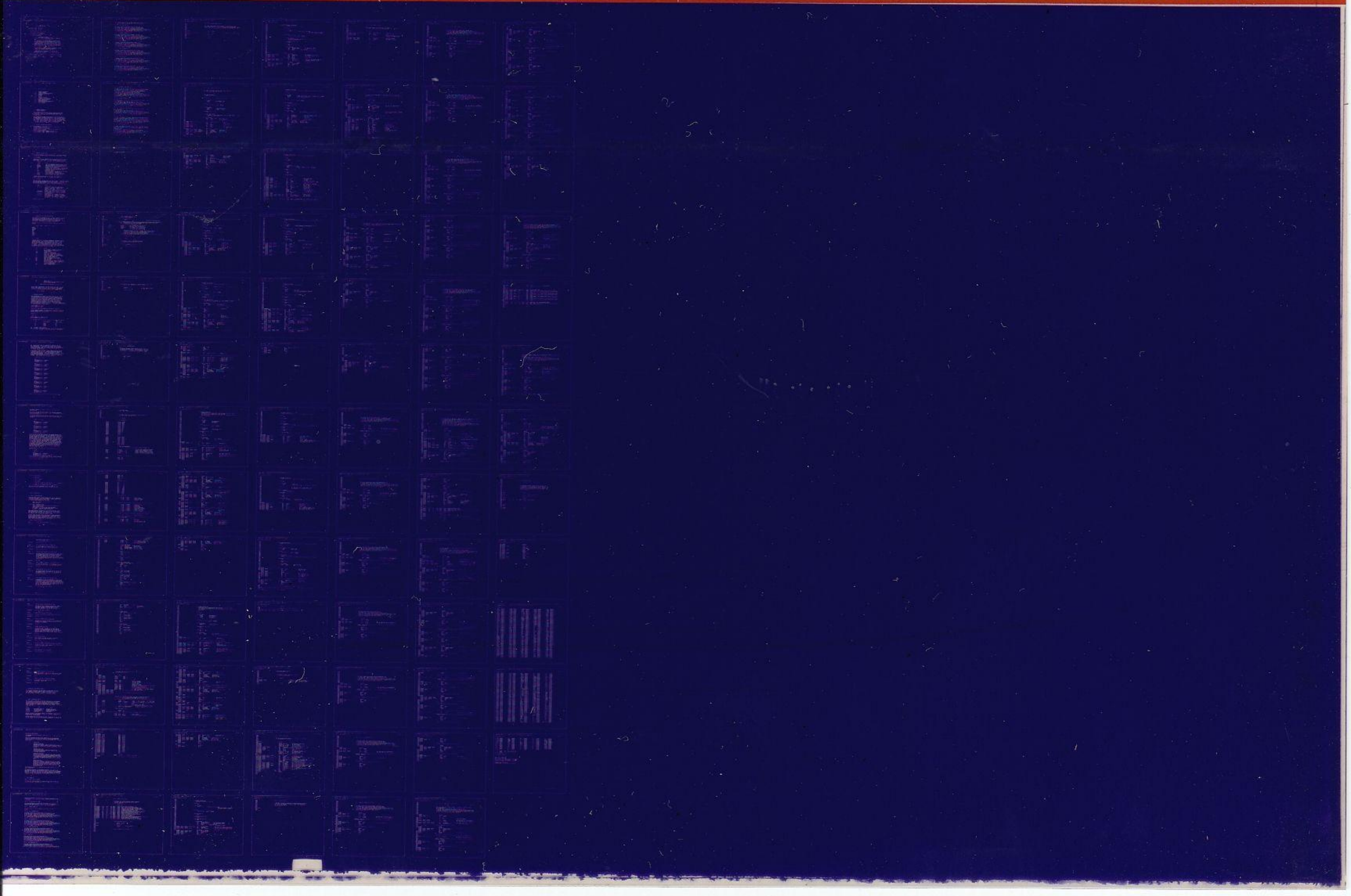
VSV21

VSV21 DIAG
CZVSWA0

AH-FF62A-MC
1 OF 1 OCT 1985

digital
MADE IN USA



```
.REM &

                    IDENTIFICATION
                    ------ -------

PRODUCT CODE:  AC-FF61A-MC

PRODUCT NAME:  CZVSWAO

PRODUCT DATE:  4 JUL 85

MAINTAINER:  CSS ENGINEERING

AUTHOR:  JOHN BOWSKILL
```


```
&
```

```
.REM &
```

TABLE OF CONTENTS

1.0 GENERAL INFORMATION

1.1 PROGRAM ABSTRACT

THIS DIAGNOSTIC VERIFIES THAT DEVICES UNDER TEST ARE VSV21 DEVICES AND CAUSES ALL VSV21 ON-BOARD MICRODIAGNOSTICS TO BE EXECUTED. IT PROVIDES COMPLETE FUNCTION LEVEL COVERAGE AND FRU CALLOUT.

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+, ACT, APT, SLIDE AND PAPER TAPE. 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 & HARDWARE REQUIREMENTS

THIS DIAGNOSTIC IS DESIGNED TO RUN ON A 11/23+ OR A 11/73 WITH A MINIMUM MEMORY OF 32 K BYTES.
THE MODULE UNDER TEST IS A M7656.
TESTS 4 , 13 , 14 , 16 AND 17 REQUIRE PART NUMBER VSV21-AJ WHICH CONSISTS OF THE FOLLOWING:
1 OFF 12-15336-01 CONN, LOOPBACK EIA DATA 25-WAY
2 OFF 70-20130-01 CONN, LOOPBACK EIA DATA 9-WAY
1 OFF 70-20131-01 CONN, LOOPBACK KEYBOARD JACK

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 STATISTICAL INFORMATION (IF IMPLEMENTED BY THE DIAGNOSTIC - 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
IBR*	INHIBIT ALL ERROR REPORTS EXCEPT FIRST LEVEL (FIRST LEVEL CONTAINS ERROR TYPE, NUMBER, PC, TEST AND UNIT)
IXR*	INHIBIT EXTENDED ERROR REPORTS (THOSE CALLED BY PRINTX MACRO'S)
PRI	DIRECT MESSAGES TO LINE PRINTER
PNT	PRINT TEST NUMBER AS TEST EXECUTES
BOE	"BELL" ON ERROR
UAM	UNATTENDED MODE (NO MANUAL INTERVENTION)
ISR	INHIBIT STATISTICAL REPORTS (DOES NOT APPLY TO DIAGNOSTICS WHICH DO NOT SUPPORT STATISTICAL REPORTING)
IDR	INHIBIT PROGRAM DROPPING OF UNITS
ADR	EXECUTE AUTODROP CODE

LOT	LOOP ON TEST
EVL	EXECUTE EVALUATION (ON DIAGNOSTICS WHICH HAVE EVALUATION SUPPORT)

*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 A DIAGNOSTIC IS STARTED, THE RUNTIME SERVICES WILL PROMPT THE USER FOR HARDWARE INFORMATION BY TYPING "CHANGE HW (L) ?" YOU MUST ANSWER "Y" AFTER A START COMMAND UNLESS THE HARDWARE INFORMATION HAS BEEN "PRELOADED" USING THE SETUP UTILITY (SEE CHAPTER 6 OF THE XXDP+ USER'S MANUAL) OR YOU WISH TO USE THE HARDCODED DEFAULTS (SEE BELOW) OR YOU HAVE JUST RUN THE DIAGNOSTICS AND WISH TO KEEP THE SAME HARDWARE INFORMATION. WHEN YOU ANSWER THIS QUESTION WITH A "Y", THE RUNTIME SERVICES WILL ASK FOR THE NUMBER OF UNITS (IN DECIMAL). YOU WILL THEN BE ASKED THE FOLLOWING QUESTIONS FOR EACH UNIT.

```
DEVICE ADDRESS (O) 172010 ?
VECTOR ADDRESS (O) 320 ?
```

WHERE DEVICE AND VECTOR ADDRESS ARE DEFINED IN SECTION 5.0.

IN THE FOLLOWING EXAMPLE THE USER SELECTS 1 UNIT WITH 172020 FOR DEVICE ADDRESS INSTEAD OF THE DEFAULT 172010 AND 324 FOR VECTOR ADDRESS INSTEAD OF THE DEFAULT 320.

```
CHANGE HW (L) ? Y

# UNITS (D) ? 1

DEVICE ADDRESS (O) 172010 ? 172020
VECTOR ADDRESS (O) 320 ? 324
```

THE HARDCODED DEFAULTS ARE FOR 4 UNITS WITH THE FOLLOWING ADRESSES:

UNIT	DEVICE ADDRESS	VECTOR ADDRESS
#1	172010	320
#2	172020	324
#3	172030	330
#4	172040	334

2.5 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 HAVE A MULTIPLEXED DEVICE SUCH AS
A MASS STORAGE CONTROLLER WITH SEVERAL DRIVES OR A COMMUNICATION
DEVICE WITH SEVERAL LINES, THIS BECOMES TEDIOUS SINCE MOST OF
THE ANSWERS ARE REPETITIOUS.

TO ILLUSTRATE A MORE EFFICIENT METHOD, SUPPOSE YOU ARE TESTING
A FICTIONAL DEVICE, THE XY11. SUPPOSE THIS DEVICE CONSISTS OF
A CONTROL MODULE WITH EIGHT UNITS (SUB-DEVICES) ATTACHED TO IT.
THESE UNITS ARE DESCRIBED BY THE OCTAL NUMBERS 0 THROUGH 7. THERE
IS ONE HARDWARE PARAMETER THAT CAN VARY AMONG UNITS CALLED THE
Q-FACTOR. THIS Q-FACTOR MAY BE 0 OR 1. BELOW IS A SIMPLE WAY
TO BUILD A TABLE FOR ONE XY11 WITH EIGHT UNITS.

```
# UNITS (D) ?  8<CR>

UNIT 1
CSR ADDRESS (O) ?  160000<CR>
SUB-DEVICE # (O) ?  0<CR>
Q-FACTOR (O) 0 ?  1<CR>

UNIT 2
CSR ADDRESS (O) ?  160000<CR>
SUB-DEVICE # (O) ?  1<CR>
Q-FACTOR (O) 1 ?  0<CR>

UNIT 3
CSR ADDRESS (O) ?  160000<CR>
SUB-DEVICE # (O) ?  2<CR>
Q-FACTOR (O) 0 ?  <CR>

UNIT 4
CSR ADDRESS (O) ?  160000<CR>
SUB-DEVICE # (O) ?  3<CR>
Q-FACTOR (O) 0 ?  <CR>

UNIT 5
CSR ADDRESS (O) ?  160000<CR>
SUB-DEVICE # (O) ?  4<CR>
Q-FACTOR (O) 0 ?  <CR>

UNIT 6
CSR ADDRESS (O) ?  160000<CR>
SUB-DEVICE # (O) ?  5<CR>
Q-FACTOR (O) 0 ?  <CR>

UNIT 7
CSR ADDRESS (O) ?  160000<CR>
SUB-DEVICE # (O) ?  6<CR>
Q-FACTOR (O) 0 ?  1<CR>

UNIT 8
CSR ADDRESS (O)  160000<CR>
SUB-DEVICE # (O) ?  7<CR>
Q-FACTOR (O) 1 ?  <CR>
```

NOTICE THAT THE DEFAULT VALUE FOR THE Q-FACTOR CHANGES WHEN A

NON-DEFAULT RESPONSE IS GIVEN. BE CAREFUL WHEN SPECIFYING MULTIPLE UNITS!

AS YOU CAN SEE FROM THE ABOVE EXAMPLE, THE HARDWARE PARAMETERS DO NOT VARY SIGNIFICANTLY FROM UNIT TO UNIT. THE PROCEDURE SHOWN IS NOT VERY EFFICIENT.

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

```
        # UNITS (D) ?  8<CR>

        UNIT 1
        CSR ADDRESS (O) ?  160000<CR>
        SUB-DEVICE # (O) ?  0,1<CR>
        Q-FACTOR (O) O ?  1,0<CR>

        UNIT 3
        CSR ADDRESS (O) ?  160000<CR>
        SUB-DEVICE # (O) ?  2-5<CR>
        Q-FACTOR (O) O ?  0<CR>

        UNIT 7
        CSR ADDRESS (O) ?  160000<CR>
        SUB-DEVICE # (O) ?  6,7<CR>
        Q-FACTOR (O) O ?  1<CR>
```

AS YOU CAN SEE IN THE ABOVE DIALOGUE, THE RUNTIME SERVICES WILL BUILD AS MANY ENTRIES AS IT CAN WITH THE INFORMATION GIVEN IN ANY ONE PASS THROUGH THE QUESTIONS. IN THE FIRST PASS TWO ENTRIES ARE BUILT SINCE TWO SUB-DEVICES AND Q-FACTORS WERE SPECIFIED. THE SERVICES ASSUME THAT THE CSR ADDRESS IS 160000 FOR BOTH SINCE IT WAS SPECIFIED ONLY ONCE. IN THE SECOND PASS, FOUR ENTRIES WERE BUILT. THIS IS BECAUSE FOUR SUB-DEVICES WERE SPECIFIED. THE "-" CONSTRUCT TELLS THE RUNTIME SERVICES TO INCREMENT THE DATA FROM THE FIRST NUMBER TO THE SECOND. IN THIS CASE, SUB-DEVICES 2, 3, 4 AND 5 WERE SPECIFIED. (IF THE SUB-DEVICE WERE SPECIFIED BY ADDRESSES, THE INCREMENT WOULD BE BY 2 SINCE ADDRESSES MUST BE ON AN EVEN BOUNDARY.) THE CSR ADDRESSES AND Q-FACTORS FOR THE FOUR ENTRIES ARE ASSUMED TO BE 160000 AND 0 RESPECTIVELY SINCE THEY WERE ONLY SPECIFIED ONCE. THE LAST TWO UNITS ARE SPECIFIED IN THE THIRD PASS.

THE WHOLE PROCESS COULD HAVE BEEN ACCOMPLISHED IN ONE PASS AS SHOWN BELOW.

```
        # UNITS (D) ?  8<CR>

        UNIT 1
        CSR ADDRESS (O) ?  160000<CR>
        SUB-DEVICE # (O) ?  0-7<CR>
        Q-FACTOR (O) O ?  0,1,0,...,1,1<CR>
```

AS YOU CAN SEE FROM THIS EXAMPLE, NULL REPLIES (COMMAS ENCLOSING A NULL FIELD) TELL THE RUNTIME SERVICES TO REPEAT THE LAST REPLY.

2.6 QUICK START-UP PROCEDURE (XXDP+)

TO START-UP THIS PROGRAM:

1. BOOT XXDP+
2. TYPE "R ZVSWA0"
3. TYPE "START"
4. ANSWER THE "CHANGE HW" QUESTION WITH "Y" (IF APPROPRIATE)
5. ANSWER ALL THE HARDWARE QUESTIONS IF ANSWER TO 4 WAS "Y"

WHEN YOU FOLLOW THIS PROCEDURE YOU WILL BE USING ONLY THE DEFAULTS FOR FLAGS. THESE DEFAULTS ARE DESCRIBED IN SECTION 2.3.

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

ERROR 1 : TEST 1 MUST BE PERFORMED FIRST

EXPLANATION: ALL TESTS (EXCEPT TEST 1) REQUIRE TEST 1 TO HAVE BEEN

RUN FIRST TO VERIFY THAT THE DEVICE IS A VSV21.
TEST 1 SHOULD NOW BE RUN.

OCCURRENCE: ANY TEST EXCEPT TEST 1.

ERROR 2 : TEST 2 MUST BE RUN FIRST TO ENABLE DMA

EXPLANATION: ALL TESTS NEEDING DMA ENABLED REQUIRE TEST 2 TO HAVE BEEN RUN FIRST. TEST 2 SHOULD NOW BE RUN.

OCCURRENCE: TESTS 15,16,17 AND 18

ERROR 3 : TIMED OUT WHILE WAITING FOR STATUS VALID

EXPLANATION: THE STATUS VALID BIT IN THE CSR FOR THE CURRENT DEVICE HAS NOT BEEN SET WITHIN A SUFFICIENTLY SHORT TIME. THIS NORMALLY MEANS THAT THE ON BOARD FIRMWARE IS WAITING FOR SOMETHING AND HAS NOT SET THE BIT. AS THIS INDICATES WHEN THE CSR IS IN A VALID STATE TO THE HOST FURTHER TESTING OF THIS DEVICE IS NOT POSSIBLE UNTIL FIXED.

OCCURRENCE: ANY TEST.

ERROR 4 : THERE HAS BEEN A POWERUP - PLEASE REPEAT TEST 1

EXPLANATION: THE VSV21 HAS BEEN RESET. IT IS NECESSARY TO DO TEST 1 AGAIN. NOTE THAT DMA WILL ALSO HAVE BEEN DISABLED.

OCCURRENCE: ANY TEST.

ERROR 5 : CONTROLLER READY NOT SET IN CSR

EXPLANATION: THE CONTROLLER READY BIT IS NOT SET IN THE CSR FOR THE CURRENT DEVICE. THIS MEANS THAT NO COMMAND CAN BE WRITTEN TO IT BY THE HOST. TRY REPEATING TEST 1 AFTER INVESTIGATION.

OCCURRENCE: ANY TEST.

ERROR 6 : STATUS PACKET SHORTER THAN SPECIFIED LENGTH

EXPLANATION: IN PROGRAMMED I/O THE HOST PROCESSES A STATUS PACKET BY READING THE PARAMETER REGISTER AND EXTRACTING THE PACKET WORD BY WORD. IN THIS CASE THE LENGTH SPECIFIED IN THE FIRST WORD INDICATES THAT THERE IS MORE DATA TO COME BUT THE PARAMETER READY BIT IS NOT SET IN THE CSR AND THUS THE PARAMETER REGISTER CANNOT BE READ TO GET THE REST OF THE STATUS PACKET.

OCCURRENCE: ANY TEST WHEN DMA IS NOT ENABLED.

ERROR 7 : STATUS AVAILABLE EXPECTED BUT NOT FOUND

EXPLANATION: THE HOST HAS SENT A COMMAND TO THE VSV21 (WHICH HAS DMA ENABLED) AND IS THUS EXPECTING A 'STATUS AVAILABLE' MESSAGE IN THE CSR WHEN THE VSV21 HAS FINISHED PROCESSING. THIS HAS NOT HAPPENED.

OCCURRENCE: ANY TEST WHEN DMA IS ENABLED.

ERROR 8 : ERROR PACKET RECEIVED FROM VSV21

EXPLANATION: THE HOST HAS RECEIVED AN ERROR PACKET FROM THE VSV21.

OCCURRENCE: ANY TEST.

ERROR 9 : FAILED TO TRANSMIT STATUS ACKNOWLEDGE

EXPLANATION: THE HOST HAS ATTEMPTED TO SEND A 'STATUS ACKNOWLEDGE' COMMAND VIA THE CSR BUT THE CONTROLLER READY BIT HAS NOT BEEN SET AFTERWARDS.

OCCURRENCE: ANY TEST.

ERROR 10: TIMEOUT WAITING FOR STATUS

EXPLANATION: THE HOST HAS SENT A COMMAND TO THE VSV21 AND IS EXPECTING A STATUS ACKNOWLEDGE VIA THE PARAMETER REGISTER. THE PARAMETER READY BIT IN THE CSR HAS NOT BEEN SET WITHIN A SUFFICIENTLY SHORT TIME AS THIS INDICATES WHEN THE HOST CAN READ THE PARAMETER REGISTER.

OCCURRENCE: ANY TEST.

ERROR 11: DMA DATA MATCH ERROR

EXPLANATION: DATA RECEIVED BACK FROM THE VSV21 IS DIFFERENT TO THAT SENT. THERE IS SOMETHING WRONG WITH DMA.

OCCURRENCE: TESTS 2,16,17

ERROR 12: FAILED TO GET MICRO DIAGNOSTICS RETURN STATUS PACKET

EXPLANATION: THE HOST HAS FAILED TO GET A RETURN STATUS PACKET AFTER INVOKING A MICRODIAGNOSTICS TEST.

OCCURRENCE: TESTS 4-19

ERROR 13: MICRO DIAGNOSTIC TEST UNSUCCESSFUL

EXPLANATION: THE MICRODIAGNOSTIC TEST HAS FAILED.

OCCURRENCE: TESTS 4-19.

ERROR 14: ERROR PACKET EXPECTED BUT NOT RECEIVED

EXPLANATION: AN INVALID COMMAND IS SENT TO THE VSV21 IN TEST 3 AND SO AN ERROR PACKET IS EXPECTED. THIS DID NOT HAPPEN.

OCCURRENCE: TEST 3.

ERROR 15: TIMED OUT WAITING TO RECEIVE AN INTERRUPT

EXPLANATION: AN INTERRUPT HAS NOT BEEN RECEIVED WITHIN A SUFFICIENTLY SHORT TIME.

OCCURRENCE: TEST 20.

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.

5.0 DEVICE INFORMATION TABLES

THE VSV21 MODULE INTERFACES TO THE HOST PROCESSOR VIA THE PROGRAMMED I/O INTERFACE. THIS USES THREE REGISTERS WITHIN A BLOCK OF FOUR ADDRESSABLE WORD LOCATIONS ON THE HOST BUS. THE BLOCK IS ASSIGNED AN ADDRESS BEGINNING ON A MODULO-4 WORD BOUNDARY VIA HARDWARE ADDRESS SELECT SWITCHES. THIS ADDRESS IS TERMED THE DEVICE ADDRESS. THE BLOCK LOOKS LIKE THIS:

ADDRESS	READ	WRITE
177XXXX0	DMA ADDRESS REGISTER	HARDWARE INITIALIZE
177XXXX2	STATUS REGISTER	COMMAND REGISTER (CSR)
177XXXX4	PARAMETER REGISTER	PARAMETER REGISTER
177XXXX6	(UNDEFINED)	(UNDEFINED)

ADDRESS 177XXXX0 IS THE DEVICE ADDRESS. THE HARDWARE P-TABLE CONSISTS OF TWO VALUES FOR EACH DEVICE. THE FIRST IS THE DEVICE ADDRESS , THE SECOND IS THE VECTOR ADDRESS.

THE DMA ADDRESS REGISTER CONTAINS THE LEAST SIGNIFICANT 16 BITS OF THE PHYSICAL ADDRESS OF THE LAST DMA OPERATION PERFORMED BY THE CONTROLLER

WHEN DMA HAS BEEN ENABLED.

THE HARDWARE INITIALIZE REGISTER , WHEN WRITTEN TO , CAUSES THE VSV21 TO BE RESET.

THE STATUS REGISTER CONTAINS STATUS INFORMATION ON THE OPERATIONAL STATUS OF THE CONTROLLER AND THE PROGRESS OF OPERATIONS WHICH ARE BEING PERFORMED BY THE CONTROLLER. THE REGISTER CONTAINS THE FOLLOWING STATUS BITS.

BIT	DESCRIPTION
12	CONTROLLER READY BIT : THIS BIT IS A LOGICAL 1 WHEN THE CONTROLLER IS READY TO ACCEPT A NEW COMMAND FROM THE HOST. A LOGICAL 0 IN THIS BIT INDICATES THAT THE CONTROLLER CANNOT ACCEPT A NEW COMMAND.
13	PARAMETER READY BIT : THIS BIT IS A LOGICAL 1 WHEN THE CONTROLLER HAS INFORMATION FOR THE HOST. IT IS A LOGICAL 0 WHEN THERE IS NO MORE INFORMATION.
14	CONTROLLER ERROR BIT : THIS BIT IS A LOGICAL 1 WHEN THE VSV21 BOARD HAS BEEN RESET. IT WILL BE CLEARED WHEN A COMMAND IS WRITTEN TO THE COMMAND REGISTER OR THE PARAMETER REGISTER IS READ (PROVIDING THE PARAMETER READY BIT IS SET). THIS BIT IS A LOGICAL 0 WHEN THE VSV21 BOARD HAS NOT JUST BEEN RESET.
15	STATUS VALID BIT : THIS BIT IS A LOGICAL 1 WHEN THE STATUS OF THE ABOVE DEFINED BITS IS VALID. A LOGICAL 0 INDICATES THAT THE STATUS OF THE ABOVE DEFINED BITS IS NOT VALID AND MAY BE UPDATED. THE HOST MUST WAIT UNTIL THIS BIT IS SET BEFORE READING THE STATUS OF THE BITS AND TAKING ANY SUBSEQUENT ACTION.

THE COMMAND REGISTER IS THE REGISTER FOR PASSING COMMANDS FROM THE HOST TO THE VSV21.

THE PARAMETER REGISTER IS A READ/WRITE REGISTER. IT IS WRITTEN TO WHEN THE HOST NEEDS TO SEND PARAMETERS PRIOR TO WRITING TO THE COMMAND REGISTER. IT IS READ BY THE HOST WHEN THE VSV21 HAS SET THE PARAMETER READY BIT IN THE STATUS REGISTER. THE HOST KEEPS READING THE PARAMETER REGISTER FOR MORE INFORMATION UNTIL THE PARAMETER READY BIT IS CLEAR.

6.0 TEST SUMMARIES

TEST 1: VERIFY VSV21 PRESENCE

This test verifies that there is a VSV21 device at the current CSR.
On the first pass it displays the version of the VSV21 and a message

warning the operator to attach external loopback connectors if he wishes tests 4 , 13 , 14 , 16 or 17 to be done. The test also disables interrupts.

TEST 2: IN DEPTH Q22 BUS TEST

This test performs an in-depth test of the Q22 bus by means of exercising the DMA mechanism in both directions. It sets up the DMA protocol and must be done before other tests requiring DMA.

TEST 3: FORCED ERROR TEST

This test sends an invalid command to the VSV21 to provoke an error packet.

TEST 4: FULL ON BOARD TESTS

This test invokes full on-board VSV21 microdiagnostics tests.
If the HOE flag is specified the VSV21 will halt on error.
If the LOE flag is specified the operator will be given the choice on the first pass of whether he wishes the on-board tests to loop on error. If he selects yes the test may hang indefinitely as the on board will not return control to the host but will continue looping.

TEST 5: ROM CHECKSUM TEST

This test invokes the on-board VSV21 ROM CHECKSUM test.
If the HOE flag is specified the VSV21 will halt on error.
If the LOE flag is specified the operator will be given the choice on the first pass of whether he wishes the on-board tests to loop on error. If he selects yes the test may hang indefinitely as the on board will not return control to the host but will continue looping.

TEST 6: NVRAM CHECKSUM TEST

This test invokes the on-board VSV21 NVRAM CHECKSUM test.
If the HOE flag is specified the VSV21 will halt on error.
If the LOE flag is specified the operator will be given the choice on the first pass of whether he wishes the on-board tests to loop on error. If he selects yes the test may hang indefinitely as the on board will not return control to the host but will continue looping.

TEST 7: RAM TEST

This test invokes the on-board VSV21 RAM test.
If the HOE flag is specified the VSV21 will halt on error.
If the LOE flag is specified the operator will be given the choice on the first pass of whether he wishes the on-board tests to loop on error. If he selects yes the test may hang indefinitely as the on board will not return control to the host but will continue looping.

TEST 8: RAM ADDRESSING TEST

This test invokes the on-board VSV21 RAM ADDRESSING test.
If the HOE flag is specified the VSV21 will halt on error.
If the LOE flag is specified the operator will be given the choice on the first pass of whether he wishes the on-board tests to loop on error.

If he selects yes the test may hang indefinitely as the on board will
not return control to the host but will continue loop'ng.

TEST 9: 68K PROCESSOR TEST

This test invokes the on-board VSV21 68K PROCESSOR test.
If tne HOE flag is specified the VSV21 will halt on error.
If the LOE flag is specified the operator will be given the choice on
the first pass of whether he wishes the on-board tests to loop on error.
If he selects yes the test may hang indefinitely as the on board will
not return control to the host but will continue looping.

TEST 10: INTERNAL EXCEPTIONS TEST

This test invokes the on-board INTERNAL EXCEPTIONS test.
If the HOE flag is specified the VSV21 will halt on error.
If the LOE flag is specified the operator will be given the choice on
the f'rst pass of whether he wishes the on-board tests to loop on error.
If he selects yes the test may hang indefinitely as the on board will
not return control to the host but will continue looping.

TEST 11: ACRCT INTERNAL TEST

This test invokes the on-board ACRCT INTERNAL test.
If the HOE flag is specified the VSV21 will halt on error.
If the LOE flag is specified the operator will be given the choice on
the first pass of whether he wishes the on-board tests to loop on error.
If he selects yes the test may hang indefinitely as the on board will
not return control to the host but will continue looping.

TEST 12: ACRCT EXTERNAL TEST

This test invokes the on-board VSV21 ACRCT EXTERNAL test.
If the HOE flag is specified the VSV21 will halt on error.
If the LOE flag is specified the operator will be given the choice on
the first pass of whether he wishes the on-board tests to loop on error.
If he selects yes the test may hang indefinitely as the on board will
not return control to the host but will continue looping.

TEST 13: DUART BASIC TEST

This test invokes the on-board VSV21 DUART BASIC test on ports 0-3.
If the HOE flag is specified the VSV21 will halt on error.
If the LOE flag is specified the operator will be given the choice on
the first pass of whether he wishes the on board tests to loop on error.
If he selects yes the test may hang indefin'tely as the on board w'll
not return control to the host but will continue looping.

TEST 14: DUART FULL TEST

This test invokes the on-board VSV21 DUART FULL test on ports 0-3.
If the HOE flag is specified the VSV21 will halt on error.
If the LOE flag is specified the operator will be given the choice on
the first pass of whether he wishes the on-board tests to loop on error.
If he selects yes the test may hang indefinitely as the on board will
not return control to the host but will continue loop'ng.

TEST 15: PERIPHERAL CONFIDENCE TEST

This test is not done if /FLA:UAM is specified on the START command.
This test allows the operator to select a port for either input or output.
If he selects output an ASCII string is written to that port.
If he selects input the first 16 characters read from that port are
displayed in octal on the console running the tests.
If the HOE flag is specified the VSV21 will halt on error.
If the LOE flag is specified the operator will be given the choice on
the first pass of whether he wishes the on-board tests to loop on error.
If he selects yes the test may hang indefinitely as the on board will
not return control to the host but will continue looping.

TEST 16: INTERNAL LOOPBACK TEST

This test causes the on-board software to perform an INTERNAL LOOPBACK
test on ports 0-3.
If the HOE flag is specified the VSV21 will halt on error.
If the LOE flag is specified the operator will be given the choice on
the first pass of whether he wishes the on-board tests to loop on error.
If he selects yes the test may hang indefinitely as the on board will
not return control to the host but will continue looping.

TEST 17: EXTERNAL I/O LOOPBACK TEST

This test causes the on-board software to perform an EXTERNAL LOOPBACK
test on ports 0-3.
If the HOE flag is specified the VSV21 will halt on error.
If the LOE flag is specified the operator will be given the choice on
the first pass of whether he wishes the on-board tests to loop on error.
If he selects yes the test may hang indefinitely as the on board will
not return control to the host but will continue looping.

TEST 18: SCREEN TEST

This test is not done if /FLA:UAM is specified on the START command.
This test displays different screen test pictures as selected by the
operator. If the HOE flag is specified the VSV21 will halt on error.
If the LOE flag is specified the operator will be given the choice on
the first pass of whether he wishes the on-board tests to loop on error.
If he selects yes the test may hang indefinitely as the on board will
not return control to the host but will continue looping.

TEST 19: NVRAM READ/WRITE TEST

This test is not done if /FLA:UAM is specified on the START command.
This test tests NVRAM read/write. On each pass the operator will be
asked if he wishes to continue as NVRAM has a limited life in terms of
read/write cycles.
If the HOE flag is specified the VSV21 will halt on error.
If the LOE flag is specified the operator will be given the choice on
the first pass of whether he wishes the on-board tests to loop on error.
If he selects yes the test may hang indefinitely as the on board will
not return control to the host but will continue looping.

TEST 20: ENABLE INTERRUPTS TEST

Th's test verif'es that interrupts are received by the host when interrupts are enabled.

&

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

                                .MCALL  SVC
000000                          SVC                             ; INITIALIZE SUPERVISOR MACROS

                        ;*******************************************************************************
                        ;       IF STRUCTURED MACROS ARE TO BE USED, ADD ".MCALL STRUCT" AND "STRUCT"
                        ;       TO INITIALIZE THE STRUCTURED MACROS.

                        SVCINS= -1      ; LIST INSTRUCTIONS, SHIFTED RIGHT
                        SVCTST= -1      ; LIST TEST TAGS, SHIFTED RIGHT
                        SVCSUB= -1      ; LIST SUBTEST TAGS, SHIFTED RIGHT
                        SVCGBL= -1      ; LIST GLOBAL TAGS, SHIFTED RIGHT
                        SVCTAG= -1      ; LIST OTHER TAGS, SHIFTED RIGHT

                        ;       CHANGE THE VALUES OF THE SVC... SYMBOLS TO BE ZERO IF YOU WISH
                        ;       TO ALIGN THE MACRO CALLS AND THEIR EXPANSIONS.  CHANGE THE
                        ;       SYMBOLS TO BE MINUS-ONE TO NOT LIST THE EXPANSIONS.  YOU MAY
                        ;       CHANGE THE SYMBOLS AT ANY POINT IN YOUR PROGRAM.
                        ;*******************************************************************************

000000                          .ENABL  ABS,AMA
002000                  .       =       2000

002000                          BGNMOD

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

002000                          POINTER ERRTBL,BGNSETUP

002000                          HEADER  CZVSWA0,A,0,15,0
```

DISPATCH TABLE

```
                                .SBTTL  DISPATCH TABLE

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

 000004                                 MAXUNIT== 4                     ; maximum number of units
 000024                                 MAXTST == 20.                   ; maximum number of tests

002122                                  DISPATCH MAXTST
```

```
                                .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.
                                ;--

002174                                  BGNHW   DFPTBL


002176  172010                         .WORD   172010
002200  000320                         .WORD   320

002202                                  ENDHW
```

DEFAULT HARDWARE P-TABLE

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


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

002202                                  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
                                ;
                                ;                                ;  BIT POSITION IN SECOND STATUS WORD
000040                          EF.START==      32.              ; (100000) START COMMAND WAS ISSUED
000037                          EF.RESTART==    31.              ; (040000) RESTART COMMAND WAS ISSUED
000036                          EF.CONTINUE==   30.              ; (020000) CONTINUE COMMAND WAS ISSUED
000035                          EF.NEW==        29.              ; (010000) A NEW PASS HAS BEEN STARTED
000034                          EF.PWR-=        28.              ; (004000) A POWER-FAIL/POWER-UP OCCURRED
                                ;
                                ;
                                ; PRIORITY LEVEL DEFINITIONS
                                ;
000340                          PRI07== 340
000300                          PRI06== 300
000240                          PRI05== 240
```

GLOBAL EQUATES SECTION SEQ 0021

```
                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
  965
  966                                   ;CSR bit meanings
  967
  968           100000                          VS.SVL  == BIT15        ; status valid
  969           040000                          VS.ERR  == BIT14        ; error ( powerup only )
  970           010000                          VS.CRY  == BIT12        ; controller ready
  971           020000                          VS.PRY  == BIT13        ; parameter ready
  972
  973
  974                                   ;Parallel interface commands ( these are written to the CSR by the host )
  975
  976           002400                          PI.CMA  == 5*400+0      ; command available
  977           003000                          PI.ACK  == 6*400+0      ; status ack
  978           001000                          PI.SEN  == 2*400+0      ; buffers sent
  979           001001                          PI.CAN  == 2*400+1      ; buffers cancelled
  980           000400                          PI.IDM  == 1*400+0      ; check ident. protocol.
  981           002000                          PI.TST  == 4*400+0      ; invoke microdiagnostics
  982           001400                          PI.IDS  == 3*400+0      ; disable interrupts
  983           001403                          PI.IEN  == PI.IDS+3     ; enable interrupts
  984           177777                          PI.RUB  == 177777       ; rubbish command to provoke error
  985
  986                                   ;This one is written to the parameter register by the on-board
  987
  988           100000                          PI.AVA  == BIT15        ; status available
  989
  990                                   ;These are DMA command codes that are put in the first word of the command
  991                                   ;buffer by the host
  992
  993           002101                          DI.RDA  == 2101         ; read data
  994           002102                          DI.WDD  == 2102         ; write dats.
  995           002103                          DI.RRA  == 2103         ; read ram protocol code.
  996           002104                          DI.WRA  == 2104         ; write ram protocol code.
  997
  998                                   ;return from the on-board
  999
```

SEQ 0022

GLOBAL EQUATES SECTION

```
1000    000011                  DI.MIC  == 9.           ; micro diagnostic result code
1001    000005                  DI.ERP  == 5.           ; error reply packet code
1002    000176                  DI.ERS  == 126.         ; error log and stop packet code
1003    000177                  DI.ERL  == 127.         ; error log packet code
1004
1005                    ;Fills in error table details
1006                            .MACRO  ERR.FILLIN      NUMBER,MESSAGE
1007                            MOV     #2,ERRTYP       ; assume hard error for now
1008                            MOV     #NUMBER,ERRNBR  ; get error number
1009                            MOV     #MESSAGE,ERRMSG ; get error message
1010                            MOV     #ERRCODE,ERRBLK ; get error routine
1011                            .ENDM
1012
1013                    ;Error return
1014                            .MACRO  ERR.RETURN
1015                            SEC
1016                            RETURN
1017                            .ENDM
1018
1019                    ;OK return
1020                            .MACRO  OK.RETURN
1021                            CLC
1022                            RETURN
1023                            .ENDM
1024
1025                    ;Put at start of each test.
1026                    ;Ensures that TEST1 must have been performed first.
1027                            .MACRO  ST.TEST,?LABEL
1028                            BGNTST
1029                            CALL    STTEST
1030                            BCC     LABEL
1031                            EXIT    TST
1032                    LABEL:
1033                            .ENDM
1034
1035                    ;Sets up POLCNT for a delay
1036                            .MACRO  SET.POLCNT
1037                            MOV     #1500.,POLCNT
1038                            .ENDM
1039
1040                    ;Writes a command to the CSR
1041                            .MACRO  DO.COMMAND      COM,LABEL
1042                            MOV     #COM,COMAND
1043                            SET.POLCNT
1044                            CALL    DOCOM
1045                            BCC     LABEL
1046                            .ENDM
1047
1048                    ;Tests manual intervention and exits test if not allowed
1049                            .MACRO  MAN.IGNORE      ?LABEL
1050                            MANUAL
1051                            BCOMPLETE       LABEL
1052                            EXIT    TST
1053                    LABEL:
1054                            .ENDM
1055
1056                    ;Tests dma done and exits test if not
```

GLOBAL EQUATES SECTION

```
1057                                    .MACRO  DMA.IGNORE      ?LABEL
1058                                    TST     DMASET(R4)              ; DMA enabled ?
1059                                    BNE     LABEL                   ; branch if yes
1060                                    ERRHRD  2,ERR2,ERRCODE          ; error exit if not
1061                                    EXIT    TST
1062                                LABEL:
1063                                    .ENDM
1064
1065                                ;Calls writeram routine
1066                                    .MACRO  DO.WRAMS        BUFADD,BUFLEN
1067                                    .IF     NB      BUFADD
1068                                    .IFT
1069                                    MOV     BUFADD,COMBUF+4
1070                                    MOV     BUFLEN,COMBUF+6
1071                                    .IFF
1072                                    MOV     #DMAOBUF,COMBUF+4
1073                                    MOV     #DMALEN,COMBUF+6
1074                                    .IFTF
1075                                    CALL    WRAMS
1076                                    .ENDC
1077                                    .ENDM
1078
1079                                ;Calls readram routine
1080                                    .MACRO  DO.RRAMS        BUFADD,BUFLEN
1081                                    .IF     NB      BUFADD
1082                                    .IFT
1083                                    MOV     BUFADD,COMBUF+4
1084                                    MOV     BUFLEN,COMBUF+6
1085                                    .IFF
1086                                    MOV     #DMAIBUF,COMBUF+4
1087                                    MOV     #DMALEN,COMBUF+6
1088                                    .IFTF
1089                                    CALL    RRAMS
1090                                    .ENDC
1091                                    .ENDM
```

GLOBAL DATA SECTION

```
                                .SBTTL  GLOBAL DATA SECTION

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

                                        ERRTBL
002202  000000                  ERRTYP::      .WORD   0
002204  000000                  ERRNBR::      .WORD   0
002206  000000                  ERRMSG::      .WORD   0
002210  000000                  ERRBLK::      .WORD   0

002212  000000                  TEMP1:: .WORD   0               ; temporary storage
002214  000000                  TEMP2:: .WORD   0               ; temporary storage
002216  000000                  POLCNT::.WORD   0               ; <0 = poll CSR forever , >0 = poll CSR
                                ;                               ; required number
002220  000000                  LOGUNT::.WORD   0               ; current unit number
002222                          PBLOC:: .BLKW   4               ; storage for sending microdiagnostic parameters
002232  000000                  CURCSR::.WORD   0               ; CSR address for current unit
002234  000000                  CURPAR::.WORD   0               ; parameter register address for current unit
002236  000000                  CURVEC::.WORD   0               ; vector address for current unit
002240  000000                  INTFLG::.WORD   0               ; <>0 = got interrupt , 0 = not got interrupt
002242  000000                  NOTEST::.WORD   0               ; 0 = tests can be done , <>0 = tests cannot
002244  000001  000002  000004  BITS::   .WORD   BIT0,BIT1,BIT2,BIT3,BIT4,BIT5,BIT6,BIT7,BIT8
002252  000010  000020  000040
002260  000100  000200  000400
002266  001000  002000  004000          .WORD   BIT9,BIT10,BIT11,BIT12,BIT13,BIT14,BIT15
002274  010000  020000  040000
002302  100000

                                ;Flags per unit (globals are offset from start of flags for that unit)
                                ;*************************************************************************
                                ;               N.B. R4 POINTS TO THE FLAGS FOR THE CURRENT UNIT
                                ;*************************************************************************

000000                                  T1DONE  == 0            ; 1 WORD: 0 = TEST1 not done , <>0 = TEST1 DONE
000002                                  DMASET  == T1DONE+2     ; 1 WORD: 0 = DMA disabled , <>0 = DMA enabled
000004                                  FTHRU   == DMASET+2     ; 2 WORDS: bit 'n' is clear if TEST 'n' not yet
                                ;                               ; done
000010                                  TMASKS  == FTHRU+4      ; MAXTST WORDS : test masks
                                ;                               ; (only applicable to Microdiagnostic tests)
000030                                  LENFLAGS == TMASKS/2+MAXTST     ; length of flags per unit in words
002304                          FLAGS:: .BLKW   <LENFLAGS*MAXUNIT>      ; the flags indexed by unit

                                ;Command and status buffers

000024                                  COMLEN  = 24
000036                                  STALEN  = 36
002604  012000  002616  017000  BUFDES: .WORD   <COMLEN*400>,COMBUF,<STALEN*400>,STABUF ; buffer descriptors
002612  002666
002614                          COMAND: .BLKW   1               ; current command
002616                          COMBUF: .BLKW   COMLEN          ; command packet buffer for DMA
002666                          STABUF: .BLKW   STALEN          ; status packet buffer for DMA.

                                ; the data to be written to RAM
```

SEQ 0025

GLOBAL DATA SECTION

```
   1141 002762                          DMAOBUF::
   1142 002762  177776                           .WORD   177776
   1143 002764  177775                           .WORD   177775
   1144 002766  177774                           .WORD   177774
   1145 002770  177773                           .WORD   177773
   1146 002772  177772                           .WORD   177772
   1147 002774  177771                           .WORD   177771
   1148 002776  177770                           .WORD   177770
   1149 003000  177760                           .WORD   177760
   1150 003002  177750                           .WORD   177750
   1151 003004  177740                           .WORD   177740
   1152 003006  177730                           .WORD   177730
   1153 003010  177720                           .WORD   177720
   1154 003012  177710                           .WORD   177710
   1155 003014  177700                           .WORD   177700
   1156 003016  177600                           .WORD   177600
   1157 003020  177500                           .WORD   177500
   1158 003022  177400                           .WORD   177400
   1159 003024  177300                           .WORD   177300
   1160 003026  177200                           .WORD   177200
   1161 003030  177100                           .WORD   177100
   1162 003032  177000                           .WORD   177000
   1163 003034  000000                           .WORD   0
   1164
   1165         000054          DMALEN = .-DMAOBUF      ; length of above data
   1166 003036                  DMAIBUF::
   1167 003036                           .BLKW   DMALEN  ; buffer to read back into .
   1168
   1169
```

GLOBAL TEXT SECTION

```
                                .SBTTL  GLOBAL TEXT SECTION

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

003166  124  105  123  ERR1::  .ASCIZ  /TEST 1 MUST BE PERFORMED FIRST/
003225  124  105  123  ERR2::  .ASCIZ  /TEST 2 MUST BE RUN FIRST TO ENABLE DMA/
003274  124  111  115  ERR3::  .ASCIZ  /TIMED OUT WHILE WAITING FOR STATUS VALID/
003345  124  110  105  ERR4::  .ASCIZ  /THERE HAS BEEN A POWERUP - PLEASE REPEAT TEST 1/
003425  103  117  116  ERR5::  .ASCIZ  /CONTROLLER READY NOT SET IN CSR/
003465  123  124  101  ERR6::  .ASCIZ  /STATUS PACKET SHORTER THAN SPECIFIED LENGTH/
003541  123  124  101  ERR7::  .ASCIZ  /STATUS AVAILABLE EXPECTED BUT NOT FOUND/
003611  105  122  122  ERR8::  .ASCIZ  /ERROR PACKET RECEIVED FROM VSV21/
003652  106  101  111  ERR9::  .ASCIZ  /FAILED TO TRANSMIT STATUS ACKNOWLEDGE/
003720  124  111  115  ERR10:: .ASCIZ  /TIMEOUT WAITING FOR STATUS/
003753  104  115  101  ERR11:: .ASCIZ  /DMA DATA MATCH ERROR/
004000  106  101  111  ERR12:: .ASCIZ  /FAILED TO GET MICRO DIAGNOSTICS RETURN STATUS PACKET/
004065  115  111  103  ERR13:: .ASCIZ  /MICRO DIAGNOSTIC TEST UNSUCCESSFUL/
004130  105  122  122  ERR14:: .ASCIZ  /ERROR PACKET EXPECTED BUT NOT RECEIVED/
004177  124  111  115  ERR15:: .ASCIZ  /TIMED OUT WAITING TO RECEIVE AN INTERRUPT/

                                ;
                                ; NAMES OF DEVICES SUPPORTED BY PROGRAM
                                ;
004251                                  DEVTYP  <VSV21>
                                ; TEST DESCRIPTION
                                ;
004260                                  DESCRIPT        <VSV21 DIAGNOSTIC>
                                        .EVEN

                                ;
                                ; FORMAT STATEMENTS USED IN PRINT CALLS
                                ;
```

GLOBAL ERROR REPORT SECTION

```
                                .SBTTL  GLOBAL ERROR REPORT SECTION

                                ;++
                                ; THE GLOBAL ERROR REPORT SECTION CONTAINS MESSAGE PRINTING AREAS
                                ; USED BY MORE THAN TEST TO OUTPUT ADDITIONAL ERROR INFORMATION.  PRINTB
                                ; (BASIC) AND PRINTX (EXTENDED) CALLS ARE USED TO CALL PRINT SERVICES.
                                ;--

004302                                  BGNMSG  ERRCODE

004302  005237  002242                 INC     NOTEST                  ;say no further tests allowed

004306                                  EXIT    MSG

                                        .EVEN

004312                                  ENDMSG
```

```
                                .SBTTL  GLOBAL SUBROUTINES SECTION

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

                                ;++
                                ; FUNCTIONAL DESCRIPTION:
                                ;    SUBROUTINE TO POLL THE CSR  OR STATUS VALID BIT TO BE SET.

                                ; INPUTS:
                                ;       POLCNT              NUMBER OF TIMES TO POLL , -1 = INFINITE

                                ; IMPLICIT INPUTS:
                                ;       CURCSR              - CSR ADDRESS TO POLL
                                ;       T1DONE              - TEST 1 DONE FLAG

                                ; OUTPUTS:
                                ;       CONDITION CODE      - CLEAR = SUCCESS
                                ;                           - SET   = FAILURE

                                ; IMPLICIT OUTPUTS:
                                ;       ERROR BLOCK         - FILLED IN ON FAILURE
                                ;       T1DONE              - CLEARED IF POWERUP
                                ;       DMASET              - CLEARED IF POWERUP

                                ; SUBORDINATE ROUTINES USED:
                                ;       NONE

                                ; FUNCTIONAL SIDE EFFECTS:
                                ;       IF POWERUP OF VSV21 FOUND FORCES TEST1 TO BE DONE AGAIN BY CLEARING
                                ;       T1DONE. ALSO CLEARS DMASET TO SAY THAT DMA IS NOT ENABLED.

                                ; CALLING SEQUENCE:
                                ;       MOV     #1500.,POLCNT   ;SET UP POLL COUNT
                                ;       CALL    POLL            ;GO TO ROUTINE
                                ;       BCS     ERROR           ;CARRY SET IF ROUTINE HAD ERROR
                                ;--
004314  032777  100000  175710  POLL::  BIT     #VS.SVL,@CURCSR         ; SVL ?
004322  001040                         BNE     2$                      ; branch if yes
004324                                  DELAY   10.                     ; delay 10 units
004354  005737  002216                 TST     POLCNT                  ; poll forever ?
004360  002755                         BLT     POLL                    ; yes
004362  005337  002216                 DEC     POLCNT                  ; no   reduce count
004366  003352                         BGT     POLL                    ; try again if still some left

004370                                  ERR.FILLIN      3,ERR3
004370  012737  000002  002202         MOV     #2,ERRTYP       ; assume hard error for now
004376  012737  000003  002204         MOV     #3,ERRNBR       ; get error 3
004404  012737  003274  002206         MOV     #ERR3,ERRMSG    ; get error ERR3
004412  012737  004302  002210         MOV     #ERRCODE,ERRBLK ; get error routine
004420                                  ERR.RETURN
004420  000261                         SEC
004422  000207                         RETURN

004424  032777  040000  175600  2$:     BIT     #VS.ERR,@CURCSR         ; ERR ?
```

GLOBAL SUBROUTINES SECTION

```
1282 004432  001425                        BEQ     3$                      ; branch if not
1283
1284                                ; powerup - initialise
1285 004434  005064  000002                CLR     DMASET(R4)              ; say DMA disabled
1286 004440  005764  000000                TST     T1DONE(R4)              ; Test 1 done yet ?
1287 004444  001420                        BEQ     3$                      ; branch if not
1288 004446  005064  000000                CLR     T1DONE(R4)              ; say Test 1 not done
1289 004452                                ERR.FILLIN      4,ERR4
     004452  012737  000002  002202        MOV     #2,ERRTYP       ; assume hard error for now
     004460  012737  000004  002204        MOV     #4,ERRSUB       ; get error 4
     004466  012737  003345  002206        MOV     #ERR4,ERRMSG    ; get error ERR4
     004474  012737  004302  002210        MOV     #ERRCODE,ERRBLK ; get error routine
1290 004502                                ERR.RETURN
     004502  000261                        SEC
     004504  000207                        RETURN
1291
1292 004506                         3$:    OK.RETURN
     004506  000241                        CLC
     004510  000207                        RETURN
1293
```

GLOBAL SUBROUTINES SECTION

```
                                ;++
                                ; FUNCTIONAL DESCRIPTION:
                                ;       SUBROUTINE TO POLL THE CSR FOR STATUS VALID AND CONTROLLER READY BITS
                                ;       TO BE SET.
                                ;
                                ; INPUTS:
                                ;       NONE
                                ;
                                ; IMPLICIT INPUTS:
                                ;       CURCSR          - CSR ADDRESS TO POLL
                                ;
                                ; OUTPUTS:
                                ;       CONDITION CODE  - CLEAR = SUCCESS
                                ;                       - SET   = FAILURE
                                ;
                                ; IMPLICIT OUTPUTS:
                                ;       ERROR BLOCK     - FILLED IN ON FAILURE
                                ;
                                ; SUBORDINATE ROUTINES USED:
                                ;       POLL
                                ;
                                ; FUNCTIONAL SIDE EFFECTS:
                                ;       NONE
                                ;
                                ; CALLING SEQUENCE:
                                ;       CALL    POLLCRY         ;GO TO ROUTINE
                                ;       BCS     ERROR           ;CARRY SET IF ROUTINE HAD ERROR
                                ;--
004512                          POLLCRY::
004512  004737  004314                  CALL    POLL                    ; poll for SVL
004516  103001                          BCC     10$                     ; branch if SVL
004520  000207                          RETURN                          ; else return if not
004522  032777  010000  175502  10$:    BIT     #VS.CRY,@CURCSR         ; CRY ?
004530  001016                          BNE     20$                     ; branch if yes
004532                                  ERR.FILLIN      5,ERR5          ; error return if not
004532  012737  000002  002202          MOV     #2,ERRTYP       ; assume hard error for now
004540  012737  000005  002204          MOV     #5,ERRNBR       ; get error 5
004546  012737  003425  002206          MOV     #ERR5,ERRMSG    ; get error ERR5
004554  012737  004302  002210          MOV     #ERRCODE,ERRBLK ; get error routine
004562                                  ERR.RETURN
004562  000261                          SEC
004564  000207                          RETURN
004566                          20$:    OK.RETURN                       ; success return
004566  000241                          CLC
004570  000207                          RETURN
```

GLOBAL SUBROUTINES SECTION

```
                                ;++
                                ; FUNCTIONAL DESCRIPTION:
                                ;    SUBROUTINE TO WRITE A COMMAND TO THE CURRENT CSR.
                                ;
                                ; INPUTS:
                                ;       POLCNT          - NUMBER OF TIMES TO POLL , -1 = INFINITE
                                ;
                                ; IMPLICIT INPUTS:
                                ;       CURCSR          - CSR ADDRESS TO POLL
                                ;
                                ; OUTPUTS:
                                ;       CONDITION CODE  - CLEAR = SUCCESS
                                ;                       - SET   = FAILURE
                                ;
                                ; IMPLICIT OUTPUTS:
                                ;       ERROR BLOCK     - FILLED IN ON FAILURE
                                ;
                                ; SUBORDINATE ROUTINES USED:
                                ;       NONE
                                ;
                                ; FUNCTIONAL SIDE EFFECTS:
                                ;       THIS DEPENDS ON THE COMMAND. E.G. IF THE COMMAND IS 'BUFFERS SENT'
                                ;       DMA WILL THEN BE ENABLED AND STATUS WILL COME BACK VIA DMA.
                                ;
                                ; CALLING SEQUENCE:
                                ;       MOV     #1500.,POLCNT   ;SET UP POLL COUNT
                                ;       CALL    DOCOM           ;GO TO ROUTINE
                                ;       BCS     ERROR           ;CARRY SET IF ROUTINE HAD ERROR
                                ;--
                                
004572  004737  004314          DOCOM:: CALL POLL                       ; ensure SVL
004576  103003                          BCC     10$                     ; branch if OK
004600                                  ERROR
004602                                  ERR.RETURN
004602  000261                          SEC
004604  000207                          RETURN
                                
004606  032777  020000  175416  10$:    BIT     #VS.PRY,@CURCSR         ; PRY ?
004614  001404                          BEQ     20$                     ; branch if not
004616  004737  005066                  CALL    GETST1                  ; get status
004622  103001                          BCC     20$                     ; branch if status OK
004624  000207                          RETURN                          ; return if not
                                
004626  032777  010000  175376  20$:    BIT     #VS.CRY,@CURCSR         ; CRY ?
004634  001017                          BNE     30$                     ; branch if yes
004636                                  ERR.FILLIN      5,ERR5
004636  012737  000002  002202          MOV     #2,ERRTYP       ; assume hard error for now
004644  012737  000005  002204          MOV     #5,ERRNBR       ; get error 5
004652  012737  003425  002206          MOV     #ERR5,ERRMSG    ; get error ERR5
004660  012737  004302  002210          MOV     #ERRCODE,ERRBLK ; get error routine
004666                                  ERROR
004670                                  ERR.RETURN
004670  000261                          SEC
004672  000207                          RETURN
                                
004674  013777  002614  175330  30$:    MOV     COMAND,@CURCSR          ; write comand to the CSR
```

GLOBAL SUBROUTINES SECTION

```
1383 004702  004737  004314          35$:    CALL    POLL                    ; wait for SVL
1384 004706  103003                          BCC     40$
1385 004710                                  ERROR
1386 004712                                  ERR.RETURN
     004712  000261                          SEC
     004714  000207                          RETURN
1387
1388 004716  032777  020000  175306  40$:    BIT     #VS.PRY,@CURCSR         ; PRY ?
1389 004724  001055                          BNE     50$                     ; branch if yes
1390
1391                                 ;interrupt mask and buffer sent commands do not expect status from the VSV21
1392
1393 004726  022737  001400  002614          CMP     #PI.IDS,COMAND          ; interrupt disable command ?
1394 004734  001453                          BEQ     60$                     ; branch if yes
1395 004736  022737  001403  002614          CMP     #PI.IEN,COMAND          ; interrupt enable command ?
1396 004744  001447                          BEQ     60$                     ; branch if yes
1397 004746  022737  001000  002614          CMP     #PI.SEN,COMAND          ; buf sent command ?
1398 004754  001443                          BEQ     60$                     ; branch if yes
1399
1400                                 ;try again to get status
1401 004756                                  DELAY   10.                     ; delay 10 units
1402 005006  005737  002216                  TST     POLCNT                  ; poll forever ?
1403 005012  002733                          BLT     35$                     ; yes
1404 005014  005337  002216                  DEC     POLCNT                  ; no   reduce count
1405 005020  003330                          BGT     35$                     ; try again if still some left
1406 005022                                  ERR.FILLIN      10.,ERR10       ; timeout
     005022  012737  000002  002202          MOV     #2,ERRTYP       ; assume hard error for now
     005030  012737  000012  002204          MOV     #10.,ERRNBR     ; get error 10.
     005036  012737  003720  002206          MOV     #ERR10,ERRMSG   ; get error ERR10
     005044  012737  004302  002210          MOV     #ERRCODE,ERRBLK ; get error routine
1407 005052                                  ERROR
1408 005054                                  ERR.RETURN
     005054  000261                          SEC
     005056  000207                          RETURN
1409
1410 005060  004737  005462          50$:    CALL    GETST2                  ; get status
1411 005064  000207                  60$:    RETURN
1412
```

```
                        ;++
                        ; FUNCTIONAL DESCRIPTION:
                        ;    SUBROUTINE TO GET A STATUS PACKET FROM THE VSV21 INTO STATUS BUFFER.
                        ;    THIS INCLUDES BOTH PROGRAMMED I/O AND DMA MODES.
                        ;    THIS MODULE IS VERY SIMILAR TO GETST2 BUT IS CALLED BEFORE A COMMAND IS
                        ;    SENT TO THE VSV21.                                          ----
                        ;
                        ; INPUTS:
                        ;       NONE
                        ;
                        ; IMPLICIT INPUTS:
                        ;       DMASET          - DMA ENABLED FLAG
                        ;       STABUF          - STATUS BUFFER
                        ;       CURCSR          - CSR ADDRESS
                        ;
                        ; OUTPUTS:
                        ;       CONDITION CODE  - CLEAR = SUCCESS
                        ;                       - SET   = FAILURE
                        ;
                        ; IMPLICIT OUTPUTS:
                        ;       ERROR BLOCK     - FILLED IN ON FAILURE
                        ;       STABUF          - STATUS BUFFER
                        ;
                        ; SUBORDINATE ROUTINES USED:
                        ;       POLL
                        ;       POLLCRY
                        ;
                        ; FUNCTIONAL SIDE EFFECTS:
                        ;       NONE
                        ;
                        ; CALLING SEQUENCE:
                        ;       CALL    GETST1          ;GO TO ROUTINE
                        ;       BCS     ERROR           ;CARRY SET IF ROUTINE HAD ERROR
                        ;--
                        ;
005066                  GETST1::
005066  005764  000002          TST     DMASET(R4)              ; DMA enabled ?
005072  001060                  BNE     30$                     ; branch if yes
                        ;
                        ;DMA not enabled - get status packet via parameter register
                        ;
                        ;       special processing if ERR set - only one word in status packet
005074  032777  040000  175130  BIT     #VS.ERR,@CURCSR         ; ERR ?
005102  001407                  BEQ     5$                      ; branch if no
005104  012737  000001  002666  MOV     #1,STABUF               ; pretend status packet has a header
005112  017737  175116  002670  MOV     @CURPAR,STABUF+2        ; get rest of packet
005120  000444                  BR      25$
                        ;
005122  017737  175106  002666  5$:     MOV     @CURPAR,STABUF          ; get status header word
005130  113701  002666          MOVB    STABUF,R1               ; get number of following status words
005134  001436                  BEQ     25$                     ; branch if none
005136  012702  002670          MOV     #<STABUF+2>,R2          ; get address of where to put them
005142  004737  004314  10$:     CALL    POLL                    ; poll
005146  103003                  BCC     20$
005150                          ERROR
005152                          ERR.RETURN
005152  000261                  SEC
```

```
GLOBAL SUBROUTINES SECTION

      005154  000207                          RETURN
1470  005156  032777  020000  175046  20$:    BIT     #VS.PRY,@CURCSR         ; PRY ?
1471  005164  001017                          BNE     22$                     ; branch if yes
1472  005166                                  ERR.FILLIN      6,ERR6
      005166  012737  000002  002202          MOV     #2,ERRTYP       ; assume hard error for now
      005174  012737  000006  002204          MOV     #6,ERRNBR       ; get error 6
      005202  012737  003465  002206          MOV     #ERR6,ERRMSG    ; get error ERR6
      005210  012737  004302  002210          MOV     #ERRCODE,ERRBLK ; get error routine
1473  005216                                  ERROR
1474  005220                                  ERR.RETURN
      005220  000261                          SEC
      005222  000207                          RETURN
1475  005224  017722  175004          22$:    MOV     @CURPAR,(R2)+           ; get a status word
1476  005230  077134                          SOB     R1,10$                  ; again
1477  005232  000434                  25$:    BR      60$
1478
1479                                  ;DMA enabled - status packet is written straight to status buffer
1480
1481  005234  022777  100000  174772  30$:    CMP     #PI.AVA,@CURPAR ; is there a packet available ?
1482  005242  001417                          BEQ     40$                     ; branch if yes
1483  005244                                  ERR.FILLIN      7,ERR7
      005244  012737  000002  002202          MOV     #2,ERRTYP       ; assume hard error for now
      005252  012737  000007  002204          MOV     #7,ERRNBR       ; get error 7
      005260  012737  003541  002206          MOV     #ERR7,ERRMSG    ; get error ERR7
      005266  012737  004302  002210          MOV     #ERRCODE,ERRBLK ; get error routine
1484  005274                                  ERROR
1485  005276                                  ERR.RETURN
      005276  000261                          SEC
      005300  000207                          RETURN
1486
1487  005302  004737  004512          40$:    CALL    POLLCRY                 ; poll for CRY
1488  005306  103003                          BCC     50$                     ; branch if OK
1489  005310                                  ERROR
1490  005312                                  ERR.RETURN                      ; return if error
      005312  000261                          SEC
      005314  000207                          RETURN
1491  005316  012777  003000  174706  50$:    MOV     #PI.ACK,@CURCSR ; send status ack
1492
1493  005324  004737  004512          60$:    CALL    POLLCRY                 ; poll for CRY
1494  005330  103017                          BCC     70$                     ; branch if OK
1495  005332                                  ERR.FILLIN      9.,ERR9         ; error
      005332  012737  000002  002202          MOV     #2,ERRTYP       ; assume hard error for now
      005340  012737  000011  002204          MOV     #9.,ERRNBR      ; get error 9.
      005346  012737  003652  002206          MOV     #ERR9,ERRMSG    ; get error ERR9
      005354  012737  004302  002210          MOV     #ERRCODE,ERRBLK ; get error routine
1496  005362                                  ERROR
1497  005364                                  ERR.RETURN
      005364  000261                          SEC
      005366  000207                          RETURN
1498
1499  005370  122737  000005  002667  70$:    CMPB    #DI.ERP,STABUF+1        ; error status packet ?
1500  005376  001410                          BEQ     75$                     ; branch if yes
1501  005400  122737  000176  002667          CMPB    #DI.ERS,STABUF+1        ; error status packet ?
1502  005406  001404                          BEQ     75$                     ; branch if yes
1503  005410  122737  000177  002667          CMPB    #DI.ERL,STABUF+1        ; error status packet ?
1504  005416  001017                          BNE     80$                     ; branch if not
1505  005420                          75$:    ERR.FILLIN      8.,ERR8
```

```
GLOBAL SUBROUTINES SECTION

        005420  012737  000002  002202          MOV     #2,ERRTYP       ; assume hard error for now
        005426  012737  000010  002204          MOV     #8.,ERRNBR      ; get error 8.
        005434  012737  003611  002206          MOV     #ERR8,ERRMSG    ; get error ERR8
        005442  012737  004302  002210          MOV     #ERRCODE,ERRBLK ; get error routine
   1506 005450                                  ERROR
   1507 005452                                  ERR.RETURN
        005452  000261                          SEC
        005454  000207                          RETURN
   1508
   1509 005456                          80$:    OK.RETURN
        005456  000241                          CLC
        005460  000207                          RETURN
   1510
```

GLOBAL SUBROUTINES SECTION

```
                                ;**
                                ; FUNCTIONAL DESCRIPTION:
                                ;    SUBROUTINE TO GET A STATUS PACKET FROM THE VSV21 INTO STATUS BUFFER.
                                ;    THIS INCLUDES BOTH PROGRAMMED I/O AND DMA MODES.
                                ;    THIS MODULE IS VERY SIMILAR TO GETST1 BUT IS CALLED AFTER A COMMAND IS SENT
                                ;    TO THE VSV21.                                          ----
                                ;
                                ; INPUTS:
                                ;       NONE
                                ;
                                ; IMPLICIT INPUTS:
                                ;       DMASET          - DMA ENABLED FLAG
                                ;       STABUF          - STATUS BUFFER
                                ;       CURCSR          - CSR ADDRESS
                                ;       L$TEST          - CURRENT TEST
                                ;
                                ; OUTPUTS:
                                ;       CONDITION CODE  - CLEAR = SUCCESS
                                ;                       - SET   = FAILURE
                                ;
                                ; IMPLICIT OUTPUTS:
                                ;       ERROR BLOCK     - FILLED IN ON FAILURE
                                ;       STABUF          - STATUS BUFFER
                                ;
                                ; SUBORDINATE ROUTINES USED:
                                ;       POLL
                                ;       POLLCRY
                                ;
                                ; FUNCTIONAL SIDE EFFECTS:
                                ;       NONE
                                ;
                                ; CALLING SEQUENCE:
                                ;       CALL    GETST2          ;GO TO ROUTINE
                                ;       BCS     ERROR           ;CARRY SET IF ROUTINE HAD ERROR
                                ;--
                                ;
005462                          GETST2::
005462  005764  000002                  TST     DMASET(R4)              ; DMA enabled ?
005466  001060                          BNE     30$                     ; branch if yes
                                ;
                                ;DMA not enabled - get status packet via parameter register
                                ;
                                ;       special processing if ERR set - only one word in status packet
005470  032777  040000  174534          BIT     #VS.ERR,@CURCSR         ; ERR ?
005476  001407                          BEQ     5$                      ; branch if no
005500  012737  000001  002666          MOV     #1,STABUF               ; pretend status packet has a header
005506  017737  174522  002670          MOV     @CURPAR,STABUF+2        ; get rest of packet
005514  000444                          BR      25$
                                ;
005516  017737  174512  002666  5$:     MOV     @CURPAR,STABUF          ; get status header word
005524  113701  002666                  MOVB    STABUF,R1               ; get number of following status words
005530  001436                          BEQ     25$                     ; branch if none
005532  012702  002670                  MOV     #<STABUF+2>,R2          ; get address of where to put them
005536  004737  004314          10$:    CALL    POLL                    ; poll
005542  103003                          BCC     20$
005544                                  ERROR
005546                                  ERR.RETURN
```

SEQ 0037

GLOBAL SUBROUTINES SECTION

```
                005546  000261                          SEC
                005550  000207                          RETURN
     1569       005552  032777  020000  174452  20$:    BIT     #VS.PRY,@CURCSR         ; PRY ?
     1570       005560  001017                          BNE     22$                     ; branch if yes
     1571       005562                                  ERR.FILLIN      6,ERR6
                005562  012737  000002  002202          MOV     #2,ERRTYP       ; assume hard error for now
                005570  012737  000006  002204          MOV     #6,ERRNBR       ; get error 6
                005576  012737  003465  002206          MOV     #ERR6,ERRMSG    ; get error ERR6
                005604  012737  004302  002210          MOV     #ERRCODE,ERRBLK ; get error routine
     1572       005612                                  ERROR
     1573       005614                                  ERR.RETURN
                005614  000261                          SEC
                005616  000207                          RETURN
     1574       005620  017722  174410          22$:    MOV     @CURPAR,(R2)+           ; get a status word
     1575       005624  077134                          SOB     R1,10$                  ; again
     1576       005626  000434                  25$:    BR      60$
     1577
     1578                                       ;DMA enabled - status packet is written straight to status buffer
     1579
     1580       005630  022777  100000  174376  30$:    CMP     #PI.AVA,@CURPAR ; is there a packet available ?
     1581       005636  001417                          BEQ     40$                     ; branch if yes
     1582       005640                                  ERR.FILLIN      7,ERR7
                005640  012737  000002  002202          MOV     #2,ERRTYP       ; assume hard error for now
                005646  012737  000007  002204          MOV     #7,ERRNBR       ; get error 7
                005654  012737  003541  002206          MOV     #ERR7,ERRMSG    ; get error ERR7
                005662  012737  004302  002210          MOV     #ERRCODE,ERRBLK ; get error routine
     1583       005670                                  ERROR
     1584       005672                                  ERR.RETURN
                005672  000261                          SEC
                005674  000207                          RETURN
     1585
     1586       005676  004737  004512          40$:    CALL    POLLCRY                 ; poll for CRY
     1587       005702  103003                          BCC     50$                     ; branch if OK
     1588       005704                                  ERROR
     1589       005706                                  ERR.RETURN                      ; return if error
                005706  000261                          SEC
                005710  000207                          RETURN
     1590       005712  012777  003000  174312  50$:    MOV     #PI.ACK,@CURCSR ; send status ack
     1591
     1592       005720  004737  004512          60$:    CALL    POLLCRY                 ; poll for CRY
     1593       005724  103017                          BCC     70$                     ; branch if OK
     1594       005726                                  ERR.FILLIN      9.,ERR9         ; error
                005726  012737  000002  002202          MOV     #2,ERRTYP       ; assume hard error for now
                005734  012737  000011  002204          MOV     #9.,ERRNBR      ; get error 9.
                005742  012737  003652  002206          MOV     #ERR9,ERRMSG    ; get error ERR9
                005750  012737  004302  002210          MOV     #ERRCODE,ERRBLK ; get error routine
     1595       005756                                  ERROR
     1596       005760                                  ERR.RETURN
                005760  000261                          SEC
                005762  000207                          RETURN
     1597
     1598       005764  122737  000005  002667  70$:    CMPB    #DI.ERP,STABUF+1        ; error status packet ?
     1599       005772  001414                          BEQ     75$                     ; branch if yes
     1600       005774  122737  000176  002667          CMPB    #DI.ERS,STABUF+1        ; error status packet ?
     1601       006002  001410                          BEQ     75$                     ; branch if yes
     1602       006004  022737  000003  002114          CMP     #3,L$TEST               ; test 3 (error expected) ?
     1603       006012  001423                          BEQ     80$                     ; branch if yes
```

GLOBAL SUBROUTINES SECTION

```
1604 006014  122737  000177  002667          CMPB    #DI.ERL,STABUF+1        ; error status packet ?
1605 006022  001017                          BNE     80$                     ; branch if not
1606 006024                          75$:    ERR.FILLIN      8.,ERR8
     006024  012737  000002  002202          MOV     #2,ERRTYP       ; assume hard error for now
     006032  012737  000010  002204          MOV     #8.,ERRNBR      ; get error 8.
     006040  012737  003611  002206          MOV     #ERR8,ERRMSG    ; get error ERR8
     006046  012737  004302  002210          MOV     #ERRCODE,ERRBLK ; get error routine
1607 006054                                  ERROR
1608 006056                                  ERR.RETURN
     006056  000261                          SEC
     006060  000207                          RETURN
1609
1610 006062                          80$:    OK.RETURN
     006062  000241                          CLC
     006064  000207                          RETURN
1611
```

GLOBAL SUBROUTINES SECTION

```
                                ;++
                                ; FUNCTIONAL DESCRIPTION:
                                ;    SUBROUTINE TO WRITE DATA USING DMA.
                                ;
                                ; INPUTS:
                                ;       NONE
                                ;
                                ; IMPLICIT INPUTS:
                                ;       COMBUF          - COMMAND BUFFER CONTAINING :
                                ;                                         DMA BUFFER ADDRESS IN COMBUF+4
                                ;                                         DMA BUFFER LENGTH IN COMBUF+6
                                ;
                                ; OUTPUTS:
                                ;       CONDITION CODE  - CLEAR = SUCCESS
                                ;                       - SET   = FAILURE
                                ;
                                ; IMPLICIT OUTPUTS:
                                ;       NONE
                                ;
                                ; SUBORDINATE ROUTINES USED:
                                ;       DOCOM
                                ;
                                ; FUNCTIONAL SIDE EFFECTS:
                                ;       NONE
                                ;
                                ; CALLING SEQUENCE:
                                ;       MOV     #BUFADD,COMBUF+4        ;GET DMA BUFFER ADDRESS
                                ;       MOV     #BUFLEN,COMBUF+6        ;GET DMA BUFFER LENGTH
                                ;       CALL    WRAMS           ;GO TO ROUTINE
                                ;       BCS     ERROR           ;CARRY SET IF ROUTINE HAD ERROR
                                ;--
006066  012737  002104  002616  WRAMS:: MOV     #DI.WRA,COMBUF          ; load write ram command code protocol.
006074  005037  002620          CLR     COMBUF+2                ; increment address to bottom 16 bits
006100  012737  000170  002626  MOV     #120.,COMBUF+8.         ; page number in ram of where to write.

006106                          DO.COMMAND      PI.CMA,1$       ; do the DMA
006106  012737  002400  002614  MOV     #PI.CMA,COMAND
006114  012737  002734  002216  MOV     #1500.,POLCNT
006122  004737  004572          CALL    DOCOM
006126  103000                  BCC     1$
006130  000207                  1$:     RETURN                  ; return failed
```

```
GLOBAL SUBROUTINES SECTION

                                ;++
                                ; FUNCTIONAL DESCRIPTION:
                                ;    SUBROUTINE TO READ DATA USING DMA.
                                ;
                                ; INPUTS:
                                ;       NONE
                                ;
                                ; IMPLICIT INPUTS:
                                ;       COMBUF          - COMMAND BUFFER CONTAINING :
                                ;                                         DMA BUFFER ADDRESS IN COMBUF+4
                                ;                                         DMA BUFFER LENGTH IN COMBUF+6
                                ;
                                ; OUTPUTS:
                                ;       CONDITION CODE  - CLEAR = SUCCESS
                                ;                       - SET   = FAILURE
                                ;
                                ; IMPLICIT OUTPUTS:
                                ;       DMA BUFFER CONTAINING THE DMA'D DATA
                                ;       CORRUPTS REGISTERS R1,R2
                                ;
                                ; SUBORDINATE ROUTINES USED:
                                ;       DOCOM
                                ;
                                ; FUNCTIONAL SIDE EFFECTS:
                                ;       NONE
                                ;
                                ; CALLING SEQUENCE:
                                ;       MOV     #BUFADD,COMBUF+4        ;GET DMA BUFFER ADDRESS
                                ;       MOV     #BUFLEN,COMBUF+6        ;GET DMA BUFFER LENGTH
                                ;       CALL    RRAMS           ;GO TO ROUTINE
                                ;       BCS     ERROR           ;CARRY SET IF ROUTINE HAD ERROR
                                ;-

006132                          RRAMS::
                                ;clear read buffer
006132  013701  002622                  MOV     COMBUF+4,R1             ; get address of read buffer
006136  013702  002624                  MOV     COMBUF+6,R2             ; get length in bytes to clear
006142  105021          10$:    CLRB    (R1)+                   ; clear read buffer
006144  077202                  SOB     R2,10$                  ; next byte
                                ;read into read buffer
006146  012737  002103  002616          MOV     #DI.RRA,COMBUF          ; load write ram command code protocol.
006154  005037  002620                  CLR     COMBUF+2                ; increment to bottom 16 bits
006160  012737  000170  002626          MOV     #120.,COMBUF+8.         ; page number in ram of where to read

006166                                  DO.COMMAND      PI.CMA,20$      ; do the DMA
006166  012737  002400  002614          MOV     #PI.CMA,COMAND
006174  012737  002734  002216          MOV     #1500.,POLCNT
006202  004737  004572                  CALL    DOCOM
006206  103000                          BCC     20$
006210  000207          20$:    RETURN                          ; return failed
```

GLOBAL SUBROUTINES SECTION SEQ 0041

```
1700                                    ;++
1701                                    ; FUNCTIONAL DESCRIPTION:
1702                                    ;    SUBROUTINE TO COMPARE DATA WRITTEN BY WRAMS WITH DATA READ BACK BY RRAMS.
1703                                    ;
1704                                    ; INPUTS:
1705                                    ;       NONE
1706                                    ;
1707                                    ; IMPLICIT INPUTS:
1708                                    ;       DMAOBUF         - ASSUMED OUTPUT BUFFER USED BY A PREVIOUS CALL TO WRAMS
1709                                    ;       DMAIBUF         - ASSUMED INPUT BUFFER USED BY A PREVIOUS CALL TO RRAMS
1710                                    ;       DMALEN          - LENGTH OF ABOVE BUFFERS
1711                                    ;
1712                                    ; OUTPUTS:
1713                                    ;       CONDITION CODE  - CLEAR = SUCCESS
1714                                    ;                         - SET   = FAILURE
1715                                    ;
1716                                    ; IMPLICIT OUTPUTS:
1717                                    ;       CORRUPTS REGISTERS R1,R2
1718                                    ;
1719                                    ; SUBORDINATE ROUTINES USED:
1720                                    ;       NONE
1721                                    ;
1722                                    ; FUNCTIONAL SIDE EFFECTS:
1723                                    ;       NONE
1724                                    ;
1725                                    ; CALLING SEQUENCE:
1726                                    ;       CALL    CRAMS           ;GO TO ROUTINE
1727                                    ;       BCS     ERROR           ;CARRY SET IF ROUTINE HAD ERROR
1728                                    ;--
1729
1730 006212  012701  002762            CRAMS:: MOV     #DMAOBUF,R1             ; get start address of data
1731 006216  012702  003036                    MOV     #DMAIBUF,R2             ; get start address of data read back
1732 006222  012703  000054                    MOV     #DMALEN,R3              ; get length in bytes
1733 006226  122122                    10$:    CMPB    (R1)+,(R2)+             ; are we o.k
1734 006230  001417                            BEQ     20$                     ; branch if yes
1735 006232                                    ERR.FILLIN      11.,ERR11       ; error return if not
     006232  012737  000002  002202            MOV     #2,ERRTYP       ; assume hard error for now
     006240  012737  000013  002204            MOV     #11.,ERRNBR     ; get error 11.
     006246  012737  003753  002206            MOV     #ERR11,ERRMSG   ; get error ERR11
     006254  012737  004302  002210            MOV     #ERRCODE,ERRBLK ; get error routine
1736 006262                                    ERROR
1737 006264                                    ERR.RETURN
     006264  000261                            SEC
     006266  000207                            RETURN
1738 006270  077322                    20$:    SOB     R3,10$                  ; branch if we have not finished
1739 006272                                    OK.RETURN
     006272  000241                            CLC
     006274  000207                            RETURN
1740
```

GLOBAL SUBROUTINES SECTION

```
                                ;++
                                ; FUNCTIONAL DESCRIPTION:
                                ;    SUBROUTINE TO SET UP TEST MASK FOR EVERY MICRODIAGNOSTIC TEST.
                                ;
                                ; INPUTS:
                                ;       NONE
                                ;
                                ; IMPLICIT INPUTS:
                                ;       L$TEST          - CURRENT TEST
                                ;
                                ; OUTPUTS:
                                ;       NONE
                                ;
                                ; IMPLICIT OUTPUTS:
                                ;       CORRUPTS REGISTERS R1,R2
                                ;       PBLOC+2         - LOCATION TO STORE TEST MASK ( TEMPORARY )
                                ;       TMASKS          - LOCATION TO STORE TEST MASK ( PERMANENT )
                                ;
                                ; SUBORDINATE ROUTINES USED:
                                ;       NONE
                                ;
                                ; FUNCTIONAL SIDE EFFECTS:
                                ;       NONE
                                ;
                                ; CALLING SEQUENCE:
                                ;       CALL    TSTMSK          ;GO TO ROUTINE
                                ;--
006276                          TSTMSK::
006276  013702  002114                  MOV     L$TEST,R2               ; get current test ( 1 - MAXTST )
006302  005302                          DEC     R2                      ; Get offset from 0
006304  006302                          ASL     R2                      ; in bytes.
006306  060402                          ADD     R4,R2                   ; Get offset from FLAGS
006310  004737  007112                  CALL    TSTONCE                 ; First time through this test ?
006314  001047                          BNE     40$                     ; branch if not
006316  012737  000400  002224          MOV     #<1*400>,PBLOC+2        ; test count = 1 ( top byte )
006324                                  RFLAGS  R1                      ; get DRS operator flag settings
006330  032701  100000                  BIT     #HOE,R1                 ; HOE bit set ?
006334  001403                          BEQ     10$                     ; branch if not
006336  152737  000100  002224          BISB    #BIT6,PBLOC+2           ; set the HOE bit
006344  032701  040000          10$:    BIT     #LOE,R1                 ; LOE bit set ?
006350  001426                          BEQ     30$                     ; branch if not
006352                                  PRINTF  #TM1                    ; print warning message
006372                                  MANUAL                          ; manual allowed ?
006374                                  BNCOMPLETE      20$             ; no - do same as op answer yes
006376                                  GMANIL  TM2,TEMP1,177777,NO     ; op wish LOE set ?
006412  005737  002212                  TST     TEMP1
006416  001403                          BEQ     30$                     ; branch if no
006420  152737  000200  002224  20$:    BISB    #BIT7,PBLOC+2           ; set the LOE bit
006426  013762  002224  000010  30$:    MOV     PBLOC+2,TMASKS(R2)      ; save the mask
006434  016237  000010  002224  40$:    MOV     TMASKS(R2),PBLOC+2      ; copy saved mask into parameter block
006442  000207                          RETURN
                                ;
006444     045     116     045  TM1:    .ASCIZ  /%N%A** AN ON-BOARD ERROR WILL HANG THIS TEST INDEFINITELY **/
006541     104     125     105  TM2:    .ASCIZ  /DUE TO ON-BOARD LOE FLAG SET - DO YOU WISH IT SET ?/
                                ;
                                        .EVEN
```

GLOBAL SUBROUTINES SECTION

```
                                ;++
                                ; FUNCTIONAL DESCRIPTION:
                                ;    SUBROUTINE TO MOVE MICRODIAGNOSTIC PARAMETERS INTO THE PARAMETER REGISTER.
                                ;
                                ; INPUTS:
                                ;       NONE
                                ;
                                ; IMPLICIT INPUTS:
                                ;       PBLOC           - BUFFER CONTAINING MICRODIAGNOSTIC PARAMETERS
                                ;       CURPAR          - CURRENT PARAMETER REGISTER ADDRESS
                                ;
                                ; OUTPUTS:
                                ;       CONDITION CODE  - CLEAR = SUCCESS
                                ;                       - SET   = FAILURE
                                ;
                                ; IMPLICIT OUTPUTS:
                                ;       CORRUPTS REGISTERS R1,R2
                                ;
                                ; SUBORDINATE ROUTINES USED:
                                ;       POLLCRY
                                ;
                                ; FUNCTIONAL SIDE EFFECTS:
                                ;       NONE
                                ;
                                ; CALLING SEQUENCE:
                                ;       CALL    MVPAR           ;GO TO ROUTINE
                                ;       BCS     ERROR           ;CARRY SET IF ROUTINE HAD ERROR
                                ;--
                                
006626  012701  000004        MVPAR:: MOV     #4,R1                   ; get length to move
006632  012702  002222                MOV     #PBLOC,R2               ; get start address of Pbloc
006636                                SET.POLCNT
006636  012737  002734  002216        MOV     #1500.,POLCNT
006644  004737  004512        10$:    CALL    POLLCRY                 ; wait for SVL CRY
006650  103003                        BCC     20$                     ; branch if OK
006652                                ERROR
006654                                ERR.RETURN
006654  000261                        SEC
006656  000207                        RETURN
006660  012277  173350        20$:    MOV     (R2)+,@CURPAR           ; move parameter into parameter reg
006664  077111                        SOB     R1,10$                  ; next
006666                                OK.RETURN
006666  000241                        CLC
006670  000207                        RETURN
```

GLOBAL SUBROUTINES SECTION

```
                                ;++
                                ; FUNCTIONAL DESCRIPTION:
                                ;    SUBROUTINE TO INVOKE A MICRODIAGNOSTIC
                                ;
                                ; INPUTS:
                                ;       NONE
                                ;
                                ; IMPLICIT INPUTS:
                                ;       POLCNT          - POLL COUNT
                                ;       PBLOC           - BUFFER FOR MICRODIAGNOSTIC PARAMETERS
                                ;       STABUF          - STATUS BUFFER FOR RETURNS FROM VSV21
                                ;
                                ; OUTPUTS:
                                ;       CONDITION CODE  - CLEAR = SUCCESS
                                ;                       - SET   = FAILURE
                                ;
                                ; IMPLICIT OUTPUTS:
                                ;       NONE
                                ;
                                ; SUBORDINATE ROUTINES USED:
                                ;       DOCOM
                                ;
                                ; FUNCTIONAL SIDE EFFECTS:
                                ;       NONE
                                ;
                                ; CALLING SEQUENCE:
                                ;       CALL    DOMIC           ;GO TO ROUTINE
                                ;       BCS     ERROR           ;CARRY SET IF ROUTINE HAD ERROR
                                ;--
                                ;
006672  012737  002000  002614  DOMIC:: MOV     #PI.TST,COMAND
006700  032737  000200  002224          BIT     #BIT7,PBLOC+2           ; LOE ?
006706  001004                        BNE     10$                     ; branch if yes
006710                                SET.POLCNT                      ; set ordinary t'meout
006710  012737  002734  002216          MOV     #1500.,POLCNT
006716  000403                        BR      20$
006720  012737  177777  002216  10$:   MOV     #-1,POLCNT              ; set infinite timeout
006726  004737  004572          20$:   CALL    DOCOM                   ; invoke micro diagnostic.
006732  103001                        BCC     30$
006734  000207                        RETURN
006736  122737  000011  002667  30$:   CMPB    #DI.MIC,STABUF+1        ; correct status return ?
006744  001417                        BEQ     40$                     ; branch if yes
006746                                ERR.FILLIN      12.,ERR12               ; error if no
006746  012737  000002  002202          MOV     #2,ERRTYP       ; assume hard error for now
005754  012737  000014  002204          MOV     #12.,ERRNBR     ; get error 12.
006762  012737  004000  002206          MOV     #ERR12,ERRMSG   ; get error ERR12
006770  012737  004302  002210          MOV     #ERRCODE,ERRBLK ; get error routine
006776                                ERROR
007000                                ERR.RETURN
007000  000261                        SEC
007002  000207                        RETURN
007004  005737  0026'0          40$:   TST     STABUF+2                ; test successful ?
007010  001417                        BEQ     50$                     ; branch if YES
007012                                ERR.FILLIN      13.,ERR13               ; error if no
007012  012737  000002  002202          MOV     #2,ERRTYP       ; assume hard error for now
007020  012737  000015  002204          MOV     #13.,ERRNBR     ; get error 13.
007026  012737  004065  002206          MOV     #ERR13,ERRMSG   ; get error ERR13
```

GLOBAL SUBROUTINES SECTION

SEQ 0045

```
      007034  012737  004302  002210          MOV     #ERRCODE,ERRBLK ; get error routine
 1887 007042                                  ERPUR
 1888 007044                                  ERR.RETURN
      007044  000261                          SEC
      007046  000207                          RETURN
 1889 007050                          50$:    OK.RETURN
      007050  000241                          CLC
      007052  000207                          RETURN
 1890
```

GLOBAL SUBROUTINES SECTION

```
                                ;++
                                ; FUNCTIONAL DESCRIPTION:
                                ;    SUBROUTINE TO MARK CURRENT TEST AS HAVING BEEN DONE AT LEAST ONCE
                                ;
                                ; INPUTS:
                                ;       NONE
                                ;
                                ; IMPLICIT INPUTS:
                                ;       L$TEST          - CURRENT TEST
                                ;
                                ; OUTPUTS:
                                ;       NONE
                                ;
                                ; IMPLICIT OUTPUTS:
                                ;       CORRUPTS REGISTERS R0,R1
                                ;       FTHRU           - FLAG BITS INDICATING IF TESTS HAVE BEEN DONE ONCE
                                ;
                                ; SUBORDINATE ROUTINES USED:
                                ;       NONE
                                ;
                                ; FUNCTIONAL SIDE EFFECTS:
                                ;       NONE
                                ;
                                ; CALLING SEQUENCE:
                                ;       CALL    MRKONCE         ;GO TO ROUTINE
                                ;--
007054                          MRKONCE::
007054  013700  002114                  MOV     L$TEST,R0               ; get current test
007060  010401                          MOV     R4,R1                   ; get address of flags
007062  020027  000017                  CMP     R0,#15.                 ; test 1-15 ?
007066  003404                          BLE     10$                     ; branch if yes
007070  162700  000017                  SUB     #15.,R0                 ; get in range for a test > 15
007074  062701  000002                  ADD     #2,R1                   ; point to second flag word (test > 15)
007100  006300                  10$:    ASL     R0                      ; get offset from start of BITS
007102  056061  002244  000004          BIS     BITS(R0),FTHRU(R1)      ; set appropriate flag bit
007110  000207                          RETURN
```

GLOBAL SUBROUTINES SECTION SEQ 0047

```
1929                                    ;++
1930                                    ; FUNCTIONAL DESCRIPTION:
1931                                    ;    SUBROUTINE TO TEST IF CURRENT TEST HAS BEEN DONE AT LEAST ONCE
1932                                    ;
1933                                    ; INPUTS:
1934                                    ;       NONE
1935                                    ;
1936                                    ; IMPLICIT INPUTS:
1937                                    ;       L$TEST          - CURRENT TEST
1938                                    ;       FTHRU           - FLAG BITS INDICATING IF TESTS HAVE BEEN DONE ONCE
1939                                    ;
1940                                    ; OUTPUTS:
1941                                    ;       Z BIT           - SET INDICATES TEST HAS BEEN DONE AT LEAST ONCE
1942                                    ;                       - CLEAR INDICATES TEST HAS NOT BEEN DONE
1943                                    ;
1944                                    ; IMPLICIT OUTPUTS:
1945                                    ;       CORRUPTS REGISTERS R0,R1
1946                                    ;
1947                                    ; SUBORDINATE ROUTINES USED:
1948                                    ;       NONE
1949                                    ;
1950                                    ; FUNCTIONAL SIDE EFFECTS:
1951                                    ;       NONE
1952                                    ;
1953                                    ; CALLING SEQUENCE:
1954                                    ;       CALL    TSTONCE         ;GO TO ROUTINE
1955                                    ;       BNE     ATLEASTONCE     ;BRANCH IF TEST DONE AT LEAST ONCE
1956                                    ;--
1957
1958 007112                             TSTONCE::
1959 007112  013700  002114                    MOV     L$TEST,R0               ; get current test
1960 007116  010401                            MOV     R4,R1                   ; get address of flags
1961 007120  020027  000017                    CMP     R0,#15.                 ; test 1-15 ?
1962 007124  003404                            BLE     10$                     ; branch if yes
1963 007126  162700  000017                    SUB     #15.,R0                 ; get in range for a test > 15
1964 007132  062701  000002                    ADD     #2,R1                   ; point to second flag word (test > 15)
1965 007136  006300                    10$:    ASL     R0                      ; get offset from start of BITS
1966 007140  036061  002244  000004            BIT     BITS(R0),FTHRU(R1)      ; test appropriate flag bit
1967 007146  000207                            RETURN
```

GLOBAL SUBROUTINES SECTION

```
                                ;++
                                ; FUNCTIONAL DESCRIPTION:
                                ;    SUBROUTINE TO BE CALLED AT START OF EACH TEST TO PERFORM INITIALISATION.
                                ;
                                ; INPUTS:
                                ;       NONE
                                ;
                                ; IMPLICIT INPUTS:
                                ;       L$TEST          - CURRENT TEST
                                ;       T1DONE          - TEST 1 DONE FLAG
                                ;
                                ; OUTPUTS:
                                ;       CONDITION CODE  - CLEAR = SUCCESS
                                ;                       - SET   = FAILURE
                                ;
                                ; IMPLICIT OUTPUTS:
                                ;       CORRUPTS REGISTER R1
                                ;       PBLOC           - FILLS IN CORRESPONDING MICRODIAGNOSTIC TEST NUMBER
                                ;
                                ; SUBORDINATE ROUTINES USED:
                                ;       NONE
                                ;
                                ; FUNCTIONAL SIDE EFFECTS:
                                ;       NONE
                                ;
                                ; CALLING SEQUENCE:
                                ;       CALL    STTEST          ;GO TO ROUTINE
                                ;       BCS     ERROR           ;BRANCH IF ERROR
                                ;--
                                ;
007150                          STTEST::
007150  005737  002242                  TST     NOTEST                  ; can test be done ?
007154  001402                          BEQ     5$                      ; branch if yes
007156                                  ERR.RETURN                      ; error return if not
007156  000261                          SEC
007160  000207                          RETURN
007162  013701  002114          5$:     MOV     L$TEST,R1               ; get current test
007166  005301                          DEC     R1                      ; Get 'n range 0 - 40
007170  006301                          ASL     R1                      ; for offset calculations.
007172  005701                          TST     R1                      ; is it test 1 ?
007174  001414                          BEQ     20$                     ; exit if yes
                                ;
                                ; Test is not 1 - ensure that test 1 has already been done
                                ;
007176  005764  000000                  TST     T1DONE(R4)              ; test 1 done ?
007202  001006                          BNE     10$                     ; branch if yes
007204                                  ERRHRD  1,ERR1,ERRCODE          ; error exit if not
007214                                  ERR.RETURN
007214  000261                          SEC
007216  000207                          RETURN
                                ;
                                ; Set up microdiagnostic test number in PBLOC
                                ;
007220  016137  007232  002222  10$:     MOV     MICNUM(R1),PBLOC        ; get microd'agnostic test number
                                ;
007226                          20$:    OK.RETURN
007226  000241                          CLC
```

GLOBAL SUBROUTINES SECTION

```
        007230  000207                          RETURN
   2021
   2022                                ;Microdiagnostic test number versus test. ( -1 is null )
   2023
   2024 007232  177777  177777  177777  MICNUM: .WORD    1., 1., 1.,0,1,2,3,4,5,6,7
   2025 007260  000010  000011  000012          .WORD   8.,9.,10.,11.,11.,11.,12.,13.,-1.
   2026
```

GLOBAL SUBROUTINES SECTION

```
                                .TITLE MISCELLANEOUS SECTIONS
                                .SBTTL  PROTECTION TABLE

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

007302                                  BGNPROT

007302  000000                          0               ;OFFSET INTO P-TABLE FOR CSR ADDRESS
007304  177777                          -1              ;OFFSET INTO P-TABLE FOR MASSBUS ADDRESS
007306  177777                          -1              ;OFFSET INTO P-TABLE FOR DRIVE NUMBER

007310                                  ENDPROT
```

INITIALIZE SECTION

```
                                .SBTTL  INITIALIZE SECTION

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

007310                                  BGNINIT

007310                                  READEF  #EF.START       ; IF started by start THEN
007316                                  BCOMPLETE       10$     ; do start coding
007320                                  READEF  #EF.RESTART     ; IF not started by restart THEN
007326                                  BNCOMPLETE      20$     ; skip start coding
007330  012700  002304          10$:    MOV     #FLAGS,R0       ; get address of FLAGS
007334  012701  000140                  MOV     #<LENFLAGS*MAXUNIT>,R1  ; get length of FLAGS
007340  005020                  15$:    CLR     (R0)+           ; clear out FLAGS
007342  077102                          SOB     R1,15$          ; next
007344  005037  002242                  CLR     NOTEST          ; say tests can be done
007350                          20$:    READEF  #EF.CONTINUE    ; IF started by continue THEN
007356                                  BCOMPLETE       END     ; don't get P-table
007360                                  READEF  #EF.NEW         ; IF not a new pass THEN
007366                                  BNCOMPLETE      NEXT    ; skip setup

007370  012737  177777  002220          MOV     #-1,LOGUNT      ; initialise unit number
007376  005237  002220          NEXT:   INC     LOGUNT          ; next unit number
007402  023737  002220  002012          CMP     LOGUNT,L$UNIT   ; IF passed operator's max unit number THEN
007410  001437                          BEQ     ABORT           ; abort pass
007412  023727  002220  000004          CMP     LOGUNT,#MAXUNIT ; IF passed max unit number THEN
007420  002033                          BGE     ABORT           ; abort pass (N.B. OPERATOR'S MAX COULD BE >)
007422                                  GPHARD  LOGUNT,R1       ; Get hardware P-table address
007432                                  BNCOMPLETE      NEXT    ; if not available get next
007434  011137  002232                  MOV     (R1),CURCSR     ; get current device address from P-table
007440  062737  000002  002232          ADD     #2,CURCSR       ; add 2 to get current CSR
007446  011137  002234                  MOV     (R1),CURPAR     ; get current device address from P-table
007452  062737  000004  002234          ADD     #4,CURPAR       ; add 4 to get current Parameter Register
007460  016137  000002  002236          MOV     2(R1),CURVEC    ; get current vector address
007466  013701  002220                  MOV     LOGUNT,R1       ; get unit number
007472  070127  000060                  MUL     #<LENFLAGS*2>,R1        ; get offset from start of FLAGS
007476  062701  002304                  ADD     #FLAGS,R1       ; get true address of FLAGS for this unit
007502  010104                          MOV     R1,R4           ; save in R4
007504                          END:    EXIT    INIT
007510                          ABORT:  DOCLN
007512                                  ENDINIT

                                        .EVEN
```

AUTODROP SECTION

```
                                .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.
                                ;--

007514                                  BGNAUTO

007514                                  ENDAUTO
```

```
                                .SBTTL  CLEANUP CODING SECTION

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

007516                                  BGNCLN

007516  023727  002220  000004          CMP     LOGUNT,#MAXUNIT         ; IF greater than max unit THEN
007524  002011                          BGE     10$                     ; exit
007526  013700  002232                  MOV     CURCSR,R0               ; get current CSR
007532  005040                          CLR      (R0)                   ; Write to the previous word (i.e. H/W
                                ;                                       ; register ) to reset VSV21.

007534  005064  000002                  CLR     DMASET(R4)              ; say DMA disabled
007540  005064  000000                  CLR     T1DONE(R4)              ; say Test 1 not done
007544  005037  002242                  CLR     NOTEST                  ; say tests can be done

007550                          10$:    EXIT    CLN

007554                                  ENDCLN
```

CLEANUP CODING SECTION

```
                                        .TITLE HARDWARE TESTS
                                        .SBTTL  TEST 1:  VERIFY VSV21 PRESENCE


                                        ;++
                                        ; This test verifies that there is a VSV21 device at the current CSR.
                                        ; On the first pass it displays the version of the VSV21. It also disables
                                        ; interrupts.
                                        ; -

007556                                          ST.TEST
007556  004737  007150                          CALL    STTEST
007562  103002                                  BCC     30000$

                                        ;SUBTEST 1 sends an interrupt mask to disable interrupts.

007570                                          BGNSUB
007572  005064  000000                          CLR     T1DONE(R4)              ; say that TEST1 has not been done
007576                                          DO.COMMAND      PI.IDS,10$      ; send over interrupt mask.
007576  012737  001400  002614                  MOV     #PI.IDS,COMAND
007604  012737  002734  002216                  MOV     #1500.,POLCNT
007612  004737  004572                          CALL    DOCOM
007616  103002                                  BCC     10$
007620                                          ESCAPE  TST
007624  005264  000000                10$:      INC     T1DONE(R4)              ; say TEST1 done so that if powerup
                                        ;                                       ; when request firmware id will get
                                        ;                                       ; error
007630                                          ENDSUB

                                        ;SUBTEST 2 requests firmware ID and displays it

007632                                          BGNSUB
007634                                          DO.COMMAND      PI.IDM,10$      ; request firmware id
007634  012737  000400  002614                  MOV     #PI.IDM,COMAND
007642  012737  002734  002216                  MOV     #1500.,POLCNT
007650  004737  004572                          CALL    DOCOM
007654  103004                                  BCC     10$
007656  005064  000000                          CLR     T1DONE(R4)              ; say that TEST1 has not been done
007662                                          ESCAPE  TST
007666  005264  000000                10$:      INC     T1DONE(R4)              ; say TEST1 done
007672  004737  007112                          CALL    TSTONCE                 ; first time through ?
007676  001044                                  BNE     20$                     ; branch if not
007700  004737  007054                          CALL    MRKONCE                 ; say not first time through
007704  000337  002670                          SWAB    STABUF+2                ; Rearrange
007710  000337  002672                          SWAB    STABUF+4                ; firmware ID
007714  000337  002674                          SWAB    STABUF+6                ; so that can be printed.
007720  000337  002676                          SWAB    STABUF+8.               ; This assumes top byte zero.
007724                                          PRINTX  #TS1,#<STABUF+2>        ; Print firmware id.
007750                                          PRINTX  #TS2
007770                                          PRINTX  #TS3
010010                                  20$:    ENDSUB

010012                                          EXIT    TST

010016      045     116     045         TS1:    .ASCIZ  /%N%AFIRMWARE ID IS : %T/
010046      045     116     062         TS2:    .ASCIZ  /%N2%ATESTS 4,13,14,16 AND 17 WILL ABORT WITH AN ERROR IF/
```

SEQ 0055

TEST 1: VERIFY VSV21 PRESENCE

```
 2219 010137     045     116     045  TS3:    .ASCIZ  /%N%AEXTERNAL LOOPBACK CONNECTORS ARE NOT ATTACHED TO PORTS 0-3/
 2220
 2221                                         .EVEN
 2222
 2223 010236                                  ENDTST
 2224
```

TEST 1: VERIFY VSV21 PRESENCE

```
                                        .SBTTL  TEST 2:  IN DEPTH Q22 BUS TEST


                                        ;++
                                        ; This test performs an in depth test of the Q22 bus by means of exercising the
                                        ; DMA mechanism in both directions. It sets up the DMA protocol and must be
                                        ; done before other tests requiring DMA.
                                        ;-

010240                                          ST.TEST
010240  004737  007150                          CALL    STTEST
010244  103002                                  BCC     30001$

                                        ;SUBTEST 1 moves DMA buffer descriptors through the parameter register

010252                                          BGNSUB
010254  012701  002604                          MOV     #BUFDES,R1              ; get address of buffer descriptors
010260  012702  000004                          MOV     #4,R2                   ; length to move

010264                                  10$:    SET.POLCNT
010264  012737  002734  002216                  MOV     #1500.,POLCNT
010272  004737  004512                          CALL    POLLCRY                 ; wait for SVL CRY
010276  103002                                  BCC     20$                     ; branch if OK
010300                                          ESCAPE  TST
010304  012177  171724                  20$:    MOV     (R1)+,@CURPAR           ; move buffer desc. into parameter reg
010310  077213                                  SOB     R2,10$                  ; next descriptor
010312                                          ENDSUB

                                        ;SUBTEST 2 does the buffer sent command

010314                                          BGNSUB
010316                                          DO.COMMAND      PI.SEN,BUFOK    ; do buffers sent command
010316  012737  001000  002614                  MOV     #PI.SEN,COMAND
010324  012737  002734  002216                  MOV     #1500.,POLCNT
010332  004737  004572                          CALL    DOCOM
010336  103016                                  BCC     BUFOK
010340                                          ENDSUB

                                        ;SUBTEST 3 does a buffers cancelled command if SUBTEST 2 failed

010342                                          BGNSUB
010344                                          DO.COMMAND      PI.CAN,10$      ; do buffers cancelled command
010344  012737  001001  002614                  MOV     #PI.CAN,COMAND
010352  012737  002734  002216                  MOV     #1500.,POLCNT
010360  004737  004572                          CALL    DOCOM
010364  103000                                  BCC     10$
010366                                  10$:    ESCAPE  TST
010372                                          ENDSUB

                                        ;SUBTEST 3 writes to RAM using DMA. This is SUBTEST 3 if 2 is successful.

010374                                  BUFOK:  BGNSUB
010376  005264  000002                          INC     DMASET(R4)              ; record DMA enabled
010402                                          DO.WRAMS                        ; write to RAM
010402  012737  002762  002622                  MOV     #DMAOBUF,COMBUF+4
010410  012737  000054  002624                  MOV     #DMALEN,COMBUF+6
```

TES 2: IN DEPTH Q22 BUS TEST

```
        010416  004737  006066                  CALL    WRAMS
   2270 010422  103002                          BCC     10$                     ; carry on if OK
   2271 010424                                  ESCAPE  TST
   2272 010430                          10$:    ENDSUB
   2273
   2274                                 ;SUBTEST 4 reads from RAM using DMA.
   2275
   2276 010432                                  BGNSUB
   2277 010434                                  DO.RRAMS                        ; read from RAM
        010434  012737  003036  002622          MOV     #DMAIBUF,COMBUF+4
        010442  012737  000054  002624          MOV     #DMALEN,COMBUF+6
        010450  004737  006132                  CALL    RRAMS
   2278 010454  103002                          BCC     10$                     ; carry on if OK
   2279 010456                                  ESCAPE  TST
   2280 010462                          10$:    ENDSUB
   2281
   2282                                 ;SUBTEST 5 compares data read back with data written
   2283
   2284 010464                                  BGNSUB
   2285 010466  004737  006212                  CALL    CRAMS                   ; compare the patterns
   2286 010472  103002                          BCC     10$                     ; carry on if OK
   2287 010474                                  ESCAPE  TST
   2288 010500                          10$:    ENDSUB
   2289
   2290 010502                                  EXIT    TST
   2291
   2292                                         .EVEN
   2293 010506                                  ENDTST
   2294
```

```
 2296
 2297                                   .SBTTL  TEST 3:  FORCED ERROR TEST
 2298
 2299
 2300                                   ;++
 2301                                   ; This test sends an invalid command to the VSV21 to provoke an error packet.
 2302                                   ;--
 2303
 2304 010510                                       ST.TEST
      010510  004737  007150                       CALL    STTEST
      010514  103002                               BCC     30002$
 2305
 2306                                   ;SUBTEST 1 sends an invalid command
 2307
 2308 010522                                       BGNSUB
 2309 010524                                       DO.COMMAND      PI.RUB,10$      ; send a rubbish command
      010524  012737  177777  002614               MOV     #PI.RUB,COMAND
      010532  012737  002734  002216               MOV     #1500.,PGLCNT
      010540  004737  004572                       CALL    DOCOM
      010544  103000                               BCC     10$
 2310 010546  122737  000177  002667  10$:        CMPB    #DI.ERL,STABUF+1        ; correct error return ?
 2311 010554  001002                               BNE     20$                     ; branch if not
 2312 010556                                       EXIT    TST                     ; received an error   exit ok
 2313
 2314 010562                                   20$:    ERR.FILLIN      14.,ERR14
 2315 010612                                           ERROR
 2316 010614                                           ESCAPE  TST
 2317 010620                                           ENDSUB
 2318
 2319 010622                                           ENDTST
```

TEST 3: FORCED ERROR TEST

```
                                .SBTTL  TEST 4:  FULL ON BOARD TESTS


                                ;++
                                ; This test invokes full on-board VSV21 microdiagnostic tests.
                                ; If the HOE flag is specified the VSV21 will halt on error.
                                ; If the LOE flag is specified the operator will be given the choice on the
                                ; first pass of whether he wishes the on-board tests to loop on error. If he
                                ; selects yes the test may hang indefinitely as the on board will not return
                                ; control to the host but will continue looping.
                                ;--

010624                                  ST.TEST
010624  004737  007150                  CALL    STTEST
010630  103002                          BCC     30003$

010636  004737  006276                  CALL    TSTMSK                  ; get test mask for microdiagnostic
010642  004737  007054                  CALL    MRKONCE                 ; mark test as entered at least once

                                ;SUBTEST 1 moves the parameters for the microdiagnostics test through the
                                ;parameter register

010646                                  BGNSUB
010650  004737  006626                  CALL    MVPAR                   ; move the parameters
010654  103002                          BCC     10$
010656                                  ESCAPE  TST
010662                          10$:    ENDSUB

                                ;SUBTEST 2 invokes the microdiagnostic test

010664                                  BGNSUB
010666  004737  006672                  CALL    DOMIC
010672  103002                          BCC     10$
010674                                  ESCAPE  TST
010700                          10$:    ENDSUB

010702                                  ENDTST
```

```
                                .SBTTL  TEST 5:  ROM CHECKSUM TEST


                                ;++
                                ; This test invokes the on-board VSV21 ROM CHECKSUM test.
                                ; If the HOE flag is specified the VSV21 will halt on error.
                                ; If the LOE flag is specified the operator will be given the choice on the
                                ; first pass of whether he wishes the on-board tests to loop on error. If he
                                ; selects yes the test may hang indefinitely as the on board will not return
                                ; control to the host but will continue looping.
                                ;--

010704                                  ST.TEST
010704  004737  007150                  CALL    STTEST
010710  103002                         BCC     30004$

010716  004737  006276                  CALL    TSTMSK                  ; get test mask for microdiagnostic
010722  004737  007054                  CALL    MRKONCE                 ; mark test as entered at least once

                                ;SUBTEST 1 moves the parameters for the microdiagnostics test through the
                                ;parameter register

010726                                  BGNSUB
010730  004737  006626                  CALL    MVPAR                   ; move the parameters
010734  103002                         BCC     10$
010736                                  ESCAPE  TST
010742                          10$:    ENDSUB

                                ;SUBTEST 2 invokes the microdiagnostic test

010744                                  BGNSUB
010746  004737  006672                  CALL    DOMIC
010752  103002                         BCC     10$
010754                                  ESCAPE  TST
010760                          10$:    ENDSUB

010762                                  ENDTST
```

TEST 5: ROM CHECKSUM TEST

```
                                .SBTTL  TEST 6:  NVRAM CHECKSUM TEST


                                ;++
                                ; This test invokes the on-board VSV21 NVRAM CHECKSUM test.
                                ; If the HOE flag is specified the VSV21 will halt on error.
                                ; If the LOE flag is specified the operator will be given the choice on the
                                ; first pass of whether he wishes the on-board tests to loop on error. If he
                                ; selects yes the test may hang indefinitely as the on board will not return
                                ; control to the host but will continue looping.
                                ;--

010764                                  ST.TEST
010764  004737  007150                  CALL    ST.TEST
010770  103002                          BCC     30005$

010776  004737  006276                  CALL    TSTMSK                  ; get test mask for microdiagnostic
011002  004737  007054                  CALL    MRKONCE                 ; mark test as entered at least once

                                ;SUBTEST 1 moves the parameters for the microdiagnostics test through the
                                ;parameter register

011006                                  BGNSUB
011010  004737  006626                  CALL    MVPAR                   ; move the parameters
011014  103002                          BCC     10$
011016                                  ESCAPE  TST
011022                          10$:    ENDSUB

                                ;SUBTEST 2 invokes the microdiagnostic test

011024                                  BGNSUB
011026  004737  006672                  CALL    DOMIC
011032  103002                          BCC     10$
011034                                  ESCAPE  TST
011040                          10$:    ENDSUB

011042                                  ENDTST
```

TEST 6: NVRAM CHECKSUM TEST

```
2433
2434                                    .SBTTL  TEST 7:  RAM TEST
2435
2436
2437                                    ;++
2438                                    ; This test invokes the on-board VSV21 RAM test.
2439                                    ; If the HOE flag is specified the VSV21 will halt on error.
2440                                    ; If the LOE flag is specified the operator will be given the choice on the
2441                                    ; first pass of whether he wishes the on-board tests to loop on error. If he
2442                                    ; selects yes the test may hang indefinitely as the on board will not return
2443                                    ; control to the host but will continue looping.
2444                                    ;--
2445
2446 011044                                     ST.TEST
     011044  004737  007150                     CALL    STTEST
     011050  103002                             BCC     30006$
2447
2448 011056  004737  006276                     CALL    TSTMSK                  ; get test mask for microdiagnostic
2449 011062  004737  007054                     CALL    MRKONCE                 ; mark test as entered at least once
2450
2451                                    ;SUBTEST 1 moves the parameters for the microdiagnostics test through the
2452                                    ;parameter register
2453
2454 011066                                     BGNSUB
2455 011070  004737  006626                     CALL    MVPAR                   ; move the parameters
2456 011074  103002                             BCC     10$
2457 011076                                     ESCAPE  TST
2458 011102                             10$:    ENDSUB
2459
2460                                    ;SUBTEST 2 invokes the microdiagnostic test
2461
2462 011104                                     BGNSUB
2463 011106  004737  006672                     CALL    DOMIC
2464 011112  103002                             BCC     10$
2465 011114                                     ESCAPE  TST
2466 011120                             10$:    ENDSUB
2467
2468 011122                                     ENDTST
```

TEST 7: RAM TEST

```
2470
2471                                    .SBTTL  TEST 8:  RAM ADDRESSING TEST
2472
2473
2474                                    ;++
2475                                    ; This test invokes the on-board VSV21 RAM ADDRESSING test.
2476                                    ; If the HOE flag is specified the VSV21 will halt on error.
2477                                    ; If the LOE flag is specified the operator will be given the choice on the
2478                                    ; first pass of whether he wishes the on-board tests to loop on error. If he
2479                                    ; selects yes the test may hang indefinitely as the on board will not return
2480                                    ; control to the host but will continue looping.
2481                                    ;--
2482
2483 011124                                       ST.TEST
     011124  004737  007150                        CALL    STTEST
     011130  103002                                BCC     30007$
2484
2485 011136  004737  006276                        CALL    TSTMSK                  ; get test mask for microdiagnostic
2486 011142  004737  007054                        CALL    MRKONCE                 ; mark test as entered at least once
2487
2488                                    ;SUBTEST 1 moves the parameters for the microdiagnostics test through the
2489                                    ;parameter register
2490
2491 011146                                       BGNSUB
2492 011150  004737  006626                        CALL    MVPAR                   ; move the parameters
2493 011154  103002                                BCC     10$
2494 011156                                       ESCAPE  TST
2495 011162                                10$:    ENDSUB
2496
2497                                    ;SUBTEST 2 invokes the microdiagnostic test
2498
2499 011164                                       BGNSUB
2500 011166  004737  006672                        CALL    DOMIC
2501 011172  103002                                BCC     10$
2502 011174                                       ESCAPE  TST
2503 011200                                10$:    ENDSUB
2504
2505 011202                                       ENDTST
```

TEST 8: RAM ADDRESSING TEST

```
                                .SBTTL  TEST 9:  68K PROCESSOR TEST


                                ;++
                                ; This test invokes the on-board VSV21 68K PROCESSOR test.
                                ; If the HOE flag is specified the VSV21 will halt on error.
                                ; If the LOE flag is specified the operator will be given the choice on the
                                ; first pass of whether he wishes the on-board tests to loop on error. If he
                                ; selects yes the test may hang indefinitely as the on board will not return
                                ; control to the host but will continue looping.
                                ;--

011204                                  ST.TEST
011204  004737  007150                  CALL    STTEST
011210  103002                          BCC     30008$

011216  004737  006276                  CALL    TSTMSK                  ; get test mask for microdiagnostic
011222  004737  007054                  CALL    MRKONCE                 ; mark test as entered at least once

                                ;SUBTEST 1 moves the parameters for the microdiagnostics test through the
                                ;parameter register

011226                                  BGNSUB
011230  004737  006626                  CALL    MVPAR                   ; move the parameters
011234  103002                          BCC     10$
011236                                  ESCAPE  TST
011242                          10$:    ENDSUB

                                ;SUBTEST 2 invokes the microdiagnostic test

011244                                  BGNSUB
011246  004737  006672                  CALL    DOMIC
011252  103002                          BCC     10$
011254                                  ESCAPE  TST
011260                          10$:    ENDSUB

011262                                  ENDTST
```

```
                                .SBTTL  TEST 10: INTERNAL EXCEPTIONS TEST


                                ;++
                                ; This test invokes the on-board INTERNAL EXCEPTIONS test.
                                ; If the HOE flag is specified the VSV21 will halt on error.
                                ; If the LOE flag is specified the operator will be given the choice on the
                                ; first pass of whether he wishes the on-board tests to loop on error. If he
                                ; selects yes the test may hang indefinitely as the on board will not return
                                ; control to the host but will continue looping.
                                ;--

011264                                  ST.TEST
011264  004737  007150                  CALL    STTEST
011270  103002                          BCC     30009$

011276  004737  006276                  CALL    TSTMSK                  ; get test mask for microdiagnostic
011302  004737  007054                  CALL    MRKONCE                 ; mark test as entered at least once

                                ;SUBTEST 1 moves the parameters for the microdiagnostics test through the
                                ;parameter register

011306                                  BGNSUB
011310  004737  006626                  CALL    MVPAR                   ; move the parameters
011314  103002                          BCC     10$
011316                                  ESCAPE  TST
011322                          10$:    ENDSUB

                                ;SUBTEST 2 invokes the microdiagnostic test

011324                                  BGNSUB
011326  004737  006672                  CALL    DOMIC
011332  103002                          BCC     10$
011334                                  ESCAPE  TST
011340                          10$:    ENDSUB

011342                                  ENDTST
```

```
2581
2582                                    .SBTTL  TEST 11: ACRCT INTERNAL TEST
2583
2584
2585                                    ;++
2586                                    ; This test invokes the on board ACRCT INTERNAL test.
2587                                    ; If the HOE flag is specified the VSV21 will halt on error.
2588                                    ; If the LOE flag is specified the operator will be given the choice on the
2589                                    ; first pass of whether he wishes the on board tests to loop on error. If he
2590                                    ; selects yes the test may hang indefinitely as the on board will not return
2591                                    ; control to the host but will continue looping.
2592                                    ;--
2593
2594 011344                                    ST.TEST
     011344  004737  007150                     CALL    STTEST
     011350  103002                             BCC     30010$
2595
2596 011356  004737  006276                     CALL    TSTMSK                  ; get test mask for microdiagnostic
2597 011362  004737  007054                     CALL    MRKONCE                 ; mark test as entered at least once
2598
2599                                    ;SUBTEST 1 moves the parameters for the microdiagnostics test through the
2600                                    ;parameter register
2601
2602 011366                                     BGNSUB
2603 011370  004737  006626                     CALL    MVPAR                   ; move the parameters
2604 011374  103002                             BCC     10$
2605 011376                                     ESCAPE  TST
2606 011402                             10$:    ENDSUB
2607
2608                                    ;SUBTEST 2 invokes the microdiagnostic test
2609
2610 011404                                     BGNSUB
2611 011406  004737  006672                     CALL    DOMIC
2612 011412  103002                             BCC     10$
2613 011414                                     ESCAPE  TST
2614 011420                             10$:    ENDSUB
2615
2616 011422                                     ENDTST
```

```
2618
2619                                    .SBTTL  TEST 12: ACRCT EXTERNAL TEST
2620
2621
2622                                    ;**
2623                                    ; This test invokes the on-board VSV21 ACRCT EXTERNAL test.
2624                                    ; If the HOE flag is specified the VSV21 will halt on error.
2625                                    ; If the LOE flag is specified the operator will be given the choice on the
2626                                    ; first pass of whether he wishes the on-board tests to loop on error. If he
2627                                    ; selects yes the test may hang indefinitely as the on board will not return
2628                                    ; control to the host but will continue looping.
2629                                    ; -
2630
2631 011424                                        ST.TEST
     011424  004737  007150                         CALL    STTEST
     011430  103002                                 BCC     30011$
2632
2633 011436  004737  006276                         CALL    TSTMSK                  ; get test mask for microdiagnostic
2634 011442  004737  007054                         CALL    MRKONCE                 ; mark test as entered at least once
2635
2636                                    ;SUBTEST 1 moves the parameters for the microdiagnostics test through the
2637                                    ;parameter register
2638
2639 011446                                         BGNSUB
2640 011450  004737  006626                         CALL    MVPAR                   ; move the parameters
2641 011454  103002                                 BCC     10$
2642 011456                                         ESCAPE  TST
2643 011462                             10$:        ENDSUB
2644
2645
2646                                    ;SUBTEST 2 invokes the microdiagnostic test
2647 011464
2648 011466  004737  006672                         BGNSUB
2649 011472  103002                                 CALL    DOMIC
2650 011474                                         BCC     10$
2651 011500                             10$:        ESCAPE  TST
2652                                                ENDSUB
2653 011502
                                                    ENDTST
```

```
 2655
 2656                                   .SBTTL  TEST 13: DUART BASIC TEST
 2657
 2658                                   ;++
 2659                                   ; This test invokes the on board VSV21 DUART BASIC test on ports 0-3.
 2660                                   ; If the HOE flag is specified the VSV21 will halt on error.
 2661                                   ; If the LOE flag is specified the operator will be given the choice on the
 2662                                   ; first pass of whether he wishes the on board tests to loop on error. If he
 2663                                   ; selects yes the test may hang indefinitely as the on board will not return
 2664                                   ; control to the host but will continue looping.
 2665                                   ;--
 2666
 2667 011504                                       ST.TEST
      011504  004737  007150                       CALL    STTEST
      011510  103002                               BCC     30012$
 2668
 2669 011516  004737  006276                       CALL    TSTMSK                  ; get test mask for microdiagnostic
 2670 011522  004737  007054                       CALL    MRKONCE                 ; mark test as entered at least once
 2671
 2672                                   ;FIRST PORT ==========-======-=========-==========-==-=======
 2673
 2674 011526  005037  002226                       CLR     PBLOC+4                 ; get port 0 into parameter block
 2675
 2676                                   ;SUBTEST 1 moves the parameters for the microdiagnostics test through the
 2677                                   ;parameter register
 2678
 2679 011532                                       BGNSUB
 2680 011534  004737  006626                       CALL    MVPAR                   ; move the parameters
 2681 011540  103002                               BCC     10$
 2682 011542                                       ESCAPE  TST
 2683 011546                           10$:        ENDSUB
 2684
 2685                                   ;SUBTEST 2 invokes the microdiagnostic test
 2686
 2687 011550                                       BGNSUB
 2688 011552  004737  006672                       CALL    DOMIC
 2689 011556  103002                               BCC     10$
 2690 011560                                       ESCAPE  TST
 2691 011564                           10$:        ENDSUB
 2692
 2693                                   ;SECOND PORT ==========-======-=========-==========-==-=======
 2694
 2695 011566  005237  002226                       INC     PBLOC+4                 ; get next port into parameter block
 2696
 2697                                   ;SUBTEST 3 moves the parameters for the microdiagnostics test through the
 2698                                   ;parameter register
 2699
 2700 011572                                       BGNSUB
 2701 011574  004737  006626                       CALL    MVPAR                   ; move the parameters
 2702 011600  103002                               BCC     10$
 2703 011602                                       ESCAPE  TST
 2704 011606                           10$:        ENDSUB
 2705
 2706                                   ;SUBTEST 4 invokes the microdiagnostic test
 2707
 2708 011610                                       BGNSUB
 2709 011612  004737  006672                       CALL    DOMIC
```

TEST 13: DUART BASIC TEST SEQ 0069

```
2710 011616  103002                    BCC     10$
2711 011620                             ESCAPE  TST
2712 011624                     10$:    ENDSUB
2713
2714                                    ;THIRD PORT ***********************************************************
2715
2716 011626  005237  002226             INC     PBLOC+4                 ; get next port into parameter block
2717
2718                                    ;SUBTEST 5 moves the parameters for the microdiagnostics test through the
2719                                    ;parameter register
2720
2721 011632                             BGNSUB
2722 011634  004737  006626             CALL    MVPAR                   ; move the parameters
2723 011640  103002                     BCC     10$
2724 011642                             ESCAPE  TST
2725 011646                     10$:    ENDSUB
2726
2727                                    ;SUBTEST 6 invokes the microdiagnostic test
2728
2729 011650                             BGNSUB
2730 011652  004737  006672             CALL    DOMIC
2731 011656  103002                     BCC     10$
2732 011660                             ESCAPE  TST
2733 011664                     10$:    ENDSUB
2734
2735                                    ;FOURTH PORT **********************************************************
2736
2737 011666  005237  002226             INC     PBLOC+4                 ; get next port into parameter block
2738
2739                                    ;SUBTEST 7 moves the parameters for the microdiagnostics test through the
2740                                    ;parameter register
2741
2742 011672                             BGNSUB
2743 011674  004737  006626             CALL    MVPAR                   ; move the parameters
2744 011700  103002                     BCC     10$
2745 011702                             ESCAPE  TST
2746 011706                     10$:    ENDSUB
2747
2748                                    ;SUBTEST 8 invokes the microdiagnostic test
2749
2750 011710                             BGNSUB
2751 011712  004737  006672             CALL    DOMIC
2752 011716  103002                     BCC     10$
2753 011720                             ESCAPE  TST
2754 011724                     10$:    ENDSUB
2755
2756
2757 011726                             ENDTST
```

```

                                   .SBTTL  TEST 14: DUART FULL TEST


                                   ;++
                                   ; This test invokes the on-board VSV21 DUART FULL test on ports 0-3.
                                   ; If the HOE flag is specified the VSV21 will halt on error.
                                   ; If the LOE flag is specified the operator will be given the choice on the
                                   ; first pass of whether he w'shes the on-board tests to loop on error. If he
                                   ; selects yes the test may hang indefinitely as the on board will not return
                                   ; control to the host but will continue looping.
                                   ;--

011730                                     ST.TEST
011730  004737  007150                     CALL    STTEST
011734  103002                             BCC     30013$

011742  004737  006276                     CALL    TSTMSK                  ; get test mask for microdiagnostic
011746  004737  007054                     CALL    MRKONCE                 ; mark test as entered at least once

                                   ;FIRST PORT ==============================================================

011752  005037  002226                     CLR     PBLOC+4                 ; get port 0 into parameter block

                                   ;SUBTEST 1 moves the parameters for the microdiagnostics test through the
                                   ;parameter reg'ster

011756                                     BGNSUB
011760  004737  006626                     CALL    MVPAR                   ; move the parameters
011764  103002                             BCC     10$
011766                                     ESCAPE  TST
011772                             10$:    ENDSUB

                                   ;SUBTEST 2 invokes the microdiagnostic test

011774                                     BGNSUB
011776  004737  006672                     CALL    DOMIC
012002  103002                             BCC     10$
012004                                     ESCAPE  TST
012010                             10$:    ENDSUB

                                   ;SECOND PORT =============================================================

012012  005237  002226                     INC     PBLOC+4                 ; get next port into parameter block

                                   ;SUBTEST 3 moves the parameters for the m'crodiagnostics test through the
                                   ;parameter reg'ster

012016                                     BGNSUB
012020  004737  006626                     CALL    MVPAR                   ; move the parameters
012024  103002                             BCC     10$
012026                                     ESCAPE  TST
012032                             10$:    ENDSUB

                                   ;SUBTEST 4 invokes the microdiagnostic test

012034                                     BGNSUB
```

SEQ 0071

TEST 14: DUART FULL TEST

```
2814 012036  004737  006672                CALL    DOMIC
2815 012042  103002                        BCC     10$
2816 012044                                ESCAPE  TST
2817 012050                         10$:   ENDSUB
2818
2819                                ;THIRD PORT ************************************************************
2820
2821 012052  005237  002226                INC     PBLOC+4                 ; get next port into parameter block
2822
2823                                ;SUBTEST 5 moves the parameters for the microdiagnostics test through the
2824                                ;parameter register
2825
2826 012056                                 BGNSUB
2827 012060  004737  006626                CALL    MVPAR                   ; move the parameters
2828 012064  103002                        BCC     10$
2829 012066                                ESCAPE  TST
2830 012072                         10$:   ENDSUB
2831
2832                                ;SUBTEST 6 invokes the microdiagnostic test
2833
2834 012074                                 BGNSUB
2835 012076  004737  006672                CALL    DOMIC
2836 012102  103002                        BCC     10$
2837 012104                                ESCAPE  TST
2838 012110                         10$:   ENDSUB
2839
2840                                ;FOURTH PORT ***********************************************************
2841
2842 012112  005237  002226                INC     PBLOC+4                 ; get next port into parameter block
2843
2844                                ;SUBTEST 7 moves the parameters for the microdiagnostics test through the
2845                                ;parameter register
2846
2847 012116                                 BGNSUB
2848 012120  004737  006626                CALL    MVPAR                   ; move the parameters
2849 012124  103002                        BCC     10$
2850 012126                                ESCAPE  TST
2851 012132                         10$:   ENDSUB
2852
2853                                ;SUBTEST 8 invokes the microdiagnostic test
2854
2855 012134                                 BGNSUB
2856 012136  004737  006672                CALL    DOMIC
2857 012142  103002                        BCC     10$
2858 012144                                ESCAPE  TST
2859 012150                         10$:   ENDSUB
2860
2861
2862 012152                                 ENDTST
```

TEST 14: DUART FULL TEST

```
                                .SBTTL  TEST 15: PERIPHERAL CONFIDENCE TEST


                                ;++
                                ; This test allows the operator to select a port for either input or output.
                                ; If he selects output an ASCII string is written to that port.
                                ; If he selects input the first 16 characters read from that port are displayed
                                ; in octal on the console running the tests.
                                ; If the HOE flag is specified the VSV21 will halt on error.
                                ; If the LOE flag is specified the operator will be given the choice on the
                                ; first pass of whether he wishes the on-board tests to loop on error. If he
                                ; selects yes the test may hang indefinitely as the on board will not return
                                ; control to the host but will continue looping.
                                ;--

 000010                                 T15RLEN=8.

 012154                                 ST.TEST
 012154  004737  007150                 CALL    STTEST
 012160  103002                         BCC     30014$

 012166                                 MAN.IGNORE                      ; ignore test if not manual

 012176                                 DMA.IGNORE                      ; ensure DMA enabled

 012220                                 GMANID  T15M1,PBLOC+4,D,177777,0,3,NO   ; get port number

 012240  004737  006276                 CALL    TSTMSK                  ; get test mask for microdiagnostic
 012244  004737  007054                 CALL    MRKONCE                 ; mark test as entered at least once

 012250                                 PRINTF  #T15M2
 012270                                 GMANIL  T15M3,T15INP,177777,NO  ; get whether input/output dev'ce
 012304  005737  013076                 TST     T15INP                  ; input device ?
 012310  001007                         BNE     30$                     ; branch if yes
 012312  052737  000001  002224         BIS     #1,PBLOC+2              ; say normal mode + wr'te
 012320  012737  000056  002230         MOV     #<T15WEN-T15WBUF>,PBLOC+6       ; put length to be DMA'd
 012326  000406                         BR      40$
 012330  052737  000002  002224 30$:    BIS     #2,PBLOC+2              ; say normal mode + read
 012336  012737  000020  002230         MOV     #<T15RLEN*2>,PBLOC+6            ; put length to be DMA'd

                                ;SUBTEST 1 DMAs test data into the VSV21 for an output dev'ce
 012344                         40$:
 012344                                 BGNSUB
 012346  005737  013076                 TST     T15INP                  ; input device ?
 012352  001013                         BNE     10$                     ; branch if yes
 012354                                 DO.WRAMS        #T15WBUF,#<T15WEN-T15WBUF>      ; DMA the data
 012354  012737  013020  002622         MOV     #T15WBUF,COMBUF+4
 012362  012737  000056  002624         MOV     #<T15WEN-T15WBUF>,COMBUF+6
 012370  004737  006066                 CALL    WRAMS
 012374  103002                         BCC     10$
 012376                                 ESCAPE  TST                     ; exit if DMA fails
 012402                         10$:    ENDSUB

                                ;SUBTEST 2 moves the parameters for the microdiagnostics test through the
                                ;parameter reg'ster
```

TEST 15: PERIPHERAL CONFIDENCE TEST SEQ 0073

```
2916 012404                                     BGNSUB
2917 012406  004737  006626                     CALL    MVPAR                   ; move the parameters
2918 012412  103002                             BCC     10$
2919 012414                                     ESCAPE  TST
2920 012420                             10$:    ENDSUB
2921
2922                                            ;SUBTEST 3 invokes the microdiagnostic test
2923
2924 012422                                     BGNSUB
2925 012424  004737  006672                     CALL    DOMIC
2926 012430  103002                             BCC     10$
2927 012432                                     ESCAPE  TST
2928 012436                             10$:    ENDSUB
2929
2930                                            ;SUBTEST 4 DMAs the data back from the VSV21 for an input dev'ce
2931
2932 012440                                     BGNSUB
2933 012442  005737  013076                     TST     T15INP                  ; input device ?
2934 012446  001441                             BEQ     30$                     ; branch if not
2935 012450                                     DO.RRAMS        #T15RBUF,#<T15RLEN*2>   ; DMA the data
     012450  012737  013000  002622             MOV     #T15RBUF,COMBUF+4
     012456  012737  000020  002624             MOV     #<T15RLEN*2>,COMBUF+6
     012464  004737  006132                     CALL    RRAMS
2936 012470  103002                             BCC     10$
2937 012472                                     ESCAPE  TST                     ; exit if DMA fails
2938 012476                             10$:    PRINTF  #T15M4                  ; print the data received
2939 012516  012701  000010                     MOV     #T15RLEN,R1             ; number to print
2940 012522  012702  013000                     MOV     #T15RBUF,R2             ; address of buffer
2941 012526                             20$:    PRINTF  #T15M5,(R2)+            ; print a word
2942 012550  077112                             SOB     R1,20$                  ; next one
2943 012552                             30$:    ENDSUB
2944
2945 012554                                     EXIT    TST
2946
2947 012560     120     114     105     T15M1:  .ASCIZ  /PLEASE ENTER PORT NUMBER/
2948 012611     045     116     045     T15M2:  .ASCIZ  /%N%AIS IT AN INPUT DEVICE ?/
2949 012645     120     114     105     T15M3:  .ASCIZ  /PLEASE ENTER "YES" FOR INPUT DEVICE OR "NO" FOR OUTPUT DEVICE/
2950 012743     045     116     045     T15M4:  .ASCIZ  /%N%ADATA RECEIVED IS : /
2951 012773     045     117     067     T15M5:  .ASCIZ  /%07/
2952                                            .EVEN
2953 013000                             T15RBUF:
2954 013000                                     .BLKW   T15RLEN
2955 013020                             T15WBUF:
2956 013020     041     100     043             .ASCIZ  /!@#$%^&*()_+1234567890QWERTYUIOP|Oasdfghjkl;'/
2957 013076                             T15WEN:
2958                                            .EVEN
2959 013076  000000                     T15INP: .WORD   0                       ; <>0 for i/p device , 0 for o/p device
2960
2961 013100                                     ENDTST
```

TEST 15: PERIPHERAL CONFIDENCE TEST

```
                                .SBTTL  TEST 16: INTERNAL LOOPBACK TEST


                                ;++
                                ; This test causes the on-board software to perform an INTERNAL LOOPBACK
                                ; test on ports 0 3.
                                ; If the HOE flag is specified the VSV21 will halt on error.
                                ; If the LOE flag is spec'fied the operator will be given the choice on the
                                ; first pass of whether he wishes the on-board tests to loop on error. If he
                                ; selects yes the test may hang indefinitely as the on board will not return
                                ; control to the host but w'll continue looping.
                                ;--

013102                                  ST.TEST
013102  004737  007150                  CALL    STTEST
013106  103002                          BCC     30017$

013114                                  DMA.IGNORE                      ; ensure DMA enabled

013136  004737  006276                  CALL    TSTMSK                  ; get test mask for microdiagnostic
013142  004737  007054                  CALL    MRKONCE                 ; mark test as entered at least once

013146  052737  000013  002224          BIS     #13,PBLOC+2             ; say local loop + read + write
013154  012737  000054  002230          MOV     #DMALEN,PBLOC+6         ; put length to be DMA'd

                                ;SUBTEST 1 DMAs test data into the VSV21

013162                                  BGNSUB
013164                                  DO.WRAMS                        ; DMA the data
013164  012737  002762  002622          MOV     #DMAOBUF,COMBUF+4
013172  012737  000054  002624          MOV     #DMALEN,COMBUF+6
013200  004737  006066                  CALL    WRAMS
013204  103002                          BCC     10$
013206                                  ESCAPE  TST                     ; exit if DMA fails
013212                          10$:    ENDSUB

                                ;FIRST PORT ============================================================

013214  005037  0022c6                  CLR     PBLOC+4                 ; get port 0 into parameter block

                                ;SUBTEST 2 moves the parameters for the microdiagnostics test through the
                                ;parameter register

013220                                  BGNSUB
013222  004737  006626                  CALL    MVPAR                   ; move the parameters
013226  103002                          BCC     10$
013230                                  ESCAPE  TST
013234                          10$:    ENDSUB

                                ;SUBTEST 3 invokes the microd'agnostic test

013236                                  BGNSUB
013240  004737  006672                  CALL    DOMIC
013244  103002                          BCC     10$
013246                                  ESCAPE  TST
013252                          10$:    ENDSUB
```

TEST 16: INTERNAL LOOPBACK TEST

```
3015
3016                                    ;SUBTEST 4 DMAs the data back from the VSV21
3017
3018 013254                                     BGNSUB
3019 013256                                     DO.RRAMS                        ; DMA the data
     013256  012737  003036  002622             MOV     #DMAIBUF,COMBUF+4
     013264  012737  000054  002624             MOV     #DMALEN,COMBUF+6
     013272  004737  006132                     CALL    RRAMS
3020 013276  103002                             BCC     10$
3021 013300                                     ESCAPE  TST                     ; exit if DMA fails
3022 013304                             10$:    ENDSUB
3023
3024                                    ;SUBTEST 5 Compares data written and read
3025
3026 013306                                     BGNSUB
3027 013310  004737  006212                     CALL    CRAMS                   ; DMA the data
3028 013314  103002                             BCC     10$
3029 013316                                     ESCAPE  TST                     ; exit if DMA fails
3030 013322                             10$:    ENDSUB
3031
3032                                    ;SECOND PORT ==========================================================
3033
3034 013324  005237  002226                     INC     PBLOC+4                 ; get next port into parameter block
3035
3036                                    ;SUBTEST 6 moves the parameters for the microdiagnostics test through the
3037                                    ;parameter register
3038
3039 013330                                     BGNSUB
3040 013332  004737  006626                     CALL    MVPAR                   ; move the parameters
3041 013336  103002                             BCC     10$
3042 013340                                     ESCAPE  TST
3043 013344                             10$:    ENDSUB
3044
3045                                    ;SUBTEST 7 invokes the microdiagnostic test
3046
3047 013346                                     BGNSUB
3048 013350  004737  006672                     CALL    DOMIC
3049 013354  103002                             BCC     10$
3050 013356                                     ESCAPE  TST
3051 013362                             10$:    ENDSUB
3052
3053                                    ;SUBTEST 8 DMAs the data back from the VSV21
3054
3055 013364                                     BGNSUB
3056 013366                                     DO.RRAMS                        ; DMA the data
     013366  012737  003036  002622             MOV     #DMAIBUF,COMBUF+4
     013374  012737  000054  002624             MOV     #DMALEN,COMBUF+6
     013402  004737  006132                     CALL    RRAMS
3057 013406  103002                             BCC     10$
3058 013410                                     ESCAPE  TST                     ; exit if DMA fails
3059 013414                             10$:    ENDSUB
3060
3061                                    ;SUBTEST 9 Compares data written and read
3062
3063 013416                                     BGNSUB
3064 013420  004737  006212                     CALL    CRAMS                   ; DMA the data
3065 013424  103002                             BCC     10$
```

SEQ 0076

TEST 16: INTERNAL LOOPBACK TEST

```
3066 013426                                     ESCAPE  TST                     ; exit if DMA fails
3067 013432                             10$:    ENDSUB
3068
3069                                    ;THIRD PORT ==-===-:=============================================-=========================
3070
3071 013434  005237  002226                     INC     PBLOC+4                 ; get next port into parameter block
3072
3073                                    ;SUBTEST 10 moves the parameters for the microd'agnostics test through the
3074                                    ,parameter register
3075
3076 013440                                     BGNSUB
3077 013442  004737  006626                     CALL    MVPAR                   , move the parameters
3078 013446  103002                             BCC     10$
3079 013450                                     ESCAPE  TST
3080 013454                             10$:    ENDSUB
3081
3082                                    ;SUBTEST 11 invokes the microdiagnostic test
3083
3084 013456                                     BGNSUB
3085 013460  004737  006672                     CALL    DOMIC
3086 013464  103002                             BCC     10$
3087 013466                                     ESCAPE  TST
3088 013472                             10$:    ENDSUB
3089
3090                                    ;SUBTEST 12 DMAs the data back from the VSV21
3091
3092 013474                                     BGNSUB
3093 013476                                     DO.RRAMS                        ; DMA the data
     013476  012737  003036  002622             MOV     #DMAIBUF,COMBUF+4
     013504  012737  000054  002624             MOV     #DMALEN,COMBUF+6
     013512  004737  006132                     CALL    RRAMS
3094 013516  103002                             BCC     10$
3095 013520                                     ESCAPE  TST                     ; exit if DMA fails
3096 013524                             10$:    ENDSUB
3097
3098                                    ;SUBTEST 13 Compares data written and read
3099
3100 013526                                     BGNSUB
3101 013530  004737  006212                     CALL    CRAMS                   ; DMA the data
3102 013534  103002                             BCC     10$
3103 013536                                     ESCAPE  TST                     ; exit if DMA fails
3104 013542                             10$:    ENDSUB
3105
3106                                    ;FOURTH PORT ***************************************************************
3107
3108 013544  005237  002226                     INC     PBLOC+4                 ; get next port into parameter block
3109
3110                                    ;SUBTEST 14 moves the parameters for the microd'agnostics test through the
3111                                    ;parameter register
3112
3113 013550                                     BGNSUB
3114 013552  004737  006626                     CALL    MVPAR                   ; move the parameters
3115 013556  103002                             BCC     10$
3116 013560                                     ESCAPE  TST
3117 013564                             10$:    ENDSUB
3118
3119                                    ;SUBTEST 15 invokes the microdiagnostic test
```

```
TEST 16: INTERNAL LOOPBACK TEST

   3120
   3121 013566                                  BGNSUB
   3122 013570  004737  006672                  CALL    DOMIC
   3123 013574  103002                          BCC     10$
   3124 013576                                  ESCAPE  TST
   3125 013602                          10$:    ENDSUB
   3126
   3127                                 ;SUBTEST 16 DMAs the data back from the VSV21
   3128
   3129 013604                                  BGNSUB
   3130 013606                                  DO.RRAMS                        ; DMA the data
        013606  012737  003036  002622          MOV     #DMAIBUF,COMBUF+4
        013614  012737  000054  002624          MOV     #DMALEN,COMBUF+6
        013622  004737  006132                  CALL    RRAMS
   3131 013626  103002                          BCC     10$
   3132 013630                                  ESCAPE  TST                     ; exit if DMA fails
   3133 013634                          10$:    ENDSUB
   3134
   3135                                 ;SUBTEST 17 Compares data written and read
   3136
   3137 013636                                  BGNSUB
   3138 013640  004737  006212                  CALL    CRAMS                   ; DMA the data
   3139 013644  103002                          BCC     10$
   3140 013646                                  ESCAPE  TST                     ; exit if DMA fails
   3141 013652                          10$:    ENDSUB
   3142
   3143
   3144 013654                                  ENDTST
```

TEST 16: INTERNAL LOOPBACK TEST

```
3146
3147                                    .SBTTL  TEST 17: EXTERNAL I/O LOOPBACK TEST
3148
3149
3150                                    ;++
3151                                    ; This test causes the on-board software to perform an EXTERNAL LOOPBACK
3152                                    ; test on ports 0-3.
3153                                    ; If the HOE flag is specified the VSV21 will halt on error.
3154                                    ; If the LOE flag is specified the operator will be given the choice on the
3155                                    ; first pass of whether he wishes the on-board tests to loop on error. If he
3156                                    ; selects yes the test may hang indefinitely as the on board will not return
3157                                    ; control to the host but will continue looping.
3158                                    ;--
3159
3160 013656                                       ST.TEST
     013656  004737  007150                        CALL    STTEST
     013662  103002                                BCC     30019$
3161
3162 013670                                       DMA.IGNORE                      ; ensure DMA enabled
3163
3164 013712  004737  006276                        CALL    TSTMSK                  ; get test mask for microdiagnostic
3165 013716  004737  007054                        CALL    MRKONCE                 ; mark test as entered at least once
3166
3167 013722  052737  000003  002224                BIS     #3,PBLOC+2              ; say normal mode + read + write
3168 013730  012737  000054  002230                MOV     #DMALEN,PBLOC+6         ; put length to be DMA'd
3169
3170                                    ;SUBTEST 1 DMAs test data into the VSV21
3171
3172 013736                                       BGNSUB
3173 013740                                       DO.WRAMS                        ; DMA the data
     013740  012737  002762  002622                MOV     #DMAOBUF,COMBUF+4
     013746  012737  000054  002624                MOV     #DMALEN,COMBUF+6
     013754  004737  006066                        CALL    WRAMS
3174 013760  103002                                BCC     10$
3175 013762                                       ESCAPE  TST                     ; exit if DMA fails
3176 013766                                    10$:    ENDSUB
3177
3178                                    ;FIRST PORT ************************************************************
3179
3180 013770  005037  002226                        CLR     PBLOC+4                 ; get port 0 into parameter block
3181
3182                                    ;SUBTEST 2 moves the parameters for the microdiagnostics test through the
3183                                    ;parameter register
3184
3185 013774                                       BGNSUB
3186 013776  004737  006626                        CALL    MVPAR                   ; move the parameters
3187 014002  103002                                BCC     10$
3188 014004                                       ESCAPE  TST
3189 014010                                    10$:    ENDSUB
3190
3191                                    ;SUBTEST 3 invokes the microdiagnostic test
3192
3193 014012                                       BGNSUB
3194 014014  004737  006672                        CALL    DOMIC
3195 014020  103002                                BCC     10$
3196 014022                                       ESCAPE  TST
3197 014026                                    10$:    ENDSUB
```

TEST 17: EXTERNAL I/O LOOPBACK TEST

```
3198
3199                                    ;SUBTEST 4 DMAs the data back from the VSV21
3200
3201 014030                                     BGNSUB
3202 014032                                     DO.RRAMS                        ; DMA the data
     014032  012737  003036  002622             MOV     @DMAIBUF,COMBUF+4
     014040  012737  000054  002624             MOV     @DMALEN,COMBUF+6
     014046  004737  006132                     CALL    RRAMS
3203 014052  103002                             BCC     10$
3204 014054                                     ESCAPE  TST                     ; exit if DMA fails
3205 014060                                10$: ENDSUB
3206
3207                                    ;SUBTEST 5 Compares data written and read
3208
3209 014062                                     BGNSUB
3210 014064  004737  006212                     CALL    CRAMS                   ; DMA the data
3211 014070  103002                             BCC     10$
3212 014072                                     ESCAPE  TST                     ; exit if DMA fails
3213 014076                                10$: ENDSUB
3214
3215                                    ;SECOND PORT ***************************************************************
3216
3217 014100  005237  002226                     INC     PBLOC+4                 ; get next port into parameter block
3218
3219                                    ;SUBTEST 6 moves the parameters for the microdiagnostics test through the
3220                                    ;parameter register
3221
3222 014104                                     BGNSUB
3223 014106  004737  006626                     CALL    MVPAR                   ; move the parameters
3224 014112  103002                             BCC     10$
3225 014114                                     ESCAPE  TST
3226 014120                                10$: ENDSUB
3227
3228                                    ;SUBTEST 7 invokes the microdiagnostic test
3229
3230 014122                                     BGNSUB
3231 014124  004737  006672                     CALL    DOMIC
3232 014130  103002                             BCC     10$
3233 014132                                     ESCAPE  TST
3234 014136                                10$: ENDSUB
3235
3236                                    ;SUBTEST 8 DMAs the data back from the VSV21
3237
3238 014140                                     BGNSUB
3239 014142                                     DO.RRAMS                        ; DMA the data
     014142  012737  003036  002622             MOV     @DMAIBUF,COMBUF+4
     014150  012737  000054  002624             MOV     @DMALEN,COMBUF+6
     014156  004737  006132                     CALL    RRAMS
3240 014162  103002                             BCC     10$
3241 014164                                     ESCAPE  TST                     ; exit if DMA fails
3242 014170                                10$: ENDSUB
3243
3244                                    ;SUBTEST 9 Compares data written and read
3245
3246 014172                                     BGNSUB
3247 014174  004737  006212                     CALL    CRAMS                   ; DMA the data
3248 014200  103002                             BCC     10$
```

TEST 1?: EXTERNAL I/O LOOPBACK TEST SEQ 0080

```
3249 014202                              ESCAPE  TST                     ; exit if DMA fails
3250 014206                         10$: ENDSUB
3251
3252                                ;THIRD PORT ********** ---------------------------------------------------
3253
3254 014210  005237  002226              INC     PBLOC+4                 ; get next port into parameter block
3255
3256                                ;SUBTEST 10 moves the parameters for the microdiagnostics test through the
3257                                ;parameter register
3258
3259 014214                              BGNSUB
3260 014216  004737  006626              CALL    MVPAR                   ; move the parameters
3261 014222  103002                      BCC     10$
3262 014224                              ESCAPE  TST
3263 014230                         10$: ENDSUB
3264
3265                                ;SUBTEST 11 invokes the microdiagnostic test
3266
3267 014232                              BGNSUB
3268 014234  004737  006672              CALL    DOMIC
3269 01423C  103002                      BCC     10$
3270 014242                              ESCAPE  TST
3271 014246                         10$: ENDSUB
3272
3273                                ;SUBTEST 12 DMAs the data back from the VSV21
3274
3275 014250                              BGNSUB
3276 014252                              DO.RRAMS                        ; DMA the data
     014252  012737  003036  002622      MOV     #DMAIBUF,COMBUF+4
     014260  012737  000054  002624      MOV     #DMALEN,COMBUF+6
     014266  004737  006132              CALL    RRAMS
3277 014272  103002                      BCC     10$
3278 014274                              ESCAPE  TST                     ; exit if DMA fails
3279 014300                         10$: ENDSUB
3280
3281                                ;SUBTEST 13 Compares data written and read
3282
3283 014302                              BGNSUB
3284 014304  004737  006212              CALL    CRAMS                   ; DMA the data
3285 014310  103002                      BCC     10$
3286 014312                              ESCAPE  TST                     ; exit if DMA fails
3287 014316                         10$: ENDSUB
3288
3289                                ;FOURTH PORT ********** ---------- ---------------------------------------
3290
3291 014320  005237  002226              INC     PBLOC+4                 ; get next port into parameter block
3292
3293                                ;SUBTEST 14 moves the parameters for the microdiagnostics test through the
3294                                ;parameter register
3295
3296 014324                              BGNSUB
3297 014326  004737  006626              CALL    MVPAR                   ; move the parameters
3298 014332  103002                      BCC     10$
3299 014334                              ESCAPE  TST
3300 014340                         10$: ENDSUB
3301
3302                                ;SUBTEST 15 invokes the microdiagnostic test
```

SEQ 0081

TEST 17: EXTERNAL I/O LOOPBACK TEST

```
3303
3304 014342                                    BGNSUB
3305 014344  004737  006672                    CALL    DOMIC
3306 014350  103002                            BCC     10$
3307 014352                                    ESCAPE  TST
3308 014356                            10$:    ENDSUB
3309
3310                                   ;SUBTEST 16 DMAs the data back from the VSV21
3311
3312 014360                                    BGNSUB
3313 014362                                    DO.RRAMS                        ; DMA the data
     014362  012737  003036  002622            MOV     #DMAIBUF,COMBUF+4
     014370  012737  000054  002624            MOV     #DMALEN,COMBUF+6
     014376  004737  006132                    CALL    RRAMS
3314 014402  103002                            BCC     10$
3315 014404                                    ESCAPE  TST                     ; exit if DMA fails
3316 014410                            10$:    ENDSUB
3317
3318                                   ;SUBTEST 17 Compares data written and read
3319
3320 014412                                    BGNSUB
3321 014414  004737  006212                    CALL    CRAMS                   ; DMA the data
3322 014420  103002                            BCC     10$
3323 014422                                    ESCAPE  TST                     ; exit if DMA fails
3324 014426                            10$:    ENDSUB
3325
3326
3327 014430                                    ENDTST
```

TEST 18: SCREEN TEST

```
3329                                    .SBTTL  TEST 18: SCREEN TEST
3330
3331
3332                                    ;++
3333                                    ; This test displays different screen test pictures as selected by the
3334                                    ; operator. If the HOE flag is specified the VSV21 will halt on error.
3335                                    ; If the LOE flag is specified the operator will be g'ven the choice on the
3336                                    ; first pass of whether he wishes the on-board tests to loop on error. If he
3337                                    ; selects yes the test may hang indefin'tely as the on board will not return
3338                                    ; control to the host but will continue looping
3339                                    ;--
3340
3341 014432                                       ST.TEST
     014432  004737  007150                        CALL    STTEST
     014436  103002                                BCC     30021$
3342
3343 014444                                       MAN.IGNORE                      ; ignore test if not manual
3344
3345 014454                                       DMA.IGNORE                      ; ensure DMA enabled
3346
3347 014476  004737  006276                        CALL    TSTMSK                  ; get testmask
3348 014502  004737  007054                        CALL    MRKONCE                 ; mark test as entered at least once
3349
3350                                    ;SUBTEST 1 DMAs the operator requested p'cture to the VSV21
3351
3352 014506                                       BGNSUB
3353 014510                                       PRINTF  #T18M1
3354 014530                                       GMANID  T18M2,TEMP1,D,177777,1,4,NO ; get current p'cture number
3355 014550  013700  002212                        MOV     TEMP1,R0                ; and save it
3356 014554  005300                                DEC     R0                      ; get in range 0 - 3
3357 014556  006300                                ASL     R0                      ; get byte offset
3358 014560                                       DO.WRAMS        PICS(R0),#<PIC2 PIC1>   ; write the picture
     014560  016037  014712  002622                MOV     PICS(R0),COMBUF+4
     014566  012737  000042  002624                MOV     #<PIC2-PIC1>,COMBUF+6
     014574  004737  006066                        CALL    WRAMS
3359 014600  103002                                BCC     20$
3360 014602                                       ESCAPE  TST
3361 014606                                20$:    ENDSUB
3362
3363                                    ;SUBTEST 2 moves the parameters for the microdiagnost'cs test tnrough the
3364                                    ;parameter register
3365
3366 014610                                       BGNSUB
3367 014612  012737  000042  002226                MOV     #<PIC2-PIC1>,PBLOC+4    ; th'rd parameter 's length of picture
3368 014620  004737  006626                        CALL    MVPAR                   ; move the parameters
3369 014624  103002                                BCC     10$
3370 014626                                       ESCAPE  TST
3371 014632                                10$:    ENDSUB
3372
3373                                    ;SUBTEST 3 invokes the microdiagnostic test
3374
3375 014634                                       BGNSUB
3376 014636  004737  006672                        CALL    DOMIC
3377 014642  103002                                BCC     10$
3378 014644                                       ESCAPE  TST
3379 014650                                10$:    ENDSUB
3380
```

SEQ 0083

TEST 18: SCREEN TEST

```
3381 014652                                    PRINTF  #T18M3                    ; Print message
3382 014672                                    GMANIL  T18M4,TEMP1,177777,YES    ; and wait.
3383
3384 014706                                    EXIT    TST
3385
3386 014712  014722  014764  015026  PICS:    .WORD   PIC1,PIC2,PIC3,PIC4
3387 014722  014001  000001  177777  PIC1:    .WORD   014001,000001,177777,004005,000000,004006,000000,004007
3388 014742  000360  004014  040000           .WORD   000360,004014,040000,004015,000000,054000,021042,000531
3389 014762  177001                           .WORD   177001
3390 014764  014001  000001  177777  PIC2:    .WORD   014001,000001,177777,004005,000000,004006,000000,004007
3391 015004  000360  004014  040000           .WORD   000360,004014,040000,004015,000000,054000,042104,000531
3392 015024  177001                           .WORD   177001
3393 015026  014001  000001  177777  PIC3:    .WORD   014001,000001,177777,004005,000000,004006,000000,004007
3394 015046  000360  004014  040000           .WORD   000360,004014,040000,004015,000000,054000,10421,000531
3395 015066  177001                           .WORD   177001
3396 015070  014001  000001  177777  PIC4:    .WORD   014001,000001,177777,004005,000000,004006,000000,004007
3397 015110  000360  004014  040000           .WORD   000360,004014,040000,004015,000000,054000,177777,000531
3398 015130  177001                           .WORD   177001
3399
3400 015132     045     116     045  T18M1:   .ASCIZ  /%N%APLEASE INPUT PICTURE NUMBER WHERE :/
3401 015202     122     105     104  T18M2:   .ASCIZ  /RED , GREEN , BLUE AND WHITE SCREENS ARE 1,2,3,4 RESP./
3402 015271     045     116     045  T18M3:   .ASCIZ  /%N%APLEASE ENTER CARRIAGE RETURN WHEN YOU HAVE FINISHED/
3403 015361     127     111     124  T18M4:   .ASCIZ  /WITH THE PICTURE/
3404
3405                                           .EVEN
3406
3407 015402                                    ENDTST
```

```
                                .SBTTL  TEST 19: NVRAM READ/WRITE TEST


                                ;++
                                ; This test tests NVRAM read/write. On each pass the operator will be asked if
                                ; he wishes to continue as NVRAM has a limited life in terms of read/write
                                ; cycles.
                                ; If the HOE flag is specified the VSV21 will halt on error.
                                ; If the LOE flag is specified the operator will be given the choice on the
                                ; first pass of whether he wishes the on-board tests to loop on error. If he
                                ; selects yes the test may hang indefinitely as the on board will not return
                                ; control to the host but will continue looping.
                                ;--

015404                                  ST.TEST
015404  004737  007150                  CALL    STTEST
015410  103002                          BCC     30024$

015416                                  MAN.IGNORE                      ; ignore test if not manual

015426  004737  006276                  CALL    TSTMSK                  ; get test mask for microdiagnostic
015432  004737  007054                  CALL    MRKONCE                 ; mark test as entered at least once

015436                                  PRINTF  #T20M1                  ; Ask if operator
015456                                  GMANIL  T20M2,TEMP1,177777,NO   ; wants to do test.
015472  005737  002212                  TST     TEMP1                   ; operator want to do test ?
015476  001002                          BNE     20$                     ; branch if yes
015500                                  EXIT    TST                     ; else exit if no
015504                          20$:

                                ;SUBTEST 1 moves the parameters for the microdiagnostics test through the
                                ;parameter register

015504                                  BGNSUB
015506  004737  006626                  CALL    MVPAR                   ; move the parameters
015512  103002                          BCC     10$
015514                                  ESCAPE  TST
015520                          10$:    ENDSUB

                                ;SUBTEST 2 invokes the microdiagnostic test

015522                                  BGNSUB
015524  004737  006672                  CALL    DOMIC
015530  103002                          BCC     10$
015532                                  ESCAPE  TST
015536                          10$:    ENDSUB

015540                                  ENDTST

015542     045     116     045  T20M1:  .ASCIZ  /%N%A** WARNING ** WRITING TO NVRAM REDUCES IT'S LIFE/
015627     104     117     040  T20M2:  .ASCIZ  /DO YOU STILL WANT TO DO THIS TEST ?/

                                        .EVEN
```

TEST 19: NVRAM READ/WRITE TEST

```
                                .SBTTL  TEST 20: ENABLE INTERRUPTS TEST


                                ;++
                                ; This test verifies that interrupts are received by the host when interrupts
                                ; are enabled.
                                ;--

015674                                  ST.TEST
015674  004737  007150                  CALL    STTEST
015700  103002                          BCC     30026$

015706                                  SETVEC  CURVEC,#INTVEC,#240     ; enable the interrupt service routine
                                ;                                       ; and g've it priority #5
015734  005037  002240                  CLR     INTFLG                  ; say no interrupts have been received

                                ;SUBTEST 1 enables interrupts and expects to get an interrupt while do'ng it

015740                                  BGNSUB
015742                                  DO.COMMAND      PI.IEN,10$      ; enable interrupts
015742  012737  001403  002614          MOV     #PI.IEN,COMAND
015750  012737  002734  002216          MOV     #1500.,POLCNT
015756  004737  004572                  CALL    DOCOM
015762  103001                          BCC     10$
015764  000437                          BR      20$                     ; branch on error
015766  012701  002734          10$:    MOV     #1500.,R1               ; count to poll
015772  005737  002240          15$:    TST     INTFLG                  ; have we received an interrupt ?
015776  001032                          BNE     20$                     ; branch if yes
016000                                  DELAY   10.                     ; delay 10 units
016030  077120                          SOB     R1,15$                  ; again
016032                                  ERR.FILLIN      15.,ERR15               ; time out
016032  012737  000002  002202          MOV     #2,ERRTYP       ; assume hard error for now
016040  012737  000017  002204          MOV     #15.,ERRNBR     ; get error 15.
016046  012737  004177  002206          MOV     #ERR15,ERRMSG   ; get error ERR15
016054  012737  004302  002210          MOV     #ERRCODE,ERRBLK ; get error routine
016062                                  ERROR
016064                          20$:    ENDSUB

                                ;SUBTEST 2 disables interrupts

016066                                  BGNSUB
016070                                  DO.COMMAND      PI.IDS,10$      ; disable interrupts
016070  012737  001400  002614          MOV     #PI.IDS,COMAND
015076  012737  002734  002216          MOV     #1500.,POLCNT
016104  004737  004572                  CALL    DOCOM
016110  103000                          BCC     10$
016112                          10$:    CLRVEC  CURVEC                  ; disable the interrupt service routine
016120                                  ENDSUB

016122                                  ENDTST

                                ;Interrupt service routine

0161 4                                  BGNSRV  INTVEC
016124  012737  000001  002240          MOV     #1,INTFLG               ; say received interrupt
016132                                  ENDSRV
```

TEST 20: ENABLE INTERRUPTS TEST

```
3506
3507                                    .TITLE PARAMETER CODING
3508
3509                                    .SBTTL  HARDWARE PARAMETER CODING SECTION
3510
3511
3512                                    ;++
3513                                    ; THE HARDWARE PARAMETER CODING SECTION CONTAINS MACROS
3514                                    ; THAT ARE USED BY THE SUPERVISOR TO BUILD P-TABLES.  THE
3515                                    ; MACROS ARE NOT EXECUTED AS MACHINE INSTRUCTIONS BUT ARE
3516                                    ; INTERPRETED BY THE SUPERVISOR AS DATA STRUCTURES.  THE
3517                                    ; MACROS ALLOW THE SUPERVISOR TO ESTABLISH COMMUNICATIONS
3518                                    ; WITH THE OPERATOR.
3519                                    ;--
3520
3521 016134                                    BGNHRD
3522
3523 016136                                    GPRMA   DEVADD,0,0,160000,177776,YES
3524
3525 016146                                    GPRMA   VECADD,2,0,0,776,YES
3526
3527 016156                                    ENDHRD
3528
3529 016156     104     105     126  DEVADD: .ASCIZ  /DEVICE ADDRESS/
3530 016175     126     105     103  VECADD: .ASCIZ  /VECTOR ADDRESS/
3531
3532                                    .EVEN
3533
3534
3535 016214                            $PATCH::
3536 016214                                    .BLKW   50.
3537
3538 016360                                    LASTAD
     016364                            L$LAST::
3539 016364                                    ENDMOD
```

HARDWARE PARAMETER CODING SECTION SEQ 0087

```
3541 016364                                BGNSETUP        4
3542 016364                                BGNPTAB
3543 016370  172010                        .WORD   172010
3544 016372  000320                        .WORD   320
3545 016374                                ENDPTAB
3546 016374                                BGNPTAB
3547 016400  172020                        .WORD   172020
3548 016402  000324                        .WORD   324
3549 016404                                ENDPTAB
3550 016404                                BGNPTAB
3551 016410  172030                        .WORD   172030
3552 016412  000330                        .WORD   330
3553 016414                                ENDPTAB
3554 016414                                BGNPTAB
3555 016420  172040                        .WORD   172040
3556 016422  000334                        .WORD   334
3557 016424                                ENDPTAB
3558 016424                                ENDSETUP
3559         000001                .END
```

Symbol table

```
ABORT   007510          C$ERDF= 000055  DMAOBU  002762 G  GETST2  005462 G  L$CO    002032 G
ADR   = 000020 G        C$ERHR= 000056  DMASET= 000002 G  G$CNTO= 000200     L$DEPO  002011 G
ASSEMB= 000010          C$ERRO= 000060  DOCOM   004572 G  G$DELM= 000372     L$DESC  004260 G
BITS    002244 G        C$ERSF= 000054  DOMIC   006672 G  G$DISP= 000003     L$DESP  002076 G
BIT0  = 000001 G        C$ERSO= 000057  EF.CON= 000036 G  G$EXCP= 000400     L$DEVP  002060 G
BIT00 = 000001 G        C$ESCA= 000010  EF.NEW= 000035 G  G$HILI= 000002     L$DISP  002124 G
BIT01 = 000002 G        C$ESEG= 000005  EF.PWR= 000034 G  G$LOLI= 000001     L$DLY   002116 G
BIT02 = 000004 G        C$ESUB= 000003  EF.RES= 000037 G  G$NO  = 000000     L$DTP   002040 G
BIT03 = 000010 G        C$ETST= 000001  EF.STA= 000040 G  G$OFFS= 000400     L$DTYP  002034 G
BIT04 = 000020 G        C$EXIT= 000032  END     007504    G$OFSI= 000376     L$DUT   002072 G
BIT05 = 000040 G        C$FREQ= 000101  ERRBLK  002210 G  G$PRMA= 000001     L$DVTY  004251 G
BIT06 = 000100 G        C$FRME= 000100  ERRCOD  004302 G  G$PRMD= 000002     L$EF    002052 G
BIT07 = 000200 G        C$GETB= 000026  ERRMSG  002206 G  G$PRML= 000000     L$ENVI  002044 G
BIT08 = 000400 G        C$GETW= 000027  ERRNBR  002204 G  G$RADA= 000140     L$ERRT  002202 G
BIT09 = 001000 G        C$GMAN= 000043  ERRTYP  002202 G  G$RADB= 000000     L$ETP   002102 G
BIT1  = 000002 G        C$GPHR= 000042  ERR1    003166 G  G$RADD= 000040     L$EXP1  002046 G
BIT10 = 002000 G        C$GPRI= 000040  ERR10   003720 G  G$RADL= 000120     L$EXP4  002064 G
BIT11 = 004000 G        C$INIT= 000011  ERR11   003753 G  G$RADO= 000020     L$EXP5  002066 G
BIT12 = 010000 G        C$INLP= 000020  ERR12   004000 G  G$XFER= 000004     L$HARD  016136 G
BIT13 = 020000 G        C$MANI= 000050  ERR13   004065 G  G$YES = 000010     L$HIME  002120 G
BIT14 = 040000 G        C$MAP = 000102  ERR14   004130 G  HELP  = 000000     L$HPCP  002016 G
BIT15 = 100000 G        C$MEM = 000031  ERR15   004177 G  HOE   = 100000 G  L$HPTP  002022 G
BIT2  = 000004 G        C$MMU = 000103  ERR2    003225 G  IBE   = 010000 G  L$HW    002176 G
BIT3  = 000010 G        C$MSG = 000023  ERR3    003274 G  IDU   = 000040 G  L$ICP   002104 G
BIT4  = 000020 G        C$OPNR= 000034  ERR4    003345 G  IER   = 020000 G  L$INIT  007310 G
BIT5  = 000040 G        C$OPNW= 000104  ERR5    003425 G  INTFLG  002240 G  L$LADP  002026 G
BIT6  = 000100 G        C$PNTB= 000014  ERR6    003465 G  INTVEC  016124 G  L$LAST  016364 G
BIT7  = 000200 G        C$PNTF= 000017  ERR7    003541 G  ISR   = 000100 G  L$LOAD  002100 G
BIT8  = 000400 G        C$PNTS= 000016  ERR8    003611 G  IXE   = 004000 G  L$LUN   002074 G
BIT9  = 001000 G        C$PNTX= 000015  ERR9    003652 G  I$AU  = 000041    L$MREV  002050 G
BOE   = 000400 G        C$PUTB= 000072  EVL   = 000004 G  I$AUTO= 000041    L$NAME  002000 G
BUFDES  002604          C$PUTW= 000073  E$END = 002100    I$CLN = 000041    L$PRIO  002042 G
BUFOK   010374          C$QIO = 000377  E$LOAD= 000035    I$DU  = 000041    L$PROT  007302 G
COMAND  002014          C$RDBU= 000007  FLAGS   002304 G  I$HRD = 000041    L$PRT   002112 G
COMBUF  002616          C$REFG= 000047  FTHRU = 000004 G  I$INIT= 000041    L$REPP  002062 G
COMLEN= 000024          C$REL = 000077  F$AU  = 000015    I$MOD = 000041    L$REV   002010 G
CRAMS   006212 G        C$RESE= 000033  F$AUTO= 000020    I$MSG = 000041    L$SPC   002056 G
CURCSR  002232 G        C$REVI= 000004  F$BGN = 000040    I$PROT= 000040    L$SPCP  002020 G
CURPAR  002234 G        C$RFLA= 000021  F$CLEA= 000007    I$PTAB= 000041    L$SPTP  002024 G
CURVEC  002236 G        C$RPT = 000025  F$DU  = 000016    I$PWR = 000041    L$STA   002030 G
C$AU  = 000052          C$SEFG= 000046  F$END = 000041    I$RPT = 000041    L$TEST  002114 G
C$AUTO= 000061          C$SPRI= 000041  F$HARD= 000004    I$SEG = 000041    L$TIML  002014 G
C$BRK = 000022          C$SVEC= 000037  F$HW  = 000013    I$SETU= 000041    L$UNIT  002012 G
C$BSEG= 000004          C$TOME= 000076  F$INIT= 000006    I$SRV = 000041    L10000  002202
C$BSUB= 000002          DEVADD  016156  F$JMP = 000050    I$SUB = 000041    L10001  004312
C$CLCK= 000062          DFPTBL  002176 G  F$MOD = 000000  I$TST = 000041    L10003  007512
C$CLEA= 000012          DIAGMC= 000000    F$MSG = 000011  J$JMP = 000167    L10004  007514
C$CLOS= 000035          DI.ERL= 000177 G  F$PROT= 000021  LENFLA= 000030 G  L10005  007554
C$CLP1= 000006          DI.ERP= 000005 G  F$PWR = 000017  LOE   = 040000 G  L10006  010236
C$CPBF= 000074          DI.ERS= 000176 G  F$RPT = 000012  LOGUNT  002220 G  L10007  007630
C$CPME= 000075          DI.MIC= 000011 G  F$SEG = 000003  LOT   = 000010 G  L10010  010010
C$CVEC= 000036          DI.RDA= 002101 G  F$SOFT= 000005  L$ACP   002110 G  L10011  010506
C$DCLN= 000044          DI.RRA= 002103 G  F$SRV = 000010  L$APT   002036 G  L10012  010312
C$DODU= 000051          DI.WDD= 002102 G  F$SUB = 000002  L$AUT   002070 G  L10013  010340
C$DRPT= 000024          DI.WRA= 002104 G  F$SW  = 000014  L$AUTO  007514 G  L10014  010372
C$DU  = 000053          DMAIBU  003036 G  F$TEST= 000001  L$CCP   002106 G  L10015  010430
C$EDIT= 000000          DMALEN= 000054    GETST1  005066 G  L$CLEA  007516 G  L10016  010462
```

Symbol table

L10017	010500	L10110	013304	NEXT	007376	TS1	010016	T12.1	011446		
L10020	010622	L10111	013322	NOTEST	002242 G	TS2	010046	T12.2	011464		
L10021	010620	L10112	013344	O$APTS=	000000	TS3	010137	T13	011504 G		
L10022	010702	L10113	013362	O$AU =	000000	T$ARGC=	000001	T13.1	011532		
L10023	010662	L10114	013414	O$BGNR=	000000	T$CODE=	001031	T13.2	011550		
L10024	010700	L10115	013432	O$BGNS=	000000	T$ERRN=	000002	T13.3	011572		
L10025	010762	L10116	013454	O$DU =	000000	T$EXCP=	000000	T13.4	011610		
L10026	010742	L10117	013472	O$ERRT=	000001	T$FLAG=	000040	T13.5	011632		
L10027	010760	L10120	013524	O$GNSW=	000000	T$FREE=	016424	T13.6	011650		
L10030	011042	L10121	013542	O$POIN=	000001	T$GMAN=	000000	T13.7	011672		
L10031	011022	L10122	013564	O$SETU=	000001	T$HILI=	000776	T13.8	011710		
L10032	011040	L10123	013602	PBLOC	002222 G	T$LAST=	000001	T14	011730 G		
L10033	011122	L10124	013634	PICS	014712	T$LOLI=	000000	T14.1	011756		
L10034	011102	L10125	013652	PIC1	014722	T$LSYM=	010000	T14.2	011774		
L10035	011120	L10126	014430	PIC2	014764	T$LTNO=	000024	T14.3	012016		
L10036	011202	L10127	013766	PIC3	015026	T$NEST=	177777	T14.4	012034		
L10037	011162	L10130	014010	PIC4	015070	T$NS0 =	000000	T14.5	012056		
L10040	011200	L10131	014026	PI.ACK=	003000 G	T$NS1 =	000004	T14.6	012074		
L10041	011262	L10132	014060	PI.AVA=	100000 G	T$NS2 =	000002	T14.7	012116		
L10042	011242	L10133	014076	PI.CAN=	001001 G	T$PCNT=	000000	T14.8	012134		
L10043	011260	L10134	014120	PI.CMA=	002400 G	T$PTAB=	010173	T15	012154 G		
L10044	011342	L10135	014136	PI.IDM=	000400 G	T$PTHV=	000004	T15INP	013076		
L10045	011322	L10136	014170	PI.IDS=	001400 G	T$PTNU=	000004	T15M1	012560		
L10046	011340	L10137	014206	PI.IEN=	001403 G	T$SAVL=	177777	T15M2	012611		
L10047	011422	L10140	014230	PI.RUB=	177777 G	T$SEGL=	177777	T15M3	012645		
L10050	011402	L10141	014246	PI.SEN=	001000 G	T$SIZE=	000020	T15M4	012743		
L10051	011420	L10142	014300	PI.TST=	002000 G	T$SUBN=	000002	T15M5	012773		
L10052	011502	L10143	014316	PNT =	001000 G	T$TAGL=	177777	T15RBU	013000		
L10053	011462	L10144	014340	POLCNT	002216 G	T$TAGN=	010175	T15RLE=	000010		
L10054	011500	L10145	014356	POLL	004314 G	T$TEMP=	000000	T15WBU	013020		
L10055	011726	L10146	014410	POLLCR	004512 G	T$TEST=	000024	T15WEN	013076		
L10056	011546	L10147	014426	PRI =	002000 G	T$TSTM=	177777	T15.1	012344		
L10057	011564	L10150	015402	PRI00 =	000000 G	T$TSTS=	000001	T15.2	012404		
L10060	011606	L10151	014606	PRI01 =	000040 G	T$$AUT=	010004	T15.3	012422		
L10061	011624	L10152	014632	PRI02 =	000100 G	T$$CLE=	010005	T15.4	012440		
L10062	011646	L10153	014650	PRI03 =	000140 G	T$$DAT=	010174	T16	013102 G		
L10063	011664	L10154	015540	PRI04 =	000200 G	T$$HAR=	010163	T16.1	013162		
L10064	011706	L10155	015520	PRI05 =	000240 G	T$$HW =	010000	T16.10	013440		
L10065	011724	L10156	015536	PRI06 =	000300 G	T$$INI=	010003	T16.11	013456		
L10066	012152	L10157	016122	PRI07 =	000340 G	T$$MSG=	010001	T16.12	013474		
L10067	011772	L10160	016064	RRAMS	006132 G	T$$PC =	000004	T16.13	013526		
L10070	012010	L10161	016120	STABUF	002666	T$$PRO=	010002	T16.14	013550		
L10071	012032	L10162	016132	STALEN=	000036	T$$PTA=	010173	T16.15	013566		
L10072	012050	L10163	016156	STTEST	007150 G	T$$SRV=	010162	T16.16	013604		
L10073	012072	L10164	016370	SVCGBL=	000000	T$$SUB=	010161	T16.17	013636		
L10074	012110	L10165	016400	SVCINS=	177777	T$$TES=	010157	T16.2	013220		
L10075	012132	L10166	016374	SVCSUB=	177777	T1	007556 G	T16.3	013236		
L10076	012150	L10167	016410	SVCTAG=	177777	T1DONE=	000000 G	T16.4	013254		
L10077	013100	L10170	016404	SVCTST=	177777	T1 1	007570	T16.5	013306		
L10100	012402	L10171	016420	S$LSYM=	010000	T1.2	007632	T16.6	013330		
L10101	012420	L10172	016414	TEMP1	002212 G	T10	011264 G	T16.7	013346		
L10102	012436	L10174	016424	TEMP2	002214 G	T10.1	011306	T16.8	013364		
L10103	012552	MAXTST=	000024 G	TMASKS=	000010 G	T10.2	011324	T16.9	013416		
L10104	013654	MAXUNI=	000004 G	TM1	006444	T11	011344 G	T17	013656 G		
L10105	013212	MICNUM	007232	TM2	006541	T11.1	011366	T17.1	013736		
L10106	013234	MRKONC	007054 G	TSTMSK	006276 G	T11.2	011404	T17.10	014214		
L10107	013252	MVPAR	006626 G	TSTONC	007112 G	T12	011424 G	T17.11	014232		

Symbol table

```
T17.12  014250          T18     014432 G        T2.3    010342          T5      010704 G        T9.2    011244
T17.13  014302          T18M1   015132          T2.4    010374          T5.1    010726          UAM   = 000200 G
T17.14  014324          T18M2   015202          T2.5    010432          T5.2    010744          VECADD  016175
T17.15  014342          T18M3   015271          T2.6    010464          T6      010764 G        VS.CRY= 010000 G
T17.16  014360          T18M4   015361          T20     015674 G        T6.1    011006          VS.ERR= 040000 G
T17.17  014412          T18.1   014506          T20M1   015542          T6.2    011024          VS.PRY= 020000 G
T17.2   013774          T18.2   014610          T20M2   015627          T7      011044 G        VS.SVL= 100000 G
T17.3   014012          T18.3   014634          T20.1   015740          T7.1    011066          WRAMS   006066 G
T17.4   014030          T19     015404 G        T20.2   016066          T7.2    011104          X$ALWA= 000000
T17.5   014062          T19.1   015504          T3      010510 G        T8      011124 G        X$FALS= 000040
T17.6   014104          T19.2   015522          T3.1    010522          T8.1    011146          X$OFFS= 000400
T17.7   014122          T2      010240 G        T4      010624 G        T8.2    011164          X$TRUE= 000020
T17.8   014140          T2.1    010252          T4.1    010646          T9      011204 G        $PATCH  016214 G
T17.9   014172          T2.2    010314          T4.2    010664          T9.1    011226

. ABS.  016424    000   (RW,I,GBL,ABS,OVR)
        000000    001   (RW,I,LCL,REL,CON)
Errors detected:  0

*** Assembler statistics


Work  file  reads: 303
Work  file writes: 304
Size of work file: 35760 Words  ( 140 Pages)
Size of core pool: 19714 Words  ( 75 Pages)
Operating  system: RSX-11M/PLUS (Under VAX/VMS)

Elapsed time: 00:07:25.81
ZVSWA0,ZVSWA0/ SP=SVC40.MLB/ML,ZVSWA0.MAC
```