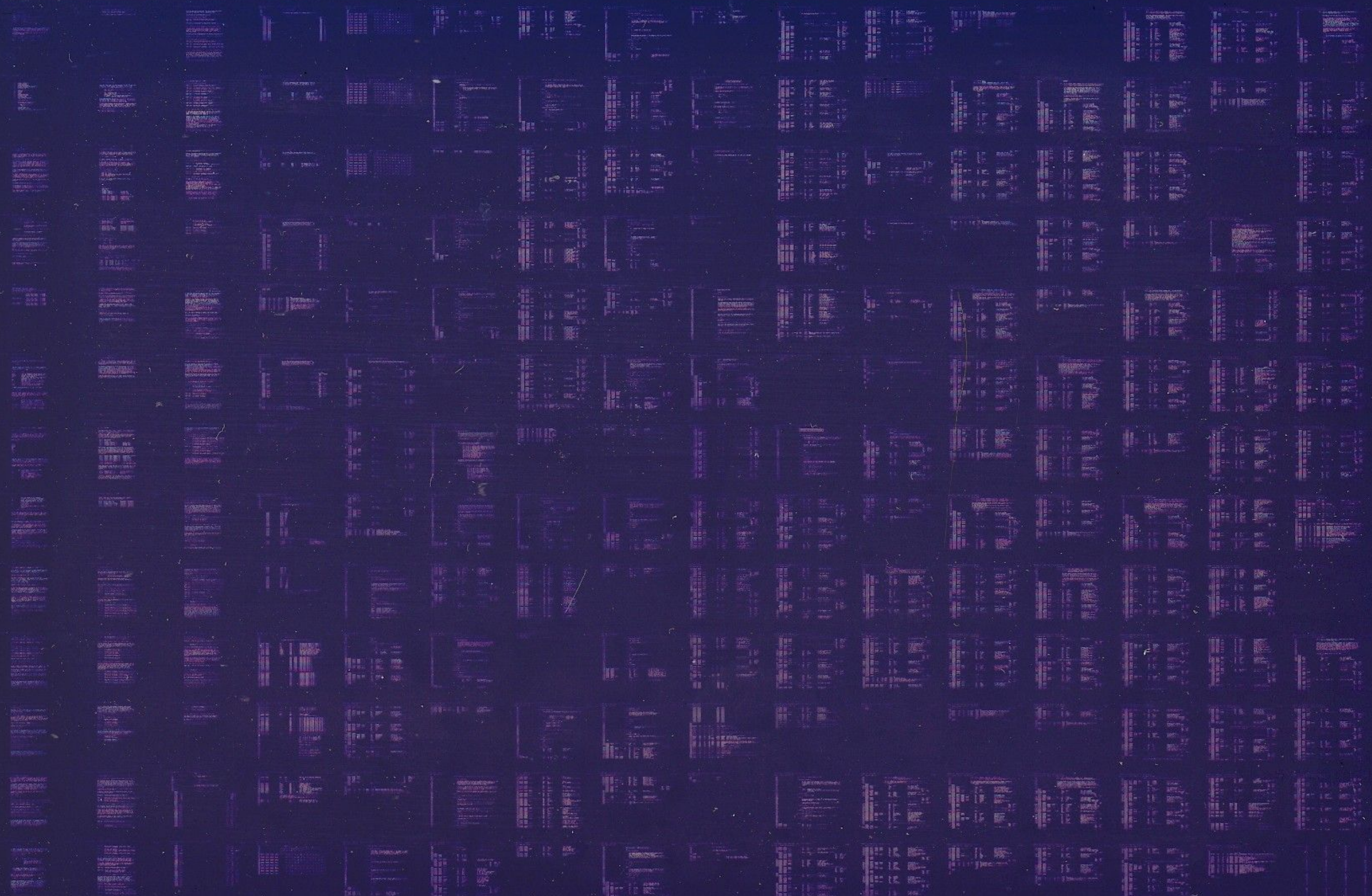


IDV 11
IAV 11

IDV/IAV-11 I/O DIAG
CZIXVAO

COPYRIGHT (c) 1983
AH-T507A-MC
FICHE 01 OF 02

JUL 1984
digital
Made In USA



IDV 11
IAV 11

IDV/IAV-11 I/O DIAG
CZIXVRO

COPYRIGHT (c) 1983
AH-T507A-MC
FICHE 02 OF 02

JUL 1984
digital
Made In USA

The image shows a grid of 16 small diagrams or data tables arranged in a 4x4 pattern on the left side of the page. Each cell contains a small, illegible diagram or table of data. The diagrams appear to be technical drawings or data tables related to the I/O diagnostic mentioned in the header. The text is too small to read, but the layout suggests a structured comparison or analysis of different components or states.

.REM 6

IDENTIFICATION

PRODUCT CODE: AC-T504A-MC
PRODUCT NAME: CZIXVA0 IDV/IAV-11 I/O MOD FAMILY DIAG
PRODUCT DATE: OCT 1983
MAINTAINER: CSS MUNICH
AUTHORS: PETER SEEBACH, DAVE HUNTER

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

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

COPYRIGHT (C) 1983 BY DIGITAL EQUIPMENT CORPORATION

THE FOLLOWING ARE TRADEMARKS OF DIGITAL EQUIPMENT CORPORATION:

DIGITAL
DEC

PDP
DECUS

UNIBUS
DECTAPE

MASSBUS

TABLE OF CONTENTS

1.0	GENERAL INFORMATION
1.1	PROGRAM ABSTRACT
1.2	SYSTEM REQUIREMENTS
1.3	RUNNING THE DIAGNOSTIC ON A FALCON.
1.4	RELATED DOCUMENTS AND STANDARDS
1.5	DIAGNOSTIC HIERARCHY PREREQUISITES
1.6	EXECUTION TIME
2.0	OPERATING INSTRUCTIONS
2.1	COMMANDS
2.2	SWITCHES
2.3	FLAGS
2.4	HARDWARE QUESTIONS
2.5	SOFTWARE QUESTIONS
2.6	EXTENDED P-TABLE DIALOGUE
2.7	CLOCK QUESTIONS
2.8	QUICK STARTUP PROCEDURE
2.9	USING THE XXDP* SETUP UTILITY
3.0	ERROR INFORMATION
4.0	PERFORMANCE AND PROGRESS REPORTS
4.1	PRINT COMMAND UTILIZATION
5.0	DEVICE INFORMATION TABLES
6.0	TEST SUMMARIES
6.1	INTERNAL LOGIC TESTS.
6.2	MANUFACTURING I/O TESTS.
6.3	FIELD INPUT/OUTPUT TESTS USING FIELD TEST CONNECTORS.
6.4	SPECIFICALLY SELECTABLE INPUT/OUTPUT TESTS.

1.0 GENERAL INFORMATION

1.1 PROGRAM ABSTRACT

THE CZIXV?? DIAGNOSTIC PROVIDES A SERIES OF TESTS TO VERIFY THE INTEGRITY AND FUNCTIONALITY OF THE IAV/IDV-11 FAMILY. THIS DIAGNOSTIC CAN BE USED BY FIELD SERVICE FOR FUNCTIONAL TESTING, BY THE ENGINEER FOR DESIGN TESTS, AND BY MANUFACTURING FOR CHECKOUT AND REPAIR.

THE FOLLOWING SPECIAL FEATURES ARE IMPLEMENTED.

THE DIAGNOSTIC IS SET UP FOR FIELD SERVICE SO THAT IF THE USER TYPES "N" TO THE "CHANGE HARDWARE" AND "CHANGE SOFTWARE" QUESTIONS, AN AUTOMATIC CONFIGURATION ROUTINE WILL BE RUN. THIS FINDS ALL DEVICES IN THE ADDRESS RANGE 171000 TO 171770, PRINTS A LIST OF ALL IAV/IDV11 DEVICES FOUND, AND CARRIES OUT THE SELECTED TESTS ON THESE DEVICES. THIS FEATURE HAS SPECIAL SIGNIFICANCE WHEN THE XXDP* SETUP UTILITY IS USED (SEE SECTION 2.9).

NOT ALL OF THE DIAGNOSTIC TESTS ARE RUN DURING EVERY PASS. WHICH TO RUN IS DETERMINED BY THE DEVICE BEING TESTED AND BY THE ANSWERS TO THE SOFTWARE QUESTIONS. FOR MORE INFORMATION, SEE SECTION 6.0.

THE "PRINT" COMMAND CAN BE USED TO OBTAIN A LIST OF TEST TITLES, A PRINTOUT OF THE IAV/IDV-11 CONFIGURATION THAT THE DIAGNOSTIC IS USING, OR A PRINTOUT OF THE ERROR STATISTICS ACCUMULATED BY THE DIAGNOSTIC. HELP ON HOW TO REPEAT THE AUTOCONFIGURATION CAN ALSO BE OBTAINED. FOR MORE INFORMATION, SEE SECTION 4.0.

IF THE EVALUATE FLAG "EVL" IS SET, ANY UNIT ON WHICH MORE THAN 5 ERRORS ARE DETECTED FOLLOWING A "START" COMMAND IS DROPPED FROM TESTING.

THE PROGRAM SUPPORTS UP TO 16 UNITS, ALL SELECTED TESTS BEING RUN ON ONE UNIT BEFORE PROCEEDING TO THE NEXT UNIT.

THIS DIAGNOSTIC HAS BEEN WRITTEN FOR USE WITH THE DIAGNOSTIC RUNTIME SERVICES SOFTWARE (SUPERVISOR). THESE SERVICES PROVIDE THE INTERFACE TO THE OPERATOR AND TO THE SOFTWARE ENVIRONMENT. THIS PROGRAM CAN BE USED WITH XXDP*.

FOR A COMPLETE DESCRIPTION OF THE RUNTIME SERVICES, REFER TO THE XXDP* USER'S MANUAL. THERE IS A BRIEF DESCRIPTION OF THE RUNTIME SERVICES IN SECTION 2 OF THIS DOCUMENT.

1.2 SYSTEM REQUIREMENTS

- A. LSI PROCESSOR WITH A MINIMUM OF 28K OF MEMORY.
- B. CONSOLE TERMINAL WITH INTERFACE ADDRESS 777560.

C. XXDP+ LOAD DEVICE (RX,RK,RL ECT.)

D. IDV/IAV-11 MODULES TO BE CHECKED.

E. FIELD CHECKOUT : FOR DIGITAL MODULES, DIGITAL TEST CONNECTOR
2G-M002A.

FOR ANALOGUE MODULES, CALIBRATED VOLTAGE
SOURCE, MEASURING EQUIPMENT AND ANALOGUE
TEST CONNECTOR.

MANUFACTURING : FOR DIGITAL MODULES, DATA LOOPBACK EQUIPMENT.
CHECKOUT

FOR ANALOGUE MODULES, CALIBRATED VOLTAGE
SOURCE AND MEASURING EQUIPMENT.

1.3 RUNNING THE DIAGNOSTIC ON A FALCON

TO RUN THE DIAGNOSTIC ON A FALCON BASED SYSTEM, A BOOTSTRAP
PROGRAM IS NEEDED IN ADDITION TO THE ABOVE REQUIERMENTS.
THIS COULD BE IN THE FALCON MACRO ODT ROM (KXT11-A2), OR ON
A MXV-11 BOARD.

IF YOU HAVE INSTALLED THE FALCON MACRO-ODT ROM KXT11-A2 FOR
BOOTING THE XXDP+ MEDIA WITH THE DIAGNOSTIC ON IT, SOME OF THE
IAV/IDV-11 DEFAULT ADDRESSES ARE USED, PREVENTING THE
DIAGNOSTICS AUTOMATIC CONFIGURATION ROUTINE FROM WORKING
CORRECTLY. TO USE THE DIAGNOSTIC, THE ADDRESSES MUST BE ENTERED
MANUALLY USING THE STARTUP QUESTIONS.

NOTE:

ONES THE XXDP+ MEDIA IS BOOTED, THE CONSOLE "BREAK" KEY SHOULD
NOT BE PRESSED AS IT MAY CAUSE ERROR MESSAGES TO BE PRINTED.

1.4 RELATED DOCUMENTS AND STANDARDS

XXDP+ USER MANUAL (CHQUS)
IAV/IDV-11 OPTION DESCRIPTION YG-C03NC-00

1.5 DIAGNOSTIC HIERARCHY PREREQUISITES

BEFORE RUNNING THIS DIAGNOSTIC, THE APPROPRIATE LSI-11 CPU,
MEMORY AND PERIPHERAL STANDARD DIAGNOSTICS SHOULD BE RUN TO VERIFY
CORRECT OPERATION OF THE SYSTEM.

1.6 EXECUTION TIME

EXECUTION TIMES LISTED BELOW ARE FOR A WHOLE PASS IN NO QUICK

VERIFY MODE WITH "UAM" FLAG SELECTED.
MANY OF THE TESTS NEED MANUAL INTERVENTIONS, SO IT MAKES NO
SENSE TO GIVE THE TIMES WITH UAM NOT SET.
EXECUTION TIMES VARY WITH THE CPU TYPE. THE FOLLOWING TIMES
ARE TYPICAL ON A PDP-11/23 SYSTEM :

FIELD SERVICE TESTS -	1 DIGITAL INPUT MODULE	18 SECONDS
	1 DIGITAL OUTPUT MODULE	18 SECONDS
	1 ANALOGUE INPUT MODULE	18 SECONDS
	1 ANALOGUE OUTPUT MODULE	18 SECONDS
MANUFACTURING TESTS -	1 DIGITAL INPUT MODULE	18 SECONDS
(BASIC LOGIC ONLY)	1 DIGITAL OUTPUT MODULE	18 SECONDS
	1 ANALOGUE INPUT MODULE	18 SECONDS
	1 ANALOGUE OUTPUT MODULE	18 SECONDS
MANUFACTURING TESTS -	1 DIGITAL INPUT MODULE	270 SECONDS
(INCL. LOOPBACK OR	1 DIGITAL OUTPUT MODULE	110 SECONDS
SETUP TESTS)	1 ANALOGUE INPUT MODULE	18 SECONDS
	1 ANALOGUE OUTPUT MODULE	18 SECONDS

2.0 OPERATING INSTRUCTIONS

THIS SECTION CONTAINS A BRIEF DESCRIPTION OF THE RUNTIME SERVICES. FOR DETAILED INFORMATION, REFER TO THE XXDP+ USER'S MANUAL (CHQUS).

2.1 COMMANDS

THERE ARE ELEVEN LEGAL COMMANDS FOR THE DIAGNOSTIC RUNTIME SERVICES (SUPERVISOR). THIS SECTION LISTS THE COMMANDS AND GIVES A VERY BRIEF DESCRIPTION OF THEM. THE XXDP+ USER'S MANUAL HAS MORE DETAILS.

COMMAND	EFFECT
START	START THE DIAGNOSTIC FROM AN INITIAL STATE
RESTART	START THE DIAGNOSTIC WITHOUT INITIALIZING
CONTINUE	CONTINUE AT TEST THAT WAS INTERRUPTED (AFTER ↑C)
PROCEED	CONTINUE FROM AN ERROR HALT
EXIT	RETURN TO XXDP+ MONITOR (XXDP+ OPERATION ONLY!)
ADD	ACTIVATE A UNIT FOR TESTING (ALL UNITS ARE CONSIDERED TO BE ACTIVE AT START TIME)
DROP	DEACTIVATE A UNIT
PRINT	PRINT TEST TITLES, IAV/IDV-11 CONFIGURATION, ERROR STATISTICS, OR HOW TO RECONFIGURE. (SEE SECTION 4.0)
DISPLAY	TYPE A LIST OF ALL DEVICE INFORMATION
FLAGS	TYPE THE STATE OF ALL FLAGS (SEE SECTION 2.3)
ZFLAGS	CLEAR ALL FLAGS (SEE SECTION 2.3)

A COMMAND CAN BE RECOGNIZED BY THE FIRST THREE CHARACTERS. SO YOU MAY, FOR EXAMPLE, TYPE "STA" INSTEAD OF "START".

2.2 SWITCHES

THERE ARE SEVERAL SWITCHES WHICH ARE USED TO MODIFY SUPERVISOR OPERATION. THESE SWITCHES ARE APPENDED TO THE LEGAL COMMANDS. ALL OF THE LEGAL SWITCHES ARE TABULATED BELOW WITH A BRIEF DESCRIPTION OF EACH. IN THE DESCRIPTIONS BELOW, A DECIMAL NUMBER IS DESIGNATED BY "DDDD".

SWITCH	EFFECT
/TESTS:LIST	EXECUTE ONLY THOSE TESTS SPECIFIED IN THE LIST. LIST IS A STRING OF TEST NUMBERS, FOR EXAMPLE - /TESTS:1:5:7-10. THIS LIST WILL CAUSE TESTS 1,5,7,8,9,10 TO BE RUN. ALL OTHER TESTS WILL NOT BE RUN.
/PASS:DDDD	EXECUTE DDDDD PASSES (DDDD = 1 TO 64000)
/FLAGS:FLGS	SET SPECIFIED FLAGS. FLAGS ARE DESCRIBED IN SECTION 2.3.
/EOP:DDDD	REPORT END OF PASS MESSAGE AFTER EVERY DDDDD PASSES ONLY. (DDDD = : TO 64000)
/UNITS:LIST	TEST/ADD/DROP ONLY THOSE UNITS SPECIFIED

IN THE LIST. LIST EXAMPLE - /UNITS:0:5:10-12
 USE UNITS 0,5,10,11,12 (UNIT NUMBERS = 0-63)

EXAMPLE OF SWITCH USAGE:

START/TESTS:1-5/PASS:1000/EOP:100

THE EFFECT OF THIS COMMAND WILL BE: 1) TESTS 1 THROUGH 5 WILL BE EXECUTED, 2) ALL UNITS WILL TESTED 1000 TIMES AND 3) THE END OF PASS MESSAGES WILL BE PRINTED AFTER EACH 100 PASSES ONLY. A SWITCH CAN BE RECOGNIZED BY THE FIRST THREE CHARACTERS. YOU MAY, FOR EXAMPLE, TYPE "/TES:1-5" INSTEAD OF "/TESTS:1-5".

BELOW IS A TABLE THAT SPECIFIES WHICH SWITCHES CAN BE USED BY EACH COMMAND.

	TESTS	PASS	FLAGS	EOP	UNITS
START	X	X	X	X	X
RESTART	X	X	X	X	X
CONTINUE		X	X	X	
PROCEED			X		
DROP					X
ADD					X
PRINT					
DISPLAY					X
FLAGS					
ZFLAGS					
EXIT					

2.3 FLAGS

FLAGS ARE USED TO SET UP CERTAIN OPERATIONAL PARAMETERS SUCH AS LOOPING ON ERROR. ALL FLAGS ARE CLEARED AT STARTUP AND REMAIN CLEARED UNTIL EXPLICITLY SET USING THE FLAGS SWITCH. FLAGS ARE ALSO CLEARED AFTER A START COMMAND UNLESS SET USING THE FLAG SWITCH. THE ZFLAGS COMMAND MAY ALSO BE USED TO CLEAR ALL FLAGS.

WITH THE EXCEPTION OF THE START AND ZFLAGS COMMANDS, NO COMMANDS AFFECT THE STATE OF THE FLAGS; THEY REMAIN SET OR CLEARED AS SPECIFIED BY THE LAST FLAG SWITCH.

FLAG	EFFECT
HOE	HALT ON ERROR - CONTROL IS RETURNED TO RUNTIME SERVICES COMMAND MODE
LOE	LOOP ON ERROR
IER*	INHIBIT ALL ERROR REPORTS
IBE*	INHIBIT ALL ERROR REPORTS EXCEPT FIRST LEVEL (FIRST LEVEL CONTAINS ERROR TYPE, NUMBER, PC, TEST AND UNIT)
IXE*	INHIBIT EXTENDED ERROR REPORTS (THOSE CALLED BY PRINTX MACROS)
PRI	DIRECT MESSAGES TO LINE PRINTER
BOE	"BELL" ON ERROR

PNT PRINT TEST NUMBER AS TEST EXECUTES
 UAM UNATTENDED MODE (NO MANUAL INTERVENTION)
 (IF SET, TEST 15,16,17,18 WILL NOT RUN)
 ISR INHIBIT STATISTICAL REPORTS (NOT APPLICABLE)
 IDR INHIBIT PROGRAM DROPPING OF UNITS (NOT
 REQUIRED SINCE UNITS ARE ONLY DROPPED IF
 EVL IS USED.
 ADR EXECUTE AUTODROP CODE
 LOT LOOP ON TEST
 EVL EXECUTE EVALUATION IE. DROP UNIT IF MORE
 THAN 5 ERRORS OCCUR AFTER A START OR RESTART
 COMMAND.

*ERROR MESSAGES ARE DESCRIBED IN SECTION 3.1

SEE THE XXDP* USER'S MANUAL FOR MORE DETAILS ON FLAGS. YOU MAY SPECIFY MORE THAN ONE FLAG WITH THE FLAG SWITCH. FOR EXAMPLE, TO CAUSE THE PROGRAM TO LOOP ON ERROR, INHIBIT ERROR REPORTS AND TYPE A "BELL" ON ERROR, YOU MAY USE THE FOLLOWING STRING:

/FLAGS:LOE:IER:BOE

2.4 HARDWARE QUESTIONS

WHEN THE DIAGNOSTIC IS STARTED, THE RUNTIME SERVICES WILL PROMPT THE USER FOR HARDWARE INFORMATION BY TYPING "CHANGE HW (L) ?"

THIS DIAGNOSTIC HAS BEEN PRELOADED SUCH THAT IF YOU ANSWER "N" TO THE QUESTION, IT WILL AUTOMATICALLY SEARCH FOR UP TO 16 IAV/IDV11 UNITS IN THE ADDRESS RANGE 171000 TO 171770. DEFAULT VECTORS EQUAL TO THE LOW 9 ADDRESS BITS WILL BE ASSUMED FOR MODULES WITH ADDRESSES OVER 171400.

TO RUN THE DIAGNOSTIC WITH SPECIFIC MODULES WHICH NEED NOT BE IN THE ABOVE ADDRESS RANGE, YOU MUST ANSWER "Y" TO THE "CHANGE HARDWARE" QUESTION. THE RUNTIME SERVICES WILL THEN ASK FOR THE NUMBER OF UNITS (IN DECIMAL). TO KEEP DOWN MEMORY REQUIREMENTS, THE MAXIMUM NUMBER OF UNITS SUPPORTED IS 16. YOU WILL THEN BE ASKED THE FOLLOWING QUESTIONS FOR EACH UNIT:

MODE REGISTER ADDRESS (O) 171000 ?

IN REPLY, YOU SHOULD ENTER AN ADDRESS IN OCTAL IN THE RANGE 160000 TO 177770.

VECTOR ADDRESS (FOR OUTPUT MODULES, TYPE "O") (O) 0 ?

PRIORITY LEVEL (FOR OUTPUT MODULES, TYPE "O") (O) 4 ?
 (FOR LSI WITH FIXED PRI. USE LEVEL 4 ONLY)

FOR OUTPUT MODULES (WHICH HAVE NO INTERRUPT LOGIC) YOU SHOULD TYPE "O" FOR THE ABOVE TWO QUESTIONS. THIS CAUSES THE DIAGNOSTIC PRINT ROUTINE TO OUTPUT "NONE" FOR THE VECTOR ADDRESS OF THE MODULES.

IS THIS MODULE CONNECTED TO ANOTHER VIA A LOOPBACK CABLE
(ONLY MANUFACTURING SHOULD ANSWER YES) (L) N ?

IF YOU ANSWER "N" TO THIS QUESTION, NO MORE HARDWARE QUESTIONS
ARE ASKED. THIS WILL BE THE NORMAL ANSWER FOR FIELD SERVICE AS
SPECIAL SIGNAL LOOPING EQUIPMENT IS REQUIRED FOR LOOPBACK TESTS.
IF YOU ANSWER "Y", THE FOLLOWING ARE ALSO ASKED:

LOW LEVEL INPUT FOR DIGITAL INPUT LOOPBACK TESTS
(FOR OUTPUT LOOPBACK TYPE "N") (L) N ?

THIS SELECTS WHETHER TO USE THE 0-5 VOLT OR 0-60 VOLT
SENSITIVITY OF THE DIGITAL INPUT MODULES FOR THE INPUT LOOPBACK TESTS.
TO FULLY TEST THE MODULES, SEVERAL PASSES SHOULD BE MADE AT BOTH
LEVELS. FOR OUTPUT LOOPBACK TESTS THE 0-60 VOLT SENSITIVITY MUST
BE USED ("N" HAS TO BE TYPED).
IF "Y" IS TYPED, THE NEXT QUESTION IS OMITTED.
OTHERWISE, THE DIAGNOSTIC ASKS:

DEBOUNCE PERIOD FOR DIGITAL INPUT LOOPBACK TESTS
500US (0), 5 MS (1), 10MS (2) OR ALL (3) (S) 3 ?

IF THE HIGH LEVEL (0-60 VOLTS) IS USED ON THE DIGITAL INPUT
MODULES, THE HARDWARE CAN BE PROGRAMMED TO WAIT FOR A SPECIFIED
INTERVAL BEFORE INTERRUPTING WITH NEW INPUT DATA. THIS QUESTION
ALLOWS YOU TO SPECIFY THAT A PARTICULAR INTERVAL BE USED. FOR
FULL TESTING OF THE MODULES, "3" SHOULD BE TYPED.
FOR OUTPUT LOOPBACK TESTS THE ANSWERING OF THIS QUESTION HAS NO
AFFECT. A FIXED 500 US INPUT DEBOUNCE WILL ALWAYS BE USED.

OTHER MODULE MODE REGISTER ADDRESS (0) 171400 ?

IN REPLY, YOU SHOULD ENTER AN ADDRESS IN OCTAL IN THE RANGE
160000 TO 177770. THE MODULE AT THIS ADDRESS WILL BE USED AS THE
OTHER HALF OF A LOOPED PAIR, BUT WILL NOT ITSELF BE TESTED
UNLESS IT IS ALSO ONE OF THE MODULES SPECIFIED BY THE FIRST
HARDWARE QUESTION UNDER ANOTHER UNIT NUMBER. IF THE UNIT IS
CONNECTED TO SEVERAL OTHER MODULES, ONLY ONE WILL BE USED FOR
LOOPBACK TESTING.

OTHER MODULE VECTOR ADDRESS (IF OUTPUT MODULE, TYPE "0") (0) 400 ?

FOR OUTPUT MODULES (WHICH HAVE NO INTERRUPT LOGIC) YOU SHOULD
TYPE "0" FOR THE ABOVE QUESTION. THIS CAUSES THE DIAGNOSTIC
PRINT ROUTINE TO OUTPUT "NONE" FOR THE VECTOR ADDRESS OF THE
MODULE.

THE FOLLOWING ILLUSTRATES THE RESPONSE TO THE HARDWARE
QUESTIONS. IN THIS EXAMPLE, THE USER RESPONSE IS UNDERLINED :-

CHANGE HARDWARE (L) ? Y <CR>

#UNITS (D) ? 2 <CR>

UNIT 0

MODE REGISTER ADDRESS (0) 171000 ? <CR>

```

VECTOR ADDRESS (FOR OUTPUT MODULES, TYPE "0") (0) 0 ? <CR>
PRIORITY LEVEL (FOR OUTPUT MODULES, TYPE "0") (0) 4 ? 0 <CR>
(FOR LSI WITH FIXED PRI. USE LEVEL 4 ONLY)
IS THIS MODULE CONNECTED TO ANOTHER VIA A LOOPBACK CABLE
(ONLY MANUFACTURING SHOULD ANSWER YES) (L) N ? <CR>
UNIT 1
MODE REGISTER ADDRESS (0) 171000 ? 171400 <CR>
VECTOR ADDRESS (FOR OUTPUT MODULES, TYPE RETURN) (0) 0 ? 400 <CR>
PRIORITY LEVEL (FOR OUTPUT MODULES, TYPE RETURN) (0) 0 ? 4 <CR>
(FOR LSI WITH FIXED PRI. USE LEVEL 4 ONLY)
IS THIS MODULE CONNECTED TO ANOTHER VIA A LOOPBACK CABLE
(ONLY MANUFACTURING SHOULD ANSWER YES) (L) N ? Y <CR>
LOW LEVEL INPUT FOR DIGITAL INPUT LOOPBACK TESTS
(FOR OUTPUT LOOPBACK TYPE "N") (L) N ? <CR>
DEBOUNCE PERIOD FOR DIGITAL INPUT LOOPBACK TESTS
500US (0), 5 MS (1), 10MS (2) OR ALL (3) (S) 3 ? <CR>
OTHER MODULE MODE REGISTER ADDRESS (0) 171000 ? <CR>
OTHER MODULE VECTOR ADDRESS (IF OUTPUT MODULE, TYPE "0") (0) 0 ? <CR>

```

IN THIS EXAMPLE, THE INTERNAL LOGIC OF AN OUTPUT MODULE AT ADDRESS 171000 WILL BE TESTED. THE MODULE WILL THEN BE USED AS THE SIGNAL SOURCE TO TEST AN INPUT MODULE AT ADDRESS 171400.

FOR THE DIGITAL LOOPBACK TESTS THE LOGIC LEVELS USED WILL BE 0 VOLTS FOR THE LOW LEVEL AND UP TO 60 VOLTS FOR THE HIGH LEVEL. IF INPUT LOOPBACK TESTS ARE RUNNING ALL DEBOUNCE INTERVALS WILL BE USED.

NOTICE THAT THE DEFAULT VALUE FOR THE PRIORITY LEVEL CHANGES WHEN A NON-DEFAULT RESPONSE IS GIVEN. THIS IS TRUE FOR ALL OF THE HARDWARE QUESTIONS, SO BE CAREFUL WHEN SPECIFYING MULTIPLE UNITS!

2.5 SOFTWARE QUESTIONS

AFTER YOU HAVE ANSWERED THE HARDWARE QUESTIONS OR AFTER A RESTART OR CONTINUE COMMAND, THE RUNTIME SERVICES WILL ASK FOR SOFTWARE PARAMETERS. THESE PARAMETERS GOVERN THE DIAGNOSTIC OPERATING MODES. YOU WILL BE PROMPTED BY "CHANGE SW (L) ?". THE NORMAL RESPONSE FOR FIELD SERVICE IS TO TYPE "N". THIS CAUSES THE DEVICE INTERNAL LOGIC TESTS AND FIELD INPUT/OUTPUT TESTS TO BE RUN.
IN A MANUFACTURING ENVIRONMENT, THE PARAMETERS SHOULD BE CHANGED

BY TYPING "Y". THE FOLLOWING QUESTIONS WILL THEN BE ASKED :

RUN MANUFACTURING TESTS (L) N ?

ANSWERING "N" TO THIS QUESTION CAUSES THE INTERNAL LOGIC AND FIELD INPUT/OUTPUT TESTS TO BE RUN. THE NEXT SOFTWARE QUESTION WILL BE "QUICK VERIFY ?". ANSWERING "Y" CAUSES THE FOLLOWING QUESTIONS TO BE ASKED:

RUN LOOPBACK AND I/O TESTS (L) Y ?

IF "Y" IS TYPED, LOOPBACK TESTS WILL BE RUN FOR DIGITAL MODULES AND INPUT/OUTPUT TESTS WILL BE RUN FOR ANALOGUE MODULES. ANSWERING "N" CAUSES ONLY THE INTERNAL LOGIC TESTS TO BE RUN AND THE NEXT QUESTION TO BE "QUICK VERIFY MODE ?". THIS IS INTENDED FOR A QUICK TEST OF THE MODULE INTEGRITY BEFORE CONNECTING UP TEST EQUIPMENT FOR FULL TESTS.

FIRST PATTERN FOR DIGITAL LOOPBACK TESTS (O) 177777 ?
SECOND PATTERN FOR DIGITAL LOOPBACK TESTS (O) 0 ?

THESE WILL BE USED AS ALTERNATING DATA PATTERNS IN DIGITAL LOOPBACK TESTS 8 AND 9.

QUICK VERIFY MODE (L) N ?

IF THE ANSWER TO THIS QUESTION IS "Y", ONLY ONE ITERATION OF EACH TEST WILL BE PERFORMED. OTHERWISE, SOME TESTING IS DONE MORE THAN ONCE. REPEATEDLY TESTING A PIECE OF LOGIC IN THIS WAY OFTEN DETECTS FAULTS WHICH A SINGLE TEST WOULD NOT. THEREFORE, TO FULLY TEST THE HARDWARE, THE ANSWER TO THIS QUESTION SHOULD BE "N".

THE FOLLOWING ILLUSTRATES THE RESPONSE TO THE SOFTWARE QUESTIONS. THE USER RESPONSE IS UNDERLINED:

```
CHANGE SOFTWARE (L) ? Y <CR>
-----
RUN MANUFACTURING TESTS (L) N ? Y <CR>
-----
RUN LOOPBACK AND I/O TESTS (L) Y ? Y <CR>
-----
FIRST PATTERN FOR DIGITAL LOOPBACK TESTS (O) 177777 ? 125252 <CR>
-----
SECOND PATTERN FOR DIGITAL LOOPBACK TESTS (O) 0 ? 52525 <CR>
-----
QUICK VERIFY MODE (L) N ? <CR>
-----
```

IN THIS EXAMPLE, FULL MANUFACTURING TESTS ARE TO BE RUN, INCLUDING LOOPBACK TESTS FOR ANY DIGITAL MODULES AND INPUT/OUTPUT TESTS FOR ANY ANALOGUE MODULES. IN THE DIGITAL LOOPBACK TESTS WITH ALTERNATING PATTERNS, THE USER HAS CHOSEN TO SWITCH ALTERNATE BITS OF THE SIGNAL RATHER THAN TO ACCEPT THE DEFAULT OF ALL ONES THEN ALL ZEROS.

2.6 EXTENDED P-TABLE DIALOGUE

WHEN YOU ANSWER THE HARDWARE QUESTIONS, YOU ARE BUILDING ENTRIES IN A TABLE THAT DESCRIBES THE DEVICES UNDER TEST. THE SIMPLEST WAY TO BUILD THIS TABLE IS TO ANSWER ALL QUESTIONS FOR EACH UNIT TO BE TESTED. IF YOU ARE TESTING SEVERAL IDENTICAL DEVICES, THIS BECOMES TEDIOUS SINCE MOST OF THE ANSWERS ARE THE SAME FOR EACH UNIT.

TO ILLUSTRATE A MORE EFFICIENT METHOD, SUPPOSE YOU ARE TESTING FOUR DIGITAL INPUT MODULES ALL CONNECTED VIA SPECIAL TEST EQUIPMENT TO ONE DIGITAL OUTPUT MODULE FOR LOOPBACK TESTS. YOU COULD ANSWER THE HARDWARE QUESTIONS FOR EACH OF THE FOUR UNITS AS SHOWN IN SECTION 2.4. HOWEVER, APART FROM THE "MODE REGISTER" AND "VECTOR ADDRESS", THE ANSWERS ARE THE SAME FOR ALL OF THE UNITS. THIS PROCEDURE IS NOT VERY EFFICIENT.

THE RUNTIME SERVICES CAN TAKE MULTIPLE UNIT SPECIFICATIONS HOWEVER. LET'S BUILD THE SAME TABLE USING THE MULTIPLE SPECIFICATION FEATURE:

```
CHANGE HARDWARE (L) ? Y <CR>
#UNITS (D) ? 4 <CR>
UNIT 0
MODE REGISTER ADDRESS (O) 171000 ? 171400,171410,171420,171430 <CR>
VECTOR ADDRESS (FOR OUTPUT MODULES, TYPE "O") (O) 0 ? 400,410,420,430<CR>
PRIORITY LEVEL (FOR OUTPUT MODULES, TYPE "O") (O) 4 ? <CR>
(FOR LSI WITH FIXED PRI. USE LEVEL 4 ONLY)
IS THIS MODULE CONNECTED TO ANOTHER VIA A LOOPBACK CABLE
(ONLY MANUFACTURING SHOULD ANSWER YES) (L) N ? Y <CR>
LOW LEVEL INPUT FOR DIGITAL INPUT LOOPBACK TESTS
(FOR OUTPUT LOOPBACK TYPE "N") (L) N ? <CR>
DEBOUNCE PERIOD FOR DIGITAL INPUT LOOPBACK TESTS
500US (0), 5 MS (1), 10MS (2) OR ALL (3) (S) 3 ? <CR>
OTHER MODULE MODE REGISTER ADDRESS (O) 171400 ? 171000 <CR>
OTHER MODULE VECTOR ADDRESS (IF OUTPUT MODULE, TYPE "O") (O) 400 ? 0 <CR>
```

AS YOU CAN SEE IN THE ABOVE DIALOGUE, THE RUNTIME SERVICES WILL BUILD AS MANY ENTRIES AS IT CAN WITH THE INFORMATION GIVEN IN ANY ONE PASS THROUGH THE QUESTIONS. IN THIS EXAMPLE, THE 4 INPUT MODULES AT ADDRESSES 174000, 174010, 174020 AND 174030 ARE SET UP WITH VECTORS 400,410,420 AND 430. ALL HAVE PRIORITY LEVEL 4 AND ALL USE HIGH LEVEL WITH ALL THREE DEBOUNCE PERIODS AND ALL ARE CONNECTED TO THE SAME OUTPUT MODULE AT ADDRESS 171000.

VECTOR ADDRESS (FOR OUTPUT MODULES, TYPE "0") (0) 0 ? 0 <CR>

PRIORITY LEVEL (FOR OUTPUT MODULES, TYPE "0") (0) 4 ? 0 <CR>
(FOR LSI WITH FIXED PRI. USE LEVEL 4 ONLY)

IS THIS MODULE CONNECTED TO ANOTHER VIA A LOOPBACK CABLE
(ONLY MANUFACTURING SHOULD ANSWER YES) (L) N ? <CR>

3.0 ERROR INFORMATION

3.1 TYPES OF ERROR MESSAGES

THERE ARE THREE LEVELS OF ERROR MESSAGES THAT MAY BE ISSUED BY A DIAGNOSTIC : GENERAL, BASIC AND EXTENDED. GENERAL ERROR MESSAGES ARE ALWAYS PRINTED UNLESS THE "IER" FLAG IS SET (SECTION 2.3). THE GENERAL ERROR MESSAGE IS OF THE FORM :

NAME TYPE NUMBER ON UNIT NUMBER TST NUMBER PC:xxxxxx
ERROR MESSAGE

WHERE NAME = DIAGNOSTIC NAME
TYPE = ERROR TYPE (SYS FATAL, DEV FATAL, HARD OR SOFT)
NUMBER = ERROR NUMBER
UNIT NUMBER = 0 - N (N IS LAST UNIT IN PTABLE)
TST NUMBER = TEST AND SUBTEST WHERE ERROR OCCURRED
PC:xxxxxx = ADDRESS OF ERROR MESSAGE CALL

BASIC ERROR MESSAGES ARE MESSAGES THAT CONTAIN SOME ADDITIONAL INFORMATION ABOUT THE ERROR. THESE ARE ALWAYS PRINTED UNLESS THE "IER" OR "IBR" FLAGS ARE SET (SECTION 2.3). THESE MESSAGES ARE PRINTED AFTER THE ASSOCIATED GENERAL MESSAGE.

EXTENDED ERROR MESSAGES CONTAIN SUPPLEMENTARY ERROR INFORMATION SUCH AS REGISTER CONTENTS OR GOOD/BAD DATA. THESE ARE ALWAYS PRINTED UNLESS THE "IER", "IBR" OR "IXR" FLAGS ARE SET (SECTION 2.3). THESE MESSAGES ARE PRINTED AFTER THE ASSOCIATED GENERAL ERROR MESSAGE AND ANY ASSOCIATED BASIC ERROR MESSAGES.

3.2 SPECIFIC ERROR MESSAGES

ALL SPECIFIC ERROR MESSAGES ARE EXPLAINED WITH THE TEST DESCRIPTIONS IN SECTION 6.0.

4.0 PERFORMANCE AND PROGRESS REPORTS

AT THE END OF EACH PASS, THE PASS COUNT IS GIVEN ALONG WITH THE TOTAL NUMBER OF ERRORS REPORTED SINCE THE DIAGNOSTIC WAS STARTED. THE "EOP" SWITCH CAN BE USED TO CONTROL HOW OFTEN THE END OF PASS MESSAGE IS PRINTED. SECTION 2.2 DESCRIBES SWITCHES.

4.1 PRINT COMMAND UTILIZATION

THE "PRINT" COMMAND CAN BE USED TO FIND OUT HOW MANY ERRORS HAVE OCCURRED ON EACH UNIT SINCE THE DIAGNOSTIC WAS STARTED.

IN ADDITION, THE COMMAND CAN BE USED TO DISPLAY THE CONFIGURATION THAT THE DIAGNOSTIC IS CURRENTLY USING, TO PRINT A LIST OF TEST TITLES, OR TO SHOW HOW TO MAKE THE DIAGNOSTIC REESTABLISH THE CONFIGURATION. THE FOLLOWING EXAMPLES SHOW HOW THE PRINT COMMAND CAN BE USED. USER INPUT IS UNDERLINED :

PRINT <CR>

TYPE T,R,C,S OR HELP (S) H ? <CR>

THE FOLLOWING COMMANDS ARE ACCEPTED :-

T - PRINT TEST TITLES
 R - PRINT HOW TO REESTABLISH THE SYSTEM CONFIGURATION
 C - PRINT CONFIGURATION TABLE CURRENTLY USED BY DIAGNOSTIC
 S - PRINT STATISTICS TABLE
 TYPE T,R,C,S OR HELP (S) H ?

IF YOU TYPE "H", "HELP" OR ANY CHARACTER OTHER THEN "T", "R", "C" OR "S", THE ROUTINE PRINTS THE ABOVE HELP MESSAGE LISTING THE ACCEPTABLE COMMANDS.

PRINT <CR>

TYPE T,R,C,S OR HELP (S) H ? T <CR>

TEST TITLES.

1	REGISTER NXM TEST	
2	RESET TEST	
3	R-W BIT TEST	
4	INTERNAL INTERRUPT TEST	- DIGITAL INPUT
5	INTERNAL LOGIC TEST	- ANALOGUE INPUT
6	INTERNAL INTERRUPT TEST	- ANALOGUE INPUT
7	INTERNAL LOGIC TEST	- ANALOGUE OUTPUT
8	DIGITAL INPUT LOOPBACK TEST	- PATTERN PAIR SELECTABLE
9	DIGITAL OUTPUT LOOPBACK TEST	- PATTERN PAIR SELECTABLE
10	DIGITAL INPUT LOOPBACK TEST	- RANDOM PATTERN
11	DIGITAL OUTPUT LOOPBACK TEST	- RANDOM PATTERN
12	DIGITAL INPUT LOOPBACK TEST	- SLIDING PATTERN
13	DIGITAL OUTPUT LOOPBACK TEST	- SLIDING PATTERN

- 14 DIGITAL INPUT LOOPBACK TEST - INTERRUPT LINE TEST
- 15 ANALOGUE INPUT TEST - FIELD AND MANUFACTURING
- 16 ANALOGUE OUTPUT TEST - FIELD AND MANUFACTURING
- 17 DIGITAL INPUT TEST - FIELD TEST
- 18 DIGITAL OUTPUT TEST - FIELD TEST
- 19 DIGITAL INPUT TEST - SPECIFICALLY SELECTABLE
- 20 DIGITAL OUTPUT TEST - SPECIFICALLY SELECTABLE
- 21 ANALOGUE INPUT (DYNAMIC) - SPECIFICALLY SELECTABLE
- 22 ANALOGUE OUTPUT (DYNAMIC) - SPECIFICALLY SELECTABLE
- 23 ANALOGUE OUTPUT (CALIBR.) - SPECIFICALLY SELECTABLE
- 24 ANALOGUE INPUT (CALIBR.) - SPECIFICALLY SELECTABLE
- 25 VISUAL LED TEST - SPECIFICALLY SELECTABLE

DR> PRINT <CR>

 TYPE T,R,C,S OR HELP (S) T ? S <CR>

IDV/IAV-11 MODULE STATISTICS.

UNIT	ERRORS	DROPPED
0	0	NO
1	6	YES
2	UNTESTED	NO

HERE, UNIT 0 HAS SHOWN NO FAULTS, UNIT 1 HAS HAD 6 ERRORS AND BEEN DROPPED FROM TESTING, AND UNIT 2 HAS NOT YET BEEN TESTED. UNIT 2 IS SHOWN AS NOT DROPPED. IF THE DIAGNOSTIC HAD NOT YET BEEN STARTED, THE UNIT WOULD STILL NOT BE SHOWN AS DROPPED (UNLIKE THE DISPLAY COMMAND).

DR> PRINT <CR>

 TYPE T,R,C,S OR HELP (S) S ? C <CR>

IDV/IAV-11 MODULE CONFIGURATION.

UNIT	ADDRESS	VECTOR ASSUMED	ID/MODE	MODULE TYPE	LOOPED	TO	AT	VECTOR
0	171000	NONE	060/000	DIG. OUT	YES	DIG. IN	171400	400
1	171400	400	030/002	DIG. IN	YES	DIG. OUT	171000	NONE
2	171010	NONE	260/000	AN. OUT	NO			
3	171410	410	100/000	AN. IN	NO			
4	171700	NONE	320/000	*** CANNOT BE TESTED WITH THIS DIAGNOSTIC ***				
5	171710	NONE	UNKNOWN	UNKNOWN	NO			
6	171020	NONE	060/000	DIG. OUT	YES	UNKNOWN	171420	420

THE THIRD COLUMN IS LABELED "VECTOR ASSUMED" FOR THE FOLLOWING REASONS. IF THE DIAGNOSTIC IS CONFIGURED AUTOMATICALLY BY ANSWERING "NO" TO THE "CHANGE HARDWARE" QUESTION, DEVICES WITH ADDRESSES LOWER THAN 171400 ARE ASSUMED TO HAVE NO VECTOR. ABOVE

THIS ADDRESS, THE VECTOR IS ASSUMED TO BE THE SAME AS THE LOW 9 BITS OF THE ADDRESS. IF THE HARDWARE QUESTIONS WERE ANSWERED, THE VECTOR IS PRINTED AS IT WAS TYPED, EXCEPT THAT ZERO VECTORS ARE PRINTED AS "NONE".

THE ABOVE EXAMPLE IS SELECTED TO ILLUSTRATE THE DIFFERENT FEATURES OF THE CONFIGURATION PRINTOUT.

UNIT 4 HAS AN ID CODE GREATER THAN 300. IT IS THEREFORE NOT ONE OF THE IXV11 DEVICES RECOGNISED BY THE DIAGNOSTIC. ALTHOUGH A WARNING IS PRINTED, BASIC LOGIC TESTS WILL BE RUN ON THE UNIT.

IF ADDRESSING THE UNIT UNDER TEST CAUSES A BUS TIMEOUT, THEN "UNKNOWN" IS PRINTED FOR THE ID/MODE AND MODULE TYPE. THIS IS SHOWN FOR UNIT 5. THIS WOULD NORMALLY ONLY OCCUR IF THE ADDRESS WAS INCORRECTLY TYPED IN THE HARDWARE QUESTIONS.

IF ADDRESSING THE OTHER MODULE CAUSES A BUS TIMEOUT, THEN "UNKNOWN" IS PRINTED FOR THE MODULE TYPE OF THE OTHER MODULE IN THE COLUMN HEADED "TO". THIS IS SHOWN FOR UNIT 6. THIS WOULD NORMALLY ONLY OCCUR IF THE ADDRESS WAS INCORRECTLY TYPED IN THE HARDWARE QUESTIONS.

DR> PRINT <CR>

TYPE T,R,C,S OR HELP (S) C ? R <CR>

TO REESTABLISH THE SYSTEM CONFIGURATION, ANSWER THE HARDWARE QUESTION TYPING "0" AS THE MODE ADDRESS FOR 16 UNITS.

EG. MODE REGISTER ADDRESS (0) 0 ? 0.....

TYPING "R" GIVES YOU INFORMATION ON HOW TO REESTABLISH THE SYSTEM CONFIGURATION. THIS IS NECESSARY IF THE H/W QUESTIONS HAVE BEEN ANSWERED OR THE HARDWARE ITSELF HAS BEEN CHANGED AND YOU WANT TO FIND OUT WHAT IS IN THE SYSTEM WITHOUT REBOOTING THE DIAGNOSTIC.

5.0 DEVICE INFORMATION TABLES

THE HARDWARE P TABLES CONTAIN 6 WORDS FOR EACH DEVICE. THESE ARE USED TO SAVE THE ANSWERS TO THE STARTUP HARDWARE QUESTIONS, AND CAN BE DISPLAYED ON THE CONSOLE BY USING EITHER THE "DISPLAY" COMMAND DESCRIBED IN SECTION 2.1 OR THE "PRINT" COMMAND DESCRIBED IN SECTION 4.

THE HARDWARE P TABLE IS SET UP FOR FIELD SERVICE FOR 16 UNITS, EACH WITH A MODE ADDRESS OF 0. IF THE USER TYPES "NO" TO THE "CHANGE HARDWARE" AND "CHANGE SOFTWARE" QUESTIONS, AN AUTOMATIC CONFIGURATION ROUTINE WILL BE RUN. THIS FINDS ALL DEVICES IN THE ADDRESS RANGE 171000 TO 171770, PRINTS A LIST OF ALL IXV11 DEVICES FOUND, AND CARRIES OUT THE SELECTED TESTS ON THESE DEVICES. THE HARDWARE TABLE SET UP BY THE CONFIGURATION ROUTINE REMAINS IN EFFECT UNTIL CHANGED BY THE HARDWARE QUESTIONS, EVEN IF THE HARDWARE CONFIGURATION ITSELF IS ALTERED.

USING THE XXDP+ SETUP UTILITY, THE TABLES CAN BE PRELOADED TO CONTAIN INFORMATION FOR SPECIFIC SYSTEMS. HOWEVER, SPECIAL CARE MUST BE TAKEN IF IT IS DESIRED TO RESTORE THE SELF CONFIGURING FEATURE OF THE DIAGNOSTIC. SEE SECTION 2.9.

6.0 TEST SUMMARIES

TESTS ARE DIVIDED INTO 4 TYPES - INTERNAL LOGIC, MANUFACTURING INPUT/OUTPUT, FIELD INPUT/OUTPUT USING FIELD TEST CONNECTORS, AND SPECIFICALLY SELECTABLE INPUT/OUTPUT.

INTERNAL LOGIC TESTS PERFORM DETAILED CHECKS ON EACH SEPARATELY ACCESSABLE PART OF THE DEVICE AND GIVE ERROR MESSAGES WHICH HELP THE FAULTY COMPONENTS TO BE IDENTIFIED. OUTPUT LINES ARE AFFECTED BUT NO TEST CONNECTORS ARE REQUIRED. THESE TESTS ARE ALWAYS RUN FOR FIELD AND MANUFACTURING.

TEST 1 :	REGISTER NXM TEST	
TEST 2 :	RESET TEST	
TEST 3 :	R-W BIT TEST	
TEST 4 :	INTERNAL INTERRUPT TEST	- DIGITAL INPUT
TEST 5 :	INTERNAL LOGIC TEST	- ANALOGUE INPUT
TEST 6 :	INTERNAL INTERRUPT TEST	- ANALOGUE INPUT
TEST 7 :	INTERNAL LOGIC TEST	- ANALOGUE OUTPUT

MANUFACTURING INPUT/OUTPUT TESTS ARE RUN IF THE USER ANSWERS "YES" TO THE "RUN MANUFACTURING TESTS" AND "RUN LOOPBACK AND I/O TESTS" SOFTWARE QUESTIONS. THE USER MUST EITHER CONNECT TEST CABLES BETWEEN THE MODULE BEING TESTED AND THE "OTHER" MODULE SPECIFIED IN THE HARDWARE QUESTIONS, OR FOR ANALOGUE MODULES, CONNECT VOLTAGE SOURCES OR MEASURING EQUIPMENT. THE ANALOGUE TESTS WILL ONLY BE CARRIED OUT IF THE "UNATTENDED MODE" FLAG IS NOT SELECTED.

TEST 8 :	DIGITAL INPUT LOOPBACK TEST	- PATTERN PAIR SELECTABLE
TEST 9 :	DIGITAL OUTPUT LOOPBACK TEST	- PATTERN PAIR SELECTABLE
TEST10 :	DIGITAL INPUT LOOPBACK TEST	- RANDOM PATTERN
TEST11 :	DIGITAL OUTPUT LOOPBACK TEST	- RANDOM PATTERN
TEST12 :	DIGITAL INPUT LOOPBACK TEST	- SLIDING PATTERN
TEST13 :	DIGITAL OUTPUT LOOPBACK TEST	- SLIDING PATTERN
TEST14 :	DIGITAL INPUT LOOPBACK TEST	- INTERRUPT LINE TEST
TEST15 :	ANALOGUE INPUT TEST	- FIELD AND MANUFACTURING
TEST16 :	ANALOGUE OUTPUT TEST	- FIELD AND MANUFACTURING

FIELD INPUT/OUTPUT TESTS ARE RUN IF THE USER ANSWERS "NO" TO THE "MANUFACTURING" SOFTWARE QUESTION AND THE "UNATTENDED MODE" FLAG IS NOT SELECTED. THE TESTS GUIDE THE USER THROUGH THE SEQUENCE NECESSARY TO TEST THE USER LINES OF EACH DEVICE USING THE TEST CONNECTORS.

TEST15 :	ANALOGUE INPUT TEST	- FIELD AND MANUFACTURING
TEST16 :	ANALOGUE OUTPUT TEST	- FIELD AND MANUFACTURING
TEST17 :	DIGITAL INPUT TEST	- FIELD TEST
TEST18 :	DIGITAL OUTPUT TEST	- FIELD TEST

SPECIFICALLY SELECTABLE TESTS ARE ONLY RUN IF THE USER SELECTS THEM BY TEST NUMBER (IE. NOT IN SEQUENCE WITH OTHER TESTS). THEY ALLOW THE USER TO READ FROM OR WRITE TO THE USER INTERFACE OF THE CHOSEN DEVICE. FOR ANALOGUE DEVICES, THE PROGRAM ALSO PERFORMS THE CONVERSIONS BETWEEN BIT PATTERNS AND VOLTAGES. ONCE

STARTED, THESE TESTS WILL RUN INDEFINATELY UNTIL "CONTROL C" IS TYPED. AT THIS POINT, THEY CAN BE RESTARTED BY TYPING "CONTINUE", OR ANOTHER TEST CAN BE SELECTED.

TEST19	: DIGITAL INPUT TEST	- SPECIFICALLY SELECTABLE
TEST20	: DIGITAL OUTPUT TEST	- SPECIFICALLY SELECTABLE
TEST21	: ANALOGUE INPUT (DYNAMIC)	- SPECIFICALLY SELECTABLE
TEST22	: ANALOGUE OUTPUT (DYNAMIC)	- SPECIFICALLY SELECTABLE
TEST23	: ANALOGUE OUTPUT (CALIBR.)	- SPECIFICALLY SELECTABLE
TEST24	: ANALOGUE INPUT (CALIBR.)	- SPECIFICALLY SELECTABLE
TEST25	: VISUAL LED TEST	- SPECIFICALLY SELECTABLE

6.1 INTERNAL LOGIC TESTS.

TEST 1 - REGISTER NXM TEST.

THIS TEST CHECKS THAT ACCESSING THE DEVICE MODE, DATA, CSA AND CSB REGISTERS (IF PRESENT) DOES NOT CAUSE A NXM TRAP. THE FOLLOWING ERROR MAY BE PRINTED :

ERROR 101 : REGISTER ADDRESSING ERROR - TRAP TO 4
REGISTER AT XXXXXX DOES NOT RESPOND

THIS COULD MEAN THAT THE DEVICE ADDRESS SWITCH IS INCORRECTLY SET, THAT THE ADDRESS WAS ENTERED INCORRECTLY IN THE STARTUP QUESTIONS, OR THAT THE DEVICE DOES NOT RESPOND.

TEST 2 - RESET TEST.

THIS TEST CHECKS THAT THE DEVICE REGISTERS ARE CORRECTLY SET OR RESET AFTER A BUS RESET. IN THE MOD REGISTER, ONLY THE LED BIT IS TESTED.

ERROR 200 : SPECIAL MODULE FOUND, CAN'T BE TESTED WITH THIS DIAGNOSTIC

ERROR 201 : LED BIT IN MOD REGISTER NOT CLEARED AFTER BUS RESET

ERROR 202 : LED BIT IN MOD REGISTER CAN'T BE SET

ERROR 203 : REGISTER INCORRECT AFTER BUS RESET
ADDRESS:AAAAAA, GOOD:GGGGGG, BAD:BBBBBB

TEST 3 - REGISTER R/W BIT TEST.

THIS TEST CHECKS THAT THE READ/WRITE BITS OF EACH REGISTER CAN ALL BE SET, ALL CLEARED AND INDIVIDUALLY SET. THE FOLLOWING ERRORS MAY BE PRINTED :

ERROR 300 : SPECIAL MODULE FOUND - CAN'T BE TESTED WITH THIS DIAGNOSTIC

ERROR 301 : REGISTER READ/WRITE BITS COULD NOT BE SET
ADDRESS:AAAAAA, GOOD:GGGGGG, BAD:BBBBBB, R/W BITS:RRRRRR

ERROR 302 : REGISTER READ/WRITE BITS COULD NOT BE CLEARED
ADDRESS:AAAAAA, GOOD:GGGGGG, BAD:BBBBBB, R/W BITS:RRRRRR

ERROR 303 : REGISTER READ/WRITE BITS COULD NOT BE INDIVIDUALLY SET
ADDRESS:AAAAAA, GOOD:GGGGGG, BAD:BBBBBB, R/W BITS:RRRRRR

ERROR 304 : MOD REGISTER CONTENTS OF DIGITAL INPUT MODULE INCORRECT
ADDRESS:AAAAAA, GOOD:GGGGGG, BAD:BBBBBB

ERROR 305 : MOD REGISTER CONTENTS OF DIGITAL INPUT MODULE INCORRECT
RT1 BIT NOT SET
ADDRESS:AAAAAA, GOOD:GGGGGG, BAD:BBBBBB

ERROR 306 : MOD REGISTER CONTENTS OF DIGITAL INPUT MODULE INCORRECT
RTO * RT1 NOT SET
ADDRESS:AAAAA, GOOD:GGGGG, BAD:BBBBB

TEST 4 - INTERNAL INTERRUPT LOGIC TEST - DIGITAL INPUT.

THIS TEST CHECKS THAT THE INTERNAL INTERRUPT LOGIC IS ABLE TO CAUSE AN INTERRUPT USING THE VECTOR AND PRIORITY LEVEL SELECTED IN THE START UP QUESTIONS. THE INTERRUPT IS GENERATED BY SETTING THE INTERRUPT ENABLE AND THEN INTERRUPT TEST BITS. AFTER THE INTERRUPT, THE CSA CONTENTS ARE ALSO CHECKED. THE FOLLOWING ERRORS MAY BE PRINTED :

- ERROR 401 : NO INTERRUPT AFTER SETTING EI * TST IR BIT IN CSA
CSA REGISTER CONTENTS :DDDDDD
- ERROR 402 : INTERRUPT DID NOT OCCUR AT THE SELECTED PRIORITY LEVEL
GOOD: GGGGGG, BAD :BBBBBB
- ERROR 403 : CSA REGISTER OF DIGITAL INPUT MODULE INCORRECT AFTER INTERRUPT
GOOD: GGGGGG, BAD :BBBBBB
- ERROR 404 : IR15 IN CSA REGISTER OF DIGITAL INPUT NOT CLEARED AFTER INTERRUPT
GOOD: GGGGGG, BAD :BBBBBB

TEST 5 - INTERNAL LOGIC TEST - ANALOGUE INPUT.

THIS TEST PERFORMS A PSEUDO CONVERSION ON EACH CHANNEL THAT IS FOUND, BY SETTING THE A/D START BIT AND THEN POLLING THE "DONE" BIT TO CHECK THAT THE CONVERSION HAS BEEN COMPLETED WITHIN THE ALLOWED TIMEOUT PERIOD (AT LEAST 10MS). BEFORE A/D START IS SET, A CHECK IS MADE THAT THE DONE BIT IS CLEARED. THE ERROR CONDITIONS ARE ALSO CHECKED, BUT NO CHECK IS MADE ON THE RESULTING INPUTS OR ON THE GAIN SETTING. THE FOLLOWING ERRORS MAY BE PRINTED :

- ERROR 501 : CSA CONTENTS INCORRECT AFTER READ DAT AND CLEAR CSA
GOOD:GGGGGG, BAD:BBBBBB, ADDRESSED CHANNEL IS : CCC
- ERROR 502 : DONE BIT IN CSA NOT SET (TIMEOUT) AFTER A/D START
GOOD:GGGGGG, BAD:BBBBBB, ADDRESSED CHANNEL IS : CCC
- ERROR 503 : ERR BIT IN CSA NOT SET AFTER A/D STAR WHEN DONE IS SET
GOOD:GGGGGG, BAD:BBBBBB, ADDRESSED CHANNEL IS : CCC
- ERROR 504 : ERR, DONE BIT IN CSA NOT CLEARED AFTER READ DAT REG.
GOOD:GGGGGG, BAD:BBBBBB, ADDRESSED CHANNEL IS : CCC
- ERROR 505 : ERR BIT IN CSA NOT SET AFTER LOADING A/D START TWICE
GOOD:GGGGGG, BAD:BBBBBB, ADDRESSED CHANNEL IS : CCC

TEST 6 - INTERNAL INTERRUPT LOGIC TEST - ANALOGUE INPUT.

THIS TEST CHECKS THAT THE INTERNAL INTERRUPT LOGIC IS ABLE TO CAUSE A DONE AND AN ERROR INTERRUPT USING THE VECTOR AND PRIORITY LEVEL SELECTED IN THE START UP QUESTIONS. THE DONE INTERRUPT IS GENERATED BY SETTING THE INTERRUPT ENABLE BIT AND THE A/D START BIT. THE ERROR INTERRUPT IS GENERATED BY SETTING THE A/D START BIT TWICE. A TIME OUT IS GENERATED IF THE INTERRUPT HAS NOT OCCURED WITHIN THE ALLOWED TIMEOUT PERIOD. THE FOLLOWING ERRORS MAY BE PRINTED :

ERROR 601 : TIME OUT - DONE BIT IN CSA NOT SET AFTER A/D START AND 606

ERROR 602 : NO INTERRUPT AFTER SETTING EI + A/D START IN CSA

ERROR 603 : PRIORITY LEVEL INCORRECT
GOOD:GGGGGG, BAD:BBBBBB

ERROR 604 : CSA CONTENTS INCORRECT AFTER CONVERSION
GOOD:GGGGGG, BAD:BBBBBB

ERROR 605 : ERROR INTERRUPT OCCURRED

ERROR 607 : NO ERROR INTERRUPT AFTER TWO A/D STARTS

TEST 7 - INTERNAL LOGIC TEST - ANALOGUE OUTPUT.

THIS TEST PERFORMS A CONVERSION ON EACH AVAILABLE CHANNEL. LOADING THE DAT REGISTER AND CHECKING THAT THE CONVERSIONS ARE COMPLETED WITHIN THE ALLOWED TIMEOUT PERIOD (READING THE READY BIT AFTER 150 US). THE FOLLOWING ERRORS MAY BE PRINTED :

ERROR 701 : CSA CONTENTS INCORRECT AFTER LOADING NEW CHANNEL
(IGNORE OPL BIT)
GOOD:GGGGGG, BAD:BBBBBB

ERROR 702 : CSA CONTENTS INCORRECT AFTER LOADING DAT REG.
READY TIMEOUT WAS 150 US
GOOD:GGGGGG, BAD:BBBBBB

6.2 MANUFACTURING I/O TESTS.

TEST 8 - DIGITAL INPUT LOOPBACK WITH SELECTABLE DATA PATTERN PAIR.

DATA TRANSFERS ARE MADE BETWEEN THE MODULE UNDER TEST (IN THIS TEST A DIGITAL INPUT) AND THE MODULE SPECIFIED AS THE "OTHER" IN THE HARDWARE QUESTIONS (IN THIS TEST A DIGITAL OUTPUT). IF LOW LEVEL IS NOT SELECTED, ONE HUNDRED CYCLES ARE PERFORMED USING THE DATA PAIR AND DEBOUNCE PERIOD SPECIFIED IN THE HARD AND SOFTWARE QUESTIONS. A DEBOUNCE PERIOD OF 10 MILLISECONDS IS THEN USED TO CHECK THAT THE DAT REGISTER DOES NOT CHANGE IN THE FIRST 2.5 MILLISECONDS.

IF LOW LEVEL IS SELECTED, ONLY 3 CYCLES ARE PERFORMED AND THE DEBOUNCE FUNCTION IS NOT CHECKED.

BY DEFAULT, THE DATA PAIR IS ALL ZEROS AND ALL ONES, BUT ANY PATTERN PAIR CAN BE SELECTED. IF THE TWO PATTERNS ARE IDENTICAL, A FIXED OUTPUT WILL RESULT.

ERROR 801 : REGISTER ADDRESSING ERROR FOR OTHER MODULE
804 AND 806 REGISTER AT AAAAAA DOSE NOT RESPOND

ERROR 802 : LOOPED DATA IN DATA REGISTER OF DIGITAL INPUT MODULE INCORRECT
AND 807 GOOD:GGGGGG, BAD:BBBBBB, MOD REGISTER CONTENTS:DDDDDD

ERROR 803 : DEBOUNCE FUNCTION ERROR -
AND 805 DATA ACCEPTED BEFORE LOADED DEBOUNCE FINISHED

TEST 9 - DIGITAL OUTPUT LOOPBACK TEST - SELECTABLE DATA PATTERN PAIR.

DATA TRANSFERS ARE MADE BETWEEN THE MODULE UNDER TEST (IN THIS TEST A DIGITAL OUTPUT) AND THE MODULE SPECIFIED AS THE "OTHER" IN THE HARDWARE QUESTIONS (IN THIS TEST A DIGITAL INPUT).

SEVERAL CYCLES ARE PERFORMED USING THE DATA PAIR FROM THE SW P-TABLE AND A DEBOUNCE PERIOD FOUND IN THE MOD REGISTER OF THE DIGITAL OUTPUT MODULE. FOR THE INPUT MODULE A DEBOUNCE OF (500US) IS USED. BY DEFAULT, THE DATA PAIR IS ALL ZEROS AND ALL ONES, BUT ANY PATTERN PAIR CAN BE SELECTED. IF THE TWO PATTERNS ARE IDENTICAL, A FIXED OUTPUT WILL RESULT.

IF A RELAY OUTPUT MODULE IS USED, A CHECK IS ALSO MADE THAT THE DAT REGISTER OF THE DIGITAL INPUT MODULE WILL NOT BE CHANGED WITHIN 2 MS OF LOADING THE OUTPUT DATA REGISTER.

THE FOLLOWING ERRORS MAY BE PRINTED :

ERROR 901 : REGISTER ADDRESSING ERROR FOR OTHER MODULE ADDRESS
903, 905 REGISTER AT AAAAAA DOSE NOT RESPOND
AND 907

ERROR 902 : DATA REGISTER OF DIGITAL OUTPUT MODULE INCORRECT
AFTER READ BACK
GOOD:GGGGGG, BAD:BBBBBB, MOD REGISTER CONTENTS:DDDDDD

ERROR 904 : RECEIVED DATA INCORRECT - CHECK OUTPUT LOGIC
GOOD:GGGGGG, BAD:BBBBBB, MOD REGISTER CONTENTS:DDDDDD

ERROR 906 : DEBOUNCE TIME INCORRECT
DATA ACCEPTED TOO FAST (LOADED DEBOUNCE NOT FINISHED)
GOOD:GGGGGG, BAD:BBBBBB, MOD REGISTER CONTENTS:DDDDDD

ERROR 908 : RECEIVED DATA INCORRECT - CHECK OUTPUT LOGIC
GOOD:GGGGGG, BAD:BBBBBB, MOD REGISTER CONTENTS:DDDDDD

TEST 10 - DIGITAL INPUT LOOPBACK TEST - RANDOM PATTERN

THIS TEST CHECKS THE DATA TRANSFER BETWEEN INPUT AND OUTPUT MODULES. THE OUTPUT MODULE IS THAT SPECIFIED AS THE "OTHER" IN THE HARDWARE QUESTIONS. SEVERAL CYCLES ARE PERFORMED USING DATA GENERATED BY A RANDOM PATTERN GENERATOR WITHIN THE PROGRAM.

IF LOW LEVEL TESTING WAS NOT SELECTED IN THE HARDWARE QUESTIONS, THE SELECTED DEBOUNCE PERIOD IS USED. IF "ALL" WAS SPECIFIED, THE PERIOD IS SELECTED AT RANDOM FROM THE 3 ALLOWED VALUES. THE FOLLOWING ERRORS MAY BE PRINTED :

ERROR 1001 : REGISTER ADDRESSING ERROR FOR OTHER MODULE
AND 1003 REGISTER AT AAAAAA DOSE NOT RESPOND

ERROR 1002 : DIGITAL INPUT DATA REGISTER INCORRECT
AND 1004 LOADED AND READ DATA NOT THE SAME
GOOD:GGGGGG, BAD:BBBBBB, MOD REGISTER CONTENTS:DDDDDD

TEST 11 - DIGITAL OUTPUT LOOPBACK TEST WITH RANDOM PATTERN

THIS TEST CHECKS THE DATA OUTPUT LINES OF DIGITAL OUTPUT MODULES BY READING THE OUTPUT DATA WITH AN INPUT MODULE. THE INPUT MODULE IS THAT SPECIFIED AS THE "OTHER" IN THE HARDWARE QUESTIONS. 500 CYCLES ARE PERFORMED USING DATA GENERATED BY A RANDOM PATTERN GENERATOR WITHIN THE PROGRAM. THE TIMEOUT PERIOD IS BASED ON THE DEBOUNCE PERIOD FOUND IN THE MOD REGISTER OF THE DIGITAL OUTPUT MODULE.

THE DIGITAL INPUT MODULE IS ALWAYS LOADED WITH A DEBOUNCE PERIOD OF 500 US. THE FOLLOWING ERRORS MAY BE PRINTED :

ERROR 1101 : REGISTER ADDRESSING ERROR FOR OTHER MODULE
AND 1103 REGISTER AT AAAAAA DOES NOT RESPOND

ERROR 1102 : DATA REGISTER OF DIGITAL OUTPUT MODULE INCORRECT
DATA CHANGED AFTER LOADING
GOOD:GGGGGG, BAD:BBBBBB, MOD REGISTER CONTENTS:DDDDDD

ERROR 1104 : LOADED AND READ DATA NOT THE SAME - CHECK OUTPUT LOGIC
GOOD:GGGGGG, BAD:BBBBBB, MOD REGISTER CONTENTS:DDDDDD

TEST 12 - DIGITAL INPUT LOOPBACK TEST WITH SLIDING PATTERN

THIS TEST TRANSFERS SLIDING ONES AND THEN SLIDING ZEROS BETWEEN OUTPUT MODULE AND INPUT MODULES. THE OUTPUT MODULE IS THAT SPECIFIED AS THE "OTHER" IN THE HARDWARE QUESTIONS.

THE TRANSFERS ARE PERFORMED USING THE LEVEL SELECT AND DEBOUNCE PERIOD SPECIFIED IN THE HARDWARE QUESTIONS.

THE FOLLOWING ERRORS MAY BE PRINTED :

ERROR 1201 : REGISTER ADDRESSING ERROR FOR OTHER MODULE
1203, 1205 REGISTER AT AAAAAA DOES NOT RESPOND
AND 1207

ERROR 1202 : DATA REGISTER OF INPUT MODULE INCORRECT
1204 AND 1208 GOOD:GGGGGG, BAD:BBBBBB, MOD REGISTER CONTENTS:DDDDDD

TEST 13 - DIGITAL OUTPUT LOOPBACK TEST WITH SLIDING PATTERN

THIS TEST CHECKS THE DATA OUTPUT LINES OF DIGITAL OUTPUT MODULES BY READING THE OUTPUT DATA WITH AN INPUT MODULE. THE INPUT MODULE IS THAT SPECIFIED AS THE "OTHER" IN THE HARDWARE QUESTIONS. SEVERAL CYCLES ARE PERFORMED USING SLIDING ONES AND THEN SLIDING ZEROS.

THE TIMEOUT PERIOD IS DEPENDENT ON THE DEBOUNCE SPECIFIED IN THE RT BITS FOUND IN THE MOD REGISTER OF THE DIGITAL OUTPUT MODULE.

THE DIGITAL INPUT MODULE IS ALWAYS LOADED WITH A DEBOUNCE PERIOD OF 5 US.

THE FOLLOWING ERRORS MAY BE PRINTED :

ERROR 1301 : REGISTER ADDRESSING ERROR FOR OTHER MODULE ADDRESS
1303 AND 1306 REGISTER AT AAAAAA DOSE NOT RESPOND

ERROR 1302 : DATA REGISTER OF OUTPUT MODULE INCORRECT AFTER LOAD
AND 1305 GOOD:GGGGGG, BAD:BBBBBB, MOD REGISTER CONTENTS:DDDDDD

ERROR 1304 : RECEIVED DATA INCORRECT - CHECK DIGITAL OUTPUT LINES
AND 1307 GOOD:GGGGGG, BAD:BBBBBB, MOD REGISTER CONTENTS:DDDDDD

TEST 14 - DIGITAL LOOPBACK TEST - INPUT MODULE INTERRUPT LINE.

THIS CHECKS THAT THE CSA INTERRUPT BIT OF THE INPUT MODULE CAN BE SET TO CAUSE AN INTERRUPT BY THE LEADING OR TRAILING EDGES OF THE MODULE DATA LINE 15. THE DEBOUNCE PERIOD FOR THE DIGITAL INPUT MODULE IS DEPENDENT ON THE OUTPUT MODULE THAT YOU SELECT AS THE "OTHER" IN THE HARDWARE QUESTIONS. A CHECK IS ALSO MADE THAT BIT 15 REMAINS HIGH AFTER THE LINE HAS REVERTED BACK TO ITS INITIAL STATE. ALL SIGNALS ARE PROVIDED BY LINES FROM THE OUTPUT MODULE.

ALSO THE BITS 0-14 ARE SWITCHED ON AND OFF FOUR TIMES AND THEN A CHECK IS MADE THAT BIT 15 IS NOT AFFECTED.

ALL OF THE FUNCTIONAL TESTS ABOVE ARE ALSO CARRIED OUT IF LOW LEVEL TESTING IS SELECTED.

THE FOLLOWING ERRORS MAY BE PRINTED :

- ERROR 1401 : REGISTER ADDRESSING ERROR FOR OTHER MODULE
AND 1410 REGISTER AT AAAAAA DOSE NOT RESPOND
- ERROR 1402 : DATA REGISTER CONTENTS OF INPUT MODULE INCORRECT
AND 1411 GOOD:GGGGGG, BAD:BBBBBB, MOD REGISTER CONTENTS:DDDDDD
- ERROR 1403 : NO INTERRUPT ON LEADING EDGE OF LINE 15
AND 1412 CSA REGISTER CONTENTS IS :DDDDDD
- ERROR 1404 : CSA REGISTER INCORRECT AFTER INTERRUPT
1407, 1413 GOOD:GGGGGG, BAD:BBBBBB
AND 1416
- ERROR 1405 : CSA REGISTER NOT LOADABLE WITH THE CORRECT DATA
AND 1414 GOOD:GGGGGG, BAD:BBBBBB
- ERROR 1406 : NO INTERRUPT ON TRAILING EDGE OF LINE 15
CSA REGISTER CONTENT IS : DDDDDD
- ERROR 1408 : INTERRUPT OCCURRED WHEN SWITCHING DATA BITS 0-14
AND 1417 CSA REGISTER CONTENT IS : DDDDDD
- ERROR 1409 : CSA REGISTER CHANGED AFTER SWITCHING DATA BITS 0-14
AND 1418 GOOD:GGGGGG, BAD:BBBBBB

TEST 15 - ANALOGUE INPUT TEST. (FIELD AND MANUFACTURING TEST)

AT FIRST THE USER IS TOLD THAT THE TEST WILL ONLY RUN CORRECTLY IF THE MODULE HAS BEEN CALIBRATED. THEN HE IS ASKED HOW MANY MUX BOARDS ARE CONNECTED TO THE CONTROLLER. THEN HE IS ASKED TO CONNECT A CALIBRATED VOLTAGE SOURCE ON THE EVEN INPUTS TO THE MODULE VIA THE ANALOGUE TEST CONNECTOR.

PROMPTED BY THE PROGRAM, THE USER THEN SETS UP A SERIES OF VOLTAGES, TYPING "CARRIAGE RETURN" AFTER EACH SO THAT THE PROGRAM CAN CHECK THE D/A CONVERSIONS. THE CONVERSIONS FOR 48.828 MV ARE PERFORMED ON ALL GAINS. OTHER CONVERSIONS ARE PERFORMED ONLY AT GAIN 1. IF "N" TO THE SW QUESTION "RUN MANUFACTURING TEST" IS ANSWERED THE FOLLOWING TOLLERANCES WILL BE USED, FOR GAIN 1,2,5 IS 2 LSB, FOR GAIN 10 IS 4 LSB, FOR GAIN 20 IS 6 LSB, FOR GAIN 50 IS 7 LSB, FOR GAIN 100 IS 12 LSB AND FOR GAIN 200 IS 20 LSB.
IF "Y" WAS ANSWERD THE TOLLERANCE WILL BE, FOR GAIN 1,2,5 IS 2 LSB, FOR GAIN 10 IS 3 LSB, FOR GAIN 20 IS 3, FOR GAIN 50 IS 5 LSB, FOR GAIN 100 IS 10 LSB AND FOR GAIN 200 IS 15 LSB.
IF MUX BOARDS ARE CONNECTED THE TOLLERANCE WILL BE 2 LSB MORE FOR GAINS OVER 10.

NEXT, THE USER IS PROMPTED TO CONNECT THE VOLTAGE SOURCE TO THE ODD INPUTS AND DO THE SAME AS FOR THE EVEN LINES AS ABOVE.

THE VALUES USED DEPEND ON WHETHER THE MODULE IS SET UP FOR UNIPOLAR OR BIPOLAR INPUT.

FINALLY, THE EXTERNAL TRIGGER FUNCTION IS TESTED BY SWITCHING THE ENABLE EXTERNAL TRIGGER ON AND OFF TO DO THE EXTERNAL START (EET IS LOOPED BACK TO THE EXTERNAL TRIGGER INPUT VIA THE ANALOGUE TEST CONNECTOR). THE EXTERNAL TRIGGER FUNCTION IS TESTED UNDER INTERRUPT.

IF THE "UAM" FLAG IS SET, THE TEST IS NOT CARRIED OUT.

THE FOLLOWING ERRORS MAY BE PRINTED :

ERROR 1501,1504: CSA CONTENTS INCORRECT
GOOD:GGGGGG, BAD:BBBBBB, ADDRESSED CHANNEL IS: CCC

ERROR 1502 : TIMEOUT - DONE BIT NOT SET AFTER A/D START
AND 1505 CSA REGISTER CONTENTS IS : DDDDDD

ERROR 1503,1506 : CONVERSION VALUE INCORRECT
GOOD:GGGGGG, BAD:BBBBBB, CSA REGISTER CONTENTS:DDDD

ERROR 1507 : NO DONE INTERRUPT AFTER EXTERNAL TRIGGER

ERROR 1508 : CSA CONTENTS INCORRECT AFTER EXTERNAL TRIGGER
GOOD:GGGGGG, BAD:BBBBBB, ADDRESSED CHANNEL IS: CCC

TEST 16 - ANALOGUE OUTPUT TEST. (FIELD AND MANUFACTURING TEST)

THE USER IS TOLD THAT THE TEST WILL ONLY RUN CORRECTLY IF THE MODULE HAS BEEN CALIBRATED.

NEXT, THE OPERATOR IS ASKED WHETHER THE MODULE IS VOLTAGE CALIBRATED OR CURRENT CALIBRATED. DEPENDING ON THE RESPONSE, THE PROGRAM PROMPTS WITH VOLTAGE OR CURRENT VALUES. IF CURRENT IS SELECTED A CHECK IS MADE THAT ALL CHANNELS ARE SHOWN AS "OPEN LINE" IN THE CSA REGISTER. THIS ASSUMES THAT NOTHING IS CONNECTED TO THE CURRENT OUTPUTS.

THE USER IS THEN ASKED TO CONNECT A DVM WITH THE CORRECT RANGE TO THE MODULE OUTPUT AT THE SPECIFIED CHANNEL. IF CURRENT WAS SELECTED, THE PROGRAM USES THE OFFSET BIT IN THE MOD REGISTER AND CHECK THAT THE OPL BIT IN THE CSA REGISTER IS CLEAR.

THE OPERATOR THEN HAS TO CHECK SEVERAL VALUES TO ENSURE THAT THE READING OF THE DVM IS THE SAME AS THE OUTPUT VALUE. ALL VALUES HAVE BEEN CHECKED BY ANSWERING THE QUESTION "IS VALUE ON DVM IN RANGE. THE PROCESS IS REPEATED FOR ALL CHANNELS.

IF "UAM" FLAG IS SET, THIS TEST WILL NOT BE CARRIED OUT.

N.B. THE DVM READINGS MAY DIFFER FROM THE PRINTED VALUES IF THE MODULE HAS BEEN RECALIBRATED BY THE CUSTOMER.

THE FOLLOWING ERRORS MAY BE PRINTED :

ERROR 1601 : OPL BIT IN CSA NOT SET
MAKE SURE THAT NOTHING IS CONNECTED TO CURRENT OUTPUT
CSA REGISTER CONTENTS IS :DDDDDD

ERROR 1602 : OPL BIT IN CSA NOT CLEARED
IS DVM CONNECTED TO THE ADDRESSED CHANNEL ?
CSA REGISTER CONTENTS IS :DDDDDD

ERROR 1603 : READY BIT IN CSA NOT SET AFTER LOADING DAT REGISTER (TIMEOUT)
AND 1606 GOOD:GGGGGG, BAD:BBBBBB

ERROR 1604 : CSA CONTENTS INCORRECT
AND 1607 GOOD:GGGGGG, BAD:BBBBBB

ERROR 1605 : MODULE DEFECT OR INCORRECTLY CALIBRATED - TRY TEST 23
AND 1608

6.3 FIELD INPUT/OUTPUT TESTS USING FIELD TEST CONNECTORS.

TEST 15 - ANALOGUE INPUT TEST. (FIELD AND MANUFACTURING TEST)

AT FIRST THE USER IS TOLD THAT THE TEST WILL ONLY RUN CORRECTLY IF THE MODULE HAS BEEN CALIBRATED. THEN HE IS ASKED HOW MANY MUX BOARDS ARE CONNECTED TO THE CONTROLLER. THEN HE IS ASKED TO CONNECT A CALIBRATED VOLTAGE SOURCE ON THE EVEN INPUTS TO THE MODULE VIA THE ANALOGUE TEST CONNECTOR.

PROMPTED BY THE PROGRAM, THE USER THEN SETS UP A SERIES OF VOLTAGES, TYPING "CARRIAGE RETURN" AFTER EACH SO THAT THE PROGRAM CAN CHECK THE D/A CONVERSIONS. THE CONVERSIONS FOR 48.828 MV ARE PERFORMED ON ALL GAINS. OTHER CONVERSIONS ARE PERFORMED ONLY AT GAIN 1. IF "N" TO THE SW QUESTION "RUN MANUFACTURING TEST" IS ANSWERED THE FOLLERING TOLLERANCES WILL BE USED, FOR GAIN 1,2,5 IS 2 LSB, FOR GAIN 10 IS 4 LSB, FOR GAIN 20 IS 6 LSB, FOR GAIN 50 IS 7 LSB, FOR GAIN 100 IS 12 LSB AND FOR GAIN 200 IS 20 LSB.

IF "Y" WAS ANSWERD THE TOLLERANCE WILL BE, FOR GAIN 1,2,5 IS 2 LSB, FOR GAIN 10 IS 3 LSB, FOR GAIN 20 IS 3, FOR GAIN 50 IS 5 LSB, FOR GAIN 100 IS 10 LSB AND FOR GAIN 200 IS 15 LSB.
IF MUX BOARDS ARE CONNECTED THE TOLLERANCE WILL BE 2 LSB MORE FOR GAINS OVER 10.

NEXT, THE USER IS PROMPTED TO CONNECT THE VOLTAGE SOURCE TO THE ODD INPUTS AND DO THE SAME AS FOR THE EVEN LINES AS ABOVE.

THE VALUES USED DEPEND ON WHETHER THE MODULE IS SET UP FOR UNIPOLAR OR BIPOLAR INPUT.

FINALLY, THE EXTERNAL TRIGGER FUNCTION IS TESTED BY SWITCHING THE ENABLE EXTERNAL TRIGGER ON AND OFF TO DO THE EXTERNAL START (EET IS LOOPED BACK TO THE EXTERNAL TRIGGER INPUT VIA THE ANALOGUE TEST CONNECTOR). THE EXTERNAL TRIGGER FUNCTION IS TESTED UNDER INTERRUPT.

IF THE "UAM" FLAG IS SET, THE TEST IS NOT CARRIED OUT.

THE FOLLOWING ERRORS MAY BE PRINTED :

ERROR 1501,1504: CSA CONTENTS INCORRECT
GOOD:GGGGGG, BAD:BBBBBB, ADDRESSED CHANNEL IS: CCC

ERROR 1502 : TIMEOUT - DONE BIT NOT SET AFTER A/D START
AND 1505 : CSA REGISTER CONTENTS IS : DDDDDD

ERROR 1503,1506 : CONVERSION VALUE INCORRECT
GOOD:GGGGGG, BAD:BBBBBB, CSA REGISTER CONTENTS:DDDDD

ERROR 1507 : NO DONE INTERRUPT AFTER EXTERNAL TRIGGER

ERROR 1508 : CSA CONTENTS INCORRECT AFTER EXTERNAL TRIGGER
GOOD:GGGGGG, BAD:BBBBBB, ADDRESSED CHANNEL IS: CCC

TEST 16 - ANALOGUE OUTPUT TEST. (FIELD AND MANUFACTURING TEST)

THE USER IS TOLD THAT THE TEST WILL ONLY RUN CORRECTLY IF THE MODULE HAS BEEN CALIBRATED.

NEXT, THE OPERATOR IS ASKED WHETHER THE MODULE IS VOLTAGE CALIBRATED OR CURRENT CALIBRATED. DEPENDING ON THE RESPONSE, THE PROGRAM PROMPTS WITH VOLTAGE OR CURRENT VALUES. IF CURRENT IS SELECTED A CHECK IS MADE THAT ALL CHANNELS ARE SHOWN AS "OPEN LINE" IN THE CSA REGISTER. THIS ASSUMES THAT NOTHING IS CONNECTED TO THE CURRENT OUTPUTS. THE USER IS THEN ASKED TO CONNECT A DVM WITH THE CORRECT RANGE TO THE MODULE OUTPUT AT THE SPECIFIED CHANNEL. IF CURRENT WAS SELECTED, THE PROGRAM USES THE OFFSET BIT IN THE MOD REGISTER AND CHECKS THAT THE OPL BIT IN THE CSA REGISTER IS CLEAR.

THE OPERATOR THEN HAS TO CHECK SEVERAL VALUES TO ENSURE THAT THE READING OF THE DVM IS THE SAME AS THE OUTPUT VALUE. WHEN ALL VALUES HAVE BEEN CHECKED AND NO ERROR FOUND, THE USER HAS TO TYPE "CARRIAGE RETURN". THE PROCESS IS REPEATED FOR ALL CHANNELS.

IF "UAM" FLAG IS SET, THIS TEST WILL NOT BE CARRIED OUT.

N.B. THE DVM READINGS MAY DIFFER FROM THE PRINTED VALUES IF THE MODULE HAS BEEN RECALIBRATED BY THE CUSTOMER.

THE FOLLOWING ERRORS MAY BE PRINTED :

- ERROR 1601 : OPL BIT IN CSA NOT SET
MAKE SURE THAT NOTHING IS CONNECTED TO CURRENT OUTPUT
CSA REGISTER CONTENTS IS :DDDDDD
- ERROR 1602 : OPL BIT IN CSA NOT CLEARED
IS DVM CONNECTED TO THE ADDRESSED CHANNEL ?
CSA REGISTER CONTENTS IS :DDDDDD
- ERROR 1603 : READY BIT IN CSA NOT SET AFTER LOADING DAT REGISTER (TIMEOUT)
AND 1606 GOOD:GGGGGG, BAD:BBBBBB
- ERROR 1604 : CSA CONTENTS INCORRECT
AND 1607 GOOD:GGGGGG, BAD:BBBBBB
- ERROR 1605 : MODULE DEFECT OR INCORRECTLY CALIBRATED - TRY TEST 23
AND 1608

TEST 17 - DIGITAL INPUT TEST.

FIRST OF ALL, THE USER IS PROMPTED TO INSERT THE TEST CONNECTOR INTO THE MODULE ON WHICH THE LED IS FLASHING. THEN A CHECK IS MADE THAT THE INPUT DATA MATCHES THE ODD PATTERNS AVAILABLE FROM THE CONNECTOR. AFTER THIS, THE USER IS ASKED TO SWITCH THE CONNECTOR TO THE EVEN POSITION AND THE DATA IS CHECKED TO BE CORRECT. THE PROGRAM THEN PROMPTS THE USER TO MOVE THE SWITCH "ON" AND "OFF" TWICE MORE. THIS CHECKS THAT THE CSA INTERRUPT

BIT IR15 CAN BE SET TO CAUSE AN INTERRUPT BY THE LEADING OR TRAILING EDGES OF THE MODULE DATA LINE 15, AND THAT IR15 OF THE CSA REMAINS SET AFTER THE DATA LINE 15 HAS REVERTED TO ITS ORIGINAL STATE.

IF THE "UAM" FLAG IS SET, THIS TEST WILL NOT BE CARRIED OUT.

THIS TEST IS CARRIED OUT IN LOW LEVEL MODE.

THE FOLLOWING ERRORS MAY BE PRINTED :

ERROR 1701 : DATA REGISTER OF DIGITAL INPUT NOT LOADED WITH EVEN DATA
GOOD:GGGGGG, BAD:BBBBBB, MOD REGISTER CONTENTS IS:DDDDDD

ERROR 1702 : DATA REGISTER OF DIGITAL INPUT NOT LOADED WITH ODD DATA
GOOD:GGGGGG, BAD:BBBBBB, MOD REGISTER CONTENTS IS:DDDDDD

ERROR 1703 : NO INTERRUPT ON LEADING EDGE OF DATA LINE 15
CSA REGISTER CONTENTS IS:DDDDDD

ERROR 1704 : CSA REGISTER OF DIGITAL INPUT MODULE INCORRECT
AND 1706 AFTER INTERRUPT
GOOD:GGGGGG, BAD:BBBBBB, MOD REGISTER CONTENTS IS:DDDDDD

ERROR 1705 : NO INTERRUPT ON TRAILING EDGE OF DATA LINE 15
CSA REGISTER CONTENTS IS:DDDDDD

ERROR 1707 : CSA DATA BIT IR15 IS NOT CORRECT - CLEARED BY DATA LINE 15
GOOD:GGGGGG, BAD:BBBBBB, MOD REGISTER CONTENTS IS:DDDDDD

TEST 18 - DIGITAL OUTPUT TEST.

THE USER IS PROMPTED TO INSERT THE DIGITAL TEST CONNECTOR INTO THE MODULE ON WHICH THE LED IS FLASHING. THEN HE IS PROMPTED TO CHECK THAT A SLIDING ONES PATTERN APPEARS ON THE TEST CONNECTOR LEDS. AFTER 15 SECONDS THE PROGRAM PROMPTS THE USER TO LOOK FOR A SLIDING ZEROS PATTERN, WHICH IS OUTPUT FOR 15 SECONDS BEFORE TERMINATING THE TEST. THIS CHECKS THAT ALL OUTPUT LINES CAN BE UNIQUELY SET AND CLEARED.

IF THE "UAM" FLAG IS SET, THIS TEST WILL NOT BE CARRIED OUT.

ERRORS : THIS TEST IS A VISUAL TEST AND HAS NO ERROR PRINTOUT.

6.4 SPECIFICALLY SELECTABLE INPUT/OUTPUT TESTS.

TEST 19 - DIGITAL INPUT TEST - SPECIFICALLY SELECTABLE

THIS TEST CONTINUOUSLY READS THE INPUT LINES AND PRINTS THE DATA TO THE CONSOLE.

THE USER IS ASKED WHICH INPUT RANGE (LOW LEVEL OR NORMAL) AND IF NORMAL IS SELECTED, WHICH DEBOUNCE PERIOD TO USE. THEN THE INPUT LINES OF THE MODULE ARE CONTINUOUSLY READ AND, IF THE SUPERVISOR FLAG "INHIBIT ERROR REPORT" IS NOT SET, THE BIT PATTERN IS OUTPUT TO THE CONSOLE. BY USING DIFFERENT INPUT VOLTAGES, THIS TEST CAN BE USED TO CHECK THE SWITCHING LEVELS. THIS TEST IS A VISUAL TEST. AN ERROR REPORT IS ONLY MADE IF THE LLS AND RT BITS IN THE MOD REGISTER CAN'T CLEAR OR SET. THE INTERRUPT FUNCTION OF THE MODULE IS NOT TESTED. TO ABORT THE TEST, THE USER MUST TYPE "CNTL C".

THE FOLLOWING ERRORS MAY BE PRINTED :

ERROR 1901 : LOW LEVEL BIT IN MOD REGISTER OF DIGITAL INPUT COULD NOT BE CLEARED

ERROR 1902 : DEBOUNCE BITS IN MOD REGISTER COULD NOT BE SET
GOOD:GGGGGG, BAD:BBBBBB

ERROR 1903 : LOW LEVEL BIT IN MOD REGISTER OF DIGITAL INPUT COULD NOT BE SET

TEST 20 - DIGITAL OUTPUT TEST - SPECIFICALLY SELECTABLE.

THE USER IS FIRST ASKED TO SELECT A PATTERN TYPE - SLIDING ONES, SLIDING ZEROS, OR A PATTERN PAIR. IN THE LAST CASE, HE IS ALSO ASKED TO ENTER THE DATA PAIR. THE PROGRAM THEN REQUESTS A TIME INTERVAL IN MILLISECONDS (DEFAULT IS ZERO) TO WAIT BETWEEN EACH PATTERN CHANGE. THE SELECTED PATTERN IS THEN OUTPUT TO THE USER LINES OF THE MODULE.

AFTER LOADING THE OUTPUT, THE DATA REGISTER IS READ BACK AND AN ERROR REPORTED IF THE LOADED DATA AND THE READ DATA ARE NOT THE SAME. THIS IS THE ONLY ERROR REPORT IN THE TEST. ALL OTHER CHECKING IS VISUAL.

TO ABORT THE TEST, "CNTL C" MUST BE TYPED.

THE FOLLOWING ERRORS MAY BE PRINTED :

ERROR 2001 : DATA REGISTER INCORRECT AFTER READ BACK
AND 2002 GOOD:GGGGG, BAD:BBBBBB

TEST 21 - ANALOGUE INPUT (DYNAMIC ROUTINE) - SPECIFICALLY
SELECTABLE.

THE USER IS ASKED WHICH CHANNELS ARE TO BE READ, WHICH GAIN AND WHICH TRIGGER (INTERNAL OR EXTERNAL) ARE TO BE USED. AFTER THIS HE IS TOLD WHICH RANGE (UNI OR BIPOLAR) THE MODULE IS SWITCHED

TO.

CONVERSIONS ARE THEN MADE TWICE ON EACH SELECTED CHANNEL. THE VALUES READ ARE PRINTED OUT AS A DUMP AFTER ALL CONVERSIONS. THE DUMPED VALUES ARE OCTAL.

PRINTOUT CAN BE DISABLED USING THE SUPERVISOR FLAG "INHIBIT ERROR REPORTS".

TO ABORT THE TEST, "CNTL C" MUST BE TYPED.

THE FOLLOWING ERRORS MAY BE PRINTED :

ERROR 2101 : CSA CONTENTS INCORRECT - SELECTED CHANNEL NOT AVAILABLE
GOOD:GGGGGG, BAD:BBBBBB, ADDRESSED CHANNEL IS:CCC

ERROR 2102 : MODULE NOT READY FOR CONVERSION - DONE BIT IN CSA IS SET

ERROR 2103 : DONE BIT IN CSA NOT SET (TIMEOUT) AFTER CONVERSION START
AND 2105

ERROR 2104 : DONE BIT IN CSA NOT CLEARED AFTER READING DATA REGISTER
AND 2106

ERROR 2107 : DUMP BUFFER FULL - MORE THEN 127 CHANNELS ADDRESSED ?
AND 2115

ERROR 2108 : CSA CONTENTS INCORRECT
2110, 2112 GCOD:GGGGGG, BAD:BBBBBB, ADDRESSED CHANNEL IS:CCC
AND 2114

ERROR 2109 : TIMEOUT - NO DONE INTERRUPT OCCURRED
AND 2111

ERROR 2113 : ERRONEOUS INTERRUPT OCCURRED

TEST 22 - ANALOGUE OUTPUT (DYNAMIC TEST) - SPECIFICALLY
SELECTABLE.

THIS TEST GENERATES EITHER A PERMANENT LOADED STATIC VALUE, OR
AN ENDLESS SYMMETRICAL VOLTAGE RAMP.

THE USER IS ASKED TO ENTER THE NUMBER OF THE FIRST AND LAST CHANNEL WHICH HE WANTS TO USE. HE THEN SELECTS THE OUTPUT PATTERN TYPE - SINGLE FIXED VALUE OR RAMP. FOR THE FIRST OF THESE, THE USER TYPES THE DESIRED OUTPUT BINARY VALUE. FOR THE RAMP PATTERN, AN ENDLESS SYMMETRICAL VOLTAGE RAMP IS GENERATED. IN THIS CASE, THE USER IS ASKED TO ENTER A VALUE "DELTA U". THIS VALUE IS THE NUMBER OF BITS BY WHICH THE PROGRAM CHANGES THE OUTPUT FOR EACH STEP OF THE RAMP, AND MUST BE IN THE RANGE 1 TO 4096. THE USER MUST ALSO SELECT A TIME INTERVAL "DELTA T" IN INCREMENT STEPS, WHICH WILL DETERMINE THE PERIOD BETWEEN EACH VALUE CHANGE. IF ZERO IS TYPED, THE VALUE WILL BE VARIED AS FAST AS THE HARDWARE ALLOWS IT.

THE CONTROL STATUS REGISTER STATUS PRINTOUT CAN BE INHIBITED BY

SETTING THE "IBE" FLAG.

THE FOLLOWING ERRORS MAY BE PRINTED :

ERROR 2201 : READY BIT IN CSA NOT SET AFTER LOADING DATA REGISTER (TIMEOUT)
2202 AND 2203 CSA REGISTER CONTENTS IS:DDDDDD

TEST 23 - ANALOGUE OUTPUT (CALIBRATION ROUTINE) - SPECIFICALLY
SELECTABLE.

THIS TEST IS FOR CALIBRATION AND ALLOWS THE USER TO
FOLLOW A CALIBRATION PROCEDURE OR TO DO A MANUAL CALIBRATION.

FIRST OF ALL, THE USER GETS SOME INFORMATION ABOUT THE
CALIBRATION EQUIPMENT. HE IS THEN PROMPTED FOR THE CHANNEL
AND THE OUTPUT TYPE (VOLTAGE OR CURRENT). IF CURRENT IS
SELECTED, THE PROGRAM LOOKS AT THE COFS BIT IN THE CSA TO FIND
OUT THE OFFSET CONFIGURATION. NOW THE OPERATOR CAN CHOOSE
BETWEEN THE CALIBRATION PROCEDURE OR MANUAL VALUES; IF MANUAL IS
USED, THE OPERATOR CAN TYPE IN A VALUE IN MILLIVOLTS OR
MICROAMPS.

OTHERWISE HE IS PROMPTED TO CALIBRATE FIRST THE OFFSET AND THEN
THE GAIN. THEN HE IS REQUIRED TO TEST THE HALF FULL SCALE. TO GO
FROM ONE STEP TO THE NEXT, THE USER MUST TYPE CARRIAGE RETURN.
ALL STEPS (OFFSET, GAIN AND HALF FULL SCALE) MUST BE REPEATED
BEFORE GOING ON TO THE NEXT CHANNEL.

THE FOLLOWING ERRORS MAY BE PRINTED :

ERROR 2301 : READY BIT IN CSA NOT SET AFTER LOADING DATA REGISTER
2302, 2303 CSA REGISTER CONTENTS IS:DDDDDD
AND 2304

TEST 24 - ANALOGUE INPUT (CALIBRATION ROUTINE) - SPECIFICALLY
SELECTABLE.

THIS TEST IS FOR CALIBRATING AN ANALOGUE INPUT MODULE .

FIRST OF ALL, THE USER IS ASKED WHETHER HE NEEDS INFORMATION
ABOUT THE CALIBRATION PROCEDURE.

HE IS THEN PROMPTED FOR THE CHANNEL ON WHICH HE HAS PUT THE
PRECISION VOLTAGE REFERENCE SOURCE .

AFTER THIS HE IS INFORMED WHICH RANGE IS SWITCHED (UNIPOLAR OR
BIPOLAR) AND ASKED TO SELECT THE DESIRED GAIN.

NOW THE PROGRAM CONTINUOUSLY READS THE INPUT AND PRINTS THE
RESULT AS AN OCTAL AND VOLTAGE VALUE.

TO ABORT THE TEST, "CNTRL C" MUST BE TYPED.

THE FOLLOWING ERRORS MAY BE PRINTED :

ERROR 2401 : CSA CONTENTS INCORRECT AFTER READING DATA REGISTER
GOOD:GGGGGG, BAD:BBBBBB

ERROR 2402 : TIMEOUT - DONE NOT SET AFTER A/D START
GOOD:GGGGGG, BAD:BBBBBB

TEST 25 - VISUAL LED TEST - SPECIFICALLY SELECTABLE.

THIS TEST IS A VISUAL TEST. IT FLASHES THE LED ON AND OFF ON EVERY MODULE IN THE SYSTEM THAT IS FOUND BY THE AUTOMATIC CONFIGURATION ROUTINE OR SELECTED IN THE HARDWARE QUESTIONS.

MUX BOARDS ARE NOT INDEPENDENTLY ADDRESSABLE, SO THEIR LEDS ARE NOT FLASHED.

THE TEST RUNS UNTIL "CNTL C" IS TYPED.

ERRORS : THIS TEST HAS NO ERROR MESSAGES.

&

```

1815 .TITLE PROGRAM HEADER AND TABLES
1816 .SBTTL PROGRAM HEADER
1842
1847
1849 000000 .ENABL ABS,AMA
1850 002000 . 2000
1852
1853 002000 BGNMOD
1854
1855 : **
1856 : THE PROGRAM HEADER IS THE INTERFACE BETWEEN
1857 : THE DIAGNOSTIC PROGRAM AND THE SUPERVISOR.
1858 : **
1859
1860 002000 POINTER ALL
1861
1878
1879 002000 HEADER CZIXV,A,0,150,0,340
002000
002000 103
002001 132
002002 111
002003 130
002004 126
002005 000
002006 000
002007 000
002010
002010 101
002011
002011 060
002012
002012 000020
002014
002014 000150
002016
002016 002246
002020
002020 003376
002022
002022 002210
002024
002024 002232
002026
002026 104004
002030
002030 000000
002032
002032 000000
002034
002034 000000
002036
002036 000000
002040
002040 002124
002042
002042 000340

```

```

L$NAME::
        .ASCII /C/
        .ASCII /Z/
        .ASCII /I/
        .ASCII /X/
        .ASCII /V/
        .BYTE 0
        .BYTE 0
        .BYTE 0
L$REV::
        .ASCII /A/
L$DEPO::
        .ASCII /O/
L$UNIT::
        .WORD T$PTHV
L$TIML::
        .WORD 150
L$HPCP::
        .WORD L$HARD
L$SPCP::
        .WORD L$SOFT
L$HPTP::
        .WORD L$HW
L$SPTP::
        .WORD L$SW
L$LADP::
        .WORD L$LAST
L$STA::
        .WORD 0
L$CO::
        .WORD 0
L$DTYP::
        .WORD 0
L$APT::
        .WORD 0
L$DTP::
        .WORD L$DISPATCH
L$PRIO::
        .WORD 340

```


PROGRAM HEADER

```

002044
002044 000000
002046
002046 000000
002050
002050 003
002051 003
002052
002052 000000
002054 000000
002056
002056 000000
002060
002060 007206
002062
002062 017320
002064
002064 000000
002066
002066 000000
002070
002070 026702
002072
002072 026610
002074
002074 000000
002076
002076 007222
002100
002100 104035
002102
002102 007176
002104
002104 022350
002106
002106 026600
002110
002110 026514
002112
002112 022342
002114
002114 000000
002116
002116 000000
002120
002120 000000
    
```

```

L$ENVI:: .WORD 0
L$EXP1:: .WORD 0
L$MREV:: .BYTE C$REVISION
          .BYTE C$EDIT
L$EF:: .WORD 0
         .WORD 0
L$SPC:: .WORD 0
L$DEVP:: .WORD L$DVTYP
L$REPP:: .WORD L$RPT
L$EXP4:: .WORD 0
L$EXP5:: .WORD 0
L$AUT:: .WORD L$AU
L$DUT:: .WORD L$DU
L$LUN:: .WORD 0
L$DESP:: .WORD L$DESC
L$LOAD:: EMT E$LOAD
L$ETP:: .WORD L$ERRTBL
L$ICP:: .WORD L$INIT
L$CCP:: .WORD L$CLEAN
L$ACP:: .WORD L$AUTO
L$PRT:: .WORD L$PROT
L$TEST:: .WORD 0
L$DLY:: .WORD 0
L$HIME:: .WORD 0
    
```

DISPATCH TABLE

```

1892
1893
1894
1895
1896
1897
1898
1899 002122
      002122 000031
      002124
      002124 026714
      002126 027220
      002130 030144
      002132 031326
      002134 032414
      002136 033674
      002140 035154
      002142 035720
      002144 037076
      002146 040216
      002150 041076
      002152 041674
      002154 042776
      002156 043702
      002160 046404
      002162 052510
      002164 056536
      002166 061070
      002170 062124
      002172 063254
      002174 064500
      002176 070704
      002200 072532
      002202 077110
      002204 102102
1900

```

.SBTTL DISPATCH TABLE

```

;
; THE DISPATCH TABLE CONTAINS THE STARTING ADDRESS OF EACH TEST.
; IT IS USED BY THE SUPERVISOR TO DISPATCH TO EACH TEST.
;

```

DISPATCH 25.

```

      .WORD 25
L$DISPATCH::
      .WORD T1
      .WORD T2
      .WORD T3
      .WORD T4
      .WORD T5
      .WORD T6
      .WORD T7
      .WORD T8
      .WORD T9
      .WORD T10
      .WORD T11
      .WORD T12
      .WORD T13
      .WORD T14
      .WORD T15
      .WORD T16
      .WORD T17
      .WORD T18
      .WORD T19
      .WORD T20
      .WORD T21
      .WORD T22
      .WORD T23
      .WORD T24
      .WORD T25

```

DEFAULT HARDWARE P-TABLE

1908
1909
1910
1911
1912
1913
1914
1915
1916
1917
1918
1928
1929
1930
1931
1932
1933
1934
1935
1936
1937
1938

002206
002206 000010
002210
002210
002210 171000
002212 000000
002214 000200
002216 000000
002220 000000
002222 000003
002224 171400
002226 000400
002230
002230

.SBTTL DEFAULT HARDWARE P-TABLE

;;
; THE DEFAULT HARDWARE P-TABLE CONTAINS DEFAULT VALUES OF
; THE TEST-DEVICE PARAMETERS. THE STRUCTURE OF THIS TABLE
; IS IDENTICAL TO THE STRUCTURE OF THE HARDWARE P-TABLES,
; AND IS USED AS A "TEMPLATE" FOR BUILDING THE P-TABLES.
;--

BGNHW DFPTBL

.WORD 171000
.WORD 0
.WORD PRI04
.WORD 0
.WORD 0
.WORD 3
.WORD 171400
.WORD 400

; MODE REGISTER ADDRESS
; VECTOR ADDRESS
; PRIORITY LEVEL
; LOOPED ?
; LOW LEVEL INPUT ? (0 = NO)
; DEBOUNCE FOR DIGITAL LOOPBACK
; 3=ALL, 0=LOW, 1=NORMAL, 2=HIGH
; OTHER MODULE ADDRESS
; OTHER MODULE VECTOR ADDRESS

.WORD L10000-L\$HW/2
L\$HW::
DFPTBL::

ENDHW

L10000:

SOFTWARE P-TABLE

1940
1941
1942
1943
1944
1945
1946
1947
1948
1949
1950
1958
1959
1960
1961
1962
1963
1964
1965

002230
002230 000005
002232
002232
002232 000000
002234 000000
002236 177777
002240 000000
002242 000000
002244
002244

.SBTTL SOFTWARE P-TABLE

; THE SOFTWARE TABLE CONTAINS VARIOUS DATA USED BY THE
; PROGRAM AS OPERATIONAL PARAMETERS. THESE PARAMETERS ARE
; SET UP AT ASSEMBLY TIME AND MAY BE VARIED BY THE OPERATOR
; AT RUN TIME.

BGNSW SFPTBL

.WORD L10001-L\$SW/2
L\$SW::
SFPTBL::

MANTST:: .WORD 0 ; RUN MANUFACTURING TESTS ? (0 = NO)
MANIO:: .WORD 0 ; RUN LOOPBACK TESTS ? (0 = NO)
PAT1:: .WORD 177777 ; 1ST PATTERN FOR DIGITAL LOOPBACK
PAT2:: .WORD 0 ; 2ND PATTERN FOR DIGITAL LOOPBACK
QVP:: .WORD 0 ; QUICK VERIFY ? (0 = NO)

ENDSW

L10001:

HARDWARE PARAMETER CODING SECTION

```

002344 003302
002346 000000
002350 000770
2000
2001 002352          104:  ENDHRD
          002352          L10002: .EVEN
2002
2009
2010
2011 002352      115      117      104  G1:  .NLIST  BEX
2012 002400      126      105      103  G2:  .ASCIZ  /MODE REGISTER ADDRESS/
2013 002456      120      122      111  G3:  .ASCIZ  /VECTOR ADDRESS (FOR OUTPUT MODULES, TYPE "0")/
2014 002533      012      015      050  G3:  .ASCII  /PRIORITY LEVEL (FOR OUTPUT MODULES, TYPE "0")/
2015 002610      111      123      040  G4:  .ASCIZ  <12><15>/(FOR LSI WITH FIXED PRI. USE LEVEL 4 ONLY)/
2016 002673      103      101      102  G4:  .ASCII  /IS THIS MODULE CONNECTED TO ANOTHER VIA A LOOPBACK /
2017 002744      131      105      123  G4:  .ASCIZ  /CABLE/<15><12>/(ONLY MANUFACTURING SHOULD ANSWER /
2018 002751      114      117      127  G14: .ASCIZ  /YES)/
2019 003031      015      012      050  G14: .ASCII  /LOW LEVEL INPUT FOR DIGITAL INPUT LOOPBACK TESTS/
2020 003077      104      105      102  G15: .ASCIZ  <15><12>/(FOR OUTPUT LOOPBACK TYPE "N") /
2021 003166      065      060      060  G15: .ASCII  /DEBOUNCE PERIOD FOR DIGITAL INPUT LOOPBACK TESTS ONLY/<15><12>
2022 003237      117      124      110  G5:  .ASCIZ  /500US (0), 5MS (1), 10MS (2), OR ALL (3)/
2023 003302      117      124      110  G6:  .ASCIZ  /OTHER MODULE MODE REGISTER ADDRESS/
2024 003361      124      131      120  G6:  .ASCII  /OTHER MODULE VECTOR ADDRESS (IF OUTPUT MODULE, /
2025          .ASCIZ  /TYPE "0")/
2026          .LIST  BEX
          .EVEN

```

.WORD G6
.WORD T\$LOLIM
.WORD T\$HILIM

SOFTWARE PARAMETER CODING SECTION

```

2028          .SBTTL  SOFTWARE PARAMETER CODING SECTION
2029
2030          ;**
2031          ; THE SOFTWARE PARAMETER CODING SECTION CONTAINS MACROS
2032          ; THAT ARE USED BY THE SUPERVISOR TO BUILD P-TABLES.  THE
2033          ; MACROS ARE NOT EXECUTED AS MACHINE INSTRUCTIONS BUT ARE
2034          ; INTERPRETED BY THE SUPERVISOR AS DATA STRUCTURES.  THE
2035          ; MACROS ALLOW THE SUPERVISOR TO ESTABLISH COMMUNICATIONS
2036          ; WITH THE OPERATOR.
2037          ;--
2038
2039 003374          BGNSFT
2039 003374 000025          L$SOFT: .WORD L10003-L$SOFT/2
2039 003376
2040
2049
2050 003376          GPRML  G10.0,-1,YES          ; MANUFACTURING TESTS ?
2050 003376 000130          .WORD T$CODE
2050 003400 003450          .WORD G10
2050 003402 177777          .WORD -1
2051 003404          XFERF  10$          ; IF NOT, BRANCH
2051 003404 017044          .WORD T$CODE
2052 003406          GPRML  G11.2,-1,YES          ; MANUFACTURING I/O TESTS ?
2052 003406 001130          .WORD T$CODE
2052 003410 003500          .WORD G11
2052 003412 177777          .WORD -1
2053 003414          XFERF  10$          ; IF NOT, BRANCH
2053 003414 013044          .WORD T$CODE
2054 003416          GPRMD  G12.4,0,-1,0,177777,YES ; LOOPBACK PATTERN 1
2054 003416 002032          .WORD T$CODE
2054 003420 003533          .WORD G12
2054 003422 177777          .WORD -1
2054 003424 000000          .WORD T$LLOLIM
2054 003426 177777          .WORD T$HILIM
2055 003430          GPRMD  G13.6,0,-1,0,177777,YES ; LOOPBACK PATTERN 2
2055 003430 003032          .WORD T$CODE
2055 003432 003604          .WORD G13
2055 003434 177777          .WORD -1
2055 003436 000000          .WORD T$LLOLIM
2055 003440 177777          .WORD T$HILIM
2056 003442          10$: GPRML  G16.10,-1,YES          ; QUICK VERIFY MODE ?
2056 003442 004130          .WORD T$CODE
2056 003444 003656          .WORD G16
2056 003446 177777          .WORD -1
2057
2058          .EVEN
2059
2060 003450          ENDSFT
2060
2061          L10003: .EVEN
2061
2068
2069
2070 003450          122      125      116  G10: .NLIST  BEX
2070 003450          122      125      116  G10: .ASCIZ  /RUN MANUFACTURING TESTS/
2071 003500          122      125      116  G11: .ASCIZ  \RUN LOOPBACK AND I/O TESTS\
2072 003533          106      111      122  G12: .ASCIZ  /FIRST PATTERN; FOR DIGITAL LOOPBACK TESTS/
2073 003604          123      105      103  G13: .ASCIZ  /SECOND PATTERN FOR DIGITAL LOOPBACK TESTS/

```

SOFTWARE PARAMETER CODING SECTION

```
2074 003656    121    125    111 G16:    .ASCIZ /QUICK VERIFY MODE/  
2075                                            .LIST    BEX  
2076                                            .EVEN  
2077  
2078  
2079 003700                                    ENDMOD  
2080
```


SOFTWARE PARAMETER CODING SECTION

2092
 2093
 2121
 2131
 2132 003700
 2133
 2134
 2135
 2136
 2137
 2138
 2153
 2154 003700

.TITLE GLOBAL AREAS
 .SBTTL GLOBAL EQUATES SECTION

BGNMOD

 ; THE GLOBAL EQUATES SECTION CONTAINS PROGRAM EQUATES THAT
 ; ARE USED IN MORE THAN ONE TEST.
 ;--

EQUALS

;
 ; BIT DIFINITIONS
 ;

100000	BIT15== 100000
040000	BIT14== 40000
020000	BIT13== 20000
010000	BIT12== 10000
004000	BIT11== 4000
002000	BIT10== 2000
001000	BIT09== 1000
000400	BIT08== 400
000200	BIT07== 200
000100	BIT06== 100
000040	BIT05== 40
000020	BIT04== 20
000010	BIT03== 10
000004	BIT02== 4
000002	BIT01== 2
000001	BIT00== 1

;
 BIT9== BIT09
 BIT8== BIT08
 BIT7== BIT07
 BIT6== BIT06
 BIT5== BIT05
 BIT4== BIT04
 BIT3== BIT03
 BIT2== BIT02
 BIT1== BIT01
 BIT0== BIT00

;
 ; EVENT FLAG DEFINITIONS
 ; EF32:EF17 RESERVED FOR SUPERVISOR TO PROGRAM COMMUNICATION

000040	EF.START== 32.	; START COMMAND WAS ISSUED
000037	EF.RESTART== 31.	; RESTART COMMAND WAS ISSUED
000036	EF.CONTINUE== 30.	; CONTINUE COMMAND WAS ISSUED
000035	EF.NEW== 29.	; A NEW PASS HAS BEEN STARTED
000034	EF.PWR== 28.	; A POWER-FAIL/POWER-UP OCCURRED

;
 ; PRIORITY LEVEL DEFINITIONS

000340
 PRI07== 340

GLOBAL EQUATES SECTION

```

000300      PRI06== 300
000240      PRI05== 240
000200      PRI04== 200
000140      PRI03== 140
000100      PRI02== 100
000040      PRI01== 40
000000      PRI00== 0

```

; OPERATOR FLAG BITS

```

000004      EVL==      4
000010      LOT==     10
000020      ADR==     20
000040      IDU==     40
000100      ISR==    100
000200      UAM==    200
000400      BOE==    400
001000      PNT==   1000
002000      PRI==   2000
004000      IXE==   4000
010000      IBE==  10000
020000      IER==  20000
040000      LOE==  40000
100000      HOE== 100000

```

2155
2156
2157
2158
2159

```

171000      IXSTA== 171000      ; FIRST STANDARD IDV/IAV-11 ADDRESS
171770      IXEND== 171770     ; LAST STANDARD IDV/IAV-11 ADDRESS

```

; NB. SEE ALSO MESSAGE "NODEV" BELOW.

GLOBAL DATA SECTION

```

2161          .SBTTL GLOBAL DATA SECTION
2162
2163          ;**
2164          ; THE GLOBAL DATA SECTION CONTAINS DATA THAT ARE USED
2165          ; IN MORE THAN ONE TEST.
2166          ;--
2167
2168 003700 000000 MOD::          .WORD 0          ; MODE REGISTER ADDRESS OF CURRENT UUT
2169 003702 000000 DAT::          .WORD 0          ; DATA REGISTER ADDRESS OF CURRENT UUT
2170 003704 000000 CSA::          .WORD 0          ; CSR REGISTER A ADDRESS OF CURRENT UUT
2171 003706 000000 CSB::          .WORD 0          ; CSR REGISTER B ADDRESS OF CURRENT UUT
2172 003710 000000 VEC::          .WORD 0          ; VECTOR ADDRESS OF CURRENT UUT
2173 003712 000000 PRIO::         .WORD 0          ; PRIORITY LEVEL OF CURRENT UUT
2174 003714 000000 COND::         .WORD 0          ; SET IF UUT IS CONNECTED TO ANOTHER
2175 003716 000000 LOWLVL::       .WORD 0          ; LOW LEVEL FOR DIGITAL INPUT LOOPBACK
2176 003720 000003 DBOUNC::         .WORD 3          ; DEBOUNC PERIOD FOR DIGITAL INPUT LOOPB.
2177 003722 000000 OTHMOD::       .WORD 0          ; OTHER MODULE MODE REGISTER ADDRESS
2178 003724 000000 OTHDAT::       .WORD 0          ; OTHER MODULE DATA REGISTER ADDRESS
2179 003726 000000 OTHCSA::       .WORD 0          ; OTHER MODULE REGISTER A ADDRESS
2180 003730 000000 OTHCSB::       .WORD 0          ; OTHER MODULE REGISTER B ADDRESS
2181 003732 000000 OTHVEC::       .WORD 0          ; OTHER MODULE VECTOR ADDRESS
2182
2183 003734 100000 ECNT::          .WORD 100000    ; ERROR COUNT FOR UUT 0 - BIT 15 IS SET TO
2184 003736 100000          .WORD 100000    ; ERROR COUNT FOR UUT 1 - FLAG NOT TESTED.
2185 003740 100000          .WORD 100000    ; ERROR COUNT FOR UUT 2
2186 003742 100000          .WORD 100000    ; ERROR COUNT FOR UUT 3
2187 003744 100000          .WORD 100000    ; ERROR COUNT FOR UUT 4
2188 003746 100000          .WORD 100000    ; ERROR COUNT FOR UUT 5
2189 003750 100000          .WORD 100000    ; ERROR COUNT FOR UUT 6
2190 003752 100000          .WORD 100000    ; ERROR COUNT FOR UUT 7
2191 003754 100000          .WORD 100000    ; ERROR COUNT FOR UUT 8
2192 003756 100000          .WORD 100000    ; ERROR COUNT FOR UUT 9
2193 003760 100000          .WORD 100000    ; ERROR COUNT FOR UUT 10
2194 003762 100000          .WORD 100000    ; ERROR COUNT FOR UUT 11
2195 003764 100000          .WORD 100000    ; ERROR COUNT FOR UUT 12
2196 003766 100000          .WORD 100000    ; ERROR COUNT FOR UUT 13
2197 003770 100000          .WORD 100000    ; ERROR COUNT FOR UUT 14
2198 003772 100000          .WORD 100000    ; ERROR COUNT FOR UUT 15
2199
2200 003774 000000 GPADD::       .WORD 0          ; PARAMETER ADDRESS FOR UNIT 0
2201 003776 000000          .WORD 0          ; PARAMETER ADDRESS FOR UNIT 1
2202 004000 000000          .WORD 0          ; PARAMETER ADDRESS FOR UNIT 2
2203 004002 000000          .WORD 0          ; PARAMETER ADDRESS FOR UNIT 3
2204 004004 000000          .WORD 0          ; PARAMETER ADDRESS FOR UNIT 4
2205 004006 000000          .WORD 0          ; PARAMETER ADDRESS FOR UNIT 5
2206 004010 000000          .WORD 0          ; PARAMETER ADDRESS FOR UNIT 6
2207 004012 000000          .WORD 0          ; PARAMETER ADDRESS FOR UNIT 7
2208 004014 000000          .WORD 0          ; PARAMETER ADDRESS FOR UNIT 8
2209 004016 000000          .WORD 0          ; PARAMETER ADDRESS FOR UNIT 9
2210 004020 000000          .WORD 0          ; PARAMETER ADDRESS FOR UNIT 10
2211 004022 000000          .WORD 0          ; PARAMETER ADDRESS FOR UNIT 11
2212 004024 000000          .WORD 0          ; PARAMETER ADDRESS FOR UNIT 12
2213 004026 000000          .WORD 0          ; PARAMETER ADDRESS FOR UNIT 13
2214 004030 000000          .WORD 0          ; PARAMETER ADDRESS FOR UNIT 14
2215 004032 000000          .WORD 0          ; PARAMETER ADDRESS FOR UNIT 15
2216
2217 004034          DROPED::       .BLKB 16.        ; UNIT DROPPED FLAGS

```


GLOBAL DATA SECTION

```

2275
2276 004134 000000      MODE:: .WORD 0          ; MODE FOR DIGITAL/ANALOGUE CONVERSIONS
2277 004136 000000      GAIN:: .WORD 0         ; GAIN FOR DIGITAL/ANALOGUE CONVERSIONS
2278
2279 004140 000000      PADD:: .WORD 0         ; ADDRESS OF PROMPT FOR DECIMAL INPUT ROUTINE
2280
2281      ;THE FOLLOWING EQUATES ARE ONLY USED IN THE TESTS
2282
2283 004142 000000      ANS:: .WORD 0          ; TEMPORARY STORE FOR MANUAL INPUTS
2284 004144 000000      ITRCNT:: .WORD 0       ; ITERRATION COUNTER
2285 004146 000010      ITRDEF:: .WORD 10      ; ITERRATION DEFAULT
2286 004150 000000      INTFLA:: .WORD 0       ; DONE INTERRUPT FLAG
2287 004152 000000      INTFL2:: .WORD 0       ; ERROR INTERRUPT FLAG
2288 004154 000020      CYCLE:: .WORD 20       ; CYCLE COUNTER FOR PATTERN PAIR LOOPBACK
2289 004156 000003      CYCLLS:: .WORD 3        ; CYCLE COUNTER FOR LLS AND SLIDING TESTS
2290 004160 000500      CYCRAD:: .WORD 500     ; CYCLE COUNTER FOR RANDOM LOOPBACK TESTS
2291 004162 000020      LLWC:: .WORD 16.       ; INPUT MODULE LOW LEVEL WAIT VALUE (10MS)
2292 004164 007777      REDTIM:: .WORD 7777    ; READY TIMEOUT COUNTER
2293 004166 000001      OUTDE1:: .WORD 1       ; OPTO OUTPUT DEBOUNCE WAIT VALUE
2294 004170 000012      OUTDE2:: .WORD 12      ; RELAY OUTPUT DEBOUNCE WAIT VALUE
2295
2296      ;THE FOLLOWING TABEL GIVE THE LOAD VALUE OF HOW OFTEN THE 500 US
2297      ;WAIT LOOP WILL BE EXECUTED.
2298      ;THE TABLE IS FOR DIGITAL INPUT MODULES ONLY AND WILL BE USED IN THE
2299      ;DIGITAL INPUT LOOPBACK TESTS.(TEST 8,10,12)
2300
2301 004172 000          DBTAB:: .BYTE 0          ; DUMMY LOCATION FOR USE OF R5
2302 004173 001          ; .BYTE 1          ; VALUE FOR INPUT MODULE DEB. 500US
2303 004174 013          ; .BYTE 11.         ; VALUE FOR INPUT MODULE DEB. 5MS
2304 004175 025          ; .BYTE 21.         ; VALUE FOR INPUT MODULE DEB. 10MS
2305
2306      .EVEN
2307
2308

```

GLOBAL DATA SECTION

2310 000012

.RADIX 10
.NLIST BEX

; ANALOGUE/DIGITAL CONVERSION TABLES USED BY ROUTINES DACON AND ADCON.

; VOLTAGE UNIPOLAR TABLE MODE 0 (0-10V)

; BITS 11 10 9 8 7 6 5 4 3 2 1 0

Line	Code	Code	Code	Code	Table	11	10	9	8	7	6	5	4	3	2	1	0	Units
2319	004176	011610	004704	002342	VUPTAB::	5000.	2500.	1250.	625.	312.	156.	78.	39.	19.	9.	4.	2.	MV G=1
2320	004226	000000	000000	000000		0.	0.	0.	0.	500.	250.	125.	63.	531.	766.	883.	441.	UV
2322	004256	004704	002342	001161		2500.	1250.	625.	312.	156.	78.	39.	19.	9.	4.	2.	1.	MV G=2
2323	004306	000000	000000	000000		0.	0.	0.	500.	250.	125.	63.	531.	766.	883.	441.	221.	UV
2325	004336	001750	000764	000372		1000.	500.	250.	125.	62.	31.	15.	7.	3.	1.	0.	0.	MV G=5
2326	004366	000000	000000	000000		0.	0.	0.	0.	500.	250.	625.	813.	906.	953.	977.	488.	UV
2328	004416	000764	000372	000175		500.	250.	125.	62.	31.	15.	7.	3.	1.	0.	0.	0.	MV G=10
2329	004446	000000	000000	000000		0.	0.	0.	500.	250.	625.	813.	906.	953.	977.	488.	244.	UV
2331	004476	000372	000175	000076		250.	125.	62.	31.	15.	7.	3.	1.	0.	0.	0.	0.	MV G=20
2332	004526	000000	000000	000764		0.	0.	500.	250.	625.	813.	906.	953.	977.	488.	244.	122.	UV
2334	004556	000144	000062	000031		100.	50.	25.	12.	6.	3.	1.	0.	0.	0.	0.	0.	MV G=50
2335	004606	000000	000000	000000		0.	0.	0.	500.	250.	125.	563.	781.	391.	195.	98.	49.	UV
2337	004636	000062	000031	000014		50.	25.	12.	6.	3.	1.	0.	0.	0.	0.	0.	0.	MV G=100
2338	004666	000000	000000	000764		0.	0.	500.	250.	125.	563.	781.	391.	195.	98.	49.	24.	UV
2340	004716	000031	000014	000006		25.	12.	6.	3.	1.	0.	0.	0.	0.	0.	0.	0.	MV G=200
2341	004746	000000	000764	000372		0.	500.	250.	125.	563.	781.	391.	195.	98.	49.	24.	12.	UV

GLOBAL DATA SECTION

2402
2403
2404
2405
2406
2407
2408
2409
2410
2411
2412
2413
2414
2415
2416
2417
2418
2419
2420
2421
2422
2423
2424
2425
2426
2427
2428
2429
2430
2431
2432
2433

006376 017500 007640 003720
006426 000000 000000 000000
006456 007640 003720 001750
006506 000000 000000 000000
006536 003100 001440 000620
006566 000000 000000 000000
006616 001440 000620 000310
006646 000000 000000 000000
006676 000620 000310 000144
006726 000000 000000 000000
006756 000240 000120 000050
007006 000000 000000 000000
007036 000120 000050 000024
007066 000000 000000 000000
007116 000050 000024 000012
007146 000000 000000 000000
000010

; CURRENT 4 - 20 MA TABLE (MODE 3)

; BITS 11 10 9 8 7 6 5 4 3 2 1 0

I4TAB:: 8000,4000,2000,1000, 500, 250, 125, 62, 31, 15, 7, 3 ; UA G=1
0, 0, 0, 0, 0, 0, 0, 500, 250, 625, 813, 906 ; NA
4000,2000,1000, 500, 250, 125, 62, 31, 15, 7, 3, 1 ; UA G=2
0, 0, 0, 0, 0, 0, 500, 250, 625, 813, 906, 953 ; NA
1600, 800, 400, 200, 100, 50, 25, 12, 6, 3, 1, 0 ; UA G=5
0, 0, 0, 0, 0, 0, 500, 250, 125, 563, 781 ; NA
800, 400, 200, 100, 50, 25, 12, 6, 3, 1, 0, 0 ; UA G=10
0, 0, 0, 0, 0, 0, 500, 250, 125, 563, 781, 390 ; NA
400, 200, 100, 50, 25, 12, 6, 3, 1, 0, 0, 0 ; UA G=20
0, 0, 0, 0, 0, 500, 250, 125, 563, 781, 390, 195 ; NA
160, 80, 40, 20, 10, 5, 2, 1, 0, 0, 0, 0 ; UA G=50
0, 0, 0, 0, 0, 0, 500, 250, 625, 313, 156, 78 ; NA
80, 40, 20, 10, 5, 2, 1, 0, 0, 0, 0, 0 ; UA G=100
0, 0, 0, 0, 0, 500, 250, 625, 313, 156, 78, 39 ; NA
40, 20, 10, 5, 2, 1, 0, 0, 0, 0, 0, 0 ; UA G=200
0, 0, 0, 0, 500, 250, 625, 313, 156, 78, 39, 20 ; NA

.RADIX 8
.LIST BEX

GLOBAL DATA SECTION

2435
2448
2449

007176
007176 000000
007200 000000
007202 000000
007204 000000

ERRTBL

ERRTYP:: .WORD 0
ERRNBR:: .WORD 0
ERRMSG:: .WORD 0
ERRBLK:: .WORD 0

L\$ERRTBL::

GLOBAL TEXT SECTION

```

2451          .SBTTL GLOBAL TEXT SECTION
2452
2453          ;**
2454          ; THE GLOBAL TEXT SECTION CONTAINS FORMAT STATEMENTS,
2455          ; MESSAGES, AND ASCII INFORMATION THAT ARE USED IN
2456          ; MORE THAN ONE TEST.
2457          ;--
2458
2459          .NLIST BEX
2460
2461          ;
2462          ; NAMES OF DEVICES SUPPORTED BY PROGRAM
2463          ;
2464          DEVTYP <IDV\IAV-11>
2465          007206
2466          007206      111      104      126          L$DVTYP::
2467          007206          .ASCIZ /IDV\IAV-11/
2468
2469          ;
2470          ; TEST DESCRIPTION
2471          ;
2472          DESCRIPT      <IDV\IAV-11 DIAGNOSTIC>
2473          007222
2474          007222      111      104      126          L$DESC::
2475          007222          .ASCIZ /IDV\IAV-11 DIAGNOST
2476
2477          ;
2478          ; FORMAT STATEMENTS USED IN PRINT CALLS
2479          ;
2480          ;
2481          ;
2482          ;
2483          ;
2484          ;
2485          ;
2486          ;
2487          ;
2488          ;
2489          ;
2490          ;
2491          ;
2492          ;
2493          ;
2494          ;
2495          ;
2496          ;
2497          ;
2498          ;
2499          ;
2500          ;
2501          ;
2502          ;

```

GLOBAL ERROR REPORT SECTION

```

2511          .SBTTL GLOBAL ERROR REPORT SECTION
2512
2513          ;**
2514          ; THE GLOBAL ERROR REPORT SECTION CONTAINS MESSAGE PRINTING AREAS
2515          ; USED BY MORE THAN TEST TO OUTPUT ADDITIONAL ERROR INFORMATION. PRINTB
2516          ; (BASIC) AND PRINTX (EXTENDED) CALLS ARE USED TO CALL PRINT SERVICES.
2517          ;--
2518
2519
2535
2536 007342          BGNMSG EER1
2537 007342          PRINTB #GOOBAD,GOOD,BAD          ; PRINT GOOD AND BAD          EER1::
2538 007342          013746 004064          MOV          BAD,-(SP)
2539 007342          007346 013746 004062          MOV          GOOD,-(SP)
2540 007342          007352 012746 007732          MOV          #GOOBAD,-(SP)
2541 007342          007356 012746 000003          MOV          #3,-(SP)
2542 007342          007362 010600          MOV          SP,R0
2543 007342          007364 104414          TRAP         C$PNTB
2544 007342          007366 062706 000010          ADD          #10,SP
2545 007372          004737 011342          JSR          PC,CHKMAX          ; CHECK FOR TOO MANY ERRORS
2546 007376          ENDMSG
2547 007376          104423          L10004:      TRAP         C$MSG
2548
2549
2550
2551 007400          BGNMSG EER2
2552 007400          PRINTB #EMG3,GOOD,BAD,@MOD          EER2::
2553 007400          017746 174274          MOV          @MOD,-(SP)
2554 007400          007404 013746 004064          MOV          BAD,-(SP)
2555 007400          007410 013746 004062          MOV          GOOD,-(SP)
2556 007400          007414 012746 010101          MOV          #EMG3,-(SP)
2557 007400          007420 012746 000004          MOV          #4,-(SP)
2558 007400          007424 010600          MOV          SP,R0
2559 007400          007426 104414          TRAP         C$PNTB
2560 007400          007430 062706 000012          ADD          #12,SP
2561 007434          004737 011342          JSR          PC,CHKMAX
2562 007440          ENDMSG
2563 007440          104423          L10005:      TRAP         C$MSG
2564
2565
2566
2567 007442          BGNMSG EER3
2568 007442          PRINTB #EMG4,@CSA          EER3::
2569 007442          017746 174236          MOV          @CSA,-(SP)
2570 007442          007446 012746 010166          MOV          #EMG4,-(SP)
2571 007442          007452 012746 000002          MOV          #2,-(SP)
2572 007442          007456 010600          MOV          SP,R0
2573 007442          007460 104414          TRAP         C$PNTB
2574 007442          007462 062706 000006          ADD          #6,SP
2575 007466          004737 011342          JSR          PC,CHKMAX
2576 007472          ENDMSG
2577 007472          104423          L10006:      TRAP         C$MSG
2578
2551
2552

```

GLOBAL ERROR REPORT SECTION

2553	007474			BGNMSG	EER4				
	007474								
2554	007474	000304		SWAB	R4			EER4::	
2555	007476			PRINTB	#EMG5,GOOD,BAD,R4				
	007476	010446							MOV R4,-(SP)
	007500	013746	004064						MOV BAD,-(SP)
	007504	013746	004062						MOV GOOD,-(SP)
	007510	012746	010230						MOV #EMG5,-(SP)
	007514	012746	000004						MOV #4,-(SP)
	007520	010600							MOV SP,R0
	007522	104414							TRAP C\$PNTB
	007524	062706	000012						ADD #12,SP
2556	007530	000304		SWAB	R4				
2557	007532	004737	011342	JSR	PC,CHKMAX				
2558	007536			ENDMSG					
	007536	104423						L10007:	TRAP C\$MSG
2559									
2560									
2561	007540			BGNMSG	EER5				
	007540								
2562	007540			PRINTB	#EMG6,GOOD,BAD,@CSA			EER5::	
	007540	017746	174140						MOV @CSA,-(SP)
	007544	013746	004064						MOV BAD,-(SP)
	007550	013746	004062						MOV GOOD,-(SP)
	007554	012746	010314						MOV #EMG6,-(SP)
	007560	012746	000004						MOV #4,-(SP)
	007564	010600							MOV SP,R0
	007566	104414							TRAP C\$PNTB
	007570	062706	000012						ADD #12,SP
2563	007574	004737	011342	JSR	PC,CHKMAX				
2564	007600			ENDMSG					
	007600	104423						L10010:	TRAP C\$MSG
2565									
2566									
2567	007602			BGNMSG	EER6				
	007602								
2568	007602			PRINTB	#EMG4,BAD			EER6::	
	007602	013746	004064						MOV BAD,-(SP)
	007606	012746	010166						MOV #EMG4,-(SP)
	007612	012746	000002						MOV #2,-(SP)
	007616	010600							MOV SP,R0
	007620	104414							TRAP C\$PNTB
	007622	062706	000006						ADD #6,SP
2569	007626	004737	011342	JSR	PC,CHKMAX				
2570	007632			ENDMSG					
	007632	104423						L10011:	TRAP C\$MSG
2571									
2572									
2573									
2574	007634			BGNMSG	EERA				
	007634								
2575	007634			PRINTB	#EMG1,R1			EERA::	
	007634	010146							MOV R1,-(SP)
	007636	012746	007762						MOV #EMG1,-(SP)

GLOBAL ERROR REPORT SECTION

007642	012746	000002							
007646	010600								MOV #2,-(SP)
007650	104414								MOV SP,R0
007652	062706	000006							TRAP C\$PNTB
2576	007656	004737	011342	JSR	PC,CHKMAX				ADD #6,SP
2577	007662			ENDMSG					
	007662								
	007662	104423						L10012:	TRAP C\$MSG
2578									
2579									
2580	007664			BGNMSG	EERB				
	007664								
2581	007664			PRINTB	#EMG2,R5,GOOD,BAD			EERB::	
	007664	013746	004064						MOV BAD,-(SP)
	007670	013746	004062						MOV GOOD,-(SP)
	007674	010546							MOV R5,-(SP)
	007676	012746	010031						MOV #EMG2,-(SP)
	007702	012746	000004						MOV #4,-(SP)
	007706	010600							MOV SP,R0
	007710	104414							TRAP C\$PNTB
	007712	062706	000012						ADD #12,SP
2582	007716	004737	011342	JSR	PC,CHKMAX				
2583	007722			ENDMSG					
	007722								
	007722	104423						L10013:	TRAP C\$MSG
2584									
2585									
2586	007724			BGNMSG	EERG				
	007724								
2587	007724	004737	011342	JSR	PC,CHKMAX			EERG::	
2588	007730			ENDMSG					
	007730								
	007730	104423						L10014:	TRAP C\$MSG
2589									
2590									
2591	007732	045	101	107	.NLIST BEX				
2592	007762	045	101	122	GOOBAD: .ASCIZ	/#AGOOD:#06#A, BAD:#06#N/			
2593	010031	045	101	101	EMG1: .ASCIZ	/#AREGISTER AT #06#A DOES NOT RESPOND#N/			
2594	010101	045	101	107	EMG2: .ASCIZ	/#AADDRESS: #06#A, GOOD:#06#A, BAD:#06#N/			
2595	010166	045	101	103	EMG3: .ASCIZ	/#AGOOD:#06#A, BAD:#06#A, MOD REGISTER CONTENTS:#06#N/			
2596	010230	045	101	107	EMG4: .ASCIZ	/#ACSA REGISTER CONTENTS IS :#06#N/			
2597	010314	045	101	107	EMG5: .ASCIZ	/#AGOOD:#06#A, BAD:#06#A, ADDRESSED CHANNEL IS:#D3#N/			
2598					EMG6: .ASCIZ	/#AGOOD:#06#A, BAD:#06#A, CSA REGISTER CONTENTS:#06#N/			
2599					.LIST BEX				
2600					.EVEN				

GLOBAL SUBROUTINES SECTION

```

2602 .SBTTL GLOBAL SUBROUTINES SECTION
2603
2604 :**
2605 : THE GLOBAL SUBROUTINES SECTION CONTAINS THE SUBROUTINES
2606 : THAT ARE USED IN MORE THAN ONE TEST.
2607 :--
2608
2609 : SUBROUTINE REGTST - GENERAL PURPOSE REGISTER TEST.
2610
2611 :**
2612 : FUNCTIONAL DESCRIPTION:
2613 :
2614 : CHECKS THAT ALL READ/WRITE BITS OF THE SELECTED REGISTER CAN BE
2615 : SET, CLEARED, AND INDIVIDUALLY SET (SLIDING ONES PATTERN).
2616 :
2617 : INPUTS:
2618 :
2619 : IF ENTERED AT LOCATION REGTST, THE LOCATIONS FOLLOWING THE
2620 : SUBROUTINE CALL MUST CONTAIN THE READ/WRITE BIT MASK, THE
2621 : ADDRESS OF THE REGISTER TO BE TESTED, AND THE FIRST ERROR NUMBER
2622 : TO BE USED (SEE CALLING SEQUENCE).
2623 :
2624 : IF ENTERED AT LOCATION REGTST1, THE READ/WRITE BIT MASK, REGISTER
2625 : ADDRESS TO BE TESTED, AND THE FIRST ERROR NUMBER MUST BE LOADED
2626 : INTO LOCATIONS MASK, REGADD, AND ERRNBR RESPECTIVELY. THIS
2627 : ALLOWS THE ARGUMENTS TO BE VARIED AT RUN TIME.
2628 :
2629 : IMPLICIT INPUTS:
2630 :
2631 : NONE.
2632 :
2633 : OUTPUTS:
2634 :
2635 : ERROR MESSAGES IF ERRORS OCCUR.
2636 :
2637 : IMPLICIT OUTPUTS:
2638 :
2639 : IF ENTERED AT LOCATION REGTST,
2640 :
2641 : MASK - CONTAINS THE READ/WRITE BIT MASK
2642 : REGADD - CONTAINS THE ADDRESS OF THE REGISTER BEING TESTED
2643 :
2644 : ALWAYS,
2645 :
2646 : MASCOM - CONTAINS THE COMPLEMENT OF THE MASK
2647 : GOOD - CONTAINS LAST EXPECTED DATA
2648 : BAD - CONTAINS LAST ACTUAL DATA
2649 : ERRNBR - CONTAINS THE INPUT ERROR NUMBER * 2
2650 : ERRTP - CONTAINS 3 (SOFT ERROR)
2651 : ERRBLK - CONTAINS ADDRESS OF REGERR (REGISTER ERROR MESSAGE)
2652 : ERRMSG - CONTAINS 3RD REGISTER ERROR MESSAGE
2653 :
2654 : SUBORDINATE ROUTINES USED:
2655 :
2656 : INSERT - ERROR INSERTION ROUTINE
2657 : CHKMAX - ERROR COUNT CHECKING ROUTINE
2658 : DRS ERROR MACRO

```

GLOBAL SUBROUTINES SECTION

```

2659
2660 ; FUNCTIONAL SIDE EFFECTS:
2661 ;
2662 ;     NONE.
2663 ;
2664 ; CALLING SEQUENCE:
2665 ;
2666 ;     EITHER FIXED PARAMETERS FOLLOW THE SUBROUTINE CALL :
2667 ;
2668 ;     EG.      CALL      REGTST
2669 ;             177          ; BIT MASK OF R/W BITS
2670 ;             CSR          ; REGISTER ADDRESS
2671 ;             200.         ; FIRST ERROR NUMBER
2672 ;
2673 ;     OR PARAMETERS ARE SET DYNAMICALLY :
2674 ;
2675 ;     EG.      MOV      #177,MASK      ; BIT MASK OF R/W BITS
2676 ;             MOV      CSR,REGADD     ; REGISTER ADDRESS
2677 ;             MOV      #200.,ERRNBR   ; FIRST ERROR NUMBER
2678 ;             CALL     REGTS1
2679 ;
2680 ; --
2681
2682 010402 REGTST::
2683 010402 017637 000000 010754 MOV      @ (SP),MASK      ; GET R/W BIT MASK
2684 010410 062716 000002          ADD      #2,(SP)          ; JUMP OVER ARGUMENT
2685 010414 017637 000000 010760 MOV      @ (SP),REGADD   ; GET REGISTER ADDRESS
2686 010422 062716 000002          ADD      #2,(SP)          ; JUMP OVER ARGUMENT
2687 010426 017637 000000 007200 MOV      @ (SP),ERRNBR   ; GET FIRST ERROR NUMBER
2688 010434 062716 000002          ADD      #2,(SP)          ; JUMP OVER ARGUMENT
2689 010440
2690 010440 013737 010754 010756 MOV      MASK,MASCOM     ; SET UP COMPLEMENT
2691 010446 005137 010756          COM      MASCOM          ; OF R/W BIT MASK
2692 010452 012737 000003 007176 MOV      #3,ERRTYP       ; SET UP FOR SOFT ERROR
2693 010460 012737 010762 007204 MOV      @REGERR,ERRBLK  ; SET UP ERROR MESSAGE ROUTINE
2694 010466 012737 011030 007202 MOV      @RERR1,ERRMSG   ; FIRST ERROR MESSAGE
2695
2696 ; CHECK THAT ALL R/W BITS CAN BE SET
2697 ;
2698 010474 013737 010754 004062 MOV      MASK,GOOD      ; SET UP EXPECTED DATA
2699 010502          BGNSEG
2700 010504 053777 004062 000246 BIS      GOOD,@REGADD    ; SET ALL R/W BITS          TRAP      C$BSEG
2701 010512 017737 000242 004064 MOV      @REGADD,BAD     ; READ THE RESULT
2702 010520 043737 010756 004064 BIC      MASCOM,BAD      ; KEEP ONLY R/W BITS
2703 010526 023737 004064 004062 CMP      BAD,GOOD        ; ALL R/W BITS SET?
2704 010534 004737 012146          CALL     INSERT          ; ALLOW FORCED ERROR PRINTOUT
2705 010540 001401          BEQ      10$           ; IF OK, BRANCH
2706 010542          ERROR          ; ELSE REPORT ERROR
2707 010544          10$:          ENDSEG          TRAP      C$ERROR
2708 010544          10000$:          TRAP      C$ESEG
2709 010544 104405
2710
2711 010546 005037 004062 CLR      GOOD            ; SET UP EXPECTED DATA

```


GLOBAL SUBROUTINES SECTION

```

2712 010552 005237 007200          INC      ERRNBR          ; NEXT ERROR NUMBER
2713 010556 012737 011102 007202  MOV      @RERR2,ERRMSG ; NEXT ERROR MESSAGE
2714
2715 010564          BGNSEG
      010564 104404
2716 010566 043777 010754 000164          BIC      MASK,@REGADD ; CLEAR ALL R/W BITS          TRAP      C$BSEG
2717 010574 017737 000160 004064          MOV      @REGADD,BAD ; READ THE RESULT
2718 010602 043737 010756 004064          BIC      MASCOM,BAD ; KEEP ONLY R/W BITS
2719 010610 023737 004064 004062          CMP      BAD,GOOD ; ALL R/W BITS CLEAR?
2720 010616 004737 012146          CALL    INSERT ; ALLOW FORCED ERROR PRINTOUT
2721 010622 001401          BEQ     20$ ; IF OK, BRANCH
2722 010624          ERROR ; ELSE REPORT ERROR
      010624 104460
2723 010626          20$: ENDSEG ; TRAP      C$ERROR
      010626          10001$: TRAP      C$ESEG
      010626 104405
2724          ;
2725          ; CHECK THAT EACH R/W BIT CAN BE SET
2726          ;
2727 010630 005237 007200          INC      ERRNBR          ; NEXT ERROR NUMBER
2728 010634 012737 011160 007202  MOV      @RERR3,ERRMSG ; NEXT ERROR MESSAGE
2729 010642 012737 000001 004062  MOV      #1,GOOD ; FIRST BIT TO TEST
2730 010650 033737 004062 010754 30$: BIT      GOOD,MASK ; R/W BIT?
2731 010656 001004          BNE     50$ ; IF YES, TEST IT
2732 010660 006337 004062          40$: ASL      GOOD ; ELSE FIND NEXT R/W BIT
2733 010664 103427          BCS     70$ ; IF ALL DONE, RETURN
2734 010666 000770          BR      30$ ; ELSE CHECK IF NEXT IS R/W
2735
2736 010670          50$: BGNSEG
      010670 104404
2737 010672 043777 010754 000060          BIC      MASK,@REGADD ; CLEAR ALL R/W BITS          TRAP      C$BSEG
2738 010700 053777 004062 000052          BIS      GOOD,@REGADD ; SET ONE BIT
2739 010706 017737 000046 004064          MOV      @REGADD,BAD ; READ IT BACK
2740 010714 043737 010756 004064          BIC      MASCOM,BAD ; KEEP ONLY R/W BITS
2741 010722 023737 004064 004062          CMP      BAD,GOOD ; ALL OTHER BITS CLEAR?
2742 010730 004737 012146          CALL    INSERT ; ALLOW FORCED ERROR PRINTOUT
2743 010734 001401          BEQ     60$ ; IF OK, BRANCH
2744 010736          ERROR ; ELSE REPORT ERROR
      010736 104460
2745 010740          60$: ENDSEG ; TRAP      C$ERROR
      010740          10002$: TRAP      C$ESEG
      010740 104405
2746 010742 000746          BR      40$ ; TEST NEXT BIT
2747
2748 010744 043777 010754 000006 70$: BIC      MASK,@REGADD ; LEAVE THE R/W BITS CLEAR
2749 010752 000207          RETURN
2750
2751
2752 010754 000000          MASK:: .WORD 0 ; BIT MASK OF READ/WRITE BITS
2753 010756 000000          MASCOM: .WORD 0 ; COMPLEMENT OF MASK
2754 010760 000000          REGADD:: .WORD 0 ; ADDRESS OF REGISTER TO BE TESTED
2755
2756 010762          BGNMSG REGERR
      010762
2757 010762          PRINTB @REGMSG,REGADD,GOOD,BAD,MASK          REGERR::
      010762 013746 010754          MOV      MASK,-(SP)
      010766 013746 004064          MOV      BAD,-(SP)

```

GLOBAL SUBROUTINES SECTION

```

010772 013746 004062
010776 013746 010760
011002 012746 011247
011006 012746 000005
011012 010600
011014 104414
011016 062706 000014
2758 011022 004737 011342 JSR PC,CHKMAX ; CHECK FOR TOO MANY ERRORS
2759 011026 ENDMSG
011026
011026 104423 L10015: TRAP C$MSG
2760
2761 .NLIST BEX
2762
2763 011030 122 105 107 RERR1: .ASCIZ #REGISTER READ/WRITE BITS COULD NOT BE SET#
2764 011102 122 105 107 RERR2: .ASCIZ #REGISTER READ/WRITE BITS COULD NOT BE CLEARED#
2765 011160 122 105 107 RERR3: .ASCIZ #REGISTER READ/WRITE BITS COULD NOT BE INDIVIDUALLY SET#
2766
2767 011247 045 101 101 REGMSG: .ASCIZ .#ADDRESS: #06#A, GOOD: #06#A, BAD: #06#A, R/W BITS: #06#N.
2768
2769 .LIST BEX
2770 .EVEN

```

```

MOV GOOD, -(SP)
MOV REGADD, -(SP)
MOV #REGMSG, -(SP)
MOV #5, -(SP)
MOV SP, R0
TRAP C$PNTB
ADD #14, SP

```

GLOBAL SUBROUTINES SECTION

```

2772      ; SUBROUTINE CHKMAX - ERROR COUNT CHECKING ROUTINE.
2773      ;
2774      ; **
2775      ; FUNCTIONAL DESCRIPTION:
2776      ;
2777      ;     SUBROUTINE TO UPDATE UNIT ERROR COUNT. IF THE PROGRAM IS LOOPING
2778      ;     ON AN ERROR, THE SUBROUTINE DOES NOTHING. OTHERWISE, THE ERROR
2779      ;     COUNT FOR THE UNIT IS INCREMENTED. IF THE ERROR COUNT EXCEEDS 5
2780      ;     AND THE USER FLAG EVL HAS BEEN SELECTED AND THE FLAG IDU IS NOT
2781      ;     SELECTED, THE UNIT IS DROPPED FROM THE TEST CYCLE.
2782      ;
2783      ; INPUTS:
2784      ;
2785      ;     NONE.
2786      ;
2787      ; IMPLICIT INPUTS:
2788      ;
2789      ;     L$LUN CONTAINS THE NUMBER OF THE UNIT CURRENTLY BEING TESTED.
2790      ;     ECNT IS THE ADDRESS OF THE ERROR COUNT FOR UNIT 0.
2791      ;
2792      ; OUTPUTS:
2793      ;
2794      ;     NONE.
2795      ;
2796      ; IMPLICIT OUTPUTS:
2797      ;
2798      ;     THE ERROR COUNT FOR THE LOGICAL UNIT BEING TESTED IS
2799      ;     INCREMENTED IF THE PROGRAM IS NOT LOOPING.
2800      ;
2801      ; SUBORDINATE ROUTINES USED:
2802      ;
2803      ;     NONE.
2804      ;
2805      ; FUNCTIONAL SIDE EFFECTS:
2806      ;
2807      ;     IF THE ERROR COUNT EXCEEDS 5 AND THE USER EVL FLAG IS SELECTED,
2808      ;     AND THE 'LOOP ON TEST' AND 'INHIBIT DROPPING OF UNITS' FLAGS ARE
2809      ;     NOT SELECTED, THE UNIT WILL BE DROPPED FROM TESTING.
2810      ;
2811      ; CALLING SEQUENCE:
2812      ;
2813      ;     JSR PC,CHKMAX
2814      ;
2815      ; --
2816
2817 011342      CHKMAX::INLOOP      ; LOOPING ON ERROR?
2818 011342      104420             ; IF YES, EXIT          TRAP   C$INLP
2819 011344      103436             ;                       BCS    10$
2820 011346      013700 002074      MOV     L$LUN,RO      ; GET CURRENT UNIT
2821 011352      006300             ASL     RO          ; CONVERT TO ERROR COUNT OFFSET
2822 011354      005260 003734      INC     ECNT(RO)   ; UPDATE THE ERROR COUNT
2823 011360      026027 003734 000005  CMP     ECNT(RO),#5 ; TOO MANY ERRORS?
2824 011366      003425             BLE     10$        ; IF NOT, JUMP
2825
2826 011370      RFLAGS RO          ; GET OPERATOR FLAGS

```

GLOBAL SUBROUTINES SECTION

```

011370 104421
2827 011372 032700 000040      BIT    #IDU,RO      ; IS DROPPING INHIBITED?
2828 011376 001021              BNE    10$         ; IF YES, EXIT
2829 011400 032700 000004      BIT    #EVL,RO     ; EVALUATE FLAG SELECTED ?
2830 011404 001416              BEQ    10$         ; IF NOT, EXIT
2831
2832 011406              PRINTF #NERRS,L$LUN ; 'TOO MANY ERRORS'
011406 013746 002074
011412 012746 011444
011416 012746 000002
011422 010600
011424 104417
011426 062706 000006
2833 011432              DODU   L$LUN       ; DROP THE UNIT
011432 013700 002074
011436 104451
2834
2835 011440              DOCLN                ; END THE SUBPASS
011440 104444
2836
2837 011442 000207              10$:  RTS    PC
2838
2839
2840 011444 045 116 045 NERRS: .NLIST BEX
                                .ASCIZ /#N#AMORE THAN 5 ERRORS ON UNIT#D2/
2841                                .LIST  BEX
2842                                .EVEN
2843
2844
2845
TRAP  C$RFLA
MOV   L$LUN,-(SP)
MOV   #NERRS,-(SP)
MOV   #2,-(SP)
MOV   SP,RO
TRAP  C$PNTF
ADD   #6,SP
MOV   L$LUN,RO
TRAP  C$DODU
TRAP  C$DCLN

```

GLOBAL SUBROUTINES SECTION

```

2847 ; SUBROUTINES WT25M, WT500 AND WT25 - DELAY ROUTINES.
2848 ;
2849 ; **
2850 ; FUNCTIONAL DESCRIPTION:
2851 ;
2852 ;     SUBROUTINE TO WAIT FOR 25 MILLISECONDS, 500 MICROSECONDS OR 25
2853 ;     MICROSECONDS.
2854 ;
2855 ;     NOTE. BECAUSE OF THE SMALL NUMBER OF PROGRAM WAIT LOOPS USED FOR
2856 ;     THE 25 MICROSECOND COUNTER, THE ACCURACY OF THE WT25 ROUTINE
2857 ;     IS LOW. THE DELAY MAY LAST UP TO 50 MICROSECONDS ON SOME
2858 ;     SLOW PROCESSORS.
2859 ;
2860 ; INPUTS:
2861 ;
2862 ;     NONE.
2863 ;
2864 ; IMPLICIT INPUTS:
2865 ;
2866 ;     THE VARIABLES CNT25M, CNT500, AND CNT25 MUST HAVE BEEN SET UP BY
2867 ;     ROUTINE SETCLK.
2868 ;
2869 ; OUTPUTS:
2870 ;
2871 ;     NONE.
2872 ;
2873 ; IMPLICIT OUTPUTS:
2874 ;
2875 ;     NONE.
2876 ;
2877 ; SUBORDINATE ROUTINES USED:
2878 ;
2879 ;     NONE.
2880 ;
2881 ; FUNCTIONAL SIDE EFFECTS:
2882 ;
2883 ;     NONE.
2884 ;
2885 ; CALLING SEQUENCE:
2886 ;
2887 ;     JSR     PC,WT25M      ; WAIT FOR 25 MILLISECONDS
2888 ;     OR JSR     PC,WT500   ; WAIT FOR 500 MICROSECONDS
2889 ;     OR JSR     PC,WT25    ; WAIT FOR 25 MICROSECONDS
2890 ;
2891 ; --
2892 ;
2893 011506 013700 011534 WT25M:: MOV     CNT25M,RO    ; GET 25 MILLISECOND WAIT COUNT
2894 011512 000405        BR          WAIT          ;
2895 ;
2896 011514 013700 011536 WT500:: MOV     CNT500,RO    ; GET 500 MICROSECOND WAIT COUNT
2897 011520 000402        BR          WAIT          ;
2898 ;
2899 011522 013700 011540 WT25::  MOV     CNT25,RO     ; GET 25 MICROSECOND WAIT COUNT
2900 ;
2901 011526 005300        WAIT:  DEC     RO          ; ALL DONE?
2902 011530 001376        BNE     WAIT        ; IF NOT, WAIT SOME MORE
2903 011532 000207        RTS     PC         ; ELSE RETURN

```

GLOBAL SUBROUTINES SECTION

2904

2905 011534 000000

2906 011536 000000

2907 011540 000000

2908

CNT25M::WORD 0

CNT500::WORD 0

CNT25::WORD 0

; COUNTER FOR 25 MILLISECOND DELAY

; COUNTER FOR 500 MICROSECOND DELAY

; COUNTER FOR 25 MICROSECOND DELAY

GLOBAL SUBROUTINES SECTION

```

2910 ; SUBROUTINE CRLF - ROUTINE TO PRINT CARRIAGE RETURN, LINE FEED.
2911
2912 ; **
2913 ; FUNCTIONAL DESCRIPTION:
2914 ;
2915 ;     PRINTS A CARRIAGE RETURN AND LINE FEED.
2916 ;
2917 ; INPUTS:
2918 ;
2919 ;     NONE.
2920 ;
2921 ; IMPLICIT INPUTS:
2922 ;
2923 ;     NONE.
2924 ;
2925 ; OUTPUTS:
2926 ;
2927 ;     A CARRIAGE RETURN AND LINE FEED ARE PRINTED.
2928 ;
2929 ; IMPLICIT OUTPUTS:
2930 ;
2931 ;     NONE.
2932 ;
2933 ; SUBORDINATE ROUTINES USED:
2934 ;
2935 ;     SUPERVISOR PRINTF MACRO.
2936 ;
2937 ; FUNCTIONAL SIDE EFFECTS:
2938 ;
2939 ;     NONE.
2940 ;
2941 ; CALLING SEQUENCE:
2942 ;
2943 ;     JSR     PC,CRLF
2944 ;
2945 ; --
2946
2947 011542 CRLF::
2948 011542 PRINTF  @LF
      011542 012746 011564 MOV @LF,-(SP)
      011546 012746 000001 MOV @1,-(SP)
      011552 010600 MOV SP,R0
      011554 104417 TRAP C$PRINTF
      011556 062706 000004 ADD @4,SP
2949 011562 000207 RTS PC
2950
2951 011564 045 116 000 LF: .ASCIZ /#N/
2952 .EVEN

```

GLOBAL SUBROUTINES SECTION

```

2954 ; SUBROUTINE WRDY - SUBROUTINE TO WAIT FOR OPERATOR READY
2955
2956 ;**
2957 ; FUNCTIONAL DESCRIPTION:
2958 ;
2959 ; THIS PRINTS A MESSAGE FOR THE OPERATOR TO TYPE 'CARRIAGE RETURN'
2960 ; TO CONTINUE. THE ROUTINE IS NORMALLY USED TO ALLOW A MESSAGE TO
2961 ; BE READ BEFORE PROCEEDING.
2962 ;
2963 ; IF MANUAL INTERVENTION IS NOT ALLOWED, THE ROUTINE DOES NOTHING.
2964 ;
2965 ; INPUTS:
2966 ;
2967 ; NONE.
2968 ;
2969 ; IMPLICIT INPUTS:
2970 ;
2971 ; NONE.
2972 ;
2973 ; OUTPUTS:
2974 ;
2975 ; 'TYPE 'CARRIAGE RETURN' TO CONTINUE OR 'CONRTOL C' TO ABORT.
2976 ;
2977 ; IMPLICIT OUTPUTS:
2978 ;
2979 ; NONE.
2980 ;
2981 ; SUBORDINATE ROUTINES USED:
2982 ;
2983 ; SUPERVISOR GMANID MACRO.
2984 ;
2985 ; FUNCTIONAL SIDE EFFECTS:
2986 ;
2987 ; NONE.
2988 ;
2989 ; CALLING SEQUENCE:
2990 ;
2991 ; JSR PC,WRDY
2992 ;
2993 ;--
2994
2995 011570
2996 011570 WRDY::
2997 011570 104450 MANUAL ; IS MANUAL INTERVENTION ALLOWED ?
2998 011572 103010 BNCOMPLETE 10$ ; IF NOT, EXIT TRAP C$MANI
2999 011574 GMANID WRDY1,WFLG,A,377,0,1,YES ; 'TYPE RETURN TO CONTINUE' BCC 10$
011574 104443 TRAP C$GMAN
011576 000406 BR 10000$
011600 011616 .WORD WFLG
011602 000152 .WORD T$CODE
011604 011620 .WORD WRDY1
011606 000377 .WORD 377
011610 000000 .WORD T$LOLIM
011612 000001 .WORD T$HILIM
011614
2999 011614 000207 10$: RTS PC 10000$:

```


GLOBAL SUBROUTINES SECTION

```
3000
3001 011616 000000          WFLG:  .WORD  0          ; FLAG FOR WARNING MESSAGE INPUT
3002
3003
3004 011620      124      131      120  WRDY1:  .NLIST  BEX
3005                                     .ASCIZ  /TYPE "CARRIAGE RETURN" TO CONTINUE OR "CONTROL C" TO ABORT./
3006                                     .LIST   BEX
                                     .EVEN
```

GLOBAL SUBROUTINES SECTION

```

3008      ; SUBROUTINE SELECT - TEST SELECT ROUTINE
3009
3010      ; **
3011      ; FUNCTIONAL DESCRIPTION:
3012      ;
3013      ;     THIS IS CALLED BY EACH TEST TO DECIDE WHETHER THE TEST SHOULD
3014      ;     BE RUN BASED ON THE DEVICE TYPE AND THE TEST MODE SELECTED IN
3015      ;     THE STARTUP QUESTIONS. A TEST SELECT MASK SUPPLIED BY THE TEST
3016      ;     IS COMPARED WITH A CONTROL MASK SET UP BY THE INITIALISATION
3017      ;     ROUTINE TO DECIDE WHETHER THE TEST IS RUN.
3018      ;
3019      ;     IF THE SOFTWARE FAULT INSERTION FLAG SFI IS SET, ALL NON-SPECIFIC
3020      ;     TESTS ARE SELECTED.
3021
3022      ; INPUTS:
3023      ;
3024      ;     THE LOCATION FOLLOWING THE SUBROUTINE CALL CONTAINS THE TEST
3025      ;     SELECT MASK IN THE FOLLOWING FORMAT :
3026      ;
3027      ;     BIT 0 IS SET IF DIGITAL INPUT MODULES ARE TO BE TESTED
3028      ;     BIT 1 IS SET IF DIGITAL OUTPUT MODULES ARE TO BE TESTED
3029      ;     BIT 2 IS SET IF ANALOGUE INPUT MODULES ARE TO BE TESTED
3030      ;     BIT 3 IS SET IF ANALOGUE OUTPUT MODULES ARE TO BE TESTED
3031      ;     BIT 4 IS SET TO FORCE TESTING OF MODULES WHICH ARE NOT
3032      ;     IDENTIFIED AS ANALOGUE OR DIGITAL
3033      ;     BITS 5, 6 AND 7 ARE UNUSED
3034      ;     BIT 8 IS SET TO INDICATE A BASIC INTERNAL LOGIC TEST
3035      ;     BIT 9 IS SET FOR FIELD INPUT/OUTPUT TESTS
3036      ;     BIT 10 IS SET FOR LOOPBACK TESTS
3037      ;     BIT 11 IS SET FOR ANALOGUE INPUT/OUTPUT TESTS USED BY
3038      ;     MANUFACTURING AND FIELD SERVICE
3039      ;     BIT 12 IS SET FOR SPECIFICALLY SELECTABLE TESTS
3040      ;     BITS 13, 14 AND 15 ARE UNUSED
3041
3042      ;     THE SECOND LOCATION FOLLOWING THE SUBROUTINE CALL CONTAINS THE
3043      ;     ADDRESS OF A TEST HEADER MESSAGE TO BE PRINTED IF THE TEST IS
3044      ;     SELECTED AND THE USER 'PNT' FLAG IS SELECTED.
3045
3046      ; IMPLICIT INPUTS:
3047      ;
3048      ;     CONMSK - TEST CONTROL MASK SET UP BY INIT CODE AT THE BEGINING
3049      ;     OF EACH SUBPASS. THE FORMAT IS AS FOLLOWS :
3050      ;
3051      ;     BIT 0 IS SET IF UUT IS DIGITAL INPUT
3052      ;     BIT 1 IS SET IF UUT IS DIGITAL OUTPUT
3053      ;     BIT 2 IS SET IF UUT IS ANALOGUE INPUT
3054      ;     BIT 3 IS SET IF UUT IS ANALOGUE OUTPUT
3055      ;     BIT 4 IS SET IF UUT IS NONE OF THE ABOVE
3056      ;     BITS 5, 6 AND 7 ARE UNUSED
3057      ;     BIT 8 IS ALWAYS SET TO SELECT BASIC INTERNAL
3058      ;     LOGIC TESTS
3059      ;     BIT 9 SET TO SELECT FIELD INPUT/OUTPUT TESTS
3060      ;     BIT 10 IS SET IF LOOPBACK TESTING IS SELECTED
3061      ;     AND ALLOWED FOR CURRENT UUT
3062      ;     BIT 11 IS SET IF MANUFACTURING HAVE SELECTED LOOPBACK
3063      ;     AND I/O TESTS
3064      ;     BIT 12 IS SET IF A SPECIFICALLY SELECTABLE TEST

```

GLOBAL SUBROUTINES SECTION

```

3065      :
3066      :           IS CHOSEN
3067      :           BITS 13, 14 AND 15 ARE UNUSED
3068      :
3069      :           SFI - IF THIS IS NON ZERO, ALL TESTS ARE SELECTED.
3070      :
3071      : OUTPUTS:
3072      :
3073      :           IF THE TEST MATCHES THE DEVICE TYPE AND TEST MODE CHOSEN VIA THE
3074      :           STARTUP QUESTIONS, THE ROUTINE EXITS WITH THE CARRY BIT CLEAR.
3075      :           IN THIS CASE, IF THE 'PNT' FLAG IS SELECTED, THE TEST HEADER IS
3076      :           PRINTED AND THE FLAGS TSTFLG AND TSUFLG ARE SET TO SHOW THAT A
3077      :           TEST HAS BEEN SELECTED.
3078      :
3079      :           IF THE TEST IS NOT SELECTED, THE CARRY BIT IS SET AND, IF THE
3080      :           'PNT' FLAG IS SELECTED, A MESSAGE 'TEST DISABLED BY STARTUP
3081      :           QUESTIONS' IS PRINTED.
3082      :
3083      :           IF THE TEST IS NOT SPECIFICALLY SELECTABLE (BIT 11 IS CLEAR IN
3084      :           THE TEST SELECT MASK), BIT 11 IS CLEARED IN THE CONTROL MASK TO
3085      :           PREVENT SUBSEQUENT SPECIFICALLY SELECTABLE TESTS FROM BEING RUN.
3086      :
3087      :           R0, R1 AND R2 ARE CORRUPTED.
3088      :
3089      : IMPLICIT OUTPUTS:
3090      :
3091      :           NONE.
3092      :
3093      : SUBORDINATE ROUTINES USED:
3094      :
3095      :           NONE.
3096      :
3097      : FUNCTIONAL SIDE EFFECTS:
3098      :
3099      :           NONE.
3100      :
3101      : CALLING SEQUENCE:
3102      :
3103      :           FIXED PARAMETERS FOLLOW THE SUBROUTINE CALL :
3104      :
3105      :           EG.      CALL      SELECT
3106      :                   777
3107      :
3108      :                   TNAME
3109      :                   BCS      TSTEND
3110      :                   : TEST SELECT MASK (BASIC TEST
3111      :                   : ON ALL DEVICE TYPES
3112      :                   : ADDRESS OF TEST HEADER
3113      :                   : BRANCH IF TEST NOT SELECTED
3113 011714
3114 011714 017601 000000
3115 011720 062716 000002
3116 011724 017602 000000
3117 011730 062716 000002
3118
3119 011734
3119 011734 104421
3120
3113      :
3114      : SELECT::
3115      : MOV      @ (SP),R1
3116      : ADD      @2,(SP)
3117      : MOV      @ (SP),R2
3118      : ADD      @2,(SP)
3119      :
3120      : RFLAGS  R0
3120      :                   : READ OPERATOR FLAGS INTO R0
3120      :                   TRAP  CSRFLA

```

GLOBAL SUBROUTINES SECTION

```

3121 011736 032701 010000          BIT    #10000,R1          ; SPECIFICALLY SELECTABLE TEST ?
3122 011742 001003                BNE    10$              ; IF YES, BRANCH
3123 011744 042737 010000 004132 10$: BIC    #10000,CONMSK    ; ELSE PREVENT SUCH TESTS
3124 011752 005737 004066          TST    SFI              ; SOFTWARE FAULT INSERTION ?
3125 011756 001007                BNE    20$              ; IF YES, SELECT THE TEST
3126 011760 130137 004132          BITB   R1,CONMSK        ; IS DEVICE THE CORRECT TYPE ?
3127 011764 001426                BEQ    30$              ; IF NOT, BRANCH
3128 011766 105001                CLRB   R1              ; IF YES, DISCARD LOW BYTE
3129 011770 030137 004132          BIT    R1,CONMSK        ; AND CHECK TEST TYPE
3130 011774 001426                BEQ    30$              ; IF WRONG, BRANCH
3131
3132          ; TEST IS SELECTED
3133
3134 011776 012737 000001 004054 20$: MOV    #1,TSTFLG          ; FLAG THAT TEST IS SELECTED
3135 012004 012737 000001 004056    MOV    #1,TSUFLG        ;
3136
3137 012012 032700 001000          BIT    #PNT,RO          ; PRINT TEST HEADER ?
3138 012016 001425                BEQ    50$              ; IF NOT, EXIT (CARRY IS CLEAR)
3139 012020                PRINTF R2              ; ELSE PRINT THE HEADER
          012020 010246
          012022 012746 000001          MOV    R2,-(SP)
          012026 010600                MOV    #1,-(SP)
          012030 104417                MOV    SP,RO
          012032 062706 000004          TRAP   C$PNTF
          3140 012036 000241                CLC                    ; CLEAR THE CARRY
          3141 012040 000414                BR     50$              ; AND EXIT
3142
3143          ; TEST IS NOT SELECTED
3144
3145 012042 032700 001000 30$:  BIT    #PNT,RO          ; PRINT TEST HEADER ?
3146 012046 001410                BEQ    40$              ; IF NOT, EXIT
3147 012050                PRINTF #TNA            ; ELSE PRINT 'NOT APPLICABLE'
          012050 012746 012074          MOV    #TNA,-(SP)
          012054 012746 000001          MOV    #1,-(SP)
          012060 010600                MOV    SP,RO
          012062 104417                TRAP   C$PNTF
          012064 062706 000004          ADD    #4,SP
3148 012070 000261 40$:  SEC                    ; SET THE CARRY BIT
3149
3150 012072 000207 50$:  RETURN                ; COMMON RETURN
3151
3152
3153 012074 045 101 040 TNA:  .NLIST BEX          TEST DISABLED BY STARTUP QUESTIONS#N/
3154                .ASCIZ /#A
3155                .LIST BEX
                .EVEN

```

GLOBAL SUBROUTINES SECTION

```

3157      ; SUBROUTINE INSERT - SUBROUTINE TO FORCE ERROR PRINTOUTS
3158
3159      ;**
3160      ; FUNCTIONAL DESCRIPTION:
3161      ;
3162      ;     THIS SUBROUTINE CAN BE USED TO FORCE PRINTOUT OF ERROR MESSAGES
3163      ;     FOR QUALITY CHECKING. IF THE FLAG 'SFI' IS NONE ZERO, THE BRANCH
3164      ;     INSTRUCTION FOLLOWING THE SUBROUTINE CALL IS SKIPPED OVER,
3165      ;     CAUSING THE ERROR MESSAGE TO BE PRINTED. IF 'SFI' IS SET TO 1,
3166      ;     THE ADDRESS OF THE SUBROUTINE CALL IS COMPARED WITH THAT OF THE
3167      ;     LAST CALL AND, IF IT HAS NOT CHANGED, THE MESSAGE IS NOT
3168      ;     PRINTED.
3169
3170      ; INPUTS:
3171      ;
3172      ;     SFI - IF ZERO, THE ROUTINE DOES NOTHING.
3173      ;           IF ONE, ERROR MESSAGES ARE PRINTED ONCE.
3174      ;           IF ANY OTHER VALUE, ERROR MESSAGES ARE ALWAYS PRINTED.
3175
3176      ; IMPLICIT INPUTS:
3177      ;
3178      ;     SEE CALLING SEQUENCE.
3179
3180      ; OUTPUTS:
3181      ;
3182      ;     LASTFA - IF THE ERROR MESSAGE IS TO BE PRINTED ONCE ONLY, LASTFA
3183      ;             IS LOADED WITH THIS SUBROUTINE RETURN ADDRESS.
3184
3185      ; IMPLICIT OUTPUTS:
3186      ;
3187      ;     NONE.
3188
3189      ; SUBORDINATE ROUTINES USED:
3190      ;
3191      ;     NONE.
3192
3193      ; FUNCTIONAL SIDE EFFECTS:
3194      ;
3195      ;     IF 'SFI' IS NONE ZERO, THE ROUTINE RETURN ADDRESS IS INCREMENTED
3196      ;     BY ONE WORD.
3197
3198      ; CALLING SEQUENCE:
3199      ;
3200      ;     A ONE WORD BRANCH INSTRUCTION MUST FOLLOW THE SUBROUTINE CALL
3201      ;     BEFORE THE ERROR PRINT CALL.
3202      ;
3203      ;     EG.      CMP      BAD,GOOD      ; REGISTER CORRECT ?
3204      ;             CALL     INSERT        ; SKIP BRANCH IF SFI FLAG SET
3205      ;             BEQ     10$           ; BRANCH IF REGISTER CORRECT
3206      ;             ERROR                    ; ELSE PRINT OUT ERROR MESSAGE
3207
3208      ;
3209
3210      ;
3211      ;
3212      ;
3213      ;
3210 012146      INSERT::
3211 012146 106746      MFPS      -(SP)          ; SAVE CONDITION CODES
3212 012150 023727 004066 000001      CMP      SFI,#1      ; INSERT FAULTS ?
3213 012156 103413      BLO      20$          ; IF NOT, BRANCH

```

GLOBAL SUBROUTINES SECTION

```

3214 012160 001007          BNE      10$          ; IF ALWAYS, BRANCH
3215 012162 026637 000002 012212    CMP      2(SP),LASTFA ; IS THIS FAULT ALREADY PRINTED?
3216 012170 001406          BEQ      20$          ; IF YES, EXIT
3217 012172 016637 000002 012212    MOV      2(SP),LASTFA ; ELSE, SAVE FAULT ADDRESS
3218 012200 062766 000002 000002 10$:  ADD      2,2(SP)      ; SKIP BRANCH TO FORCE PRINTOUT
3219 012206 106426          20$:  MTPS     (SP)+      ; RESTORE CONDITION CODES
3220 012210 000207          RETURN                    ; AND RETURN
3221
3222 012212 000000          LASTFA: .WORD 0      ; ADDRESS OF ROUTINE CALL
3223

```

GLOBAL SUBROUTINES SECTION

```

3225 ; SUBROUTINE CONFIG - ROUTINE TO FIND THE DEVICE CONFIGURATION AUTOMATICALLY.
3226
3227 ;**
3228 ; FUNCTIONAL DESCRIPTION:
3229 ;
3230 ; THIS ROUTINE SEARCHES THE IDV/IAV-11 ADDRESS RANGE (171000-171770) FOR A
3231 ; RESPONDING ADDRESS. FOR THE FIRST ADDRESS ON A 4 WORD BOUNDARY (171XX0)
3232 ; WHICH DOES NOT CAUSE AN NXM TRAP, AN ENTRY IS MADE IN THE HARDWARE
3233 ; P TABLE FOR UNIT L$LUN. FOR ADDRESSES OVER 171400, A VECTOR IS
3234 ; ASSUMED BASED ON THE LOW 8 BITS OF THE DEVICE ADDRESS.
3235 ;
3236 ; EACH CALL TO THE ROUTINE WILL CAUSE ONE MORE UNIT TO BE SEARCHED FOR.
3237 ; IF THE UNIT FOUND IS THE LAST IN THE IDV/IAV-11 RANGE, L$UNITS IS SET TO
3238 ; THE TOTAL NUMBER OF UNITS. IF NO UNIT IS FOUND, THE PROGRAM EXITS
3239 ; WITH THE CARRY BIT SET AND L$UNIT SET TO L$LUN.
3240 ;
3241 ; MODULES WITH MODE REGISTER CONTENTS NOT CORRESPONDING TO AN IDV/IAV11
3242 ; ARE SET UP, BUT CAUSE AN 'UNIDENTIFIED MODULE' MESSAGE TO BE OUTPUT.
3243 ;
3244 ; INPUTS:
3245 ;
3246 ; L$LUN - NUMBER FOR THE NEXT UNIT FOUND.
3247 ;
3248 ; IMPLICIT INPUTS:
3249 ;
3250 ; STADD MUST BE SET TO 171000 PRIOR TO THE FIRST CALL TO INITIALISE
3251 ; THE SEARCH AREA.
3252 ;
3253 ; ADDRESSES DEFINED BY "IXSTA" TO "IXEND" OF THE I/O PAGE ARE READ.
3254 ;
3255 ; THE WORD L$LUN*2 AFTER LABEL GPADD MUST CONTAIN THE PARAMETER
3256 ; TABLE ADDRESS FOR THE UNIT.
3257 ;
3258 ; OUTPUTS:
3259 ;
3260 ; THE HARDWARE P TABLE AND L$UNIT ARE SET UP TO INCLUDE UP TO 16
3261 ; UNITS FOUND IN THE ABOVE ADDRESS RANGE.
3262 ;
3263 ; A MESSAGE 'UNIDENTIFIED MODULE FOUND AT ADDRESS NNNNNN' MAY BE
3264 ; PRINTED.
3265 ;
3266 ; IMPLICIT OUTPUTS:
3267 ;
3268 ; IF THE UNIT FOUND IS THE LAST IN THE IDV/IAV-11 RANGE, L$UNIT IS SET UP
3269 ; TO CONTAIN THE TOTAL NUMBER OF UNITS FOUND (L$LUN+1).
3270 ;
3271 ; IF NO MORE UNITS ARE FOUND, THE CARRY BIT IS SET AND L$LUN IS
3272 ; PLACED IN L$UNIT.
3273 ;
3274 ; SUBORDINATE ROUTINES USED:
3275 ;
3276 ; NXM - NON EXISTANT MEMORY TRAP ROUTINE.
3277 ; WRDY - ROUTINE TO WAIT FOR OPERATOR TO TYPE 'RETURN'.
3278 ;
3279 ; FUNCTIONAL SIDE EFFECTS:
3280 ;
3281 ; NXMFLG MAY BE SET.

```

GLOBAL SUBROUTINES SECTION

```

3282
3283      ; CALLING SEQUENCE:
3284      ;
3285      ; EG. JSR    PC,CONFIG
3286      ;         BCS    INIUUT
3287      ;
3288      ; --
3289
3290      CONFIG::
3291      012214 010146      MOV    R1,-(SP)      ; SAVE R1 TO R4
3292      012216 010246      MOV    R2,-(SP)
3293      012220 010346      MOV    R3,-(SP)
3294      012222 010446      MOV    R4,-(SP)
3295
3296      012224 013702 012514      MOV    STADD,R2      ; START SEARCH FROM THIS ADDRESS
3297      012230 012737 000001 012516      MOV    #1,NOUNIT    ; ASSUME NO UNIT WILL BE FOUND
3298      012236      SETVEC  #4,#NXM,#340    ; SET UP NXM TRAP
3299
3300      012264 005037 004060      10$:   CLR    NXMFLG      ; CLEAR THE NXM FLAG
3301      012270 005712      TST    (R2)        ; CHECK THE ADDRESS
3302      012272 005737 004060      TST    NXMFLG      ; ANYTHING THERE ?
3303      012276 001006      BNE    20$        ; IF NOT, TRY NEXT DEVICE ADDRESS
3304      012300 005737 012516      TST    NOUNIT      ; IF YES, IS IT THE 2ND ONE THIS CALL ?
3305      012304 001422      BEQ    30$        ; IF YES, SET UP THE P TABLE
3306      012306 005037 012516      CLR    NOUNIT      ; IF IT'S THE FIRST, FLAG UNIT FOUND
3307      012312 010201      MOV    R2,R1      ; AND SAVE THE ADDRESS
3308
3309      012314 062702 000010      20$:   ADD    #10,R2      ; GET THE NEXT ADDRESS
3310      012320 020227 171770      CMP    R2,#IXEND    ; OUT OF THE IXV11 RANGE ?
3311      012324 003757      BLE    10$        ; IF NOT, GO BACK
3312      012326 012702 171000      MOV    #IXSTA,R2    ; ELSE START AGAIN NEXT CALL
3313      012332 013737 002074 002012      MOV    L$LUN,L$UNIT ; SAVE THE UNIT NUMBER
3314      012340 005737 012516      TST    NOUNIT      ; WERE ANY UNITS FOUND ?
3315      012344 001047      BNE    70$        ; IF NOT, EXIT
3316      012346 005237 002012      INC    L$UNIT      ; ELSE WE HAVE THE LAST UNIT
3317
3318      012352 021127 140000      30$:   CMP    (R1),#140000 ; IS MODULE ID OK ?
3319      012356 103413      BLO    40$        ; IF YES, BRANCH
3320      012360      PRINTF #C01,R1      ; ELSE PRINT 'UNIDENTIFIED'
3321
3322      012360 010146      MOV    R1,-(SP)
3323      012362 012746 012520      MOV    #C01,-(SP)
3324      012366 012746 000002      MOV    #2,-(SP)
3325      012372 010600      MOV    SP,R0
3326      012374 104417      TRAP  C$PNTF
3327      012376 062706 000006      ADD    #6,SP
3328      012402 004737 011570      JSR    PC,WRDY     ; WAIT FOR OPERATOR TO TYPE 'RETURN'
3329
3330      012406 013700 002074      40$:   MOV    L$LUN,R0    ; FORM OFFSET FOR LUN
3331      012412 006300      ASL    R0
3332      012414 016003 003774      MOV    GPADD(R0),R3 ; GET P TABLE ADDRESS IN R3
3333      012420 010123      MOV    R1,(R3)    ; SAVE THE DEVICE ADDRESS

```


GLOBAL SUBROUTINES SECTION

```

3327
3328 012422 042701 177000      BIC    #177000,R1      ; CALCULATE THE VECTOR ADDRESS
3329 012426 032701 000400      BIT    #400,R1        ; ADDRESS OVER 171400 ?
3330 012432 001001              BNE    50#            ; IF YES, BRANCH
3331 012434 005001              CLR    R1             ; ELSE ZERO THE VECTOR
3332 012436 010123      50#:  MOV    R1,(R3).      ; AND SAVE IT
3333 012440 001402              BEQ    60#            ; IF 0, BRANCH
3334 012442 012701 000200      MOV    #200,R1        ; ELSE SET UP DEFAULT PRIORITY
3335 012446 010123      60#:  MOV    R1,(R3).      ; SAVE THE PRIORITY
3336 012450 005023              CLR    (R3).          ; FLAG NOT LOOPED
3337 012452 005023              CLR    (R3).          ; NO LOW LEVEL
3338 012454 012723 000003      MOV    #3,(R3).       ; ALL DEBOUNCE PERIODS
3339 012460 005023              CLR    (R3).          ; NO OTHER DEVICE
3340 012462 005023              CLR    (R3).          ; OR VECTOR
3341
3342 012464 010237 012514      70#:  MOV    R2,STADD   ; SAVE THE NEXT SEARCH ADDRESS
3343 012470      CLRVEC #4          ; RESTORE THE NXM TRAP CATCHER
      012470 012700 000004      MOV    #4,R0         ;
      012474 104436              TRAP C:VEC          ;
3344 012476 012604      MOV    (SP),R4        ; RESTORE R4 TO R1
3345 012500 012603      MOV    (SP),R3        ;
3346 012502 012602      MOV    (SP),R2        ;
3347 012504 012601      MOV    (SP),R1        ;
3348 012506 006237 012516      ASR    NOUNIT         ; IF NO UNIT, SET THE CARRY BIT
3349 012512 000207      RTS    PC             ; AND RETURN
3350
3351 012514 171000      STADD:: .WORD IXSTA   ; START ADDRESS OF SEARCH AREA
3352 012516 000000      NOUNIT: .WORD 0      ; SET TO SHOW NO UNIT FOUND
3353
3354
3355 012520 045 116 045 C01:  .NLIST BEX
      .ASCIZ /#UNIDENTIFIED MODULE FOUND AT ADDRESS #06#A./
3356      .LIST BEX
3357      .EVEN

```

GLOBAL SUBROUTINES SECTION

```

3359 ; SUBROUTINE CONPRI - SUBROUTINE TO PRINT A CONFIGURATION TABLE.
3360
3361 ;**
3362 ; FUNCTIONAL DESCRIPTION:
3363 ;
3364 ; THIS ROUTINE PRINTS A CONFIGURATION TABLE OF ALL UNITS LISTED IN
3365 ; THE HARDWARE PARAMETER TABLE. LISTED ARE THE MODE REGISTER ADDRESS,
3366 ; VECTOR, DEVICE TYPE, WHETHER LOOPED AND IF SO, THE ADDRESS AND VECTOR
3367 ; OF THE DEVICE TO WHICH IT IS LOOPED.
3368
3369 ; INPUTS:
3370 ;
3371 ; NONE.
3372
3373 ; IMPLICIT INPUTS:
3374 ;
3375 ; THE HARDWARE P TABLE.
3376 ;
3377 ; IF NO UNITS ARE CONFIGURED, THE GLOBAL MESSAGE LABELLED "NODEV"
3378 ; IS PRINTED.
3379
3380 ; OUTPUTS:
3381 ;
3382 ; THE CONFIGURATION TABLE IS PRINTED.
3383
3384 ; IMPLICIT OUTPUTS:
3385 ;
3386 ; IF NO UNITS ARE CONFIGURED (L$UNIT = 0), A MESSAGE 'NO DEVICES
3387 ; FOUND' IS PRINTED.
3388
3389 ; SUBORDINATE ROUTINES USED:
3390 ;
3391 ; NXM - NON EXISTANT MEMORY TRAP ROUTINE.
3392 ; CRLF - LINE FEED PRINT ROUTINE.
3393
3394 ; FUNCTIONAL SIDE EFFECTS:
3395 ;
3396 ; REGISTERS R1 TO R5 ARE CORRUPTED.
3397
3398 ; CALLING SEQUENCE:
3399 ;
3400 ; JSR PC,CONPRI
3401 ;
3402 ; --
3403
3404 012600 CONPRI::
3405 012600 SETVEC 04,0NXM,0340 ; CATCH REFERENCES TO NXM
3406 012600 012746 000340 MOV 0340,-(SP)
3407 012604 012746 017266 MOV 0NXM,-(SP)
3408 012610 012746 000004 MOV 04,-(SP)
3409 012614 012746 000003 MOV 03,-(SP)
3410 012620 104437 TRAP C$SVEC
3411 012622 062706 000010 ADD 010,SP
3412 3406 012626 005737 002012 TST L$UNIT ; ANY UNITS CONFIGURED ?
3413 3407 012632 001012 BNE 10, ; IF YES, BRANCH
3414 3408 012634 PRINTF 0NODEV ; ELSE PRINT 'NO DEVICES FOUND'
3415 012634 012746 007250 MOV 0NODEV,-(SP)

```

GLOBAL SUBROUTINES SECTION

012640	012746	000001				MOV	#1,-(SP)
012644	010600					MOV	SP,R0
012646	104417					TRAP	C\$PNTF
012650	062706	000004				ADD	#4,SP
3409 012654	000137	013704		JMP	CONEX		; AND EXIT
3410							
3411 012660			10\$:	PRINTF	#CP1		; PRINT CONFIGURATION HEADER
012660	012746	013720				MOV	#CP1,-(SP)
012664	012746	000001				MOV	#1,-(SP)
012670	010600					MOV	SP,R0
012672	104417					TRAP	C\$PNTF
012674	062706	000004				ADD	#4,SP
3412 012700				PRINTF	#CP1A		
012700	012746	013771				MOV	#CP1A,-(SP)
012704	012746	000001				MOV	#1,-(SP)
012710	010600					MOV	SP,R0
012712	104417					TRAP	C\$PNTF
012714	062706	000004				ADD	#4,SP
3413 012720				PRINTF	#CP1B		
012720	012746	014043				MOV	#CP1B,-(SP)
012724	012746	000001				MOV	#1,-(SP)
012730	010600					MOV	SP,R0
012732	104417					TRAP	C\$PNTF
012734	062706	000004				ADD	#4,SP
3414 012740				PRINTF	#CP1C		
012740	012746	014146				MOV	#CP1C,-(SP)
012744	012746	000001				MOV	#1,-(SP)
012750	010600					MOV	SP,R0
012752	104417					TRAP	C\$PNTF
012754	062706	000004				ADD	#4,SP
3415 012760	005001						
3416 012762	010100		20\$:	CLR	R1		; START WITH FIRST UNIT
3417 012764	006300			MOV	R1,R0		; FORM PARAMETER TABLE OFFSET
3418 012766	016002	003774		ASL	R0		
3419 012772	012203			MOV	GPADD(R0),R2		; GET THE UNIT P TABLE
3420 012774				MOV	(R2),R3		; SAVE THE UNIT ADDRESS
012774	010346			PRINTF	#CP2,R1,R3		; PRINT UNIT NO. AND ADDRESS
012776	010146					MOV	R3,-(SP)
013000	012746	014217				MOV	R1,-(SP)
013004	012746	000003				MOV	#CP2,-(SP)
013010	010600					MOV	#3,-(SP)
013012	104417					MOV	SP,R0
013014	062706	000010				TRAP	C\$PNTF
013014	062706	000010				ADD	#10,SP
3421 013020	005722			TST	(R2),		; IS THERE A VALID VECTOR ?
3422 013022	001413			BEQ	30\$; IF NOT, BRANCH
3423 013024				PRINTF	#CP3,-2(R2)		; ELSE PRINT THE VECTOR
013024	016246	177776				MOV	-2(R2),-(SP)
013030	012746	014230				MOV	#CP3,-(SP)
013034	012746	000002				MOV	#2,-(SP)
013040	010600					MOV	SP,R0
013042	104417					TRAP	C\$PNTF
013044	062706	000006				ADD	#6,SP
3424 013050	000410			BR	40\$		
3425 013052			30\$:	PRINTF	#CP4		; PRINT 'NONE'
013052	012746	014237				MOV	#CP4,-(SP)
013056	012746	000001				MOV	#1,-(SP)
013062	010600					MOV	SP,R0

GLOBAL SUBROUTINES SECTION

```

013064 104417
013066 062706 000004 TRAP C$PNTF
3426 ADD #4,SP
3427 013072 005037 004060 40$: CLR NXMFLG ; GET READY FOR NXM TRAP
3428 013076 112305 MOV (R3)+,R5 ; GET MODE REGISTER LOW BYTE
3429 013100 111304 MOV (R3),R4 ; AND HIGH BYTE
3430 013102 042704 177400 BIC #177400,R4 ; DISCARD REGISTER HIGH BYTE
3431 013106 005737 004060 TST NXMFLG ; WAS THERE AN NXM TRAP ?
3432 013112 001421 BEQ 50$ ; IF NOT, BRANCH
3433 013114 PRINTF #CP10 ; ELSE PRINT UNKNOWN ID/MODE
013114 012746 014400 MOV #CP10,-(SP)
013120 012746 000001 MOV #1,-(SP)
013124 010600 MOV SP,R0
013126 104417 TRAP C$PNTF
013130 062706 000004 ADD #4,SP
3434 013134 PRINTF #CP10A ; AND UNKNOWN MODULE TYPE
013134 012746 014415 MOV #CP10A,-(SP)
013140 012746 000001 MOV #1,-(SP)
013144 010600 MOV SP,R0
013146 104417 TRAP C$PNTF
013150 062706 000004 ADD #4,SP
3435 013154 000464 BR 110$ ; AND SEE IF LOOPED
3436
3437 013156 50$: PRINTF #CP5,R4,R5 ; PRINT OUT ID/MODE
013156 010546 MOV R5,-(SP)
013160 010446 MOV R4,-(SP)
013162 012746 014251 MOV #CP5,-(SP)
013166 012746 000003 MOV #3,-(SP)
013172 010600 MOV SP,R0
013174 104417 TRAP C$PNTF
013176 062706 000010 ADD #10,SP
3438
3439 013202 020427 000037 CMP R4,#37 ; IS MODULE DIGITAL INPUT ?
3440 013206 101003 BHI 60$ ; IF NOT, BRANCH
3441 013210 012703 014446 MOV #DI,R3 ; ELSE SAVE 'DIG. IN' STRING
3442 013214 000421 BR 90$ ; GO TO PRINT MODULE TYPE
3443 013216 020427 000077 60$: CMP R4,#77 ; IS MODULE DIGITAL OUTPUT ?
3444 013222 101003 BHI 70$ ; IF NOT, BRANCH
3445 013224 012703 014457 MOV #DO,R3 ; ELSE SAVE 'DIG OUT' STRING
3446 013230 000413 BR 90$ ; GO TO PRINT MODULE TYPE
3447 013232 020427 000177 70$: CMP R4,#177 ; IS MODULE ANALOGUE INPUT ?
3448 013236 101003 BHI 80$ ; IF NOT, BRANCH
3449 013240 012703 014470 MOV #AI,R3 ; ELSE SAVE 'AN IN' STRING
3450 013244 000405 BR 90$ ; GO TO PRINT MODULE TYPE
3451 013246 020427 000277 80$: CMP R4,#277 ; IS MODULE ANALOGUE OUTPUT ?
3452 013252 101014 BHI 100$ ; IF NOT, BRANCH
3453 013254 012703 014501 MOV #AO,R3 ; ELSE SAVE 'AN OUT' STRING
3454 013260 90$: PRINTF #CP6,R3 ; PRINT OUT DEVICE TYPE
013260 010346 MOV R3,-(SP)
013262 012746 014266 MOV #CP6,-(SP)
013266 012746 000002 MOV #2,-(SP)
013272 010600 MOV SP,R0
013274 104417 TRAP C$PNTF
013276 062706 000006 ADD #6,SP
3455 013302 000411 BR 110$ ;
3456 013304 100$: PRINTF #CP7 ; PRINT 'CANNOT BE TESTED'
013304 012746 014274 MOV #CP7,-(SP)

```

GLOBAL SUBROUTINES SECTION

```

013310 012746 000001
013314 010600
013316 104417
013320 062706 000004
3457 013324 000561 BR 210$ ; GO TO NEXT DEVICE
3458
3459 013326 005722 110$: TST (R2)+ ; SKIP THE PRIORITY
3460 013330 005722 TST (R2)+ ; IS DEVICE LOOPED ?
3461 013332 001012 BNE 120$ ; IF YES, BRANCH
3462 013334 PRINTF #CP9 ; ELSE PRINT 'NO'
013334 012746 014367
013340 012746 000001
013344 010600
013346 104417
013350 062706 000004
3463 013354 000137 013670 JMP 210$ ; AND GO TO NEXT DEVICE
3464
3465 013360 120$: PRINTF #CP8 ; PRINT 'YES'
013360 012746 014356
013364 012746 000001
013370 010600
013372 104417
013374 062706 000004
3466
3467 013400 005037 004060 CLR NXMFLG ; GET READY FOR NXM TRAP
3468 013404 005722 TST (R2)+ ; SKIP LOW LEVEL QUESTION
3469 013406 005722 TST (R2)+ ; SKIP DBOUNCE QUESTION
3470 013410 012203 MOV (R2)+,R3 ; GET OTHER MODULE ADDRESS
3471 013412 105723 TSTB (R3)+ ; GET ID CODE
3472 013414 111304 MOVB (R3),R4 ; INTO R4
3473 013416 042704 177400 BIC #177400,R4 ; DISCARD REGISTER TOP BYTE
3474 013422 005737 004060 TST NXMFLG ; WAS THERE AN NXM TRAP ?
3475 013426 001411 BEQ 130$ ; IF NOT, BRANCH
3476 013430 PRINTF #CP10A ; ELSE PRINT UNKNOWN OTHER MODULE TYPE
013430 012746 014415
013434 012746 000001
013440 010600
013442 104417
013444 062706 000004
3477 013450 000451 BR 190$ ;
3478
3479 013452 020427 000037 130$: CMP R4,#37 ; IS MODULE DIGITAL INPUT ?
3480 013456 101003 BHI 140$ ; IF NOT, BRANCH
3481 013460 012703 014446 MOV #DI,R3 ; ELSE SAVE 'DIG. IN' STRING
3482 013464 000421 BR 170$ ; GO TO PRINT MODULE TYPE
3483 013466 020427 000077 140$: CMP R4,#77 ; IS MODULE DIGITAL OUTPUT ?
3484 013472 101003 BHI 150$ ; IF NOT, BRANCH
3485 013474 012703 014457 MOV #DO,R3 ; ELSE SAVE 'DIG OUT' STRING
3486 013500 000413 BR 170$ ; GO TO PRINT MODULE TYPE
3487 013502 020427 000177 150$: CMP R4,#177 ; IS MODULE ANALOGUE INPUT ?
3488 013506 101003 BHI 160$ ; IF NOT, BRANCH
3489 013510 012703 014470 MOV #AI,R3 ; ELSE SAVE 'AN IN' STRING
3490 013514 000405 BR 170$ ; GO TO PRINT MODULE TYPE
3491 013516 020427 000277 160$: CMP R4,#277 ; IS MODULE ANALOGUE OUTPUT ?
3492 013522 101014 BHI 180$ ; IF NOT, BRANCH
3493 013524 012703 014501 MOV #AO,R3 ; ELSE SAVE 'AN OUT' STRING
3494 013530 170$: PRINTF #CP6,R3 ; PRINT OUT DEVICE TYPE

```

GLOBAL SUBROUTINES SECTION

```

013530 010346
013532 012746 014266
013536 012746 000002
013542 010600
013544 104417
013546 062706 000006
3495 013552 000410
3496 013554 180$: BR 190$
PRINTF #CP10 ; PRINT 'UNKNOWN'
013554 012746 014400
013560 012746 000001
013564 010600
013566 104417
013570 062706 000004
3497
3498 013574 190$: PRINTF #CP11,-2(R2) ; PRINT OTHER MODULE ADDRESS
013574 016246 177776
013600 012746 014433
013604 012746 000002
013610 010600
013612 104417
013614 062706 000006
3499 013620 005712
3500 013622 001011
3501 013624 TST (R2) ; IS THERE A VALID VECTOR ?
BNE 200$ ; IF YES, BRANCH
PRINTF #CP4 ; ELSE PRINT 'NONE'
013624 012746 014237
013630 012746 000001
013634 010600
013636 104417
013640 062706 000004
3502 013644 000411
3503 013646 200$: BR 210$
PRINTF #CP12,(R2) ; AND GO TO NEXT UNIT
; PRINT OTHER VECTOR
013646 011246
013650 012746 014437
013654 012746 000002
013660 010600
013662 104417
013664 062706 000006
3504
3505 013670 005201
3506 013672 020137 002012
3507 013676 002002
3508 013700 000137 012762
3509 013704 004737 011542
3510 013710 CONEX: JSR PC,CRLF ; PRINT A LINE FEED
CLRVEC #4 ; RESTORE SUPERVISOR NXM TRAP CATCHER
013710 012700 000004
013714 104436
3511 013716 000207
3512
3513
3514
3515 013720 045 116 045 CP1: .ASCIZ \#N#S24#AIDV/IAV-11 MODULE CONFIGURATION.\
3516 013771 045 116 045 CP1A: .ASCIZ /#N#S24#A-----/
3517 014043 045 116 062 CP1B: .ASCII \#N2#AUNIT ADDRESS VECTOR ID/MODE MODULE LOOPED TO\
3518 014136 045 123 071 .ASCIZ \#S9#AAT\
3519 014146 045 101 040 CP1C: .ASCIZ \#A VECTOR#N#S15#AASSUMED#S13#ATYPE#N\
3520
3521 014217 045 116 045 CP2: .ASCIZ \#N#D3#09\

```

GLOBAL SUBROUTINES SECTION

3522	014230	045	123	065	CP3:	.ASCIZ	\S5#03\	
3523	014237	045	123	064	CP4:	.ASCIZ	\S4#ANONE\	
3524	014251	045	123	063	CP5:	.ASCIZ	\S3#03#A/#03\	
3525	014266	045	123	063	CP6:	.ASCIZ	\S3#T\	
3526	014274	045	101	040	CP7:	.ASCIZ	\#A *** CANNOT BE TESTED WITH THIS DIAGNOSTIC ***\	
3527	014356	045	101	040	CP8:	.ASCIZ	\#A YES\	
3528	014367	045	101	040	CP9:	.ASCIZ	\#A NO\	
3529	014400	045	101	040	CP10:	.ASCIZ	\#A UNKNOWN\	
3530	014415	045	101	040	CP10A:	.ASCIZ	\#A UNKNOWN \	
3531	014433	045	117	071	CP11:	.ASCIZ	\#09\	
3532	014437	045	123	065	CP12:	.ASCIZ	\S5#03\	
3533								
3534	014446	104	111	107	DI:	.ASCIZ	/DIG. IN /	
3535	014457	104	111	107	DO:	.ASCIZ	/DIG. OUT/	
3536	014470	101	116	056	AI:	.ASCIZ	/AN. IN /	
3537	014501	101	116	056	AO:	.ASCIZ	/AN. OUT/	
3538								
3539						.LIST	BEX	
3540						.EVEN		

GLOBAL SUBROUTINES SECTION

```

3542      ; SUBROUTINE DACON - DIGITAL TO ANALOGUE CONVERSION ROUTINE.
3543
3544      ;**
3545      ; FUNCTIONAL DESCRIPTION:
3546
3547      ;     THIS CONVERTS A 12 BIT DIGITAL PATTERN INTO A 2 WORD ANALOGUE
3548      ;     OUTPUT VALUE.
3549
3550      ; INPUTS:
3551
3552      ;     MODE : 0 = UNIPOLAR (0 TO 10 VOLTS)      BINARY CODED
3553      ;             1 = BIPOLAR (-10 TO +10 VOLTS)  OFFSET BINARY CODED
3554      ;             2 = 0 - 20 MA                    BINARY CODED
3555      ;             3 = 4 - 20 MA                    BINARY CODED
3556
3557      ;     GAIN : 0 - 7 = GAINS 1,2,5,10,20,50,100,200
3558
3559      ;     R1  : 12 BIT INPUT PATTERN.
3560
3561      ; IMPLICIT INPUTS:
3562      ;     VITAB AND ITAB : DIGITAL/ANALOGUE CONVERSION TABLES.
3563
3564      ; OUTPUTS:
3565
3566      ;     R1 - MILLIVOLTS (MODES 0 AND 1)
3567      ;           MICROAMPS (MODES 2 AND 3)
3568
3569      ;     R2 - MICROVOLTS (MODES 0 AND 1)
3570      ;           NANOAMPS (MODES 2 AND 3)
3571
3572      ; IMPLICIT OUTPUTS:
3573      ;     NONE.
3574
3575      ; SUBORDINATE ROUTINES USED:
3576      ;     NONE.
3577
3578      ; FUNCTIONAL SIDE EFFECTS:
3579      ;     NONE.
3580
3581      ; CALLING SEQUENCE:
3582      ;     NONE.
3583
3584      ;
3585      ;
3586      ;
3587      ; EG.  MOV    #1,MODE      ; BIPOLAR CONVERSION
3588      ;        MOV    #6,GAIN    ; GAIN = 100
3589      ;        MOV    #7777,R1   ; ALL BITS SET
3590      ;        JSR    PC,DACON
3591
3592      ;
3593      ;
3594      ;
3594 014512      DACON::
3595 014512      MOV    R3,-(SP)    ; SAVE R3
3596 014514      MOV    R4,-(SP)    ; AND R4
3597 014516      MOV    MODE,R3    ; GET MODE
3598 014522      TST    R3         ; MODE 0 ?
004134

```


GLOBAL SUBROUTINES SECTION

```

3599 014524 001003          BNE      10$
3600 014526 012700 004176  MOV      @VUPTAB,R0
3601 014532 000414          BR       PSDA
3602
3603 014534 005303          10$:    DEC      R3
3604 014536 001003          BNE      20$
3605 014540 012700 004776  MOV      @VBPTAB,R0
3606 014544 000407          BR       PSDA
3607
3608 014546 005303          20$:    DEC      R3
3609 014550 001003          BNE      30$
3610 014552 012700 005576  MOV      @IOTAB,R0
3611 014556 000402          BR       PSDA
3612
3613 014560 012700 006376  30$:    MOV      @I4TAB,R0
3614
3615 014564 013703 004136  PSDA:    MOV      GAIN,R3
3616 014570 005303          30$:    DEC      R3
3617 014572 100403          BMI      40$
3618 014574 062700 000060  ADD      @48.,R0
3619 014600 000773          BR       30$
3620
3621 014602 011004          40$:    MOV      (R0),R4
3622 014604 010103          MOV      R1,R3
3623 014606 005001          CLR      R1
3624 014610 005002          CLR      R2
3625 014612 006303          ASL      R3
3626 014614 006303          ASL      R3
3627 014616 006303          ASL      R3
3628 014620 006303          ASL      R3
3629
3630 014622 006303          50$:    ASL      R3
3631 014624 103011          BCC      70$
3632 014626 066002 000030  ADD      24.(R0),R2
3633 014632 020227 001750  CMP      R2,@1000.
3634 014636 002403          BLT      60$
3635 014640 162702 001750  SUB      @1000.,R2
3636 014644 005201          INC      R1
3637 014646 061001          60$:    ADD      (R0),R1
3638 014650 062700 000002  70$:    ADD      @2,R0
3639 014654 005703          TST      R3
3640 014656 001361          BNE      50$
3641
3642 014660 023727 004134 000001 80$:    CMP      MODE,#1
3643 014666 001007          BNE      90$
3644 014670 160401          SUB      R4,R1
3645 014672 002005          BGE      90$
3646 014674 005702          TST      R2
3647 014676 001403          BEQ      90$
3648 014700 162702 001750  SUB      @1000.,R2
3649 014704 005201          INC      R1
3650
3651 014706 023727 004134 000003 90$:    CMP      MODE,#3
3652 014714 001002          BNE      100$
3653 014716 062701 007640  ADD      @4000.,R1
3654
3655 014722 012604          100$:   MOV      (SP),R4

```

```

: IF NOT, BRANCH
: GET CONVERSION TABLE FOR MODE 0
:
: MODE 1 ?
: IF NOT, BRANCH
: GET CONVERSION TABLE FOR MODE 1
:
: MODE 2 ?
: IF NOT, BRANCH
: GET CONVERSION TABLE FOR MODE 2
:
: USE I4TAB FOR MODE 3
: GET THE GAIN
: HAVE WE GOT THE CORRECT TABLE ?
: IF YES, BRANCH
: ELSE INCREASE THE TABLE OFFSET
: AND TRY AGAIN
:
: SAVE HIGH BIT VALUE
: SAVE THE BIT PATTERN
: CLEAR THE OUTPUT REGISTERS
:
: SHIFT OUT UNUSED BITS
: ( BITS 15 - 12 )
:
: TEST A BIT
: IF CLEAR, BRANCH
: ELSE ADD IN LOW VALUE
: OVERFLOW OF LOW WORD ?
: IF NOT, BRANCH
: ELSE CARRY FROM LOW WORD
: TO HIGH WORD
: AND ADD IN HIGH VALUE
: GET NEXT TABLE ENTRY
: ALL BITS PROCESSED ?
: IF NOT, DO MORE BITS
: BIPOLAR VOLTAGE CONVERSION ?
: IF NOT, BRANCH
: ELSE MAKE BIPOLAR
: IF STILL POSITIVE, BRANCH
: DECIMAL PART ZERO ?
: IF YES, BRANCH
: ELSE BORROW FROM HIGH PART
:
: 4 - 20 MA MODE ?
: IF NOT, BRANCH
: ELSE ADD IN BASE VALUE
: RESTORE R4

```

GLOBAL SUBROUTINES SECTION

3656 014724 012603
3657 014726 000207

MOV (SP)+,R3 ; AND R3
RTS PC ;

GLOBAL SUBROUTINES SECTION

```

3659      ; SUBROUTINE ADCON - ANALOGUE TO DIGITAL CONVERSION ROUTINE.
3660
3661      ;**
3662      ; FUNCTIONAL DESCRIPTION:
3663      ;
3664      ;     THIS CONVERTS A 2 WORD ANALOGUE VALUE INTO A 12 BIT DIGITAL OUTPUT
3665      ;     PATTERN. THE INPUT IS ROUNDED UP OR DOWN TO THE NEAREST LSB VALUE.
3666      ;
3667      ; INPUTS:
3668      ;
3669      ;     MODE : 0 = UNIPOLAR (0 TO 10 VOLTS)      BINARY CODED
3670      ;           1 = BIPOLAR (-10 TO +10 VOLTS)  OFFSET BINARY CODED
3671      ;           2 = 0 - 20 MA                    BINARY CODED
3672      ;           3 = 4 - 20 MA                    BINARY CODED
3673      ;
3674      ;     GAIN : 0 - 7 = GAINS 1,2,5,10,20,50,100,200
3675      ;
3676      ;     R1 - MILLIVOLTS (MODES 0 AND 1)
3677      ;           MICROAMPS (MODES 2 AND 3)
3678      ;
3679      ;     R2 - MICROVOLTS (MODES 0 AND 1)
3680      ;           NANOAMPS (MODES 2 AND 3)
3681      ;
3682      ; IMPLICIT INPUTS:
3683      ;
3684      ;     VITAB AND ITAB : DIGITAL/ANALOGUE CONVERSION TABLES.
3685      ;
3686      ; OUTPUTS:
3687      ;
3688      ;     R1 : 12 BIT INPUT PATTERN.
3689      ;
3690      ; IMPLICIT OUTPUTS:
3691      ;
3692      ;     NONE.
3693      ;
3694      ; SUBORDINATE ROUTINES USED:
3695      ;
3696      ;     NONE.
3697      ;
3698      ; FUNCTIONAL SIDE EFFECTS:
3699      ;
3700      ;     NONE.
3701      ;
3702      ; CALLING SEQUENCE:
3703      ;
3704      ; EG.  MOV    #1,MODE      ; BIPOLAR CONVERSION
3705      ;       MOV    #6,GAIN    ; GAIN = 100
3706      ;       MOV    #-4,R1    ; -4.001 MILLIVOLTS
3707      ;       MOV    #-1,R2    ; IN R1/R2
3708      ;       JSR    PC,ADCON
3709      ;
3710      ; --
3711
3712 014730      ADCON::
3713 014730      010346      MOV    R3,-(SP)      ; SAVE R3
3714 014732      010446      MOV    R4,-(SP)      ; AND R4
3715 014734      013703      004134      MOV    MODE,R3      ; GET MODE

```

GLOBAL SUBROUTINES SECTION

3716	014740	005703			TST	R3		; MODE 0 ?
3717	014742	001003			BNE	10\$; IF NOT, BRANCH
3718	014744	012700	004176		MOV	#VUPTAB,R0		; GET CONVERSION TABLE FOR MODE 0
3719	014750	000416			BR	PSAD		
3720								
3721	014752	005303		10\$:	DEC	R3		; MODE 1 ?
3722	014754	001003			BNE	20\$; IF NOT, BRANCH
3723	014756	012700	004776		MOV	#VBPTAB,R0		; GET CONVERSION TABLE FOR MODE 1
3724	014762	000411			BR	PSAD		
3725								
3726	014764	005303		20\$:	DEC	R3		; MODE 2 ?
3727	014766	001003			BNE	30\$; IF NOT, BRANCH
3728	014770	012700	005576		MOV	#IOTAB,R0		; GET CONVERSION TABLE FOR MODE 2
3729	014774	000404			BR	PSAD		
3730								
3731	014776	012700	006376	30\$:	MOV	#I4TAB,R0		; USE I4TAB FOR MODE 3
3732	015002	162701	007640		SUB	#4000.,R1		; AND SUBTRACT BASE VALUE
3733								
3734	015006	013703	004136	PSAD:	MOV	GAIN,R3		; GET THE GAIN
3735	015012	005303		10\$:	DEC	R3		; HAVE WE GOT THE CORRECT TABLE ?
3736	015014	100403			BMI	20\$; IF YES, BRANCH
3737	015016	062700	000060		ADD	#48.,R0		; ELSE INCREASE THE TABLE OFFSET
3738	015022	000773			BR	10\$; AND TRY AGAIN
3739								
3740	015024	023727	004134	000001	20\$:	CMP	MODE,#1	; BIPOLAR MODE ?
3741	015032	001011			BNE	30\$; IF NOT, BRANCH
3742	015034	061001			ADD	(R0),R1		; ELSE CONVERT BIPOLAR TO UNIPOLAR
3743	015036	020127	023420		CMP	R1,#10000.		; WAS PREVIOUS VALUE NEGATIVE ?
3744	015042	001005			BNE	30\$; IF NOT, BRANCH
3745	015044	005702			TST	R2		; IS DECIMAL PART ZERO ?
3746	015046	001403			BEQ	30\$; IF YES, BRANCH
3747	015050	062702	001750		ADD	#1000.,R2		; ELSE BORROW FROM HIGH PART
3748	015054	005301			DEC	R1		
3749								
3750	015056	016003	000026	30\$:	MOV	22.(R0),R3		; GET ROUNDING VALUES FROM LOWEST
3751	015062	016004	000056		MOV	46.(R0),R4		; SIGNIFICANT BIT
3752	015066	006203			ASR	R3		; DIVIDE BY 2
3753	015070	103002			BCC	40\$; IF NO CARRY SKIP NEXT COMMAND
3754	015072	062704	001000		ADD	#1000,R4		; ADD CARRY
3755	015076	006204		40\$:	ASR	R4		; DIDIDE BY 2
3756	015100	060402			ADD	R4,R2		; ROUND UP THE INPUT VALUE
3757	015102	020227	001750		CMP	R2,#1000.		; LOWER PART IS MODULO 1000
3758	015106	002403			BLT	50\$		
3759	015110	162702	001750		SUB	#1000.,R2		; IF OVERFLOW, CARRY OVER TO
3760	015114	005201			INC	R1		; HIGH PART
3761	015116	060301		50\$:	ADD	R3,R1		; ADD IN HIGH PART OF ROUNDING FACTOR
3762	015120	012703	000020		MOV	#20,R3		; INITIALISE WORKING REGISTER
3763								
3764	015124	020110		60\$:	CMP	R1,(R0)		; COMPARE HIGH VALUE WITH TABLE ENTRY
3765	015126	002415			BLT	90\$; IF LESS, DON'T SET BIT
3766	015130	003003			BGT	70\$; IF MORE, SET THE BIT
3767	015132	020260	000030		CMP	R2,24.(R0)		; OTHERWISE, MUST CHECK THE LOW VALUE
3768	015136	002411			BLT	90\$; IF LESS, DON'T SET THE BIT
3769								
3770	015140	166002	000030	70\$:	SUB	24.(R0),R2		; SUBTRACT THE TABLE ENTRIES
3771	015144	002003			BGE	80\$; BRANCH IF NO BORROW NEEDED
3772	015146	062702	001750		ADD	#1000.,R2		; ELSE ADD TO LOW WORD

GLOBAL SUBROUTINES SECTION

```
3773 015152 005301
3774 015154 161001      80$: DEC R1 ; FROM HIGH WORD
3775 015156 052703 000001 SUB (R0),R1 ; AND LOW WORDS
3776 015162 062700 000002 BIS #1,R3 ; AND SET THE OUTPUT BIT
3777 015166 006303 90$: ADD #2,R0 ; AND NEXT TABLE ENTRY
3778 015170 103355 ASL R3 ; READY FOR NEXT BIT
3779 BCC 60$ ; IF 12 BITS NOT DONE, GO BACK
3780 015172 006203 100$: ASR R3 ; GET THE PATTERN AGAIN
3781 015174 010301 MOV R3,R1 ; SET UP OUTPUT REGISTER
3782 015176 012604 MOV (SP)+,R4 ; RESTORE R4
3783 015200 012603 MOV (SP)+,R3 ; AND R3
3784 015202 000207 RTS PC ;
```

GLOBAL SUBROUTINES SECTION

```

3786 ; SUBROUTINE DECIN - SIGNED DECIMAL INPUT ROUTINE.
3787
3788 ;**
3789 ; FUNCTIONAL DESCRIPTION:
3790 ;
3791 ; THIS SOLICITS A SIGNED DECIMAL NUMBER FROM THE OPERATOR.
3792 ;
3793 ; INPUTS:
3794 ;
3795 ; PADD - THE ADDRESS OF THE PROMPT STRING FOR THE INPUT.
3796 ;
3797 ; THE OPERATOR IS PROMPTED FOR A NUMBER WHICH CAN BE UP TO 10 DIGITS
3798 ; LONG INCLUDING AN OPTIONAL + OR - SIGN AND DECIMAL POINT. THE
3799 ; STRING SHOULD BE IN 'PRINT' FORMAT ( IE..ASCIZ /#N#AINPUT VALUE/ ).
3800 ;
3801 ; IMPLICIT INPUTS:
3802 ;
3803 ; NONE.
3804 ;
3805 ; OUTPUTS:
3806 ;
3807 ; R1 - INTEGER PART OF OPERATOR INPUT
3808 ; R2 - DECIMAL PART OF OPERATOR INPUT
3809 ;
3810 ;
3811 ; IMPLICIT OUTPUTS:
3812 ;
3813 ; ERROR MESSAGES ARE PRINTED IF THE OPERATOR TYPES AN ILLEGAL CHARACTER,
3814 ; AN INTEGER PART OVER 32767, OR A DECIMAL PART WITH MORE THAN 3 DIGITS.
3815 ;
3816 ; AN EXAMPLE STRING IS APPENDED TO THE INPUT PROMPT.
3817 ;
3818 ; SUBORDINATE ROUTINES USED:
3819 ;
3820 ; NONE.
3821 ;
3822 ; FUNCTIONAL SIDE EFFECTS:
3823 ;
3824 ; NONE.
3825 ;
3826 ; CALLING SEQUENCE:
3827 ;
3828 ; EG. MOV @MADD,PADD ; LOAD THE PROMPT MESSAGE ADDRESS
3829 ; JSR PC,DECIN
3830 ;
3831 ; --
3832
3833 015204 DECIN:: MOV @NR1,R0 ; CLEAR NUMBER AND STRING LOCATIONS
3834 015204 012700 015556 MOV @7,R1 ; 2 WORDS FOR INTEGER AND DECIMAL
3835 015210 012701 000007 10$: CLR (R0). ; PARTS AND 10 BYTES FOR THE INPUT
3836 015214 005020 DEC R1 ; STRING.
3837 015216 005301 BNE 10$ ;
3838 015220 001375 PRINTF PADD ; PRINT THE PROMPT
3839
3840 015222 MOV PADD,-(SP)
015222 013746 004140 MOV #1,-(SP)
015226 012746 000001

```

GLOBAL SUBROUTINES SECTION

```

015232 010600
015234 104417
015236 062706 000004
3841 015242 GMANID GETNUM,SNUM,A,-1,0,10.,NO ; GET THE NUMBER STRING
015242 104443
015244 000406
015246 015562
015250 000142
015252 015575
015254 177777
015256 000000
015260 000012
015262
3842 015262 012700 015562 MOV #SNUM,R0 ; POINT TO THE START
3843 015266 012701 015556 MOV #NR1,R1 ; ASSUME INTEGER PART FIRST
3844
3845 015272 121027 000053 CMPB (R0),#'. ; IS 1ST CHARACTER A . ?
3846 015276 001430 BEQ 60$ ; IF YES, BRANCH
3847 015300 121027 000055 CMPB (R0),#'- ; IS IT A - ?
3848 015304 001425 BEQ 60$ ; IF YES, BRANCH
3849 015306 121027 000056 20$: CMPB (R0),#'. ; IS IT A . ?
3850 015312 001017 BNE 50$ ; IF NOT, BRANCH
3851
3852 015314 012701 015560 30$: MOV #NR2,R1 ; START ON DECIMAL PART
3853 015320 105760 000002 TSTB 2(R0) ; FORCE TO 3 DIGITS
3854 015324 001003 BNE 40$ ;
3855 015326 112760 000060 000002 MOVB #'0,2(R0) ; IE. REPLACE NULLS
3856 015334 105760 000003 40$: TSTB 3(R0) ;
3857 015340 001007 BNE 60$ ; WITH ZEROS
3858 015342 112760 000060 000003 MOVB #'0,3(R0) ;
3859 015350 000403 BR 60$ ;
3860
3861 015352 105710 50$: TSTB (R0) ; END OF STRING ?
3862 015354 001451 BEQ 110$ ; IF YES, FINISH UP
3863 015356 000402 BR 70$ ; ELSE GET NEXT DIGIT
3864
3865 015360 005200 60$: INC R0 ; SKIP OVER THE SIGN OR POINT
3866 015362 000751 BR 20$ ;
3867
3868 015364 121027 000060 70$: CMPB (R0),#60 ; IS CHARACTER A VALID NUMBER ?
3869 015370 002403 BLT 80$ ; IF TOO LOW, ASK AGAIN
3870 015372 121027 000071 CMPB (R0),#71 ;
3871 015376 003411 BLE 90$ ; IF NOT TOO HIGH, BRANCH
3872
3873 015400 80$: PRINTF #DECIN3 ; PRINT 'ILLEGAL CHARACTER'
015400 012746 015734 MOV #DECIN3,-(SP)
015404 012746 000001 MOV #1,-(SP)
015410 010600 MOV SP,R0
015412 104417 TRAP C$PNTF
015414 062706 000004 ADD #4,SP
3874 015420 000671 BR DECIN ; AND ASK AGAIN
3875
3876 015422 021127 006314 90$: CMP (R1),#3276. ; NUMBER TOO HIGH ?
3877 015426 101013 BMI 100$ ; IF YES, BRANCH
3878
3879 015430 006311 ASL (R1) ; ELSE MULTIPLY BY 10
3880 015432 011102 MOV (R1),R2 ;

```

GLOBAL SUBROUTINES SECTION

SEQ 0094

```

3881 015434 006311          ASL      (R1)          ; READY FOR NEXT CHARACTER
3882 015436 006311          ASL      (R1)          ;
3883 015440 060211          ADD      R2,(R1)      ;
3884                                     ;
3885 015442 112002          MOVVB   (R0),R2       ; SAVE THE CHARACTER
3886 015444 162702 000060  SUB      #60,R2       ; CONVERT TO NUMBER
3887 015450 060211          ADD      R2,(R1)     ; AND ADD TO ACCUMULATOR
3888 015452 100401          BMI      100$       ; IF OVERFLOW, REPORT ERROR
3889                                     ;
3890 015454 000714          BR      20$         ; AND GET NEXT CHARACTER
3891                                     ;
3892 015456                                     ; PRINT 'NUMBER TOO BIG'
      015456 012746 015615 100$: PRINTF  #DECIN1
      015462 012746 000001                                     ;
      015466 010600                                     MOV    #DECIN1,-(SP)
      015470 104417                                     MOV    #1,-(SP)
      015472 062706 000004                                     MOV    SP,R0
      015476 000642                                     TRAP   C$PNTF
3893 015476 000642          BR      DECIN        ADD    #4,SP
3894                                     ; AND GET ANOTHER
3895 015500 013701 015556 110$: MOV     NR1,R1          ; SET UP OUTPUT REGISTERS
3896 015504 013702 015560  MOV     NR2,R2          ;
3897 015510 020227 001750  CMP     R2,#1000.     ; DECIMAL PART TOO BIG ?
3898 015514 103411          BLO    120$         ; IF NOT, BRANCH
3899 015516          PRINTF  #DECIN2          ; PRINT 'ONLY 3 DIGITS ALLOWED'
      015516 012746 015653                                     MOV    #DECIN2,-(SP)
      015522 012746 000001                                     MOV    #1,-(SP)
      015526 010600                                     MOV    SP,R0
      015530 104417                                     TRAP   C$PNTF
      015532 062706 000004                                     ADD    #4,SP
3900 015536 000622          BR      DECIN
3901                                     ;
3902 015540 123727 015562 000055 120$: CMPB   SNUM,#'-'      ; WAS STRING NEGATIVE ?
3903 015546 001002          BNE    130$         ; IF NOT, BRANCH
3904 015550 005401          NEG    R1           ; ELSE NEGATE THE OUTPUT
3905 015552 005402          NEG    R2           ;
3906                                     ;
3907 015554 000207          RTS     PC          ; AND RETURN
3908                                     ;
3909                                     .NLIST BEX
3910                                     ;
3911 015556 000000          NR1:   .WORD  0      ; STORE FOR INTEGER PART
3912 015560 000000          NR2:   .WORD  0      ; STORE FOR DECIMAL PART
3913 015562 055 061 062  SNUM:  .ASCIZ  /-12345.678/ ; STORE FOR INPUT STRING
3914                                     ;
3915 015575 050 105 107  GETNUM: .ASCIZ  /(EG. 12345.678)/ ; PROMPT FOR INPUT
3916 015615 045 116 045  DECIN1: .ASCIZ  /#N#AMUST BE LESS THAN 32768#N/
3917 015653 045 116 045  DECIN2: .ASCIZ  /#N#AONLY 3 DIGITS MAY FOLLOW THE DECIMAL POINT#N/
3918 015734 045 116 045  DECIN3: .ASCIZ  /#N#AILLEGAL CHARACTER#N/
3919                                     ;
3920                                     .LIST  BEX
3921                                     .EVEN

```


GLOBAL SUBROUTINES SECTION

```

3923 ; SUBROUTINE DECOUT - SIGNED DECIMAL OUTPUT ROUTINE
3924
3925 ;**
3926 ; FUNCTIONAL DESCRIPTION:
3927 ;
3928 ;     ROUTINE TO PRINT A SIGNED DECIMAL NUMBER.
3929 ;
3930 ; INPUTS:
3931 ;
3932 ;     R1 - INTEGER PART OF NUMBER TO BE PRINTED
3933 ;     R2 - DECIMAL PART OF NUMBER TO BE PRINTED
3934 ;
3935 ; IMPLICIT INPUTS:
3936 ;
3937 ;     NONE.
3938 ;
3939 ; OUTPUTS:
3940 ;
3941 ;     THE NUMBER IS PRINTED AS FOLLOWS : -12345.678
3942 ;     ( NO SIGN IS PRINTED FOR POSITIVE NUMBERS )
3943 ;     IF THE DECIMAL PART (R2) IS ZERO, THE NUMBER
3944 ;     IS PRINTED AS -12345
3945 ;
3946 ; IMPLICIT OUTPUTS:
3947 ;
3948 ;     NONE.
3949 ;
3950 ; SUBORDINATE ROUTINES USED:
3951 ;
3952 ;     NONE.
3953 ;
3954 ; FUNCTIONAL SIDE EFFECTS:
3955 ;
3956 ;     NONE.
3957 ;
3958 ; CALLING SEQUENCE:
3959 ;
3960 ;     EG. MOV     @-10.,R1           ; PRINT -10.001
3961 ;           MOV     @-1,R2           ;
3962 ;           JSR     PC,DECOUT
3963 ;
3964 ; --
3965
3966 015764 DECOUT::
3967 015764 010146     MOV     R1,-(SP)           ; SAVE R1
3968 015766 010246     MOV     R2,-(SP)           ; SAVE R2
3969
3970 015770 005701     TST     R1               ; R1 > 0 ?
3971 015772 002402     BLT     10$              ; IF NOT, BRANCH
3972 015774 005702     TST     R2               ; R2 POSITIVE ?
3973 015776 002012     BGE     20$              ; IF YES, BRANCH
3974 016000 005401 10$: NEG     R1               ; ELSE MAKE POSITIVE
3975 016002 005402     NEG     R2
3976 016004           PRINTF @DEC01           ; AND PRINT ...
           MOV     @DEC01,-(SP)
           MOV     @1,-(SP)
           MOV     SP,R0

```

GLOBAL SUBROUTINES SECTION

```

016016 104417
016020 062706 000004
3977
3978 016024 20$: PRINTF #DEC02,R1 ; PRINT THE INTEGER PART
016024 010146
016026 012746 016106
016032 012746 000002
016036 010600
016040 104417
016042 062706 000006
3979 016046 005702
3980 016050 001411
3981 016052
016052 010246
016054 012746 016112
016060 012746 000002
016064 010600
016066 104417
016070 062706 000006
3982 016074 012602
3983 016076 012601
3984 016100 000207
3985
3986
3987 016102 045 101 055 DEC01:
3988 016106 045 104 065 DEC02:
3989 016112 045 101 056 DEC03:
3990
3991
3992

TRAP C$PNTF
ADD #4,SP

MOV R1,-(SP)
MOV #DEC02,-(SP)
MOV #2,-(SP)
MOV SP,R0
TRAP C$PNTF
ADD #6,SP

TST R2 ; DECIMAL PART = 0 ?
BEQ DECEX ; IF YES BRANCH
PRINTF #DEC03,R2 ; ELSE, PRINT DECIMAL PART

MOV R2,-(SP)
MOV #DEC03,-(SP)
MOV #2,-(SP)
MOV SP,R0
TRAP C$PNTF
ADD #6,SP

DECEX: MOV (SP)+,R2 ; RESTORE R2
MOV (SP)+,R1 ; AND R1
RTS PC ; AND RETURN

.NLIST BEX
.ASCIZ /#A-/
.ASCIZ /#D5/
.ASCIZ /#A.#Z3/
.LIST BEX
.EVEN

```

GLOBAL SUBROUTINES SECTION

3994
3995
3996
3997
3998
3999
4000
4001
4002
4003
4004
4005
4006
4007
4008
4009
4010
4011
4012
4013
4014
4015
4016
4017
4018
4019
4020
4021
4022
4023
4024
4025
4026
4027

```

:*****
: SUBROUTINE WRITE - WRITE ROUTINE FOR OTHER MODULE ADDRESS
:-----
: FUNCTIONAL DESCRIPTION:
:
:   SUBROUTINE FOR LOADING THE DATA REGISTER OF A DIGITAL OUTPUT
:   MODULE. THE OUTPUT MODULE IS SPECIFIED FOR THE LOOPBACK TESTS
:   AS 'OTHER' IN THE HARDWARE P TABLE.
:   THIS SUBROUTINE ALSO DOES A CHECK TO THE OTHER MODULE
:   ADDRESS. IF THE ADDRESS IS NOT AVAILABLE IT WILL REPORT
:   AN ERROR.
:
: INPUTS:
:
:   THE CONTENTS OF R1 WILL BE USED TO LOAD THE DATA REGISTER
:
: IMPLICIT INPUTS:
:
:   ERRNBR - CONTAINS THE ERROR NUMBER
:
: OUTPUTS:
:
:   ERROR MESSAGES IF OUTPUT ADDRESS IS NOT AVAILABLE (NXM).
:
: SUBORDINATE ROUTINES USED:
:
:   DRS ERROR MACRO
:
: CALLING SEQUENCE:
:
:   CALL WRITE OR JSR PC,WRITE
:-----

```

```

4028 016122 010146 WRITE:: MOV R1,-(SP) ;SAVE R1
4029 016124 SETVEC #4,#NXM,#PRIO7 ;SET UP VECTOR
      016124 012746 000340 MOV #PRIO7,-(SP)
      016130 012746 017266 MOV #NXM,-(SP)
      016134 012746 000004 MOV #4,-(SP)
      016140 012746 000003 MOV #3,-(SP)
      016144 104437 TRAP C#SVEC
      016146 062706 000010 ADD #10,SP
4030 016152 005037 004060 CLR NXMFLG ;CLEAR NXM FLAG
4031 016156 010177 165542 MOV R1,@OTHDAT ;LOAD DIGITAL OUTPUT REGISTER
4032 016162 CLRVEC #4 ;VECTOR 4 TO NORMAL STATE
      016162 012700 000004 MOV #4,R0
      016166 104436 TRAP C#CVEC
4033 016170 005737 004060 TST NXMFLG ;WAS MODULE ADDRESS AVAILABLE
4034 016174 000241 CLC ;CLEAR CARRY
4035 016176 004737 012146 CALL INSERT ;SKIP BRANCH IF 'SFI' IS SET
4036 016202 001415 BEQ 10# ;BRANCH IF YES
4037 016204 013701 003722 MOV OTHMOD,R1 ;SET UP DATA FOR ERROR REPORT
4038 016210 012737 000003 007176 MOV #3,ERRTYP ;SET UP FOR SOFT ERROR
4039 016216 012737 016242 007202 MOV #WRERM,ERRMSG ;LOAD ERROR MESSAGES
4040 016224 012737 007634 007204 MOV #EERA,ERRBLK ;LOAD ERROR MESSAGES ROUTINE
4041 016232 ERROR ;ERROR HANDLER
      016232 104460 TRAP C#ERROR

```

GLOBAL SUBROUTINES SECTION

```

4042 016234 000261
4043 016236 012601
4044 016240 000207
4045
4046
4047 016242 122 105 107 WRERM: .NLIST BEX
4048 .ASCIZ /REGISTER ADDRESSING ERROR FOR OTHER MODULE ADDRESS/
4049 .LIST BEX
4050 .EVEN

```

```

10$: SEC
MOV (SP)+,R1 ;SET CARRY FOR MARK THE ERROR
RETURN ;RESTORE R1
;AND RETURN

```

GLOBAL SUBROUTINES SECTION

4052
4053
4054
4055
4056
4057
4058
4059
4060
4061
4062
4063
4064
4065
4066
4067
4068
4069
4070
4071
4072
4073
4074
4075
4076
4077
4078
4079
4080
4081
4082
4083
4084
4085
4086
4087
4088

```

:.....
: SUBROUTINE READ - READ ROUTINE FOR OTHER MODULE ADDRESS
:-----
:
: FUNCTIONAL DESCRIPTION:
:
:   SUBROUTINE FOR READING THE DATA REGISTER OF A DIGITAL INPUT
:   MODULE THAT IS SPECIFIED FOR THE LOOPBACK TESTS AS 'OTHER'
:   IN THE HARDWARE P TABLE.
:   THIS SUBROUTINE ALSO DOES A CHECK TO THE OTHER MODULE
:   ADDRESS. IF THE ADDRESS IS NOT AVAILABLE IT WILL REPORT
:   AN ERROR.
:
: INPUTS:
:
:   NONE.
:
: IMPLICIT INPUTS:
:
:   ERRNBR - CONTAINS ERROR NUMBER FOR ERROR HANDLER
:
: OUTPUTS:
:
:   ERROR MESSAGES IF INPUT ADDRESS IS NOT AVAILABLE (NXM).
:   THE LOCATION 'BAD' WILL CONTAIN THE CONTENTS OF THE
:   DATA REGISTER OF THE DIGITAL INPUT MODULE.
:
: SUBORDINATE ROUTINES USED:
:
:   DRS ERROR MACRO
:
: CALLING SEQUENCE:
:
:   CALL READ OR JSR PC,READ
:-----

```

```

4089 016326 010146
4090 016330
      016330 012746 000340
      016334 012746 017266
      016340 012746 000004
      016344 012746 000003
      016350 104437
      016352 062706 000010
4091 016356 005037 004060
4092 016362 017737 165336 004064
4093 016370
      016370 012700 000004
      016374 104436
4094 016376 005737 004060
4095 016402 000241
4096 016404 004737 012146
4097 016410 001415
4098 016412 013701 003722
4099 016416 012737 000003 007176
4100 016424 012737 016450 007202

READ:: MOV R1, -(SP) ;SAVE R1
      SETVEC #4, @NXM, @PRI07 ;SET UP VECTOR

      MOV @PRI07, -(SP)
      MOV @NXM, -(SP)
      MOV #4, -(SP)
      MOV #3, -(SP)
      TRAP C$SVEC
      ADD #10, SP

      CLR NXMFLG ;CLEAR NXM FLAG
      MOV @0THDAT, BAD ;READ DIGITAL INPUT REGISTER
      CLRVEC #4 ;VECTOR 4 TO NORMAL STATE

      MOV #4, R0
      TRAP C$CVEC

      TST NXMFLG ;WAS MODULE ADDRESS AVAILABLE
      CLC ;CLEAR CARRY
      CALL INSERT ;SKIP BRANCH IF 'SFI' IS SET
      BEQ 10$ ;BRANCH IF YES
      MOV 0THMOD, R1 ;SET UP DATA FOR ERROR REPORT
      MOV #3, ERRTP ;SET UP FOR SOFT ERROR
      MOV @REERM, ERRMSG ;LOAD ERROR MESSAGES

```


GLOBAL SUBROUTINES SECTION

```

4113
4114
4115
4116
4117
4118
4119
4120
4121
4122
4123
4124
4125
4126
4127
4128
4129
4130
4131
4132
4133
4134
4135
4136
4137
4138
4139
4140
4141
4142
4143
4144
4145
4146
4147 016534 013746 016572
4148 016540 013700 016574
4149 016544 006316
4150 016546 005500
4151 016550 006200
4152 016552 005516
4153 016554 061600
4154 016556 005600
4155 016560 012637 016572
4156 016564 010037 016574
4157 016570 000207
4158
4159 016572 135753
4160 016574 024674
4161

```

```

:.....
: SUBROUTINE - RANDOM
:-----
: FUNCTIONAL DESCRIPTION:
:
: THIS ROUTINE GENERATES A RANDOM PATTERN. THE PATTERN IS
: STORED IN LOCATION RB.
:
: INPUTS:
:
: NONE.
:
: IMPLICIT INPUTS:
:
: RA AND RB
:
: OUTPUTS:
:
: RB - CONTAINS THE RANDOM PATTERN
: RA - CONTAINS A SECOND RANDOM PATTERN
:
: IMPLICIT OUTPUTS:
:
: NONE.
:
: SUBORDINATE ROUTINES USED:
:
: NONE.
:
: CALLING SEQUENCE:
:
: CALL RANDOM OR JSR PC,RANDOM
:-----
RANDOM: MOV RA, -(SP) ; PUSH RA TO STACK
      MOV RB, R0 ; GET THE LAST RANDOM PATTERN
      ASL @SP ; SHIFT SP (=RA) LEFT
      ADC R0 ; IF CARRY IS SET ADD TO R0 (=RB)
      ASR R0 ; THEN SHIFT THE RESULT RITH
      ADC @SP ; IF CARRY IS SET ADD TO SP (=RA)
      ADD @SP, R0 ; ADD SP (=RA) AND R0 (=RB)
      SBC R0 ; SUBTRACT CARRY IF SET FROM RB
      MOV (SP)+, RA ; LOAD NEW VALUE INTO LOCATION RA
      MOV R0, RB ; LOAD LOCATION RB WITH NEW PAT.
      RETURN ;
RA:: .WORD 135753 ; START PATTERN FOR RB
RB:: .WORD 24674 ; STORAGE FOR RANDOM PATTERN

```

GLOBAL SUBROUTINES SECTION

4163
4164
4165
4166
4167
4168
4169
4170
4171
4172
4173
4174
4175
4176
4177
4178
4179
4180
4181
4182
4183
4184
4185
4186
4187
4188
4189
4190
4191
4192
4193
4194
4195
4196
4197
4198
4199
4200
4201

```

:.....
: SUBROUTINE - FLASH
:-----
: FUNCTIONAL DESCRIPTION:
:
: THIS SUBROUTINE FLASHES THE SELECTED MODULE
: LED ON AND OFF WHILE READING THE CONSOLE KEYBOARD INPUT.
: THE EXPECTED INPUTS ARE 'Y' OR 'N' OTHER INPUTS
: (EXCEPT THE CNTL C) WILL CAUSE A PROMT TO REPEAT
: WITH THE CORRECT CHARACTER.
:
: INPUTS:
:
: CONSOLE KEYBOARD INPUT.
:
: IMPLICIT INPUTS:
:
: NONE.
:
: OUTPUTS:
:
: FLSANS - CONTAINS THE KEYBOARD INPUT YES OR NO
: FLSANS ZERO = NO
: FLSANS ONE = YES
:
: IMPLICIT OUTPUTS:
:
: NONE.
:
: SUBORDINATE ROUTINES USED:
:
: BREAK - DRS MACRO (THIS MACRO LOOKS FOR CNTL C)
: PRINTF - DRS MACRO, THIS MACRO PRINTS A MESSAGES ON THE CONSOLE
:
: CALLING SEQUENCE:
:
: CALL FLASH OR JSR PC,FLASH

```

```

4202 016576 105737 177560 FLASH:: TSTB TKS ;TEST THE KEYBOARD STATUS REG.
4203 016602 100002 BPL 10$ ;BRANCH IF NOTHING FOUND
4204 016604 BREAK ;LOOK FOR 'CNTL C' TRAP C$BRK
016604 104422
4205 016606 000421 BR 20$ ;
4206 016610 042777 000100 165062 10$: BIC #100,@MOD ;SWITH MODULE LED OFF
4207 016616 012701 000012 40$: MOV #10.,R1 ;SET UP WAIT COUNTER
4208 016622 004737 011506 30$: CALL WT25M ;WAIT FOR 25 MS
4209 016626 005301 DEC R1 ;ARE 250 MS OVER
4210 016630 001374 BNE 30$ ;BRANCH IF NO
4211 016632 032777 000100 165040 BIT #100,@MOD ;IS THE MODULE LED SWITCHED ON
4212 016640 001356 BNE FLASH ;BRANCH IF YES
4213 016642 052777 000100 165030 BIS #100,@MOD ;OTHERWISE SWITCH IT ON
4214 016650 000762 BR 40$ ;AND BRANCH TO WAIT LOOP
4215 016652 012737 000001 017000 20$: MOV #1,FLSANS ;SAVE 'YES' ANSWER
4216 016660 013737 177562 004064 MOV TKB,BAD ;GET CHARACTER
4217 016666 042737 000200 004064 BIC #200,BAD ;DISCARD PARITY BIT
4218 016674 122737 000131 004064 CMPB #'Y,BAD ;WAS THE TYPED CHARACTER A 'Y' ?

```


GLOBAL SUBROUTINES SECTION

```

4219 016702 001432          BEQ      50$          ;BRANCH IF YES
4220 016704 122737 000131 004064  CMPB   #'Y,BAD      ;WAS IT A LOWESCASE 'Y' ?
4221 016712 001426          BEQ      50$          ;BRANCH IF YES
4222 016714 005037 017000    CLR     FLSANS      ;SAVE 'NO' ANSWER
4223 016720 122737 000116 004064  CMPB   #'N,BAD      ;WAS THE TYPED CHARACTER A 'N' ?
4224 016726 001420          BEQ      50$          ;BRANCH IF YES
4225 016730 122737 000116 004064  CMPB   #'N,BAD      ;WAS IT A LOWERCASE 'N' ?
4226 016736 001414          BEQ      50$          ;BRANCH IF YES
4227 016740 013737 177562 177566  MOV     TKB,TPB     ;ECHO THE CHARACTER
4228 016746          PRINTF  #PROMT      ;IF NO PRINT A PROMT
          016746 012746 017002          MOV     #PROMT,-(SP)
          016752 012746 000001          MOV     #1,-(SP)
          016756 010600          MOV     SP,RO
          016760 104417          TRAP   C$PNTF
          016762 062706 000004          ADD     #4,SP
4229 016766 000703          BR      FLASH      ;AND GO BACK
4230 016770 013737 177562 177566 50$:  MOV     TKB,TPB     ;ECHO THE CHARACTER
4231 016776 000207          RETURN
4232
4233          ; CONSOLE EQUATES
4234
4235          177560          TKS = 177560      ;KEYBOARD STATUS REGISTER
4236          177562          TKB = 177562      ;KEYBOARD DATA REGISTER
4237          177566          TPB = 177566      ;PRINTER DATA BUFFER
4238
4239 017000 000000          FLSANS::          .WORD 0      ;SAVE LOCATION FOR KEYBOARD BUFFER
4240
4241
4242 017002 045 116 045 PROMT: .NLIST  BEX
4243          .ASCIZ  /#N#AUNAUTHORIZED CHARACTER, TYPE ONLY 'Y' OR 'N' /
4244          .LIST  BEX
          .EVEN

```

GLOBAL SUBROUTINES SECTION

```

4246
4247
4248
4249
4250
4251
4252
4253
4254
4255
4256
4257
4258
4259
4260
4261
4262
4263
4264
4265
4266
4267
4268
4269
4270
4271
4272
4273
4274
4275
4276
4277
4278
4279
4280
4281
4282
4283
4284 017066
4285 017066 005037 017172
4286 017072 105737 177560
4287 017076 100032
4288 017100
      017100 104422
4289 017102 013737 177562 004064
4290 017110 042737 000200 004064
4291 017116 022737 000015 004064
4292 017124 001004
4293 017126 012737 000001 017172
4294 017134 000413
4295 017136 013737 177562 177566 20$:
4296 017144
      017144 012746 017174
      017150 012746 000001
      017154 010600
      017156 104417
      017160 062706 000004

:*****
: SUBROUTINE CALRET - CONSOLE READ ROUTINE
:*****
:
: FUNCTIONAL DESCRIPTION:
:
: THIS SUBROUTINE READS THE CONSOLE KEYBOARD WITHOUT STOPPING
: THE PROGRAM SEQUENCING. THE READING WILL BE DONE WHEN YOU
: CALL THIS SUBROUTINE BUT NOT UNDER INTERRUPT.
: CALRET AFFECTS THE CARRY BIT. IF THE OPERATOR TYPED A CARRIGE
: RETURN, THE CARRY BIT WILL BE SET. IF THE OPERATOR TYPED
: ANY OTHER CHARACTERS OR NO CHARACTER, THE CARRY BIT WILL BE CLEARED.
: THE ROUTINE WILL ALSO TAKE CARE OF THE 'CNTL C'.
:
: INPUTS:
:
: CONSOLE KEYBOARD BUFFER AND STATUS
:
: IMPLICIT INPUTS:
:
: NONE
:
: OUTPUTS:
:
: IF READ CHARACTER WAS A CARRIGE RETURN, THE CARRY BIT IS SET.
: IF READ CHARACTER WAS ANY OTHER CHARACTERS OR NO CHARACTER
: WAS TYPED, THE CARRY BIT WILL BE CLEARED.
: IF READ CHARACTER WAS A CNTL C, THE SUPERVISOR WILL HANDLE IT.
:
: SUBORDINATE ROUTINES USED:
:
: BREAK - DRS MACRO ;THIS MACRO TAKES CARE OF CNTL C
:
: CALLING SEQUENCE:
:
: CALL CALRET OR JSR PC,CALRET
:*****
CALRET::
      CLR CARRFL ;TEMPORY STORE FOR CARRY BIT
      TSTB TKS ;TEST THE KEYBOARD STATUS REG.
      BPL 10$ ;BRANCH IF NOTHING WAS TYPED
      BREAK ;WAS THE TYPED CHAR. A 'CNTL C'
:
: TRAP C$BRK
      MOV TKB,BAD ;GET OPERATOR INPUT
      BIC #200,BAD ;CLEAR KEYBOARD BUFFER PARITY BIT
      CMP #15,BAD ;WAS THE TYPED CHARACTER A RETURN ?
      BNE 20$ ;BRANCH IF NO
      MOV #1,CARRFL ;SET CARRY FLAG
      BR 10$ ;AND BRANCH
      MOV TKB,TPB ;ECHO THE CHARACTER
      PRINTF #RETME1 ;PRINT THAT ONLY CARRIGE RETURN WILL
:
: TRAP C$PNTF
      MOV #RETME1,-(SP)
      MOV #1,-(SP)
      MOV SP,RO
      TRAP C$PNTF
      ADD #4,SP

```

GLOBAL SUBROUTINES SECTION

```

4297
4298 017164 006237 017172      104:   ASR   CARRFL      ;DO SOMETHING
4299 017170 000207              RETURN      ;AFFECT THE CARRY
4300
4301      ; CONSOLE EQUATES
4302
4303      177560      TKS = 177560      ;KEYBOARD STATUS REGISTER
4304      177562      TKB = 177562      ;KEYBOARD DATA REGISTER
4305      177566      TPB = 177566      ;PRINTER DATA BUFFER
4306
4307 017172 000000      CARRFL: .WORD 0      ;SAVE LOCATION FOR CARRY BIT
4308
4309
4310 017174 045 116 045 RETME1: .NLIST BEX
                                .ASCIZ /#N#AType CNTL C TO ABORT OR RETURN TO GO TO THE NEXT STEP/
4311                                .LIST BEX
4312                                .EVEN
4313

```

GLOBAL SUBROUTINES SECTION

```

4315
4316
4317
4318
4319
4320
4321
4322
4323
4324
4325
4326
4327
4328
4329
4330 017266
      017266
4331 017266 012737 000001 004060
4332 017274
      017274
      017274 000002
4333
4334
4335
4336
4337
4338
4339
4340
4341
4342
4343
4344 017276
      017276
4345 017276 012737 000001 004150
4346 017304
      017304
      017304 000002
4347
4348
4349
4350
4351
4352
4353
4354
4355
4356
4357
4358 017306
      017306
4359 017306 012737 000001 004152
4360 017314
      017314
      017314 000002
4361
4362

```

```

;*****;
; INTERRUPT SERVICE ROUTINES ;
;*****;

; INTERRUPT SERVICE ROUTINE NXM - NON EXISTANT MEMORY TRAP.

; **
; THIS ROUTINE SETS A FLAG NXMFLG TO 1. IT IS EXECUTED WHEN A NON
; EXISTANT MEMORY TRAP OCCURS IF VECTOR 4 HAS BEEN LOADED WITH THE
; ADDRESS NXM.
;
; NXMFLG SHOULD BE CLEARED IMMEDIATELY BEFORE EXECUTING CODE WHICH MAY
; ADDRESS NON EXISTANT MEMORY.
; --

      BGNSRV  NXM
      MOV     #1,NXMFLG           ; FLAG NXM TRAP      NXM::
      ENDSRV
                                          L10016:
                                          RTI

;*****;
; INTERRUPT SERVICE ROUTINE - INTSR
;*****;

; **
; THIS ROUTINE SETS A FLAG (INTFLA) TO 1. IT WILL BE EXECUTED
; WHEN AN DONE INTERRUPT HAS OCCURRED.
; --

      BGNSRV  INTSR
      MOV     #1,INTFLA          ; INTERRUPT FLAG  INTSR::
      ENDSRV
                                          L10017:
                                          RTI

;*****;
; INTERRUPT SERVICE ROUTINE - INTSR2
;*****;

; **
; THIS ROUTINE SETS A FLAG (INTFL2) TO 1. IT WILL BE EXECUTED
; WHEN AN ERROR INTERRUPT HAS OCCURRED.
; --

      BGNSRV  INTSR2
      MOV     #1,INTFL2          ; ERROR INTERRUPT FLAG INTSR2::
      ENDSRV
                                          L10020:
                                          RTI

```

GLOBAL SUBROUTINES SECTION

```

4363
4364 ;.....
4365 ; INTERRUPT SERVICE ROUTINE - CLINT
4366 ;.....
4367
4368 ;...
4369 ; THIS ROUTINE IS A DUMY SERVICE FOR THE LINE TIME CLOCK INTERRUPTS
4370 ; IT WILL BE EXECUTED WHEN AN INTERRUPT AT VECTOR 100 IS OCCURED.
4371 ; ---
4372 017316          BGNSRV CLINT
      017316
4373
4374 017316          ENDSRV          ;NO ACTION IN THIS ROUTINE
      017316
      017316          L10021:
4375          000002          RTI

```

GLOBAL SUBROUTINES SECTION

4377
4378
4379
4380 017320
4381
4382

ENDMOD

GLOBAL SUBROUTINES SECTION

```

4394 .TITLE MISCELLANEOUS SECTIONS
4395 .SBTTL REPORT CODING SECTION
4423
4424 017320          BGNMOD
4425
4426 ; PRINT ROUTINE
4427
4428 ;**
4429 ; FUNCTIONAL DESCRIPTION:
4430 ;
4431 ;     PRINTS OUT TEST TITLES, A CONFIGURATION TABLE OR A STATISTICS
4432 ;     TABLE FOR THE UNITS UNDER TEST. WHICH TO PRINT IS DETERMINED BY
4433 ;     USER INPUT.
4434 ;
4435 ;     THE CONFIGURATION TABLE SHOWS THE HARDWARE CHARACTERISTICS OF
4436 ;     THE UNIT UNDER TEST AND IF PRESENT, OF THE UNIT TO WHICH IT IS
4437 ;     CONNECTED FOR LOOPBACK TESTING. IF THE FIRST UNIT MODE ADDRESS
4438 ;     IS ZERO (IE. NO TESTS HAVE BEEN RUN AND NO HARDWARE PARAMETERS
4439 ;     HAVE BEEN SET UP) THE AUTOMATIC CONFIGURATION ROUTINE 'CONFIG'
4440 ;     IS CALLED TO ASCERTAIN THE HARDWARE CONFIGURATION.
4441 ;
4442 ;     THE STATISTICS TABLE DISPLAYS THE NUMBER OF ERRORS WHICH THE
4443 ;     DIAGNOSTIC HAS DETECTED FOR EACH UNIT, AND WHETHER THE UNIT HAS
4444 ;     BEEN DROPPED FROM TESTING.
4445 ;
4446 ; INPUTS:
4447 ;
4448 ;     THE USER IS ASKED TO TYPE A CHARACTER INDICATING WHETHER TO
4449 ;     PRINT THE TEST TITLES, THE CONFIGURATION TABLE, THE STATISTICS
4450 ;     TABLE OR HOW TO REESTABLISH THE SYSTEM CONFIGURATION.
4451 ;
4452 ; IMPLICIT INPUTS:
4453 ;
4454 ;     THE HARDWARE PARAMETER TABLE IS READ FOR THE CONFIGURATION
4455 ;     PRINTOUT.
4456 ;
4457 ;     THE ERROR TABLE 'ECNT' IS USED FOR THE STATISTICS PRINTOUT.
4458 ;
4459 ;     TEST TITLES ARE ASSUMED TO BE LABELLED WITH THE FORMAT TDMNN,
4460 ;     WHERE NN IS THE TEST NUMBER. NTESTS AT THE START OF THE ROUTINE
4461 ;     MUST EQUAL THE NUMBER OF TESTS IN THE DIAGNOSTIC.
4462 ;
4463 ; OUTPUTS:
4464 ;
4465 ;     EITHER A LIST OF TEST TITLES, A CONFIGURATION TABLE OR A
4466 ;     STATISTICS TABLE ARE OUTPUT.
4467 ;
4468 ;     IF ANY UNIT HAS BEEN DROPPED OR DESELECTED USING THE "UNITS"
4469 ;     SWITCH, A MESSAGE "PLEASE TYPE ADD" MAY BE PRINTED.
4470 ;
4471 ; IMPLICIT OUTPUTS:
4472 ;
4473 ;     NONE.
4474 ;
4475 ; SUBORDINATE ROUTINES USED:
4476 ;
4477 ;     CONFIG - AUTOMATIC CONFIGURATION ROUTINE.

```

REPORT CODING SECTION

```

4478      :      CONPRI - CONFIGURATION PRINT ROUTINE.
4479      :      CRLF  - LINE FEED PRINT ROUTINE.
4480      :
4481      :      FUNCTIONAL SIDE EFFECTS:
4482      :
4483      :      IF NO HARDWARE PARAMETERS ARE SET UP, THE AUTOMATIC
4484      :      CONFIGURATION ROUTINE IS CALLED TO GENERATE A HARDWARE P TABLE.
4485      :
4486      :      REGISTERS R1 TO R5 ARE CORRUPTED.
4487      :
4488      :      CALLING SEQUENCE:
4489      :
4490      :      INVOKED BY THE OPERATOR PRINT COMMAND.
4491      :
4492      :
4493      :
4494      :
4495      :
4496      :
4497      :
4498      :
4499      :
4500      :
4501      :
4502      :
4503      :
4504      :
4505      :
4506      :
4507      :
4508      :
4509      :
4510      :
4511      :
4512      :

```

000031				NTESTS=25.		: 25 TESTS FOT TITLE PRINTOUT
017320				BGNRPT		
017320						L\$RPT::
104450	ASK:			MANUAL		: IS MANUAL INTERVATION ALLOWED ?
103412				BCOMPLETE 10\$: IF YES, BRANCH (UAM NOT SET)
012746		020654		PRINTF @PRA		: PRINT THAT UAM HAS TO BE SET
012746		000001				TRAP C\$MANI
010600						BCS 10\$
104417						MOV @PRA,-(SP)
062706		000004				MOV @1,-(SP)
017344				EXIT RPT		: EXIT PRINT ROUTINE
000167						MOV SP,RO
002770						TRAP C\$PNTF
104443	10\$:			GMANID PR1,CHAR,A,377,1,4,YES;		ADD @4,SP
000406						.WORD J\$JMP
020454						.WORD L10022-2-
000152						TRAP C\$GMAN
020746						BR 10000\$
000377						.WORD CHAR
000001						.WORD T\$CODE
000004						.WORD PR1
						.WORD 377
						.WORD T\$LLOLIM
						.WORD T\$HILIM
023727		020454	000122	CMP CHAR,@'R		: REESTABLISH SYSTEM CONFIG. ?
001457				BEQ RECON		: IF YES, OUTPUT INFORMATION
023727		020454	000124	CMP CHAR,@'T		: TEST LIST REQUESTED ?
001515				BEQ TITLE		: IF YES, OUTPUT TITLES
023727		020454	000103	CMP CHAR,@'C		: CONFIGURATION REQUESTED ?
001574				BEQ CON		: IF YES, OUTPUT CONFIGURATION
023727		020454	000123	CMP CHAR,@'S		: STATISTICS REQUESTED ?
001002				BNE HEL		: IF NOT, PRINT THE HELP MESSAGE
000137		020260		JMP STAT		: IF YES, OUTPUT STATISTICS
017434	HEL:			PRINTF @PR2		: OTHERWISE, PRINT THE HELP MESSAGE
012746		020773				MOV @PR2,-(SP)
012746		000001				MOV @1,-(SP)
010600						MOV SP,RO

REPORT CODING SECTION

```

4528 017672 012703 000022      10$:  MOV      #18.,R3          ; WAIT AFTER 18 LINES
4529
4530 017676          20$:  PRINTF  #TNUM,R1        ; PRINT TEST NUMBER
      017676 010146
      017700 012746 020650
      017704 012746 000002
      017710 010600
      017712 104417
      017714 062706 000006
4531 017720          PRINTF  (R2)          ; AND TITLE
      017720 011246
      017722 012746 000001
      017726 010600
      017730 104417
      017732 062706 000004
4532 017736 062702 000002      ADD      #2,R2          ; GET ADDRESS OF NEXT TITLE
4533 017742 005201          INC      R1            ; AND NEXT TEST NUMBER
4534 017744 020127 000031      CMP      R1,#NTTESTS  ; ALL PRINTED ?
4535 017750 003015          BGT      30$          ; IF YES, EXIT
4536 017752 005303          DEC      R3            ; 18 LINES OUTPUT ?
4537 017754 001350          BNE      20$          ; IF NOT, BRANCH
4538 017756          GMANID  RDY,RFLG,A,377,0,1,YES ; ELSE WAIT FOR OPERATOR TO READ
      017756 104443
      017760 000406
      017762 020462
      017764 000152
      017766 020612
      017770 000377
      017772 000000
      017774 000001
      017776
4539 017776 004737 011542      JSR      PC,CRLF      ; PRINT A LINE FEED
4540 020002 000733          BR       10$          ; AND THEN CONTINUE
4541
4542 020004 000137 020444      30$:  JMP      PREX          ; EXIT
4543
4544 020010 005037 020450      CON:  CLR      STFLG      ; ASSUME DIAGNOSTIC IS NOT STARTED
4545 020014 005037 020452      CLR      GPFLG      ; FLAG NO GPHARDS YET EXECUTED
4546 020020 005737 003774      TST      GPADD      ; ARE ANY PARAMETER ADDRESSES SET UP ?
4547 020024 001402          BEQ      10$          ; IF NOT, BRANCH
4548 020026 005237 020450      INC      STFLG      ; ELSE FLAG DIAGNOSTIC IS STARTED
4549
4550 020032 005737 002012      10$:  TST      L$UNIT      ; ANY UNITS SET UP ?
4551 020036 001404          BEQ      20$          ; IF NOT, BRANCH
4552 020040 023727 002012 000020      CMP      L$UNIT,#16. ; TOO MANY UNITS SET UP ?
4553 020046 003403          BLE      30$          ; IF NOT, BRANCH
4554 020050 012737 000020 002012 20$:  MOV      #16.,L$UNIT  ; SET UP 16 UNITS
4555
4556 020056 013746 002074      30$:  MOV      L$LUN,-(SP) ; SAVE THE UNIT NUMBER BEING TESTED
4557 020062 005037 002074      CLR      L$LUN      ; START WITH UNIT 0
4558
4559 020066 013701 003774          MOV      GPADD,R1    ; GET FIRST PARAM. ADDRESS
4560 020072 005711          TST      (R1)        ; ANYTHING IN IT
4561 020074 001003          BNE      40$          ; IF YES, BRANCH
4562 020076 012737 171000 012514      MOV      #171000,STADD ; ELSE START FOR FIRST IAV/IDV ADDR.
4563
4564 020104 013701 002074      40$:  MOV      L$LUN,R1    ; FORM OFFSET FOR UNIT TABLES

```


REPORT CODING SECTION

```

4598 020322 020137 002012      20$:  CMP      R1,L$UNIT      ; ALL UNITS REPORTED ?
4599 020326 001444              BEQ      60$              ; IF YES, EXIT
4600 020330 010104              MOV      R1,R4           ; FORM OFFSET TO ERROR COUNT
4601 020332 006304              ASL      R4              ;
4602 020334 016405 003734      MOV      ECNT(R4),R5     ; GET UNIT'S ERROR COUNT
4603 020340 005705              TST      R5              ; IS IT NEGATIVE ?
4604 020342 100423              BMI      40$            ; IF YES, REPORT UNTESTED
4605
4606 020344 012703 022334      MOV      #NO,R3          ; ASSUME UNIT IS NOT DROPPED
4607 020350 105761 004034      TSTB    DROPE(R1)       ; CHECK IF IT IS
4608 020354 001402              BEQ      30$            ; IF IT IS NOT, BRANCH
4609 020356 012703 022330      MOV      #YES,R3         ; OTHERWISE PRINT YES
4610 020362              30$:  PRINTF   #PR5,R1,R5,R3 ; ELSE PRINT STATISTICS
      020362 010346              MOV      R3,-(SP)
      020364 010546              MOV      R5,-(SP)
      020366 010146              MOV      R1,-(SP)
      020370 012746 022112      MOV      #PR5,-(SP)
      020374 012746 000004      MOV      #4,-(SP)
      020400 010600              MOV      SP,R0
      020402 104417              TRAP    C$PNTF
      020404 062706 000012      ADD     #12,SP
4611 020410 000411              BR       50$            ; AND LOOK FOR MORE UNITS
4612
4613 020412              40$:  PRINTF   #PR4,R1      ; PRINT 'UNTESTED'
      020412 010146              MOV      R1,-(SP)
      020414 012746 022061      MOV      #PR4,-(SP)
      020420 012746 000002      MOV      #2,-(SP)
      020424 010600              MOV      SP,R0
      020426 104417              TRAP    C$PNTF
      020430 062706 000006      ADD     #6,SP
4614
4615 020434 005201              50$:  INC      R1          ; PREPARE FOR NEXT UNIT
4616 020436 000731              BR       20$            ; IF NOT, REPORT THE NEXT
4617
4618 020440 004737 011542      60$:  JSR      PC,CRLF     ; PRINT A LINE FEED
4619
4620 020444              PREX:  EXIT      RPT
      020444 000167
      020446 001670              .WORD   J$JMP
              .WORD   L10022-2-.
4621
4622 020450 000000              STFLG: .WORD   0          ; SET IF DIAGNOSTIC HAS BEEN STARTED
4623 020452 000000              GPFLG: .WORD   0          ; SET IF ANY GPHARDS ARE EXECUTED
4624 020454 110 040 040 040 CHAR:  .ASCIZ  /H /      ; STORE FOR OPERATOR INPUT
      020457 040 040 000
4625 020462 000000              RFLG:  .WORD   0          ; FLAG FOR 'TYPE RETURN FOR MORE TITLES
4626
4650
4651 020464              TADS:  TITLES          ; LIST OF TEST TITLE ADDRESSES
      020464 027116              .WORD   TSHD1
      020466 027562              .WORD   TSHD2
      020470 030626              .WORD   TSHD3
      020472 031760              .WORD   TSHD4
      020474 033204              .WORD   TSHD5
      020476 034524              .WORD   TSHD6
      020500 035424              .WORD   TSHD7
      020502 036570              .WORD   TSHD8

```

REPORT CODING SECTION

020504	037616				.WORD	TSHD9
020506	040676				.WORD	TSHD10
020510	041400				.WORD	TSHD11
020512	042572				.WORD	TSHD12
020514	043450				.WORD	TSHD13
020516	045612				.WORD	TSHD14
020520	051006				.WORD	TSHD15
020522	054530				.WORD	TSHD16
020524	057454				.WORD	TSHD17
020526	061424				.WORD	TSHD18
020530	062450				.WORD	TSHD19
020532	063746				.WORD	TSHD20
020534	067340				.WORD	TSHD21
020536	071566				.WORD	TSHD22
020540	075120				.WORD	TSHD23
020542	100100				.WORD	TSHD24
020544	102236				.WORD	TSHD25
4655						
4656					.NLIST	BEX
4657						
4658	020546	045	116	045	TT:	.ASCII /%N%ATEST TITLES./
4659	020566	045	116	045		.ASCIZ /%N%A-----%N2/
4660						
4661	020612	124	131	120	RDY:	.ASCIZ /TYPE "RETURN" FOR MORE TITLES/
4662						
4663	020650	045	104	063	TNUM:	.ASCIZ /%D3/
4664						
4665	020654	045	116	045	PRA:	.ASCIZ /%N%ATO USE THE PRINT COMMAND, PLEASE CLEAR THE "UAM" FLAG/
4666						
4667	020746	124	131	120	PR1:	.ASCIZ /TYPE T,R,C,S OR HELP/
4668						
4669	020773	045	116	045	PR2:	.ASCII /%N%ATHE FOLLOWING COMMANDS ARE ACCEPTED :-/
4670	021045	045	116	062		.ASCIZ /%N2%AT - PRINT TEST TITLES/
4671	021100	045	116	045	PR2A:	.ASCIZ /%N%AC - PRINT CONFIGURATION TABLE CURRENTLY USED BY DIAGNOSTIC/
4672	021177	045	116	045	PR2B:	.ASCIZ /%N%AR - PRINT HOW TO REESTABLISH THE SYSTEM CONFIGURATION/
4673	021271	045	116	045	PR2C:	.ASCIZ /%N%AS - PRINT STATISTICS TABLE%N/
4674						
4675	021332	045	116	045	PR2D:	.ASCIZ /%N%ATO REESTABLISH THE SYSTEM CONFIGURATION, ANSWER THE/
4676	021422	045	116	045	PR2E:	.ASCIZ /%N%AHARDWARE QUESTION TYPING "0" AS THE MODE ADDRESS FOR 16 UNITS./
4677	021525	045	116	045	PR2F:	.ASCIZ /%N%N%AEG. MODE REGISTER ADDRESS (0) 0 ? 0...../
4678	021621	045	116	045	PR2G:	.ASCIZ /%N%A-----/
4679	021715	045	116	062	PR3:	.ASCII \%N2%AIDV/IAV-11 MODULE STATISTICS.\
4680	021757	045	116	045		.ASCIZ /%N%A-----/
4681	022022	045	116	062	PR3A:	.ASCIZ /%N2%AUNIT ERRORS DROPPED%N/
4682						
4683	022061	045	116	045	PR4:	.ASCIZ /%N%D3%A UNTESTED NO/
4684						
4685	022112	045	116	045	PR5:	.ASCIZ /%N%D3%S5%D3%S7%T/
4686						
4687	022133	045	116	045	PR6:	.ASCII /%N%AUNIT DROPPED OR DESELECTED - PLEASE TYPE "ADD"/
4688	022215	040	101	116		.ASCIZ / AND TRY AGAIN/
4689						
4690	022234	045	116	045	PR7:	.ASCIZ /%N%PLEASE TYPE "START" OR "RESTART" TO CONTINUE TESTING.%N/
4691						
4692	022330	131	105	123	YES:	.ASCIZ /YES/
4693	022334	116	117	000	NO:	.ASCIZ /NO/
4694						

REPORT CODING SECTION

4695
4696
4697
4698 022340
022340
022340 104425

.LIST BEX
.EVEN
ENDRPT

L10022: TRAP CSRPT

PROTECTION TABLE

```
4700          .SBTTL  PROTECTION TABLE
4701
4702          ;++
4703          ; THIS TABLE IS USED BY THE RUNTIME SERVICES
4704          ; TO PROTECT THE LOAD MEDIA.
4705          ;--
4706
4707 022342      BGNPROT
4708          L$PROT::
4709 022342 000000      0          ;OFFSET INTO P-TABLE FOR MODE ADDRESS
4710 022344 177777     -1          ;OFFSET INTO P-TABLE FOR MASSBUS ADDRESS
4711 022346 177777     -1          ;OFFSET INTO P-TABLE FOR DRIVE NUMBER
4712
4713 022350      ENDPROT
4714
```

INITIALIZE SECTION

```

4729          .SBTTL  INITIALIZE SECTION
4730
4731          ;
4732          ; THE INITIALIZE SECTION CONTAINS THE CODING THAT IS PERFORMED
4733          ; AT THE BEGINNING OF EACH PASS.
4734          ;
4735
4736 022350          BGNINIT
4737          L$INIT::
4761
4762 022350          START: READEF  @EF.START          ; IS THIS A NEW START ?
4763          022350 012700 000040
4764          022354 104447
4765          022356          BNCOMPLETE RESTR1          ; IF NOT, BRANCH
4766          022356 103126
4767          022360          SETVEC  @14,@106736,@340; *** JUST FOR DEBUG PROGRAM ***
4768          022360 012746 000340
4769          022364 012746 106736
4770          022370 012746 000014
4771          022374 012746 000003
4772          022400 104437
4773          022402 062706 000010
4774          022406          SETVEC  @100,@CLINT,@340; IGNORE FURTHER INTERRUPTS TO VECTOR 100
4775          022406 012746 000340
4776          022412 012746 017316
4777          022416 012746 000100
4778          022422 012746 000003
4779          022426 104437
4780          022430 062706 000010
4781          022434 005737 002232
4782          022440 001024
4783          022442          TST      MANTST          ; DOING MANUFACTURING TESTS ?
4784          022442 012746 023652
4785          022446 012746 000001
4786          022452 010600
4787          022454 104417
4788          022456 062706 000004
4789          022462          PRINTF  @WARN          ; PRINT 'DISCONNECT EXTERNAL EQUIPMENT'
4790          022462 012746 023756
4791          022466 012746 000001
4792          022472 010600
4793          022474 104417
4794          022476 062706 000004
4795          022502 004737 011570
4796          022506 004737 011542
4797          022512          JSR      PC,WRDY          ; WAIT FOR OPERATOR TO TYPE 'RETURN'
4798          022512 104433
4799          022514 004737 024222
4800          022514          JSR      PC,CRLF          ; PRINT A LINE FEED
4801          022520          BRESET          ; RESET THE SYSTEM
4802          022520 104433
4803          022524          JSR      PC,SETCLK          ; SET UP CLOCK COUNTER
4804          022526          TST      L$UNIT          ; ANY UNITS CONFIGURED ?
4805          022526 001404
4806          022526          BEQ      20$          ; IF NOT, BRANCH
4807          022526 023727 002012 000020
4808          022534          CMP      L$UNIT,@16.          ; TOO MANY UNITS SET UP ?
4809          022536 012737 000020 002012 20$:
4810          022536          BLE      30$          ; IF NOT, BRANCH
4811          022544          MOV      @16.,L$UNIT          ; ELSE, SET UP 16 UNITS
4812          005037 023650          30$: CLR      ACFLG          ; CLEAR AUTO CONFIGURATION FLAG

```


INITIALIZE SECTION

```

4782 022550 012737 171000 012514      MOV      #171000,STADD ; START ADDRESS FOR SEARCH
4783 022556 012700 004070              MOV      #LOPFLG,RO  ; FLAG THAT LOOP CONFIGURATION IS NOT CHECKED
4784 022562 012701 000020              MOV      #16.,R1    ; FOR POSSIBLE 16 UNITS
4785 022566 005020          50$:      CLR      (RO).       ; CLEAR THE FLAG
4786 022570 005301              DEC      R1         ; ALL DONE ?
4787 022572 001375              BNE     50$        ; IF NOT, DO THE NEXT
4788
4789
4790 022574 012700 004034              MOV      #DROPPED,RO ; GET UNIT DROPPED TABLE ADDRESS
4791 022600 012701 000020              MOV      #16.,R1    ; THERE ARE 16 UNITS
4792 022604 105020          55$:      CLRB   (RO).       ; CLEAR ALL 16 DROPPED UNIT FLAGS
4793 022606 005301              DEC      R1         ; ...
4794 022610 001375              BNE     55$        ; ...
4795 022612 012700 003734              MOV      #ECNT,RO   ; GET ERROR COUNT FOR UUT 0
4796 022616 012701 000020              MOV      #16.,R1    ; THERE ARE 16 UUT'S
4797 022622 012720 100000          60$:      MOV      #100000,(RO). ; INIT THE ERROR COUNT
4798 022626 005301              DEC      R1         ; ALL CLEARED ?
4799 022630 001374              BNE     60$        ; IF NOT, CLEAR NEXT COUNT
4800 022632 000404              BR      STARES     ; ELSE, START TESTING WITH FIRST UUT
4801
4802 022634          RESTRT: REDEF  #EF.RESTART ; IS THIS A RESTART ?
      022634 012700 000037              MOV      #EF.RESTART,RO
      022640 104447              TRAP   C$REFG
4803 022642          BNCOMPLETE NEWST ; IF NOT, BRANCH
      022642 103005              BCC    NEWST
4804
4805 022644 005037 004054          STARES: CLR      TSTFLG ; SHOW NO TESTS HAVE BEEN RUN THIS PASS
4806 022650 005037 004056              CLR      TSUFLG   ; OR FOR THIS UNIT
4807 022654 000536              BR      INIUUT    ; START TESTING WITH FIRST UUT
4808
4809 022656          NEWST: REDEF  #EF.NEW ; IS THIS A NEW PASS ?
      022656 012700 000035              MOV      #EF.NEW,RO
      022662 104447              TRAP   C$REFG
4810 022664          BNCOMPLETE CONT ; IF NOT, BRANCH
      022664 103030              BCC    CONT
4811 022666 005737 004054              TST     TSTFLG   ; WERE ANY TESTS SELECTED LAST TIME ?
4812 022672 001011              BNE     10$      ; IF YES, BRANCH
4813 022674          PRINTF #NTEST ; IF NOT, TELL THE USER
      022674 012746 024056              MOV     #NTEST,-(SP)
      022700 012746 000001              MOV     #1,-(SP)
      022704 010600              MOV     SP,RO
      022706 104417              TRAP   C$PNTF
      022710 062706 000004              ADD     #4,SP
4814 022714          DOCLN ; AND RETURN TO THE SUPERVISOR
      022714 104444              TRAP   C$DCLN
4815
4816 022716 005037 004054          10$:   CLR      TSTFLG ; INIT TEST FLAG AGAIN
4817 022722 005737 023650              TST     ACFLG    ; JUST DONE A RUN OF AUTO CONFIGURATION ?
4818 022726 001511              BEQ     INIUUT   ; IF NOT, TEST FIRST UNIT
4819 022730 005037 023650              CLR     ACFLG    ; ELSE, FLAG THAT THE RUN IS FINISHED
4820 022734 004737 012600              JSR    PC,CONPRI ; PRINT THE CONFIGURATION
4821 022740 004737 011570              JSR    PC,WRDY   ; WAIT FOR OPERATOR TO TYPE 'RETURN'
4822 022744 000502              BR     INIUUT    ; THEN TEST THE FIRST UNIT
4823
4824 022746          CONT: REDEF  #EF.CONTINUE ; IS THIS A CONTINUE ?
      022746 012700 000036              MOV     #EF.CONTINUE,RO
      022752 104447              TRAP   C$REFG

```

INITIALIZE SECTION

```

4825 022754          BNCOMPLETE PWRFL          ; IF NOT, BRANCH
      022754 103025
4826 022756          SETVEC #4,#NXM,#PRI07 ; IGNORE NXM TRAPS
      022756 012746 000340
      022762 012746 017266
      022766 012746 000004
      022772 012746 000003
      022776 104437
      023000 062706 000010
4827 023004 052777 000100 160666
4828 023012          BIS #100,#MOD          ; ELSE, LIGHT OUT LED
      023012 012700 000004          CLRVEC #4 ; RESTORE THE SUPERVISOR NXM VECTOR
      023016 104436
4829 023020          JSR PC,LOPCHK          ; CHECK THE LOOP CONFIGURATION
4830 023024          JMP END                ; AND CONTINUE
4831
4832 023030          PWRFL: READEF #EF,PWR      ; IS THIS A POWER FAIL
      023030 012700 000034
      023034 104447
4833 023036          BNCOMPLETE NXTUUT        ; IF NOT, MUST BE NEXT UNIT
      023036 103023
4834 023040          SETVEC #4,#NXM,#PRI07 ; IGNORE NXM TRAPS
      023040 012746 000340
      023044 012746 017266
      023050 012746 000004
      023054 012746 000003
      023060 104437
      023062 062706 000010
4835 023066 052777 000100 160604
4836 023074          BIS #100,#MOD          ; ELSE, LIGHT OUT LED
      023074 012700 000004          CLRVEC #4 ; RESTORE THE SUPERVISOR NXM VECTOR
      023100 104436
4837 023102          JMP PSEUL1              ; AND CONTINUE
4838
4839 023106          NXTUUT: TST TSUFLG        ; WERE ANY TESTS RUN ON THE LAST UNIT ?
4840 023112          BNE 10$                 ; IF YES, BRANCH
4841 023114          PRINTF #NTEST1,L$LUN      ; ELSE PRINT A WARNING
      023114 013746 002074
      023120 012746 024152
      023124 012746 000002
      023130 010600
      023132 104417
      023134 062706 000006
4842 023140          JSR PC,WRDY              ; LET THE OPERATOR READ IT
4843 023144          CLR TSUFLG              ; SHOW NO TESTS FOR NEXT UNIT
4844 023150          BR NEXT                 ; AND TEST THE NEXT UNIT
4845
4846 023152          INIUUT: MOV #-1,L$LUN      ; INITIALIZE LOGICAL UNIT NUMBER.
4847 023160          NEXT: INC L$LUN          ; NEXT LOGICAL UNIT TO BE TESTED ?
4848 023164          CMP L$LUN,L$UNIT        ; ALL UNITS TRIED ?
4849 023172          BGE INIUUT              ; IF YES, START AGAIN
4850
4851 023174          MOV L$LUN,R5             ; SAVE UNIT NUMBER
4852 023200          ASL R5                  ; FORM OFFSET
4853
4854 023202          GPHARD L$LUN,R1          ; GET PARAMETER TABLE ADDRESS IN R1
      023202 013700 002074

```

INITIALIZE SECTION

```

023206 104442
023210 010001
4855 023212          BNCOMPLETE NEXT          ; IF DROPPED, GET THE NEXT          TRAP    C$GPHRD
                                MOV          RO,R1
023212 103362
4856 023214 010165 003774      MOV          R1,GPADD(R5)      ; ELSE SAVE THE ADDRESS
4857 023220 005711          TST          (R1)             ; MODE ADDRESS = 0 ?
4858 023222 001024          BNE          20$             ; IF NOT, BRANCH
4859 023224 004737 012214      JSR          PC,CONFIG        ; ELSE DO AUTO CONFIGURATION FOR THIS UNIT
4860 023230 012737 000001 023650  MOV          #1,ACFLG         ; AND FLAG THAT WE ARE DOING IT
4861 023236 103014          BCC          10$             ; BRANCH IF ANOTHER UNIT FOUND
4862 023240 005737 002012      TST          L$UNIT          ; ARE THERE ANY UNITS TO TEST ?
4863 023244 001342          BNE          INIUUT          ; IF YES, START AGAIN WITH THE FIRST UNIT
4864 023246          PRINTF  #NODEV           ; ELSE PRINT "NO UNITS FOUND"
                                MOV          #NODEV,-(SP)
                                MOV          #1,-(SP)
                                MOV          SP,RO
                                TRAP    C$PNTF
                                ADD          #4,SP
                                MOV          #4,SP
                                TRAP    C$DCLN
023246 012746 007250
023252 012746 000001
023256 010600
023260 104417
023262 062706 000004
4865 023266          DOCLN              ; AND RETURN TO THE SUPERVISOR
                                TRAP    C$DCLN
023266 104444
4866
4867 023270 016501 003774      10$: MOV          GPADD(R5),R1  ; GET PARAMETER TABLE ADDRESS IN R1
4868
4869 023274 004737 025146      20$: JSR          PC,LOPCHK    ; CHECK THE LOOP CONFIGURATION
4870
4871 023300 042765 100000 003734  BIC          #100000,ECNT(R5); FLAG UNIT IS BEING TESTED
4872
4873 023306 011137 003700      MOV          (R1),MOD        ; SAVE NEW MODE REGISTER ADDRESS
4874 023312 011137 003702      MOV          (R1),DAT        ; SAVE NEW DATA REGISTER ADDRESS
4875 023316 062737 000002 003702  ADD          #2,DAT
4876 023324 011137 003704      MOV          (R1),CSA        ; SAVE NEW CSA REGISTER ADDRESS
4877 023330 062737 000004 003704  ADD          #4,CSA
4878 023336 012137 003706      MOV          (R1),,CSB       ; SAVE NEW CSB REGISTER ADDRESS
4879 023342 062737 000006 003706  ADD          #6,CSB
4880
4881 023350 012137 003710      MOV          (R1),,VEC       ; SAVE NEW VECTOR ADDRESS
4882 023354 012137 003712      MOV          (R1),,PRIO      ; SAVE NEW PRIORITY
4883 023360 012137 003714      MOV          (R1),,COND      ; SAVE WHETHER OTHER MODULE CONNECTED
4884 023364 012137 003716      MOV          (R1),,LOWLVL    ; SAVE LOW LEVEL AND
4885 023370 012137 003720      MOV          (R1),,DBOUNC    ; DEBOUNCE ANSWER FOR DIGITAL INPUT LOOPB.
4886 023374 011137 003722      MOV          (R1),OTHMOD     ; SAVE OTHER MODULE MODE ADDRESS
4887 023400 011137 003724      MOV          (R1),OTHDAT     ; AND DATA REGISTER ADDRESS
4888 023404 062737 000002 003724  ADD          #2,OTHDAT
4889 023412 011137 003726      MOV          (R1),OTHCSA     ; AND REGISTER A ADDRESS
4890 023416 062737 000004 003726  ADD          #4,OTHCSA
4891 023424 012137 003730      MOV          (R1),,OTHCSB    ; AND REGISTER B ADDRESS
4892 023430 062737 000006 003730  ADD          #6,OTHCSB
4893 023436 011137 003732      MOV          (R1),OTHVEC     ; AND OTHER MODULE VECTOR ADDRESS
4894
4895 023442          SETVEC  #4,#NXM,#PRIO7    ; IGNORE NXM TRAPS
                                MOV          #PRIO7,-(SP)
                                MOV          #NXM,-(SP)
                                MOV          #4,-(SP)
                                MOV          #3,-(SP)
                                TRAP    C$SVEC
                                ADD          #10,SP
023442 012746 000340
023446 012746 017266
023452 012746 000004
023456 012746 000003
023462 104437
023464 062706 000010
4896 023470 052777 000100 160202  BIS          #100,#MOD        ; SWITCH ON UUT LED

```

INITIALIZE SECTION

```

4897 023476 017701 160176      MOV      @MOD,R1      ; GET MODE REGISTER CONTENTS
4898 023502      CLRVEC   #4          ; RESTORE NXM TRAP CATCHER
      023502 012700 000004      MOV      #4,R0
      023506 104436      TRAP    C$CVEC
4899
4900 023510 012702 000020      MOV      #20,R2      ; ASSUME UNIDENTIFIABLE MODULE
4901 023514 020127 140000      CMP      R1,#140000  ; IS ID TOO BIG ?
4902 023520 103013      BHIS    PSEUL1      ; IF YES, BRANCH
4903 023522 012702 000010      MOV      #10,R2     ; ELSE ASSUME ANALOGUE OUTPUT
4904 023526 006301      ASL     R1          ; SHIFT OUT TOP BIT
4905 023530 103407      BCS    PSEUL1      ; IF ID > 177, ID IS ANALOGUE OUTPUT
4906 023532 006202      ASR     R2          ; ELSE ASSUME ANALOGUE INPUT
4907 023534 006301      ASL     R1          ; SHIFT OUT NEXT BIT
4908 023536 103404      BCS    PSEUL1      ; IF ID > 77, ID IS ANALOGUE INPUT
4909 023540 006202      ASR     R2          ; ELSE ASSUME DIGITAL OUTPUT
4910 023542 006301      ASL     R1          ; SHIFT OUT TOP BIT
4911 023544 103401      BCS    PSEUL1      ; IF ID > 37, ID IS DIGITAL OUTPUT
4912 023546 006202      ASR     R2          ; ELSE ID IS DIGITAL INPUT
4913
4914 023550 052702 000400      PSEUL1: BIS    #400,R2  ; FORCE BASIC TESTS TO BE RUN
4915 023554 005737 002232      TST     MANTST     ; MANUFACTURING TESTS SELECTED ?
4916 023560 001003      BNE     40$        ; IF YES, BRANCH
4917 023562 052702 001000      BIS    #1000,R2   ; ELSE FORCE FIELD I/O TESTS
4918 023566 000412      BR     50$        ;
4919 023570 005737 002234      40$:   TST     MANIO  ; MANUFACTURING I/O TESTS SELECTED ?
4920 023574 001407      BEQ    50$        ; IF NOT, BRANCH
4921 023576 052702 004000      BIS    #4000,R2  ; IF YES, SELECT THEM
4922 023602 005737 003714      TST     COND      ; OTHER MODULE CONNECTED ?
4923 023606 001402      BEQ    50$        ; IF NOT, BRANCH
4924 023610 052702 002000      BIS    #2000,R2  ; ELSE SELECT LOOPBACK TESTS
4925 023614 052702 010000      50$:   BIS    #10000,R2 ; ASSUME SPECIFICALLY SELECTED TEST
4926 023620 010237 004132      MOV     R2,CONMSK ; SAVE TEST CONTROL MASK
4927
4928 023624 005037 004130      CLR     LOTFLA    ; INIT LOOP ON TEST FLAG
4929 023630      RFLAGS  R0        ; GET OPERATOR FLAGS
      023630 104421      TRAP    C$RFLA
4930 023632 032700 000010      BIT     #LOT,R0   ; LOOP ON TEST SELECTED ?
4931 023636 001402      BEQ    END        ; IF NOT, BRANCH
4932 023640 005237 004130      INC    LOTFLA    ; ELSE SET FLAG
4933
4934 023644      END:   EXIT     INIT
      023644 104432      TRAP    C$EXIT
      023646 000352      .WORD  L10024-
4935
4936 023650 000000      ACFLG: .WORD    0      ; SET IF AUTO CONFIGURATION IS TO BE DONE
4937
4938      .NLIST  BEX
4939 023652      045    116    045  WARN:  .ASCIZ  /#N#ATHE FOLLOWING TESTS MAY GENERATE SIGNALS ON THE OUTPUT MODULES./
4940 023756      045    116    045  WARN1: .ASCII  /#N#AEXTERNAL EQUIPMENT SHOULD BE DISCONNECTED BEFORE/
4941 024042      040    120    122  .ASCIZ  / PROCEEDING./
4942
4943 024056      045    116    045  NTEST: .ASCIZ  /#N#AND TESTS WERE RUN - CHECK ANSWERS TO STARTUP QUESTIONS./
4944 024152      045    116    045  NTEST1: .ASCIZ  /#N#AND TESTS WERE RUN ON UNIT #D3#A./
4945      .LIST   BEX
4946      .EVEN
4947
4948 024220      ENDTNIT

```

INITIALIZE SECTION

024220
024220 104411
4949
4961

L10024: TRAP C\$INIT

INITIALIZE SECTION

```

4963 :*****
4964 : SUBROUTINES USED DURING INITIALISATION. *
4965 :*****
4966
4967 .SBTTL SETCLK - ROUTINE TO SET UP DELAY COUNTS
4968
4969
4970 :**
4971 : FUNCTIONAL DESCRIPTION:
4972 :
4973 :     THIS ROUTINE SETS UP 3 DELAY VARIABLES CALLED CNT25M, CNT500, AND
4974 :     CNT25. THESE GIVE DELAYS OF APPROXIMATELY 25 MILLISECONDS, 500
4975 :     MICROSECONDS OR 25 MICROSECONDS RESPECTIVELY IF USED AS FOLLOWS:-
4976 :
4977 :             MOV     CNTXXX,RO
4978 :             1$: DEC  RO
4979 :             BNE    1$
4980 :
4981 :     THE COUNTS ARE DERIVED FROM AN L CLOCK IF THERE IS ONE.
4982 :     OTHERWISE, THE OPERATOR IS ASKED TO TYPE 2 CHARACTERS ON THE
4983 :     CONSOLE 6 SECONDS APART.
4984 :
4985 : INPUTS:
4986 :
4987 :     NONE.
4988 :
4989 : IMPLICIT INPUTS:
4990 :
4991 :     IF CNT25M IS NOT ZERO (ALREADY SET UP), THE ROUTINE DOES NOTHING.
4992 :
4993 : OUTPUTS:
4994 :
4995 :     CONSOLE MESSAGE IF THERE IS NO L CLOCK ON THE SYSTEM.
4996 :
4997 : IMPLICIT OUTPUTS:
4998 :
4999 :     CNT25M CONTAINS THE COUNT REQUIRED FOR 25 MILLISECONDS.
5000 :     CNT500 CONTAINS THE COUNT REQUIRED FOR 500 MICROSECONDS.
5001 :     CNT25 CONTAINS THE COUNT REQUIRED FOR 25 MICROSECONDS.
5002 :
5003 : SUBORDINATE ROUTINES USED:
5004 :
5005 :     CRLF - LINE FEED PRINT ROUTINE.
5006 :     CLINT - DUMMY CLOCK INTERRUPT SERVICE ROUTINE
5007 :
5008 : FUNCTIONAL SIDE EFFECTS:
5009 :
5010 :     RO TO R5 ARE CORRUPTED.
5011 :
5012 :     IF A LINE TIME CLOCK IS FOUND, VECTOR 100 IS SET UP SO THAT
5013 :     INTERRUPTS TO IT ARE IGNORED. THE SETVEC MACRO CAN BE USED TO
5014 :     SET UP THE VECTOR FOR A DEVICE INTERRUPT.
5015 :
5016 : CALLING SEQUENCE:
5017 :
5018 :     JSR     PC,SETCLK
5019 :

```

SETCLK - ROUTINE TO SET UP DELAY COUNTS

```

5020
5021 024222 005737 011534      SETCLK: TST      CNT25M      ; COUNTERS ALREADY SET UP?
5022 024226 001402              BEQ      10$              ; IF NOT, BRANCH
5023 024230 000137 025072      JMP      SETEX            ; IF YES, EXIT
5024
5025 024234 005004              10$:  CLR      R4          ; CLEAR A COUNTER
5026 024236              GETPRI   R2              ; SAVE CURRENT PRIORITY IN R2
                                TRAP      C$GPRI
                                MOV      RO,R2
5027 024242 005037 025074      CLR      CLKFLG          ; ASSUME THERE IS NO CLOCK WITH A CSR
5028 024246              CLOCK   L,R1            ; GET ADDRESS OF CLOCK TABLE
                                MOV      #L,R0
                                TRAP      C$CLKC
                                MOV      RO,R1
                                TRAP      C$SVEC
                                ADD     #10,SP
5029 024256              SETVEC  #4,#NXM,#340    ; SET UP CLOCK CSA TRAP
                                MOV      #340,-(SP)
                                MOV      #NXM,-(SP)
                                MOV      #4,-(SP)
                                MOV      #3,-(SP)
                                TRAP
5030 024304 005037 004060      CLR      NXMFLG          ; CLEAR NXM FLAG
5031 024310 005771 000000      TST      @(R1)           ; ACCESS THE CLOCK ADDRESS
5032 024314 005737 004060      TST      NXMFLG          ; DOES THE CLOCK HAVE A REGISTER ?
5033 024320 001005              BNE     LCLOCK           ; IF NOT, BRANCH
5034 024322 005237 025074      INC     CLKFLG          ; ELSE FLAG THERE IS A CLOCK CSR
5035 024326 012771 000100 000000  MOV     #100,@(R1)       ; AND SET IT UP TO INTERRUPT
5036
5037      ; USE THE L CLOCK
5038
5039 024334              LCLOCK: CLRVEC  #4      ; SET VECTOR 4 TO UNUSED POOL
                                MOV      #4,R0
                                TRAP      C$CVEC
5040 024342 012703 000006              MOV     #6,R3            ; IF 50 HZ, 100 MS = 5 INTERRUPTS
5041 024346 026127 000006 000062  CMP     6(R1),#50.       ; 50 HZ CORRECT?
5042 024354 001401              BEQ     10$              ; IF YES, BRANCH
5043 024356 005203              INC     R3                ; ELSE ALLOW 6 INTERRUPTS
5044
5045 024360 010305              10$:  MOV     R3,R5          ; SAVE NUMBER OF INTERRUPTS
5046 024362              SETVEC  #100,#KLINT,#340; SET UP THE CLOCK VECTOR
                                MOV      #340,-(SP)
                                MOV      #KLINT,-(SP)
                                MOV      #100,-(SP)
                                MOV      #3,-(SP)
                                TRAP      C$SVEC
                                ADD     #10,SP
5047
5048 024410              SETPRI  #0              ; TO WAIT FOR 1ST INTERRUPT
                                ; AND DROP THE PRIORITY
                                MOV      #0,R0
                                TRAP      C$SPRI
5049 024416 005000              20$:  CLR     R0              ; CLEAR R0 AND THE CARRY BIT
5050 024420 020305              CMP     R3,R5            ; HAS COUNT BEEN DROPPED ?
5051 024422 001004              BNE     30$              ; IF YES, START THE COUNTERS
5052 024424 005300              DEC     R0                ; WAITED TOO LONG ?
5053 024426 001374              BNE     20$              ; IF NOT, WAIT LONGER
5054 024430 000137 024542      JMP     USCLOK           ; IF YES, ASSUME NO CLOCK
5055

```

SETCLK - ROUTINE TO SET UP DELAY COUNTS

```

5056 024434 005005      30$: CLR R5          ; CLEAR THE HIGH COUNTER
5057 024436 005204      40$: INC R4          ; COUNT THE DELAY FOR 5 OR 6 INTERRUPTS
5058 024440 001376      BNE 40$          ;
5059 024442 105205      INCB R5         ;
5060 024444 001374      BNE 40$         ;
5061 024446 000435      BR USCLOK       ; IF TOO LONG, ASSUME NO CLOCK
5062
5063 024450 005303      KLINT: DEC R3      ; 5 OR 6 INTERRUPTS?
5064 024452 001401      BEQ 40$       ; IF YES, TIDY UP
5065 024454 000002      RTI          ; ELSE KEEP COUNTING
5066
5067 024456      40$: SETPRI R2          ; RESTORE THE PRIORITY
      024456 010200
      024460 104441
5068 024462      SETVEC #100,#CLINT,#340; IGNORE FURTHER INTERRUPTS TO VECTOR 100
      024462 012746 000340
      024466 012746 017316
      024472 012746 000100
      024476 012746 000003
      024502 104437
      024504 062706 000010
      MOV R2,R0
      TRAP C$SPRI
      MOV #340,-(SP)
      MOV #CLINT,-(SP)
      MOV #100,-(SP)
      MOV #3,-(SP)
      TRAP C$SVEC
      ADD #10,SP
5069 024510 022626      CMP (SP)+,(SP)+ ; TIDY UP THE STACK
5070 024512 005737 025074 TST CLKFLG      ; CAN WE DISABLE A CLOCK ?
5071 024516 001402      BEQ 50$       ; IF NOT, BRANCH
5072 024520 005071 000000 CLR @R1         ; ELSE, DISABLE CLOCK INTERRUPTS
5073
5074 024524 000241      50$: CLC          ; DIVIDE THE 100 MILLISECOND COUNTERS
5075 024526 006005      ROR R5        ; BY 4 TO GIVE 25 MILLISECONDS
5076 024530 006004      ROR R4        ;
5077 024532 000241      CLC          ;
5078 024534 006005      ROR R5        ;
5079 024536 006004      ROR R4        ;
5080 024540 000524      BR SAVCNT     ; AND SAVE THE COUNT
5081
5082 ; COME HERE IF NOT ENOUGH CLOCK INTERRUPTS OCCUR BEFORE THE COUNTERS OVERFLOW
5083 ;
5084 024542      USCLOK: SETPRI R2          ; RESTORE THE PRIORITY
      024542 010200
      024544 104441
      MOV R2,R0
      TRAP C$SPRI
5085 024546      SETVEC #100,#CLINT,#340; IGNORE FURTHER INTERRUPTS TO VECTOR 100
      024546 012746 000340
      024552 012746 017316
      024556 012746 000100
      024562 012746 000003
      024566 104437
      024570 062706 000010
      MOV #340,-(SP)
      MOV #CLINT,-(SP)
      MOV #100,-(SP)
      MOV #3,-(SP)
      TRAP C$SVEC
      ADD #10,SP
5086 024574 005737 025074 TST CLKFLG      ; CAN WE DISABLE A CLOCK ?
5087 024600 001402      BEQ NOCLOK    ; IF NOT, BRANCH
5088 024602 005071 000000 CLR @R1         ; ELSE DISABLE CLOCK INTERRUPTS
5089
5090 ; USE THE CONSOLE FOR TIMING
5091 ;
5092      177560      TKS=177560      ; KEYBOARD STATUS REGISTER
5093      177562      TKB=177562      ; KEYBOARD DATA BUFFER
5094      177564      TPS=177564     ; PRINTER STATUS REGISTER
5095      177566      TPB=177566     ; PRINTER DATA BUFFER
5096

```


SETCLK - ROUTINE TO SET UP DELAY COUNTS

```

5097 024606          NOCLOK: SETVEC #60,#TTINT,#340 ; SET UP INTERRUPT VECTOR
      024606 012746 000340          MOV #340,-(SP)
      024612 012746 024740          MOV #TTINT,-(SP)
      024616 012746 000060          MOV #60,-(SP)
      024622 012746 000003          MOV #3,-(SP)
      024626 104437          TRAP C$SVEC
      024630 062706 000010          ADD #10,SP
5098 024634          PRINTF #TIMMSG          ; 'TYPE 2 CHARACTERS 6 SECONDS APART'
      024634 012746 025076          MOV #TIMMSG,-(SP)
      024640 012746 000001          MOV #1,-(SP)
      024644 010600          MOV SP,R0
      024646 104417          TRAP C$PNTF
      024650 062706 000004          ADD #4,SP
5099
5100 024654 105737 177560          10$: TSTB TKS          ; IS FIRST CHARACTER READY?
5101 024660 100375          BPL 10$          ; IF NOT, WAIT
5102 024662 013700 177562          MOV TKB,R0          ; ELSE GET THE CHARACTER
5103 024666 042700 177600          BIC #177600,R0          ; DISCARD UNWANTED BITS
5104 024672 020027 000003          CMP R0,#3          ; IF 'C, RETURN TO SUPERVISOR
5105 024676 001001          BNE 20$          ;
5106 024700          DOCLN          ;
      TRAP C$DCLN
5107
5108 024702 013737 177562 177566 20$: MOV TKB,TPB          ; NOW ECHO THE CHARACTER
5109 024710          SETPRI #0          ; DROP THE PRIORITY
      MOV #0,R0
      TRAP C$SPRI
      024710 012700 000000
      024714 104441
5110 024716 012737 000100 177560          MOV #100,TKS          ; ALLOW INTERRUPTS
5111
5112 024724 012705 000360          30$: MOV #240,,R5          ; SET UP MODULO 240 COUNTER
5113 024730 005305          40$: DEC R5          ; START COUNTING
5114 024732 001376          BNE 40$          ; R5 IS MODULO 240 COUNTER
5115 024734 005204          INC R4          ; UPDATE THE COUNTER
5116 024736 000772          BR 30$          ; 6 SECONDS/240 = 25 MILLISECONDS
5117
5118 024740          TTINT: SETPRI R2          ; RESTORE THE PRIORITY
      024740 010200          MOV R2,R0
      024742 104441          TRAP C$SPRI
5119 024744          CLRVEC #60          ; AND THE KEYBOARD VECTOR
      024744 012700 000060          MOV #60,R0
      024750 104436          TRAP C$CVEC
5120 024752 022626          CMP (SP)+,(SP)+          ; TIDY UP THE STACK
5121 024754 005037 177560          CLR TKS          ; DISABLE INTERRUPTS
5122 024760 013700 177562          MOV TKB,R0          ; ELSE GET THE CHARACTER
5123 024764 042700 177600          BIC #177600,R0          ; DISCARD UNWANTED BITS
5124 024770 020027 000003          CMP R0,#3          ; IF 'C, RETURN TO SUPERVISOR
5125 024774 001001          BNE 10$          ;
5126 024776          DOCLN          ;
      TRAP C$DCLN
      024776 104444
5127 025000 013737 177562 177566 10$: MOV TKB,TPB          ; ELSE, ECHO THE CHARACTER
5128 025006 004737 011542          JSR PC,CRLF          ; AND PRINT A LINE FEED
5129
5130          ;
5131          ; SAVE THE COUNTERS
5132 025012 010437 011534          SAVCNT: MOV R4,CNT25M          ; SAVE THE 25 MILLISECONDS COUNTER
5133 025016 012700 000062          MOV #50,,R0          ; NOW DIVIDE BY 50
5134 025022 062704 000031          ADD #25,,R4          ; TO NEAREST 50

```

SETCLK - ROUTINE TO SET UP DELAY COUNTS

```

5135 025026 005001          CLR    R1          ; INITIALISE RESULT
5136 025030 160004          10$:  SUB    R0,R4        ; REMAINDER < 0 ?
5137 025032 002402          BLT    20$        ; IF YES, BRANCH
5138 025034 005201          INC    R1          ; ELSE INCREMENT RESULT
5139 025036 000774          BR     10$        ; AND TRY AGAIN
5140 025040 010137 011536   20$:  MOV    R1,CNT500 ; SAVE THE 500 MICROSECONDS COUNTER
5141
5142 025044 012700 000024   MOV    #20.,R0    ; NOW DIVIDE BY 20
5143 025050 062701 000012   ADD    #10.,R1    ; TO NEAREST 20
5144 025054 005002          CLR    R2          ; INITIALISE RESULT
5145 025056 160001          30$:  SUB    R0,R1        ; REMAINDER < 0 ?
5146 025060 002402          BLT    40$        ; IF YES, BRANCH
5147 025062 005202          INC    R2          ; ELSE INCREMENT RESULT
5148 025064 000774          BR     30$        ; AND TRY AGAIN
5149 025066 010237 011540   40$:  MOV    R2,CNT25 ; SAVE THE 25 MICROSECONDS COUNTER
5150
5151 025072 000207          SETEX: RTS    PC    ; RETURN
5152
5153 025074 000000          CLKFLG: .WORD 0    ; SET IF DRS FINDS A CLOCK WITH A CSR
5154
5155                          .NLIST BEX
5156 025076 045 116 045 TIMMSG: .ASCIZ /#N#ATYPE 2 CHARACTERS 6 SECONDS APART >/
5157                          .LIST BEX
5158                          .EVEN

```

LOPCHK - LOOPBACK CHECKING ROUTINE.

5160
5161
5162
5163
5164
5165
5166
5167
5168
5169
5170
5171
5172
5173
5174
5175
5176
5177
5178
5179
5180
5181
5182
5183
5184
5185
5186
5187
5188
5189
5190
5191
5192
5193
5194
5195
5196
5197
5198
5199
5200
5201
5202
5203
5204
5205
5206
5207
5208
5209
5210
5211 025146
5212 025146 010146
5213 025150 010246
5214 025152 010346
5215
5216 025154 013701 002074

```
.SBTTL LOPCHK - LOOPBACK CHECKING ROUTINE.
; **
; FUNCTIONAL DESCRIPTION:
;
; IF UNIT L$LUN IN THE HARDWARE P TABLE IS SPECIFIED AS LOOPED, THE
; ROUTINE CHECKS THAT THE OTHER MODULE CAN BE ADDRESSED AND THAT IT
; IS THE CORRECT TYPE FOR LOOPING TO THE UUT. IF EITHER CHECK FAILS,
; A MESSAGE IS PRINTED AND THE UNIT IS FLAGGED IN THE P TABLE AS NOT
; LOOPED.
;
; INPUTS:
;
; L$LUN - NUMBER OF UNIT TO CHECK
; LOPFLG - ADDRESS OF FLAG FOR UNIT 0. IF THE FLAG FOR UNIT L$LUN
; IS NON ZERO, THE UNIT HAS BEEN CHECKED AND THE ROUTINE
; DOES NOTHING.
;
; IMPLICIT INPUTS:
;
; THE HARDWARE P TABLE.
;
; OUTPUTS:
;
; IF MODULES SPECIFIED AS 'OTHER' IN THE HARDWARE QUESTIONS DO NOT
; RESPOND OR ARE NOT OF THE CORRECT TYPE FOR LOOPING, AN ERROR
; MESSAGE IS PRINTED.
;
; LOPFLG FOR THE SPECIFIED UNIT IS SET TO FLAG THAT THE LOOPBACK
; CONFIGURATION HAS BEEN CHECKED.
;
; IMPLICIT OUTPUTS:
;
; IF THE CONFIGURATION IS NOT CORRECT, THE INCORRECT MODULES ARE
; FLAGGED AS NO LONGER LOOPED IN THE HARDWARE P TABLE.
;
; SUBORDINATE ROUTINES USED:
;
; NXM : NON EXISTANT MEMORY TRAP ROUTINE.
;
; FUNCTIONAL SIDE EFFECTS:
;
; IF A LOOPED MODULE DOES NOT CHECK CORRECTLY, IT IS FLAGGED AS
; NOT LOOPED IN THE HARDWARE P TABLE.
;
; CALLING SEQUENCE:
;
; JSR PC,LOPCHK
;
; --
LOPCHK:
MOV R1,-(SP) ; SAVE REGISTERS R1 TO R3
MOV R2,-(SP) ;
MOV R3,-(SP) ;
MOV L$LUN,R1 ; GET UNIT OFFSET
```

LOPCHK - LOOPBACK CHECKING ROUTINE.

```

5217 025160 006301          ASL      R1          ;
5218 025162 005761 004070  TST      LOPFLG(R1)  ; LOOP CONFIGURATION ALREADY CHECKED ?
5219 025166 001402          BEQ      20$         ; IF NOT, CHECK IT
5220 025170 000137 025630 10$:     JMP      LOPEX       ; EXIT
5221
5222 025174 005237 004070 20$:     INC      LOPFLG      ; SHOW CONFIGURATION IS CHECKED
5223 025200          SETVEC  #4,#NXM,#PRI07 ; SET UP NXM VECTOR
                                MOV      #PRI07,-(SP)
                                MOV      #NXM,-(SP)
                                MOV      #4,-(SP)
                                MOV      #3,-(SP)
                                TRAP    C$SVEC
                                ADD     #10,SP
5224
5225 025226 016102 003774 30$:     MOV      GPADD(R1),R2 ; GET UNIT'S P TABLE
5226 025232 005762 000006  TST      6(R2)       ; IS UNIT LOOPED ?
5227 025236 001574          BEQ      LOPEX       ; IF NOT, EXIT
5228 025240 005037 004060  CLR      NXMFLG      ; ELSE CLEAR NXM FLAG
5229 025244 005772 000010  TST      @10(R2)     ; ACCESS THE OTHER UNIT ADDRESS
5230 025250 005737 004060  TST      NXMFLG      ; DOES OTHER UNIT EXIST ?
5231 025254 001413          BEQ      40$         ; IF YES, BRANCH
5232 025256          PRINTF #LOP1,14(R2) ; ELSE PRINT 'OTHER UNIT DOESN'T RESPOND'
                                MOV      14(R2),-(SP)
                                MOV      #LOP1,-(SP)
                                MOV      #2,-(SP)
                                MOV      SP,RO
                                TRAP    C$PNTF
                                ADD     #6,SP
5233 025302 000534          BR       110$        ; AND DESELECT LOOPING
5234
5235 025304 027227 000014 140000 40$:   CMP      @14(R2),#140000 ; IS OTHER UNIT IXV11 ?
5236 025312 103413          BLO     50$         ; IF YES, BRANCH
5237 025314          PRINTF #LOP2,14(R2) ; ELSE PRINT 'OTHER UNIT NOT IXV11'
                                MOV      14(R2),-(SP)
                                MOV      #LOP2,-(SP)
                                MOV      #2,-(SP)
                                MOV      SP,RO
                                TRAP    C$PNTF
                                ADD     #6,SP
5238 025340 000515          BR       110$        ; DESELECT LOOPING
5239
5240 025342 027227 000000 020000 50$:   CMP      @R2,#20000    ; IS UUT DIGITAL INPUT ?
5241 025350 103013          BHI     60$         ; IF NOT, BRANCH
5242 025352 012703          MOV     #LOP3,R3     ; SAVE DIGITAL INPUT MESSAGE ADDRESS
5243 025356 027227 000014 020000  CMP      @14(R2),#20000 ; IS OTHER UNIT DIGITAL OUTPUT ?
5244 025364 103471          BLO     100$        ; IF NOT, DESELECT LOOPING
5245 025366 027227 000014 037400  CMP      @14(R2),#37400 ;
5246 025374 101065          BHI     100$        ;
5247 025376 000514          BR      LOPEX       ; OTHERWISE, DO NOTHING
5248
5249 025400 027227 000000 040000 60$:   CMP      @R2,#40000    ; IS UUT DIGITAL OUTPUT ?
5250 025406 103007          BHI     70$         ; IF NOT, BRANCH
5251 025410 012703          MOV     #LOP4,R3     ; SAVE DIGITAL OUTPUT MESSAGE ADDRESS
5252 025414 027227 000014 017400  CMP      @14(R2),#17400 ; IS OTHER UNIT DIGITAL INPUT ?
5253 025422 101052          BHI     100$        ; IF NOT, DESELECT LOOPING
5254 025424 000501          BR      LOPEX       ; OTHERWISE DO NOTHING
5255

```

LOPCHK - LOOPBACK CHECKING ROUTINE.

```

5256 025426 027227 000000 100000 70:  CMP      @R2),#100000 ; IS UNIT ANALOGUE INPUT ?
5257 025434 103013          ; BHIS      80:      ; IF NOT, BRANCH
5258 025436 012703 026166          ; MOV      @LOP5,R3 ; SAVE ANALOGUE INPUT MESSAGE ADDRESS
5259 025442 027227 000014 100000          ; CMP      @14(R2),#100000 ; IS OTHER UNIT ANALOGUE OUTPUT ?
5260 025450 103437          ; BLO      100:      ; IF NOT, DESELECT LOOPING
5261 025452 027227 000014 137400          ; CMP      @14(R2),#137400 ;
5262 025460 101033          ; BHI      100:      ;
5263 025462 000462          ; BR       LOPEX    ; OTHERWISE, DO NOTHING
5264
5265 025464 027227 000000 140000 80:  CMP      @R2),#140000 ; IS UUT ANALOGUE INPUT ?
5266 025472 103012          ; BHIS      90:      ; IF NOT, BRANCH
5267 025474 012703 026265          ; MOV      @LOP6,R3 ; SAVE ANALOGUE OUTPUT MESSAGE ADDRESS
5268 025500 027227 000014 040000          ; CMP      @14(R2),#40000 ; IS OTHER UNIT ANALOGUE INPUT ?
5269 025506 103420          ; BLO      100:      ; IF NOT, DESELECT LOOPING
5270 025510 027227 000014 077400          ; CMP      @14(R2),#77400 ;
5271 025516 101444          ; BLOS     LOPEX    ; IF YES, DO NOTHING
5272
5273 025520          90:  PRINTF  @LOP7,(R2) ; PRINT 'UNKNOWN MODULE CANNOT BE LOOPED'
5274 025520 011246          ;
5275 025522 012746 026365          ;
5276 025526 012746 000002          ;
5277 025532 010600          ;
5278 025534 104417          ;
5279 025536 062706 000006          ;
5280 025542 005062 000006          ;
5281 025546 000430          ;
5282          CLR      6(R2) ; CLEAR P TABLE LOOPED PARAMETER
5283          BR       LOPEX ; AND EXIT
5284
5285 025550          100: PRINTF  R3,(R2),14(R2) ; PRINT 'CAN'T BE LOOPED'
5286 025550 016246 000014          ;
5287 025554 011246          ;
5288 025556 010346          ;
5289 025560 012746 000003          ;
5290 025564 010600          ;
5291 025566 104417          ;
5292 025570 062706 000010          ;
5293
5294 025574 005062 000006          110: CLR      6(R2) ; CLEAR P TABLE LOOPED PARAMETER
5295 025600          PRINTF  @LOPDES,L$LUN ; PRINT 'LOOPING DESELECTED'
5296 025600 013746 002074          ;
5297 025604 012746 026444          ;
5298 025610 012746 000002          ;
5299 025614 010600          ;
5300 025616 104417          ;
5301 025620 062706 000006          ;
5302 025624 004737 011570          ;
5303          JSR      PC,WRD1 ; WAIT FOR OPERATOR TO TYPE 'RETURN'
5304
5305          LOPEX: CLRVEC  #4 ; RESTORE SUPERVISOR NXM TRAP
5306
5307          MOV      (SP),R3 ; RESTORE REGISTERS R1 TO R3
5308          MOV      (SP),R2 ;
5309          MOV      (SP),R1 ;
5310          RTS      PC ; AND RETURN
5311
5312          .NLIST  BEX
5313
5314 025646          045      116      062  LOP1: .ASCIIZ /#N2#AOTHER DEVICE AT #06#A DOES NOT RESPOND./

```

LOPCHK - LOOPBACK CHECKING ROUTINE.

```

5292
5293 025723      045      116      062 LOP2:  .ASCIZ  \#N2#AOTHER DEVICE AT #06#A NOT IXV11.\
5294
5295 025771      045      116      062 LOP3:  .ASCII  /#N2#ADIGITAL INPUT AT #06#A CANNOT BE LOOPED TO DEVICE AT/
5296 026062      040      045      117      .ASCII  / #06/
5297
5298 026067      045      116      062 LOP4:  .ASCII  /#N2#ADIGITAL OUTPUT AT #06#A CANNOT BE LOOPED TO DEVICE AT/
5299 026161      040      045      117      .ASCIZ  / #06/
5300
5301 026166      045      116      062 LOP5:  .ASCII  /#N2#AANALOGUE INPUT AT #06#A CANNOT BE LOOPED TO DEVICE AT/
5302 026260      040      045      117      .ASCIZ  / #06/
5303
5304 026265      045      116      062 LOP6:  .ASCII  /#N2#AANALOGUE OUTPUT AT #06#A CANNOT BE LOOPED TO DEVICE AT/
5305 026360      040      045      117      .ASCIZ  / #06/
5306
5307 026365      045      116      062 LOP7:  .ASCIZ  /#N2#AUNKNOWN MODULE AT #06#A CANNOT BE LOOPED./
5308
5309 026444      045      116      045 LOPDES: .ASCIZ  \#N#ALOOPING DESELECTED FOR UNIT #D2#A.\
5310
5311
5312
                    .LIST  BEX
                    .EVEN
    
```

AUTODROP SECTION

5314
5315
5316
5317
5318
5319
5320
5321
5322
5323 026514
026514
5324
5331 026514
026514 012746 000340
026520 012746 017266
026524 012746 000004
026530 012746 000003
026534 104437
026536 062706 000010
5332 026542 005037 004060
5333 026546 005777 155126
5334
5335
5336
5337
5338
5339 026552 005737 004060
5340 026556 001404
5341 026560
026560 013700 002074
026564 104451
5342 026566
026566 104444
5343 026570
026570 012700 000004
026574 104436
5344 026576
026576
026576 104461

.SBTTL AUTODROP SECTION

; THIS CODE IS EXECUTED IMMEDIATELY AFTER THE INITIALIZE CODE IF
; THE "ADR" FLAG WAS SET. THE UNIT(S) UNDER TEST ARE CHECKED TO
; SEE IF THEY WILL RESPOND. THOSE THAT DON'T ARE IMMEDIATELY
; DROPPED FROM TESTING.

BGNAUTO

L\$AUTO::

SETVEC 04,0NXM,0PRI07 ; SET UP NON - EXISTENT MEMORY TRAP VECTOR.
MOV 0PRI07,-(SP)
MOV 0NXM,-(SP)
MOV 04,-(SP)
MOV 03,-(SP)
TRAP C\$SVEC
ADD 010,SP

CLR NXMFLG ; CLEAR NON - EXISTENT MEMORY FLAG
TST 0MOD ; REFERENCE MEMORY ADDRESS FOR THE DEVICE
; TO SEE IF IT EXISTS.

; IF THE DEVICE DOESN'T EXIST, THE RESULTANT TRAP TO VECTOR 04 WILL
; CAUSE THE FLAG NXMFLG TO BE SET (SEE INTERRUPT ROUTINE NXM).

TST NXMFLG ; WAS THERE A TRAP ?
BEQ 10\$; BRANCH IF NOT
DODU L\$LUN ; ELSE DROP THE DEVICE

DOCLN ; CLEAN UP CODE. MOV L\$LUN,RO
TRAP C\$DODU

10\$: CLRVEC 04 ; RETURN VECTOR 04 TO NORMAL STATE TRAP C\$DOCLN

MOV 04,RO
TRAP C\$CVEC

ENDAUTO

L10025:
TRAP C\$AUTO

CLEANUP CODING SECTION

5346
5347
5348
5349
5350
5351
5352
5353
5354
5363
5364
5365
5366
5378
5379
5380
5381

026600
026600
026600 104433
026602 104432
026604 000002
026606
026606
026606 104412

.SBTTL CLEANUP CODING SECTION

; THE CLEANUP CODING SECTION CONTAINS THE CODING THAT IS PERFORMED
; AFTER THE HARDWARE TESTS HAVE BEEN PERFORMED.

BGNCLN

L\$CLEAN::

BRESET

; DO A BUS RESET TO SWITCH OFF ALL LEDS

EXIT CLN

TRAP C\$RESET

TRAP C\$EXIT
.WORD L10026-

.EVEN

ENDCLN

L10026:

TRAP C\$CLEAN

DROP UNIT SECTION

```

5383          .SBTTL  DROP UNIT SECTION
5384
5385          ;**
5386          ; THE DROP-UNIT SECTION CONTAINS THE CODING THAT CAUSES A DEVICE
5387          ; TO NO LONGER BE TESTED.
5388          ;--
5389
5390 026610          BGNDU
5391          026610          L$DU::
5400
5401 026610 010026          MOV    RO,(SP)+          ; SAVE RO CONTENTS
5402 026612 112760 000001 004034          MOVB   #1,DROPED(RO)          ; FLAG UNIT DROPPED IN PARAM TABLE
5403 026620 014600          MOV    -(SP),RO          ; GET ORIGINAL RO CONTENTS
5404 026622          PRINTF  #DROPD,RO          ; 'UNIT DROPPED'
5405          026622 010046
5406          026624 012746 026650          MOV    RO,-(SP)
5407          026630 012746 000002          MOV    #DROPD,-(SP)
5408          026634 010600          MOV    #2,-(SP)
5409          026636 104417          MOV    SP,RO
5410          026640 062706 000006          TRAP  C$PNTF
5411          5405          ADD    #6,SP
5412
5413          5406          EXIT    DU
5414          026644 000167          .WORD  J$JMP
5415          026646 000030          .WORD  L10027-2-.
5416
5417          5408
5418          5420
5419          5421          .NLIST  BEX
5420          5422 026650 045 116 045 DROPD: .ASCIZ /#N#AUNIT #D2#A DROPPED/
5421          5423          .LIST  BEX
5422          5424          .EVEN
5423          5425
5424          5426 026700          ENDDU
5425          026700
5426          026700 104453          L10027: TRAP  C$DU

```

ADD UNIT SECTION

```

5428          .SBTTL  ADD UNIT SECTION
5429
5430          : **
5431          : THE ADD-UNIT SECTION CONTAINS ANY CODE THE PROGRAMMER WISHES
5432          : TO BE EXECUTED IN CONJUNCTION WITH THE ADDING OF A UNIT BACK
5433          : TO THE TEST CYCLE.
5434          : --
5435
5436 026702          BGNAU
5437 026702          L$AU::
5446
5447 026702 105060 004034  CLRB  DROPED(RO)          ; FLAG UNIT NOT DROPPED IN PARAM TABLE
5448
5449 026706          EXIT  AU
5449 026706 000167          .WORD  JSJMP
5449 026710 000000          .WORD  L10030-2-
5450
5462
5463          .EVEN
5464
5465 026712          ENDAU
5465 026712          L10030:
5465 026712 104452          TRAP  C$AU
5466
5467 026714          ENDMOD
5468
5469

```

ADD UNIT SECTION

5471
5482
5483
5518
5519 026714
5520
5521
5522
5523
5524
5525
5532
5538 026714
026714
5539 026714 004737 011714
5540 026720 000417
5541 026722 027116
5542 026724 103465
5543 026726 005037 004144
5544 026732
026732 012746 000340
026736 012746 027112
026742 012746 000004
026746 012746 000003
026752 104437
026754 062706 000010
5545 026760 013701 003700
5546 026764 162701 000002
5547 026770 012702 000004
5548 026774 005003
5549 026776 062701 000002
5550 027002
027002 104404
5551 027004 005004
5552 027006 005711
5553 027010 005704
5554 027012 004737 012146
5555 027016 001405
5556 027020 005203
5557 027022
027022 104456
027024 000145
027026 027150
027030 007634
5558 027032
027032
027032 104405
5559 027034 005302
5560 027036 001357
5561 027040 005703
5562 027042 001404
5563 027044
027044 013700 002074
027050 104451
5564 027052
027052 104444
5565 027054 005737 002242

```
.TITLE HARDWARE TESTS
.SBTTL TEST 1: REGISTER NXM TEST.
      BGNMOD
;.....
;          TEST 1 - REGISTER NXM TEST.
;
;          THIS TEST CHECKS THAT ACCESSING THE DEVICE MODE, DATA, CSA AND
;          CSB REGISTERS (IF PRESENT) DOES NOT CAUSE A NXM TRAP.
;.....
      BGNTST
                                T1::
5539  CALL    SELECT                ;CALL SELECT ROUTINE
5540  .WORD   417                    ;GIVE TEST PARAMETER
5541  TSHD1   ;GIVE TEST HEADER ADDRESS
5542  BCS     EXQV1                  ;IF CARRY IS SET, EXIT TEST
5543  CLR     ITRCNT                 ;CLEAR ITERATION COUNTER
5544  SETVEC  #4,#LOCATE,#PRI07      ;SET UP INTERRUPT ROUTINE
                                MOV     #PRI07,-(SP)
                                MOV     #LOCATE,-(SP)
                                MOV     #4,-(SP)
                                MOV     #3,-(SP)
                                TRAP    C$SVEC
                                ADD     #10,SP
5545  ITRAC1: MOV    MOD,R1           ;GET FIRST REGISTER ADDRESS
5546  SUB     #2,R1
5547  MOV     #4,R2
5548  CLR     R3
5549  10$:   ADD     #2,R1           ;SET COUNTER FOR 4 REGISTERS
5550  BGNSEG ;CLEAR LOCATION FOR ERROR MARK
;          ;GET REGISTER ADDRESS
;
;                                TRAP    C$BSEG
;
5551  CLR     R4
5552  TST    (R1)                   ;TEST REGISTER ADDRESS
5553  TST    R4                     ;WAS THERE A TRAP?
5554  CALL   INSERT                 ;SKIP BRANCH IF "SFI" IS SET
5555  BEQ    20$                    ;IF NO, BRANCH
5556  INC    R3                     ;MARK THE ERROR
5557  ERHRD  101,E101,EERA         ;ERROR HANDLER
;                                TRAP    C$ERHRD
;                                .WORD   101
;                                .WORD   E101
;                                .WORD   EERA
5558  20$:   ENDSEG
;
;                                10000$:
;                                TRAP    C$ESEG
5559  DEC    R2                     ;ALL REGISTERS TESTED
5560  BNE    10$                    ;IF NO, BRANCH
5561  TST    R3                     ;WAS THERE AN ERROR
5562  BEQ    30$                    ;IF NO, DON'T DROP THE UNIT
5563  DODU   L$LUN                  ;DROP THE UNIT UNDER TEST
;                                MOV     L$LUN,R0
;                                TRAP    C$DODU
5564  DOCLN
;                                ;RUN THE CLEAN UP ROUTINE
;                                TRAP    C$DCLN
5565  30$:   TST    QVP             ;IS QUICK VERIFY PASS SELECTED?
```

TEST 1: REGISTER NXM TEST.

```

5566 027060 001007          BNE      EXQV1          ;IF YES EXIT TEST
5567 027062 005237 004144    INC      ITRCNT          ;ITERATION COUNTER + 1
5568 027066 023737 004146 004144  CMP     ITRDEF,ITRCNT    ;DEFAULT ITERATION EXECUTED
5569 027074 001401          BEQ     EXQV1          ;IF YES EXIT TEST
5570 027076 000730          BR      ITRAC1         ;IF NO, TEST ITERATION
5571 027100          EXQV1: CLRVEC  #4          ;
      027100 012700 000004          ;
      027104 104436          ;
5577 027106          TSTEN1: EXIT  TST          ;
      027106 104432          ;
      027110 000106          ;
5578          ;
5590 027112          BGNSRV  LOCATE          ;SERVICE ROUTINE LOCATE
      027112          ;
5591 027112 005204          INC      R4          ;LOCATE::
5592 027114          ENDSRV          ;INCREMENTS R4 IF A TRAP TO 4
      027114          ;HAS OCCURRED
      027114 000002          ;
5593          ;
5594          ;
5595 027116          045 123 062 TSHD1:: .NLIST BEX
5596 027150          122 105 107 E101: .ASCIZ /#S2#AREGISTER NXM TEST.#N/
5597          ;
5598          ;
5599          ;
5600 027216          ;
      027216          ;
      027216 104401          ;

```

L10031: TRAP C\$ETST

TEST 2: RESET TEST

```

5608 .SBTTL TEST 2: RESET TEST
5609 :*****
5610 : TEST 2 - RESET TEST.
5611 : THIS TEST CHECKS THAT THE DEVICE REGISTERS ARE CORRECTLY SET OR RESET
5612 : AFTER A BUS RESET. IN THE MOD REGISTER ONLY THE LED BIT IS TESTED .
5613 :*****
5614 027220 BGNTST
5615 027220 004737 011714 CALL SELECT ;CALL SELECT ROUTINE T2::
5616 027224 000417 .WORD 417 ;GIVE TEST PARAMETER
5617 027226 027562 TSHD2 ;GIVE TEST HEADER ADDRESS
5618 027230 103552 BCS EXQV2 ;IF CARRY IS SET, EXIT TEST
5619 027232 005037 004144 CLR ITRCNT ;CLEAR ITERATION COUNTER
5620 027236 005001 ITRAC2: CLR R1 ;CLEAR TEMPORARY STORE
5621 027240 113701 004132 MOVB CONMSK,R1 ;GET MODULE TYPE
5622 027244 032701 000020 BIT #20,R1 ;DON'T TEST SPECIAL MODULES
5623 027250 004737 012146 CALL INSERT ;SKIP BRANCH IF "SFI" IS SET
5624 027254 001404 BEQ 10$ ;BRANCH IF NO SPEC. MODULE
5625 027256 ERRSOFT 200,E200,EERG ;ERROR HANDLER
5626 027256 104457 TRAP C$ERSOFT
5627 027260 000310 .WORD 200
5628 027262 027605 .WORD E200
5629 027264 007724 .WORD EERG
5630 027266 012702 177777 10$: MOV #-1,R2 ;MODULE IDENTIFICATION
5631 027272 005202 20$: INC R2 ;...
5632 027274 006201 ASR R1 ;...
5633 027276 103375 BCC 20$ ;...
5634 027300 006302 ASL R2 ;MULTIPLY BY 10 TO GET TABLE
5635 027302 006302 ASL R2 ;OFFSET
5636 027304 006302 ASL R2 ;...
5637 027306 005003 CLR R3 ;START WITH DAT REGISTER
5638 027310 104404 BGNSEG
5639 027312 BRESET ;DO A BUS RESET TRAP C$BSEG
5640 027312 104433 TRAP C$RESET
5641 027314 032777 000100 154356 BIT #100,@MOD ;IS LED BIT CLEARED ?
5642 027322 004737 012146 CALL INSERT ;SKIP BRANCH IF "SFI" IS SET
5643 027326 001404 BEQ 30$ ;BRANCH IF YES
5644 027330 ERRSOFT 201,E201,EERG ;ERROR HANDLER
5645 027330 104457 TRAP C$ERSOFT
5646 027332 000311 .WORD 201
5647 027334 027703 .WORD E201
5648 027336 007724 .WORD EERG
5649 027340 30$: CKLOOP ;
5650 027340 104406 TRAP C$CLP1
5651 027342 052777 000100 154330 BIS #100,@MOD ;SWITCH ON THE MODULE LED
5652 027350 032777 000100 154322 BIT #100,@MOD ;IS LED BIT NOW SET ?
5653 027356 004737 012146 CALL INSERT ;SKIP BRANCH IF "SFI" IS SET
5654 027362 001004 BNE 40$ ;BRANCH IF YES
5655 027364 ERRSOFT 202,E202,EERG ;ERROR HANDLER
5656 027364 104457 TRAP C$ERSOFT
5657 027366 000312 .WORD 202
5658 027370 027770 .WORD E202
5659 027372 007724 .WORD EERG
5660 027374 40$: ENDSEG ;
5661 027374 104405 10000$: TRAP C$ESEG

```

TEST 2: RESET TEST

```

5647 027376 013705 003702      MOV    DAT,R5      ;GET FIRST REGISTER ADDRESS
5648 027402 062702 000002      ADD    #2,R2      ;POINT TABLE OFFSET TO DAT CONT.
5649 027406      BGNSEG
5650 027410 104404      BRESET           ;DO A BUS RESET           TRAP    C#BSEG
5651 027412 052777 000100 154260      BIS    #100,@MOD   ;SWITCH ON THE MODULE LED           TRAP    C#RESET
5652 027420 016237 030102 004062 50$:  MOV    R5AV(R2),GOOD ;GET FIRST COMPARE CONTENTS FROM TABLE
5653 027426 011537 004064      MOV    @R5,BAD    ;GET FIRST REGISTER CONTENTS
5654 027432 005703      TST    R3         ;IS THIS THE DAT REGISTER?
5655 027434 001006      BNE    60$       ;IF NO, BRANCH
5656 027436 032737 000005 004132      BIT    #5,CONMSK  ;IS THIS AN INPUT MODULE?
5657 027444 001402      BEQ    60$       ;IF NO, BRANCH
5658 027446 005037 004064      CLR    BAD       ;IF YES, CLEAR DAT CONTENTS
5659 027452 042737 000040 004064 60$:  BIC    #40,BAD   ;MASK OUT OPL BIT FOR ANA. OUTPUT
5660 027460 023737 004062 004064      CMP    GOOD,BAD  ;CMP TABLE CONT.WITH REG. CONT.
5661 027466 004737 012146      CALL  INSERT     ;SKIP BRANCH IF "SFI" IS SET
5662 027472 001404      BEQ    70$       ;
5663 027474      ERRSOFT 203,E203,EERB ;ERROR HANDLER
5664 027474 104457      TRAP   C$ERSOFT
5665 027476 000313      .WORD 203
5666 027500 030035      .WORD E203
5667 027502 007664      .WORD EERB
5668 027504 70$:  CKLOOP      ;
5669 027504 104406      TRAP   C$CLP1
5670 027506 005203      INC    R3        ;COUNTER FOR NEXT REGISTER
5671 027510 022703 000003      CMP    #3,R3    ;3 REGISTERS TESTED ?
5672 027514 001405      BEQ    80$      ;BRANCH IF YES
5673 027516 062702 000002      ADD    #2,R2    ;LOAD NEXT TABLE ADDR.
5674 027522 062705 000002      ADD    #2,R5    ;LOAD NEXT REGISTER ADDR.
5675 027526 000734      BR    50$      ;
5676 027530 80$:  ENDSEG      ;
5677 027530 10001$: TRAP   C$ESEG
5678 027532 005737 002242      TST    QVP      ;IS QUICK VERIFY PASS SELECTED?
5679 027536 001007      BNE    EXQV2    ;IF YES EXIT TEST
5680 027540 005237 004144      INC    ITRCNT   ;ITERATION COUNTER + 1
5681 027544 023737 004146 004144      CMP    ITRDEF,ITRCNT ;DEFAULT ITERATION EXECUTED
5682 027552 001401      BEQ    EXQV2    ;IF YES EXIT TEST
5683 027554 000630      BR    ITRAC2   ;IF NO, TEST ITERATION
5684 027556 104432      EXIT    TST     ;
5685 027560 000362      TRAP   C$EXIT
5686 027560 000362      .WORD L10033-.
5687 027562 045 123 062 TSHD2: .NLIST BEX
5688 027605 123 120 105 E200: .ASCIZ /*S2#ARESET TEST.#N/
5689 027703 114 105 104 E201: .ASCIZ /SPECIAL MODULE FOUND - CAN'T BE TESTED WITH THIS DIAGNOSTIC /
5690 027770 114 105 104 E202: .ASCIZ /LED BIT IN MOD REGISTER NOT CLEARED AFTER BUS RESET /
5691 030035 122 105 107 E203: .ASCIZ /LED BIT IN MOD REGISTER CAN'T BE SET/
5692 030035 122 105 107 E203: .ASCIZ /REGISTER INCORRECT AFTER BUS RESET /
5693 .EVEN
5694 030102 000000 000000 000000 R5AV: .WORD 0,0,0,0 ;MASKS FOR DIGITAL INPUT
5695 030112 000000 000000 000000 .WORD 0,0,0,0 ;MASKS FOR DIGITAL OUTPUT
5696 030122 000000 000000 100000 .WORD 0,0,100000,0 ;MASKS FOR ANALOGUE INPUT
5697 030132 000000 000000 100200 .WORD 0,0,100200,0 ;MASKS FOR ANALOGUE OUTPUT
5698 .LIST
5699 .EVEN

```

TEST 2: RESET TEST

SEQ 0141

5693 030142
030142
030142 104401

ENDTST

L10033: TRAP C\$ETST

TEST 3: REGISTER R/W BIT TEST.

```

5695 .SBTTL TEST 3: REGISTER R/W BIT TEST.
5696 ;*****
5697 ;
5698 ;
5699 ;
5700 ;THIS TEST CHECKS THAT THE READ/WRITE BITS OF EACH REGISTER CAN ALL BE
5701 ;SET, ALL CLEARED AND INDIVIDUALLY SET.
5702 ;*****
5702 030144          BGNTST
5703 030144 004737 011714          CALL SELECT          ;CALL SELECT ROUTINE
5704 030150 000417          .WORD 417          ;TEST SELECT MASK
5705 030152 030626          TSHD3          ;TEST HEADER ADDRESS
5706 030154 103002          BCC 1$          ;IF CARRY IS SET, DON'T BRANCH
5707 030156          EXIT TST          ;EXIT TEST IF CARRY IS SET
5708 030162 005037 004144          1$: CLR ITRCNT          ;CLEAR ITERATION COUNTER
5709 030166 113701 004132          MOVB CONMSK,R1        ;GET MODULE TYPE
5710 030172 032701 000020          BIT #20,R1          ;TEST ONLY ANAL.+ DIGITAL MODULES
5711 030176 004737 012146          CALL INSERT          ;SKIP BRANCH IF "SFI" IS SET
5712 030202 001404          BEQ ITRAC3          ;...
5713 030204          ERRSOFT 300,E300,EERG ;ERROR HANDLER
5714 030214          ITRAC3: SETPRI #PRI07 ;DISABLE INTERRUPTS
5715 030222 005001          CLR R1          ;CLEAR TEMPORARY STORE
5716 030224 113701 004132          MOVB CONMSK,R1        ;GET MODULE TYPE
5717 030230 012702 177777          MOV #-1,R2          ;
5718 030234 005202          10$: INC R2          ;MODULE IDENTIFICATION
5719 030236 006201          ASR R1          ;...
5720 030240 103375          BCC 10$          ;...
5721 030242 006302          ASL R2          ;MULTIPLY BY 8. TO GET TABLE
5722 030244 006302          ASL R2          ;OFFSET
5723 030246 006302          ASL R2          ;
5724 030250 005003          CLR R3          ;START WITH MOD REGISTER
5725 030252 013737 003700 010760          MOV MOD,REGADD        ;LOAD FIRST REGISTER ADDRESS
5726 030260 012737 000455 007200          20$: MOV #301.,ERRNBR      ;LOAD FIRST ERROR NUMBER
5727 030266 016237 031264 010754          MOV RWMAC(R2),MASK    ;GET R/W MASK FORM TABLE
5728 030274 004737 010440          CALL REGTS1          ;CALL REGISTER TEST
5729 030300 005203          INC R3          ;COUNTER FOR NEXT REGISTER
5730 030302 022703 000004          CMP #4,R3          ;4 REGISTERS TESTED ?
5731 030306 001406          BEQ 30$          ;BRANCH IF YES
5732 030310 062702 000002          ADD #2,R2          ;LOAD NEXT TABLE ADDR.
5733 030314 062737 000002 010760          ADD #2,REGADD        ;NEXT REGISTER ADDRESS
5734 030322 000756          BR 20$          ;TEST AGAIN WITH NEW PARAMETER
5735
5736 ;THE FOLLOWING CODE IS FOR TESTING THE RT1 BIT IN A DIGITAL INPUT MODULE.
5737
5738 030324          30$: BGNSEG
5739 030326 104404          MOVB CONMSK,R1        ;GET MODULE TYPE
5740 030332 032701 000001          BIT #1,R1          ;IS MODULE A DIGITAL INPUT?
5741 030336 001515          BEQ 70$          ;BRANCH IF NO
5735
5736
5737
5738
5739
5740
5741

```


TEST 3: REGISTER R/W BIT TEST.

```

5781
5782 030626      045      123      062  TSHD3: .NLIST BEX
5783 030652      123      120      105  E300:  .ASCIZ /#S2#AR-W BIT TEST#N/
5784 030747      115      117      104  E304:  .ASCIZ /SPECIAL MODULE FOUND - CAN'T BE TESTED WITH THIS DIAGNOSTIC /
5785 031037      115      117      104  E305:  .ASCIZ /MOD REGISTER CONTENTS OF DIGITAL INPUT MODULE INCORRECT/
5786 031126      012      015      122  E306:  .ASCIZ /MOD REGISTER CONTENTS OF DIGITAL INPUT MODULE INCORRECT/
5787 031150      115      117      104  E306:  .ASCIZ /MOD REGISTER CONTENTS OF DIGITAL INPUT MODULE INCORRECT/
5788 031237      012      015      122  E306:  .ASCIZ <12><15>/RT0 + RT1 NOT SET/
5789
5790
5791 031264 000011 000000 070000 RWMASK: .LIST BEX
5792 031272 000000 177777 000000 .EVEN
5793 031274 000000 177777 000000 .WORD 11,0,70000,0 ;R/W MASKS FOR DIGITAL INPUT
5794 031302 000000 077136 001420 .WORD 0,177777,0,0 ;R/W MASKS FOR DIGITAL OUTPUT
5795 031304 000000 000000 077136 .WORD 0,0,77136,0 ;R/W MASKS FOR ANALOGUE INPUT
5796 031312 000000 000000 001420 .WORD 0,0,1420,0 ;R/W MASKS FOR ANALOGUE OUTPUT
5797 031314 000000 000000 001420 .WORD 0,0,1420,0 ;R/W MASKS FOR ANALOGUE OUTPUT
5798 031322 000000
5799
5795 .EVEN
5796 031324 .ENDTST
5797 031324 104401
5798 031326
5799
L10034: TRAP C#ETST
ENDMOD

```

TEST 3: REGISTER R/W BIT TEST.

```

5803           .TITLE HARDWARE TESTS
5848 031326           BGNMOD
5849           .SBTTL TEST 4: INTERNAL INTERRUPT LOGIC TEST - DIGITAL INPUT.
5850           ;*****
5851           ; TEST 4 - INTERNAL INTERRUPT LOGIC TEST - DIGITAL INPUT.
5852           ;
5853           ; THIS TEST CHECKS THAT THE INTERNAL INTERRUPT LOGIC IS ABLE TO CAUSE AN
5854           ; INTERRUPT USING THE VECTOR AND PRIORITY LEVEL SELECTED IN THE START UP
5855           ; QUESTIONS. THE INTERRUPT IS GENERATED BY SETTING THE INTERRUPT ENABLE
5856           ; AND THEN INTERRUPT TEST BITS. AFTER INTERRUPT, THE CSA CONTENTS
5857           ; ARE ALSO CHECKED.
5858           ;*****
5859 031326           BGNTST 4
                    T4::
5860 031326 004737 011714          CALL SELECT          ; JUMP TO SELECT ROUTINE
5861 031332 000401                .WORD 401              ; GIVE TEST PARAMETER
5862 031334 031760                TSHD4                ; GIVE TEST HEADER
5863 031336 103002                BCC 1$              ; BRANCH IF CARRY IS CLEARED
5864 031340                EXIT TST                    ; EXIT TEST IF CARRY WAS SET
                    TRAP C$EXIT
5865 031344 005037 004144          1$: CLR ITRCNT          ; CLEAR ITERATION COUNTER
5866 031350                ITRAC4: BGNSEG                .WORD L10035-.
                    TRAP C$BSEG
5867 031352 005037 004150          CLR INTFLA          ; CLEAR INTERRUPT FLAG
5868 031356 005077 152322          CLR @CSA           ; CLEAR CONTROL STATUS REGISTER
5869 031362                SETVEC VEC,@INTSR,@PRI07    ; SET VECTOR AND SERVICE ROUTINE
                    MOV @PRI07,-(SP)
                    MOV @INTSR,-(SP)
                    MOV VEC,-(SP)
                    MOV @3,-(SP)
                    TRAP C$SVEC
                    ADD @10,SP
5870 031410                SETPRI @PRI07              ; DISABLE INTERRUPT AT THIS POINT
                    MOV @PRI07,R0
                    TRAP C$SPRI
5871 031416 012777 040000 152260  MOV @40000,@CSA    ;;; SET IE BIT IN CSA
5872 031424 052777 000400 152252  BIS @400,@CSA      ;;; SET TST IR BIT IN CSA REGISTER
5873 031432 012701 000340          MOV @PRI07,R1     ;;; LOAD R1 WITH PRIORITY NUMBER
5874 031436 004737 011522          10$: CALL WT25     ;;; WAIT 20 US
5875 031442 005737 004150          TST INTFLA       ; INTERRUPT OCCURRED ?
5876 031446 001015                RNE 20$           ; BRANCH IF YES
5877 031450 162701 000040          SUB @40,R1       ; DECREMENT PRIORITY
5878 031454                SETPRI R1                  ; SET PRIORITY
                    MOV R1,R0
                    TRAP C$SPRI
5879 031460 020127 000100          CMP R1,@PRI02    ; IS THE PRIORITY OVER 2?
5880 031464 004737 012146          CALL INSERT     ; SKIP BRANCH IF SFI FLAG SET
5881 031470 001362                BNE 10$         ; IF YES, BRANCH
5882 031472                ERRSOFT 401,E401,EER3       ; ERROR HANDLER
                    TRAP C$ERRSOFT
                    .WORD 401
                    .WORD E401
                    .WORD EER3
5883 031502                20$: CKLOOP                ; BRANCH TO BGNSEG IF LOE IS SET
5884 031504                SETPRI @PRI07              ; DISABLE INTERRUPT AT THIS POINT
                    TRAP C$CLP1
    
```

TEST 4: INTERNAL INTERRUPT LOGIC TEST - DIGITAL INPUT.

```

031504 012700 000340
031510 104441
5885 031512 062701 000040      ADD    #40,R1      ;CORRECT PRI. FOR CMP (-INC BY 1)
5886 031516 020137 003712      CMP    R1,PRI0    ;CHECK INTERRUPT PRIORITY
5887 031522 004737 012146      CALL  INSERT     ;SKIP BRANCH IF SFI FLAG SET
5888 031526 001430              BEQ    30$        ;BRANCH IF PRIORITY WAS CORRECT
5889                000005      .REPT  5         ;SET UP DATA FOR ERROR MESSAGES
5890                ASR    R1         ;...
5891                .ENDR              ;...
5892 031542 010137 004064      MOV    R1,BAD     ;SET UP DATA FOR ERROR MESSAGE
5893 031546 013737 003712 004062  MOV    PRI0,GOOD  ;...
5894                000005      .REPT  5         ;...
5895                ASR    GOOD       ;...
5896                .ENDR              ;...
5897 031600                ERRSOFT 402,E402,EER1 ;ERROR HANDLER
031600 104457
031602 000622
031604 032122
031606 007342
5898 031610                30$:  CKLOOP      ;BRANCH TO BGNSEG IF LOE IS SET
031610 104406                TRAP  C$CLP1
5899 031612 017737 152066 004064  MOV    @CSA,BAD   ;GET CSA CONTENTS
5900 031620 022737 140000 004064  CMP    #140000,BAD ;IE AND IR SHOULD BE SET
5901 031626 004737 012146      CALL  INSERT     ;SKIP BRANCH IF SFI FLAG SET
5902 031632 001407              BEQ    40$        ;BRANCH IF YES
5903 031634 012737 140000 004062  MOV    #140000,GOOD ;SET UP DATA FOR ERROR MESSAGES
5904 031642                ERRSOFT 403,E403,EER1 ;ERROR HANDLER
031642 104457
031644 000623
031646 032211
031650 007342
5905 031652                40$:  CKLOOP      ;BRANCH TO BGNSEG IF LOE IS SET
031652 104406                TRAP  C$CLP1
5906 031654 052777 100000 152022  BIS    #100000,@CSA ;CLEAR IR 15 BIT IN CSA
5907 031662 017737 152016 004064  MOV    @CSA,BAD   ;GET CSA CONTENTS
5908 031670 022737 040000 004064  CMP    #40000,BAD ;IS IR15 BIT CLEARED
5909 031676 004737 012146      CALL  INSERT     ;SKIP BRANCH IF SFI FLAG SET
5910 031702 001407              BEQ    50$        ;BRANCH IF YES
5911 031704 012737 040000 004062  MOV    #40000,GOOD ;SET UP DATA FOR ERROR MESSAGES
5912 031712                ERRSOFT 404,E404,EER1 ;ERROR HANDLER
031712 104457
031714 000624
031716 032310
031720 007342
5913 031722                50$:  ENDSEG
031722                10000$: TRAP  C$ESEG
031722 104405
5914 031724 005737 002242      TST    QVP        ;IS QUICK VERIFY PASS SELECTED?
5915 031730 001007              BNE    EXQV4     ;IF YES, EXIT TEST
5916 031732 005237 004144              INC    ITRCNT    ;ITERATION COUNTER + 1
5917 031736 023737 004146 004144  CMP    ITRDEF,ITRCNT ;DEFAULT ITERATION EXECUTED
5918 031744 001401              BEQ    EXQV4     ;IF YES, EXIT TEST
5919 031746 000600              BR    ITRAC4    ;IF NO, TEST ITERATION
5920 031750 005077 151730      EXQV4: CLR    @CSA  ;CLEAR CSA REGISTER
5921 031754                EXIT    TST      ;EXIT TETS
031754 104432
031756 000434                TRAP  C$EXIT
                                .WORD  L10035-

```

TEST 4: INTERNAL INTERRUPT LOGIC TEST - DIGITAL INPUT.

```

5922
5923 031760      045      123      062 TSHD4:: .NLIST BEX
5924 032040      116      117      040 E401:  .ASCIZ /S2#INTERNAL INTERRUPT TEST - DIGITAL INPUT.#N/
5925 032122      111      116      124 E402:  .ASCIZ /NO INTERRUPT AFTER SETTING EI * TST IR BIT IN CSA/
5926 032211      103      123      101 E403:  .ASCIZ /INTERUPT DID NOT OCCUR AT THE SELECTED PRIORITY LEVEL/
5927 032310      111      122      061 E404:  .ASCIZ /CSA REGISTER OF DIGITAL INPUT MODULE INCORRECT AFTER INTERRUPT/
5928
5929
5930
5931 032412      .LIST BEX
      032412      .EVEN
      032412 104401 .ENDTST

```

L10035: TRAP C#ETST

TEST 5: INTERNAL LOGIC TEST - ANALOGUE INPUT.

```

5933 .SBTTL TEST 5: INTERNAL LOGIC TEST - ANALOGUE INPUT.
5934 ;*****
5935 ; TEST 5 - INTERNAL LOGIC TEST - ANALOGUE INPUT.
5936 ;
5937 ;THIS TEST PERFORMS A PSEUDO CONVERSION ON EACH CHANNEL THAT IS FOUND, BY
5938 ;SETTING THE A/D START BIT AND THEN POLLING THE 'DONE' BIT TO CHECK THAT
5939 ;THE CONVERSION HAS BEEN COMPLETED WITHIN THE ALLOWED TIMEOUT PERIOD (AT
5940 ;LEAST 10MS).
5941 ;BEFORE A/D START IS SET, A CHECK IS MADE THAT THE DONE BIT IS CLEARED.
5942 ;THE ERROR CONDITIONS ARE ALSO CHECKED, BUT NO CHECK IS MADE ON THE RESULTING
5943 ;INPUTS OR ON THE GAIN SETTING.
5944 ;*****
5945 032414 BGNTST
5946 032414 004737 011714 CALL SELECT ;CALL SELECT ROUTINE
5947 032420 000404 .WORD 404 ;GIVE TEST PARAMETER (BASIC,AIP)
5948 032422 033204 TSHD5 ;GIVE TEST HEADER ADDRESS
5949 032424 103002 BCC 1$ ;BRANCH IF CARRY IS CLEARED
5950 032426 EXIT TST ;EXIT TEST IF CARRY WAS SET
5951 032432 005037 004144 1$: CLR ITRCNT ;CLEAR ITERATION COUNTER
5952 032436 005003 ITRAC5: CLR R3 ;CLEAR MAXIMUM CHANNEL COUNTER
5953 032440 005004 CLR R4 ;GET FIRST CHANNEL
5954 032442 104404 10$: BGNSEG
5955 032444 005777 151232 TST @DAT ;READ DAT TO CLEAR DONE+ERR IN CSA
5956 032450 005077 151230 CLR @CSA ;CLEAR CONTROL STATUS REGISTER
5957 032454 050477 151224 BIS R4,@CSA ;LOAD CHANNEL NUMBER
5958 032460 017737 151220 004064 MOV @CSA,BAD ;GET CSA CONTENTS
5959 032466 012737 100000 004062 MOV @100000,GOOD ;SET UP GOOD DATA
5960 032474 050437 004062 BIS R4,GOOD ;SAVE CHANNEL NUMBER IN GOOD
5961 032500 023737 004062 004064 CMP GOOD,BAD ;IS CHANNEL AVAILABLE AND DONE CLEARED
5962 032506 004737 012146 CALL INSERT ;SKIP BRANCH IF SFI FLAG SET
5963 032512 001404 BEQ 20$ ;BRANCH IF YES
5964 032514 ERRSOFT 501,E501,EER4 ;ERROR HANDLER
5965 032524 104406 20$: CKLOOP ;BRANCH BACK TO BGNSEG
5966 032526 052777 000001 151150 BIS #1,@CSA ;START A CONVERSION
5967 032534 012701 000024 MOV #20.,R1 ;SET UP TIME OUT COUNTER
5968 032540 004737 011514 30$: CALL WT500 ;WAIT 500 US
5969 032544 017737 151134 004064 MOV @CSA,BAD ;GET CSA CONTENTS
5970 032552 032737 000200 004064 BIT #200,BAD ;POLL THE DONE BIT
5971 032560 001015 BNE 40$ ;BRANCH IF DONE BIT WAS SET
5972 032562 005301 DEC R1 ;DECREMENT TIMEOUT COUNTER
5973 032564 004737 012146 CALL INSERT ;SKIP BRANCH IF SFI FLAG SET
5974 032570 001363 BNE 30$ ;BRANCH IF 4 MS NOT REACHED
5975 032572 012737 100200 004062 MOV #100200,GOOD ;SET UP DATA FOR ERROR MESSAGES
5976 032600 050437 004062 BIS R4,GOOD ;LOAD CHANNEL NUMBER INTO GOOD
5977 032604 ERRSOFT 502,E502,EER4 ;ERROR HANDLER
5978 032604 104457 TRAP C$ERSOFT
5979 032606 000766 .WORD 502
5980 032610 033345 .WORD E502

```

TEST 5: INTERNAL LOGIC TEST - ANALOGUE INPUT.

SEQ 0149

```

032612 007474
5978 032614          40$: CKLOOP          ;BRANCH TO BGNSEG          .WORD  EER4
032614 104406          TRAP  C$CLP1
5979 032616 052777 000001 151060  BIS  #1,@CSA          ;SET A/D START BIT TO GET AN ERR
5980 032624 017737 151054 004064  MOV  @CSA,BAD          ;GET CSA CONTENTS
5981 032632 012737 100240 004062  MOV  #100240,GOOD      ;SET UP GOOD DATA FOR COMPARISON
5982 032640 050437 004062          BIS  R4,GOOD          ;...
5983 032644 023737 004062 004064  CMP  GOOD,BAD          ;ERROR BIT SHOULD NOW BE SET
5984 032652 004737 012146          CALL INSERT          ;SKIP BRANCH IF SFI FLAG SET
5985 032656 001404          BEQ  50$              ;BRANCH IF YES
5986 032660          ERRSOFT 503,E503,EER4 ;ERROR HANDLER
032660 104457          TRAP  C$ERSOFT
032662 000767          .WORD  503
032664 033427          .WORD  E503
032666 007474          .WORD  EER4
5987 032670          50$: ENDSEG
032670          10000$: TRAP  C$ESEG
032670 104405          BGNSEG
5988 032672 017737 151002 004064  MOV  @DAT,BAD          ;CLEAR DONE + ERR BY READING DAT REG.
032672 104404          TRAP  C$BSEG
5989 032674 017737 150776 004064  MOV  @CSA,BAD          ;GET CSA CONTENTS
5990 032702 017737 150776 004064  MOV  #100000,GOOD      ;SET UP GOOD DATA FOR COMPARISON
5991 032710 012737 100000 004062  BIS  R4,GOOD          ;...
5992 032716 050437 004062          CMP  GOOD,BAD          ;DONE + ERR BIT SHOULD NOW BE CLEARED
5993 032722 023737 004062 004064  CALL INSERT          ;SKIP BRANCH IF SFI FLAG SET
5994 032730 004737 012146          BEQ  60$              ;BRANCH IF YES
5995 032734 001404          ERRSOFT 504,E504,EER4 ;ERROR HANDLER
032736 104457          TRAP  C$ERSOFT
032740 000770          .WORD  504
032742 033517          .WORD  E504
032744 007474          .WORD  EER4
5997 032746          60$: CKLOOP          ;BRANCH TO BGNSEG IF LOE IS SET
032746 104406          TRAP  C$CLP1
5998 032750 052777 000001 150726  BIS  #1,@CSA          ;SET A/D START BIT IN CSA
5999 032756 052777 000001 150720  BIS  #1,@CSA          ;SET A/D START A SECOND TIME
6000 032764 017737 150714 004064  MOV  @CSA,BAD          ;GET CSA CONTENTS
6001 032772 012737 100040 004062  MOV  #100040,GOOD      ;SET UP GOOD DATA FOR COMPARISON
6002 033000 050437 004062          BIS  R4,GOOD          ;...
6003 033004 023737 004062 004064  CMP  GOOD,BAD          ;ERROR BIT SHOULD BE SET
6004 033012 004737 012146          CALL INSERT          ;SKIP BRANCH IF SFI FLAG SET
6005 033016 001404          BEQ  70$              ;BRANCH IF YES
6006 033020          ERRSOFT 505,E505,EER4 ;ERROR HANDLER
033020 104457          TRAP  C$ERSOFT
033022 000771          .WORD  505
033024 033604          .WORD  E505
033026 007474          .WORD  EER4
6007 033030          70$: CKLOOP          ;BRANCH BACK TO BGNSEG (LOE SET)
033030 104406          TRAP  C$CLP1
6008 033032          ENDSEG
033032          10001$: TRAP  C$ESEG
033032 104405
6009 033034 017737 150642 004064  MOV  @DAT,BAD          ;CLEAR ERROR BIT IN CSA
6010 033042 005203          INC  R3                ;MARK CHANNEL NUMBER
6011 033044 022703 000017          CMP  #15.,R3          ;16 CHANNELS TESTED ?
6012 033050 002027          BGE  90$              ;BRANCH IF NOT
6013 033052 017737 150626 004064  MOV  @CSA,BAD          ;GET CSA CONTENTS

```

TEST 5: INTERNAL LOGIC TEST - ANALOGUE INPUT.

```

6014 033060 042737 100377 004064      BIC      #100377,BAD      ;MASK OUT UNUSED BITS
6015 033066 023727 004064 077400      CMP      BAD,#77400      ;HAVE WE REACHED LAST CHA. IN LAST BANK
6016 033074 103026                      BHIS     110$           ;EXIT IF YES
6017 033076 062704 000400                      ADD      #400,R4        ;INCREMENT CHANNEL NUMBER
6018 033102 005077 150576      80$:    CLR      @CSA          ;CLEAR OLD CHANNEL NUMBER
6019 033106 050477 150572                      BIS      R4,@CSA        ;LOAD NEW CHANNEL NUMBER
6020 033112 032777 100000 150564      BIT      #100000,@CSA   ;IS THE LOADED CHANNEL AVAILABLE ?
6021 033120 001407                      BEQ      100$           ;BRANCH IF NOT
6022 033122 005003                      CLR      R3             ;POINT TO FIRST CHANNEL IN BANK
6023 033124 000137 032442                      JMP      10$            ;IF YES GO TO TEST CHANNEL
6024 033130 062704 000400      90$:    ADD      #400,R4        ;GET NEXT CHANNEL
6025 033134 000137 032442                      JMP      10$            ;REPEAT WITH NEXT CHANNEL
6026 033140 062704 010000      100$:   ADD      #10000,R4       ;POINT TO NEXT CHANNEL BANK
6027 033144 020427 070000                      CMP      R4,#70000      ;HAVE WE REACHED THE LAST CHA.
6028 033150 101754                      BLOS     80$            ;BRANCH IF NOT
6029 033152 005737 002242      110$:   TST      QVP            ;IS QUICK VERIFY PASS SELECTED?
6030 033156 001010                      BNE      EXQV5          ;IF YES, EXIT TEST
6031 033160 005237 004144                      INC      ITRCNT         ;ITERATION COUNTER + 1
6032 033164 023737 004146 004144      CMP      ITRDEF,ITRCNT  ;DEFAULT ITERATION EXECUTED
6033 033172 001402                      BEQ      EXQV5          ;IF YES, EXIT TEST
6034 033174 000137 032436                      JMP      ITRACS         ;IF NO, TEST ITERATION
6035 033200      EXQV5: EXIT      TST      ;EXIT TEST
      033200 104432
      033202 000470

```

TRAP C\$EXIT
.WORD L10036-

```

6036
6037
6038 033204      045      123      062 TSHD5: .NLIST BEX
6039 033261      103      123      101 E501: .ASCIZ /#S2#INTERNAL LOGIC TEST - ANALOGUE INPUT.#N/
6040 033345      104      117      116 E502: .ASCIZ /CSA CONTENTS INCORRECT AFTER READ DAT AND CLEAR CSA/
6041 033427      105      122      122 E503: .ASCIZ /DONE BIT IN CSA NOT SET (TIMEOUT) AFTER A/D START/
6042 033517      105      122      122 E504: .ASCIZ \ERR BIT IN CSA NOT SET AFTER A/D START WHEN DONE IS SET\
6043 033604      105      122      122 E505: .ASCIZ /ERR, DONE BIT IN CSA NOT CLEARED AFTER READ DAT REG./
6044
6045
6046 033672
      033672
      033672 104401

```

L10036: TRAP C\$ETST

TEST 6: INTERNAL INTERRUPT LOGIC TEST - ANALOGUE INPUT.

```

6048 .SBTTL TEST 6: INTERNAL INTERRUPT LOGIC TEST - ANALOGUE INPUT.
6049 ;*****
6050 ; TEST 6 - INTERNAL INTERRUPT LOGIC TEST - ANALOGUE INPUT.
6051 ;
6052 ; THIS TEST CHECKS THAT THE INTERNAL INTERRUPT LOGIC IS ABLE TO CAUSE A
6053 ; DONE AND AN ERROR INTERRUPT USING THE VECTOR AND PRIORITY LEVEL SELECTED
6054 ; IN THE START UP QUESTIONS.
6055 ; THE DONE INTERRUPT IS GENERATED BY SETTING THE INTERRUPT ENABLE
6056 ; BIT AND THE A/D START BIT.
6057 ; THE ERROR INTERRUPT IS GENERATED BY SETTING THE A/D START BIT TWICE.
6058 ; A TIME OUT IS GENERATED IF THE INTERRUPT HAS NOT OCCURED WITHIN THE
6059 ; ALLOWED TIMEOUT PERIOD.
6060 ;*****
6061 033674          BGNTST
        033674
6062 033674 004737 011714          CALL    SELECT          T6::
6063 033700 000404          .WORD   404             ;CALL SELECT ROUTINE
6064 033702 034524          TSHD6          ;GIVE TEST PARAMETER (BASIC+AIP)
6065 033704 103002          BCC     1$           ;GIVE TEST HEADER ADDRESS
6066 033706          EXIT    TST             ;BRANCH IF CARRY IS CLEARED
        033706 104432          ;EXIT TEST IF CARRY WAS SET
        033710 001242          TRAP    C$EXIT
6067 033712 005037 004144          1$:   CLR     ITRCNT          ;CLEAR ITERATION COUNTER
        033716          SETVEC  VEC,#INTSR,#PRI07 ;SET DONE VECTOR AND SERVICE ROUTINE
        033716 012746 000340          MOV     #PRI07,-(SP)
        033722 012746 017276          MOV     #INTSR,-(SP)
        033726 013746 003710          MOV     VEC,-(SP)
        033732 012746 000003          MOV     #3,-(SP)
        033736 104437          TRAP   C$SVEC
        033740 062706 000010          ADD    #10,SP
6069 033744 013702 003710          MOV    VEC,R2          ;GET DONE VECTOR
6070 033750 062702 000004          ADD    #4,R2          ;CREATE ERROR VECTOR (VEC+4)
6071 033754          SETVEC  R2,#INTSR2,#PRI07 ;SET ERROR VEC. AND SERVICE ROUTINE
        033754 012746 000340          MOV     #PRI07,-(SP)
        033760 012746 017306          MOV     #INTSR2,-(SP)
        033764 010246          MOV     R2,-(SP)
        033766 012746 000003          MOV     #3,-(SP)
        033772 104437          TRAP   C$SVEC
        033774 062706 000010          ADD    #10,SP
6072 034000          ITRAC6: BGNSEG
        034000 104404          TRAP   C$BSEG
6073 034002          SETPRI  #PRI07          ;DISABLE INTERRUPT AT THIS POINT
        034002 012700 000340          MOV     #PRI07,R0
        034006 104441          TRAP   C$SPRI
6074 034010 005037 004150          CLR    INTFLA          ;CLEAR INTERRUPT FLAG
6075 034014 005037 004152          CLR    INTFL2         ;CLEAR ERROR INTERRUPT FLAG
6076 034020 005777 147656          TST    @DAT           ;READ DAT TO CLEAR ERR.DONE IN CSA
6077 034024 005077 147654          CLR    @CSA          ;CLEAR CONTROL STATUS REGISTER
6078 034030 012701 000340          MOV    #PRI07,R1     ;:::LOAD R1 WITH PRIORITY NUMBER
6079 034034 012777 000100 147642          MOV    #100,@CSA     ;:::SET INTERRUPT ENABLE BIT
6080 034042 052777 000001 147634          BIS    #1,@CSA       ;:::START THE CONVERSION
6081 034050 012703 000036          MOV    #30.,R3      ;:::SET UP TIME OUT COUNTER
6082 034054 004737 011514          CALL   WT500         ;:::WAIT 500 US
6083 034060 032777 000200 147616          BIT    #200,@CSA     ;:::POLL THE DONE BIT
6084 034066 001010          BNE    20$           ;:::BRANCH IF DONE BIT WAS SET
6085 034070 005303          DEC    R3            ;:::DECREMENT TIMEOUT COUNTER
6086 034072 004737 012146          CALL   INSERT        ;:::SKIP BRANCH IF SFI FLAG SET

```


TEST 6: INTERNAL INTERRUPT LOGIC TEST - ANALOGUE INPUT.

```

6121                                     ;NOW WE TEST THE ERROR INTERRUPT
6122
6123 034316                               BGNSEG
6124 034316 104404
6125 034320 005777 147356                TST    @DAT                TRAP    C$BSEG
6126 034324 005737 004152                TST    INTFL2            ;READ DAT TO CLEAR ERR+DONE IN CSA
6127 034330 004737 012146                CALL   INSERT           ;ERROR INTERRUPT OCCURRED ?
6128 034334 001404                        BEQ    60$              ;SKIP BRANCH IF SFI FLAG SET
6128 034336                               ERRSOFT 605,E605,EERG    ;BRANCH IF NO
6128 034336 104457                        ;ERROR HANDLER
6128 034340 001135                        TRAP   C$ERSOFT
6128 034342 035052                        .WORD 605
6128 034344 007724                        .WORD E605
6129 034346                               .WORD EERG
6129 034346 104406                        60$: CKLOOP            ;
6130 034350 013703 000036                MOV    30.,R3           ;SET TIMEOUT COUNTER
6131 034354 052777 000001 147322        BIS    #1,@CSA         ;START A CONVERSION (SET A/D BIT)
6132 034362 004737 011514                CALL   WT500           ;WAIT 500 US
6133 034366 032777 000200 147310        BIT    #200,@CSA      ;POLL THE DONE BIT
6134 034374 001010                        BNE    80$            ;BRANCH IF DONE BIT WAS SET
6135 034376 005303                        DEC    R3              ;DECREMENT TIMEOUT COUNTER
6136 034400 004737 012146                CALL   INSERT           ;SKIP BRANCH IF SFI FLAG SET
6137 034404 001366                        BNE    70$            ;BRANCH IF 4 MS NOT REACHED
6138 034406                               ERRSOFT 606,E601,EERG    ;ERROR HANDLER
6138 034406 104457                        TRAP   C$ERSOFT
6138 034410 001136                        .WORD 606
6138 034412 034605                        .WORD E601
6138 034414 007724                        .WORD EERG
6139 034416                               80$: CKLOOP            ;
6139 034416 104406
6140 034420 052777 000001 147256        BIS    #1,@CSA         ;SET A/D START A SECOND TIME
6141 034426 004737 011522                CALL   WT25           ;WAIT 25 US FOR INTERRUPT
6142 034432 005737 004152                TST    INTFL2         ;ERROR INTERRUPT OCCURRED ?
6143 034436 004737 012146                CALL   INSERT           ;SKIP BRANCH IF SFI FLAG SET
6144 034442 001004                        BNE    90$            ;BRANCH IF YES
6145 034444                               ERRSOFT 607,E606,EERG    ;ERROR HANDLER
6145 034444 104457                        TRAP   C$ERSOFT
6145 034446 001137                        .WORD 607
6145 034450 035102                        .WORD E606
6145 034452 007724                        .WORD EERG
6146 034454                               90$: CKLOOP            ;
6146 034454 104406
6147 034456 005777 147220                TST    @DAT                TRAP   C$CLP1
6148 034462                               ENDSEG                ;READ DAT TO CLEAR ERR+DONE IN CSA
6148 034462 104405
6149 034464                               SETPRI #PRI07          ;DISABLE INTERRUPTS
6149 034464 012700 000340                TRAP   C$ESEG
6149 034470 104441                        MOV    #PRI07,R0
6149 034472 005737 002242                TRAP   C$SPRI
6150 034476 001010                        TST    QVP             ;IS QUICK VERIFY PASS SELECTED?
6151 034500 005237 004144                BNE    EXQV6          ;IF YES, EXIT TEST
6152 034504 023737 004146 004144        INC    ITRCNT         ;ITERATION COUNTER + 1
6153 034512 000137 034000                CMP    ITRDEF,ITRCNT  ;DEFAULT ITERATION EXECUTED
6154 034514 000137 034000                BEQ    EXQV6          ;IF YES, EXIT TEST
6155 034520 104432                JMP    ITRAC6         ;IF NO, TEST ITERATION
6156 034520 104432                EXQV6: EXIT          TST
6156 034520 104432                ;
6156 034520 104432                TRAP   C$EXIT

```

TEST 6: INTERNAL INTERRUPT LOGIC TEST - ANALOGUE INPUT.

```

034522 000430 .WORD L10037-.
6157
6158
6159 034524 045 123 062 TSHD6: .NLIST BEX
6160 034605 124 111 115 E601: .ASCIZ /S2#AINTERNAL INTERRUPT TEST - ANALOGUE INPUT.#N/
6161 034670 116 117 040 E602: .ASCIZ \TIME OUT - DONE BIT IN CSA NOT SET AFTER A/D START\
6162 034751 120 122 111 E603: .ASCIZ \NO INTERRUPT AFTER SETTING EI + A/D START IN CSA\
6163 035002 103 123 101 E604: .ASCIZ /PRIORITY LEVEL INCORRECT/
6164 035052 105 122 122 E605: .ASCIZ /CSA CONTENTS INCORRECT AFTER CONVERSION/
6165 035102 116 117 040 E606: .ASCIZ /ERROR INTERRUPT OCCURED/
6166 .LIST BEX
6167 .EVEN
6168 035152 .ENDTST
035152
035152 104401 L10037: TRAP C$ETST

```

TEST 7: INTERNAL LOGIC TEST - ANALOGUE OUTPUT.

```

6170 .SBTTL TEST 7: INTERNAL LOGIC TEST - ANALOGUE OUTPUT.
6171 ;*****
6172 ; TEST 7 - INTERNAL LOGIC TEST - ANALOGUE OUTPUT.
6173 ;
6174 ; THIS TEST PERFORMS A CONVERSION ON EACH AVAILABLE CHANNEL, LOADING THE
6175 ; DAT REGISTER AND CHECKING THAT THE CONVERSIONS ARE COMPLETED WITHIN THE
6176 ; ALLOWED TIMEOUT PERIOD ( READING THE READY BIT AFTER 150 US).
6177 ;*****
6178 035154 BGNTST
        035154
6179 035154 004737 011714 CALL SELECT ;CALL SELECT ROUTINE
6180 035160 000410 .WORD 410 ;GIVE TEST PARAMETER (BASIC.AOP)
6181 035162 035424 TSHD7 ;GIVE TEST HEADER ADDRESS
6182 035164 103515 BCS EXQV7 ;IF CARRY IS SET, EXIT TEST
6183 035166 005037 004144 CLR ITRCNT ;CLEAR ITERATION COUNTER
6184 035172 005002 ITRAC7: CLR R2 ;LOAD R2 WITH FIRST CHAN. NUMBER
6185 035174 012704 012525 MOV #12525,R4 ;SET UP DATA FOR DATA REGISTER
6186 035200 104404 10$ : BGNSEG
        035200 104404 TRAP C$BSEG
6187 035202 042777 003400 146474 BIC #3400,@CSA ;CLEAR CHANNEL BITS
6188 035210 050277 146470 BIS R2,@CSA ;LOAD CHANNEL NUMBER
6189 035214 017737 146464 004064 MOV @CSA,BAD ;GET CSA CONTENTS
6190 035222 012737 100200 004062 MOV #100200,GOOD ;LOAD GOOD DATA FOR COMPARISON
6191 035230 050237 004062 BIS R2,GOOD ;...
6192 035234 042737 000040 004064 BIC #40,BAD ;MASK OUT OPL BIT
6193 035242 023737 004062 004064 CMP GOOD,BAD ;CHAVAI,READY SHOULD BE SET
6194 035250 004737 012146 CALL INSERT ;SKIP BRANCH IF SFI FLAG SET
6195 035254 001404 BEQ 20$ ;BRANCH IF YES
6196 035256 ERRSOFT 701,E701,EER1 ;ERROR HANDLER
        035256 104457 TRAP C$ERSOFT
        035260 001275 .WORD 701
        035262 035502 .WORD E701
        035264 007342 .WORD EER1
6197 035266 20$ : ENDSEG ;BRANCH TO BGNSEG IF LOE IS SET
        035266 104405 10000$: TRAP C$ESEG
6198 035270 006104 ROL R4 ;CHANGE DATA FOR LOAD DAT REGISTER
6199 035272 BGNSEG
        035272 104404 TRAP C$BSEG
6200 035274 010477 146402 MOV R4,@DAT ;LOAD DATA REGISTER
6201 035300 012703 000006 MOV #6,R3 ;LOAD WAIT COUNTER
6202 035304 004737 011522 30$ : CALL WT25 ;WAIT 25 US
6203 035310 005303 DEC R3 ;DECREMENT COUNTER
6204 035312 001374 BNE 30$ ;BRANCH IF NOT ZERO
6205 035314 017737 146364 004064 MOV @CSA,BAD ;GET CSA CONTENTS
6206 035322 042737 000040 004064 BIC #40,BAD ;MASK OUT OPL BIT
6207 035330 023737 004062 004064 CMP GOOD,BAD ;COMPARE GOOD AND BAD
6208 035336 004737 012146 CALL INSERT ;SKIP BRANCH IF SFI FLAG SET
6209 035342 001404 BEQ 40$ ;BRANCH IF YES
6210 035344 ERRSOFT 702,E702,EER1 ;ERROR HANDLER
        035344 104457 TRAP C$ERSOFT
        035346 001276 .WORD 702
        035350 035606 .WORD E702
        035352 007342 .WORD EER1
6211 035354 40$ : ENDSEG ;BRANCH TO BGNSEG IF LOE IS SET
        035354 104405 10001$: TRAP C$ESEG
        035354 104405

```

TEST 7: INTERNAL LOGIC TEST - ANALOGUE OUTPUT.

```

6212 035356 000302          SWAB R2          ;SWAB HIGH AND LOW BYTE
6213 035360 120227 000003  CMPB R2,#3      ;ALL 4 CHANNELS EXECUTED?
6214 035364 001403          BEQ 60$         ;BRANCH IF YES
6215 035366 005202          INC R2          ;NEXT CHANNEL NUMBER
6216 035370 000302          SWAB R2          ;SWAB LOW AND HIGH BYTE
6217 035372 000702          BR 10$         ;PERFORM A CONV. WITH NEXT CHAN.
6218 035374 005737 002242 60$: TST QVP         ;IS QUICK VERIFY PASS SELECTED?
6219 035400 001007          BNE EXQV7      ;IF YES EXIT TEST
6220 035402 005237 004144  INC ITRCNT     ;ITERATION COUNTER + 1
6221 035406 023737 004146 004144  CMP ITRDEF,ITRCNT ;DEFAULT ITERATION EXECUTED
6222 035414 001401          BEQ EXQV7      ;IF YES, EXIT TEST
6223 035416 000665          BR ITRAC7      ;IF NO, TEST ITERATION
6224 035420          EXQV7: EXIT TST          ;
        035420 104432          TRAP C$EXIT
        035422 000274          .WORD L10040-.
6225          .NLIST BEX
6226 035424 045 123 062 TSHD7: .ASCIZ /#S2#AINTERNAL LOGIC TEST - ANALOGUE OUTPUT.#N/
6227 035502 103 123 101 E701: .ASCII /CSA CONTENTS INCORRECT AFTER LOADING NEW CHANNEL /<12><15>
6228 035565 050 111 107 .ASCIZ /(IGNORE OPL BIT)/
6229 035606 103 123 101 E702: .ASCII /CSA CONTENTS INCORRECT AFTER LOADING DAT REG./<12><15>
6230 035665 122 105 101 .ASCIZ /READY TIMEOUT WAS 150US/
6231          .LIST BEX
6232          .EVEN
6233 035716          ENDTST
        035716          L10040: TRAP C$ETST
        035716 104401
6234
6240
6241
6253

```

HARDWARE TESTS MACRO M1200 26-OCT-83 15:22 PAGE 68

TEST 7: INTERNAL LOGIC TEST - ANALOGUE OUTPUT.

SEQ 0157

6261 035720
6262

ENDMOD

TEST 7: INTERNAL LOGIC TEST - ANALOGUE OUTPUT.

```

6266                                     .TITLE HARDWARE TESTS
6312 035720                             BGNMOD
6313
6314                                     .SBTTL TEST 8: DIGITAL INPUT LOOPBACK TEST WITH SELECTABLE DATA PATTERN PAIR.
6315 ;*****
6316 ; TEST 8 - DIGITAL INPUT LOOPBACK WITH SELECTABLE DATA PATTERN PAIR.
6317 ;
6318 ; DATA TRANSFERS ARE MADE BETWEEN THE MODULE UNDER TEST (IN THIS
6319 ; TEST A DIGITAL INPUT) AND THE MODULE SPECIFIED AS THE "OTHER" IN
6320 ; THE HARDWARE QUESTIONS (IN THIS TEST A DIGITAL OUTPUT). IF LOW
6321 ; LEVEL IS NOT SELECTED, ONE HUNDRED CYCLES ARE PERFORMED USING
6322 ; THE DATA PAIR AND DEBOUNCE PERIOD SPECIFIED IN THE SOFTWARE
6323 ; AND HARDWARE QUESTIONS. A DEBOUNCE PERIOD OF 10 MILLISECONDS IS
6324 ; THEN USED TO CHECK THAT THE DAT REGISTER DOES NOT CHANGE IN THE
6325 ; FIRST 2.5 MILLISECONDS.
6326 ;
6327 ; IF LOW LEVEL IS SELECTED, ONLY 4 CYCLES ARE PERFORMED AND THE
6328 ; DEBOUNCE FUNCTION IS NOT CHECKED.
6329 ;
6330 ; BY DEFAULT, THE DATA PAIR IS ALL ZEROS AND ALL ONES, BUT ANY
6331 ; PATTERN PAIR CAN BE SELECTED. IF THE TWO PATTERNS ARE IDENTICAL,
6332 ; A FIXED OUTPUT WILL RESULT.
6333 ;*****
6334 035720                             BGNTST 8.
6335 035720 004737 011714                CALL SELECT          ;CALL SELECT ROUTINE
6336 035724 002001                        .WORD 2001          ;GIVE TEST PARAMETER
6337 035726 036570                        TSMDB              ;GIVE TEST HEADER ADDRESS
6338 035730 103002                        BCC 1$             ;IF CARRY IS CLEARED BRANCH
6339 035732                        EXIT TST                  ;IF CARRY IS SET, EXIT TEST
6340 035732 104432                        TRAP C$EXIT
6341 035734 001140                        .WORD L10041-.
6342 035736 005037 004144                1$: CLR ITRCNT      ;CLEAR ITERATION COUNTER
6343 035742 013737 004154 036566        ITRAB: MOV CYCLE,SAVB ;LOAD CYCLE COUNTER
6344 035750 005002                        10$: CLR R2         ;FLAG FOR PATTERN CHOICE
6345 035752 005737 003716                TST LOWLVL        ;IS LOW LEVEL SELEC. IN P-TABLE?
6346 035756 001402                        BEQ 20$           ;BRANCH IF NO
6347 035760 000137 036360                JMP LLPAB         ;JUMP TO LOW LEVEL PART
6348 035764 013705 003720                20$: MOV DBOUNC,R5 ;GET DEBOUNCE FROM P TABLE
6349 035770 005205                        INC R5            ;CREATE DEBOUNCE VALUE FOR MOD
6350 035772 042705 177774                BIC @177774,R5   ;ONLY BITS 1 AND 2 ARE USED
6351 035776 022737 000003 003720        CMP @3,DBOUNC    ;ALL DEBOUNCE PERIODS SELECTED ?
6352 036004 001002                        BNE 30$          ;BRANCH IF NO
6353 036006 012705 000001                MOV @1,R5        ;GET FIRST DEBOUNCE
6354 036012 110577 145662                30$: MOVB R5,@MOD ;LOAD INPUT DEBOUNCE INTO MOD REGISTER
6355 036016 052777 000100 145654        BIS @100,@MOD    ;SET MODULE LED BIT AGAIN
6356 036024 116503 004172                MOVB DBTAB(R5),R3 ;GET DEBOUNCE WAIT VALUE FROM TABLE
6357 036030 013701 002236                40$: MOV PAT1,R1  ;GET FIRST PATTERN FROM P TABLE
6358 036034                        50$: BGNSEG
6359 036034 104404                        TRAP C$BSEG
6360 036036 010304                        MOV R3,R4        ;LOAD WAIT COUNTER
6361 036040 012737 001441 007200        MOV @801.,ERRNBR ;ERROR NUMBER FOR WRITE SUBROU.
6362 036046 004737 016122                CALL WRITE       ;CALL WRITE ROUTINE
6363 036052 103002                        BCC 60$          ;BRANCH IF NO ADDR. ERROR FOUND
6364 036054                        EXIT TST         ;EXIT TEST IF ADDRESSING ERROR
6365 036056 104432                        TRAP C$EXIT
6366 036056 001016                        .WORD L10041-.
    
```


TEST 8: DIGITAL INPUT LOOPBACK TEST WITH SELECTABLE DATA PATTE

```

6362 036060 063704 004166      60$:  ADD    OUTDE1,R4      ;ADD OPTO OUTPUT DEBOUNCE
6363 036064 032777 000001 145630  BIT    #1,80THMOD    ;IS DEBOUNCE OF OUTPUT 500 US ?
6364 036072 001002                BNE    70$           ;BRANCH IF YES
6365 036074 063704 004170                ADD    OUTDE2,R4      ;IF NO ADD RLEAY DEBOUNCE
6366 036100 004737 011514      70$:  JSR    PC,WT500      ;WAIT 500 US
6367 036104 005304                DEC    R4             ;DECREMENT DEBOUNCE WAIT COUNTER
6368 036106 001374                BNE    70$           ;BRANCH IF COUNTER IS NOT ZERO
6369 036110 017737 145566 004064  MOV    @DAT,BAD       ;READ DIGITAL INPUT PATTERN
6370 036116 020137 004064  CMP    R1,BAD         ;CMP READ AND LOADED DATA
6371 036122 004737 012146  CALL  INSERT          ;SKIP BRANCH IF 'SFI' IS SET
6372 036126 001406                BEQ    80$           ;BRANCH IF EQUAL
6373 036130 010137 004062  MOV    R1,GOOD        ;GET GOOD PATTERN
6374 036134                ERRSOF 802,E802,EER2 ;ERROR HANDLER
        036134 104457
        036136 001442
        036140 036665
        036142 007400
6375 036144      80$:  ENDSEG
        036144
        036144 104405
6376 036146 005702                TST    R2             ;IS THE 2ND PATTERN USED
6377 036150 001004                BNE    90$           ;BRANCH IF YES
6378 036152 005202                INC    R2             ;INC FLAG TO SHOW 2ND PATTERN
6379 036154 013701 002240  MOV    PAT2,R1        ;GET SECOND PATTERN FROM P TABLE
6380 036160 000725                BR     50$           ;BRANCH TO OUTPUT LOOP
6381 036162 022737 000003 003720 90$:  CMP    #3,DBOUNC     ;ALL DEBOUNCE SELECTED?
6382 036170 001004                BNE    100$          ;BRANCH IF NO
6383 036172 005205                INC    R5             ;POINT TO NEXT DEBOUNCE VALUE
6384 036174 022705 000004  CMP    #4,R5         ;ALL DEBOUNCE PERIODS EXECUTED
6385 036200 001304                BNE    30$           ;BRANCH IF NO
6386 036202 005337 036566 100$: DEC    SAV8       ;DECREMENT CYCLE COUNTER
6387 036206 001260                BNE    10$           ;BRANCH IF NO ZERO
6388
6389
6390
6391 036210      PSUE8: BGNSEG
        036210 104404
6392 036212 005001                CLR    R1             ;GET VALUE FOR WRITE ROUTINE
6393 036214 012704 000062  MOV    #50.,R4        ;WAIT COUNTER FOR CLEAR DAT REG.
6394 036220 012737 001443 007200  MOV    #803.,ERRNBR   ;LOAD ERROR NUMBER
6395 036226 004737 016122  JSR    PC,WRITE       ;CLEAR OUTPUT DAT REGISTER
6396 036232 103002                BCC   10$           ;BRANCH IF NO ADDR. ERROR FOUND
6397 036234                EXIT  TST            ;EXIT TEST IF ADDRESSING ERROR
        036234 104432
        036236 000636
6398 036240 004737 011514      10$: JSR    PC,WT500      ;...
6399 036244 005304                DEC    R4             ;...
6400 036246 001374                BNE    10$           ;...
6401 036250 052777 000003 145422  BIS    #3,@MOD        ;LOAD INPUT DEBOUNCE OF 10 MS
6402 036256 012701 177777  MOV    #177777,R1     ;LOAD VALUE FOR OUTPUT DAT REG.
6403 036262 012704 000004  MOV    #4,R4          ;LOAD (2MS) WAIT COUNTER
6404 036266 012737 001444 007200  MOV    #804.,ERRNBR   ;LOAD ERROR NUMBER
6405 036274 004737 016122  JSR    PC,WRITE       ;SEND DATA
6406 036300 103002                BCC   20$           ;BRANCH IF NO ADDR. ERROR FOUND
6407 036302                EXIT  TST            ;EXIT TEST IF ADDRESSING ERROR
        036302 104432
        036304 000570
        TRAP    C$ESEG
        .WORD  802
        .WORD  E802
        .WORD  EER2
        TRAP    C$BSEG
        .WORD  L10041..
        TRAP    C$EXIT
        .WORD  L10041..

```

TEST 8: DIGITAL INPUT LOOPBACK TEST WITH SELECTABLE DATA PATTE

```

6408 036306 004737 011514          20$: JSR    PC,WT500          ;WAIT 500 US
6409 036312 005304                   DEC    R4                ;...
6410 036314 001374                   BNE    20$              ;BRANCH IF NOT FINISHED
6411 036316 017737 145360 004064    MOV    @DAT,BAD        ;READ INPUT DATA REG.
6412 036324 020137 004064          CMP    R1,BAD         ;DATA SHOULD NOT BE CHANGED
6413 036330 004737 012146          CALL   INSERT         ;SKIP BRANCH IF 'SFI' IS SET
6414 036334 001006                   BNE    30$              ;BRANCH IF OK
6415 036336 005037 004062          CLR    GOOD           ;SET UP DATA FOR ERROR HANDLER
6416 036342                   ERRSOFT 805,E803,EER2 ;ERROR HANDLER
                                TRAP    C$ERSOFT
                                .WORD   805
                                .WORD   E803
                                .WORD   EER2
6417 036352                   30$: ENDSEG
                                ;
                                10001$: TRAP    C$ESEG
6418 036354 000137 036534          JMP    QVR8           ;SKIP LOW LEVEL PART
6419
6420
6421
6422 036360 013737 004156 036566    LLPA8: MOV    CYCLLS,SAV8 ;LOAD LOW LEVEL CYCLE COUNTER
6423 036366 052777 000002 145304    BIS    #2,@MOD        ;LOAD SMS INPUT DEBOUNCE
6424 036374 042777 000001 145276    BIC    #1,@MOD
6425 036402 052777 000010 145270    BIS    #10,@MOD
6426 036410 005002                   10$: CLR    R2          ;...
6427 036412 013701 002236          MOV    PAT1,R1        ;SWITCH INPUT MODULE TO LLS
6428 036416 013704 004162          20$: MOV    LLWC,R4    ;FLAG FOR PATTERN MARKING
6429 036422 012737 001446 007200    MOV    #806.,ERRNBR  ;GET FIRST PATTERN
6430 036430                   BGNSEG                ;GET LOW LEVEL DEBOUNCE TIME
6431 036432 004737 016122          CALL   WRITE          ;LOAD ERROR NUMBER
6432 036436 103002                   BCC    30$            ;CALL WRITE ROUTINE
6433 036440                   EXIT    TST           ;BRANCH IF NO ADDR. ERROR FOUND
6434 036444 004737 011514          30$: JSR    PC,WT500   ;EXIT TEST IF ADDRESSING ERROR
6435 036450 005304                   DEC    R4             ;TRAP    C$BSEG
6436 036452 001374                   BNE    30$            ;BRANCH IF NO ADDR. ERROR FOUND
6437 036454 017737 145222 004064    MOV    @DAT,BAD        ;EXIT TEST IF ADDRESSING ERROR
6438 036462 020137 004064          CMP    R1,BAD         ;TRAP    C$EXIT
6439 036466 004737 012146          CALL   INSERT         ;.WORD   L10041..
6440 036472 001406                   BEQ    40$            ;WAIT 500 US
6441 036474 010137 004062          MOV    R1,GOOD        ;DECREMENT WAIT COUNTER
6442 036500                   ERRSOFT 807,E802,EER2 ;BRANCH IF COUNTER IS NOT ZERO
6443 036510                   40$: ENDSEG          ;READ DIGITAL INPUT PATTERN
6444 036512 005702                   TST    R2             ;CMP READ AND LOADED DATA
6445 036514 001004                   BNE    50$            ;SKIP BRANCH IF 'SFI' IS SET
6446 036516 005202                   INC    R2             ;BRANCH IF EQUAL
6447 036520 013701 002240          MOV    PAT2,R1        ;GET GOOD PATTERN
6448 036524 000734                   BR     20$            ;GET LOW LEVEL DEBOUNCE TIME
6449 036526 005337 036566          50$: DEC    SAV8      ;LOAD ERROR NUMBER
                                TRAP    C$ERSOFT
                                .WORD   807
                                .WORD   E802
                                .WORD   EER2
                                10002$: TRAP    C$ESEG
; WAS THE 2ND PATTERN USED?
; BRANCH IF YES
; INC FLAG TO SHOW 2ND PATTERN
; GET SECOND PATTERN FROM P TABLE
; BRANCH TO OUTPUT LOOP
; IS CYCLE COUNTER OVER?

```

TEST 8: DIGITAL INPUT LOOPBACK TEST WITH SELECTABLE DATA PATTE

```

6450 036532 001326
6451 036534 005737 002242          QVR8:  BNE 10$          ;BRANCH IF NO
6452 036540 001010                    TST QVP          ;IS QUICK VERIFY PASS SELECTED?
6453 036542 005237 004144                    BNE EXQV8       ;IF YES, EXIT TEST
6454 036546 023737 004146 004144          INC ITRCNT      ;ITERATION COUNTER . 1
6455 036554 001402                    CMP ITRDEF,ITRCNT ;DEFAULT ITERATION EXECUTED
6456 036556 000137 035742          BEQ EXQV8       ;IF YES, EXIT TEST
6457 036562 104432          EXQV8: EXIT ITR8 ;IF NO, TEST ITERATION
      036564 000310          TST            ;EXIT TEST
                                TRAP C$EXIT
                                .WORD L10041-.
6458
6459
6460 036566 000000          SAV8: .WORD          ;
6461
6462
6463          .NLIST BEX
6464 036570 045 123 062 TSHD8: .ASCIZ /*S2#ADIGITAL INPUT LOOPBACK TEST - PATTERN PAIR SELECTABLE#N/
6465 036665 114 117 120 E802: .ASCIZ /LOPPED DATA IN DATA REGISTER OF DIGITAL INPUT MODULE INCORRECT/
6466 036764 104 105 102 E803: .ASCII /DEBOUNCE FUNCTION ERROR/<12><15>
6467 037015 104 101 124          .ASCIZ /DATA ACCEPTED BEFORE LOADED DEBOUNCE FINISHED/
6468          .LIST BEX
6469          .EVEN
6470
6471 037074          ENDTST
      037074
      037074 104401          L10041: TRAP C$ETST
    
```

TEST 9: DIGITAL OUTPUT LOOPBACK TEST WITH SELECTABLE DATA PATT

```

6473 .SBTTL TEST 9: DIGITAL OUTPUT LOOPBACK TEST WITH SELECTABLE DATA PATTERN PAIR.
6474 ;*****
6475 ; TEST 9 - DIGITAL OUTPUT LOOPBACK TEST - SELECTABLE DATA PATTERN PAIR.
6476 ;
6477 ;DATA TRANSFERS ARE MADE BETWEEN THE MODULE UNDER TEST (IN THIS TEST A
6478 ;DIGITAL OUTPUT) AND THE MODULE SPECIFIED AS THE 'OTHER' IN THE HARDWARE
6479 ;QUESTIONS (IN THIS TEST A DIGITAL INPUT). SEVERAL CYCLES
6480 ;ARE PERFORMED USING THE DATA PAIR FROM THE SW P-TABLE AND A DEBOUNCE
6481 ;PERIOD FOUND IN THE MOD REGISTER OF THE DIGITAL OUTPUT MODULE .
6482 ;FOR THE INPUT MODULE THE DEBOUNCE OF (500US) IS USED.
6483 ;BY DEFAULT, THE DATA PAIR IS ALL ZEROS AND ALL ONES,
6484 ;BUT ANY PATTERN PAIR CAN BE SELECTED. IF THE TWO PATTERNS ARE IDENTICAL,
6485 ;A FIXED OUTPUT WILL RESULT.
6486 ;IF A RELAY OUTPUT MODULE IS USED, A CHECK IS ALSO MADE THAT THE
6487 ;DAT REGISTER OF THE DIGITAL INPUT MODULE WILL NOT BE CHANGED
6488 ;TOO FAST AFTER LOADING THE OUTPUT DATA REGISTER.
6489 ;*****
6490 037076 BGNTS1
6491 037076 004737 011714 CALL SELECT T9::
6492 037102 002002 .WORD 2002 ;CALL SELECT ROUTINE
6493 037104 037616 TSM09 ;GIVE TEST PARAMETER
6494 037106 103002 BCC 1$ ;GIVE TEST HEADER ADDRESS
6495 037110 104432 EXIT TST ;IF CARRY IS CLEARED, BRANCH
;IF CARRY IS SET, EXIT TEST
;*****
6496 037114 005037 004144 1$: CLR ITRCNT ;CLEAR ITERATION COUNTER
6497 037120 013705 004154 ITR9: MOV CYCLE,R5 ;LOAD CYCLE COUNTER
6498 037124 005002 10$: CLR R2 ;FLAG FOR PATTERN CHOICE
6499 037126 012737 001605 007200 MOV #901.,ERRNBR ;LOAD ERROR NUMBER
6500 037134 004737 016326 CALL READ ;LOAD ERROR NUMBER
6501 037140 103002 BCC 20$ ;TEST OTHER MODULE ADDRESS
6502 037142 104432 EXIT TST ;BRANCH IF NO ADDR. ERROR FOUND
;EXIT TEST IF ADDRESSING ERROR
;*****
6503 037146 052777 000001 144546 20$: BIS #1,@0THMOD ;LOAD 500 US INPUT DEBOUNCE
6504 037154 142777 000002 144540 BICB #2,@0THMOD ;
6505 037162 012703 000001 MOV #1,R3 ;LOAD INPUT DEBOUNCE COUNTER
6506 037166 032777 000001 144504 BIT #1,@MOD ;FIND DEBOUNCE OF OUTPUT MODULE
6507 037174 001003 BNE 30$ ;BRANCH IF RTO IS SET (500 US)
6508 037176 063703 004170 ADD OUTDE2,R3 ;ADD RELAY OUTPUT MODULE DEB
6509 037202 000402 BR 40$ ;BRANCH OVER NEXT COMMAND
6510 037204 063703 004166 30$: ADD OUTDE1,R3 ;ADD OPTP OUTPUT DEB.
6511 037210 013701 002236 40$: MOV PAT1,R1 ;GET FIRST PATTERN
6512 037214 010304 50$: MOV R3,R4 ;SAVE WAIT COUNTER VALUE
6513 037216 010177 144460 MOV R1,@DAT ;LOAD OUTPUT DATA REGISTER
6514 037222 004737 011514 60$: JSR PC,WT500 ;WAIT 500 US
6515 037226 005304 DEC R4 ;DECREMENT WAIT COUNTER
6516 037230 001374 BNE 60$ ;BRANCH IF COUNTER IS NOT ZERO
6517 037232 017737 144444 004064 MOV @DAT,BAD ;READ LOADED DATA BACK
6518 037240 020137 004064 CMP R1,BAD ;ARE THE LOADED DATA STILL THERE?
6519 037244 004737 012146 CALL INSERT ;SKIP BRANCH IF 'SFI' IS SET
6520 037250 001406 BEQ 70$ ;BRANCH IF YES
6521 037252 010137 004062 MOV R1,GOOD ;SET UP DATA FOR ERROR REPORT
6522 037256 ERRSOF 902,E902,EER2 ;ERROR HANDLER
037256 104457 TRAP C$ERSOFT
037260 001606 .WORD 902

```

TEST 9: DIGITAL OUTPUT LOOPBACK TEST WITH SELECTABLE DATA PATT

```

037262 037714
037264 007400
6523 037266 012737 001607 007200 70$: MOV #903.,ERRNBR ;LOAD ERROR NUMBER
6524 037274 004737 016326 CALL READ ;READ DATA REG. OF INPUT MODULE
6525 037300 103002 BCC 80$ ;BRANCH IF NO ADDR. ERROR FOUND
6526 037302 EXIT TST ;EXIT TEST IF ADDRESSING ERROR
037302 104432 TRAP C$EXIT
037304 000710 .WORD L10042-.
6527 037306 020137 004064 80$: CMP R1,BAD ;CMP READ AND LOADED DATA
6528 037312 004737 012146 CALL INSERT ;SKIP BRANCH IF 'SFI' IS SET
6529 037316 001406 BEQ 90$ ;BRANCH IF EQUAL
6530 037320 010137 004062 MOV R1,GOOD ;GET GOOD PATTERN
6531 037324 ERRSOFT 904,E903,EER2 ;ERROR HANDLER
037324 104457 TRAP C$ERSOFT
037326 001610 .WORD 904
037330 040017 .WORD E903
037332 007400 .WORD EER2
6532 037334 005702 90$: TST R2 ;IS THE 2ND PATTERN USED?
6533 037336 001004 BNE 100$ ;BRANCH IF YES
6534 037340 005202 INC R2 ;INC FLAG TO SHOW 2ND PATTERN
6535 037342 013701 002240 MOV PAT2,R1 ;GET SECOND PATTERN FROM P TABLE
6536 037346 000722 BR 50$ ;BRANCH TO OUTPUT LOOP
6537 037350 005305 100$: DEC R5 ;ALL CYCLES PERFORMED?
6538 037352 001264 BNE 10$ ;BRANCH IF NO
6539
6540 ;NOW WE WILL TEST THAT THE OUTPUT DEBOUNCE TIME IS NOT FINISHED TOO FAST
6541 ;ONLY IF WE HAVE A RELAY OUTPUT (5 MS DEBOUNCE).
6542
6543 037354 BGNSEG
037354 104404 TRAP C$BSEG
6544 037356 032777 000001 144314 BIT #1,@MOD ;IS THE DEBOUNCE OF OUTPUT 500 US
6545 037364 001076 BNE 170$ ;SKIP THE FOLLOWING PART IF YES
6546 037366 012777 177777 144306 MOV #177777,@DAT ;LOAD OUTPUT MODULE DAT REGISTER
6547 037374 010304 MOV R3,R4 ;GET IN AND OUTPUT DEBOUNCE
6548 037376 004737 011514 110$: JSR PC,WT500 ;INPUT DAT REGISTER WILL ALSO
6549 037402 005304 DEC R4 ;BE LOADED
6550 037404 001374 BNE 110$ ;...
6551 037406 005077 144270 CLR @DAT ;CLEAR OUTPUT DATA REGISTER
6552 037412 012704 000002 MOV #2,R4 ;LOAD WAIT COUNTER
6553 037416 004737 011522 120$: JSR PC,WT25 ;WAIT 25 US
6554 037422 005304 DEC R4 ;...
6555 037424 001374 BNE 120$ ;BRANCH IF NOT FINISHED
6556 037426 012737 001611 007200 MOV #905.,ERRNBR ;ERROR NUMBER FOR READ ROUT.
6557 037434 004737 016326 CALL READ ;READ LOOPED DATA
6558 037440 103002 BCC 130$ ;BRANCH IF NO ADDR. ERROR FOUND
6559 037442 EXIT TST ;EXIT TEST IF ADDRESSING ERROR
037442 104432 TRAP C$EXIT
037444 000550 .WORD L10042-.
6560 037446 022737 177777 004064 130$: CMP #177777,BAD ;DATA REG OF INPUT SHOULD NOT CHANGE
6561 037454 004737 012146 CALL INSERT ;SKIP BRANCH IF 'SFI' IS SET
6562 037460 001407 BEQ 140$ ;BRANCH IF YES
6563 037462 012737 177777 004062 MOV #177777,GOOD ;SET UP DATA FOR ERROR HANDLER
6564 037470 ERRSOFT 906,E904,EER2 ;ERROR HANDLER
037470 104457 TRAP C$ERSOFT
037472 001612 .WORD 906
037474 040074 .WORD E904
037476 007400 .WORD EER2

```

TEST 9: DIGITAL OUTPUT LOOPBACK TEST WITH SELECTABLE DATA PATT

```

6565 037500          140$: CKLOOP          ;
      037500 104406          ;
6566 037502 010304          ;LOAD WAIT COUNTER      TRAP  C$CLP1
6567 037504 004737 011514 150$: MOV R3,R4 ;WAIT FOR 500 US
6568 037510 005304          ;
6569 037512 001374          ;...
6570 037514 012737 001613 007200 BNE 150$ ;BRANCH IF NOT FINISHED
6571 037522 004737 016326 MOV #907.,ERRNBR ;LOAD ERROR NUMBER
6572 037526 103002          CALL READ ;READ DATA REG. OF INPUT MODULE
6573 037530          EXIT TST ;BRANCH IF NO ADDR. ERROR FOUND
      037530 104432          ;EXIT TEST IF ADDRESSING ERROR
      037532 000462          ;
6574 037534 005737 004064 160$: TST BAD ;OUT+INPUT DATA SHOULD BE EQUAL
6575 037540 004737 012146 CALL INSERT ;SKIP BRANCH IF 'SFI' IS SET
6576 037544 001406          BEQ 170$ ;BRANCH IF EQUAL (ZERO)
6577 037546 005037 004062 CLR GOOD ;GET GOOD PATTERN
6578 037552          ERRSOFT 908,E903,EER2 ;ERROR HANDLER
      037552 104457          ;
      037554 001614          ;
      037556 040017          ;
      037560 007400          ;
6579 037562          170$: ENDSEG          ;
      037562          ;
      037562 104405          ;
6580 037564 005737 002242 TST QVP ;IS QUICK VERIFY PASS SELECTED?
6581 037570 001010          BNE EXQV9 ;IF YES, EXIT TEST
6582 037572 005237 004144 INC ITRCNT ;ITERATION COUNTER + 1
6583 037576 023737 004146 004144 CMP ITRDEF,ITRCNT ;DEFAULT ITERATION EXECUTED
6584 037604 001402          BEQ EXQV9 ;IF YES, EXIT TEST
6585 037606 000137 037120 JMP ITRA9 ;IF NO, TEST ITERATION
6586 037612          EXQV9: EXIT TST ;EXIT TEST
      037612 104432          ;
      037614 000400          ;
6587
6588
6589
6590 037616 045 123 062 TSHD9:: .NLIST BEX
6591 037714 104 101 124 E902: .ASCIZ /#S2#ADIGITAL OUTPUT LOOPBACK TEST - PATTERN PAIR SELECTABLE#N/
6592 037777 101 106 124 .ASCIZ /DATA REGISTER OF DIGITAL OUTPUT MODULE INCORRECT /<12><15>
6593 040017 122 105 103 E903: .ASCIZ /AFTER READ BACK/
6594 040074 104 105 102 E904: .ASCIZ /RECEIVED DATA INCORRECT - CHECK OUTPUT LOGIC/
6595 040125 104 101 124 .ASCIZ /DEBOUNCE TIME INCORRECT/<12><15>
6596 .ASCIZ /DATA ACCEPTED TOO FAST (LOADED DEBOUNCE NOT FINISHED)/
6597 .LIST BEX
6598 .EVEN
6599 040214          ENDTST
      040214
      040214 104401          ;
                                L10042: TRAP C$ETST

```

TEST 10: DIGITAL INPUT LOOPBACK TEST - RANDOM PATTERN

```

6601 .SBTTL TEST 10: DIGITAL INPUT LOOPBACK TEST - RANDOM PATTERN
6602 :*****
6603 : TEST 10 - DIGITAL LOOPBACK TEST WITH RANDOM PATTERN AND RANDOM OR
6604 : SELECTABLE DEBOUNCE PERIOD.
6605 :
6606 : THIS TEST CHECKS THE DATA TRANSFER BETWEEN INPUT AND OUTPUT
6607 : MODULES. THE OUTPUT MODULE IS THAT SPECIFIED AS THE "OTHER" IN
6608 : THE HARDWARE QUESTIONS. SEVERAL CYCLES ARE PERFORMED USING DATA
6609 : GENERATED BY A RANDOM PATTERN GENERATOR WITHIN THE PROGRAM.
6610 :
6611 : IF LOW LEVEL TESTING WAS NOT SELECTED IN THE HARDWARE QUESTIONS,
6612 : THE SELECTED DEBOUNCE PERIOD IS USED. IF "ALL" WAS SPECIFIED,
6613 : THE PERIOD IS SELECTED AT RANDOM FROM THE 3 ALLOWED VALUES.
6614 :*****
6615 040216 BGNTST
        040216 T10::
6616 040216 004737 011714 CALL SELECT ;CALL SELECT ROUTINE
6617 040222 002001 .WORD 2001 ;GIVE TEST PARAMETER
6618 040224 040676 TSHD10 ;GIVE TEST HEADER ADDRESS
6619 040226 103002 BCC 1$ ;IF CARRY IS CLEARED, BRANCH
6620 040230 EXIT TST ;IF CARRY IS SET, EXIT TEST
        040230 104432 TRAP C$EXIT
        040232 000642 .WORD L10043-.
6621 040234 005037 004144 1$: CLR ITRCNT ;CLEAR ITERATION COUNTER
6622 040240 013702 004160 ITRA10: MOV CYCRAD,R2 ;LOAD CYCLE COUNTER
6623 040244 005737 003716 TST LOWLVL ;IS LOW LEVEL SELEC. IN P-TABLE?
6624 040250 001402 BEQ 10$ ;BRANCH IF NO
6625 040252 000137 040504 JMP LLPA10 ;JUMP TO LOW LEVEL PART
6626 040256 013705 003720 10$: MOV DBOUNC,R5 ;GET DEBOUNCE FROM P TABLE
6627 040262 005205 INC R5 ;CREATE DEBOUNCE VALUE FOR MOD
6628 040264 042705 177774 BIC #177774,R5 ;WE ONLY USE BITS 1 AND 2
6629 040270 022737 000003 003720 CMP #3,DBOUNC ;ALL DEBOUNCE PERIODS SELECTED ?
6630 040276 001002 BNE 30$ ;BRANCH IF NO
6631 040300 012705 000001 20$: MOV #1,R5 ;GET FIRST DEBOUNCE
6632 040304 110577 143370 30$: MOV R5,@MOD ;LOAD DEBOUNCE INTO MOD REGISTER
6633 040310 052777 000100 143362 BIS #100,@MOD ;SWITCH ON THE MODULE LED AGAIN
6634 040316 116503 004172 MOV DBTA8(R5),R3 ;GET WAIT VALUE FROM TABLE
6635 040322 104404 40$: BGNSEG
        040322 104404 TRAP C$BSEG
6636 040324 013701 016574 MOV RB,R1 ;GET FIRST RANDOM PATTERN
6637 040330 010304 50$: MOV R3,R4 ;SAVE WAIT COUNTER VALUE
6638 040332 012737 001751 007200 MOV #1001.,ERRNBR ;LOAD ERROR NUMBER FOR SUBROU.
6639 040340 004737 016122 JSR PC,WRITE ;CALL WRITE ROUTINE
6640 040344 103002 BCC 60$ ;BRANCH IF NO ADDR. ERROR FOUND
6641 040346 EXIT TST ;EXIT TEST IF ADDRESSING ERROR
        040346 104432 TRAP C$EXIT
        040350 000524 .WORD L10043-.
6642 040352 063704 004166 60$: ADD OUTDE1,R4 ;ADD OPTO OUTPUT DEBOUNCE
6643 040356 032777 000001 143336 BIT #1,@OTHMOD ;IS DEBOUNCE OF OUTPUT 500 US ?
6644 040364 001002 BNE 70$ ;BRANCH IF YES
6645 040366 063704 004170 ADD OUTDE2,R4 ;IF NO ADD RELAY DEBOUNCE
6646 040372 004737 011514 70$: JSR PC,WT500 ;WAIT 500 US
6647 040376 005304 DEC R4 ;DECREMENT WAIT COUNTER
6648 040400 001374 BNE 70$ ;BRANCH IF COUNTER IS NOT ZERO
6649 040402 017737 143274 004064 MOV @DAT,BAD ;READ DIGITAL INPUT PATTERN
6650 040410 020137 004064 CMP R1,BAD ;CMP READ AND LOADED DATA
6651 040414 004737 012146 CALL INSERT ;SKIP BRANCH IF 'SFI' IS SET

```

TEST 10: DIGITAL INPUT LOOPBACK TEST - RANDOM PATTERN

```

6652 040420 001406          BEQ      80$          ;BRANCH IF EQUAL
6653 040422 010137 004062    MOV      R1,GOOD     ;GET GOOD PATTERN
6654 040426          ERRSOFT 1002,E1002,EER2 ;ERROR HANDLER
        040426 104457          TRAP    C$ERSOFT
        040430 001752          .WORD  1002
        040432 040762          .WORD  E1002
        040434 007400          .WORD  EER2
6655 040436          80$:   ENDSEG
        040436          10000$: TRAP    C$ESEG
        040436 104405
6656 040440 004737 016534    CALL    RANDOM       ;GENERATE NEXT PATTERN
6657 040444 022737 000003 003720    CMP     #3,DBOUNC   ;ALL DEBOUNCE REQUESTED ?
6658 040452 001010          BNE     90$         ;BRANCH IF NO
6659 040454 005205          INC     R5          ;NEXT DEBOUNCE
6660 040456 022705 000004    CMP     #4,R5       ;ALL DEBOUNCE PERIODS EXECUTED?
6661 040462 001310          BNE     30$         ;BRANCH IF NO
6662 040464 005302          DEC     R2          ;ALL CYCLES PERFORMED ?
6663 040466 001304          BNE     20$         ;BRANCH IF NO
6664 040470 000137 040644    JMP     QVR10       ;SKIP LOW LEVEL PART
6665 040474 005302 90$:   DEC     R2          ;ALL CYCLES PERFORMED ?
6666 040476 001311          BNE     40$         ;BRANCH IF NO
6667 040500 000137 040644    JMP     QVR10       ;SKIP LOW LEVEL PART
6668
6669          ;THIS PART WILL TEST THAT THE LOW LEVEL SELECT WILL WORK WITH RANDOM
6670          ;PATTERN. THE DEBOUNCE OF THE DIGITAL INPUT MODULE IS FIXED AT 5 MS
6671
6672 040504 052777 000002 143166 LLPA10: BIS     #2,&MOD   ;LOAD A DEBOUNCE OF 5 MS
6673 040512 042777 000001 143160    BIC     #1,&MOD     ;CLEAR DEBOUNCE OF 5 US
6674 040520 052777 000010 143152    BIS     #10,&MOD   ;SWITCH INPUT MODULE TO LLS
6675 040526 013702 004156          MOV     CYCLLS,R2  ;LOAD CYCLE COUNTER
6676 040532          BGNSEG
        040532 104404          TRAP    C$BSEG
6677 040534 013701 016574 10$:   MOV     RB,R1       ;LOAD DATA
6678 040540 013704 004162          MOV     LLWC,R4    ;LOAD LLS WAIT COUNTER
6679 040544 012737 001753 007200    MOV     #1003.,ERRNBR ;LOAD ERROR NUMBER
6680 040552 004737 016122          JSR     PC,WRITE   ;SEND DATA AND ADD OUTPUT DEBOU.
6681 040556 103002          BCC     20$        ;BRANCH IF NO ADDR. ERROR FOUND
6682 040560          EXIT    TST       ;EXIT TEST IF ADDRESSING ERROR
        040560 104432          TRAP    C$EXIT
        040562 000312          .WORD  L10043..
6683 040564 004737 011514 20$:   JSR     PC,WT500   ;WAIT 500 US
6684 040570 005304          DEC     R4
6685 040572 001374          BNE     20$        ;BRANCH IF NOT FINISHED
6686 040574 017737 143102 004064    MOV     @DAT,BAD   ;READ INPUT DATA
6687 040602 020137 004064          CMP     R1,BAD    ;DATA SHOULD NOT BE CHANGED
6688 040606 004737 012146          CALL    INSERT    ;SKIP BRANCH IF 'SFI' IS SET
6689 040612 001406          BEQ     30$        ;BRANCH IF OK
6690 040614 010137 004062          MOV     R1,GOOD   ;SET UP DATA FOR ERROR HANDLER
6691 040620          ERRSOFT 1004,E1002,EER2 ;ERROR HANDLER
        040620 104457          TRAP    C$ERSOFT
        040622 001754          .WORD  1004
        040624 040762          .WORD  E1002
        040626 007400          .WORD  EER2
6692 040630          30$:   CKLOOP
        040630 104406          TRAP    C$CLP1
6693 040632 004737 016534    CALL    RANDOM
6694 040636 005302          DEC     R2
        ;CREATE NEXT PATTERN
        ;IS CYCLE COUNTER OVER?

```


TEST 10: DIGITAL INPUT LOOPBACK TEST - RANDOM PATTERN

```

6695 040640 001335          BNE      10$          ;BRANCH IF NO
6696 040642          ENDSEG
        040642          10001$: TRAP      C$ESEG
        040642 104405
6697 040644 005737 002242    QVR10: TST      QVP          ;IS QUICK VERIFY PASS SELECTED?
6698 040650 001010          BNE      EXQV10    ;IF YES, EXIT TEST
6699 040652 005237 004144    INC      ITRCNT    ;ITERATION COUNTER + 1
6700 040656 023737 004146 004144  CMP      ITRDEF,ITRCNT ;DEFAULT ITERATION EXECUTED
6701 040664 001402          BEQ      EXQV10    ;IF YES, EXIT TEST
6702 040666 000137 040240    JMP      ITRA10    ;IF NO, TEST ITERATION
6703 040672          EXQV10: EXIT    TST          ;EXIT TEST
        040672 104432          TRAP      C$EXIT
        040674 000200          .WORD    L10043-.
6704
6705
6706          .NLIST    BEX
6707 040676      045      123      062    TSHD10: .ASCII /S2ADIGITAL INPUT LOOPBACK TEST - RANDOM PATTERN/N/
6708 040762      104      111      107    E1002: .ASCII /DIGITAL INPUT DATA REGISTER INCORRECT /<12><15>/
6709 041032      114      117      101    .ASCII /LOADED AND READ DATA NOT THE SAME/
6710          .LIST    BEX
6711          .EVEN
6712
6713 041074          ENDTST
        041074          L10043: TRAP      C$ETST
        041074 104401

```

TEST 11: DIGITAL OUTPUT LOOPBACK TEST - RANDOM PATTERN

```

6715 .SBTTL TEST 11: DIGITAL OUTPUT LOOPBACK TEST - RANDOM PATTERN
6716 ;*****
6717 ; TEST 11 - DIGITAL OUTPUT LOOPBACK TEST WITH RANDOM PATTERN
6718 ;
6719 ; THIS TEST CHECKS THE DATA OUTPUT LINES OF DIGITAL OUTPUT MODULES
6720 ; BY READING THE OUTPUT DATA WITH AN INPUT MODULE. THE INPUT MODULE IS THAT
6721 ; SPECIFIED AS THE 'OTHER' IN THE HARDWARE QUESTIONS. 500 CYCLES ARE
6722 ; PERFORMED USING DATA GENERATED BY A RANDOM PATTERN GENERATOR WITHIN THE
6723 ; PROGRAM. THE TIMEOUT PERIOD IS BASED ON THE DEBOUNCE PERIOD FOUND IN THE
6724 ; MOD REGISTER OF THE DIGITAL OUTPUT MODULE.
6725 ; THE DIGITAL INPUT MODULE IS ALWAYS LOADED WITH A DEBOUNCE OF 500 US.
6726 ;*****
6727 041076 BGNTST
        041076
6728 041076 004737 011714 CALL SELECT ;CALL SELECT ROUTINE
6729 041102 002002 .WORD 2002 ;GIVE TEST PARAMETER
6730 041104 041400 TSHD11 ;GIVE TEST HEADER ADDRESS
6731 041106 103002 BCC 1$ ;IF CARRY IS CLEARED, BRANCH
6732 041110 EXIT TST ;IF CARRY IS SET, EXIT TEST
        TRAP C$EXIT
        .WORD L10044-.
6733 041114 005037 004144 1$: CLR ITRCNT ;CLEAR ITERATION COUNTER
6734 041120 013702 004160 ITRA11: MOV CYCRAD,R2 ;LOAD CYCLE COUNTER
6735 041124 012737 002115 007200 MOV #1101.,ERRNBR ;LOAD ERROR NUMBER
6736 041132 004737 016326 CALL READ ;TEST OTHER MODULE ADDRESS (INPUT)
6737 041136 103002 BCC 10$ ;BRANCH IF NO ADDR. ERROR FOUND
6738 041140 EXIT TST ;EXIT TEST IF ADDRESSING ERROR
        TRAP C$EXIT
        .WORD L10044-.
6739 041144 152777 000001 142550 10$: BISB #1,@OTHMOD ;SET 500 US DEBOUNCE IN INPUT
6740 041152 042777 000002 142542 BIC #2,@OTHMOD ;CLEAR 5 MS DEBOUNCE IN INPUT MOD
6741 041160 012703 000001 MOV #1,R3 ;LOAD WAIT COUNTER WITH INPUT DEB.
6742 041164 032777 000001 142506 BIT #1,@MOD ;FIND DEBOUNCE OF OUTPUT MODULE
6743 041172 001003 BNE 20$ ;BRANCH IF RTO IS SET (500 US)
6744 041174 063703 004170 ADD OUTDE2,R3 ;ADD RELAY OUTPUT MODULE DEB.
6745 041200 000402 BR 30$ ;
6746 041202 063703 004166 20$: ADD OUTDE1,R3 ;ADD DEBOUNCE OF OPTO OUTPUT MODULE
6747 041206 013701 016574 30$: MOV RB,R1 ;GET FIRST RANDOM PATTERN
6748 041212 BGNSEG
        TRAP C$BSEG
        .WORD
6749 041214 010304 40$: MOV R3,R4 ;SAVE WAIT COUNTER VALUE
6750 041216 010177 142460 MOV R1,@DAT ;LOAD OUTPUT DATA REGISTER
6751 041222 004737 011514 50$: JSR PC,WT500 ;WAIT 500 US
6752 041226 005304 DEC R4 ;DECREMENT WAIT COUNTER
6753 041230 001374 BNE 50$ ;BRANCH IF COUNTER IS NOT ZERO
6754 041232 017737 142444 004064 MOV @DAT,BAD ;READ LOADED DATA BACK
6755 041240 020137 004064 CMP R1,BAD ;LOADED DATA STILL THERE?
6756 041244 004737 012146 CALL INSERT ;SKIP BRANCH IF 'SFI' IS SET
6757 041250 001406 BEQ 60$ ;BRANCH IF YES
6758 041252 010137 004062 MOV R1,GOOD ;SET UP DATA FOR ERROR REPORT
6759 041256 ERRSOFT 1102,E1102,EER2 ;ERROR HANDLER
        TRAP C$ERSOFT
        .WORD 1102
        .WORD E1102
        .WORD EER2
6760 041266 60$: CKLOOP ;
        TRAP C$CLP1
        .WORD 104406
    
```

TEST 11: DIGITAL OUTPUT LOOPBACK TEST - RANDOM PATTERN

```

6761 041270 012737 002117 007200      MOV    #1103.,ERRNBR      ;LOAD ERROR NUMBER
6762 041276 004737 016326                CALL   READ              ;READ D/O LINES WITH DI MODULE
6763 041302 103002                BCC    70$              ;BRANCH IF NO ADDR. ERROR FOUND
6764 041304                EXIT   TST              ;EXIT TEST IF ADDRESSING ERROR
      041304 104432
      041306 000364                TRAP   C$EXIT
      041310 020137 004064      70$:  CMP    R1,BAD        ;CMP READ AND LOADED DATA
      041314 004737 012146                CALL   INSERT          ;SKIP BRANCH IF 'SFI' IS SET
6767 041320 001406                BEQ    80$              ;BRANCH IF EQUAL
6768 041322 010137 004062      MOV    R1,GOOD        ;GET GOOD PATTERN
6769 041326                ERRSOFT 1104,E1103,EER2 ;ERROR HANDLER
      041326 104457                TRAP   C$ERSOFT
      041330 002120                .WORD 1104
      041332 041603                .WORD E1103
      041334 007400                .WORD EER2
6770 041336                80$:  ENDSEG
      041336                10000$: TRAP   C$ESEG
      041336 104405
6771 041340 004737 016534      CALL   RANDOM          ;GENERATE NEXT PATTERN
6772 041344 005302                DEC    R2              ;ALL CYCLES PERFORMED ?
6773 041346 001317                BNE    30$              ;BRANCH IF NO
6774 041350 005737 002242      TST    QVP            ;IS QUICK VERIFY PASS SELECTED?
6775 041354 001007                BNE    EXQV11          ;IF YES, EXIT TEST
6776 041356 005237 004144      INC    ITRCNT         ;ITERATION COUNTER + 1
6777 041362 023737 004146 004144    CMP    ITRDEF,ITRCNT  ;DEFAULT ITERATION EXECUTED
6778 041370 001401                BEQ    EXQV11          ;IF YES, EXIT TEST
6779 041372 000652                BR     ITR11           ;IF NO, TEST ITERATION
6780 041374                EXQV11: EXIT   TST    ;EXIT TEST
      041374 104432                TRAP   C$EXIT
      041376 000274                .WORD 110444-.
6781
6782                .NLIST BEX
6783 041400      045      123      062  TSHD11: .ASCIZ /#S2#ADIGITAL OUTPUT LOOPBACK TEST - RANDOM PATTERN#N/
6784 041465      104      101      124  E1102: .ASCIZ /DATA REGISTER OF DIGITAL OUTPUT MODULE INCORRECT /<12><15>
6785 041550      104      101      124      .ASCIZ /DATA CHANGED AFTER LOADING/
6786 041603      114      117      101  E1103: .ASCIZ /LOADED AND READ DATA NOT THE SAME - CHECK OUTPUT LOGIC/
6787                .LIST BEX
6788                .EVEN
6789
6790 041672                ENDTST
      041672
      041672 104401                L10044: TRAP   C$ETST

```

TEST 12: DIGITAL INPUT LOOPBACK TEST - SLIDING PATTERN

```

6792 .SBTTL TEST 12: DIGITAL INPUT LOOPBACK TEST - SLIDING PATTERN
6793 ;*****
6794 ; TEST 12 - DIGITAL INPUT LOOPBACK TEST WITH SLIDING PATTERN
6795 ;
6796 ;THIS TEST TRANSFERS SLIDING ONES AND THEN SLIDING ZEROS BETWEEN
6797 ;OUTPUT MODULE AND INPUT MODULES. THE OUTPUT MODULE THAT SPECIFIED
6798 ;AS THE 'OTHER' IN THE HARDWARE QUESTIONS.
6799 ;THE TRANSFERS ARE PERFORMED USING THE LEVEL SELECT AND THE DEBOUNCE
6800 ;PERIOD SPECIFIED IN THE HARDWARE QUESTIONS.
6801 ;*****
6802 041674 BGNTST
        041674
6803 041674 004737 011714 CALL SELECT T12::
        041700 002001 .WORD 2001 ;CALL SELECT ROUTINE
6804 041702 042572 TSHD12 ;GIVE TEST PARAMETER
6805 041704 103002 BCC 1$ ;GIVE TEST HEADER ADDRESS
6806 041706 041706 EXIF TST ;IF CARRY IS CLEARED, BRANCH
        041706 104432 ;IF CARRY IS SET, EXIT TEST
        041710 001064 TRAP C$EXIT
6808 041712 005037 004144 1$: CLR ITRCNT ;CLEAR ITERATION COUNTER
        041716 013702 004156 ITRA12: MOV CYCLLS,R2 ;LOAD CYCLE COUNTER
6809 041722 005737 003716 TST LOWLVL ;IS LOW LEVEL SELEC. IN P-TABLE
6810 041726 001402 BEQ 10$ ;BRANCH IF NO
6811 041730 000137 042270 JMP LLPA12 ;JUMP TO LOW LEVEL PART
6812 041734 013705 003720 10$: MOV DBOUNC,R5 ;GET DEBOUNCE FROM P TABLE
6813 041740 005205 INC R5 ;CREATE DEBOUNCE VALUE FOR MOD
6814 041742 042705 177774 BIC @177774,R5 ;WE ONLY USE BITS 1 AND 2
6815 041746 022737 000003 003720 CMP @3,DBOUNC ;ALL DEBOUNCE PERIODS SELECTED ?
6816 041754 001002 BNE 20$ ;BRANCH IF NO
6817 041756 012705 000001 20$: MOV @1,R5 ;GET FIRST DEBOUNCE
6818 041762 110577 141712 30$: MOVB R5,@MOD ;LOAD DEBOUNCE INTO MOD REGISTER
6819 041766 052777 000100 141704 BIS @100,@MOD ;SWITCH ON THE MODULE LED
6820 041774 012701 000001 MOV @1,R1 ;GET MASK FOR SLIDING ONES
6821 042000 042000 104404 40$: BGNSEG TRAP C$BSEG
6822 042002 116503 004172 MOVB DBTA8(R5),R3 ;GET WAIT VALUE FROM TABLE
6823 042006 012737 002261 007200 MOV @1201.,ERRNBR ;LOAD ERROR NUMBER
6824 042014 004737 016122 JSR PC,WRITE ;CALL WRITE ROUTINE
6825 042020 103002 BCC 50$ ;BRANCH IF NO ADDR. ERROR FOUND
6826 042022 042022 104432 EXIT TST ;EXIT TEST IF ADDRESSING ERROR
        042024 000750 TRAP C$EXIT
6827 042026 063703 004166 141662 50$: ADD OUTDE1,R3 ;ADD OPTO OUTPUT DEBOUNCE
6828 042032 032777 000001 BIT @1,@OTHMOD ;IS DEBOUNCE OF OUTPUT 500 US ?
6829 042040 001002 BNE 60$ ;BRANCH IF YES
6830 042042 063703 004170 ADD OUTDE2,R3 ;IF NO, ADD RELAY DEBOUNCE
6831 042046 010304 60$: MOV R3,R4 ;SAVE WAIT COUNTER VALUE
6832 042050 004737 011514 70$: JSR PC,WT500 ;WAIT 500 US
6833 042054 005304 DEC R4 ;DECREMENT WAIT COUNTER
6834 042056 001374 BNE 70$ ;BRANCH IF COUNTER IS NOT ZERO
6835 042060 017737 141616 004064 MOV @DAT,BAD ;READ DIGITAL INPUT PATTERN
6836 042066 020137 004064 CMP R1,BAD ;CMP READ AND LOADED DATA
6837 042072 004737 012146 CALL INSERT ;SKIP BRANCH IF 'SFI' IS SET
6838 042076 001406 BEQ 80$ ;BRANCH IF EQUAL
6839 042100 010137 004062 MOV R1,GOOD ;GET GOOD PATTERN
6840 042104 104457 ERRSOFT 1202,E1202,EEP2 ;ERROR HANDLER
        042104 TRAP C$ERRSOFT

```

TEST 12: DIGITAL INPUT LOOPBACK TEST - SLIDING PATTERN

```

042106 002262
042110 042660
042112 007400
6842 042114 80$: ENDSEG
042114
042114 104405
6843 042116 006301
6844 042120 103327
6845
6846
6847
6848 042122 012701 177776
6849 042126 90$:
042126 104404 BGNSEG
6850 042130 010304 MOV R3,R4 ;SAVE WAIT COUNTER VALUE
6851 042132 012737 002263 007200 MOV #1203.,ERRNBR ;LOAD ERROR NUMBER
6852 042140 004737 016122 JSR PC,WRITE ;CALL WRITE ROUTINE
6853 042144 103002 BCC 100$ ;BRANCH IF NO ADDR. ERROR FOUND
6854 042146 EXIT TST ;EXIT TEST IF ADDRESSING ERROR
042146 104432 TRAP C$EXIT
042150 000624 .WORD L10045..
6855 042152 004737 011514 100$: JSR PC,WT500 ;WAIT 500 US
6856 042156 005304 DEC R4 ;DECREMENT WAIT COUNTER
6857 042160 001374 BNE 100$ ;BRANCH IF COUNTER IS NOT ZERO
6858 042162 017737 141514 004064 MOV @DAT,BAD ;READ DIGITAL INPUT PATTERN
6859 042170 020137 004064 CMP R1,BAD ;CMP READ AND LOADED DATA
6860 042174 004737 012146 CALL INSERT ;SKIP BRANCH IF 'SFI' IS SET
6861 042200 001406 BEQ 110$ ;BRANCH IF EQUAL
6862 042202 010137 004062 MOV R1,GOOD ;GET GOOD PATTERN
6863 042206 ERRSOFT 1204,E1202,EER2 ;ERROR HANDLER
042206 104457 TRAP C$ERSOFT
042210 002264 .WORD 1204
042212 042660 .WORD E1202
042214 007400 .WORD EER2
6864 042216 110$: ENDSEG
042216
042216 104405 10001$: TRAP C$ESEG
6865 042220 006301 ASL R1 ;NEXT SLIDING ZEROS DATA
6866 042222 103002 BCC 120$ ;BRANCH IF ALL DATA LINES WERE ZEROS
6867 042224 005501 ADC R1 ;ADD THE CARRY AND REPEAT
6868 042226 000737 BR 90$ ;WITH NEW DATA
6869 042230 022737 000003 003720 120$: CMP #3,DBOUNC ;ALL DEBOUNCE SELECTED?
6870 042236 001010 BNE 130$ ;BRANCH IF NO
6871 042240 005205 INC R5 ;NEXT DEBOUNCE
6872 042242 022705 000004 CMP #4,R5 ;ALL DEBOUNCE PERIODS EXECUTED
6873 042246 001245 BNE 30$ ;BRANCH IF NO
6874 042250 005302 DEC R2 ;ALL CYCLES PERFORMED ?
6875 042252 001241 BNE 20$ ;BRANCH IF NO
6876 042254 000137 042540 JMP QVR12 ;SKIP LOW LEVEL PART
6877 042260 005302 130$: DEC R2 ;ALL CYCLES PERFORMED ?
6878 042262 001246 BNE 40$ ;BRANCH IF NO
6879 042264 000137 042540 JMP QVR12 ;JUMP TO QUICK VERIFY ROUTINE
6880
6881 ;THIS PART WILL TEST THAT THE LOW LEVEL SELECT WILL WORK WITH SLIDING
6882 ;PATTERNS. THE DEBOUNCE OF THE DIGITAL INPUT MODULE WILL FIX 5 MS
6883 ;BUT THE WAIT LOOP FOR LLS WILL BE 10 MS.
6884

```


TEST 12: DIGITAL INPUT LOOPBACK TEST - SLIDING PATTERN

```

042516 042660
042520 007400
6927 042522 80$: ENDSEG
042522
042522 104405
6928 042524 006301
6929 042526 103002
6930 042530 005501
6931 042532 000737
6932 042534 005302
6933 042536 001267
6934 042540 005737 002242
6935 042544 001010
6936 042546 005237 004144
6937 042552 023737 004146 004144
6938 042560 001402
6939 042562 000137 041716
6940 042566
042566 104432
042570 000204
6941
6942
6943
6944 042572 045 123 062 TSHD12: .NLIST BEX
6945 042660 104 101 124 E1202: .ASCII /#S2#ADIGITAL INPUT LOOPBACK TEST - SLIDING PATTERN#N/
6946 042732 114 117 101 .ASCII /DATA REGISTER OF INPUT MODULE INCORRECT /<12><15>#N/
6947 .LIST BEX
6948 .EVEN
6949
6950 042774
042774
042774 104401

```

.WORD E1202
.WORD EER2

10003\$: TRAP C#ESEG

TRAP C#EXIT
.WORD L10045-

L10045: TRAP C#ETST

```

;NEXT SLIDING ZERO DATA
;BRANCH IF ALL ZEROS WHERE DONE
;ADD THE CARRY
;AND REPEAT
;ALL CYCLES PERFORMED ?
;BRANCH IF NO
;IS QUICK VERIFY PASS SELECTED?
;IF YES EXIT TEST
;ITERATION COUNTER + 1
;DEFAULT ITERATION EXECUTED
;IF YES, EXIT TEST
;IF NO, TEST ITERATION
;EXIT TEST

```

TEST 13: DIGITAL OUTPUT LOOPBACK TEST - SLIDING PATTERN

```

6952 .SBTTL TEST 13: DIGITAL OUTPUT LOOPBACK TEST - SLIDING PATTERN
6953 ;*****
6954 ; TEST 13 - DIGITAL OUTPUT LOOPBACK TEST WITH SLIDING PATTERN
6955 ;
6956 ; THIS TEST CHECKS THE DATA OUTPUT LINES OF DIGITAL OUTPUT MODULES
6957 ; BY READING THE OUTPUT DATA WITH AN INPUT MODULE. THE INPUT MODULE IS
6958 ; THAT SPECIFIED AS THE 'OTHER' IN THE HARDWARE QUESTIONS. SEVERAL CYCLES ARE
6959 ; PERFORMED USING SLIDING ONES AND THEN SLIDING ZEROS.
6960 ; THE TIMEOUT IS DEPENDENT ON THE DEBOUNCE PERIOD SPECIFIED BY RT BITS
6961 ; IN THE MOD REGISTER OF THE DIGITAL OUTPUT MODULE.
6962 ; THE DIGITAL INPUT MODULE IS ALWAYS LOADED WITH A DEBOUNCE PRIOD OF 5 US.
6963 ;*****
6964 042776          BGNTST
        042776
6965 042776 004737 011714          CALL SELECT          ;CALL SELECT ROUTINE
6966 043002 002002          .WORD 2002          ;GIVE TEST PARAMETER
6967 043004 043450          TSHD13          ;GIVE TEST HEADER ADDRESS
6968 043006 103002          BCC 1$          ;IF CARRY IS CLEARED, BRANCH
6969 043010          EXIT TST          ;IF CARRY IS SET, EXIT TEST
        043010 104432          TRAP C$EXIT
        043012 000666          .WORD L10046-.
6970 043014 005037 004144          1$: CLR ITRCNT          ;CLEAR ITERATION COUNTER
6971 043020 013702 004156          ITRA13: MOV CYCLLS,R2      ;LOAD CYCLE COUNTER
6972 043024 012737 002425 007200  MOV #1301.,ERRNBR    ;LOAD ERROR NUMBER
6973 043032 004737 016326          CALL READ          ;TEST OTHER MODULE ADDRESS
6974 043036 103002          BCC 10$          ;BRANCH IF NO ADDR. ERROR FOUND
6975 043040          EXIT TST          ;EXIT TEST IF ADDRESSING ERROR
        043040 104432          TRAP C$EXIT
        043042 000636          .WORD L10046-.
6976 043044 152777 000001 140650 10$: BISB #1,@OTHMOD      ;LOAD 5 US DEBOUNCE IN INPUT MOD
6977 043052 042777 000002 140642          BIC #2,@OTHMOD
6978 043060 012703 000001          MOV #1,R3          ;...
6979 043064 032777 000001 140606          BIT #1,@MOD      ;LOAD WAIT COUNTER WITH INPUT DEB.
6980 043072 001003          BNE 20$          ;FIND DEBOUNCE OF OUTPUT MODULE
6981 043074 063703 004170          ADD OUTDE2,R3     ;BRANCH IF RTO IS SET (500 US)
6982 043100 000402          BR 30$          ;ADD RELAY OUTPUT MODULE DEB.
6983 043102 063703 004166          20$: ADD OUTDE1,R3  ;BRANCH OVER NEXT COMMAND
6984 043106 012701 000001          30$: MOV #1,R1      ;ADD OPTO OUTPUT DEB.
6985 043112          40$: BGNSEG          ;GET SLIDING ONES' MASK
        043112 104404          TRAP C$BSEG
6986 043114 010304          MOV R3,R4          ;SAVE WAIT COUNTER VALUE
6987 043116 010177 140560          MOV R1,@DAT      ;LOAD OUTPUT DATA REGISTER
6988 043122 004737 011514          50$: JSR PC,WT500  ;WAIT 500 US
6989 043126 005304          DEC R4          ;DECREMENT WAIT COUNTER
6990 043130 001374          BNE 50$          ;BRANCH IF COUNTER IS NOT ZERO
6991 043132 017737 140544 004064          MOV @DAT,BAD     ;READ LOADED DATA BACK
6992 043140 020137 004064          CMP R1,BAD       ;LOADED DATA STILL THERE?
6993 043144 004737 012146          CALL INSERT      ;SKIP BRANCH IF 'SFI' IS SET
6994 043150 001406          BEQ 60$          ;BRANCH IF YES
6995 043152 010137 004062          MOV R1,GOOD      ;SET UP DATA FOR ERROR REPORT
6996 043156          ERRSOFT 1302,E1302,EER2 ;ERROR HANDLER
        043156 104457          TRAP C$ERSOFT
        043160 002426          .WORD 1302
        043162 043536          .WORD E1302
        043164 007400          .WORD EER2
6997 043166          60$: CKLOOP          ;
        043166 104406          TRAP C$CLP1

```


TEST 13: DIGITAL OUTPUT LOOPBACK TEST - SLIDING PATTERN

```

6998 043170 012737 002427 007200      MOV    #1303.,ERRNBR      ;LOAD ERROR NUMBER
6999 04317 004737 016326      CALL   READ              ;READ D/O LINES WITH DI MODULE
7000 04320 103002      BCC    70$              ;BRANCH IF NO ADDR. ERROR FOUND
7001 043204      EXIT   TST              ;EXIT TEST IF ADDRESSING ERROR
      043204 104432      TRAP  C$EXIT
      043206 000472      .WORD L10046-.
7002 043210 020137 004064      70$:  CMP    R1,BAD          ;CMP READ AND LOADED DATA
7003 043214 004737 012146      CALL   INSERT          ;SKIP BRANCH IF 'SFI' IS SET
7004 043220 001406      BEQ    80$            ;BRANCH IF EQUAL
7005 043222 010137 004062      MOV    R1,GOOD        ;GET GOOD PATTERN
7006 043226      ERRSOFT 1304,E1303,EER2 ;ERROR HANDLER
      043226 104457      TRAP  C$ERSOFT
      043230 002430      .WORD 1304
      043232 043622      .WORD E1303
      043234 007400      .WORD EER2
7007 043236      80$:  ENDSEG
      043236      10000$: TRAP  C$ESEG
      043236 104405
7008 043240 006301      ASL    R1              ;NEXT SLIDING ONES DATA
7009 043242 103323      BCC    40$            ;BRANCH IF CARRY IS CLEARED
7010
7011      ;THE FOLLOWING CODE IS FOR SLIDING ZERO
7012
7013 043244 012701 177776      90$:  MOV    #177776,R1    ;LOAD MASK FOR SLIDING ZEROS
7014 043250      BGNSEG
      043250 104404      TRAP  C$BSEG
7015 043252 010304      MOV    R3,R4          ;SAVE WAIT COUNTER VALUE
7016 043254 010177 140422      MOV    R1,@DAT        ;LOAD OUTPUT DATA REGISTER
7017 043260 004737 011514      100$: JSR    PC,WT500     ;WAIT 500 US
7018 043264 005304      DEC    R4             ;DECREMENT WAIT COUNTER
7019 043266 001374      BNE    100$          ;BRANCH IF COUNTER IS NOT ZERO
7020 043270 017737 140406 004064      MOV    @DAT,BAD       ;READ DIGITAL INPUT PATTERN
7021 043276 020137 004064      CMP    R1,BAD        ;CMP READ AND LOADED DATA
7022 043302 004737 012146      CALL   INSERT          ;SKIP BRANCH IF 'SFI' IS SET
7023 043306 001406      BEQ    110$          ;BRANCH IF EQUAL
7024 043310 010137 004062      MOV    R1,GOOD        ;GET GOOD PATTERN
7025 043314      ERRSOFT 1305,E1302,EER2 ;ERROR HANDLER
      043314 104457      TRAP  C$ERSOFT
      043316 002431      .WORD 1305
      043320 043536      .WORD E1302
      043322 007400      .WORD EER2
7026 043324      110$: CKLOOP
      043324 104406      TRAP  C$CLP1
7027 043326 012737 002432 007200      MOV    #1306.,ERRNBR  ;LOAD ERROR NUMBER
7028 043334 004737 016326      CALL   READ           ;READ D/O LINES WITH DI MODULE
7029 043340 103002      BCC    120$          ;BRANCH IF NO ADDR. ERROR FOUND
7030 043342      EXIT   TST           ;EXIT TEST IF ADDRESSING ERROR
      043342 104432      TRAP  C$EXIT
      043344 000334      .WORD L10046-.
7031 043346 020137 004064      120$: CMP    R1,BAD          ;CMP READ AND LOADED DATA
7032 043352 004737 012146      CALL   INSERT          ;SKIP BRANCH IF 'SFI' IS SET
7033 043356 001406      BEQ    130$            ;BRANCH IF EQUAL
7034 043360 010137 004062      MOV    R1,GOOD        ;GET GOOD PATTERN
7035 043364      ERRSOFT 1307,E1303,EER2 ;ERROR HANDLER
      043364 104457      TRAP  C$ERSOFT
      043366 002433      .WORD 1307
      043370 043622      .WORD E1303

```

TEST 13: DIGITAL OUTPUT LOOPBACK TEST - SLIDING PATTERN

```

043372 007400
7036 043374 130$: ENDSEG .WORD EER2
043374 104405 10001$: TRAP C$ESEG
7037 043376 006301 ASL R1 ;NEXT SLIDING ZEROS DATA
7038 043400 103002 BCC 140$ ;BRANCH IF ALL LINES WHERE ZERO
7039 043402 005501 ADC R1 ;ADD THE CARRY TO R1
7040 043404 000721 BR 90$ ;AND REPEAT
7041 043406 005302 140$: DEC R2 ;ALL CYCLES PERFORMED ?
7042 043410 001402 BEQ 150$ ;BRANCH IF YES
7043 043412 000137 043106 JMP 30$ ;IF NO, DO NEXT CYCLE
7044 043416 005737 002242 150$: TST QVP ;IS QUICK VERIFY PASS SELECTED?
7045 043422 001010 BNE EXQV13 ;IF YES EXIT TEST
7046 043424 005237 004144 INC ITRCNT ;ITERATION COUNTER + 1
7047 043430 023737 004146 004144 CMP ITRDEF,ITRCNT ;DEFAULT ITERATION EXECUTED
7048 043436 001402 BEQ EXQV13 ;IF YES, EXIT TEST
7049 043440 000137 043020 JMP ITRA13 ;IF NO, TEST ITERATION
7050 043444 EXQV13: EXIT TST ;EXIT TEST
043444 104432 TRAP C$EXIT
043446 000232 .WORD L10046-
7051
7052 .NLIST BEX
7053 043450 045 123 062 TSHD13: .ASCIZ /#S2#ADIGITAL OUTPUT LOOPBACK TEST - SLIDING PATTERN#N/
7054 043536 104 101 124 E1302: .ASCIZ /DATA REGISTER OF OUTPUT MODULE INCORRECT AFTER LOAD/
7055 043622 122 105 103 E1303: .ASCIZ /RECEIVED DATA INCORRECT - CHECK OUTPUT LINES/
7056 .LIST BEX
7057 .EVEN
7058
7059 043700 ENDTST
043700
043700 104401 L10046: TRAP C$ETST

```

TEST 14: DIGITAL INPUT LOOPBACK TEST - INPUT MODULE INTERRUPT

```

7061 .SBTTL TEST 14: DIGITAL INPUT LOOPBACK TEST - INPUT MODULE INTERRUPT LINE.
7062 ;*****
7063 ; TEST 14 - DIGITAL LOOPBACK TEST - INPUT MODULE INTERRUPT LINE.
7064 ;
7065 ; THIS CHECKS THAT THE CSA INTERRUPT BIT OF THE INPUT MODULE CAN BE SET
7066 ; TO CAUSE AN INTERRUPT BY THE LEADING OR TRAILING EDGES OF THE MODULE
7067 ; DATA LINE 15. THE DEBOUNCE PERIOD FOR THE DIGITAL INPUT MODULE
7068 ; IS DEPENDENT ON THE OUTPUT MODULE THAT YOU SELECT AS THE 'OTHER' IN THE
7069 ; HARDWARE QUESTIONS. IT IS ALSO CHECKED THAT BIT 15 REMAINS HIGH AFTER THE
7070 ; LINE HAS REVERTED BACK TO ITS INITIAL STATE. ALL SIGNALS ARE PROVIDED BY
7071 ; LINES FROM THE OUTPUT MODULE.
7072 ; ALSO THE BITS 0-14 ARE SWITCHED ON AND OFF FOUR TIMES AND THEN A CHECK IS
7073 ; MADE THAT BIT 15 IS NOT AFFECTED.
7074 ; ALL OF THE FUNCTIONAL TESTS ABOVE ARE ALSO CARRIED OUT IF LOW LEVEL
7075 ; TESTING IS SELECTED.
7076 ;*****
7077 043702          BGNTST
7078 043702          T14::
7079 043702 004737 011714 CALL SELECT          ;CALL SELECT ROUTINE
7080 043706 002001          .WORD 2001          ;GIVE TEST PARAMETER
7081 043710 045612          TSHD14          ;GIVE TEST HEADER ADDRESS
7082 043712 103002          BCC 1$          ;IF CARRY IS SET, EXIT TEST
7083 043714          EXIT TST          ;EXIT TEST
7084 043714 104432          TRAP C$EXIT
7085 043716 002464          .WORD L10047-.
7086 043720 005037 004144 1$: CLR ITRCNT          ;CLEAR ITERATION COUNTER
7087 043724 ITRA14: SETVEC VEC,#INTSR,#PRI07 ;SET VECTOR AND SERVICE ROUTINE
7088 043724 012746 000340 MOV #PRI07,-(SP)
7089 043730 012746 017276 MOV #INTSR,-(SP)
7090 043734 013746 003710 MOV VEC,-(SP)
7091 043740 012746 000003 MOV #3,-(SP)
7092 043744 104437          TRAP C$SVEC
7093 043746 062706 000010 ADD #10,SP
7094 043752          SETPRI #PRI07          ;DISABLE INTERRUPT
7095 043752 012700 000340 MOV #PRI07,RO
7096 043756 104441          TRAP C$SPRI
7097 043760 005037 004150 CLR INTFLA          ;CLEAR INTERRUPT FLAG
7098 043764 012737 002571 007200 MOV #1401.,ERRNBR ;LOAD ERROR NUMBER
7099 043772 004737 016122 JSR PC,WRITE          ;TEST SELECTED OUTPUT ADDR.
7100 043776 103002          BCC 10$          ;BRANCH IF NO ADDR. ERROR FOUND
7101 044000          EXIT TST          ;EXIT TEST IF ADDRESSING ERROR
7102 044000 104432          TRAP C$EXIT
7103 044002 002400          .WORD L10047-.
7104 044004 005737 003716 10$: TST LOWLVL          ;IS LOW LEVEL SELEC. IN P-TABLE
7105 044010 001402          BEQ 20$          ;BRANCH IF NO
7106 044012 000137 044704 JMP LLPA14          ;JUMP TO LOW LEVEL PART
7107 044016 112777 000001 137654 20$: MOVB #1,@MOD          ;LOAD 500US AS INPUT DEBOUNCE
7108 044024 012703 000003 MOV #3,R3          ;LOAD DEBOUNCE WAIT COUNTER WITH 1.5MS
7109 044030 032777 000001 137664 BIT #1,@OTHMOD          ;IS DEBOUNCE OF OUTPUT MOD 500 US?
7110 044036 001007          BNE 30$          ;BRANCH IF 500 US DEBOUNCE IS FOUND
7111 044040 112777 000002 137632 MOVB #2,@MOD          ;IF NO, LOAD 5 MS AS INPUT DEBOUNCE
7112 044046 012703 000024 MOV #20.,R3          ;LOAD DEBOUNCE WAIT COUNTER WITH 10MS
7113 044052 062703 000010 ADD #8.,R3          ;ADD 4 MS DEBOUNCE TOLLERANCE
7114 044056 052777 000100 137614 30$: BIS #100,@MOD          ;SWITCH ON THE MODULE LED AGAIN
7115 044064 104404          TRAP C$BSEG
7116 044066 005001          CLR R1          ;DATA FOR WRITE ROUTINE

```

TEST 14: DIGITAL INPUT LOOPBACK TEST - INPUT MODULE INTERRUPT

```

7104 044070 010304          40$:  MOV    R3,R4          ;SAVE WAIT COUNTER
7105 044072 004737 016122    JSR    PC,WRITE      ;SEND DATA
7106 044076 103002          BCC    50$          ;BRANCH IF NO ADDR. ERROR FOUND
7107 044100          EXIT    TST          ;EXIT TEST IF ADDRESSING ERROR
      044100 104432          TRAP   C$EXIT
      044102 002300          .WORD L10047-.
7108 044104 004737 011514    50$:  JSR    PC,WT500      ;WAIT 500 US
7109 044110 005304          DEC    R4           ;...
7110 044112 001374          BNE    50$          ;BRANCH IF NOT FINISHED
7111 044114 017737 137562 004064  MOV    @DAT,BAD     ;GET DATA REGISTER CONTENTS
7112 044122 005737 004064    TST    BAD          ;IS DATA REG. OF INPUT MODULE ZERO
7113 044126 004737 012146    CALL   INSERT       ;SKIP BRANCH IF 'SFI' IS SET
7114 044132 001406          BEQ    60$          ;BRANCH IF YES
7115 044134 005037 004062    CLR    GOOD         ;SET UP DATA FOR ERROR HANDLER
7116 044140          ERRSOFT 1402,E1402,EER2 ;ERROR HANDLER
      044140 104457          TRAP   C$ERSOFT
      044142 002572          .WORD 1402
      044144 045703          .WORD E1402
      044146 007400          .WORD EER2
7117 044150          60$:  CKLOOP          ;
      044150 104406          TRAP   C$CLP1
7118 044152          SETPRI  #PRI00      ;ENABLE INTERRUPTS
      044152 012700 000000          MOV    #PRI00,R0
      044156 104441          TRAP   C$SPRI
7119 044160 012701 100000    MOV    #100000,R1   ;DATA (BIT 15 SET) FOR WRITE ROUTINE
7120 044164 012777 060000 137512  MOV    #60000,@CSA ;ENABLE INT. * LEADING EDGE
7121 044172 010304          MOV    R3,R4       ;RELOAD WAIT COUNTER
7122 044174 004737 016122    JSR    PC,WRITE      ;SEND DATA
7123 044200 103002          BCC    70$          ;BRANCH IF NO ADDR. ERROR FOUND
7124 044202          EXIT    TST          ;EXIT TEST IF ADDRESSING ERROR
      044202 104432          TRAP   C$EXIT
      044204 002176          .WORD L10047-.
7125 044206 004737 011514    70$:  JSR    PC,WT500      ;WAIT 500 US
7126 044212 005304          DEC    R4           ;...
7127 044214 001374          BNE    70$          ;BRANCH IF NOT FINISHED
7128 044216 005737 004150    TST    INTFLA       ;WAS THERE AN INTERRUPT ON LEAD.
7129 044222 004737 012146    CALL   INSERT       ;SKIP BRANCH IF 'SFI' IS SET
7130 044226 001004          BNE    80$          ;BRANCH IF YES
7131 044230          ERRSOFT 1403,E1403,EER3 ;ERROR HANDLER
      044230 104457          TRAP   C$ERSOFT
      044232 002573          .WORD 1403
      044234 045764          .WORD E1403
      044236 007442          .WORD EER3
7132 044240          80$:  CKLOOP          TRAP   C$CLP1
      044240 104406          ;
7133 044242 017737 137436 004064  MOV    @CSA,BAD     ;GET CSA CONTENTS
7134 044250 022737 160000 004064  CMP    #160000,BAD  ;IS IR,IE AND ELE SET
7135 044256 004737 012146    CALL   INSERT       ;SKIP BRANCH IF 'SFI' IS SET
7136 044262 001407          BEQ    90$          ;BRANCH IF YES
7137 044264 012737 160000 004062  MOV    #160000,GOOD ;SET UP DATA FOR ERROR HANDLER
7138 044272          ERRSOFT 1404,E1404,EER1 ;ERROR HANDLER
      044272 104457          TRAP   C$ERSOFT
      044274 002574          .WORD 1404
      044276 046034          .WORD E1404
      044300 007342          .WORD EER1
7139 044302          90$:  ENDSEG
      044302

```

10000\$:

TEST 14: DIGITAL INPUT LOOPBACK TEST - INPUT MODULE INTERRUPT

```

044302 104405                                TRAP    C$ESEG
7140
7141      ;NOW WE TEST THAT WE GET AN INTERRUPT ON THE TRAILING EDGE OF BIT 15
7142
7143 044304                                BGNSEG
044304 104404                                TRAP    C$BSEG
7144 044306 012701 077777                    MOV     #77777,R1                    ;DATA (BIT 15 CLEAR) FOR WRITE ROUTINE
7145 044312 012777 050000 137364            MOV     #50000,@CSA                  ;ENABLE INT. + TRAILING EDGE
7146 044320 052777 100000 137356            BIS     #100000,@CSA                 ;CLEAR IR BIT IN CSA
7147 044326 017737 137352 004064            MOV     @CSA,BAD                      ;GET CSA CONTENTS
7148 044334 022737 050000 004064            CMP     #50000,BAD                   ;IS IR CLEARED AND EI+ETE SET?
7149 044342 004737 012146                    CALL    INSERT                         ;SKIP BRANCH IF 'SFI' IS SET
7150 044346 001407 012146                    BEQ     100$                           ;BRANCH IF YES
7151 044350 012737 050000 004062            MOV     #50000,GOOD                  ;SET UP DATA FOR ERROR HANDLER
7152 044356                                ERRSOFT 1405,E1405,EER1                ;ERROR HANDLER
044356 104457                                TRAP    C$ERSOFT
044360 002575                                .WORD  1405
044362 046103                                .WORD  E1405
044364 007342                                .WORD  EER1
7153 044366                                100$:  CKLOOP                          ;
044366 104406                                TRAP    C$CLP1
7154 044370 005037 004150                    CLR     INTFLA                        ;CLEAR INTERRUPT FLAG
7155 044374 010304 004150                    MOV     R3,R4                          ;RELOAD WAIT COUNTER
7156 044376 004737 016122                    JSR     PC,WRITE                       ;SEND DATA
7157 044402 103002 016122                    BCC     110$                           ;BRANCH IF NO ADDR. ERROR FOUND
7158 044404                                EXIT     TST                            ;EXIT TEST IF ADDRESSING ERROR
044404 104432                                TRAP    C$EXIT
044406 001774                                .WORD  L10047-.
7159 044410 004737 011514                    110$:  JSR     PC,WT500                 ;WAIT 500 US
7160 044414 005304 011514                    DEC     R4
7161 044416 001374 011514                    BNE     110$
7162 044420 005737 004150                    TST     INTFLA
7163 044424 004737 012146                    CALL    INSERT                         ;WAS THERE AN INTERRUPT ON TRAIL.?
7164 044430 001004 012146                    BNE     120$                           ;SKIP BRANCH IF 'SFI' IS SET
7165 044432                                ERRSOFT 1406,E1406,EER3                ;BRANCH IF YES
044432 104457                                ;ERROR HANDLER
044434 002576                                TRAP    C$ERSOFT
044436 046163                                .WORD  1406
044440 007442                                .WORD  E1406
044442 007442                                .WORD  EER3
7166 044442                                120$:  CKLOOP                          ;
044442 104406                                TRAP    C$CLP1
7167 044444 017737 137234 004064            MOV     @CSA,BAD                      ;GET CSA CONTENTS
7168 044452 022737 150000 004064            CMP     #150000,BAD                   ;IS IR,IE AND ELE SET
7169 044460 004737 012146                    CALL    INSERT                         ;SKIP BRANCH IF 'SFI' IS SET
7170 044464 001407 012146                    BEQ     130$                           ;BRANCH IF YES
7171 044466 012737 150000 004062            MOV     #150000,GOOD                  ;SET UP DATA FOR ERROR HANDLER
7172 044474                                ERRSOFT 1407,E1404,EER1                ;ERROR HANDLER
044474 104457                                TRAP    C$ERSOFT
044476 002577                                .WORD  1407
044500 046034                                .WORD  E1404
044502 007342                                .WORD  EER1
7173 044504                                130$:  ENDSEG
044504                                10001$: TRAP    C$ESEG
044504 104405
7174
7175      ;WE NOW SWITCH BITS 0-14 ON AND OFF FOUR TIMES
7176      ;AND TEST THAT WE GET NO INTERRUPT

```

TEST 14: DIGITAL INPUT LOOPBACK TEST - INPUT MODULE INTERRUPT

```

7177
7178 044506 012705 000004      MOV      #4,R5          ;LOAD FOUR TIMES LOOP COUNTER
7179 044512 012777 070000 137164  MOV      #70000,@CSA    ;SET IE,ETE AND ELE IN CSA
7180 044520 052777 100000 137156  BIS      #100000,@CSA   ;CLEAR IR BIR IN CSA
7181 044526
      044526 104404                                TRAP      C$BSEG
7182 044530 005001      140$: CLR      R1          ;DATA FOR WRITE ROUTINE
7183 044532 005037 004150      150$: CLR      INTFLA    ;CLEAR INTERRUPT FLAG
7184 044536 010304      MOV      R3,R4        ;RELOAD WAIT COUNTER
7185 044540 004737 016122      JSR      PC,WRITE     ;SEND DATA AND ADD OUTPUT DEBOU.
7186 044544 103002      BCC      160$        ;BRANCH IF NO ADDR. ERROR FOUND
7187 044546
      044546 104432                                TRAP      C$EXIT
      044550 001632                                .WORD    L10047-.
7188 044552 004737 011514      160$: JSR      PC,WT500  ;WAIT 500 US
7189 044556 005304      DEC      R4          ;...
7190 044560 001374      BNE      160$        ;BRANCH IF NOT FINISHED
7191 044562 005737 004150      TST      INTFLA     ;NO INTERRUPT SHOULD BE SHOWN
7192 044566 004737 012146      CALL     INSERT     ;SKIP BRANCH IF 'SFI' IS SET
7193 044572 001404      BEQ      170$        ;BRANCH IF NO INTERRUPT OCCURRED
7194 044574
      044574 104457                                TRAP      C$ERSOFT
      044576 002600                                .WORD    1408
      044600 046234                                .WORD    E1408
      044602 007442                                .WORD    EER3
7195 044604
      044604 104406      170$: CKLOOP                                TRAP      C$CLP1
7196 044606 017737 137072 004064      MOV      @CSA,BAD    ;GET CSA CONTENTS
7197 044614 022737 070000 004064      CMP      #70000,BAD  ;IS IR,IE AND ETE SET
7198 044622 004737 012146      CALL     INSERT     ;SKIP BRANCH IF 'SFI' IS SET
7199 044626 001407      BEQ      180$        ;BRANCH IF YES
7200 044630 012737 070000 004062      MOV      #70000,GOOD ;SET UP DATA FOR ERROR HANDLER
7201 044636
      044636 104457                                TRAP      C$ERSOFT
      044640 002601                                .WORD    1409
      044642 046315                                .WORD    E1409
      044644 007342                                .WORD    EER1
7202 044646
      044646 104406      180$: CKLOOP                                ;
      ;
      ;                                TRAP      C$CLP1
7203 044650 005705      TST      R5          ;HAVE WE DONE THIS ROUTINE 4 TIMES?
7204 044652 001006      BNE      190$        ;BRANCH IF YES
7205 044654 005305      DEC      R5          ;DECREMENT FOUR TIMES LOOP COUNTER
7206 044656 005701      TST      R1          ;USED DATA ALL ONES?
7207 044660 001723      BEQ      140$        ;BRANCH IF YES
7208 044662 012701 077777      MOV      #77777,R1   ;IF NO, LOAD ONES DATA WRITE ROUTINE
7209 044666 000721      BR       150$        ;BRANCH BACK TO WRITE ROUTINE
7210 044670
      044670 012700 000340      190$: SETPRI  #PRI07  ;DISABLE INTERRUPTS
      044674 104441                                MOV      #PRI07,R0
      ;                                TRAP      C$SPRI
7211 044676
      044676                                ENDSEG
      044676 104405                                10002$: TRAP      C$ESEG
7212 044700 000137 045560      JMP      QVR14       ;JUMP TO QUICK VERIFY ROUTINE
7213
7214
7215
7216
      ;IF LOW LEVEL WAS REQUESTED IN: THE SOFTWARE P-TABLE, THE FOLLOWING
      ;CODE WILL BE EXECUTED. WE WILL TEST THE SAME AS ABOVE BUT WITH LOW LEVEL

```

TEST 14: DIGITAL INPUT LOOPBACK TEST - INPUT MODULE INTERRUPT

```

7217                                     ;DEBOUNCE VALUE. THE INPUT MODULE DEBOUNCE WILL BE SELECTED AS 5 MS.
7218
7219
7220 044704 052777 000002 136766 LLPA14: BIS      #2,@MOD      ;LOAD AN INPUT DEBOUNCE OF 5 MS
7221 044712 042777 000001 136760      BIC      #1,@MOD      ;...
7222 044720 052777 000010 136752      BIS      #10,@MOD     ;SWITCH INPUT MODULE TO LLS
7223 044726 013702 004156          MOV      CYCLLS,R2    ;LOAD CYCLE COUNTER
7224 044732 013703 004162          MOV      LLWC,R3     ;LOAD LLS WAIT COUNTER
7225 044736          BGNSEG
7226 044740 005001          CLR      R1          ;DATA FOR WRITE ROUTINE
7227 044742 010304          MOV      R3,R4      ;SAVE WAIT COUNTER
7228 044744 012737 002602 007200 10$:  MOV      #1410.,ERRNBR ;LOAD ERROR NUMBER FOR WRITE ROUT.
7229 044752 004737 016122          JSR      PC,WRITE    ;SEND DATA
7230 044756 103002          BCC     20$         ;BRANCH IF NO ADDR. ERROR FOUND
7231 044760          EXIT     TST          ;EXIT TEST IF ADDRESSING ERROR
7232 044760 104432          TRAP   C$EXIT     ;...
7233 044762 001420          .WORD  L10047-.
7234 044764 004737 011514          20$:  JSR      PC,WT500   ;WAIT 500 US
7235 044770 005304          DEC     R4          ;...
7236 044772 001374          BNE     20$         ;BRANCH IF NOT FINISHED
7237 044774 017737 136702 004064 30$:  MOV      @DAT,BAD    ;GET DATA REGISTER CONTENTS
7238 045002 005737 004064          TST     BAD         ;IS DATA REG. OF INPUT MODULE ZERO?
7239 045006 004737 012146          CALL   INSERT       ;SKIP BRANCH IF 'SFI' IS SET
7240 045012 001406          BEQ     30$         ;BRANCH IF YES
7241 045014 005037 004062          CLR     GOOD        ;SET UP DATA FOR ERROR HANDLER
7242 045020          ERRSOFT 1411,E1402,EER2 ;ERROR HANDLER
7243 045020 104457          TRAP   C$ERSOFT   ;...
7244 045022 002603          .WORD  1411
7245 045024 045703          .WORD  E1402
7246 045026 007400          .WORD  EER2
7247 045030          30$:  CKLOOP
7248 045030 104406          :
7249 045032          SETPRI #PRI00      ;ENABLE INTERRUPTS
7250 045032 012700 000000          MOV     #PRI00,RO   ;...
7251 045036 104441          TRAP   C$SPRI
7252 045040 012701 100000          MOV     #100000,R1  ;DATA (BIT 15 SET) FOR WRITE ROUTINE
7253 045044 012777 060000 136632 40$:  MOV     #60000,@CSA ;ENABLE INT. + LEADING EDGE
7254 045052 010304          MOV     R3,R4      ;RELOAD WAIT COUNTER
7255 045054 004737 016122          JSR     PC,WRITE    ;SEND DATA
7256 045060 103002          BCC     40$         ;BRANCH IF NO ADDR. ERROR FOUND
7257 045062          EXIT     TST          ;EXIT TEST IF ADDRESSING ERROR
7258 045062 104432          TRAP   C$EXIT     ;...
7259 045064 001316          .WORD  L10047-.
7260 045066 004737 011514          40$:  JSR     PC,WT500   ;WAIT 500 US
7261 045072 005304          DEC     R4          ;...
7262 045074 001374          BNE     40$         ;BRANCH IF NOT FINISHED
7263 045076 005737 004150          TST     INTFLA     ;WAS THERE AN INTERRUPT?
7264 045102 004737 012146          CALL   INSERT       ;SKIP BRANCH IF 'SFI' IS SET
7265 045106 001004          BNE     50$         ;BRANCH IF YES
7266 045110          ERRSOFT 1412,E1403,EER3 ;ERROR HANDLER
7267 045110 104457          TRAP   C$ERSOFT   ;...
7268 045112 002604          .WORD  1412
7269 045114 045764          .WORD  E1403
7270 045116 007442          .WORD  EER3
7271 045120          50$:  CKLOOP
7272 045120 104406          TRAP   C$CLP1

```

TEST 14: DIGITAL INPUT LOOPBACK TEST - INPUT MODULE INTERRUPT

```

7257 045122 017737 136556 004064      MOV      @CSA,BAD      ;GET CSA CONTENTS
7258 045130 022737 160000 004064      CMP      #160000,BAD  ;IS IR,IE AND ELE SET
7259 045136 004737 012146              CALL     INSERT       ;SKIP BRANCH IF 'SFI' IS SET
7260 045142 001407              BEQ      60$          ;BRANCH IF YES
7261 045144 012737 160000 004062      MOV      #160000,GOOD ;SET UP DATA FOR ERROR HANDLER
7262 045152              ERRSOFT 1413,E1404,EER1 ;ERROR HANDLER
              045152 104457              TRAP    C$ERSOFT
              045154 002605              .WORD  1413
              045156 046034              .WORD  E1404
              045160 007342              .WORD  EER1
7263 045162              60$:      ENDSEG
              045162              10003$:
              045162 104405              TRAP    C$ESEG
7264
7265              ;WE NOW TEST THAT WE GET AN INTERRUPT ON THE TRAILING EDGE OF BIT 15
7266
7267 045164              BGNSEG
              045164 104404              TRAP    C$BSEG
7268 045166 012701 077777              MOV      #77777,R1    ;DATA (BIT 15 CLEAR) FOR WRITE ROUTINE
7269 045172 012777 050000 136504      MOV      #50000,@CSA ;ENABLE INT. + TRAILING EDGE
7270 045200 052777 100000 136476      BIS      #100000,@CSA ;CLEAR IR BIT IN CSA
7271 045206 017737 136472 004064      MOV      @CSA,BAD    ;GET CSA CONTENTS
7272 045214 022737 050000 004064      CMP      #50000,BAD  ;IS IR CLEARED AND EI+ETE SET
7273 045222 004737 012146              CALL     INSERT       ;SKIP BRANCH IF 'SFI' IS SET
7274 045226 001407              BEQ      70$          ;BRANCH IF YES
7275 045230 012737 050000 004062      MOV      #50000,GOOD ;SET UP DATA FOR ERROR HANDLER
7276 045236              ERRSOFT 1414,E1405,EER1 ;ERROR HANDLER
              045236 104457              TRAP    C$ERSOFT
              045240 002606              .WORD  1414
              045242 046103              .WORD  E1405
              045244 007342              .WORD  EER1
7277 045246              70$:      CKLOOP
              045246 104406              TRAP    C$CLP1
7278 045250 005037 004150      CLR      INTFLA      ;CLEAR INTERRUPT FLAG
7279 045254 010304              MOV      R3,R4       ;RELOAD WAIT COUNTER
7280 045256 004737 016122      JSR      PC,WRITE    ;SEND DATA
7281 045262 103002              BCC      80$         ;BRANCH IF NO ADDR. ERROR FOUND
7282 045264              EXIT      TST        ;EXIT TEST IF ADDRESSING ERROR
              045264 104432              TRAP    C$EXIT
              045266 001114              .WORD  L10047-.
7283 045270 004737 011514      80$:      JSR      PC,WT500    ;WAIT 500 US
7284 045274 005304              DEC      R4          ;...
7285 045276 001374              BNE      80$         ;BRANCH IF NOT FINISHED
7286 045300 005737 004150      TST      INTFLA     ;WAS THERE AN INTERRUPT ON TRAIL?
7287 045304 004737 012146      CALL     INSERT       ;SKIP BRANCH IF 'SFI' IS SET
7288 045310 001004              BNE      90$         ;BRANCH IF YES
7289 045312              ERRSOFT 1415,E1406,EER3 ;ERROR HANDLER
              045312 104457              TRAP    C$ERSOFT
              045314 002607              .WORD  1415
              045316 046163              .WORD  E1406
              045320 007442              .WORD  EER3
7290 045322              90$:      CKLOOP
              045322 104406              TRAP    C$CLP1
7291 045324 017737 136354 004064      MOV      @CSA,BAD    ;GET CSA CONTENTS
7292 045332 022737 150000 004064      CMP      #150000,BAD ;IS IR,IE AND ELE SET
7293 045340 004737 012146              CALL     INSERT       ;SKIP BRANCH IF 'SFI' IS SET
7294 045344 001407              BEQ      100$        ;BRANCH IF YES

```


TEST 14: DIGITAL INPUT LOOPBACK TEST - INPUT MODULE INTERRUPT

```

7295 045346 012737 150000 004062      MOV      #150000,GOOD      ;SET UP DATA FOR ERROR HANDLER
7296 045354      ERRSOFT 1416,E1404,EER1  ;ERROR HANDLER
      045354 104457
      045356 002610          TRAP    C$ERSOFT
      045360 046034          .WORD  1416
      045362 007342          .WORD  E1404
7297 045364      100$:  ENDSEG          .WORD  EER1
      045364
      045364 104405          10004$: TRAP    C$ESEG
7298
7299
7300      ;WE NOW SWITCH BITS 0-14 ON AND OFF FOUR TIMES AND TEST THAT WE
7301      ;GET NO INTERRUPT
7302 045366 012705 000004      MOV      #4,R5            ;LOAD FOUR TIMES LOOP COUNTER
7303 045372 012777 070000 136304      MOV      #70000,@CSA     ;SET IE,ETE AND ELE IN CSA
7304 045400 052777 100000 136276      BIS      #100000,@CSA   ;CLEAR IR BIT IN CSA
7305 045406      BGNSEG
      045406 104404          TRAP    C$BSEG
7306 045410 005001
7307 045412 005037 004150 110$:  CLR      R1            ;DATA FOR WRITE ROUTINE
7308 045416 010304 120$:  CLR      INTFLA        ;CLEAR INTERRUPT FLAG
7309 045420 004737 016122      MOV      R3,R4          ;RELOAD WAIT COUNTER
7310 045424 103002      JSR      PC,WRITE      ;SEND DATA AND ADD OUTPUT DEBOU.
7311 045426      BCC     130$          ;BRANCH IF NO ADDR. ERROR FOUND
      045426 104432      EXIT     TST          ;EXIT TEST IF ADDRESSING ERROR
      045430 000752          TRAP    C$EXIT
7312 045432 004737 011514 130$:  JSR      PC,WT500      ;WAIT 500 US
7313 045436 005304      DEC     R4            ;...
7314 045440 001374      BNE     130$          ;BRANCH IF NOT FINISHED
7315 045442 005737 004150      TST     INTFLA        ;WAS THERE AN INTERRUPT?
7316 045446 004737 012146      CALL    INSERT        ;SKIP BRANCH IF 'SFI' IS SET
7317 045452 001404      BEQ     140$          ;BRANCH IF YES
7318 045454      ERRSOFT 1417,E1408,EER3 ;ERROR HANDLER
      045454 104457          TRAP    C$ERSOFT
      045456 002611          .WORD  1417
      045460 046234          .WORD  E1408
      045462 007442          .WORD  EER3
7319 045464      140$:  CKLOOP          TRAP    C$CLP1
      045464 104406
7320 045466 017737 136212 004064      MOV      @CSA,BAD      ;GET CSA CONTENTS
7321 045474 022737 070000 004064      CMP      #70000,BAD    ;IS IR,IE AND ETE SET
7322 045502 004737 012146      CALL    INSERT        ;SKIP BRANCH IF 'SFI' IS SET
7323 045506 001407      BEQ     150$          ;BRANCH IF YES
7324 045510 012737 070000 004062      MOV      #70000,GOOD   ;SET UP DATA FOR ERROR HANDLER
7325 045516      ERRSOFT 1418,E1409,EER1 ;ERROR HANDLER
      045516 104457          TRAP    C$ERSOFT
      045520 002612          .WORD  1418
      045522 046315          .WORD  E1409
      045524 007342          .WORD  EER1
7326 045526      150$:  CKLOOP          TRAP    C$CLP1
      045526 104406
7327 045530 005705      TST     R5            ;HAVE WE DONE THIS ROUTINE 4 TIMES?
7328 045532 001006      BNE     160$          ;BRANCH IF YES
7329 045534 005305      DEC     R5            ;DECREMENT FOUR TIMES LOOP COUNTER
7330 045536 005701      TST     R1            ;USED DATA ALL ONES?
7331 045540 001723      BEQ     110$          ;BRANCH IF YES
7332 045542 012701 077777      MOV      #77777,R1    ;IF NO, LOAD ONES DATA WRITE ROUTINE

```

TEST 14: DIGITAL INPUT LOOPBACK TEST - INPUT MODULE INTERRUPT

```

7333 045546 000721
7334 045550 160$: BR 120$ ;BRANCH BACK TO WRITE ROUTINE
045550 012700 000340 SETPRI @PRI07 ;DISABLE INTERRUPTS
045554 104441
7335 045556 ENDSEG MOV @PRI07,R0
045556 104405 TRAP C$SPRI
045556 104405 10005$: TRAP C$ESEG
7336 045560 005737 002242 QVR14: TST QVP ;IS QUICK VERIFY PASS SELECTED?
7337 045564 001010 BNE EXQV14 ;IF YES, EXIT TEST
7338 045566 005237 004144 INC ITRCNT ;ITERATION COUNTER = 1
7339 045572 023737 004146 004144 CMP ITRDEF,ITRCNT ;DEFAULT ITERATION EXECUTED
7340 045600 001402 BEQ EXQV14 ;IF YES, EXIT TEST
7341 045602 000137 043724 JMP ITRA14 ;IF NO, TEST ITERATION
7342 045606 EXQV14: EXIT TST ;EXIT TEST
045606 104432 TRAP C$EXIT
045610 000572 .WORD L10047-
7343
7344
7345
7346 045612 045 123 062 TSHD14: .NLIST BEX
7347 045703 104 101 124 E1402: .ASCIZ /#S2#ADIGITAL INPUT LOOPBACK TEST - INTERRUPT LINE TEST#N/
7348 045764 116 117 040 E1403: .ASCIZ /DATA REGISTER CONTENTS OF INPUT MODULE INCORRECT/
7349 046034 103 123 101 E1404: .ASCIZ /NO INTERRUPT ON LEADING EDGE OF LINE 15/
7350 046103 103 123 101 E1405: .ASCIZ /CSA REGISTER INCORRECT AFTER INTERRUPT/
7351 046163 116 117 040 E1406: .ASCIZ /CSA REGISTER NOT LOADABLE WITH THE CORRECT DATA/
7352 046234 111 116 124 E1408: .ASCIZ /NO INTERRUPT ON TRAILING EDGE OF LINE 15/
7353 046315 103 123 101 E1409: .ASCIZ /INTERRUPT OCCURRED WHEN SWITCHING DATA BITS 0-14/
7354 .ASCIZ /CSA REGISTER CHANGED AFTER SWITCHING DATA BITS 0-14/
7355 .LIST BEX
7356 .EVEN
7357 046402 ENDTST
046402
046402 104401 L10047: TRAP C$ETST
7358
7370
7371 .EVEN
7372

```

HARDWARE TESTS MACRO M1200 26 OCT-83 15:22 PAGE 78

D15

TEST 14: DIGITAL INPUT LOOPBACK TEST - INPUT MODULE INTERRUPT

SEQ 0185

7380 046404
7381

ENDMOD

TEST 14: DIGITAL INPUT LOOPBACK TEST - INPUT MODULE INTERRUPT

7385
 7396
 7432
 7433 046404
 7434
 7435
 7436
 7437
 7438
 7439
 7440
 7441
 7442
 7443
 7444
 7445
 7446
 7447
 7448
 7449
 7450
 7451
 7452
 7453
 7454
 7455
 7456
 7457
 7458
 7459
 7460
 7461
 7462
 7463
 7464
 7465
 7466
 7467
 7468
 7469 046404
 046404
 7470 046404 004737 011714
 7471 046410 005004
 7472 046412 051006
 7473 046414 103002
 7474 046416
 046416 104432
 046420 004066
 7475 046422
 046422 104450
 7476 046424
 046424 103416
 7477 046426
 046426 104421
 7478 046430 032700 001000
 7479 046434 001410
 7480 046436

.TITLE HARDWARE TESTS

```

        BGNMOD
    .SBTTL TEST 15: ANALOGUE INPUT TEST - FIELD AND MANUFACTURING TEST.
    ;*****
    ; TEST 15 - ANALOGUE INPUT TEST.
    ;
    ;AT FIRST THE USER IS TOLD THAT THE TEST WILL ONLY RUN CORRECTLY IF THE
    ;MODULE HAS BEEN CALIBRATED CORRECTLY.
    ;THEN HE IS ASKED HOW MANY MUXBOARDS ARE CONNECTED THE THE CONTROLLER.
    ;THEN HE IS ASKED TO CONNECT A CALIBRATED VOLTAGE SOURCE ON THE EVEN
    ;INPUTS TO THE MODULE VIA THE ANALOGUE INPUT TEST CONNECTOR.
    ;PROMPTED BY THE PROGRAM, THE USER THEN
    ;SETS UP A SERIES OF VOLTAGES, TYPING 'CARRIAGE RETURN' AFTER EACH SO
    ;THAT THE PROGRAM CAN CHECK THE D/A CONVERSIONS.
    ;THE CONVERSION FOR 48.828 MV ARE PERFORMED ON ALL GAINS. OTHER
    ;CONVERSIONS ARE PERFORMED ONLY IN GAIN 1.
    ;IF "N" TO THE SW QUESTION "RUN MANUFACTURING TEST" WAS ANSWERED THE FOLLOWING
    ;TOLLERANCE MAY BE USED , FOR GAIN 1,2,5 IS 2 LSB, FOR GAIN 10 IS 4 LSB,
    ;FOR GAIN 20 IS 6, FOR GAIN 50 IS 7 LSB, FOR GAIN 100 IS 12 LSB AND FOR
    ;GAIN 200 IS 20 LSB.
    ;IF "Y" WAS ANSWERED THEN THE TOLLERANCE WILL BE , FOR GIAN 1,2,5 IS 2 LSB
    ;FOR GAIN 10 IS 3 LSB, FOR GIAN 20 IS 3 LSB, FOR GAIN 50 IS 5 LSB, FOR GAIN
    ;100 IS 10 LSB AND FOR GAIN 200 IS 15 LSB. IF A MUX BOARD IS CONNECTED WE WILL
    ;ADD 2 LSB MORE FOR GAINS OVER 10.
    ;NEXT THE USER IS PROMPTED TO CONNECT THE VOLTAGE SOURCE TO THE ODD INPUTS
    ;AND DO THE SAME AS FOR THE EVEN LINES ABOVE.
    ;THE VALUES USED DEPEND ON WHETHER THE MODULE IS
    ;SET UP FOR UNIPOLAR OR BIPOLAR INPUT.
    ;
    ;FINALLY, THE EXTERNAL TRIGGER FUNCTION IS TESTED BY SWITCHING THE
    ;ENABLE EXTERNAL TRIGGER ON AND OFF TO DO THE EXTERNAL START (EET IS LOOPED
    ;BACK TO THE EXTERNAL TRIGGER INPUT VIA THE ANALOGUE INPUT TEST CONNECTOR).
    ;THE EXTERNAL TRIGGER FUNCTION IS TESTED UNDER INTERRUPT.
    ;
    ;IF THE "UAM" FLAG IS SET, THIS TEST IS NOT CARRIED OUT.
    ;TO RUN THIS TEST SUCCESSFULLY, THE DEVICE HAS TO BE CALIBRATED CORRECTLY.
    ;*****
    
```

```

        BGNTST 15.
        T15::
        CALL SELECT          ;CALL SELECT ROUTINE
        .WORD 5004           ;GIVE TEST PARAMETER
        TSHD15              ;GIVE TEST HEADER ADDRESS
        BCC 1$              ;IF CARRY IS SET, EXIT TEST
        EXIT TST            ;EXIT TEST
                                TRAP C$EXIT
                                .WORD L10050-
    1$: MANUAL                ;IS MANUAL INTERVENTION ALLOWED?
                                TRAP C$MANI
        BCOMPLETE 10$       ;IF YES, BRANCH (UAM FLAG NOT SET)
                                BCS 10$
        RFLAGS RO           ;READ OPERATOR FLAGS INTO RO
                                TRAP C$RFLA
        BIT #PNT,RO         ;PRINT MESSAGES ?
        BEQ 3$              ;IF NO, EXIT
        PRINTF #!M15        ;IF YES PRINT TEST DISABLED
    
```


TEST 15: ANALOGUE INPUT TEST - FIELD AND MANUFACTURING TEST.

```

046674 000406
046676 050656
046700 000042
046702 051340
046704 177777
046706 000000
046710 000007
046712
7500 046712 012737 050726 050674      MOV    #FILTOL,TOLOF
7501 046720 005737 002232              TST    MANTST
7502 046724 001411              BEQ    PS151
7503 046726 012737 050746 050674      MOV    #MAFTOL,TOLOF
7504 046734 005737 050656              TST    ANS151
7505 046740 001403              BEQ    PS151
7506 046742 012737 050766 050674      MOV    #MUXTOL,TOLOF
7507 046750 012703 000014      PS151: MOV    #14,R3
7508 046754 012737 000024 050670      MOV    #24,VOLTE
7509 046762 005737 050664              TST    BIPOL
7510 046766 001004              BNE    10$
7511 046770 005003              CLR    R3
7512 046772 012737 000010 050670      MOV    #10,VOLTE
7513 047000 005737 050666      10$:  TST    MUXC
7514 047004 001001              BNE    20$
7515 047006 005004              CLR    R4
7516 047010 042704 007400      20$:  BIC    #7400,R4
7517 047014 005037 050662              CLR    ANS153
7518 047020              PRINTF #MES152
047020 012746 051415
047024 012746 000001
047030 010600
047032 104417
047034 062706 000004
7519 047040 016301 050676      MOV    VOLVA(R3),R1
7520 047044 016302 050700      MOV    VOLVA+2(R3),R2
7521 047050 004737 015764      CALL   DECOU
7522 047054              PRINTF #OUT15
047054 012746 051455
047060 012746 000001
047064 010600
047066 104417
047070 062706 000004
7523 047074              30$:  GMANIL MES153,ANS152,-1,YES
047074 104443
047076 000404
047100 050660
047102 000130
047104 051474
047106 177777
047110
7524 047110 005737 050660      TST    ANS152
7525 047114 001767              BEQ    30$
7526 047116 005037 004134              CLR    MODE
7527 047122 005737 050664              TST    BIPOL
7528 047126 001403              BEQ    50$
7529 047130 012737 000001 004134      MOV    #1,MODE
7530 047136 022763 000060 050676 50$:  CMP    #48.,VOLVA(R3)
7531 047144 001016              BNE    40$

```

```

BR      10000$
.WORD  ANS151
.WORD  T$CODE
.WORD  MES151
.WORD  -1
.WORD  T$LOLIM
.WORD  T$HILIM

10000$:
;GET FIELD TOLLERANCE TABLE
;ARE MANUFACTURING TESTS REQUIRED ?
;BRANCH IF NOT
;GET MANUFACTURING TOLLERANCE TABLE
;ARE THERE SOME MUX BOARDS CONNECTED
;BRANCH IF NOT
;GET MUX TOLLERANCE TABLE
;POINT R3 TO BIPOLAR TABLE (VOLVA+14)
;LOAD BIPOLAR VOLTAGE TABLE END ADDRESS
;IS BIPOLAR SELECTED ?
;BRANCH IF YES
;POINT R3 TO UNIPOLAR TABLE (VOLVA)
;LOAD UNIP. VOLTAGE TABLE END ADDR.
;ARE WE TESTING A MUX BOARDED ?
;BRANCH IF YES
;BEGIN WITH CHANNEL ZERO
;BEGIN WITH FIRST CHANNEL IN BANK
;CLEAR ODD/EVEN FLAG
;PRINT VOLT SOURCE ADJUSTMENT
MOV    #MES152,-(SP)
MOV    #1,-(SP)
MOV    SP,R0
TRAP   C$PNTF
ADD    #4,SP

;LOAD DECOU INPUT (R1)
;...
;PRINT DECIMAL NUMBER
;PRINT VOLT
MOV    #OUT15,-(SP)
MOV    #1,-(SP)
MOV    SP,R0
TRAP   C$PNTF
ADD    #4,SP

;AND SOURCE CONNECTION
TRAP   C$GMAN
BR      10001$
.WORD  ANS152
.WORD  T$CODE
.WORD  MES153
.WORD  -1

10001$:
;TEST ANSWER OF SOURCE CONNECTION
;GO ON IF ANSWER WAS YES
;LOAD ADCON INPUT (UNIPOLAR)
;IS BIPOLAR SELECTED ?
;BRANCH IF YES
;LOAD ADCON INPUT (BIPOLAR)
;IS THE USED VOLTAGE 48.828 MV
;BRANCH IF NOT

```

TEST 15: ANALOGUE INPUT TEST - FIELD AND MANUFACTURING TEST.

7532	047146	012702	003720		MOV	#2000.,R2							
7533	047152				PRINTF	#PME155							
	047152	012746	052467										
	047156	012746	000001										
	047162	010600											
	047164	104417											
	047166	062706	000004										
7534	047172	004737	011506	41\$:	CALL	WT25M							
7535	047176	005302			DEC	R2							
7536	047200	001374			BNE	41\$							
7537	047202	005005		40\$:	CLR	R5							
7538	047204	010537	004136	60\$:	MOV	R5,GAIN							
7539	047210				BGNSEG								
	047210	104404											
7540	047212	042777	077536	134464	BIC	#77536,@CSA							
7541	047220	006305			ASL	R5							
7542	047222	050577	134456		BIS	R5,@CSA							
7543	047226	012737	100000	004062	MOV	#100000,GOOD							
7544	047234	050537	004062		BIS	R5,GOOD							
7545	047240	006205			ASR	R5							
7546	047242	050477	134436		BIS	R4,@CSA							
7547	047246	017737	134432	004064	MOV	@CSA,BAD							
7548	047254	050437	004062		BIS	R4,GOOD							
7549	047260	023737	004062	004064	CMF	GOOD,BAD							
7550	047266	004737	012146		CALL	INSERT							
7551	047272	001404			BEQ	70\$							
7552	047274				ERRSOFT	1501,E1501,EER4							
	047274	104457											
	047276	002735											
	047300	051726											
	047302	007474											
7553	047304			70\$:	CKLOOP								
	047304	104406											
7554	047306	052777	000001	134370	BIS	#1,@CSA							
7555	047314	005002			CLR	R2							
7556	047316	017737	134362	004064	MOV	@CSA,BAD							
7557	047324	032737	000200	004064	BIT	#200,BAD							
7558	047332	001013			BNE	90\$							
7559	047334	005302			DEC	R2							
7560	047336	004737	012146		CALL	INSERT							
7561	047342	001365			BNE	80\$							
7562	047344	052737	000200	004062	BIS	#200,GOOD							
7563	047352				ERRSOFT	1502,E1502,EER4							
	047352	104457											
	047354	002736											
	047356	051755											
	047360	007474											
7564	047362			90\$:	CKLOOP								
	047362	104406											
7565													
7566													
7567													
7568	047364	012702	000012		MOV	#10.,R2							
7569	047370	004737	011514	100\$:	CALL	WT500							
7570	047374	005302			DEC	R2							
7571	047376	001374			BNE	100\$							
7572	047400	017737	134276	004064	MOV	@DAT,BAD							

TEST 15: ANALOGUE INPUT TEST - FIELD AND MANUFACTURING TEST.

```

7664 050104 010400          MOV      R4,R0          ;GET R4 CONTENTS (CHANNEL)
7665 050106 042700 170377  BIC      #170377,R0     ;CLEAR UNNEEDED BITS
7666 050112 022700 007400  CMP      #7400,R0      ;ALL CHANNELS TESTED ?
7667 050116 001404          BEQ      230$          ;BRANCH IF YES
7668 050120 062704 000400  ADD      #400,R4       ;INCREMENT CHANNEL NUMBER
7669 050124 000137 047202  JMP      40$           ;BRANCH IF NOT
7670 050130 023703 050670  230$:   CMP      VOLTE,R3    ;ALL VOLTAGES REQUESTED ?
7671 050134 001414          BEQ      240$          ;BRANCH IF YES
7672 050136 062703 000004  ADD      #4,R3         ;IF NO, POINT TO NEXT VALUE
7673 050142 005737 050666  TST     MUXC           ;ARE WE TESTING MUX BOARDS ?
7674 050146 001402          BEQ      233$          ;BRANCH IF NOT
7675 050150 062703 000004  ADD      #4,R3         ;IF YES SKIP HALF SCALE TESTING
7676 050154 005737 050662  233$:   TST     ANS153     ;ARE WE TESTING ODD CHANNELS
7677 050160 001016          BNE      250$          ;BRANCH IF YES
7678 050162 000137 047000  JMP      10$           ;AND REPEAT THE TEST WITH FIRST CHA.
7679 050166 005737 050662  240$:   TST     ANS153     ;WAS ODD AND EVEN CHA. TESTED ?
7680 050172 001056          BNE      270$          ;BRANCH IF YES
7681
7682          ;NOW WE DO CONVERSION WITH VOLTAGE SOURCE ON ODD CHANNELS
7683
7684 050174 012737 000001 050662  MOV      #1,ANS153     ;SET ODD/EVEN FLAG
7685 050202 012703 000014  MOV      #14,R3        ;POINT R3 TO BIPOLAR TABLE (VOLVA+14)
7686 050206 005737 050664  TST     BIPOL          ;IS BIPOLAR SELECTED ?
7687 050212 001001          BNE      250$          ;BRANCH IF YES
7688 050214 005003          CLR      R3            ;POINT R3 TO UNIPOLAR TABLE (VOLVA)
7689 050216 042704 007400  250$:   BIC      #7400,R4     ;CLEAR BITS FOR CHANNEL 0-15
7690 050222 052704 000400  BIS      #400,R4       ;LOAD FIRST ODD CHANNEL
7691 050226          PRINTF #MES152     ;PRINT VOLT SOURCE ADJUSTMENT
          MOV      #MES152,-(SP)
          MOV      #1,-(SP)
          MOV      SP,R0
          TRAP    C$PNTF
          ADD      #4,SP
          MOV      #OUT15,-(SP)
          MOV      #1,-(SP)
          MOV      SP,R0
          TRAP    C$PNTF
          ADD      #4,SP
7692 050246 016301 050676  MOV      VOLVA(R3),R1   ;LOAD DECOUT INPUT (R1)
7693 050252 016302 050700  MOV      VOLVA+2(R3),R2 ;...
7694 050256 004737 015764  CALL     DECOUT         ;PRINT DECIMAL NUMBER
7695 050262          PRINTF #OUT15     ;PRINT VOLT
          MOV      #OUT15,-(SP)
          MOV      #1,-(SP)
          MOV      SP,R0
          TRAP    C$PNTF
          ADD      #4,SP
7696 050302 050302 104443  260$:   GMANIL MES154,ANS152,-1,YES ;CONNECT VOLT SOURCE TO J3 (ODD)
          TRAP    C$GMAN
          BR      10004$
          .WORD   ANS152
          .WORD   T$CODE
          .WORD   MES154
          .WORD   -1
          10004$:
7697 050316 005737 050660  TST     ANS152         ;TEST ANSWER
7698 050322 001767          BEQ      260$          ;ASK AGAIN IF ANSWER IS NO
7699 050324 000137 047136  JMP      50$           ;REPEAT TEST WITH ODD CHANNELS
7700
7701          ;---
7702
7703 050330 005737 050666  270$:   TST     MUXC         ;ARE WE TESTING THE CONTROLLER ?

```

TEST 15: ANALOGUE INPUT TEST - FIELD AND MANUFACTURING TEST.

```

7704 050334 001427          BEQ      EXTRI          ;BRANCH IF YES TO EXT.TRIGGER PART
7705 050336 023737 050656 050666 MUXPAR: CMP      ANS151,MUXC      ;ALL BOARDS TESTED ?
7706 050344 001542          BEQ      EXQV15         ;IF YES, EXIT TEST
7707 050346 062704 000400      ADD      #400,R4         ;POINT TO NEXT BANK
7708 050352 042777 077536 133324 BIC      #77536,@CSA     ;CLEAR CSA REGISTER
7709 050360 050477 133320      BIS      R4,@CSA        ;LOAD CHANNEL IN CSA
7710 050364          GMANIL  MES155,ANS152,-1,YES ;PUT TEST CONNECTOR TO NEXT BANK
      050364 104443          TRAP     C$GMAN
      050366 000404          BR       10005$
      050370 050660          .WORD   ANS152
      050372 000130          .WORD   T$CODE
      050374 051666          .WORD   MES155
      050376 177777          .WORD   -1
      050400          10005$:
7711 050400 042704 007400      BIC      #7400,R4        ;BEGIN TEST ITERATION WITH AN EVEN CHA.
7712 050404 005237 050666      INC      MUXC
7713 050410 000137 046750      JMP      PS151          ;COUNT MUX BOARDS
7714
7715
7716          ;THE FOLLOWING CODE IS USED FOR TEST EXTERNAL TRIGGER FUNCTION
7717
7718 050414          EXTRI: SETVEC  VEC,#INTSR,#PRI07 ;SET VECTOR 1 AND SERVICE ROUTINE
      050414 012746 000340          MOV      #PRI07,-(SP)
      050420 012746 017276          MOV      #INTSR,-(SP)
      050424 013746 003710          MOV      VEC,-(SP)
      050430 012746 000003          MOV      #3,-(SP)
      050434 104437          TRAP    C$SVEC
      050436 062706 000010          ADD      #10,SP
7719 050442          SETPRI  #PRI07          ;DISABLE INTERRUPTS
      050442 012700 000340          MOV      #PRI07,RO
      050446 104441          TRAP    C$SPRI
7720 050450          BGNSEG
      050450 104404          TRAP    C$BSEG
7721 050452 005037 004150      CLR      INTFLA        ;CLEAR INTERRUPT FLAG
7722 050456 005005      CLR      R5            ;CLEAR GAIN POINTER
7723 050460 042777 077536 133216 20$: BIC      #77536,@CSA     ;CLEAR ALL R/W BITS IN CSA
7724 050466 050477 133212      BIS      R4,@CSA        ;LOAD CHANNEL
7725 050472          SETPRI  #PRI00          ;ALLOW INTERRUPTS
      050472 012700 000000          MOV      #PRI00,RO
      050476 104441          TRAP    C$SPRI
7726 050500 052777 000120 133176      BIS      #120,@CSA      ;SET EET + IE IN CSA
7727 050506 042777 000020 133170      BIC      #20,@CSA       ;DO EXTERNAL START (= CLR EET)
7728 050514 005003      CLR      R3            ;CLEAR TIMEOUT COUNTER
7729 050516 005737 004150      40$: TST      INTFLA        ;WAS THERE A DONE INTERRUPT?
7730 050522 001010      BNE     50$            ;BRANCH IF YES
7731 050524 005303      DEC      R3            ;DECREMENT TIMEOUT COUNTER
7732 050526 004737 012146      CALL    INSERT         ;SKIP BRANCH IF 'SFI' IS SET
7733 050532 001371      BNE     40$            ;BRANCH IF NOT ZERO
7734 050534          ERRSOFT 1507,E1507,EERG ;ERROR HANDLER
      050534 104457          TRAP    C$ERSOFT
      050536 002743          .WORD   1507
      050540 052063          .WORD   E1507
      050542 007724          .WORD   EERG
7735 050544          50$: CKLOOP          ;
      050544 104406          TRAP    C$CLP1
7736 050546 017737 133132 004064      MOV      @CSA,BAD       ;GET CSA CONTENTS
7737 050554 012737 100300 004062      MOV      #100300,GOOD  ;SET UP DATA FOR ERROR MESSAGES

```

TEST 15: ANALOGUE INPUT TEST - FIELD AND MANUFACTURING TEST.

```

7738 050562 050437 004062      BIS      R4,GOOD      ;... (CHANNEL)
7739 050566 050537 004062      BIS      R5,GOOD      ;... (GAIN )
7740 050572 023737 004062 004064  CMP      GOOD,BAD     ;IS CSA CONTENTS OK AFTER CONVERSION?
7741 050600 004737 012146      CALL     INSERT       ;SKIP BRANCH IF 'SFI' IS SET
7742 050604 001404      BEQ      60$          ;BRANCH IF DONE IS CLEARED
7743 050606      ERRSOFT 1508,E1508,EER4 ;ERROR HANDLER
      050606 104457      TRAP     C$ERSOFT
      050610 002744      .WORD   1508
      050612 052134      .WORD   E1508
      050614 007474      .WORD   EER4
7744 050616      60$:  CKLOOP
      050616 104406      TRAP     C$CLP1
7745 050620 042777 000100 133056  BIC      #100,@CSA    ;CLEAR INTERRUPT ENABLE BIT
7746 050626 005037 004150      CLR     INTFLA       ;CLEAR DONE INTERRUPT FLAG
7747 050632      SETPRI #PRI07       ;DISABLE INTERRUPTS
      050632 012700 000340      MOV     #PRI07,RO
      050636 104441      TRAP     C$SPRI
7748 050640 005777 133036      TST     @DAT         ;READ DAT TO CLR DONE+ERR IN CSA
7749 050644      ENDSEG
      050644      10006$:
7750 050646 000137 050336      JMP     MUXPAR       ;JUMP TO MUX PART
      TRAP     C$ESEG
7751
7752 050652      EXQV15: EXIT  TST
      050652 104432      TRAP     C$EXIT
      050654 001632      .WORD   L10050-.
7753
7754 050656 000000      ANS151: .WORD 0      ;SAVE LOCATION FOR MUX BOARD ANSWER
7755 050660 000001      ANS152: .WORD 1      ;SAVE LOCATION FOR SOURCE CONNECTION ANSWER
7756 050662 000000      ANS153: .WORD 0      ;LOCATION FOR ODD/EVEN FLAG
7757 050664 000000      BIPOL:  .WORD 0      ;BIPOLAR FLAG
7758 050666 000000      MUXC:   .WORD 0      ;MUX BOARD COUNTER
7759 050670 000000      VOLTE:  .WORD 0      ;SAVE LOCATION FOR VOLVA TABLE END ADDR.
7760 050672 000002      TOLRA:  .WORD 2      ;LOCATION TO STORE THE TOLLERANS VALUE
7761 050674 000000      TOLOF:  .WORD 0      ;STORE TO SAVE TABLE POINTER
7762
7763 050676 000060 001474      VOLVA:  .WORD 48.,828. ;UNIPOLAR LSB VALUE
7764 050702 011610 000000      .WORD 5000.,0        ;UNIPOLAR HALF SCALE VALUE
7765 050706 023413 000144      .WORD 9995.,100.    ;UNIPOLAR FULL SCALE VALUE
7766
7767 050712 154365 177634      .WORD -9995.,-100.  ;BIPOLAR FULL NEG. VALUE
7768 050716 000060 001474      .WORD 48.,828.     ;BIPOLAR ZERO SCALE VALUE
7769 050722 023413 000144      .WORD 9995.,100.   ;BIPOLAR FULL POS. VALUE
7770
7771
7772      ;  TOLLERANCE TABLE
7773
7774      ;  GAIN  1 2 5 10 20 50 100 200
7775
7776 050726 000002 000002 000002 000002  FILTOL: .WORD 2,2,2, 4, 6, 7,12., 20. ;FIELD TABLE
      050734 000004 000006 000007
      050742 000014 000024
7777 050746 000002 000002 000002 000002  MAFTOL: .WORD 2,2,2, 3, 3, 5,10., 15. ;MANUFACTURING TABLE (NO MUX)
      050754 000003 000003 000005
      050762 000012 000017
7778 050766 000002 000002 000002 000002  MUXTOL: .WORD 2,2,2, 3, 5, 6,12., 17. ;MANUFACTURING TABLE (WITH MUX)
      050774 000003 000005 000006

```


TEST 16: ANALOGUE OUTPUT TEST - FIELD AND MANUFACTURING TEST

```

7809 .SBTTL TEST 16: ANALOGUE OUTPUT TEST - FIELD AND MANUFACTURING TEST
7810 :*****
7811 : TEST 16 - ANALOGUE OUTPUT TEST.
7812 :
7813 : THE USER IS TOLD THAT THE TEST WILL ONLY
7814 : RUN CORRECTLY IF THE MODULE HAS BEEN CALIBRATED.
7815 :
7816 : NEXT, THE OPERATOR IS ASKED WHETHER THE MODULE IS VOLTAGE
7817 : CALIBRATED OR CURRENT CALIBRATED. DEPENDING ON THE RESPONSE, THE
7818 : PROGRAM PROMPTS WITH VOLTAGE OR CURRENT VALUES.
7819 : IF CURRENT IS USED A CHECK IS MADE THAT ALL CHANNELS ARE SHOWN
7820 : AS "OPEN LINE" IN THE CSA REGISTER.
7821 : THIS ASSUMES THAT NOTHING IS CONNECTED TO THE CURRENT OUTPUTS.
7822 : THE USER IS THEN ASKED TO CONNECT A DVM WITH THE CORRECT RANGE
7823 : TO THE MODULE OUTPUT AT THE SPECIFIED CHANNEL. IF CURRENT WAS
7824 : SELECTED, THE PROGRAM USES THE OFFSET BIT IN THE MOD REGISTER AND
7825 : CHECKS THAT THE OPL BIT IN THE CSA REGISTER IS CLEAR.
7826 :
7827 : THE OPERATOR THEN HAS TO CHECK SEVERAL VALUES TO ENSURE THAT THE
7828 : READING OF THE DVM IS THE SAME AS THE OUTPUT VALUE. ALL
7829 : VALUES HAVE BEEN CHECKED BY ANSWERING THE QUESTION "IS VALUE ON
7830 : DVM IN RANGE. THE PROCESS IS REPEATED FOR ALL
7831 : CHANNELS.
7832 :
7833 : IF "UAM" FLAG IS SET, THIS TEST WILL NOT BE CARRIED OUT.
7834 :
7835 : N.B. THE DVM READINGS MAY DIFFER FROM THE PRINTED VALUES IF THE
7836 : MODULE HAS BEEN RECALIBRATED BY THE CUSTOMER.
7837 :*****

```

```

7838 052510 BGNTST
7839 052510 004737 011714 CALL SELECT ;CALL SELECT ROUTINE
7840 052514 001010 .WORD 1010 ;GIVE TEST PARAMETER
7841 052516 054530 TSHD16 ;GIVE TEST HEADER ADDRESS
7842 052520 103002 BCC 1$ ;IF CARRY IS SET, EXIT TEST
7843 052522 EXIT TST ;EXIT TEST
7844 052526 104432 TRAP C$EXIT
7845 052530 004010 .WORD L10051-.
7846 052532 104450 1$: MANUAL ;IS MANUAL INTERVENTION ALLOWED?
7847 052534 103416 BCOMPLETE ITRA16 ;BRANCH IF YES (UAM FLAG NOT SET)
7848 052538 104421 RFLAGS R0 ;READ OPERATOR FLAGS INTO R0
7849 052542 032700 001000 BIT #PNT,R0 ;PRINT MESSAGES ?
7850 052546 001410 BEQ 3$ ;IF NO, EXIT
7851 052550 012746 054623 PRINTF #TM16 ;IF YES PRINT TEST DISABLED
7852 052554 012746 000001 MOV #TM16, -(SP)
7853 052558 010600 MOV #1, -(SP)
7854 052562 104417 000004 MOV SP,R0
7855 052566 062706 000004 TRAP C$PNTF
7856 052570 000004 ADD #4,SP
7857 052574 104432 3$: EXIT TST ;EXIT TEST
7858 052578 003750 TRAP C$EXIT
7859 052582 005001 ITRA16: CLR R1 ;CLEAR TEMPORARY STORE
7860 052586 005004 CLR R4 ;GET FIRST CHANNEL NUMBER

```

TEST 16: ANALOGUE OUTPUT TEST - FIELD AND MANUFACTURING TEST

SEQ 0197

```

7853 052572          PRINTF #PME166          ;MARK FOR CALIBRATION WITH
      052572 012746 055666          MOV #PME166, -(SP)
      052576 012746 000001          MOV #1, -(SP)
      052602 010600          MOV SP, R0
      052604 104417          TRAP C$PNTF
      052606 062706 000004          ADD #4, SP
7854 052612          PRINTF #PME167          ;TEST 23
      052612 012746 055757          MOV #PME167, -(SP)
      052616 012746 000001          MOV #1, -(SP)
      052622 010600          MOV SP, R0
      052624 104417          TRAP C$PNTF
      052626 062706 000004          ADD #4, SP
7855 052632          GMANIL MES164,ANS164,-1,NO ;VOLTAGE OR CURRENT
      052632 104443          TRAP C$GMAN
      052634 000404          BR 10000$
      052636 054404          .WORD ANS164
      052640 000120          .WORD T$CODE
      052642 055521          .WORD MES164
      052644 177777          .WORD -1
      052646
7856 052646 005737 054404          TST ANS164          10000$:
7857 052652 001002          BNE 10$           ;IS CURRENT CALIBRATED?
7858 052654 000137 053744          JMP VOLPA        ;SKIP VOLTAGE ROUTINE IF YES
7859 052660          10$: BGNSEG          ;JUMP TO VOLTAGE ROUTINE
      052660 104404
7860 052662 000304          SWAB R4          TRAP C$BSEG
7861 052664 005077 131014          CLR @CSA        ;CHANNEL NUMBER IN HIGH BYTE
7862 052670 050477 131010          BIS R4,@CSA    ;SET UP CSA
7863 052674 032777 000010 130776          BIT #10,@MOD  ;LOAD CHANNEL INTO CSA
7864 052702 001013          BNE 12$        ;IS 4-20 MA RANGE SELECTED
7865 052704 012777 000010 130770          MOV #10,@DAT  ;BRANCH IF YES
7866 052712 013703 004164          MOV REDTIM,R3 ;LOAD DAT FOR 'OPL' CORRECTION
7867 052716 032777 000200 130760 13$: BIT #200,@CSA ;LOAD READY TIMEOUT COUNTER
7868 052724 001002          BNE 12$        ;IS READY BIT SET?
7869 052726 005303          DEC R3         ;BRANCH IF YES
7870 052730 001372          BNE 13$        ;IS TIMEOUT OVER?
7871 052732 004737 011514          CALL WT500     ;BRANCH IF NOT ZERO
7872 052736 017737 130742 004064          MOV @CSA,BAD  ;WAIT 500 US
7873 052744 032737 000040 004064          BIT #40,BAD   ;GET CSA CONTENTS
7874 052752 004737 012146          CALL INSERT    ;IS OLP BIT SET
7875 052756 001004          BNE 20$        ;SKIP BRANCH IF "SFI" IS SET
7876 052760          ERRSOFT 1601,E1601,EER6 ;BRANCH IF YES
      052760 104457          ;ERROR HANDLER
      052762 003101          TRAP C$ERSOFT
      052764 056016          .WORD 1601
      052766 007602          .WORD E1601
7877 052770          20$: ENDSEG          .WORD EER6
      052770 104405          10001$:
7878 052772 000304          SWAB R4          TRAP C$ESEG
7879 052774 005204          INC R4          ;CHANNEL NUMBER IN LOW BYTE
7880 052776 022704 000004          CMP #4,R4      ;CREATE NEXT CHANNEL NUMBER
7881 053002 001326          BNE 10$        ;ALL 4 CHANNELS DONE ?
7882
7883 053004 017737 130670 004064          MOV @MOD,BAD  ;BRANCH IF NO
7884 053012 012737 000003 004134          MOV #3,MODE   ;GET MOD REGISTER CONTENTS
7885 053020 012737 000036 054524          MOV #36,TABOFF ;SET UP INPUT FOR DACON ROUTINE
      ;PIONT TO CURRENT OFFSET TABLE

```

TEST 16: ANALOGUE OUTPUT TEST - FIELD AND MANUFACTURING TEST

```

7886 053026 032737 000010 004064      BIT      @10,BAD      ;IS AN OFFSET SELECETED?
7887 053034 001005                    BNE      30$        ;BRANCH IF YES
7888 053036 005037 054524                    CLR      TABOFF     ;CLEAR TABLE POINTER
7889 053042 012737 000002 004134      MOV      @2,MODE    ;SET UP INPUT FOR DACON ROUTINE
7890 053050 005037 004136      30$:    CLR      GAIN      ;...
7891 053054 005004                    CLR      R4         ;CHANNEL COUNTER
7892 053056 012737 000003 054526      40$:    MOV      @3.,VALCNT ;LOAD VALUE COUNTER
7893 053064                    PRINTF   @PME162,R4  ;PROMT FOR DVM CHAN. CONNECTION
      053064 010446                    MOV      R4,-(SP)
      053066 012746 055055                    MOV      @PME162,-(SP)
      053072 012746 000002                    MOV      @2,-(SP)
      053076 010600                    MOV      SP,R0
      053100 104417                    TRAP    C$PNTF
      053102 062706 000006                    ADD     @6,SP
7894 053106                    PRINTF   @PME165      ;PROMT FOR DVM ADJUSTMENT
      053106 012746 055151                    MOV      @PME165,-(SP)
      053112 012746 000001                    MOV      @1,-(SP)
      053116 010600                    MOV      SP,R0
      053120 104417                    TRAP    C$PNTF
      053122 062706 000004                    ADD     @4,SP
7895 053126                    PRINTF   @PME16A      ;
      053126 012746 055213                    MOV      @PME16A,-(SP)
      053132 012746 000001                    MOV      @1,-(SP)
      053136 010600                    MOV      SP,R0
      053140 104417                    TRAP    C$PNTF
      053142 062706 000004                    ADD     @4,SP
7896 053146                    GMANIL  MES162,ANS161,-1,YES ;...
      053146 104443                    TRAP    C$GMAN
      053150 000404                    BR      10002$
      053152 054402                    .WORD  ANS161
      053154 000130                    .WORD  T$CODE
      053156 055305                    .WORD  MES162
      053160 177777                    .WORD  -1
      053162                    10002$:
7897 053162 000304                    SWAB    R4          ;LOAD CHANNEL IN HIGH BYTE
7898 053164 010477 130514                    MOV     R4,@CSA    ;LOAD CSA WITH CHANNEL NUMBER
7899 053170 013705 054524                    MOV     TABOFF,R5  ;POINT TO THE FIRST VALUE
7900 053174 016501 054430      50$:    MOV     CUROUT(R5),R1 ;GET OUTPUT VALUE FROM TABLE
7901 053200                    BGNSEG
      053200 104404                    TRAP    C$BSEG
7902 053202 032777 000010 130470      BIT     @10,@MOD   ;IS 4-20 MA RANGE SELECTED
7903 053210 001002                    BNE     60$        ;BRANCH IF YES
7904 053212 005701                    TST    R1          ;ARE WE USEING 0 MA
7905 053214 001415                    BEQ    70$        ;SKIP OPL TESTING IF YES
7906 053216 017737 130462 004064      60$:    MOV     @CSA,BAD   ;GET CSA CONTENTS
7907 053224 032737 000040 004064      BIT     @40,BAD   ;IS OPL BIT CLEARED ?
7908 053232 004737 012146                    CALL   INSERT     ;SKIP BRANCH IF "SFI" IS SET
7909 053236 001404                    BEQ    70$        ;BRANCH IF YES
7910 053240                    ERRSOFT 1602,E1602,EER3 ;ERROR HANDLER
      053240 104457                    TRAP    C$ERSOFT
      053242 003102                    .WORD  1602
      053244 056134                    .WORD  E1602
      053246 007442                    .WORD  EER3
7911 053250      70$:    CKLOOP
      053250 104406                    TRAP    C$CLP1
7912 053252 010177 130424                    MOV     R1,@DAT   ;LOAD OUTPUT VALUE INTO DATA REG.
7913 053256 013703 004164                    MOV     REDTIM,R3 ;LOAD READY TIMEOUT COUNTER

```


TEST 16: ANALOGUE OUTPUT TEST - FIELD AND MANUFACTURING TEST

	053522	010600						MOV	SP,R0
	053524	104417						TRAP	C\$PNTF
	053526	062706	000004					ADD	#4,SP
7943									
7944	053532	006237	054402	ASR	ANS161				;DIVIDE BY TWO FOR VOLTAGE PRINTOUT
7945	053536	006003		ROR	R3				;DIVIDE BY TWO FOR VOLTAGE PRINTOUT
7946	053540			PRINTF	#PME164,ANS161,R3				;PRINT VOLT ON 500 OHM RESISTOR
	053540	010346						MOV	R3,-(SP)
	053542	013746	054402					MOV	ANS161,-(SP)
	053546	012746	055603					MOV	#PME164,-(SP)
	053552	012746	000003					MOV	#3,-(SP)
	053556	010600						MOV	SP,R0
	053560	104417						TRAP	C\$PNTF
	053562	062706	000010					ADD	#10,SP
7947	053566	062705	000002	ADD	#2,R5				;POINT TO INTEGER DIFFERENTS VALUE
7948	053572	016501	054430	MOV	CUROUT(R5),R1				;AND GET IT
7949	053576	062705	000002	ADD	#2,R5				;POINT TO DECIMAL DIFFERENTS VALUE
7950	053602	016502	054430	MOV	CUROUT(R5),R2				;AND GET IT
7951	053606			PRINTF	#PLMI				;PRINT +/-
	053606	012746	056457					MOV	#PLMI,-(SP)
	053612	012746	000001					MOV	#1,-(SP)
	053616	010600						MOV	SP,R0
	053620	104417						TRAP	C\$PNTF
	053622	062706	000004					ADD	#4,SP
7952	053626	004737	015764	CALL	DECOUT				;PRINT DIFFERENTS VALUE
7953	053632			PRINTF	#OUT2				;PRINT 'MILLIVOTS'
	053632	012746	056516					MOV	#OUT2,-(SP)
	053636	012746	000001					MOV	#1,-(SP)
	053642	010600						MOV	SP,R0
	053644	104417						TRAP	C\$PNTF
	053646	062706	000004					ADD	#4,SP
7954									
7955	053652			GMANIL	MES165,ANS161,-1,NO				;IS VALUE ON DVM IN RANGE ?
	053652	104443						TRAP	C\$GMAN
	053654	000404						BR	10004\$
	053656	054402						.WORD	ANS161
	053660	000120						.WORD	T\$CODE
	053662	055017						.WORD	MES165
	053664	177777						.WORD	-1
	053666								10004\$:
7956	053666	005737	054402	TST	ANS161				;TEST ANSWER
7957	053672	001004		BNE	110\$;BRANCH IF YES
7958	053674			ERRSOFT	1605,E1608,EERG				;ERROR HANDLER
	053674	104457						TRAP	C\$ERRSOFT
	053676	003105						.WORD	1605
	053700	056371						.WORD	E1608
	053702	007724						.WORD	EERG
7959	053704	062705	000002	110\$:	ADD	#2,R5			;POINT TO NEXT OUTPUT VALUE
7960	053710	005337	054526		DEC	VALCNT			;ALL VALUES USED
7961	053714	001402			BEQ	120\$;BRANCH IF YES
7962	053716	000137	053174		JMP	50\$;IF NO, REPEATED WITH NEXT VALUE
7963	053722	000304		120\$:	SWAB	R4			;CHANNEL NUMBER IN LOW BYTE
7964	053724	005204			INC	R4			;CREATE NEXT CHANNEL NUMBER
7965	053726	022704	000004		CMP	#4,R4			;ALL 4 CHANNELS DONE ?
7966	053732	001002			BNE	130\$;BRANCH IF NO
7967	053734	000137	054376		JMP	EXQV16			;SKIP VOLTAGE OUTPUT CHECK
7968	053740	000137	053056	130\$:	JMP	40\$;

TEST 16: ANALOGUE OUTPUT TEST - FIELD AND MANUFACTURING TEST

```

7969
7970          ;**PART 2 WILL DO THE SAME WITH THE VOLTAGES OUTPUTS (EXCEPT THE OPL TESTING)
7971
7972 053744 012737 000000 004134 VOLPA: MOV    #0,MODE          ;SET UP DACON ROUTINE INPUT
7973 053752 005037 004136          CLR    GAIN          ;...
7974 053756 005004          CLR    R4           ;START WITH CHANNEL 0
7975 053760 012737 000003 054526 10$: MOV    #3.,VALCNT      ;LOAD OUTPUT VALUE COUNTER
7976 053766          PRINTF #PME163,R4      ;PROMT FOR DVM CHAN. CONNECTION
      053766 010446          MOV    R4,-(SP)
      053770 012746 055346          MOV    #PME163,-(SP)
      053774 012746 000002          MOV    #2,-(SP)
      054000 010600          MOV    SP,R0
      054002 104417          TRAP  C$PNTF
      054004 062706 000006          ADD    #6,SP
7977 054010          GMANIL MES163,ANS161,-1,YES ;PROMT FOR DVM ADJUST
      054010 104443          TRAP  C$GMAN
      054012 000404          BR    10005$
      054014 054402          .WORD ANS161
      054016 000130          .WORD T$CODE
      054020 055441          .WORD MES163
      054022 177777          .WORD -1
      054024          10005$:
7978 054024 000304          SWAB   R4           ;LOAD CHANNEL IN HIGH BYTE
7979 054026 010477 127652          MOV    R4,@CSA      ;LOAD CHANNEL IN CSA
7980 054032 005005          CLR    R5           ;POINT TO THE FIRST VALUE
7981 054034 016501 054406 20$: MOV    VOLOUT(R5),R1 ;GET OUTPUT VALUE FROM TABLE
7982 054040          BGNSEG
      054040 104404          TRAP  C$BSEG
7983 054042 010177 127634          MOV    R1,@DAT      ;LOAD OUTPUT DATA REGISTER
7984 054046 013703 004164          MOV    REDTIM,R3    ;LOAD READY TIMEOUT COUNTER
7985 054052 032777 000200 127624 40$: BIT    #200,@CSA    ;IS READY BIT NOW SET?
7986 054060 004737 012146          CALL  INSERT        ;SKIP BRANCH IF "SFI" IS SET
7987 054064 001006          BNE   50$          ;BRANCH IF YES
7988 054066 005303          DEC   R3           ;DECREMENT READY TIMEOUT COUNT.
7989 054070 001370          BNE   40$          ;BRANCH IF COUNTER IS NOT ZERO
7990 054072          ERRSOFT 1606,E1603,EER3
      054072 104457          TRAP  C$ERSOFT
      054074 003106          .WORD 1606
      054076 056244          .WORD E1603
      054100 007442          .WORD EER3
7991 054102          50$: CKLOOP
      054102 104406          TRAP  C$CLP1
7992 054104 017737 127574 004064          MOV    @CSA,BAD     ;GET CSA CONTENTS
7993 054112 012737 100200 004062          MOV    #100200,GOOD ;LOAD GOOD DATA
7994 054120 050437 004062          BIS   R4,GOOD       ;LOAD CHANNEL NUMBER
7995 054124 042737 000040 004064          BIC   #40,BAD        ;MASK OUT OPL BIT
7996 054132 023737 004062 004064          CMP   GOOD,BAD      ;CSA CONTENTS CORRECT AFTER READY SET?
7997 054140 004737 012146          CALL  INSERT        ;SKIP BRANCH IF "SFI" IS SET
7998 054144 001404          BEQ   30$          ;BRANCH IF YES
7999 054146          ERRSOFT 1607,E1604,EER1 ;ERROR HANDLER
      054146 104457          TRAP  C$ERSOFT
      054150 003107          .WORD 1607
      054152 056342          .WORD E1604
      054154 007342          .WORD EER1
8000 054156          30$: ENDSEG
      054156          10006$:
      054156 104405          TRAP  C$ESEG

```

TEST 16: ANALOGUE OUTPUT TEST - FIELD AND MANUFACTURING TEST

```

8001 054160          PRINTF #PME161          ;PRINT OPERATOR ACTION
      054160 012746 054725          MOV      #PME161,-(SP)
      054164 012746 000001          MOV      #1,-(SP)
      054170 010600          MOV      SP,R0
      054172 104417          TRAP     C$PNTF
      054174 062706 000004          ADD      #4,SP
8002 054200          CALL      DACON          ;CALL D/A CONVERSION ROUTINE
8003 054204 004737 015764          CALL      DECOUT        ;PRINT OUT THE RESULT
8004 054210          PRINTF #OUT2          ;PRINT 'MILLIVOLTS'
      054210 012746 056516          MOV      #OUT2,-(SP)
      054214 012746 000001          MOV      #1,-(SP)
      054220 010600          MOV      SP,R0
      054222 104417          TRAP     C$PNTF
      054224 062706 000004          ADD      #4,SP
8005 054230 062705 000002          ADD      #2,R5          ;POINT TO INTEGER DIFFERENTS VALUE
8006 054234 016501 054406          MOV      VOLOUT(R5),R1 ;AND GET IT
8007 054240 062705 000002          ADD      #2,R5          ;POINT TO DECIMAL DIFFERENTS VALUE
8008 054244 016502 054406          MOV      VOLOUT(R5),R2 ;AND GET IT
8009 054250          PRINTF #PLMI          ;PRINT ./-
      054250 012746 056457          MOV      #PLMI,-(SP)
      054254 012746 000001          MOV      #1,-(SP)
      054260 010600          MOV      SP,R0
      054262 104417          TRAP     C$PNTF
      054264 062706 000004          ADD      #4,SP
8010 054270 004737 015764          CALL      DECOUT        ;PRINT THE DIFFERENTS VALUE
8011 054274          PRINTF #OUT2          ;PRINT 'MILLIVOLTS'
      054274 012746 056516          MOV      #OUT2,-(SP)
      054300 012746 000001          MOV      #1,-(SP)
      054304 010600          MOV      SP,R0
      054306 104417          TRAP     C$PNTF
      054310 062706 000004          ADD      #4,SP
8012 054314          GMANIL MES165,ANS161,-1,NO ;IS THE VALUE ON THE DVM IN RANGE
      054314 104443          TRAP     C$GMAN
      054316 000404          BR      10007$
      054320 054402          .WORD  ANS161
      054322 000120          .WORD  T$CODE
      054324 055017          .WORD  MES165
      054326 177777          .WORD  -1
      054330          10007$:
8013 054330 005737 054402          TST      ANS161          ;TEST ANSWER
8014 054334 001004          BNE     60$             ;BRANCH IF YES
8015 054336          ERRSOFT 1608,E1608,EERG ;ERROR HANDLER
      054336 104457          TRAP     C$ERSOFT
      054340 003110          .WORD  1608
      054342 056371          .WORD  E1608
      054344 007724          .WORD  EERG
8016 054346 062705 000002          60$: ADD      #2,R5          ;POINT TO THE NEXT OUTPUT VALUE
8017 054352 005337 054526          DEC      VALCNT        ;ALL OUTPUT VALUES USED
8018 054356 001226          BNE     20$             ;BRANCH IF NO
8019 054360 000304          SWAB    R4             ;CHANNEL NUMBER IN LOW BYTE
8020 054362 005204          INC     R4             ;NEXT CHANNEL NUMBER
8021 054364 022704 000004          CMP     #4,R4          ;ALL 4 CHANNELS DONE
8022 054370 001402          BEQ     EXQV16         ;EXIT IF YES
8023 054372 000137 053760          JMP     10$            ;BRANCH IF NO
8024 054376          EXQV16: EXIT          TST
      054376 104432          TRAP     C$EXIT
      054400 002134          .WORD  L10051

```

TEST 16: ANALOGUE OUTPUT TEST - FIELD AND MANUFACTURING TEST

```

8025
8026 054402 000000          ANS161: .WORD 0          ;TEMPORARY STORE
8027 054404 000000          ANS164: .WORD 0          ;STORE FOR VOLTAGE OR CURRENT ANSWER
8028
8029 054406 000000 000002 000620 VOLOUT: .WORD 0,2,400.    ;OUTPUT VALUE + DIFFERENTS FROM THAT
8030 054414 004000 000002 000620          .WORD 4000,2,400.    ;... (= +/- 1LSB DIFFERENCE)
8031 054422 007777 000002 000620          .WORD 7777,2,400.    ;...
8032
8033 054430 000000 000011 001130 CUROUT: .WORD 0,9.,600.,4,600. ;OUT VALUE+DIFFERENTS+500 OHM DIFF.
      054436 000004 001130          .WORD 4000,9.,600.,4,600. ;... (= +/- 2LSB DIFFERENCE)
8034 054442 004000 000011 001130          .WORD 7777,9.,600.,4,600. ;...
      054450 000004 001130
8035 054454 007777 000011 001130          .WORD 0,8.,0,4,0      ;OFFSET OUT+DIFFER.+500 OHM DIFFER.
      054462 000004 001130          .WORD 4000,8.,0,4,0   ;... (= +/- 2LSB DIFFERENCE)
8036
8037 054466 000000 000010 000000          .WORD 7777,8.,0,4,0   ;...
      054474 000004 000000
8038 054500 004000 000010 000000          .WORD 0,8.,0,4,0
      054506 000004 000000
8039 054512 007777 000010 000000          .WORD 0,8.,0,4,0
      054520 000004 000000
8040
8041 054524 000000          TABOFF: .WORD 0        ;CUROUT TABLE PIONTER
8042 054526 000000          VALCNT: .WORD 0        ;VALUE COUNTER
8043
8044          .NLIST BEX
8045 054530          045      123      062 TSMD16: .ASCIZ /#S2#ANALOGUE OUTPUT TEST - FIELD AND MANUFACTURING TEST#N/
8046 054623          045      101      124 TM16: .ASCIZ /#ATEST DISABLED - NO MANUAL INTERVENTION ALLOWED (UAM FLAG SET)#N/
8047
8048 054725          045      116      045 PME161: .ASCIZ /#N#ACOMPARE THE PRINTED VALUE WITH THE VALUE ON THE DVM#N/
8049 055017          012      015      012 MES165: .ASCIZ <12><15><12><15>/IS VALUE ON DVM IN RANGE /
8050 055055          045      116      045 PME162: .ASCIZ /#N#ACONNECT YOUR DVM TO THE CURRENT OUTPUT AT CHANNEL #02#N/
8051 055151          045      101      124 PME165: .ASCIZ /#ATO MEASURE THE 20000 UA RANGE#N/
8052 055213          045      101      050 PME16A: .ASCIZ /#A(OOR THE 10000 MV RANGE IF YOU USE A 500 OHM RESISTOR)#N/
8053 055305          124      131      120 MES162: .ASCIZ /TYPE CARRIAGE RETURN TO CONTINUE/
8054 055346          045      116      045 PME163: .ASCIZ /#N#ACONNECT YOUR DVM TO THE VOLTAGES OUTPUT AT CHANNEL #02/
8055 055441          101      116      104 MES163: .ASCIZ /AND ADJUST IT TO MEASURE THE 0 - 10000 MV RANGE/
8056 055521          101      122      105 MES164: .ASCIZ /ARE THE ANALOGUE OUTPUTS FOR CURRENT CALIBRATED ?/
8057 055603          045      116      045 PME164: .ASCIZ /#N#A= #D5#A.#Z3#A MILLIVOLTS ON 500 OHM RESISTOR /
8058 055666          045      116      045 PME166: .ASCIZ /#N#ATHIS TEST WILL ONLY RUN CORRECTLY IF THE MODULE IS#N/
8059 055757          045      101      103 PME167: .ASCIZ /#ACALIBRATED (USE TEST 23)#N#N/
8060
8061 056016          117      120      114 E1601: .ASCII /OPL BIT IN CSA NOT SET/<12><15>
8062 056046          115      101      113          .ASCIZ /MAKE SURE THAT NOTHING IS CONNECTED TO CURRENT OUTPUT/
8063 056134          117      120      114 E1602: .ASCII /OPL BIT IN CSA NOT CLEARED/<12><15>
8064 056170          111      123      040          .ASCIZ /IS DVM CONNECTED TO THE ADDRESSED CHANNEL ?/
8065 056244          122      105      101 E1603: .ASCIZ /READY BIT IN CSA NOT SET AFTER LOADING DAT REGISTER (TIMEOUT)/
8066 056342          103      123      101 E1604: .ASCIZ /CSA CONTENTS INCORRECT/
8067 056371          115      117      104 E1608: .ASCIZ /MODULE DEFECT OR INCORRECTLY CALIBRATED - TRY TEST 23/
8068
8069 056457          045      101      040 PLMI: .ASCIZ \#A TOLLERANCE +/-\
8070 056501          045      101      040 OUT1: .ASCIZ /#A MICROAMPS/
8071 056516          045      101      040 OUT2: .ASCIZ /#A MILLIVOLTS/
8072          .LIST BEX
8073          .EVEN
8074 056534          .ENDTST
      056534

```

L10051:

HARDWARE TESTS MACRO M1200 26-OCT-83 15:22 PAGE 81-8

J16

TEST 16: ANALOGUE OUTPUT TEST - FIELD AND MANUFACTURING TEST

SEQ 0204

056534 104401

TRAP C\$ETST

TEST 17: DIGITAL INPUT TEST - USING FIELD TEST CONNECTOR.

```

8076 .SBTTL TEST 17: DIGITAL INPUT TEST - USING FIELD TEST CONNECTOR.
8077 :*****
8078 : TEST 17 - DIGITAL INPUT TEST.
8079 :
8080 :FIRST OF ALL, THE USER IS PROMPTED TO INSERT THE TEST CONNECTOR INTO THE
8081 :MODULE ON WHICH THE LED IS FLASHING. THEN A CHECK IS MADE THAT THE INPUT
8082 :DATA MATCHES THE ODD PATTERNS AVAILABLE FROM THE CONNECTOR. AFTER THIS,
8083 :THE USER IS ASKED TO SWITCH THE CONNECTOR TO THE EVEN POSITION AND THE
8084 :DATA IS CHECKED TO BE CORRECT. THE PROGRAM THEN PROMPTS THE USER TO MOVE
8085 :THE SWITCH "ON" AND "OFF" TWICE MORE. THIS CHECKS THAT THE CSA INTERRUPT
8086 :BIT IR15 CAN BE SET TO CAUSE AN INTERRUPT BY THE LEADING OR TRAILING
8087 :EDGES OF THE MODULE DATA LINE 15, AND THAT IR15 OF THE CSA REMAINS SET
8088 :AFTER THE DATA LINE 15 HAS REVERTED TO ITS ORIGINAL STATE.
8089 :
8090 :IF 'UAM' FLAG IS SET, THIS TEST WILL NOT BE CARRIED OUT.
8091 :
8092 :THIS TEST IS CARRIED OUT IN LOW LEVEL MODE.
8093 :*****
8094 056536          BGNTST
8095 056536 004737 011714          CALL SELECT          ;CALL SELECT ROUTINE
8096 056542 001001          .WORD 1001          ;GIVE TEST PARAMETER
8097 056544 057454          TSHD17          ;GIVE TEST HEADER ADDRESS
8098 056546 103002          BCC 10$          ;IF CARRY IS SET, EXIT TEST
8099 056550          EXIT TST          ;EXIT TEST
8100 056552 104432          TRAP C$EXIT
8100 056554 002314          .WORD L10052-.
8101 056554 104450          10$: MANUAL          ;IS MANUAL INTERVENTION ALLOWED?
8101 056556 103416          BCOMPLETE 20$          ;IF YES, BRANCH (UAM FLAG NOT SET)
8102 056560          RFLAGS RO          ;READ OPERATOR FLAGS INTO RO
8103 056562 104421          BCS 20$          ;PRINT MESSAGES ?
8104 056566 001410          TRAP C$RFLA
8105 056570          BIT #PNT,RO          ;IF NO, EXIT
8105 056574 012746 057544          BEQ 3$          ;IF YES PRINT TEST DISABLED
8105 056600 010600          MOV #TM17,-(SP)
8105 056602 104417          MOV #1,-(SP)
8105 056604 062706 000004          MOV SP,RO
8106 056610          TRAP C$PNTF
8106 056610 104432          ADD #4,SP
8106 056612 002254          TRAP C$EXIT
8107 056614 052777 000010 125056 20$: BIS #10,@MOD          ;SELECT LOW LEVEL RANGE
8108 056622          PRINTF #WME171          ;ASK FOR TEST CONNECTOR
8108 056622 012746 057646          MOV #WME171,-(SP)
8108 056626 012746 000001          MOV #1,-(SP)
8108 056632 010600          MOV SP,RO
8108 056634 104417          TRAP C$PNTF
8108 056636 062706 000004          ADD #4,SP
8109 056642          PRINTF #WME172          ;ASK FOR TEST CONNECTOR
8109 056642 012746 057747          MOV #WME172,-(SP)
8109 056646 012746 000001          MOV #1,-(SP)
8109 056652 010600          MOV SP,RO
8109 056654 104417          TRAP C$PNTF
8109 056656 062706 000004          ADD #4,SP

```

TEST 17: DIGITAL INPUT TEST - USING FIELD TEST CONNECTOR.

8110	056662	004737	016576	CALL	FLASH				;CALL FLASHING LED ROUTINE
8111	056666	005737	017000	TST	FLSANS				;WAS THE ANSWER NO?
8112	056672	001753		BEQ	30\$;ASK AGAIN IF ANSWER WAS NO
8113	056674	005037	004144	CLR	ITRCNT				;CLEAR ITERATION COUNTER
8114	056700			SETVEC	VEC,#INTSR,#PRI07				;SET UP VECTOR AND SERVICE ROUT.
	056700	012746	000340					MOV	#PRI07,-(SP)
	056704	012746	017276					MOV	#INTSR,-(SP)
	056710	013746	003710					MOV	VEC,-(SP)
	056714	012746	000003					MOV	#3,-(SP)
	056720	104437						TRAP	C\$SVEC
	056722	062706	000010					ADD	#10,SP
8115	056726	005001		ITRA17:	CLR	R1			;CLEAR TEMPORARY STORE
8116	056730				BGNSEG				
	056730	104404						TRAP	C\$BSEG
8117	056732	005037	004150	CLR	INTFLA				;CLEAR INTERRUPT FLAG
8118	056736	017737	124740	MOV	@DAT,BAD	004064			;READ INPUT
8119	056744	022737	125252	CMP	#125252,BAD	004064			;IS ODD PATTERN FOUND ?
8120	056752	004737	012146	CALL	INSERT				;SKIP BRANCH IF "SFI" IS SET
8121	056756	001407		BEQ	10\$;IF YES BRANCH
8122	056760	012737	125252	MOV	#125252,GOOD	004062			;SET UP DATA FOR ERROR HANDLER
8123	056766			ERRSOFT	1701,E1701,EER2				;ERROR HANDLER
	056766	104457						TRAP	C\$ERSOFT
	056770	003245						.WORD	1701
	056772	060356						.WORD	E1701
	056774	007400						.WORD	EER2
8124	056776			10\$:	CKLOOP				
	056776	104406						TRAP	C\$CLP1
8125	057000			20\$:	GMANIL	MES171,ANS,-1,NO			;PROMPT FOR EVEN PATTERN
	057000	104443						TRAP	C\$GMAN
	057002	000404						BR	10001\$
	057004	004142						.WORD	ANS
	057006	000120						.WORD	T\$CODE
	057010	060035						.WORD	MES171
	057012	177777						.WORD	-1
	057014								10001\$:
8126	057014	005737	004142	TST	ANS				;CHECK THE ANSWER
8127	057020	001767		BEQ	20\$;IF ANSWER WAS NO, ASK AGAIN
8128	057022	017737	124654	MOV	@DAT,BAD	004064			;READ INPUT
8129	057030	022737	052525	CMP	#52525,BAD	004064			;EVEN PATTERN FOUND
8130	057036	004737	012146	CALL	INSERT				;SKIP BRANCH IF "SFI" IS SET
8131	057042	001407		BEQ	30\$;IF YES, BRANCH
8132	057044	012737	052525	MOV	#52525,GOOD	004062			;SET UP DATA FOR ERROR HANDLER
8133	057052			ERRSOFT	1702,E1702,EER2				;ERROR HANDLER
	057052	104457						TRAP	C\$ERSOFT
	057054	003246						.WORD	1702
	057056	060447						.WORD	E1702
	057060	007400						.WORD	EER2
8134	057062			30\$:	CKLOOP				
	057062	104406						TRAP	C\$CLP1
8135	057064	012777	040000	MOV	#40000,@CSA	124612			;SET IE IN CSA REGISTER
8136	057072	052777	020000	BIS	#20000,@CSA	124604			;SET ELE IN CSA REGISTER
8137	057100			SETPRI	#PRI00				;ENABLE INTERRUPT
	057100	012700	000000					MOV	#PRI00,R0
	057104	104441						TRAP	C\$SPRI
8138	057106			40\$:	GMANIL	MES172,ANS,-1,NO			;ASK ODD POSITION
	057106	104443						TRAP	C\$GMAN
	057110	000404						BR	10002\$

TEST 17: DIGITAL INPUT TEST - USING FIELD TEST CONNECTOR.

```

8167 057322 004737 012146          CALL    INSERT          ;SKIP BRANCH IF "SFI" IS SET
8168 057326 001407                   BEQ     901              ;BRANCH IF YES
8169 057330 012737 150000 004062    MOV     @150000,GOOD    ;SET UP DATA FOR ERROR HANDLER
8170 057336                   ERRSOFT 1706,E1704,EER2 ;ERROR HANDLER
                                TRAP    C#ERSOFT
                                .WORD   1706
                                .WORD   E1704
                                .WORD   EER2
8171 057346                   901:   CKLOOP
8172 057350                   GMANIL MES172,ANS,-1,NO ;ASK ODD POSITION
                                TRAP    C#CLP1
                                .WORD   100051
                                BR       100051
                                .WORD   ANS
                                .WORD   T#CODE
                                .WORD   MES172
                                .WORD   -1
8173 057364 005737 004142          TST     ANS              ;CHECK THE ANSWER
8174 057370 001766                   BEQ     901              ;IF ANSWER WAS NO, ASK AGAIN
8175 057372 017737 124306 004064    MOV     @CSA,BAD        ;GET CSRA CONTENTS
8176 057400 022737 150000 004064    CMP     @150000,BAD     ;IS IR15,EI AND ETE15 STILL SET
8177 057406 004737 012146          CALL    INSERT          ;SKIP BRANCH IF "SFI" IS SET
8178 057412 001407                   BEQ     1001             ;BRANCH IF YES
8179 057414 012737 150000 004062    MOV     @150000,GOOD    ;SET UP DATA FOR ERROR HANDLER
8180 057422                   ERRSOFT 1707,E1707,EER2 ;ERROR HANDLER
                                TRAP    C#ERSOFT
                                .WORD   1707
                                .WORD   E1707
                                .WORD   EER2
8181 057432                   1001: ENDSEG
                                TRAP    C#ESEG
                                .WORD   100031
                                .WORD   100031
8182 057434                   SETPRI @PRI07          ;DISABLE INTERRUPTS
                                MOV     @PRI07,RO
                                TRAP    C#SPRI
8183 057442                   CLRVEC VEC             ;RESTORE INTERRUPT VECTOR
                                MOV     VEC,RO
                                TRAP    C#CVEC
8184 057450                   EXQV17: EXIT          TST
                                TRAP    C#EXIT
                                .WORD   L10052-.
8185 057452 001414                   .NLIST BEX
8186 057454 045 123 062 TSHD17:;.ASCIZ /#S2#ADIGITAL INPUT TEST - USING FIELD TEST CONNECTOR.#N/
8187 057544 045 101 124 TM17: .ASCIZ /#ATEST DISABLED - NO MANUAL INTERVENTION ALLOWED (UAM FLAG SET)#N/
8188
8189 057646 045 116 045 WME171: .ASCIZ /#N#AIS THE DIGITAL TEST CONNECTOR SWITCHED TO THE ODD POSITION#N/
8190 057747 045 101 101 WME172: .ASCIZ /#AAND PLUGGED INTO THE BOARD WITH THE FLASHING LED ? /
8191 060035 115 117 126 MES171: .ASCII /MOVE THE SWITCH TO THE EVEN POSITION/<12><15>
8192 060103 124 131 120 .ASCIZ /TYPE Y AND THEN CR IF DONE/
8193 060136 116 117 127 MES172: .ASCII /NOW MOVE THE SWITCH BACK TO THE ODD POSITION/<12><15>
8194 060214 124 131 120 .ASCIZ /TYPE Y AND THEN CR IF DONE/
8195 060247 115 117 126 MES173: .ASCII /MOVE THE SWITCH TO THE EVEN POSITION AGAIN/<12><15>
8196 060323 124 131 120 .ASCIZ /TYPE Y AND THEN CR IF DONE/
8197 060356 104 101 124 E1701: .ASCIZ /DATA REGISTER OF DIGITAL INPUT NOT LOADED WITH EVEN DATA/
8198 060447 104 101 124 E1702: .ASCIZ /DATA REGISTER OF DIGITAL INPUT NOT LOADED WITH ODD DATA/
8199 060537 116 117 040 E1703: .ASCIZ /NO INTERRUPT ON LEADING EDGE OF DATA LINE 15/

```

TEST 17: DIGITAL INPUT TEST - USING FIELD TEST CONNECTOR.

8200	060614	103	123	101	E1704:	.ASCII	/CSA REGISTER OF DIGITAL INPUT MODULE INCORRECT/<12><15>
8201	060674	101	106	124		.ASCIZ	/AFTER INTERRUPT/
8202	060714	116	117	040	E1705:	.ASCIZ	/NO INTERRUPT ON TRAILING EDGE OF DATA LINE 15/
8203	060772	103	123	101	E1707:	.ASCIZ	/CSA DATA BIT IR15 IS NOT CORRECT - CLEARED BY DATA LINE 15/
8204						.LIST	BEX
8205						.EVEN	
8206	061066					ENDTST	
	061066						
	061066	104401					

L10052: TRAP C\$ETST

TEST 18: DIGITAL OUTPUT TEST - USING FIELD TEST CONNECTOR.

```

8208 .SBTTL TEST 18: DIGITAL OUTPUT TEST - USING FIELD TEST CONNECTOR.
8209 ;*****
8210 ; TEST 18 - DIGITAL OUTPUT TEST.
8211 ;
8212 ;THE USER IS PROMPTED TO INSERT THE DIGITAL TEST CONNECTOR INTO THE MODULE
8213 ;ON WHICH THE LED IS FLASHING. THEN HE IS PROMPTED TO CHECK THAT A SLIDING
8214 ;ONES PATTERN APPEARS ON THE TEST CONNECTOR LEDS. AFTER 15 SECONDS THE PROGRAM
8215 ;PROMPTS THE USER TO LOOK FOR A SLIDING ZEROS PATTERN, WHICH IS OUTPUT FOR
8216 ;15 SECONDS BEFORE TERMINATING THE TEST. THIS CHECKS THAT ALL OUTPUT LINES
8217 ;CAN BE UNIQUELY SET AND CLEARED.
8218 ;
8219 ;IF THE 'UAM' FLAG IS SET, THIS TEST WILL NOT BE CARRIED OUT.
8220 ;*****
8221 061070          BGNTST
      061070
8222 061070 004737 011714          CALL SELECT          T18::
8223 061074 001002                .WORD 1002          ;CALL SELECT ROUTINE
8224 061076 061424                TSHD18          ;GIVE TEST PARAMETER (FIELD/DO)
8225 061100 103546                BCS EXQV18      ;GIVE TEST HEADER
8226 061102                MANUAL          ;IF CARRY IS SET, EXIT TEST
      061102 104450                ;IS MANUAL INTERVENTION ALLOWED?
8227 061104                BCOMPLETE 10$          TRAP C$MANI
      061104 103416                ;IF YES BRANCH (UAM FLAG NOT SET)
8228 061106                RFLAGS RO          BCS 10$
      061106 104421                ;READ OPERATOR FLAGS INTO RO          TRAP C$RFLA
8229 061110 032700 001000          BIT @PNT,RO          ;PRINT MESSAGES ?
8230 061114 001410                BEQ 3$          ;IF NO, EXIT
8231 061116                PRINTF @TM18          ;IF YES PRINT TEST DISABLED
      061116 012746 061515          MOV @TM18,-(SP)
      061122 012746 000001          MOV @1,-(SP)
      061126 010600                MOV SP,RO
      061130 104417                TRAP C$PNTF
      061132 062706 000004          ADD @4,SP
8232 061136                3$: EXIT TST          ;EXIT TEST
      061136 104432                TRAP C$EXIT
      061140 000762                .WORD L10053-.
8233 061142 005737 004130          10$: TST LOTFLA          ;IS LOOP ON TEST FLAG SET ?
8234 061146 001027                BNE 30$          ;IF NO, BRANCH
8235 061150 005337 004130          DEC LOTFLA          ;IF YES, DECREMENT IT
8236 061154                20$: PRINTF @WME181          ;ASK FOR TEST CONNECTOR
      061154 012746 061617          MOV @WME181,-(SP)
      061160 012746 000001          MOV @1,-(SP)
      061164 010600                MOV SP,RO
      061166 104417                TRAP C$PNTF
      061170 062706 000004          ADD @4,SP
8237 061174                PRINTF @WME184          ;...
      061174 012746 061712          MOV @WME184,-(SP)
      061200 012746 000001          MOV @1,-(SP)
      061204 010600                MOV SP,RO
      061206 104417                TRAP C$PNTF
      061210 062706 000004          ADD @4,SP
8238 061214 004737 016576          CALL FLASH          ;CALL FLASHING LED ROUTINE
8239 061220 005737 017000          TST FLSANS          ;IF THE ANSWER IS NO
8240 061224 001753                BEQ 20$          ;BRANCH BACK AND ASK AGAIN
8241 061226 005037 004144          30$: CLR ITRCNT          ;CLEAR I TERRATION COUNTER
8242 061232 005037 061422          ITRA18: CLR SLZERO          ;CLEAR FLAG
8243 061236                PRINTB @WME182          ;PRINT SLIDING ONE MESSAGES

```


HARDWARE TESTS MACRO M1200 26 OCT 83 15:22 PAGE 83.2

F1

TEST 18: DIGITAL OUTPUT TEST - USING FIELD TEST CONNECTOR.

SEQ 0212

8297

HARDWARE TESTS MACRO M1200 26-OCT-83 15:22 PAGE 85

TEST 18: DIGITAL OUTPUT TEST - USING FIELD TEST CONNECTOR.

8305 062124
8306

ENDMOD

G1

SEQ 0213

TEST 18: DIGITAL OUTPUT TEST - USING FIELD TEST CONNECTOR:

```

8310          .TITLE HARDWARE TESTS
8355
8356 062124          BGNMOD
8357          .SBTTL TEST 19: DIGITAL INPUT TEST - SPECIFICALLY SELECTABLE
8358          :*****
8359          : TEST 19 - DIGITAL INPUT TEST - SPECIFICALLY SELECTABLE
8360          :
8361          : THIS TEST CONTINUOUSLY READS THE INPUT LINES AND PRINTS THE DATA
8362          : TO THE CONSOLE.
8363          :
8364          : THE USER IS ASKED WHICH INPUT RANGE (LOW LEVEL OR NORMAL) AND IF
8365          : NORMAL IS SELECTED, WHICH DEBOUNCE PERIOD TO USE. THEN THE INPUT
8366          : LINES OF THE MODULE ARE CONTINUOUSLY READ AND, IF THE SUPERVISOR
8367          : FLAG "INHIBIT ERROR REPORT" IS NOT SET, THE BIT PATTERN IS
8368          : OUTPUT TO THE CONSOLE. BY USING DIFFERENT INPUT VOLTAGES, THIS
8369          : TEST CAN BE USED TO CHECK THE SWITCHING LEVELS. THIS TEST IS A
8370          : VISUAL TEST. AN ERROR REPORT IS ONLY MADE IF THE LLS AND RT BITS
8371          : IN THE MOD REGISTER CAN'T CLEAR OR SET. THE INTERRUPT FUNCTION
8372          : OF THE MODULE IS NOT TESTED. TO ABORT THE TEST, THE USER MUST
8373          : TYPE "CNTL C".
8374          :
8375          :*****
8376 062124          BGNTST 19.
8377 062124          CALL          SELECT          T19::          ;CALL SELECT ROUTINE
8378 062130          004737 011714          .WORD          10001          ;GIVE TEST PARAMETER (SPEC/DI)
8379 062132          062450          TSHD19          ;GIVE TEST HEADER
8380 062134          103002          BCC          ITRA19          ;IF CARRY IS SET, EXIT TEST
8381 062136          104432          EXIT          TST          ;EXIT TEST
8382 062140          001112          TRAP          C$EXIT
8383 062142          104443          .WORD          L10054-.
8384 062144          000404          TRAP          C$GMAN
8385 062146          004142          BR          10000$
8386 062150          000120          .WORD          ANS
8387 062152          063020          .WORD          T$CODE
8388 062154          177777          .WORD          MMES1
8389 062156          005737 004142          .WORD          -1
8390 062162          001067          TST          ANS          10000$:          ;IS LOW LEVEL SELECTED?
8391 062164          042777 000010 121506          BNE          30$          ;IF YES, DON'T ASK FOR DEBOUNCE
8392 062172          017737 121502 004064          BIC          #10,@MOD          ;CLEAR LOW LEVEL BIT IN MOD REG.
8393 062200          032737 000010 004064          MOV          @MOD,BAD          ;GET MOD REGISTER CONTENTS
8394 062206          004737 012146          BIT          #10,BAD          ;IS LOW LEVEL BIT IN MOD CLEARED?
8395 062212          001404          CALL          INSERT          ;SKIP BRANCH IF 'SFI' IS SET
8396 062214          104457          BEQ          10$          ;BRANCH IF YES
8397 062216          003555          ERRSOFT 1901,E1901,EERG          ;ERROR HANDLER
8398 062220          062535          TRAP          C$ERSOFT
8399 062222          007724          .WORD          1901
8400 062224          104406          .WORD          E1901
8401 062226          104443          .WORD          EERG
8402 062230          000406          10$:          CKLOOP          ;
8403 062232          062446          GMANID MMES2,ANS1,0,-1,1,3,NO          ;ASK FOR DEBOUNCE PERIOD
8404          TRAP          C$CLP1
8405          BR          10001$
8406          .WORD          ANS1

```


TEST 19: DIGITAL INPUT TEST - SPECIFICALLY SELECTABLE

```

062234 000022 .WORD T$CODE
062236 063042 .WORD MMES2
062240 177777 .WORD -1
062242 000001 .WORD T$LLOLIM
062244 000003 .WORD T$HILIM
062246
10001$:
8393 062246 053777 062446 121424 BIS ANS1,@MOD ;SET RESPONSE TIME IN MOD REG.
8394 062254 032737 000001 062446 BIT #1,ANS1 ;IS ONLY RTO USED ?
8395 062262 001003 BNE 20$ ;BRANCH IF NO
8396 062264 042777 000002 121406 BIC #2,@MOD ;IF YES, CLEAR RT1 BIT
8397 062272 117737 121402 004064 20$: MOVB @MOD,BAD ;GET MOD REGISTER CONTENTS
8398 062300 123737 004064 062446 CMPB BAD,ANS1 ;DEBOUNCE CORRECT LOADED
8399 062306 004737 012146 CALL INSERT ;SKIP BRANCH IF 'SFI' IS SET
8400 062312 001435 BEQ 40$ ;BRANCH IF YES
8401 062314 013737 062446 004062 MOV ANS1,GOOD ;SET UP DATA FOR ERROR HANDLER
8402 062322 052737 002100 004062 BIS #2100,GOOD ;...
8403 062330 ERRSOFT 1902,E1902,EER1 ;ERROR HANDLER
062330 104457 TRAP C$ERSOFT
062332 003556 .WORD 1902
062334 062641 .WORD E1902
062336 007342 .WORD EER1
8404 062340 000422 BR 40$ ;SKIP THE NEXT COMMAND LINE
8405 062342 052777 000010 121330 30$: BIS #10,@MOD ;SET LOW LEVEL
8406 062350 004737 011506 CALL WT25M ;WAIT 25 MS
8407 062354 017737 121320 004064 MOV @MOD,BAD ;GET MOD REGISTER CONTENTS
8408 062362 032737 000010 004064 BIT #10,BAD ;IS LOW LEVEL SELECTED
8409 062370 004737 012146 CALL INSERT ;SKIP BRANCH IF 'SFI' IS SET
8410 062374 001004 BNE 40$ ;BRANCH IF YES
8411 062376 ERRSOFT 1903,E1903,EERG ;ERROR HANDLER
062376 104457 TRAP C$ERSOFT
062400 003557 .WORD 1903
062402 062720 .WORD E1903
062404 007724 .WORD EERG
8412 062406 40$: CKLOOP ;
062406 104406 TRAP C$CLP1
8413 062410 017702 121266 MOV @DAT,R2 ;GET INPUT VALUE
8414 062414 PRINTB #FORMAT,R2,R2 ;PRINT INPUT VALUE BIN. * OCTAL
062414 010246 MOV R2,-(SP)
062416 010246 MOV R2,-(SP)
062420 012746 063171 MOV #FORMAT,-(SP)
062424 012746 000003 MOV #3,-(SP)
062430 010600 MOV SP,R0
062432 104414 TRAP C$PNTB
062434 062706 000010 ADD #10,SP
8415 062440 000762 BR 40$ ;GO INTO THE PRINTOUT LOOP
8416 062442 EXQV19: EXIT TST ;EXIT TEST
062442 104432 TRAP C$EXIT
062444 000606 .WORD L10054..
8417
8418 062446 000002 ANS1: .WORD 2
8419
8420 .NLIST BEX
8421 062450 045 123 062 TSHD19: .ASCIZ /#S2#ADIGITAL INPUT TEST - SPECIFICALLY SELECTABLE.#N/
8422 062535 114 117 127 E1901: .ASCIZ /LOW LEVEL BIT IN MOD REGISTER OF DIGITAL INPUT COULD NOT BE CLEARED
8423 062641 104 105 102 E1902: .ASCIZ /DEBOUNCE BITS IN MOD REGISTER COULD NOT BE SET/
8424 062720 114 117 127 E1903: .ASCIZ /LOW LEVEL BIT IN MOD REGISTER OF DIGITAL INPUT COULD NOT BE SET
8425 063020 114 117 127 MMES1: .ASCIZ /LOW LEVEL INPUT ?/

```

TEST 19: DIGITAL INPUT TEST - SPECIFICALLY SELECTABLE

8426	063042	127	110	111	MMES2:	.ASCII	/WHICH DEBOUNCE PERIOD DO YOU WANT TO USE ? /<12><15>
8427	063117	105	116	124		.ASCIZ	/ENTER 1 FOR 500US, 2 FOR 5MS, 3 FOR 10MS./
8428	063171	045	116	045	FORMAT:	.ASCIZ	/NAREAD INPUT VALUE = #B16#A (OCTAL = #06#A)/
8429						.LIST	BEX
8430						.EVEN	
8431	063252					ENDTST	
	063252						
	063252	104401					

L10054: TRAP C\$ETST

TEST 20: DIGITAL OUTPUT TEST - SPECIFICALLY SELECTABLE

```

8433 .SBTTL TEST 20: DIGITAL OUTPUT TEST - SPECIFICALLY SELECTABLE
8434 :*****
8435 : TEST 20 - DIGITAL OUTPUT TEST - SPECIFICALLY SELECTABLE.
8436 :
8437 : THE USER IS FIRST ASKED TO SELECT A PATTERN TYPE - SLIDING ONES,
8438 : SLIDING ZEROS, OR A PATTERN PAIR. IN THE LAST CASE, HE IS ALSO
8439 : ASKED TO ENTER THE DATA PAIR. THE PROGRAM THEN REQUESTS A TIME
8440 : INTERVAL IN MILLISECONDS (DEFAULT IS ZERO) TO WAIT BETWEEN EACH
8441 : PATTERN CHANGE. THE SELECTED PATTERN IS THEN OUTPUT TO THE USER
8442 : LINES OF THE MODULE.
8443 :
8444 : AFTER LOADING THE OUTPUT, THE DATA REGISTER IS READ BACK AND AN
8445 : ERROR REPORTED IF THE LOADED DATA AND THE READ DATA ARE NOT THE
8446 : SAME. THIS IS THE ONLY ERROR REPORT IN THE TEST. ALL OTHER
8447 : CHECKING IS VISUAL.
8448 :
8449 : TO ABORT THE TEST, "CNTL C" MUST BE TYPED.
8450 :*****
8451 063254 BGNTST
      063254
8452 063254 004737 011714 CALL SELECT ;CALL SELECT ROUTINE
8453 063260 010002 .WORD 10002 ;GIVE TEST PARAMETER (SPEC/DI)
8454 063262 063746 TSHD20 ;GIVE TEST HEADER
8455 063264 103002 BCC ITRA20 ;IF CARRY IS SET EXIT TEST
8456 063266 EXIT TST ;EXIT TEST
      063266 104432
      063270 001206
8457 063272 ITRA20: GMANID MMES71,ANS2,0,-1,1,3,YES ;GET MANUAL PARAMETER
      063272 104443
      063274 000406
      063276 063736
      063300 000032
      063302 064104
      063304 177777
      063306 000001
      063310 000003
      063312
      063312 022737 000003 063736 CMP #3,ANS2 ;IS IT A PATTERN PAIR?
8459 063320 001410 BEQ 10$ ;BRANCH IF YES
8460 063322 022737 000001 063736 CMP #1,ANS2 ;IS IT SLIDING ONES?
8461 063330 001445 BEQ 20$ ;IF YES, BRANCH
8462 063332 022737 000002 063736 CMP #2,ANS2 ;IS IT SLIDING ZEROS?
8463 063340 001444 BEQ 30$ ;BRANCH IF YES
8464 063342 10$: GMANID MMES72,FIRPA,0,-1,0,177777,YES ;GET FIRST PATTERN
      063342 104443
      063344 000406
      063346 063740
      063350 000032
      063352 064214
      063354 177777
      063356 000000
      063360 177777
      063362
8465 063362 GMANID MMES73,SECPA,0,1,0,177777,YES ;GET SECOND PATTERN
      063362 104443
      063364 000406
      063366 063742

```

TEST 20: DIGITAL OUTPUT TEST - SPECIFICALLY SELECTABLE

063370	000032								.WORD	T\$CODE
063372	064234								.WORD	MME573
063374	177777								.WORD	-1
063376	000000								.WORD	T\$LOLIM
063400	177777								.WORD	T\$HILIM
063402										
8466	063402			GMANID	MME574,TIMIN,D,-1,1,5000,YES			10002\$:		
	063402	104443								
	063404	000406							TRAP	C\$GMAN
	063406	063744							BR	10003\$
	063410	000052							.WORD	TIMIN
	063412	064254							.WORD	T\$CODE
	063414	177777							.WORD	MME574
	063416	000001							.WORD	-1
	063420	005000							.WORD	T\$LOLIM
	063422								.WORD	T\$HILIM
8467	063422			PRINTB	#WME52			10003\$:		
	063422	012746	064421							
	063426	012746	000001							
	063432	010600							MOV	#WME52,-(SP)
	063434	104414							MOV	#1,-(SP)
	063436	062706	000004						MOV	SP,RO
8468	063442	000464							TRAP	C\$PNTB
8469	063444	012704	000001	20\$:	BR	80\$			ADD	#4,SP
8470	063450	000402			MOV	#1,R4				
8471	063452	012704	177776	30\$:	BR	40\$				
8472					MOV	#177776,R4				
8473										
8474										
8475	063456			40\$:	GMANID	MME574,TIMIN,D,-1,1,5000,YES				
	063456	104443								
	063460	000406								
	063462	063744							TRAP	C\$GMAN
	063464	000052							BR	10004\$
	063466	064254							.WORD	TIMIN
	063470	177777							.WORD	T\$CODE
	063472	000001							.WORD	MME574
	063474	005000							.WORD	-1
	063476								.WORD	T\$LOLIM
	063476								.WORD	T\$HILIM
8476	063476			PRINTB	#WME51			10004\$:		
	063476	012746	064342							
	063502	012746	000001							
	063506	010600							MOV	#WME51,-(SP)
	063510	104414							MOV	#1,-(SP)
	063512	062706	000004						MOV	SP,RO
8477	063516	013703	063744	50\$:	TRAP	C\$PNTB			ADD	#4,SP
8478	063522				MOV	TIMIN,R3				
	063522	104404			BGNSEG					
8479	063524	010477	120152						TRAP	C\$BSEG
8480	063530	005303		60\$:	MOV	R4,@DAT				
8481	063532	004737	011514		DEC	R3				
8482	063536	004737	011514		JSR	PC,WT500				
8483	063542	005703			JSR	PC,WT500				
8484	063544	001371			TST	R3				
8485	063546	017737	120130		BNE	60\$				
8486	063554	023704	004064		MOV	@DAT,BAD				
8487	063560	004737	012146		CMP	BAD,R4				
					CALL	INSERT				

;SKIP BRANCH IF 'SFI' IS SET

TEST 20: DIGITAL OUTPUT TEST - SPECIFICALLY SELECTABLE

```

8488 063564 001406          BEQ      70$          ;BRANCH IF EQUAL
8489 063566 010437 004062  MOV      R4,GOOD     ;SET UP DATA FOR ERRMES.
8490 063572          ERRSOF 2001,E2001,EER1 ;ERROR HANDLER
      063572 104457          ;
      063574 003721          TRAP    C$ERSOFT
      063576 064034          .WORD  2001
      063600 007342          .WORD  E2001
      063600 007342          .WORD  EER1
8491          70$:  ENDSEG          10005$:
      063602          ROL      R4          ;NEXT DATA
      063602 104405          ADC      R4          ;ADD CARRY IF SET
8492 063604 006104          BREAK          ;OPERATER INTERR. ENB.
8493 063606 005504          BR      50$          TRAP    C$BRK
8494 063610          ;SEND ROUTINE FOR PATTERN PAIR
      063610 104422          ;FIRST PATTERN TO OUTPUT
8495 063612 000741          MOV      FIRPA,@DAT  ;THIS IS FOR ERR HANDLER
8496          MOV      FIRPA,R4  ;OPERATOR INTRR. ENB.
8497          BREAK          TRAP    C$BRK
8498          CLR      R2          ;FLAG FOR PATTERN CHOICE
8499 063614 013777 063740 120060 80$:  MOV      FIRPA,@DAT  ;LOAD TIME INTERVAL COUNTER
8500 063622 013704 063740          MOV      FIRPA,R4  ;DEC TIME INTERVAL CON.
8501 063626          BREAK          ;WAIT 500 US
      063626 104422          ;WAIT 500 US
8502 063630 005002          CLR      R2          ;IS THE TIME OVER?
8503 063632 013703 063744 90$:  MOV      TIMIN,R3  ;IF YES, SEND SECOND PAT.
8504 063636 005303 100$:  DEC      R3          ;READ THE DATA REG. BACK
8505 063640 004737 011514  JSR      PC,WT500  ;CMP READ + LOADED DATA
8506 063644 004737 011514  JSR      PC,WT500  ;SKIP BRANCH IF 'SFI' IS SET
8507 063650 005703          TST      R3          ;BRANCH IF EQUAL
8508 063652 001371          BNE     100$        ;SET UP DATA FOR ERRMES.
8509 063654 017737 120022 004064  MOV      @DAT,BAD  ;ERROR HANDLER
8510 063662 023704 004064          CMP      BAD,R4
8511 063666 004737 012146          CALL    INSERT
8512 063672 001406          BEQ     110$
8513 063674 010437 004062  MOV      R4,GOOD
8514 063700          ERRSOF 2002,E2001,EER1 ;
      063700 104457          TRAP    C$ERSOFT
      063702 003722          .WORD  2002
      063704 064034          .WORD  E2001
      063706 007342          .WORD  EER1
8515 063710 005702          110$:  TST      R2          ;IS R2 CLEARED?
8516 063712 001340          BNE     80$
8517 063714 005202          INC     R2          ;BRANCH IF R2 = 1
8518 063716 013777 063742 117756  MOV      SECPA,@DAT ;INCREMENT R2
8519 063724 013704 063742          MOV      SECPA,R4  ;SECOND PATTERN TO OUTP.
8520 063730 000740          BR      90$
8521 063732          EXQV20: EXIT    TST          ;THIS IS FOR ERR HANDLER
      063732 104432          ;GO INTO OUTPUT LOOP
      063734 000542          TRAP    C$EXIT
      063736 000001          .WORD  L10055-.
8522 063740 000000          ANS2:  .WORD  1
8523 063742 177777          FIRPA:  .WORD  0          ;STORE FOR FIRST PATTERN
8524 063744 001000          SECPA:  .WORD  177777    ;STORE FOR SECOND PATTERN
8525          TIMIN:  .WORD  1000 ;STORE FOR TIME INTERVAL
8526          .NLIST  BEX
8527 063746          045    123    062  TSHD20: .ASCIZ /#S2#ADIGITAL OUTPUT TEST - SPECIFICALLY SELECTABLE.#N/
8528 064034          104    101    124  E2001: .ASCIZ /DATA REGISTER INCORRECT AFTER READ BACK/
8529 064104          105    116    124  MMES71: .ASCII /ENTER PATTERN: TYPE/<12><15>
8530 064130          061    040    106  .ASCIZ /1 FOR SLIDING ONES, 2 FOR ZEROS, 3 FOR PATTERN PAIR/

```

TEST 20: DIGITAL OUTPUT TEST - SPECIFICALLY SELECTABLE

8531	064214	106	111	122	MMES72: .ASCIZ	/FIRST PATTERN /
8532	064234	123	105	103	MMES73: .ASCIZ	/SECOND PATTERN /
8533	064254	105	116	124	MMES74: .ASCIZ	/ENTER TIME INTERVAL (MS) BETWEEN EACH PATTERN CHANGE /
8534	064342	045	101	127	WMES1: .ASCIZ	/AWORKING, OUTPUT DATA ARE SLIDING PATTERN/N/
8535	064421	045	101	127	WMES2: .ASCIZ	/AWORKING, OUTPUT DATA ARE PATTERN PAIRS/N/
8536					.LIST	BEX
8537					.EVEN	
8538	064476				ENDTST	
	064476					
	064476	104401				

L10055: TRAP C\$ETST

TEST 21: ANALOGUE INPUT (DYNAMIC ROUTINE) - SPECIFICALLY SELEC

```

8540 .SBTTL TEST 21: ANALOGUE INPUT (DYNAMIC ROUTINE) - SPECIFICALLY SELECTABLE
8541 :*****
8542 : TEST 21 - ANALOGUE INPUT (DYNAMIC ROUTINE) - SPECIFICALLY
8543 : SELECTABLE.
8544 :
8545 : THE USER IS ASKED WHICH CHANNELS ARE TO BE READ, WHICH GAIN AND
8546 : WHICH TRIGGER (INTERNAL OR EXTERNAL) ARE TO BE USED. AFTER THIS
8547 : HE IS TOLD WHICH RANGE (UNI OR BIPOLAR) THE MODULE IS SWITCHED
8548 : TO.
8549 :
8550 : CONVERSIONS ARE THEN MADE TWICE ON EACH SELECTED CHANNEL. THE
8551 : VALUES READ ARE PRINTED OUT AS A DUMP AFTER ALL CONVERSIONS. THE
8552 : DUMPED VALUES ARE OCTAL.
8553 :
8554 : PRINTOUT CAN BE DISABLED USING THE SUPERVISOR FLAG "INHIBIT
8555 : ERROR REPORTS".
8556 :
8557 : TO ABORT THE TEST, "CNL C" MUST BE TYPED.
8558 :*****
8559 064500 BGNTST
      064500
8560 064500 004737 011714 CALL SELECT ;CALL SELECT ROUTINE
      064504 010004 .WORD 10004 ;GIVE TEST PARAMETER (SPEC/DI)
8561 064506 067340 TSHD21 ;GIVE TEST HEADER
8562 064510 103002 BCC ITRC21 ;IF CARRY IS SET, EXIT TEST
8563 064512 EXIT TST ;EXIT TEST
      064512 104432 TRAP C$EXIT
      064514 004166 .WORD L10056-
8564 064516 ITRC21: GMANID MES211,ANS211,D,77400,0,127,,YES ;GET 1ST CHANNEL TO BE TESTED
      064516 104443 TRAP C$GMAN
      064520 000406 BR 10000$
      064522 066330 .WORD ANS211
      064524 000052 .WORD T$CODE
      064526 067434 .WORD MES211
      064530 077400 .WORD 77400
      064532 000000 .WORD T$LOLIM
      064534 000177 .WORD T$HILIM
      064536
      10000$:
8565 064536 GMANID MES212,ANS212,D,77400,0,127,,YES ;GET LAST CHANNEL TO BE TESTED
      064536 104443 TRAP C$GMAN
      064540 000406 BR 10001$
      064542 066332 .WORD ANS212
      064544 000052 .WORD T$CODE
      064546 067452 .WORD MES212
      064550 077400 .WORD 77400
      064552 000000 .WORD T$LOLIM
      064554 000177 .WORD T$HILIM
      064556
      10001$:
8566 064556 023737 066332 066330 CMP ANS212,ANS211 ;IS LAST CHA. GREATER THAN 1ST
8567 064564 002011 BGE 10$ ;BRANCH IF YES
8568 064566 PRINTF @PME211 ;PRINT OPERATOR MISTAKE
      064566 012746 067571 MOV @PME211, -(SP)
      064572 012746 000001 MOV #1, -(SP)
      064576 010600 MOV SP, R0
      064600 104417 TRAP C$PNTF
      064602 062706 000004 ADD #4, SP
8570 064606 000743 BR ITRC21 ;AND BRANCH BACK

```

TEST 21: ANALOGUE INPUT (DYNAMIC ROUTINE) - SPECIFICALLY SELEC

```

8571 064610          10$:  GMANID MES213,ANS213,0,16,0,7,YES      ;ASK FOR GAIN
      064610 104443
      064612 000406
      064614 066334
      064616 000032
      064620 067467
      064622 000016
      064624 000000
      064626 000007
      064630
      10002$:
      8572 064630 032777 000020 117042      BIT      @20,@MOD      ;IS BIPOLAR MODE SELECTED ?
      8573 064636 001011
      8574 064640          BNE      20$
      064640 012746 067747          PRINTF   @PME213      ;PRINT UNIPOLAR MESSAGES
      064644 012746 000001
      064650 010600
      064652 104417
      064654 062706 000004
      8575 064660 000410          BR       30$
      8576 064662          PRINTF   @PME214      ;AND SKIP BIPOLAR MESSAGES
      064662 012746 070022          ;PRINT BIPOLAR MESSAGES
      064666 012746 000001
      064672 010600
      064674 104417
      064676 062706 000004
      8577 064702          30$:  GMANIL MES214,ANS214,-1,YES      ;ASK FOR TRIGGER
      064702 104443
      064704 000404
      064706 066336
      064710 000130
      064712 067550
      064714 177777
      064716
      10003$:
      8578 064716 042777 000016 116760      BIC      @16,@CSA      ;CLEAR GAIN BITS IN CSA
      8579 064724 053777 066334 116752      BIS      ANS213,@CSA   ;LOAD GAIN INTO CSA
      8580 064732 005001
      8581 064734 013704 066330          INTR:  CLR      R1      ;POINT TABLE POINT TO FIRST ADDRESS
      8582 064740          10$:  BGNSEG
      064740 104404
      8583 064742 005777 116734          ;GET FIRST CHANNEL
      8584 064746 042777 177400 116730      TST      @DAT          ;READ DAT TO CLEAR DONE-ERR IN CSA
      8585 064754 050477 116724          ;CLEAR HIGH BYTE OF CSA
      8586 064760 017737 116720 004064      BIS      R4,@CSA      ;LOAD CHANNEL INTO CSA
      8587 064766 032737 100000 004064      MOV      @CSA,BAD     ;GET CSA CONTENTS
      8588 064774 004737 012146          ;IS THE LOADED CHANNEL AVAILABLE?
      8589 065000 001014          CALL   INSERT         ;SKIP BRANCH IF 'SFI' IS SET
      8590 065002 012737 100000 004062      BNE      20$
      8591 065010 050437 004062          ;BRANCH IF YES
      8592 065014 053737 066334 004062      MOV      @100000,GOOD ;SET UP DATA FOR ERROR MESSAGES
      8593 065022          BIS      R4,GOOD      ;SET CHANNEL NUMBER
      065022 104457          BIS      ANS213,GOOD  ;SET GAIN
      065024 004065          ERRSOFT 2101,E2101,EER4 ;ERROR HANDLER
      065026 070074
      065030 007474
      8594 065032          20$:  CKLOOP
      065032 104406
      8595 065034 032777 000200 116642      BIT      @200,@CSA    ;IS INTERFACE READY FOR CONVERSION?

```


TEST 21: ANALOGUE INPUT (DYNAMIC ROUTINE) - SPECIFICALLY SELEC

```

8596 065042 004737 012146          CALL    INSERT          ;SKIP BRANCH IF 'SFI' IS SET
8597 065046 001404          BEQ     30$             ;BRANCH IF DONE IS CLEARED
8598 065050          ERRSOFT 2102,E2102,EERG ;ERROR HANDLER
      065050 104457          TRAP   C$ERSOFT
      065052 004066          .WORD 2102
      065054 070164          .WORD E2102
      065056 007724          .WORD EERG
8599 065060          30$:   CKLOOP          TRAP   C$CLP1
      065060 104406          ;IS EXTERNAL TRIGGER USED?
8600 065062 005737 066336          TST    ANS214          ;BRANCH IF YES TO EXTERNAL PART
8601 065066 001175          BNE    EXTR21          ;LOAD DONE TIMEOUT COUNTER
8602 065070 005003          CLR    R3              ;START THE FIRST CONVERSION
8603 065072 052777 000001 116604          BIS    @1,@CSA         ;CONVERSION DONE (DONE SET )
8604 065100 032777 000200 116576 40$:   BIT    @200,@CSA       ;BRANCH IF YES
8605 065106 001010          BNE    50$             ;DECREMENT READY WAIT COUNTER
8606 065110 005303          DEC    R3              ;SKIP BRANCH IF 'SFI' IS SET
8607 065112 004737 012146          CALL    INSERT          ;BRANCH IF TIMEOUT COUNTER IS NOT ZERO
8608 065116 001370          BNE    40$             ;ERROR HANDLER
8609 065120          ERRSOFT 2103,E2103,EERG
      065120 104457          TRAP   C$ERSOFT
      065122 004067          .WORD 2103
      065124 070255          .WORD E2103
      065126 007724          .WORD EERG
8610 065130          50$:   ENDSEG          TRAP   C$ESEG
      065130 104405          10004$:
8611 065132 017761 116544 066340          MOV    @DAT,DMPTAB(R1) ;READ DATA REGISTER AND SAVE IT
8612 065140 062701 000002          ADD    @2,R1           ;INCREMENT DUMP TABLE ADDRESS
8613 065144 032777 000200 116532          BIT    @200,@CSA       ;DONE CLEARED AFTER READ DAT REG.?
8614 065152 004737 012146          CALL    INSERT          ;SKIP BRANCH IF 'SFI' IS SET
8615 065156 001404          BEQ    60$             ;BRANCH IF YES
8616 065160          ERRSOFT 2104,E2104,EERG ;ERROR HANDLER
      065160 104457          TRAP   C$ERSOFT
      065162 004070          .WORD 2104
      065164 070346          .WORD E2104
      065166 007724          .WORD EERG
8617 065170 005003          60$:   CLR    R3              ;LOAD DONE TIMEOUT COUNTER
8618 065172 052777 000001 116504          BIS    @1,@CSA         ;START SECOND CONVERSION
8619 065200 032777 000200 116476 70$:   BIT    @200,@CSA       ;CONVERSION DONE (DONE SET )
8620 065206 001010          BNE    80$             ;BRANCH IF YES
8621 065210 005303          DEC    R3              ;DECREMENT DONE TIMEOUT COUNTER
8622 065212 004737 012146          CALL    INSERT          ;SKIP BRANCH IF 'SFI' IS SET
8623 065216 001370          BNE    70$             ;BRANCH IF TIMEOUT COUNTER IS NOT ZERO
8624 065220          ERRSOFT 2105,E2103,EERG ;ERROR HANDLER
      065220 104457          TRAP   C$ERSOFT
      065222 004071          .WORD 2105
      065224 070255          .WORD E2103
      065226 007724          .WORD EERG
8625 065230 017761 116446 066340 80$:   MOV    @DAT,DMPTAB(R1) ;READ DATA REGISTER AND SAVE IT
8626 065236 062701 000002          ADD    @2,R1           ;INCREMENT DUMP TABLE ADDRESS
8627 065242 032777 000200 116434          BIT    @200,@CSA       ;DONE CLEARED AFTER READ DAT REG.?
8628 065250 004737 012146          CALL    INSERT          ;SKIP BRANCH IF 'SFI' IS SET
8629 065254 001404          BEQ    90$             ;BRANCH IF YES
8630 065256          ERRSOFT 2106,E2104,EERG ;ERROR HANDLER
      065256 104457          TRAP   C$ERSOFT
      065260 004072          .WORD 2106
      065262 070346          .WORD E2104

```

TEST 21: ANALOGUE INPUT (DYNAMIC ROUTINE) - SPECIFICALLY SELEC

```

      065264 007724
8631 065266 020437 066332      90$:  CMP      R4,ANS212      ;HAVE WE READ THE LAST CHANNEL ?
      065272 002017      BGE      DMPR      ;IF YES, BRANCH TO DUMP ROUTINE
8633 065274 062704 000400      ADD      @400,R4      ;INCREMENT CHANNEL
8634 065300 022701 001000      CMP      @512.,R1      ;IS THE TOP ADDRESS OF TABLE REACHED?
8635 065304 004737 012146      CALL     INSERT      ;SKIP BRANCH IF 'SFI' IS SET
8636 065310 001006      BNE      100$      ;BRANCH IF NO
8637 065312      ERRSOF 2107,E2107,EERG ;ERROR HANDLER
      065312 104457
      065314 004073
      065316 070433
      065320 007724
      065322 000137 064516      100$:  JMP      ITRC21      ;
      065326 000137 064740      JMP      10$      ;AND DO CONVERSION WITH NEXT CHA.
8640
8641      ;THE NEXT CODE WILL DUMP THE CONVERSION ON THE CONSOLE TERMINAL
8642
8643 065332      DMPR:  RFLAGS R0      ;READ SUPERVISOR FLAGS
      065332 104421
      065334 032700 010000      BIT      @IBE,R0      TRAP      C$RFLA
      065340 001402      BEQ      10$      ;IS IBE FLAG SET?
      065342 000137 064732      JMP      INTR      ;SKIP JUMP INSTRUCTION IF NO
      065346 162701 000002      10$:  SUB      @2,R1      ;DISABLE DUMP PRINT OUT
      065352 010102      MOV      R1,R2      ;RESTORE LAST R1 INCREMENT
      065354 005001      CLR      R1      ;SAVE DUMP TABLE END ADDRESS
      065356 013704 066330      MOV      ANS211,R4      ;POINT TO FIRST VALUE
      065362 000304      SWAB     R4      ;SET UP R4 WITH FIRST CHANNEL NO.
      065364 005003      CLR      R3      ;SWAB HIGH BYTE INTO LOW BYTE
8653 065366      20$:  PRINTF @PME212,DMPTAB(R1),R4 ;CLEAR CHANNEL MARKER
      065366 010446
      065370 016146 066340      MOV      R4,-(SP)      ;DUMP THE CONVERSION TABLE
      065374 012746 067665      MOV      DMPTAB(R1),-(SP)
      065400 012746 000003      MOV      @PME212,-(SP)
      065404 010600
      065406 104417
      065410 062706 000010      MOV      SP,R0
      065414 005203      TRAP     C$PNTF
      065416 022703 000002      ADD      @10,SP
      065422 001002      INC      R3      ;INCREMENT CHANNEL MARKER
      065424 005204      CMP      @2,R3      ;IS CHANNEL NUMBER PRINTED TWICE ?
      065426 005003      BNE      30$      ;BRANCH IF NO
      065430 062701 000002      INC      R4      ;IF YES, INCREMENT CHANNEL NUMBER
      065434 020201      CLR      R3      ;CLEAR CHANNEL MARKER
      065436 002353      30$:  ADD      @2,R1      ;INCREMENT TABLE ADDRESS
      065440 004737 011542      CMP      R2,R1      ;ALL VALUES PRINTED ?
      065444 005737 066336      BGE      20$      ;BRANCH IF NO
      065450 001002      CALL     CRLF      ;PRINT A CRLF BETWEEN EVERY DUMP
      065452 000137 064732      TST      ANS214      ;IS EXTERNAL TRIGGER USED?
      065456 000137 065552      BNE      40$      ;BRANCH IF YES
8667
8668      40$:  JMP      INTR      ;REPEAT INTERNAL TEST
8669
8670      JMP      EXR      ;REPEAT EXTERNAL TEST
      065462 012746 000340      ;THE FOLLOWING CODE IS USED IF EXTERNAL TRIGGER WAS REQUESTED
      065466 012746 017276      EXTR21: SETVEC VEC,@INTSR,@PRI07 ;SET VECTOR 1 AND SERVICE ROUTINE
      065472 013746 003710
      065476 012746 000003      MOV      @PRI07,-(SP)
      MOV      @INTSR,-(SP)
      MOV      VEC,-(SP)
      MOV      @3,-(SP)

```

TEST 21: ANALOGUE INPUT (DYNAMIC ROUTINE) - SPECIFICALLY SELEC

	065502	104437									TRAP	C\$SVEC
	065504	062706	000010								ADD	#10,SP
8671	065510	013702	003710			MOV	VEC,R2					;GET VECTOR 1
8672	065514	062702	000004			ADD	#4,R2					;CREATE ERR VECTOR (VECTOR 2)
8673	065520					SETVEC	R2,#INTSR2,#PRI07					;SET VECTOR 2 AND SERVICE ROUTINE
	065520	012746	000340								MOV	#PRI07,-(SP)
	065524	012746	017306								MOV	#INTSR2,-(SP)
	065530	010246									MOV	R2,-(SP)
	065532	012746	000003								MOV	#3,-(SP)
	065536	104437									TRAP	C\$SVEC
	065540	062706	000010								ADD	#10,SP
8674	065544					SETPRI	#PRI07					;DISABLE INTERRUPTS
	065544	012700	000340								MOV	#PRI07,RO
	065550	104441									TRAP	C\$SPRI
8675	065552	017737	116124	004064	EXR:	MOV	@DAT,BAD					;READ DAT TO CLEAR DONE IN CSA
8676	065560	005037	004152			CLR	INTFL2					;CLEAR ERR INTERRUPT FLAG
8677	065564	005001				CLR	R1					;POINT TABLE POINT TO FIRST ADDRESS
8678	065566	013704	066330			MOV	ANS211,R4					;GET FIRST CHANNEL
8679	065572	042777	177536	116104	10%:	BIC	#177536,@CSA					;CLEAR R/W BITS IN CSA
8680	065600	050477	116100			BIS	R4,@CSA					;LOAD CHANNEL INTO CSA
8681	065604	053777	066334	116072		BIS	ANS213,@CSA					;LOAD GAIN INTO CSA
8682	065612	017737	116066	004064		MOV	@CSA,BAD					;GET CSA CONTENTS
8683	065620	012737	100000	004062		MOV	#100000,GOOD					;SET UP DATA FOR ERROR MESSAGES
8684	065626	050437	004062			BIS	R4,GOOD					;SET CHANNEL NUMBER
8685	065632	053737	066334	004062		BIS	ANS213,GOOD					;SET GAIN
8686	065640	023737	004062	004064		CMP	GOOD,BAD					;IS CSA CONTENTS CORRECT?
8687	065646	004737	012146			CALL	INSERT					;SKIP BRANCH IF 'SFI' IS SET
8688	065652	001404				BEQ	20%					;BRANCH IF YES
8689	065654					ERRSOFT	2108,E2108,EER4					;ERROR HANDLER
	065654	104457									TRAP	C\$ERSOFT
	065656	004074									.WORD	2108
	065660	070521									.WORD	E2108
	065662	007474									.WORD	EER4
8690	065664				20%:	SETPRI	#PRI00					;ALLOW INTERRUPTS
	065664	012700	000000								MOV	#PRI00,RO
	065670	104441									TRAP	C\$SPRI
8691	065672	052777	000120	116004		BIS	#120,@CSA					;SET EET + EI BITS IN CSA
8692	065700	005037	004150		30%:	CLR	INTFLA					;CLEAR INTERRUPT FLAG
8693	065704	005003				CLR	R3					;CLEAR TIMEOUT COUNTER
8694	065706	005737	004150		40%:	TST	INTFLA					;WAS THERE A DONE INTERRUPT?
8695	065712	001010				BNE	50%					;BRANCH IF YES
8696	065714	005303				DEC	R3					;DECREMENT TIMEOUT COUNTER
8697	065716	004737	012146			CALL	INSERT					;SKIP BRANCH IF 'SFI' IS SET
8698	065722	001371				BNE	40%					;BRANCH IF NOT ZERO
8699	065724					ERRSOFT	2109,E2109,EERG					;ERROR HANDLER
	065724	104457									TRAP	C\$ERSOFT
	065726	004075									.WORD	2109
	065730	070603									.WORD	E2109
	065732	007724									.WORD	EERG
8700	065734	005037	004150		50%:	CLR	INTFLA					;CLEAR INTERRUPT FLAG
8701	065740	017737	115740	004064		MOV	@CSA,BAD					;GET CSA CONTENTS
8702	065746	012737	100320	004062		MOV	#100320,GOOD					;SET UP DATA FOR ERROR MESSAGES
8703	065754	050437	004062			BIS	R4,GOOD					;SET CHANNEL NUMBER
8704	065760	053737	066334	004062		BIS	ANS213,GOOD					;SET GAIN
8705	065766	023737	004062	004064		CMP	GOOD,BAD					;IS CSA CONTENTS OK AFTER CONVERSION?
8706	065774	004737	012146			CALL	INSERT					;SKIP BRANCH IF 'SFI' IS SET
8707	066000	001404				BEQ	70%					;BRANCH IF DONE IS CLEARED

TEST 21: ANALOGUE INPUT (DYNAMIC ROUTINE) - SPECIFICALLY SELEC

```

8708 066002          ERRSOFT 2110,E2108,EER4      ;ERROR HANDLER
      066002 104457
      066004 004076
      066006 070521
      066010 007474
      8709 066012 017761 115664 066340 70$:  MOV    @DAT,DMPTAB(R1)      ;SAVE DATA REGISTER CONTENTS
      8710 066020 062701 000002          ADD    #2,R1              ;INCREMENT DUMP TABLE ADDRESS
      8711 066024 017737 115654 004064      MOV    @CSA,BAD           ;GET CSA CONTENTS
      8712 066032 012737 100120 004062      MOV    #100120,GOOD      ;SET UP DATA FOR ERROR MESSAGES
      8713 066040 050437 004062          BIS    R4,GOOD           ;SET CHANNEL NUMBER
      8714 066044 053737 066334 004062      BIS    ANS213,GOOD       ;SET GAIN
      8715 066052 023737 004062 004064      CMP    GOOD,BAD          ;IS INTERFACE READY FOR CONVERSION?
      8716 066060 004737 012146          CALL  INSERT             ;SKIP BRANCH IF 'SFI' IS SET
      8717 066064 001404          BEQ   90$                ;BRANCH IF DONE IS CLEARED
      8718 066066          ERRSOFT 2112,E2108,EER4      ;ERROR HANDLER
      066066 104457
      066070 004100
      066072 070521
      066074 007474
      8719 066076 005737 004152          90$:  TST    INTFL2         ;WAS THERE AN ERROR INTERRUPT?
      8720 066102 004737 012146          CALL  INSERT             ;SKIP BRANCH IF 'SFI' IS SET
      8721 066106 001404          BEQ   100$              ;BRANCH IF NO
      8722 066110          ERRSOFT 2113,E2110,EERG      ;ERROR HANDLER
      066110 104457
      066112 004101
      066114 070650
      066116 007724
      8723 066120 005037 004150          100$: CLR    INTFLA          ;CLEAR DONE INTERRUPT FLAG
      8724 066124 005037 004152          CLR    INTFL2          ;CLEAR ERROR INTERRUPT FLAG
      8725 066130 005003          CLR    R3              ;SET UP WAIT COUNTER
      8726 066132 005737 004150          170$: TST    INTFLA          ;WAS THERE A DONE INTERRUPT?
      8727 066136 001010          BNE   180$             ;BRANCH IF YES
      8728 066140 005303          DEC    R3              ;DECREMENT TIMEOUT COUNTER
      8729 066142 004737 012146          CALL  INSERT             ;SKIP BRANCH IF 'SFI' IS SET
      8730 066146 001371          BNE   170$             ;BRANCH IF NOT ZERO
      8731 066150          ERRSOFT 2114,E2109,EERG      ;ERROR HANDLER
      066150 104457
      066152 004102
      066154 070603
      066156 007724
      8732 066160 017761 115516 066340 180$:  MOV    @DAT,DMPTAB(R1)      ;SAVE DATA REGISTER CONTENTS
      8733 066166 062701 000002          ADD    #2,R1              ;INCREMENT DUMP TABLE ADDRESS
      8734 066172 022701 001000          CMP    #512.,R1         ;IS TOP ADDRESS OF TABLE REACHED?
      8735 066176 001004          BNE   120$             ;BRANCH IF NO
      8736 066200          ERRSOFT 2115,E2107,EERG      ;ERROR HANDLER
      066200 104457
      066202 004103
      066204 070433
      066206 007724
      8737 066210 020437 066332          120$: CMP    R4,ANS212      ;HAVE WE REACHED THE LAST CHANNEL?
      8738 066214 002036          BGE   130$             ;IF YES, BRANCH TO DUMP ROUTINE
      8739 066216 062704 000400          ADD    #400,R4          ;INCREMENT CHANNEL NUMBER
      8740 066222 042777 177400 115454      BIC    #177400,@CSA     ;CLEAR HIGH BYTE OF CSA
      8741 066230 050477 115450          BIS    R4,@CSA          ;LOAD CHANNEL INTO CSA
      8742 066234 017737 115444 004064      MOV    @CSA,BAD         ;GET CSA CONTENTS
      8743 066242 012737 100120 004062      MOV    #100120,GOOD      ;SET UP DATA FOR ERROR MESSAGES
      8744 066250 050437 004062          BIS    R4,GOOD           ;SET CHANNEL NUMBER

```

TEST 21: ANALOGUE INPUT (DYNAMIC ROUTINE) - SPECIFICALLY SELEC

```

8745 066254 053737 066334 004062      BIS      ANS213,GOOD      ;SET GAIN
8746 066262 023737 004062 004064      CMP      GOOD,BAD       ;IS CSA CONTENTS CORRECT?
8747 066270 004737 012146      CALL     INSERT         ;SKIP BRANCH IF 'SFI' IS SET
8748 066274 001404      BEQ      110$           ;BRANCH IF YES
8749 066276      ERRSOFT 2116,E2108,EER4   ;ERROR HANDLER
      066276 104457      TRAP     C$ERSOFT
      066300 004104      .WORD   2116
      066302 070521      .WORD   E2108
      066304 007474      .WORD   EER4
8750 066306 000137 065700      110$:   JMP      30$           ;DO CONVERSION WITH NEXT CHANNEL
8751 066312      130$:   SETPRI  #PRI07     ;DISABLE INTERRUPTS
      066312 012700 000340      MOV     #PRI07,R0
      066316 104441      TRAP    C$SPRI
8752 066320 000137 065332      JMP     DMPR           ;JUMP TO DUMP ROUTINE
8753
8754 066324      EXQV21: EXIT  TST       ;EXIT TEST
      066324 104432      TRAP     C$EXIT
      066326 002354      .WORD   L10056-.
8755
8756 066330 000000      ANS211: .WORD  0       ;SAVE LOCATION FOR FIRST CHANNEL
8757 066332 000000      ANS212: .WORD  0       ;SAVE LOCATION FOR LAST CHANNEL
8758 066334 000000      ANS213: .WORD  0       ;SAVE LOCATION FOR THE GAIN
8759 066336 000000      ANS214: .WORD  0       ;SAVE LOCATION FOR TRIGGER
8760
8761 066340      DMPTAB: .BLKW  256.    ;DUMP TABLE AREA
8762
8763      .NLIST  BEX
8764 067340      045     123     062  TSHD21: .ASCIZ  /%S2%AANALOGUE INPUT (DYNAMIC ) - SPECIFICALLY SELECTABLE.%N/
8765 067434      106     111     122  MES211: .ASCIZ  /FIRST CHANNEL/
8766 067452      114     101     123  MES212: .ASCIZ  /LAST CHANNEL/
8767 067467      107     101     111  MES213: .ASCIZ  /GAIN (TYPE 0 - 7 = GAIN 1,2,5,10,20,50,100,200)/
8768 067550      105     130     124  MES214: .ASCIZ  /EXTERNAL TRIGGER/
8769
8770 067571      045     101     114  PME211: .ASCIZ  /%ALAST CHANNEL HAS TO BE GREATER OR EQUAL TO THE 1ST ONE.%N/
8771 067665      045     116     045  PME212: .ASCIZ  /%N%ACONVERSION VALUE (OCTAL):%06%A, CHANNEL:%D3/
8772 067747      045     116     045  PME213: .ASCIZ  /%N%AMODULE IS SWITCHED TO UNIPOLAR RANGE%N/
8773 070022      045     116     045  PME214: .ASCIZ  /%N%AMODULE IS SWITCHED TO BIPOLAR RANGE%N/
8774
8775 070074      103     123     101  E2101: .ASCIZ  /CSA CONTENTS INCORRECT - SELECTED CHANNEL NOT AVAILABLE/
8776 070164      115     117     104  E2102: .ASCIZ  /MODULE NOT READY FOR CONVERSION - DONE BIT IN CSA IS SET/
8777 070255      104     117     116  E2103: .ASCIZ  /DONE BIT IN CSA NOT SET (TIMEOUT) AFTER CONVERSION START/
8778 070346      104     117     116  E2104: .ASCIZ  /DONE BIT IN CSA NOT CLEARED AFTER READ DATA REGISTER/
8779 070433      104     125     115  E2107: .ASCIZ  /DUMP BUFFER FULL - MORE THEN 127 CHANNELS ADDRESSED ?/
8780 070521      103     123     101  E2108: .ASCIZ  /CSA CONTENTS INCORRECT - AFTER READ DATA REGISTER/
8781 070603      124     111     115  E2109: .ASCIZ  /TIMEOUT - NO DONE INTERRUPT OCCURRED/
8782 070650      105     122     122  E2110: .ASCIZ  /ERROR INTERRUPT OCCURRED/
8783      .LIST  BEX
8784      .EVEN
8785 070702      ENDTST
      070702
      070702 104401      L10056: TRAP  C$ETST

```

TEST 22: ANALOGUE OUTPUT (DYNAMIC TEST) - SPECIFICALLY SELECTA

```

8787 .SBTTL TEST 22: ANALOGUE OUTPUT (DYNAMIC TEST) - SPECIFICALLY SELECTABLE.
8788 :*****
8789 : TEST 22 - ANALOGUE OUTPUT (DYNAMIC TEST) - SPECIFICALLY
8790 : SELECTABLE.
8791 :
8792 : THIS TEST GENERATES EITHER A PERMANENT LOADED STATIC VALUE, OR AN ENDLESS
8793 : SYMMETRICAL VOLTAGE RAMP.
8794 :
8795 : THE USER IS ASKED TO ENTER THE NUMBER OF THE FIRST AND LAST
8796 : CHANNEL WHICH HE WANTS TO USE. HE THEN SELECTS THE OUTPUT PATTERN TYPE
8797 : - SINGLE FIXED VALUE OR RAMP. FOR THE FIRST OF THESE, THE USER
8798 : TYPES THE DESIRED OUTPUT BINARY VALUE. FOR THE RAMP PATTERN, AN
8799 : ENDLESS SYMMETRICAL VOLTAGE RAMP IS GENERATED. IN THIS CASE, THE USER IS
8800 : ASKED TO ENTER A VALUE 'DELTA U'. THIS VALUE IS THE NUMBER OF BITS BY
8801 : WHICH THE PROGRAM CHANGES THE OUTPUT FOR EACH STEP OF THE RAMP, AND MUST
8802 : BE IN THE RANGE 1 TO 4096. THE USER MUST ALSO SELECT A TIME INTERVAL
8803 : 'DELTA T' IN INCREMENT STEPS, WHICH WILL DETERMINE THE PERIOD BETWEEN EACH
8804 : VALUE CHANGE. IF ZERO IS TYPED, THE VALUE WILL BE VARIED AS FAST
8805 : AS THE HARDWARE ALLOWS IT.
8806 :
8807 : THE CONTROL STATUS REGISTER STATUS PRINTOUT CAN BE INHIBITED BY
8808 : SETTING THE 'IBE' FLAG.
8809 :*****
8810 070704          BGNTST
      070704
8811 070704 004737 011714          CALL SELECT          T22::          ;CALL SELECT ROUTINE
8812 070710 010010          .WORD 10010          ;GIVE TEST PARAMETER (SPEC/DI)
8813 070712 071566          TSHD22          ;GIVE TEST HEADER
8814 070714 103002          BCC ITRC22          ;IF CARRY IS CLEARED, EXIT TEST
8815 070716          EXIT TST          ;EXIT TEST
      070716 104432          TRAP C$EXIT
      070720 001610          .WORD L10057-
8816 070722          ITRC22: GMANID MES221,ANS221,D,1400,0,7,YES ;GET 1ST CHANNEL TO BE TESTED
      070722 104443          TRAP C$GMAN
      070724 000406          BR 10000$
      070726 071552          .WORD ANS221
      070730 000052          .WORD T$CODE
      070732 071662          .WORD MES221
      070734 001400          .WORD 1400
      070736 000000          .WORD T$L0LIM
      070740 000007          .WORD T$HILIM
      070742
8817 070742          10000$:
      070742 104443          10$: GMANID MES222,ANS222,D,1400,0,7,YES ;GET LAST CHANNEL TO BE TESTED
      070744 000406          TRAP C$GMAN
      070746 071554          BR 10001$
      070750 000052          .WORD ANS222
      070752 071700          .WORD T$CODE
      070754 001400          .WORD MES222
      070756 000000          .WORD 1400
      070760 000007          .WORD T$L0LIM
      070762          .WORD T$HILIM
8818 070762 023737 071554 071552          CMP ANS222,ANS221          10001$:
8819 070770 002011          BGE 20$          ;LAST CHA. GREATER THAN 1ST?
8820 070772          PRINTF #PME221          ;BRANCH IF YES
      070772 012746 072162          ;PRINT OPERATOR MISTAKE
      070776 012746 000001          MOV #PME221,-(SP)
      MOV #1,-(SP)

```

TEST 22: ANALOGUE OUTPUT (DYNAMIC TEST) - SPECIFICALLY SELECTA

```

071002 010600
071004 104417
071006 062706 000004
8821 071012 000743
8822 071014 104443
071016 000406
071020 071556
071022 000032
071024 071715
071026 177777
071030 000001
071032 000002
071034
8823 071034 022737 000001 071556
8824 071042 001431
8825 071044 104443
071046 000406
071050 071560
071052 000052
071054 071767
071056 177777
071060 000001
071062 010000
071064
8826 071064 104443
071066 000406
071070 071562
071072 000052
071074 072040
071076 177777
071100 000000
071102 177777
071104
8827 071104 012746 072332
071110 012746 000001
071114 010600
071116 104417
071120 062706 000004
8828 071124 000470
8829
8830
8831
8832 071126 104443
071130 000406
071132 071564
071134 000032
071136 072121
071140 177777
071142 000000
071144 007777
071146
8833 071146 005002

```

```

          BR      ITRC22
20$:      GMANID  MES223,ANS223,0,-1,1,2,YES
          ;AND BRANCH BACK
          ;GET PATTERN TYPE
          TRAP    C$GMAN
          BR      10002$
          .WORD   ANS223
          .WORD   T$CODE
          .WORD   MES223
          .WORD   -1
          .WORD   T$LOLIM
          .WORD   T$HILIM
          10002$:
8823      CMP     #1,ANS223
8824      BEQ     40$
8825      GMANID  MES224,ANS224,D,-1,1,4096.,YES
          ;IS STATIC VALUE USED ?
          ;BRANCH IF YES
          ;GET DELTA U
          TRAP    C$GMAN
          BR      10003$
          .WORD   ANS224
          .WORD   T$CODE
          .WORD   MES224
          .WORD   -1
          .WORD   T$LOLIM
          .WORD   T$HILIM
          10003$:
8826      GMANID  MES225,ANS225,D,-1,0,177777,YES
          ;GET DELTA T
          TRAP    C$GMAN
          BR      10004$
          .WORD   ANS225
          .WORD   T$CODE
          .WORD   MES225
          .WORD   -1
          .WORD   T$LOLIM
          .WORD   T$HILIM
          10004$:
8827      PRINTF  #PME223
          ;PRINT HOW TO CONTINUE
          MOV     #PME223,-(SP)
          MOV     #1,-(SP)
          MOV     SP,RO
          TRAP    C$PNTF
          ADD     #4,SP
          ;SKIP NEXT GMANID
          BR      110$
          ;THE NEXT CODE FOR FIXED VALUE OUTPUT
          40$:   GMANID  MES226,ANS226,0,-1,0,7777,YES
          ;GET STATIC VALUE
          TRAP    C$GMAN
          BR      10005$
          .WORD   ANS226
          .WORD   T$CODE
          .WORD   MES226
          .WORD   -1
          .WORD   T$LOLIM
          .WORD   T$HILIM
          10005$:
8833      CLR     R2
          ;CLEAR INHIBIT CSA PRINT.

```

TEST 22: ANALOGUE OUTPUT (DYNAMIC TEST) - SPECIFICALLY SELECTA

```

8834 071150 005037 071562          CLR      ANS225          ;SET DELTA T TO ZERO
8835 071154 004737 017066      50$:    CALL      CALRET      ;READ OPERATOR KEYBOARD ACTION
8836 071160 103762                BCS      40$           ;BRANCH IF CR WAS TYPED TO NEXT VALUE
8837 071162 013704 071552          MOV      ANS221,R4      ;SAVE CHANNEL
8838 071166 010477 112512      60$:    MOV      R4,@CSA    ;LOAD CHANNEL
8839 071172 013703 004164          MOV      REDTIM,R3     ;LOAD READY TIME OUT COUNTER
8840 071176 032777 000200 112500 70$:    BIT      #200,@CSA    ;READY SET
8841 071204 001010                BNE      80$           ;BRANCH IF YES
8842 071206 005303                DEC      R3            ;DECREMENT TIME OUT COUNTER
8843 071210 004737 012146          CALL     INSERT        ;SKIP BRANCH IF 'SFI' IS SET
8844 071214 001370                BNE      70$           ;BRANCH IF COUNTER IS NOT ZERO
8845 071216                ERRSOFT 2201,E2201,EER3 ;ERROR HANDLER
                                TRAP      C$ERSOFT
                                .WORD    2201
                                .WORD    E2201
                                .WORD    EER3
                                ;
8846 071226 005702      80$:    TST      R2            ;WAS CSA MESSAGES PRINTED ONCE?
8847 071230 001012                BNE      90$           ;SKIP PRINTOUT IF YES
8848 071232                PRINTB  #PME222,@CSA ;PRINT CSA STATUS
                                MOV      @CSA,-(SP)
                                MOV      #PME222,-(SP)
                                MOV      #2,-(SP)
                                MOV      SP,R0
                                TRAP      C$PNTB
                                ADD      #6,SP
8849 071256 062706 000006 112416 90$:    MOV      ANS226,@DAT    ;LOAD DATA REG. WITH OPERATOR INPUT
8850 071264 020437 071554          CMP      R4,ANS222     ;HAVE WE READ THE LAST CHANNEL ?
8851 071270 103003                BHS      100$          ;BRANCH IF YES
8852 071272 062704 000400          ADD      #400,R4       ;INCREMENT CHANNEL
8853 071276 000733                BR       60$           ;GO INTO LOOP, EXIT WITH CNTL C
8854 071300 012702 000001      100$:   MOV      #1,R2        ;SET INHIBIT CSA PRINTOUT FLAG
8855 071304 000723                BR       50$           ;BRANCH
8856
8857                                ;THE FOLLOWING CODE CREATES A RAMP ON THE OUTPUT
8858
8859 071306 005002      110$:   CLR      R2            ;CLR DELTA U SAVE LOCATION
8860 071310 013704 071552          MOV      ANS221,R4     ;GET FIRST CHANNEL
8861 071314 000412                BR       130$          ;
8862 071316 004737 017066      120$:   CALL     CALRET        ;READ OPERATOR KEYBOARD ACTION
8863 071322 103650                BCS      30$           ;IF CR WAS TYPED, BRANCH
8864 071324 013704 071552          MOV      ANS221,R4     ;GET FIRST CHANNEL NUMBER
8865 071330 063702 071560          ADD      ANS224,R2     ;GET DELTA U
8866 071334 022702 007777          CMP      #4095.,R2    ;FULL SCALE REACHED
8867 071340 103436                BLO      180$          ;IF YES, BRANCH TO NEGATIVE RAMP
8868 071342 013701 071562      130$:   MOV      ANS225,R1     ;LOAD DELTA T COUNTER
8869 071346 010477 112332          MOV      R4,@CSA      ;LOAD CHANNEL NUMBER
8870 071352 013703 004164          MOV      REDTIM,R3    ;LOAD READY TIMEOUT COUNTER
8871 071356 032777 000200 112320 140$:   BIT      #200,@CSA    ;READY SET
8872 071364 001010                BNE      150$          ;BRANCH IF YES
8873 071366 005303                DEC      R3            ;DECREMENT TIMEOUT COUNTER
8874 071370 004737 012146          CALL     INSERT        ;SKIP BRANCH IF 'SFI' IS SET
8875 071374 001370                BNE      140$          ;BRANCH IF COUNTER IS NOT ZERO
8876 071376                ERRSOFT 2202,E2201,EER3 ;ERROR HANDLER
                                TRAP      C$ERSOFT
                                .WORD    2202
                                .WORD    E2201
                                .WORD    EER3
                                ;
                                071376 104457
                                071400 004232
                                071402 072430
                                071404 007442

```


TEST 22: ANALOGUE OUTPUT (DYNAMIC TEST) - SPECIFICALLY SELECTA

```

8877 071406 010277 112270      150$:  MOV      R2,@DAT      ;LOAD DELTA U IN DATA REG.
8878 071412 005701              TST      R1                ;IS DELTA T ZERO ?
8879 071414 001402              BEQ      170$              ;BRANCH IF YES
8880 071416 005301      160$:  DEC      R1                ;IS DELTA T OVER ?
8881 071420 001376              BNE      160$              ;BRANCH IF NO
8882 071422 020437 071554      170$:  CMP      R4,ANS222      ;HAVE WE READ THE LAST CHANNEL ?
8883 071426 103333              BHIS     120$              ;IF YES, BRANCH TO FIRST CHANNEL
8884 071430 062704 000400      ADD      @400,R4          ;IF NO, INCREMENT CHANNEL
8885 071434 000742              BR       130$              ;AND DO NEXT ONE WITH SAME DELTA U
8886
8887
8888      ;THE FOLLOWING CODE IS FOR NEGATIVE RAMP
8889 071436 013704 071552      180$:  MOV      ANS221,R4      ;GET FIRST CHANNEL NUMBER
8890 071442 163702 071560      SUB      ANS224,R2      ;DO NEGATIVE RAMP
8891 071446 005702              TST      R2                ;HAVE WE REACHED THE BOTTOM (ZERO)
8892 071450 001716              BEQ      110$              ;IF YES, BRANCH TO POSITIV RAMP
8893 071452 013701 071562      190$:  MOV      ANS225,R1      ;LOAD DELTA T
8894 071456 010477 112222      MOV      R4,@CSA        ;LOAD CHANNEL NUMBER
8895 071462 013703 004164      MOV      REDTIM,R3      ;LOAD READY TIMEOUT COUNTER
8896 071466 032777 000200 112210 200$:  BIT      @200,@CSA      ;IS READY THERE ?
8897 071474 001010              BNE      210$              ;BRANCH IF YES
8898 071476 005303              DEC      R3                ;DECREMENT TIMEOUT COUNTER
8899 071500 004737 012146      CALL     INSERT          ;SKIP BRANCH IF 'SFI' IS SET
8900 071504 001370              BNE      200$              ;BRANCH IF COUNTER IS NOT ZERO
8901 071506              ERRSOFT 2203,E2201,EER3 ;ERROR HANDLER
      TRAP      C$ERSOFT
      .WORD     2203
      .WORD     E2201
      .WORD     EER3
8902 071516 010277 112160      210$:  MOV      R2,@DAT
8903 071522 005701              TST      R1                ;...
8904 071524 001402              BEQ      230$              ;IS DELTA T ZERO ?
8905 071526 005301      220$:  DEC      R1                ;BRANCH IF YES
8906 071530 001376              BNE      220$              ;DECREMENT DELTA T COUNTER
8907 071532 020437 071554      230$:  CMP      R4,ANS222      ;UNTILL WE HAVE ZERO
8908 071536 103337              BHIS     180$              ;HAVE WE READ THE LAST CHANNEL ?
8909 071540 062704 000400      ADD      @400,R4          ;BRANCH IF YES
8910 071544 000742              BR       190$              ;INCREMENT CHANNEL
8911 071546              EXQV22: EXIT      TST      ;DO NEXT CHANNEL WITH SAME DELTA U
      ;EXIT TEST
      TRAP      C$EXIT
      .WORD     L10057-.
8912
8913 071552 000000      ANS221: .WORD     0        ;SAVE LOCATION FOR FIRST CHANNEL
8914 071554 000000      ANS222: .WORD     0        ;SAVE LOCATION FOR SECOND CHANNEL
8915 071556 000002      ANS223: .WORD     2        ;SAVE LOCATION FOR PATTERN TYPE
8916 071560 000200      ANS224: .WORD     128.    ;SAVE LOCATION FOR DELTA U
8917 071562 000144      ANS225: .WORD     100.    ;SAVE LOCATION FOR DELTA T
8918 071564 004000      ANS226: .WORD     4000    ;SAVE LOCATIN FOR STATIC VALUE
8919
8920      .NLIST      BEX
8921 071566      045      123      062  TSHD22: .ASCIZ  /#S2#ANALOGUE OUTPUT (DYNAMIC) - SPECIFICALLY SELECTABLE.#N/
8922 071662      106      111      122  MES221: .ASCIZ  /FIRST CHANNEL/
8923 071700      114      101      123  MES222: .ASCIZ  /LAST CHANNEL/
8924 071715      120      101      124  MES223: .ASCIZ  /PATTERN TYPE (1 = SINGLE FIXED, 2 = RAMP)/
8925 071767      116      125      115  MES224: .ASCIZ  /NUMBER OF BINARY STEPS (DELTA U)(1-4096)/
8926 072040      124      111      115  MES225: .ASCIZ  /TIME INTERVAL (INCREMENTS) BETWEEN DELTA U STEPS/
8927 072121      105      116      124  MES226: .ASCIZ  /ENTER FIXED VALUE (OCTAL 0-7777)/

```

TEST 22: ANALOGUE OUTPUT (DYNAMIC TEST) - SPECIFICALLY SELECTA

```

8928
8929 072162      045      101      114  PME221: .ASCIZ /#ALAST CHANNEL HAS TO BE GREATER OR EQUAL TO THE FIRST ONE.#N/
8930 07226J     045      116      045  PME222: .ASCIZ /#N#ACONTROL STATUS REGISTER STATUS :#06#N/
8931 072332     045      116      045  PME223: .ASCIZ /#N#ATYPE CNTL C TO ABORT OR CR FOR NEXT DELTA T AND U VALUE#N/
8932 072430     122      105      101  E2201: .ASCIZ /READY BIT IN CSA NOT SET AFTER LOADING DATA REGISTER (TIMEOUT)/
8933             .LIST      BEX
8934             .EVEN
8935 072530             .ENDTST
      072530
      072530 104401

```

L10057: TRAP C\$ETST

TEST 23: ANALOGUE OUTPUT (CALIBRATION ROUTINE)

```

8937 .SBTTL TEST 23: ANALOGUE OUTPUT (CALIBRATION ROUTINE)
8938 ;*****
8939 ; TEST 23 - ANALOGUE OUTPUT (CALIBRATION ROUTINE) - SPECIFICALLY
8940 ; SELECTABLE.
8941 ;
8942 ; THIS TEST IS FOR CALIBRATION AND ALLOWS THE USER TO
8943 ; FOLLOW A CALIBRATION PROCEDURE OR TO DO A MANUAL CALIBRATION.
8944 ;
8945 ; FIRST OF ALL, THE USER GETS SOME INFORMATION ABOUT THE
8946 ; CALIBRATION EQUIPMENT. HE IS THEN IS PROMPTED FOR THE CHANNEL
8947 ; AND THE OUTPUT TYPE (VOLTAGE OR CURRENT). IF CURRENT IS
8948 ; SELECTED, THE PROGRAM LOOKS AT THE COFS BIT IN THE CSA TO FIND
8949 ; OUT THE OFFSET CONFIGURATION. NOW THE OPERATOR CAN CHOOSE
8950 ; BETWEEN THE CALIBRATION PROCEDURE OR MANUAL VALUES; IF MANUAL IS
8951 ; USED, THE OPERATOR CAN TYPE IN A VALUE IN MILLIVOLTS OR
8952 ; MICROAMPS.
8953 ;
8954 ; OTHERWISE HE IS PROMPTED TO CALIBRATE FIRST THE OFFSET AND THEN
8955 ; THE GAIN. THEN HE IS REQUIRED TO TEST THE HALF FULL SCALE. TO GO
8956 ; FROM ONE STEP TO THE NEXT, THE USER MUST TYPE CARRIAGE RETURN.
8957 ; ALL STEPS (OFFSET,GAIN AND HALF FULL SCALE) MUST BE REPEATED
8958 ; BEFORE GOING ON TO THE NEXT CHANNEL.
8959 ;*****
8960 072532 BGNTST
      072532
8961 072532 004737 011714 CALL SELECT ;CALL SELECT ROUTINE
8962 072536 010010 .WORD 10010 ;GIVE TEST PARAMETER (SPEC/AO)
8963 072540 075120 TSHD23 ;GIVE TEST HEADER
8964 072542 103002 BCC ITRC23 ;IF CARRY IS CLEARED, EXIT TEST
8965 072544 EXIT TST ;EXIT TEST
      072544 104432 TRAP C$EXIT
      072546 004340 .WORD L10060-.
8966 072550 ITRC23: PRINTF #PME23A ;PRINT CALIBRATION EQUIPMENT
      072550 012746 075402 MOV #PME23A,-(SP)
      072554 012746 000001 MOV #1,-(SP)
      072560 010600 MOV SP,RO
      072562 104417 TRAP C$PNTF
      072564 062706 000004 ADD #4,SP
8967 072570 PRINTF #PME23B ;PRINT CALIBRATION EQUIPMENT
      072570 012746 075530 MOV #PME23B,-(SP)
      072574 012746 000001 MOV #1,-(SP)
      072600 010600 MOV SP,RO
      072602 104417 TRAP C$PNTF
      072604 062706 000004 ADD #4,SP
8968 072610 PRINTF #PME23D ;PRINT CALIBRATION EQUIPMENT
      072610 012746 075626 MOV #PME23D,-(SP)
      072614 012746 000001 MOV #1,-(SP)
      072620 010600 MOV SP,RO
      072622 104417 TRAP C$PNTF
      072624 062706 000004 ADD #4,SP
8969 072630 PRINTF #PME23E ;PRINT CALIBRATION EQUIPMENT
      072630 012746 075715 MOV #PME23E,-(SP)
      072634 012746 000001 MOV #1,-(SP)
      072640 010600 MOV SP,RO
      072642 104417 TRAP C$PNTF
      072644 062706 000004 ADD #4,SP
8970 072650 005005 CHA23: CLR R5 ;CLEAR SECOND PASS COUNTER

```

TEST 23: ANALOGUE OUTPUT (CALIBRATION ROUTINE)

```

8971 072652          GMANID MES231,ANS231,D,3400,0,3,YES ;WHICH CHANNEL SHOULD CALIBR.
      072652 104443          TRAP      C%GMAN
      072654 000406          BR        10000%
      072656 074752          .WORD    ANS231
      072660 000052          .WORD    T%CODE
      072662 075214          .WORD    MES231
      072664 003400          .WORD    3400
      072666 000000          .WORD    T%LOLIM
      072670 000003          .WORD    T%HILIM
      072672          10000%:
8972 072672 042777 003400 111004 BIC      @3400,@CSA ;CLEAR CHANNEL NUMBER
8973 072700 053777 074752 110776 BIS      ANS231,@CSA ;LOAD CHANNEL NUMBER
8974 072706 013704 074752          MOV      ANS231,R4 ;GET CHANNEL NUMBER
8975 072712 000304          SWAB     R4 ;SWAB HIGH BYTE INTO LOW BYTE
8976 072714 032777 100000 110762 BIT      @100000,@CSA ;IS THE CHANNEL AVAILABLE ?
8977 072722 001011          BNE     10% ;BRANCH IF YES
8978 072724          PRINTF @PME233 ;PRINT CHA. NOT AVAILABLE
      072724 012746 076121          MOV      @PME233,-(SP)
      072730 012746 000001          MOV      @1,-(SP)
      072734 010600          MOV      SP,R0
      072736 104417          TRAP    C%PNTF
      072740 062706 000004          ADD     @4,SP
8979 072744 000741          BR      CHA23 ;
8980 072746          10%: GMANID MES232,ANS232,0,-1,0,1,YES ;CURRENT OR VOLTAGE ?
      072746 104443          TRAP    C%GMAN
      072750 000406          BR      10001%
      072752 074754          .WORD    ANS232
      072754 000032          .WORD    T%CODE
      072756 075243          .WORD    MES232
      072760 177777          .WORD    -1
      072762 000000          .WORD    T%LOLIM
      072764 000001          .WORD    T%HILIM
      072766          10001%:
8981 072766          GMANIL MES233,ANS233,-1,YES ;PROCEDURE OR MANUAL ?
      072766 104443          TRAP    C%GMAN
      072770 000404          BR      10002%
      072772 074756          .WORD    ANS233
      072774 000130          .WORD    T%CODE
      072776 075320          .WORD    MES233
      073000 177777          .WORD    -1
      073002          10002%:
8982 073002 005037 004134 CLR      MODE ;LOAD DACON INPUT (UNIPOLAR)
8983 073006 005737 074754 TST     ANS232 ;IS VOLTAGE USED ?
8984 073012 001012          BNE     20% ;BRANCH IF YES
8985 073014 012737 000002 004134 MOV      @2,MODE ;LOAD DACON INPUT (0-20MA)
8986 073022 032777 000010 110650 BIT      @10,@MOD ;READ OFFSET CONFIGURATION
8987 073030 001403          BEQ     20% ;BRANCH IF 0 MA OFFSET
8988 073032 012737 000003 004134 MOV      @3,MODE ;LOAD DACON INPUT (4-20MA)
8989 073040 005037 004136          CLR     GAIN ;LOAD DACON INPUT (GAIN)
8990 073044 022737 000001 074756 20%: CMP      @1,ANS233 ;IS MANUAL VALUE USED ?
8991 073052 001402          BEQ     30% ;BRANCH IF NO
8992 073054 000137 074544          JMP     MAN23 ;IF YES JUMP TO MANUAL ROUT.
8993 073060 005002          30%: CLR     R2 ;CLEAR INHIBIT PRINT FLAG
8994 073062 013703 004164          40%: MOV     REDTIM,R3 ;LOAD TIMEOUT COUNTER
8995 073066 032777 000200 110610 50%: BIT     @200,@CSA ;READY SET ?
8996 073074 001010          BNE     60% ;BRANCH IF YES
8997 073076 005303          DEC     R3 ;DECREMENT TIMEOUT COUNTER

```

TEST 23: ANALOGUE OUTPUT (CALIBRATION ROUTINE)

```

8998 073100 004737 012146          CALL  INSERT          ;SKIP BRANCH IF 'SFI' IS SET
8999 073104 001370          BNE   50$             ;BRANCH IF NOT ZERO
9000 073106          ERRSOFT 2301,E2301,EER3 ;ERROR HANDLER
           073106 104457          TRAP  C$ERRSOFT
           073110 004375          .WORD 2301
           073112 077021          .WORD E2301
           073114 007442          .WORD EER3

9001
9002
9003 073116 012777 000001 110556 60$: MOV   #1, @DAT          ;LOAD DATA REG.
9004 073124 005702          TST   R2              ;WAS THE OFFSET CALIB. MES. PRINTED
9005 073126 001175          BNE   90$             ;BRANCH IF YES
9006 073130 012701 000001          MOV   #1, R1          ;SETUP R1 FOR PRINTOUT
9007 073134          PRINTF #PME235, R1, R4 ;PRINT CHANNEL NUMBER * DAT CONT.
           073134 010446          MOV   R4, -(SP)
           073136 010146          MOV   R1, -(SP)
           073140 012746 076275          MOV   #PME235, -(SP)
           073144 012746 000003          MOV   #3, -(SP)
           073150 010600          MOV   SP, R0
           073152 104417          TRAP  C$PNTF
           073154 062706 000010          ADD   #10, SP
9008 073160 013703 004134          MOV   MODE, R3        ;LOAD TABLE START ADDRESS (O/P=1)
9009 073164 006303          ASL   R3              ;
9010 073166 006303          ASL   R3              ;
9011 073170 006303          ASL   R3              ;
9012 073172 016301 074760          MOV   TAB23(R3), R1   ;LOAD R1 FOR DECOUT ROUTINE
9013 073176 016302 074762          MOV   TAB23+2(R3), R2 ;LOAD R2 FOR DECOUT ROUTINE
9014 073202 005737 074754          TST   ANS232         ;WAS VOLTAGE RANGE REQUESTED
9015 073206 001025          BNE   66$             ;BRANCH IF YES
9016 073210 032777 000010 110462          BIT   #10, @MOD      ;IS OFFSET BIT SET ?
9017 073216 001011          BNE   62$             ;BRANCH IF YES
9018 073220          PRINTF #PME231         ;PRINT 0 - 20 MA RANGE IS SELECTED
           073220 012746 075753          MOV   #PME231, -(SP)
           073224 012746 000001          MOV   #1, -(SP)
           073230 010600          MOV   SP, R0
           073232 104417          TRAP  C$PNTF
           073234 062706 000004          ADD   #4, SP
9019 073240 000410          BR    66$             ;SKIP 4 - 20 MA PRINTOUT
9020 073242          PRINTF #PME232         ;PRINT 4 - 20 MA RANGE IS SELECTED
           073242 012746 076040          MOV   #PME232, -(SP)
           073246 012746 000001          MOV   #1, -(SP)
           073252 010600          MOV   SP, R0
           073254 104417          TRAP  C$PNTF
           073256 062706 000004          ADD   #4, SP
9021 073262          PRINTF #PME234         ;PRINT OFFSET CALIBRATION
           073262 012746 076205          MOV   #PME234, -(SP)
           073266 012746 000001          MOV   #1, -(SP)
           073272 010600          MOV   SP, R0
           073274 104417          TRAP  C$PNTF
           073276 062706 000004          ADD   #4, SP
9022 073302 004737 015764          CALL  DECOUT         ;CALL DECIMAL OUT ROUTINE
9023 073306 005737 074754          TST   ANS232         ;IS VOLTAGE USED ?
9024 073312 001043          BNE   70$             ;BRANCH IF YES
9025 073314          PRINTF #OUT231        ;PRINT 'MICROAMPS'
           073314 012746 077002          MOV   #OUT231, -(SP)
           073320 012746 000001          MOV   #1, -(SP)
           073324 010600          MOV   SP, R0

```

TEST 23: ANALOGUE OUTPUT (CALIBRATION ROUTINE)

073326	104417						TRAP	C\$PNTF
9026	073330	062706	000004				ADD	#4,SP
9027	073334	006201		ASR	R1		;DIVIDE BY TWO FOR VOLTAGE VALUE	
9028	073336	006002		ROR	R2		;DIVIDE BY TWO ADD CARRY FOR VOLTAGE	
	073340			PRINTF	#PME239,R1,R2		;PRINT VOLTAGE ON 500 OHM RESISTOR	
	073340	010246					MOV	R2,-(SP)
	073342	010146					MOV	R1,-(SP)
	073344	012746	076576				MOV	#PME239,-(SP)
	073350	012746	000003				MOV	#3,-(SP)
	073354	010600					MOV	SP,RO
	073356	104417					TRAP	C\$PNTF
9029	073360	062706	000010				ADD	#10,SP
9030	073364	016301	074764	MOV	TAB23*4(R3),R1		;GET VOLTAGE ON 500 OHM TOLLERANCE	
9031	073370	016302	074766	MOV	TAB23*6(R3),R2		;...	
	073374			PRINTF	#TOL1,R1,R2		;AND PRINT IT	
	073374	010246					MOV	R2,-(SP)
	073376	010146					MOV	R1,-(SP)
	073400	012746	076707				MOV	#TOL1,-(SP)
	073404	012746	000003				MOV	#3,-(SP)
	073410	010600					MOV	SP,RO
	073412	104417					TRAP	C\$PNTF
9032	073414	062706	000010				ADD	#10,SP
9033	073420	000426		BR	80\$;AND SKIP MILLIVOLTS PRINTOUT	
	073422			PRINTF	#OUT232		;PRINT 'MILLIVOLTS'	
	073422	012746	076762				MOV	#OUT232,-(SP)
	073426	012746	000001				MOV	#1,-(SP)
	073432	010600					MOV	SP,RO
	073434	104417					TRAP	C\$PNTF
9034	073436	062706	000004				ADD	#4,SP
9035	073442	016301	074764	MOV	TAB23*4(R3),R1		;GET VOLTAGE TOLLERANCE	
9036	073446	016302	074766	MOV	TAB23*6(R3),R2		;...	
	073452			PRINTF	#TOL1,R1,R2		;AND PRINT IT	
	073452	010246					MOV	R2,-(SP)
	073454	010146					MOV	R1,-(SP)
	073456	012746	076707				MOV	#TOL1,-(SP)
	073462	012746	000003				MOV	#3,-(SP)
	073466	010600					MOV	SP,RO
	073470	104417					TRAP	C\$PNTF
9037	073472	062706	000010				ADD	#10,SP
	073476			80\$: PRINTF	#PME236		;PRINT HOW TO CONTINUE	
	073476	012746	076352				MOV	#PME236,-(SP)
	073502	012746	000001				MOV	#1,-(SP)
	073506	010600					MOV	SP,RO
	073510	104417					TRAP	C\$PNTF
	073512	062706	000004				ADD	#4,SP
9038	073516	012702	000001				;SET INHIBIT OFFSET PRINTOUT	
9039	073522	004737	017066	90\$: CALL	CALRET		;READ OPERATOR KEYBOARD ACTION	
9040	073526	103402		BCS	95\$;BRANCH IF CARRIEGE RETURN TO NEXT	
9041	073530	000137	073062	JMP	40\$;REPEAT IF NO CARRIEGE RETURN FOUND	
9042								
9043	073534	005002		95\$: CLR	R2		;	
9044	073536	013703	004164	100\$: MOV	REDTIM,R3		;LOAD READY TIMEOUT COUNTER	
9045	073542	032777	000200	110134 110\$: BIT	#200,@CSA		;READY SET	
9046	073550	001010		BNE	120\$;BRANCH IF YES	
9047	073552	005303		DEC	R3		;DECREMENT TIMEOUT COUNTER	
9048	073554	004737	012146	CALL	INSERT		;SKIP BRANCH IF 'SFI' IS SET	
9049	073560	001370		BNE	110\$;BRANCH IF COUNTER IS NOT ZERO	

TEST 23: ANALOGUE OUTPUT (CALIBRATION ROUTINE)

```

9050 073562          ERRSOFT 2302,E2301,EER3
      073562 104457
      073564 004376
      073566 077021
      073570 007442
9051 073572 012777 007777 110102 120$: MOV    #7777,@DAT
9052 073600 005702          TST    R2
9053 073602 001145          BNE    150$
9054 073604 012701 007777          MOV    #7777,R1
9055 073610          PRINTF #PME235,R1,R4
      073610 010446
      073612 010146
      073614 012746 076275
      073620 012746 000003
      073624 010600
      073626 104417
      073630 062706 000010
9056 073634 013703 004134          MOV    MODE,R3
9057 073640 006303          ASL    R3
9058 073642 006303          ASL    R3
9059 073644 006303          ASL    R3
9060 073646 016301 075020          MOV    TAB23.40(R3),R1
9061 073652 016302 075022          MOV    TAB23.42(R3),R2
9062 073656          PRINTF #PME237
      073656 012746 076421
      073662 012746 000001
      073666 010600
      073670 104417
      073672 062706 000004
9063 073676 004737 015764          CALL   DECOUT
9064 073702 005737 074754          TST    ANS232
9065 073706 001043          BNE    130$
9066 073710          PRINTF #OUT231
      073710 012746 077002
      073714 012746 000001
      073720 010600
      073722 104417
      073724 062706 000004
9067 073730 006201          ASR    R1
9068 073732 006002          ROR    R2
9069 073734          PRINTF #PME239,R1,R2
      073734 010246
      073736 010146
      073740 012746 076576
      073744 012746 000003
      073750 010600
      073752 104417
      073754 062706 000010
9070 073760 016301 075024          MOV    TAB23.44(R3),R1
9071 073764 016302 075026          MOV    TAB23.46(R3),R2
9072 073770          PRINTF #TOL1,R1,R2
      073770 010246
      073772 010146
      073774 012746 076707
      074000 012746 000003
      074004 010600
      074006 104417

```

```

;ERROR HANDLER
      TRAP   C$ERRSOFT
      .WORD  2302
      .WORD  E2301
      .WORD  EER3
;LOAD DATA REG.
;WAS THE GAIN CALIB. MES. PRINTED
;BRANCH IF YES
;SETUP R1 FOR PRINTOUT
;PRINT CHANNEL NUMBER + DAT CONT.
      MOV    R4,-(SP)
      MOV    R1,-(SP)
      MOV    #PME235,-(SP)
      MOV    #3,-(SP)
      MOV    SP,R0
      TRAP   C$PNTF
      ADD    #10,SP
;LOAD TABLE START POINT (O/P=7777)
;BUILD EVEN ADDRESS
;
;...
;LOAD R1 FOR DECOUT ROUTINE
;LOAD R2 FOR DECOUT ROUTINE
;PRINT GAIN CALIBRATION
      MOV    #PME237,-(SP)
      MOV    #1,-(SP)
      MOV    SP,R0
      TRAP   C$PNTF
      ADD    #4,SP
;CALL DECIMAL OUT ROUTINE
;IS VOLTAGE USED ?
;BRANCH IF YES
;PRINT 'MICROAMPS'
      MOV    #OUT231,-(SP)
      MOV    #1,-(SP)
      MOV    SP,R0
      TRAP   C$PNTF
      ADD    #4,SP
;DIVIDE BY TWO FOR VOLTAGE VALUE
;DIVIDD BY TWO ADD CARRY FOR VOLTAGE
;PRINT VOLTAGE ON 500 OHM RESISTOR
      MOV    R2,-(SP)
      MOV    R1,-(SP)
      MOV    #PME239,-(SP)
      MOV    #3,-(SP)
      MOV    SP,R0
      TRAP   C$PNTF
      ADD    #10,SP
;GET VOLTAGE ON 500 OHM TOLLERANCE
;...
;AND PRINT IT
      MOV    R2,-(SP)
      MOV    R1,-(SP)
      MOV    #TOL1,-(SP)
      MOV    #3,-(SP)
      MOV    SP,R0
      TRAP   C$PNTF

```

TEST 23: ANALOGUE OUTPUT (CALIBRATION ROUTINE)

SEQ 0238

```

9073 074010 062706 000010
9074 074014 000426
          130$: BR      140$
          PRINTF  #OUT232
          ;AND SKIP MILLIVOLT PRINTOUT
          ;PRINT 'MILLIVOLTS'
          ADD      #10,SP
          MOV      #OUT232,-(SP)
          MOV      #1,-(SP)
          MOV      SP,R0
          TRAP    C:PNTF
          ADD      #4,SP
9075 074016 012746 076762
          074022 012746 000001
          074026 010600
          074030 104417
          074032 062706 000004
9076 074036 016301 075024
          074042 016302 075026
          MOV      TAB23+44(R3),R1
          ;GET VOLTAGE ON 500 OHM TOLLERANCE
          ;...
          PRINTF  #TOL1,R1,R2
          ;AND PRINT IT
          MOV      R2,-(SP)
          MOV      R1,-(SP)
          MOV      #TOL1,-(SP)
          MOV      #3,-(SP)
          MOV      SP,R0
          TRAP    C:PNTF
          ADD      #10,SP
9077 074046 010246
          074050 010146
          074052 012746 076707
          074056 012746 000003
          074062 010600
          074064 104417
          074066 062706 000010
9078 074072 012746 076352
          074072 012746 000001
          074102 010600
          074104 104417
          074106 062706 000004
9079 074112 012702 000001
          074116 004737 017066
          074122 103402
          074124 000137 073536
          MOV      #1,R2
          ;SET INHIBIT GAIN PRINTOUT
          CALL    CALRET
          ;READ OPERATOR KEYBOARD ACTION
          BCS     155$
          ;BRANCH IF CARRIAGE RETURN
          JMP     100$
          ;REPEAT IF NO CARRIGE RETURN FOUO
          CLR     R2
          ;CLEAR INHIBIT HALF SCALE PRI. FLAG
          MOV     REDTIM,R3
          ;LOAD READY TIMEOUT COUNTER
          160$: BIT     #200,@CSA
          170$: BNE     180$
          ;READY SET
          ;BRANCH IF YES
          DEC     R3
          ;DECREMENT TIMEOUT COUNTER
          CALL    INSERT
          ;SKIP BRANCH IF 'SFI' IS SET
          BNE     170$
          ;BRANCH IF NOT ZERO
          ERRSOFT 2303,E2301,EER3
          ;ERROR HANDLER
          TRAP    C:ERSOFT
          .WORD  2303
          .WORD  E2301
          .WORD  EER3
9083 074130 005002
          074132 013703 004164
          074136 032777 000200 107540
          074144 001010
          074146 005303
          074150 004737 012146
          074154 001370
          074156 104457
          074160 004377
          074162 077021
          074164 007442
9092 074166 012777 004000 107506 180$: MOV     #4000,@DAT
          074174 005702
          074176 001145
          074200 012701 004000
          074204 010446
          074206 010146
          074210 012746 076275
          074214 012746 000003
          074220 010600
          074222 104417
          074224 062706 000010
          074230 013703 004134
          074234 006303
          074236 006303
          074240 006303
          MOV      MODE,R3
          ;LOAD TABLE START POINT (O/P=4000)
          ASL     R3
          ;BUILD EVEN ADDRESS
          ASL     R3
          ;...
          ASL     R3
          ;...
          MOV      R4,-(SP)
          MOV      R1,-(SP)
          MOV      #PME235,-(SP)
          MOV      #3,-(SP)
          MOV      SP,R0
          TRAP    C:PNTF
          ADD      #10,SP

```


TEST 23: ANALOGUE OUTPUT (CALIBRATION ROUTINE)

9101	074242	016301	075060	MOV	TAB23+100(R3),R1	;LOAD R1 FOR DECOUT ROUTINE
9102	074246	016302	075062	MOV	TAB23+102(R3),R2	;LOAD R2 FOR DECOUT ROUTINE
9103	074252			PRINTF	#PME238	;PRINT HALF SCALE CALIBRATION
	074252	012746	076507			MOV #PME238,-(SP)
	074256	012746	000001			MOV #1,-(SP)
	074262	010600				MOV SP,R0
	074264	104417				TRAP C#PNTF
	074266	062706	000004			ADD #4,SP
9104	074272	004737	015764	CALL	DECOUT	;CALL DECIMAL OUT ROUTINE
9105	074276	005737	074754	TST	ANS232	;IS VOLTAGE USED ?
9106	074302	001043		BNE	190\$;BRANCH IF YES
9107	074304			PRINTF	#OUT231	;PRINT 'MICROAMPS'
	074304	012746	077002			MOV #OUT231,-(SP)
	074310	012746	000001			MOV #1,-(SP)
	074314	010600				MOV SP,R0
	074316	104417				TRAP C#PNTF
	074320	062706	000004			ADD #4,SP
9108	074324	006201		ASR	R1	;DIVIDE BY TWO FOR VOLTAGE VALUE
9109	074326	006002		ROR	R2	;DIVIDE BY TWO ADD CARRY FOR VOLTAGE
9110	074330			PRINTF	#PME239,R1,R2	;PRINT VOLTAGE ON 500 OHM RESISTOR
	074330	010246				MOV R2,-(SP)
	074332	010146				MOV R1,-(SP)
	074334	012746	076576			MOV #PME239,-(SP)
	074340	012746	000003			MOV #3,-(SP)
	074344	010600				MOV SP,R0
	074346	104417				TRAP C#PNTF
	074350	062706	000010			ADD #10,SP
9111	074354	016301	075064	MOV	TAB23+104(R3),R1	;GET VOLTAGE ON 500 OHM TOLLERANCE
9112	074360	016302	075066	MOV	TAB23+106(R3),R2	;...
9113	074364			PRINTF	#TOL1,R1,R2	;AND PRINT IT
	074364	010246				MOV R2,-(SP)
	074366	010146				MOV R1,-(SP)
	074370	012746	076707			MOV #TOL1,-(SP)
	074374	012746	000003			MOV #3,-(SP)
	074400	010600				MOV SP,R0
	074402	104417				TRAP C#PNTF
	074404	062706	000010			ADD #10,SP
9114	074410	000426		BR	200\$;AND SKIP MILLIVOLT PRINTOUT
9115	074412			PRINTF	#OUT232	;PRINT 'MILLIVOLTS'
	074412	012746	076762			MOV #OUT232,-(SP)
	074416	012746	000001			MOV #1,-(SP)
	074422	010600				MOV SP,R0
	074424	104417				TRAP C#PNTF
	074426	062706	000004			ADD #4,SP
9116	074432	016301	075064	MOV	TAB23+104(R3),R1	;GET VOLTAGE ON 500 OHM TOLLERANCE
9117	074436	016302	075066	MOV	TAB23+106(R3),R2	;...
9118	074442			PRINTF	#TOL1,R1,R2	;AND PRINT IT
	074442	010246				MOV R2,-(SP)
	074444	010146				MOV R1,-(SP)
	074446	012746	076707			MOV #TOL1,-(SP)
	074452	012746	000003			MOV #3,-(SP)
	074456	010600				MOV SP,R0
	074460	104417				TRAP C#PNTF
	074462	062706	000010			ADD #10,SP
9119	074466			PRINTF	#PME236	;PRINT HOW TO CONTINUE
	074466	012746	076352			MOV #PME236,-(SP)
	074472	012746	000001			MOV #1,-(SP)

TEST 23: ANALOGUE OUTPUT (CALIBRATION ROUTINE)

```

074476 010600
074500 104417
074502 062706 000004
9120 074506 012702 000001
9121 074512 004737 017066
9122 074516 103402
9123 074520 000137 074132
9124 074524 005705
9125 074526 001402
9126 074530 000137 072650
9127 074534 012705 000001
9128 074540 000137 073060
9129
9130
9131
9132
9133 074544 005737 074754
9134 074550 001004
9135 074552 012737 077002 004140
9136 074560 000403
9137 074562 012737 076762 004140
9138 074570
074570 012746 076655
074574 012746 000001
074600 010600
074602 104417
074604 062706 000004
9139 074610 004737 015204
9140 074614 004737 014730
9141 074620 005002
9142 074622 013703 004164
9143 074626 032777 000200 107050
9144 074634 001010
9145 074636 005303
9146 074640 004737 012146
9147 074644 001370
9148 074646
074646 104457
074650 004400
074652 077021
074654 007442
9149 074656 010177 107020
9150 074662 004737 017066
9151 074666 103726
9152 074670 005702
9153 074672 001353
9154 074674
074674 010446
074676 010146
074700 012746 076275
074704 012746 000003
074710 010600
074712 104417
074714 062706 000010
9155 074720
074720 012746 076352
074724 012746 000001

```

210\$: MOV #1,R2 ;SET INHIBIT PRNTOU FLAG
 CALL CALRET ;READ OPERATOR KEYBOARD ACTION
 BCS 215\$;IF CARRIGE RETURN NEXT STEP
 JMP 160\$;BRANCH IF NO CARRIAGE RETURN
 215\$: TST R5 ;WAS IT THE SECOND PASS ?
 BEQ 220\$;IF NO BRANCH
 JMP CHA23 ;IF YES JUMP TO CHANNEL PRINTOUT
 220\$: MOV #1,R5 ;SET SECOND PASS INDICATOR
 JMP 30\$;DO THE SECOND PASS

;THE FOLLOWING PART IS USED IF THE OPERATOR WANTS MANUAL VALUES FOR
 ;CALIBRATION

MAN23: TST ANS232 ;IS VOLTAGE USED ?
 BNE 10\$;BRANCH IF YES
 MOV #OUT231,PADD ;LOAD 'MICROAMPS' INFORMATION
 BR 20\$;AND JUMP
 10\$: MOV #OUT232,PADD ;LOAD 'MILLIVOLTS' INFORMATION
 20\$: PRINTF #PRMST ;PRINT ENTER MANUAL VALUE

MOV #PRMST,-(SP)
 MOV #1,-(SP)
 MOV SP,R0
 TRAP C\$PNTF
 ADD #4,SP

CALL DECIN ;REQUEST A DEC. NUM. FROM OPERATOR
 CALL ADCON ;CONVERT THE ANALOGUE VALUE TO DIGI.
 CLR R2 ;CLEAR INHIBIT MESSAGES FLAG
 30\$: MOV REDTIM,R3 ;LOAD TIMEOUT COUNTER
 40\$: BIT #200,@CSA ;READY SET
 BNE 50\$;BRANCH IF YES
 DEC R3 ;DECREMENT TIMEOUT COUNTER
 CALL INSERT ;SKIP BRANCH IF 'SFI' IS SET
 BNE 40\$;BRANCH IF COUNTER IS NOT ZERO
 ERRSOFT 2304,E2301,EER3 ;ERROR HANDLER

TRAP C\$ERSOFT
 .WORD 2304
 .WORD E2301
 .WORD EER3

50\$: MOV R1,@DAT ;LOAD DIGITAL VALUE INTO DATA REGISTER
 CALL CALRET ;READ OPERATOR KEYBOARD ACTION
 BCS MAN23 ;BRANCH IF 'CR' WAS TYPED
 TST R2 ;INHIBIT THE PRINT OUT ?
 BNE 30\$;BRANCH IF YES
 PRINTF #PME235,R1,R4 ;PRINT CHANNEL NUMBER * DAT CONT.

MOV R4,-(SP)
 MOV R1,-(SP)
 MOV #PME235,-(SP)
 MOV #3,-(SP)
 MOV SP,R0
 TRAP C\$PNTF
 ADD #10,SP

60\$: PRINTF #PME236 ;PRINT HOW TO CONTINUE

MOV #PME236,-(SP)
 MOV #1,-(SP)

TEST 23: ANALOGUE OUTPUT (CALIBRATION ROUTINE)

```

074730 010600
074732 104417
074734 062706 000004
9156 074740 012702 000001
9157 074744 000726
9158 074746
074746 104432
074750 002136
9159
9160 074752 000000
9161 074754 000000
9162 074756 000001
9163
9164
9165
9166 074760 000002 000671 000002 TAB23: .WORD 2.,441.,2,400. ;MODE 0 (LOADED VALUE INTO DAT = 1)
074766 000620
9167 074770 154365 000000 000002 .WORD -9995.,000,2,400. ;MODE 1 (LOADED VALUE INTO DAT = 1)
074776 000620
9168 075000 000004 001563 000004 .WORD 4.,883.,4,800. ;MODE 2 (LOADED VALUE INTO DAT = 1)
075006 001440
9169 075010 007643 000000 000004 .WORD 4003.,000,4,000 ;MODE 3 (LOADED VALUE INTO DAT = 1)
075016 000000
9170
9171 075020 023415 000000 000002 .WORD 9997.,000,2,400. ;MODE 0 (LOADED VALUE INTO DAT = 7777)
075026 000620
9172 075030 023413 000000 000002 .WORD 9995.,000,2,400. ;MODE 1 (LOADED VALUE INTO DAT = 7777)
075036 000620
9173 075040 047033 000000 000004 .WORD 19995.,000,4,800. ;MODE 2 (LOADED VALUE INTO DAT = 7777)
075046 001440
9174 075050 047034 000000 000004 .WORD 19996.,000,4,000 ;MODE 3 (LOADED VALUE INTO DAT = 7777)
075056 000000
9175
9176 075060 011610 000000 000002 .WORD 5000.,000,2,400. ;MODE 0 (LOADED VALUE INTO DAT = 4000)
075066 000620
9177 075070 000000 000000 000002 .WORD 000.,000,2,400. ;MODE 1 (LOADED VALUE INTO DAT = 4000)
075076 000620
9178 075100 023420 000000 000004 .WORD 10000.,000,4,800. ;MODE 2 (LOADED VALUE INTO DAT = 4000)
075106 001440
9179 075110 027340 000000 000004 .WORD 12000.,000,4,000 ;MODE 3 (LOADED VALUE INTO DAT = 4000)
075116 000000
9180
9181
9182
9183 075120 045 123 062 TSHD23: .NLIST BEX
9184 075214 103 110 101 MES231: .ASCIZ /S2#ANALOGUE OUTPUT (CALIBR.) - SPECIFICALLY SELECTABLE.#N/
9185 075243 103 101 114 MES232: .ASCIZ /CHANNEL TO CALIBRATE ?/
9186 075320 104 117 040 MES233: .ASCIZ /CALIBRATE FOR CURRENT (0) OR FOR VOLTAGE (1)/
9187
9188 075402 045 116 045 PME23A: .ASCII /#ACALIBRATION EQUIPMENT#N/
9189 075435 045 116 045 .ASCIZ /#ADIGITAL VOLTMETER TO MEASURE THE VOLTAGE AND CURRENT#N/
9190 075530 045 101 117 PME23B: .ASCIZ /#AOUTPUTS. THE DVM SHOULD HAVE AN ACCURACY OF 0.01 PERCENT.#N/
9191 075626 045 101 106 PME23D: .ASCIZ /#AFOR CURRENT ADJUSTMENT ATTACH THE 500 OHM RESISTOR#N/
9192 075715 045 101 111 PME23E: .ASCIZ /#AIN PARALLEL WITH THE DVM.#N/
9193
9194 075753 045 116 045 PME231: .ASCIZ /#A0 - 20 M# RANGE IS SELECTED (COFS BIT CLEARED)#N/
9195 076040 045 116 045 PME232: .ASCIZ /#A4 - 20 MA RANGE IS SELECTED (COFS BIT SET)#N/

```

```

MOV SP,R0
TRAP C$PNTF
ADD #4,SP
TRAP C$EXIT
.WORD L10060-

```

```

MOV #1,R2
BR 30$
EXQV23: EXIT TST

```

```

;SET INHIBIT PRINT FLAG
;GO INTO LOOP
;EXIT TEST
;SAVE LOCATION FOR CHANNEL
;SAVE LOCATION FOR CURRENT OR VOLT
;SAVE LOCATION FOR OPERATOR VALUE

```

```

ANS231: .WORD 0
ANS232: .WORD 0
ANS233: .WORD 1

```

```

TAB23: .WORD 2.,441.,2,400. ;MODE 0 (LOADED VALUE INTO DAT = 1)
.WORD -9995.,000,2,400. ;MODE 1 (LOADED VALUE INTO DAT = 1)
.WORD 4.,883.,4,800. ;MODE 2 (LOADED VALUE INTO DAT = 1)
.WORD 4003.,000,4,000 ;MODE 3 (LOADED VALUE INTO DAT = 1)
.WORD 9997.,000,2,400. ;MODE 0 (LOADED VALUE INTO DAT = 7777)
.WORD 9995.,000,2,400. ;MODE 1 (LOADED VALUE INTO DAT = 7777)
.WORD 19995.,000,4,800. ;MODE 2 (LOADED VALUE INTO DAT = 7777)
.WORD 19996.,000,4,000 ;MODE 3 (LOADED VALUE INTO DAT = 7777)
.WORD 5000.,000,2,400. ;MODE 0 (LOADED VALUE INTO DAT = 4000)
.WORD 000.,000,2,400. ;MODE 1 (LOADED VALUE INTO DAT = 4000)
.WORD 10000.,000,4,800. ;MODE 2 (LOADED VALUE INTO DAT = 4000)
.WORD 12000.,000,4,000 ;MODE 3 (LOADED VALUE INTO DAT = 4000)

```

TEST 23: ANALOGUE OUTPUT (CALIBRATION ROUTINE)

```

9196 076121      045      101      103  PME233: .ASCIZ  /#ACHANNEL NOT AVAILABLE, ENTER AVAILABLE CHANNEL #N/
9197 076205      045      116      045  PME234: .ASCIZ  /#N#AADJUST THE OFFSET TRIMPOT UNTIL THE DVM INDICATES :/
9198 076275      045      116      045  PME235: .ASCIZ  /#N#ABIT PATTERN #06#A OUTPUT TO CHANNEL :#D3/
9199 076352      045      116      045  PME236: .ASCIZ  /#N#ATYPE CARRIAGE RETURN TO CONTINUE#N/
9200 076421      045      116      045  PME237: .ASCIZ  /#N#AADJUST THE GAIN TRIMPOT UNTIL THE DVM INDICATES :/
9201 076507      045      116      045  PME238: .ASCIZ  /#N#ANOW CHECK THE HALF SCALE THE DVM SHOULD INDICATE :/
9202 076576      045      101      075  PME239: .ASCIZ  /#A= -#D5#A.#23#A MILLIVOLT ON 500 OHM RESISTOR/
9203 076655      045      116      045  PRMST:  .ASCIZ  /#N#AENTER MANUAL VALUE IN/
9204
9205 076707      045      101      040  TOL1:   .ASCIZ  \#A TOLLERANCE +/- #D1#A.#D3#A MILLIVOLTS#N\
9206 076762      045      101      040  OUT232: .ASCIZ  /#A MILLIVOLTS#N/
9207 077002      045      101      040  OUT231: .ASCIZ  /#A MICROAMPS#N/
9208
9209 077021      122      105      101  E2301:  .ASCIZ  /READY BIT IN CSA NOT SET AFTER LOADING DATA REGISTER/
9210          .LIST  BEX
9211          .EVEN
9212 077106          .ENDTST

```

077106 104401

L10060: TRAP C\$ETST

TEST 24: ANALOGUE INPUT (CALIBRATION ROUTINE)

```

9214 .SBTTL TEST 24: ANALOGUE INPUT (CALIBRATION ROUTINE)
9215 ;*****
9216 ; TEST 24 - ANALOGUE INPUT (CALIBRATION ROUTINE) - SPECIFICALLY
9217 ; SELECTABLE.
9218 ;
9219 ; THIS TEST IS FOR CALIBRATING AN ANALOGUE INPUT MODULE .
9220 ;
9221 ; FIRST OF ALL, THE USER IS ASKED WHETHER HE NEEDS INFORMATION ABOUT
9222 ; THE CALIBRATION PROCEDURE.
9223 ; HE IS THEN PROMPTED FOR THE CHANNEL ON WHICH HE HAS PUT THE PRECISION
9224 ; VOLTAGE REFERENCE SOURCE .
9225 ; AFTER THIS HE IS INFORMED WHICH RANGE IS SWITCHED (UNIPOLAR OR BIPOLAR)
9226 ; AND ASKED TO SELECT THE DESIRED GAIN.
9227 ; NOW THE PROGRAM CONTINUOUSLY READS THE INPUT AND PRINTS THE RESULT
9228 ; AS AN OCTAL AND VOLTAGE VALUE.
9229 ; TO ABORT THE TEST, "CNL C" MUST BE TYPED.
9230 ;*****
9231 077110          BGNTSI
9232 077110          T24::
9233 077110 004737 011714 CALL SELECT ;CALL SELECT ROUTINE
9234 077114 010004 .WORD 10004 ;GIVE TEST PARAMETER (SPEC/AI)
9235 077116 100100 TSHD24 ;GIVE TEST HEADER
9236 077120 103002 BCC ITRC24 ;IF CARRY IS CLEARED EXIT TEST
9237 077122          EXIT TST ;EXIT TEST
9238 077122 104432          TRAP C$EXIT
9239 077124 002754          .WORD L10061-.
9240 077126          ITRC24: GMANIL MES241,ANS241,-1,YES ;DO YOU WANT CALIBRATION HELP
9241 077126 104443          TRAP C$GMAN
9242 077130 000404          BR 10000$
9243 077132 100072          .WORD ANS241
9244 077134 000130          .WORD T$CODE
9245 077136 100173          .WORD MES241
9246 077140 177777          .WORD -1
9247 077142          10000$:
9248 077142 005737 100072 TST ANS241 ;DO YOU WANT HELP
9249 077146 001545 BEQ 20$ ;BRANCH IF ANSWER WAS NO
9250 077150 032777 000020 104522 BIT #20,@MOD ;IS UNIPOLAR RANGE SWITCHED
9251 077156 001461 BEQ 10$ ;BRANCH IF YES
9252 077160          PRINTF @PME242 ;PRINT BIPOLAR MESSAGES
9253 077160 012746 101530 MOV @PME242,-(SP)
9254 077164 012746 000001 MOV #1,-(SP)
9255 077170 010600 MOV SP,RO
9256 077172 104417 TRAP C$PNTF
9257 077174 062706 000004 ADD #4,SP
9258 077200          PRINTF @PME24A ;CALIBRATION INFORMATION GENERAL
9259 077200 012746 100364 MOV @PME24A,-(SP)
9260 077204 012746 000001 MOV #1,-(SP)
9261 077210 010600 MOV SP,RO
9262 077212 104417 TRAP C$PNTF
9263 077214 062706 000004 ADD #4,SP
9264 077220          PRINTF @PME24B ;CALIBRATION INFORMATION FOR BIPOLAR
9265 077220 012746 100474 MOV @PME24B,-(SP)
9266 077224 012746 000001 MOV #1,-(SP)
9267 077230 010600 MOV SP,RO
9268 077232 104417 TRAP C$PNTF

```

TEST 24: ANALOGUE INPUT (CALIBRATION ROUTINE)

```

9247 077234 062706 000004
077240 012746 100604 PRINTF @PME24C
077244 012746 000001
077250 010600
077252 104417
9248 077254 062706 000004
077260 012746 100675 PRINTF @PME24D
077264 012746 000001
077270 010600
077272 104417
9249 077274 062706 000004
077300 012746 100750 PRINTF @PME24E
077304 012746 000001
077310 010600
077312 104417
9250 077314 062706 000004
077320 000460 BR 20$
9251
9252 077322 10$: PRINTF @PME24A
077322 012746 100364
077326 012746 000001
077332 010600
077334 104417
9253 077336 062706 000004
077342 012746 101604 PRINTF @PME243
077346 012746 000001
077352 010600
077354 104417
9254 077356 062706 000004
077362 012746 101036 PRINTF @PME24F
077366 012746 000001
077372 010600
077374 104417
9255 077376 062706 000004
077402 012746 101145 PRINTF @PME24G
077406 012746 000001
077412 010600
077414 104417
9256 077416 062706 000004
077422 012746 101235 PRINTF @PME24H
077426 012746 000001
077432 010600
077434 104417
9257 077436 062706 000004
077442 012746 101307 PRINTF @PME24I
077446 012746 000001
077452 010600
077454 104417
077456 062706 000004
    
```

```

ADD @4,SP
;CALIBRATION INFORMATION FOR BIPOLAR
MOV @PME24C,-(SP)
MOV @1,-(SP)
MOV SP,RO
TRAP C$PNTF
ADD @4,SP
;CALIBRATION INFORMATION FOR BIPOLAR
MOV @PME24D,-(SP)
MOV @1,-(SP)
MOV SP,RO
TRAP C$PNTF
ADD @4,SP
;CALIBRATION INFORMATION FOR BIPOLAR
MOV @PME24E,-(SP)
MOV @1,-(SP)
MOV SP,RO
TRAP C$PNTF
ADD @4,SP
;AND SKIP UNIPOLAR MESSAGE
;CALIBRATION INFORMATION GENERAL
MOV @PME24A,-(SP)
MOV @1,-(SP)
MOV SP,RO
TRAP C$PNTF
ADD @4,SP
;PRINT UNIPOLAR MESSAGES
MOV @PME243,-(SP)
MOV @1,-(SP)
MOV SP,RO
TRAP C$PNTF
ADD @4,SP
;CALIBRATION INFORMATION FOR UNIPOLAR
MOV @PME24F,-(SP)
MOV @1,-(SP)
MOV SP,RO
TRAP C$PNTF
ADD @4,SP
;CALIBRATION INFORMATION FOR UNIPOLAR
MOV @PME24G,-(SP)
MOV @1,-(SP)
MOV SP,RO
TRAP C$PNTF
ADD @4,SP
;CALIBRATION INFORMATION FOR UNIPOLAR
MOV @PME24H,-(SP)
MOV @1,-(SP)
MOV SP,RO
TRAP C$PNTF
ADD @4,SP
;CALIBRATION INFORMATION FOR UNIPOLAR
MOV @PME24I,-(SP)
MOV @1,-(SP)
MOV SP,RO
TRAP C$PNTF
ADD @4,SP
    
```

TEST 24: ANALOGUE INPUT (CALIBRATION ROUTINE)

```

9258
9259 077462 005037 004134 20$: CLR MODE ;LOAD DACON INPUT (UNIPOLAR)
9260 077466 032777 000020 104204 BIT #20,@MOD ;IS UNIPOLAR RANGE SWITCHED
9261 077474 001403 BEQ 30$ ;BRANCH IF YES
9262 077476 012737 000001 004134 MOV #1,MODE ;LOAD DACON INPUT (BIPOLAR)
9263 077504 30$: GMANID MES242,ANS242,D,77400,0,15.,YES ;GET USED CHANNEL
    077504 104443 TRAP C$GMAN
    077506 000406 BR 10001$
    077510 100074 .WORD ANS242
    077512 000052 .WORD T$CODE
    077514 100246 .WORD MES242
    077516 077400 .WORD 77400
    077520 000000 .WORD T$LOLIM
    077522 000017 .WORD T$HILIM
    077524 10001$:
9264 077524 042777 077536 104152 BIC #77536,@CSA ;CLEAR CSA R/W BITS
9265 077532 053777 100074 104144 BIS ANS242,@CSA ;LOAD CHANNEL NUMBER
9266 077540 013704 100074 MOV ANS242,R4 ;AND GET CHANNEL NUMBER
9267 077544 032777 100000 104132 BIT #100000,@CSA ;IS THE CHANNEL AVAILABLE
9268 077552 001011 BNE 40$ ;BRANCH IF YES
9269 077554 PRINTF #PME241 ;PRINT CHA. NOT AVAILABLE
    077554 012746 101375 MOV #PME241,-(SP)
    077560 012746 000001 MOV #1,-(SP)
    077564 010600 MOV SP,R0
    077566 104417 TRAP C$PNTF
    077570 062706 000004 ADD #4,SP
9270 077574 000743 BR 30$ ;
9271
9272 077576 40$: GMANID MES243,ANS243,0,16,0,7,YES ;GET GAIN
    077576 104443 TRAP C$GMAN
    077600 000406 BR 10002$
    077602 100076 .WORD ANS243
    077604 000032 .WORD T$CODE
    077606 100306 .WORD MES243
    077610 000016 .WORD 16
    077612 000000 .WORD T$LOLIM
    077614 000007 .WORD T$HILIM
    077616 10002$:
9273 077616 053777 100076 104060 BIS ANS243,@CSA ;LOAD GAIN INTO CSA
9274 077624 006237 100076 ASR ANS243 ;CHANGE GAIN BIT POSITION
9275 077630 013737 100076 004136 MOV ANS243,GAIN ;LOAD DACON INPUT (GAIN)
9276 077636 006337 100076 ASL ANS243 ;CHANGE BACK GAIN BIT POSI.
9277
9278 077642 005777 104034 TST @DAT ;READ DAT FOR CLR DONE+ERR BIT IN CSA
9279 077646 017737 104032 004064 50$: MOV @CSA,BAD ;GET CSA CONTENTS
9280 077654 032737 000200 004064 BIT #200,BAD ;READY FOR CONVERSION (DONE CLEARED)
9281 077662 004737 012146 CALL INSERT ;SKIP BRANCH IF 'SFI' IS SET
9282 077666 001414 BEQ 60$ ;BRANCH IF YES
9283 077670 012737 100000 004062 MOV #100000,GOOD ;SET UP GOOD DATA
9284 077676 013737 100076 004062 MOV ANS243,GOOD ;SET UP GOOD DATA
9285 077704 050437 004062 BIS R4,GOOD ;...
9286 077710 ERRSOF T 2401,E2401,EER1 ;ERROR HANDLER
    077710 104457 TRAP C$ERSOF T
    077712 004541 .WORD 2401
    077714 101746 .WORD E2401
    077716 007342 .WORD EER1
9287 077720 005003 60$: CLR R3 ;CLEAR DONE TIMEOUT COUNTER

```

TEST 24: ANALOGUE INPUT (CALIBRATION ROUTINE)

```

9288 077722 052777 000001 103754      BIS      #1,@CSA      ;START A CONVERSION
9289 077730 017737 103750 004064 70$:  MOV      @CSA,BAD    ;GET CSA CONTENTS
9290 077736 032737 000200 004064      BIT      #200,BAD    ;CONVERSION DONE (DONE SET) ?
9291 077744 001013      BNE      80$        ;BRANCH IF YES
9292 077746 005303      DEC      R3         ;DECREMENT DONE WAIT COUNTER
9293 077750 004737 012146      CALL    INSERT      ;SKIP BRANCH IF 'SFI' IS SET
9294 077754 001365      BNE      70$        ;BRANCH IF NO TIMEOUT
9295 077756 052737 000200 004062      BIS      #200,GOOD   ;SET UP GOOD DATA
9296 077764      ERRSOFT 2402,E2402,EER1 ;ERROR HANDLER
          077764 104457      TRAP    C$ERSOFT
          077766 004542      .WORD  2402
          077770 002031      .WORD  E2402
          077772 007342      .WORD  EER1

9297
9298 077774 017737 103702 004062 80$:  MOV      @DAT,GOOD   ;GET DAT CONTENTS
9299 100002 013701 004062      MOV      GOOD,R1    ;LOAD DACON INPUT WITH DAT CONTENTS
9300 100006 004737 004512      CALL    DACON       ;DIGITAL TO ANALOGUE CONVERSION
9301 100012      PRINTF  #PME244,GOOD ;PRINT FOUND VALUES
          100012 013746 004062      MOV      GOOD,-(SP)
          100016 012746 101661      MOV      #PME244,-(SP)
          100022 012746 000002      MOV      #2,-(SP)
          100026 010600      MOV      SP,R0
          100030 104417      TRAP    C$PNTF
          100032 062706 000006      ADD     #6,SP

9302 100036 004737 015764      CALL    DECOUT      ;CALL DECIMAL OUT ROUTINE
9303 100042      PRINTF  #OUT24     ;PRINT 'MILLIVOLTS'
          100042 012746 101731      MOV      #OUT24,-(SP)
          100046 012746 000001      MOV      #1,-(SP)
          100052 010600      MOV      SP,R0
          100054 104417      TRAP    C$PNTF
          100056 062706 000004      ADD     #4,SP

9304 100062 000137 077646      JMP     50$        ;GO INTO LOOP
9305 100066      EXIT     TST
          100066 104432      TRAP    C$EXIT
          100070 002010      .WORD  L10061-

9306
9307 100072 000000      ANS241: .WORD  0    ;LOCATION FOR HELP ANSWER
9308 100074 000000      ANS242: .WORD  0    ;SAVE LOCATION FOR CHANNEL NUMBER
9309 100076 000000      ANS243: .WORD  0    ;SAVE LOCATION FOR GAIN VALUE
9310
9311      .NLIST  BEX
9312 100100      045     123     062  TSHD24: .ASCIZ  /#S2#ANALOGUE INPUT (CALIBR.) - SPECIFICALLY SELECTABLE.#N/
9313 100173      104     117     040  MES241: .ASCIZ  /DO YOU NEED CALIBRATION HELP INFORMATION ?/
9314 100246      125     123     105  MES242: .ASCIZ  /USED CHANNEL FOR VOLTAGE SOURCE/
9315 100306      107     101     111  MES243: .ASCIZ  /GAIN (TYPE 0-7 = GIAN 1.2.5.10.20.50.100.200)/
9316
9317 100364      045     116     045  PME24A: .ASCII  /#N#ACALIBRATION INFORMATION#N/
9318 100421      045     101     107      .ASCIZ  /#AGAIN 1 IS USED FOR FOLLOWING PROCEDURE#N/
9319
9320 100474      045     116     045  PME24B: .ASCII  /#N#N#AFOR BIPOLAR ADJUSTMENT#N#N/
9321 100534      045     101     101      .ASCIZ  /#AADJUST VOLTAGE SOURCE TO -9.9951 V#N/
9322 100604      045     101     101  PME24C: .ASCIZ  /#AADJUST OFFSET TRIMPOT UNTIL 0001 (-9.9951 V) APPEARS#N/
9323 100675      045     116     045  PME24D: .ASCIZ  /#N#N#AADJUST VOLTAGE SOURCE TO +9.9902 V#N/
9324 100750      045     101     101  PME24E: .ASCIZ  /#AADJUST GAIN TRIMPOT UNTIL 7776 (9.9902 V) APPEARS#N/
9325
9326 101036      045     116     045  PME24F: .ASCII  /#N#N#AFOR UNIPOLAR ADJUSTMENT#N#N/
9327 101077      045     101     101      .ASCIZ  /#AADJUST VOLTAGE SOURCE TO 0.0024 V#N/

```


TEST 24: ANALOGUE INPUT (CALIBRATION ROUTINE)

```

9328 101145 045 101 101 PME24G: .ASCIZ /#AADJUST OFFSET TRIMPOT UNTIL 0001 (0.0024 V) APPEARS#N/
9329 101235 045 116 045 PME24H: .ASCIZ /#N#N#AADJUST VOLTAGE SOURCE TO 9.9952 V#N/
9330 101307 045 101 101 PME24I: .ASCIZ /#AADJUST GAIN TRIMPOT UNTIL 7776 (9.9952 V) APPEARS#N/
9331
9332 101375 045 101 103 PME24J: .ASCIZ /#ACHANNEL NOT AVAILABLE, ENTER CORRECT CHANNEL OR #N/
9333 101461 045 101 106 .ASCIZ /#AFIND THE MISTAKE WITH OTHER TESTS.#N/
9334 101530 045 116 045 PME24K: .ASCIZ /#N#AMODULE IS SWITCHED TO BIPOLAR RANGE#N#N/
9335 101604 045 116 045 PME24L: .ASCIZ /#N#AMODULE IS SWITCHED TO UNIPOLAR RANGE#N#N/
9336 101661 045 116 045 PME24M: .ASCIZ /#N#AOCTAL VALUE :#06#A, EQUIVALENT TO :/
9337
9338 101731 045 101 040 OUT24: .ASCIZ /#A MILLIVOLT/
9339
9340 101746 103 123 101 E2401: .ASCIZ /CSA CONTENTS INCORRECT AFTER READING DATA REGISTER/
9341 102031 124 111 115 E2402: .ASCIZ \TIMEOUT - DONE NOT SET AFTER A/D START\
9342 .LIST BEX
9343 .EVEN
9344 102100 .ENDTST
102100
102100 104401

```

```

L10061: TRAP C#ETST

```

TEST 25: VISUAL LED TEST - SPECIFICALLY SELECTABLE.

```

9346          .SBTTL TEST 25: VISUAL LED TEST - SPECIFICALLY SELECTABLE.
9347          :.....
9348          : TEST 25 - VISUAL LED TEST - SPECIFICALLY SELECTABLE.
9349          :
9350          : THIS TEST IS A VISUAL TEST. IT FLASHES THE LED ON AND OFF ON
9351          : EVERY MODULE IN THE SYSTEM THAT IS FOUND BY THE AUTOMATIC
9352          : CONFIGURATION ROUTINE OR SELECTED IN THE HARDWARE QUESTIONS.
9353          :
9354          : MUX BOARDS ARE NOT INDEPENDENTLY ADDRESSABLE, SO THEIR LEDS ARE
9355          : NOT FLASHED.
9356          :
9357          : THE TEST RUNS UNTIL "CNTL C" IS TYPED.
9358          :.....
9359
9360 102102          BGNTST
9361 102102          004737 011714          CALL SELECT          ; CALL THE SELECT ROUTINE
9362 102106          010377          .WORD 10377          ; SPECIFIC TEST, ALL DEVICES
9363 102110          102236          TSMD25          ; TEST HEADER ADDRESS
9364 102112          103002          BCC 10$          ; BRANCH IF THE TEST IS SELECTED
9365 102114          104432          EXIT TST          ; OTHERWISE, EXIT THE TEST
9366 102116          000202          TRAP C$EXIT
9367 102120          005001          10$: CLR R1          ; START BY SWITCHING LEDS ON
9368
9369 102122          062701 000100          20$: ADD #100,R1          ; IF BIT 6 IS CLEAR, SET IT
9370 102126          042701 000200          BIC #200,R1          ; IF IT IS SET, CLEAR IT
9371
9372 102132          005037 002074          CLR L$LUN          ; START WITH UNIT 0
9373 102136          013703 002074          30$: MOV L$LUN,R3          ; FORM OFFSET FOR PARAMETER ADDRESS
9374 102142          006303          ASL R3          ;
9375 102144          016302 003774          MOV GPADD(R3),R2          ; GET PARAMETER ADDRESS IN R2
9376 102150          001006          BNE 35$          ; IF ADDRESS IS SET UP, BRANCH
9377 102152          013700 002074          GPHARD L$LUN,R2          ; ELSE, GET PARAM ADDRESS FROM DRS
9378 102152          013700 002074          MOV L$LUN,R0
9379 102156          104442          TRAP C$GPHRD
9380 102160          010002          MOV R0,R2
9381 102162          010263 003774          MOV R2,GPADD(R3)          ; AND SAVE IT IN THE TABLE
9382 102166          005712          35$: TST (R2)          ; MODE ADDRESS = 0 ?
9383 102170          001002          BNE 40$          ; IF NOT, BRANCH
9384 102172          004737 012214          JSR PC,CONFIG          ; ELSE DO AUTO CONFIGURATION FOR THIS UNIT
9385 102176          010172 000000          40$: MOV R1,@(R2)          ; SWITCH THE LED OVER
9386 102202          005237 002074          INC L$LUN          ; GO TO NEXT UNIT
9387 102206          023737 002074 002012          CMP L$LUN,L$UNITS          ; ALL CHANGED ?
9388 102214          002750          BLT 30$          ; IF NOT, SWITCH OVER THE NEXT
9389
9387 102216          012703 000024          MOV #20,R3          ; WAIT FOR 0.5 SECONDS
9388 102222          004737 011506          50$: JSR PC,WI25M          ; 25 MILLISECONDS WAIT ROUTINE
9389 102226          005303          DEC R3          ; * 20 = 0.5 SECONDS
9390 102230          001374          BNE 50$          ; GO ON IF 0.5 SEC. ARE OVER
9391 102232          104422          BREAK          ; ALLOW OPERATOR INPUT
9392 102234          000732          BR 20$          ; DO IT ALL AGAIN
9393
9394          .NLIST BEX
9395 102236          045 123 062 TSMD25:;.ASCIZ /#52#AVISUAL LED TEST - SPECIFICALLY SELECTABLE.#N/

```

TEST 25: VISUAL LED TEST - SPECIFICALLY SELECTABLE.

9396
9397
9398
9399 102320
102320
102320 104401

.LIST BEX
.EVEN
ENDTST

L10062: TRAP CSETST

E4

HARDWARE TESTS MACRO M1200 26-OCT-83 15:22 PAGE 94

SEQ 0250

TEST 25: VISUAL LED TEST - SPECIFICALLY SELECTABLE.

9407
9413
9419
9436 102322
9437

ENDMOD

TEST 25: VISUAL LED TEST - SPECIFICALLY SELECTABLE.

```

9441          .TITLE CLOSE SECTION
9452
9481
9482 102322          BGNMOD
9483
9484 102322          $PATCH::
9485 102322          .BLKW  500
9486
9493
9495 103522          .BLKB  400-<.E377>          ; SHIFT TO CORRECT FOR LSI BUG
9497 104000          LASTAD
                                     .EVEN
                                     .WORD T$FREE
                                     .WORD T$SIZE
          104000 104504
          104002 000240
          104004
9498 104004          L$LAST::
                                     ENDMOD

```

TEST 25: VISUAL LED TEST - SPECIFICALLY SELECTABLE.

9500
 9501
 9502
 9515
 9516 104004
 9517 000020
 9518
 9519
 9520
 9521

BGNSETUP 16.
 .REPT 16.
 BGNPTAB
 .WORD 0,0,0,0,0,0,0,0
 ENDPTAB
 .ENDR

104004 104034
 104006 000010
 104010
 104030
 104030 104060
 104032 000010
 104034
 104054
 104054 104104
 104056 000010
 104060
 104100
 104100 104130
 104102 000010
 104104
 104124
 104124 104154
 104126 000010
 104130
 104150
 104150 104200
 104152 000010
 104154
 104174
 104174 104224
 104176 000010
 104200
 104220
 104220 104250
 104222 000010
 104224
 104244
 104244 104274
 104246 000010
 104250
 104270
 104270 104320
 104272 000010
 104274
 104314
 104314 104344
 104316 000010
 104320
 104340
 104340 104370
 104342 000010
 104344

.WORD L10064
 .WORD L10065-./2-1
 L10063:
 L10065:
 .WORD L10066
 .WORD L10067-./2-1
 L10064:
 L10067:
 .WORD L10070
 .WORD L10071-./2-1
 L10066:
 L10071:
 .WORD L10072
 .WORD L10073-./2-1
 L10070:
 L10073:
 .WORD L10074
 .WORD L10075-./2-1
 L10072:
 L10075:
 .WORD L10076
 .WORD L10077-./2-1
 L10074:
 L10077:
 .WORD L10100
 .WORD L10101-./2-1
 L10076:
 L10101:
 .WORD L10102
 .WORD L10103-./2-1
 L10100:
 L10103:
 .WORD L10104
 .WORD L10105-./2-1
 L10102:
 L10105:
 .WORD L10106
 .WORD L10107-./2-1
 L10104:
 L10107:
 .WORD L10110
 .WORD L10111-./2-1
 L10106:
 L10111:
 .WORD L10112
 .WORD L10113-./2-1
 L10110:

TEST 25: VISUAL LED TEST - SPECIFICALLY SELECTABLE.

```

104364
104364 104414
104366 000010
104370
104410
104410 104440
104412 000010
104414
104434
104434 104464
104436 000010
104440
104460
104460 000000
104462 000010
104464
104504
9522 104504
9523 000001

```

.END ENDSETUP

```

L10113:
      .WORD L10114
      .WORD L10115-./2-1
L10112:
L10115:
      .WORD L10116
      .WORD L10117-./2-1
L10114:
L10117:
      .WORD L10120
      .WORD L10121-./2-1
L10116:
L10121:
      .WORD 0
      .WORD L10123-./2-1
L10120:
L10123:

```

SYMBOL TABLE

ACFLG	023650		BIT7	= 000200	G	C#DODU=	000051	DMPR	065332	EXTR21	065462
ADCON	014730	G	BIT8	= 000400	G	C#DRPT=	000024	DMP TAB	066340	E#END	= 002100
ADR	= 000020	G	BIT9	= 001000	G	C#DU	= 000053	DO	014457	E#LOAD=	000035
AI	014470		BOE	= 000400	G	C#EDIT=	000003	DROPD	026650	E1002	040762
ANS	004142	G	CALRET	017066	G	C#ERDF=	000055	DROPE D	004034	E101	027150
ANS1	062446		CARRFL	017172		C#ERHR=	000056	ECNT	003734	E1102	041465
ANS151	050656		CHAR	020454		C#ERRO=	000060	EERA	007634	E1103	041603
ANS152	050660		CHA23	072650		C#ERSF=	000054	EERB	007664	E1202	042660
ANS153	050662		CHKMAX	011342	G	C#ERSO=	000057	EERG	007724	E1302	043536
ANS161	054402		CLINT	017316	G	C#ESCA=	000010	EER1	007342	E1303	043622
ANS164	054404		CLKFLG	025074		C#ESEG=	000005	EER2	007400	E1402	045703
ANS2	063736		CNT25	011540	G	C#ESUB=	000003	EER3	007442	E1403	045764
ANS211	066330		CNT25M	011534	G	C#ETST=	000001	EER4	007474	E1404	046034
ANS212	066332		CNT500	011536	G	C#EXIT=	000032	EER5	007540	E1405	046103
ANS213	066334		CON	020010		C#GETB=	000026	EER6	007602	E1406	046163
ANS214	066336		COND	003714	G	C#GETW=	000027	EF.CON=	000036	E1408	046234
ANS221	071552		CONEX	013704		C#GMAN=	000043	EF.NEW=	000035	E1409	046315
ANS222	071554		CONFIG	012214	G	C#GPHR=	000042	EF.PWR=	000034	E1501	051726
ANS223	071556		CONMSK	004132	G	C#GPLO=	000030	EF.RES=	000037	E1502	051755
ANS224	071560		CONPRI	012600	G	C#GPRI=	000040	EF.STA=	000040	E1503	052030
ANS225	071562		CONT	022746		C#INIT=	000011	EMG1	007762	E1507	052063
ANS226	071564		CO1	012520		C#INLP=	000020	EMG2	010031	E1508	052134
ANS231	074752		CP1	013720		C#MANI=	000050	EMG3	010101	E1601	056016
ANS232	074754		CP1A	013771		C#MEM	= 000031	EMG4	010166	E1602	056134
ANS233	074756		CP1B	014043		C#MSG	= 000023	EMG5	010230	E1603	056244
ANS241	100072		CP1C	014146		C#OPEN=	000034	EMG6	010314	E1604	056342
ANS242	100074		CP10	014400		C#PNTB=	000014	END	023644	E1608	056371
ANS243	100076		CP10A	014415		C#PNTF=	000017	ERRBLK	007204	E1701	060356
AO	014501		CP11	014433		C#PNTS=	000016	ERRMSG	007202	E1702	060447
ASK	017320		CP12	014437		C#PNTX=	000015	ERRNBR	007200	E1703	060537
ASKADD	020234		CP2	014217		C#QIO	= 000377	ERRTYP	007176	E1704	060614
ASSEMB=	000010		CP3	014230		C#RDBU=	000007	EVL	= 000004	E1705	060714
BAD	004064	G	CP4	014237		C#REFG=	000047	EXQV1	027100	E1707	060772
BIPOL	050664		CP5	014251		C#RESE=	000033	EXQV10	040672	E1901	062535
BIT0	= 000001	G	CP6	014266		C#REVI=	000003	EXQV11	041374	E1902	062641
BIT00	= 000001	G	CP7	014274		C#RFLA=	000021	EXQV12	042566	E1903	062720
BIT01	= 000002	G	CP8	014356		C#RPT	= 000025	EXQV13	043444	E200	027605
BIT02	= 000004	G	CP9	014367		C#SEFG=	000046	EXQV14	045606	E2001	064034
BIT03	= 000010	G	CRLF	011542	G	C#SPRI=	000041	EXQV15	050652	E201	027703
BIT04	= 000020	G	CSA	003704	G	C#SVEC=	000037	EXQV16	054376	E202	027770
BIT05	= 000040	G	CSB	003706	G	C#TPRI=	000013	EXQV17	057450	E203	030035
BIT06	= 000100	G	CUROUT	054430		DACON	014512	EXQV18	061416	E2101	070074
BIT07	= 000200	G	CYCLE	004154	G	DAT	003702	EXQV19	062442	E2102	070164
BIT08	= 000400	G	CYCLLS	004156	G	DBOUNC	003720	EXQV2	027556	E2103	070255
BIT09	= 001000	G	CYCRAD	004160	G	DBTAB	004172	EXQV20	063732	E2104	070346
BIT1	= 000002	G	C#AU	= 000052		DECEX	016074	EXQV21	066324	E2107	070433
BIT10	= 002000	G	C#AUTO=	000061		DECIN	015204	EXQV22	071546	E2108	070521
BIT11	= 004000	G	C#BRK	= 000022		DECIN1	015615	EXQV23	074746	E2109	070603
BIT12	= 010000	G	C#BSEG=	000004		DECIN2	015653	EXQV3	030622	E2110	070650
BIT13	= 020000	G	C#BSUB=	000002		DECIN3	015734	EXQV4	031750	E2201	072430
BIT14	= 040000	G	C#CEFG=	000045		DECOUT	015764	EXQV5	033200	E2301	077021
BIT15	= 100000	G	C#CLCK=	000062		DEC01	016102	EXQV6	034520	E2401	101746
BIT2	= 000004	G	C#CLEA=	000012		DEC02	016106	EXQV7	035420	E2402	102031
BIT3	= 000010	G	C#CLOS=	000035		DEC03	016112	EXQV8	036562	E300	030652
BIT4	= 000020	G	C#CLP1=	000006		DFPTBL	002210	EXQV9	037612	E304	030747
BIT5	= 000040	G	C#CVEC=	000036		DI	014446	EXR	065552	E305	031037
BIT6	= 000100	G	C#DCLN=	000044		DIAGMC=	000000	EXTRI	050414	E306	031150

SYMBOL TABLE

L10071 104100
L10072 104130
L10073 104124
L10074 104154
L10075 104150
L10076 104200
L10077 104174
L10100 104224
L10101 104220
L10102 104250
L10103 104244
L10104 104274
L10105 104270
L10106 104320
L10107 104314
L10110 104344
L10111 104340
L10112 104370
L10113 104364
L10114 104414
L10115 104410
L10116 104440
L10117 104434
L10120 104464
L10121 104460
L10123 104504
MAFTOL 050746
MANIO 002234 G
MANTST 002232 G
MAN23 074544
MASCOM 010756
MASK 010754 G
MES151 051340
MES152 051415
MES153 051474
MES154 051571
MES155 051666
MES162 055305
MES163 055441
MES164 055521
MES165 055017 G
MES171 060035
MES172 060136
MES173 060247
MES211 067434
MES212 067452
MES213 067467
MES214 067550
MES221 071662
MES222 071700
MES223 071715
MES224 071767
MES225 072040
MES226 072121
MES231 075214
MES232 075243
MES233 075320

MES241 100173
MES242 100246
MES243 100306
MMES1 063020
MMES2 063042
MMES71 064104
MMES72 064214
MMES73 064234
MMES74 064254
MOD 003700 G
MODE 004134 G
MUXC 050666
MUXPAR 050336
MUXTOL 050766
NERRS 011444
NEWST 022656
NEXT 023160
NH = 000032
NO 022334
NOCLK 024606
NODEV 007250 G
NOUNIT 012516
NR1 015556
NR2 015560
NTEST 024056
NTESTS= 000031
NTEST1 024152
NXM 017266 G
NXMFLG 004060 G
NXTUUT 023106
ONEFIL= 000001
OTHCSA 003726 G
OTHCSB 003730 G
OTHDAT 003724 G
OTHMOD 003722 G
OTHVEC 003732 G
OUTDE1 004166 G
OUTDE2 004170 G
OUT1 056501
OUT15 051455
OUT2 056516
OUT231 077002
OUT232 076762
OUT24 101731
O\$APTS= 000001
O\$AU = 000001
O\$BGNR= 000001
O\$BGNS= 000001
O\$DU = 000001
O\$ERRT= 000001
O\$GNSW= 000001
O\$POIN= 000001
O\$SETU= 000001
PADD 004140 G
PAT1 002236 G
PAT2 002240 G
PLMI 056457

PME151 052212
PME152 052265
PME153 052337
PME154 052430
PME155 052467
PME16A 055213
PME161 054725
PME162 055055
PME163 055346
PME164 055603
PME165 055151
PME166 055666
PME167 055757
PME211 067571
PME212 067665
PME213 067747
PME214 070022
PME221 072162
PME222 072260
PME223 072332
PME23A 075402
PME23B 075530
PME23D 075626
PME23E 075715
PME23I 075753
PME232 076040
PME233 076121
PME234 076205
PME235 076275
PME236 076352
PME237 076421
PME238 076507
PME239 076576
PME24A 100364
PME24B 100474
PME24C 100604
PME24D 100675
PME24E 100750
PME24F 101036
PME24G 101145
PME24H 101235
PME24I 101307
PME24J 101375
PME242 101530
PME243 101604
PME244 101661
PNT = 001000 G
PRA 020654
PREX 020444
PRI = 002000 G
PRIO 003712 G
PRIO0 = 000000 G
PRIO1 = 000040 G
PRIO2 = 000100 G
PRIO3 = 000140 G
PRIO4 = 000200 G
PRIO5 = 000240 G

PRI06 = 000300 G
PRI07 = 000340 G
PRMST 076655
PROMT 017002
PR1 020746
PR2 020773
PR2A 021100
PR2B 021177
PR2C 021271
PR2D 021332
PR2E 021422
PR2F 021525
PR2G 021621
PR3 021715
PR3A 022022
PR4 022061
PR5 022112
PR6 022133
PR7 022234
PSAD 015006
PSDA 014564
PSEUL1 023550
PSUE8 036210
PS151 046750
PWRFL 023030
QVP 002242 G
QVR10 040644
QVR12 042540
QVR14 045560
QVR8 036534
RA 016572 G
RANDOM 016534 G
RB 016574 G
RDY 020612
READ 016326 G
RECON 017536
REDTIM 004164 G
REERM 016450
REGADD 010760 G
REGERR 010762 G
REGMSG 011247
REGTST 010402 G
REGTS1 010440 G
RERR1 011030
RERR2 011102
RERR3 011160
RESTR 022634
RTIME1 017174
RFLG 020462
RSAV 030102
RWMAK 031264
SAVCNT 025012
SAV8 036566
SECPA 063742
SELECT 011714 G
SETCLK 024222
SETEX 025072

SFI 004066 G
SFPTBL 002232 G
SLZERO 061422
SNUM 015562
STADD 012514 G
STARES 022644
START 022350
STAT 020260
STFLG 020450
SVCGBL = 000000
SVCINS = 000001
SVCSUB = 000001
SVCTAG = 000001
SVCTST = 000001
S\$LSYM = 010000
TABOFF 054524
TAB23 074760
TADS 020464
TIMIN 063744
TIMMSG 025076
TITLE 017642
TKB = 177562
TKS = 177560
TM15 051100
TM16 054623
TM17 057544
TM18 061515
TNA 012074
TNUM 020650
TLOF 050674
TOLRA 050672
TOL1 076707
TPB = 177566
TPS = 177564
TSHD1 027116 G
TSHD10 040676 G
TSHD11 041400 G
TSHD12 042572 G
TSHD13 043450 G
TSHD14 045612 G
TSHD15 051006 G
TSHD16 054530 G
TSHD17 057454 G
TSHD18 061424 G
TSHD19 062450 G
TSHD2 027562 G
TSHD20 063746 G
TSHD21 067340 G
TSHD22 071566 G
TSHD23 075120 G
TSHD24 100100 G
TSHD25 102236 G
TSHD3 030626 G
TSHD4 031760 G
TSHD5 033204 G
TSHD6 034524 G
TSHD7 035424 G

SYMBOL TABLE

TSHD8	036570	G	T\$PTAB=	010122	T\$\$PRO=	010023	T24	077110	G	WMES1	064342		
TSHD9	037616	G	T\$PTHV=	000020	T\$\$PTA=	010122	T25	102102	G	WMES2	064421		
TSTEN1	027106		T\$PTNU=	000020	T\$\$RPT=	010022	T3	030144	G	WME151	051202		
TSTFLG	004054	G	T\$SAVL=	177777	T\$\$SEG=	010004	T4	031326	G	WME152	051304		
TSUFLG	004056	G	T\$SEGL=	177777	T\$\$SOF=	010003	T5	032414	G	WME171	057646		
TT	020546		T\$SEKO=	010004	T\$\$SRV=	010032	T6	033674	G	WME172	057747		
TTINT	024740		T\$SIZE=	000240	T\$\$SW =	010001	T7	035154	G	WME181	061617		
T\$ARGC=	000001		T\$SUBN=	000000	T\$\$TES=	010062	T8	035720	G	WME182	061744		
T\$CODE=	000032		T\$TAGL=	177777	T1	026714	G	T9	037076	G	WME183	062030	
T\$ERRN=	004542		T\$TAGN=	010124	T10	040216	G	UAM	=	000200	G	WME184	061712
T\$EXCP=	000000		T\$TEMP=	000000	T11	041076	G	USCLOK	024542		WRDY	011570	G
T\$FLAG=	000040		T\$TEST=	000031	T12	041674	G	VALCNT	054526		WRDY1	011620	
T\$FREE=	104504		T\$TSTM=	177777	T13	042776	G	VBPTAB	004776	G	WRERM	016242	
T\$GMAN=	000000		T\$TSTS=	000001	T14	043702	G	VEC	003710	G	WRITE	016122	G
T\$HILI=	000007		T\$\$AU =	010030	T15	046404	G	VOLOUT	054406		WT25	011522	G
T\$LAST=	000001		T\$\$AUT=	010025	T16	052510	G	VOLPA	053744		WT25M	011506	G
T\$LOLI=	000000		T\$\$CLE=	010026	T17	056536	G	VOLTE	050670		WT500	011514	G
T\$LSYM=	010000		T\$\$DAT=	010123	T18	061070	G	VOLVA	050676		X\$ALWA=	000000	
T\$LTND=	000031		T\$\$DU =	010027	T19	062124	G	VUPTAB	004176	G	X\$FALS=	000040	
T\$NEST=	177777		T\$\$HAR=	010002	T2	027220	G	WAIT	011526		X\$OFFS=	000400	
T\$NSO =	000000		T\$\$HW =	010000	T20	063254	G	WARN	023652		X\$TRUE=	000020	
T\$NS1 =	000001		T\$\$INI=	010024	T21	064500	G	WARN1	023756		YES	022330	
T\$NS2 =	000003		T\$\$MSG=	010015	T22	070704	G	WFLG	011616		\$PATCH	102322	G
T\$PCNT=	000000		T\$\$PC =	000020	T23	072532	G						

. ABS. 104504 000
000000 001
ERRORS DETECTED: 0

VIRTUAL MEMORY USED: 31296 WORDS (123 PAGES)
DYNAMIC MEMORY: 17890 WORDS (68 PAGES)
ELAPSED TIME: 00:11:46
ZIXVAO.BIN,ZIXVAO.SEQ/CR/-SP=[50,200]SVC/ML,[7,116]ZIXVAO.SRC