	T46 TEST CLOCK TO COUNT UP AT 100HZ
646	T47 TEST THAT CLOCK ENABLE DOES NOT CLEAR ON DONE (MODE 1) 1 KHZ
665	T50 TEST THAT CLOCK ENABLE DOES NOT CLEAR DONE (MODE 1 100HZ)
682	T51 CLOCK PRE-INTERRUPT SETUP
695	T52 TEST THAT THE CLOCK INTERRUPTS AT LEVEL INDICATED -1
710	T53 TEST THAT THE CLOCK DOES NOT INTERRUPT AT LEVEL INDICATED
741	T54 TEST 1MHZ REPEATABILITY
751	T55 TEST 100 KHZ REPEATABILITY
760	T56 TEST 10 KHZ REPEATABILITY
769	T57 TEST 1KHZ REPEATABILITY



TABLE OF CONTENTS

779	T60	TEST 100HZ REPEATIBILITY
789	T61	TEST THAT RESET SETS VC READY BIT
797	T62	TEST THAT VC MODE BIT 2 CAN BE SET AND CLEARED
811	T63	TEST THAT VC MODE BIT 3 CAN BE SET
820	T64	TEST THAT VC INTERRUPT ENABLE (BIT 6) CAN BE SET
828	T65	TEST THAT CHANNEL (BIT 9) CAN BE SET
836	T66	TEST THAT STORE (BIT 10) CAN BE SET
843	T67	TEST THAT WRITE THRU (BIT 11) CAN BE SET
850	T70	TEST THAT ERASE (BIT 12) CAN BE SET
859	T71	TEST THAT THE X REGISTER CAN BE CLEARED
867	T72	TEST THAT THE X REGISTER CAN BE LOADED WITH 01777
875	T73	TEST THAT THE X REGISTER CAN BE LOADED WITH 0525
883	T74	TEST THAT THE X REGISTER CAN BE LOADED WITH 01252
892	T75	TEST THAT THE X REGISTER CAN HOLD A COUNT PATTERN
905	T76	TEST THAT THE Y REGISTER CAN BE CLEARED
913	T77	TEST THAT THE Y REGISTER CAN BE LOADED WITH 01777
921	T100	TEST THAT THE Y REGISTER CAN BE LOADED WITH 0525
930	T101	TEST THAT THE Y REGISTER CAN BE LOADED WITH 01252
938	T102	TEST THAT THE Y REGISTER CAN HOLD A COUNT PATTERN
951	T103	TEST THAT THE X-Y REGISTERS CAN HOLD DIFFERENT DATA
967	T104	TEST THAT WHEN INTENSIFY BIT IS SET THAT THE VC READY BIT CLEARS
985	T105	TEST THAT VC MODE 1 (INTENSIFY ON X) CLEARS THE READY FLAG
1016	T106	TEST THAT VC MODE 2 (INTENSIFY ON Y) CLEARS THE READY FLAG
1043	T107	TEST WHEN ERASE IS SET, VC READY BIT CLEARS AND SET AFTER DELAY
1064	T110	SCOPE PRE-INTERRUPT SETUP
1074	T111	TEST THAT THE DISPLAY DOES INTERRUPT AT LEVEL INDICATED =1
1087	T112	TEST THAT THE DISPLAY DOES NOT INTERRUPT AT LEVEL INDICATED
1107	T113	TEST THAT RESET CLEARS VC MODE BITS
1116	T114	TEST THAT RESET CLEARS INTERRUPT ENABLE, CHANNEL, STORE, WRITE THRU
1125	T115	TEST THAT RESET CLEARS X REGISTER
1134	T116	TEST THAT RESET CLEARS Y REGISTER
1144	T117	DOES EXTERNAL ENABLE (BIT 4) SET
1152	T120	DOES CLOCK OVERFLOW ENABLE (BIT 5) SET
1160	T121	DOES AD INTERRUPT ENABLE (BIT 6) SET
1168	T122	DOES MUX CHANNEL (BIT 8) SET
1176	T123	DOES MUX CHANNEL (BIT 9) SET
1184	T124	DOES MUX CHANNEL (BIT 10) SET
1192	T125	DOES MUX CHANNEL (BIT 11) SET
1200	T126	DOES UNIPOLAR/BIPOLAR (BIT 13) SET
1208	T127	TEST FOR PROPER SELECTION OF THE LOW BYTE OPERATION
1218	T130	TEST FOR PROPER SELECTION OF THE HIGH BYTE OPERATION
1229	T131	TEST AD 00 (BIT 0) CAN BE SET AND CLEARED
1249	T132	DOES AD DONE (BIT 7) SET AND CLEAR
1274	T133	TEST THAT THE CONVERTED NUMBER = 1777 (9H BIT 10=1)
1290	T134	TEST THAT NO EXTERNAL CONVERSIONS INPUT
1303	T135	TEST THAT CLOCK CAN START A CONVERSION
1310	T136	A TO D PRE-INTERRUPT SETUP
1329	T137	TEST THAT A TO D INTERRUPTS AT LEVEL INDICATED =1
1345	T140	TEST THAT THE A TO D DOES NOT INTERRUPT AT LEVEL INDICATED
1371	T141	TEST THAT RESET CLEARS MUX AND UNIPOLAR BITS
1381	T142	TEST THAT RESET CLEARS EXT AND INTERRUPT ENABLE BITS
1389	T143	TEST THAT RESET CLEARS AD DONE
1398	T144	TEST THAT RESET CLEARS AD BUFFER REG
1409	T145	LOAD DIFFERENT NUMBERS INTO DIFFERENT REG.
1447	T146	DETERMINE IF MORE AR11'S ARE TO BE TESTED
1460		END OF PASS ROUTINE



1479	CONVERT BINARY TO DECIMAL AND TYPE ROUTINE
1577	SCOPE HANDLER ROUTINE
1578	ERROR HANDLER ROUTINE
1579	TTY INPUT ROUTINE
1580	READ AN OCTAL NUMBER FROM THE TTY
1583	ERROR MESSAGE TIMEOUT ROUTINE
1586	POWER DOWN AND UP ROUTINES
1591	BINARY TO OCTAL (ASCII) AND TYPE
1592	TYPE ROUTINE
1593	APT COMMUNICATIONS ROUTINE
1594	TRAP DECODER
(3)	TRAP TABLE







(1)	010000	SW12=	10000
(1)	004000	SW11=	4000
(1)	002000	SW10=	2000
(1)	001000	SW09=	1000
(1)	000400	SW08=	400
(1)	000200	SW07=	200
(1)	000100	SW06=	100
(1)	000040	SW05=	40
(1)	000020	SW04=	20
(1)	000010	SW03=	10
(1)	000004	SW02=	4
(1)	000002	SW01=	2
(1)	000001	SW00=	1
(1)		.EQUIV	SW09,SW9
(1)		.EQUIV	SW08,SW8
(1)		.EQUIV	SW07,SW7
(1)		.EQUIV	SW06,SW6
(1)		.EQUIV	SW05,SW5
(1)		.EQUIV	SW04,SW4
(1)		.EQUIV	SW03,SW3
(1)		.EQUIV	SW02,SW2
(1)		.EQUIV	SW01,SW1
(1)		.EQUIV	SW00,SW0

(1)		I=DATA BIT DEFINITIONS (BIT0 TO BIT15)	
(1)	100000	BIT15=	100000
(1)	040000	BIT14=	40000
(1)	020000	BIT13=	20000
(1)	010000	BIT12=	10000
(1)	004000	BIT11=	4000
(1)	002000	BIT10=	2000
(1)	001000	BIT09=	1000
(1)	000400	BIT08=	400
(1)	000200	BIT07=	200
(1)	000100	BIT06=	100
(1)	000040	BIT05=	40
(1)	000020	BIT04=	20
(1)	000010	BIT03=	10
(1)	000004	BIT02=	4
(1)	000002	BIT01=	2
(1)	000001	BIT00=	1
(1)		.EQUIV	BIT09,BIT9
(1)		.EQUIV	BIT08,BIT8
(1)		.EQUIV	BIT07,BIT7
(1)		.EQUIV	BIT06,BIT6
(1)		.EQUIV	BIT05,BIT5
(1)		.EQUIV	BIT04,BIT4
(1)		.EQUIV	BIT03,BIT3
(1)		.EQUIV	BIT02,BIT2
(1)		.EQUIV	BIT01,BIT1
(1)		.EQUIV	BIT00,BIT0

(1)		I=BASIC "CPU" TRAP VECTOR ADDRESSES		
(1)	000004	ERRVEC=	4	;;TIME OUT AND OTHER ERRORS
(1)	000010	RESVEC=	10	;;RESERVED AND ILLEGAL INSTRUCTIONS
(1)	000014	TBITVEC=	14	;;"T" BIT



(1)	000014	TRTVEC= 14	;;TRACE TRAP
(1)	000014	BPTVEC= 14	;;BREAKPOINT TRAP (BPT)
(1)	000020	IOTVEC= 20	;;INPUT/OUTPUT TRAP (IOT) **SCOPE**
(1)	000024	PWRVEC= 24	;;POWER FAIL
(1)	000030	EMTVEC= 30	;;EMULATOR TRAP (EMT) **ERROR**
(1)	000034	TRAPVEC=34	;;"TRAP" TRAP
(1)	000060	TKVEC= 60	;;TTY KEYBOARD VECTOR
(1)	000064	TPVEC= 64	;;TTY PRINTER VECTOR
(1)	000240	PIRQVEC=240	;;PROGRAM INTERRUPT REQUEST VECTOR
14	170400	ABASE=170400	
15	000340	AVECT1=340	
16	000200	APRIOR=200	



```

18
(1)          .SBTTL OPERATIONAL SWITCH SETTINGS
(1)          ;o
(1)          ;o      SWITCH              USE
(1)          ;o      -----              -----
(1)          ;o          15              HALT ON ERROR
(1)          ;o          14              LOOP ON TEST
(1)          ;o          13              INHIBIT ERROR TYPEOUTS
(1)          ;o          12              STORAGE SCOPE CONNECTED
(1)          ;o          11              INHIBIT ITERATIONS
(1)          ;o          10              ALL 1'S JUMPER IS INSTALLED
(1)          ;o           9              LOOP ON ERROR
(1)          ;o           8              LOOP ON TEST IN SWR<710>
19
(1)          .SBTTL TRAP CATCHER
(1)          ;o
(1)          ;o      .=0
(1)          ;o      ;oALL UNUSED LOCATIONS FROM 0 - 776 CONTAIN A ".+2,HALT"
(1)          ;o      ;oSEQUENCE TO CATCH ILLEGAL TRAPS AND INTERRUPTS
(1)          ;o      ;oLOCATION 0 CONTAINS 0 TO CATCH IMPROPERLY LOADED VECTORS
(1)          ;o      .=176
(1)          ;o      DISPREG: .WORD 0          ;;SOFTWARE DISPLAY REGISTER
(1)          ;o      SWREG:  .WORD 0          ;;SOFTWARE SWITCH REGISTER
(1)          .SBTTL STARTING ADDRESS(ES)
(1)          ;o      JMP      000BEGIN ;;JUMP TO STARTING ADDRESS OF PROGRAM
20          ;o      JMP      000BEGIN1 ;;JUMP TO RESTART ADDRESS
(1)          ;o      000200 000137 001576
(1)          ;o      000204 000137 001602

```







```

26
(1)          .SBTTL  COMMON TAGS
(1)
(2)          ;;*****
(1)          ;THIS TABLE CONTAINS VARIOUS COMMON STORAGE LOCATIONS
(1)          ;USED IN THE PROGRAM.
(1)
(1)          001100          .01100
(1) 001100          000000          SBTAG:          .WORD          0          ;;START OF COMMON TAGS
(1) 001102          000          STSTNM: .BYTE          0          ;;CONTAINS THE TEST NUMBER
(1) 001103          000          SERFLG: .BYTE          0          ;;CONTAINS ERROR FLAG
(1) 001104          000000          SICNT: .WORD          0          ;;CONTAINS SUBTEST ITERATION COUNT
(1) 001106          000000          SLPADR: .WORD          0          ;;CONTAINS SCOPE LOOP ADDRESS
(1) 001110          000000          SLPERR: .WORD          0          ;;CONTAINS SCOPE RETURN FOR ERRORS
(1) 001112          000000          SERTTL: .WORD          0          ;;CONTAINS TOTAL ERRORS DETECTED
(1) 001114          000          SITEMB: .BYTE          0          ;;CONTAINS ITEM CONTROL BYTE
(1) 001115          001          SERMAX: .BYTE          1          ;;CONTAINS MAX. ERRORS PER TEST
(1) 001116          000000          SERRPC: .WORD          0          ;;CONTAINS PC OF LAST ERROR INSTRUCTION
(1) 001120          000000          SBDADR: .WORD          0          ;;CONTAINS ADDRESS OF 'GOOD' DATA
(1) 001122          000000          SBDADR: .WORD          0          ;;CONTAINS ADDRESS OF 'BAD' DATA
(1) 001124          000000          SBDAT: .WORD          0          ;;CONTAINS 'GOOD' DATA
(1) 001126          000000          SBDAT: .WORD          0          ;;CONTAINS 'BAD' DATA
(1) 001130          000000          .WORD          0          ;;RESERVED--NOT TO BE USED
(1) 001132          000000          .WORD          0
(1) 001134          000000          .WORD          0
(1) 001136          177570          SWR: .WORD          DSWR          ;;ADDRESS OF SWITCH REGISTER
(1) 001140          177570          DISPLAY: .WORD          DDISP          ;;ADDRESS OF DISPLAY REGISTER
(1) 001142          177560          STKS: 177560          ;;TTY KBD STATUS
(1) 001144          177562          STKB: 177562          ;;TTY KBD BUFFER
(1) 001146          177564          STPB: 177564          ;;TTY PRINTER STATUS REG. ADDRESS
(1) 001150          177566          STPB: 177566          ;;TTY PRINTER BUFFER REG. ADDRESS
(1) 001152          000          SNULL: .BYTE          0          ;;CONTAINS NULL CHARACTER FOR FILLS
(1) 001153          002          SFILLS: .BYTE          2          ;;CONTAINS # OF FILLER CHARACTERS REQUIRED
(1) 001154          012          SFILLC: .BYTE          12          ;;INSERT FILL CHARS. AFTER A "LINE FEED"
(1) 001155          000          STPFLE: .BYTE          0          ;;"TERMINAL AVAILABLE" FLAG (BIT<07>=0=YES)
(1) 001156          000000          SREGAD: .WORD          0          ;;CONTAINS THE ADDRESS FROM
(1)          ;;WHICH (SREGO) WAS OBTAINED
(3) 001160          000000          SREGO: .WORD          0          ;;CONTAINS ((SREGAD)+0)
(3) 001162          000000          SREG1: .WORD          0          ;;CONTAINS ((SREGAD)+2)
(1) 001164          000000          STINES: 0          ;;MAX. NUMBER OF ITERATIONS
(1) 001166          000000          SESCAPE: 0          ;;ESCAPE ON ERROR ADDRESS
(1) 001170          077          SQUES: .ASCII          /?/          ;;QUESTION MARK
(1) 001171          015          SCRFL: .ASCII          <15>          ;;CARRIAGE RETURN
(1) 001172          000012          SLP: .ASCII          <12>          ;;LINE FEED

```



```
(2)                                     ;*****  
(2)                                     ;  
(2)                                     ;  
(2)                                     ;  
(3)                                     ;*****  
(2)                                     ;  
(2) 001174                               .EVEN  
(2) 001174 000000                      SMAIL:                               ;IAPT MAILBOX  
(2) 001176 000000                      SMSGTY: .WORD  AMSGTY                ;MESSAGE TYPE CODE  
(2) 001200 000000                      SFATAL: .WORD  AFATAL                ;FATAL ERROR NUMBER  
(2) 001202 000000                      STESTN: .WORD  ATESTN               ;TEST NUMBER  
(2) 001204 000000                      SPASS: .WORD   APASS                 ;PASS COUNT  
(2) 001206 000000                      SDEVCT: .WORD  ADEVCT               ;DEVICE COUNT  
(2) 001208 000000                      SUNIT: .WORD   AUNIT                ;I/O UNIT NUMBER  
(2) 001210 000000                      SMSGAD: .WORD  AMSGAD               ;MESSAGE ADDRESS  
(2) 001212 000000                      SMSGLG: .WORD  AMSGLG               ;MESSAGE LENGTH  
(2) 001214                               SETABLE:                             ;IAPT ENVIRONMENT TABLE  
(2) 001214 000                          SENVI: .BYTE   AENV                 ;ENVIRONMENT BYTE  
(2) 001215 000                          SENVM: .BYTE   AENVM                ;ENVIRONMENT MODE BITS  
(2) 001216 000000                      SSWREG: .WORD  ASWREG               ;IAPT SWITCH REGISTER  
(2) 001220 000000                      SUSWR: .WORD   AUSWR                ;USER SWITCHES  
(2) 001222 000000                      SPCUOP: .WORD  ACPUOP              ;CPU TYPE, OPTIONS  
(2)                                     ;  
(2)                                     ;  
(2)                                     ;  
(2)                                     ;  
(2)                                     ;  
(2)                                     ;  
(2) 001224 000                          SHAMS1: .BYTE  AHAMS1               ;HIGH ADDRESS, H.S. BYTE  
(2) 001225 000                          SHAMP1: .BYTE  AHAMP1               ;MEM. TYPE, BLK#1  
(2)                                     ;  
(2)                                     ;  
(2)                                     ;  
(2)                                     ;  
(2) 001226 000000                      SHADR1: .WORD  AHADR1               ;HIGH ADDRESS, BLK#1  
(2)                                     ;  
(2) 001230 000                          SHAMS2: .BYTE  AHAMS2               ;HIGH ADDRESS, H.S. BYTE  
(2) 001231 000                          SHAMP2: .BYTE  AHAMP2               ;MEM. TYPE, BLK#2  
(2) 001232 000000                      SHADR2: .WORD  AHADR2               ;MEM. LAST ADDRESS, BLK#2  
(2) 001234 000                          SHAMS3: .BYTE  AHAMS3               ;HIGH ADDRESS, H.S. BYTE  
(2) 001235 000                          SHAMP3: .BYTE  AHAMP3               ;MEM. TYPE, BLK#3  
(2) 001236 000000                      SHADR3: .WORD  AHADR3               ;MEM. LAST ADDRESS, BLK#3  
(2) 001240 000                          SHAMS4: .BYTE  AHAMS4               ;HIGH ADDRESS, H.S. BYTE  
(2) 001241 000                          SHAMP4: .BYTE  AHAMP4               ;MEM. TYPE, BLK#4  
(2) 001242 000000                      SHADR4: .WORD  AHADR4               ;MEM. LAST ADDRESS, BLK#4  
(2) 001244 340                          SVECT1: .BYTE  AVECT1               ;INTERRUPT VECTOR#1  
(2) 001245 000                          SVECT2: .BYTE  AVECT2               ;INTERRUPT VECTOR#2  
(2) 001246 200                          SPRIOR: .BYTE  APRIOR               ;BUS PRIORITY #1, #2  
(2) 001247 000                          ;  
(2)                                     ;  
(2) 001250 170400                      SBASE: .WORD  ABASE                 ;BASE ADDRESS OF EQUIPMENT UNDER TEST  
(2) 001252 000000                      SDEVN: .WORD  ADEVN                 ;DEVICE MAP  
(2) 001254 000000                      SCDW1: .WORD  ACDW1                 ;CONTROLLER DESCRIPTION WORD#1  
(2) 001256 000000                      SCDW2: .WORD  ACDW2                 ;CONTROLLER DESCRIPTION WORD#2  
(2) 001260 000000                      SDDW0: .WORD  ADDW0                 ;DEVICE DESCRIPTOR WORD#0  
(2) 001262 000000                      SDDW1: .WORD  ADDW1                 ;DEVICE DESCRIPTOR WORD#1  
(2) 001264 000000                      SDDW2: .WORD  ADDW2                 ;DEVICE DESCRIPTOR WORD#2
```





50	001356	000000		0				
51								
52			ITEM	5				
53	001360	014253		EM5	ICLOCK INTERRUPT ERROR			
54	001362	015040		DM5	IERRPC CSR			
55	001364	015532		DT5	ISERRPC CSR			
56	001366	000000		0				
57								
58			ITEM	6				
59	001370	014301		EM6	ICLOCK COUNTER REGISTER IN ERROR			
60	001372	015055		DM6	IERRPC CSC CKCNTR EXPECTED			
61	001374	015540		DT6	ISERRPC CSC SBDDAT SGDDAT			
62	001376	000000		0				
63								
64			ITEM	7				
65	001400	014341		EM7	ICLOCK COUNTED IN ERROR			
66	001402	015055		DM6	IERRPC CSC CKCNTR EXPECTED			
67	001404	015540		DT6	ISERRPC CSC SBDDAT SGDDAT			
68	001406	000000		0				
69								
70			ITEM	10				
71	001410	014370		EM10	ICLOCK REPEATABILITY FAILED			
72	001412	015115		DM10	IERRPC CSC TIME1 TIME2			
73	001414	015540		DT6	ISERRPC CSC SBDDAT SGDDAT			
74	001416	000000		0				
75								
76			ITEM	11				
77	001420	014423		EM11	IVC STATUS REGISTER IN ERROR			
78	001422	015153		DM11	IERRPC VCAOR VCSTAT EXPECTED			
79	001424	015552		DT11	ISERRPC VCSTAT SBDDAT SGDDAT			
80	001426	000000		0				
81								
82			ITEM	12				
83	001430	014457		EM12	IX REGISTER IN ERROR			
84	001432	015213		DM12	IERRPC VCXPOS X POS. EXPECTED			
85	001434	015564		DT12	ISERRPC VCXPOS SBDDAT SGDDAT			
86	001436	000000		0				
87								
88			ITEM	13				
89	001440	014503		EM13	IY REGISTER IN ERROR			
90	001442	015254		DM13	IERRPC VCYPOS Y POS. EXPECTED			
91	001444	015576		DT13	ISERRPC VCYPOS SBDDAT SGDDAT			
92	001446	000000		0				
93								
94			ITEM	14				
95	001450	014527		EM14	ISCOPE INTERRUPT ERROR			
96	001452	015315		DM14	IERRPC VCAOR			
97	001454	015610		DT14	ISERRPC VCSTAT			
98	001456	000000		0				
99								
100			ITEM	15				
101	001460	014555		EM15	IDevice BUS ERROR			
102	001462	015334		DM15	IERRPC BASE ADDRESS ACTUAL ADDRESS			
103	001464	015616		DT15	ISERRPC ARBADD SBDDAT			
104	001466	000000		0				
105								



107			ITEM	16				
108	001470	014576		EM16	INCORRECT A/D BUFFER DATA			
109	001472	015364		DM16	IERRPC	ADDR	BUFFER	EXPECTED
110	001474	015626		DT16	IERRPC	ADDR	SBDDAT	SGDDAT
111	001476	000000		0				
112								
113			ITEM	17				
114	001500	014630		EM17	DUAL REGISTER ADDRESSING			
115	001502	015425		DM17	IERRPC	BUFADR	READ	EXPECTED
116	001504	015640		DT17	IERRPC	BUFADR	SBDDAT	SGDDAT
117	001506	000000		0				

119				
120	001510	000300	ADBRL: 300	JA TO D BR LEVEL
121	001512	000300	CKBRL: 300	JCLOCK BR LEVEL
122	001514	000200	VCBRL: 200	JSCOPE BR LEVEL
123				
124			PROGRAM WILL CHANGE THE NEXT 16 LOCATIONS	
125				
126	001516	170400	ADCS: 170400	JA TO D STATUS REG
127	001520	170402	ADDR: 170402	JA TO D BUFFER
128				
129	001522	170404	CSR: 170404	JCLOCK STATUS REGISTER
130	001524	170406	CSS: 170406	JCLOCK PRESET BUFFER
131				
132	001526	170410	VCSTAT: 170410	JVC STATUS REGISTER
133	001530	170412	VCXREG: 170412	JVC X AXIS REGISTER
134	001532	170414	VCYREG: 170414	JVC Y AXIS REGISTER
135				
136				
137	001534	170416	CSC: 170416	JCLOCK COUNTER REGISTER
138				
139	001536	000340	ADINT: 340	JA TO D INTERRUPT VECTOR
140	001540	000342	ADINT1: 342	
141				
142	001542	000344	KWIV: 344	JCLOCK INTERRUPT VECTOR
143	001544	000346	KWIV8: 346	
144				
145	001546	000350	VCIV: 350	JSCOPE INTERRUPT VECTOR
146	001550	000352	VCIV8: 352	
147				
148	001552	170400	ARBADD: 170400	JCURRENT DEVICE ADDRESS
149	001554	000340	ARBVCT: 340	JCURRENT DEVICE VECTOR
150	001556	000000	NMBEXT: 0	JNUMBER OF ADDITIONAL AR11'S
151	001560	000000	NBEXT: 0	
152	001562	000000	COUNT: 0	
153	001564	000000	DELAY: 0	
154	001566	000000	TEMP: 0	
155	001570	000000	SWITCH: 0	
156	001572	000000	BRLEV1: 0	
157	001574	000000	BRLEV2: 0	
158				



160									
161	001576	005000				BEGIN:	CLR	R0	ICLEAR R0
162	001600	000402					BR	R0EG	
163	001602	012700	177777			BEGIN1:	MOV	R-1,R0	ILOAD R0
164	001606	000005				RBEG:	RESET		
165	001610	012706	001100				MOV	@STACK,SP	ILOAD STACK
166	001614	012737	001642	000004			MOV	@18,004	ILOAD BUS ERROR
167	001622	013702	001250				MOV	@BASE,R2	ILOAD STARTING ADDRESS
168	001626	005003					CLR	R3	ICLEAR COUNT
169	001630	005712				28:	TST	(R2)	I TEST IF EXISTENT
170	001632	062702	000020				ADD	@20,R2	IEXIST, UPDATE TEST ADDRESS
171	001636	005203					INC	R3	IUPDATE # OF AR11'S
172	001640	000773					BR	29	
173	001642	022626				18:	CMP	(SP)+,(SP)+	IPOP STACK
174	001644	005703					TST	R3	I TEST IF FIRST DOES EXIST
175	001646	001001					BNE	38	IBR
176	001650	000000					HALT		I FIRST AR11 DOES NOT EXIST
177									ICHECK THE PROGRAM DEVICE ADDRESS
178	001652	005303				38:	DEC	R3	IADJUST R3
179	001654	010337	001556				MOV	R3,NMBEXT	ISAVE THE NUMBER OF ADDITIONAL AR11
180	001660	012737	000006	000004			MOV	@6,004	I RESET BUS ERROR
181	001666	005037	000006				CLR	006	
182	001672	013737	001250	001552			MOV	@BASE,ARBADD	ILOAD FIRST ADDRESS
183	001700	013737	001244	001554			MOV	@VECT1,ARBVCT	ILOAD FIRST VECTOR
184	001706	013737	001556	001560			MOV	NMBEXT,NBEXT	ILOAD NUMBER OF AR11'S
185	001714	000005				RBEG1:	RESET		
186									ICLEAR THE COMMON TAGS (SCHTAG) AREA
(1)	001716	012706	001100				MOV	@SCHTAG,R6	IFIRST LOCATION TO BE CLEARED
(1)	001722	005026					CLR	(R6)+	ICLEAR MEMORY LOCATION
(1)	001724	022706	001126				CMP	@SBODAT,R6	IDONE?
(1)	001730	001374					BNE	.-6	ILOOP BACK IF NO
(1)	001732	012706	001100				MOV	@STACK,SP	ISETUP THE STACK POINTER
(1)									IIINITIALIZE A FEW VECTORS
(1)	001736	012737	015652	000020			MOV	@SCOPE,@IOTVEC	IIOT VECTOR FOR SCOPE ROUTINE
(1)	001744	012737	000340	000022			MOV	@340,@IOTVEC+2	IILEVEL 7
(1)	001752	012737	016132	000030			MOV	@ERROR,@EMTVEC	IIEMT VECTOR FOR ERROR ROUTINE
(1)	001760	012737	000340	000032			MOV	@340,@EMTVEC+2	IILEVEL 7
(1)	001766	012737	020440	000034			MOV	@TRAP,@TRAPVEC	II TRAP VECTOR FOR TRAP CALLS
(1)	001774	012737	000340	000036			MOV	@340,@TRAPVEC+2	IILEVEL 7
(1)	002002	012737	017242	000024			MOV	@SPHRDN,@SPHRVEC	IIPOWER FAILURE VECTOR
(1)	002010	012737	000340	000026			MOV	@340,@SPHRVEC+2	IILEVEL 7
(1)	002016	013737	013310	013302			MOV	@ENDCT,@EOPCT	IISETUP END-OF-PROGRAM COUNTER
(1)	002024	005037	001164				CLR	@TIMES	IIINITIALIZE NUMBER OF ITERATIONS
(1)	002030	005037	001166				CLR	@ESCAPE	IICLEAR THE ESCAPE ON ERROR ADDRESS
(1)	002034	112737	000001	001115			MOV	@1,@ERRMAX	IIALLOW ONE ERROR PER TEST
(1)	002042	012737	002042	001106			MOV	@.,@SLPADR	IIINITIALIZE THE LOOP ADDRESS FOR SCOPE
(1)	002050	012737	002050	001110			MOV	@.,@SLPERR	IISETUP THE ERROR LOOP ADDRESS
(2)									II SIZE FOR A HARDWARE SWITCH REGISTER, IF NOT FOUND OR IT IS
(2)									II EQUAL TO A "-1", SETUP FOR A SOFTWARE SWITCH REGISTER.
(2)	002056	013746	000004				MOV	@ERRVEC,-(SP)	II SAVE ERROR VECTOR
(2)	002062	012737	002120	000004			MOV	@648,@ERRVEC	IISET UP ERROR VECTOR
(2)	002070	012737	177570	001136			MOV	@DSHR,SWR	IISETUP FOR A HARDWARE SWICH REGISTER
(2)	002076	012737	177570	001140			MOV	@DISP,DISPLAY	IIAND A HARDWARE DISPLAY REGISTER
(2)	002104	022777	177777	177024			CMP	R-1,@SWR	II TRY TO REFERENCE HARDWARE SWR
(2)	002112	001013					BNE	658	II BRANCH IF NO TIMEOUT TRAP OCCURRED
(2)									II AND THE HARDWARE SWR IS NOT = -1

(2)	002114	005737	000001			TST	001	;;FORCE A TRAP THROUGH ERRVEC
(2)	002120	012737	000176	001136	648:	MOV	0SWREG,SWR	;;POINT TO SOFTWARE SWR
(2)	002126	012737	000174	001140		MOV	0DISPREG,DISPLAY	;;POINT TO SOFTWARE DISPLAY REG
(2)	002134	012716	002142			MOV	0658,(SP)	;;REPLACE OLD PC WITH NEW
(2)	002140	000002				RTI		;;RESTORE PC AND PSW
(2)	002142	012637	000004		658:	MOV	(SP)+,06ERRVEC	;;RESTORE ERROR VECTOR
(1)								
(1)	002146				SARG11			
(2)	002146	005037	001202			CLR	0PASS	;;CLEAR PASS COUNT
(2)	002152	132737	000200	001215		BITB	0APTSIZE,SENVH	;;TEST USER SIZE UNDER APT
(2)	002160	001403				BEO	648	;;YES,USE NON-APT SWITCH
(2)	002162	012737	001216	001136		MOV	0SSWREG,SWR	;;NO,USE APT SWITCH REGISTER
(2)	002170				648:			
187	002170	000005			R0EG21	RESET		
188	002172	012702	000232			MOV	0232,R2	;;LOAD R2
189	002176	012701	000230			MOV	0230,R1	;;LOAD R1
190	002202	010221			58:	MOV	R2,(R1)+	;;LOAD .+2
191	002204	005021				CLR	(R1)+	;;LOAD HALT
192	002206	010102				MOV	R1,R2	;;LOAD R2
193	002210	005722				TST	(R2)+	;;BUMP R2
194	002212	020227	001002			CMP	R2,01002	;;TEST FOR LAST
195	002216	001371				BNE	58	;;BR UNTIL DONE
196	002220	005700				TST	R0	;;TEST R0
197	002222	001402				BEO	28	;;BR IF CLEARED
198	002224	000137	002420			JMP	48	;;INHIBIT TYP0UT
199	002230	005737	000042		28:	TST	0042	;;TEST ACT-11 OR DDP
200	002234	001402				BEO	38	;;BR IF CLEARED
201	002236	000137	002420			JMP	48	;;INHIBIT TYP0UT
202	002242				38:			
(1)	002242	104400	002250			TYPE	,658	;;TYPE ASCIZ STRING
(1)	002246	000415				BR	648	;;GET OVER THE ASCIZ
(1)					;;658:	.ASCIZ	<15><12>/AR-11	DIAGNOSTIC TEST I/
(1)	002302				648:			
203	002302	104400	002310			TYPE	,678	;;TYPE ASCIZ STRING
(1)	002306	000414				BR	668	;;GET OVER THE ASCIZ
(1)					;;678:	.ASCIZ	<15><12>/MAINDEC-11-DZARA-B/<15><12>	
(1)	002340				668:			
204	002340	013746	001556			MOV	NMBEXT,-(SP)	;;PUSH ON STACK
205	002344	104402				TYP0S		;;TYPE OCTAL
206	002346	000002				.WORD	2	
207	002350	104400	002356			TYPE	,698	;;TYPE ASCIZ STRING
(1)	002354	000421				BR	688	;;GET OVER THE ASCIZ
(1)					;;698:	.ASCIZ	/(0) ADDITIONAL AR11'S CONNECTED/<15><12>	
(1)	002420				688:			
208								
209	002420	000240			48:	NOP		
210	002422	012700	001516			MOV	0ADCS,R0	;;LOAD POINTER
211	002426	013720	001532		108:	MOV	ARBADD,(R0)+	
212	002432	022700	001536			CMP	0ADINT,R0	;;TEST FOR END
213	002436	001373				BNE	108	
214	002440	013720	001534		118:	MOV	ARBVCT,(R0)+	;;LOAD VECTOR
215	002444	022700	001532			CMP	0ARBADD,R0	
216	002450	001373				BNE	118	
217	002452	012700	001520			MOV	0ADDBR,R0	
218	002456	012701	000002			MOV	02,R1	
219	002462	060120			128:	ADD	R1,(R0)+	



220	002464	005721		TST	(R1)←
221	002466	022701	000020	CMF	020,R1
222	002472	001373		BNE	128
223	002474	005720		TST	(R0)←
224	002476	012701	000002	MOV	02,R1
225	002502	060120	138:	ADD	R1,(R0)←
226	002504	005721		TST	(R1)←
227	002506	022701	000014	CMF	010,R1
228	002512	001373		BNE	138

```

230                                     ;|.....
(3)                                     ;|TEST 1          TEST EACH BUS ADDRESS FOR TIMEOUT
(3)                                     ;|.....
(2) 002514 000004                       TST11  SCOPE
(1) 002516 012737 000100 001164         MOV      0100,STIMES      ;|DO 100 ITERATIONS
231                                     ;|
232 002524 012737 002574 000004         MOV      018,004        ;|LOAD BUS RETURN
233 002532 013737 001516 001126         MOV      ADCS,SDDAT    ;|LOAD WITH BUS ADDRESS TO BE TESTED
234                                     ;|
235 002540 005777 176362                 281     TST      0SDDAT    ;|TEST ADDRESS
236 002544 023737 001126 001534         CMP      SDDAT,CSC     ;|TEST FOR LAST
237 002552 001404                       BEQ      38             ;|BR IF DONE
238 002554 062737 000002 001126         ADD      02,SDDAT      ;|MAKE NEXT ADDRESS
239 002562 000766                       BR       28            ;|TRY MORE
240 002564 012737 000006 000004 381     MOV      06,004        ;|
241 002572 000407                       BR       TST2         ;|BR TO NEXT TEST
242                                     ;|
243 002574 022626                 181     CMP      (SP)+,(SP)+   ;|CLEAN STACK
244 002576 012737 000006 000004         MOV      06,004        ;|RESET 4
245 002604 104015                       ERROR   15             ;|DEVICE BUS ERROR
246 002606 000137 013166                 JMP      BYPASS        ;|DONT TEST ANY MORE
247

```



```
249
250
(3)
(3)
(2) 002612 000004
251 002614 012737 000000 001124
252 002622 013777 001124 176674
253 002630 017737 176670 001126
254 002636 023737 001124 001126
255 002644 001401
256 002646 104004
257
258
(3)
(3)
(2) 002650 000004
259 002652 012737 000377 001124
260 002660 013777 001124 176636
261 002666 017737 176632 001126
262 002674 023737 001124 001126
263 002702 001401
264 002704 104004
265
266
(3)
(3)
(2) 002706 000004
267 002710 012737 000125 001124
268 002716 013777 001124 176600
269 002724 017737 176574 001126
270 002732 023737 001124 001126
271 002740 001401
272 002742 104004
273
274
(3)
(3)
(2) 002744 000004
275 002746 012737 000252 001124
276 002754 013777 001124 176542
277 002762 017737 176536 001126
278 002770 023737 001124 001126
279 002776 001401
280 003000 104004
281
```

```

//*****
//TEST 2 TEST THAT THE PRESET BUFFER CAN HOLD 00
//*****
TST2: SCOPE
MOV 00,SGDDAT ILOAD EXPECTED
MOV SGDDAT,PCSB ILOAD PRESET BUFFER
MOV PCSB,SBDDAT IREAD REG.
CMP SGDDAT,SBDDAT ICOMPARE
BEQ TST3 IIBR IF EQUAL
ERROR 4 IERROR, COUNTER PRESET FAILED TO CLEAR

//*****
//TEST 3 TEST THE COUNTER PRESET BUFFER CAN HOLD 0377
//*****
TST3: SCOPE
MOV 0377,SGDDAT ILOAD EXPECTED
MOV SGDDAT,PCSB ILOAD PRESET BUFFER
MOV PCSB,SBDDAT IREAD REG.
CMP SGDDAT,SBDDAT ICOMPARE
BEQ TST4 IIBR IF EQUAL
ERROR 4 IERROR, COUNTER PRESET FAILED TO LOAD

//*****
//TEST 4 TEST THAT PRESET BUFFER CAN HOLD 0125
//*****
TST4: SCOPE
MOV 0125,SGDDAT ILOAD EXPECTED
MOV SGDDAT,PCSB ILOAD PRESET BUFFER
MOV PCSB,SBDDAT IREAD REG.
CMP SGDDAT,SBDDAT ICOMPARE
BEQ TST5 IIBR IF EQUAL
ERROR 4 IERROR, COUNTER PRESET FAILED TO LOAD

//*****
//TEST 5 TEST THAT PRESET BUFFER CAN HOLD 0252
//*****
TST5: SCOPE
MOV 0252,SGDDAT ILOAD EXPECTED
MOV SGDDAT,PCSB ILOAD PRESET
MOV PCSB,SBDDAT IREAD REG.
CMP SGDDAT,SBDDAT ICOMPARE
BEQ TST6 IIBR IF EQUAL
ERROR 4 IERROR, COUNTER PRESET FAILED TO LOAD
```

```
283 //*****  
(3) ;*TEST 6 TEST THAT PRESET BUFFER CAN HOLD A COUNT PATTERN  
(3) //*****  
(2) 003002 000004 TST6: SCOPE  
(1) 003004 012737 000100 001164 MOV 0100,STIMES ;100 100 ITERATIONS  
284 003012 012737 003024 001110 MOV 019,SLPERR ;LOAD ERROR SCOPE RETURN  
285 003020 005037 001124 CLR SDDAT ;CLEAR PATTERN  
286 003024 013777 001124 176472 18: MOV SDDAT,0CSB ;LOAD REG.  
287 003032 017737 176466 001126 MOV 0CSB,SDDAT ;READ REG.  
288 003040 023737 001124 001126 CMP SDDAT,SDDAT ;COMPARE  
289 003046 001401 BEQ 28 ;JBR IF EQUAL  
290 003050 104004 ERROR 4 ;PRESET BUFFER FAILED TO HOLD A COUNT PATTERN  
291  
292 003052 105237 001124 28: INCB SDDAT ;UPDATE PATTERN  
293 003056 001362 BNE 18 ;JBR UNTIL DONE  
294  
295 //*****  
(3) ;*TEST 7 TEST INIT TO CLEAR COUNT PRESET BUFFER WHEN IT IS =-1  
(3) //*****  
(2) 003060 000004 TST7: SCOPE  
(1) 003062 012737 000020 001164 MOV 020,STIMES ;100 20 ITERATIONS  
296 003070 012737 000000 001124 MOV 00,SDDAT ;LOAD EXPECTED  
297 003076 012777 177777 176420 MOV 0-1,0CSB  
298 003104 000005 RESET  
299 003106 017737 176412 001126 MOV 0CSB,SDDAT ;READ REG.  
300 003114 001401 BEQ TST10 ;JBR IF EQUAL  
301 003116 104004 ERROR 4 ;ERROR, INIT FAILED TO CLEAR CSB  
302  
303 //*****  
(3) ;*TEST 10 TEST THAT THE COUNTER CAN HOLD 00  
(3) //*****  
(2) 003120 000004 TST10: SCOPE  
304 003122 012737 000000 001124 MOV 00,SDDAT ;LOAD EXPECTED  
305 003130 013777 001124 176366 MOV SDDAT,0CSB ;LOAD PRESET BUFFER  
306 003136 017737 176372 001126 MOV 0CSB,SDDAT ;READ COUNTER  
307 003144 023737 001124 001126 CMP SDDAT,SDDAT ;COMPARE  
308 003152 001401 BEQ TST11 ;JBR IF EQUAL  
309 003154 104006 ERROR 6 ;ERROR, COUNTER FAILED TO CLEAR  
310  
311 //*****  
(3) ;*TEST 11 TEST THE COUNTER CAN HOLD 0377  
(3) //*****  
(2) 003156 000004 TST11: SCOPE  
312 003160 012737 000377 001124 MOV 0377,SDDAT ;LOAD EXPECTED  
313 003166 013777 001124 176330 MOV SDDAT,0CSB ;LOAD PRESET BUFFER  
314 003174 017737 176334 001126 MOV 0CSB,SDDAT ;READ COUNTER  
315 003202 023737 001124 001126 CMP SDDAT,SDDAT ;COMPARE  
316 003210 001401 BEQ TST12 ;JBR IF EQUAL  
317 003212 104006 ERROR 6 ;ERROR, COUNTER FAILED TO LOAD  
318
```



```
320
321
(3)
(3)
(2) 003214 000004
322 003216 012737 000125 001124
323 003224 013777 001124 176272
324 003232 017737 176276 001126
325 003240 023737 001124 001126
326 003246 001401
327 003250 104006
328
329
(3)
(3)
(2) 003252 000004
330 003254 012737 000252 001124
331 003262 013777 001124 176234
332 003270 017737 176240 001126
333 003276 023737 001124 001126
334 003304 001401
335 003306 104006
336
337
(3)
(3)
(2) 003310 000004
(1) 003312 012737 000100 001164
338 003320 012737 003332 001110
339 003326 005037 001124
340 003332 013777 001124 176164 101
341 003340 017737 176170 001126
342 003346 023737 001124 001126
343 003354 001401
344 003356 104006
345
346 003360 105237 001124 281
347 003364 001362
348
349
(3)
(3)
(2) 003366 000004
(1) 003370 012737 000020 001164
350 003376 012737 000000 001124
351 003404 012777 177777 176112
352 003412 000009
353 003414 017737 176114 001126
354 003422 001401
355 003424 104006
356

//*****
//TEST 12 TEST THAT COUNTER CAN HOLD 0125
//*****
TST12: SCOPE
MOV 0125,SGDDAT ILOAD EXPECTED
MOV SGDDAT,PCSB ILOAD PRESET BUFFER
MOV PCSC,SBDDAT IREAD COUNTER
CMP SBDDAT,SBDDAT ICOMPARE
BEG TST13 IIBR IF EQUAL
ERROR 6 IERROR, COUNTER FAILED TO LOAD

//*****
//TEST 13 TEST THAT COUNTER CAN HOLD 0252
//*****
TST13: SCOPE
MOV 0252,SGDDAT ILOAD EXPECTED
MOV SGDDAT,PCSB ILOAD PRESET
MOV PCSC,SBDDAT IREAD COUNTER
CMP SGDDAT,SBDDAT ICOMPARE
BEG TST14 IIBR IF EQUAL
ERROR 6 IERROR, COUNTER FAILED TO LOAD

//*****
//TEST 14 TEST THAT COUNTER CAN HOLD A COUNT PATTERN
//*****
TST14: SCOPE
MOV 0100,STIMES IIBR 100 ITERATIONS
MOV 010,SLPERR ILOAD ERROR SCOPE RETURN
CLR SBDDAT ICLEAR PATTERN
MOV SBDDAT,PCSB ILOAD PRESET
MOV PCSC,SBDDAT IREAD COUNTER
CMP SBDDAT,SBDDAT ICOMPARE
BEG 28 IIBR IF EQUAL
ERROR 6 ICOUNTER FAILED TO HOLD A COUNT PATTERN
28: INCB SBDDAT IUPDATE PATTERN
BNE 18 IIBR UNTIL DONE

//*****
//TEST 15 TEST INIT TO CLEAR COUNTER WHEN IT IS =1
//*****
TST15: SCOPE
MOV 020,STIMES IIBR 20 ITERATIONS
MOV 00,SBDDAT ILOAD EXPECTED
MOV 0-1,PCSB
RESET
MOV PCSC,SBDDAT IREAD COUNTER
BEG TST16 IIBR IF EQUAL
ERROR 6 IERROR, INIT FAILED TO CLEAR COUNTER
```



```
358
359
(3)
(3)
(2) 003426 000004
360 003430 012737 000021 001124
361 003436 013777 001124 176056
362 003444 017737 176052 001126
363 003452 023737 001124 001126
364 003460 001401
365 003462 104003
366
367
(3)
(3)
(2) 003464 000004
368 003466 012737 000022 001124
369 003474 013777 001124 176020
370 003502 017737 176014 001126
371 003510 023737 001124 001126
372 003516 001401
373 003520 104003
374
375
(3)
(3)
(2) 003522 000004
376 003524 012737 000024 001124
377 003532 013777 001124 175762
378 003540 017737 175756 001126
379 003546 023737 001124 001126
380 003554 001401
381 003556 104003
382
383
(3)
(3)
(2) 003560 000004
384 003562 012737 000030 001124
385 003570 013777 001124 175724
386 003576 017737 175720 001126
387 003604 023737 001124 001126
388 003612 001401
389 003614 104003
390

//*****
//TEST 16 TEST ENABLE COUNTER (BIT 0) CAN BE SET
//*****
TST16: SCOPE
MOV 00174|0170,SGDDAT ILOAD EXPECTED
MOV SGDDAT,PCSR
MOV PCSR,SGDDAT IREAD REG.
CMP SGDDAT,SGDDAT ICOMPARE
BEQ TST17 IIFR IF EQUAL
ERROR 3 IERROR COUNTER ENABLE FAILED TO SET

//*****
//TEST 17 TEST RATE SELECT (BIT 1) MAY BE SET
//*****
TST17: SCOPE
MOV 00174|0171,SGDDAT ILOAD EXPECTED
MOV SGDDAT,PCSR
MOV PCSR,SGDDAT IREAD REG.
CMP SGDDAT,SGDDAT ICOMPARE
BEQ TST20 IIFR IF EQUAL
ERROR 3 IERROR, RATE BIT 1 FAILED TO SET

//*****
//TEST 20 TEST THAT RATE SELECT (BIT 2) MAY BE SET
//*****
TST20: SCOPE
MOV 00174|0172,SGDDAT ILOAD EXPECTED
MOV SGDDAT,PCSR
MOV PCSR,SGDDAT IREAD REG.
CMP SGDDAT,SGDDAT ICOMPARE
BEQ TST21 IIFR IF EQUAL
ERROR 3 IERROR, RATE BIT 2 FAILED TO SET

//*****
//TEST 21 TEST THAT RATE SELECT (BIT 3) MAY BE SET
//*****
TST21: SCOPE
MOV 00174|0173,SGDDAT ILOAD EXPECTED
MOV SGDDAT,PCSR
MOV PCSR,SGDDAT IREAD REG.
CMP SGDDAT,SGDDAT ICOMPARE
BEQ TST22 IIFR IF EQUAL
ERROR 3 IERROR, RATE BIT 3 FAILED TO SET
```



```
392 (3) (3) (2) 003616 000004 000120 001124
393 003620 012737 000120 001124
394 003626 013777 001124 175666
395 003634 017737 175662 001126
396 003642 023737 001124 001126
397 003650 001401
398 003652 104003
399
400 (3) (3) (2) 003654 000004 000420 001124
401 003656 012737 000420 001124
402 003664 013777 001124 175630
403 003672 017737 175620 001126
404 003700 023737 001124 001126
405 003706 001401
406 003710 104003
407
408 (3) (3) (2) 003712 000004 040020 001124
409 003714 012737 040020 001124
410 003722 013777 001124 175572
411 003730 017737 175566 001126
412 003736 023737 001124 001126
413 003744 001401
414 003746 104003
415
416 (3) (3) (2) 003750 000004 000220 001124
417 003752 012737 000220 001124
418 003760 013777 001124 175534
419 003766 017737 175530 001126
420 003774 023737 001124 001126
421 004002 001401
422 004004 104003
423 (3) (3) (2) 004006 000004 100020 001124
424 004010 012737 100020 001124
425 004016 013777 001124 175476
426 004024 017737 175472 001126
427 004032 023737 001124 001126
428 004040 001401
429 004042 104003
```

```
*****
TEST 22 TEST CLOCK INTERRUPT ENABLE (BIT 6) CAN BE SET
*****
```

```
TST22: SCOPE
MOV @BIT6:BIT4,SGDDAT ILOAD EXPECTED
MOV SGDDAT,PCSR ILOAD REG.
MOV @CSR,SBDDAT IREAD REG.
CMP SGDDAT,SBDDAT ICOMPARE
BEQ TST23 IIBR IF EQUAL
ERROR 3 IERROR, CLOCK INTERRUPT ENABLE FAILED TO SET
```

```
*****
TEST 23 TEST MODE (BIT 8) CAN BE SET
*****
```

```
TST23: SCOPE
MOV @BIT8:BIT4,SGDDAT ILOAD EXPECTED
MOV SGDDAT,PCSR ILOAD REG.
MOV @CSR,SBDDAT IREAD REG.
CMP SGDDAT,SBDDAT ICOMPARE
BEQ TST24 IIBR IF EQUAL
ERROR 3 IERROR, CSR NOT = 420
```

```
*****
TEST 24 TEST EXT INTERRUPT ENABLE (BIT 14) CAN BE SET
*****
```

```
TST24: SCOPE
MOV @BIT14:BIT4,SGDDAT ILOAD EXPECTED
MOV SGDDAT,PCSR ILOAD REG.
MOV @CSR,SBDDAT IREAD REG.
CMP SGDDAT,SBDDAT ICOMPARE
BEQ TST25 IIBR IF EQUAL
ERROR 3 IERROR, EXT INTERRUPT ENABLE FAILED TO SET
```

```
*****
TEST 25 TEST THAT CLK DONE (BIT 7) CAN BE SET
*****
```

```
TST25: SCOPE
MOV @BIT7:BIT4,SGDDAT ILOAD EXPECTED
MOV SGDDAT,PCSR ILOAD REG.
MOV @CSR,SBDDAT IREAD REG.
CMP SGDDAT,SBDDAT ICOMPARE
BEQ TST26 IIBR IF EQUAL
ERROR 3 ICLOCK DONE FAILED TO SET
```

```
*****
TEST 26 TEST THAT CLK EXT INPUT (BIT 15) CAN BE SET
*****
```

```
TST26: SCOPE
MOV @BIT15:BIT4,SGDDAT ILOAD EXPECTED
MOV SGDDAT,PCSR ILOAD REG.
MOV @CSR,SBDDAT IREAD REG.
CMP SGDDAT,SBDDAT ICOMPARE
BEQ TST27 IIBR IF EQUAL
ERROR 3 ICLOCK EXT INPUT FAILED TO SET
```



```
431 (3) (3) (2) 004044 000004
(1) 004046 012737 000010 001164
432 004054 005077 175442
433 004060 012737 000020 001124
434 004066 005037 001566
435 004072 017737 175424 001126 191
436 004100 005737 001126
437 004104 100001
438 004106 104003
439 004110 005237 001566 281
440 004114 001366
441 (3) (3) (2) 004116 000004
(1) 004120 012737 000004 001164
442 004126 005077 175370
443 004132 005077 175366
444 004136 012777 000002 175356
445 004144 005037 001566
446 004150 013737 001566 001124 191
447 004156 017737 175352 001126
448 004164 023737 001124 001126
449 004172 001401
450 004174 104007
451 004176 117737 175320 001126 281
452 004204 100005
453 004206 012737 000022 001124
454 004214 104003
455 004216 000411
456 004220 052777 004000 175274 381
457 004226 042777 004000 175266
458 004234 105237 001566
459 004240 001343
460 (3) (3) (2) 004242 000004
(1) 004244 005077 175252
461 004250 012777 177777 175246
462 004256 012777 000002 175236
463 004264 012737 000222 001124
464 004272 052777 004000 175222
465 004300 042777 004000 175214
466 004306 017737 175210 001126
467 004314 105737 001126
468 004320 100401
469 004322 104003
```

;;\*\*\*\*\*  
;TEST 27 TEST THAT THE EXT FLAG DOES NOT SET FROM THE OUTSIDE SOURCE  
;;\*\*\*\*\*  
TST27: SCOPE  
MOV 010,STIMES ;100 10 ITERATIONS  
CLR 0CSR  
MOV 0BIT4,SGDDAT ;LOAD EXPECTED  
CLR TEMP  
MOV 0CSR,SGDDAT ;READ REG  
TST SGDDAT ;TEST BIT15  
BPL 28 ;BR IF CLEARED  
ERROR 3 ;ERROR EXT FLG SET IN ERROR  
INC TEMP  
BNE 18  
;;\*\*\*\*\*  
;TEST 30 MAINT. COUNT THE COUNTER REGISTER AT RATE 1MHZ  
;;\*\*\*\*\*  
TST30: SCOPE  
MOV 04,STIMES ;100 4 ITERATIONS  
CLR 0CSR  
CLR 0CSB  
MOV 02,0CSR ;SET UP 1MHZ RATE  
CLR TEMP ;CLEAR PATTERN  
MOV TEMP,SGDDAT ;LOAD EXPECTED  
MOV 0CSB,SGDDAT ;READ COUNTER  
CMP SGDDAT,SGDDAT ;COMP  
BEQ 28 ;BR IF EQUAL  
ERROR 7 ;ERROR, CLOCK COUNTER  
;BUFFER COUNTED IN ERROR  
MOV 0CSR,SGDDAT ;READ REG  
BPL 38 ;BR IF CLEAR  
MOV 022,SGDDAT ;LOAD EXPECTED STATUS  
ERROR 3 ;CLK DONE SET IN ERROR  
BR TST31 ;BR TO SCOPE  
DIS 0BIT11,0CSR ;MAINT COUNT  
BIC 0BIT11,0CSR  
INCB TEMP  
BNE 18 ;BRANCH IF NOT FULL COUNT  
;;\*\*\*\*\*  
;TEST 31 TEST THAT OVERFLOW SET CLK DONE (BIT 7)  
;;\*\*\*\*\*  
TST31: SCOPE  
CLR 0CSR ;CLEAR STATUS  
MOV 0-1,0CSB ;LOAD PRESET  
MOV 0BIT1,0CSR ;LOAD 1 MHZ. RATE  
MOV 0BIT7!BIT4!BIT1,SGDDAT ;LOAD EXPECTED  
DIS 0BIT11,0CSR ;MAINT COUNT  
BIC 0BIT11,0CSR ;CLEAR BIT  
MOV 0CSR,SGDDAT ;READ REG.  
TSTB SGDDAT ;TEST BIT 7  
BMI TST32 ;BR IF MINUS  
ERROR 3 ;ERROR, OVERFLOW FAILED TO SET BIT 7



```
473                                     ;);*****  
(3) ;)TEST 32      MAINT. COUNT THE COUNTER REGISTER AT RATE 0100KHZ  
(3) ;);*****  
(2) 004324 000004 T8732: SCOPE  
(1) 004326 012737 000004 001164      MOV      04,STIMES      ;)DO 4 ITERATIONS  
474 004334 005077 175162              CLR      0CSR          ;)CLEAR CLOCK STATUS  
475 004340 005077 175160              CLR      0C80          ;)CLEAR PRESET  
476 004344 005037 001124              CLR      8GDDAT  
477 004350 012777 000004 175144      MOV      04,0CSR      ;)LOAD STATUS, 100KHZ RATE  
478 004356 005037 001124              CLR      8GDDAT  
479 004362 017737 175146 001126 18:  MOV      0C8C,8GDDAT  ;)READ COUNTER  
480 004370 023737 001124 001126      CMP      8GDDAT,8GDDAT ;)COMPARE  
481 004376 001401              BEQ      28            ;)BR IF EQUAL  
482 004400 104007              ERROR     7            ;)ERROR, CLOCK COUNTER BUFFER  
483                                     ;)COUNTED IN ERROR, FAULT IS PROBABLY IN THE  
484                                     ;)CLOCK UP COUNT OR RATE SELECTION LOGIC  
485 004402 012737 000012 001562 28:  MOV      010.,COUNT  ;)LOAD COUNT  
486 004410 052777 004000 175104 38:  BIS      0BIT11,0CSR  ;)MAINT COUNT  
487 004416 042777 004000 175076      BIC      0BIT11,0CSR  
488 004424 005337 001562              DEC      COUNT        ;)DONE  
489 004430 001367              BNE      38            ;)BR IF NOT  
490 004432 105237 001124              INCB    8GDDAT        ;)INCREMENT EXPECTED VALUE  
491 004436 001351              BNE      18  
492  
493                                     ;);*****  
(3) ;)TEST 33      MAINT. COUNT THE COUNTER REGISTER AT RATE 010KHZ  
(3) ;);*****  
(2) 004440 000004 T8733: SCOPE  
(1) 004442 012737 000004 001164      MOV      04,STIMES      ;)DO 4 ITERATIONS  
494 004450 005077 175046              CLR      0CSR          ;)CLEAR CLOCK STATUS  
495 004454 005077 175044              CLR      0C80          ;)CLEAR PRESET  
496 004460 012777 000006 175034      MOV      06,0CSR      ;)LOAD STATUS, 10KHZ RATE  
497 004466 005037 001124              CLR      8GDDAT  
498 004472 017737 175036 001126 18:  MOV      0C8C,8GDDAT  ;)READ COUNTER  
499 004500 023737 001124 001126      CMP      8GDDAT,8GDDAT ;)COMPARE  
500 004506 001401              BEQ      28            ;)BR IF EQUAL  
501 004510 104007              ERROR     7            ;)ERROR, CLOCK COUNTER BUFFER  
502                                     ;)COUNTED IN ERROR, FAULT IS PROBABLY IN THE  
503                                     ;)CLOCK UP COUNT OR RATE SELECTION LOGIC  
504 004512 012737 000144 001562 28:  MOV      0100.,COUNT ;)LOAD COUNT  
505 004520 052777 004000 174774 38:  BIS      0BIT11,0CSR  ;)MAINT COUNT  
506 004526 042777 004000 174766      BIC      0BIT11,0CSR  
507 004534 005337 001562              DEC      COUNT        ;)DONE  
508 004540 001367              BNE      38            ;)BR IF NOT  
509 004542 105237 001124              INCB    8GDDAT        ;)UPDATE PATTERN  
510 004546 001351              BNE      18
```

```
512  
513  
(3)  
(3)  
(2) 004550 000004  
(1) 004552 012737 000004 001164  
514 004560 005077 174736  
515 004564 005077 174734  
516 004570 012737 000001 001124  
517 004576 012777 000010 174716  
518 004604 012737 001750 001562  
519 004612 052777 004000 174702 181  
520 004620 042777 004000 174674  
521 004626 005337 001562  
522 004632 001367  
523 004634 017737 174674 001126  
524 004642 023737 001124 001126  
525 004650 001401  
526 004652 104007  
527  
528  
529  
530  
(3)  
(3)  
(2) 004654 000004  
(1) 004656 012737 000004 001164  
531 004664 005077 174632  
532 004670 005077 174630  
533 004674 012737 000001 001124  
534 004702 012777 000012 174612  
535 004710 012737 023420 001562  
536 004716 052777 004000 174576 181  
537 004724 042777 004000 174570  
538 004732 005337 001562  
539 004736 001367  
540 004740 017737 174570 001126  
541 004746 023737 001124 001126  
542 004754 001401  
543 004756 104007  
544  
545  
546
```

```
*****  
TEST 34 MAINT. COUNT THE COUNTER REGISTER AT RATE #10KHZ  
*****  
TST34: SCOPE  
MOV 04,STIMES ;DO 4 ITERATIONS  
CLR 0CSR ;CLEAR CLOCK STATUS  
CLR 0CS0 ;CLEAR PRESET  
MOV 01,SGDDAT ;LOAD EXPECTED  
MOV 010,0CSR ;LOAD STATUS, 1 KHZ RATE  
MOV 01000.,COUNT ;SET UP A COUNTER  
BIS 0BIT11,0CSR ;GENERATE MAINT COUNT  
BIC 0BIT11,0CSR  
DEC COUNT  
BNE 18 ;BR  
MOV 0CSC,SBDDAT ;READ COUNTER  
CMP SGDDAT,SBDDAT ;COMPARE  
BEG TST35 ;BR IF EQUAL  
ERROR 7 ;ERROR, CLOCK COUNTER BUFFER  
 ;COUNTED IN ERROR, FAULT IS PROBABLY IN THE  
 ;CLOCK UP COUNT OR RATE SELECTION LOGIC
```

```
530  
(3)  
(3)  
(2) 004654 000004  
(1) 004656 012737 000004 001164  
531 004664 005077 174632  
532 004670 005077 174630  
533 004674 012737 000001 001124  
534 004702 012777 000012 174612  
535 004710 012737 023420 001562  
536 004716 052777 004000 174576 181  
537 004724 042777 004000 174570  
538 004732 005337 001562  
539 004736 001367  
540 004740 017737 174570 001126  
541 004746 023737 001124 001126  
542 004754 001401  
543 004756 104007  
544  
545  
546
```

```
*****  
TEST 35 MAINT. COUNT THE COUNTER REGISTER AT RATE #100KHZ  
*****  
TST35: SCOPE  
MOV 04,STIMES ;DO 4 ITERATIONS  
CLR 0CSR ;CLEAR CLOCK STATUS, 100KHZ RATE  
CLR 0CS0 ;CLEAR PRESET  
MOV 01,SGDDAT ;LOAD EXPECTED  
MOV 012,0CSR ;LOAD STATUS, 100 KHZ  
MOV 010000.,COUNT ;SET UP A COUNTER  
BIS 0BIT11,0CSR ;GENERATE MAINT COUNT  
BIC 0BIT11,0CSR  
DEC COUNT  
BNE 18 ;BR  
MOV 0CSC,SBDDAT ;READ COUNTER  
CMP SGDDAT,SBDDAT ;COMPARE  
BEG TST36 ;BR IF EQUAL  
ERROR 7 ;ERROR, CLOCK COUNTER BUFFER  
 ;COUNTED IN ERROR, FAULT IS PROBABLY IN THE  
 ;CLOCK UP COUNT OR RATE SELECTION LOGIC
```



```
548 //*****  
(3) /*TEST 36 TEST THAT RESET CLEARS RATE SELECT AND MODE BITS  
(3) //*****  
(2) 004760 000004 TST36: SCOPE  
(1) 004762 012737 000010 001164 MOV #10,STIMES //DO 10 ITERATIONS  
549 004770 012777 000416 174524 MOV #BIT0,BIT3,BIT2,BIT1,PCSR //SET MODE BITS  
550 004776 012737 000020 001124 MOV #20,SGDDAT  
551 005004 000005 RESET  
552 005006 017737 174510 001126 MOV PCSR,SBDDAT //READ REG.  
553 005014 023737 001124 001126 CMP SGDDAT,SBDDAT //COMPARE  
554 005022 001401 BEQ TST37 //BR IF EQUAL  
555 005024 104003 ERROR 3 //ERROR, RESET FAILED TO CLEAR RATE OR MODE BITS  
556  
557 //*****  
(3) /*TEST 37 TEST THAT RESET CLEARS CLK INTERRUPT ENABLE  
(3) //*****  
(2) 005026 000004 TST37: SCOPE  
(1) 005030 012737 000010 001164 MOV #10,STIMES //DO 10 ITERATIONS  
558 005036 012777 040100 174456 MOV #BIT6,BIT14,PCSR //SET CLK INT ENABLE  
559 005044 012737 000020 001124 MOV #20,SGDDAT  
560 005052 000005 RESET  
561 005054 017737 174442 001126 MOV PCSR,SBDDAT  
562 005062 023737 001124 001126 CMP SGDDAT,SBDDAT  
563 005070 001401 BEQ TST40 //BR IF EQUAL  
564 005072 104003 ERROR 3 //ERROR, RESET FAILED TO CLEAR CLK INT ENABLE  
565  
566 //*****  
(3) /*TEST 40 TEST THAT RESET CLEARS CLK FLAGS  
(3) //*****  
(2) 005074 000004 TST40: SCOPE  
(1) 005076 012737 000010 001164 MOV #10,STIMES //DO 10 ITERATIONS  
567 005104 012777 100200 174410 MOV #BIT7,BIT15,PCSR //SET CLK FLAGS  
568 005112 012737 000020 001124 MOV #20,SGDDAT  
569 005120 000005 RESET  
570 005122 017737 174374 001126 MOV PCSR,SBDDAT  
571 005130 023737 001124 001126 CMP SGDDAT,SBDDAT  
572 005136 001401 BEQ TST41 //BR IF EQUAL  
573 005140 104003 ERROR 3 //ERROR, RESET FAILED TO CLEAR CLK FLAGS  
574  
575 //*****  
(3) /*TEST 41 TEST THAT RESET CLEARS COUNTER ENABLE  
(3) //*****  
(2) 005142 000004 TST41: SCOPE  
(1) 005144 012737 000010 001164 MOV #10,STIMES //DO 10 ITERATIONS  
576 005152 012777 000000 174344 MOV #0,PCSR //CLEAR COUNTER  
577 005160 012777 000001 174334 MOV #BIT0,PCSR //LOAD COUNTER ENABLE  
578 005166 012737 000020 001126 MOV #20,SBDDAT  
579 005174 000005 RESET  
580 005176 017737 174320 001126 MOV PCSR,SBDDAT  
581 005204 023737 001124 001126 CMP SGDDAT,SBDDAT  
582 005212 001401 BEQ TST42 //BR IF EQUAL  
583 005214 104003 ERROR 3 //ERROR, RESET FAILED TO CLEAR COUNTER ENABLE  
584  
585
```

```
587  
(3) //*****  
(3) //TEST 42 TEST CLOCK TO COUNT UP AT 1 MHZ  
(2) //*****  
(1) 005216 000004 TST42: SCOPE  
588 005220 012737 000100 001164 MOV #100,STIMES //DO 100 ITERATIONS  
589 005226 012737 000003 014056 MOV #3,RATE //SELECT MODE 0, 1MHZ., GO  
590 005234 012737 000222 001124 MOV @BIT7:BIT4:BIT1,SGDDAT  
591 005242 004737 013376 JSR PC,UPCNT  
592 005246 017737 174250 001126 MOV @CSR,SDDAT //READ REG  
593 005254 023737 001124 001126 CMP SGDDAT,SDDAT //COMPARE  
594 005262 001401 BEQ 18 //BR IF EQUAL  
595 005264 104003 ERROR 3 //ERROR, 1MHZ RATE FAILED TO SET DONE  
596 005266 012737 000000 001124 18: MOV #0,SGDDAT  
597 005274 017737 174234 001126 MOV @CSR,SDDAT  
598 005302 023737 001124 001126 CMP SGDDAT,SDDAT  
599 005310 001401 BEQ TST43 //BR IF EQUAL  
600 005312 104006 ERROR 6 //IN MODE 0 CLOCK COUNTER WAS  
//LOADED ON CLK OVERFLOW
```

```
601  
602 //*****  
(3) //TEST 43 TEST CLOCK TO COUNT UP AT 100KHZ  
(3) //*****  
(2) 005314 000004 TST43: SCOPE  
(1) 005316 012737 000100 001164 MOV #100,STIMES //DO 100 ITERATIONS  
603 005324 012737 000003 014056 MOV #5,RATE //SELECT MODE 0, 100KHZ., GO  
604 005332 012737 000224 001124 MOV @BIT7:BIT4:BIT2,SGDDAT  
605 005340 004737 013376 JSR PC,UPCNT  
606 005344 017737 174152 001126 MOV @CSR,SDDAT //READ REG  
607 005352 023737 001124 001126 CMP SGDDAT,SDDAT //COMPARE  
608 005360 001401 BEQ 18 //BR IF EQUAL  
609 005362 104003 ERROR 3 //ERROR, 100KHZ. RATE FAILED TO SET DONE  
610 005364 012737 000000 001124 18: MOV #0,SGDDAT  
611 005372 017737 174136 001126 MOV @CSR,SDDAT  
612 005400 023737 001124 001126 CMP SGDDAT,SDDAT  
613 005406 001401 BEQ TST44 //BR IF EQUAL  
614 005410 104006 ERROR 6 //IN MODE 0 CLOCK COUNTER WAS  
//LOADED ON CLK OVERFLOW
```

```
615  
616 //*****  
(3) //TEST 44 TEST CLOCK TO COUNT UP AT 10 KHZ  
(3) //*****  
(2) 005412 000004 TST44: SCOPE  
(1) 005414 012737 000100 001164 MOV #100,STIMES //DO 100 ITERATIONS  
618 005422 012737 000007 014056 MOV #7,RATE //SELECT MODE 0, 10KHZ., GO  
619 005430 012737 000226 001124 MOV @BIT7:BIT4:BIT2:BIT1,SGDDAT  
620 005436 004737 013376 JSR PC,UPCNT  
621 005442 017737 174054 001126 MOV @CSR,SDDAT //READ REG  
622 005450 023737 001124 001126 CMP SGDDAT,SDDAT //COMPARE  
623 005456 001401 BEQ TST45 //BR IF EQUAL  
624 005460 104007 ERROR 7 //ERROR, 10 KHZ. FAILED TO SET DONE  
625
```



```
627
628
(3)
(3)
(2) 005462 000004
(1) 005464 012737 000100 001164
629 005472 012737 000011 014056
630 005500 012737 000230 001124
631 005506 004737 013376
632 005512 017737 174004 001126
633 005520 023737 001124 001126
634 005526 001401
635 005530 104007
636
637
(3)
(3)
(2) 005532 000004
(1) 005534 012737 000100 001164
638 005542 012737 000013 014056
639 005550 012737 000232 001124
640 005556 004737 013376
641 005562 017737 173734 001126
642 005570 023737 001124 001126
643 005576 001401
644 005600 104007
645
646
(3)
(3)
(2) 005602 000004
(1) 005604 012737 000100 001164
647 005612 012737 000411 014056
648 005620 012737 000631 001124
649 005626 004737 013376
650 005632 017737 173664 001126
651 005640 042777 000001 173654
652 005646 023737 001124 001126
653 005654 001401
654 005656 104007
655
656 005660 012737 000376 001124 101
657 005666 017737 173642 001126
658 005674 023737 001124 001126
659 005702 001401
660 005704 104006
661
662

//*****
//TEST 45 TEST CLOCK TO COUNT UP AT 1KHZ
//*****
TST45: SCOPE
MOV #100,STIMES //DO 100 ITERATIONS
MOV #11,RATE //SELECT MODE 0, 1 KHZ, GO
MOV #BIT7:BIT4:BIT3,SGDDAT
JSR PC,UPCNT
MOV #CSR,SBDDAT //READ REG
CMP SGDDAT,SBDDAT //COMPARE
BEQ TST46 //BR IF EQUAL
ERROR 7 //ERROR, 1KHZ FAILED TO SET DONE

//*****
//TEST 46 TEST CLOCK TO COUNT UP AT 100HZ
//*****
TST46: SCOPE
MOV #100,STIMES //DO 100 ITERATIONS
MOV #13,RATE //SELECT MODE 0, 100 HZ, GO
MOV #BIT7:BIT4:BIT3:BIT1,SGDDAT //LOAD EXPECTED
JSR PC,UPCNT
MOV #CSR,SBDDAT //READ REG.
CMP SGDDAT,SBDDAT //COMPARE
BEQ TST47 //BR IF EQUAL
ERROR 7 //ERROR, 100HZ FAILED TO SET DONE

//*****
//TEST 47 TEST THAT CLOCK ENABLE DOES NOT CLEAR ON DONE (MODE 1) 1 KHZ
//*****
TST47: SCOPE
MOV #100,STIMES //DO 100 ITERATIONS
MOV #411,RATE //SELECT MODE 1, 1KHZ., GO
MOV #631,SGDDAT //LOAD EXP
JSR PC,UPCNT
MOV #CSR,SBDDAT //READ REG
DIB #BIT0,CSR //CLEAR ENABLE
CMP SGDDAT,SBDDAT //COMPARE
BEQ 18 //BR IF EQUAL
ERROR 7 //ERROR, 1 KHZ, FAILED TO OVERFLOW
// AND CONTINUE COUNTING

MOV #376,SGDDAT
MOV #CSR,SBDDAT
CMP SGDDAT,SBDDAT
BEQ TST50 //BR IF EQUAL
ERROR 6 //IN MODE 1, CLOCK COUNTER FAILED
// TO BE RE-LOADED ON CLOCK OVERFLOW
```

```
664  
665  
(3) //*****  
(3) //TEST 50 TEST THAT CLOCK ENABLE DOES NOT CLEAR DONE (MODE 1 100KHZ)  
(2) //*****  
(1) 005706 000004 TST50: SCOPE  
005710 012737 000100 001164 MOV 0100,STIMES //DO 100 ITERATIONS  
666 005716 012737 000413 014056 MOV 0413,RATE //MODE 1 100KHZ GO  
667 005724 012737 000633 001124 MOV 0633,SGDDAT //LOAD EXPECTED  
668 005732 004737 013376 JSR PC,UPCNT  
669 005736 017737 173560 001126 MOV 0CSR,SBDDAT //READ REG  
670 005744 042777 000001 173550 BIC 00170,0CSR //CLEAR ENABLE  
671 005752 023737 001124 001126 CMP SGDDAT,SBDDAT //COMPARE  
672 005760 001401 BEQ 18 //BR IF EQUAL  
673 005762 104007 ERROR 7 //ERROR, 100KHZ FAILED TO  
674 // OVERFLOW AND CONTINUE COUNTING  
675 005764 012737 000376 001124 18: MOV 0376,SGDDAT  
676 005772 017737 173536 001126 MOV 0CSC,SBDDAT  
677 006000 023737 001124 001126 CMP SGDDAT,SBDDAT  
678 006006 001401 BEQ TST51 //BR IF EQUAL  
679 006010 134006 ERROR 6 //IN MODE 1, CLOCK COUNTER FAILED  
680 // TO BE RE-LOADED ON CLOCK OVERFLOW  
681  
682 //*****  
(3) //TEST 51 CLOCK PRE-INTERRUPT SETUP  
(3) //*****  
(2) 006012 000004 TST51: SCOPE  
(1) 006014 012737 000001 001164 MOV 01,STIMES //DO 1 ITERATION  
683 006022 042737 177437 001512 BIC 0177437,CKBRL  
684 006030 001001 BNE .+4  
685 006032 000000 HALT //ERROR, BR LEVEL INDICATED FOR CLOCK WAS 0  
686 006034 022737 000340 001512 CMP 0340,CKBRL  
687 006042 001001 BNE .+4  
688 006044 000000 HALT //ERROR BR LEVEL FOR CLOCK WAS 7  
689  
690 006046 013737 001512 001572 MOV CKBRL,BRLEV1  
691 006054 162737 000040 001572 SUB 040,BRLEV1  
692 006062 013737 001512 001574 MOV CKBRL,BRLEV2  
693 006070 000005 RESET
```



```
695                                     ;;;;;;;;;;;;;;
(3) ;TEST 52 TEST THAT THE CLOCK INTERRUPTS AT LEVEL INDICATED -1
(3) ;;;;;;;;;;;;;;
(2) 006072 000004 TST521 SCOPE
696 006074 012777 006160 173440 MOV 028,0KWIV ILOAD VECTOR
697 006102 005077 173414 CLR 0CSR
698 006106 012777 177777 173410 MOV 0-1,0CSB ILOAD PRESET
699 006114 012777 000102 173400 MOV 0BIT6|0IT1,0CSR ILOAD RATE AND INT ENABLE
700 006122 012700 002000 MOV 02000,R0 ISET UP DELAY
701 006126 013737 001572 177776 MOV BRLEV1,PSW
702 006134 052777 004000 173360 DIS 0BIT11,0CSR ICLOCK
703 006142 042777 004000 173352 BIC 0BIT11,0CSR
704 006150 005300 181 DEC R0
705 006152 001376 BNE 18 IDELAY
706 006154 104005 ERROR 5 IERROR, CLOCK FAILED TO INTERRUPT
707 006156 000401 BR T8T53 IINEXT TEST
708 006160 022626 281 CMP (SP)+,(SP)+
709
710 ;;;;;;;;;;;;;;
(3) ;TEST 53 TEST THAT THE CLOCK DOES NOT INTERRUPT AT LEVEL INDICATED
(3) ;;;;;;;;;;;;;;
(2) 006162 000004 TST531 SCOPE
711 006164 012777 006246 173350 MOV 018,0KWIV ISET UP INTERRUPT RETURN STATUS
712 006172 012777 000000 173344 MOV 00,0KWIVS
713 006200 005077 173316 CLR 0CSR
714 006204 012777 177776 173312 MOV 0-4,0CSB
715 006212 012777 000102 173302 MOV 0BIT6|0IT1,0CSR IENABLE INTERRUPT
716 006220 013737 001574 177776 MOV BRLEV2,PSW
717 006226 052777 000001 173266 DIS 0BIT0,0CSR
718 006234 012700 001000 MOV 01000,R0
719 006240 005300 DEC R0
720 006242 001376 BNE 18 IJBR TO NEXT TEST
721 006244 000403 BR 28 IERROR, INTERRUPT OCCURED IN ERROR
722 006246 104005 181 ERROR 5 IAT LEVEL INDICATED
723 006250 022626 CMP (SP)+,(SP)+ IJBR TO SCOPE
724 006252 000421 BR T8T54
725
726
727 ;SUBTEST, TEST THAT IF PRIORITY IS LOWERED AGAIN
728 ;NO INTERRUPT SHOULD OCCUR
729
730 006254 012777 006302 173260 281 MOV 038,0KWIV IRESET THE VECTOR
731 006262 012700 000010 MOV 010,R0
732 006266 005037 177776 CLR PSW
733 006272 005300 DEC R0
734 006274 001376 BNE 18 IDELAY
735 006276 104005 ERROR 5 IERROR, INIT DONE FAILED TO CLEAR INT REG.
736 006300 000406 BR T8T54 IJBR TO NEXT TEST
737 006302 022626 381 CMP (SP)+,(SP)+
738 006304 013777 001544 173230 MOV KWIVS,0KWIV
739 006312 005077 173226 CLR 0KWIVS
```

```
741 (3) (3) (2) 006316 000004 (1) 006320 012737 000100 001164 742 006326 004537 013676 743 006332 000002 744 006334 000002 745 006336 000001 746 006340 000036 747 006342 104007 748 006344 003401 749 006346 104010 750 (3) (3) (2) 006350 000004 (1) 006352 012737 000100 001164 752 006360 004537 013676 753 006364 000004 754 006366 000002 755 006370 000001 756 006372 000454 757 006374 104007 758 006376 003401 759 006400 104010 760 (3) (3) (2) 006402 000004 (1) 006404 012737 000100 001164 761 006412 004537 013676 762 006416 000006 763 006420 000002 764 006422 000001 765 006424 005670 766 006426 104007 767 006430 003401 768 006432 104010 769 (3) (3) (2) 006434 000004 (1) 006436 012737 000100 001164 770 006444 004537 013676 771 006450 000010 772 006452 000002 773 006454 000001 774 006456 072460 775 006460 104007 776 006462 003401 777 006464 104010
```

```

//*****
//TEST 54 TEST 1MHZ REPEATABILITY
//*****
TST54: SCOPE
MOV #100,STIMES //DO 100 ITERATIONS
JSR RS,REPEAT //TEST REPEATABILITY
2 //CLOCK RATE, 1MHZ
2 //CLOCK DEV.
1 //MIN. COUNT
30. //DELAY
ERROR 7 //ERROR, FAILED TO REACH MIN. COUNT
BLE TST55 //BR IF LESS OR EQUAL
ERROR 10 //ERROR, CLOCK REPEATABILITY >2
//TRY CHANGING I.C. "EQ0" FROM 0640 TO DEC 300
//*****
//TEST 55 TEST 100 KHZ REPEATABILITY
//*****
TST55: SCOPE
MOV #100,STIMES //DO 100 ITERATIONS
JSR RS,REPEAT //TEST REPEATABILITY
4 //CLOCK RATE, 100KHZ
2 //CLOCK DEV.
1 //MIN. COUNT
300. //DELAY
ERROR 7 //ERROR, FAILED TO REACH MIN. COUNT
BLE TST56 //BR IF LESS OR EQUAL
ERROR 10 //ERROR, CLOCK REPEATABILITY >2
//*****
//TEST 56 TEST 10 KHZ REPEATABILITY
//*****
TST56: SCOPE
MOV #100,STIMES //DO 100 ITERATIONS
JSR RS,REPEAT //TEST REPEATABILITY
6 //CLOCK RATE, 10KHZ
2 //CLOCK DEV.
1 //MIN. COUNT
3000. //DELAY
ERROR 7 //ERROR, FAILED TO REACH MIN. COUNT
BLE TST57 //BR IF LESS OR EQUAL
ERROR 10 //ERROR, CLOCK REPEATABILITY >2
//*****
//TEST 57 TEST 1KHZ REPEATABILITY
//*****
TST57: SCOPE
MOV #100,STIMES //DO 100 ITERATIONS
JSR RS,REPEAT //TEST REPEATABILITY
10 //CLOCK RATE, 1KHZ
2 //CLOCK DEV.
1 //MIN. COUNT
30000. //DELAY
ERROR 7 //ERROR, FAILED TO REACH MIN. COUNT
BLE TST60 //BR IF LESS OR EQUAL
ERROR 10 //ERROR, CLOCK REPEATABILITY >2

```



```
779      ))*****  
(3)      )-TEST 60      TEST 100HZ REPEATIBILITY  
(3)      ))*****  
(2) 006466 000004      TST60: SCOPE  
(1) 006470 012737 000100 001164      MOV      0100,STIMES      )100 100 ITERATIONS  
780 006476 004537 013676      JSR      R9,REPEAT      )  
781 006502 000012      )12      )CLOCK RATE, 100HZ  
782 006504 000002      )2      )CLOCK DEV.  
783 006506 000001      )1      )MIN. COUNT  
784 006510 177777      )-1      )DELAY  
785 006512 104007      ERROR 7      )ERROR, FAILED TO REACH MIN. COUNT  
786 006514 063401      BLE     TST61      )1BR IF LESS OR EQUAL  
787 006516 104010      ERROR 10      )ERROR CLOCK REPEATABILITY >2  
788  
789      ))*****  
(3)      )-TEST 61      TEST THAT RESET SETS VC READY BIT  
(3)      ))*****  
(2) 006520 000004      TST61: SCOPE  
(1) 006522 012737 000010 001164      MOV      010,STIMES      )100 10 ITERATIONS  
790 006530 012737 000200 001124      MOV      0017,SGDDAT      )LOAD EXPECTED  
791 006536 000005      RESET  
792 006540 017737 172762 001126      MOV      0VCSTAT,SGDDAT      )READ REGISTER  
793 006546 023737 001124 001126      CMP      SGDDAT,SGDDAT      )COMPARE  
794 006554 001401      BEQ     TST62      )1BR IF SET  
795 006556 104011      ERROR 11      )RESET FAILED TO SET READY  
796  
797      ))*****  
(3)      )-TEST 62      TEST THAT VC MODE BIT 2 CAN BE SET AND CLEARED  
(3)      ))*****  
(2) 006560 000004      TST62: SCOPE  
798 006562 012777 000004 172736      MOV      0BIT2,0VCSTAT      )LOAD DISPLAY STATUS  
799 006570 012737 000200 001124      MOV      0BIT7,0BIT2,SGDDAT      )LOAD EXPECTED  
800 006576 017737 172724 001126      MOV      0VCSTAT,SGDDAT      )READ REG  
801 006604 023737 001124 001126      CMP      SGDDAT,SGDDAT      )COMPARE  
802 006612 001401      BEQ     10      )1BR IF EQUAL  
803 006614 104011      ERROR 11      )ERROR, VC STATUS NOT = 204  
804 006616 005077 172704 181      CLR      0VCSTAT      )CLEAR STATUS  
805 006622 012737 000200 001124      MOV      0BIT7,SGDDAT      )LOAD EXPECTED  
806 006630 017737 172672 001126      MOV      0VCSTAT,SGDDAT      )READ REG  
807 006636 023737 001124 001126      CMP      SGDDAT,SGDDAT      )COMPARE  
808 006644 001401      BEQ     TST63      )1BR IF CLEARED  
809 006646 104011      ERROR 11      )MODE FAILED TO CLEAR  
810  
811      ))*****  
(3)      )-TEST 63      TEST THAT VC MODE BIT 3 CAN BE SET  
(3)      ))*****  
(2) 006650 000004      TST63: SCOPE  
812 006652 012777 000010 172646      MOV      0BIT3,0VCSTAT      )LOAD  
813 006660 012737 000210 001124      MOV      0BIT7,0BIT3,SGDDAT      )LOAD EXPECTED  
814 006666 017737 172634 001126      MOV      0VCSTAT,SGDDAT      )READ REG  
815 006674 023737 001124 001126      CMP      SGDDAT,SGDDAT      )COMPARE  
816 006702 001401      BEQ     TST64      )1BR IF EQUAL  
817 006704 104011      ERROR 11      )ERROR, VC STATUS NOT = 210  
818
```

```
020 (3) (3) (2) 006706 000004
021 006710 012777 000100 172610
022 006716 012737 000300 001124
023 006724 017737 172576 001126
024 006732 023737 001124 001126
025 006740 001401
026 006742 104011
027
028 (3) (3) (2) 006744 000004
029 006746 012777 001000 172552
030 006754 012737 001200 001124
031 006762 017737 172540 001126
032 006770 023737 001124 001126
033 006776 001401
034 007000 104011
035
036 (3) (3) (2) 007002 000004
037 007004 012777 002000 172514
038 007012 012737 002200 001124
039 007020 017737 172502 001126
040 007026 023737 001124 001126
041 007034 001401
042 007036 104011
043
044 (3) (3) (2) 007040 000004
045 007042 012777 004000 172456
046 007050 012737 004200 001124
047 007056 017737 172444 001126
048 007064 023737 001124 001126
049 007072 001401
050 007074 104011
051 (3) (3) (2) 007076 000004
052 007100 012777 010000 172420
053 007106 012737 010000 001124
054 007114 017737 172406 001126
055 007122 023737 001124 001126
056 007130 001401
057 007132 104011
058 007134 005077 172366 101 CLR OVCSTAT

//*****
//TEST 64 TEST THAT VC INTERRUPT ENABLE (BIT 6) CAN BE SET
//*****
TST64: SCOPE
MOV OBIT6,OVCSTAT ILOAD DISPLAY STATUS
MOV OBIT7:BIT6,SGDDAT ILOAD EXPECTED
MOV OVCSTAT,SBDDAT IREAD REG
CMP SGDDAT,SBDDAT ICOMPARE
BEQ TST65 IIBR IF EQUAL
ERROR 11 IERROR, VC STATUS NOT = 300

//*****
//TEST 65 TEST THAT CHANNEL (BIT 9) CAN BE SET
//*****
TST65: SCOPE
MOV OBIT9,OVCSTAT ILOAD DISPLAY STATUS
MOV OBIT9:BIT7,SGDDAT ILOAD EXPECTED
MOV OVCSTAT,SBDDAT IREAD REG
CMP SGDDAT,SBDDAT ICOMPARE
BEQ TST66 IIBR IF EQUAL
ERROR 11 IERROR, VC STATUS NOT = 1200

//*****
//TEST 66 TEST THAT STORE (BIT 10) CAN BE SET
//*****
TST66: SCOPE
MOV OBIT10,OVCSTAT ILOAD DISPLAY STATUS
MOV OBIT10:BIT7,SGDDAT ILOAD EXPECTED
MOV OVCSTAT,SBDDAT IREAD REG
CMP SGDDAT,SBDDAT ICOMPARE
BEQ TST67 IIBR IF EQUAL
ERROR 11 IERROR, VC STATUS NOT = 2200

//*****
//TEST 67 TEST THAT WRITE THRU (BIT 11) CAN BE SET
//*****
TST67: SCOPE
MOV OBIT11,OVCSTAT ILOAD DISPLAY STATUS
MOV OBIT11:BIT7,SGDDAT ILOAD EXPECTED
MOV OVCSTAT,SBDDAT IREAD REG
CMP SGDDAT,SBDDAT ICOMPARE
BEQ TST70 IIBR IF EQUAL
ERROR 11 IERROR, VC STATUS NOT = 4200

//*****
//TEST 70 TEST THAT ERASE (BIT 12) CAN BE SET
//*****
TST70: SCOPE
MOV OBIT12,OVCSTAT ISET ERASE BIT
MOV OBIT12,SGDDAT ILOAD EXPECTED
MOV OVCSTAT,SBDDAT IREAD REG
CMP SGDDAT,SBDDAT ICOMPARE
BEQ 10 IIBR IF SET
ERROR 11 IERROR, ERASE BIT FAILED TO SET
```



```
059                                     ))*****  
(3)                                     ))TEST 71      TEST THAT THE X REGISTER CAN BE CLEARED  
(3)                                     ))*****  
(2) 007140 000004                       TST711  SCOPE  
060 007142 012737 000000 001124         MOV      00,SGDDAT           ;LOAD EXPECTED  
061 007150 013777 001124 172352         MOV      SGDDAT,OVXREG      ;LOAD REG  
062 007156 017737 172346 001126         MOV      OVXREG,SGDDAT     ;READ REG  
063 007164 023737 001124 001126         CMP      SGDDAT,SGDDAT     ;COMPARE  
064 007172 001401                       BEQ      TST72              ;BR IF EQUAL  
065 007174 104012                       ERROR    12                 ;ERROR, VC XREGISTER NOT = 0  
066  
067                                     ))*****  
(3)                                     ))TEST 72      TEST THAT THE X REGISTER CAN BE LOADED WITH 01777  
(3)                                     ))*****  
(2) 007176 000004                       TST721  SCOPE  
068 007200 012737 001777 001124         MOV      01777,SGDDAT      ;LOAD EXPECTED  
069 007206 013777 001124 172314         MOV      SGDDAT,OVXREG      ;LOAD REG  
070 007214 017737 172310 001126         MOV      OVXREG,SGDDAT     ;READ REG  
071 007222 023737 001124 001126         CMP      SGDDAT,SGDDAT     ;COMPARE  
072 007230 001401                       BEQ      TST73              ;BR IF EQUAL  
073 007232 104012                       ERROR    12                 ;ERROR, VC X REGISTER NOT = 1777  
074  
075                                     ))*****  
(3)                                     ))TEST 73      TEST THAT THE X REGISTER CAN BE LOADED WITH 0525  
(3)                                     ))*****  
(2) 007234 000004                       TST731  SCOPE  
076 007236 012737 000525 001124         MOV      0525,SGDDAT      ;LOAD EXPECTED  
077 007244 013777 001124 172256         MOV      SGDDAT,OVXREG      ;LOAD REG,  
078 007252 017737 172252 001126         MOV      OVXREG,SGDDAT     ;READ REG  
079 007260 023737 001124 001126         CMP      SGDDAT,SGDDAT     ;COMPARE  
080 007266 001401                       BEQ      TST74              ;BR IF EQUAL  
081 007270 104012                       ERROR    12                 ;ERROR, VC X REGISTER NOT = 525  
082  
083                                     ))*****  
(3)                                     ))TEST 74      TEST THAT THE X REGISTER CAN BE LOADED WITH 01252  
(3)                                     ))*****  
(2) 007272 000004                       TST741  SCOPE  
084 007274 012737 001252 001124         MOV      01252,SGDDAT     ;LOAD EXPECTED  
085 007302 013777 001124 172220         MOV      SGDDAT,OVXREG      ;LOAD REG,  
086 007310 017737 172214 001126         MOV      OVXREG,SGDDAT     ;READ REG  
087 007316 023737 001124 001126         CMP      SGDDAT,SGDDAT     ;COMPARE  
088 007324 001401                       BEQ      TST75              ;BR IF EQUAL  
089 007326 104012                       ERROR    12                 ;ERROR, VC X REGISTER NOT = 1252  
090
```

```
092  
(3)  
(3)  
(2) 007330 000000  
(1) 007332 012737 000100 001164  
093 007340 012737 007352 001110  
094 007346 005037 001124  
095 007352 013777 001124 172150 101  
096 007360 017737 172140 001126  
097 007366 023737 001124 001126  
098 007374 001401  
099 007376 104012  
900  
901 007400 005237 001124 201 INC SGDDAT ILOAD EXPECTED  
902 007404 022737 002000 001124 CMP #2000,SGDDAT ILOAD Y REG  
903 007412 001357 BNE 19 IREAD REG  
904  
905  
(3)  
(3)  
(2) 007414 000000  
906 007416 012737 000000 001124  
907 007424 013777 001124 172100  
908 007432 017737 172074 001126  
909 007440 023737 001124 001126  
910 007446 001401  
911 007450 104013  
912  
913  
(3)  
(3)  
(2) 007452 000000  
914 007454 012737 001777 001124  
915 007462 013777 001124 172042  
916 007470 017737 172036 001126  
917 007476 023737 001124 001126  
918 007504 001401  
919 007506 104013  
920  
921  
(3)  
(3)  
(2) 007510 000000  
922 007512 012737 000525 001124  
923 007520 013777 001124 172004  
924 007526 017737 172000 001126  
925 007534 023737 001124 001126  
926 007542 001401  
927 007544 104013  
928
```

```
))  
I*TEST 75 TEST THAT THE X REGISTER CAN HOLD A COUNT PATTERN  
))
```

```
TST75: SCOPE  
MOV #100,STIMES ILOAD 100 ITERATIONS  
MOV #19,SLPERR ILOAD SCOPE ERROR RETURN  
CLR SGDDAT ICLEAR EXPECTED  
MOV SGDDAT,OVXREG ILOAD REG  
MOV OVXREG,SBDDAT IREAD REG  
CMP SGDDAT,SBDDAT ICOMPARE  
BEQ 29 IIBR IF EQUAL  
ERROR 12 IVC X REG FAILED TO HOLD A COUNT PATTERN  
  
29: INC SGDDAT IUPDATE PATTERN  
CMP #2000,SGDDAT IFINISHED?  
BNE 19 IIBR IF NOT
```

```
))  
I*TEST 76 TEST THAT THE Y REGISTER CAN BE CLEARED  
))
```

```
TST76: SCOPE  
MOV #0,SGDDAT ILOAD EXPECTED  
MOV SGDDAT,OVYREG ILOAD Y REG  
MOV OVYREG,SBDDAT IREAD REG  
CMP SGDDAT,SBDDAT ICOMPARE  
BEQ TST77 IIBR IF EQUAL  
ERROR 13 IERROR, VC Y REGISTER NOT = 0
```

```
))  
I*TEST 77 TEST THAT THE Y REGISTER CAN BE LOADED WITH 01777  
))
```

```
TST77: SCOPE  
MOV #1777,SGDDAT ILOAD EXPECTED  
MOV SGDDAT,OVYREG ILOAD Y REG  
MOV OVYREG,SBDDAT IREAD REG  
CMP SGDDAT,SBDDAT ICOMPARE  
BEQ TST100 IIBR IF EQUAL  
ERROR 13 IERROR, VC Y REGISTER NOT = 1777
```

```
))  
I*TEST 100 TEST THAT THE Y REGISTER CAN BE LOADED WITH 0525  
))
```

```
TST100: SCOPE  
MOV #525,SGDDAT ILOAD EXPECTED  
MOV SGDDAT,OVYREG ILOAD Y REG  
MOV OVYREG,SBDDAT IREAD REG  
CMP SGDDAT,SBDDAT ICOMPARE  
BEQ TST101 IIBR IF EQUAL  
ERROR 13 IERROR, VC Y REGISTER NOT = 525
```



```
930 //*****  
(3) ;*TEST 101 TEST THAT THE Y REGISTER CAN BE LOADED WITH #1252  
(3) //*****  
(2) 007546 000004 TST101: SCOPE  
931 007550 012737 001252 001124 MOV #1252,SGDDAT ;LOAD EXPECTED  
932 007556 013777 001124 171746 MOV SGDDAT,OVYREG ;LOAD Y REG  
933 007564 017737 171742 001126 MOV OVYREG,SBDDAT ;READ REG  
934 007572 023737 001124 001126 CMP SGDDAT,SBDDAT ;COMPARE  
935 007600 001401 BEQ TST102 ;BR IF EQUAL  
936 007602 104013 ERROR 13 ;ERROR, VC Y REGISTER NOT = 1252  
937  
938 //*****  
(3) ;*TEST 102 TEST THAT THE Y REGISTER CAN HOLD A COUNT PATTERN  
(3) //*****  
(2) 007604 000004 TST102: SCOPE  
(1) 007606 012737 000100 001164 MOV #100,STIMES ;DO 100 ITERATIONS  
939 007614 012737 007626 001110 MOV #18,SLPERR ;LOAD SCOPE ERROR RETURN  
940 007622 005037 001124 CLR SGDDAT ;CLEAR EXPECTED  
941 007626 013777 001124 171676 10: MOV SGDDAT,OVYREG ;LOAD REG  
942 007634 017737 171672 001126 MOV OVYREG,SBDDAT ;READ REG  
943 007642 023737 001124 001126 CMP SGDDAT,SBDDAT ;COMPARE  
944 007650 001401 BEQ 20 ;BR IF EQUAL  
945 007652 104013 ERROR 13 ;VC Y REG FAILED TO HOLD A COUNT PATTERN  
946  
947 007654 005237 001124 20: INC SGDDAT ;UPDATE PATTERN  
948 007660 022737 002000 001124 CMP #2000,SGDDAT ;FINISHED?  
949 007666 001357 BNE 10 ;BR IF NOT  
950  
951 //*****  
(3) ;*TEST 103 TEST THAT THE X-Y REGISTERS CAN HOLD DIFFERENT DATA  
(3) //*****  
(2) 007670 000004 TST103: SCOPE  
952 007672 012777 001252 171630 MOV #1252,OVXREG ;LOAD X REGISTER  
953 007700 012737 000525 001124 MOV #525,SGDDAT ;LOAD EXPECTED  
954 007706 013777 001124 171616 MOV SGDDAT,OVYREG ;LOAD Y REG  
955 007714 017737 171612 001126 MOV OVYREG,SBDDAT ;READ REG  
956 007722 023737 001124 001126 CMP SGDDAT,SBDDAT ;COMPARE  
957 007730 001401 BEQ 10 ;BR IF EQUAL  
958 007732 104013 ERROR 13 ;ERROR, SELECTED Y REGISTER INCORRECTLY  
959  
960 007734 012737 001252 001124 10: MOV #1252,SGDDAT ;LOAD EXPECTED  
961 007742 017737 171562 001126 MOV OVXREG,SBDDAT ;READ REG  
962 007750 023737 001124 001126 CMP SGDDAT,SBDDAT ;COMPARE  
963 007756 001401 BEQ TST104 ;BR IF EQUAL  
964 007760 104012 ERROR 12 ;ERROR, SELECTED X REGISTER INCORRECTLY  
965
```

```
967 //*****  
(3) /*TEST 104 TEST THAT WHEN INTENSIFY BIT IS SET THAT THE VC READY BIT CLEARS  
(3) //*****  
(2) 007762 000004 TST104: SCOPE  
968 ) AND THEN SETS AFTER A DELAY  
969 007764 012700 001000 MOV #1000,R0 ;LOAD EXPECTED  
970 007770 012737 000000 001124 MOV #0,SGDDAT ;INTENSIFY  
971 007776 012777 000001 171522 MOV #BIT0,OVSTAT ;READ REG  
972 010004 017737 171516 001126 MOV OVSTAT,SBDDAT ;TEST READY  
973 010012 105737 001126 TSTR SBDDAT ;IBR IF NOT SET  
974 010016 100002 BPL IS ;READY FAILED TO CLEAR  
975 010020 104011 ERROR 11 ;IBR TO SCOPE  
976 010022 000414 BR TST105 ;NEXT TEST  
977 010024 105777 171476 18: TSTR OVSTAT ;DELAY  
978 010030 100411 BHI TST105 ;LOAD EXPECTED  
979 010032 005300 DEC R0 ;READ REG  
980 010034 001373 BNE IS ;READY FAILED TO SET AFTER A DELAY  
981 010036 012737 000200 001124 MOV #BIT7,SGDDAT  
982 010044 017737 171456 001126 MOV OVSTAT,SBDDAT  
983 010052 104011 ERROR 11  
984  
985 //*****  
(3) /*TEST 105 TEST THAT VC MODE 1 (INTENSIFY ON X) CLEARS THE READY FLAG  
(3) //*****  
(2) 010054 000004 TST105: SCOPE  
986 ) AND THEN SETS IT  
987 010056 012700 001000 MOV #1000,R0 ;SET UP DELAY  
988 010062 012737 000204 001124 MOV #BIT7:BIT2,SGDDAT ;LOAD EXPECTED  
989 010070 012777 000004 171430 MOV #BIT2,OVSTAT ;LOAD MODE 1  
990 010076 017737 171424 001126 MOV OVSTAT,SBDDAT ;READ REG  
991 010104 105737 001126 TSTR SBDDAT ;TEST READY  
992 010110 100402 BHI 20 ;IBR IF READY STILL SET  
993 010112 104011 ERROR 11 ;ERROR, IN MODE 1 READY SHOULD NOT  
994 ; CLEAR UNTIL X IS LOADED  
995 010114 000432 BR TST106 ;IBR TO SCOPE  
996  
997 010116 012737 000004 001124 28: MOV #BIT2,SGDDAT ;LOAD EXPECTED  
998 010124 005077 171400 CLR OVXREG ;ADDRESS X AXIS  
999 010130 017737 171372 001126 MOV OVSTAT,SBDDAT ;READ REG  
1000 010136 105737 001126 TSTR SBDDAT ;TEST READY  
1001 010142 000240 NOP  
1002 010144 100002 BPL IS ;IBR IF CLEAR  
1003 010146 104011 ERROR 11 ;ERROR, MODE 1 LOAD X FAILED TO CLEAR READY FLAG  
1004 010150 000414 BR TST106 ;IBR TO SCOPE  
1005  
1006 010152 105777 171350 18: TSTR OVSTAT ;TEST READY  
1007 010156 100411 BHI TST106 ;NEXT TEST  
1008 010160 005300 DEC R0 ;DELAY  
1009 010162 001373 BNE IS ;TEST READY AGAIN  
1010 010164 012737 000204 001124 MOV #BIT7:BIT2,SGDDAT ;LOAD EXPECTED  
1011 010172 017737 171330 001126 MOV OVSTAT,SBDDAT ;READ REG  
1012 010200 104011 ERROR 11 ;ERROR, READY FAILED TO SET  
1013 ; AFTER MODE 1 OPERATION  
1014
```



```
1016 (3)
1017 (3)
1018 (2) 010202 000004
1019 010204 012700 001000
1020 010210 012737 000210 001124
1021 010216 012777 000010 171302
1022 010224 017737 171276 001126
1023 010232 105737 001126
1024 010236 100402
1025 010240 104011
1026 010242 000431
1027 010244 012737 000010 001124 25:
1028 010252 005077 171254
1029 010256 017737 171244 001126
1030 010264 105737 001126
1031 010270 100002
1032 010272 104011
1033 010274 000414
1034 010276 105777 171224 18:
1035 010302 100411
1036 010304 005300
1037 010306 001373
1038 010310 012737 000210 001124
1039 010316 017737 171204 001126
1040 010324 104011
1041
1042
1043 (3)
1044 (3)
1045 (2) 010326 000004
1046 010330 032777 010000 170600
1047 010336 001430
1048 010340 012700 000010
1049 010344 005037 001566
1050 010350 012777 002000 171150
1051 010356 052777 010000 171142
1052 010364 105777 171136
1053 010370 100002
1054 010372 104011
1055 010374 000411
1056 010376 105777 171124 18:
1057 010402 100406
1058 010404 005337 001566
1059 010410 001372
1060 010412 005300
1061 010414 001370
1062 010416 104011

//*****
//TEST 106 TEST THAT VC MODE 2 (INTENSIFY ON Y) CLEARS THE READY FLAG
//*****
TST106: SCOPE
) AND THEN SETS IT
MOV 01000,R0 ISET UP DELAY
MOV 00IT7:BIT3,SGDDAT ILOAD EXPECTED
MOV 00IT3,OVSTAT ILOAD MODE 2
MOV OVSTAT,SBDDAT IREAD REG
TSTB SBDDAT ITEST READY
BMY 25 IJBR IF SET
ERROR 11 IERROR, IN MODE 2 READY SHOULD NOT CLEAR
UNTIL Y IS LOADED
BR TST107 IJBR TO SCOPE
MOV 00IT3,SGDDAT ILOAD EXPECTED
CLR OVCYREG IADDRESS Y AXIS
MOV OVSTAT,SBDDAT IREAD REG
TSTB SBDDAT ITEST READY
BPL 15 IJBR IF CLEARED
ERROR 11 IERROR, MODE 2 LOAD Y FAILED TO CLEAR READY FLAG
BR TST107 IJBR TO SCOPE
TSTB OVSTAT ITEST READY
BMY TST107 IJNEXT TEST
DEC R0 IDELAY
BNE 15 IJTEST READY AGAIN
MOV 00IT7:BIT3,SGDDAT ILOAD EXPECTED
MOV OVSTAT,SBDDAT
ERROR 11 IERROR, READY FAILED TO SET
AFTER MODE 2 OPERATION

//*****
//TEST 107 TEST WHEN ERASE IS SET, VC READY BIT CLEARS AND SET AFTER DELAY
//*****
TST107: SCOPE
BIT 00IT12,00HR ITEST BIT 12
BEG TST110 IJBYPASS IF NO STORAGE SCOPE
MOV 010,R0
CLR TEMP ICLEAR DELAY
MOV 00IT10,OVSTAT ISET STORE MODE
BIT 00IT12,OVSTAT ISET ERASE BIT
TSTB OVSTAT ITEST THAT READY CLEARS
BPL 15 IJBR IF CLEARED
ERROR 11 IERROR, READY FAILED TO RESET
BR TST110 IJBR TO SCOPE
TSTB OVSTAT ITEST FOR READY
BMY TST110 IJBR IF SET
DEC TEMP IDELAY
BNE 15 IJBR IF NOT READY
DEC R0 IDECREMENT COUNTER
BNE 15 IJBR IF NOT DONE
ERROR 11 IERROR, ERASE CLEARED READY AND FAILED
TO SET READY AFTER A DELAY
```

```
1064 (3) //*****  
1065 (3) //TEST 110 SCOPE PRE-INTERRUPT SETUP  
1066 (2) //*****  
1067 (1) TST110: SCOPE  
1068 010420 000004 MOV 01,STIMES //DO 1 ITERATION  
1069 010422 012737 000001 001164 BIC 0177437,VCBRL //MASK TO PSW  
1070 010430 042737 177437 001514 BNE .+4 //LOCATION VCBRL CONTAINED A BR LEVEL 0  
1071 010436 001001 HALT //LOCATION VCBRL CONTAINED A BR LEVEL 0  
1072 010440 000000 //LOCATION VCBRL CONTAINED A BR LEVEL 0  
1073 010442 022737 000340 001514 CMP 0340,VCBRL //LOCATION VCBRL CONTAINS BR LEVEL 7  
1074 010450 001001 BNE .+4 //LOCATION VCBRL CONTAINS BR LEVEL 7  
1075 010452 000000 HALT //LOCATION VCBRL CONTAINS BR LEVEL 7  
1076 010454 013737 001514 001572 MOV VCBRL,BRLEV1 //SET UP BR LEVELS  
1077 010462 162737 000040 001572 SUB 040,BRLEV1 // -1  
1078 010470 013737 001514 001574 MOV VCBRL,BRLEV2 // 0  
1079 (3) //*****  
1080 (3) //TEST 111 TEST THAT THE DISPLAY DOES INTERRUPT AT LEVEL INDICATED -1  
1081 (2) //*****  
1082 (1) TST111: SCOPE  
1083 010476 000004 MOV 040,STIMES //DO 40 ITERATIONS  
1084 010500 012737 000040 001164 MOV 0340,PSW  
1085 010506 012737 000340 177776 MOV 018,0VCIV //SET INTERRUPT VECTOR  
1086 010514 012777 010556 171024 MOV 0400,RO //SET UP DELAY  
1087 010522 012700 000400 MOV BRLEV1,PSW //START DISPLAY  
1088 010526 013737 001572 177776 MOV 0BIT6,0BIT0,0VCSTAT //DELAY  
1089 010534 012777 000101 170764 DEC RO  
1090 010542 005300 BNE .-2  
1091 010544 001376 CLR 0VCSTAT //DO NOT LET INTERRUPT ENABLE SET  
1092 010546 005077 170754 ERROR 14 //ERROR, VC FAILED TO INTERRUPT  
1093 010552 104014 BR TST112 //NEXT TEST  
1094 010554 000404 181 MOV VCIV8,0VCIV //RESET VECTOR  
1095 010556 013777 001550 170762 181 CMP (SP)+,(SP)+ //POP STACK  
1096 010564 022626 //*****  
1097 (3) //TEST 112 TEST THAT THE DISPLAY DOES NOT INTERRUPT AT LEVEL INDICATED  
1098 (3) //*****  
1099 (2) TST112: SCOPE  
1100 (1) MOV 040,STIMES //DO 40 ITERATIONS  
1101 010566 000004 MOV 0340,PSW  
1102 010570 012737 000040 001164 MOV 018,0VCIV //SET INTERRUPT VECTOR  
1103 010576 012737 000340 177776 MOV 0400,RO //SET UP DELAY  
1104 010604 012777 010640 170734 MOV BRLEV2,PSW //START DISPLAY  
1105 010612 012700 004000 DEC RO //DELAY  
1106 010616 013737 001574 177776 BNE .-2  
1107 010624 012777 000101 170674 BR 2E //  
1108 010632 005300 CLR 0VCSTAT //DO NOT LET INTERRUPT ENABLE SET  
1109 010634 001376 ERROR 14 //ERROR VC INTERRUPTED IN ERROR  
1110 010636 000404 BR TST113 //NEXT TEST  
1111 010640 005077 170662 181 MOV 038,0VCIV //LOAD RETURN VECTOR  
1112 010644 104014 281 BIC 0340,PSW //LOWER PSW  
1113 010646 000416 CLR 0VCSTAT //CLEAR INT ENABLE  
1114 010650 012777 010674 170670 281 ERROR 14 //LOWERING THE PRIORITY FAILED TO ALLOW INTERRUPT  
1115 010654 042737 000340 177776 BR TST113 //NEXT TEST  
1116 010664 005077 170636 381 MOV VCIV8,0VCIV //RESET VECTOR  
1117 010670 104014 381 CMP (SP)+,(SP)+ //POP STACK  
1118 010672 000404  
1119 010674 013777 001550 170644 381  
1120 010702 022626
```



```
1107 (3) (3) (2) 010704 000004 (1) 010706 012737 000040 001164 1108 010714 012777 000014 170604 1109 010722 012737 000200 001124 1110 010730 000005 1111 010732 017737 170570 001126 1112 010740 023737 001124 001126 1113 010746 001401 1114 010750 104011 1115 1116 (3) (3) (2) 010752 000004 (1) 010754 012737 000040 001164 1117 010762 012777 007100 170536 1118 010770 012737 000200 001124 1119 010776 000005 1120 011000 017737 170522 001126 1121 011006 023737 001124 001126 1122 011014 001401 1123 011016 104011 1124 1125 (3) (3) (2) 011020 000004 (1) 011022 012737 000040 001164 1126 011030 012777 177777 170472 1127 011036 012737 000000 001124 1128 011044 000005 1129 011046 017737 170456 001126 1130 011054 023737 001124 001126 1131 011062 001401 1132 011064 104011 1133 1134 (3) (3) (2) 011066 000004 (1) 011070 012737 000040 001164 1135 011076 012777 177777 170426 1136 011104 012737 000000 001124 1137 011112 000005 1138 011114 017737 170412 001126 1139 011122 023737 001124 001126 1140 011130 001401 1141 011132 104011 1142
```

```
;;*****  
;TEST 113 TEST THAT RESET CLEARS VC MODE BITS  
;;*****  
TST113: SCOPE  
NOV 040,STIMES ;DO 40 ITERATIONS  
NOV 0BIT3;BIT2,0VCSTAT  
NOV 0BIT7,SGDDAT ;LOAD EXPECTED  
RESET  
NOV 0VCSTAT,SBDDAT ;READ STATUS  
CMP SGDDAT,SBDDAT ;COMPARE  
BEQ TST114 ;BR IF EQUAL  
ERROR 11 ;ERROR, RESET FAILED TO CLEAR VC STATUS REG  
  
;;*****  
;TEST 114 TEST THAT RESET CLEARS INTERRUPT ENABLE, CHANNEL, STORE, WRITE THRU  
;;*****  
TST114: SCOPE  
NOV 040,STIMES ;DO 40 ITERATIONS  
NOV 0BIT11;BIT10;BIT9;BIT6,0VCSTAT  
NOV 0BIT7,SGDDAT ;LOAD EXPECTED  
RESET  
NOV 0VCSTAT,SBDDAT ;READ STATUS  
CMP SGDDAT,SBDDAT ;COMPARE  
BEQ TST115 ;BR IF EQUAL  
ERROR 11 ;ERROR, RESET FAILED TO CLEAR VC STATUS  
  
;;*****  
;TEST 115 TEST THAT RESET CLEARS X REGISTER  
;;*****  
TST115: SCOPE  
NOV 040,STIMES ;DO 40 ITERATIONS  
NOV 0-1,0VCXREG  
NOV 00,SGDDAT ;LOAD EXPECTED  
RESET  
NOV 0VCXREG,SBDDAT ;READ REG  
CMP SGDDAT,SBDDAT ;COMPARE  
BEQ TST116 ;BR IF EQUAL  
ERROR 11 ;ERROR, RESET FAILED TO CLEAR VC X REGISTER  
  
;;*****  
;TEST 116 TEST THAT RESET CLEARS Y REGISTER  
;;*****  
TST116: SCOPE  
NOV 040,STIMES ;DO 40 ITERATIONS  
NOV 0-1,0VCYREG  
NOV 00,SGDDAT ;LOAD EXPECTED  
RESET  
NOV 0VCYREG,SBDDAT ;READ REG  
CMP SGDDAT,SBDDAT ;COMPARE  
BEQ TST117 ;BR IF EQUAL  
ERROR 11 ;ERROR, RESET FAILED TO CLEAR VC Y REGISTER
```

```
1144 (3) //*****
(3) //TEST 117 DOES EXTERNAL ENABLE (BIT 4) SET
(2) //*****
011134 000004 TST117: SCOPE
1145 011136 012777 000020 170352 MOV @BIT4,@ADCS ISET BIT 4
1146 011144 012737 000020 001124 MOV @BIT4,@GDDAT ILOAD EXPECTED
1147 011152 017737 170340 001126 MOV @ADCS,@SDDAT IREAD REG
1148 011160 023737 001124 001126 CMP @GDDAT,@SDDAT ICOMPARE
1149 011166 001401 BEQ TST120 //BR IF SET
1150 011170 104001 ERROR 1 IEXT ENABLE BIT 4 FAILED TO SET
1151
1152 (3) //*****
(3) //TEST 120 DOES CLOCK OVERFLOW ENABLE (BIT 5) SET
(2) //*****
011172 000004 TST120: SCOPE
1153 011174 012777 000040 170314 MOV @BIT5,@ADCS ISET BIT 5
1154 011202 012737 000040 001124 MOV @BIT5,@GDDAT ILOAD EXPECTED
1155 011210 017737 170302 001126 MOV @ADCS,@SDDAT IREAD REG
1156 011216 023737 001124 001126 CMP @GDDAT,@SDDAT ICOMPARE
1157 011224 001401 BEQ TST121 //BR IF SET
1158 011226 104001 ERROR 1 ICLOCK OVERFLOW ENABLE FAILED TO SET
1159
1160 (3) //*****
(3) //TEST 121 DOES AD INTERRUPT ENABLE (BIT 6) SET
(2) //*****
011230 000004 TST121: SCOPE
1161 011232 012777 000100 170256 MOV @BIT6,@ADCS ISET BIT 6
1162 011240 012737 000100 001124 MOV @BIT6,@GDDAT ILOAD EXPECTED
1163 011246 017737 170244 001126 MOV @ADCS,@SDDAT IREAD REG
1164 011254 023737 001124 001126 CMP @GDDAT,@SDDAT ICOMPARE
1165 011262 001401 BEQ TST122 //BR IF SET
1166 011264 104001 ERROR 1 IAD INTERRUPT ENABLE FAILED TO SET
1167
1168 (3) //*****
(3) //TEST 122 DOES MUX CHANNEL (BIT 8) SET
(2) //*****
011266 000004 TST122: SCOPE
1169 011270 012777 000400 170220 MOV @BIT8,@ADCS ISET BIT 8
1170 011276 012737 000400 001124 MOV @BIT8,@GDDAT ILOAD EXPECTED
1171 011304 017737 170206 001126 MOV @ADCS,@SDDAT IREAD REG
1172 011312 023737 001124 001126 CMP @GDDAT,@SDDAT ICOMPARE
1173 011320 001401 BEQ TST123 //BR IF SET
1174 011322 104001 ERROR 1 IMUX BIT 8 FAILED TO SET
1175
1176 (3) //*****
(3) //TEST 123 DOES MUX CHANNEL (BIT 9) SET
(2) //*****
011324 000004 TST123: SCOPE
1177 011326 012777 001000 170162 MOV @BIT9,@ADCS ISET BIT 9
1178 011334 012737 001000 001124 MOV @BIT9,@GDDAT ILOAD EXPECTED
1179 011342 017737 170150 001126 MOV @ADCS,@SDDAT IREAD REGISTER
1180 011350 023737 001124 001126 CMP @GDDAT,@SDDAT ICOMPARE
1181 011356 001401 BEQ TST124 //BR IF EQUAL
1182 011360 104001 ERROR 1 IMUX BIT 9 FAILED TO SET
```



```
1184      ;);
(3)      ;);
(3)      ;);
(2)      011362 000004
1185      011364 012777 002000 170124      MOV      @BIT10,@ADCS      ;SET BIT 10
1186      011372 012737 002000 001124      MOV      @BIT10,SGDDAT    ;LOAD EXPECTED
1187      011400 017737 170112 001126      MOV      @ADCS,SBDDAT    ;READ REG.
1188      011406 023737 001124 001126      CMP      SGDDAT,SBDDAT    ;COMPARE
1189      011414 001401      BEQ      TST125          ;BR IF EQUAL
1190      011416 104001      ERROR    1              ;MUX BIT 10 FAILED TO SET
1191
1192      ;);
(3)      ;);
(3)      ;);
(2)      011420 000004
1193      011422 012777 004000 170066      MOV      @BIT11,@ADCS      ;SET BIT 11
1194      011430 012737 004000 001124      MOV      @BIT11,SGDDAT    ;LOAD EXPECTED
1195      011436 017737 170054 001126      MOV      @ADCS,SBDDAT    ;READ REGISTER
1196      011444 023737 001124 001126      CMP      SGDDAT,SBDDAT    ;COMPARE
1197      011452 001401      BEQ      TST126          ;BR IF EQUAL
1198      011454 104001      ERROR    1              ;MUX BIT 11 FAILED TO SET
1199
1200      ;);
(3)      ;);
(3)      ;);
(2)      011456 000004
1201      011460 012777 020000 170030      MOV      @BIT13,@ADCS      ;SET BIT 13
1202      011466 012737 020000 001124      MOV      @BIT13,SGDDAT    ;LOAD EXPECTED
1203      011474 017737 170016 001126      MOV      @ADCS,SBDDAT    ;READ REGISTER
1204      011502 023737 001124 001126      CMP      SGDDAT,SBDDAT    ;COMPARE
1205      011510 001401      BEQ      TST127          ;BR IF EQUAL
1206      011512 104001      ERROR    1              ;UNIPOLAR /BIPOLAR BIT FAILED TO SET
1207
1208      ;);
(3)      ;);
(3)      ;);
(2)      011514 000004
1209      011516 012737 001000 001124      MOV      @BIT9,SGDDAT     ;LOAD EXPECTED VALUE
1210      011524 012777 001040 167764      MOV      @BIT9,@BITS,@ADCS ;LOAD A TO D REG.
1211      011532 105077 167760      CLRB     @ADCS           ;CLEAR THE LOW BYTE
1212      011536 017737 167754 001126      MOV      @ADCS,SBDDAT    ;READ RESULT
1213      011544 023737 001124 001126      CMP      SBDDAT,SBDDAT    ;COMPARE RESULTS
1214      011552 001401      BEQ      TST130          ;BR IF THE SAME
1215      011554 104001      ERROR    1              ;FAILED TO CLEAR ONLY THE LOW BYTE OF A TO D
1216
```

```
1218                                     ;*****  
(3) ;TEST 130 TEST FOR PROPER SELECTION OF THE HIGH BYTE OPERATION  
(3) ;*****  
(2) 011556 000004 TST130: SCOPE  
1219 011560 012737 000040 001124 MOV 0BIT5,SGDDAT ILOAD EXPECTED  
1220 011566 013737 001516 001566 MOV ADCS,TEMP IGET BASE ADDRESS OF A TO D  
1221 011574 005237 001566 INC TEMP IMAKE HIGH BYTE ADDRESS  
1222 011600 012777 001040 167710 MOV 0BIT9,0BIT5,0ADCS ILOAD A TO D REG  
1223 011606 105077 167754 CLRB 0TEMP I SHOULD CLEAR HIGH BYTE ONLY  
1224 011612 017737 167700 001126 MOV 0ADCS,0BDDAT IREAD A TO D REG.  
1225 011620 023737 001124 001126 CMP SGDDAT,0BDDAT ICOMPARE RESULTS  
1226 011626 001401 BEQ TST131 IJBR IF THE SAME  
1227 011630 104001 ERROR 1 IFAILED TO CLEAR ONLY THE HIGH BYTE OF A TO D  
1228  
1229                                     ;*****  
(3) ;TEST 131 TEST AD 60 (BIT 0) CAN BE SET AND CLEARED  
(3) ;*****  
(2) 011632 000004 TST131: SCOPE  
1230  
1231 011634 017700 167660 MOV 0ADDBR,R0 IFAKE READ  
1232 011640 005077 167652 CLR 0ADCS  
1233 011644 012737 000001 001124 MOV 0BIT0,SGDDAT ILOAD EXPECTED  
1234 011652 013777 001124 167636 MOV SGDDAT,0ADCS IREAD REGISTER  
1235 011660 017737 167632 001126 MOV 0ADCS,0BDDAT ICOMPARE  
1236 011666 023737 001124 001126 CMP SGDDAT,0BDDAT ICOMPARE  
1237 011674 001401 BEQ 10 IJBR IF SET  
1238 011676 104001 ERROR 1 IA TO D 60 FAILED TO SET  
1239 011700 012700 001000 101 MOV 01000,R0  
1240 011704 005300 DEC R0  
1241 011706 001376 BNE .-2  
1242 011710 017737 167602 001126 MOV 0ADCS,0BDDAT IREAD REG.  
1243 011716 012737 000200 001124 MOV 0BIT7,SGDDAT ILOAD EXPECTED  
1244 011724 032737 000001 001126 BIT 0BIT0,0BDDAT ITEST BIT  
1245 011732 001401 BEQ TST132 IJBR IF CLEARED  
1246 011734 104001 ERROR 1 IA TO D 60 FAILED TO CLEAR
```



1248  
1249  
(3)  
(3)  
(2)  
1250  
1251  
1252  
1253  
1254  
1255  
1256  
1257  
1258  
1259  
1260  
1261  
1262  
1263  
1264  
1265  
1266  
1267  
1268  
1269  
1270  
1271  
1272  
1273  
1274  
(3)  
(3)  
(2)  
1275  
1276  
1277  
1278  
1279  
1280  
1281  
1282  
1283  
1284  
1285  
1286  
1287

011736 000004  
011740 017700 167554  
011744 012700 000000  
011750 012777 000001 167540  
011756 012737 000200 001124  
011764 105777 167526  
011770 100406  
011772 005200  
011774 001373  
011776 017737 167514 001126  
012004 104001  
012006 017700 167506  
012012 012737 000000 001124  
012020 017737 167472 001126  
012026 105737 001126  
012032 100001  
012034 104001  
012036 017737 167450 001126  
012044 005037 001124  
012050 032737 000001 001126  
012056 001401  
012060 104001  
012062 000004  
012064 005077 167426  
012070 005277 167422  
012074 105777 167416  
012100 100373  
012102 012737 001777 001124  
012110 017737 167404 001126  
012116 032777 002000 167012  
012124 001405  
012126 023737 001124 001126  
012134 001401  
012136 104016

\*\*\*\*\*  
TEST 132 DOES AD DONE (BIT 7) SET AND CLEAR  
\*\*\*\*\*

TEST132: SCOPE  
MOV @ADDR,R0  
MOV @R0  
MOV @BIT0,@ADCS  
MOV @BIT7,@SODAT  
TSTB @ADCS  
BHI 28  
INC R0  
BNE 18  
MOV @ADCS,@SODAT  
ERROR 1  
MOV @ADDR,R0  
MOV @R0,@SODAT  
MOV @ADCS,@SODAT  
TSTB @SODAT  
BPL 38  
ERROR 1  
MOV @ADCS,@SODAT  
CLR @SODAT  
BIT @BIT0,@SNR  
BEQ TEST133  
ERROR 1

ILOAD EXPECTED  
IREAD REGISTER  
IERROR, A TO D DONE FAILED TO SET  
ICLEAR EXPECTED  
IREAD REG  
IIBR IF CLEARED  
IERROR, DONE FAILED TO CLEAR UPON  
IREADING CONVERTED VALUE  
IREAD REG  
ICLEAR EXPECTED  
IIBR IF CLEARED  
IERROR, AD GO FAILED TO CLEAR

\*\*\*\*\*  
TEST 133 TEST THAT THE CONVERTED NUMBER = 1777 (SN BIT 10=1)  
\*\*\*\*\*

TEST133: SCOPE  
CLR @ADCS  
INC @ADCS  
TSTB @ADCS  
BPL 18  
MOV @1777,@SODAT  
MOV @ADDR,@SODAT  
BIT @BIT10,@SNR  
BEQ TEST134  
CMP @SODAT,@SODAT  
BEQ TEST134  
ERROR 16

INIT  
CONVERT  
LOOP  
ILOAD EXPECTED  
IREAD REG  
ITEST BIT 10  
IIBR IF CLEARED  
ICOMPARE  
IIBR IF EQUAL  
I CONVERTED VALUE NOT = 1777  
OPERATOR INFORMED THE PROGRAM THAT THE  
ALL 1'S JUMPER HAD BEEN INSTALLED

```
1289  
1290  
(3)  
(3)  
(2) 012140 000004  
(1) 012142 012737 000004 001164  
1291 012150 005077 167342  
1292 012154 012737 000020 001124  
1293 012162 017700 167332  
1294 012166 052777 000020 167322  
1295 012174 012700 000000  
1296 012200 005200 101 INC R0  
1297 012202 001376 101 BNE 10  
1298 012204 017737 167306 001126 MOV 0ADCS,0DDAT IREAD REG  
1299 012212 105737 001126 TST0 0DDAT  
1300 012216 100001 BPL TST135 I1BR IF PLUS  
1301 012220 104001 ERROR 1 IERROR EXTERNAL CONVERSION
```

```
1302  
1303  
(3)  
(3)  
(2) 012222 000004  
(1) 012224 012737 000004 001164  
1304 012232 005077 167260  
1305 012236 005077 167260  
1306 012242 012737 000240 001124  
1307 012250 012737 000003 014056  
1308 012256 052777 000040 167232  
1309 012264 004737 013376  
1310 012270 012700 000040  
1311 012274 005300 101 DEC R0  
1312 012276 001376 101 BNE 10 IDELAY  
1313 012300 017737 167212 001126 MOV 0ADCS,0DDAT IREAD REGISTER  
1314 012306 023737 001124 001126 CMP 0DDAT,0DDAT ICOMPARE  
1315 012314 001401 BEQ TST136 I1BR IF EQUAL  
1316 012316 104001 ERROR 1 ICLOCK OVERFLOW FAILED TO START A CONVERSION
```



```
1318      ;/*****  
(3)      ;TEST 136      A TO D PRE-INTERRUPT SETUP  
(3)      ;/*****  
(2) 012320 000000      TST136: SCOPE  
(1) 012322 012737 000001 001164      MOV      01,STIMES      ;/DO 1 ITERATION  
1319 012330 042737 177437 001510      BIC      0177437,ADDR1      ;MASK TO BITS  
1320 012336 001001      BNE      .+4  
1321 012340 000000      HALT                                ;BR LEVEL INDICATED IS 0  
1322 012342 022737 000340 001510      CMP      0340,ADDR1      ;IS IT BR LEVEL 7  
1323 012350 001001      BNE      .+4  
1324 012352 000000      HALT                                ;BR LEVEL INDICATED IS 7  
1325 012354 013737 001510 001572      MOV      ADDR1,BRLEV1  
1326 012362 162737 000040 001572      SUB      040,BRLEV1  
1327 012370 013737 001510 001574      MOV      ADDR1,BRLEV2  
1328  
1329      ;/*****  
(3)      ;TEST 137      TEST THAT A TO D INTERRUPTS AT LEVEL INDICATED -1  
(3)      ;/*****  
(2) 012376 000004      TST137: SCOPE  
(1) 012400 012737 000040 001164      MOV      040,STIMES      ;/DO 40 ITERATIONS  
1330 012406 005077 167104      CLR      0ADCS      ;CLEAR A TO D STATUS  
1331 012412 017700 167102      MOV      0ADDBR,R0      ;READ ADDRUFF  
1332 012416 012777 012454 167112      MOV      010,0ADINT      ;LOAD RETURN ADDRESS  
1333 012424 013737 001572 177776      MOV      BRLEV1,PSW      ;BR LEVEL -1  
1334 012432 052777 000101 167056      BIS      0BIT6,0BITS,0ADCS      ;SET THE AD GO AND INTERRUPT ENABLE BIT  
1335 012440 012700 000100      MOV      0100,R0  
1336 012444 005300      DEC      R0      ;DELAY  
1337 012446 001376      BNE      .-2  
1338 012450 104002      ERROR 2      ;ERROR, A TO D FAILED TO INTERRUPT  
1339 012452 000403      BR      TST140      ;/NEXT TEST  
1340 012454 005077 167036 181      CLR      0ADCS      ;CLEAR ERROR BIT  
1341 012460 022626      CMP      (SP)+,(SP)+  
1342
```

```
1344
1345
(3)
(3)
(2) 012462 000004
(1) 012464 012737 000040 001164
1346 012472 005077 167020
1347 012476 017700 167016
1348 012502 013737 001974 177776
1349 012510 012777 012536 167020
1350 012516 052777 000101 166772
1351 012524 012700 001000
1352 012530 005300
1353 012532 001376
1354 012534 000405
1355 012536 005077 166754 181
1356 012542 022626
1357 012544 104002
1358 012546 000423
1359 012550 012777 012576 166760 381
1360 012556 012700 001000
1361 012562 005037 177776
1362 012566 005300
1363 012570 001376
1364 012572 104002
1365 012574 000410
1366 012576 022626 281
1367 012600 005077 166712
1368 012604 013777 001540 166724
1369 012612 005077 166722
1370
1371
(3)
(3)
(2) 012616 000004
(1) 012620 012737 000040 001164
1372 012626 005037 001124
1373 012632 012777 027400 166656
1374 012640 000305
1375 012642 017737 166650 001126
1376 012650 001401
1377 012652 104001
1378

//*****
//TEST 140 TEST THAT THE A TO D DOES NOT INTERRUPT AT LEVEL INDICATED
//*****
TST140: SCOPE
MOV 040,STIMES //DO 40 ITERATIONS
CLR 0ADCS //CLEAR A TO D STATUS
MOV 0ADDR,R0 //READ ADDR
MOV 0RLEV2,PSW //R LEVEL
MOV 010,0ADINT //LOAD RETURN ADDRESS
BIS 0BIT6,0BITS,0ADCS //SET AD GO AND INTERRUPT ENABLE BIT
MOV 01000,R0
DEC R0 //DELAY
BNE ,-2
BR 30 //NEXT SUB-TEST
CLR 0ADCS //CLEAR ERROR FLAG
CMP (SP)+,(SP)+
ERROR 2 //A TO D INTERRUPTED ON LEVEL INDICATED
BR TST141 //NEXT TEST
MOV 020,0ADINT //RELOAD VECTOR
MOV 01000,R0 //LOAD COUNT
CLR PSW //LOWER PSW
DEC R0
BNE ,-2 //DELAY
ERROR 2 //LOWERING PSW FAILED TO ALLOW AD INTERRUPT
BR TST141 //NEXT TEST
CMP (SP)+,(SP)+
CLR 0ADCS
MOV ADINT1,0ADINT
CLR 0ADINT1

//*****
//TEST 141 TEST THAT RESET CLEARS MUX AND UNIPOLAR BITS
//*****
TST141: SCOPE
MOV 040,STIMES //DO 40 ITERATIONS
CLR 0GDDAT //CLEAR EXPECTED
MOV 0BIT13,0BIT11,0BIT10,0BIT9,0BITS,0ADCS
RESET
MOV 0ADCS,0GDDAT //READ REG
BEG TST142 //R IF CLEARED
ERROR 1 //ERROR, RESET FAILED TO CLEAR MUX BITS
```



```
1380
1381      ;*****
(3)      ;TEST 142      TEST THAT RESET CLEARS EXT AND INTERRUPT ENABLE BITS
(3)      ;*****
(2) 012654 000004      TST142: SCOPE
(1) 012656 012737 000040 001164      MOV      040,STIMES      ;100 40 ITERATIONS
1382 012664 012777 000120 166624      MOV      0BIT6,RIY4,0ADCS
1383 012672 005037 001124      CLR      SGDDAT      ;CLEAR EXPECTED
1384 012676 000005      RESET
1385 012700 017737 166612 001126      MOV      0ADCS,SDDAT      ;READ REG
1386 012706 001401      BEQ      TST143      ;BR IF CLEARED
1387 012710 104001      ERROR    1      ;ERROR, RESET FAILED TO CLEAR ENABLES
1388
1389      ;*****
(3)      ;TEST 143      TEST THAT RESET CLEARS AD DONE
(3)      ;*****
(2) 012712 000004      TST143: SCOPE
(1) 012714 012737 000040 001164      MOV      040,STIMES      ;100 40 ITERATIONS
1390 012722 012777 000001 166566      MOV      0BIT0,0ADCS
1391 012730 005037 001124      CLR      SGDDAT
1392 012734 105777 166556      TSTB    0ADCS
1393 012740 100375      BPL      =4
1394 012742 000005      RESET
1395 012744 017737 166546 001126      MOV      0ADCS,SDDAT
1396 012752 001401      BEQ      TST144      ;BR IF CLEARED
1397 012754 104001      ERROR    1      ;ERROR, RESET FAILED TO CLEAR DONE FLAG
1398
(3)      ;TEST 144      TEST THAT RESET CLEARS AD BUFFER REG
(3)      ;*****
(2) 012756 000004      TST144: SCOPE
(1) 012760 012737 000040 001164      MOV      040,STIMES      ;100 40 ITERATIONS
1399 012766 012777 000001 166522      MOV      01,0ADCS      ;CONVERT
1400 012774 105777 166516      TSTB    0ADCS      ;WAIT
1401 013000 100375      BPL      10
1402 013002 000005      RESET
1403 013004 017737 166510 001126      MOV      0ADDBR,SDDAT      ;READ REGISTER
1404 013012 005037 001124      CLR      SGDDAT      ;LOAD EXPECTED
1405 013016 023737 001124 001126      CMP      SGDDAT,SDDAT      ;COMPARE
1406 013024 001401      BEQ      TST145      ;BR IF CLEARED
1407 013026 104016      ERROR    16      ;RESET FAILED TO CLEAR AD BUFFER REGISTER
```

```

1409          ;*****
(3)          ;TEST 145      LOAD DIFFERENT NUMBERS INTO DIFFERENT REG.
(3)          ;*****
(2) 013030 000004          TST145: SCOPE
(1) 013032 012737 000020 001164      MOV      #20,STIMES      ;DO 20 ITERATIONS
1410 013040 000009          RESET
1411 013042 013700 001934          MOV      CSC,R0          ;LOAD STARTING ADDRESS
1412 013046 062700 000002          ADD      #2,R0          ;ADJUST ADDRESS
1413 013052 012701 013162          MOV      @BUPNUM+20,R1  ;LOAD STARTING TABLE ADDRESS
1414 013056 012702 000010          MOV      @B.,R2        ;LOAD COUNT
1415 013062 014140          18:  MOV      -(R1),-(R0)    ;LOAD THE REG
1416 013064 005302          DEC      R2             ;DONE ALL
1417 013066 001375          BNE     19             ;BR IF NOT
1418
1419 013070 013700 001916          MOV      ADCB,R0        ;LOAD STARTING POINTER
1420 013074 012701 013142          MOV      @BUPNUM,R1    ;LOAD STARTING POINTER <EXPECTED>
1421 013100 012702 000010          MOV      @B.,R2        ;LOAD # OF REG
1422
1423 013104 011137 001124          38:  MOV      (R1),SGDDAT  ;READ REG
1424 013110 011037 001126          MOV      (R0),SBDDAT  ;READ REG
1425 013114 023737 001124 001126  CMP      SGDDAT,SBDDAT ;COMPARE
1426 013122 001403          BEQ     48             ;BR IF EQUAL
1427 013124 010037 013162          MOV      R0,BUFADR     ;SAVE BUS ADDRESS
1428 013130 104017          ERROR  17            ;INCORRECT DATA, REG. WAS CHANGED IN ERROR
1429 013132 022021          48:  CMP      (R0)+,(R1)+ ;UPDATE POINTERS
1430 013134 005302          DEC      R2             ;DONE ALL REG
1431 013136 001362          BNE     38            ;BR IF NOT
1432 013140 000411          BR      TST146        ;NEXT TEST
1433
1434          ;
1435          ;
1436 013142 027560          BUFNUM: 27560        ;A TO D STATUS
1437 013144 000000          0                    ;A TO D BUFFER
1438 013146 140736          140736              ;CLOCK STATUS
1439 013150 000377          377                  ;CLOCK PRESET
1440 013152 007214          7214                 ;VC STATUS
1441 013154 001292          1292                 ;VC X POS
1442 013156 000525          525                  ;VC Y POS
1443 013160 000377          377                  ;CLOCK COUNTER
1444
1445 013162 170400          BUFADR: 170400      ;BUS ADDRESS OF REG IN ERROR
1446
1447          ;*****
(3)          ;TEST 146      DETERMINE IF MORE AR11'S ARE TO BE TESTED
(3)          ;*****
(2) 013164 000004          TST146: SCOPE
1448 013166 009737 001960          BYPASS: TST      NBEXT
1449 013172 001411          BEQ     19            ;TEST IF ANY
1450 013174 062737 000020 001552  ADD      #20,ARBADD    ;BR IF NONE
1451 013202 062737 000020 001554  ADD      #20,ARBVCT    ;UPDATE DEVICE ADDRESS
1452 013210 005337 001960          DEC      NBEXT        ;UPDATE DEVICE VECTOR
1453 013214 000413          BR      BYPASS1      ;ANOTHER ONE ?
1454 013216 013737 001290 001552  18:  MOV      @BASE,ARBADD  ;BR IF ANOTHER
1455 013224 013737 001244 001554  MOV      @VECT1,ARBVCT ;RELOAD ADDRESS
1456 013232 013737 001956 001560  MOV      @NBEXT,NBEXT  ;RELOAD VECTOR
1457 013240 000137 013254          JMP     SEOP          ;RELOAD NUMBER
                          ;DONE

```



1458 013244 012700 177777  
1459 013250 000137 002170  
1460

BYPAS1: MOV 0-1,R0  
JMP RREG2 ;TEST ANOTHER UNIT

(1) .SBTTL END OF PASS ROUTINE

(1) ;\*\*\*\*\*  
(1) ;INCREMENT THE PASS NUMBER (SPASS)  
(1) ;INDICATE END-OF-PROGRAM AFTER 1 PASSES THRU THE PROGRAM  
(1) ;TYPE "END PASS 0XXXXX" (WHERE XXXXX IS A DECIMAL NUMBER)  
(1) ;IF THERES A MONITOR GO TO IT  
(1) ;IF THERE ISN'T JUMP TO BYPAS1

(1) 013254  
(1) 013254 000004  
(1) 013256 005037 001102  
(1) 013262 005037 001164  
(1) 013266 005237 001202  
(1) 013272 042737 100000 001202  
(1) 013300 005327  
(1) 013302 000001  
(1) 013304 003022  
(1) 013306 012737  
(1) 013310 000001  
(1) 013312 013302  
(1) 013314 104400 013361  
(2) 013320 013746 001202  
(2) 013324 104404  
(1) 013326 104400 013356  
(1) 013332 013700 000042  
(1) 013336 001405  
(1) 013340 000005  
(1) 013342 004710  
(1) 013344 000240  
(1) 013346 000240  
(1) 013350 000240  
(1) 013352  
(1) 013352 000137  
(1) 013354 013244  
(1) 013356 377 377 000  
(1) 013361 015 042412 042116  
(1) 013366 050040 051501 020123  
(1) 013374 000043  
1461

SEOP1  
SCOPE  
CLR STSTNM ;ZERO THE TEST NUMBER  
CLR STIMES ;ZERO THE NUMBER OF ITERATIONS  
INC SPASS ;INCREMENT THE PASS NUMBER  
BIC 0100000,SPASS ;DON'T ALLOW A NEG. NUMBER  
DEC (PC)+ ;LOOP?  
SEOPCT: .WORD 1  
BST SDOAGN ;YES  
MOV (PC)+,0(PC)+ ;RESTORE COUNTER  
SENDCT: .WORD 1  
SEOPCT  
TYPE ,SENDMG ;TYPE "END PASS 0"  
MOV SPASS,-(SP) ;SAVE SPASS FOR TYPEOUT  
TYPDB ;GO TYPE--DECIMAL ASCII WITH SIGN  
TYPE ,SENULL ;TYPE A NULL CHARACTER  
SGET42: MOV 0042,R0 ;GET MONITOR ADDRESS  
BEQ SDOAGN ;BRANCH IF NO MONITOR  
RESET ;CLEAR THE WORLD  
SENDAD: JBR PC,(R0) ;GO TO MONITOR  
NOP ;SAVE ROOM  
NOP ;FOR  
NOP ;ACT11  
SDOAGN: JMP 0(PC)+ ;RETURN  
SRTNAD: .WORD BYPAS1  
SENULL: .BYTE -1,-1,0 ;NULL CHARACTER STRING  
SENDMG: .ASCIZ <15><12>/END PASS 0/





(1)	013554	116663	000001	177777		MOVB	1(SP),-1(R3)	;;YES--SET THE SIGN
(1)	013562	052702	000060		681	BIS	0'0,R2	;;MAKE THE BCD DIGIT ASCII
(1)	013566	052702	000040		781	BIS	0' ,R2	;;MAKE IT A SPACE IF NOT ALREADY A DIGIT
(1)	013572	110223				MOVB	R2,(R3)+	;;PUT THIS CHARACTER IN THE OUTPUT BUFFER
(1)	013574	005720				TST	(R0)+	;;JUST INCREMENTING
(1)	013576	020027	000010			CMP	R0,010	;;CHECK THE TABLE INDEX
(1)	013602	002746				BLT	28	;;GO DO THE NEXT DIGIT
(1)	013604	003002				BGT	88	;;GO TO EXIT
(1)	013606	010502				MOV	R5,R2	;;GET THE LSD
(1)	013610	000764				BR	68	;;GO CHANGE TO ASCII
(1)	013612	105726			881	TSTB	(SP)+	;;WAS THE LSD THE FIRST NON-ZERO?
(1)	013614	100003				BPL	98	;;BR IF NO
(1)	013616	116663	177777	177776		MOVB	-1(SP),-2(R3)	;;YES--SET THE SIGN FOR TYPING
(1)	013624	105013			981	CLRB	(R3)	;;SET THE TERMINATOR
(3)	013626	012605				NOV	(SP)+,R5	;;POP STACK INTO R5
(3)	013630	012603				NOV	(SP)+,R3	;;POP STACK INTO R3
(3)	013632	012602				NOV	(SP)+,R2	;;POP STACK INTO R2
(3)	013634	012601				NOV	(SP)+,R1	;;POP STACK INTO R1
(3)	013636	012600				NOV	(SP)+,R0	;;POP STACK INTO R0
(1)	013640	104400	013666			TYPE	,SDBLK	;;NOW TYPE THE NUMBER
(1)	013644	016666	000002	000004		NOV	2(SP),4(SP)	;;ADJUST THE STACK
(1)	013652	012616				NOV	(SP)+,(SP)	
(1)	013654	000002				RTI		;;RETURN TO USER
(1)	013656	023420			SDBL:	10000.		
(1)	013660	001750				1000.		
(1)	013662	000144				100.		
(1)	013664	000012				10.		
(1)	013666	000004			SDBLK:	.BLKW	4	

```

1481          ;SUBROUTINE TO TEST THE CLOCK REPEATIBILITY
1482          ;FIRST CLEAR CLOCK STATUS AND PRESET BUFFER
1483          ; THEN ENABLE THE CLOCK TO COUNT AT A RATE.
1484          ; DECREMENT R0 FOR SOME PERIOD OF TIME, WHEN R0 = 0
1485          ; SAVE THE COUNTER VALUE AND REPEAT THIS OPERATION AGAIN
1486          ;THEN COMPARE THE FIRST TIMED VALUE TO THE SECOND TIMED VALUE
1487          ; <MACHINE AND MEMORY TIMING NOT IMPORTANT>
1488          ; TO BE WITHIN THE VALUE SPECIFIED BY LOCATION CNTDEV
1489          ; IF GREATER THAN EXPECTED IT IS AN ERROR.
1490          ; ALSO TEST THAT THE COUNTER HAS REACHED A MIN. COUNT
1491
1492          013676 005077 165620          REPEAT: CLR          0CSR          ;STOP THE CLOCK
1493          013702 005077 165616          CLR          0CSR          ;CLEAR THE BUFFER
1494          013706 012537 014056          MOV          (R5)+,RATE    ;SET UP RATE
1495          013712 012537 014060          MOV          (R5)+,CNTDEV  ;SET UP CNT. DEV
1496          013716 012537 014062          MOV          (R5)+,MINCNT  ;SET UP MIN COUNT
1497          013722 012537 014064          MOV          (5)+,CKDLY    ;SAVE DELAY
1498          013726 004737 014016          JSR          PC,108        ;DUMMY TO CHARGE THE "CACHE"
1499          013732 004737 014016          JSR          PC,108        ;ENABLE THE CLOCK
1500          013736 010037 001124          MOV          R0,SGDDAT     ;SAVE FIRST TIME
1501          013742 004737 014016          JSR          PC,108        ;ENABLE THE CLOCK
1502          013746 010037 001126          MOV          R0,SGDDAT     ;SAVE SECOND TIME
1503          013752 013700 001124          MOV          SGDDAT,R0     ;GET FIRST TIME AGAIN
1504          013756 163700 001126          SUB          SGDDAT,R0     ;SUBTRACT SECOND TIME
1505          013762 100001
1506          013764 005400
1507          013766 023737 001126 014062 381  CMP          SGDDAT,MINCNT  ;MAGNITUDE OF DIFFERENCE IN R0
1508          013774 002004
1509          013776 013737 014062 001124  BGE          48            ;COMPARE TO MIN. COUNT
1510          014004 000205
1511          014006 005725
1512          014010 020037 014060          MOV          MINCNT,SGDDAT ;BRANCH IF GREATER
1513          014014 000205
1514
1515          014016 013700 014064          RTS          R5            ;LOAD SGDDAT FOR TYPE-OUT
1516          014022 013777 014056 165472 881  TST          (R5)+
1517          014030 005277 165466          CMP          R0,CNTDEV    ;UPDATE THE STACK
1518          014034 005300
1519          014036 001376
1520          014040 017700 165470          CMP          R0,CNTDEV    ;COMPARE TO DEVIATION
1521          014044 005077 165452          MOV          CKDLY,R0
1522          014050 005077 165450          MOV          RATE,0CSR
1523          014054 000207
1524
1525          014056 000000          INC          0CSR
1526          014060 000000          DEC          R0
1527          014062 000000          BNE          18
1528          014064 000000          MOV          0CSC,R0
1529
1530          014066 000000          CLR          0CSR
1531          014068 000000          CLR          0CSR
1532          014070 000000          RTS          PC
1533
1534          014072 000000          ;GET DELAY
1535          014074 000000          ;LOAD RATE
1536          014076 000000          ;ENABLE CLOCK
1537          014078 000000          ;DELAY
1538
1539          014080 000000          ;READ COUNTER
1540          014082 000000          ;STOP CLOCK
1541          014084 000000          ;CLEAR BUFFER
1542          014086 000000          ;EXIT
1543
1544          014088 000000          ;CLOCK RATE
1545          014090 000000          ;CLOCK DEV.
1546          014092 000000          ;MIN. COUNT
1547          014094 000000
1548          014096 000000
1549          014098 000000
1550          014100 000000
1551          014102 000000
1552          014104 000000
1553          014106 000000
1554          014108 000000
1555          014110 000000
1556          014112 000000
1557          014114 000000
1558          014116 000000
1559          014118 000000
1560          014120 000000
1561          014122 000000
1562          014124 000000
1563          014126 000000
1564          014128 000000
1565          014130 000000
1566          014132 000000
1567          014134 000000
1568          014136 000000
1569          014138 000000
1570          014140 000000
1571          014142 000000
1572          014144 000000
1573          014146 000000
1574          014148 000000
1575          014150 000000
1576          014152 000000
1577          014154 000000
1578          014156 000000
1579          014158 000000
1580          014160 000000
1581          014162 000000
1582          014164 000000
1583          014166 000000
1584          014168 000000
1585          014170 000000
1586          014172 000000
1587          014174 000000
1588          014176 000000
1589          014178 000000
1590          014180 000000
1591          014182 000000
1592          014184 000000
1593          014186 000000
1594          014188 000000
1595          014190 000000
1596          014192 000000
1597          014194 000000
1598          014196 000000
1599          014198 000000
1600          014200 000000
1601          014202 000000
1602          014204 000000
1603          014206 000000
1604          014208 000000
1605          014210 000000
1606          014212 000000
1607          014214 000000
1608          014216 000000
1609          014218 000000
1610          014220 000000
1611          014222 000000
1612          014224 000000
1613          014226 000000
1614          014228 000000
1615          014230 000000
1616          014232 000000
1617          014234 000000
1618          014236 000000
1619          014238 000000
1620          014240 000000
1621          014242 000000
1622          014244 000000
1623          014246 000000
1624          014248 000000
1625          014250 000000
1626          014252 000000
1627          014254 000000
1628          014256 000000
1629          014258 000000
1630          014260 000000
1631          014262 000000
1632          014264 000000
1633          014266 000000
1634          014268 000000
1635          014270 000000
1636          014272 000000
1637          014274 000000
1638          014276 000000
1639          014278 000000
1640          014280 000000
1641          014282 000000
1642          014284 000000
1643          014286 000000
1644          014288 000000
1645          014290 000000
1646          014292 000000
1647          014294 000000
1648          014296 000000
1649          014298 000000
1650          014300 000000
1651          014302 000000
1652          014304 000000
1653          014306 000000
1654          014308 000000
1655          014310 000000
1656          014312 000000
1657          014314 000000
1658          014316 000000
1659          014318 000000
1660          014320 000000
1661          014322 000000
1662          014324 000000
1663          014326 000000
1664          014328 000000
1665          014330 000000
1666          014332 000000
1667          014334 000000
1668          014336 000000
1669          014338 000000
1670          014340 000000
1671          014342 000000
1672          014344 000000
1673          014346 000000
1674          014348 000000
1675          014350 000000
1676          014352 000000
1677          014354 000000
1678          014356 000000
1679          014358 000000
1680          014360 000000
1681          014362 000000
1682          014364 000000
1683          014366 000000
1684          014368 000000
1685          014370 000000
1686          014372 000000
1687          014374 000000
1688          014376 000000
1689          014378 000000
1690          014380 000000
1691          014382 000000
1692          014384 000000
1693          014386 000000
1694          014388 000000
1695          014390 000000
1696          014392 000000
1697          014394 000000
1698          014396 000000
1699          014398 000000
1700          014400 000000
1701          014402 000000
1702          014404 000000
1703          014406 000000
1704          014408 000000
1705          014410 000000
1706          014412 000000
1707          014414 000000
1708          014416 000000
1709          014418 000000
1710          014420 000000
1711          014422 000000
1712          014424 000000
1713          014426 000000
1714          014428 000000
1715          014430 000000
1716          014432 000000
1717          014434 000000
1718          014436 000000
1719          014438 000000
1720          014440 000000
1721          014442 000000
1722          014444 000000
1723          014446 000000
1724          014448 000000
1725          014450 000000
1726          014452 000000
1727          014454 000000
1728          014456 000000
1729          014458 000000
1730          014460 000000
1731          014462 000000
1732          014464 000000
1733          014466 000000
1734          014468 000000
1735          014470 000000
1736          014472 000000
1737          014474 000000
1738          014476 000000
1739          014478 000000
1740          014480 000000
1741          014482 000000
1742          014484 000000
1743          014486 000000
1744          014488 000000
1745          014490 000000
1746          014492 000000
1747          014494 000000
1748          014496 000000
1749          014498 000000
1750          014500 000000
1751          014502 000000
1752          014504 000000
1753          014506 000000
1754          014508 000000
1755          014510 000000
1756          014512 000000
1757          014514 000000
1758          014516 000000
1759          014518 000000
1760          014520 000000
1761          014522 000000
1762          014524 000000
1763          014526 000000
1764          014528 000000
1765          014530 000000
1766          014532 000000
1767          014534 000000
1768          014536 000000
1769          014538 000000
1770          014540 000000
1771          014542 000000
1772          014544 000000
1773          014546 000000
1774          014548 000000
1775          014550 000000
1776          014552 000000
1777          014554 000000
1778          014556 000000
1779          014558 000000
1780          014560 000000
1781          014562 000000
1782          014564 000000
1783          014566 000000
1784          014568 000000
1785          014570 000000
1786          014572 000000
1787          014574 000000
1788          014576 000000
1789          014578 000000
1790          014580 000000
1791          014582 000000
1792          014584 000000
1793          014586 000000
1794          014588 000000
1795          014590 000000
1796          014592 000000
1797          014594 000000
1798          014596 000000
1799          014598 000000
1800          014600 000000
1801          014602 000000
1802          014604 000000
1803          014606 000000
1804          014608 000000
1805          014610 000000
1806          014612 000000
1807          014614 000000
1808          014616 000000
1809          014618 000000
1810          014620 000000
1811          014622 000000
1812          014624 000000
1813          014626 000000
1814          014628 000000
1815          014630 000000
1816          014632 000000
1817          014634 000000
1818          014636 000000
1819          014638 000000
1820          014640 000000
1821          014642 000000
1822          014644 000000
1823          014646 000000
1824          014648 000000
1825          014650 000000
1826          014652 000000
1827          014654 000000
1828          014656 000000
1829          014658 000000
1830          014660 000000
1831          014662 000000
1832          014664 000000
1833          014666 000000
1834          014668 000000
1835          014670 000000
1836          014672 000000
1837          014674 000000
1838          014676 000000
1839          014678 000000
1840          014680 000000
1841          014682 000000
1842          014684 000000
1843          014686 000000
1844          014688 000000
1845          014690 000000
1846          014692 000000
1847          014694 000000
1848          014696 000000
1849          014698 000000
1850          014700 000000
1851          014702 000000
1852          014704 000000
1853          014706 000000
1854          014708 000000
1855          014710 000000
1856          014712 000000
1857          014714 000000
1858          014716 000000
1859          014718 000000
1860          014720 000000
1861          014722 000000
1862          014724 000000
1863          014726 000000
1864          014728 000000
1865          014730 000000
1866          014732 000000
1867          014734 000000
1868          014736 000000
1869          014738 000000
1870          014740 000000
1871          014742 000000
1872          014744 000000
1873          014746 000000
1874          014748 000000
1875          014750 000000
1876          014752 000000
1877          014754 000000
1878          014756 000000
1879          014758 000000
1880          014760 000000
1881          014762 000000
1882          014764 000000
1883          014766 000000
1884          014768 000000
1885          014770 000000
1886          014772 000000
1887          014774 000000
1888          014776 000000
1889          014778 000000
1890          014780 000000
1891          014782 000000
1892          014784 000000
1893          014786 000000
1894          014788 000000
1895          014790 000000
1896          014792 000000
1897          014794 000000
1898          014796 000000
1899          014798 000000
1900          014800 000000
1901          014802 000000
1902          014804 000000
1903          014806 000000
1904          014808 000000
1905          014810 000000
1906          014812 000000
1907          014814 000000
1908          014816 000000
1909          014818 000000
1910          014820 000000
1911          014822 000000
1912          014824 000000
1913          014826 000000
1914          014828 000000
1915          014830 000000
1916          014832 000000
1917          014834 000000
1918          014836 000000
1919          014838 000000
1920          014840 000000
1921          014842 000000
1922          014844 000000
1923          014846 000000
1924          014848 000000
1925          014850 000000
1926          014852 000000
1927          014854 000000
1928          014856 000000
1929          014858 000000
1930          014860 000000
1931          014862 000000
1932          014864 000000
1933          014866 000000
1934          014868 000000
1935          014870 000000
1936          014872 000000
1937          014874 000000
1938          014876 000000
1939          014878 000000
1940          014880 000000
1941          014882 000000
1942          014884 000000
1943          014886 000000
1944          014888 000000
1945          014890 000000
1946          014892 000000
1947          014894 000000
1948          014896 000000
1949          014898 000000
1950          014900 000000
1951          014902 000000
1952          014904 000000
1953          014906 000000
1954          014908 000000
1955          014910 000000
1956          014912 000000
1957          014914 000000
1958          014916 000000
1959          014918 000000
1960          014920 000000
1961          014922 000000
1962          014924 000000
1963          014926 000000
1964          014928 000000
1965          014930 000000
1966          014932 000000
1967          014934 000000
1968          014936 000000
1969          014938 000000
1970          014940 000000
1971          014942 000000
1972          014944 000000
1973          014946 000000
1974          014948 000000
1975          014950 000000
1976          014952 000000
1977          014954 000000
1978          014956 000000
1979          014958 000000
1980          014960 000000
1981          014962 000000
1982          014964 000000
1983          014966 000000
1984          014968 000000
1985          014970 000000
1986          014972 000000
1987          014974 000000
1988          014976 000000
1989          014978 000000
1990          014980 000000
1991          014982 000000
1992          014984 000000
1993          014986 000000
1994          014988 000000
1995          014990 000000
1996          014992 000000
1997          014994 000000
1998          014996 000000
1999          014998 000000
2000          015000 000000
2001          015002 000000
2002          015004 000000
2003          015006 000000
2004          015008 000000
2005          015010 000000
2006          015012 000000
2007          015014 000000
2008          015016 000000
2009          015018 000000
2010          015020 000000
2011          015022 000000
2012          015024 000000
2013          015026 000000
2014          015028 000000
2015          015030 000000
2016          015032 000000
2017          015034 000000
2018          015036 000000
2019          015038 000000
2020          015040 000000
2021          015042 000000
2022          015044 000000
2023          015046 000000
2024          015048 000000
2025          015050 000000
2026          015052 000000
2027          015054 000000
2028          015056 000000
2029          015058 000000
2030          015060 000000
2031          015062 000000
2032          015064 000000
2033          015066 000000
2034          015068 000000
2035          015070 000000
2036          015072 000000
2037          015074 000000
2038          015076 000000
2039          015078 000000
2040          015080 000000
2041          015082 000000
2042          015084 000000
2043          015086 000000
2044          015088 000000
2045          015090 000000
2046          015092 000000
2047          015094 000000
2048          015096 000000
2049          015098 000000
2050          015100 000000
2051          015102 000000
2052          015104 000000
2053          015106 000000
2054          015108 000000
2055          015110 000000
2056          015112 000000
2057          015114 000000
2058          015116 000000
2059          015118 000000
2060          015120 000000
2061          015122 000000
2062          015124 000000
2063          015126 000000
2064          015128 000000
2065          015130 000000
2066          015
```



1530	014066	020101	047524	042040	EM11	.ASCIZ /A TO D STATUS REGISTER IN ERROR/
	014074	051440	040524	052524		
	014102	020123	042522	044507		
	014110	052123	051105	044440		
	014116	020116	051105	047522		
	014124	000122				
1531	014126	020101	047524	042040	EM21	.ASCIZ /A TO D INTERRUPT ERROR/
	014134	044440	052116	051105		
	014142	052522	052120	042440		
	014150	051122	051117	000		
1532	014155	103	047514	045503	EM31	.ASCIZ /CLOCK STATUS REGISTER IN ERROR/
	014162	051440	040524	052524		
	014170	020123	042522	044507		
	014176	052123	051105	044440		
	014204	020116	051105	047522		
	014212	000122				
1533	014214	046103	041517	020113	EM41	.ASCIZ /CLOCK PRESET REGISTER IN ERROR/
	014222	051120	051505	052105		
	014230	051040	043505	051511		
	014236	042524	020122	047111		
	014244	042440	051122	051117		
	014252	000				
1534	014253	103	047514	045503	EM51	.ASCIZ /CLOCK INTERRUPT ERROR/
	014260	044440	052116	051105		
	014266	052522	052120	042440		
	014274	051122	051117	000		
1535	014301	103	047514	045503	EM61	.ASCIZ /CLOCK COUNTER REGISTER IN ERROR/
	014306	041440	052517	052116		
	014314	051105	051040	043505		
	014322	051511	042524	020122		
	014330	047111	042440	051122		
	014336	051117	000			
1536	014341	103	047514	045503	EM71	.ASCIZ /CLOCK COUNTED IN ERROR/
	014346	041440	052517	052116		
	014354	042105	044440	020116		
	014362	051105	047522	000122		
1537	014370	046103	041517	020113	EM101	.ASCIZ /CLOCK REPEATABILITY FAILED/
	014376	042522	042520	052101		
	014404	041101	046111	052111		
	014412	020131	040506	046111		
	014420	042105	000			
1538						
1539	014423	126	020103	052123	EM111	.ASCIZ /VC STATUS REGISTER IN ERROR/
	014430	052101	051525	051040		
	014436	043505	051511	042524		
	014444	020122	047111	042440		
	014452	051122	051117	000		
1540	014457	130	051040	043505	EM121	.ASCIZ /X REGISTER IN ERROR/
	014464	051511	042524	020122		
	014472	047111	042440	051122		
	014500	051117	000			
1541	014503	131	051040	043505	EM131	.ASCIZ /V REGISTER IN ERROR/
	014510	051511	042524	020122		
	014516	047111	042440	051122		
	014524	051117	000			
1542	014527	123	047503	042520	EM141	.ASCIZ /SCOPE INTERRUPT ERROR/

	014534	044440	052116	051105					
	014542	052522	052120	042440					
	014550	051122	051117	000					
1543	014555	104	053105	041511	EM151	.ASCIZ	/DEVICE BUS ERROR/		
	014562	020105	052502	020123					
	014570	051105	047522	000122					
1544									
1545	014576	047111	047503	051122	EM161	.ASCIZ	"INCORRECT A/D BUFFER DATA"		
	014604	041505	020124	027501					
	014612	020104	052502	043106					
	014620	051105	042040	052101					
	014626	000101							
1546	014630	044103	047101	042507	EM171	.ASCIZ	/CHANGED REGISTER IN ERROR/		
	014636	020104	042522	044507					
	014644	052123	051105	044440					
	014652	020116	051105	047522					
	014660	000122							
1547	014662	051105	050122	020103	DM11	.ASCIZ	/ERRPC ADCS ADSTAT EXPECTED/		
	014670	020040	040440	041504					
	014676	020123	020040	042101					
	014704	052123	052101	042440					
	014712	050130	041505	042524					
	014720	000104							
1548	014722	051105	050122	020103	DM21	.ASCIZ	/ERRPC ADCS/		
	014730	020040	040440	041504					
	014736	000123							
1549	014740	051105	050122	020103	DM31	.ASCIZ	/ERRPC CSR CKSTAT EXPECTED/		
	014746	020040	041440	051123					
	014754	020040	020040	045503					
	014762	052123	052101	042440					
	014770	050130	041505	042524					
	014776	000104							
1550	015000	051105	050122	020103	DM41	.ASCIZ	/ERRPC CSB CKBUFF EXPECTED/		
	015006	020040	041440	041123					
	015014	020040	020040	045503					
	015022	052502	043106	042440					
	015030	050130	041505	042524					
	015036	000104							
1551	015040	051105	050122	020103	DM51	.ASCIZ	/ERRPC CSR/		
	015046	020040	041440	051123					
	015054	000							
1552	015055	105	051122	041520	DM61	.ASCIZ	/ERRPC CSC CKCNTR EXPECTED/		
	015062	020040	020040	051503					
	015070	020103	020040	041440					
	015076	041513	052116	020122					
	015104	054105	042520	052103					
	015112	042105	000						
1553	015115	105	051122	041520	DM101	.ASCIZ	/ERRPC CSC TIME1 TIME2/		
	015122	020040	020040	051503					
	015130	020103	020040	052040					
	015136	046511	030505	020040					
	015144	052040	046511	051105					
	015152	000							
1554	015153	105	051122	041520	DM111	.ASCIZ	/ERRPC VCADR VCSTAT EXPECTED/		
	015160	020040	053040	040503					
	015166	051104	020040	053040					



	015174	051503	040524	020124					
	015202	054105	042520	052103					
	015210	042105	000						
1555	015213	105	051122	041520	DM121	.ASCIZ	/ERRPC	VCXREG	X POS. EXPECTED/
	015220	020040	053040	054103					
	015226	042522	020107	054040					
	015234	050040	051517	020056					
	015242	042440	050130	041505					
	015250	042524	000104						
1556	015254	051105	050122	020103	DM131	.ASCIZ	/ERRPC	VCYREG	Y POS. EXPECTED/
	015262	020040	041526	051131					
	015270	043505	020040	020131					
	015276	047520	027123	020040					
	015304	054105	042520	052103					
	015312	042105	000						
1557	015315	105	051122	041520	DM141	.ASCIZ	/ERRPC	VCSTAT/	
	015322	020040	053040	051503					
	015330	040524	000124						
1558	015334	051105	050122	020103	DM151	.ASCIZ	/ERRPC	BASE	ACTUAL/
	015342	020040	041040	051501					
	015350	020105	020040	040440					
	015356	052103	040525	000114					
1559	015364	051105	050122	020103	DM161	.ASCIZ	/ERRPC	ADDR	READ EXPECTED/
	015372	020040	042101	041104					
	015400	020122	020040	051040					
	015406	040505	020104	020040					
	015414	054105	042520	052103					
	015422	042105	000						
1560	015425	105	051122	041520	DM171	.ASCIZ	/ERRPC	BUFADR	READ EXPECTED/
	015432	020040	041040	043125					
	015440	042101	020122	020040					
	015446	042522	042101	020040					
	015454	042440	050130	041505					
	015462	042524	000104						
1561						.EVEN			
1562	015466	001116	001516	001126	DT11	SERRPC,ADCS,SBDDAT,SGDDAT,0			
	015474	001124	000000						
1563	015500	001116	001516	000000	DT21	SERRPC,ADCS,0			
1564	015506	001116	001522	001126	DT31	SERRPC,CSR,SBDDAT,SGDDAT,0			
	015514	001124	000000						
1565	015520	001116	001524	001126	DT41	SERRPC,CSB,SBDDAT,SGDDAT,0			
	015526	001124	000000						
1566	015532	001116	001522	000000	DT51	SERRPC,CSR,0			
1567	015540	001116	001534	001126	DT61	SERRPC,CSC,SBDDAT,SGDDAT,0			
	015546	001124	000000						
1568	015552	001116	001526	001126	DT111	SERRPC,VCSTAT,SBDDAT,SGDDAT,0			
	015560	001124	000000						
1569	015564	001116	001530	001126	DT121	SERRPC,VCXREG,SBDDAT,SGDDAT,0			
	015572	001124	000000						
1570	015576	001116	001532	001126	DT131	SERRPC,VCYREG,SBDDAT,SGDDAT,0			
	015604	001124	000000						
1571	015610	001116	001526	000000	DT141	SERRPC,VCSTAT,0			
1572	015616	001116	001532	001126	DT151	SERRPC,ARBADD,SBDDAT,0			
	015624	000000							
1573	015626	001116	001520	001126	DT161	SERRPC,ADDR,SBDDAT,SGDDAT,0			
	015634	001124	000000						

MAINDEC-11-DZARA-B MACY11 27(732) 17-MAY-76 09149 PAGE 41-3  
DZARAB.P11 CONVERT BINARY TO DECIMAL AND TYPE ROUTINE

SEQ 0062

1574 015640 001110 013162 001126 DT17: SERRPC,BUFAOR,SDDAT,SGDDAT,0  
015646 001124 000000







```
(1) 016060 105237 001102          SSVLAD: INCB      STSTNM          ;;COUNT TEST NUMBERS
(1) 016064 113737 001102 001200      MOVB      STSTNM,STESTN  ;;SET TEST NUMBER IN APT MAILBOX
(1) 016072 011637 001106          MOV      (SP),SLPADR    ;;SAVE SCOPE LOOP ADDRESS
(1) 016076 011637 001110          MOV      (SP),SLPERR    ;;SAVE ERROR LOOP ADDRESS
(1) 016102 005037 001166          CLR      SEESCAPE      ;;CLEAR THE ESCAPE FROM ERROR ADDRESS
(1) 016106 112737 000001 001115      MOVB      01,SEEMAX     ;;ONLY ALLOW ONE(1) ERROR ON NEXT TEST
(1) 016114 013777 001102 163016      SOVER:  MOV      STSTNM,ODISPLAY ;;DISPLAY TEST NUMBER
(1) 016122 013716 001106          MOV      SLPADR,(SP)    ;;FUDGE RETURN ADDRESS
(1) 016126 000002          RTI          ;;FIXES PS
(1) 016130 003720          SMXCNT: 2000.         ;;MAX. NUMBER OF ITERATIONS
```

```
1570  
(1) .SBTTL ERROR HANDLER ROUTINE
```

```
(1) ;;*****  
(1) ;;THIS ROUTINE WILL INCREMENT THE ERROR FLAG AND THE ERROR COUNT,  
(1) ;;SAVE THE ERROR ITEM NUMBER AND THE ADDRESS OF THE ERROR CALL  
(1) ;;AND GO TO SERRTYP ON ERROR  
(1) ;;THE SWITCH OPTIONS PROVIDED BY THIS ROUTINE ARE:  
(1) ;;SW15=1 HALT ON ERROR  
(1) ;;SW13=1 INHIBIT ERROR TYPEOUTS  
(1) ;;SW09=1 LOOP ON ERROR  
(1) ;;CALL  
(1) ;; ERROR N ;;ERROR=EMT AND N=ERROR ITEM NUMBER
```

```
(1) 016132          SERROR:  
(1) 016132 105237 001103          700:  INCB      SERFLG          ;;SET THE ERROR FLAG  
(1) 016136 001775          BEQ      70          ;;DON'T LET THE FLAG GO TO ZERO  
(1) 016140 013777 001102 162772      MOV      STSTNM,ODISPLAY ;;DISPLAY TEST NUMBER AND ERROR FLAG  
(1) 016146 005237 001112          INC      SERTTL        ;;INC THE ERROR COUNT  
(1) 016152 011637 001116          MOV      (SP),SERRPC    ;;GET ADDRESS OF ERROR INSTRUCTION  
(1) 016156 162737 000002 001116      SUB      02,SERRPC  
(1) 016164 117737 162726 001114      MOVB      0SERRPC,SITEMB ;;STRIP AND SAVE THE ERROR ITEM CODE  
(1) 016172 032777 020000 162736      BIT      0BIT13,0SWR    ;;SKIP TYPEOUT IF SET  
(1) 016200 001004          BNE      200          ;;SKIP TYPEOUTS  
(1) 016202 004737 017106          JSR      PC,SERRTYP     ;;GO TO USER ERROR ROUTINE  
(1) 016206 104400 001171          TYPE      ,SCLRF
```

```
(1) 016212          200:  
(1) 016212 122737 000001 001214      CMPO     0APTENV,SENV   ;;RUNNING IN APT MODE  
(1) 016220 001007          BNE      20          ;;NO,SKIP APT ERROR REPORT  
(1) 016222 113737 001114 016234      MOVB      SITEMB,215   ;;SET ITEM NUMBER AS ERROR NUMBER  
(1) 016230 004737 020210          JSR      PC,0ATY4      ;;REPORT FATAL ERROR TO APT
```

```
(1) 016234          210:  .BYTE 0  
(1) 016235          .BYTE 0  
(1) 016236 000777          220:  BR      220          ;;APT ERROR LOOP  
(1) 016240 005777 162672          200:  TST      0SWR          ;;HALT ON ERROR  
(1) 016244 100001          BPL      30          ;;SKIP IF CONTINUE  
(1) 016246 000000          HALT          ;;HALT ON ERROR!
```

```
(1) 016250 032777 001000 162660          30:  BIT      0BIT09,0SWR   ;;LOOP ON ERROR SWITCH SET?  
(1) 016256 001402          BEQ      40          ;;BR IF NO  
(1) 016260 013716 001110          MOV      SLPERR,(SP)   ;;FUDGE RETURN FOR LOOPING  
(1) 016264 005737 001166          40:  TST      SEESCAPE      ;;CHECK FOR AN ESCAPE ADDRESS  
(1) 016270 001402          BEQ      50          ;;BR IF NONE  
(1) 016272 013716 001166          MOV      SEESCAPE,(SP) ;;FUDGE RETURN ADDRESS FOR ESCAPE
```

```
(1) 016276          50:  
(1) 016276 022737 013342 000042          CMP      0SENDAD,0042  ;;ACT-11 AUTO-ACCEPT?  
(1) 016304 001001          BNE      60          ;;BRANCH IF NO
```



```

(1) 016306 000000          HALT          IYES
(1) 016310          681          RTI          IRETURN
(1) 016310 000002
1579
(1)          .SBTTL  TTY INPUT ROUTINE
(1)
(2)          I*****
(1)          I*SOFTWARE SWITCH REGISTER CHANGE ROUTINE.
(1)          I*ROUTINE IS ENTERED FROM THE TRAP HANDLER, AND WILL
(1)          I*SERVICE THE TEST FOR CHANGE IN SOFTWARE SWITCH REGISTER TRAP CALL
(1)          I*WHEN OPERATING IN TTY FLAG MODE.
(1) 016312 022737 000176 001136 SCKSWR: CMP      @SWREG,SWR      IIS THE SOFT-SWR SELECTED?
(1) 016320 001073          BNE      148          IIBRANCH IF NO
(1) 016322 109777 162614          TSTB     @STKB          IICHR THERE?
(1) 016326 100070          BPL      148          IIF NO, DON'T WAIT AROUND
(1) 016330 117746 162610 281     MOVB     @STKB,-(SP)      IISAVE THE CHAR
(1) 016334 042716 177600          BIC     @C177,(SP)     IISTRIP-OFF THE ASCII
(1) 016340 022726 000007          CMP     @7,(SP)+      IIS IT A CONTROL G?
(1) 016344 001061          BNE     148          IINO, RETURN TO USER
(1) 016346 104400 016755          TYPE    .SCNTLG      IYES, ECHO CONTROL G
(1)
(1) 016352 104400 016762 681     TYPE    .SMWR          IITYPE CURRENT CONTENTS
(2) 016356 013746 000176          MOV     SWREG,-(SP)    IISAVE SWREG FOR TYPEOUT
(2) 016362 104401          TYPE    .SMNEW        IIPROMPT FOR NEW SWR
(1) 016364 104400 016773          CLR     -(SP)         IICLEAR COUNTER
(1) 016370 005046          CLR     -(SP)         IITHE NEW SWR
(1) 016372 005046          RDCHR          IISGET NEXT CHAR
(1) 016374 104406 781
(1) 016376 022716 000025 881     CMP     @25,(SP)      IIS IT A CONTROL UT
(1) 016402 001005          BNE     98          IIBRANCH IF NO
(1) 016404 104400 016750          TYPE    .SCNTLU      IYES, ECHO IT
(1) 016410 062706 000006          ADD     @6,SP         IIGNORE PREVIOUS INPUT
(1) 016414 000756          BR      68          IILET'S TRY IT AGAIN
(1)
(1) 016416 022716 000015 981     CMP     @15,(SP)     IIS IT A <CR>?
(1) 016422 001011          BNE     118          IIBRANCH IF NO
(1) 016424 009766 000004          TST     4(SP)         IYES, IS IT THE FIRST CHART
(1) 016430 001403          BEQ     108          IIBRANCH IF YES
(1) 016432 016677 000002 162476 1081    MOV     2(SP),@SWR    IISAVE NEW SWR
(1) 016440 062706 000006          ADD     @6,SP         IICLEAR UP STACK
(1) 016444 000017          BR      138          IIRETURN TO USER
(1) 016446 022716 000012 1181    CMP     @12,(SP)     IIS IT A <LF>
(1) 016452 001017          BNE     158          IIBRANCH IF NO
(1) 016454 009766 000004          TST     4(SP)         IYES, IS IT THE FIRST CHART
(1) 016460 001403          BEQ     128          IYES
(1) 016462 016677 000002 162446 1281    MOV     2(SP),@SWR    IISAVE NEW SWR
(1) 016470 062706 000006          ADD     @6,SP         IICLEAR UP STACK
(1) 016474 013716 000046          MOV     @046,(SP)     IISGET RESTART
(1) 016500 062716 000010          ADD     @10,(SP)     IIAADDRESS
(1) 016504 104400 001171 1381    TYPE    .SCRLF      IIECHO <CR> AND <LF>
(1) 016510 000002 1481    RTI          IIRETURN
(1) 016512 004737 020122 1581    JBR     PC,STYPEC    IIECHO CHAR
(1) 016516 042726 177770          BIC     @177770,(SP)+ IIRESTRICT TO 0-7
(1) 016522 009766 000002          TST     2(SP)         IIS THIS THE FIRST CHAR
(1) 016526 001403          BEQ     168          IIBRANCH IF YES
    
```



```

(1) 016530 C06316          ASL      (SP)          ;;NO, SHIFT PRESENT
(1) 016532 006316          ASL      (SP)          ;; CHAR OVER TO MAKE
(1) 016534 006316          ASL      (SP)          ;; ROOM FOR NEW ONE.
(1) 016536 005266 000002   168: INC      2(SP)          ;;KEEP COUNT OF CHAR
(1) 016542 056616 177776   RIS      -2(SP),(SP)   ;;SET IN NEW CHAR
(1) 016546 008712          BR       79            ;;GET THE NEXT ONE
(2)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1) 016550 011646          SRDCHR: MOV      (SP),-(SP)   ;;PUSH DOWN THE PC
(1) 016552 016666 000004 000002   MOV      4(SP),2(SP)   ;;SAVE THE PS
(1) 016560 105777 162356   18:  TSTB     0STKS      ;;WAIT FOR
(1) 016564 100375          BPL      19            ;;A CHARACTER
(1) 016566 117766 162352 000004   MOVB    0STKB,4(SP)   ;;READ THE TTY
(1) 016574 042766 177600 000004   BIC     0^C<177>,4(SP) ;;GET RID OF JUNK IF ANY
(1) 016602 026627 000004 000140   CMP     4(SP),0140    ;;IS IT UPPER CASE?
(1) 016610 002407          BLT      29            ;;BRANCH IF YES
(1) 016612 026627 000004 000175   CMP     4(SP),0175    ;;IS IT A SPECIAL CHAR?
(1) 016620 003003          BGT      29            ;;BRANCH IF YES
(1) 016622 042766 000040 000004   BIC     040,4(SP)     ;;MAKE IT UPPER CASE
(1) 016630 000002          28:  RTI              ;;GO BACK TO USER
(2)
(1)
(1)
(1)
(1)
(1)
(1) 016632 010346          SRDLIN: MOV      R3, -(SP)   ;;SAVE R3
(1) 016634 012703 016740   18:  MOV      0STYIN,R3   ;;GET ADDRESS
(1) 016640 022703 016750   28:  CMP      0STYIN+0,,R3  ;;BUFFER FULL?
(1) 016644 101405          BLOS     49            ;;OR IF YES
(1) 016646 104406          RDCHR    ;;GO READ ONE CHARACTER FROM THE TTY
(1) 016650 112613          MOVB    (SP)+,(R3)    ;;GET CHARACTER
(1) 016652 122713 000177   108: CMPB    0177,(R3)     ;;IS IT A RUBOUT
(1) 016656 001003          BNE     39            ;;SKIP IF NOT
(1) 016660 104400 001170   48:  TYPE     ,0QUES      ;;TYPE A '?'
(1) 016664 000763          BR       19            ;;CLEAR THE BUFFER AND LOOP
(1) 016666 111337 016736   38:  MOVB    (R3),09      ;;ECHO THE CHARACTER
(1) 016672 104400 016736          TYPE     ,09
(1) 016676 122723 000015          CMPB    015,(R3)+    ;;CHECK FOR RETURN
(1) 016702 001356          BNE     29            ;;LOOP IF NOT RETURN
(1) 016704 105063 177777          CLRB    -1(R3)        ;;CLEAR RETURN (THE 15)
(1) 016710 104400 001172          TYPE     ,SLF        ;;TYPE A LINE FEED
(1) 016714 012603          MOV     (SP)+,R3      ;;RESTORE R3
(1) 016716 011646          MOV     (SP),-(SP)    ;;ADJUST THE STACK AND PUT ADDRESS OF THE
(1) 016720 016666 000004 000002   MOV     4(SP),2(SP)   ;; FIRST ASCII CHARACTER ON IT
(1) 016726 012766 016740 000004   MOV     0STYIN,4(SP)
(1) 016734 000002          RTI
(1) 016736 000          98:  .BYTE   0            ;;RETURN
(1) 016737 000          .BYTE   0            ;;STORAGE FOR ASCII CHAR. TO TYPE
(1) 016737 000          .BYTE   0            ;;TERMINATOR
    
```



```
(1) 016740 000010          STTYIN: .BLKB      0          ;;RESERVE 8 BYTES FOR TTY INPUT
(1) 016750 052536 005015    000      SCNTLU: .ASCIZ    /U/<15><12>  ;;CONTROL "U"
(1) 016755      136 006507 000012    SCNTLG: .ASCIZ    /G/<15><12>  ;;CONTROL "G"
(1) 016762 005015 053523 020122    SHSWR: .ASCIZ    <15><12>/SWR 0 /
(1) 016770 020075      000          SHNEW: .ASCIZ    / NEW 0 /
(1) 016773      040 047040 053505
(1) 017000 036440 000040

1500
(1)          .SBTTL  READ AN OCTAL NUMBER FROM THE TTY
(1)
(2)          ;;*****
(1)          ;;THIS ROUTINE WILL READ AN OCTAL (ASCII) NUMBER FROM THE TTY AND
(1)          ;;CHANGE IT TO BINARY.
(1)          ;;CALL:
(1)          ;;      RDOCT          ;;READ AN OCTAL NUMBER
(1)          ;;      RETURN HERE    ;;LOW ORDER BITS ARE ON TOP OF THE STACK
(1)          ;;                      ;;HIGH ORDER BITS ARE IN SHIOCT
(1)
(1) 017004 011646          SRDOCT: NOV      (SP),-(SP)      ;;PROVIDE SPACE FOR THE
(1) 017006 016666 000004 000002      NOV      4(SP),2(SP)      ;;INPUT NUMBER
(3) 017014 010046          NOV      R0,-(SP)        ;;PUSH R0 ON STACK
(3) 017016 010146          NOV      R1,-(SP)        ;;PUSH R1 ON STACK
(3) 017020 010246          NOV      R2,-(SP)        ;;PUSH R2 ON STACK
(1) 017022 104407          10:      ROLIN          ;;READ AN ASCII LINE
(1) 017024 012600          NOV      (SP)+,R0      ;;GET ADDRESS OF 1ST CHARACTER
(1) 017026 005001          CLR      R1            ;;CLEAR DATA WORD
(1) 017030 005002          CLR      R2
(1) 017032 112046          20:      MOVB      (R0)+,-(SP)    ;;PICKUP THIS CHARACTER
(1) 017034 001012          BEQ      Z0            ;;IF ZERO GET OUT
(1) 017036 006301          ASL      R1            ;;02
(1) 017040 006102          ROL      R2
(1) 017042 006301          ASL      R1            ;;04
(1) 017044 006102          ROL      R2
(1) 017046 006301          ASL      R1            ;;08
(1) 017050 006102          ROL      R2
(1) 017052 042716 177770      BIC      0^C7,(SP)      ;;STRIP THE ASCII JUNK
(1) 017056 062601          ADD      (SP)+,R1      ;;ADD IN THIS DIGIT
(1) 017060 000764          BR       Z0            ;;LOOP
(1) 017062 005726          30:      TST      (SP)+
(1) 017064 010166 000012      NOV      R1,12(SP)     ;;CLEAN TERMINATOR FROM STACK
(1) 017070 010237 017104      NOV      R2,SHIOCT    ;;SAVE THE RESULT
(3) 017074 012602          NOV      (SP)+,R2
(3) 017076 012601          NOV      (SP)+,R1
(3) 017100 012600          NOV      (SP)+,R0
(1) 017102 000002          RTI
(1) 017104 000000          SHIOCT: .WORD      0          ;;HIGH ORDER BITS GO HERE
```

1502  
1503

(1)  
(1)  
(2)  
(1)  
(1)  
(1)  
(1)  
(1)  
(1) 017106  
(1) 017106 104400 001171  
(1) 017112 010046  
(1) 017114 005000  
(1) 017116 193700 001114  
(1) 017122 001004  
(1)  
(2) 017124 013746 001116  
(2)  
(2) 017130 104401  
(1) 017132 000426  
(1) 017134 005300  
(1) 017136 006300  
(1) 017140 006300  
(1) 017142 006300  
(1) 017144 062700 001320  
(1) 017150 012037 017160  
(1) 017154 001404  
(1) 017156 104400  
(1) 017160 000000  
(1) 017162 104400 001171  
(1) 017166 012037 017176  
(1) 017172 001404  
(1) 017174 104400  
(1) 017176 000000  
(1) 017200 104400 001171  
(1) 017204 011000  
(1) 017206 001004  
(1) 017210 012600  
(1) 017212 104400 001171  
(1) 017216 000207  
(2) 017220  
(2) 017220 013046  
(2) 017222 104401  
(1) 017224 005710  
(1) 017226 001770  
(1) 017230 104400 017236  
(1) 017234 000771  
(1) 017236 020040 000  
(1) 017242

.SBTTL ERROR MESSAGE TYPEOUT ROUTINE

\*\*\*\*\*  
THIS ROUTINE USES THE "ITEM CONTROL BYTE" (SITEMB) TO DETERMINE WHICH  
ERROR IS TO BE REPORTED. IT THEN OBTAINS, FROM THE "ERROR TABLE" (SERRTB),  
AND REPORTS THE APPROPRIATE INFORMATION CONCERNING THE ERROR.

SERRTYP:

```
TYPE      ,SCLRF      ;; "CARRIAGE RETURN" & "LINE FEED"
MOV       RB,-(SP)    ;;SAVE RB
CLR       RB          ;;PICKUP THE ITEM INDEX
BISB     @SITEMB,RB
BNE      18          ;;IF ITEM NUMBER IS ZERO, JUST
                          ;;TYPE THE PC OF THE ERROR
                          ;;SAVE SERRPC FOR TYPEOUT
                          ;;ERROR ADDRESS
                          ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
                          ;;GET OUT
                          ;;ADJUST THE INDEX SO THAT IT WILL
                          ;;WORK FOR THE ERROR TABLE
MOV       SERRPC,-(SP)
TYP0C
BR        68
18:      DEC         RB
ASL      RB
ASL      RB
ASL      RB
ADD      @SERRTB,RB   ;;FORM TABLE POINTER
MOV      (RB)+,28     ;;PICKUP "ERROR MESSAGE" POINTER
BEG      38           ;;SKIP TYPEOUT IF NO POINTER
TYPE     0            ;;TYPE THE "ERROR MESSAGE"
28:      .WORD      0  ;;"ERROR MESSAGE" POINTER GOES HERE
TYPE     ,SCLRF      ;; "CARRIAGE RETURN" & "LINE FEED"
38:      MOV       (RB)+,48
BEG      58           ;;SKIP TYPEOUT IF 0
TYPE     0            ;;TYPE THE "DATA HEADER"
48:      .WORD      0  ;;"DATA HEADER" POINTER GOES HERE
TYPE     ,SCLRF      ;; "CARRIAGE RETURN" & "LINE FEED"
58:      MOV       (RB),RB
BNE      78           ;;PICKUP "DATA TABLE" POINTER
68:      MOV       (SP)+,RB
TYPE     ,SCLRF      ;; "CARRIAGE RETURN" & "LINE FEED"
78:      RTS        PC  ;;RETURN
MOV      0(RB)+,-(SP) ;;SAVE 0(RB)+ FOR TYPEOUT
TYP0C
TST      (RB)        ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
BEG      68          ;;IS THERE ANOTHER NUMBER?
TYPE     ,88        ;;OR IF NO
BR        78         ;;TYPE TWO(2) SPACES
88:      .ASCIZ    / /  ;;LOOP
          .EVEN     ;;TWO(2) SPACES
```



1585

1586

(1)

(1)

(2)

(1)

(1)

(1)

(3)

(3)

(3)

(3)

(3)

(3)

(3)

(1)

(1)

(1)

(1)

(1)

(1)

(2)

(1)

(1)

(1)

(1)

(1)

(3)

(3)

(3)

(3)

(3)

(3)

(3)

(1)

(1)

(1)

(1)

(1)

(1)

(1)

(1)

(1)

(1)

1587

(1)

(1)

(1)

(1)

(1)

(1)

(1)

1588

.SBTTL POWER DOWN AND UP ROUTINES

;;\*\*\*\*\*

;;POWER DOWN ROUTINE

SPWRDN: MOV @SILLUP,@SPHRVEC ;;SET FOR FAST UP  
MOV @340,@SPHRVEC+2 ;;PRIO:7  
MOV R0,-(SP) ;;PUSH R0 ON STACK  
MOV R1,-(SP) ;;PUSH R1 ON STACK  
MOV R2,-(SP) ;;PUSH R2 ON STACK  
MOV R3,-(SP) ;;PUSH R3 ON STACK  
MOV R4,-(SP) ;;PUSH R4 ON STACK  
MOV R5,-(SP) ;;PUSH R5 ON STACK  
MOV @SWR,-(SP) ;;PUSH @SWR ON STACK  
MOV SP,@SAVR6 ;;SAVE SP  
MOV @SPWRUP,@SPHRVEC ;;SET UP VECTOR  
HALT  
BR -2 ;;HANG UP

;;\*\*\*\*\*

;;POWER UP ROUTINE

SPWRUP: MOV @SILLUP,@SPHRVEC ;;SET FOR FAST DOWN  
MOV @SAVR6,SP ;;GET SP  
CLR @SAVR6 ;;WAIT LOOP FOR THE TTY  
18: INC @SAVR6 ;;WAIT FOR THE INC  
BNE 18 ;;OP WORD  
MOV (SP)+,@SWR ;;POP STACK INTO @SWR  
MOV (SP)+,R5 ;;POP STACK INTO R5  
MOV (SP)+,R4 ;;POP STACK INTO R4  
MOV (SP)+,R3 ;;POP STACK INTO R3  
MOV (SP)+,R2 ;;POP STACK INTO R2  
MOV (SP)+,R1 ;;POP STACK INTO R1  
MOV (SP)+,R0 ;;POP STACK INTO R0  
MOV @SPWRDN,@SPHRVEC ;;SET UP THE POWER DOWN VECTOR  
MOV @340,@SPHRVEC+2 ;;PRIO:7  
TYPE PWRMSG ;;REPORT THE POWER FAILURE  
SPWRMSG: .WORD PWRMSG ;;POWER FAIL MESSAGE POINTER  
MOV (PC)+,(SP) ;;RESTART AT BEGIN1  
SPWRAD: .WORD BEGIN1 ;;RESTART ADDRESS  
RTI  
SILLUP: HALT ;;THE POWER UP SEQUENCE WAS STARTED  
BR -2 ;; BEFORE THE POWER DOWN WAS COMPLETE  
SAVR6: @ ;;PUT THE SP HERE  
PWRMSG: .ASCII <15><12>/RESTARTING AFTER A POWER FAILURE/<15><12><12>

.EVEN





```

(1) 017620 005704          TST      R4          ;;SUPPRESS THIS 0?
(1) 017622 001403          BEQ      58          ;;OR IF YES
(1) 017624 005204          48: INC      R4          ;;DON'T SUPPRESS ANYMORE 0'S
(1) 017626 052703 000060  BIS      0',R3       ;;MAKE THIS DIGIT ASCII
(1) 017632 052703 000040  58: BIS      0',R3       ;;MAKE ASCII IF NOT ALREADY
(1) 017636 110337 017702  MOV     R3,88       ;;SAVE FOR TYPING
(1) 017642 104400 017702  TYPE    ,88        ;;GO TYPE THIS DIGIT
(1) 017646 105337 017704  78: DECB   SOCNT     ;;COUNT BY 1
(1) 017652 003347          BGT     28          ;;OR IF MORE TO DO
(1) 017654 002402          BLY     68          ;;OR IF DONE
(1) 017656 005204          INC     R4          ;;INSURE LAST DIGIT ISN'T A BLANK
(1) 017660 000744          BR      28          ;;GO DO THE LAST DIGIT
(1) 017662 012605          68: MOV     (SP)+,R5   ;;RESTORE R5
(1) 017664 012604          MOV     (SP)+,R4   ;;RESTORE R4
(1) 017666 012603          MOV     (SP)+,R3   ;;RESTORE R3
(1) 017670 016666 000002 000004  MOV     2(SP),4(SP) ;;SET THE STACK FOR RETURNING
(1) 017676 012616          MOV     (SP)+,(SP)
(1) 017700 000002          RTI                    ;;RETURN
(1) 017702 000          88: .BYTE   0          ;;STORAGE FOR ASCII DIGIT
(1) 017703 000          .BYTE   0          ;;TERMINATOR FOR TYPE ROUTINE
(1) 017704 000          SOCNT:  .BYTE   0          ;;OCTAL DIGIT COUNTER
(1) 017705 000          88FILL: .BYTE   0          ;;ZERO FILL SWITCH
(1) 017706 000000          88MODE: .WORD   0          ;;NUMBER OF DIGITS TO TYPE

1592
(1)          .SBTTL  TYPE ROUTINE
(1)
(1)          ;;*****
(1)          ;;ROUTINE TO TYPE ASCII MESSAGE. MESSAGE MUST TERMINATE WITH A 0 BYTE.
(1)          ;;THE ROUTINE WILL INSERT A NUMBER OF NULL CHARACTERS AFTER A LINE FEED.
(1)          ;;NOTE1:  SNUL CONTAINS THE CHARACTER TO BE USED AS THE FILLER CHARACTER.
(1)          ;;NOTE2:  SPILLS CONTAINS THE NUMBER OF FILLER CHARACTERS REQUIRED.
(1)          ;;NOTE3:  SPILLC CONTAINS THE CHARACTER TO FILL AFTER.
(1)          ;;
(1)          ;;CALL:
(1)          ;;1) USING A TRAP INSTRUCTION
(1)          ;;      TYPE      ,MESADR          ;;MESADR IS FIRST ADDRESS OF AN ASCII STRING
(1)          ;;OR
(1)          ;;      TYPE
(1)          ;;      MESADR
(1)          ;;
(1)          STYPE:  TSTB   STPFLG          ;;IS THERE A TERMINAL?
(1)          017710 105737 001155          BPL     18          ;;OR IF YES
(1)          017714 100002          HALT                    ;;HALT HERE IF NO TERMINAL
(1)          017716 000000          BR      38          ;;LEAVE
(1)          017720 000430          18: MOV     R0,-(SP)   ;;SAVE R0
(1)          017722 010046          MOV     02(SP),R0   ;;GET ADDRESS OF ASCII STRING
(1)          017724 017600 000002          CMPB   @APTENV,SENV ;;RUNNING IN APT MODE
(1)          017730 122737 000001 001214          BNE    628          ;;NO,GO CHECK FOR APT CONSOLE
(1)          017736 001011          BITB   @APTSPOOL,SENVM ;;SPOOL MESSAGE TO APT
(1)          017740 132737 000100 001215          BNE    628          ;;NO,GO CHECK FOR CONSOLE
(1)          017746 001405          MOV     R0,618      ;;SETUP MESSAGE ADDRESS FOR APT
(1)          017750 010037 017760          JSR    PC,SATV3     ;;SPOOL MESSAGE TO APT
(1)          017754 004737 020200          .WORD   0          ;;MESSAGE ADDRESS
(1)          017760 000000          618:  .WORD   0          ;;APT CONSOLE SUPPRESSED
(1)          017762 132737 000040 001215  628:  BITB   @APTCSUP,SENVM ;;YES,SKIP TYPE OUT
(1)          017770 001003          BNE    608

```

```

(1) 017772 112046          291  MOV8  (R0)+,-(SP)  ;;PUSH CHARACTER TO BE TYPED ONTO STACK
(1) 017774 001005          BNE  49           ;;BR IF IT ISN'T THE TERMINATOR
(1) 017776 005726          TST  (SP)+       ;;IF TERMINATOR POP IT OFF THE STACK
(1) 020000 012600          603: MOV  (SP)+,R0   ;;RESTORE R0
(1) 020002 062716 000002  391  ADD  02,(SP)     ;;ADJUST RETURN PC
(1) 020006 000002          RTI                    ;;RETURN
(1) 020010 122716 000011  491  CMP8  0MT,(SP)   ;;BRANCH IF <MT>
(1) 020014 001430          BEQ  89           ;;BRANCH IF NOT <CRLF>
(1) 020016 122716 000200  CMP8  0CRLF,(SP)
(1) 020022 001006          BNE  99           ;;POP <CR><LF> EQUIV
(1) 020024 005726          TST  (SP)+       ;;TYPE A CR AND LF
(1) 020030 001171          TYPE
(1) 020032 105037 020166  SCRLF
(1) 020036 000755          CLRB  SCHARCNT   ;;CLEAR CHARACTER COUNT
(1) 020040 004737 020122  BR  29           ;;GET NEXT CHARACTER
(1) 020044 123726 001154  JSR  PC,STYPEC   ;;GO TYPE THIS CHARACTER
(1) 020050 001350          CMP8  0FILLC,(SP)+
(1) 020052 013746 001152  BNE  29           ;;IS IT TIME FOR FILLER CHARS.?
(1)                                ;;IF NO GO GET NEXT CHAR.
(1)                                ;;GET # OF FILLER CHARS. NEEDED
(1)                                ;;AND THE NULL CHAR.
(1) 020056 105366 000001  791  DECB  1(SP)     ;;DOES A NULL NEED TO BE TYPED?
(1) 020062 002770          BLY  69           ;;BR IF NO--SO POP THE NULL OFF OF STACK
(1) 020064 004737 020122  JSR  PC,STYPEC   ;;GO TYPE A NULL
(1) 020070 105337 020166  DECB  SCHARCNT
(1) 020074 000770          BR  79           ;;DO NOT COUNT AS A COUNT
(1)                                ;;LOOP

(1)                                ;HORIZONTAL TAB PROCESSOR
(1)
(1)
(1) 020076 112716 000040  891  MOV8  0' ,(SP)   ;;REPLACE TAB WITH SPACE
(1) 020102 004737 020122  991  JSR  PC,STYPEC   ;;TYPE A SPACE
(1) 020106 132737 000007 020166  BIT8  07,SCHARCNT
(1) 020114 001372          BNE  99           ;;BRANCH IF NOT AT
(1) 020116 005726          TST  (SP)+       ;;TAB STOP
(1) 020120 000724          BR  29           ;;POP SPACE OFF STACK
(1) 020122 105777 161020  STYPEC: TST8  0STPB
(1) 020126 100375          BPL  STYPEC      ;;GET NEXT CHARACTER
(1) 020130 116677 000002 161012  MOV8  2(SP),0STPB
(1) 020136 122766 000015 000002  CMP8  0CR,2(SP)  ;;WAIT UNTIL PRINTER IS READY
(1) 020144 001003          BNE  19           ;;LOAD CHAR TO BE TYPED INTO DATA REG.
(1) 020146 105037 020166  CLRB  SCHARCNT   ;;IS CHARACTER A CARRIAGE RETURN?
(1) 020152 000406          BR  STYPEX      ;;BRANCH IF NO
(1) 020154 122766 000012 000002  191  CMP8  0LF,2(SP)  ;;YES--CLEAR CHARACTER COUNT
(1) 020162 001402          BEQ  STYPEX      ;;EXIT
(1) 020164 105227          INCB  (PC)+      ;;IS CHARACTER A LINE FEED?
(1) 020166 000000          SCHARCNT: WORD  0  ;;BRANCH IF YES
(1) 020170 000207          STYPEX: RTS  PC  ;;COUNT THE CHARACTER
(1)                                ;;CHARACTER COUNT STORAGE

(1)                                ;SBTTL APT COMMUNICATIONS ROUTINE
(1)
(1)
(2)
(1) 020172 112737 000001 020436  SATY1: MOV8  01,0PFLG  ;;*****
(1) 020200 112737 000001 020434  SATY3: MOV8  01,0MFLG  ;;TO REPORT FATAL ERROR
(1) 020206 000403          BR  SATYC        ;;TO TYPE A MESSAGE
(1) 020210 112737 000001 020436  SATY4: MOV8  01,0PFLG
(2) 020216          SATYC:          ;;TO ONLY REPORT FATAL ERROR
    
```



APT COMMUNICATIONS ROUTINE

```

(3) 020216 010046      MOV      R0,-(SP)      ;;PUSH R0 ON STACK
(3) 020220 010146      MOV      R1,-(SP)      ;;PUSH R1 ON STACK
(1) 020222 105737 020434  TSTB    SMFLG         ;;SHOULD TYPE A MESSAGE?
(1) 020226 001450      BEQ      58           ;;IF NOT: BR
(1) 020230 122737 000001 001214  CMPS    SPTENV,SENV   ;;OPERATING UNDER APT?
(1) 020236 001031      BNE     38           ;;IF NOT: BR
(1) 020240 132737 000100 001215  BITB    SPTSPool,SENVH ;;SHOULD SPOOL MESSAGE?
(1) 020246 001425      BEQ     38           ;;IF NOT: BR
(1) 020250 017600 000004      MOV     04(SP),R0     ;;GET MESSAGE ADDR.
(1) 020254 062766 000002 000004  ADD     02,4(SP)      ;;BUMP RETURN ADDR.
(1) 020262 005737 001174      TST     SMSTYPE       ;;SEE IF DONE W/ LAST XMISSION?
(1) 020266 001375      BNE     18           ;;IF NOT: WAIT
(1) 020270 010037 001210      MOV     R0,SMSTAD     ;;PUT ADDR IN MAILBOX
(1) 020274 105720 281      TSTB    (R0)+        ;;FIND END OF MESSAGE
(1) 020276 001376      BNE     28
(1) 020300 163700 001210      SUB     SMSTAD,R0     ;;SUB START OF MESSAGE
(1) 020304 006200      ASR     R0           ;;GET MESSAGE LGTH IN WORDS
(1) 020306 010037 001212      MOV     R0,SMSTLGT    ;;PUT LENGTH IN MAILBOX
(1) 020312 012737 000004 001174  MOV     04,SMSTYPE    ;;TELL APT TO TAKE MSG.
(1) 020320 000413      BR      58
(1) 020322 017637 000004 020346 381      MOV     04(SP),48     ;;PUT MSG ADDR IN JSR LINKAGE
(1) 020330 062766 000002 000004  ADD     02,4(SP)      ;;BUMP RETURN ADDRESS
(3) 020336 013746 177776      MOV     177776,-(SP)  ;;PUSH 177776 ON STACK
(1) 020342 004737 017710      JSR     PC,STYPE     ;;CALL TYPE MACRO
(1) 020346 000000      .WORD  0
(1) 020350      .WORD  581
(1) 020350 105737 020436 1001      TSTB    SFPLG         ;;SHOULD REPORT FATAL ERROR?
(1) 020354 001416      BEQ     128          ;;IF NOT: BR
(1) 020356 005737 001214      TST     SENV         ;;RUNNING UNDER APT?
(1) 020362 001413      BEQ     128          ;;IF NOT: BR
(1) 020364 005737 001174 1181      TST     SMSTYPE       ;;FINISHED LAST MESSAGE?
(1) 020370 001375      BNE     118          ;;IF NOT: WAIT
(1) 020372 017637 000004 001176      MOV     04(SP),SFATAL ;;GET ERROR #
(1) 020400 062766 000002 000004  ADD     02,4(SP)      ;;BUMP RETURN ADDR.
(1) 020406 005237 001174      INC     SMSTYPE       ;;TELL APT TO TAKE ERROR
(1) 020412 105037 020436 1281      CLRB   SFPLG         ;;CLEAR FATAL FLAG
(1) 020416 105037 020435      CLRB   SLPLG         ;;CLEAR LOG FLAG
(1) 020422 105037 020434      CLRB   SMFLG         ;;CLEAR MESSAGE FLAG
(3) 020426 012601      MOV     (SP)+,R1     ;;POP STACK INTO R1
(3) 020430 012600      MOV     (SP)+,R0     ;;POP STACK INTO R0
(1) 020432 000207      RTS     PC           ;;RETURN
(1) 020434      000      SMFLG: .BYTE 0     ;;MESSAGE FLAG
(1) 020435      000      SLPLG: .BYTE 0     ;;LOG FLAG
(1) 020436      000      SFPLG: .BYTE 0     ;;FATAL FLAG
(1)      020440      .EVEN
(1)      000200      APTSIZE=200
(1)      000001      APTENV=001
(1)      000100      APTSPool=100
(1)      000040      APTCSUP=040
1594
(1)      .SBTTL TRAP DECODER
(1)
(2)
(1)      ;;*****
(1)      ;;THIS ROUTINE WILL PICKUP THE LOWER BYTE OF THE "TRAP" INSTRUCTION
(1)      ;;AND USE IT TO INDEX THROUGH THE TRAP TABLE FOR THE STARTING ADDRESS
(1)      ;;OF THE DESIRED ROUTINE. THEN USING THE ADDRESS OBTAINED IT WILL

```





ABASE = 170400	140	26												
ACDW1 = 000000	26													
ACDW2 = 000000	26													
ACPUOP = 000000	26													
ADDR1 = 001510	1200	1319*	1322	1325	1327									
ADCS = 001516	1260	210	233	1145*	1147	1153*	1155	1161*	1163	1169*	1171	1177*	1179	
	1185*	1187	1193*	1195	1201*	1203	1210*	1211*	1212	1220	1222*	1224	1232*	
	1234*	1235	1242	1252*	1254	1258	1262	1267	1275*	1276*	1277	1291*	1294*	
	1298	1304*	1308*	1313	1330*	1334*	1340*	1346*	1350*	1355*	1367*	1373*	1375	
	1382*	1385	1390*	1392	1395	1399*	1400	1419	1562	1563				
ADDR = 001520	1270	217	1231	1250	1260	1280	1293	1331	1347	1403	1573			
ADDW0 = 000000	26													
ADDW1 = 000000	26													
ADDW10 = 000000	26													
ADDW11 = 000000	26													
ADDW12 = 000000	26													
ADDW13 = 000000	26													
ADDW14 = 000000	26													
ADDW15 = 000000	26													
ADDW2 = 000000	26													
ADDW3 = 000000	26													
ADDW4 = 000000	26													
ADDW5 = 000000	26													
ADDW6 = 000000	26													
ADDW7 = 000000	26													
ADDW8 = 000000	26													
ADDW9 = 000000	26													
ADEVCT = 000000	26													
ADEVH = 000000	26													
ADINT = 001536	1398	212	1332*	1349*	1359*	1360*								
ADINT1 = 001540	1400	1368	1369*											
AENV = 000000	26													
AENVH = 000000	26													
AFATAL = 000000	26													
ANADR1 = 000000	26													
ANADR2 = 000000	26													
ANADR3 = 000000	26													
ANADR4 = 000000	26													
ANAMS1 = 000000	26													
ANAMS2 = 000000	26													
ANAMS3 = 000000	26													
ANAMS4 = 000000	26													
ANSGAD = 000000	26													
ANSGLG = 000000	26													
ANSCTY = 000000	26													
ANTYP1 = 000000	26													
ANTYP2 = 000000	26													
ANTYP3 = 000000	26													
ANTYP4 = 000000	26													
APASS = 000000	26													
APRIOR = 000200	160	26												
APTCBU = 000040	1592	1593*												
APTENV = 000001	1578	1592	1593*											
APTSIZ = 000200	186	1593*												
APTSPO = 000100	1592	1593*												
ARBAOD = 001552	1400	182*	211	215	1450*	1454*	1572							

AROVCT	001554	1490	1830	214	14510	14550												
ASHREG	000000	26																
ATESTN	000000	26																
AUNIT	000000	26																
AUSWR	000000	26																
AVECT1	000340	150	26															
AVECT2	000000	26																
BEGIN	001576	19	1610															
BEGIN1	001602	20	1630	1506														
BIT0	000001	130	360	577	651	670	717	971	1079	1092	1233	1244	1252	1269				
		1334	1350	1390														
BIT00	000001	130																
BIT01	000002	130																
BIT02	000004	130																
BIT03	000010	130																
BIT04	000020	130																
BIT05	000040	130																
BIT06	000100	130																
BIT07	000200	130																
BIT08	000400	130	1577															
BIT09	001000	130	1577	1570														
BIT1	000002	130	360	464	465	549	589	619	639	690	715							
BIT10	002000	130	837	830	1040	1117	1185	1186	1201	1373								
BIT11	004000	130	457	490	466	467	486	487	505	506	519	520	536	537				
		702	703	844	845	1117	1193	1194	1373	1577								
BIT12	010000	130	851	852	1044	1049												
BIT13	020000	130	1201	1202	1373	1570												
BIT14	040000	130	409	550	1577													
BIT15	100000	130	424	567														
BIT2	000004	130	376	540	604	619	790	799	900	909	947	1010	1100					
BIT3	000010	130	304	549	630	639	812	813	1019	1020	1027	1030	1100					
BIT4	000020	130	360	360	376	304	303	401	409	417	424	433	465	509				
		604	619	630	639	1145	1146	1292	1294	1302								
BIT5	000040	130	1153	1154	1210	1219	1222	1306	1300	1310								
BIT6	000100	130	393	550	699	715	821	822	1079	1092	1117	1161	1162	1334				
		1350	1302															
BIT7	000200	130	417	465	567	509	604	619	630	639	790	799	805	813				
		822	830	830	845	901	900	1010	1019	1030	1109	1110	1243	1253				
		1306																
BIT8	000400	130	401	549	1169	1170	1373											
BIT9	001000	130	829	830	1117	1177	1170	1209	1210	1222	1373							
BPTVEC	000014	130																
BRLEV1	001572	1560	6900	6910	701	10710	10720	1078	13250	13260	1333							
BRLEV2	001574	1570	6920	716	10730	1091	13270	1340										
BUFADR	013162	14270	14450	1574														
BUFNUM	013142	1413	1420	14360														
BYPASS	013166	246	14400															
BYPAS1	013244	1493	14500	1460														
CKBRL	001512	1210	6030	606	690	692												
CKDLY	014064	14970	1515	15200														
CKSWR	104405	1577	15940															
CNYDEV	014060	14950	1512	15260														
COUNT	001562	1520	4050	4000	5040	5070	5100	5210	5350	5300								
CR	000015	130	1592															
CRLF	000200	130	1592															
CSD	001524	1300	2520	253	2600	261	2600	269	2760	277	2860	287	2970	299				



		305*	313*	323*	331*	340*	351*	443*	463*	475*	495*	515*	532*	576*
CSC	001534	698*	714*	1469*	1493*	1522*	1565							
		1370	236	306	314	324	332	341	353	447	479	498	523	548
CSR	001522	596	611	657	676	1411	1520	1567						
		1290	361*	362	369*	370	377*	378	385*	386	394*	395	402*	403
		410*	411	418*	419	425*	426	432*	435	442*	444*	452	457*	458*
		462*	464*	466*	467*	468	474*	477*	486*	487*	494*	496*	505*	506*
		514*	517*	519*	520*	531*	534*	536*	537*	549*	552	558*	561	567*
		570	577*	580	591	606	621	632	641	650	651*	669	670*	697*
		699*	702*	703*	713*	715*	717*	1305*	1468*	1470*	1472	1492*	1516*	1517*
		1521*	1564	1566										
DDISP	177570	130	26	186										
DELAY	001564	1530	1471*	1475*										
DM1	014662	30	1547*											
DM10	015115	72	1553*											
DM11	015153	78	1554*											
DM12	015213	84	1555*											
DM13	015254	90	1556*											
DM14	015315	96	1557*											
DM15	015334	102	1558*											
DM16	015364	109	1559*											
DM17	015425	115	1560*											
DM2	014722	36	1548*											
DM3	014740	42	1549*											
DM4	015000	48	1550*											
DM5	015040	54	1551*											
DM6	015055	60	66	1552*										
DISPLA	001140	260	186*	1577*	1578*									
DISPRE	000174	190	186											
DSWR	177570	130	26	186										
DT1	015466	31	1562*											
DT11	015552	79	1568*											
DT12	015564	85	1569*											
DT13	015576	91	1570*											
DT14	015610	97	1571*											
DT15	015616	103	1572*											
DT16	015626	110	1573*											
DT17	015640	116	1574*											
DT2	015500	37	1563*											
DT3	015506	43	1564*											
DT4	015520	49	1565*											
DT5	015532	55	1566*											
DT6	015540	61	67	73	1567*									
ENTVEC	000030	130	186*											
EM1	014866	29	1530*											
EM10	014370	71	1537*											
EM11	014423	77	1538*											
EM12	014457	83	1540*											
EM13	014503	89	1541*											
EM14	014527	95	1542*											
EM15	014555	101	1543*											
EM16	014576	108	1545*											
EM17	014630	114	1546*											
EM2	014126	35	1531*											
EM3	014155	41	1532*											
EM4	014214	47	1533*											



EMS	014253	93	15340																		
EM6	014301	90	15350																		
EM7	014341	65	15360																		
ERRVEC	000004	130	1860	15770																	
GNB	***** U	19	202	203	207	1594															
HT	000011	130	1592																		
IOTVEC	000020	130	1860																		
KWIV	001542	1420	6960	7110	7300	7300															
KWIV8	001544	1430	7120	730	7300																
LP	000012	130	1592																		
MINCNT	014062	14960	1507	1509	15270																
NDEXT	001560	1510	1840	1448	14520	14560															
NMDEXT	001556	1500	1790	184	204	1456															
PC	%000007	130	5900	6050	6200	6310	6400	6490	6600	13090	14600	14980	14990	15010							
		15230	15780	15790	15830	1586	15920	15930													
PIRQ	177772	130																			
PIRQVE	000240	130																			
PR0	000000	130																			
PR1	000040	130																			
PR2	000100	130																			
PR3	000140	130																			
PR4	000200	130																			
PR5	000240	130																			
PR6	000300	130																			
PR7	000340	130																			
PS	177776	130																			
PSW	177776	130	7010	7160	7320	10750	10780	10880	10910	11000	13330	13400	13610								
PWRMSG	017414	1500	15070																		
PWRVEC	000024	130	1860	15860																	
RATE	014056	5000	6030	6180	6290	6380	6470	6660	13070	1470	14940	1516	15250								
RBE6	001606	162	1640																		
RBE61	001714	1050																			
RBE62	002170	1070	1459																		
RDCHR	104406	1570	15940																		
RDLIN	104407	1500	15940																		
RDOCT	104410	15940																			
REPEAT	013676	742	752	761	770	780	14920														
RESVEC	000010	130																			
R0	%000000	130	1610	1630	196	2100	2110	212	2140	215	2170	2190	223	2250							
		7000	7040	7180	7190	7310	7330	9690	9790	9870	10000	10180	10360	10460							
		10990	10770	10800	10900	10930	12310	12390	12400	12500	12510	12560	12600	12930							
		12950	12960	13100	13110	13310	13350	13360	13470	13510	13520	13600	13620	14110							
		14120	14190	14190	1424	1427	1429	14500	14600	14790	1500	1502	15030	15040							
		15060	1512	15150	15180	15200	15000	15030	15060	15920	15930	15940									
R1	%000001	130	1890	1900	1910	192	2100	219	220	221	2240	225	226	227							
		14130	1415	14200	1423	1429	14790	15000	15060	15930											
R2	%000002	130	1670	169	1700	1800	190	1920	193	194	14140	14160	14210	14300							
		14700	15000	15060																	
R3	%000003	130	1680	1710	174	1700	179	14790	15790	15860	15910										
R4	%000004	130	15060	15910																	
R5	%000005	130	7420	7520	7610	7700	7800	14790	1494	1495	1496	15100	1511	15130							
		15060	15910																		
R6	%000006	130	1860																		
R7	%000007	130																			
SP	%000006	130	1650	175	1860	2040	243	700	723	737	1006	1105	1341	1356							
		1360	14600	14790	15770	15780	15790	15800	15830	15860	15910	15920	15930	15940							





TST12	003214	316	3210	
TST120	011172	1149	11520	
TST121	011230	1157	11600	
TST122	011266	1165	11680	
TST123	011324	1173	11760	
TST124	011362	1181	11840	
TST125	011420	1189	11920	
TST126	011456	1197	12000	
TST127	011514	1205	12080	
TST13	003252	326	3290	
TST130	011556	1214	12180	
TST131	011632	1226	12290	
TST132	011736	1245	12490	
TST133	012062	1270	12740	
TST134	012140	1282	1284	12900
TST135	012222	1300	13030	
TST136	012320	1315	13180	
TST137	012376	13290		
TST14	003310	334	3370	
TST140	012462	1339	13450	
TST141	012616	1350	1365	13710
TST142	012654	1376	13810	
TST143	012712	1386	13890	
TST144	012756	1396	13980	
TST145	013030	1406	14090	
TST146	013164	1432	14470	
TST15	003366	3490		
TST16	003426	354	3590	
TST17	003464	364	3670	
TST2	002612	241	2500	
TST20	003522	372	3750	
TST21	003560	380	3830	
TST22	003616	388	3920	
TST23	003654	397	4000	
TST24	003712	405	4080	
TST25	003750	413	4160	
TST26	004006	421	4230	
TST27	004044	428	4310	
TST3	002650	255	2580	
TST30	004116	4410		
TST31	004242	456	4610	
TST32	004324	470	4730	
TST33	004440	4930		
TST34	004550	5130		
TST35	004654	525	5300	
TST36	004760	542	5480	
TST37	005026	554	5570	
TST4	002706	263	2660	
TST40	005074	563	5660	
TST41	005142	572	5750	
TST42	005216	582	5870	
TST43	005314	598	6020	
TST44	005412	613	6170	
TST45	005462	623	6280	
TST46	005532	634	6370	
TST47	005602	643	6460	





270	287*	288	299*	306*	307	314*	315	324*	325	332*	333	341*
342	353*	362*	363	370*	371	378*	379	386*	387	395*	396	403*
404	411*	412	419*	428	426*	427	435*	436	447*	448	452*	468*
469	479*	480	498*	499	523*	524	540*	541	552*	553	561*	562
570*	571	578*	580*	581	591*	592	596*	597	606*	607	611*	612
621*	622	632*	633	641*	642	650*	652	657*	658	669*	671	676*
677	792*	793	800*	801	806*	807	814*	815	823*	824	831*	832
839*	840	846*	847	853*	854	862*	863	870*	871	878*	879	886*
887	896*	897	908*	909	916*	917	924*	925	933*	934	942*	943
955*	956	961*	962	972*	973	982*	990*	991	999*	1000	1011*	1021*
1022	1029*	1030	1039*	1111*	1112	1120*	1121	1129*	1130	1138*	1139	1147*
1148	1155*	1156	1163*	1164	1171*	1172	1179*	1180	1187*	1188	1195*	1196
1203*	1204	1212*	1213	1224*	1225	1235*	1236	1242*	1244	1250*	1262*	1263
1267*	1269	1280*	1283	1298*	1299	1313*	1314	1375*	1385*	1395*	1403*	1405
1424*	1425	1502*	1504	1507	1562	1564	1565	1567	1568	1569	1570	1572
1573	1574											

SCDW1	001254	260										
SCDW2	001256	260										
SCNARC	020166	15920*										
SCKSWR	016312	15790	1594									
SCMYAG	001100	260	106									
SCM1	= 000002	260										
SCM2	= 000004	260										
SCM3	= 000002	260										
SCNYLG	016755	15790										
SCNYLU	016750	15790										
SCPUOP	001222	260										
SCRLF	001171	260	1578	1579	1583	1592						
SDBLK	013666	14790										
SDDW0	001260	260										
SDDW1	001262	260										
SDDW10	001304	260										
SDDW11	001306	260										
SDDW12	001310	260										
SDDW13	001312	260										
SDDW14	001314	260										
SDDW15	001316	260										
SDDW2	001264	260										
SDDW3	001266	260										
SDDW4	001270	260										
SDDW5	001272	260										
SDDW6	001274	260										
SDDW7	001276	260										
SDDW8	001300	260										
SDDW9	001302	260										
SDEVCT	001204	260										
SDEVM	001252	260										
SDOAGN	013352	14600										
SDYBL	013656	14790										
SENDAD	013342	23	14600	1578								
SENDCT	013310	106	14600									
SENDMG	013361	14600										
SENULL	013356	14600										
SENV	001214	260	1578	1592	1593							
SENVH	001215	260	106	1592	1593							
SEOP	013254	1457	14600									









	1398	1409	1447	1460	1577	1578	1586							
SSWREG 001216	260	186												
SSWRMK= 000000	18	1577												
STESTN 001200	260	1577*												
STIMES 001164	260	186*	230*	283*	295*	337*	349*	431*	441*	473*	493*	513*	530*	
	548*	557*	566*	575*	587*	602*	617*	628*	637*	646*	665*	682*	741*	
	751*	760*	769*	779*	789*	892*	938*	1064*	1074*	1087*	1107*	1116*	1125*	
	1134*	1290*	1303*	1318*	1329*	1345*	1371*	1381*	1389*	1398*	1409*	1460*	1577*	
STKB 001144	260	1579												
STKS 001142	260	1579												
STN = 000147	60	12	2300	241	2500	255	2580	263	2660	271	2740	279	2830	
	2950	300	3030	308	3110	316	3210	326	3290	334	3370	3490	354	
	3590	364	3670	372	3750	380	3830	388	3920	397	4000	405	4080	
	413	4160	421	4230	428	4310	4410	456	4610	470	4730	4930	5130	
	525	5300	542	5480	554	5570	563	5660	572	5750	582	5870	598	
	6020	613	6170	623	6280	634	6370	643	6460	659	6650	678	6820	
	6950	707	7100	724	736	7410	748	7510	758	7600	767	7690	776	
	7790	786	7890	794	7970	808	8110	816	8200	825	8280	833	8360	
	841	8430	848	8500	8590	864	8670	872	8750	880	8830	888	8920	
	9050	910	9130	918	9210	926	9300	935	9380	9510	963	9670	976	
	978	9850	995	1004	1007	10160	1026	1033	1035	10430	1045	1053	1056	
	10640	10740	1084	10870	1090	1103	11070	1113	11160	1122	11250	1131	11340	
	1140	11440	1149	11520	1157	11600	1165	11680	1173	11760	1181	11840	1189	
	11920	1197	12000	1205	12080	1214	12180	1226	12290	1245	12490	1270	12740	
	1282	1284	12900	1300	13030	1315	13180	13290	1339	13450	1358	1365	13710	
	1376	13810	1386	13890	1396	13980	1406	14090	1432	14470				
STPB 001150	260	1592*												
STPFLG 001155	260	1592												
STPS 001146	260	1592												
STRAP 020440	186	15940												
STRP = 000011	15940													
STRPAD 020462	15940													
STSYM 001004	250													
STSTNH 001102	260	1460*	1577*	1578										
STTYIN 016740	15790													
STYPDN= ***** U	1594													
STYPD8 013452	14790	1594												
STYPE 017710	15920	1593	1594											
STYPEC 020122	1579	15920												
STYPEX 020170	15920													
STYPOC 017506	15910	1594												
STYPON 017522	15910	1594												
STYPOS 017462	15910	1594												
SUNIT 001206	260													
SUNITM 001010	250													
SUSWR 001220	260													
SVECT1 001244	260	183	1455											
SVECT2 001245	260													
SXTSTR 015664	15770													
SSGET4= 000000	14600													
SOFILL 017705	15910*													
SADCAT= ***** U	1577	1578												
. = 020504	190	230	240	250	260	186	2030	684	687	720	734	1066	1069	
	1081	1094	1241	1320	1323	1337	1353	1363	1393	1460	14790	1577	1578	
	15790	15830	1586	1592	15930									
.SASTA= ***** U	1593													

.SX = 001000

250



COMHEN	130															
ENDCOM	130															
ERROR	130	245	256	264	272	280	290	301	309	317	327	335	344	355	365	
	373	381	389	398	406	414	422	429	438	450	455	471	482	501	526	
	543	555	564	573	583	594	599	609	614	624	635	644	654	660	673	
	679	706	722	735	747	749	757	759	766	768	775	777	785	787	795	
	803	809	817	826	834	842	849	856	865	873	881	889	899	911	919	
	927	936	945	958	964	975	983	993	1003	1012	1024	1032	1040	1052	1061	
	1083	1097	1102	1114	1123	1132	1141	1150	1158	1166	1174	1182	1190	1198	1206	
	1215	1227	1238	1246	1259	1265	1271	1285	1301	1316	1338	1357	1364	1377	1387	
	1397	1407	1420													
ESCAPE	130															
GETPRI	130															
MULT	130															
NEWST	130	230	250	258	266	274	283	295	303	311	321	329	337	349	359	
	367	375	383	392	400	408	416	423	431	441	461	473	493	513	530	
	548	557	566	575	587	602	617	628	637	646	665	682	695	710	741	
	751	760	769	779	789	797	811	820	828	836	843	850	859	867	875	
	883	892	905	913	921	930	938	951	967	985	1016	1043	1064	1074	1087	
	1107	1116	1125	1134	1144	1152	1160	1168	1176	1184	1192	1200	1208	1218	1229	
	1249	1274	1290	1303	1318	1329	1345	1371	1381	1389	1398	1409	1447			
POP	130	1479	1500	1506	1593											
PUSH	130	1479	1500	1506	1593											
REPORT	130															
SCOPE	130	230	250	258	266	274	283	295	303	311	321	329	337	349	359	
	367	375	383	392	400	408	416	423	431	441	461	473	493	513	530	
	548	557	566	575	587	602	617	628	637	646	665	682	695	710	741	
	751	760	769	779	789	797	811	820	828	836	843	850	859	867	875	
	883	892	905	913	921	930	938	951	967	985	1016	1043	1064	1074	1087	
	1107	1116	1125	1134	1144	1152	1160	1168	1176	1184	1192	1200	1208	1218	1229	
	1249	1274	1290	1303	1318	1329	1345	1371	1381	1389	1398	1409	1447			
SETPRI	130															
SETTRA	15948															
SETUP	130	186														
SKIP	130	241	255	263	271	279	289	300	308	316	326	334	343	354	364	
	372	380	388	397	405	413	421	428	437	449	453	456	470	481	500	
	525	542	554	563	572	582	593	598	608	613	623	634	643	653	659	
	672	678	707	721	724	736	748	758	767	776	786	794	802	808	816	
	825	833	841	848	855	864	872	880	888	898	910	918	926	935	944	
	957	963	974	976	978	992	995	1002	1004	1007	1009	1023	1026	1031	1033	
	1035	1037	1045	1051	1053	1056	1058	1060	1064	1095	1098	1103	1113	1122	1131	
	1140	1149	1157	1165	1173	1181	1189	1197	1205	1214	1226	1237	1245	1264	1270	
	1282	1284	1300	1315	1339	1354	1358	1365	1376	1386	1396	1406	1426	1432		
SLASH	130															
SPACE	130															
STARS	130	23	25	26	230	250	258	266	274	283	295	303	311	321	329	
	337	349	359	367	375	383	392	400	408	416	423	431	441	461	473	
	493	513	530	548	557	566	575	587	602	617	628	637	646	665	682	
	695	710	741	751	760	769	779	789	797	811	820	828	836	843	850	
	859	867	875	883	892	905	913	921	930	938	951	967	985	1016	1043	
	1064	1074	1087	1107	1116	1125	1134	1144	1152	1160	1168	1176	1184	1192	1200	
	1208	1218	1229	1249	1274	1290	1303	1318	1329	1345	1371	1381	1389	1398	1409	
	1447	1460	1479	1577	1578	1579	1580	1583	1586	1591	1592	1593	1594			
SWRSU	130	1860														
TRMTRP	15948															
TYPBIN	130															

DZARAB.P11

CROSS REFERENCE TABLE -- MACRO NAMES

TYPDEC	130	1460													
TYPNAM	130														
TYPNUM	130														
TYPOCS	130														
TYPOCT	130	1579	1503												
TYPTXT	130	202	203	207											
SSCHRE	260														
SSCHTM	260														
SSESCA	130														
SSNEWT	130	230	250	250	266	274	283	295	303	311	321	329	337	349	359
	367	375	383	392	400	400	416	423	431	441	461	473	493	513	530
	548	557	566	575	587	602	617	628	637	646	665	682	695	710	741
	751	760	769	779	789	797	811	820	828	836	843	850	859	867	875
	883	892	905	913	921	930	938	951	967	985	1016	1043	1064	1074	1087
	1107	1116	1125	1134	1144	1152	1160	1168	1176	1184	1192	1200	1208	1218	1229
	1249	1274	1290	1303	1318	1329	1345	1371	1381	1389	1398	1409	1447		
SSSET	15940														
SSSETH	1060														
SSSKIP	130	241	255	263	271	279	300	308	316	326	334	354	364	372	380
	388	397	405	413	421	428	456	470	525	542	554	563	572	582	590
	613	623	634	643	659	678	707	724	736	748	758	767	776	786	794
	800	816	825	833	841	848	864	872	880	888	910	918	926	935	963
	976	978	995	1004	1007	1026	1033	1035	1045	1053	1055	1084	1098	1103	1113
	1122	1131	1140	1149	1157	1165	1173	1181	1189	1197	1205	1214	1226	1245	1270
	1282	1284	1300	1315	1339	1350	1365	1376	1386	1396	1406	1432			
.EQUAY	70	13													
.HEADE	70	12													
.SETUP	80	105													
.SWRMI	90	10													
.SWRLO	100														
.SACT1	100	23													
.SAPTB	100	260													
.SAPTH	100	25													
.SAPTY	100	1593													
.SCATC	70	19													
.SCHTA	70	26													
.SEOP	70	1460													
.SERRO	70	1570													
.SERRY	90	1503													
.SPARM	80														
.SPOWE	80	1506													
.SRDOC	90	1500													
.SREAD	80	1579													
.SSAVE	80														
.SSCOP	80	1577													
.SSPAC	80														
.SSWDO	80														
.STRAP	80	1594													
.STYPD	90	1479													
.STYPE	70	80	1592												
.STVPO	70	1591													



ADD	170	219	225	230	1412	1450	1451	1475	1479	1479	1500	1503	1591	1592	1593
ASL	1579	1500	1503	1594											
ASLB	1479														
ASR	1593														
BCC	1479														
BEQ	186	197	200	237	255	263	271	279	289	300	308	316	326	334	343
	354	364	372	380	380	397	405	413	421	420	449	481	500	525	542
	554	563	572	582	593	598	600	613	623	634	643	653	659	672	678
	794	802	808	816	825	833	841	848	855	864	872	880	888	898	910
	918	926	935	944	957	963	1045	1113	1122	1131	1140	1149	1157	1165	1173
	1181	1189	1197	1205	1214	1226	1237	1245	1270	1282	1284	1315	1376	1386	1396
	1406	1426	1440	1460	1577	1578	1579	1580	1583	1591	1592	1593			
BGE	1500	1577													
BGT	1460	1479	1579	1591											
BHI	1577														
BIC	450	467	487	506	520	537	651	670	683	703	1065	1100	1319	1460	1579
	1500	1591													
BIS	457	466	486	505	519	536	702	717	1049	1294	1300	1334	1350	1479	1579
	1591														
BISB	1583														
BIT	1044	1244	1269	1281	1577	1578									
BITB	186	1592	1593												
BLE	740	750	767	776	786										
BLOS	1579														
BLT	1479	1579	1591	1592											
BMI	470	978	992	1007	1023	1035	1056	1255	1473	1479					
BNE	175	186	195	213	216	222	228	293	347	440	460	489	491	500	510
	522	539	684	687	705	720	734	903	949	980	1009	1037	1050	1060	1066
	1069	1081	1094	1241	1257	1297	1312	1320	1323	1337	1353	1363	1417	1431	1476
	1479	1510	1577	1578	1579	1583	1586	1591	1592	1593					
BPL	437	453	974	1002	1031	1051	1264	1278	1300	1393	1401	1479	1505	1578	1579
	1591	1592													
BR	162	172	202	203	207	239	241	456	707	721	724	736	976	995	1004
	1026	1033	1053	1084	1095	1098	1103	1339	1354	1358	1365	1432	1453	1479	1577
	1578	1579	1580	1583	1586	1591	1592	1593							
CLR	161	168	181	186	191	205	339	432	434	442	443	445	462	474	475
	476	478	494	495	497	514	515	531	532	697	713	732	739	804	857
	894	940	998	1028	1047	1082	1096	1101	1232	1268	1275	1291	1304	1305	1330
	1340	1346	1355	1361	1367	1369	1372	1383	1391	1404	1460	1460	1479	1492	1493
	1521	1522	1577	1579	1580	1583	1586	1591							
CLRB	1211	1223	1479	1577	1579	1592	1593								
CMP	173	186	194	212	215	221	227	236	243	254	262	270	278	288	307
	315	325	333	342	363	371	379	387	396	404	412	420	427	448	480
	499	524	541	553	562	571	581	592	597	607	612	622	633	642	652
	650	671	677	686	700	723	737	793	801	807	815	824	832	840	847
	854	863	871	879	887	897	902	909	917	925	934	943	948	956	962
	1060	1086	1105	1112	1121	1130	1139	1148	1156	1164	1172	1180	1188	1196	1204
	1213	1225	1236	1283	1314	1322	1341	1356	1366	1405	1425	1429	1474	1479	1507
	1512	1577	1578	1579											
CMPB	1577	1578	1579	1592	1593										
DEC	170	400	507	521	538	704	719	733	979	1000	1036	1057	1059	1080	1093
	1240	1311	1336	1392	1362	1416	1430	1492	1460	1510	1503				
DECB	1591	1592													
ENT	13														
HALT	19	176	685	688	1067	1070	1321	1324	1578	1586	1592				
INC	171	439	901	947	1221	1256	1276	1296	1460	1479	1517	1577	1578	1579	1586



INCB	1591	1593													
IOY	292	346	459	490	509	1577	1578	1592							
JMP	13														
JSR	19	20	198	201	246	1457	1459	1460							
MOV	590	603	620	631	640	649	660	742	752	761	770	780	1509	1460	1498
	1499	1501	1578	1579	1592	1593									
	163	165	166	167	179	180	182	183	184	186	188	189	190	192	204
	210	211	214	217	218	224	230	232	233	240	244	251	252	253	259
	260	261	267	268	269	275	276	277	283	284	286	287	295	296	297
	299	304	309	306	312	313	314	322	323	324	330	331	332	337	338
	340	341	349	350	351	353	360	361	362	368	369	370	376	377	378
	384	385	386	393	394	398	401	402	403	409	410	411	417	418	419
	424	425	426	431	433	435	441	444	446	447	454	463	464	465	468
	473	477	479	485	493	496	498	504	513	516	517	518	523	530	533
	534	535	540	540	549	550	552	557	558	559	561	566	567	568	570
	575	576	577	578	580	587	588	589	591	595	596	602	603	604	606
	610	611	617	618	619	621	628	629	630	632	637	638	639	641	646
	647	648	650	656	657	665	666	667	669	675	676	682	690	692	696
	698	699	700	701	711	712	714	715	716	718	730	731	738	741	751
	760	769	779	789	790	792	798	799	800	805	806	812	813	814	821
	822	823	829	830	831	837	838	839	844	845	846	851	852	853	860
	861	862	868	869	870	876	877	878	884	885	886	892	893	895	896
	906	907	908	914	915	916	922	923	924	931	932	933	938	939	941
	942	952	953	954	955	960	961	969	970	971	972	981	982	987	988
	989	990	997	999	1010	1011	1018	1019	1020	1021	1027	1029	1030	1039	1046
	1048	1064	1071	1073	1074	1078	1076	1077	1078	1079	1085	1087	1088	1089	1090
	1091	1092	1099	1104	1107	1108	1109	1111	1116	1117	1118	1120	1125	1126	1127
	1129	1134	1135	1136	1138	1148	1146	1147	1153	1154	1155	1161	1162	1163	1169
	1170	1171	1177	1178	1179	1185	1186	1187	1193	1194	1195	1201	1202	1203	1209
	1210	1212	1219	1220	1222	1224	1231	1233	1234	1235	1239	1242	1243	1250	1251
	1252	1253	1258	1260	1261	1262	1267	1279	1280	1290	1292	1293	1295	1298	1303
	1306	1307	1310	1313	1318	1325	1327	1329	1331	1332	1333	1335	1345	1347	1348
	1349	1351	1359	1360	1368	1371	1373	1375	1381	1382	1385	1389	1390	1395	1398
	1399	1403	1409	1411	1413	1414	1415	1419	1420	1421	1423	1424	1427	1454	1455
	1456	1458	1460	1469	1470	1471	1479	1494	1495	1496	1497	1500	1502	1503	1509
	1515	1516	1520	1577	1578	1579	1580	1583	1586	1591	1592	1593	1594		
NOVB	186	452	1479	1577	1578	1579	1580	1591	1592	1593	1594				
NEG	1479	1506	1591												
NOP	209	1001	1460												
REBET	164	185	187	298	352	551	560	569	579	693	791	1110	1119	1120	1137
	1374	1384	1394	1402	1410	1460									
ROL	1580	1591													
RTI	186	1479	1577	1578	1579	1580	1586	1591	1592						
RTB	1477	1510	1513	1523	1583	1592	1593	1594							
SUB	691	1072	1326	1479	1584	1578	1593								
TRAP	1594														
TST	169	174	186	193	196	199	220	223	226	235	436	1448	1479	1511	1577
	1578	1579	1580	1583	1591	1592	1593	1594							
TSTB	469	973	977	991	1000	1006	1022	1030	1034	1050	1055	1254	1263	1277	1299
	1392	1400	1472	1479	1577	1579	1592	1593							
.ASCII	26														
.ASCIZ	26	202	203	207	1460	1930	1931	1932	1933	1934	1935	1936	1937	1939	1940
	1541	1542	1543	1545	1546	1547	1548	1549	1550	1551	1552	1553	1554	1555	1556
.BLKB	1557	1558	1559	1560	1579	1583	1587								
.BLKW	1579														
	1479														



	26	1468	1578	1579	1591	1593									
.BYTE	26														
.ENABL	4														
.END	1595														
.ENDC	12	13	18	19	23	25	26	185	186	202	203	207	230	241	250
	255	258	263	266	271	274	279	283	289	295	300	303	308	311	316
	321	326	329	334	337	343	349	354	359	364	367	372	375	380	383
	388	392	397	400	405	408	413	416	421	423	428	431	437	441	449
	453	456	461	470	473	481	493	500	513	525	530	542	548	554	557
	563	566	572	575	582	587	593	598	602	608	613	617	623	628	634
	637	643	646	653	659	665	672	678	682	695	707	710	721	724	736
	741	748	751	758	760	767	769	776	779	786	789	794	797	802	808
	811	816	820	825	828	833	836	841	843	848	850	855	859	864	867
	872	875	880	883	888	892	898	905	910	913	918	921	926	930	935
	938	944	951	957	963	967	974	976	978	985	992	995	1002	1004	1007
	1009	1016	1023	1026	1031	1033	1035	1037	1043	1045	1051	1053	1056	1058	1060
	1064	1074	1084	1087	1095	1098	1103	1107	1113	1116	1122	1125	1131	1134	1140
	1144	1149	1152	1157	1160	1165	1168	1173	1176	1181	1184	1189	1192	1197	1200
	1205	1208	1214	1218	1226	1229	1237	1245	1249	1264	1270	1274	1282	1284	1290
	1300	1303	1315	1318	1329	1339	1345	1354	1358	1365	1371	1376	1381	1386	1389
	1396	1398	1406	1409	1426	1432	1447	1460	1479	1577	1578	1579	1580	1583	1586
	1591	1592	1593	1594											
.EQUIV	13														
.EVEN	26	202	203	207	1561	1583	1588	1593							
.IF	12	13	18	19	23	25	26	185	186	202	203	207	230	241	250
	255	258	263	266	271	274	279	283	289	295	300	303	308	311	316
	321	326	329	334	337	343	349	354	359	364	367	372	375	380	383
	388	392	397	400	405	408	413	416	421	423	428	431	437	441	449
	453	456	461	470	473	481	493	500	513	525	530	542	548	554	557
	563	566	572	575	582	587	593	598	602	608	613	617	623	628	634
	637	643	646	653	659	665	672	678	682	695	707	710	721	724	736
	741	748	751	758	760	767	769	776	779	786	789	794	797	802	808
	811	816	820	825	828	833	836	841	843	848	850	855	859	864	867
	872	875	880	883	888	892	898	905	910	913	918	921	926	930	935
	938	944	951	957	963	967	974	976	978	985	992	995	1002	1004	1007
	1009	1016	1023	1026	1031	1033	1035	1037	1043	1045	1051	1053	1056	1058	1060
	1064	1074	1084	1087	1095	1098	1103	1107	1113	1116	1122	1125	1131	1134	1140
	1144	1149	1152	1157	1160	1165	1168	1173	1176	1181	1184	1189	1192	1197	1200
	1205	1208	1214	1218	1226	1229	1237	1245	1249	1264	1270	1274	1282	1284	1290
	1300	1303	1315	1318	1329	1339	1345	1354	1358	1365	1371	1376	1381	1386	1389
	1396	1398	1406	1409	1426	1432	1447	1460	1479	1577	1578	1579	1580	1583	1586
	1591	1592	1593	1594											
.IFF	13	18	23	25	26	186	230	241	250	255	258	263	266	271	274
	279	283	289	295	300	303	308	311	316	321	326	329	334	337	343
	349	354	359	364	367	372	375	380	383	388	392	397	400	405	408
	413	416	421	423	428	431	437	441	449	453	456	461	470	473	481
	493	500	513	525	530	542	548	554	557	563	566	572	575	582	587
	593	598	602	608	613	617	623	628	634	637	643	646	653	659	665
	672	678	682	695	707	710	721	724	736	741	748	751	758	760	767
	769	776	779	786	789	794	797	802	808	811	816	820	825	828	833
	836	841	843	848	850	855	859	864	867	872	875	880	883	888	892
	898	905	910	913	918	921	926	930	935	938	944	951	957	963	967
	974	976	978	985	992	995	1002	1004	1007	1009	1016	1023	1026	1031	1033
	1035	1037	1043	1045	1051	1053	1056	1058	1060	1064	1074	1084	1087	1095	1098
	1103	1107	1113	1116	1122	1125	1131	1134	1140	1144	1149	1152	1157	1160	1165
	1168	1173	1176	1181	1184	1189	1192	1197	1200	1205	1208	1214	1218	1226	1229
	1237	1245	1249	1264	1270	1274	1282	1284	1290	1300	1303	1315	1318	1329	1339



	1345	1354	1358	1365	1371	1376	1381	1386	1389	1396	1398	1406	1409	1426	1432
	1447	1460	1479	1577	1578	1579	1580	1583	1586	1591	1592	1593	1594		
.IFT	202	203	207	1577	1578	1579	1580								
.IFTF	202	203	207	1577	1578	1579	1580								
.IIF	12	18	19	26	186	1460	1577	1578	1579	1583	1592	1594			
.IRP	105	230	250	258	266	274	243	295	303	311	321	329	337	349	359
	367	375	383	392	400	408	416	423	431	441	461	473	493	513	530
	548	557	566	575	587	602	617	628	637	646	665	682	695	710	741
	751	760	769	779	789	797	811	820	828	836	843	850	859	867	875
	883	892	905	913	921	930	938	951	967	985	1016	1043	1064	1074	1087
	1107	1116	1125	1134	1144	1152	1160	1168	1176	1184	1192	1200	1208	1218	1229
	1249	1274	1290	1303	1318	1329	1345	1371	1381	1389	1398	1409	1447	1479	1577
	1580	1586	1593												
.LIST	2	11	13	18	19	26	105	186	202	203	207	230	250	258	266
	274	283	295	303	311	321	329	337	349	359	367	375	383	392	400
	408	416	423	431	441	461	473	493	513	530	548	557	566	575	587
	602	617	628	637	646	665	682	695	710	741	751	760	769	779	789
	797	811	820	828	836	843	850	859	867	875	883	892	905	913	921
	930	938	951	967	985	1016	1043	1064	1074	1087	1107	1116	1125	1134	1144
	1152	1160	1168	1176	1184	1192	1200	1208	1218	1229	1249	1274	1290	1303	1318
	1329	1345	1371	1381	1389	1398	1409	1447	1460	1577	1578	1579	1594		
.MACRO	18	26	186	1594											
.MCALL	7	8	9	10	13	26	186								
.NLIST	1	3	13	18	19	26	105	186	202	203	207	230	250	258	266
	274	283	295	303	311	321	329	337	349	359	367	375	383	392	400
	408	416	423	431	441	461	473	493	513	530	548	557	566	575	587
	602	617	628	637	646	665	682	695	710	741	751	760	769	779	789
	797	811	820	828	836	843	850	859	867	875	883	892	905	913	921
	930	938	951	967	985	1016	1043	1064	1074	1087	1107	1116	1125	1134	1144
	1152	1160	1168	1176	1184	1192	1200	1208	1218	1229	1249	1274	1290	1303	1318
	1329	1345	1371	1381	1389	1398	1409	1447	1460	1577	1578	1579	1594		
.PAGE	26														
.REPT	19	26													
.SBTTL	13	18	19	23	25	26	230	250	250	266	274	283	295	303	311
	321	329	337	349	359	367	375	383	392	400	408	416	423	431	441
	461	473	493	513	530	548	557	566	575	587	602	617	628	637	646
	665	682	695	710	741	751	760	769	779	789	797	811	820	828	836
	843	850	859	867	875	883	892	905	913	921	930	938	951	967	985
	1016	1043	1064	1074	1087	1107	1116	1125	1134	1144	1152	1160	1168	1176	1184
	1192	1200	1208	1218	1229	1249	1274	1290	1303	1318	1329	1345	1371	1381	1389
	1398	1409	1447	1460	1479	1577	1578	1579	1580	1583	1586	1591	1592	1593	1594
.TITLE	12														
.WORD	19	23	25	26	206	1460	1580	1583	1586	1591	1592	1593			

ERRORS DETECTED: 0  
DEFAULT GLOBALS GENERATED: 0

\*DZARAB,DZARAB/CRF=DZARAB.P11  
RUN-TIME: 79 48 8 SECONDS  
RUN-TIME RATIO: 416/136=3.0  
CORE USED: 25K (50 PAGES)