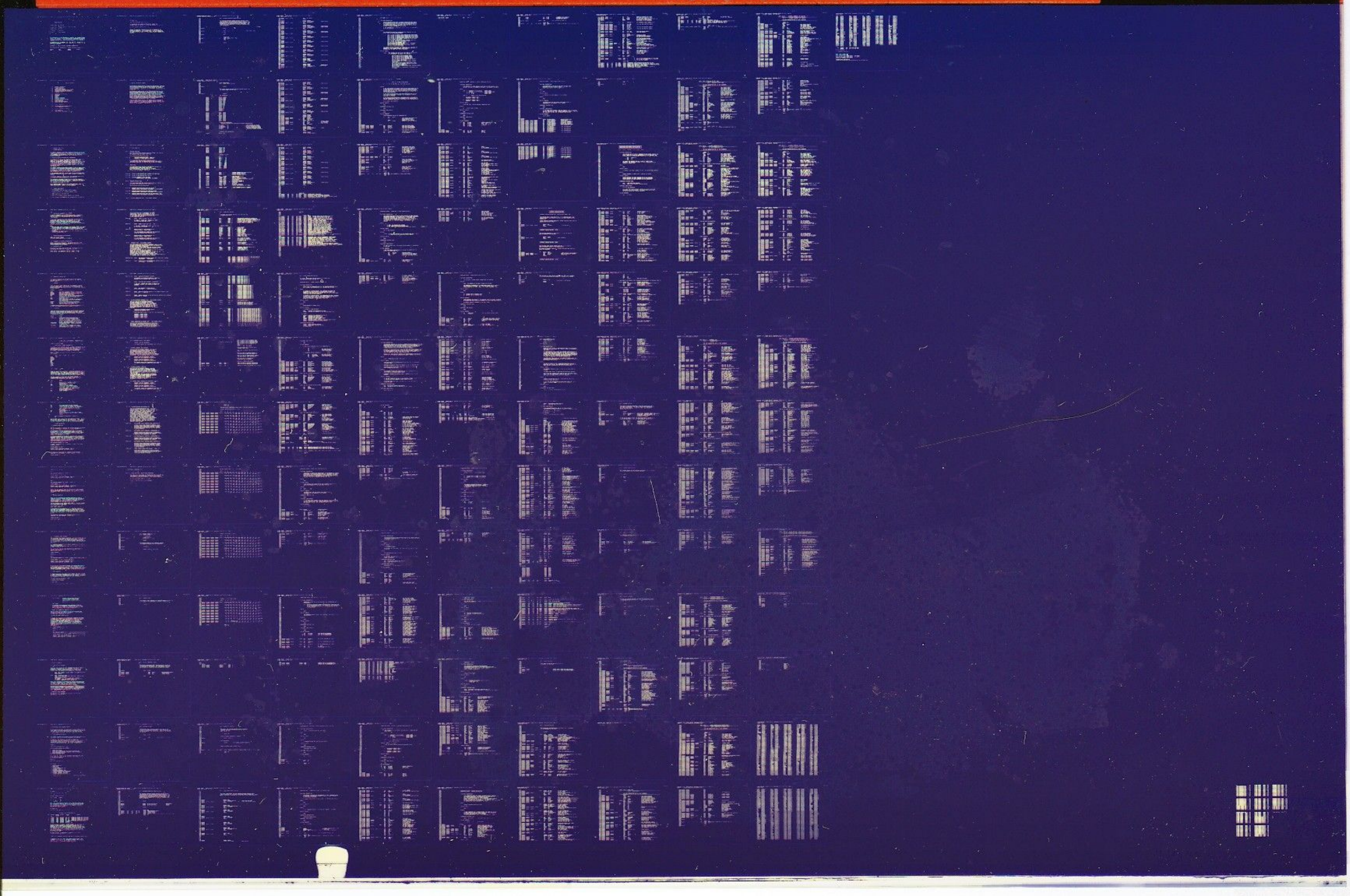
IOV11-O IOV11-O 5 CHA CNT DIA
CZIOVAO

AH-T977A-MC
FICHE 01 OF 01

APR 1985
digital
Made In USA



.REM &

IDENTIFICATION

Product Code: AC-T976A-MC

Product Name: CZIDVAO IDV11-D 5 CHA. CNT DIAG

Product Date: SEP 1984

Maintainer: CSS Munich

Authors: Peter Seebach

TABLE OF CONTENTS

1.0 General information

1.1 Program abstract

The CZIDV?? diagnostic provides a series of tests to verify the integrity and functionality of the IDV11-D five channel counter. 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 five channel counter test if a counter is found . this feature has special significance when the XXDP+ SETUP utility is used (see section 2.9).

The "PRINT" command can be used to obtain a list of test titles, a printout of the IAV/IDV11 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 proceding 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. IDV11-D five channel counter module to be checked.

e. Test connector (2G-M008A-00) and voltage source (21-24V DC)

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:
A) Once the XXDP+ media is booted, the console "BREAK" key should not be pressed as it may cause error messages to be printed.
B) I/O Page addresses from 160000 to 173776 are used by the KXT11-A2 ODT prom, so the first IAV/IDV11 address must be 174000 or higher.
C) FALCON dose not support vectors over 374. Therefore you can't use IAV/IDV11 vectors over 374.

1.4 Related documents and standards

XXDP+ User manual (CHQUS)
IAV/IDV11 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 vary with the CPU type. The following times are typical on a PDP-11/23 + system for one unit:

quick verify = no ,execution time for 1 pass = 30 sec.
quick verify = yes,execution time for 1 pass = 9 sec.

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 "ddddd".

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:DDDDD	execute ddddd passes (ddddd = 1 to 64000)
/FLAGS:FLGS	set specified flags. flags are described in section 2.3.
/EOP:DDDDD	report end of pass message after every ddddd passes only. (ddddd = 1 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)
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 (0) 171400 ?

In reply, you should enter an address in octal in the range 160000 to 177770.

VECTOR ADDRESS (0) 400 ?

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

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 (O) 171400 ? <cr>
                                  ----
VECTOR ADDRESS (O) 400 ? <cr>
                        ----
PRIORITY LEVEL (FOR OUTPUT MODULES, TYPE "O") (O) 4 ? 0 <cr>
(FOR LSI WITH FIXED PRI. USE LEVEL 4 ONLY)                -----


UNIT 1

MODE REGISTER ADDRESS (O) 171000 ? 171410 <cr>
                                  -----------
VECTOR ADDRESS (O) 0 ? 410 <cr>
                      --------
PRIORITY LEVEL (FOR OUTPUT MODULES, TYPE RETURN) (O) 0 ? 4 <cr>
(FOR LSI WITH FIXED PRI. USE LEVEL 4 ONLY)                 ---- -
```

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".

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.

```
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>
                      -----
QUICK VERIFY MODE (L) N ? <cr>
                         -- -
```

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 IDV-11 D modules.
You could answer the hardware questions for each of the four units as shown in section 2.4.

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 (0) 171400 ? 171400,171410,171420,171430 <cr>
                                   --------------------------------
VECTOR ADDRESS (0) 400 ? 400,410,420,430<cr>
                         ------------------
PRIORITY LEVEL (FOR OUTPUT MODULES, TYPE "0") (0) 4 ? <cr>
(FOR LSI WITH FIXED PRI. USE LEVEL 4 ONLY)             ----
```

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 IDV11-D modules at addresses 174000, 174010, 174020 and 174030 are set up with vectors 400,410,420 and 430. All have priority level 4

2.7 Clock questions

If there is no line time clock on the system, the user is asked to type 2 characters 6 seconds apart on the console. This should be done as accurately as possible since the interval is used by the diagnostic to calculate values for device timeouts.

2.8 Quick start-up procedure (XXDP+)

To start-up this program:

1. Boot XXDP+

2. Give the date and answer XXDP + questions

3. Type "R ZIDV??". (Normally the revision and patch level are typed insted of the question marks. The form shown here causes the latest version to be run.)

4. Type "START"

5. For standard configurations using addresses 171400 to 171770 answer the "CHANGE HW" question with "N". To test specific devices or those at non-standard addresses answer "Y" and answer all of the hardware questions.

6. Answer the "CHANGE SW" question with "N"

When you follow this procedure you will be using only the defaults for flags and software parameters. These defaults are described in sections 2.3 and 2.5.

2.9 Using the XXDP+ SETUP utility

To enable the diagnostic to automatically establish the IXV11 configuration in the field, the hardware p table is preset for 16 units, each with a mode address of 0. If the XXDP+ SETUP utility is used to preset the p table for a particular configuration, then the diagnostic will not automatically establish the configuration at startup. In this case, the diagnostic can be made to do automatic configuration by answering the hardware questions to give 16 units with mode addresses of 0.

```
EG. CHANGE HARDWARE (L) ? Y <cr>
                          ------
    #UNITS (D) ? 16 <cr>
                 -------
    UNIT 0

    MODE REGISTER ADDRESS (O) 0 ? 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0 <cr>
                                  ----------------------------------
    VECTOR ADDRESS (FOR OUTPUT MODULES, TYPE "0") (O) 0 ? 0 <cr>
                                                          ------
    PRIORITY LEVEL (FOR OUTPUT MODULES, TYPE "0") (O) 4 ? 0 <cr>
    (FOR LSI WITH FIXED PRI. USE LEVEL 4 ONLY)            -----
```

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  AM 9513 SUBREGISTER TEST
  5  INTERRUPT TEST
  6  REFERENCE FREQUENCY TEST
  7  SIGNAL GENERATION ON COUNTER INPUT
  8  EXTERNAL LOOPBACK, NOT ISOLATED
  9  EXTERNAL LOOPBACK, ISOLATED
 10  UP/DOWN COUNTING APPLICATION TEST

 11  VISUAL LED TEST                    - SPECIFICALLY SELECTABLE
```

```
DR> PRINT <cr>
    ---------
TYPE T,R,C,S OR HELP (S) T ? S <cr>
                           -- -

IDV/IAV11 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/IAV11 MODULE CONFIGURATION.

UNIT  ADDRESS  VECTOR  ID/MODE    MODULE              COMMEND
               ASSUMED             TYPE

  0   171000    NONE   060/000   DIG. OUT   CANNOT BE TESTED WITH THIS DIAG
  1   171410     410   030/002   DIG. IN    CANNOT BE TESTED WITH THIS DIAG
  2   171010    NONE   260/000   AN.  OUT   CANNOT BE TESTED WITH THIS DIAG
  3   171420     420   100/000   AN.  IN    CANNOT BE TESTED WITH THIS DIAG
  4   171700    NONE   320/000   *** CANNOT BE TESTED WITH THIS DIAG ***
  5   171710    NONE   UNKNOWN   UNKNOWN
  6   171400     400   300/000   FIVE CHA. CNT
```

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.

```
---------------------- ---------------------------------------- -
* NOTE: This diagnostic can only test the five channel counter module
------------ -   ---------------------------------------- -------
```

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.

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 3 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 IDV11-D tests if a counter was found. 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

Test 1 - Register NXM Test.

This test checks that accessing the device MODE, CSR, CCR and INR registers 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 MOD AND INR registers are correctly set or reset after a bus reset.
In the MOD register, only the LED bit is tested.
NOTE: The CSR and the CCR register can not be cleared by init so they will not tested in this test.

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 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

TEST 4 - AM 9513 SUBREGISTER TEST - FIVE CHANNEL COUNTER

THIS IS THE FIRST TEST TO REFERENCE THE AM9513
SUBREGISTERS. FOR ALL FIVE CHANNELS IT TESTS
THAT IN TOGGLE MODE THE OUTPUT CAN BE SET AND CLEARED.
IT THEN TESTS THAT THE LOAD AND HOLD REGISTERS
CAN BE LOADED AND READ OUT.

ERROR 401, 403 ERROR - UNABLE TO CLEAR 'OUT' BIT , CHA.: X

IN TOGGLE A 'CLEAR TOUT' COMMAND WAS LOADED.
THE STATE WAS READ BACK AS SET.

ERROR 402 ERROR - UNABLE TO SET 'OUT' BIT , CHA.: X

A 'SET TOUT' COMMAND WAS LOADED, AND THE STATE
WAS READ BACK AS CLEARED.

ERROR 404 ERROR WRITING TO LOAD REGISTER OF CHA.: X
WRITTEN: XXXXXX, READ: XXXXXX

THE LOAD REGISTER WAS LOADED WITH THE SPECIFIED
VALUE, BUT ANOTHER VALUE WAS READ BACK.

ERROR 405 ERROR WRITING TO HOLD REGISTER OF CHA.: X
WRITTEN: XXXXXX, READ: XXXXXX

THE HOLD REGISTER WAS LOADED WITH THE SPECIFIED
VALUE, BUT ANOTHER VALUE WAS READ BACK.

TEST 5 - INTERRUPT TEST - FIVE CHANNEL COUNTER

THIS TEST CHECKS THAT THE INTERNAL INTERRUPT LOGIC IS ABLE
TO CAUSE AN INTERRUPT USING THE VECTOR AND THE PRIORITY LEVEL
SELECTED IN THE START UP QUESTION.
AT FIRST WE SELECT TOUT TOGGEL MODE FOR THE FIRST CHANNEL.
THEN WE CLEAR TOUT AND SET THE INTERRUPT ENABALE BIT FOR
THE SELECTED CHANNEL.
NOW WE TEST THAT THE 'SET TOUT' COMMAND WILL SET THE INTERRUPT
REQUEST BIT IN THE INTERRUPT REGISTER.
AFTER THAT WE SET THE MASTER ENABLE BIT TO LOOK THAT THE
INTERRUPT WILL ALSO CAUSE AN INTERRUPT FOR THE
SELECTED CHANNEL AND AT THE CORRECT PRIORITY LEVEL.
NOW WE CHECK THAT THE INTERRUPT HAS CLEARED THE MASTER ENABLE BIT.
THE SEQUENCE ABOVE IS REPEATED FOR ALL CHANNELS.

ERROR 501,502 ERROR - INR REGISTER INCORRECT AFTER LOAD 'CLEAR TOUT' COMMAND
GOOD:NNNNNN BAD:NNNNNN

ERROR 503 ERROR - IR BIT IN THE INR REGISTER NOT RESETED AFTER BIT CLEAR
GOOD:NNNNNN BAD:NNNNNN

ERROR 504,505 ERROR - UNEXPECTED INTERRUPT DETECTED

AN INTERRUPT WAS DETECTED BEFORE ALL CONDITIONS
REQUIRED FOR AN INTERRUPT WERE SATISFIED.

ERROR 506 ERROR - EXPECTING ONE INTERRUPT, ENCOUNTERED :XXX

A 'SET TOUT' COMMAND WAS LOADED TO PRODUCE ONE
INTERRUPT, HOWEVER, THEN INDICATED NUMBER WAS
DETECTED.

ERROR 507 ERROR - BAD INTERRUPT VECTOR DETECTED
SET UP VECTOR:NNN FOUND VECTOR:NNN

ERROR 508 ERROR - INTERRUPT DID NOT OCCURED AT THE CORRECT PRIORITY LEVEL
SET UP:NNN FOUND:NNN

ERROR 509 ERROR - MASTER CLEAR BIT IN INR REGISTER NOT RESETED AFTER INTERRUPT
INR CONTENS IS : NNNNNN

TEST 6 - AM 9513 REFERENCE FREQUENCY TEST - FIVE CHANNEL COUNTER

THIS IS A TEST OF THE REFERENCE FREQUENCY.
THE 5MHZ FREQUENCY IS SELECTED AND THE LOAD
REGISTER IS LOADED WITH 10000. ALL FIVE COUNTERS
ARE LOADED AND ARMED FOR THE DURATION OF THREE
NOP INSTRUCTIONS. A TEST IS MADE THAT ALL COUNTERS
HAVE A VALUE DIFFERENT THAT 10000 AND THAT THE
DIFFERENCE BETWEEN ALL FIVE COUNTERS IS NOT GREATER
THAN TWO.

ERROR 601 UNEXPECTED DIFFERENCE BETWEEN COUNTER
SHOULD NOT BE GREATER THAN 2

COUNT DOWN FROM 10,000 AT 5 MHZ FOR 3 NOP INSTRUCTIONS

COUNTER 1 VALUE: XXXXXX
COUNTER 2 VALUE: XXXXXX
COUNTER 3 VALUE: XXXXXX
COUNTER 4 VALUE: XXXXXX
COUNTER 5 VALUE: XXXXXX

TEST 7 - SIGNAL GENERATION ON COUNTER INPUT - FIVE CHANNEL COUNTER

THIS IS A TEST OF SIGNAL GENERATION ON TIMER INPUT.
THE INTERNAL LOOPBACK MODE IS SELECTED AND THE OUTPUT
FROM CHANNEL N-1 IS USED AS INPUT TO THE COUNTER.
THE COUNTER IS ALLOWED TO RUN FOR FIVE NOP INSTRUCTIONS.
A TEST IS THEN MADE THAT THE COUNTER INCREMENTED.

THIS IS REPEATED FOR ALL FIVE COUNTER.

ERORO 701 COUNTRE X DID NOT INXCREMENT

THE COUNTER SHOULD HAVE INCREMENTED AWAY FROM ZERO.

TEST 8 - EXTERNAL LOOPBACK, NOT ISOLATED

THIS IS A TEST OF EXTERNAL LOOPBACK, NOT ISOLATED.
THE FREQUENCY OUTPUT CONTROL REGISTER AND THE
5MHZ LOOPBACK FREQUENCY ARE SELECTED. THE SOURCE
AND GATE OUTPUTS ARE THEN USED AS INPUT TO THE
COUNTER. THE COUNTER IS ALLOWED TO RUN FOR THREE
NOP INSTRUCTIONS, AND A TEST IS MADE THAT THE COUNTER
INCREMENTED. THIS IS REPEATED FOR ALL FIVE COUNTERS.

ERROR 801, COUNTER X DID NOT INCREMENT, SOURCE X
ENSURE THAT TEST CONNECTOR IS INSTALLED

802 COUNTER X DID NOT INCREMENT, GATE X
ENSURE THAT TEST CONNECTOR IS INSTALLED

THE COUNTER SHOULD HAVE INCREMENTED AWAY FROM ZERO.
THE INPUT WAS EITHER A SOURCE OR A GATE SIGNAL.

TEST 9 - EXTERNAL LOOPBACK, ISOLATED

THIS IS A TEST OF EXTERNAL LOOPBACK, ISOLATED.
THE FREQUENCY OUTPUT FOUT IS SWITCHED OFF THEN
COUNTER 1 IS SET UP FOR 100 KHZ SIGNAL GENERATION ON
OUT1. OUT1 IS LOOPBACK TO SOURCE 1•2 OF COUNTER 2.
TEHN THE COUNTER IS ALLOWED TO RUN FOR 100 PASSES OF A
THREE NOP LOOP.
A TEST IS THEN MADE THAT THE COUNTER INCREMENTED.
THIS SEQUENCE ABOVE IS REPEATED WITH ALL COUNTERS.
COUNTER 2 IS USED FOR SIGNAL GENERATION AND OUT2
IS LOOPBACK TO SOURCE 3•4 OF COUNTER 3.
COUNTER 3 IS USED FOR SIGNAL GENERATION AND OUT3
IS LOOPBACK TO SOURCE 5 • GATE 1.
E.T.C.

THE LOOPBACK SIGNAL FREQUENCY IS APPOXIMATLY 100 kHz.

ERROR 901, COUNTER X DID NOT INCREMENT, SOURCE X
ENSURE THAT TEST CONNECTOR IS INSTALLED

902 COUNTER X DID NOT INCREMENT, GATE X
ENSURE THAT TEST CONNECTOR IS INSTALLED

THE COUNTER SHOULD HAVE INCREMENTED AWAY FROM ZERO.
THE INPUT WAS EITHER A SOURCE OR A GATE SIGNAL.

TEST 10 - UP/DOWN COUNTING APPLICATION TEST

THIS IS AN UP/DOWN APPLICATION TEST. THIS TEST USES THE OUTPUTS FROM TOUT N= 3 AND 4 AS INPUT TO COUNTERS 1 AND 2. WHEN TOUT N=4 SETS AND TOUT N=3 IS SET, THEN COUNTER 1 SHOULD INCREMENT. THE SIGNAL TOUT N=3 SETS WHILE TOUT N=4 IS RESET, AND COUNTER 2 SHOULD NOT INCREMENT. THIS SYNCHRONOUS SETTING AND CLEARING OF TOUT N=3 AND 4,' OUT OF PHASE BY 90 DEGREES, IS REPEATED 10000 TIMES. AFTER THIS LOOP COUNTER 1 MUST HAVE A VALUE OF 10000 AND COUNTER 2 MUST HAVE A VALUE OF ZERO. THE SAME LOOP IS THEN PERFORMED WITH TOUT N=3,4 OUT OF PHASE BY 90 DEGREES IN THE OTHER DIRECTION. FOR THIS TEST COUNTER 2 MUST HAVE A VALUE OF 10000 AND COUNTER 1 MUST CONTAIN A VALUE OF ZERO.

ERROR 1001 BAD COUNTER VALUE, EXPECTED 10000, DETECTED XXX

OUTPUTS 3 AND 4 WERE TOGGLED TO PRODUCE A COUNT OF 10000. THESE SIGNALS WERE INPUTED TO COUNTER 1 WHICH WAS SET TO THE ACTIVE HIGH MODE. A COUNT OF 10000 SHOULD HAVE RESULTED, BUT THE INDICATED VALUE WAS DETECTED.

ERROR 1002 BAD COUNTER VALUE, EXPECTED 0, ENCOUNTERED XXX

OUTPUTS 3 AND 4 WERE USED AS INPUT TO COUNTER 2. COUNTER 2 WAS SET TO ACTIVE LOW, AND THESE SIGNALS SHOULD NOT HAVE INCREMENTED THE COUNTER.

ERROR 1003 BAD COUNTER VALUE, EXPECTED 0, ENCOUNTERED XXX

OUTPUTS 3 AND 4 WERE USED AS INPUT TO COUNTER 1. COUNTER 1 WAS SET TO ACTIVE LOW, AND THESE SIGNALS SHOULD NOT HAVE INCREMENTED THE COUNTER.

ERROR 1004 BAD COUNTER VALUE, EXPECTED 10000, DETECTED XXX

OUTPUTS 3 AND 4 WERE TOGGLED TO PRODUCE A COUNT OF 10000. THESE SIGNALS WERE INPUTED TO COUNTER 2 WHICH WAS SET TO THE ACTIVE HIGH MODE. A COUNT OF 10000 SHOULD HAVE RESULTED, BUT THE INDICATED VALUE WAS DETECTED.

ERROR 1005 ERROR LOADING TOGGLE, WRITTEN XXX, READ: XXX

EVERY TIME A TOGGLE OUTPUT IS SET OF CLEARED, THE SIGNAL IS READ BACK AND TESTED FOR THE CORRECT STATE. IF THE EXPECTED STATE IS NOT DETECTED, THIS ERROR MESSAGE IS PRODUCED.

6.1 SECIFICALLY SELECTABLE TEST

Test 11 - Visual LED Test - Specifically Selectable.

This test is a visual test. It fleshes 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.

The test runs until "CNTL C" is typed.

Errors : This test has no error messages.

```
                                .TITLE PROGRAM HEADER AND TABLES
                                .SBTTL  PROGRAM HEADER


                                        .ENABL  ABS,AMA
                                .       .       2000

                                        BGNMOD

                               ;++
                               ; THE PROGRAM HEADER IS THE INTERFACE BETWEEN
                               ; THE DIAGNOSTIC PROGRAM AND THE SUPERVISOR.
                               ;--

                                        POINTER ALL


                                        HEADER  CZIDV,A,0,150,0,340
```

```
                                .SBTTL  DISPATCH TABLE

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

002122                                  DISPATCH 11.
```

```
                                .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.
                                ;--

002152                                  BGNHW   DFPTBL

002154  171400                                 .WORD   171400          ; MODE REGISTER ADDRESS
002156  000400                                 .WORD   400             ; VECTOR ADDRESS
002160  000200                                 .WORD   PRI04           ; PRIORITY LEVEL

002162                                  ENDHW
```

```
                                .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.
                                ;--

002162                                  BGNSW   SFPTBL

002164  000000                QVP::               .WORD   0       ; QUICK VERIFY ? (0 = NO)

002166                                  ENDSW
```

```
                                .SBTTL  HARDWARE PARAMETER CODING SECTION

                                ;++
                                ; THE HARDWARE PARAMETER CODING SECTION CONTAINS MACROS
                                ; THAT ARE USED BY THE SUPERVISOR TO BUILD P-TABLES.  THE
                                ; MACROS ARE NOT EXECUTED AS MACHINE INSTRUCTIONS BUT ARE
                                ; INTERPRETED BY THE SUPERVISOR AS DATA STRUCTURES.  THE
                                ; MACROS ALLOW THE SUPERVISOR TO ESTABLISH COMMUNICATIONS
                                ; WITH THE OPERATOR.
                                ;--

002166                                  BGNHRD

002170                                  GPRMA   G1,0,0,0,177776,YES             ; MODE ADDRESS
002200                                  GPRMA   G2,2,0,0,770,YES                ; VECTOR
002210                                  GPRMD   G3,4,0,340,0,6,YES              ; PRIORITY

002222                          10$:    ENDHRD

                                        .NLIST  BEX
002222  115  117  104  G1:     .ASCIZ  /MODE REGISTER ADDRESS/
002250  126  105  103  G2:     .ASCIZ  /VECTOR ADDRESS /
002270  120  122  111  G3:     .ASCII  /PRIORITY LEVEL /
002307  012  015  050          .ASCIZ  <12><15>/(FOR LSI WITH FIXED PRI. USE LEVEL 4 ONLY)/
                                        .LIST   BEX
                                        .EVEN
```

```
                                .SBTTL  SOFTWARE PARAMETER CODING SECTION

                                ;++
                                ; THE SOFTWARE PARAMETER CODING SECTION CONTAINS MACROS
                                ; THAT ARE USED BY THE SUPERVISOR TO BUILD P-TABLES.  THE
                                ; MACROS ARE NOT EXECUTED AS MACHINE INSTRUCTIONS BUT ARE
                                ; INTERPRETED BY THE SUPERVISOR AS DATA STRUCTURES.  THE
                                ; MACROS ALLOW THE SUPERVISOR TO ESTABLISH COMMUNICATIONS
                                ; WITH THE OPERATOR.
                                ;--

002364                                  BGNSFT

                                10$:    GPRML   G16,0,-1,YES                    ; QUICK VERIFY MODE ?
                                        .EVEN

002374                                  ENDSFT


                                        .NLIST  BEX
002374  121  125  111  G16:     .ASCIZ  /QUICK VERIFY MODE/
                                        .LIST   BEX
                                        .EVEN


002416                          ENDMOD
```

```
                                .TITLE GLOBAL AREAS
                                .SBTTL  GLOBAL EQUATES SECTION


002416                                  BGNMOD

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


002416                                  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
                                ;
001000                          BIT9==  BIT09
000400                          BIT8==  BIT08
000200                          BIT7==  BIT07
000100                          BIT6==  BIT06
000040                          BIT5==  BIT05
000020                          BIT4==  BIT04
000010                          BIT3==  BIT03
000004                          BIT2==  BIT02
000002                          BIT1==  BIT01
000001                          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
```

```
                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
1349
1350
1351                                    ;
1352                                    ;
1353            000002                  MREA==  2               ;COUNTER COMMAND AND STATUS REGISTER
1354            000004                  MREB==  4               ;COUNTER CONTROL REGISTER
1355            000006                  MREC==  6               ;COUNTER INTERRUPT REGISTER
1356
1357            177400                  B==     177400          ;HIGH BITS MUST BE SET
1358            177500                  C$LOA== 100!B           ;LOAD COUNTER
1359            177440                  C$ARM==  40!B           ;ARM ALL SELECTED COUNTERS
1360            177600                  C$DAS== 200!B           ;SAVE AND DISARM ALL SELECTED COUNTERS
1361            177777                  C$MAR== 377!B           ;MASTER RESET
1362            177750                  C$IN1== 350!B           ;INITIALIZE COMMAND #1
1363            177757                  C$IN2== 357!B           ;INITIALIZE COMMAND #2
1364            177740                  C$CTN== 340!B           ;CLEAR TOUT
1365            177750                  C$STN== 350!B           ;SET TOUT
1366            177700                  C$DAC== 300!B           ;DISARM ALL COUNTERS
1367            000042                  M$TOG==  42             ;COUNTER MODE REGISTER, OUT TOGGLE
1368
1369            171000                  IXSTA== 171000          ; FIRST STANDARD IDV/IAV-11 ADDRESS
1370            171770                  IXEND== 171770          ; LAST STANDARD IDV/IAV-11 ADDRESS
1371
```

GLOBAL DATA SECTION

```
                                        .SBTTL  GLOBAL DATA SECTION

                                        ;++
                                        ; THE GLOBAL DATA SECTION CONTAINS DATA THAT ARE USED
                                        ; IN MORE THAN ONE TEST.
                                        ;--

002416  000000                  MOD::          .WORD   0       ; MODE REGISTER ADDRESS OF CURRENT UUT
002420  000000                  CSR::          .WORD   0       ; COMMAND AND STATUS REGISTER ADDRESS OF UUT
002422  000000                  CCR::          .WORD   0       ; COUNTER CONTROL REGISTER A ADDRESS OF UUT
002424  000000                  INR::          .WORD   0       ; INTERRUPT REGISTER ADDRESS OF CURRENT UUT
002426  000000                  VEC::          .WORD   0       ; VECTOR ADDRESS OF CURRENT UUT
002430  000000                  PRIO::         .WORD   0       ; PRIORITY LEVEL OF CURRENT UUT

                                        ;
                                        ; WORKING STORAGE
                                        ;
002432  000000                  SAVPRI::       .WORD   0       ; SAVED PRIORITY
002434  000000                  BITMSK::       .WORD   0       ; FOR BIT MAP
002436  000000                  BITMS1::       .WORD   0       ; GENERAL BIT MASK
002440  000000                  BITMS2::       .WORD   0       ; GENERAL BIT MASK
002442  000000                  NUMBER::       .WORD   0       ; GENERAL NUMBER
002444  000000                  COUNTR::       .WORD   0       ; GENERAL COUNTER
002446  000000                  GROUP::        .WORD   0       ; CURRENT GROUP
002450  000000                  ELEMNT::       .WORD   0       ; CURRENT ELEMENT
002452                          CNTVAL::       .BLKW   5       ; HOLDS VALUES FROM COUNTERS
002464  000000                  SRC::          .WORD   0       ; SOURCE FOR A TEST
002466  000000                  LOOP::         .WORD   0       ; GENERAL LOOP COUNT

002470  000000                  NXMFLG::       .WORD   0       ; SET IF NXM TRAP OCCURS
002472  000000                  BIV::          .WORD   0       ; HOLD VECTOR OF INTERRUPT
                                                               ; IF INTERRUPT DID NOT OCCUR THROUGH
                                                               ; SPECIFIED VECTOR

002474  000000                  GOOD::         .WORD   0       ; EXPECTED CONTENTS
002476  000000                  BAD::          .WORD   0       ; ACTUAL CONTENTS
002500  000000                  SFI::          .WORD   0       ; FLAG TO FORCE ERROR PRINTOUTS

002502  000000                  ANS::          .WORD   0       ; TEMPORARY STORE FOR MANUAL INPUTS
002504  000000                  ITRCNT::       .WORD   0       ; ITERRATION COUNTER
002506  000010                  ITRDEF::       .WORD   10      ; ITERRATION DEFAULT
002510  000000                  INTFLA::       .WORD   0       ; DONE INTERRUPT FLAG
002512  000000                  INTFL2::       .WORD   0       ; ERROR INTERRUPT FLAG


002514  100000                  ECNT::         .WORD   100000  ; ERROR COUNT FOR UUT 0 - BIT 15 IS SET TO
002516  100000                                 .WORD   100000  ; ERROR COUNT FOR UUT 1 - FLAG NOT TESTED.
002520  100000                                 .WORD   100000  ; ERROR COUNT FOR UUT 2
002522  100000                                 .WORD   100000  ; ERROR COUNT FOR UUT 3
002524  100000                                 .WORD   100000  ; ERROR COUNT FOR UUT 4
002526  100000                                 .WORD   100000  ; ERROR COUNT FOR UUT 5
002530  100000                                 .WORD   100000  ; ERROR COUNT FOR UUT 6
```

GLOBAL DATA SECTION

```
1430 002532  100000                          .WORD   100000  ; ERROR COUNT FOR UUT 7
1431 002534  100000                          .WORD   100000  ; ERROR COUNT FOR UUT 8
1432 002536  100000                          .WORD   100000  ; ERROR COUNT FOR UUT 9
1433 002540  100000                          .WORD   100000  ; ERROR COUNT FOR UUT 10
1434 002542  100000                          .WORD   100000  ; ERROR COUNT FOR UUT 11
1435 002544  100000                          .WORD   100000  ; ERROR COUNT FOR UUT 12
1436 002546  100000                          .WORD   100000  ; ERROR COUNT FOR UUT 13
1437 002550  100000                          .WORD   100000  ; ERROR COUNT FOR UUT 14
1438 002552  100000                          .WORD   100000  ; ERROR COUNT FOR UUT 15
1439
1440 002554  000000                GPADD::   .WORD   0       ; PARAMETER ADDRESS FOR UNIT 0
1441 002556  000000                          .WORD   0       ; PARAMETER ADDRESS FOR UNIT 1
1442 002560  000000                          .WORD   0       ; PARAMETER ADDRESS FOR UNIT 2
1443 002562  000000                          .WORD   0       ; PARAMETER ADDRESS FOR UNIT 3
1444 002564  000000                          .WORD   0       ; PARAMETER ADDRESS FOR UNIT 4
1445 002566  000000                          .WORD   0       ; PARAMETER ADDRESS FOR UNIT 5
1446 002570  000000                          .WORD   0       ; PARAMETER ADDRESS FOR UNIT 6
1447 002572  000000                          .WORD   0       ; PARAMETER ADDRESS FOR UNIT 7
1448 002574  000000                          .WORD   0       ; PARAMETER ADDRESS FOR UNIT 8
1449 002576  000000                          .WORD   0       ; PARAMETER ADDRESS FOR UNIT 9
1450 002600  000000                          .WORD   0       ; PARAMETER ADDRESS FOR UNIT 10
1451 002602  000000                          .WORD   0       ; PARAMETER ADDRESS FOR UNIT 11
1452 002604  000000                          .WORD   0       ; PARAMETER ADDRESS FOR UNIT 12
1453 002606  000000                          .WORD   0       ; PARAMETER ADDRESS FOR UNIT 13
1454 002610  000000                          .WORD   0       ; PARAMETER ADDRESS FOR UNIT 14
1455 002612  000000                          .WORD   0       ; PARAMETER ADDRESS FOR UNIT 15
1456
1457 002614                       DROPED::   .BLKB   16.     ; UNIT DROPPED FLAGS
1458
1459 002634  000000               TSTFLG::   .WORD   0       ; CLEARED AT START OF EACH PASS
1460                                                         ; SET IF ANY TEST IS SELECTED
1461 002636  000000               TSUFLG::   .WORD   0       ; CLEARED AT START OF EACH UNIT'S PASS
1462                                                         ; SET IF ANY TEST IS SELECTED
1463
1464
1465
1466 002640  000000               LOPFLG::   .WORD   0       ; SET IF LOOP CHECK ROUTINE CALLED FOR UNIT 0
1467 002642  000000                          .WORD   0       ; SET IF LOOP CHECK ROUTINE CALLED FOR UNIT 1
1468 002644  000000                          .WORD   0       ; SET IF LOOP CHECK ROUTINE CALLED FOR UNIT 2
1469 002646  000000                          .WORD   0       ; SET IF LOOP CHECK ROUTINE CALLED FOR UNIT 3
1470 002650  000000                          .WORD   0       ; SET IF LOOP CHECK ROUTINE CALLED FOR UNIT 4
1471 002652  000000                          .WORD   0       ; SET IF LOOP CHECK ROUTINE CALLED FOR UNIT 5
1472 002654  000000                          .WORD   0       ; SET IF LOOP CHECK ROUTINE CALLED FOR UNIT 6
1473 002656  000000                          .WORD   0       ; SET IF LOOP CHECK ROUTINE CALLED FOR UNIT 7
1474 002660  000000                          .WORD   0       ; SET IF LOOP CHECK ROUTINE CALLED FOR UNIT 8
1475 002662  000000                          .WORD   0       ; SET IF LOOP CHECK ROUTINE CALLED FOR UNIT 9
1476 002664  000000                          .WORD   0       ; SET IF LOOP CHECK ROUTINE CALLED FOR UNIT 10
1477 002666  000000                          .WORD   0       ; SET IF LOOP CHECK ROUTINE CALLED FOR UNIT 11
1478 002670  000000                          .WORD   0       ; SET IF LOOP CHECK ROUTINE CALLED FOR UNIT 12
1479 002672  000000                          .WORD   0       ; SET IF LOOP CHECK ROUTINE CALLED FOR UNIT 13
1480 002674  000000                          .WORD   0       ; SET IF LOOP CHECK ROUTINE CALLED FOR UNIT 14
1481 002676  000000                          .WORD   0       ; SET IF LOOP CHECK ROUTINE CALLED FOR UNIT 15
1482
1483 002700  000000               LOTFLA::   .WORD   0       ; SET BY INIT CODE IF LOOP ON TEST IS SELECTED
1484
1485 002702  000000               CONMSK::   .WORD   0       ; TEST CONTROL MASK SET UP BY INIT CODE
1486
```

```
                                                        ; BIT 0 IS SET IF UUT IS DIGITAL INPUT
                                                        ; BIT 1 IS SET IF UUT IS DIGITAL OUTPUT
                                ; LOW BYTE IS DEVICE CODE   ; BIT 2 IS SET IF UUT IS ANALOGUE INPUT
                                                        ; BIT 3 IS SET IF UUT IS ANALOGUE OUTPUT
                                                        ; BIT 4 IS SET IF UUT IS NONE OF THE
                                                        ; ABOVE
                                                        ; BITS 5, 6 AND 7 ARE UNUSED

                                                        ; BIT 8 IS ALWAYS SET TO SELECT BASIC
                                                        ; INTERNAL LOGIC TESTS

                                                        ; BIT 9 SET TO SELECT FIELD INPUT/OUTPUT
                                                        ; TESTS

                                                        ; BIT 10 IS SET IF LOOPBACK TESTING IS
                                                        ; SELECTED AND ALLOWED FOR CURRENT UUT

                                                        ; BIT 11 IS SET IF MANUFACTURING LOOPBACK
                                                        ; AND INPUT/OUTPUT TESTS ARE SELECTED

                                                        ; BIT 12 IS SET IF A SPECIFICALLY
                                                        ; SELECTABLE TEST IS CHOSEN

                                                        ; BITS 13, 14 AND 15 ARE UNUSED

002704  000000          MODE::  .WORD   0              ; MODE FOR DIGITAL/ANALOGUE CONVERSIONS
002706  000000          GAIN::  .WORD   0              ; GAIN FOR DIGITAL/ANALOGUE CONVERSIONS

002710  000000          PADD::  .WORD   0              ; ADDRESS OF PROMPT FOR DECIMAL INPUT ROUTINE


                                .EVEN
```

GLOBAL DATA SECTION

```
1522          000012                          .RADIX  10
1523                                          .NLIST  BEX
1524
1525                                  ; ANALOGUE/DIGITAL CONVERSION TABLES USED BY ROUTINES DACON AND ADCON.
1526
1527                                  ; VOLTAGE UNIPOLAR TABLE MODE 0 (0-10V)
1528
1529                                  ; BITS    11   10    9    8    7    6    5    4    3    2    1    0
1530
1531 002712  011610  004704  002342  VUPTAB:: 5000,2500,1250, 625, 312, 156,  78,  39,  19,   9,   4,   2 ; mV G=1
1532 002742  000000  000000  000000             0,   0,   0,   0, 500, 250, 125,  63, 531, 766, 883, 441 ; uV
1533
1534 002772  004704  002342  001161           2500,1250, 625, 312, 156,  78,  39,  19,   9,   4,   2,   1 ; mV G=2
1535 003022  000000  000000  000000             0,   0,   0, 500, 250, 125,  63, 531, 766, 883, 441, 221 ; uV
1536
1537 003052  001750  000754  000372           1000, 500, 250, 125,  62,  31,  15,   7,   3,   1,   0,   0 ; mV G=5
1538 003102  000000  000000  000000             0,   0,   0,   0, 500, 250, 625, 813, 906, 953, 977, 488 ; uV
1539
1540 003132  000764  000372  000175            500, 250, 125,  62,  31,  15,   7,   3,   1   0,   0,   0 ; mV G=10
1541 003162  000000  000000  000000              0,   0,   0, 500, 250, 625, 813, 906, 953, 977, 488, 244 ; uV
1542
1543 003212  000372  000175  000076            250, 125,  62,  31,  15,   7,   3,   1,   0,   0,   0,   0 ; mV G=20
1544 003242  000000  000000  000764              0,   0, 500, 250, 625, 813, 906, 953, 977, 488, 244, 122 ; uV
1545
1546 003272  000144  000062  000031            100,  50,  25,  12,   6,   3,   1,   0,   0,   0,   0,   0 ; mV G=50
1547 003322  000000  000000  000000              0,   0,   0, 500, 250, 125, 563, 781, 391, 195,  98,  49 ; uV
1548
1549 003352  000062  000031  000014             50,  25,  12,   6,   3,   1,   0,   0,   0,   0,   0,   0 ; mV G=100
1550 003402  000000  000000  000764              0,   0, 500, 250, 125, 563, 781, 391, 195,  98,  49,  24 ; uV
1551
1552 003432  000031  000014  000006             25,  12,   6,   3,   1,   0,   0,   0,   0,   0,   0,   0 ; mV G=200
1553 003462  000000  000764  000372              0, 500, 250, 125, 563, 781, 391, 195,  98,  49,  24,  12 ; uV
1554
```

```
1556                                    ;VOLTAGE BIPOLAR TABLE (-10 - +10V) MODE 1
1557
1558                                    ; BITS    11   10    9    8    7    6    5    4    3    2    1    0
1559
1560 003512  023420  011610  004704  VBPTAB::10000,5000,2500,1250, 625, 312, 156,  78,  39,  19,   9,   4 ; mV G=1
1561 003542  000000  000000  000000            0,   0,   0,   0,   0, 500, 250, 125,  63, 531, 766, 883 ; uV
1562
1563 003572  011610  004704  002342         5000,2500,1250, 625, 312, 156,  78,  39,  19,   9,   4,   2 ; mV G=2
1564 003622  000000  000000  000000            0,   0,   0,   0, 500, 250, 125,  63, 531, 766, 883, 441 ; uV
1565
1566 003652  003720  001750  000764         2000,1000, 500, 250, 125,  62,  31,  15,   7,   3,   1,   0 ; mV G=5
1567 003702  000000  000000  000000            0,   0,   0,   0,   0, 500, 250, 625, 813, 906, 953, 977 ; uV
1568
1569 003732  001750  000764  000372         1000, 500, 250, 125,  62,  31,  15,   7,   3,   1,   0,   0 ; mV G=10
1570 003762  000000  000000  000000            0,   0,   0,   0, 500, 250, 625, 813, 906, 953, 977, 488 ; uV
1571
1572 004012  000764  000372  000175          500, 250, 125,  62,  31,  15,   7,   3,   1,   0,   0,   0 ; mV G=20
1573 004042  000000  000000  000000            0,   0,   0, 500, 250, 625, 813, 906, 953, 977, 488, 244 ; uV
1574
1575 004072  000310  000144  000062          200, 100,  50,  25,  12,   6,   3,   1,   0,   0,   0,   0 ; mV G=50
1576 004122  000000  000000  000000            0,   0,   0,   0, 500, 250, 125, 563, 781, 391, 195,  98 ; uV
1577
1578 004152  000144  000062  000031          100,  50,  25,  12,   6,   3,   1,   0,   0,   0,   0,   0 ; mV G=100
1579 004202  000000  000000  000000            0,   0,   0, 500, 250, 125, 563, 781, 391, 195,  98,  49 ; uV
1580
1581 004232  000062  000031  000014           50,  25,  12,   6,   3,   1,   0,   0,   0,   0,   0,   0 ; mV G=200
1582 004262  000000  000000  000764            0,   0, 500, 250, 125, 563, 781, 391, 195,  98,  49,  24 ; uV
1583
```

GLOBAL DATA SECTION

```
1585                                    ;CURRENT 0-20 mA TABLE - MODE 2
1586
1587                                    ; BITS    11   10    9    8    7    6    5    4    3    2    1    0
1588
1589 004312  023420  011610  004704  IOTAB:: 10000,5000,2500,1250, 625, 312, 156,  78,  39,  19,   9,   4 ; uA G=1
1590 004342  000000  000000  000000            0,   0,   0,   0,   0, 500, 250, 125,  63, 531, 766, 883 ; nA
1591
1592 004372  011610  004704  002342          5000,2500,1250, 625, 312, 156,  78,  39,  19,   9,   4,   2 ; uA G=2
1593 004422  000000  000000  000000            0,   0,   0,   0, 500, 250, 125,  63, 531, 766, 883, 441 ; nA
1594
1595 004452  003720  001750  000764          2000,1000, 500, 250, 125,  62,  31,  15,   7,   3,   1,   0 ; mA G=5
1596 004502  000000  000000  000000            0,   0,   0,   0,   0, 500, 250, 625, 813, 906, 953, 977 ; nA
1597
1598 004532  001750  000764  000372          1000, 500, 250, 125,  62,  31,  15,   7,   3,   1,   0,   0 ; uA G=10
1599 004562  000000  000000  000000            0,   0,   0,   0, 500, 250, 625, 813, 906, 953, 977, 488 ; nA
1600
1601 004612  000764  000372  000175           500, 250, 125,  62,  31,  15,   7,   3,   1,   0,   0,   0 ; uA G=20
1602 004642  000000  000000  000000            0,   0,   0, 500, 250, 625, 813, 906, 953, 977, 488, 244 ; nA
1603
1604 004672  000310  000144  000062           200, 100,  50,  25,  12,   6,   3,   1,   0,   0,   0,   0 ; uA G=50
1605 004722  000000  000000  000000            0,   0,   0,   0, 500, 250, 125, 563, 781, 391, 195,  98 ; nA
1606
1607 004752  000144  000062  000031           100,  50,  25,  12,   6,   3,   1,   0,   0,   0,   0,   0 ; uA G=100
1608 005002  000000  000000  000000            0,   0,   0, 500, 250, 125, 563, 781, 391, 195,  98,  49 ; nA
1609
1610 005032  000062  000031  000014            50,  25,  12,   6,   3,   1,   0,   0,   0,   0,   0,   0 ; uA G=200
1611 005062  000000  000000  000764             0,   0, 500, 250, 125, 563, 781, 391, 195,  98,  49,  24 ; nA
1612
```

```
1614
1615                                    ; CURRENT 4 - 20 MA TABLE (MODE 3)
1616
1617                                    ; BITS    11   10    9    8    7    6    5    4    3    2    1    0
1618
1619 005112  017500  007640  003720  I4TAB:: 8000,4000,2000,1000, 500, 250, 125,  62,  31,  15,   7,   3 ; uA G=1
1620 005142  000000  000000  000000            0,   0,   0,   0,   0,   0,   0, 500, 250, 625, 813, 906 ; nA
1621
1622 005172  007640  003720  001750          4000,2000,1000, 500, 250, 125,  62,  31,  15,   7,   3,   1 ; uA G=2
1623 005222  000000  000000  000000            0,   0,   0,   0,   0,   0, 500, 250, 625, 813, 906, 953 ; nA
1624
1625 005252  003100  001440  000620          1600, 800, 400, 200, 100,  50,  25,  12,   6,   3,   1,   0 ; uA G=5
1626 005302  000000  000000  000000            0,   0,   0,   0,   0,   0,   0, 500, 250, 125, 563, 781 ; nA
1627
1628 005332  001440  000620  000310           800, 400, 200, 100,  50,  25,  12,   6,   3,   1,   0,   0 ; uA G=10
1629 005362  000000  000000  000000            0,   0,   0,   0,   0,   0, 500, 250, 125, 563, 781, 390 ; nA
1630
1631 005412  000620  000310  000144           400, 200, 100,  50,  25,  12,   6,   3,   1,   0,   0,   0 ; uA G=20
1632 005442  000000  000000  000000            0,   0,   0,   0,   0, 500, 250, 125, 563, 781, 390, 195 ; nA
1633
1634 005472  000240  000120  000050           160,  80,  40,  20,  10,   5,   2,   1,   0,   0,   0,   0 ; uA G=50
1635 005522  000000  000000  000000            0,   0,   0,   0,   0,   0, 500, 250, 625, 313, 156,  78 ; nA
1636
1637 005552  000120  000050  000024            80,  40,  20,  10,   5,   2,   1,   0,   0,   0,   0,   0 ; uA G=100
1638 005602  000000  000000  000000            0,   0,   0,   0,   0, 500, 250, 625, 313, 156,  78,  39 ; nA
1639
1640 005632  000050  000024  000012            40,  20,  10.  5,   2,   1,   0,   0,   0,   0,   0,   0 ; uA G=200
1641 005662  000000  000000  000000            0,   0,   0,   0, 500, 250, 625, 313, 156,  78,  39,  20 ; nA
1642
1643
1644          000010                          .RADIX  8
1645                                          .LIST   BEX
```

GLOBAL DATA SECTION

```
1647
1660
1661 005712                                  ERRTBL
     005712  000000              ERRTYP::          .WORD   0
     005714  000000              ERRNBR::          .WORD   0
     005716  000000              ERRMSG::          .WORD   0
     005720  000000              ERRBLK::          .WORD   0
```

```
                                .SBTTL  GLOBAL TEXT SECTION

                                ;++
                                ; THE GLOBAL TEXT SECTION CONTAINS FORMAT STATEMENTS,
                                ; MESSAGES, AND ASCII INFORMATION THAT ARE USED IN
                                ; MORE THAN ONE TEST.
                                ;--

                                        .NLIST  9EX

                                ;
                                ; NAMES OF DEVICES SUPPORTED BY PROGRAM
                                ;
005722                                  DEVTYP  <IDV11-D>


                                ; TEST DESCRIPTION
                                ;
005732                                  DESCRIPT        <FIVE CHANNEL COUNTER DIAGNOSTIC>


                                ;
                                ; FORMAT STATEMENTS USED IN PRINT CALLS
                                ;

005772  045  116  045  NODEV:: .ASCIZ  \%N%A*** NO DEVICES FOUND IN RANGE 171000 TO 171770 ***%N\

                                        .LIST   BEX
                                        .EVEN
```

```
1723                                    .SBTTL  GLOBAL ERROR REPORT SECTION
1724
1725                                    ;++
1726                                    ; THE GLOBAL ERROR REPORT SECTION CONTAINS MESSAGE PRINTING AREAS
1727                                    ; USED BY MORE THAN TEST TO OUTPUT ADDITIONAL ERROR INFORMATION.  PRINTB
1728                                    ; (BASIC) AND PRINTX (EXTENDED) CALLS ARE USED TO CALL PRINT SERVICES.
1729                                    ;-
1730
1731
1747
1748 006064                                BGNMSG  EER1
1749 006064                                PRINTB  #GOOBAD,GOOD,BAD        ; PRINT GOOD AND BAD
1750 006114  004737  014434                JSR     PC,CHKMAX               ; CHECK FOR TOO MANY ERRORS
1751 006120                                ENDMSG
1752
1753
1754 006122                                BGNMSG  EER2
1755 006122                                PRINTB  #EMG3,GOOD,BAD,@MOD
1756 006156  004737  014434                JSR     PC,CHKMAX
1757 006162                                ENDMSG
1758
1759 006164                                BGNMSG  EER3
1760 006164                                PRINTB  #EMG4,@CSR
1761 006210  004737  014434                JSR     PC,CHKMAX
1762 006214                                ENDMSG
1763
1764
1765
1766 006216                                BGNMSG  EER6
1767 006216                                PRINTB  #EMG4,BAD
1768 006242  004737  014434                JSR     PC,CHKMAX
1769 006246                                ENDMSG
1770
1771
1772
1773 006250                                BGNMSG  EERA
1774 006250                                PRINTB  #EMG1,R1
1775 006272  004737  014434                JSR     PC,CHKMAX
1776 006276                                ENDMSG
1777
1778
1779 006300                                BGNMSG  EERB
1780 006300                                PRINTB  #EMG2,R5,GOOD,BAD
1781 006332  004737  014434                JSR     PC,CHKMAX
1782 006336                                ENDMSG
1783
1784
1785 006340                                BGNMSG  EERG
1786 006340  004737  014434                JSR     PC,CHKMAX
1787 006344                                ENDMSG
1788
1789
1790
1791
1792
1793 006346                                BGNMSG  ERR104
1794 006346                                PRINTB  #E104,NUMBER            ;ERROR MESSAGE
```

GLOBAL ERROR REPORT SECTION

```
1795 006372  004737  014434                JSR     PC,CHKMAX
1796 006376                                ENDMSG
1797
1798 006400                                BGNMSG  ERR105
1799 006400                                PRINTB  @E105,NUMBER            ;ERROR MESSAGE
1800 006424  004737  014434                JSR     PC,CHKMAX
1801 006430                                ENDMSG
1802
1803 006432                                BGNMSG  ERR106
1804 006432                                PRINTB  @E106,GROUP             ;ERROR MESSAGE
1805 006456                                PRINTB  @E106A,COUNTR,R3        ;ERROR MESSAGE
1806 006504  004737  014434                JSR     PC,CHKMAX
1807 006510                                ENDMSG
1808
1809 006512                                BGNMSG  ERR107
1810 006512                                PRINTB  @E107,GROUP             ;ERROR MESSAGE
1811 006536                                PRINTB  @E107A,COUNTR,R3        ;ERROR MESSAGE
1812 006564  004737  014434                JSR     PC,CHKMAX
1813 006570                                ENDMSG
1814
1815 006572                                BGNMSG  ERR108
1816 006572                                PRINTB  @E108,NUMBER            ;ERROR MESSAGE
1817 006616  004737  014434                JSR     PC,CHKMAX
1818 006622                                ENDMSG
1819
1820 006624                                BGNMSG  ERR109
1821 006624                                PRINTB  @E109,NUMBER            ;ERROR MESSAGE
1822 006650  004737  014434                JSR     PC,CHKMAX
1823 006654                                ENDMSG
1824
1825 006656                                BGNMSG  ERR110
1826 006656                                PRINTB  @E110,NUMBER            ;ERROR MESSAGE
1827 006702  004737  014434                JSR     PC,CHKMAX
1828 006706                                ENDMSG
1829
1830 006710                                BGNMSG  ERR111
1831 006710                                PRINTB  @E111,NUMBER            ;ERROR MESSAGE
1832 006734  004737  014434                JSR     PC,CHKMAX
1833 006740                                ENDMSG
1834
1835 006742                                BGNMSG  ERR112
1836 006742                                PRINTB  @E112,NUMBER            ;ERROR MESSAGE
1837 006766  004737  014434                JSR     PC,CHKMAX
1838 006772                                ENDMSG
1839
1840 006774                                BGNMSG  ERR501
1841 006774                                PRINTB  @E501
1842 007014                                PRINTB  @GOODBAD,R2,BAD         ;ERROR MESSAGE
1843 007042  004737  014434                JSR     PC,CHKMAX
1844 007046                                ENDMSG
1845
1846 007050                                BGNMSG  ERR502
1847 007050                                PRINTB  @E502                   ;ERROR MESSAGE
1848 007070                                PRINTB  @GOODBAD,R2,BAD         ;ERROR MESSAGE
1849 007116  004737  014434                JSR     PC,CHKMAX
1850 007122                                ENDMSG
1851
```

```
1852 007124                                    BGNMSG  ERR503
1853 007124                                    PRINTB  @E113                   ;ERROR MESSAGE
1854 007144  004737  014434                    JSR     PC,CHKMAX
1855 007150                                    ENDMSG
1856
1857 007152                                    BGNMSG  ERR504
1858 007152                                    PRINTB  @E114,INTFLA            ;ERROR MESSAGE
1859 007176  004737  014434                    JSR     PC,CHKMAX
1860 007202                                    ENDMSG
1861
1862 007204                                    BGNMSG  ERR505
1863 007204                                    PRINTB  @E114A                  ;ERROR MESSAGE
1864 007224                                    PRINTB  @E114B,VEC,BIV          ;ERROR MESSAGE
1865 007254  004737  014434                    JSR     PC,CHKMAX
1866 007260                                    ENDMSG
1867
1868
1869 007262                                    BGNMSG  ERR506
1870 007262                                    PRINTB  @E506                   ;ERROR MESSAGE
1871 007302                                    PRINTB  @GOODBAD,GOOD,BAD       ;ERROR MESSAGE
1872 007332  004737  014434                    JSR     PC,CHKMAX
1873 007336                                    ENDMSG
1874
1875
1876 007340                                    BGNMSG  ERR507
1877 007340                                    PRINTB  @E507                   ;ERROR MESSAGE
1878 007360                                    PRINTB  @E507A,BAD              ;ERROR MESSAGE
1879 007404  004737  014434                    JSR     PC,CHKMAX
1880 007410                                    ENDMSG
1881
1882
1883 007412                                    BGNMSG  ERR115
1884 007412                                    PRINTB  @E115,NUMBER            ;ERROR MESSAGE
1885 007436  004737  014434                    JSR     PC,CHKMAX
1886 007442                                    ENDMSG
1887
1888 007444                                    BGNMSG  ERR130
1889 007444                                    PRINTB  @STRVA1
1890 007464                                    PRINTB  @STRVA2
1891 007504  012702  000001                    MOV     #1,R2
1892 007510  012701  002452                    MOV     @CNTVAL,R1
1893 007514  012103                    60$:    MOV     (R1)+,R3
1894 007516                                    PRINTB  @CNTXT,R2,R3
1895 007542  005202                            INC     R2                      ;CHANNEL NUMBER
1896 007544  020227  000005                    CMP     R2,#5                   ;ALL CHANNELS?
1897 007550  003761                            BLE     60$                     ;MORE TO PRINT
1898 007552  004737  014434                    JSR     PC,CHKMAX
1899 007556                                    ENDMSG
1900
1901 007560                                    BGNMSG  ERR116
1902 007560                                    PRINTB  @E116,NUMBER,R2         ;SOURCE
1903 007606                                    PRINTB  @E116B
1904 007626  004737  014434                    JSR     PC,CHKMAX
1905 007632                                    ENDMSG
1906
1907 007634                                    BGNMSG  ER116A
1908 007634                                    PRINTB  @E116A,NUMBER,R2
```

```
1909 007662                                    PRINTB  #E116B
1910 007702  004737  014434                    JSR     PC,CHKMAX
1911 007706                                    ENDMSG
1912
1913 007710                                    BGNMSG  ER116B
1914 007710                                    PRINTB  #E116,R2,R3             ;SOURCE
1915 007734                                    PRINTB  #E116B
1916 007754  004737  014434                    JSR     PC,CHKMAX
1917 007760                                    ENDMSG
1918
1919 007762                                    BGNMSG  ER116C
1920 007762                                    PRINTB  #E116A,R2,R3
1921 010006                                    PRINTB  #E116B
1922 010026  004737  014434                    JSR     PC,CHKMAX
1923 010032                                    ENDMSG
1924
1925 010034                                    BGNMSG  ERR120
1926 010034                                    PRINTB  #E120,R5                ;ERROR MESSAGE
1927 010056                                    PRINTB  #E116B
1928 010076  004737  014434                    JSR     PC,CHKMAX
1929 010102                                    ENDMSG
1930
1931 010104                                    BGNMSG  ERR121
1932 010104                                    PRINTB  #E121,R5                ;ERROR MESSAGE
1933 010126  004737  014434                    JSR     PC,CHKMAX
1934 010132                                    ENDMSG
1935
1936 010134                                    BGNMSG  ERR122
1937 010134                                    PRINTB  #E121,R5                ;ERROR MESSAGE
1938 010156  004737  014434                    JSR     PC,CHKMAX
1939 010162                                    ENDMSG
1940
1941 010164                                    BGNMSG  ERR123
1942 010164                                    PRINTB  #E120,R5                ;ERROR MESSAGE
1943 010206                                    PRINTB  #E116B
1944 010226  004737  014434                    JSR     PC,CHKMAX
1945 010232                                    ENDMSG
1946
1947 010234                                    BGNMSG  ERR124
1948 010234                                    PRINTB  #E124,(R5),R0           ;ERROR MESSAGE
1949 010260                                    PRINTB  #E116B
1950 010300  004737  014434                    JSR     PC,CHKMAX
1951 010304                                    ENDMSG
1952
1953
1954
1955
1956
1957
1958
1959
1960                                           .NLIST BEX
1961 010306       045       101       107     GOOBAD: .ASCIZ  /%AGOOD:%O6%A, BAD:%O6%N/
1962 010336       045       101       122     EMG1:   .ASCIZ  /%AREGISTER AT %O6%A DOES NOT RESPOND%N/
1963 010405       045       101       101     EMG2:   .ASCIZ  /%AADDRESS: %O6%A, GOOD:%O6%A, BAD:%O6%N/
1964 010455       045       101       107     EMG3:   .ASCIZ  /%AGOOD:%O6%A, BAD:%O6%A, MOD REGISTER CONTENTS:%O6%N/
1965 010542       045       101       103     EMG4:   .ASCIZ  /%ACSR REGISTER CONTENTS IS :%O6%N/
```

```
                                        .LIST BEX
                                        .EVEN


                                                .NLIST  BEX
 010604      105     122     122  E102:     .ASCIZ  /%AERROR - SPECIAL MODULE BIT IN MADR IS NOT SET/
 010662      105     122     122  E103:     .ASCIZ  /%AERROR - 'CT MOD' BIT IN MEDR IS NOT SET/
 010732      045     101     105  E104:     .ASCIZ  /%AERROR - UNABLE TO CLEAR 'OUT' BIT NR: %D%N/
 011007      045     101     105  E105:     .ASCIZ  /%AERROR - UNABLE TO SET 'OUT' BIT NR: %D%N/
 011062      045     101     105  E106:     .ASCIZ  /%AERROR - WRITING LOAD REGISTER OF CHA.:%D1%N/
 011140      045     101     054  E106A:    .ASCIZ  /%A, WRITTEN: %O6%A,  READ: %O6%N/
 011201      045     101     105  E107:     .ASCIZ  /%AERROR - WRITING HOLD REGISTER OF CHA.:%D1%N/
 011257      045     101     054  E107A:    .ASCIZ  /%A, WRITTEN: %O6%A,  READ:  %O6%N/
 011321      045     101     105  E108:     .ASCIZ  /%AERROR - CAN NOT CLEAR 'EN' BIT NR: %D%N/
 011373      045     101     105  E109:     .ASCIZ  /%AERROR - CAN NOT SET 'EN' BIT NR: %D%N/
 011443      045     101     105  E110:     .ASCIZ  /%AERROR - CAN NOT CLEAR 'INT' BIT NR: %D%N/
 011516      045     101     105  E111:     .ASCIZ  /%AERROR - CAN NOT SET 'INT' BIT NR: %D%N/
 011567      045     101     105  E112:     .ASCIZ  /%AERROR - 'INT' BIT NR: %D1%A HAS RESET%N/
 011641      045     101     105  E113:     .ASCIZ  /%AERROR - UNEXPECTED INTERRUPT DETECTED%N/
 011713      045     101     105  E114:     .ASCIZ  /%AERROR - EXPECTING ONE INTERRUPT, ENCOUNTERED %D5%N/
 012000      045     101     105  E114A:    .ASCIZ  /%AERROR - BAD INTERRUPT VECTOR DETECTED%N/
 012052      045     101     123  E114B:    .ASCIZ  /%ASET UP VECTOR :%O3%A  FOUND VECTOR :%O3%N/
 012126      045     101     103  E115:     .ASCIZ  /%ACOUNTER %D1%A  DID NOT INCREMENT%N/
 012173      045     101     125  STRVA1:   .ASCII  /%AUNEXPECTED DIFFERENCE BETWEEN COUNTERS %N/
 012246      045     101     123            .ASCIZ  /%ASHOULD NOT BE GREATER THAN 2%N/
 012307      045     101     103  STAVA2:   .ASCIZ  /%ACOUNT DOWN FROM 10,000 AT 5 MHZ FOR 3 NOP INSTRUCTIONS.%N/
 012403      045     101     103  CNTXT:    .ASCIZ  /%ACOUNTER %D1%A  VALUE: %D6%N/
                                        .EVEN
 012442      045     101     103  E116:     .ASCIZ  /%ACOUNTER %D1%A  DID NOT INCREMENT%A, SOURCE %D2%N/
 012525      045     101     103  E116A:    .ASCIZ  /%ACOUNTER %D1%A  DID NOT INCREMENT%A, GATE %D2%N/
 012606      045     101     105  E116B:    .ASCIZ  /%AENSURE THAT TEST CONNECTOR IS INSTALLED%N/
 012662      045     101     102  E120:     .ASCIZ  /%ABAD COUNTER VALUE, EXPECTED 10000, DETECTED: %D5%N/
 012747      045     101     102  E121:     .ASCIZ  /%ABAD COUNTER VALUE, EXPECTED 0, DETECTED: %D5%N/
 013030      045     101     105  E124:     .ASCIZ  /%AERROR LOADING TOGGLE, WRITTEN %O6%A, READ: %O6%N/
 013113      045     101     111  E501:     .ASCIZ  /%AINR REGISTER INCORRECT%N/
 013146      045     101     111  E502:     .ASCIZ  /%AIR BIT IN INR REGISTER NOT RESETED AFTER BIT CLEAR%N/
 013235      045     101     111  E506:     .ASCIZ  /%AINTERRUPT DID NOT OCCURRDED AT THE CORRECT PRIORITY LEVEL%N/
 013333      045     101     115  E507:     .ASCIZ  /%AMASTER ENB BIT IN INR REGISTER NOT CLEARED AFTER INTERRUPT%N/
 013432      045     101     111  E507A:    .ASCIZ  /%AINR REGISTER CONTENS IS :%O6%N/
                                        .EVEN
                                        .NLIST  BEX
```

```
                        .SBTTL  GLOBAL SUBROUTINES SECTION

                        ;**
                        ; THE GLOBAL SUBROUTINES SECTION CONTAINS THE SUBROUTINES
                        ; THAT ARE USED IN MORE THAN ONE TEST.
                        ;-

                        ; SUBROUTINE REGTST - GENERAL PURPOSE REGISTER TEST.

                        ;**
                        ; FUNCTIONAL DESCRIPTION:
                        ;
                        ;       CHECKS THAT ALL READ/WRITE BITS OF THE SELECTED REGISTER CAN BE
                        ;       SET, CLEARED, AND INDIVIDUALLY SET (SLIDING ONES PATTERN).
                        ;
                        ; INPUTS:
                        ;
                        ;       IF ENTERED AT LOCATION REGTST, THE LOCATIONS FOLLOWING THE
                        ;       SUBROUTINE CALL MUST CONTAIN THE READ/WRITE BIT MASK, THE
                        ;       ADDRESS OF THE REGISTER TO BE TESTED, AND THE FIRST ERROR NUMBER
                        ;       TO BE USED (SEE CALLING SEQUENCE).
                        ;
                        ;       IF ENTERED AT LOCATION REGTS1, THE READ/WRITE BIT MASK, REGISTER
                        ;       ADDRESS TO BE TESTED, AND THE FIRST ERROR NUMBER MUST BE LOADED
                        ;       INTO LOCATIONS MASK, REGADD, AND ERRNBR RESPECTIVELY. THIS
                        ;       ALLOWS THE ARGUMENTS TO BE VARIED AT RUN TIME.
                        ;
                        ; IMPLICIT INPUTS:
                        ;
                        ;       NONE.
                        ;
                        ; OUTPUTS:
                        ;
                        ;       ERROR MESSAGES IF ERRORS OCCUR.
                        ;
                        ; IMPLICIT OUTPUTS:
                        ;
                        ;       IF ENTERED AT LOCATION REGTST,
                        ;
                        ;       MASK    - CONTAINS THE READ/WRITE BIT MASK
                        ;       REGADD  - CONTAINS THE ADDRESS OF THE REGISTER BEING TESTED
                        ;
                        ;       ALWAYS,
                        ;
                        ;       MASCOM - CONTAINS THE COMPLEMENT OF THE MASK
                        ;       GOOD   - CONTAINS LAST EXPECTED DATA
                        ;       BAD    - CONTAINS LAST ACTUAL DATA
                        ;       ERRNBR - CONTAINS THE INPUT ERROR NUMBER * 2
                        ;       ERRTYP - CONTAINS 3 (SOFT ERROR)
                        ;       ERRBLK - CONTAINS ADDRESS OF REGERR (REGISTER ERROR MESSAGE)
                        ;       ERRMSG - CONTAINS 3RD REGISTER ERROR MESSAGE
                        ;
                        ; SUBORDINATE ROUTINES USED:
                        ;
                        ;       INSERT - ERROR INSERTION ROUTINE
                        ;       CHKMAX - ERROR COUNT CHECKING ROUTINE
                        ;       DRS ERROR MACRO
```

```
                                ;
                                ; FUNCTIONAL SIDE EFFECTS:
                                ;
                                ;       NONE.
                                ;
                                ; CALLING SEQUENCE:
                                ;
                                ;       EITHER FIXED PARAMETERS FOLLOW THE SUBROUTINE CALL :
                                ;
                                ;       EG.     CALL    REGTST
                                ;               177                     ; BIT MASK OF R/W BITS
                                ;               CSR                     ; REGISTER ADDRESS
                                ;               200.                    ; FIRST ERROR NUMBER
                                ;
                                ;       OR PARAMETERS ARE SET DYNAMICALLY :
                                ;
                                ;       EG.     MOV     #177,MASK       ; BIT MASK OF R/W BITS
                                ;               MOV     CSR,REGADD      ; REGISTER ADDRESS
                                ;               MOV     #200.,ERRNBR    ; FIRST ERROR NUMBER
                                ;               CALL    REGTS1
                                ;
                                ;--

                                REGTST::
013474  017637  000000  014046          MOV     @(SP),MASK              ; GET R/W BIT MASK
013502  062716  000002                  ADD     #2,(SP)                 ; JUMP OVER ARGUMENT
013506  017637  000000  014052          MOV     @(SP),REGADD            ; GET REGISTER ADDRESS
013514  062716  000002                  ADD     #2,(SP)                 ; JUMP OVER ARGUMENT
013520  017637  000000  005714          MOV     @(SP),ERRNBR            ; GET FIRST ERROR NUMBER
013526  062716  000002                  ADD     #2,(SP)                 ; JUMP OVER ARGUMENT
013532                          REGTS1::
013532  013737  014046  014050          MOV     MASK,MASCOM             ; SET UP COMPLEMENT
013540  005137  014050                  COM     MASCOM                  ; OF R/W BIT MASK
013544  012737  000003  005712          MOV     #3,ERRTYP               ; SET UP FOR SOFT ERROR
013552  012737  014054  005720          MOV     #REGERR,ERRBLK          ; SET UP ERROR MESSAGE ROUTINE
013560  012737  014122  005716          MOV     #RERR1,ERRMSG           ; FIRST ERROR MESSAGE
                                ;
                                ; CHECK THAT ALL R/W BITS CAN BE SET
                                ;
013566  013737  014046  002474          MOV     MASK,GOOD               ; SET UP EXPECTED DATA
013574                                  BGNSEG
013576  053777  002474  000246          BIS     GOOD,@REGADD            ; SET ALL R/W BITS
013604  017737  000242  002476          MOV     @REGADD,BAD             ; READ THE RESULT
013612  043737  014050  002476          BIC     MASCOM,BAD              ; KEEP ONLY R/W BITS
013620  023737  002476  002474          CMP     BAD,GOOD                ; ALL R/W BITS SET?
013626  004737  015240                  CALL    INSERT                  ; ALLOW FORCED ERROR PRINTOUT
013632  001401                          BEQ     10$                     ; IF OK, BRANCH
013634                                  ERROR                           ; ELSE REPORT ERROR
013636                          10$:    ENDSEG
                                ;
                                ; CHECK THAT ALL R/W BITS CAN BE CLEARED
                                ;
013640  005037  002474                  CLR     GOOD                    ; SET UP EXPECTED DATA
013644  005237  005714                  INC     ERRNBR                  ; NEXT ERROR NUMBER
013650  012737  014174  005716          MOV     #RERR2,ERRMSG           ; NEXT ERROR MESSAGE

013656                                  BGNSEG
```

```
2132 013660  043777  014046  000164          BIC     MASK,@REGADD            ; CLEAR ALL R/W BITS
2133 013666  017737  000160  002476          MOV     @REGADD,BAD             ; READ THE RESULT
2134 013674  043737  014050  002476          BIC     MASCOM,BAD              ; KEEP ONLY R/W BITS
2135 013702  023737  002476  002474          CMP     BAD,GOOD                ; ALL R/W BITS CLEAR?
2136 013710  004737  015240                  CALL    INSERT                  ; ALLOW FORCED ERROR PRINTOUT
2137 013714  001401                          BEQ     20$                     ; IF OK, BRANCH
2138 013716                                  ERROR                           ; ELSE REPORT ERROR
2139 013720                          20$:    ENDSEG
2140                                         ;
2141                                         ; CHECK THAT EACH R/W BIT CAN BE SET
2142                                         ;
2143 013722  005237  005714                  INC     ERRNBR                  ; NEXT ERROR NUMBER
2144 013726  012737  014252  005716          MOV     #RERR3,ERRMSG           ; NEXT ERROR MESSAGE
2145 013734  012737  000001  002474          MOV     #1,GOOD                 ; FIRST BIT TO TEST
2146 013742  033737  002474  014046  30$:    BIT     GOOD,MASK               ; R/W BIT?
2147 013750  001004                          BNE     50$                     ; IF YES, TEST IT
2148 013752  006337  002474          40$:    ASL     GOOD                    ; ELSE FIND NEXT R/W BIT
2149 013756  103427                          BCS     70$                     ; IF ALL DONE, RETURN
2150 013760  000770                          BR      30$                     ; ELSE CHECK IF NEXT IS R/W
2151
2152 013762                          50$:    BGNSEG
2153 013764  043777  014046  000060          BIC     MASK,@REGADD            ; CLEAR ALL R/W BITS
2154 013772  053777  002474  000052          BIS     GOOD,@REGADD            ; SET ONE BIT
2155 014000  017737  000046  002476          MOV     @REGADD,BAD             ; READ IT BACK
2156 014006  043737  014050  002476          BIC     MASCOM,BAD              ; KEEP ONLY R/W BITS
2157 014014  023737  002476  002474          CMP     BAD,GOOD                ; ALL OTHER BITS CLEAR?
2158 014022  004737  015240                  CALL    INSERT                  ; ALLOW FORCED ERROR PRINTOUT
2159 014026  001401                          BEQ     60$                     ; IF OK, BRANCH
2160 014030                                  ERROR                           ; ELSE REPORT ERROR
2161 014032                          60$:    ENDSEG
2162 014034  000746                          BR      40$                     ; TEST NEXT BIT
2163
2164 014036  043777  014046  000006  70$:    BIC     MASK,@REGADD            ; LEAVE THE R/W BITS CLEAR
2165 014044  000207                          RETURN
2166
2167
2168 014046  000000                  MASK::  .WORD   0                       ; BIT MASK OF READ/WRITE BITS
2169 014050  000000                  MASCOM: .WORD   0                       ; COMPLEMENT OF MASK
2170 014052  000000                  REGADD::.WORD   0                       ; ADDRESS OF REGISTER TO BE TESTED
2171
2172 014054                                  BGNMSG  REGERR
2173 014054                                  PRINTB  #REGMSG,REGADD,GOOD,BAD,MASK
2174 014114  004737  014434                  JSR     PC,CHKMAX               ; CHECK FOR TOO MANY ERRORS
2175 014120                                  ENDMSG
2176
2177                                         .NLIST BEX
2178
2179 014122     122     105     107  RERR1:  .ASCIZ  <REGISTER READ/WRITE BITS COULD NOT BE SET>
2180 014174     122     105     107  RERR2:  .ASCIZ  <REGISTER READ/WRITE BITS COULD NOT BE CLEARED>
2181 014252     122     105     107  RERR3:  .ASCIZ  <REGISTER READ/WRITE BITS COULD NOT BE INDIVIDUALLY SET>
2182
2183 014341     045     101     101  REGMSG: .ASCIZ  .<ADDRESS: <06>A, GOOD: <06>A, BAD: <06>A, R/W BITS: <06>N.
2184
2185                                         .LIST BEX
2186                                         .EVEN
```

```
                        ;+
                        ; SUBROUTINE CHKMAX - ERROR COUNT CHECKING ROUTINE.
                        ;
                        ;++
                        ; FUNCTIONAL DESCRIPTION:
                        ;
                        ;       SUBROUTINE TO UPDATE UNIT ERROR COUNT. IF THE PROGRAM IS LOOPING
                        ;       ON AN ERROR, THE SUBROUTINE DOES NOTHING. OTHERWISE, THE ERROR
                        ;       COUNT FOR THE UNIT IS INCREMENTED. IF THE ERROR COUNT EXCEEDS 5
                        ;       AND THE USER FLAG EVL HAS BEEN SELECTED AND THE FLAG IDU IS NOT
                        ;       SELECTED, THE UNIT IS DROPPED FROM THE TEST CYCLE.
                        ;
                        ; INPUTS:
                        ;
                        ;       NONE.
                        ;
                        ; IMPLICIT INPUTS:
                        ;
                        ;       L$LUN CONTAINS THE NUMBER OF THE UNIT CURRENTLY BEING TESTED.
                        ;       ECNT IS THE ADDRESS OF THE ERROR COUNT FOR UNIT 0.
                        ;
                        ; OUTPUTS:
                        ;
                        ;       NONE.
                        ;
                        ; IMPLICIT OUTPUTS:
                        ;
                        ;       THE ERROR COUNT FOR THE LOGICAL UNIT BEING TESTED IS
                        ;       INCREMENTED IF THE PROGRAM IS NOT LOOPING.
                        ;
                        ; SUBORDINATE ROUTINES USED:
                        ;
                        ;       NONE
                        ;
                        ; FUNCTIONAL SIDE EFFECTS:
                        ;
                        ;       IF THE ERROR COUNT EXCEEDS 5 AND THE USER EVL FLAG IS SELECTED,
                        ;       AND THE 'LOOP ON TEST' AND 'INHIBIT DROPPING OF UNITS' FLAGS ARE
                        ;       NOT SELECTED, THE UNIT WILL BE DROPPED FROM TESTING.
                        ;
                        ; CALLING SEQUENCE:
                        ;
                        ;       JSR PC,CHKMAX
                        ;
                        ;--
                        ;
014434                  CHKMAX::INLOOP                  ; LOOPING ON ERROR?
014436                          BCOMPLETE 10$           ; IF YES, EXIT
                        ;
014440  013700  002074          MOV     L$LUN,R0        ; GET CURRENT UNIT
014444  006300                  ASL     R0              ; CONVERT TO ERROR COUNT OFFSET
014446  005260  002514          INC     ECNT(R0)        ; UPDATE THE ERROR COUNT
014452  026027  002514  000005  CMP     ECNT(R0),#5     ; TOO MANY ERRORS?
014460  003425                  BLE     10$             ; IF NOT, JUMP
                        ;
014462                          RFLAGS  R0              ; GET OPERATOR FLAGS
014464  032700  000040          BIT     #IDU,R0         ; IS DROPPING INHIBITED?
014470  001021                  BNE     10$             ; IF YES, EXIT
```

```
2245 014472  032700  000004                BIT     @EVL,R0          ; EVALUATE FLAG SELEC ED ?
2246 014476  001416                        BEQ     10$              ; IF NOT, EXIT
2247
2248 014500                                PRINTF  @NERRS,L$LUN     ; 'TOO MANY ERRORS'
2249 014524                                DODU    L$LUN            ; DROP THE UNIT
2250
2251 014532                                DOCLN                    ; END THE SUBPASS
2252
2253 014534  000207                 10$:   RTS     PC
2254
2255                                       .NLIST  BEX
2256 014536     045     116     045  NERRS: .ASCIZ  /<N>AMORE THAN 5 ERRORS ON UNIT<D2/
2257                                       .LIST   BEX
2258                                       .EVEN
2259
2260
2261
```

```
                                ; SUBROUTINES WT25M, WT500 AND WT25 - DELAY ROUTINES.
                                ;
                                ;++
                                ; FUNCTIONAL DESCRIPTION:
                                ;
                                ;       SUBROUTINE TO WAIT FOR 25 MILLISECONDS, 500 MICROSECONDS OR 25
                                ;       MICROSECONDS.
                                ;
                                ;       NOTE. BECAUSE OF THE SMALL NUMBER OF PROGRAM WAIT LOOPS USED FOR
                                ;             THE 25 MICROSECOND COUNTER, THE ACCURACY OF THE WT25 ROUTINE
                                ;             IS LOW. THE DELAY MAY LAST UP TO 50 MICROSECONDS ON SOME
                                ;             SLOW PROCESSORS.
                                ;
                                ; INPUTS:
                                ;
                                ;       NONE.
                                ;
                                ; IMPLICIT INPUTS:
                                ;
                                ;       THE VARIABLES CNT25M, CNT500, AND CNT25 MUST HAVE BEEN SET UP BY
                                ;       ROUTINE SETCLK.
                                ;
                                ; OUTPUTS:
                                ;
                                ;       NONE.
                                ;
                                ; IMPLICIT OUTPUTS:
                                ;
                                ;       NONE.
                                ;
                                ; SUBORDINATE ROUTINES USED:
                                ;
                                ;       NONE.
                                ;
                                ; FUNCTIONAL SIDE EFFECTS:
                                ;
                                ;       NONE.
                                ;
                                ; CALLING SEQUENCE:
                                ;
                                ;       JSR     PC,WT25M        ; WAIT FOR 25 MILLISECONDS
                                ;    OR JSR     PC,WT500        ; WAIT FOR 500 MICROSECONDS
                                ;    OR JSR     PC,WT25         ; WAIT FOR 25 MICROSECONDS
                                ;
                                ;--

014600  013700  014626          WT25M:: MOV     CNT25M,R0       ; GET 25 MILLISECOND WAIT COUNT
014604  000405                          BR      WAIT            ;

014606  013700  014630          WT500:: MOV     CNT500,R0       ; GET 500 MICROSECOND WAIT COUNT
014612  000402                          BR      WAIT            ;

014614  013700  014632          WT25::  MOV     CNT25,R0        ; GET 25 MICROSECOND WAIT COUNT

014620  005300                  WAIT:   DEC     R0              ; ALL DONE?
014622  001376                          BNE     WAIT            ; IF NOT, WAIT SOME MORE
014624  000207                          RTS     PC              ; ELSE RETURN
```

L4

GLOBAL SUBROUTINES SECTION

```
   2320
   2321 014626  000000                CNT25M::.WORD   0          ; COUNTER FOR 25 MILLISECOND DELAY
   2322 014630  000000                CNT500::.WORD   0          ; COUNTER FOR 500 MICROSECOND DELAY
   2323 014632  000000                CNT25:: .WORD   0          ; COUNTER FOR 25 MICROSECOND DELAY
   2324
```

```
                        .SUBROUTINE CRLF - ROUTINE TO PRINT CARRIAGE RETURN, LINE FEED.

                        ;++
                        ; FUNCTIONAL DESCRIPTION:
                        ;
                        ;       PRINTS A CARRIAGE RETURN AND LINE FEED.
                        ;
                        ; INPUTS:
                        ;
                        ;       NONE.
                        ;
                        ; IMPLICIT INPUTS:
                        ;
                        ;       NONE.
                        ;
                        ; OUTPUTS:
                        ;
                        ;       A CARRIAGE RETURN AND LINE FEED ARE PRINTED.
                        ;
                        ; IMPLICIT OUTPUTS:
                        ;
                        ;       NONE.
                        ;
                        ; SUBORDINATE ROUTINES USED:
                        ;
                        ;       SUPERVISOR PRINTF MACRO.
                        ;
                        ; FUNCTIONAL SIDE EFFECTS:
                        ;
                        ;       NONE.
                        ;
                        ; CALLING SEQUENCE:
                        ;
                        ;       JSR     PC,CRLF
                        ;
                        ;--

014634                  CRLF::
014634                          PRINTF  #LF
014654  000207                  RTS     PC

014656      045     116     000 LF:     .ASCIZ  /%N/
                                .EVEN
```

```
                                ; SUBROUTINE WRDY - SUBROUTINE TO WAIT FOR OPERATOR READY
                                ;
                                ;**
                                ; FUNCTIONAL DESCRIPTION:
                                ;
                                ;       THIS PRINTS A MESSAGE FOR THE OPERATOR TO TYPE 'CARRIAGE RETURN'
                                ;       TO CONTINUE. THE ROUTINE IS NORMALLY USED TO ALLOW A MESSAGE TO
                                ;       BE READ BEFORE PROCEDING.
                                ;
                                ;       IF MANUAL INTERVENTION IS NOT ALLOWED, THE ROUTINE DOES NOTHING.
                                ;
                                ; INPUTS:
                                ;
                                ;       NONE.
                                ;
                                ; IMPLICIT INPUTS:
                                ;
                                ;       NONE.
                                ;
                                ; OUTPUTS:
                                ;
                                ;       'TYPE 'CARRIAGE RETURN' TO CONTINUE OR 'CONRTOL C' TO ABORT.
                                ;
                                ; IMPLICIT OUTPUTS:
                                ;
                                ;       NONE.
                                ;
                                ; SUBORDINATE ROUTINES USED:
                                ;
                                ;       SUPERVISOR GMANID MACRO.
                                ;
                                ; FUNCTIONAL SIDE EFFECTS:
                                ;
                                ;       NONE.
                                ;
                                ; CALLING SEQUENCE:
                                ;
                                ;       JSR     PC,WRDY
                                ;
                                ;--
                                WRDY::
                                        MANUAL                                  ; IS MANUAL INTERVENTION ALLOWED ?
                                        BNCOMPLETE 10$                          ; IF NOT, EXIT
                                        GMANID  WRDY1,WFLG,A,377,0,1,YES ; 'TYPE RETURN TO CONTINUE'
                                10$:    RTS     PC
                                WFLG:   .WORD   0               ; FLAG FOR WARNING MESSAGE INPUT
                                        .NLIST  BEX
                                WRDY1:  .ASCIZ  /TYPE "CARRIAGE RETURN" TO CONTINUE OR "CONTROL C" TO ABORT./
                                        .LIST   BEX
                                        .EVEN
```

```
2424                            ; SUBROUTINE SELECT - TEST SELECT ROUTINE
2425                            ;
2426                            ;++
2427                            ; FUNCTIONAL DESCRIPTION:
2428                            ;
2429                            ;       THIS IS CALLED BY EACH TEST TO DECIDE WHETHER THE TEST SHOULD
2430                            ;       BE RUN BASED ON THE DEVICE TYPE AND THE TEST MODE SELECTED IN
2431                            ;       THE STARTUP QUESTIONS. A TEST SELECT MASK SUPPLIED BY THE TEST
2432                            ;       IS COMPARED WITH A CONTROL MASK SET UP BY THE INITIALISATION
2433                            ;       ROUTINE TO DECIDE WHETHER THE TEST IS RUN.
2434                            ;
2435                            ;       IF THE SOFTWARE FAULT INSERTION FLAG SFI IS SET, ALL NON-SPECIFIC
2436                            ;       TESTS ARE SELECTED.
2437                            ;
2438                            ; INPUTS:
2439                            ;
2440                            ;       THE LOCATION FOLLOWING THE SUBROUTINE CALL CONTAINS THE TEST
2441                            ;       SELECT MASK IN THE FOLLOWING FORMAT :
2442                            ;
2443                            ;               BIT 0 IS SET IF DIGITAL INPUT MODULES ARE TO BE TESTED
2444                            ;               BIT 1 IS SET IF DIGITAL OUTPUT MODULES ARE TO BE TESTED
2445                            ;               BIT 2 IS SET IF ANALOGUE INPUT MODULES ARE TO BE TESTED
2446                            ;               BIT 3 IS SET IF ANALOGUE OUTPUT MODULES ARE TO BE TESTED
2447                            ;               BIT 4 IS SET IF FIVE CHANNEL COUNTER ARE TO BE TESTED
2448                            ;               BITS 5, 6 AND 7 ARE UNUSED
2449                            ;               BIT 8 IS SET TO INDICATE A BASIC INTERNAL LOGIC TEST
2450                            ;               BIT 9 IS SET FOR FIELD INPUT/OUTPUT TESTS
2451                            ;               BIT 10 IS SET FOR LOOPBACK TESTS
2452                            ;               BIT 11 IS SET FOR ANALOGUE INPUT/OUTPUT TESTS USED BY
2453                            ;               MANUFACTURING AND FIELD SERVICE
2454                            ;               BIT 12 IS SET FOR SPECIFICALLY SELECTABLE TESTS
2455                            ;               BITS 13, 14 AND 15 ARE UNUSED
2456                            ;
2457                            ;       THE SECOND LOCATION FOLLOWING THE SUBROUTINE CALL CONTAINS THE
2458                            ;       ADDRESS OF A TEST HEADER MESSAGE TO BE PRINTED IF THE TEST IS
2459                            ;       SELECTED AND THE USER 'PNT' FLAG IS SELECTED.
2460                            ;
2461                            ; IMPLICIT INPUTS:
2462                            ;
2463                            ;       CONMSK - TEST CONTROL MASK SET UP BY INIT CODE AT THE BEGINING
2464                            ;                OF EACH SUBPASS. THE FORMAT IS AS FOLLOWS :
2465                            ;
2466                            ;                       BIT 0 IS SET IF UUT IS DIGITAL INPUT
2467                            ;                       BIT 1 IS SET IF UUT IS DIGITAL OUTPUT
2468                            ;                       BIT 2 IS SET IF UUT IS ANALOGUE INPUT
2469                            ;                       BIT 3 IS SET IF UUT IS ANALOGUE OUTPUT
2470                            ;                       BIT 4 IS SET IF UUT IS FIVE CHANNEL COUNTER
2471                            ;                       BITS 5, 6 AND 7 ARE UNUSED
2472                            ;                       BIT 8 IS ALWAYS SET TO SELECT BASIC INTERNAL
2473                            ;                       LOGIC TESTS
2474                            ;                       BIT 9 SET TO SELECT FIELD INPUT/OUTPUT TESTS
2475                            ;                       BIT 10 IS SET IF LOOPBACK TESTING IS SELECTED
2476                            ;                       AND ALLOWED FOR CURRENT UUT
2477                            ;                       BIT 11 IS SET IF MANUFACTURING HAVE SELECTED LOOPBACK
2478                            ;                       AND I/O TESTS
2479                            ;                       BIT 12 IS SET IF A SPECIFICALLY SELECTABLE TEST
2480                            ;                       IS CHOSEN
```

```
                        ;                       BITS 13, 14 AND 15 ARE UNUSED
                        ;
                        ;       SF1 - IF THIS IS NON ZERO, ALL TESTS ARE SELECTED.
                        ;
                        ; OUTPUTS:
                        ;
                        ;       IF THE TEST MATCHES THE DEVICE TYPE AND TEST MODE CHOSEN VIA THE
                        ;       STARTUP QUESTIONS, THE ROUTINE EXITS WITH THE CARRY BIT CLEAR.
                        ;       IN THIS CASE, IF THE 'PNT' FLAG IS SELECTED, THE TEST HEADER IS
                        ;       PRINTED AND THE FLAGS TSTFLG AND TSUFLG ARE SET TO SHOW THAT A
                        ;       TEST HAS BEEN SELECTED.
                        ;
                        ;       IF THE TEST IS NOT SELECTED, THE CARRY BIT IS SET AND, IF THE
                        ;       'PNT' FLAG IS SELECTED, A MESSAGE 'TEST DISABLED BY STARTUP
                        ;       QUESTIONS' IS PRINTED.
                        ;
                        ;       IF THE TEST IS NOT SPECIFICALLY SELECTABLE (BIT 11 IS CLEAR IN
                        ;       THE TEST SELECT MASK), BIT 11 IS CLEARED IN THE CONTROL MASK TO
                        ;       PREVENT SUBSEQUENT SPECIFICALLY SELECTABLE TESTS FROM BEING RUN.
                        ;
                        ;       R0, R1 AND R2 ARE CORRUPTED.
                        ;
                        ; IMPLICIT OUTPUTS:
                        ;
                        ;       NONE.
                        ;
                        ; SUBORDINATE ROUTINES USED:
                        ;
                        ;       NONE.
                        ;
                        ; FUNCTIONAL SIDE EFFECTS:
                        ;
                        ;       NONE.
                        ;
                        ; CALLING SEQUENCE:
                        ;
                        ;       FIXED PARAMETERS FOLLOW THE SUBROUTINE CALL :
                        ;
                        ;       EG.     CALL    SELECT
                        ;               777                     ; TEST SELECT MASK (BASIC TEST
                        ;                                       ; ON ALL DEVICE TYPES
                        ;               TNAME                   ; ADDRESS OF TEST HEADER
                        ;               BCS     TSTEND          ; BRANCH IF TEST NOT SELECTED
                        ;
                        ;--
015006                  SELECT::
015006  017601  000000          MOV     @(SP),R1                ; SAVE TEST SELECT MASK
015012  062716  000002          ADD     #2,(SP)                 ; JUMP OVER THE ARGUEMENT
015016  017602  000000          MOV     @(SP),R2                ; SAVE TEST HEADER ADDRESS
015022  062716  000002          ADD     #2,(SP)                 ; JUMP OVER THE ARGUEMENT

015026                          RFLAGS  R0                      ; READ OPERATOR FLAGS INTO R0

015030  032701  010000          BIT     #10000,R1               ; SPECIFICALLY SELECTABLE TEST ?
015034  001003                  BNE     10$                     ; IF YES, BRANCH
```

```
2538 015036  042737  010000  002702          BIC     #10000,CONMSK           ; ELSE PREVENT SUCH TESTS
2539 015044  005737  002500          10$:    TST     SFI                     ; SOFTWARE FAULT INSERTION ?
2540 015050  001007                          BNE     20$                     ; IF YES, SELECT THE TEST
2541 015052  130137  002702                  BITB    R1,CONMSK               ; IS DEVICE THE CORRECT TYPE ?
2542 015056  001426                          BEQ     30$                     ; IF NOT, BRANCH
2543 015060  105001                          CLRB    R1                      ; IF YES, DISCARD LOW BYTE
2544 015062  030137  002702                  BIT     R1,CONMSK               ; AND CHECK TEST TYPE
2545 015066  001422                          BEQ     30$                     ; IF WRONG, BRANCH
2546
2547                                 ; TEST IS SELECTED
2548
2549 015070  012737  000001  002634  20$:    MOV     #1,TSTFLG               ; FLAG THAT TEST IS SELECTED
2550 015076  012737  000001  002636          MOV     #1,TSUFLG               ;
2551
2552 015104  032700  001000                  BIT     #PNT,R0                 ; PRINT TEST HEADER ?
2553 015110  001425                          BEQ     50$                     ; IF NOT, EXIT (CARRY IS CLEAR)
2554 015112                                  PRINTF  R2                      ; ELSE PRINT THE HEADER
2555 015130  000241                          CLC                             ; CLEAR THE CARRY
2556 015132  000414                          BR      50$                     ; AND EXIT
2557
2558                                 ; TEST IS NOT SELECTED
2559
2560 015134  032700  001000          30$:    BIT     #PNT,R0                 ; PRINT TEST HEADER ?
2561 015140  001410                          BEQ     40$                     ; IF NOT, EXIT
2562 015142                                  PRINTF  #TNA                    ; ELSE PRINT 'NOT APPLICABLE'
2563 015162  000261                  40$:    SEC                             ; SET THE CARRY BIT
2564
2565 015164  000207                  50$:    RETURN                          ; COMMON RETURN
2566
2567                                         .NLIST  BEX
2568 015166      045     101     040  TNA:    .ASCIZ  /%A  TEST DISABLED BY STARTUP QUESTIONS%N/
2569                                         .LIST   BEX
2570                                         .EVEN
```

```
                                ; SUBROUTINE INSERT - SUBROUTINE TO FORCE ERROR PRINTOUTS
                                ;
                                ;++
                                ; FUNCTIONAL DESCRIPTION:
                                ;
                                ;       THIS SUBROUTINE CAN BE USED TO FORCE PRINTOUT OF ERROR MESSAGES
                                ;       FOR QUALITY CHECKING. IF THE FLAG 'SFI' IS NONE ZERO, THE BRANCH
                                ;       INSTRUCTION FOLLOWING THE SUBROUTINE CALL IS SKIPPED OVER,
                                ;       CAUSING THE ERROR MESSAGE TO BE PRINTED. IF 'SFI' IS SET TO 1,
                                ;       THE ADDRESS OF THE SUBROUTINE CALL IS COMPARED WITH THAT OF THE
                                ;       LAST CALL AND, IF IT HAS NOT CHANGED, THE MESSAGE IS NOT
                                ;       PRINTED.
                                ;
                                ; INPUTS:
                                ;
                                ;       SFI - IF ZERO, THE ROUTINE DOES NOTHING.
                                ;             IF ONE, ERROR MESSAGES ARE PRINTED ONCE.
                                ;             IF ANY OTHER VALUE, ERROR MESSAGES ARE ALWAYS PRINTED.
                                ;
                                ; IMPLICIT INPUTS:
                                ;
                                ;       SEE CALLING SEQUENCE.
                                ;
                                ; OUTPUTS:
                                ;
                                ;       LASTFA - IF THE ERROR MESSAGE IS TO BE PRINTED ONCE ONLY, LASTFA
                                ;                IS LOADED WITH THIS SUBROUTINE RETURN ADDRESS.
                                ;
                                ; IMPLICIT OUTPUTS:
                                ;
                                ;       NONE.
                                ;
                                ; SUBORDINATE ROUTINES USED:
                                ;
                                ;       NONE.
                                ;
                                ; FUNCTIONAL SIDE EFFECTS:
                                ;
                                ;       IF 'SFI' IS NONE ZERO, THE ROUTINE RETURN ADDRESS IS INCREMENTED
                                ;       BY ONE WORD.
                                ;
                                ; CALLING SEQUENCE:
                                ;
                                ;       A ONE WORD BRANCH INSTRUCTION MUST FOLLOW THE SUBROUTINE CALL
                                ;       BEFORE THE ERROR PRINT CALL.
                                ;
                                ;       EG.     CMP     BAD,GOOD        ; REGISTER CORRECT ?
                                ;               CALL    INSERT          ; SKIP BRANCH IF SFI FLAG SET
                                ;               BEQ     10$             ; BRANCH IF REGISTER CORRECT
                                ;               ERROR                   ; ELSE PRINT OUT ERROR MESSAGE
                                ;
                                ;--
                                
015240                          INSERT::
015240  106746                          MFPS    -(SP)                   ; SAVE CONDITION CODES
015242  023727  002500  000001          CMP     SFI,#1                  ; INSERT FAULTS ?
015250  103413                          BLO     20$                     ; IF NOT, BRANCH
```

```
2629 015252  001007                          BNE     10$                     ; IF ALWAYS, BRANCH
2630 015254  026637  000002  015304          CMP     2(SP),LASTFA            ; IS THIS FAULT ALREADY PRINTED?
2631 015262  001406                          BEQ     20$                     ; IF YES, EXIT
2632 015264  016637  000002  015304          MOV     2(SP),LASTFA            ; ELSE, SAVE FAULT ADDRESS
2633 015272  062766  000002  000002  10$:    ADD     #2,2(SP)                ; SKIP BRANCH TO FORCE PRINTOUT
2634 015300  106426                  20$:    MTPS    (SP)+                   ; RESTORE CONDITION CODES
2635 015302  000207                          RETURN                          ; AND RETURN
2636
2637 015304  000000                  LASTFA: .WORD   0                       ; ADDRESS OF ROUTINE CALL
2638
```

```
                        ; SUBROUTINE CONFIG - ROUTINE TO FIND THE DEVICE CONFIGURATION AUTOMATICALLY.
                        ;
                        ;++
                        ; FUNCTIONAL DESCRIPTION:
                        ;
                        ;       THIS ROUTINE SEARCHES THE IDV/IAV-11 ADDRESS RANGE (171000-171770) FOR A
                        ;       RESPONDING ADDRESS. FOR THE FIRST ADDRESS ON A 4 WORD BOUNDARY (171XX0)
                        ;       WHICH DOES NOT CAUSE AN NXM TRAP, AN ENTRY IS MADE IN THE HARDWARE
                        ;       P TABLE FOR UNIT L$LUN. FOR ADDRESSES OVER 171400, A VECTOR IS
                        ;       ASSUMED BASED ON THE LOW 8 BITS OF THE DEVICE ADDRESS.
                        ;
                        ;       EACH CALL TO THE ROUTINE WILL CAUSE ONE MORE UNIT TO BE SEARCHED FOR.
                        ;       IF THE UNIT FOUND IS THE LAST IN THE IDV/IAV-11 RANGE, L$UNITS IS SET TO
                        ;       THE TOTAL NUMBER OF UNITS. IF NO UNIT IS FOUND, THE PROGRAM EXITS
                        ;       WITH THE CARRY BIT SET AND L$UNIT SET TO L$LUN.
                        ;
                        ;       MODULES WITH MODE REGISTER CONTENTS NOT CORRESPONDING TO AN IDV/IAV11
                        ;       ARE SET UP, BUT CAUSE AN 'UNIDENTIFIED MODULE' MESSAGE TO BE OUTPUT.
                        ;
                        ; INPUTS:
                        ;
                        ;       L$LUN - NUMBER FOR THE NEXT UNIT FOUND.
                        ;
                        ; IMPLICIT INPUTS:
                        ;
                        ;       STADD MUST BE SET TO 171000 PRIOR TO THE FIRST CALL TO INITIALISE
                        ;       THE SEARCH AREA.
                        ;
                        ;       ADDRESSES DEFINED BY "IXSTA" TO "IXEND" OF THE I/O PAGE ARE READ.
                        ;
                        ;       THE WORD L$LUN+2 AFTER LABEL GPADD MUST CONTAIN THE PARAMETER
                        ;       TABLE ADDRESS FOR THE UNIT.
                        ;
                        ; OUTPUTS:
                        ;
                        ;       THE HARDWARE P TABLE AND L$UNIT ARE SET UP TO INCLUDE UP TO 16
                        ;       UNITS FOUND IN THE ABOVE ADDRESS RANGE.
                        ;
                        ;       A MESSAGE 'UNIDENTIFIED MODULE FOUND AT ADDRESS nnnnnn' MAY BE
                        ;       PRINTED.
                        ;
                        ; IMPLICIT OUTPUTS:
                        ;
                        ;       IF THE UNIT FOUND IS THE LAST IN THE IDV/IAV-11 RANGE, L$UNIT IS SET UP
                        ;       TO CONTAIN THE TOTAL NUMBER OF UNITS FOUND (L$LUN+1).
                        ;
                        ;       IF NO MORE UNITS ARE FOUND, THE CARRY BIT IS SET AND L$LUN IS
                        ;       PLACED IN L$UNIT.
                        ;
                        ; SUBORDINATE ROUTINES USED:
                        ;
                        ;       NXM - NON EXISTANT MEMORY TRAP ROUTINE.
                        ;       WRDY - ROUTINE TO WAIT FOR OPERATOR TO TYPE 'RETURN'.
                        ;
                        ; FUNCTIONAL SIDE EFFECTS:
                        ;
                        ;       NXMFLG MAY BE SET.
```

```
                                        ;
                                        ; CALLING SEQUENCE:
                                        ;
                                        ;   EG. JSR     PC,CONFIG
                                        ;       BCS     INIUUT                  ; IF NO MORE UNITS, START AGAIN
                                        ;
                                        ;--

015306                                  CONFIG::
015306  010146                                  MOV     R1,-(SP)                ; SAVE R1 TO R4
015310  010246                                  MOV     R2,-(SP)                ;
015312  010346                                  MOV     R3,-(SP)                ;
015314  010446                                  MOV     R4,-(SP)                ;

015316  013702  015572                          MOV     STADD,R2                ; START SEARCH FROM THIS ADDRESS
015322  012737  000001  015574                  MOV     #1,NOUNIT               ; ASSUME NO UNIT WILL BE FOUND
015330                                          SETVEC  #4,#NXM,#340            ; SET UP NXM TRAP

015356  005037  002470                  10$:    CLR     NXMFLG                  ; CLEAR THE NXM FLAG
015362  005712                                  TST     (R2)                    ; CHECK THE ADDRESS
015364  005737  002470                          TST     NXMFLG                  ; ANYTHING THERE ?
015370  001006                                  BNE     20$                     ; IF NOT, TRY NEXT DEVICE ADDRESS
015372  005737  015574                          TST     NOUNIT                  ; IF YES, IS IT THE 2ND ONE THIS CALL ?
015376  001422                                  BEQ     30$                     ; IF YES, SET UP THE P TABLE
015400  005037  015574                          CLR     NOUNIT                  ; IF IT'S THE FIRST, FLAG UNIT FOUND
015404  010201                                  MOV     R2,R1                   ; AND SAVE THE ADDRESS

015406  062702  000010                  20$:    ADD     #10,R2                  ; GET THE NEXT ADDRESS
015412  020227  171770                          CMP     R2,#IXEND               ; OUT OF THE IXV11 RANGE ?
015416  003757                                  BLE     10$                     ; IF NOT, GO BACK
015420  012702  171000                          MOV     #IXSTA,R2               ; ELSE START AGAIN NEXT CALL
015424  013737  002074  002012                  MOV     L$LUN,L$UNIT            ; SAVE THE UNIT NUMBER
015432  005737  015574                          TST     NOUNIT                  ; WERE ANY UNITS FOUND ?
015436  001041                                  BNE     70$                     ; IF NOT, EXIT
015440  005237  002012                          INC     L$UNIT                  ; ELSE WE HAVE THE LAST UNIT

015444  021127  177400                  30$:    CMP     (R1),#177400            ; IS MODULE ID OK ?
015450  103413                                  BLO     40$                     ; IF YES, BRANCH
015452                                          PRINTF  #C01,R1                 ; ELSE PRINT 'UNIDENTIFIED'
015474  004737  014662                          JSR     PC,WRDY                 ; WAIT FOR OPERATOR TO TYPE 'RETURN'

015500  013700  002074                  40$:    MOV     L$LUN,R0                ; FORM OFFSET FOR LUN
015504  006300                                  ASL     R0                      ;
015506  016003  002554                          MOV     GPADD(R0),R3            ; GET P TABLE ADDRESS IN R3
015512  010123                                  MOV     R1,(R3)+                ; SAVE THE DEVICE ADDRESS

015514  042701  177000                          BIC     #177000,R1              ; CALCULATE THE VECTOR ADDRESS
015520  032701  000400                          BIT     #400,R1                 ; ADDRESS OVER 171400 ?
015524  001001                                  BNE     50$                     ; IF YES, BRANCH
015526  005001                                  CLR     R1                      ; ELSE ZERO THE VECTOR
015530  010123                  50$:    MOV     R1,(R3)+                ; AND SAVE IT
015532  001402                                  BEQ     60$                     ; IF 0, BRANCH
015534  012701  000200                          MOV     #200,R1                 ; ELSE SET UP DEFAULT PRIORITY
015540  010123                  60$:    MOV     R1,(R3)+                ; SAVE THE PRIORITY

015542  010237  015572          70$:    MOV     R2,STADD                ; SAVE THE NEXT SEARCH ADDRESS
015546                                  CLRVEC  #4                      ; RESTORE THE NXM TRAP CATCHER
```

GLOBAL SUBROUTINES SECTION

```
2754 015554  012604                          MOV     (SP)+,R4                ; RESTORE R4 TO R1
2755 015556  012603                          MOV     (SP)+,R3                ;
2756 015560  012602                          MOV     (SP)+,R2                ;
2757 015562  012601                          MOV     (SP)+,R1                ;
2758 015564  006237  015574                  ASR     NOUNIT                  ; IF NO UNIT, SET THE CARRY BIT
2759 015570  000207                          RTS     PC                      ; AND RETURN
2760
2761 015572  171000                  STADD:: .WORD   IXSTA                   ; START ADDRESS OF SEARCH AREA
2762 015574  000000                  NOUNIT: .WORD   0                       ; SET TO SHOW NO UNIT FOUND
2763
2764                                          .NLIST  BEX
2765 015576     045     116     045  C01:    .ASCIZ  /%N%AUNIDENTIFIED MODULE FOUND AT ADDRESS %O6%A./
2766                                          .LIST   BEX
2767                                          .EVEN
```

```
2769                                    ; SUBROUTINE CONPRI - SUBROUTINE TO PRINT A CONFIGURATION TABLE.
2770
2771                                    ;++
2772                                    ; FUNCTIONAL DESCRIPTION:
2773                                    ;
2774                                    ;       THIS ROUTINE PRINTS A CONFIGURATION TABLE OF ALL UNITS LISTED IN
2775                                    ;       THE HARDWARE PARAMETER TABLE. LISTED ARE THE MODE REGISTER ADDRESS,
2776                                    ;       VECTOR AND  DEVICE TYPE.
2777                                    ;
2778                                    ; INPUTS:
2779                                    ;
2780                                    ;       NONE.
2781                                    ;
2782                                    ; IMPLICIT INPUTS:
2783                                    ;
2784                                    ;       THE HARDWARE P TABLE.
2785                                    ;
2786                                    ;       IF NO UNITS ARE CONFIGURED, THE GLOBAL MESSAGE LABELLED "NODEV"
2787                                    ;       IS PRINTED.
2788                                    ;
2789                                    ; OUTPUTS:
2790                                    ;
2791                                    ;       THE CONFIGURATION TABLE IS PRINTED.
2792                                    ;
2793                                    ; IMPLICIT OUTPUTS:
2794                                    ;
2795                                    ;       IF NO UNITS ARE CONFIGURED (L$UNIT = 0), A MESSAGE 'NO DEVICES
2796                                    ;       FOUND' IS PRINTED.
2797                                    ;
2798                                    ; SUBORDINATE ROUTINES USED:
2799                                    ;
2800                                    ;       NXM - NON EXISTANT MEMORY TRAP ROUTINE.
2801                                    ;       CRLF - LINE FEED PRINT ROUTINE.
2802                                    ;
2803                                    ; FUNCTIONAL SIDE EFFECTS:
2804                                    ;
2805                                    ;       REGISTERS R1 TO R5 ARE CORRUPTED.
2806                                    ;
2807                                    ; CALLING SEQUENCE:
2808                                    ;
2809                                    ;       JSR     PC,CONPRI
2810                                    ;
2811                                    ;--
2812
2813 015656                            CONPRI::
2814 015656                                    SETVEC  #4,#NXM,#340            ; CATCH REFERENCES TO NXM
2815 015704  005737  002012                    TST     L$UNIT                  ; ANY UNITS CONFIGURED ?
2816 015710  001012                            BNE     10$                     ; IF YES, BRANCH
2817 015712                                    PRINTF  #NODEV                  ; ELSE PRINT 'NO DEVICES FOUND'
2818 015732  000137  016452                    JMP     CONEX                   ; AND EXIT
2819
2820 015736                            10$:    PRINTF  #CP1                    ; PRINT CONFIGURATION HEADER
2821 015756                                    PRINTF  #CP1A                   ;
2822 015776                                    PRINTF  #CP1B                   ;
2823 016016                                    PRINTF  #CP1C                   ;
2824 016036  005001                            CLR     R1                      ; START WITH FIRST UNIT
2825 016040  010100                    20$:    MOV     R1,R0                   ; FORM PARAMETER TABLE OFFSET
```

```
2826 016042  006300                        ASL     R0                      ;
2827 016044  016002  002554                MOV     GPADD(R0),R2            ; GET THE UNIT P TABLE
2828 016050  012203                        MOV     (R2)+,R3                ; SAVE THE UNIT ADDRESS
2829 016052                                PRINTF  #CP2,R1,R3              ; PRINT UNIT NO. AND ADDRESS
2830 016076  005722                        TST     (R2)+                   ; IS THERE A VALID VECTOR ?
2831 016100  001413                        BEQ     30$                     ; IF NOT, BRANCH
2832 016102                                PRINTF  #CP3,-2(R2)             ; ELSE PRINT THE VECTOR
2833 016126  000410                        BR      40$                     ;
2834 016130                        30$:    PRINTF  #CP4                    ; PRINT 'NONE'
2835
2836 016150  005037  002470        40$:    CLR     NXMFLG                  ; GET READY FOR NXM TRAP
2837 016154  112305                        MOVB    (R3)+,R5                ; GET MODE REGISTER LOW BYTE
2838 016156  111304                        MOVB    (R3),R4                 ; AND HIGH BYTE
2839 016160  042704  177400                BIC     #177400,R4              ; DISCARD REGISTER HIGH BYTE
2840 016164  005737  002470                TST     NXMFLG                  ; WAS THERE AN NXM TRAP ?
2841 016170  001421                        BEQ     50$                     ; IF NOT, BRANCH
2842 016172                                PRINTF  #CP10                   ; ELSE PRINT UNKNOWN ID/MODE
2843 016212                                PRINTF  #CP10A                  ; AND UNKNOWN MODULE TYPE
2844 016232  000501                        BR      110$                    ; AND SEE IF LOOPED
2845
2846 016234                        50$:    PRINTF  #CP5,R4,R5              ; PRINT OUT ID/MODE
2847
2848 016260  022704  000300                CMP     #300,R4                 ; IS MODULE A FIVE CHA. COUNTER ?
2849 016264  101014                        BHI     55$                     ; IF NOT, BRANCH
2850 016266  012703  017222                MOV     #DCNT,R3                ; ELSE SAVE 'FIVE CHA. COUNTER' STRING
2851 016272                                PRINTF  #CP6,R3                 ; PRINTOUT DEVICE TYPE
2852 016314  000450                        BR      110$                    ;
2853 016316  020427  000037        55$:    CMP     R4,#37                  ; IS MODULE DIGITAL INPUT ?
2854 016322  101003                        BHI     60$                     ; IF NOT, BRANCH
2855 016324  012703  017156                MOV     #DI,R3                  ; ELSE SAVE 'DIG. IN' STRING
2856 016330  000421                        BR      90$                     ; GO TO PRINT MODULE TYPE
2857 016332  020427  000077        60$:    CMP     R4,#77                  ; IS MODULE DIGITAL OUTPUT ?
2858 016336  101003                        BHI     70$                     ; IF NOT, BRANCH
2859 016340  012703  017167                MOV     #DO,R3                  ; ELSE SAVE 'DIG OUT' STRING
2860 016344  000413                        BR      90$                     ; GO TO PRINT MODULE TYPE
2861 016346  020427  000177        70$:    CMP     R4,#177                 ; IS MODULE ANALOGUE INPUT ?
2862 016352  101003                        BHI     80$                     ; IF NOT, BRANCH
2863 016354  012703  017200                MOV     #AI,R3                  ; ELSE SAVE 'AN IN' STRING
2864 016360  000405                        BR      90$                     ; GO TO PRINT MODULE TYPE
2865 016362  020427  000277        80$:    CMP     R4,#277                 ; IS MODULE ANALOGUE OUTPUT ?
2866 016366  101013                        BHI     100$                    ; IF NOT, BRANCH
2867 016370  012703  017211                MOV     #AO,R3                  ; ELSE SAVE 'AN OUT' STRING
2868 016374                        90$:    PRINTF  #CP6,R3                 ; PRINT OUT DEVICE TYPE
2869 016416                        100$:   PRINTF  #CP7                    ; PRINT 'CANNOT BE TESTED'
2870
2871
2872 016436  005201                110$:   INC     R1                      ; PREPARE FOR NEXT UNIT
2873 016440  020137  002012                CMP     R1,L$UNIT               ; ALL UNITS DONE ?
2874 016444  002002                        BGE     CONEX                   ; IF YES, EXIT
2875 016446  000137  016040                JMP     20$                     ; ELSE DISPLAY THE NEXT
2876 016452  004737  014634        CONEX:  JSR     PC,CRLF                 ; PRINT A LINE FEED
2877 016456                                CLRVEC  #4                      ; RESTORE SUPERVISOR NXM TRAP CATCHER
2878 016464  000207                        RTS     PC
2879
2880                                       .NLIST  BEX
2881
2882 016466     045     116     045  CP1:    .ASCIZ  \%N%S24%AIDV/IAV-11 MODULE CONFIGURATION.\
```

L5


```
2883 016537  045  116  045  CP1A:   .ASCIZ  /≤N≤S24≤A---------- ----  /
2884 016611  045  116  062  CP1B:   .ASCIZ  \≤N2≤AUNIT  ADDRESS  VECTOR  ID/MODE    MODULE       COMMENT≤N\
2885 016707  045  123  061  CP1C:   .ASCIZ  \≤S15≤AASSUMED≤S13≤ATYPE≤N\
2886 016741  045  116  045  CP2:    .ASCIZ  \≤N≤03≤09\
2887 016752  045  123  065  CP3:    .ASCIZ  \≤S5≤03\
2888 016761  045  123  064  CP4:    .ASCIZ  \≤S4≤ANONE\
2889 016773  045  123  063  CP5:    .ASCIZ  \≤S3≤03≤A/≤03\
2890 017010  045  123  063  CP6:    .ASCIZ  \≤S3≤T\
2891 017016  045  101  040  CP7:    .ASCIZ  \≤A   * CANNOT BE TESTED WITH THIS DIAG *\
2892 017066  045  101  040  CP8:    .ASCIZ  \≤A    YES\
2893 017077  045  101  040  CP9:    .ASCIZ  \≤A     NO\
2894 017110  045  101  040  CP10:   .ASCIZ  \≤A    UNKNOWN\
2895 017125  045  101  040  CP10A:  .ASCIZ  \≤A    UNKNOWN \
2896 017143  045  117  071  CP11:   .ASCIZ  \≤09\
2897 017147  045  123  065  CP12:   .ASCIZ  \≤S5≤03\
2898
2899 017156  104  111  107  DI:     .ASCIZ  /DIG. IN /
2900 017167  104  111  107  DO:     .ASCIZ  /DIG. OUT/
2901 017200  101  116  056  AI:     .ASCIZ  /AN.  IN /
2902 017211  101  116  056  AO:     .ASCIZ  /AN.  OUT/
2903 017222  106  111  126  DCNT:   .ASCIZ  /FIVE CHA. COUNTER/
2904
2905                        .LIST   BEX
2906                        .EVEN
```

```
                                        ; SUBROUTINE DACON - DIGITAL TO ANALOGUE CONVERSION ROUTINE.
                                        ;
                                        ;++
                                        ; FUNCTIONAL DESCRIPTION:
                                        ;
                                        ;       THIS CONVERTS A 12 BIT DIGITAL PATTERN INTO A 2 WORD ANALOGUE
                                        ;       OUTPUT VALUE.
                                        ;
                                        ; INPUTS:
                                        ;
                                        ;       MODE : 0 = UNIPOLAR (0 TO 10 VOLTS)     BINARY CODED
                                        ;              1 = BIPOLAR  (-10 TO +10 VOLTS)  OFFSET BINARY CODED
                                        ;              2 = 0 - 20 mA                    BINARY CODED
                                        ;              3 = 4 - 20 mA                    BINARY CODED
                                        ;
                                        ;       GAIN : 0 - 7 = GAINS 1,2,5,10,20,50,100,200
                                        ;
                                        ;       R1   : 12 BIT INPUT PATTERN.
                                        ;
                                        ; IMPLICIT INPUTS:
                                        ;
                                        ;       VITAB AND ITAB : DIGITAL/ANALOGUE CONVERSION TABLES.
                                        ;
                                        ; OUTPUTS:
                                        ;
                                        ;       R1 - MILLIVOLTS (MODES 0 AND 1)
                                        ;            MICROAMPS  (MODES 2 AND 3)
                                        ;
                                        ;       R2 - MICROVOLTS (MODES 0 AND 1)
                                        ;            NANOAMPS   (MODES 2 AND 3)
                                        ;
                                        ; IMPLICIT OUTPUTS:
                                        ;
                                        ;       NONE.
                                        ;
                                        ; SUBORDINATE ROUTINES USED:
                                        ;
                                        ;       NONE.
                                        ;
                                        ; FUNCTIONAL SIDE EFFECTS:
                                        ;
                                        ;       NONE.
                                        ;
                                        ; CALLING SEQUENCE:
                                        ;
                                        ;  EG.  MOV     #1,MODE                    ; BIPOLAR CONVERSION
                                        ;       MOV     #6,GAIN                    ; GAIN = 100
                                        ;       MOV     #7777,R1                   ; ALL BITS SET
                                        ;       JSR     PC,DACON
                                        ;
                                        ;--
                                        DACON::
017244  010346                                  MOV     R3,-(SP)                   ; SAVE R3
017246  010446                                  MOV     R4,-(SP)                   ; AND R4
017250  013703  002704                          MOV     MODE,R3                    ; GET MODE
017254  005703                                  TST     R3                         ; MODE 0 ?
```

```
2965 017256  001003                        BNE     10$                ; IF NOT, BRANCH
2966 017260  012700  002712                MOV     #VUPTAB,R0         ; GET CONVERSION TABLE FOR MODE 0
2967 017264  000414                        BR      PSDA               ;
2968
2969 017266  005303                 10$:   DEC     R3                 ; MODE 1 ?
2970 017270  001003                        BNE     20$                ; IF NOT, BRANCH
2971 017272  012700  003512                MOV     #VBPTAB,R0         ; GET CONVERSION TABLE FOR MODE 1
2972 017276  000407                        BR      PSDA
2973
2974 017300  005303                 20$:   DEC     R3                 ; MODE 2 ?
2975 017302  001003                        BNE     30$                ; IF NOT, BRANCH
2976 017304  012700  004312                MOV     #IOTAB,R0          ; GET CONVERSION TABLE FOR MODE 2
2977 017310  000402                        BR      PSDA               ;
2978
2979 017312  012700  005112          30$:   MOV     #I4TAB,R0          ; USE I4TAB FOR MODE 3
2980
2981 017316  013703  002706          PSDA:  MOV     GAIN,R3            ; GET THE GAIN
2982 017322  005303                 30$:   DEC     R3                 ; HAVE WE GOT THE CORRECT TABLE ?
2983 017324  100403                        BMI     40$                ; IF YES, BRANCH
2984 017326  062700  000060                ADD     #48.,R0            ; ELSE INCREASE THE TABLE OFFSET
2985 017332  000773                        BR      30$                ; AND TRY AGAIN
2986
2987 017334  011004                 40$:   MOV     (R0),R4            ; SAVE HIGH BIT VALUE
2988 017336  010103                        MOV     R1,R3              ; SAVE THE BIT PATTERN
2989 017340  005001                        CLR     R1                 ; CLEAR THE OUTPUT REGISTERS
2990 017342  005002                        CLR     R2                 ;
2991 017344  006303                        ASL     R3                 ; SHIFT OUT UNUSED BITS
2992 017346  006303                        ASL     R3                 ; ( BITS 15 - 12 )
2993 017350  006303                        ASL     R3                 ;
2994 017352  006303                        ASL     R3                 ;
2995
2996 017354  006303                 50$:   ASL     R3                 ; TEST A BIT
2997 017356  103011                        BCC     70$                ; IF CLEAR, BRANCH
2998 017360  066002  000030                ADD     24.(R0),R2         ; ELSE ADD IN LOW VALUE
2999 017364  020227  001750                CMP     R2,#1000.          ; OVERFLOW OF LOW WORD ?
3000 017370  002403                        BLT     60$                ; IF NOT, BRANCH
3001 017372  162702  001750                SUB     #1000.,R2          ; ELSE CARRY FROM LOW WORD
3002 017376  005201                        INC     R1                 ; TO HIGH WORD
3003 017400  061001                 60$:   ADD     (R0),R1            ; AND ADD IN HIGH VALUE
3004 017402  062700  000002          70$:   ADD     #2,R0              ; GET NEXT TABLE ENTRY
3005 017406  005703                        TST     R3                 ; ALL BITS PROCESSED ?
3006 017410  001361                        BNE     50$                ; IF NOT, DO MORE BITS
3007
3008 017412  023727  002704  000001  80$:   CMP     MODE,#1            ; BIPOLAR VOLTAGE CONVERSION ?
3009 017420  001007                        BNE     90$                ; IF NOT,BRANCH
3010 017422  160401                        SUB     R4,R1              ; ELSE MAKE BIPOLAR
3011 017424  002005                        BGE     90$                ; IF STILL POSITIVE, BRANCH
3012 017426  005702                        TST     R2                 ; DECIMAL PART ZERO ?
3013 017430  001403                        BEQ     90$                ; IF YES, BRANCH
3014 017432  162702  001750                SUB     #1000.,R2          ; ELSE BORROW FROM HIGH PART
3015 017436  005201                        INC     R1                 ;
3016
3017 017440  023727  002704  000003  90$:   CMP     MODE,#3            ; 4 - 20 mA MODE ?
3018 017446  001002                        BNE     100$               ; IF NOT, BRANCH
3019 017450  062701  007640                ADD     #4000.,R1          ; ELSE ADD IN BASE VALUE
3020
3021 017454  012604                 100$:  MOV     (SP)+,R4           ; RESTORE R4
```

```
   3022 017456  012603                    MOV     (SP)+,R3            ; AND R3
   3023 017460  000207                    RTS     PC                  ;
```

```
                                ; SUBROUTINE ADCON - ANALOGUE TO DIGITAL CONVERSION ROUTINE.
                                ;
                                ;++
                                ; FUNCTIONAL DESCRIPTION:
                                ;
                                ;       THIS CONVERTS A 2 WORD ANALOGUE VALUE INTO A 12 BIT DIGITAL OUTPUT
                                ;       PATTERN. THE INPUT IS ROUNDED UP OR DOWN TO THE NEAREST LSF VALUE.
                                ;
                                ; INPUTS:
                                ;
                                ;       MODE : 0 = UNIPOLAR (0 TO 10 VOLTS)     BINARY CODED
                                ;              1 = BIPOLAR  (-10 TO +10 VOLTS)  OFFSET BINARY CODED
                                ;              2 = 0 - 20 mA                    BINARY CODED
                                ;              3 = 4 - 20 mA                    BINARY CODED
                                ;
                                ;       GAIN : 0 - 7 = GAINS 1,2,5,10,20,50,100,200
                                ;
                                ;       R1 - MILLIVOLTS (MODES 0 AND 1)
                                ;            MICROAMPS  (MODES 2 AND 3)
                                ;
                                ;       R2 - MICROVOLTS (MODES 0 AND 1)
                                ;            NANOAMPS   (MODES 2 AND 3)
                                ;
                                ; IMPLICIT INPUTS:
                                ;
                                ;       VITAB AND ITAB : DIGITAL/ANALOGUE CONVERSION TABLES.
                                ;
                                ; OUTPUTS:
                                ;
                                ;       R1    : 12 BIT INPUT PATTERN.
                                ;
                                ; IMPLICIT OUTPUTS:
                                ;
                                ;       NONE.
                                ;
                                ; SUBORDINATE ROUTINES USED:
                                ;
                                ;       NONE.
                                ;
                                ; FUNCTIONAL SIDE EFFECTS:
                                ;
                                ;       NONE.
                                ;
                                ; CALLING SEQUENCE:
                                ;
                                ;  EG.  MOV     #1,MODE                 ; BIPOLAR CONVERSION
                                ;       MOV     #6,GAIN                 ; GAIN = 100
                                ;       MOV     #-4,R1                  ; -4.001 MILLIVOLTS
                                ;       MOV     #-1,R2                  ; IN R1/R2
                                ;       JSR     PC,ADCON
                                ;
                                ;--
017462                          ADCON::
017462  010346                          MOV     R3,-(SP)                ; SAVE R3
017464  010446                          MOV     R4,-(SP)                ; AND R4
017466  013703  002704                  MOV     MODE,R3                 ; GET MODE
```

```
3082 017472  005703                        TST     R3              ; MODE 0 ?
3083 017474  001003                        BNE     10$             ; IF NOT, BRANCH
3084 017476  012700  002712                MOV     #VUPTAB,R0      ; GET CONVERSION TABLE FOR MODE 0
3085 017502  000416                        BR      PSAD            ;
3086
3087 017504  005303                10$:    DEC     R3              ; MODE 1 ?
3088 017506  001003                        BNE     20$             ; IF NOT, BRANCH
3089 017510  012700  003512                MOV     #VBPTAB,R0      ; GET CONVERSION TABLE FOR MODE 1
3090 017514  000411                        BR      PSAD
3091
3092 017516  005303                20$:    DEC     R3              ; MODE 2 ?
3093 017520  001003                        BNE     30$             ; IF NOT, BRANCH
3094 017522  012700  004312                MOV     #IOTAB,R0       ; GET CONVERSION TABLE FOR MODE 2
3095 017526  000404                        BR      PSAD            ;
3096
3097 017530  012700  005112          30$:  MOV     #I4TAB,R0       ; USE I4TAB FOR MODE 3
3098 017534  162701  007640                SUB     #4000.,R1       ; AND SUBTRACT BASE VALUE
3099
3100 017540  013703  002706          PSAD: MOV     GAIN,R3         ; GET THE GAIN
3101 017544  005303                10$:    DEC     R3              ; HAVE WE GOT THE CORRECT TABLE ?
3102 017546  100403                        BMI     20$             ; IF YES, BRANCH
3103 017550  062700  000060                ADD     #48.,R0         ; ELSE INCREASE THE TABLE OFFSET
3104 017554  000773                        BR      10$             ; AND TRY AGAIN
3105
3106 017556  023727  002704  000001  20$:  CMP     MODE,#1         ; BIPOLAR MODE ?
3107 017564  001011                        BNE     30$             ; IF NOT, BRANCH
3108 017566  061001                        ADD     (R0),R1         ; ELSE CONVERT BIPOLAR TO UNIPOLAR
3109 017570  020127  023420                CMP     R1,#10000.      ; WAS PREVIOUS VALUE NEGATIVE ?
3110 017574  001005                        BNE     30$             ; IF NOT, BRANCH
3111 017576  005702                        TST     R2              ; IS DECIMAL PART ZERO ?
3112 017600  001403                        BEQ     30$             ; IF YES, BRANCH
3113 017602  062702  001750                ADD     #1000.,R2       ; ELSE BORROW FROM HIGH PART
3114 017606  005301                        DEC     R1              ;
3115
3116 017610  016003  000026          30$:  MOV     22.(R0),R3      ; GET ROUNDING VALUES FROM LOWEST
3117 017614  016004  000056                MOV     46.(R0),R4      ; SIGNIFICANT BIT
3118 017620  006203                        ASR     R3              ; DIVIDE BY 2
3119 017622  103002                        BCC     40$             ; IF NO CARRY SKIP NEXT COMMAND
3120 017624  062704  001000                ADD     #1000,R4        ; ADD CARRY
3121 017630  006204                40$:    ASR     R4              ; DIDIDE BY 2
3122 017632  060402                        ADD     R4,R2           ; ROUND UP THE INPUT VALUE
3123 017634  020227  001750                CMP     R2,#1000.       ; LOWER PART IS MODULO 1000
3124 017640  002403                        BLT     50$             ;
3125 017642  162702  001750                SUB     #1000.,R2       ; IF OVERFLOW, CARRY OVER TO
3126 017646  005201                        INC     R1              ; HIGH PART
3127 017650  060301                50$:    ADD     R3,R1           ; ADD IN HIGH PART OF ROUNDING FACTOR
3128 017652  012703  000020                MOV     #20,R3          ; INITIALISE WORKING REGISTER
3129
3130 017656  020110                60$:    CMP     R1,(R0)         ; COMPARE HIGH VALUE WITH TABLE ENTRY
3131 017660  002415                        BLT     90$             ; IF LESS, DON'T SET BIT
3132 017662  003003                        BGT     70$             ; IF MORE, SET THE BIT
3133 017664  020260  000030                CMP     R2,24.(R0)      ; OTHERWISE, MUST CHECK THE LOW VALUE
3134 017670  002411                        BLT     90$             ; IF LESS, DON'T SET THE BIT
3135
3136 017672  166002  000030          70$:  SUB     24.(R0),R2      ; SUBTRACT THE TABLE ENTRIES
3137 017676  002003                        BGE     80$             ; BRANCH IF NO BORROW NEEDED
3138 017700  062702  001750                ADD     #1000.,R2       ; ELSE ADD TO LOW WORD
```

```
3139 017704  005301                      DEC     R1                      ; FROM HIGH WORD
3140 017706  161001              80$:    SUB     (R0),R1                 ; AND LOW WORDS
3141 017710  052703  000001              BIS     #1,R3                   ; AND SET THE OUTPUT BIT
3142 017714  062700  000002      90$:    ADD     #2,R0                   ; AND NEXT TABLE ENTRY
3143 017720  006303                      ASL     R3                      ; READY FOR NEXT BIT
3144 017722  103355                      BCC     60$                     ; IF 12 BITS NOT DONE, GO BACK
3145
3146 017724  006203              100$:   ASR     R3                      ; GET THE PATTERN AGAIN
3147 017726  010301                      MOV     R3,R1                   ; SET UP OUTPUT REGISTER
3148 017730  012604                      MOV     (SP)+,R4                ; RESTORE R4
3149 017732  012603                      MOV     (SP)+,R3                ; AND R3
3150 017734  000207                      RTS     PC                      ;
```

```
                        ; SUBROUTINE DECIN - SIGNED DECIMAL INPUT ROUTINE.
                        ;
                        ;++
                        ; FUNCTIONAL DESCRIPTION:
                        ;
                        ;       THIS SOLICITS A SIGNED DECIMAL NUMBER FROM THE OPERATOR.
                        ;
                        ; INPUTS:
                        ;
                        ;       PADD - THE ADDRESS OF THE PROMPT STRING FOR THE INPUT.
                        ;
                        ;       THE OPERATOR IS PROMPTED FOR A NUMBER WHICH CAN BE UP TO 10 DIGITS
                        ;       LONG INCLUDING AN OPTIONAL + OR - SIGN AND DECIMAL POINT. THE
                        ;       STRING SHOULD BE IN 'PRINT' FORMAT ( IE..ASCIZ /#N#AINPUT VALUE/ ).
                        ;
                        ; IMPLICIT INPUTS:
                        ;
                        ;       NONE.
                        ;
                        ; OUTPUTS:
                        ;
                        ;       R1 - INTEGER PART OF OPERATOR INPUT
                        ;       R2 - DECIMAL PART OF OPERATOR INPUT
                        ;
                        ;
                        ; IMPLICIT OUTPUTS:
                        ;
                        ;       ERROR MESSAGES ARE PRINTED IF THE OPERATOR TYPES AN ILLEGAL CHARACTER,
                        ;       AN INTEGER PART OVER 32767, OR A DECIMAL PART WITH MORE THAN 3 DIGITS.
                        ;
                        ;       AN EXAMPLE STRING IS APPENDED TO THE INPUT PROMPT.
                        ;
                        ; SUBORDINATE ROUTINES USED:
                        ;
                        ;       NONE.
                        ;
                        ; FUNCTIONAL SIDE EFFECTS:
                        ;
                        ;       NONE.
                        ;
                        ; CALLING SEQUENCE:
                        ;
                        ;   EG. MOV     #MADD,PADD              ; LOAD THE PROMPT MESSAGE ADDRESS
                        ;       JSR     PC,DECIN
                        ;
                        ;--
017736                  DECIN::
017736  012700  020310          MOV     #NR1,R0                 ; CLEAR NUMBER AND STRING LOCATIONS
017742  012701  000007          MOV     #7,R1                   ; 2 WORDS FOR INTEGER AND DECIMAL
017746  005020          10$:    CLR     (R0)+                   ; PARTS AND 10 BYTES FOR THE INPUT
017750  005301          DEC     R1                              ; STRING.
017752  001375          BNE     10$                             ;

017754                          PRINTF  PADD                    ; PRINT THE PROMPT
017774                          GMANID  GETNUM,SNUM,A,-1,0,10.,NO ; GET THE NUMBER STRING
020014  012700  020314          MOV     #SNUM,R0                ; POINT TO THE START
```

```
3209 020020  012701  020310                MOV     #NR1,R1          ; ASSUME INTEGER PART FIRST
3210
3211 020024  121027  000053                CMPB    (R0),#'+         ; IS 1ST CHARACTER A + ?
3212 020030  001430                        BEQ     60$              ; IF YES, BRANCH
3213 020032  121027  000055                CMPB    (R0),#'-         ; IS IT A - ?
3214 020036  001425                        BEQ     60$              ; IF YES, BRANCH
3215 020040  121027  000056          20$:  CMPB    (R0),#'.         ; IS IT A . ?
3216 020044  001017                        BNE     50$              ; IF NOT, BRANCH
3217
3218 020046  012701  020312          30$:  MOV     #NR2,R1          ; START ON DECIMAL PART
3219 020052  105760  000002                TSTB    2(R0)            ; FORCE TO 3 DIGITS
3220 020056  001003                        BNE     40$              ;
3221 020060  112760  000060  000002        MOVB    #'0,2(R0)        ; IE. REPLACE NULLS
3222 020066  105760  000003          40$:  TSTB    3(R0)            ;
3223 020072  001007                        BNE     60$              ; WITH ZEROS
3224 020074  112760  000060  000003        MOVB    #'0,3(R0)        ;
3225 020102  000403                        BR      60$              ;
3226
3227 020104  105710                50$:    TSTB    (R0)             ; END OF STRING ?
3228 020106  001451                        BEQ     110$             ; IF YES, FINISH UP
3229 020110  000402                        BR      70$              ; ELSE GET NEXT DIGIT
3230
3231 020112  005200                60$:    INC     R0               ; SKIP OVER THE SIGN OR POINT
3232 020114  000751                        BR      20$              ;
3233
3234 020116  121027  000060          70$:  CMPB    (R0),#60         ; IS CHARACTER A VALID NUMBER ?
3235 020122  002403                        BLT     80$              ; IF TOO LOW, ASK AGAIN
3236 020124  121027  000071                CMPB    (R0),#71         ;
3237 020130  003411                        BLE     90$              ; IF NOT TOO HIGH, BRANCH
3238
3239 020132                        80$:    PRINTF  #DECIN3          ; PRINT 'ILLEGAL CHARACTER'
3240 020152  000671                        BR      DECIN            ; AND ASK AGAIN
3241
3242 020154  021127  006314          90$:  CMP     (R1),#3276.      ; NUMBER TOO HIGH ?
3243 020160  101013                        BHI     100$             ; IF YES, BRANCH
3244
3245 020162  006311                        ASL     (R1)             ; ELSE MULTIPLY BY 10
3246 020164  011102                        MOV     (R1),R2          ;
3247 020166  006311                        ASL     (R1)             ; READY FOR NEXT CHARACTER
3248 020170  006311                        ASL     (R1)             ;
3249 020172  060211                        ADD     R2,(R1)          ;
3250
3251 020174  112002                        MOVB    (R0)+,R2         ; SAVE THE CHARACTER
3252 020176  162702  000060                SUB     #60,R2           ; CONVERT TO NUMBER
3253 020202  060211                        ADD     R2,(R1)          ; AND ADD TO ACCUMULATOR
3254 020204  100401                        BMI     100$             ; IF OVERFLOW, REPORT ERROR
3255
3256 020206  000714                        BR      20$              ; AND GET NEXT CHARACTER
3257
3258 020210                        100$:   PRINTF  #DECIN1          ; PRINT 'NUMBER TOO BIG'
3259 020230  000642                        BR      DECIN            ; AND GET ANOTHER
3260
3261 020232  013701  020310        110$:   MOV     NR1,R1           ; SET UP OUTPUT REGISTERS
3262 020236  013702  020312                MOV     NR2,R2           ;
3263 020242  020227  001750                CMP     R2,#1000.        ; DECIMAL PART TOO BIG ?
3264 020246  103411                        BLO     120$             ; IF NOT, BRANCH
3265 020250                                PRINTF  #DECIN2          ; PRINT 'ONLY 3 DIGITS ALLOWED'
```

```
3266 020270  000622                          BR      DECIN
3267
3268 020272  123727  020314  000055  120$:   CMPB    SNUM,#'-                ; WAS STRING NEGATIVE ?
3269 020300  001002                          BNE     130$                    ; IF NOT, BRANCH
3270 020302  005401                          NEG     R1                      ; ELSE NEGATE THE OUTPUT
3271 020304  005402                          NEG     R2                      ;
3272
3273 020306  000207                  130$:   RTS     PC                      ; AND RETURN
3274
3275                                         .NLIST  BEX
3276
3277 020310  000000                  NR1:    .WORD   0                       ; STORE FOR INTEGER PART
3278 020312  000000                  NR2:    .WORD   0                       ; STORE FOR DECIMAL PART
3279 020314    055     061     062  SNUM:   .ASCIZ  /-12345.678/             ; STORE FOR INPUT STRING
3280
3281 020327    050     105     107  GETNUM: .ASCIZ  /(EG. 12345.678)/                ; PROMPT FOR INPUT
3282 020347    045     116     045  DECIN1: .ASCIZ  /%N%AMUST BE LESS THAN 32768%N/
3283 020405    045     116     045  DECIN2: .ASCIZ  /%N%AONLY 3 DIGITS MAY FOLLOW THE DECIMAL POINT%N/
3284 020466    045     116     045  DECIN3: .ASCIZ  /%N%AILLEGAL CHARACTER%N/
3285
3286                                         .LIST   BEX
3287                                         .EVEN
```

```
                                        ; SUBROUTINE DECOUT - SIGNED DECIMAL OUTPUT ROUTINE
                                        ;
                                        ;++
                                        ; FUNCTIONAL DESCRIPTION:
                                        ;
                                        ;       ROUTINE TO PRINT A SIGNED DECIMAL NUMBER.
                                        ;
                                        ; INPUTS:
                                        ;
                                        ;       R1 - INTEGER PART OF NUMBER TO BE PRINTED
                                        ;       R2 - DECI .AL PART OF NUMBER TO BE PRINTED
                                        ;
                                        ; IMPLICIT INPUTS:
                                        ;
                                        ;       NONE.
                                        ;
                                        ; OUTPUTS:
                                        ;
                                        ;       THE NUMBER IS PRINTED AS FOLLOWS : -12345.678
                                        ;       ( NO SIGN IS PRINTED FOR POSITIVE NUMBERS )
                                        ;       IF THE DECIMAL PART (R2) IS ZERO, THE NUMBER
                                        ;       IS PRINTED AS -12345
                                        ;
                                        ; IMPLICIT OUTPUTS:
                                        ;
                                        ;       NONE.
                                        ;
                                        ; SUBORDINATE ROUTINES USED:
                                        ;
                                        ;       NONE.
                                        ;
                                        ; FUNCTIONAL SIDE EFFECTS:
                                        ;
                                        ;       NONE.
                                        ;
                                        ; CALLING SEQUENCE:
                                        ;
                                        ;   EG. MOV     #-10.,R1                ; PRINT -10.001
                                        ;       MOV     #-1,R2                  ;
                                        ;       JSR     PC,DECOUT
                                        ;
                                        ;--
                                        DECOUT::
020516  010146                          MOV     R1,-(SP)                ; SAVE R1
020520  010246                          MOV     R2,-(SP)                ; SAVE R2
                                        
020522  005701                          TST     R1                      ; R1 > 0 ?
020524  002402                          BLT     10$                     ; IF NOT, BRANCH
020526  005702                          TST     R2                      ; R2 POSITIVE ?
020530  002012                          BGE     20$                     ; IF YES, BRANCH
020532  005401                  10$:    NEG     R1                      ; ELSE MAKE POSITIVE
020534  005402                          NEG     R2                      ;
020536                                  PRINTF  #DEC01                  ; AND PRINT '-'
                                        
020556                          20$:    PRINTF  #DEC02,R1               ; PRINT THE INTEGER PART
020600  005702                          TST     R2                      ; DECIMAL PART = 0 ?
```

```
3346 020602  001411                          BEQ     DECEX                   ; IF YES BRANCH
3347 020604                                  PRINTF  @DEC03,R2               ; ELSE, PRINT DECIMAL PART
3348 020626  012602                  DECEX:  MOV     (SP)+,R2                ; RESTORE R2
3349 020630  012601                          MOV     (SP)+,R1                ; AND R1
3350 020632  000207                          RTS     PC                      ; AND RETURN
3351
3352                                         .NLIST  BEX
3353 020634     045     101     055  DEC01:  .ASCIZ  /%A-/
3354 020640     045     104     065  DEC02:  .ASCIZ  /%D5/
3355 020644     045     101     056  DEC03:  .ASCIZ  /%A.%Z3/
3356                                         .LIST   BEX
3357                                         .EVEN
3358
```

```
                        ;*****************************************************************
                        ;       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
                        ;
                        ;-----------------------------------------------------------------
020654  013746  020712  RANDOM::MOV     RA,-(SP)        ;PUSH RA TO STACK
020660  013700  020714          MOV     RB,R0           ;GET THE LAST RANDOM PATTERN
020664  006316                ASL     @SP             ;SIFT SP (=RA) LEFT
020666  005500                ADC     R0              ;IF CARRY IS SET ADD TO R0 (=RB)
020670  006200                ASR     R0              ;THEN SHIFT THE RESULT RITH
020672  005516                ADC     @SP             ;IF CARRY IS SET ADD TO SP (=RA)
020674  061600                ADD     @SP,R0          ;ADD SP (=RA) AND R0 (=RB)
020676  005600                SBC     R0              ;SUBTRACT CARRY IF SET FROM RB
020700  012637  020712          MOV     (SP)+,RA        ;LOAD NEW VALUE INTO LOCATION RA
020704  010037  020714          MOV     R0,RB           ;LOAD LOCATION RB WITH NEW PAT.
020710  000207                RETURN                  ;

020712  135753  RA::     .WORD   135753          ;START PATTERN FOR RB
020714  024674  RB::     .WORD   24674           ;STORAGE FOR RANDOM PATTERN
```

```
3410                            ;*****************************************************************
3411                            ;       SUBROUTINE - FLASH
3412                            ;-------------------------------------------------- ------ ------ ---
3413                            ; FUNCTIONAL DESCRIPTION:
3414                            ;
3415                            ;       THIS SUBROUTINE FLASHES THE SELECTED MODULE
3416                            ;       LED ON AND OFF WHILE READING THE CONSOLE KEYBOARD INPUT.
3417                            ;       THE EXPECTED INPUTS ARE 'Y' OR 'N' AND THEN A CR OTHER INPUTS
3418                            ;       (EXEPT THE CNTL C) WILL CAUSE A PROMT TO REPEAT
3419                            ;       WITH THE CORRECT CHARACTER.
3420                            ;
3421                            ; INPUTS:
3422                            ;
3423                            ;       CONSOLE KEYBOARD INPUT.
3424                            ;
3425                            ; IMPLICIT INPUTS:
3426                            ;
3427                            ;       NONE.
3428                            ;
3429                            ; OUTPUTS:
3430                            ;
3431                            ;       FLSANS - CONTAINS THE KEYBOARD INPUT YES OR NO
3432                            ;                FLSANS ZERO = NO
3433                            ;                FLSANS ONE  = YES
3434                            ;
3435                            ; IMPLICIT OUTPUTS:
3436                            ;
3437                            ;       NONE.
3438                            ;
3439                            ; SUBORDINATE ROUTINES USED:
3440                            ;
3441                            ;       BREAK - DRS MACRO (THIS MACRO LOOKS FOR CNTL C)
3442                            ;       PRINTF- DRS MACRO, THIS MACRO PRINTS A MESSAGES ON THE CONSOLE
3443                            ;
3444                            ; CALLING SEQUENCE:
3445                            ;
3446                            ;       CALL FLASH OR  JSR PC,FLASH
3447                            ;
3448                            ;-----------------------------------------------------------------
3449 020716  105737  177560     FLASH:: TSTB    TKS             ;TEST THE KEYBOARD STATUS REG.
3450 020722  100002             10$:    BPL     10$             ;BRANCH IF NOTHING FOUND
3451 020724                             BREAK                   ;LOOK FOR 'CNTL C'
3452 020726  000421                     BR      20$             ;
3453 020730  042777  000100  161460     BIC     #100,@MOD       ;SWITH MODULE LED OFF
3454 020736  012701  000012     40$:    MOV     #10.,R1         ;SET UP WAIT COUNTER
3455 020742  004737  014600     30$:    CALL    WT25M           ;WAIT FOR 25 MS
3456 020746  005301                     DEC     R1              ;ARE 250 MS OVER
3457 020750  001374                     BNE     30$             ;BRANCH IF NO
3458 020752  032777  000100  161436     BIT     #100,@MOD       ;IS THE MODULE LED SWITCHED ON
3459 020760  001356                     BNE     FLASH           ;BRANCH IF YES
3460 020762  052777  000100  161426     BIS     #100,@MOD       ;OTHERWISE SWITCH IT ON
3461 020770  000762                     BR      40$             ;AND BRANCH TO WAIT LOOP
3462 020772  012737  000001  021170 20$:    MOV     #1,FLSANS       ;SAVE 'YES' ANSWER
3463 021000  013737  177562  002476     MOV     TKB,BAD         ;GET CHARACTER
3464 021006  042737  000200  002476     BIC     #200,BAD        ;DISCARD PARITY BIT
3465 021014  122737  000131  002476     CMPB    #'Y,BAD         ;WAS THE TYPED CHARACTER A 'Y' ?
3466 021022  001432                     BEQ     50$             ;BRANCH IF YES
```

```
3467 021024  122737  000171  002476          CMPB    #'y,BAD         ;WAS IT A LOWESCASE 'y' ?
3468 021032  001426                          BEQ     50$             ;BRANCH IF YES
3469 021034  005037  021170                  CLR     FLSANS          ;SAVE 'NO' ANSWER
3470 021040  122737  000116  002476          CMPB    #'N,BAD         ;WAS THE TYPED CHARACTER A 'N' ?
3471 021046  001420                          BEQ     50$             ;BRANCH IF YES
3472 021050  122737  000156  002476          CMPB    #'n,BAD         ;WAS IT A LOWERCASE 'n' ?
3473 021056  001414                          BEQ     50$             ;BRANCH IF YES
3474 021060  013737  177562  177566          MOV     TKB,TPB         ;ECHO THE CHARACTER
3475 021066                                  PRINTF  #PROMT          ;IF NO PRINT A PROMT
3476 021106  000703                          BR      FLASH           ;AND GO BACK
3477 021110  013737  177562  177566  50$:    MOV     TKB,TPB         ;ECHO THE CHARACTER
3478 021116  105737  177560          60$:    TSTB    TKS             ;TEST THE KEYBOARD STATUS REG.
3479 021122  100375                          BPL     60$             ;BRANCH IF NOTHING FOUND
3480 021124  013737  177562  002476          MOV     TKB,BAD         ;GET FOUND CHARACTER
3481 021132  042737  000200  002476          BIC     #200,BAD        ;MASK OUT PARITY BIT
3482 021140  122737  000015  002476          CMPB    #15,BAD         ;WAS IT A <CR>
3483 021146  001404                          BEQ     70$             ;BRANCH IF YES
3484 021150  013737  177562  177566          MOV     TKB,TPB         ;ECHO THE CHARACTER
3485 021156  000657                          BR      FLASH
3486 021160  013737  177562  177566  70$:    MOV     TKB,TPB         ;ECHO THE CHARACTER
3487 021166  000207                          RETURN
3488
3489                                  ; CONSOLE EQUATES
3490
3491         177560                           TKS = 177560           ;KEYBOARD STATUS REGISTER
3492         177562                           TKB = 177562           ;KEYBOARD DATA REGISTER
3493         177566                           TPB = 177566           ;PRINTER DATA BUFFER
3494
3495 021170  000000          FLSANS:.         .WORD   0              ;SAVE LOCATION FOR KEYBOARD BUFFER
3496
3497                                  .NLIST  BEX
3498 021172     045     116     045  PROMT:  .ASCIZ  /%N%AUNAUTHORIZED CHARACTER, TYPE ONLY 'Y' OR 'N' THEN <CR>  /
3499                                  .LIST   BEX
3500                                  .EVEN
```

```
                        ;*****************************************************************
                        ; 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 SOUBROUTINE 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
                        ;
                        ;*****************************************************************
021270                  CALRET::
021270  005037  021374          CLR     CARRFL          ;TEMPORY STORE FOR CARRY BIT
021274  105737  177560          TSTB    TKS             ;TEST THE KEYBOARD STATUS REG.
021300  100032                  BPL     10$             ;BRANCH IF NOTHING WAS TYPED
021302                          BREAK                   ;WAS THE TYPED CHAR. A 'CNTL C'
021304  013737  177562  002476  MOV     TKB,BAD         ;GET OPERATOR INPUT
021312  042737  000200  002476  BIC     #200,BAD        ;CLEAR KEYBOARD BUFFER PARITY BIT
021320  022737  000015  002476  CMP     #15,BAD         ;WAS THE TYPED CHARACTER A RETURN ?
021326  001004                  BNE     20$             ;BRANCH IF NO
021330  012737  000001  021374  MOV     #1,CARRFL       ;SET CARRY FLAG
021336  000413                  BR      10$             ;AND BRANCH
021340  013737  177562  177566  20$:    MOV     TKB,TPB         ;ECHO THE CHARACTER
021346                          PRINTF  #RETME1         ;PRINT THAT ONLY CARRIGE RETURN WILL
                                                        ;DO SOMETHING
021366  006237  021374          10$:    ASR     CARRFL          ;AFFECT THE CARRY
021372  000207                  RETURN

                        ; CONSOLE EQUATES
```

```
3559            177560                          TKS = 177560            ;KEYBOARD STATUS REGISTER
3560            177562                          TKB = 177562            ;KEYBOARD DATA REGISTER
3561            177566                          TPB = 177566            ;PRINTER DATA BUFFER
3562
3563 021374    000000                  CARRFL: .WORD   0       ;SAVE LOCATION FOR CARRY BIT
3564
3565                                            .NLIST  BEX
3566 021376       045     116     045   RETME1: .ASCIZ  /<15><12>ATYPE CNTL C TO ABORT OR RETURN TO GO TO THE NEXT STEP/
3567                                            .LIST   BEX
3568                                            .EVEN
3569
```

```
3571                                    ; SUBROUTINE AMREST - AM9513 MODULE RESET ROUTINE.
3572                                    ;
3573                                    ;++
3574                                    ; FUNCTIONAL DESCRIPTION:
3575                                    ;
3576                                    ;       THIS SUBROUTINE IS CALLED AT THE START OF TEST WHICH
3577                                    ;       REFERENCE THE AM9513 COUNTER MODULE.  IT RESETS THAT
3578                                    ;       MODULE AND PLACES IT INTO A KNOWN STATE.
3579                                    ;
3580                                    ; INPUTS:
3581                                    ;
3582                                    ;       NONE.
3583                                    ;
3584                                    ; IMPLICIT INPUTS:
3585                                    ;
3586                                    ;       R1 CONTAINS THE BASE ADDRESS OF THE INTERFACE
3587                                    ;
3588                                    ; OUTPUTS:
3589                                    ;
3590                                    ;       NONE.
3591                                    ;
3592                                    ; IMPLICIT OUTPUTS:
3593                                    ;
3594                                    ;       THE ERROR COUNT FOR THE LOGICAL UNIT BEING TESTED IS
3595                                    ;       INCREMENTED IF THE PROGRAM IS NOT LOOPING.
3596                                    ;
3597                                    ; SUBORDINATE ROUTINES USED:
3598                                    ;
3599                                    ;       NONE.
3600                                    ;
3601                                    ; FUNCTIONAL SIDE EFFECTS:
3602                                    ;
3603                                    ;       IF THE ERROR COUNT EXCEEDS 5 AND THE USER EVL FLAG IS SELECTED,
3604                                    ;       AND THE 'LOOP ON TEST' AND 'INHIBIT DROPPING OF UNITS' FLAGS ARE
3605                                    ;       NOT SELECTED, THE UNIT WILL BE DROPPED FROM TESTING.
3606                                    ;
3607                                    ; CALLING SEQUENCE:
3608                                    ;
3609                                    ;       JSR PC,AMREST
3610                                    ;
3611                                    ;--
3612 021470                            AMREST::
3613 021470                                    SETVEC  #4,#NXM,#PRI07          ;SETUP TO CATCH TRAPS
3614 021516  012761  177777  000002            MOV     #177777,MREA(R1)        ;MASTER RESET
3615 021524  012761  177750  000002            MOV     #C$IN1,MREA(R1)         ;INITIALIZE COMMAND #1
3616 021532  012761  177757  000002            MOV     #C$IN2,MREA(R1)         ;INITIALIZE COMMAND #2
3617 021540  012761  177637  000002            MOV     #177637,MREA(R1)        ;DISARM ALL COUNTERS
3618 021546  012761  177427  000002            MOV     #177427,MREA(R1)        ;SELECT FREQ.OUTPUT REG.
3619 021554  012761  070000  000002            MOV     #70000,MREA(R1)         ;INIT FREQ.OUTPUT REG.
3620 021562  012761  177411  000002            MOV     #177411,MREA(R1)
3621 021570  012761  000010  000004            MOV     #10,MREB(R1)             ;SET LOAD REGISTER
3622 021576  012761  177412  000002            MOV     #177412,MREA(R1)
3623 021604  012761  000010  000004            MOV     #10,MREB(R1)             ;SET LOAD REGISTER
3624 021612  012761  177413  000002            MOV     #177413,MREA(R1)
3625 021620  012761  000010  000004            MOV     #10,MREB(R1)             ;SET LOAD REGISTER
3626 021626  012761  177414  000002            MOV     #177414,MREA(R1)
3627 021634  012761  000010  000004            MOV     #10,MREB(R1)             ;SET LOAD REGISTER
```

```
3628 021642  012761  177415  000002          MOV     #177415,MREA(R1)
3629 021650  012761  000010  000004          MOV     #10,MREB(R1)            ;SET LOAD REGISTER
3630 021656  012761  177421  000002          MOV     #177421,MREA(R1)
3631 021664  012761  000000  000004          MOV     #0,MREB(R1)             ;CLEAR HOLD REGISTER
3632 021672  012761  177422  000002          MOV     #177422,MREA(R1)
3633 021700  012761  000000  000004          MOV     #0,MREB(R1)             ;CLEAR HOLD REGISTER
3634 021706  012761  177423  000002          MOV     #177423,MREA(R1)
3635 021714  012761  000000  000004          MOV     #0,MREB(R1)             ;CLEAR HOLD REGISTER
3636 021722  012761  177424  000002          MOV     #177424,MREA(R1)
3637 021730  012761  000000  000004          MOV     #0,MREB(R1)             ;CLEAR HOLD REGISTER
3638 021736  012761  177425  000002          MOV     #177425,MREA(R1)
3639 021744  012761  000000  000004          MOV     #0,MREB(R1)             ;CLEAR HOLD REGISTER
3640 021752  012761  177537  000002          MOV     #177537,MREA(R1)        ;LOAD ALL COUNTERS
3641 021760  005037  002510                  CLR     INTFLA                  ;INIT INTERRUPT COUNT
3642 021764  000207                          RTS     PC
3643
```

```
                                        ;*****************************;
                                        ; 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.
                                ;--

021766                                  BGNSRV  NXM
021766  012737  000001  002470          MOV     #1,NXMFLG               ; FLAG NXM TRAP
021774                                  ENDSRV


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

                                ;++
                                ;THIS ROUTINE INCREMENTS A FLAG (INTFLA). IT WILL BE EXECUTED
                                ;WHEN AN INTERRUPT HAS OCCURRED.
                                ;--

021776                                  BGNSRV  INTSR
021776  005237  002510                  INC     INTFLA          ; COUNT INTERRUPTS
022002                                  ENDSRV


                                ;*************************************************************
                                ; INTERRUPT SERVICE ROUTINE - BADIV
                                ;*************************************************************

                                ;++
                                ;THIS ROUTINE LOADS VECTOR ADDRESS INTO BIV. IT IS EXECUTED
                                ;WHEN AN INTERRUPT OCCURS TO THE WRONG VECTOR.
                                ;
                                ;BIV SHOULD BE CLEARED IMMEDIATELY BEFORE EXECUTING CODE WHICH MAY
                                ;CAUSE THE INTERRUPT
                                ;--

022004                                  BGNSRV  BADIV
022004  011637  002472                  MOV     (SP),BIV                ;LOAD INTERRUPT VECTOR ADDRESS
022010  162737  000004  002472          SUB     #4,BIV                  ;BACK UP TO CORRECT ADDRESS
022016  062706  000004                  ADD     #4,SP                   ;CLEAR STACK
022022                                  ENDSRV


                                ;*************************************************************
                                ; INTERRUPT SERVICE ROUTINE - CLINT
                                ;*************************************************************
```

```
                                ;***
                                ;THIS ROUTINE IS A DUMY SERVICE FOR THE LINE TIME CLOCK INTERRUPTS
                                ;IT WILL BE EXECUTED WHEN AN INTERRUPT AT VECTOR 100 IS OCCURED.
                                ;---

022024                                  BGNSRV  CLINT
                                                        ;NO ACTION IN THIS ROUTINE
022024                                  ENDSRV
022026                                  ENDMOD
```

```
                                .TITLE MISCELLANEOUS SECTIONS
                                .SBTTL  REPORT CODING SECTION

                                        BGNMOD

                                ; PRINT ROUTINE

                                ;++
                                ; FUNCTIONAL DESCRIPTION:
                                ;
                                ;       PRINTS OUT TEST TITLES, A CONFIGURATION TABLE OR A STATISTICS
                                ;       TABLE FOR THE UNITS UNDER TEST. WHICH TO PRINT IS DETERMINED BY
                                ;       USER INPUT.
                                ;
                                ;       THE CONFIGURATION TABLE SHOWS THE HARDWARE CHARACTERISTICS OF
                                ;       THE UNIT UNDER TEST AND IF PRESENT, OF THE UNIT TO WHICH IT IS
                                ;       CONNECTED FOR LOOPBACK TESTING. IF THE FIRST UNIT MODE ADDRESS
                                ;       IS ZERO (IE. NO TESTS HAVE BEEN RUN AND NO HARDWARE PARAMETERS
                                ;       HAVE BEEN SET UP) THE AUTOMATIC CONFIGURATION ROUTINE 'CONFIG'
                                ;       IS CALLED TO ASCERTAIN THE HARDWARE CONFIGURATION.
                                ;
                                ;       THE STATISTICS TABLE DISPLAYS THE NUMBER OF ERRORS WHICH THE
                                ;       DIAGNOSTIC HAS DETECTED FOR EACH UNIT, AND WHETHER THE UNIT HAS
                                ;       BEEN DROPPED FROM TESTING.
                                ;
                                ; INPUTS:
                                ;
                                ;       THE USER IS ASKED TO TYPE A CHARACTER INDICATING WHETHER TO
                                ;       PRINT THE TEST TITLES, THE CONFIGURATION TABLE, THE STATISTICS
                                ;       TABLE OR HOW TO REESTABLISH THE SYSTEM CONFIGURATION.
                                ;
                                ; IMPLICIT INPUTS:
                                ;
                                ;       THE HARDWARE PARAMETER TABLE IS READ FOR THE CONFIGURATION
                                ;       PRINTOUT.
                                ;
                                ;       THE ERROR TABLE 'ECNT' IS USED FOR THE STATISTICS PRINTOUT.
                                ;
                                ;       TEST TITLES ARE ASSUMED TO BE LABELLED WITH THE FORMAT TDHDnn,
                                ;       WHERE nn IS THE TEST NUMBER. NTESTS AT THE START OF THE ROUTINE
                                ;       MUST EQUAL THE NUMBER OF TESTS IN THE DIAGNOSTIC.
                                ;
                                ; OUTPUTS:
                                ;
                                ;       EITHER A LIST OF TEST TITLES, A CONFIGURATION TABLE OR A
                                ;       STATISTICS TABLE ARE OUTPUT.
                                ;
                                ;       IF ANY UNIT HAS BEEN DROPPED OR DESELECTED USING THE "UNITS"
                                ;       SWITCH, A MESSAGE "PLEASE TYPE ADD" MAY BE PRINTED.
                                ;
                                ; IMPLICIT OUTPUTS:
                                ;
                                ;       NONE.
                                ;
                                ; SUBORDINATE ROUTINES USED:
                                ;
                                ;       CONFIG - AUTOMATIC CONFIGURATION ROUTINE.
```

```
                                ;       CONPRI - CONFIGURATION PRINT ROUTINE.
                                ;       CRLF   - LINE FEED PRINT ROUTINE.
                                ;
                                ; FUNCTIONAL SIDE EFFECTS:
                                ;
                                ;       IF NO HARDWARE PARAMETERS ARE SET UP, THE AUTOMATIC
                                ;       CONFIGURATION ROUTINE IS CALLED TO GENERATE A HARDWARE P TABLE.
                                ;
                                ;       REGISTERS R1 TO R5 ARE CORRUPTED.
                                ;
                                ; CALLING SEQUENCE:
                                ;
                                ;       INVOKED BY THE OPERATOR PRINT COMMAND.
                                ;
                                ;--

000013                          NTESTS=11.                      ; 11 TESTS FOT TITLE PRINTOUT

022026                          BGNRPT
022026                  ASK:    MANUAL                          ; IS MANUAL INTERVATION ALLOWED ?
022030                          BCOMPLETE 10$                   ; IF YES, BRANCH (UAM NOT SET)
022032                          PRINTF  @PRA                    ; PRINT THAT UAM HAS TO BE SET
022052                          EXIT    RPT                     ; EXIT PRINT ROUTINE
022056                  10$:    GMANID  PR1,CHAR,A,377,1,4,YES; PROMPT FOR A COMMAND
022076  023727  023162  000122          CMP     CHAR,#'R                ; REESTABLISH SYSTEM CONFIG. ?
022104  001457                        BEQ     RECON                   ; IF YES, OUTPUT INFORMATION
022106  023727  023162  000124        CMP     CHAR,#'T                ; TEST LIST REQUESTED ?
022114  001515                        BEQ     TITLE                   ; IF YES, OUTPUT TITLES
022116  023727  023162  000103        CMP     CHAR,#'C                ; CONFIGURATION REQUESTED ?
022124  001574                        BEQ     CON                     ; IF YES, OUTPUT CONFIGURATION
022126  023727  023162  000123        CMP     CHAR,#'S                ; STATISTICS REQUESTED ?
022134  001002                        BNE     HEL                     ; IF NOT, PRINT THE HELP MESSAGE
022136  000137  022766                JMP     STAT                    ; IF YES, OUTPUT STATISTICS
022142                  HEL:    PRINTF  @PR2                    ; OTHERWISE, PRINT THE HELP MESSAGE
022162                          PRINTF  @PR2A                   ;
022202                          PRINTF  @PR2B                   ;
022222                          PRINTF  @PR2C                   ;
022242  000671                BR      ASK                     ; AND PROMPT FOR COMMAND AGAIN
022244                  RECON:  PRINTF  @PR2D                   ; PRINT HOW TO REESTABLISH
022264                          PRINTF  @PR2E                   ; THE SYSTEM CONFIGURATION
022304                          PRINTF  @PR2F                   ; ...
022324                          PRINTF  @PR2G                   ; ...
022344  000137  023152          JMP     PREX                    ; AND EXIT
022350                  TITLE:  PRINTF  @TT                     ; TEST LIST HEADER
022370  012701  000001          MOV     #1,R1                   ; START WITH TEST 1
022374  012702  023172          MOV     #TADS,R2                ; START OF LIST OF TITLE ADDRESSES
022400  012703  000022  10$:    MOV     #18.,R3                 ; WAIT AFTER 18 LINES
022404                  20$:    PRINTF  @TNUM,R1                ; PRINT TEST NUMBER
022426                          PRINTF  (R2)                    ; AND TITLE
022444  062702  000002          ADD     #2,R2                   ; GET ADDRESS OF NEXT TITLE
022450  005201                  INC     R1                      ; AND NEXT TEST NUMBER
022452  020127  000013          CMP     R1,#NTESTS              ; ALL PRINTED ?
```

```
3866 022456  003015                        BGT     30$                     ; IF YES, EXIT
3867 022460  005303                        DEC     R3                      ; 18 LINES OUTPUT ?
3868 022462  001350                        BNE     20$                     ; IF NOT, BRANCH
3869 022464                                GMANID  RDY,RFLG,A,377,0,1,YES  ; ELSE WAIT FOR OPERATOR TO READ
3870 022504  004737  014634                JSR     PC,CRLF                 ; PRINT A LINE FEED
3871 022510  000733                        BR      10$                     ; AND THEN CONTINUE
3872
3873 022512  000137  023152          30$:  JMP     PREX                    ; EXIT
3874
3875 022516  005037  023156          CON:  CLR     STFLG                   ; ASSUME DIAGNOSTIC IS NOT STARTED
3876 022522  005037  023160                CLR     GPFLG                   ; FLAG NO GPHARDS YET EXECUTED
3877 022526  005737  002554                TST     GPADD                   ; ARE ANY PARAMETER ADDRESSES SET UP ?
3878 022532  001402                        BEQ     10$                     ; IF NOT, BRANCH
3879 022534  005237  023156                INC     STFLG                   ; ELSE FLAG DIAGNOSTIC IS STARTED
3880
3881 022540  005737  002012          10$:  TST     L$UNIT                  ; ANY UNITS SET UP ?
3882 022544  001404                        BEQ     20$                     ; IF NOT, BRANCH
3883 022546  023727  002012  000020        CMP     L$UNIT,#16.             ; TOO MANY UNITS SET UP ?
3884 022554  003403                        BLE     30$                     ; IF NOT, BRANCH
3885 022556  012737  000020  002012  20$:  MOV     #16.,L$UNIT             ; SET UP 16 UNITS
3886
3887 022564  013746  002074          30$:  MOV     L$LUN,-(SP)             ; SAVE THE UNIT NUMBER BEING TESTED
3888 022570  005037  002074                CLR     L$LUN                   ; START WITH UNIT 0
3889
3890 022574  013701  002554                MOV     GPADD,R1                ; GET FIRST PARAM. ADDRESS
3891 022600  005711                        TST     (R1)                    ; ANYTHING IN IT
3892 022602  001003                        BNE     40$                     ; IF YES, BRANCH
3893 022604  012737  171000  015572        MOV     #171000,STADD           ; ELSE START FOR FIRST IAV/IDV ADDR.
3894
3895 022612  013701  002074          40$:  MOV     L$LUN,R1                ; FORM OFFSET FOR UNIT TABLES
3896 022616  006301                        ASL     R1                      ;
3897 022620  005761  002554                TST     GPADD(R1)               ; PARAMETER ADDRESS SET UP ?
3898 022624  001010                        BNE     50$                     ; IF YES, BRANCH
3899 022626                                GPHARD  L$LUN,GPADD(R1)         ; ELSE, GET P TABLE ADDRESS
3900 022640                                BNCOMPLETE ASKADD               ; IF DESELECTED, REQUEST OPERATOR ADD
3901 022642  005237  023160                INC     GPFLG                   ; FLAG THAT GPHARD HAS BEEN USED
3902
3903 022646  005771  002554          50$:  TST     @GPADD(R1)              ; MODE ADDRESS OF UNIT = 0 ?
3904 022652  001002                        BNE     60$                     ; IF NOT, BRANCH
3905 022654  004737  015306          55$:  JSR     PC,CONFIG               ; DO AUTO CONFIGURATION
3906
3907 022660  005237  002074          60$:  INC     L$LUN                   ; NEXT UNIT
3908 022664  023737  002074  002012        CMP     L$LUN,L$UNIT            ; ALL DONE ?
3909 022672  002747                        BLT     40$                     ; IF NOT, DO THE NEXT
3910 022674  012637  002074                MOV     (SP)+,L$LUN             ; RESTORE THE UNIT NUMBER
3911
3912 022700  004737  015656          70$:  JSR     PC,CONPRI               ; PRINT THE CONFIGURATION
3913
3914 022704  005737  023156                TST     STFLG                   ; WAS DIAGNOSTIC STARTED ?
3915 022710  001520                        BEQ     PREX                    ; IF NOT, EXIT
3916 022712  005737  023160                TST     GPFLG                   ; HAVE WE USED ANY GPHARDS ?
3917 022716  001515                        BEQ     PREX                    ; IF NOT, EXIT
3918 022720                          80$:  PRINTF  @PR7                    ; ELSE PRINT 'USE STA OR RES'
3919 022740  000504                        BR      PREX                    ; AND EXIT
3920
3921 022742                          ASKADD: PRINTF @PR6                   ; ASK OPERATOR TO ADD UNITS
3922 022762  000137  023152                JMP     PREX                    ; AND EXIT
```

```
3923
3924 022766                          STAT:   PRINTF  #PR3                    ; PRINT STATISTICS HEADER
3925 023006                                  PRINTF  #PR3A                   ;
3926
3927 023026  005001                          CLR     R1                      ; START WITH FIRST UNIT
3928
3929 023030  020137  002012          20$:    CMP     R1,L$UNIT               ; ALL UNITS REPORTED ?
3930 023034  001444                          BEQ     60$                     ; IF YES, EXIT
3931 023036  010104                          MOV     R1,R4                   ; FORM OFFSET TO ERROR COUNT
3932 023040  006304                          ASL     R4                      ;
3933 023042  016405  002514                  MOV     ECNT(R4),R5             ; GET UNIT'S ERROR COUNT
3934 023046  005705                          TST     R5                      ; IS IT NEGATIVE ?
3935 023050  100423                          BMI     40$                     ; IF YES, REPORT UNTESTED
3936
3937 023052  012703  025006                  MOV     #NO,R3                  ; ASSUME UNIT IS NOT DROPPED
3938 023056  105761  002614                  TSTB    DROPED(R1)              ; CHECK IF IT IS
3939 023062  001402                          BEQ     30$                     ; IF IT IS NOT, BRANCH
3940 023064  012703  025002                  MOV     #YES,R3                 ; OTHERWISE PRINT YES
3941 023070                          30$:    PRINTF  #PR5,R1,R5,R3           ; ELSE PRINT STATISTICS
3942 023116  000411                          BR      50$                     ; AND LOOK FOR MORE UNITS
3943
3944 023120                          40$:    PRINTF  #PR4,R1                 ; PRINT 'UNTESTED'
3945
3946 023142  005201                  50$:    INC     R1                      ; PREPARE FOR NEXT UNIT
3947 023144  000731                          BR      20$                     ; IF NOT, REPORT THE NEXT
3948
3949 023146  004737  014634          60$:    JSR     PC,CRLF                 ; PRINT A LINE FEED
3950
3951 023152                          PREX:   EXIT    RPT
3952
3953 023156  000000                  STFLG:  .WORD   0                       ; SET IF DIAGNOSTIC HAS BEEN STARTED
3954 023160  000000                  GPFLG:  .WORD   0                       ; SET IF ANY GPHARDS ARE EXECUTED
3955 023162     110     040     040  CHAR:   .ASCIZ  /H    /                 ; STORE FOR OPERATOR INPUT
     023165     040     040     000
3956 023170  000000                  RFLG:   .WORD   0                       ; FLAG FOR 'TYPE RETURN FOR MORE TITLES
3957
3981
3982 023172                          TADS:   TITLES                          ; LIST OF TEST TITLE ADDRESSES

     023172  030206                          .WORD   TSHD1
     023174  030574                          .WORD   TSHD2
     023176  031216                          .WORD   TSHD3
     023200  032120                          .WORD   TSHD4
     023202  033474                          .WORD   TSHD5
     023204  034076                          .WORD   TSHD6
     023206  034576                          .WORD   TSHD7
     023210  035270                          .WORD   TSHD8
     023212  036110                          .WORD   TSHD9
     023214  037122                          .WORD   TSHD10
     023216  037350                          .WORD   TSHD11
3986
3987                                         .NLIST BEX
3988
3989 023220     045     116     045  TT:     .ASCII  /#N#ATEST TITLES./
3990 023240     045     116     045          .ASCIZ  /#N#A-----------#N2/
3991
3992 023264     124     131     120  RDY:    .ASCIZ  /TYPE "RETURN" FOR MORE TITLES/
```

```
3993
3994 023322      045     104     063  TNUM:    .ASCIZ  /%D3/
3995
3996 023326      045     116     045  PRA:     .ASCIZ  /%N%ATO USE THE PRINT COMMAND, PLEASE CLEAR THE "UAM" FLAG/
3997
3998 023420      124     131     120  PR1:     .ASCIZ  /TYPE T,R,C,S OR HELP/
3999
4000 023445      045     116     045  PR2:     .ASCII  /%N%ATHE FOLLOWING COMMANDS ARE ACCEPTED :-/
4001 023517      045     116     062           .ASCIZ  /%N2%AT - PRINT TEST TITLES/
4002 023552      045     116     045  PR2A:    .ASCIZ  /%N%AC - PRINT CONFIGURATION TABLE CURRENTLY USED BY DIAGNOSTIC/
4003 023651      045     116     045  PR2B:    .ASCIZ  /%N%AR - PRINT HOW TO REESTABLISH THE SYSTEM CONFIGURATION/
4004 023743      045     116     045  PR2C:    .ASCIZ  /%N%AS - PRINT STATISTICS TABLE%N/
4005
4006 024004      045     116     045  PR2D:    .ASCIZ  /%N%ATO REESTABLISH THE SYSTEM CONFIGURATION, ANSWER THE/
4007 024074      045     116     045  PR2E:    .ASCIZ  /%N%AHARDWARE QUESTION TYPING "0" AS THE MODE ADDRESS FOR 16 UNITS./
4008 024177      045     116     045  PR2F:    .ASCIZ  /%N%N%AEG.  MODE REGISTER ADDRESS (0) 0 ?  0..... ...........
4009 024273      045     116     045  PR2G:    .ASCIZ  /%N%A                                        ----------------/
4010 024367      045     116     062  PR3:     .ASCII  \%N2%AIDV/IAV-11 MODULE STATISTICS.\
4011 024431      045     116     045           .ASCIZ  /%N%A-----------------------------/
4012 024474      045     116     062  PR3A:    .ASCIZ  /%N2%AUNIT   ERRORS   DROPPED%N/
4013
4014 024533      045     116     045  PR4:     .ASCIZ  /%N%D3%A   UNTESTED    NO/
4015
4016 024564      045     116     045  PR5:     .ASCIZ  /%N%D3%S5%D3%S7%T/
4017
4018 024605      045     116     045  PR6:     .ASCII  /%N%AUNIT DROPPED OR DESELECTED - PLEASE TYPE "ADD"/
4019 024667      040     101     116           .ASCIZ  / AND TRY AGAIN/
4020
4021 024706      045     116     045  PR7:     .ASCIZ  /%N%APLEASE TYPE "START" OR "RESTART" TO CONTINUE TESTING.%N/
4022
4023 025002      131     105     123  YES:     .ASCIZ  /YES/
4024 025006      116     117     000  NO:      .ASCIZ  /NO/
4025
4026                                           .LIST BEX
4027                                           .EVEN
4028
4029 025012                                    ENDRPT
```

```
                                .SBTTL  PROTECTION TABLE

                                ;++
                                ; THIS TABLE IS USED BY THE RUNTIME SERVICES
                                ; TO PROTECT THE LOAD MEDIA.
                                ;--

                                        BGNPROT

025014                                  0               ;OFFSET INTO P-TABLE FOR MODE ADDRESS
025016  177777                          1               ;OFFSET INTO P-TABLE FOR MASSBUS ADDRESS
025020  177777                          -1              ;OFFSET INTO P-TABLE FOR DRIVE NUMBER

025022                                  ENDPROT
```

```
                                .SBTTL  INITIALIZE SECTION

                                ;++
                                ; THE INITIALIZE SECTION CONTAINS THE CODING THAT IS PERFORMED
                                ; AT THE BEGINNING OF EACH PASS.
                                ;--

025022                                  BGNINIT


025022  005037  026214          START:  CLR     LOOPEX          ; CLEAR LOOP FLAG
025026                                  READEF  #EF.START       ; IS THIS A  NEW START ?
025034                                  BNCOMPLETE RESTRT       ; IF NOT, BRANCH
025036                                  SETVEC  #14,#113240,#340; *** JUST FOR DEBUG PROGRAM ***
025064                                  SETVEC  #100,#CLINT,#340; IGNORE FURTHER INTERRUPTS TO VECTOR 100
025112                                  PRINTF  #WARN           ;CONNECT TEST CONNECTOR
025132                                  PRINTF  #WARN1          ;AND VOLTAGE SOURCE
025152  004737  014662                  JSR     PC,WRDY         ;WAIT FOR OPERATORTO TYPE RETURN
025156  004737  014634                  JSR     PC,CRLF         ;PRINT A LINE FEED

025162                          10$:    BRESET                  ; RESET THE SYSTEM
025164  004737  026650                  JSR     PC,SETCLK       ; SET UP CLOCK COUNTER

025170  005737  002012                  TST     L$UNIT          ; ANY UNITS CONFIGURED ?
025174  001404                          BEQ     20$             ; IF NOT, BRANCH
025176  023727  002012  000020          CMP     L$UNIT,#16.     ; TOO MANY UNITS SET UP ?
025204  003403                          BLE     30$             ; IF NOT, BRANCH
025206  012737  000020  002012  20$:    MOV     #16.,L$UNIT     ; ELSE, SET UP 16 UNITS

025214  005037  026212          30$:    CLR     ACFLG           ; CLEAR AUTO CONFIGURATION FLAG


025220  012700  002640                  MOV     #LOPFLG,R0      ; FLAG THAT LOOP CONFIGURATION IS NOT CHECKED
025224  012701  000020                  MOV     #16.,R1         ; FOR POSSIBLE 16 UNITS
025230  005020                  50$:    CLR     (R0)+           ; CLEAR THE FLAG
025232  005301                          DEC     R1              ; ALL DONE ?
025234  001375                          BNE     50$             ; IF NOT, DO THE NEXT


025236  012700  002614                  MOV     #DROPED,R0      ; GET UNIT DROPPED TABLE ADDERSS
025242  012701  000020                  MOV     #16.,R1         ; THERE ARE 16 UNITS
025246  105020                  55$:    CLRB    (R0)+           ; CLEAR ALL 16 DROPPED UNIT FLAGS
025250  005301                          DEC     R1              ; ...
025252  001375                          BNE     55$             ; ...
025254  012700  002514                  MOV     #ECNT,R0        ; GET ERROR COUNT FOR UUT 0
025260  012701  000020                  MOV     #16.,R1         ; THERE ARE 16 UUT'S
025264  012720  100000          60$:    MOV     #100000,(R0)+   ; INIT THE ERROR COUNT
025270  005301                          DEC     R1              ; ALL CLEARED ?
025272  001374                          BNE     60$             ; IF NOT, CLEAR NEXT COUNT
025274  000404                          BR      STARES          ; ELSE, START TESTING WITH FIRST UUT

025276                          RESTRT: READEF  #EF.RESTART     ; IS THIS A RESTART ?
025304                                  BNCOMPLETE NEWST        ; IF NOT, BRANCH

025306  005037  002634          STARES: CLR     TSTFLG          ; SHOW NO TESTS HAVE BEEN RUN THIS PASS
025312  005037  002636                  CLR     TSUFLG          ; OR FOR THIS UNIT
025316  000534                          BR      INIUUT          ; START TESTING WITH FIRST UUT
```

```
4140
4141 025320                            NEWST:  READEF  #EF.NEW         ; IS THIS A NEW PASS ?
4142 025326                                    BNCOMPLETE CONT         ; IF NOT, BRANCH
4143 025330  005737  002634                    TST     TSTFLG          ; WERE ANY TESTS SELECTED LAST TIME ?
4144 025334  001011                            BNE     10$             ; IF YES, BRANCH
4145 025336                                    PRINTF  #NTEST          ; IF NOT, TELL THE USER
4146 025356                                    DOCLN                   ; AND RETURN TO THE SUPERVISOR
4147
4148 025360  005037  002634            10$:    CLR     TSTFLG          ; INIT TEST FLAG AGAIN
4149 025364  005737  026212                    TST     ACFLG           ; JUST DONE A RUN OF AUTO CONFIGURATION ?
4150 025370  001507                            BEQ     INIUUT          ; IF NOT, TEST FIRST UNIT
4151 025372  005037  026212                    CLR     ACFLG           ; ELSE, FLAG THAT THE RUN IS FINISHED
4152 025376  004737  015656                    JSR     PC,CONPRI       ; PRINT THE CONFIGURATION
4153 025402  004737  014662                    JSR     PC,WRDY         ; WAIT FOR OPERATOR TO TYPE 'RETURN'
4154 025406  000500                            BR      INIUUT          ; THEN TEST THE FIRST UNIT
4155
4156 025410                            CONT:   READEF  #EF.CONTINUE    ; IS THIS A CONTINUE ?
4157 025416                                    BNCOMPLETE PWRFL        ; IF NOT, BRANCH
4158 025420                                    SETVEC  #4,#NXM,#PRI07  ; IGNORE NXM TRAPS
4159 025446  052777  000100  154742            BIS     #100,@MOD       ; ELSE, LIGHT UUT LED
4160 025454                                    CLRVEC  #4              ; RESTORE THE SUPERVISOR NXM VECTOR
4161 025462  000137  026202                    JMP     END             ; AND CONTINUE
4162
4163 025466                            PWRFL:  READEF  #EF.PWR         ; IS THIS A POWER FAIL
4164 025474                                    BNCOMPLETE NXTUUT       ; IF NOT, MUST BE NEXT UNIT
4165 025476                                    SETVEC  #4,#NXM,#PRI07  ; IGNORE NXM TRAPS
4166 025524  052777  000100  154664            BIS     #100,@MOD       ; ELSE, LIGHT UUT LED
4167 025532                                    CLRVEC  #4              ; RESTORE THE SUPERVISOR NXM VECTOR
4168 025540  000137  026162                    JMP     PSEUL1          ; AND CONTINUE
4169
4170 025544  005737  002636            NXTUUT: TST     TSUFLG          ; WERE ANY TESTS RUN ON THE LAST UNIT ?
4171 025550  001014                            BNE     10$             ; IF YES, BRANCH
4172 025552                                    PRINTF  #NTEST1,L$LUN   ; ELSE PRINT A WARNING
4173 025576  004737  014662                    JSR     PC,WRDY         ; LET THE OPERATOR READ IT
4174 025602  005037  002636            10$:    CLR     TSUFLG          ; SHOW NO TESTS FOR NEXT UNIT
4175 025606  000422                            BR      NEXT            ; AND TEST THE NEXT UNIT
4176
4177 025610  012737  177777  002074    INIUUT: MOV     #-1,L$LUN       ; INITIALIZE LOGICAL UNIT NUMBER.
4178 025616  005237  026214                    INC     LOOPEX          ; ARE WE IN A UNIT LOOP
4179 025622  022737  000002  026214            CMP     #2,LOOPEX       ; ... ?
4180 025630  001011                            BNE     NEXT            ; BRANCH IF NO
4181 025632                                    PRINTF  #WRSEL          ; IF YES PRINT NOT CORRECT UNIT SELECTED
4182 025652                                    DOCLN                   ; AND RETURN TO THE SUPERVISOR
4183 025654  005237  002074            NEXT:   INC     L$LUN           ; NEXT LOGICAL UNIT TO BE TESTED ?
4184 025660  023737  002074  002012            CMP     L$LUN,L$UNIT    ; ALL UNITS TRIED ?
4185 025666  002350                            BGE     INIUUT          ; IF YES, START AGAIN
4186
4187 025670  013705  002074                    MOV     L$LUN,R5        ; SAVE UNIT NUMBER
4188 025674  006305                            ASL     R5              ; FORM OFFSET
4189
4190 025676                                    GPHARD  L$LUN,R1        ; GET PARAMETER TABLE ADDRESS IN R1
4191 025706                                    BNCOMPLETE NEXT         ; IF DROPPED, GET THE NEXT
4192 025710  010165  002554                    MOV     R1,GPADD(R5)    ; ELSE SAVE THE ADDRESS
4193 025714  005711                            TST     (R1)            ; MODE ADDRESS = 0 ?
4194 025716  001024                            BNE     [illegible]     ; IF NOT, BRANCH
4195 025720  004737  015306                    JSR     PC,CONFIG       ; ELSE DO AUTO CONFIGURATION FOR THIS UNIT
4196 025724  012737  000001  026212            MOV     #1,ACFLG        ; AND FLAG THAT WE ARE DOING IT
```

```
4197 025732  103014                        BCC     10$             ; BRANCH IF ANOTHER UNIT FOUND
4198 025734  005737  002012                TST     L$UNIT          ; ARE THERE ANY UNITS TO TEST ?
4199 025740  001323                        BNE     INIUUT          ; IF YES, START AGAIN WITH THE FIRST UNIT
4200 025742                                PRINTF  @NODEV          ; ELSE PRINT "NO UNITS FOUND"
4201 025762                                DOCLN                   ; AND RETURN TO THE SUPERVISOR
4202
4203 025764  016501  002554          10$:  MOV     GPADD(R5),R1    ; GET PARAMETER TABLE ADDRESS IN R1
4204
4205 025770  011137  002416          20$:  MOV     (R1),MOD        ; SAVE NEW MODE REGISTER ADDRESS
4206 025774  011137  002420                MOV     (R1),CSR        ;
4207 026000  062737  000002  002420        ADD     #2,CSR          ;
4208 026006  011137  002422                MOV     (R1),CCR        ; SAVE NEW CCR REGISTER ADDRESS
4209 026012  062737  000004  002422        ADD     #4,CCR          ;
4210 026020  012137  002424                MOV     (R1)+,INR       ; SAVE NEW INR REGISTER ADDRESS
4211 026024  062737  000006  002424        ADD     #6,INR          ;
4212
4213 026032  012137  002426                MOV     (R1)+,VEC       ; SAVE NEW VECTOR ADDRESS
4214 026036  012137  002430                MOV     (R1)+,PRIO      ; SAVE NEW PRIORITY
4215
4216 026042  005037  002470                CLR     NXMFLG          ;
4217 026046                                SETVEC  #4,#NXM,#PRI07  ; IGNORE NXM TRAPS
4218 026074  017701  154316                MOV     @MOD,R1         ; GET MODE REGISTER CONTENTS
4219 026100                                CLRVEC  #4              ; RESTORE NXM TRAP CATCHER
4220
4221 026106  000301                        SWAB    R1              ; SWAB MOD REGISTER CONTENS
4222 026110  005737  002470                TST     NXMFLG          ; IS THE USED ADDRESS ACCESSABLE ?
4223 026114  001006                        BNE     70$             ; BRANCH IF NOT
4224 026116  120127  000300                CMPB    R1,#300         ; IS IT A FIVE CHANNEL COUNTER ?
4225 026122  001254                        BNE     NEXT            ; IF NOT BRANCH
4226 026124  052777  000100  154264        BIS     #100,@MOD       ; SWITCH ON UUT LED
4227 026132  042765  100000  002514  70$:  BIC     #100000,ECNT(R5); FLAG UNIT IS BEING TESTED
4228 026140  012702  000420                MOV     #420,R2         ; SET UP FOR FIVE CHANNEL COUNTER
4229 026144  000402                        BR      60$             ; HERE IF ID IS 300 (FIVE CHA. COUNTER)
4230 026146  012702  000040          50$:  MOV     #40,R2          ;
4231 026152  052702  010000          60$:  BIS     #10000,R2       ; ASSUME SPECIFICALLY SELECTED TEST
4232 026156  010237  002702                MOV     R2,CONMSK       ; SAVE TEST CONTROL MASK
4233
4234 026162  005037  002700          PSEUL1: CLR   LOTFLA          ; INIT LOOP ON TEST FLAG
4235 026166                                RFLAGS  R0              ; GET OPERATOR FLAGS
4236 026170  032700  000010                BIT     #LOT,R0         ; LOOP ON TEST SELECTED ?
4237 026174  001402                        BEQ     END             ; IF NOT, BRANCH
4238 026176  005237  002700                INC     LOTFLA          ; ELSE SET FLAG
4239
4240 026202  005037  026214          END:  CLR     LOOPEX          ;
4241 026206                                EXIT    INIT
4242
4243 026212  000000          ACFLG:  .WORD   0               ; SET IF AUTO CONFIGURATION IS TO BE DONE
4244 026214  000000          LOOPEX: .WORD   0               ; SET IF NOT THE COORECT UNIT IS SELECTED
4245                                                         ; IN THE START COMMAND
4246                                .NLIST  BEX
4247
4248 026216      045      116      045  WARN:   .ASCIZ  /#N#ACONNECT TEST CONNECTOR AND REQUIRED VOLTAGE/
4249 026276      045      116      045  WARN1:  .ASCIZ  /#N#ATO FIVE CHANNEL COUNTER MODULES THAT SHOULD BE TESTED./
4250 026371      045      116      045  NTEST:  .ASCIZ  /#N#ANO TESTS WERE RUN - CHECK ANSWERS TO STARTUP QUESTIONS./
4251 026465      045      116      045  NTEST1: .ASCIZ  /#N#ANO TESTS WERE RUN ON UNIT #D3#A./
4252 026532      045      116      045  WRSEL:  .ASCII  /#N#ASELECTED UNIT IS NOT A FIVE CHANNEL COUNTER.#N/
4253 026614      045      101      117          .ASCIZ  /#AOR ID CODE IS INCORRECT/
```

```
4254                                    .LIST   BEX
4255                                    .EVEN
4256
4257 026646                             ENDINIT
4258
4270
```

```
                        ;#############################################
                        ; SUBROUTINES USED DURING INITIALISATION. #
                        ;#############################################

                        .SBTTL  SETCLK - ROUTINE TO SET UP DELAY COUNTS

                        ;++
                        ; FUNCTIONAL DESCRIPTION:
                        ;
                        ;       THIS ROUTINE SETS UP 3 DELAY VARIABLES CALLED CNT25M, CNT500, AND
                        ;       CNT25. THESE GIVE DELAYS OF APPROXIMATELY 25 MILLISECONDS, 500
                        ;       MICROSECONDS OR 25 MICROSECONDS RESPECTIVELY IF USED AS FOLLOWS:
                        ;
                        ;               MOV     CNTXXX,R0
                        ;           1$: DEC     R0
                        ;               BNE     1$
                        ;
                        ;       THE COUNTS ARE DERIVED FROM AN L CLOCK IF THERE IS ONE.
                        ;       OTHERWISE, THE OPERATOR IS ASKED TO TYPE 2 CHARACTERS ON THE
                        ;       CONSOLE 6 SECONDS APART.
                        ;
                        ; INPUTS:
                        ;
                        ;       NONE.
                        ;
                        ; IMPLICIT INPUTS:
                        ;
                        ;       IF CNT25M IS NOT ZERO (ALREADY SET UP), THE ROUTINE DOES NOTHING.
                        ;
                        ; OUTPUTS:
                        ;
                        ;       CONSOLE MESSAGE IF THERE IS NO L CLOCK ON THE SYSTEM.
                        ;
                        ; IMPLICIT OUTPUTS:
                        ;
                        ;       CNT25M CONTAINS THE COUNT REQUIRED FOR 25 MILLISECONDS.
                        ;       CNT500 CONTAINS THE COUNT REQUIRED FOR 500 MICROSECONDS.
                        ;       CNT25 CONTAINS THE COUNT REQUIRED FOR 25 MICROSECONDS.
                        ;
                        ; SUBORDINATE ROUTINES USED:
                        ;
                        ;       CRLF  - LINE FEED PRINT ROUTINE.
                        ;       CLINT - DUMMY CLOCK INTERRUPT SERVICE ROUTINE
                        ;
                        ; FUNCTIONAL SIDE EFFECTS:
                        ;
                        ;       R0 TO R5 ARE CORRUPTED.
                        ;
                        ;       IF A LINE TIME CLOCK IS FOUND, VECTOR 100 IS SET UP SO THAT
                        ;       INTERRUPTS TO IT ARE IGNORED. THE SETVEC MACRO CAN BE USED TO
                        ;       SET UP THE VECTOR FOR A DEVICE INTERRUPT.
                        ;
                        ; CALLING SEQUENCE:
                        ;
                        ;       JSR     PC,SETCLK
                        ;
                        ;--
```

```
026650  005737  014626          SETCLK: TST     CNT25M          ; COUNTERS ALREADY SET UP?
026654  001402                          BEQ     10$             ; IF NOT, BRANCH
026656  000137  027524                  JMP     SETEX           ; IF YES, EXIT
026662  005004                  10$:    CLR     R4              ; CLEAR A COUNTER
026664                                  GETPRI  R2              ; SAVE CURRENT PRIORITY IN R2
026670  005037  027526                  CLR     CLKFLG          ; ASSUME THERE IS NO CLOCK WITH A CSR
026674                                  CLOCK   L,R1            ; GET ADDRESS OF CLOCK TABLE
026704                                  SETVEC  #4,#NXM,#340    ; SET UP CLOCK CSA TRAP
026732  005037  002470                  CLR     NXMFLG          ; CLEAR NXM FLAG
026736  005771  000000                  TST     @(R1)           ; ACCESS THE CLOCK ADDRESS
026742  005737  002470                  TST     NXMFLG          ; *DON'T DELETE, NEEDED FOR FALCON*
026746  005737  002470                  TST     NXMFLG          ; DOES THE CLOCK HAVE A REGISTER ?
026752  001005                          BNE     LCLOCK          ; IF NOT, BRANCH
026754  005237  027526                  INC     CLKFLG          ; ELSE FLAG THERE IS A CLOCK CSR
026760  012771  000100  000000          MOV     #100,@(R1)      ; AND SET IT UP TO INTERRUPT
                                ;
                                ; USE THE L CLOCK
                                ;
026766                          LCLOCK: CLRVEC  #4              ; SET VECTOR 4 TO UNUSED POOL
026774  012703  000006                  MOV     #6,R3           ; IF 50 HZ, 100 MS = 5 INTERRUPTS
027000  026127  000006  000062          CMP     6(R1),#50.      ; 50 HZ CORRECT?
027006  001401                          BEQ     10$             ; IF YES, BRANCH
027010  005203                          INC     R3              ; ELSE ALLOW 6 INTERRUPTS
027012  010305                  10$:    MOV     R3,R5           ; SAVE NUMBER OF INTERRRUPTS
027014                                  SETVEC  #100,#KLINT,#340; SET UP THE CLOCK VECTOR
                                                                ; TO WAIT FOR 1ST INTERRUPT
027042                                  SETPRI  #0              ; AND DROP THE PRIORITY
027050  005000                          CLR     R0              ; CLEAR R0 AND THE CARRY BIT
027052  020305                  20$:    CMP     R3,R5           ; HAS COUNT BEEN DROPPED ?
027054  001004                          BNE     30$             ; IF YES, START THE COUNTERS
027056  005300                          DEC     R0              ; WAITED TOO LONG ?
027060  001374                          BNE     20$             ; IF NOT, WAIT LONGER
027062  000137  027174                  JMP     USCLOK          ; IF YES, ASSUME NO CLOCK
027066  005005                  30$:    CLR     R5              ; CLEAR THE HIGH COUNTER
027070  005204                  40$:    INC     R4              ; COUNT THE DELAY FOR 5 OR 6 INTERRUPTS
027072  001376                          BNE     40$             ;
027074  105205                          INCB    R5              ;
027076  001374                          BNE     40$             ;
027100  000435                          BR      USCLOK          ; IF TOO LONG, ASSUME NO CLOCK
027102  005303                  KLINT:  DEC     R3              ; 5 OR 6 INTERRUPTS?
027104  001401                          BEQ     40$             ; IF YES, TIDY UP
027106  000002                          RTI                     ; ELSE KEEP COUNTING
027110                          40$:    SETPRI  R2              ; RESTORE THE PRIORITY
027114                                  SETVEC  #100,#CLINT,#340; IGNORE FURTHER INTERRUPTS TO VECTOR 100
027142  022626                          CMP     (SP)+,(SP)+     ; TIDY UP THE STACK
027144  005737  027526                  TST     CLKFLG          ; CAN WE DISABLE A CLOCK ?
027150  001402                          BEQ     50$             ; IF NOT, BRANCH
027152  005071  000000                  CLR     @(R1)           ; ELSE, DISABLE CLOCK INTERRUPTS
027156  000241                  50$:    CLC                     ; DIVIDE THE 100 MILLISECOND COUNTERS
027160  006005                          ROR     R5              ; BY 4 TO GIVE 25 MILLISECONDS
```

```
 4386 027162  006004                          ROR     R4              ;
 4387 027164  000241                          CLC                     ;
 4388 027166  006005                          ROR     R5              ;
 4389 027170  006004                          ROR     R4              ;
 4390 027172  000524                          BR      SAVCNT          ; AND SAVE THE COUNT
 4391                                 ;
 4392                                 ; COME HERE IF NOT ENOUGH CLOCK INTERRUPTS OCCUR BEFORE THE COUNTERS OVERFLOW
 4393                                 ;
 4394 027174                          USCLOK: SETPRI  R2              ; RESTORE THE PRIORITY
 4395 027200                                  SETVEC  #100,#CLINT,#340; IGNORE FURTHER INTERRUPTS TO VECTOR 100
 4396 027226  005737  027526                  TST     CLKFLG          ; CAN WE DISABLE A CLOCK ?
 4397 027232  001402                          BEQ     NOCLOK          ; IF NOT, BRANCH
 4398 027234  005071  000000                  CLR     @(R1)           ; ELSE DISABLE CLOCK INTERRUPTS
 4399                                 ;
 4400                                 ; USE THE CONSOLE FOR TIMING
 4401                                 ;
 4402         177560                          TKS=177560              ; KEYBOARD STATUS REGISTER
 4403         177562                          TKB=177562              ; KEYBOARD DATA BUFFER
 4404         177564                          TPS=177564              ; PRINTER STATUS REGISTER
 4405         177566                          TPB=177566              ; PRINTER DATA BUFFER
 4406
 4407 027240                          NOCLOK: SETVEC  #60,#TTINT,#340 ; SET UP INTERRUPT VECTOR
 4408 027266                                  PRINTF  #TIMMSG         ; 'TYPE 2 CHARACTERS 6 SECONDS APART'
 4409
 4410 027306  105737  177560          10$:    TSTB    TKS             ; IS FIRST CHARACTER READY?
 4411 027312  100375                          BPL     10$             ; IF NOT, WAIT
 4412 027314  013700  177562                  MOV     TKB,R0          ; ELSE GET THE CHARACTER
 4413 027320  042700  177600                  BIC     #177600,R0      ; DISCARD UNWANTED BITS
 4414 027324  020027  000003                  CMP     R0,#3           ; IF ^C, RETURN TO SUPERVISOR
 4415 027330  001001                          BNE     20$             ;
 4416 027332                                  DOCLN                   ;
 4417
 4418 027334  013737  177562  177566  20$:    MOV     TKB,TPB         ; NOW ECHO THE CHARACTER
 4419 027342                                  SETPRI  #0              ; DROP THE PRIORITY
 4420 027350  012737  000100  177560          MOV     #100,TKS        ; ALLOW INTERRUPTS
 4421
 4422 027356  012705  000360          30$:    MOV     #240.,R5        ; SET UP MODULO 240 COUNTER
 4423 027362  005305                  40$:    DEC     R5              ; START COUNTING
 4424 027364  001376                          BNE     40$             ; R5 IS MODULO 240 COUNTER
 4425 027366  005204                          INC     R4              ; UPDATE THE COUNTER
 4426 027370  000772                          BR      30$             ; 6 SECONDS/240 = 25 MILLISECONDS
 4427
 4428 027372                          TTINT:  SETPRI  R2              ; RESTORE THE PRIORITY
 4429 027376                                  CLRVEC  #60             ; AND THE KEYBOARD VECTOR
 4430 027404  022626                          CMP     (SP)+,(SP)+     ; TIDY UP THE STACK
 4431 027406  005037  177560                  CLR     TKS             ; DISABLE INTERRUPTS
 4432 027412  013700  177562                  MOV     TKB,R0          ; ELSE GET THE CHARACTER
 4433 027416  042700  177600                  BIC     #177600,R0      ; DISCARD UNWANTED BITS
 4434 027422  020027  000003                  CMP     R0,#3           ; IF ^C, RETURN TO SUPERVISOR
 4435 027426  001001                          BNE     10$             ;
 4436 027430                                  DOCLN                   ;
 4437 027432  013737  177562  177566  10$:    MOV     TKB,TPB         ; ELSE, ECHO THE CHARACTER
 4438 027440  004737  014634                  JSR     PC,CRLF         ; AND PRINT A LINE FEED
 4439                                 ;
 4440                                 ; SAVE THE COUNTERS
 4441                                 ;
 4442 027444  010437  014626          SAVCNT: MOV     R4,CNT25M       ; SAVE THE 25 MILLISECONDS COUNTER
```

SETCLK - ROUTINE TO SET UP DELAY COUNTS

```
4443 027450  012700  000062                MOV     #50.,R0         ; NOW DIVIDE BY 50
4444 027454  062704  000031                ADD     #25.,R4         ; TO NEAREST 50
4445 027460  005001                        CLR     R1              ; INITIALISE RESULT
4446 027462  160004                10$:    SUB     R0,R4           ; REMAINDER < 0 ?
4447 027464  002402                        BLT     20$             ; IF YES, BRANCH
4448 027466  005201                        INC     R1              ; ELSE INCREMENT RESULT
4449 027470  000774                        BR      10$             ; AND TRY AGAIN
4450 027472  010137  014630          20$:    MOV     R1,CNT500       ; SAVE THE 500 MICROSECONDS COUNTER
4451
4452 027476  012700  000024                MOV     #20.,R0         ; NOW DIVIDE BY 20
4453 027502  062701  000012                ADD     #10.,R1         ; TO NEAREST 20
4454 027506  005002                        CLR     R2              ; INITIALISE RESULT
4455 027510  160001                30$:    SUB     R0,R1           ; REMAINDER < 0 ?
4456 027512  002402                        BLT     40$             ; IF YES, BRANCH
4457 027514  005202                        INC     R2              ; ELSE INCREMENT RESULT
4458 027516  0C0774                        BR      30$             ; AND TRY AGAIN
4459 027520  010237  014632          40$:    MOV     R2,CNT25        ; SAVE THE 25 MICROSECONDS COUNTER
4460
4461 027524  000207                SETEX:  RTS     PC              ; RETURN
4462
4463 027526  000000                CLKFLG: .WORD   0               ; SET IF DRS FINDS A CLOCK WITH A CSR
4464
4465                                       .NLIST  BEX
4466 027530     045     116     045 TIMMSG: .ASCIZ  /%N%A*YPE 2 CHARACTERS 6 SECONDS APART >/
4467                                       .LIST   BEX
4468                                       .EVEN
```

```
                                .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.
                                ;--

027600                                  BGNAUTO

027600                                  SETVEC  #4,#NXM,#PRI07  ; SET UP NON - EXISTENT MEMORY TRAP VECTOR.
027626  005037  002470                  CLR     NXMFLG          ; CLEAR NON - EXISTENT MEMORY FLAG
027632  005777  152560                  TST     @#MOD           ; 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).

027636  005737  002470                  TST     NXMFLG          ; WAS THERE A TRAP ?
027642  001404                          BEQ     10$             ; BRANCH IF NOT
027644                                  DODU    L$LUN           ; ELSE DROP THE DEVICE
027652                                  DOCLN                   ; CLEAN UP CODE.
027654                          10$:    CLRVEC  #4              ; RETURN VECTOR 04 TO NORMAL STATE
027662                                  ENDAUTO
```

```
                                .SBTTL  CLEANUP CODING SECTION

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

027664                                  BGNCLN


                                        BRESET                  ; DO A BUS RESET TO SWITCH OFF ALL LEDS
027665                                  EXIT    CLN


                                        .EVEN

027672                                  ENDCLN
```

```
                                .SBTTL  DROP UNIT SECTION

                                ;++
                                ; THE DROP-UNIT SECTION CONTAINS THE CODING THAT CAUSES A DEVICE
                                ; TO NO LONGER BE TESTED.
                                ;--

027674                                  BGNDU


027674  010026                          MOV     R0,(SP)+                ; SAVE R0 CONTENTS
027676  112760  000001  002614          MOVB    #1,DROPED(R0)           ; FLAG UNIT DROPPED IN PARAM TABLE
027704  014600                          MOV     -(SP),R0                ; GET ORIGINAL R0 CONTENTS
027706                                  PRINTF  #DROPD,R0               ; 'UNIT DROPPED'


027730                                  EXIT    DU


                                        .NLIST  BEX
027734      045     116     045 DROPD:  .ASCIZ  /%N%AUNIT %D2%A DROPPED/
                                        .LIST   BEX
                                        .EVEN

027764                                  ENDDU
```

```
                                .SBTTL  ADD UNIT SECTION

                                ;++
                                ; THE ADD-UNIT SECTION CONTAINS ANY CODE THE PROGRAMMER WISHES
                                ;  TO BE EXECUTED IN CONJUNCTION WITH THE ADDING OF A UNIT BACK
                                ;  TO THE TEST CYCLE.
                                ;--

027766                                  BGNAU

027766  105060  002614                  CLRB    DROPED(R0)              ; FLAG UNIT NOT DROPPED IN PARAM TABLE

027772                                  EXIT    AU

                                .EVEN

027776                                  ENDAU

030000                                  ENDMOD
```

```
                                .TITLE HARDWARE TESTS

                                .SBTTL  TEST 1: Register NXM Test.

030000                                  BGNMOD
                                ;*******************************************************************************
                                ;               Test 1 - Register NXM Test.
                                ;
                                ;       FOR TEST DESCRIPTION SEE 6.0 TEST SUMMERIES
                                ;*******************************************************************************

030000                                  BGNTST
030000  004737  015006                  CALL    SELECT                  ;CALL SELECT ROUTINE
030004  000420                          .WORD   420                     ;GIVE TEST PARAMETER
030006  030206                          TSHD1                           ;GIVE TEST HEADER ADDRESS
030010  103467                          BCS     EXQV1                   ;IF CARRY IS SET, EXIT TEST
030012  005037  002504                  CLR     ITRCNT                  ;CLEAR ITERATION COUNTER
030016                                  SETVEC  #4,#LOCATE,#PRI07       ;SET UP INTERRUPT ROUTINE
030044  013701  002416          ITRAC1: MOV     MOD,R1                  ;GET FIRST REGISTER ADDRESS
030050  162701  000002                  SUB     #2,R1                   ;
030054  012702  000004                  MOV     #4,R2                   ;SET COUNTER FOR 4 REGISTERS
030060  005003                          CLR     R3                      ;CLEAR LOCATION FOR ERROR MARK
030062  062701  000002          10$:    ADD     #2,R1                   ;GET REGISTER ADDRESS
030066                                  BGNSEG                          ;
030070  005004                          CLR     R4                      ;
030072  005711                          TST     (R1)                    ;TEST REGISTER ADDRESS
030074  005704                          TST     R4                      ;WAS THERE A TRAP?
030076  004737  015240                  CALL    INSERT                  ;SKIP BRANCH IF "SFI" IS SET
030102  001407                          BEQ     20$                     ;IF NO, BRANCH
030104  004737  015240                  CALL    INSERT                  ;SKIP BRANCH IF "SFI" IS SET
030110  005203                          INC     R3                      ;MARK THE ERROR
030112                                  ERRHRD  101,E101,EERA           ;ERROR HANDLER
030122                          20$:    ENDSEG                          ;
030124  005302                          DEC     R2                      ;ALL REGISTERS TESTED
030126  001355                          BNE     10$                     ;IF NO, BRANCH
030130  005703                          TST     R3                      ;WAS THERE AN ERROR
030132  001404                          BEQ     30$                     ;IF NO, DON'T DROP THE UNIT
030134                                  DODU    L$LUN                   ;DROP THE UNIT UNDER TEST
030142                                  DOCLN                           ;RUN THE CLEAN UP ROUTINE
030144  005737  002164          30$:    TST     QVP                     ;IS QUICK VERIFY PASS SELECTED?
030150  001007                          BNE     EXQV1                   ;IF YES EXIT TEST
030152  005237  002504                  INC     ITRCNT                  ;ITERATION COUNTER + 1
030156  023737  002506  002504          CMP     ITRDEF,ITRCNT           ;DEFAULT ITERATION EXECUTED
030164  001401                          BEQ     EXQV1                   ;IF YES EXIT TEST
030166  000726                          BR      ITRAC1                  ;IF NO, TEST ITERATION
030170                          EXQV1:  CLRVEC  #4                      ;
030176                          TSTEN1: EXIT    TST

030202                                  BGNSRV  LOCATE                  ;SERVICE ROUTINE LOCATE
030202  005204                          INC     R4                      ;INCREMENTS R4 IF A TRAP TO 4
030204                                  ENDSRV                          ;HAS OCCURRED

                                .NLIST  BEX
030206  045 123 062             TSHD1:: .ASCIZ  /%S2%AREGISTER NXM TEST.%N/
030240  122 105 107             E101:   .ASCIZ  /REGISTER ADDRESSING ERROR - TRAP TO 4/
                                .LIST   BEX
                                .EVEN
```

TEST 1: Register NXM Test.

```
4755 030306                                    ENDTST
```

```
4757                                    .SBTTL  TEST 2: Reset Test
4758                                    ;*************************************************************************
4759                                    ;               Test 2 - Reset Test.
4760                                    ;
4761                                    ;       FOR TEST DESCRIPTION SEE 6.0 TEST SUMMERIES
4762                                    ;*************************************************************************
4763 030310                                    BGNTST
4764 030310  004737  015006                    CALL    SELECT                  ;CALL SELECT ROUTINE
4765 030314  000420                            .WORD   420                     ;GIVE TEST PARAMETER
4766 030316  030574                            TSHD2                           ;GIVE TEST HEADER ADDRESS
4767 030320  103523                            BCS     EXQV2                   ;IF CARRY IS SET, EXIT TEST
4768 030322  005037  002504                    CLR     ITRCNT                  ;CLEAR ITERATION COUNTER
4769 030326  005001                    ITRAC2: CLR     R1                      ;CLEAR TEMPORARY STORE
4770                                    ;       MOVB    CONMSK,R1               ;GET MODULE TYPE (FOR LATER USE)
4771 030330  012701  000001                    MOV     #1,R1                   ;CNMASK FOR 5 CHA. COUNTER
4772 030334  012702  177777            10$:    MOV     #-1,R2                  ;MODULE IDENTIFICATION
4773 030340  005202                    20$:    INC     R2                      ;...
4774 030342  006201                            ASR     R1                      ;...
4775 030344  103375                            BCC     20$                     ;...
4776 030346  006302                            ASL     R2                      ;MULTYPLY BY 10 TO GET TABLE
4777 030350  006302                            ASL     R2                      ;OFFSET
4778 030352  006302                            ASL     R2                      ;...
4779 030354  012703  000004                    MOV     #4,R3                   ;START WITH INR REGISTER
4780 030360                                    BGNSEG
4781 030362                                    BRESET                          ;DO A BUS RESET
4782 030364  032777  000100  152024            BIT     #100,@MOD               ;IS LED BIT CLEARED ?
4783 030372  004737  015240                    CALL    INSERT                  ;SKIP BRANCH IF "SFI" IS SET
4784 030376  001404                            BEQ     30$                     ;BRANCH IF YES
4785 030400                                    ERRSOFT 201,E201,EERG           ;ERROR HANDLER
4786 030410                            30$:    CKLOOP                          ;
4787 030412  052777  000100  151776            BIS     #100,@MOD               ;SWITCH ON THE MODULE LED
4788 030420  032777  000100  151770            BIT     #100,@MOD               ;IS LED BIT NOW SET ?
4789 030426  004737  015240                    CALL    INSERT                  ;SKIP BRANCH IF "SFI" IS SET
4790 030432  001004                            BNE     40$                     ;BRANCH IF YES
4791 030434                                    ERRSOFT 202,E202,EERG           ;ERROR HANDLER
4792 030444                            40$:    ENDSEG                          ;
4793 030446  013705  002424                    MOV     INR,R5                  ;GET INR REGISTER ADDRESS
4794 030452  062702  000006                    ADD     #6,R2                   ;POINT TABLE OFFSET TO INR CONT.
4795 030456                                    BGNSEG
4796 030460  016237  031016  002474    50$:    MOV     RSAV(R2),GOOD           ;GET FIRST COMPARE CONTENTS FROM TABLE
4797 030466  011537  002476                    MOV     @R5,BAD                 ;GET FIRST REGISTER CONTENTS
4798 030472  023737  002474  002476            CMP     GOOD,BAD                ;CMP TABLE CONT.WITH REG. CONT.
4799 030500  004737  015240                    CALL    INSERT                  ;SKIP BRANCH IF "SFI" IS SET
4800 030504  001404                            BEQ     70$                     ;
4801 030506                                    ERRSOFT 203,E203,EERB           ;ERROR HANDLER
4802 030516                            70$:    CKLOOP                          ;
4803 030520  022703  000004                    CMP     #4,R3                   ;ALL REGISTER TESTED ?
4804 030524  001406                            BEQ     80$                     ;BRANCH IF YES
4805 030526  005203                            INC     R3                      ;COUNTER FOR NEXT REGISTER
4806 030530  062702  000002                    ADD     #2,R2                   ;LOAD NEXT TABLE ADDR.
4807 030534  062705  000002                    ADD     #2,R5                   ;LOAD NEXT REGISTER ADDR.
4808 030540  000747                            BR      50$                     ;
4809 030542                            80$:    ENDSEG                          ;
4810 030544  005737  002164                    TST     QVP                     ;IS QUICK VERIFY PASS SELECTED?
4811 030550  001007                            BNE     EXQV2                   ;IF YES EXIT TEST
4812 030552  005237  002504                    INC     ITRCNT                  ;ITERATION COUNTER + 1
4813 030556  022737  000002  002504            CMP     #2,ITRCNT               ;TWO ITERATION EXECUTED
```

TEST 2: Reset Test

```
4814 030564  001401                          BEQ     EXQV2                   ;IF YES EXIT TEST
4815 030566  000657                          BR      ITRAC2                  ;IF NO, TEST ITERATION
4816 030570                          EXQV2:  EXIT    TST
4817                                         .NLIST  BEX
4818 030574     045     123     062  TSHD2:: .ASCIZ  /#S2#ARESET TEST.#N/
4819 030617     114     105     104  E201:   .ASCIZ  /LED BIT IN MOD REGISTER NOT CLEARED AFTER BUS RESET /
4820 030704     114     105     104  E202:   .ASCIZ  /LED BIT IN MOD REGISTER CAN'T BE SET/
4821 030751     122     105     107  E203:   .ASCIZ  /REGISTER INCORRECT AFTER BUS RESET /
4822                                         .EVEN
4823
4824                                 ;               1.REG., 2.REG., 3., 4.
4825 031016  000000  000000  000000  RSAV:   .WORD    0,      0,      0, 60340 ;REG. MASKS FOR 5 CHA. COUNTER
4826                                         .LIST   BEX
4827                                         .EVEN
4828 031026                                  ENDTST
```

```
                                .SBTTL  TEST 3: Register R/W Bit Test.
                                ;*****************************************************************
                                ;               Test 3 - Register R/W Bit Test.
                                ;
                                ;       FOR TEST DESCRIPTION SEE 6.0 TEST SUMMERIES
                                ;*****************************************************************
031030                                  BGNTST
031030  004737  015006                  CALL    SELECT                  ;CALL SELECT ROUTINE
031034  000420                          .WORD   420                     ;TEST SELECT MASK
031036  031216                          TSHD3                           ;TEST HEADER ADDRESS
031040  103002                          BCC     1$                      ;IF CARRY IS SET, DON'T BRANCH
031042                                  EXIT    TST                     ;EXIT TEST IF CARRY IS SET
031046  005037  002504          1$:     CLR     ITRCNT                  ;CLEAR ITERATION COUNTER
031052                          ITRAC3: SETPRI  #PRI07                  ;DISABLE INTERRUPTS
031060  005001                          CLR     R1                      ;CLEAR TEMPORARY STORE
                                ;       MOVB    CONMSK,R1               ;GET MODULE TYPE (FOR LATER USE)
031062  012701  000001                  MOV     #1,R1                   ;MASK FOR 5 CHA. COUNTER
031066  012702  177777                  MOV     #-1,R2                  ;
031072  005202                  10$:    INC     R2                      ;MODULE IDENTIFICATION
031074  006201                          ASR     R1                      ;...
031076  103375                          BCC     10$                     ;...
031100  006302                          ASL     R2                      ;MULTYPLY BY 8. TO GET TABLE
031102  006302                          ASL     R2                      ;OFFSET
031104  006302                          ASL     R2                      ;
031106  012703  000001                  MOV     #1,R3                   ;START WITH FIRST REGISTER
031112  013737  002416  014052          MOV     MOD,REGADD              ;LOAD FIRST REGISTER ADDRESS
031120  012737  000455  005714  20$:    MOV     #301.,ERRNBR            ;LOAD FIRST ERROR NUMBER
031126  016237  031242  014046          MOV     RWMAK(R2),MASK          ;GET R/W MASK FORM TABLE
031134  004737  013532                  CALL    REGTS1                  ;CALL REGISTER TEST
031140  022703  000004                  CMP     #4,R3                   ;4 REGISTERS TESTED ?
031144  001407                          BEQ     30$                     ;BRANCH IF YES
031146  005203                          INC     R3                      ;COUNTER FOR NEXT REGISTER
031150  062702  000002                  ADD     #2,R2                   ;LOAD NEXT TABLE ADDR.
031154  062737  000002  014052          ADD     #2,REGADD               ;NEXT REGISTER ADDRESS
031162  000756                          BR      20$                     ;TEST AGAIN WITH NEW PARAMETER
                                
                                
031164  005737  002164          30$:    TST     QVP                     ;IS QUICK VERIFY PASS SELECTED?
031170  001010                          BNE     EXQV3                   ;IF YES, EXIT TEST
031172  005237  002504                  INC     ITRCNT                  ;ITERATION COUNTER + 1
031176  023737  002506  002504          CMP     ITRDEF,ITRCNT           ;DEFAULT ITERATION EXECUTED
031204  001402                          BEQ     EXQV3                   ;IF YES, EXIT TEST
031206  000137  031052                  JMP     ITRAC3                  ;IF NO, TEST ITERATION
031212                          EXQV3:  EXIT TST
                                
                                        .NLIST  BEX
031216      045     123     062 TSHD3:: .ASCIZ  /%S2%AR-W BIT TEST%N/
                                        .LIST   BEX
                                        .EVEN
                                
                                ;                               1.REG., 2.REG., 3.   ,4.
031242  000000  000000  000000 RWMAK:  .WORD    0,      0,      0   , 117400 ;R/W MASKS FOR 5 CHA. CNT
031250  117400
                                        .EVEN
031252                                  ENDTST
```

```
                                        .SBTTL TEST 4  --  AM 9513 SUBREGISTER TESTS
                                        ;*****************************************************************
                                        ;       TEST 4  -- AM  9513 SUBREGISTER TESTS
                                        ;
                                        ;       FOR TEST DESCRIPTION SEE 6.0 TEST SUMMERIES
                                        ;*****************************************************************

                                                .DSABLE LSB
031254                                          BGNTST
031254  004737  015006                          CALL    SELECT                  ;CALL SELECT ROUTINE
031260  000420                                  .WORD   420                     ;TEST SELECT MASK
031262  032120                                  TSHD4                           ;TEST HEADER ADDRESS
031264  103002                                  BCC     1$                      ;IF CARRY IS SET, DON'T BRANCH
031266                                          EXIT    TST                     ;EXIT TEST IF CARRY IS SET
031272  005037  002504                  1$:     CLR     ITRCNT                  ;CLEAR ITERATION COUNTER
031276  013701  002416                          MOV     MOD,R1                  ;GET FIRST REGISTER ADDRESS
031302  004737  021470                          JSR     PC,AMREST               ;RESET THE MODULE
031306  013701  002416                  ITRAC4: MOV     MOD,R1                  ;GET FIRST REGISTER ADDRESS
031312  005037  002470                          CLR     NXMFLG                  ;TO DETECT IF A TRAP OCCURRED
                                        ;
                                        ; TEST LOADING CONTROL & STATUS REGISTER AND READING RESULTS
                                        ;
031316  012737  000002  002434                  MOV     #2,BITMSK               ;BIT MASK
031324  012737  000001  002442                  MOV     #1,NUMBER               ;COUNTER CHANNEL

031332                                  10$:    BGNSEG                          ;BEGINNING OF LOOP ON ERROR SEGMENT
031334  012700  177400                          MOV     #B,R0                   ;LOAD MASK FOR MODE REGISTER AND
031340  063700  002442                          ADD     NUMBER,R0               ;ADD CHANNEL NR
031344  010061  000002                          MOV     R0,MREA(R1)             ;SELECT MODE REGISTER
031350  012761  000042  000004                  MOV     #M$TOG,MREB(R1)         ;SET TO TOGGLE MODE
031356  012700  177740                          MOV     #C$CTN,R0               ;GET TOUT COMMAND
031362  063700  002442                          ADD     NUMBER,R0               ;ADD CHANNEL NUMBER
031366  010061  000002                          MOV     R0,MREA(R1)             ;LOAD INTO MREA (CSR)
031372  016102  000002                          MOV     MREA(R1),R2             ;GET RESULT
031376  033702  002434                          BIT     BITMSK,R2               ;TEST THAT OUTX BIT IS CLEARED
031402  004737  015240                          CALL    INSERT                  ;TEST IF ERROR MESSAGE IS REQUIRED
031406  001405                                  BEQ     20$                     ;OK
031410                                          ERRHRD  401,,ERR104             ;ERROR HANDLER
031420                                          CKLOOP                          ;IF LOOP ON ERROR
031422                                  20$:    ENDSEG                          ;END SEGMENT
031424                                          BGNSEG                          ;BEGINNING OF LOOP ON ERROR SEGMENT
031426  012700  177750                          MOV     #C$STN,R0               ;SET TOUT COMMAND
031432  063700  002442                          ADD     NUMBER,R0               ;BIT NUMBER (1-5)
031436  010061  000002                          MOV     R0,MREA(R1)             ;LOAD INTO MREA
031442  016102  000002                          MOV     MREA(R1),R2             ;GET RESULT
031446  033702  002434                          BIT     BITMSK,R2               ;TEST THAT OUTX BIT IS SET
031452  004737  015240                          CALL    INSERT                  ;TEST IF ERROR MESSAGE IS REQUIRED
031456  001005                                  BNE     30$                     ;OK
031460                                          ERRHRD  402,,ERR105             ;ERROR HANDLER
031470                                          CKLOOP                          ;IF LOOP ON ERROR
031472                                  30$:    ENDSEG                          ;END SEGMENT
                                        ;
031474                                          BGNSEG                          ;BEGINNING OF LOOP ON ERROR SEGMENT
031476  012700  177740                          MOV     #C$CTN,R0               ;CLEAR TOUT COMMAND
031502  063700  002442                          ADD     NUMBER,R0               ;BIT NUMBER (1-5)
031506  010061  000002                          MOV     R0,MREA(R1)             ;LOAD INTO MREA
031512  016102  000002                          MOV     MREA(R1),R2             ;GET RESULT
```

```
4943 031516  033702  002434                    BIT     BITMSK,R2               ;TEST THAT OUT BIT IS CLEARED AGAIN
4944 031522  004737  015240                    CALL    INSERT                  ;TEST IF ERROR MESSAGE IS REQUIRED
4945 031526  001405                            BEQ     40$                     ;OK
4946 031530                                    ERRHRD  403,,ERR104             ;ERROR HANDLER
4947 031540                                    CKLOOP                          ;IF LOOP ON ERROR
4948 031542                              40$:  ENDSEG                          ;END SEGMENT
4949
4950 031544  006337  002434                    ASL     BITMSK                  ;SHIFT MASK BIT
4951 031550  005237  002442                    INC     NUMBER                  ;GET NEXT CHA. NUMBER
4952 031554  023727  002442  000005            CMP     NUMBER,#5               ;ALL CHA. DONE ?
4953 031562  003663                            BLE     10$                     ;NO, THEN BRANCH
4954                                           ;
4955                                           ; LOAD AND READ LOAD REGISTERS
4956                                           ;
4957 031564  012737  177410  002450            MOV     #8!10,ELEMNT            ;GET LOAD REGISTER MASK
4958 031572  012737  000001  002446            MOV     #1,GROUP                ;GET CHA. NUMBER
4959 031600  012737  000002  002444  50$:       MOV     #2,COUNTR               ;INITIAL VALUE TO LOAD
4960
4961 031606                                    BGNSEG
4962 031610  013700  002450              60$:  MOV     ELEMNT,R0               ;BUILD ADDRESS
4963 031614  063700  002446                    ADD     GROUP,R0                ;ADD CHA. NUMBER
4964 031620  010061  000002                    MOV     R0,MREA(R1)             ;SELECT ADDRESS
4965 031624  013761  002444  000004            MOV     COUNTR,MREB(R1)         ;LOAD VALUE
4966 031632  010061  000002                    MOV     R0,MREA(R1)             ;SELECT ADDRESS
4967 031636  016103  000004                    MOV     MREB(R1),R3             ;GET RESULT
4968 031642  020337  002444                    CMP     R3,COUNTR               ;IS IT THE SAME VALUE?
4969 031646  004737  015240                    CALL    INSER                   ;TEST IF ERROR MESSAGE IS REQUIRED
4970 031652  001410                            BEQ     70$                     ;OK
4971                                           ;
4972                                           ; ERROR DETECTED
4973                                           ;
4974 031654                                    ERRHRD  404,,ERR106             ;ERROR HANDLER
4975 031664                                    CKLOOP                          ;IF LOOP ON ERROR
4976 031666  004737  015240                    CALL    INSERT                  ;SKIP BRANCH IF "SFI" IS SET
4977 031672  000404                            BR      73$                     ;TAKE NEXT CHANNEL
4978
4979 031674  062737  000201  002444  70$:       ADD     #201,COUNTR             ;BUMP COUNTER
4980 031702  103342                            BCC     60$                     ;GO UNTIL OVERFLOW
4981 031704                              73$:  ENDSEG
4982 031706  005237  002446                    INC     GROUP                   ;BUMP CHA.NUMBER
4983 031712  023727  002446  000005            CMP     GROUP,#5                ;HIGH LIMIT
4984 031720  003727                            BLE     50$                     ;GO FOR NEXT CHANNEL
4985                                           ;
4986                                           ; LOAD AND READ HOLD REGISTERS
4987                                           ;
4988 031722  012737  177420  002450            MOV     #8!20,ELEMNT            ;GET HOLD REGISTER MASK
4989 031730  012737  000001  002446            MOV     #1,GROUP                ;GET CHANNEL NUMBER
4990 031736  012737  000002  002444  80$:       MOV     #2,COUNTR               ;INITIAL VALUE TO LOAD
4991
4992 031744                                    BGNSEG
4993 031746  013700  002450              90$:  MOV     ELEMNT,R0               ;BUILD ADDRESS
4994 031752  063700  002446                    ADD     GROUP,R0                ;ADD CHANNEL NUMBER
4995 031756  010061  000002                    MOV     R0,MREA(R1)             ;SELECT ADDRESS
4996 031762  013761  002444  000004            MOV     COUNTR,MREB(R1)         ;LOAD VALUE
4997 031770  010061  000002                    MOV     R0,MREA(R1)             ;SELECT ADDRESS
4998 031774  016103  000004                    MOV     MREB(R1),R3             ;GET RESULT
4999 032000  020337  002444                    CMP     R3,COUNTR               ;IS IT THE SAME VALUE?
```

```
 5000 032004  004737  015240                    CALL    INSERT                  ;TEST IF ERROR MESSAGE IS REQUIRED
 5001 032010  001410                            BEQ     100$                    ;OK
 5002                                   ;
 5003                                   ; ERROR DETECTED
 5004                                   ;
 5005 032012                                    ERRHRD  405,,ERR107             ;ERROR HANDLER
 5006 032022                                    CKLOOP                          ;IF LOOP ON ERROR
 5007 032024  004737  015240                    CALL    INSERT                  ;SKIP BRANCH IF "SFI" IS SET
 5008 032030  000404                            BR      110$                    ;LEAVE TEST
 5009
 5010 032032  062737  000201  002444  100$:     ADD     #201,COUNTR             ;BUMP COUNTER
 5011 032040  103342                            BCC     90$                     ;GO UNTIL OVERFLOW
 5012 032042                            110$:   ENDSEG
 5013 032044  005237  002446                    INC     GROUP                   ;BUMP GROUP NUMBER
 5014 032050  023727  002446  000005            CMP     GROUP,#5                ;HIGH LIMIT
 5015 032056  003727                            BLE     80$                     ;GO FOR NEXT GROUP
 5016
 5017 032060  005737  002164                    TST     QVP                     ;IS QUICK VERIFY PASS SELECTED
 5018 032064  001010                            BNE     EXS4                    ;YES
 5019 032066  005237  002504                    INC     ITRCNT                  ;BUMP ITERATION COUNT
 5020 032072  023737  002506  002504            CMP     ITRDEF,ITRCNT           ;DEFAULT ITERATION COUNT
 5021 032100  001402                            BEQ     EXS4                    ;TIME TO EXIT
 5022 032102  000137  031306                    JMP     ITRAC4                  ;DO THE TEST AGAIN
 5023
 5024 032106                            EXS4:   CLRVEC  #4                      ;RESET TRAP VECTOR
 5025 032114                                    EXIT    TST
 5026
 5027                                           .NLIST  BEX
 5028 032120      045     123     062   TSHD4::.ASCIZ   /%S2%AAM 9513 SUBREGISTER TEST %N/
 5029                                           .LIST   BEX
 5030                                           .EVEN
 5031
 5032 032162                                    ENDTST
 5033
```

```
5035
5036                                    .SBTTL TEST 5  --  INTERRUPT TEST
5037                                    ;*****************************************************************************
5038                                    ;       TEST 5  --  INTERRUPT TEST
5039                                    ;
5040                                    ;       FOR TEST DESCRIPTION SEE 6.0 TEST SUMMERIES
5041                                    ;*****************************************************************************
5042
5043 032164                                    BGNTST
5044 032164  004737  015006                     CALL    SELECT                  ;CALL SELECT ROUTINE
5045 032170  000420                             .WORD   420                     ;TEST SELECT MASK
5046 032172  033474                             TSHD5                           ;TEST HEADER ADDRESS
5047 032174  103002                             BCC     1$                      ;IF CARRY IS SET, DON'T BRANCH
5048 032176                                     EXIT    TST                     ;EXIT TEST IF CARRY IS SET
5049 032202  005037  002504            1$:      CLR     ITRCNT                  ;CLEAR ITERATION COUNTER
5050
5051 032206  004737  021470                     JSR     PC,AMREST               ;RESET THE MODULE
5052 032212                                     SETVEC  VEC,#INTSR,#PRI07       ;SETUP INTERRUPT SERVICE ROUTINE
5053 032240                                     CLRVEC  #400                    ;CATCH SUPERVISOR BUG
5054 032246                                     SETVEC  VEC,#INTSR,#PRI07       ;...
5055 032274                                     SETVEC  #4,#BADIV,#PRI07        ;SETUP TO CATCH TRAPS
5056 032322                                     GETPRI  SAVPRI                  ; SAVE CURRENT PRIORITY IN R2
5057 032330                                     SETPRI  #0                      ; AND DROP THE PRIORITY
5058 032336  005037  002510            ITRAC5: CLR     INTFLA                  ;INIT INTERRUPT FLAG
5059 032342  005037  002470                     CLR     NXMFLG                  ;TO DETECT IF A TRAP OCCURRED
5060 032346  005037  002472                     CLR     #IV                     ;ZERO COUNT
5061                                    ;
5062                                    ; TEST 'IR' BITS WITHOUT MASTER ENABLE SET
5063                                    ;
5064 032352  012737  000400  002434             MOV     #400,BITMSK             ;MASK FOR 'EN' BITS
5065 032360  012737  000001  002474             MOV     #1,GOOD                 ;MASK FOR 'IR' BITS
5066 032366  012737  000001  002442             MOV     #1,NUMBER               ;MASK FOR CHANNEL NUMBER
5067
5068 032374                            10$:     BGNSEG                          ;BEGINNING OF LOOP ON ERROR SEGMENT
5069 032376  005077  150022                     CLR     @INR                    ;DISABLE ENABLE BITS
5070
5071 032402  013777  002442  150010             MOV     NUMBER,@CSR             ;SELECT MODE REGISTER AND CHANNEL
5072 032410  012777  000042  150004             MOV     #M$TOG,@CCR             ;LOAD TOGGLE MODE
5073 032416  012702  177740                     MOV     #C$CTN,R2               ;GET CLEAR TOUT COMMAND
5074 032422  063702  002442                     ADD     NUMBER,R2               ;ADD CHANNEL NUMBER
5075 032426  010277  147766                     MOV     R2,@CSR                 ;OUTPUT TO DEFINATE STATE (CLEAR TOUT)
5076 032432  012702  177750                     MOV     #C$STN,R2               ;GET SET TOUT COMMAND
5077 032436  063702  002442                     ADD     NUMBER,R2               ;ADD CHANNEL NUMBER
5078 032442  010277  147752                     MOV     R2,@CSR                 ;SET TOUT (COURS IR SET)
5079 032446  017737  147752  002476             MOV     @INR,BAD                ;GET INTERRUPT REGISTER CONTENTS
5080 032454  022737  060340  002476             CMP     #60340,BAD              ;IR BITS SHOULD BE CLEARED
5081 032462  004737  015240                     CALL    INSERT                  ;TEST IF ERROR MESSAGE IS REQUIRED
5082 032466  001406                             BEQ     20$                     ;OK
5083 032470  012702  060340                     MOV     #60340,R2               ;SET UP FOR ERROR MESSAGES
5084 032474                                     ERRHRD  501,,ERR501             ;ERROR HANDLER
5085 032504                            20$:     ENDSEG                          ;END SEGMENT
5086
5087 032506                                     BGNSEG
5088 032510  013777  002434  147706             MOV     BITMSK,@INR             ;SET ENABLE BIT
5089 032516  013777  002442  147674             MOV     NUMBER,@CSR             ;SELECT MODE REGISTER AND CHANNEL
5090 032524  012777  000042  147670             MOV     #M$TOG,@CCR             ;SET TO TOGGLE MODE
5091 032532  012702  177740                     MOV     #C$CTN,R2               ;GET CLEAP TOUT COMMAND
```

```
5092 032536  063702  002442           ADD     NUMBER,R2               ;ADD CHANNEL NUMBER
5093 032542  010277  147652           MOV     R2,@CSR                 ;OUTPUT TO DEFINATE STATE (CLEAR TOUT)
5094 032546  012702  177750           MOV     #C$STN,R2               ;GET SET TOUT COMMAND
5095 032552  063702  002442           ADD     NUMBER,R2               ;ADD CHANNEL NUMBER
5096 032556  010277  147636           MOV     R2,@CSR                 ;SET TOUT (COUSE IR SET)
5097 032562  012777  000037  147634   MOV     #37,@INR                ;CLEAR EN BIT ONLY (LEAVE IR BITS)
5098 032570  017737  147630  002476   MOV     @INR,BAD                ;GET INTERRUPT REGISTER CONTENTS
5099 032576  013702  002474           MOV     GOOD,R2                 ;SET UP FOR ERROR MESSAGES
5100 032602  052702  060340           BIS     #60340,R2               ;SET UNUSED BITS
5101 032606  020237  002476           CMP     R2,BAD                  ;IR BITS SHOULD BE SET
5102 032612  004737  015240           CALL    INSERT                  ;TEST IF ERROR MESSAGE IS REQUIRED
5103 032616  001404                   BEQ     30$                     ;OK
5104 032620                           ERRHRD  502,,ERR501             ;ERROR HANDLER
5105 032630                   30$:    ENDSEG
5106
5107 032632                           BGNSEG
5108 032634  043777  002474  147562   BIC     GOOD,@INR               ;CLEAR IR BIT
5109 032642  017737  147556  002476   MOV     @INR,BAD                ;GET INR CONTENTS
5110 032650  022737  060340  002476   CMP     #60340,BAD              ;IR BIT SOULD BE CLEARED
5111 032656  004737  015240           CALL    INSERT                  ;TEST IF ERROR MESSAGE IS REQUIRED
5112 032662  001406                   BEQ     40$                     ;OK
5113 032664  012702  060340           MOV     #60340,R2               ;SET UP FOR ERRO MESSAGES
5114 032670                           ERRHRD  503,,ERR502             ;ERROR HANDLER
5115 032700                   40$:    ENDSEG
5116
5117 032702  006337  002434           ASL     BITMSK                  ;SHIFT 'EN' MASK BIT
5118 032706  006337  002474           ASL     GOOD                    ;SHIFT 'IR' MASK BIT
5119 032712  005237  002442           INC     NUMBER                  ;INCREMENT CHA. BIT NUMBER
5120 032716  023727  002442  000005   CMP     NUMBER,#5               ;ALL 5 CHANNELS DONE ?
5121 032724  003002                   BGT     50$                     ;BRANCH IF NO
5122 032726  000137  032374           JMP     10$                     ;YES
5123                           ;
5124                           ; NO INTERRUPTS SHOULD HAVE BEEN GENERATED UP TO THIS POINT
5125                           ;
5126 032732  005737  002510   50$:    TST     INTFLA                  ;TEST INTERRUPT COUNT
5127 032736  004737  015240           CALL    INSERT                  ;TEST IF ERROR MESSAGE IS REQUIRED
5128 032742  001404                   BEQ     60$                     ;OK
5129 032744                           ERRHRD  504,,ERR503             ;ERROR HANDLER
5130                           ;
5131                           ; TEST THAT INTERRUPTS CAN BE GENERATED
5132                           ;
5133
5134
5135 032754                   60$:    BGNSEG
5136 032756  012777  117400  147440   MOV     #117400,@INR            ;***SET MASTER ENABLE AND INT ENB****
5137 032764  000240                   NOP                             ;VERY SHORT WAIT
5138 032766  005737  002510           TST     INTFLA                  ;NO INTERRUPT SHOULD APEARE
5139 032772  004737  015240           CALL    INSERT                  ;TEST IF ERROR MESSAGE IS REQUIRED
5140 032776  001404                   BEQ     70$                     ;OK
5141 033000                           ERRHRD  505,,ERR503             ;ERROR HANDLER
5142
5143 033010                   70$:    ENDSEG
5144 033012  012737  000400  002434   MOV     #400,BITMSK             ;MASK FOR 'EN' BITS
5145 033020  012737  000001  002474   MOV     #1,GOOD                 ;MASK FOR 'IR' BITS
5146 033026  012737  000001  002442   MOV     #1,NUMBER               ;MASK FOR CHANNEL NUMBER
5147
5148 033034                   200$:   BGNSEG                          ;BEGINNING OF LOOP ON ERROR SEGMENT
```

```
5149
5150 033036  012701  000340                 MOV     #PRI07,R1          ;LOAD R1 WITH PRIORITY NUMBER
5151 033042                                 SETPRI  #PRI07             ;DISABLE INTERRUPTS AT THIS POINT
5152 033050  005037  002472                 CLR     BIV                ;
5153 033054  005037  002510                 CLR     INTFLA             ;CLEAR INTERRUPT FLAG
5154 033060  052777  100000  147336         BIS     #100000,@INR       ;SET MASTER ENABLE IN INR
5155 033066  013777  002442  147324         MOV     NUMBER,@CSR        ;SELECT MODE REGISTER AND CHANNEL
5156 033074  012777  000042  147320         MOV     #M$TOG,@CCR        ;SET TO TOGGLE MODE
5157 033102  012702  177740                 MOV     #C$CTN,R2          ;GET CLEAR TOUT COMMAND
5158 033106  063702  002442                 ADD     NUMBER,R2          ;ADD CHANNEL NUMBER
5159 033112  010277  147302                 MOV     R2,@CSR            ;OUTPUT TO DEFINATE STATE (CLEAR TOUT)
5160 033116  012702  177750                 MOV     #C$STN,R2          ;GET SET TOUT COMMAND
5161 033122  063702  002442                 ADD     NUMBER,R2          ;ADD CHANNEL NUMBER
5162 033126  010277  147266                 MOV     R2,@CSR            ;SET TOUT (GENERATE INTERRUPT)
5163
5164 033132  023727  002510  000001  110$:  CMP     INTFLA,#1          ;DID INTERRUPT OCCUR
5165 033140  004737  015240                 CALL    INSERT             ;TEST IF ERROR MESSAGE IS REQUIRED
5166 033144  001431                         BEQ     90$                ;OK
5167 033146  005737  002472                 TST     BIV                ;DID AN INTERRUPT OCCUR?
5168 033152  001021                         BNE     100$               ;YES, AT ANOTHER ADDRESS
5169 033154  162701  000040                 SUB     #40,R1             ;DECREMENT PRIORITY
5170 033160                                 SETPRI  R1                 ;SET PRIORITY
5171 033164  020127  000100                 CMP     R1,#PRI02          ;IS PRIORITY OVER 2 ?
5172 033170  004737  015240                 CALL    INSERT             ;TEST IF ERROR MESSAGE IS REQUIRED
5173 033174  001356                         BNE     110$               ;BRANCH IF YES
5174 033176                                 ERRHRD  506.,ERR504        ;ERROR HANDLER
5175 033206                                 CKLOOP
5176 033210  004737  015240                 CALL    INSERT             ;TEST IF ERROR MESSAGE IS REQUIRED
5177 033214  000444                         BR      120$               ;JOIN COMMON CODE
5178 033216                         100$:   ERRHRD  507.,ERR505        ;ERROR HANDLER
5179 033226                                 CKLOOP                     ;
5180 033230  062701  000040         90$:    ADD     #40,R1             ;CORRECT PRI FOR CMP
5181 033234  020137  002430                 CMP     R1,PRIO            ;PRIORITY CORRECT  ?
5182 033240  004737  015240                 CALL    INSERT             ;TEST IF ERROR MESSAGE IS REQUIRED
5183 033244  001430                         BEQ     120$               ;BRANCH IF YES
5184         000005                         .REPT   5                  ;SET UP DATA FOR ERROR MESSAGES
5185                                        ASR     R1                 ;...
5186                                        .ENDR                      ;...
5187 033260  010137  002476                 MOV     R1,BAD             ;SET UP DATA FOR ERROR REPORT
5188 033264  013737  002430  002474         MOV     PRIO,GOOD          ;...
5189         000005                         .REPT   5                  ;
5190                                        ASR     GOOD               ;
5191                                        .ENDR                      ;
5192 033316                                 ERRHRD  508.,ERR506        ;ERROR HANDLER
5193
5194 033326                         120$:   ENDSEG
5195
5196 033330  017737  147070  002476         MOV     @INR,BAD           ;GET INR REGISTER CONTENS
5197 033336  012737  100000  002474         MOV     #100000,GOOD       ;SET UP FOR ERROR HANDLER
5198 033344  033737  002474  002476         BIT     GOOD,BAD           ;IS MASTER ENABLE CLEARED ?
5199 033352  004737  015240                 CALL    INSERT             ;TEST IF ERROR MESSAGE IS REQUIRED
5200 033356  001404                         BEQ     122$               ;BRANCH IF YES
5201 033360                                 ERRHRD  509.,ERR507        ;ERROR HANDLER
5202
5203 033370  006337  002434         122$:   ASL     BITMSK             ;SHIFT 'EN' MASK BIT
5204 033374  005237  002442                 INC     NUMBER             ;INCREMENT CHANNEL BIT NUMBER
5205 033400  023727  002442  000005         CMP     NUMBER,#5          ;ALL CHANNELS DONE ?
```

```
 5206 033406  003002                          BGT     130$                    ;IF YES TERMINATE TEST
 5207 033410  000137  033034                  JMP     200$                    ;NO
 5208
 5209 033414  005737  002164          130$:   TST     QVP                     ;IS QUICK VERIFY PASS SELECTED
 5210 033420  001010                          BNE     EXS5                    ;YES
 5211 033422  005237  002504                  INC     ITRCNT                  ;BUMP ITERATION COUNT
 5212 033426  023737  002506  002504          CMP     ITRDEF,ITRCNT           ;DEFAULT ITERATION COUNT
 5213 033434  001402                          BEQ     EXS5                    ;TIME TO EXIT
 5214 033436  000137  032336                  JMP     ITRAC5                  ;DO THE TEST AGAIN
 5215
 5216 033442  005077  146756          EXS5:   CLR     @INR                    ;DISABLE INTERRUPTS
 5217 033446                                  SETPRI  SAVPRI                  ; RESTORE THE PRIORITY
 5218 033454                                  CLRVEC  #4                      ;RESET TRAP VECTOR
 5219 033462                                  CLRVEC  VEC                     ;DEVICE INTERRUPT ADDRESS
 5220 033470                                  EXIT    TST
 5221
 5222                                         .NLIST  BEX
 5223 033474     045     123     062  TSMD5::.ASCIZ   /%S2%AINTERRUPT TEST %N/
 5224                                         .LIST   BEX
 5225                                         .EVEN
 5226
 5227 033524                                  ENDTST
```

```
5229
5230                                    .SBTTL  TEST 6  --  REFERENCE FREQUENCY TEST
5231                                    ;*************************************************************************
5232                                    ;       TEST 6  --  REFERENCE FREQUENCY TEST
5233                                    ;
5234                                    ;       FOR TEST DESCRIPTION SEE 6.0 TEST SUMMERIES
5235                                    ;*************************************************************************
5236
5237                                            .DSABLE LSB
5238 033526                                     BGNTST
5239 033526  004737  015006                     CALL    SELECT                  ;CALL SELECT ROUTINE
5240 033532  000420                             .WORD   420                     ;TEST SELECT MASK
5241 033534  034076                             TSHD6                           ;TEST HEADER ADDRESS
5242 033536  103002                             BCC     1$                      ;IF CARRY IS SET, DON'T BRANCH
5243 033540                                     EXIT    TST                     ;EXIT TEST IF CARRY IS SET
5244 033544  005037  002504             1$:     CLR     ITRCNT                  ;CLEAR ITERATION COUNTER
5245
5246 033550  013701  002416                     MOV     MOD,R1                  ;GET FIRST REGISTER ADDRESS
5247 033554  004737  021470                     JSR     PC,AMREST               ;RESET THE MODULE
5248 033560                             ITRAC6:
5249 033560  012737  000001  002442             MOV     #1,NUMBER               ;CHANNEL NUMBER
5250
5251                                    ;SET UP ALL COUNTERS FOR 5 MHZ AND LOAD VALUE 10000
5252
5253 033566  012700  177400             10$:    MOV     #B,R0                   ;SELECT MODE REGISTER
5254 033572  063700  002442                     ADD     NUMBER,R0               ;ADD IN CHANNEL NR
5255 033576  010061  000002                     MOV     R0,MREA(R1)             ;SELECT MODE REGISTER
5256 033602  012761  005440  000004             MOV     #005440,MREB(R1)        ;SELECT 5MHZ
5257 033610  012700  177410                     MOV     #B!10,R0                ;SELECT LOAD REGISTER
5258 033614  063700  002442                     ADD     NUMBER,R0               ;SELECT CHANNEL
5259 033620  010061  000002                     MOV     R0,MREA(R1)             ;ADDRESS
5260 033624  012761  023420  000004             MOV     #10000.,MREB(R1)        ;LOAD VALUE
5261 033632  005237  002442                     INC     NUMBER                  ;BUMP CHANNEL NUMBER
5262 033636  023727  002442  000005             CMP     NUMBER,#5               ;ALL CHANNELS?
5263 033644  003750                             BLE     10$                     ;MORE TO DO
5264                                    ;
5265                                    ; LOAD AND ARM ALL COUNTERS
5266                                    ;
5267 033646  012761  177537  000002             MOV     #C$LOA!37,MREA(R1)      ;START ALL COUNTER
5268 033654  012761  177477  000002             MOV     #C$ARM!37,MREA(R1)      ;START ALL COUNTER
5269 033662  000240                             NOP
5270 033664  000240                             NOP                             ;LET COUNTERS RUN
5271 033666  000240                             NOP
5272 033670  012761  177637  000002             MOV     #C$DAS!37,MREA(R1)      ;SAVE COUNTERS
5273                                    ;
5274                                    ; READ OUT COUNTERS AND SAVE IN CNTVAL TABLE
5275                                    ;
5276 033676  012704  002452                     MOV     #CNTVAL,R4              ;SAVE VALUES HERE
5277 033702  012737  000001  002442             MOV     #1,NUMBER               ;  A. NUMBER
5278 033710  012700  177420             20$:    MOV     #B!20,R0                ;  T HOLD REGISTER
5279 033714  063700  002442                     ADD     NUMBER,R0               ;ANNEL NUMBER
5280 033720  010061  000002                     MOV     R0,MREA(R1)             ;ELECT COUNTER
5281 033724  016124  000004                     MOV     MREB(R1),(R4)+          ;GET COUNTER VALUE
5282 033730  005237  002442                     INC     NUMBER                  ;BUMP CHANNEL NUMBER
5283 033734  023727  002442  000005             CMP     NUMBER,#5               ;ALL CHANNELS?
5284 033742  003762                             BLE     20$                     ;MORE TO DO
5285                                    ;
```

```
                                        ; COMPUTE MIN AND MAX VALUES OF ALL COUNTERS
                                        ;
033744  012702  002452                        MOV     #CNTVAL,R2              ;
033750  012703  000005                        MOV     #5,R3                   ;LOAD LOOP COUNT
033754  012700  023420                        MOV     #10000.,R0              ;MIN
033760  005004                                CLR     R4                      ;MAX
033762  020012                        30$:    CMP     R0,(R2)                 ;LOAD NEW MINIMUM?
033764  003401                                BLE     40$                     ;NO
033766  011200                                MOV     (R2),R0                 ;LOAD NEW MINIMUM
033770  020412                        40$:    CMP     R4,(R2)                 ;NEW MAXIMUM?
033772  003001                                BGT     50$                     ;NO
033774  011204                                MOV     (R2),R4                 ;NEW MAXIMUM
033776  005722                        50$:    TST     (R2)+                   ;BUMP POINTER
034000  005303                                DEC     R3                      ;
034002  001367                                BNE     30$                     ;LOOP TO FIND MAX & MIN
                                        ;
                                        ; SEE IF THE COUNTER VALUES WERE IN EXPECTED RANGE
                                        ;
034004  020427  023420                        CMP     R4,#10000.              ;DID COUNTER MOVE?
034010  002015                                BGE     70$                     ;STRANGE VALUES
034012  005700                                TST     R0                      ;ROUGH TEST
034014  003413                                BLE     70$                     ;
034016  010402                                MOV     R4,R2                   ;GET DIFFERENCE (MAX-MIN)
034020  160002                                SUB     R0,R2
034022  020227  000002                        CMP     R2,#2                   ;ALLOWABLE DIFFERENCE
034026  004737  015240                        CALL    INSERT                  ;SKIP BRANCH IF "SFI" IS SET
034032  003404                                BLE     70$                     ;BRANCH IF YES (LESS THEN 3)
034034                                        ERRHRD  601,,ERR130
034044  005737  002164                70$:    TST     QVP                     ;IS QUICK VERIFY PASS SELECTED
034050  001010                                BNE     EXS6                    ;YES
034052  005237  002504                        INC     ITRCNT                  ;BUMP ITERATION COUNT
034056  023737  002506  002504                CMP     ITRDEF,ITRCNT           ;DEFAULT ITERATION COUNT
034064  001402                                BEQ     EXS6                    ;TIME TO EXIT
034066  000137  033560                        JMP     ITRAC6                  ;DO THE TEST AGAIN

034072                                  EXS6:   EXIT    TST

                                                .NLIST  BEX
034076      045     123     062         TSHD6::.ASCIZ   /%S2%AREFERENCE FREQUENCY TEST %N/
                                                .LIST   BEX
                                                .EVEN

034140                                          ENDTST
```

```
                                        .SBTTL TEST 7  --  SIGNAL GENERATION ON COUNTER INPUT
                                        ;*****************************************************************
                                        ;       TEST 7  --  SIGNAL GENERATION ON COUNTER INPUT
                                        ;
                                        ;       FOR TEST DESCRIPTION SEE 6.0 TEST SUMMERIES
                                        ;*****************************************************************


                                                .DSABLE LSB
034142                                          BGNTST
034142  004737  015006                          CALL    SELECT                  ;CALL SELECT ROUTINE
034146  000420                                  .WORD   420                     ;TEST SELECT MASK
034150  034576                                  TSHD7                           ;TEST HEADER ADDRESS
034152  103002                                  BCC     1$                      ;IF CARRY IS SET, DON'T BRANCH
034154                                          EXIT    TST                     ;EXIT TEST IF CARRY IS SET
034160  005037  002504                  1$:     CLR     ITRCNT                  ;CLEAR ITERATION COUNTER

034164  013701  002416                          MOV     MOD,R1                  ;GET FIRST REGISTER ADDRESS
034170  004737  021470                          JSR     PC,AMREST               ;RESET THE MODULE
034174                                  ITRAC7:
034174  012737  000001  002442                  MOV     #1,NUMBER               ;CHANNEL NUMBER
034202  012737  000001  002434                  MOV     #1,BITMSK
034210  012700  177400                  10$:    MOV     #8,R0                   ;SELECT MODE REGISTER
034214  063700  002442                          ADD     NUMBER,R0               ;ADD IN CHANNEL NR
034220  010061  000002                          MOV     R0,MREA(R1)             ;SELECT MODE REGISTER
034224  012761  000050  000004                  MOV     #000050,MREB(R1)        ;SELECT INTERNAL LOOP BACK
034232  012700  177410                          MOV     #8!10,R0                ;SELECT LOAD REGISTER
034236  063700  002442                          ADD     NUMBER,R0               ;SELECT CHANNEL
034242  010061  000002                          MOV     R0,MREA(R1)             ;ADDRESS
034246  005061  000004                          CLR     MREB(R1)                ;INIT COUNTER
034252  012700  177500                          MOV     #C$LOA,R0               ;LOAD
034256  063700  002434                          ADD     BITMSK,R0               ;ADD IN CHANNEL NR
034262  010061  000002                          MOV     R0,MREA(R1)             ;STARTS COUNTER

034266                                          BGNSEG                          ;BEGINNING OF LOOP ON ERROR SEGMENT
034270  013737  002434  002436                  MOV     BITMSK,BITMS1           ;BUILD SECOND BIT MASK
034276  006237  002436                          ASR     BITMS1
034302  013702  002442                          MOV     NUMBER,R2               ;COMPUTE TCN-1
034306  005302                                  DEC     R2
034310  001005                                  BNE     20$
034312  012702  000005                          MOV     #5,R2                   ;WRAP AROUND
034316  012737  177420  002436                  MOV     #8!20,BITMS1            ;BIT MASK
034324  012700  177400          20$:    MOV     #8,R0                   ;SELECT MODE REGISTER
034330  060200                                  ADD     R2,R0                   ;ADD IN CHANNEL NR
034332  010061  000002                          MOV     R0,MREA(R1)             ;SELECT MODE REGISTER
034336  012761  005440  000004                  MOV     #5440,MREB(R1)          ;SELECT 5MHZ FREQUENCY
034344  012700  177410                          MOV     #8!10,R0                ;LOAD AND ARM TAK COUNTER
034350  060200                                  ADD     R2,R0                   ;CHANNEL NUMBER
034352  010061  000002                          MOV     R0,MREA(R1)             ;ADDRESS
034356  012761  000003  000004                  MOV     #3,MREB(R1)             ;SETUP COUNTER
034364  012700  177500                          MOV     #C$LOA,R0
034370  063700  002436                          ADD     BITMS1,R0
034374  010061  000002                          MOV     R0,MREA(R1)             ;START COUNTER
034400  012700  177440                          MOV     #C$ARM,R0
034404  063700  002436                          ADD     BITMS1,R0
034410  063700  002434                          ADD     BITMSK,R0               ;ADD IN CHANNEL NR
034414  010061  000002                          MOV     R0,MREA(R1)             ;START COUNTER
```

```
5386 034420  000240                          NOP
5389 034422  000240                          NOP
5390 034424  000240                          NOP
5391 034426  000240                          NOP                     ;LET COUNTER GO
5392 034430  000240                          NOP
5393 034432  012700  177700                  MOV     #C$DAC,R0       ;DISARM COUNTER
5394 034436  063700  002436                  ADD     BITMS1,R0       ;CHANNEL NUMBER
5395 034442  010061  000002                  MOV     R0,MREA(R1)
5396 034446  012700  177600                  MOV     #C$DAS,R0       ;DISARM COUNTER
5397 034452  063700  002434                  ADD     BITMSK,R0       ;CHANNEL NUMBER
5398 034456  010061  000002                  MOV     R0,MREA(R1)
5399
5400 034462  012700  177420                  MOV     #8!20,R0        ;HOLD REGISTER
5401 034466  063700  002442                  ADD     NUMBER,R0
5402 034472  010061  000002                  MOV     R0,MREA(R1)     ;COUNTER -> HOLD REGISTER
5403 034476  016103  000004                  MOV     MREB(R1),R3
5404 034502  004737  015240                  CALL    INSERT          ;TEST IF ERROR MESSAGE IS REQUIRED
5405 034506  001005                         BNE     11$             ;DID CLOCK TICK
5406 034510                                  ERRHRD  701,,ERR115     ;ERROR HANDLER
5407 034520                                  CKLOOP                  ;IF LOOP ON ERROR
5408 034522                          11$:    ENDSEG                  ;END SEGMENT
5409 034524  006337  002434                  ASL     BITMSK          ;SHIFT TO NEXT BIT
5410 034530  005237  002442                  INC     NUMBER          ;BUMP CHANNEL NUMBER
5411 034534  023727  002442  000005         CMP     NUMBER,#5       ;ALL CHANNELS?
5412 034542  003622                         BLE     10$             ;MORE TO DO
5413
5414 034544  005737  002164                  TST     QVP             ;IS QUICK VERIFY PASS SELECTED
5415 034550  001010                         BNE     EXS7            ;YES
5416 034552  005237  002504                  INC     ITRCNT          ;BUMP ITERATION COUNT
5417 034556  023737  002506  002504         CMP     ITRDEF,ITRCNT   ;DEFAULT ITERATION COUNT
5418 034564  001402                         BEQ     EXS7            ;TIME TO EXIT
5419 034566  000137  034174                  JMP     ITRAC7          ;DO THE TEST AGAIN
5420
5421 034572                          EXS7:   EXIT    TST
5422
5423                                         .NLIST  BEX
5424 034576    045    123    062   TSHD7::.ASCIZ   /%S2%ASIGNAL GENERATION ON COUNTER INPUT %N/
5425                                         .LIST   BEX
5426                                         .EVEN
5427
5428 034652                                  ENDTST
5429
```

```
5431                                            .SBTTL TEST 8  --  EXTERNAL LOOPBACK, NOT ISOLATED
5432                                            ;*****************************************************************
5433                                            ;       TEST 8  --  EXTERNAL LOOPBACK, NOT ISOLATED
5434                                            ;
5435                                            ;       FOR TEST DESCRIPTION SEE 6.0 TEST SERIES
5436                                            ;*****************************************************************
5437
5438                                                    .DSABLE LSB
5439 034654                                            BGNTST
5440 034654  004737  015006                            CALL    SELECT                  ;CALL SELECT ROUTINE
5441 034660  000420                                    .WORD   420                     ;TEST SELECT MASK
5442 034662  035270                                    TSHD8                           ;TEST HEADER ADDRESS
5443 034664  103002                                    BCC     1$                      ;IF CARRY IS SET, DON'T BRANCH
5444 034666                                            EXIT    TST                     ;EXIT TEST IF CARRY IS SET
5445 034672  005037  002504                    1$:     CLR     ITRCNT                  ;CLEAR ITERATION COUNTER
5446
5447 034676  013701  002416                            MOV     MOD,R1                  ;GET FIRST REGISTER ADDRESS
5448 034702  004737  021470                            JSR     PC,AMRST                ;RESET THE MODULE
5449 034706  012761  177427  000002                    MOV     #8!27,MREA(R1)          ;SELECT FREQ OUTPUT CNTRL REG
5450 034714  012761  061000  000004                    MOV     #61000,MREB(R1)         ;SELECT 5MH LOOPBACK FREQUENCY
5451 034722                                    ITRAC8:
5452 034722  012737  000001  002442                    MOV     #1,NUMBER               ;CHANNEL NUMBER
5453 034730  012737  000001  002434                    MOV     #1,BITMSK
5454 034736  012737  000400  002464            10$:    MOV     #400,SRC                ;INIT SOURCE
5455 034744  012737  000012  002466                    MOV     #10.,LOOP               ;SETUP LOOP COUNT
5456 034752                                    20$:    BGNSEG                          ;BEGINNING OF LOOP ON ERROR SEGMENT
5457 034754  012700  177400                            MOV     #8,R0                   ;SELECT MODE REGISTER
5458 034760  063700  002442                            ADD     NUMBER,R0               ;ADD IN CHANNEL NR
5459 034764  010061  000002                            MOV     R0,MREA(R1)             ;SELECT MODE REGISTER
5460 034770  012700  000010                            MOV     #10,R0                  ;COUNT UP
5461 034774  063700  002464                            ADD     SRC,R0                  ;SOURCE OR GATE
5462 035000  010061  000004                            MOV     R0,MREB(R1)             ;SELECT SOURCE
5463 035004  012700  177410                            MOV     #8!10,R0                ;SELECT LOAD REGISTER
5464 035010  063700  002442                            ADD     NUMBER,R0               ;SELECT CHANNEL
5465 035014  010061  000002                            MOV     R0,MREA(R1)             ;ADDRESS
5466 035020  005061  000004                            CLR     MREB(R1)                ;INIT COUNTER
5467 035024  012700  177500                            MOV     #C$LOA,R0               ;LOAD
5468 035030  063700  002434                            ADD     BITMSK,R0               ;ADD IN CHANNEL NR
5469 035034  010061  000002                            MOV     R0,MREA(R1)             ;STARTS COUNTER
5470 035040  012700  177440                            MOV     #C$ARM,R0               ;ARM
5471 035044  063700  002434                            ADD     BITMSK,R0               ;ADD IN CHANNEL NR
5472 035050  010061  000002                            MOV     R0,MREA(R1)             ;STARTS COUNTER
5473
5474 035054  000240                                    NOP
5475 035056  000240                                    NOP                             ;LET COUNTER GO
5476 035060  000240                                    NOP
5477
5478 035062  012700  177600                            MOV     #C$DAS,R0               ;DISARM COUNTER
5479 035066  063700  002434                            ADD     BITMSK,R0               ;CHANNEL NUMBER
5480 035072  010061  000002                            MOV     R0,MREA(R1)
5481
5482 035076  012700  177420                            MOV     #8!20,R0                ;HOLD REGISTER
5483 035102  063700  002442                            ADD     NUMBER,R0
5484 035106  010061  000002                            MOV     R0,MREA(R1)             ;COUNTER -> HOLD REGISTER
5485 035112  016103  000004                            MOV     MREB(R1),R3
5486 035116  004737  015240                            CALL    INSERT                  ;TEST IF ERROR MESSAGE IS REQUIRED
5487 035122  001026                                    BNE     50$                     ;DID CLOCK TICK
```

```
5488 035124  013702  002464                 MOV     SRC,R2                  ;SOURCE OR GATE
5489 035130  000302                         SWAB    R2                      ;GET TO LOW ORDER
5490 035132  042702  177740                 BIC     #177740,R2              ;CLEAR OTHER BITS
5491 035136  020227  000005                 CMP     R2,#5                   ;SOURCE
5492 035142  004737  015240                 CALL    INSERT                  ;TEST IF ERROR MESSAGE IS REQUIRED
5493 035146  003005                         BGT     30$                     ;NO, GATE
5494 035150                                 ERRMRD  801.,ER116              ;ERROR HANDLER
5495 035160  000402                         BR      40$
5496 035162  162702  000005          30$:   SUB     #5,R2
5497 035166                         40$:    ERRMRD  802.,ER116A
5498 035176                                 CKLOOP                          ;IF LOOP ON ERROR
5499 035200                         50$:    ENDSEG                          ;END SEGMENT
5500
5501 035202  062737  000400  002464         ADD     #400,SRC                ;BUMP TO NEXT SOURCE
5502 035210  005337  002466                 DEC     LOOP                    ;ADJUST LOOP COUNT
5503 035214  001256                         BNE     20$                     ;LOOP FOR NEXT SOURCE
5504
5505 035216  006337  002434                 ASL     BITMSK                  ;SHIFT TO NEXT BIT
5506 035222  005237  002442                 INC     NUMBER                  ;BUMP CHANNEL NUMBER
5507 035226  023727  002442  000005         CMP     NUMBER,#5               ;ALL CHANNELS?
5508 035234  003640                         BLE     10$                     ;MORE TO DO
5509
5510 035236  005737  002164                 TST     QVP                     ;IS QUICK VERIFY PASS SELECTED
5511 035242  001010                         BNE     EXS8                    ;YES
5512 035244  005237  002504                 INC     ITRCNT                  ;BUMP ITERATION COUNT
5513 035250  023737  002506  002504         CMP     ITRDEF,ITRCNT           ;DEFAULT ITERATION COUNT
5514 035256  001402                         BEQ     EXS8                    ;TIME TO EXIT
5515 035260  000137  034722                 JMP     ITRAC8                  ;DO THE TEST AGAIN
5516
5517 035264                         EXS8:   EXIT    TST
5518
5519                                        .NLIST  BEX
5520 035270     045     123     062  TSHD8::.ASCIZ   /%S2%AEXTERNAL LOOPBACK TEST, NOT ISOLATED%N/
5521                                        .LIST   BEX
5522                                        .EVEN
5523
5524 035344                                 ENDTST
5525
```

```
5527                                    .SBTTL TEST 9  --  EXTERNAL LOOPBACK, ISOLATED
5528                                    ;*****************************************************************
5529                                    ;       TEST 9  --  EXTERNAL LOOPBACK, ISOLATED
5530                                    ;
5531                                    ;       FOR TEST DESCRIPTION SEE 6.0 TEST SUMMERIES
5532                                    ;*****************************************************************
5533
5534                                            .DSABLE LSB
5535 035346                                    BGNTST
5536 035346  004737  015006                    CALL    SELECT                  ;CALL SELECT ROUTINE
5537 035352  000420                            .WORD   420                     ;TEST SELECT MASK
5538 035354  036110                            TSHD9                           ;TEST HEADER ADDRESS
5539 035356  103002                            BCC     1$                      ;IF CARRY IS SET, DON'T BRANCH
5540 035360                                    EXIT    TST                     ;EXIT TEST IF CARRY IS SET
5541 035364  005037  002504            1$:     CLR     ITRCNT                  ;CLEAR ITERATION COUNTER
5542
5543 035370  013701  002416                    MOV     MOD,R1                  ;GET FIRST REGISTER ADDRESS
5544 035374  004737  021470                    JSR     PC,AMREST               ;RESET THE MODULE
5545 035400  012761  177427  000002            MOV     #B!27,MREA(R1)          ;SELECT FREQ OUTPUT CNTRL REG
5546 035406  012761  070000  000004            MOV     #70000,MREB(R1)         ;SELECT 5MH LOOPBACK FREQ. FOUT=OFF
5547 035414                            ITRAC9:
5548 035414  012737  000001  002442            MOV     #1,NUMBER               ;CHANNEL NUMBER (FOR INC)
5549 035422  012737  000001  002434            MOV     #1,BITMSK               ;CHANNEL NUMBER (FOR SHIFT)
5550 035430  012737  000400  002464            MOV     #400,SRC                ;INIT SOURCE
5551
5552                                    ;SETUP FIRST COUTER FOR 100 KHZ GENERATION ON COUNTER OUTPUT.
5553
5554 035436  012700  177400            10$:    MOV     #B,R0                   ;GET MODE REGISTER MASK
5555 035442  063700  002442                    ADD     NUMBER,R0               ;ADD IN CHANNEL NR
5556 035446  010061  000002                    MOV     R0,MREA(R1)             ;SELECT MODE REGISTER
5557 035452  012761  005442  000004            MOV     #5442,MREB(R1)          ;LOAD TOGGLE AT 5 MHZ ,
5558 035460  012700  177410                    MOV     #B!10,R0                ;GET LOAD REGISTER MASK
5559 035464  063700  002442                    ADD     NUMBER,R0               ;ADD IN CHANNEL NR
5560 035470  010061  000002                    MOV     R0,MREA(R1)             ;SELECT LOAD REGISTER
5561 035474  012761  000031  000004            MOV     #25.,MREB(R1)           ;LOAD COUNT VALUE
5562                                    ;
5563                                    ; SELECT SECOND COUNTER
5564                                    ;
5565 035502  013737  002434  002436            MOV     BITMSK,BITMS1           ;NEXT COUNTER
5566 035510  006337  002436                    ASL     BITMS1                  ;NEXT COUNTER
5567 035514  013702  002442                    MOV     NUMBER,R2               ;USE N+1
5568 035520  005202                            INC     R2                      ;...
5569 035522  020227  000005                    CMP     R2,#5                   ;TEST FOR WRAP AROUND (ALL CNT DONE?)
5570 035526  003405                            BLE     20$                     ;BRANCH IF NOT
5571 035530  012702  000001                    MOV     #1,R2                   ;START OVER WITH FIRST COUNTER
5572 035534  012737  000001  002436            MOV     #1,BITMS1               ;...
5573
5574 035542  012704  000002            20$:    MOV     #2,R4                   ;DO TWICE
5575 035546                            30$:    BGNSEG                          ;BEGINNING OF LOOP ON ERROR SEGMENT
5576 035550  012700  177400                    MOV     #B,R0                   ;GET MODE REGISTER MASK
5577 035554  060200                            ADD     R2,R0                   ;ADD IN CHANNEL NR
5578 035556  010061  000002                    MOV     R0,MREA(R1)             ;SELECT MODE REGISTER
5579 035562  012700  000050                    MOV     #50,R0                  ;GET COUNT UP MASK
5580 035566  063700  002464                    ADD     SRC,R0                  ;SOURCE OR GATE
5581 035572  010061  000004                    MOV     R0,MREB(R1)             ;SELECT COUNT UP
5582 035576  012700  177410                    MOV     #B!10,R0                ;GET LOAD REGISTER MASK
5583 035602  060200                            ADD     R2,R0                   ;ADD IN CHA. NUMBER
```

```
5584 035604  010061  000002                MOV     R0,MREA(R1)             ;AND LOAD IT
5585 035610  005061  000004                CLR     MPEB(R1)                ;INIT COUNTER
5586 035614  012700  177500                MOV     #C$LOA,R0               ;GET LOAD COMMAND
5587 035620  063700  002436                ADD     BITMS1,R0               ;ADD IN CHANNEL NR
5588 035624  010061  000002                MOV     R0,MREA(R1)             ;READY TO START SECOND COUNTER
5589
5590 035630  012700  177500                MOV     #C$LOA,R0               ;GET LOAD MASK
5591 035634  063700  002434                ADD     BITMSK,R0               ;ADD IN CHANNEL NR
5592 035640  010061  000002                MOV     R0,MREA(R1)             ;READY TO START FIRST COUNTER
5593
5594                                ;ENABLE BOTH COUNTER FOR COUNTING
5595
5596 035644  012700  177440                MOV     #C$ARM,R0               ;GET ARM COMMAND
5597 035650  063700  002434                ADD     BITMSK,R0               ;ADD IN COUNTER NR
5598 035654  063700  002436                ADD     BITMS1,R0               ;ADD IN SECOND COUNTER NR
5599 035660  010061  000002                MOV     R0,MREA(R1)             ;START BOTH COUNTER (LOAD ARM)
5600
5601 035664  012700  000144                MOV     #100.,R0                ;LET COUNTER RUN AT 100 KZ
5602 035670  000240          40$:          NOP
5603 035672  000240                        NOP                             ;LET COUNTER GO
5604 035674  000240                        NOP
5605 035676  005300                        DEC     R0                      ;SHORT LOOP COUNT
5606 035700  001373                        BNE     40$
5607
5608                                ;STOP COUNTERS
5609
5610 035702  012700  177600                MOV     #C$DAS,R0               ;GET DISARM COMMAND
5611 035706  063700  002434                ADD     BITMSK,R0               ;ADD IN COUNTER NR
5612 035712  063700  002436                ADD     BITMS1,R0               ;ADD IN SECOND COUNTER NUMBER
5613 035716  010061  000002                MOV     R0,MREA(R1)             ;STOP BOTH COUNTER (DISARM)
5614
5615                                ;CHECK SECOND COUNTER CONTENTS
5616
5617 035722  012700  177420                MOV     #B!20,R0                ;HOLD REGISTER
5618 035726  060200                        ADD     R2,R0                   ;ADD CNT NUMBER
5619 035730  010061  000002                MOV     R0,MREA(R1)             ;COUNTER -> HOLD REGISTER
5620 035734  016103  000004                MOV     MREB(R1),R3             ;GET COUNTER CONTENTS
5621 035740  004737  015240                CALL    INSERT                  ;TEST IF ERROR MESSAGE IS REQUIRED
5622 035744  001024                        BNE     60$                     ;DID CLOCK TICK
5623 035746  013703  002464                MOV     SRC,R3                  ;SOURCE OR GATE
5624 035752  000303                        SWAB    R3                      ;GET TO LOW ORDER
5625 035754  042703  177740                BIC     #177740,R3              ;CLEAR OTHER BITS
5626 035760  020327  000005                CMP     R3,#5                   ;SOURCE
5627 035764  003006                        BGT     50$                     ;NO, GATE
5628 035766                                ERRHRD  901,,ER116B             ;ERROR HANDLER FOR SORCE
5629 035776                                CKLOOP                          ;IF LOOP ON ERROR
5630 036000  000406                        BR      60$                     ;
5631 036002  162703  000005        50$:    SUB     #5,R3                   ;
5632 036006                                ERRHRD  902,,ER116C             ;ERROR HANDLER FOR GATE
5633 036016                        60$:    ENDSEG                          ;END SEGMENT
5634
5635 036020  062737  000400  002464        ADD     #400,SRC                ;BUMP TO NEXT SOURCE
5636 036026  005304                        DEC     R4                      ;ADDJUST LOOP COUNT
5637 036030  001246                        BNE     30$                     ;LOOP FOR NEXT SOURCE
5638
5639 036032  006337  002434                ASL     BITMSK                  ;SHIFT TO NEXT COUNTER
5640 036036  005237  002442                INC     NUMBER                  ;BUMP CHANNEL COUNTER
```

```
 5641 036042  023727  002442  000005          CMP     NUMBER,#5               ;ALL COUNTERS DONE ?
 5642 036050  003002                          BGT     70$                     ;EXIT IF YES
 5643 036052  000137  035436                  JMP     10$                     ;MORE TO DO
 5644
 5645 036056  005737  002164          70$:    TST     QVP                     ;IS QUICK VERIFY PASS SELECTED
 5646 036062  001010                          BNE     EXS9                    ;YES
 5647 036064  005237  002504                  INC     ITRCNT                  ;BUMP ITERATION COUNT
 5648 036070  023737  002506  002504          CMP     ITRDEF,ITRCNT           ;DEFAULT ITERATION COUNT
 5649 036076  001402                          BEQ     EXS9                    ;TIME TO EXIT
 5650 036100  000137  035414                  JMP     ITRAC9                  ;DO THE TEST AGAIN
 5651
 5652 036104                          EXS9:   EXIT    TST
 5653
 5654                                         .NLIST  BEX
 5655 036110     045     123     062  TSHD9::.ASCIZ   /#S2#AEXTERNAL LOOPBACK TEST,ISOLATED.#N/
 5656                                         .LIST   BEX
 5657                                         .EVEN
 5658
 5659 036160                                 ENDTST
```

```
                                        .SBTTL TEST 10  --  UP/DOWN COUNTING APPLICATION TEST
                                        ;*****************************************************************
                                        ;         TEST 10  -   UP/DOWN COUNTING APPLICATION TEST
                                        ;
                                        ;       FOR TEST DESCRIPTION SEE 6.0 TEST SUMMERIES
                                        ;*****************************************************************
036162                                          BGNTST
036162  004737  015006                          CALL    SELECT                  ;CALL SELECT ROUTINE
036166  000420                                  .WORD   420                     ;TEST SELECT MASK
036170  037122                                  TSHD10                          ;TEST HEADER ADDRESS
036172  103002                                  BCC     1$                      ;IF CARRY IS SET, DON'T BRANCH
036174                                          EXIT    TST                     ;EXIT TEST IF CARRY IS SET
036200  012737  000004  002504          1$:     MOV     #4,ITRCNT               ;ITERATION COUNTER
036206  005037  000000G                         CLR     WORFLA                  ;CLEAR WORKING FLAG
036212  013701  002416                          MOV     MOD,R1                  ;GET FIRST REGISTER ADDRESS
036216  004737  021470                          JSR     PC,AMREST               ;RESET THE MODULE
036222                                  ITR10:
036222  012761  177427  000002                  MOV     #B!27,MREA(R1)          ;SELECT FREQ OUTPUT CNTRL REG
036230  012761  070000  000004                  MOV     #70000,MREB(R1)         ;FOUT OFF
036236  012761  177403  000002                  MOV     #B!3,MREA(R1)           ;SELECT MODE REGISTER
036244  012761  000042  000004                  MOV     #42,MREB(R1)            ;OUT TOGGLE
036252  012761  177404  000002                  MOV     #B!4,MREA(R1)           ;SELECT MODE REGISTER
036260  012761  000042  000004                  MOV     #42,MREB(R1)            ;OUT TOGGLE
036266  012761  177401  000002                  MOV     #B!1,MREA(R1)           ;SELECT MODE REG
036274  012761  042450  000004                  MOV     #42450,MREB(R1)         ;ACTIVE HIGH LEVEL, GATE 2, SRC 5
036302  004537  037022                          JSR     R5,SETMRA               ;SET & TEST MREA
036306  000344                                  .WORD   344                     ;TOUT N = 4
036310  004537  037022                          JSR     R5,SETMRA               ;SET & TEST MREA
036314  000343                                  .WORD   343                     ;TOUT N = 3
036316  012761  177411  000002                  MOV     #B!11,MREA(R1)          ;SELECT 1 LOAD REGISTER
036324  005061  000004                          CLR     MREB(R1)                ;INIT COUNTER
036330  012761  177402  000002                  MOV     #B!2,MREA(R1)           ;SELECT MODE REG
036336  012761  122450  000004                  MOV     #122450,MREB(R1)        ;ACTIVE LOW, GATE 2, SRC 5
036344  012761  177412  000002                  MOV     #B!12,MREA(R1)          ;SELECT 2 LOAD REGISTER
036352  005061  000004                          CLR     MREB(R1)                ;INIT COUNTER
036356  012703  023420                          MOV     #10000.,R3              ;LOOP COUNT
036362  012761  177503  000002                  MOV     #C!LOA!3,MREA(R1)       ;STARTS COUNTER
036370  012761  177443  000002                  MOV     #C!ARM!3,MREA(R1)       ;STARTS COUNTER
036376  004537  037022          10$:            JSR     R5,SETMRA               ;SET & TEST MREA
036402  000354                                  .WORD   354                     ;SET TOUT N = 4
036404  004537  037022                          JSR     R5,SETMRA               ;SET & TEST MREA
036410  000353                                  .WORD   353                     ;SET TOUT N = 3
036412  004537  037022                          JSR     R5,SETMRA               ;SET & TEST MREA
036416  000344                                  .WORD   344                     ;CLEAR TOUT N = 4
036420  004537  037022                          JSR     R5,SETMRA               ;SET & TEST MREA
036424  000343                                  .WORD   343                     ;CLEAR TOUT N = 3
036426                                          BREAK
036430  077316                                  SOB     R3,10$                  ;LOOP
                                        
036432  012761  177603  000002                  MOV     #C!DAS!3,MREA(R1)       ;DISARM AND SAVE COUNTERS
036440  012761  177421  000002                  MOV     #B!21,MREA(R1)          ;COUNTER -> HOLD REGISTER #1
036446  016105  000004                          MOV     MREB(R1),R5
036452  020527  023420                          CMP     R5,#10000.
036456  004737  015240                          CALL    INSERT                  ;TEST IF ERROR MESSAGE IS REQUIRED
036462  001404                                  BEQ     20$                     ;IF EXPECTED NUMBER OF COUNTS
036464                                          ERRHRD  1001,,ERR120            ;ERROR HANDLER
                                        
```

```
HARDWARE TESTS  MACRO V05.00  Wednesday 03-Oct-84 14:03  Page 61-1
TEST 10  --  UP/DOWN COUNTING APPLICATION TEST

 5718 036474  012761  177422  000002  20$:    MOV     #B!22,MREA(R1)          ;SELECT HOLD REGISTER #2
 5719 036502  016105  000004                  MOV     MREB(R1),R5
 5720 036506  005705                          TST     R5                      ;OTHER SHOULD HAVE NOT COUNTED
 5721 036510  004737  015240                  CALL    INSERT                  ;TEST IF ERROR MESSAGE IS REQUIRED
 5722 036514  001404                          BEQ     30$                     ;IF EXPECTED NUMBER OF COUNTS
 5723 036516                                  ERRHRD  1002,,ERR121            ;ERROR HANDLER
 5724                                 ;
 5725                                 ; NOW DO THE SAME TEST WITH OTHER PHASE
 5726                                 ;
 5727 036526  012761  177411  000002  30$:    MOV     #B!11,MREA(R1)          ;SELECT 1 LOAD REGISTER
 5728 036534  005061  000004                  CLR     MREB(R1)                ;INIT COUNTER
 5729 036540  012761  177412  000002          MOV     #B!12,MREA(R1)          ;SELECT 2 LOAD REGISTER
 5730 036546  005061  000004                  CLR     MREB(R1)                ;INIT COUNTER
 5731 036552  012703  023420                  MOV     #10000.,R3              ;LOOP COUNT
 5732 036556  004537  037022                  JSR     R5,SETMRA               ;SET & TEST MREA
 5733 036562  000344                          .WORD   344                     ;CLEAR TOUT N = 4
 5734 036564  004537  037022                  JSR     R5,SETMRA               ;SET & TEST MREA
 5735 036570  000343                          .WORD   343                     ;CLEAR TOUT N = 3
 5736 036572  012761  177503  000002          MOV     #C$LOA!3,MREA(R1)       ;STARTS COUNTER
 5737 036600  012761  177443  000002          MOV     #C$ARM!3,MREA(R1)       ;STARTS COUNTER
 5738 036606  004537  037022          40$:    JSR     R5,SETMRA               ;SET & TEST MREA
 5739 036612  000353                          .WORD   353                     ;SET TOUT N = 3
 5740 036614  004537  037022                  JSR     R5,SETMRA               ;SET & TEST MREA
 5741 036620  000354                          .WORD   354                     ;SET TOUT N = 4
 5742 036622  004537  037022                  JSR     R5,SETMRA               ;SET & TEST MREA
 5743 036626  000343                          .WORD   343                     ;CLEAR TOUT N = 3
 5744 036630  004537  037022                  JSR     R5,SETMRA               ;SET & TEST MREA
 5745 036634  000344                          .WORD   344                     ;CLEARTOUT N = 4
 5746 036636  077315                          SOB     R3,40$
 5747
 5748 036640  012761  177603  000002          MOV     #C$DAS!3,MREA(R1)       ;DISARM AND SAVE COUNTERS
 5749 036646  012761  177421  000002          MOV     #B!21,MREA(R1)          ;COUNTER -> HOLD REGISTER #1
 5750 036654  016105  000004                  MOV     MREB(R1),R5
 5751 036660  005705                          TST     R5                      ;OTHER SHOULD HAVE NOT COUNTED
 5752 036662  004737  015240                  CALL    INSERT                  ;TEST IF ERROR MESSAGE IS REQUIRED
 5753 036666  001404                          BEQ     50$                     ;DID CLOCK TICK
 5754 036670                                  ERRHRD  1003,,ERR122            ;ERROR HANDLER
 5755 036700                          50$:
 5756 036700  012761  177422  000002          MOV     #B!22,MREA(R1)          ;COUNTER -> HOLD REGISTER #1
 5757 036706  016105  000004                  MOV     MREB(R1),R5
 5758 036712  020527  023420                  CMP     R5,#10000.
 5759 036716  004737  015240                  CALL    INSERT                  ;TEST IF ERROR MESSAGE IS REQUIRED
 5760 036722  001404                          BEQ     60$                     ;DID CLOCK TICK
 5761 036724                                  ERRHRD  1004,,ERR123            ;ERROR HANDLER
 5762
 5763 036734  005737  002164          60$:    TST     QVP                     ;IS QUICK VERIFY PASS SELECTED
 5764 036740  001026                          BNE     EXS10                   ;YES
 5765 036742                                  BREAK
 5766 036744  005737  000000G                 TST     WORFLA                  ;IS WORKING PRINTED ONCE ?
 5767 036750  001012                          BNE     120$                    ;BRANCH IF YES
 5768 036752                                  PRINTF  #WOR                    ;PRINT TEST IS WORKING
 5769 036772  005237  000000G                 INC     WORFLA                  ;
 5770 036776  005237  002504          120$:   INC     ITRCNT                  ;BUMP ITERATION COUNT
 5771 037002  023737  002506  002504          CMP     ITRDEF,ITRCNT           ;DEFAULT ITERATION COUNT
 5772 037010  001402                          BEQ     EXS10                   ;TIME TO EXIT
 5773 037012  000137  036222                  JMP     ITR10                   ;DO THE TEST AGAIN
 5774
```

```
5775 037016                                 EXS10:  EXIT    TST
5776
5777 037022  011500                         SETMRA: MOV     (R5),R0
5778 037024  052700  177400                        BIS     #8,R0                   ;SET HIGH ORDER BYTE
5779 037030  010061  000002                        MOV     R0,MREA(R1)             ;SET OUTPUT
5780 037034  016100  000002                        MOV     MREA(R1),R0             ;GET VALUE BACK
5781 037040  011502                                MOV     (R5),R2
5782 037042  042702  177770                        BIC     #177770,R2              ;GET NUMBER
5783 037046  011546                                MOV     (R5),-(SP)              ;WHAT TYPE OF COMMAND
5784 037050  042716  177407                        BIC     #177407,(SP)            ;GET COMMAND
5785 037054  022627  000350                        CMP     (SP)+,#350              ;SET?
5786 037060  001004                                BNE     1$                      ;NO
5787 037062  136200  037113                        BITB    BITMAP-1(R2),R0         ;IS SOURCE SET?
5788 037066  001010                                BNE     10$                     ;YES, AS IT SHOULD BE
5789 037070  000403                                BR      3$                      ;ERROR
5790 037072  136200  037113                1$:     BITB    BITMAP-1(R2),R0         ;OUTPUT CLEARED?
5791 037076  001404                                BEQ     10$                     ;YES, AS IT SHOULD BE
5792 037100                                 3$:     ERRHRD  1005,,ERR124            ;ERROR HANDLER
5793 037110  005725                         10$:    TST     (R5)+
5794 037112  000205                                RTS     R5
5795
5796 037114      002      004      010    BITMAP: .BYTE   2,4,10,20,40
     037117      020      040
5797                                               .EVEN
5798
5799                                               .NLIST  BEX
5800 037122      045      123      062    TSHD10::.ASCIZ  /%S2%AUP\DOWN COUNTING APPLICATION TEST %N/
5801 037174      045      116      045    WOR:    .ASCIZ  /%N%AWORKING%N/
5802                                               .LIST   BEX
5803                                               .EVEN
5804
5805 037212                                        ENDTST
```

```
                                        .SBTTL  TEST 11:  Visual LED Test - Specifically Selectable.
                                        ;*************************************************************************
                                        ;       Test 11 - Visual LED Test - Specifically Selectable.
                                        ;
                                        ;       FOR TEST DESCRIPTION SEE 6.0 TEST SUMMERIES
                                        ;*************************************************************************

037214                                          BGNTST
037214  004737  015006                        CALL    SELECT                  ; CALL THE SELECT ROUTINE
037220  010377                                .WORD   10377                   ; SPECIFIC TEST, ALL DEVICES
037222  037350                                TSHD11                          ; TEST HEADER ADDRESS
037224  103002                                BCC     10$                     ; BRANCH IF THE TEST IS SELECTED
037226                                        EXIT    TST                     ; OTHERWISE, EXIT THE TEST

037232  005001                        10$:    CLR     R1                      ; START BY SWITCHING LEDS ON

037234  062701  000100                20$:    ADD     #100,R1                 ; IF BIT 6 IS CLEAR, SET IT
037240  042701  000200                        BIC     #200,R1                 ; IF IT IS SET, CLEAR IT

037244  005037  002074                        CLR     L$LUN                   ; START WITH UNIT 0
037250  013703  002074                30$:    MOV     L$LUN,R3                ; FORM OFFSET FOR PARAMETER ADDRESS
037254  006303                                ASL     R3                      ;
037256  016302  002554                        MOV     GPADD(R3),R2            ; GET PARAMETER ADDRESS IN R2
037262  001006                                BNE     35$                     ; IF ADDRESS IS SET UP, BRANCH
037264                                        GPHARD  L$LUN,R2                ; ELSE, GET PARAM ADDRESS FROM DRS
037274  010263  002554                        MOV     R2,GPADD(R3)            ; AND SAVE IT IN THE TABLE
037300  005712                        35$:    TST     (R2)                    ; MODE ADDRESS = 0 ?
037302  001002                                BNE     40$                     ; IF NOT, BRANCH
037304  004737  015306                        JSR     PC,CONFIG               ; ELSE DO AUTO CONFIGURATION FOR THIS UNIT
037310  010172  000000                40$:    MOV     R1,@(R2)                ; SWITCH THE LED OVER
037314  005237  002074                        INC     L$LUN                   ; GO TO NEXT UNIT
037320  023737  002074  002012                CMP     L$LUN,L$UNITS           ; ALL CHANGED ?
037326  002750                                BLT     30$                     ; IF NOT, SWITCH OVER THE NEXT

037330  012703  000024                        MOV     #20.,R3                 ; WAIT FOR 0.5 SECONDS
037334  004737  014600                50$:    JSR     PC,WT25M                ; 25 MILLISECONDS WAIT ROUTINE
037340  005303                                DEC     R3                      ; * 20 = 0.5 SECONDS
037342  001374                                BNE     50$                     ; GO ON IF 0.5 SEC. ARE OVER
037344                                        BREAK                           ; ALLOW OPERATOR INPUT
037346  000732                                BR      20$                     ; DO IT ALL AGAIN

                                        .NLIST  BEX
037350      045     123     062     TSHD11::.ASCIZ  /<S2>AVISUAL LED TEST - SPECIFICALLY SELECTABLE.<N>/
                                        .LIST   BEX
                                        .EVEN

037432                                          ENDTST

037434                                          ENDMOD
```

```
                                .TITLE CLOSE SECTION

                                        BGNMOD

                                $PATCH::
                                        .BLKW   500

                                        .BLKB   400-<.&377>             ; SHIFT TO CORRECT FOR LSI BUG
                                        LASTAD
                                L$LAST::
                                        ENDMOD
```

```
5919
5920
5921
5934
5935 041004                                    BGNSETUP        16.
5936          000020                           .REPT   16.
5937                                           BGNPTAB
5938                                           .WORD   0,0,0,0,0,0,0,0
5939                                           ENDPTAB
5940                                           .ENDR
5941 041504                                    ENDSETUP
5942          000001                 .END
```

Symbol table

```
ACFLG   026212          CON     022516          C$GETW= 000027          EF.NEW= 000035 G        E104    010732
ADCON   017462 G        CONEX   016452          C$GMAN= 000043          EF.PWR= 000034 G        E105    011007
ADR   = 000020 G        CONFIG  015306 G        C$GPHR= 000042          EF.RES= 000037 G        E106    011062
AI      017200          CONMSK  002702 G        C$GPLO= 000030          EF.STA= 000040 G        E106A   011140
AMREST  021470 G        CONPRI  015656 G        C$GPRI= 000040          ELEMNT  002450 G        E107    011201
ANS     002502 G        CONT    025410          C$INIT= 000011          EMG1    010336          E107A   011257
AO      017211          COUNTR  002444 G        C$INLP= 000020          EMG2    010405          E108    011321
ASK     022026          CO1     015576          C$IN1 = 177750 G        EMG3    010455          E109    011373
ASKADD  022742          CP1     016466          C$IN2 = 177757 G        EMG4    010542          E110    011443
ASSEMB= 000010          CP1A    016537          C$LOA = 177500 G        END     026202          E111    011516
B     = 177400 G        CP1B    016611          C$MANI= 000050          ERRBLK  005720 G        E112    011567
BAD     002476 G        CP1C    016707          C$MAR = 177777 G        ERRMSG  005716 G        E113    011641
BADIV   022004 G        CP10    017110          C$MEM = 000031          ERRNBR  005714 G        E114    011713
BITMAP  037114          CP10A   017125          C$MSG = 000023          ERRTYP  005712 G        E114A   012000
BITMSK  002434 G        CP11    017143          C$OPEN= 000034          ERR104  006346 G        E114B   012052
BITMS1  002436 G        CP12    017147          C$PNTB= 000014          ERR105  006400 G        E115    012126
BITMS2  002440 G        CP2     016741          C$PNTF= 000017          ERR106  006432 G        E116    012442
BIT0  = 000001 G        CP3     016752          C$PNTS= 000016          ERR107  006512 G        E116A   012525
BIT00 = 000001 G        CP4     016761          C$PNTX= 000015          ERR108  006572 G        E116B   012606
BIT01 = 000002 G        CP5     016773          C$QIO = 000377          ERR109  006624 G        E120    012662
BIT02 = 000004 G        CP6     017010          C$ROBU= 000007          ERR110  006656 G        E121    012747
BIT03 = 000010 G        CP7     017016          C$REFG= 000047          ERR111  006710 G        E124    013030
BIT04 = 000020 G        CP8     017066          C$RESE= 000033          ERR112  006742 G        E201    030617
BIT05 = 000040 G        CP9     017077          C$REVI= 000003          ERR115  007412 G        E202    030704
BIT06 = 000100 G        CRLF    014634 G        C$RFLA= 000021          ERR116  007560 G        E203    030751
BIT07 = 000200 G        CSR     002420 G        C$RPT = 000025          ERR120  010034 G        E501    013113
BIT08 = 000400 G        C$ARM = 177440 G        C$SEFG= 000046          ERR121  010104 G        E502    013146
BIT09 = 001000 G        C$AU  = 000052          C$SPRI= 000041          ERR122  010134 G        E506    013235
BIT1  = 000002 G        C$AUTO= 000061          C$STN = 177750 G        ERR123  010164 G        E507    013333
BIT10 = 002000 G        C$BRK = 000022          C$SVEC= 000037          ERR124  010234 G        E507A   013432
BIT11 = 004000 G        C$BSEG= 000004          C$TPRI= 000013          ERR130  007444 G        FLASH   020716 G
BIT12 = 010000 G        C$BSUB= 000002          DACON   017244 G        ERR501  006774 G        FLSANS  021170 G
BIT13 = 020000 G        C$CEFG= 000045          DCNT    017222          ERR502  007050 G        F$AU  = 000015
BIT14 = 040000 G        C$CLCK= 000062          DECEX   020626          ERR503  007124 G        F$AUTO= 000020
BIT15 = 100000 G        C$CLEA= 000012          DECIN   017736 G        ERR504  007152 G        F$BGN = 000040
BIT2  = 000004 G        C$CLOS= 000035          DECIN1  020347          ERR505  007204 G        F$CLEA= 000007
BIT3  = 000010 G        C$CLP1= 000006          DECIN2  020405          ERR506  007262 G        F$DU  = 000016
BIT4  = 000020 G        C$CTN = 177740 G        DECIN3  020466          ERR507  007340 G        F$END = 000041
BIT5  = 000040 G        C$CVEC= 000036          DECOUT  020516 G        ER116A  007634 G        F$HARD= 000004
BIT6  = 000100 G        C$DAC = 177700 G        DECO1   020634          ER116B  007710 G        F$HW  = 000013
BIT7  = 000200 G        C$DAS = 177600 G        DECO2   020640          ER116C  007762 G        F$INIT= 000006
BIT8  = 000400 G        C$DCLN= 000044          DECO3   020644          EVL   = 000004 G        F$JMP = 000050
BIT9  = 001000 G        C$DODU= 000051          DFPTBL  002154 G        EXQV1   030170          F$MOD = 000000
BIV     002472 G        C$DRPT= 000024          DI      017156          EXQV2   030570          F$MSG = 000011
BOE   = 000400 G        C$DU  = 000053          DIAGMC= 000000          EXQV3   031212          F$PROT= 000021
CALRET  021270 G        C$EDIT= 000003          DO      017167          EXS10   037016          F$PWR = 000017
CARRFL  021374          C$ERDF= 000055          DROPD   027734          EXS4    032106          F$RPT = 000012
CCR     002422 G        C$ERHR= 000056          DROPED  002614 G        EXS5    033442          F$SEG = 000003
CHAR    023162          C$ERRO= 000060          ECNT    002514 G        EXS6    034072          F$SOFT= 000005
CHKMAX  014434 G        C$ERSF= 000054          EERA    006250 G        EXS7    034572          F$SRV = 000010
CLINT   022024 G        C$ERSO= 000057          EERB    006300 G        EXS8    035264          F$SUB = 000002
CLKFLG  027526          C$ESCA= 000010          EERG    006340 G        EXS9    036104          F$SW  = 000014
CNTVAL  002452 G        C$ESEG= 000005          EER1    006064 G        E$END = 002100          F$TEST= 000001
CNTXT   012403          C$ESUB= 000003          EER2    006122 G        E$LOAD= 000035          GAIN    002706 G
CNT25   014632 G        C$ETST= 000001          EER3    006164 G        E101    030240          GETNUM  020327
CNT25M  014626 G        C$EXIT= 000032          EER6    006216 G        E102    010604          GOOBAD  010306
CNT500  014630 G        C$GETB= 000026          EF.CON= 000036 G        E103    010662          GOOD    002474 G
```

Symbol table

```
GPADD   002554 G        I$DU  = 000041        L$HPCP  002016 G        L10037  007760        L10131  041440
GPFLG   023160          I$HRD = 000041        L$HPTP  002022 G        L10040  010032        L10132  041434
GROUP   002446 G        I$INIT= 000041        L$HW    002154 G        L10041  010102        L10133  041464
G$CNTO= 000200          I$MOD = 000041        L$ICP   002104 G        L10042  010132        L10134  041460
G$DELM= 000372          I$MSG = 000041        L$INIT  025022 G        L10043  010162        L10136  041504
G$DISP= 000003          I$PROT= 000040        L$LADP  002026 G        L10044  010232        MASCOM  014050
G$EXCP= 000400          I$PTAB= 000041        L$LAST  041004 G        L10045  010304        MASK    014046 G
G$HILI= 000002          I$PWR = 000041        L$LOAD  002100 G        L10046  014120        MOD     002416 G
G$LOLI= 000001          I$RPT = 000041        L$LUN   002074 G        L10047  021774        MODE    002704 G
G$NO  = 000000          I$SEG = 000041        L$MREV  002050 G        L10050  022002        MREA  = 000002 G
G$OFFS= 000400          I$SETU= 000041        L$NAME  002000 G        L10051  022022        MREB  = 000004 G
G$OFSI= 000376          I$SFT = 000041        L$PRIO  002042 G        L10052  022024        MREC  = 000006 G
G$PRMA= 000001          I$SRV = 000041        L$PROT  025014 G        L10053  025012        M$TOG = 000042 G
G$PRMD= 000002          I$SUB = 000041        L$PRT   002112 G        L10055  026646        NERRS   014536
G$PRML= 000000          I$TST = 000041        L$REPP  002062 G        L10056  027662        NEWST   025320
G$RADA= 000140          IOTAB   004312 G      L$REV   002010 G        L10057  027672        NEXT    025654
G$RADB= 000000          I4TAB   005112 G      L$RPT   022026 G        L10060  027764        NH    = 000014
G$RADD= 000040          J$JMP = 000167        L$SOFT  002366 G        L10061  027776        NO      025006
G$RADL= 000120          KLINT   027102        L$SPC   002056 G        L10062  030306        NOCLOK  027240
G$RADO= 000020          LASTFA  015304        L$SPCP  002020 G        L10063  030204        NODEV   005772 G
G$XFER= 000004          LCLOCK  026766        L$SPTP  002024 G        L10064  031026        NOUNIT  015574
G$YES = 000010          LF      014656        L$STA   002030 G        L10065  031252        NR1     020310
G1      002222          LOCATE  030202 G      L$SW    002164 G        L10066  032162        NR2     020312
G16     002374          LOE   = 040000 G      L$TEST  002114 G        L10067  033524        NTEST   026371
G2      002250          LOOP    002466 G      L$TIML  002014 G        L10070  034140        NTESTS= 000013
G3      002270          LOOPEX  026214        L$UNIT  002012 G        L10071  034652        NTEST1  026465
HEL     022142          LOPFLG  002640 G      L10000  002162        L10072  035344        NUMBER  002442 G
HELP  = 000000          LOT   = 000010 G      L10001  002166        L10073  036160        NXM     021766 G
HOE   = 100000 G        LOTFLA  002700 G      L10002  002222        L10074  037212        NXMFLG  002470 G
IBE   = 010000 G        L$ACP   002110 G      L10003  002374        L10075  037432        NXTUUT  025544
IDU   = 000040 G        L$APT   002036 G      L10004  006120        L10076  041010        ONEFIL= 000001
IER   = 020000 G        L$AU    027766 G      L10005  006162        L10077  041034        O$APTS= 000001
INIUUT  025610          L$AUT   002070 G      L10006  006214        L10100  041030        O$AU  = 000001
INR     002424 G        L$AUTO  027600 G      L10007  006246        L10101  041060        O$BGNR= 000001
INSERT  015240 G        L$CCP   002106 G      L10010  006276        L10102  041054        O$BGNS= 000001
INTFLA  002510 G        L$CLEA  027664 G      L10011  006336        L10103  041104        O$DU  = 000001
INTFL2  002512 G        L$CO    002032 G      L10012  006344        L10104  041100        O$ERRT= 000001
INTSR   021776 G        L$DEPO  002011 G      L10013  006376        L10105  041130        O$GNSW= 000001
ISR   = 000100 G        L$DESC  005732 G      L10014  006430        L10106  041124        O$POIN= 000001
ITRAC1  030044          L$DESP  002076 G      L10015  006510        L10107  041154        O$SETU= 000001
ITRAC2  030326          L$DEVP  002060 G      L10016  006570        L10110  041150        PADD    002710 G
ITRAC3  031052          L$DISP  002124 G      L10017  006622        L10111  041200        PNT   = 001000 G
ITRAC4  031306          L$DLY   002116 G      L10020  006654        L10112  041174        PRA     023326
ITRAC5  032336          L$DTP   002040 G      L10021  006706        L10113  041224        PREX    023152
ITRAC6  033560          L$DTYP  002034 G      L10022  006740        L10114  041220        PRI   = 002000 G
ITRAC7  034174          L$DU    027674 G      L10023  006772        L10115  041250        PRIO    002430 G
ITRAC8  034722          L$DUT   002072 G      L10024  007046        L10116  041244        PRIO0 = 000000 G
ITRAC9  035414          L$DVTY  005722 G      L10025  007122        L10117  041274        PRIO1 = 000040 G
ITRCNT  002504 G        L$EF    002052 G      L10026  007150        L10120  041270        PRIO2 = 000100 G
ITRDEF  002506 G        L$ENVI  002044 G      L10027  007202        L10121  041320        PRIO3 = 000140 G
ITR10   036222          L$ERPT  005712 G      L10030  007260        L10122  041314        PRIO4 = 000200 G
IXE   = 004000 G        L$ETP   002102 G      L10031  007336        L10123  041344        PRIO5 = 000240 G
IXEND = 171770 G        L$EXP1  002046 G      L10032  007410        L10124  041340        PRIO6 = 000300 G
IXSTA = 171000 G        L$EXP4  002064 G      L10033  007442        L10125  041370        PRIO7 = 000340 G
I$AU  = 000041          L$EXP5  002066 G      L10034  007556        L10126  041364        PROMT   021172
I$AUTO= 000041          L$HARD  002170 G      L10035  007632        L10127  041414        PR1     023420
I$CLN = 000041          L$HIME  002120 G      L10036  007706        L10130  041410        PR2     023445
```

Symbol table

```
PR2A    023552          RSAV    031016          TPS   - 177564          T$PTAB- 010135          T10     036162 G
PR2B    023651          RUNAK   031242          TSHD1   030206 G        T$PTHV- 000020          T11     037214 G
PR2C    023743          SAVCNT  027444          TSHD10  037122 G        T$PTNU- 000020          T2      030310 G
PR2D    024004          SAVPRI  002432 G        TSHD11  037350 G        T$SAVL- 177777          T3      031030 G
PR2E    024074          SELECT  015006 G        TSHD2   030574 G        T$SEGL- 177777          T4      031254 G
PR2F    024177          SETCLK  026650          TSHD3   031216 G        T$SEKO- 010000          T5      032164 G
PR2G    024273          SETEX   027524          TSHD4   032120 G        T$SIZE- 000240          T6      033526 G
PR3     024367          SETMRA  037022          TSHD5   033474 G        T$SUBN- 000000          T7      034142 G
PR3A    024474          SFI     002500 G        TSHD6   034076 G        T$TAGL- 177777          T8      034654 G
PR4     024533          SFPTBL  002164 G        TSHD7   034576 G        T$TAGN- 010137          T9      035346 G
PR5     024564          SNUM    020314          TSHD8   035270 G        T$TEMP- 000000          UAM   - 000200 G
PR6     024605          SRC     002464 G        TSHD9   036110 G        T$TEST- 000013          USCLOK  027174
PR7     024706          STADD   015572 G        TSTEN1  030176          T$TSTM- 177777          VBPTAB  003512 G
PSAD    017540          STARES  025306          TSTFLG  002634 G        T$TSTS- 000001          VEC     002426 G
PSDA    017316          START   025022          TSUFLG  002636 G        T$$AU - 010061          VUPTAB  002712 G
PSEUL1  026162          STAT    022766          TT      023220          T$$AUT- 010056          WAIT    014620
PUREFL  025466          STAVA2  012307          TTINT   027372          T$$CLE- 010057          WARN    026216
QVP     002164 G        STFLG   023156          T$ARGC- 000001          T$$DAT- 010136          WARN1   026276
RA      020712 G        STRVA1  012173          T$CODE- 000152          T$$DU - 010060          WFLG    014710
RANDOM  020654 G        STRVA2- ****** GX       T$ERRN- 001755          T$$HAR- 010002          WOR     037174
RB      020714 G        SVCGBL- 000000          T$EXCP- 000000          T$$HW - 010000          WORFLA- ****** GX
RDY     023264          SVCINS- 177777          T$FLAG- 000040          T$$INI- 010055          WRDY    014662 G
RECON   022244          SVCSUB- 177777          T$FREE- 041504          T$$MSG- 010046          WRDY1   014712
REGADD  014052 G        SVCTAG- 177777          T$GMAN- 000000          T$$PC - 000020          WRSEL   026532
REGERR  014054 G        SVCTST- 177777          T$HILI- 000001          T$$PRO- 010054          WT25    014614 G
REGMSG  014341          S$LSYM- 010000          T$LAST- 000001          T$$PTA- 010135          WT25M   014600 G
REGTST  013474 G        TADS    023172          T$LOLI- 000000          T$$RPT- 010053          WT500   014606 G
REGTS1  013532 G        TIMSG   027530          T$LSYM- 010000          T$$SEG- 010000          X$ALWA- 000000
RERR1   014122          TITLE   022350          T$LTNO- 000013          T$$SOF- 010003          X$FALS- 000040
RERR2   014174          TKB   - 177562          T$NEST- 177777          T$$SRV- 010063          X$OFFS- 000400
RERR3   014252          TKS   - 177560          T$NS0 - 000000          T$$SW - 010001          X$TRUE- 000020
RESTRT  025276          TNA     015166          T$NS1 - 000001          T$$TES- 010075          YES     025002
RETME1  021376          TNUM    023322          T$NS2 - 000003          T%      030000 G        $PATCH  037434 G
RFLG    023170          TPB   - 177566          T$PCNT- 000000

. ABS.  041504    000   (RW,I,GBL,ABS,OVR)
        000000    001   (RW,I,LCL,REL,CON)
Errors detected:  0

*** Assembler statistics


Work  file  reads: 233
Work  file writes: 236
Size of work file: 28944 Words  ( 114 Pages)
Size of core pool: 19990 Words  ( 76 Pages)
Operating  system: RSX-11M/PLUS

Elapsed time: 00:08:43.64
ZIDVA0.BIN,ZIDVA0.SEQ/-SP=[50,200]SVC/ML,[53,53]ZIDVA0.SRC
```