

DHV 11

DHV-11 FUNC TST PART 1
CVDHABO

COPYRIGHT (c) 1983
AH-T653B-MC
FICHE 1 OF 2

JAN 1984
digital
Made In USA

DHV11

DHV-11 FUNC TST PART1
CYDHABO

COPYRIGHT (c) 1983
AH-T653B-MC
FICHE 2 OF 2

JAN 1984
digital
Made In USA

.REF 6

IDENTIFICATION

PRODUCT CODE: AC T652B MC
PRODUCT NAME: CVDHABO DMV 11 FUNC TST PART1
PRODUCT DATE: 09 OCTOBER 1983
MAINTAINER: EDSME DIAGNOSTICS GROUP
AUTHOR: BERT KLEINSCHMIDT
TONY GRIMSHAW
MODIFIED BY: BERT KLEINSCHMIDT
ANTHONY HART

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

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

COPYRIGHT (C) 1983 BY DIGITAL EQUIPMENT CORPORATION

THE FOLLOWING ARE TRADEMARKS OF DIGITAL EQUIPMENT CORPORATION:

DIGITAL
DEC

PDP
DECJS

UNIBUS
DECTAPE

MASSBUS

***** MODIFICATION HISTORY *****

ORIGINAL RELEASE: 31 OCT 83 (EDITED 11 JUL 83)
BERT KLEINSCHMIDT

VERSION B0 09 OCT-83 BERT KLEINSCHMIDT
FIXED TYPOGRAPHICAL ERRORS.
MOVED TEST FROM CVDHB (PART 2) INTO THIS PROGRAM:
OLD CVDHB (VERSION A) TESTS 2 THROUGH 8 ARE
NOW NEW CVDHA (VERSION B) TESTS 20 THROUGH 26.
MODIFIED TEST 13 (SELFTEST FAIL TEST) TO ALLOW USE OF CVDHA
IN MANUFACTURING WITH VERSION 0 DHV ROM CODE.

TABLE OF CONTENTS

1.0	GENERAL PROGRAM CONSIDERATIONS
1.1	PROGRAM ABSTRACT
1.2	SYSTEM REQUIREMENTS
1.3	RELATED DOCUMENTS AND STANDARDS
1.4	DIAGNOSTIC HIERARCHY PREREQUISITES
2.0	OPERATING INSTRUCTIONS
2.1	COMMANDS
2.2	SWITCHES
2.3	FLAGS
2.4	EXTENDED COMMAND SYNTAX
2.4.1	START COMMAND
2.4.1.1	TESTS SWITCH (/TESTS:<TEST-LIST>)
2.4.1.2	PASS SWITCH (/PASS:<PASS-CNT>)
2.4.1.3	FLAGS SWITCH (/FLAGS:<FLAG-LIST>)
2.4.1.4	END OF PASS SWITCH (/EOP:<INCR>)
2.4.1.5	EFFECT OF START COMMAND
2.4.2	RESTART COMMAND
2.4.2.1	TESTS, PASS, AND FLAGS SWITCHES
2.4.2.2	UNITS SWITCH (/UNITS:<UNIT-LIST>)
2.4.2.3	EFFECT OF RESTART COMMAND
2.4.3	CONTINUE COMMAND
2.4.3.1	FLAG SWITCH (/FLAGS:<FLAG LIST>)
2.4.3.2	EFFECT OF CONTINUE COMMAND
2.4.4	PROCEED COMMAND
2.4.4.1	FLAGS SWITCH (/FLAGS:<FLAG-LIST>)
2.4.4.2	EFFECT OF PROCEED COMMAND
2.4.5	ADD COMMAND
2.4.6	EFFECT OF ADD COMMAND
2.4.7	DROP COMMAND
2.4.8	EFFECT OF DROP COMMAND
2.4.9	PRINT COMMAND
2.4.9.1	EFFECT OF PRINT COMMAND
2.4.10	DISPLAY COMMAND
2.4.10.1	EFFECT OF DISPLAY COMMAND
2.4.11	FLAGS COMMAND
2.4.11.1	EFFECT OF FLAGS COMMAND
2.4.12	ZFLAGS COMMAND
2.4.13	ZFLAGS COMMAND
2.4.14	CONTROL CHARACTERS
2.5	HARDWARE QUESTIONS
2.6	SOFTWARE QUESTIONS
2.7	EXTENDED P-TABLE DIALOGUE
2.8	QUICK START-UP PROCEDURE (XXDP.)
3.0	ERROR INFORMATION
3.1	TYPES OF ERROR MESSAGES
3.2	ERROR MESSAGES
4.0	PERFORMANCE AND PROGRESS REPORTS
5.0	TEST SUMMARIES
6.0	EXAMPLE ERROR FREE PASS

1.0 GENERAL PROGRAM CONSIDERATIONS

1.1 PROGRAM ABSTRACT

CVDHA IS PART ONE OF THE DHV-11 FUNCTIONAL VERIFICATION TEST. THIS PART OF THE TEST VERIFIES THAT THE RESET, REGISTER ACCESS, AND INTERRUPT FUNCTIONS OF THE BOARD ARE FUNCTIONING CORRECTLY.

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 THE OPERATING INSTRUCTIONS-COMMANDS OF THIS DOCUMENT.

1.2 SYSTEM REQUIREMENTS

THE FOLLOWING HARDWARE IS REQUIRED TO RUN THE DHV FVT:

- 0 LSI-11 PROCESSOR WITH AT LEAST 32 KBYTES OF RAM.
- 0 DHV11 BOARDS INSTALLED ON THE Q-BUS.
- 0 APPROPRIATE PROGRAM LOAD DEVICE SUPPORTING XXDP+ MEDIA OR A DOWN LINE LOADING SYSTEM.

1.3 RELATED DOCUMENTS AND STANDARDS

- 0 DHV 11 HARDWARE MANUAL - THIS MANUAL DESCRIBES THE FUNCTIONS AND USES OF THE DHV 11 DEVICE.
- 0 XXDP+ USER'S MANUAL - DESCRIBES THE RUNNING OF DIAGNOSTICS UNDER THE XXDP+ MONITOR.

1.4 DIAGNOSTIC HIERARCHY PREREQUISITES

THE LSI-11 PROCESSOR, THE Q-BUS, THE SYSTEM MEMORY, THE CONSOLE TERMINAL, AND THE LOAD MEDIA ARE ASSUMED TO HAVE BEEN TESTED AND FOUND WORKING BEFORE THIS PROGRAM IS RUN.

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 - SEE PERFORMANCE AND PROGRESS REPORTS SECTION OF THIS DOCUMENT)
DISPLAY	TYPE A LIST OF ALL DEVICE INFORMATION
FLAGS	TYPE THE STATE OF ALL FLAGS (SEE FLAGS SECTION)
ZFLAGS	CLEAR ALL FLAGS (SEE FLAGS SECTION)

A COMMAND CAN BE RECOGNIZED BY THE FIRST THREE CHARACTERS. SO YOU MAY, FOR EXAMPLE, TYPE "STA" INSTEAD OF "START". MORE INFORMATION CAN BE FOUND WITHIN THE SECTION LABELLED EXTENDED COMMAND SYNTAX

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.
EXECUTE DDDDD PASSES (DDDDD = 1 TO 64000)
SET SPECIFIED FLAGS. SEE THE FLAGS SECTION
OF THIS DOCUMENT.
REPORT END OF PASS MESSAGE AFTER EVERY
DDDDD PASSES ONLY. (DDDDD = 1 TO 64000)
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)

/PASS:DDDDD
/FLAGS:FLGS
/EOP:DDDDD
/UNITS:LIST

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)

*SEE THE ERROR INFORMATION SECTION OF THIS DOCUMENT.

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 EXTENDED COMMAND SYNTAX

2.4.1 START COMMAND -

```
*****  
STA(RT)/TESTS:<TEST-LIST>/PASS:<PASS-CNT>/FLAGS:  
  <FLAG-LIST>/EOP:<INCR>  
*****
```

2.4.1.1 TESTS SWITCH (/TESTS:<TEST-LIST>)

<TEST-LIST> IS A SEQUENCE OF DECIMAL NUMBERS (1;2 ETC.) OR RANGES OF DECIMAL NUMBERS (1-5;8-10 ETC.), SEPERATED BY COLONS, THAT SPECIFY THE TESTS TO BE EXECUTED. TESTS WILL BE EXECUTED IN NUMERICAL ORDER REGARDLESS OF THE ORDER OF SPECIFICATION. THE DEFAULT IS TO EXECUTE ALL TESTS. ON THIS AND ALL SWITCHES, THE ANGLE BRACKETS <> ARE PUNCTUATION USED IN THE DEFINITION ONLY, AND ARE NOT TO BE TYPED BY THE OPERATOR. SEE EXAMPLE AT END OF "EFFECT OF START COMMAND" SECTION.

2.4.1.2 PASS SWITCH (/PASS:<PASS-CNT>)

<PASS-CNT> IS A DECIMAL NUMBER INDICATING THE DESIRED NUMBER OF PASSES. A PASS IS DEFINED AS THE EXECUTION OF THE FULL DIAGNOSTIC (ALL SELECTED TESTS). THE DEFAULT IS NON-ENDING EXECUTION. IN THIS CASE, EXIT FROM THE PROGRAM IS ACCOMPLISHED EITHER BY TYPING A CONTROL/C OR BY OCCURANCE OF AN ERROR WITH THE HALT ON ERROR FLAG BEING SET. THE EXIT IS A RETURN TO COMMAND MODE. SEE EXAMPLE AT END OF "EFFECT OF START COMMAND" SECTION.

2.4.1.3 FLAGS SWITCH (/FLAGS:<FLAG-LIST>) -

<FLAG-LIST> IS A SEQUENCE OF ELEMENTS OF THE FORM <FLAG>, <FLAG=1>, OR <FLAG=C>, SEPARATED BY COLONS, WHERE <FLAG> HAS ONE OF THE FOLLOWING VALUES:

HOE	HALT ON ERROR, CAUSING COMMAND MODE TO BE ENTERED WHEN AN ERROR IS ENCOUNTERED.
LOE	LOOP ON ERROR, CAUSING THE DIAGNOSTIC TO LOOP CONTINUOUSLY WITHIN THE SMALLEST DEFINED BLOCK OF CODING (SEGMENT, SUBTEST, OR TEST) CONTAINING THE ERROR.
IER	INHIBIT ERROR REPORTING.
IBE	INHIBIT BASIC ERROR REPORTS.
IXE	INHIBIT EXTENDED ERROR REPORTS.
PRI	DIRECT ALL MESSAGES TO A LINE PRINTER.
PNT	PRINT NUMBER OF TEST BEING EXECUTED.
BOE	BELL ON ERROR.
UAM	RUN IN UNATTENDED MODE, BYPASSING MANUAL

INTERVENTION.
ISR INHIBIT STATISTICAL REPORTS.
IDU INHIBIT DROPPING OF UNITS BY DIAGNOSTIC.
LOT LOOP ON TEST.
THE FLAGS NAMED OR EQUATED TO 1 ARE SET, THOSE EQUATED TO 0
ARE CLEARED. A FLAG NOT SPECIFIED IS CLEARED. IF THE FLAGS
SWITCH IS NOT GIVEN ALL FLAGS ARE CLEARED. SEE EXAMPLE AT
END OF "EFFECT OF START COMMAND" SECTION.

2.4.1.4 END OF PASS SWITCH (/EOP:<INCR>)

<INCR> IS A DECIMAL NUMBER INDICATING HOW OFTEN (IN
TERMS OF PASSES) IT IS DESIRED THAT THE END OF PASS MESSAGE
BE PRINTED. THE DEFAULT IS AT THE END OF EVERY PASS. SEE
EXAMPLE AT END OF "EFFECT OF START COMMAND" SECTION.

2.4.1.5 EFFECT OF START COMMAND -

THE EFFECT OF THE START COMMAND IS TO INITIATE THE
HARDWARE PARAMETER DIALOGUE, THE SOFTWARE PARAMETER
DIALOGUE, THE INITIALIZATION QUESTIONS, AND THEN THE
DIAGNOSTIC COMMENCES TESTING.

THE HARDWARE PARAMETER DIALOGUE COMMENCES WITH THE
QUESTION "# UNITS (D) ?" TO WHICH THE OPERATOR SHOULD REPLY
WITH THE NUMBER OF UNITS TO BE TESTED. FOLLOWING THIS ARE
THE QUESTIONS WHEREBY THE P-TABLES THEMSELVES ARE BUILT.
EACH P-TABLE IS A CORE-RESIDENT TABLE CONTAINING ALL THE
HARDWARE INFORMATION FOR ONE COMPLETE UNIT. EACH QUESTION IS
FOLLOWED BY THE RESPONSE RADIX (D FOR DECIMAL, B FOR BINARY,
O FOR OCTAL, L FOR YES/NO) IN PARENTHESES AND THE DEFAULT
VALUE AFTER THE PARENTHESES. FOR THE ACTUAL HARDWARE
P TABLE QUESTIONS SEE THE "HARDWARE PARAMETERS" SECTION.

FOLLOWING THE HARDWARE QUESTIONS ARE THE SOFTWARE
QUESTIONS TO BUILD THE SOFTWARE TABLES, WHICH DEFINE
OPERATING PARAMETERS OF THE DIAGNOSTIC PROGRAM. THESE
QUESTIONS ARE DESCRIBED IN THE "SOFTWARE PARAMETERS"
SECTION.

EXAMPLE:

STA/TESTS:1:3-4:/PASS:3/FLAGS:IER:HOE=1

THIS COMMAND WILL CAUSE THREE PASSES TO BE MADE, WITH
EACH PASS CONSISTING OF TESTS 1,3, AND 4. THERE IS NO
DIFFERENCE BETWEEN SAYING <FLAG> AND SAYING <FLAG=1>. THE
NOTATION <FLAG=0> IS MEANINGFUL ONLY ON A COMMAND OTHER THAN
START TO CLEAR A FLAG THAT WAS PREVIOUSLY SET. NOTE THAT ON
ALL COMMANDS ONLY THE FIRST THREE LETTERS ARE SCANNED.

2.4.2 RESTART COMMAND -

RES(TART)/TESTS:<TEST-LIST>/PASS:<PASS-CNT>/FLAGS:
<FLAG-LIST>/UNITS:<UNIT-LIST>

2.4.2.1 TESTS, PASS, AND FLAGS SWITCHES -

<TEST LIST>, <PASS-CNT>, AND <FLAG-LIST> ARE AS IN THE
START COMMAND.

2.4.2.2 UNITS SWITCH (/UNITS:<UNIT-LIST>) - <UNIT-LIST> IS
A SEQUENCE OF DECIMAL NUMBERS (0,1 ETC.) OR RANGES OF
DECIMAL NUMBERS (0-5, 8-10 ETC.) THAT SPECIFY THE UNITS TO
BE TESTED. THE NUMBERS ARE SEPARATED BY COLONS. THE
NUMBERS MAY RANGE FROM 0 THRU N 1 (N IS THE NUMBER OF UNITS
SPECIFIED IN THE PREVIOUS START COMMAND). THE NUMBER
INDICATES THE POSITION OF THE P-TABLE AS THE DATA WAS
ENTERED DURING THE HARDWARE DIAGLOGUE. THE UNITS WHICH ARE
SELECTED MUST NOT HAVE BEEN DROPPED BY THE DROP COMMAND.
SEE THE DISCUSSION OF ADD AND DROP COMMANDS BELOW. DEFAULT
IS TO TEST ALL UNITS WHICH HAVE NOT BEEN DROPPED BY A DROP
COMMAND.

2.4.2.3 EFFECT OF RESTART COMMAND -

THE RESTART COMMAND DIFFERS FROM THE START COMMAND IN
THAT THE P-TABLES FROM THE PREVIOUS START COMMAND (THERE
MUST HAVE BEEN ONE) ARE USED, INSTEAD OF NEW ONES BEING
BUILT. THE SOFTWARE DIALOGUE MAY OPTIONALLY BE REEXECUTED
(OPERATOR WILL BE ASKED). THE COMMAND CAN BE USED AFTER
COMMAND MODE HAS BEEN REENTERED IN ANY OF THE THREE NORMAL
WAYS: A) THE REQUESTED NUMBER OF PASSES HAVE BEEN MADE, B)
AN ERROR WAS ENCOUNTERED WITH THE HALT ON ERROR FLAG SET, OR
C) A CONTROL/C WAS ENTERED BY THE OPERATOR.

2.4.3 CONTINUE COMMAND -

CON(TINUE)/PASS:<PASS-CNT>/FLAGS:<FLAG-LIST>

2.4.3.1 FLAG SWITCH (/FLAGS:<FLAG-LIST>)

<FLAG-LIST> IS SAME AS IN THE START COMMAND, BUT UNSPECIFIED FLAGS RETAIN THEIR CURRENT VALUE.

2.4.3.2 EFFECT OF CONTINUE COMMAND -

CONTINUE MUST FOLLOW A START OR RESTART, AND COMMAND MODE MUST HAVE BEEN ENTERED DUE TO A HALT ON ERROR OR A CONTROL/C. THE EFFECT OF THE COMMAND IS TO GO TO THE BEGINNING OF THE TEST THAT WAS BEING EXECUTED WHEN THE HALT OR CONTROL/C TOOK PLACE. SOFTWARE DIALOGUE MAY OPTIONALLY BE REEXECUTED. HARDWARE PARAMETERS MAY NOT BE CHANGED.

2.4.4 PROCEED COMMAND -

PRO(CEED)/FLAGS:<FLAG-LIST>

2.4.4.1 FLAGS SWITCH (/FLAGS:<FLAG-LIST>) -

<FLAG-LIST> IS AS IN THE START COMMAND, BUT UNSPECIFIED FLAGS RETAIN THEIR CURRENT VALUE.

2.4.4.2 EFFECT OF PROCEED COMMAND -

PROCEED MUST FOLLOW A START, RESTART, OR CONTINUE. COMMAND MODE MUST HAVE BEEN ENTERED VIA A HALT ON ERROR. THE EFFECT OF THE COMMAND IS TO BEGIN EXECUTION AT THE LOCATION FOLLOWING THE ERROR CALL. NEITHER HARDWARE NOR SOFTWARE PARAMETERS MAY BE ALTERED.

2.4.5 ADD COMMAND -

ADD/UNITS:<UNIT-LIST>

2.4.6 EFFECT OF ADD COMMAND - THE UNITS SPECIFIED ARE ADDED TO THE TEST SEQUENCE. EACH UNIT MUST HAVE A P-TABLE IN MEMORY DUE TO AN EARLIER HARDWARE DIALOGUE. THIS COMMAND MUST BE FOLLOWED BY A RESTART OR CONTINUE. THE UNITS SWITCH MUST BE SPECIFIED. THE ADD COMMAND IS MEANINGFUL ONLY FOR UNITS THAT WERE PREVIOUSLY DROPPED.

2.4.7 DROP COMMAND

DRO(P)/UNITS:<UNIT-LIST>

2.4.8 EFFECT OF DROP COMMAND - THE UNITS SPECIFIED WILL BE DROPPED FROM TESTING. THE UNITS WILL BE RESELECTED ONLY BY THE EXECUTION OF AN ADD OR START COMMAND. THE UNITS SWITCH MUST BE ENTERED. THIS COMMAND MUST BE FOLLOWED BY A RESTART OR A CONTINUE COMMAND.

2.4.9 PRINT COMMAND -

PRI(NT)

2.4.9.1 EFFECT OF PRINT COMMAND - ERROR SUMMARY REPORTING IS NOT IMPLEMENTED IN THIS DIAGNOSTIC, SO THIS COMMAND HAS NO EFFECT.

2.4.10 DISPLAY COMMAND -

DIS(PLAY)/UNITS:<UNIT LIST>

2.4.10.1 EFFECT OF DISPLAY COMMAND -

THE HARDWARE P-TABLES FOR ALL UNITS ARE PRINTED IN THE FORMAT IN WHICH THEY WERE ENTERED.

2.4.11 FLAGS COMMAND -

FLA(GS)

2.4.11.1 EFFECT OF FLAGS COMMAND -

THE CURRENT SETTINGS OF ALL FLAGS ARE PRINTED.

2.4.12 ZFLAGS COMMAND -

ZFL(AGS)

2.4.13 ZFLAGS COMMAND

ALL FLAGS ARE CLEARED.

2.4.14 CONTROL CHARACTERS

- C A CONTROL/C (C) ENTERED DURING THE EXECUTION OF A DIAGNOSTIC CAUSES A RETURN TO COMMAND MODE.
- Z A CONTROL/Z (Z) ENTERED DURING ONE OF THE TWO OPERATOR DIALOGUES-- HARDWARE P-TABLE DIALOGUE OR SOFTWARE P-TABLE DIALOGUE CAUSES THE DEFAULTS TO BE TAKEN FOR THE REMAINDER OF THAT DIALOGUE.
- O A CONTROL/O (O) ENTERED DURING THE EXECUTION OF A DIAGNOSTIC CAUSES ALL TELETYPE OUTPUT TO BE SUPPRESSED FOR THE REMAINDER OF THE DIAGNOSTIC OR UNTIL ANOTHER CONTROL/O IS TYPED, WHICH RESTORES NORMAL TELETYPE OUTPUT.

2.5 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). 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:

1. CSR ADDRESS - THIS QUESTION REQUESTS THE CSR ADDRESS OF THE SPECIFIED DHV11. THE DEFAULT ANSWER FOR THIS QUESTION IS THE LOWEST ADDRESS IN THE PDP 11 FLOATING ADDRESS SPACE IN WHICH A DHV-11 CAN BE PLACED (160460 OCTAL).
2. INTERRUPT VECTOR ADDRESS - THIS QUESTION REQUESTS THE INTERRUPT VECTOR ADDRESS OF THE SPECIFIED DHV11.
3. ACTIVE LINES BIT MAP - THIS QUESTION REQUESTS AN OCTAL BIT MAP OF THE SERIAL COMMUNICATION LINES ON THE DHV11 WHICH ARE BEING SELECTED FOR TESTING. IF THE BIT IN THE BIT MAP IS SET WHICH CORRESPONDS TO A PARTICULAR LINE (I.E. BIT 3 FOR LINE 3) THAT LINE WILL BE TESTED BY THE FVT.
4. BR LEVEL - THIS QUESTIONS REQUESTS THE INTERRUPT BR LEVEL OF THE DHV11.

2.6 SOFTWARE QUESTIONS

AFTER YOU HAVE ANSWERED THE HARDWARE QUESTIONS OR AFTER A RESTART OR CONTINUE COMMAND, THE RUNTIME SERVICES WILL ASK FOR SOFTWARE PARAMETERS. THESE PARAMETERS WILL GOVERN SOME DIAGNOSTIC SPECIFIC OPERATION MODES. YOU WILL BE PROMPTED BY "CHANGE SW (L) ?" IF YOU WISH TO CHANGE ANY PARAMETERS, ANSWER BY TYPING "Y". THE FOLLOWING SOFTWARE P-TABLE QUESTIONS ARE ASKED BY THE PROGRAM IF THE OPERATOR INDICATES THAT THE SOFTWARE PARAMETERS ARE TO BE CHANGED.

1. REPORT UNIT NUMBER AS EACH UNIT IS TESTED - THIS QUESTION ASKS WHETHER THE PROGRAM SHOULD REPORT THE NUMBER OF THE UNIT WHICH IT IS TESTING AS IT BEGINS TO TEST EACH UNIT.
2. NUMBER OF INDIVIDUAL DATA ERRORS TO REPORT ON A LINE - THIS QUESTION ASKS FOR THE NUMBER OF DATA ERRORS WHICH SHOULD BE REPORTED INDIVIDUALLY BY THIS PROGRAM FOR EACH LINE FOR EACH TRANSMISSION TEST. ERRORS WHICH ARE NOT REPORTED INDIVIDUALLY ARE REPORTED IN SUMMARY ERROR REPORTS.
3. ROM VERSION PRINTOUT ON THE FIRST PASS - THIS QUESTION ASKS WHETHER THE PROGRAM SHOULD PRINT OUT THE VERSIONS OF THE ON BOARD 8051 PROCESSOR ROMS DURING THE FIRST PASS OF THE PROGRAM.

2.7 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 (0) ? 8<CR>

UNIT 1

CSR ADDRESS (0) ? 160000<CR>

SUB-DEVICE # (0) ? 0<CR>

Q FACTOR (0) 0 ? 1<CR>

UNIT 2

CSR ADDRESS (0) ? 160000<CR>

SUB DEVICE # (0) ? 1<CR>

Q FACTOR (0) 1 ? 0<CR>

UNIT 3

CSR ADDRESS (0) ? 160000<CR>

SUB DEVICE # (0) ? 2<CR>

Q FACTOR (0) 0 ? <CR>

.
.
.

UNIT 7

CSR ADDRESS (0) ? 160000<CR>

SUB-DEVICE # (0) ? 6<CR>

Q-FACTOR (0) 0 ? 1<CR>

UNIT 8

CSR ADDRESS (0) 160000<CR>

SUB-DEVICE # (0) ? 7<CR>

Q FACTOR (0) 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 (0) ? 8<CR>

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

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

UNIT 7
CSR ADDRESS (0) ? 160000<CR>
SUB-DEVICE # (0) ? 6,7<CR>
Q-FACTOR (0) 0 ? 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 (0) ? 8<CR>

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

2.8 QUICK START UP PROCEDURE (XXDP.)

TO START UP THIS PROGRAM:

1. BOOT XXDP.
2. GIVE THE DATE AND ANSWER THE LSI AND 50HZ (IF THERE IS A CLOCK AND THE QUESTION IS ASKED) QUESTIONS
3. TYPE 'R NAME', WHERE NAME IS THE NAME OF THE BIN OR BIC FILE FOR THIS PROGRAM
4. TYPE 'START'
5. ANSWER THE "CHANGE HW" QUESTION WITH "Y"
6. ANSWER ALL THE HARDWARE QUESTIONS
7. ANSWER THE "CHANGE SW" QUESTION WITH "N"

WHEN YOU FOLLOW THIS PROCEDURE YOU WILL BE USING ONLY THE DEFAULTS FOR FLAGS AND SOFTWARE PARAMETERS. FOR DEFAULT INFORMATION SEE THE SECTIONS WITHIN THIS DOCUMENT ON FLAGS, AND HARDWARE QUESTIONS.

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 (SEE THE FLAGS SECTION OF THIS DOCUMENT).

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 (SEE THE FLAGS SECTION OF THIS DOCUMENT).

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 (SEE THE

FLAGS SECTION OF THIS DOCUMENT).
THESE MESSAGES ARE PRINTED AFTER THE ASSOCIATED GENERAL ERROR
MESSAGE AND ANY ASSOCIATED BASIC ERROR MESSAGES.

3.2 ERROR MESSAGES

THIS PROGRAM IS INTENDED TO PROVIDE A GO/NO-GO INDICATION OF THE
FUNCTIONALITY OF DHV 11 BOARDS. TO EXECUTE THE PROGRAM IN THIS MODE
THE OPERATOR CAN RUN WITH THE INHIBIT BASIC ERROR REPORTING SWITCH.
IN THIS MODE THE PROGRAM PRINTS ERROR MESSAGES WHICH CONTAIN THE ERROR
MESSAGE HEADER DESCRIBED ABOVE, PLUS THE NAME OF THE FAILING TEST.
FOR A LIST OF THE TEST NAMES IN THIS PROGRAM SEE THE TEST SUMMARIES
SECTION OF THIS DOCUMENT. AN EXAMPLE OF SUCH AN ERROR MESSAGE IS THE
FOLLOWING:

CVDHA DVC FTL ERR 01603 ON UNIT 02 TST 015 SUB 000 PC: 015244
DEVICE REGISTER WORD READ/WRITE TEST

THIS ERROR INDICATES THAT A FATAL ERROR WAS ENCOUNTERED WITHIN THE
TEST WHICH TESTS THE READ/WRITE CAPABILITY OF THE DHV 11 REGISTERS.

IF THE OPERATOR REQUIRES MORE EXTENSIVE ERROR REPORTING HE CAN RUN
WITH ALL ERROR REPORTING ENABLED BY NOT USING THE INHIBIT REPORTING
SWITCHES. THE ABOVE ERROR MESSAGE WOULD THEN BECOME THE FOLLOWING:

CVDHA DVC FTL ERR 01603 ON UNIT 02 TST 015 SUB 000 PC: 015244
DEVICE REGISTER WORD READ/WRITE TEST
BAD BIT(S) IN DEVICE TBUFFAD1 REGISTER FOR LINE 7 (D).
EXPECTED DATA: 000000 (0).
ACTUAL DATA: 000023 (0).

4.0 PERFORMANCE AND PROGRESS REPORTS

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

5.0 TEST SUMMARIES

THE FOLLOWING TESTS ARE INCLUDED WITHIN CVDMA:

1. DEVICE REGISTER ADDRESS TEST - VERIFIES THAT THE UUT REGISTERS WILL RESPOND WITH THE PROPER Q-BUS HANDSHAKING WHEN ACCESSED. VERIFIES THAT THE UUT IS AT THE PROPER ADDRESS.
2. MASTER.RESET (SELFTEST) TEST - VERIFIES THAT THE MASTER.RESET BIT CLEARS WITHIN A SPECIFIED TIME OF IT BEING SET.
3. MASTER.RESET (SKIP SELFTEST) TEST - VERIFIES THAT THE MASTER.RESET BIT CLEARS WITHIN A SHORT TIME AFTER IT IS SET IF THE SKIP SELFTEST SEQUENCE IS USED.
4. RX.CHARACTER FIELD TEST - VERIFIES THAT THE DATA BITS OF THE CODES IN THE FIFO AFTER A RESET AND SKIP SELFTEST ARE CONSISTANT WITH THE SKIP SELFTEST CODES.
5. RECEPTION FLAG FIELD TEST - VERIFIES THAT THE 3 DATA STATUS BITS (OVERRUN, FRAMING, AND PARITY ERROR BITS) ARE ALL SET ON ALL OF THE SKIP SELFTEST CODES IN THE FIFO AFTER A RESET AND SKIP SELFTEST SEQUENCE.
6. RX.DATA.AVAIL TEST - VERIFIES THAT THE RX.DATA.AVAIL BIT IS SET WHEN THE SKIP SELFTEST CODES ARE IN THE FIFO AND THAT IT CLEARS AFTER THEY ARE READ.
7. RX.DATA.VALID TEST - VERIFIES THAT THE RX.DATA.VALID BIT IS SET FOR EACH VALID SKIP SELFTEST CODE IN THE FIFO AND CLEAR AFTER ALL VALID CODES ARE READ.
8. RX.LINE FIELD TEST - VERIFIES THAT THE RX.LINE FIELDS ARE CORRECT FOR THE SKIP SELFTEST CODES.
9. BMP RUN TEST - THIS TEST RUNS THE BMP AND VERIFIES THAT IT DOES NOT FAIL WITHIN A SPECIFIED PERIOD. THIS TEST SHOULD SIGNAL PROBLEMS THAT THE BMP CODES COULD CAUSE WITH LATER TESTS.
10. SKIP SELFTEST TEST - THIS TEST VERIFIES THAT IF THE SELFTEST IS SKIPPED THE PROPER CODES ARE PLACED IN THE FIFO AND THAT NO ERRORS ARE ENCOUNTERED.

11. DIAGNOSTIC.FAIL (SKIP SELFTEST) TEST - THIS TEST VERIFIES, BY USING THE SKIP SELFTEST SEQUENCE, THAT THE DIAGNOSTIC.FAIL BIT WILL GO TO BOTH THE ACTIVE AND INACTIVE STATES.
12. SELFTEST RUN TEST - VERIFIES THAT NO ERRORS ARE FOUND BY THE EXECUTION OF THE SELFTEST.
13. SELFTEST FAIL TEST VERIFIES THAT THE SELFTEST WILL REPORT ERRORS CORRECTLY WHEN IT IS FORCED TO FAIL.
14. ROM VERSION PRINTOUT TEST IF REQUESTED, REPORTS THE VERSION NUMBERS OF THE 8051 ROMS.
15. WORD ACCESS READ/WRITE TEST - VERIFIES THAT THE REGISTERS RESPOND CORRECTLY TO WORD READ AND WRITE ACCESSES.
16. WORD ACCESS READ/MODIFY/WRITE TEST VERIFIES THAT THE REGISTERS RESPOND CORRECTLY TO READ/MODIFY/WRITE WORD ACCESSES.
17. BYTE ACCESS READ/WRITE TEST VERIFIES THAT THE REGISTERS RESPOND CORRECTLY TO BYTE READ AND WRITE ACCESSES.
18. BYTE ACCESS READ/MODIFY/WRITE TEST VERIFIES THAT THE REGISTERS RESPOND CORRECTLY TO READ/MODIFY/WRITE BYTE ACCESSES.
19. ID.BIT TEST - VERIFIES THAT THE ID.BIT READS AS A ZERO.
20. NO TX.DATA.VALID/NO TX.ACTION TEST VERIFIES THAT IF A DATA WORD IS WRITTEN WITHOUT THE TX.DATA.VALID BIT SET, NO TX.ACTION IS GENERATED. THIS TEST DOES NOT REQUIRE THAT CHARACTERS ARE TXED.
21. TX.DATA.VALID / TX.ACTION TEST - VERIFIES THAT IF A DATA WORD IS WRITTEN WITH THE TX.DATA.VALID BIT SET, IT GENERATES A CORRESPONDING TX.ACTION. THIS TEST DOES NOT REQUIRE THAT CHARACTERS ARE TXED.
22. TX.ENABLE INACTIVE TEST - VERIFIES THAT IF THE TX.ENABLE BIT IS CLEAR NO TRANSMISSION OCCURS.
23. TX.ENABLE ACTIVE TEST VERIFIES THAT TX OCCURS IF THE TX.ENABLE IS SET.
24. INTERRUPTS TEST VERIFIES THAT THE TX AND RX INTERRUPTS ARE FUNCTIONING CORRECTLY.
25. BR LEVEL TEST VERIFIES THAT THE UUT GENERATES TX AND RX INTERRUPTS AT THE CORRECT BR LEVEL.
26. DIAG FIELD (BMP) TEST VERIFIES THAT A REQUEST FOR BMP CODE REPORTING IS ANSWERED BY THE UUT WITHIN THE SPECIFIED TIME.

27. REPORT BMP CODES TEST THIS PSEUDO TEST REPORTS THE FIRST 32
BMP CODES WHICH WERE DISCOVERED IN THE FIFO DURING THE
EXECUTION OF THE OTHER TESTS. THIS AVOIDS THE INTERRUPTION
OF OTHER TESTS BY THESE CODES, IF THEY ARE NOT CRITICAL TO
THE TESTS BEING PERFORMED.

6.0 EXAMPLE ERROR FREE PASS

THE FOLLOWING IS AN EXAMPLE OF AN ERROR FREE PASS DIALOGUE:

.R CVDHABO
CVDHABO.BIC

DRS
CVDHA-B 0
DHV 11 FUNC TST PART1
UNIT IS DHV 11
RESTART ADDR: 147670
DR>STA

CHANGE HW (L) ? Y

UNITS (D) ? 2

UNIT 0
CSR ADDRESS: (0) 160460 ? Y2

UNIT 1
CSR ADDRESS: (0) 160460 ? 160040
INTERRUPT VECTOR ADDRESS: (0) 300 ? 320
ACTIVE LINE BIT MAP: (0) 377 ? <CR>
INTERRUPT BR LEVEL: (0) 4? <CR>

CHANGE SW (L) ? Y

REPORT UNIT NUMBER AS EACH UNIT IS TESTED: (L) Y ? <CR>
NUMBER OF INDIVIDUAL DATA ERRORS TO REPORT ON A LINE: (D) 0 ? 4
ROM VERSION PRINTOUT ON THE FIRST PASS: (L) Y ?

TESTING UNIT : 0(D)

ROM VERSION NUMBERS: PROC_1 = 2(D) PROC_2 = 2(D)

TESTING UNIT : 1(D)

ROM VERSION NUMBERS: PROC_1 = 2(D) PROC 2 = 2(D)

CVDHA EOP 1
0 CUMULATIVE ERRORS

TESTING UNIT : 0(D)
*C

DR> EXIT

```
1048      &      .LIST SEQ,LOC,BIN,MEB
1049      .NLIST CND
1050
1051      ;*****
1052      ;
1053      ;          FVTA.PHD
1054      ;
1055      ;*****
1056
1057
1058
1059
1060      .SBTTL  PROGRAM HEADER
1061
1062
1063      .MCALL  SVC
1064 000000      SVC          ; INITIALIZE SUPERVISOR MACROS
1065
1066      ;*****
1067      ;      IF STRUCTURED MACROS ARE TO BE USED, ADD .MCALL STRUCT AND STRUCT
1068      ;      TO INITIALIZE THE STRUCTURED MACROS.
1069      ;
1070      SVCINS= 1      ; LIST INSTRUCTIONS, SHIFTED RIGHT
1071      SVCTST= 1      ; LIST TEST TAGS, SHIFTED RIGHT
1072      SVCSUB= 1      ; LIST SUBTEST TAGS, SHIFTED RIGHT
1073      SVCGBL= 1      ; LIST GLOBAL TAGS, SHIFTED RIGHT
1074      SVCTAG= 1      ; LIST OTHER TAGS, SHIFTED RIGHT
1075
1076      ;      CHANGE THE VALUES OF THE SVC... SYMBOLS TO BE ZERO IF YOU WISH
1077      ;      TO ALIGN THE MACRO CALLS AND THEIR EXPANSIONS.  CHANGE THE
1078      ;      SYMBOLS TO BE MINUS ONE TO NOT LIST THE EXPANSIONS.  YOU MAY
1079      ;      CHANGE THE SYMBOLS AT ANY POINT IN YOUR PROGRAM.
1080      ;*****
1081
1082 000000      .ENABL  ABS
1083      .ENABL  AMA
1084      .      "      2000
1085
1086 002000      BGNMOD
1087
1088      ;**
1089      ; THE PROGRAM HEADER IS THE INTERFACE BETWEEN
1090      ; THE DIAGNOSTIC PROGRAM AND THE SUPERVISOR.
1091      ;
1092
1093 002000      POINTER BGNRPT,BGNSW,BGNSFT,BGNDU,ERRTBL
1094
1111
1112 002000      HEADER  CVDHA,B,0,16,0,PRI07
1113      002000
1114      002000      103
1115      002001      126
1116      002002      104
```

```
LSNAME::
      .ASCII /C/
      .ASCII /V/
      .ASCII /D/
```

002003	110
002004	101
002005	000
002006	000
002007	000
002010	
002010	102
002011	
002011	060
002012	
002012	000000
002014	
002014	000016
002016	
002016	036662
002020	
002020	037054
002022	
002022	002214
002024	
002024	002226
002026	
002026	037372
002030	
002030	000000
002032	
002032	000000
002034	
002034	000000
002036	
002036	000000
002040	
002040	002124
002042	
002042	000340
002044	
002044	000000
002046	
002046	000000
002050	
002050	003
002051	003
002052	
002052	000000
002054	000000
002056	
002056	000000
002060	
002060	004036
002062	
002062	024232
002064	
002064	000000
002066	
002066	000000
002070	
002070	000000

	.ASCII	/M/
	.ASCII	/A/
	.BYTE	0
	.BYTE	0
	.BYTE	0
L\$REV::		
	.ASCII	/B/
L\$JEPO::		
	.ASCII	/O/
L\$UNIT::		
	.WORD	0
L\$TIML::		
	.WORD	16
L\$MPCP::		
	.WORD	L\$HARD
L\$SPCP::		
	.WORD	L\$SOFT
L\$MPTP::		
	.WORD	L\$MM
L\$SPTP::		
	.WORD	L\$SW
L\$LADP::		
	.WORD	L\$LAST
L\$STA::		
	.WORD	0
L\$CO::		
	.WORD	0
L\$DTYP::		
	.WORD	0
L\$APT::		
	.WORD	0
L\$DTP::		
	.WORD	L\$DISPATCH
L\$PRIO::		
	.WORD	PRI07
L\$ENVI::		
	.WORD	0
L\$EXP1::		
	.WORD	0
L\$MREV::		
	.BYTE	C\$REVISION
	.BYTE	C\$EDIT
L\$EF::		
	.WORD	0
	.WORD	0
L\$SPC::		
	.WORD	0
L\$DEVP::		
	.WORD	L\$DVTP
L\$REPP::		
	.WORD	L\$RPY
L\$EXP4::		
	.WORD	0
L\$EXP5::		
	.WORD	0
L\$AUT::		
	.WORD	0

002072
002072 025120
002074
002074 000000
002076
002076 004046
002100
002100 104035
002102
002102 003766
002104
002104 024246
002106
002106 025102
002110
002110 025100
002112
002112 024240
002114
002114 000000
002116
002116 000000
002120
002120 000000

1113

L\$DJT::
L\$DJT:: .WORD L\$DJ
L\$LUN::
L\$LUN:: .WORD 0
L\$DESC::
L\$DESC:: .WORD L\$DESC
L\$LOAD::
L\$LOAD:: EMT E\$LOAD
L\$ETP::
L\$ETP:: .WORD L\$ERRTBL
L\$ICP::
L\$ICP:: .WORD L\$INIT
L\$CCP::
L\$CCP:: .WORD L\$CLEAN
L\$ACP::
L\$ACP:: .WORD L\$AUTO
L\$PRT::
L\$PRT:: .WORD L\$PRU
L\$TEST::
L\$TEST:: .WORD 0
L\$DLT::
L\$DLT:: .WORD 0
L\$HIME::
L\$HIME:: .WORD 0

1125
1126
1127
1128
1129
1130
1131
1132

.SBTTL DISPATCH TABLE

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

DISPATCH 27

.WORD 27
I \$DISPATCH::
.WORD T1
.WORD T2
.WORD T3
.WORD T4
.WORD T5
.WORD T6
.WORD T7
.WORD T8
.WORD T9
.WORD T10
.WORD T11
.WORD T12
.WORD T13
.WORD T14
.WORD T15
.WORD T16
.WORD T17
.WORD T18
.WORD T19
.WORD T20
.WORD T21
.WORD T22
.WORD T23
.WORD T24
.WORD T25
.WORD T26
.WORD T27

002122 000033
002124 025236
002126 025526
002130 025756
002132 026222
002134 026420
002136 026612
002140 027030
002142 027236
002144 027444
002146 027632
002150 030046
002152 030274
002154 030556
002156 031134
002160 031534
002162 031752
002164 032174
002166 032446
002170 032724
002172 033040
002174 033256
002176 033520
002200 034026
002202 034372
002204 035424
002206 036314
002210 036600

1133

```

1141
1142 ;*****
1143 ;
1144 ;          FVTA.DHT
1145 ;
1146 ;*****
*****
1147
1148
1149
1150 .SBTTL  DEFAULT HARDWARE P TABLE
1151
1152 ;**
1153 ; THE DEFAULT HARDWARE P-TABLE CONTAINS DEFAULT VALUES OF
1154 ; THE TEST DEVICE PARAMETERS.  THE STRUCTURE OF THIS TABLE
1155 ; IS IDENTICAL TO THE STRUCTURE OF THE HARDWARE P TABLES.
1156 ; AND IS USED AS A 'TEMPLATE' FOR BUILDING THE P-TABLES.
1157 ;--
1158
1159 002212          BGNHW  DFPTBL
      002212      000004
      002214
      002214
1160
1161 002214      160460          .WORD  160460  ;DEFAULT CSR ADDRESS
1162 002216      000300          .WORD  300    ;DEFAULT VECTOR ADDRESS
1163 002220      177777          .WORD  177777  ;DEFAULT ACTIVE LINES BIT MAP
1164 002222          004          .BYTE  4      ;DEFAULT BR LEVEL
1165
1166
1167 002224          ENDPHW
      002224

```

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

L10000:


```

1194
1195 ;*****
1196 ;
1197 ;          FVTA.EQU
1198 ;
1199 ;*****
1200
1201
1202 .SBTTL GLOBAL EQUATES SECTION
1203
1204
1205
1206
1207
1208
1209
1210
1211
1212
1213
1214
1215
1216
1217
1218
1219
1220
1221         000010          NUMLNS==10          ;NUMBER OF LINES ON DMV11 IS 1.
1222         000377          MAPLNS==377         ;BIT MAP OF LINES ON DMV11.
1223
1224 ;***** DEVICE REGISTER OFFSETS FROM THE CSR'S ADDRESS *****
1225         000000          CSRO==0             ;CSR REGISTER OFFSET FROM THE CSR ADDRESS
1226         000002          RBUF0==2           ;RECEIVE REGISTER OFFSET FROM THE CSR ADDRESS
1227         000002          TXCHRO==2          ;TRANSMIT REGISTER OFFSET FROM THE CSR ADDRESS
1228         000004          LPRO==4            ;LINE PARAMETER REGISTER OFFSET FROM THE CSR ADDRESS
1229         000006          STAT0==6           ;STATUS REGISTER OFFSET FROM THE CSR ADDRESS
1230         000010          LNCTRO==10         ;LINE CONTROL REGISTER OFFSET FROM THE CSR ADDRESS
1231         000012          TXAD10==12         ;TRANSMIT ADDRESS 1 REGISTER OFFSET FROM THE CSR ADDRESS
1232         000014          TXAD20==14         ;TRANSMIT ADDRESS 2 REGISTER OFFSET FROM THE CSR ADDRESS
1233         000016          TXBFC0==16         ;TRANSMIT COUNT REGISTER OFFSET FROM THE CSR ADDRESS
1234
1235 ;***** EQUATES USED WITH RESPECT TO THE RX BUFFER *****
1236         000020          RXBETX==16.        ;LEVEL OF RX BUFFER AT WHICH TO RE ENABLE TRANSMISSION.
1237         000030          RXBDTX==24.        ;LEVEL OF RX BUFFER AT WHICH TO DISABLE TRANSMISSION.
1238         000100          RXBFUL==64.        ;TOTAL CHARACTER CAPACITY OF THE RX BUFFER.
1239
1240
1241
1242
1243
1244
1245
1246
1247
1248
1249
1250
1251
1252
1253
1254
1255 002232          EQUALS
1256
1257 ;
1258 ; BIT DIFINITIONS
1259 ;
1260         100000          BIT15== 100000
1261         040000          BIT14== 40000
1262         020000          BIT13== 20000
1263         010000          BIT12== 10000
1264         004000          BIT11== 4000
1265         002000          BIT10== 2000
1266         001000          BIT09== 1000
1267         000400          BIT08== 400
1268         000200          BIT07== 200
1269         000100          BIT06== 100
1270         000040          BIT05== 40
1271         000020          BIT04== 20
1272         000010          BIT03== 10
1273         000004          BIT02== 4
1274         000002          BIT01== 2

```

```

000001      BIT00== 1
;
001000      BIT9==  BIT09
000400      BIT8==  BIT08
000200      BIT7==  BIT07
000100      BIT6==  BIT06
000040      BIT5==  BIT05
000020      BIT4==  BIT04
000010      BIT3==  BIT03
000004      BIT2==  BIT02
000002      BIT1==  BIT01
000001      BIT0==  BIT00
;
; EVENT FLAG DEFINITIONS
; EF32:EF17 RESERVED FOR SUPERVISOR TO PROGRAM COMMUNICATION
;
000040      EF.START==      32.      ; START COMMAND WAS ISSUED
000037      EF.RESTART==    31.      ; RESTART COMMAND WAS ISSUED
000036      EF.CONTINUE==   30.      ; CONTINUE COMMAND WAS ISSUED
000035      EF.NEW==        29.      ; A NEW PASS HAS BEEN STARTED
000034      EF.PWR==        28.      ; A POWER FAIL/POWER UP OCCURRED
;
; PRIORITY LEVEL DEFINITIONS
;
000340      PRI07== 340
000300      PRI06== 300
000240      PRI05== 240
000200      PRI04== 200
000140      PRI03== 140
000100      PRI02== 100
000040      PRI01== 40
000000      PRI00== 0
;
; OPERATOR FLAG BITS
;
000004      EVL==      4
000010      LOT==     10
000020      ADR==     20
000040      IDU==     40
000100      ISR==    100
000200      UAM==    200
000400      BOE==    400
001000      PNT==   1000
002000      PRI==   2000
004000      IXE==   4000
010000      IBE==  10000
020000      IER==  20000
040000      LOE==  40000
100000      MOE== 100000

```

```

1258
1259
1260
1261
1262
1263
1264
1265
1266
1267
1268
1269
1270
1271
1272
1273
1274
1275
1276
1277
1278 002232 000300
1279 002234 000304
1280 002236 000377
1281 002240 000000
1282 002242 004
1283
1284
1285
1286
1287
1288 002244
1289 002244 160020
1290 002246 160022
1291 002250 160024
1292 002252 160026
1293 002254 160030
1294 002256 160032
1295 002260 160034
1296 002262 160036
1297
1298
1299
1300
1301
1302 002264 137660
1303 002266 177777
1304 002270 000007
1305 002272 177777
1306 002274 166051
1307 002276 000000
1308 002300 077700
1309 002302 000000
1310
1311
1312
1313
1314 002304 006002

```

```

;*****
;
;          FVTA.GDT
;
;*****

.SBTTL  GLOBAL DATA SECTION

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

;*****
;          UNIT VARIABLE AREA
;*****

RXVECA:: .WORD 300      ;RX VECTOR ADDRESS.
TXVECA:: .WORD 304      ;TX VECTOR ADDRESS.
ACTLNS:: .WORD 377      ;ACTIVE LINE BIT MAP.
UNITN:: .WORD 0         ;UNIT NUMBER.
BRLEVL:: .BYTE 4        ;INTERRUPT BUS REQUEST LEVEL
               .EVEN

;*****
;          DEVICE REGISTER ADDRESS TABLE
;*****
DRADR::
CSRA:: .WORD 160020 ;DMV-11 CSR ADDRESS
TXCHA:: RBUFA:: .WORD 160022 ;DMV-11 RECEIVE/TRANSMIT BUFFER ADDRESS
LPRA:: .WORD 160024 ;DMV-11 LINE PARAMETER REGISTER ADDRESS
STATA:: .WORD 160026 ;DMV-11 STATUS REGISTER ADDRESS
LNCTRA:: .WORD 160030 ;DMV-11 LINE CONTROL REGISTER ADDRESS
TXAD1A:: .WORD 160032 ;DMV-11 TRANSMIT BUFFER 1 REGISTER ADDRESS
TXAD2A:: .WORD 160034 ;DMV-11 TRANSMIT BUFFER 2 REGISTER ADDRESS
TXBFCA:: .WORD 160036 ;DMV-11 TRANSMIT BUFFER COUNT REGISTER ADDRESS

;*****
;          BIT MASK TABLE OF UN-USED DMV DEVICE REGISTER BITS.
;*****
UNBITB:: .WORD 137660 ;UNUSED BIT MASK FOR THE CSR
               .WORD 177777 ;UNUSED BIT MASK FOR THE RBUF/TX REG
               .WORD 7 ;UNUSED BIT MASK FOR THE LPR
               .WORD 177777 ;UNUSED BIT MASK FOR THE STAT
               .WORD 166051 ;UNUSED BIT MASK FOR THE LNCTRL
               .WORD 0 ;UNUSED BIT MASK FOR THE TBUFFAD1
               .WORD 77700 ;UNUSED BIT MASK FOR THE TBUFFAD2
               .WORD 0 ;UNUSED BIT MASK FOR THE TBUFFCT

;*****
;          REGISTER MESSAGE ADDRESS TABLE
;*****
RMA1B:: .WORD DROOMG ;ADDRESS OF 'CSR' MESSAGE.

```

```

1315 002306 006006 .WORD DR02MG ;ADDRESS OF "RBUF" MESSAGE.
1316 002310 006013 .WORD DR04MG ;ADDRESS OF "LPR" MESSAGE.
1317 002312 006017 .WORD DR06MG ;ADDRESS OF "STAT" MESSAGE.
1318 002314 006024 .WORD DR10MG ;ADDRESS OF "LNCTRL" MESSAGE.
1319 002316 006033 .WORD DR12MG ;ADDRESS OF "TBUFAD1" MESSAGE.
1320 002320 006044 .WORD DR14MG ;ADDRESS OF "TBUFAD2" MESSAGE.
1321 002322 006055 .WORD DR16MG ;ADDRESS OF "TBUFFT" MESSAGE.
1322
1323
1324 ;*****
1325 ; ASSORTED GLOBAL VARIABLES:
1326 ;*****
1326 002324 000000 BUFPTR:: .WORD 0 ;STORAGE FOR RECEIVE CHARACTER BUFFER POINTER.
1327 002326 000001 TSTNUM:: .WORD 1 ;STORAGE FOR THE TEST NUMBER.
1328 002330 000000 IESTAT:: .WORD 0 ;STORAGE FOR THE INTERRUPT ENABLE BIT STATES.
1329 002332 000000 PASCNT:: .WORD 0 ;STO'G FOR PASS COUNT USED IN ROM VERSION# "ST.
1330 002334 000000 RXINTC:: .WORD 0 ;STORAGE FOR RECEIVER INTERRUPT FLAGS.
1331 002336 000000 RXINTF:: .WORD 0 ;STORAGE FOR RECEIVER INTERRUPT FLAGS.
1332 002340 000000 TXINTC:: .WORD 0 ;STORAGE FOR TRANSMIT INTERRUPT COUNT.
1333 002342 000000 TXINTF:: .WORD 0 ;STORAGE FOR TRANSMIT INTERRUPT FLAGS.
1334 002344 000000 TP4VEC:: .WORD 0 ;STORAGE FOR THE NORMAL 004 TRAP VECTOR.
1335 002346 000000 TP4FLG:: .WORD 0 ;FLAGS SET WHEN AN EXPECTED 004 TRAP OCCURS.
1336 002350 000000 WORD1:: .WORD 0 ;LOCATION FOR PASSING INDIRECT PARAMETERS.
1337 002352 000000 CTRLCF:: .WORD 0 ;STORAGE FOR THE CONTROL-C FLAG.
1338
1339 ;*****
1340 ; LINE TIME CLOCK VARIABLES AND STORAGE.
1341 ;*****
1342 002354 177546 CLKCSR:: .WORD 177546 ;CSR ADDRESS OF THE LTC.
1343 002356 000300 CLKBRL:: .WORD PRI06 ;INTERRUPT PRIORITY LEVEL OF THE LTC.
1344 002360 000100 CLKVEC:: .WORD 100 ;INTERRUPT VECTOR ADDRESS OF THE LTC.
1345 002362 000074 CLKHRZ:: .WORD 60. ;INTERRUPT FREQUENCY OF THE LTC.
1346 002364 000000 TIMER1:: .WORD 0 ;HARDWARE CLOCK COUNTER #1.
1347 002366 000000 TIMER2:: .WORD 0 ;HARDWARE CLOCK COUNTER #2.
1348 002370 000170 TIMER3:: .WORD 120. ;HARDWARE BREAK COUNTER LOCATION.
1349 002372 000170 BCOUNT:: .WORD 120. ;BREAK COUNT VALUE IN CLOCK TICKS.
1350 002374 000021 MSTICK:: .WORD 17. ;NUMBER OF MILLI-SECONDS PER LTC TICK.
1351 002376 000062 MSLCNT:: .WORD 62 ;LOOP COUNT (USED BY MSLOOP) TO DELAY 1 MS.
1352
1353 ;*****
1354 ; MEMORY MANAGEMENT VARIABLES AND FLAGS.
1355 ;*****
1356 002400 177572 MMSRO:: .WORD 177572 ;ADDRESS OF MEM MGT STATUS REGISTER #0.
1357 002402 000000 MMPRES:: .WORD 0 ;MEM MGT PRESENT FLAG (0 IF MM NOT PRESENT).
1358 002404 000000 MMENAB:: .WORD 0 ;MEM MGT ENABLED FLAG (0 IF MM NOT ENABLED).
1359 002406 172340 PAROA:: .WORD 172340 ;ADDRESS OF MEM MGT PAR #0.
1360
1361 ;*****
1362 ; TABLE OF WORDS WITH CORRESPONDING BIT SET FOR GENERATION OF BIT MAPS.
1363 ;*****
1364 002410 000001 BITTBL:: .WORD 1 ;BIT 0 SET.
1365 002412 000002 .WORD 2 ;BIT 1 SET.
1366 002414 000004 .WORD 4 ;BIT 2 SET.
1367 002416 000010 .WORD 10 ;BIT 3 SET.
1368 002420 000020 .WORD 20 ;BIT 4 SET.
1369 002422 000040 .WORD 40 ;BIT 5 SET.
1370 002424 000100 .WORD 100 ;BIT 6 SET.
1371 002426 000200 .WORD 200 ;BIT 7 SET.

```

```

1372 002430 000400      .WORD 400      ;BIT 8 SET.
1373 002432 001000      .WORD 1000     ;BIT 9 SET.
1374 002434 002000      .WORD 2000     ;BIT 10 SET.
1375 002436 004000      .WORD 4000     ;BIT 11 SET.
1376 002440 010000      .WORD 10000    ;BIT 12 SET.
1377 002442 020000      .WORD 20000    ;BIT 13 SET.
1378 002444 040000      .WORD 40000    ;BIT 14 SET.
1379 002446 100000      .WORD 100000   ;BIT 15 SET.
1380
1381
1382      ;*****
1382      ;* GPR SAVE AREA ZERO.
1382      ;*****
1383
1383      GPRSOB::      ;BASE OF GPR SAVE AREA NUMBER ZERO.
1384 002450      .WORD 0      ;WORD 1, STORAGE FOR R1.
1385 002450 000000      .WORD 0      ;WORD 2, STORAGE FOR R2.
1386 002452 000000      .WORD 0      ;WORD 3, STORAGE FOR R3.
1387 002454 000000      .WORD 0      ;WORD 4, STORAGE FOR R4.
1388 002456 000000      .WORD 0      ;WORD 5, STORAGE FOR R5.
1389 002460 000000
1390
1391      ;*****
1392      ;* TRANSMISSION AND RECEPTION VARIABLES, POINTERS, AND FLAGS.
1392      ;*****
1393
1393      ERSNRF:: .WORD 0      ;ERROR SUMMARY REPORT FLAGS.
1394 002462 000000      ERCNTB:: .BLKW 16.      ;TABLE OF ERROR COUNTERS.
1395 002464
1396
1397      ;*****
1398      ; STORAGE AREA FOR THE BMP CODE QUEUE.
1398      ;*****
1399
1399      BMPCOP:: .WORD 0      ;POINTER USED TO ACCESS THE NEXT CELL IN QUE.
1400 002524 000000      BMPCQB:: .BLKW 64.      ;STORAGE FOR 32 CELLS, TEST# PLUS BMP CODE.
1401 002526      BMPCQE::      ;LAST ADDRESS PLUS 2 OF THE BMP CODE QUEUE.
1402 002726
1403
1404      ;*****
1405      ; GENERAL TABLE AND BUFFER AREA--513 WORDS.
1405      ;*****
1406 002726      BUFBAS::      ;BASE OF MEMORY BUFFER.
1407 002726      ERLTBL:: .BLKW 128.      ;FIRST HALF OF GENERAL TABLE OR BUFFER.
1408 003326      BUFMID:: .BLKW 64.      ;SECOND HALF OF GENERAL TABLE OR BUFFER.
1409 003526      BUF3QT:: .BLKW 64.      ;LAST QUARTER OF THE BUFFER AREA.
1410 003726      BUFEND::      ;END OF GENERAL PURPOSE MEMORY BUFFER.
1411 003726      ENDETBL:: .BLKW 16.      ;BUFFER OVERFLOW SPACE.
1412
1425      ERRRTBL
1425 003766
1425 003766 000000      ERRRTBL:: L$ERRRTBL::
1425 003770 000000      ERRRTYP:: .WORD 0
1425 003772 000000      ERRNBR:: .WORD 0
1425 003774 000000      ERRMSG:: .WORD 0
1426      ERRBLK:: .WORD 0
1427
1427      .EVEN

```

-12

```

1429 .SBTTL GPR HANDLING ROUTINES FOR SUBROUTINE CALLS.
1430 :*****
1431 :*
1432 :*   THERE ARE 4 ROUTINES AND MACRO DEFINITIONS USED FOR THE HANDLING OF
1433 :*   GPR VALUES DURING SUBROUTINE CALLS WITHIN THIS PROGRAM. THE FOUR
1434 :*   ROUTINES/MACRO CALLS HAVE THE FOLLOWING NAMES:
1435 :*
1436 :*   SAVE - MACRO DEFINITION USED AT THE BEGINNING OF A SUBROUTINE TO
1437 :*           SAVE THE GPR CONTENTS FOR LATER RESTORATION.
1438 :*   PASS - MACRO DEFINITION USED AT THE END OF A SUBROUTINE TO RESTORE
1439 :*           THE PREVIOUSLY SAVED GPR CONTENTS AND TO LEAVE THE CONTENTS
1440 :*           OF THE SPECIFIED GPR(S) INTACT (NOT RESTORED).
1441 :*   PREGOS - SUBROUTINE WHICH IS CALLED FROM THE SAVE AND PASS MACRO
1442 :*             EXPANSIONS WHICH ACTUALLY PERFORMS THE ACTIONS ON THE GPRS.
1443 :*
1444 :*   DURING A SUBROUTINE WHICH USES THESE GPR SAVE ROUTINES THE VALUES
1445 :*   OF THE GPRS ARE STORED ON THE STACK IN THE FOLLOWING STACK FRAME:
1446 :*
1447 :*       SP      -> RET PC INTO PREGOS ROUTINE.
1448 :*       SP+2    -> GPR R0 CONTENTS.
1449 :*       SP+4    -> GPR R1 CONTENTS.
1450 :*       SP+6    -> GPR R2 CONTENTS.
1451 :*       SP+8    -> GPR R3 CONTENTS.
1452 :*       SP+10   -> GPR R4 CONTENTS.
1453 :*       SP+12   -> GPR R5 CONTENTS.
1454 :*       SP+14   -> RET PC INTO CALLER OF SUB'INE WHICH CALLED PREGOS.
1455 :*
1456 :*   EACH LEVEL OF SUB'INE CALLING USES 8 WORDS OF STACK OVERHEAD.
1457 :*   THE SAVE AND PASS MACROS CAN ALSO BE USED IN "STRAIGHT LINE CODE"
1458 :*   TO SAVE AND RESTORE THE GPR VALUES. IN ANY CASE, AFTER THE
1459 :*   ISSUING OF A PASS CALL THE GPRS WILL BE RESTORED TO THE VALUES
1460 :*   THEY HAD PRIOR TO THE LAST SAVE CALL (EXCEPT FOR THE EXCEPTED,
1461 :*   OR PASSED INTACT, GPRS SPECIFIED AS PARAMETERS TO THE PASS CALL)
1462 :*   AND THE SP WILL ALSO BE RESTORED TO ITS CONDITION BEFORE THE LAST
1463 :*   SAVE CALL. THE PROGRAMMER MUST BE SURE THAT THE SP HAS THE SAME
1464 :*   VALUE WHEN THE PASS MACRO IS CALLED AS IT HAD IMMEDIATELY AFTER
1465 :*   THE SAVE MACRO WAS CALLED.
1466 :*****

```

```
1467      .SBTTL GPR FRAME ACCESS EQUATES
1468      ;***
1469      ;EQUATES THAT ALLOW ACCESS TO THE STACK FRAME. THESE ARE THE
1470      ;OFFSETS INTO THE STACK FOR REGISTERS SAVED DURING THE PREGOS
1471      ;ROUTINE.
1472      ;
1473      1474      000036      LPCSLT**      36      ;OFFSET FOR LAST RETURN PC.
1475      000016      PCSLOT**      16      ;OFFSET FOR RETURN PC.
1476      000014      R5SLOT**      14      ;OFFSET FOR R5.
1477      000012      R4SLOT**      12      ;OFFSET FOR R4.
1478      000010      R3SLOT**      10      ;OFFSET FOR R3.
1479      000006      R2SLOT**      6       ;OFFSET FOR R2.
1480      000004      R1SLOT**      4       ;OFFSET FOR R1.
1481      000002      R0SLOT**      2       ;OFFSET FOR R0.
```

```

1483 .SBTTL GLOBAL MACRO DEFINITION SAVE
1484 :*****
1485 :* THIS MACRO IS USED AT THE BEGINNING OF A SUBROUTINE TO SAVE THE
1486 :* CONTENTS OF THE GPRS R0 THRU R5.
1487 :*
1488 :* INPUTS: SP UNCHANGED SINCE SUBROUTINE WAS ENTERED
1489 :* R5SLOT OFFSET TO STACK SLOT FOR R5 (EQUATED TO 14 OCTAL)
1490 :*
1491 :* OUTPUTS: GPR SAVE AREA ON THE STACK IS LOADED WITH THE CONTENTS OF GPR,
1492 :* TOP OF STACK - LOADED WITH THE RETURN ADDRESS INTO PREG05
1493 :*
1494 :* CALLING SEQUENCE: SAVE
1495 :*
1496 :* COMMENTS: NO ARGUMENTS ARE ALLOWED.
1497 :* THE PASS MACRO SHOULD BE CALLED TO RESTORE THE GPR VALUES.
1498 :*
1499 :* SUBORDINATE ROUTINES CALLED: PREG05.
1500 :*****
1501
1502 .MACRO SAVE
1503 .LIST
1504 JSR R5,PREG05 ;CALL REGISTER SAVE SUBRT.
1505 .NLIST
1506 .ENDM SAVE

```

```
1508 .SBTTL GLOBAL MACRO DEFINITION PASS
1509 ;*****
1510 ;* THIS MACRO IS USED IN CONJUNCTION WITH THE SAVE MACRO. IT IS
1511 ;* CALLED AT END OF A SUBROUTINE TO PASS PARAMETERS IN GPRS BACK TO THE
1512 ;* CALLING ROUTINE BY ALTERING THE GPR SAVE AREA ON THE STACK AND THEN
1513 ;* RETURNING TO PREG05 TO RESTORE THE GPRS TO THEIR SAVED VALUES.
1514 ;*
1515 ;* INPUTS: ONLY ALLOWED ARGUMENTS ARE "R0" THRU "R5".
1516 ;* ROSLOT THRU RSSLOT MUST BE EQUATED TO THEIR RESPECTIVE GPR SAVE
1517 ;* SLOT OFFSETS BEFORE CALLING THIS MACRO.
1518 ;*
1519 ;* OUTPUTS: THE GPR VALUES ARE PUT IN THEIR RESPECTIVE SLOTS ON THE STACK.
1520 ;*
1521 ;* CALLING SEQUENCE: PASS R0,R1,...
1522 ;*
1523 ;* COMMENTS: ANY COMBINATION OF GPR ARGUMENTS MAY BE LISTED IN ANY ORDER.
1524 ;* FOR EXAMPLE, THE FOLLOWING ARE LEGAL:
1525 ;* PASS R1
1526 ;* PASS R4,R0,R2
1527 ;* THE GPRS LISTED AS ARGUMENTS WILL BE PASSED INTACT TO THE
1528 ;* CALLING ROUTINE, ALL OTHER GPRS WILL BE RESTORED.
1529 ;* THE SP MUST BE AT ITS ORIGINAL VALUE WHEN PASS IS CALLED.
1530 ;*
1531 ;* THE MACRO CALL
1532 ;* PASS R0,R3
1533 ;* EXPANDS INTO THE FOLLOWING ASSEMBLY CODE:
1534 ;* MOV R0,ROSLOT(SP) ;PUT R0 IN STACK SLOT.
1535 ;* MOV R3,R3SLOT(SP) ;PUT R3 IN STACK SLOT.
1536 ;* JSR PC,@(SP)+ ;RETURN TO PREG05 SUBRT.
1537 ;* IN THIS EXAMPLE GPRS R1, R2, R4, AND R5 WILL BE RESTORED TO
1538 ;* THEIR VALUES CONTAINED IN THE STACK FRAME AND R0 AND R3
1539 ;* WILL BE LEFT AT THEIR VALUES PRIOR TO THIS PASS CALL.
1540 ;*
1541 ;* SUBORDINATE ROUTINES CALLED: (PREGRT - LABEL WITHIN PREG05, VALUE ON STACK.)
1542 ;*****
1543
1544 .MACRO PASS A,B,C,D,E,F
1545 .IRP X,<A,B,C,D,E,F>
1546 .IF NB,X
1547 .LIST
1548 MOV X,X'SLOT(SP) ;PUT X IN STACK SLOT.
1549 .NLIST
1550 .ENDC
1551 .ENDM
1552 .LIST
1553 JSR PC,@(SP)+ ;RETURN TO PREG05 SUBRT.
1554 .NLIST
1555 .ENDM PASS
```

```

1557 .SBTTL GLOBAL SUBROUTINE                                PREG05
1558 :*****
1559 :* PRESERVE REGISTERS R0 THROUGH R5 FOR SUBROUTINE CALLS.
1560 :*
1561 :* INPUTS: THE RETURN ADDRESS BACK INTO THE CALLING ROUTINE MUST BE IN
1562 :* GPR R5. (I.E. MACROS USE JSR R5,PREG05".)
1563 :*
1564 :* OUTPUTS: REGISTERS R0 THROUGH R5 ARE SAVED ON THE STACK.
1565 :*
1566 :* CALLING SEQUENCE: SAVE ;MACRO EXPANSION CALLS PREG05.
1567 :* [SUBROUTINE CODE]...
1568 :* PASS ;MACRO EXPANSION RECALLS PREG05.
1569 :*
1570 :* COMMENTS: THIS ROUTINE IS RE-ENTRANT.
1571 :*
1572 :* PARAMETERS MAY BE PASSED OUT OF A SUBROUTINE BY MODIFYING THE
1573 :* REGISTER SAVE AREA ON THE STACK. USE THE PASS GPRN MACRO
1574 :* TO RETURN GPR VALUES INTACT.
1575 :* USE THE RNSLOT OFFSETS FROM THE SP TO PASS OTHER PARAMETERS.
1576 :* [EXAMPLE: MOV VALUE,R0SLOT(SP) ]
1577 :* MAKE SURE THE SP IS AT ITS ORIGINAL VALUE WHEN YOU DO THIS.
1578 :*
1579 :* SUBORDINATE ROUTINES CALLED: NONE.
1580 :*****
1581
1582 003776 PREG05: ;R5 HAS BEEN LOADED ON THE STACK BY THE SUBROUTINE CALL
1583 003776 010446 MOV R4,-(SP) ;SAVE R4
1584 004000 010346 MOV R3,-(SP) ;SAVE R3
1585 004002 010246 MOV R2,-(SP) ;SAVE R2
1586 004004 010146 MOV R1,-(SP) ;SAVE R1
1587 004006 010046 MOV R0,-(SP) ;SAVE R0
1588 004010 010546 MOV R5,-(SP) ;PUSH RETURN PC ON TOP OF STACK
1589 004012 016605 000014 MOV R5SLOT(SP),R5 ;RESTORE R5 TO VALUE IT HAD BEFORE CALLS
1590
1591 004016 004736 JSR PC,@(SP)+ ;CALL THE SUBROUTINE AT THE RETURN ADDRESS
1592 ;FROM THE PREG05 CALL, PUTTING THE PRESENT
1593 ;PC ON THE STACK AS A RETURN ADDRESS INTO
1594 ;THIS (PREG05) ROUTINE.
1595
1596 :***
1597 :* THE FOLLOWING CODE IS EXECUTED WHEN THE CALLING ROUTINE DOES A
1598 :* "RETURN" [JSR PC,@(SP)+] USING THE PC DEPOSITED ON THE STACK ABOVE.
1599 :* ---
1600
1601 004020 012605 PREGRT: MOV (SP)+,R5 ;PUT RETURN PC IN R5.
1602 004022 012600 MOV (SP)+,R0 ;RESTORE R0.
1603 004024 012601 MOV (SP)+,R1 ;RESTORE R1.
1604 004026 012602 MOV (SP)+,R2 ;RESTORE R2.
1605 004030 012603 MOV (SP)+,R3 ;RESTORE R3.
1606 004032 012604 MOV (SP)+,R4 ;RESTORE R4.
1607
1608 004034 000205 RTS R5 ;RETURN TO THE SUBROUTINE WHICH CALLED PREG05.
1609 ;RESTORING R5 IN THE PROCESS.

```

```

1611 .SBTTL GLOBAL TEXT SECTION
1613 ;*****
1614 ;
1615 ; FVTSKL1.P11
1616 ;*****
1617 ;
1619 ;
1620 ;
1621 ;**
1622 ; THE GLOBAL TEXT SECTION CONTAINS FORMAT STATEMENTS,
1623 ; MESSAGES, AND ASCII INFORMATION THAT ARE USED IN
1624 ; MORE THAN ONE TEST.
1625 ;--
1626 ;
1627 ; NAMES OF DEVICES SUPPORTED BY PROGRAM
1628 ;
1629 ;
1630 004036 DEVTYP <DHV 11>
      004036
      004036 104 110 126 L$DVTYP::
      004041 055 061 061 .ASCIZ /DHV 11/
      004044 000 .EVEN
1631
1637
1638 ; TEST DESCRIPTION
1639 ;
1640 004046 DESCRIPT <DHV-11 FUNC TST PART1>
      004046
      004046 104 110 126 L$DESC::
T1/      004051 055 061 061 .ASCIZ /DHV 11 FUNC TST PAR
      004054 040 106 125
      004057 116 103 040
      004062 124 123 124
      004065 040 120 101
      004070 122 124 061
      004073 000 .EVEN
1641 .EVEN
1642
1649

```

```

1651
1652      ;*****
1653      ;
1654      ;           FVTA.FMT
1655      ;
1656      ;*****
1657
1658
1659
1660      ;
1661      ; FORMAT STATEMENTS USED IN PRINT CALLS
1662      ;
1663      ; ***** FORMAT STATEMENTS *****
1664 004074 045 116 045 MFUNIT:: .ASCIZ /%N% A TESTING UNIT :%D4%A(D)%N/
      004077 101 040 124
      004102 105 123 124
      004105 111 116 107
      004110 040 125 116
      004113 111 124 040
      004116 072 045 104
      004121 064 045 101
      004124 050 104 051
      004127 045 116 000
1665 004132 045 124 045 EF0503:: .ASCIZ /%T%N/
      004135 116 000
1666 004137 045 101 040 EF0505:: .ASCIZ /%A %D5%A ILLEGAL INTERRUPTS RECEIVED.%N/
      004142 040 040 040
      004145 045 104 065
      004150 045 101 040
      004153 111 114 114
      004156 105 107 101
      004161 114 040 111
      004164 116 124 105
      004167 122 122 125
      004172 120 124 123
      004175 040 122 105
      004200 103 105 111
      004203 126 105 104
      004206 056 045 116
      004211 000
1667 004212 045 116 045 EF1401:: .ASCIZ /%N% A ROM VERSION NUMBERS: PROC_1 = %D2%A(D) PROC_2 = %D2%A(D)%N/
      004215 101 040 122
      004220 117 115 040
      004223 126 105 122
      004226 123 111 117
      004231 116 040 116
      004234 125 115 102
      004237 105 122 123
      004242 072 040 120
      004245 122 117 103
      004250 137 061 040
      004253 075 040 045
      004256 104 062 045
      004261 101 050 104
      004264 051 040 040
      004267 040 120 122
      004272 117 103 137

```

	004275	062	040	075	
	004300	040	045	104	
	004303	062	045	101	
	004306	050	104	051	
	004311	045	116	000	
1668	004314	045	124	045	EF1402:: .ASCIZ /#T#A ROM VERSION NUMBER #T#N/
	004317	101	040	122	
	004322	117	115	040	
	004325	126	105	122	
	004330	123	111	117	
	004333	116	040	116	
	004336	125	115	102	
	004341	105	122	040	
	004344	045	124	045	
	004347	116	000		
1669	004351	045	101	040	EF1601:: .ASCIZ /#A #T#A ABORTED #N/
	004354	040	045	124	
	004357	045	101	040	
	004362	101	102	117	
	004365	122	124	105	
	004370	104	040	045	
	004373	116	000		
1670	004375	045	101	040	EF1602:: .ASCIZ /#A EXPECTED DATA: #06#A (0).#N/
	004400	040	040	040	
	004403	105	130	120	
	004406	105	103	124	
	004411	105	104	040	
	004414	104	101	124	
	004417	101	072	040	
	004422	045	117	066	
	004425	045	101	040	
	004430	050	117	051	
	004433	056	045	116	
	004436	000			
1671	004437	045	101	040	EF1603:: .ASCIZ /#A ACTUAL DATA: #06#A (0).#N/
	004442	040	040	040	
	004445	101	103	124	
	004450	125	101	114	
	004453	040	104	101	
	004456	124	101	072	
	004461	040	040	040	
	004464	045	117	066	
	004467	045	101	040	
	004472	050	117	051	
	004475	056	045	116	
	004500	000			
1672	004501	045	101	040	EF1604:: .ASCIZ /#A BAD BIT(S) IN DEVICE #T#A REGISTER FOR LINE #D2#A (D).#N/
	004504	040	102	101	
	004507	104	040	102	
	004512	111	124	050	
	004515	123	051	040	
	004520	111	116	040	
	004523	104	105	126	
	004526	111	103	105	
	004531	040	045	124	
	004534	045	101	040	
	004537	122	105	107	

	004542	111	123	124	
	004545	105	122	040	
	004550	106	117	122	
	004553	040	114	111	
	004556	116	105	040	
	004561	045	104	062	
	004564	045	101	040	
	004567	050	104	051	
	004572	056	045	116	
	004575	000			
1673	004576	045	101	040	EF3001:: .ASCIZ /#A EXPECTED OR CORRECT VALUE: #03#N/
	004601	040	040	040	
	004604	105	130	120	
	004607	105	103	124	
	004612	105	104	040	
	004615	117	122	040	
	004620	103	117	122	
	004623	122	105	103	
	004626	124	040	126	
	004631	101	114	125	
	004634	105	072	040	
	004637	045	117	063	
	004642	045	116	000	
1674	004645	045	101	040	EF3002:: .ASCIZ /#A ACTUAL OR MEASURED VALUE: #03#N/
	004650	040	040	040	
	004653	101	103	124	
	004656	125	101	114	
	004661	040	117	122	
	004664	040	115	105	
	004667	101	123	125	
	004672	122	105	104	
	004675	040	040	126	
	004700	101	114	125	
	004703	105	072	040	
	004706	045	117	063	
	004711	045	116	000	
1675	004714	045	101	040	EF9001:: .ASCIZ /#A UNEXPECTED #T#A FOUND IN RECEIVE CHAR FIFO:#N/
	004717	040	125	116	
	004722	105	130	120	
	004725	105	103	124	
	004730	105	104	040	
	004733	045	124	045	
	004736	101	040	106	
	004741	117	125	116	
	004744	104	040	111	
	004747	116	040	122	
	004752	105	103	105	
	004755	111	126	105	
	004760	040	103	110	
	004763	101	122	040	
	004766	106	111	106	
	004771	117	072	045	
	004774	116	000		
1676	004776	045	101	040	EF9002:: .ASCIZ /#A CODE IS ASSOCIATED WITH LINE: #02#A(D)#N/
	005001	040	040	040	
	005004	103	117	104	
	005007	105	040	111	

	005012	123	040	101		
	005015	123	123	117		
	005020	103	111	101		
	005023	124	105	104		
	005026	040	127	111		
	005031	124	110	040		
	005034	114	111	116		
	005037	105	072	040		
	005042	045	104	062		
	005045	045	101	050		
	005050	104	051	045		
	005053	116	000			
1677	005055	045	101	040	EF9003:: .ASCIZ /A	CODE IS: 03A(0)N/
	005060	040	040	040		
	005063	040	040	103		
	005066	117	104	105		
	005071	040	111	123		
	005074	072	040	045		
	005077	117	063	045		
	005102	101	050	117		
	005105	051	045	116		
	005110	000				
1678	005111	045	101	040	EF9004:: .ASCIZ /A	STA VALUE: 03A(0)N/
	005114	040	040	040		
	005117	045	124	045		
	005122	101	040	126		
	005125	101	114	125		
	005130	105	072	040		
	005133	045	117	063		
	005136	045	101	050		
	005141	117	051	045		
	005144	116	000			
1679	005146	045	101	040	EF9005:: .ASCIZ /A	STA VALUE: NONEN/
	005151	040	040	040		
	005154	045	124	045		
	005157	101	040	126		
	005162	101	114	125		
	005165	105	072	040		
	005170	116	117	116		
	005173	105	045	116		
	005176	000				
1680	005177	045	101	040	EF9006:: .ASCIZ /A	STA 02A(0)N/
	005202	040	045	124		
	005205	045	101	040		
	005210	045	104	062		
	005213	045	101	050		
	005216	104	051	045		
	005221	116	000			
1681	005223	045	101	040	EF9010:: .ASCIZ /A	NUMBER OF ERRORS DETECTED ON LINE 02A(0) IS 05A(0)N
	005226	040	040	040		
	005231	116	125	115		
	005234	102	105	122		
	005237	040	117	106		
	005242	040	105	122		
	005245	122	117	122		
	005250	123	040	104		
	005253	105	124	105		

	005256	103	124	105	
	005261	104	040	117	
	005264	116	040	114	
	005267	111	116	105	
	005272	040	045	104	
	005275	062	045	101	
	005300	050	104	051	
	005303	040	111	123	
	005306	040	045	104	
	005311	065	045	101	
	005314	050	104	051	
	005317	045	116	000	
1682	005322	045	101	040	EF9016:: .ASCIZ /%A UNEXPECTED %T%A FOR LINE #D2%A(D) IN FIFO AFTER RESET: %N/
	005325	040	125	116	
	005330	105	130	120	
	005333	105	103	124	
	005336	105	104	040	
	005341	045	124	045	
	005344	101	040	106	
	005347	117	122	040	
	005352	114	111	116	
	005355	105	040	045	
	005360	104	062	045	
	005363	101	050	104	
	005366	051	040	111	
	005371	116	040	106	
	005374	111	106	117	
	005377	040	101	106	
	005402	124	105	122	
	005405	040	122	105	
	005410	123	105	124	
	005413	072	045	116	
	005416	000			
1683	005417	045	101	040	EF9017:: .ASCIZ /%A %T%A (WITH ERROR FLAGS) IS %06%A(0) %N/
	005422	040	040	040	
	005425	045	124	045	
	005430	101	040	050	
	005433	127	111	124	
	005436	110	040	105	
	005441	122	122	117	
	005444	122	040	106	
	005447	114	101	107	
	005452	123	051	040	
	005455	111	123	040	
	005460	045	117	066	
	005463	045	101	050	
	005466	117	051	045	
	005471	116	000		
1684	005473	045	101	040	EF9018:: .ASCII /%A %T%A IN SELFTEST CODE FIFO SLOT FOR LINE #D2/
	005476	040	045	124	
	005501	045	101	040	
	005504	111	116	040	
	005507	123	105	114	
	005512	106	124	105	
	005515	123	124	040	
	005520	103	117	104	
	005523	105	040	106	

	005526	111	106	117	
	005531	040	123	114	
	005534	117	124	040	
	005537	106	117	122	
	005542	040	114	111	
	005545	116	105	040	
	005550	045	104	062	
1685	005553	045	101	050	.ASCIZ /A(D) AFTER RESET.N/
	005556	104	051	040	
	005561	101	106	124	
	005564	105	122	040	
	005567	122	105	123	
	005572	105	124	056	
	005575	045	116	000	
1686	005600	045	101	040	EF9019:: .ASCIZ /A T A 06A(0)N/
	005603	040	045	124	
	005606	045	101	040	
	005611	045	117	066	
	005614	045	101	050	
	005617	117	051	045	
	005622	116	000		
1687	005624	045	101	040	EF9301:: .ASCIZ /A T D2A(D), BMP CODE REPORTED :03A(0)N/
	005627	045	124	045	
	005632	104	062	045	
	005635	101	050	104	
	005640	051	054	040	
	005643	040	102	1	
	005646	120	040	103	
	005651	117	104	105	
	005654	040	122	105	
	005657	120	117	122	
	005662	124	105	104	
	005665	040	072	045	
	005670	117	063	045	
	005673	101	050	117	
	005676	051	045	116	
	005701	000			
1688	005702	045	101	040	EF9302:: .ASCIZ /A OVERFLOW OCCURRED (MORE THAN 31 BMP CODES FOUND IN QUEUE)N/
	005705	040	117	126	
	005710	105	122	106	
	005713	114	117	127	
	005716	040	117	103	
	005721	103	125	122	
	005724	122	105	104	
	005727	040	050	115	
	005732	117	122	105	
	005735	040	124	110	
	005740	101	116	040	
	005743	063	061	040	
	005746	102	115	120	
	005751	040	103	117	
	005754	104	105	123	
	005757	040	106	117	
	005762	125	116	104	
	005765	040	111	116	
	005770	040	121	125	
	005773	105	125	105	

005776 051 045 116
006001 000

1689
1690
1701
1702

.EVEN

```

1711
1712
1713
1714
1715
1716
;*****
;
;          FVTA.MSG
;*****

1717
1718
1719
1720
1721 006002      103      123      122  DR00MG:: .ASCIZ /CSR/
      006005      000
1722 006006      122      102      125  DR02MG:: .ASCIZ /RBUF/
      006011      106      000
1723 006013      114      120      122  DR04MG:: .ASCIZ /LPR/
      006016      000
1724 006017      123      124      101  DR06MG:: .ASCIZ /STAT/
      006022      124      000
1725 006024      114      116      103  DR10MG:: .ASCIZ /LNCTRL/
      006027      124      122      114
      006032      000
1726 006033      124      102      125  DR12MG:: .ASCIZ /TBUFFAD1/
      006036      106      106      101
      006041      104      061      000
1727 006044      124      102      125  DR14MG:: .ASCIZ /TBUFFAD2/
      006047      106      106      101
      006052      104      062      000
1728 006055      124      102      125  DR16MG:: .ASCIZ /TBUFFCT/
      006060      106      106      103
      006063      124      000
1729 006065      104      105      126  EM0103:: .ASCIZ /DEVICE REGISTER ACCESS ERRORS/
      006070      111      103      105
      006073      040      122      105
      006076      107      111      123
      006101      124      105      122
      006104      040      101      103
      006107      103      105      123
      006112      123      040      105
      006115      122      122      117
      006120      122      123      000
1730 006123      115      101      123  EM0201:: .ASCIZ /MASTER RESET (PERFORM SELFTEST) TEST /
      006126      124      105      122
      006131      040      122      105
      006134      123      105      124
      006137      040      050      120
      006142      105      122      106
      006145      117      122      115
      006150      040      123      105
      006153      114      106      124
      006156      105      123      124
      006161      051      040      124
      006164      105      123      124
      006167      040      000
1731 006171      040      040      115  EM0202:: .ASCIZ / MASTER RESET BIT DID NOT CLEAR AFTER BOARD RESET./
      006174      101      123      124
      006177      105      122      040
      006202      122      105      123

```

	006205	105	124	040	
	006210	102	111	124	
	006213	040	104	111	
	006216	104	040	116	
	006221	117	124	040	
	006224	103	114	105	
	006227	101	122	040	
	006232	101	106	124	
	006235	105	122	040	
	006240	102	117	101	
	006243	122	104	040	
	006246	122	105	123	
	006251	105	124	056	
	006254	000			
1732	006255	040	040	040	.ASCIZ / WAITED 5 SECONDS. BIT DEFECTIVE OR FIRMWARE MUNG./
	006260	040	127	101	
	006263	111	124	105	
	006266	104	040	065	
	006271	040	123	105	
	006274	103	117	116	
	006277	104	123	056	
	006302	040	040	102	
	006305	111	124	040	
	006310	104	105	106	
	006313	105	103	124	
	006316	111	126	105	
	006321	040	117	122	
	006324	040	106	111	
	006327	122	115	127	
	006332	101	122	105	
	006335	040	110	125	
	006340	116	107	056	
	006343	000			
1733	006344	040	040	115	EM0203:: .ASCIZ / MASTER RESET BIT CLEAR IMMEDIATELY AFTER BOARD RESET./
	006347	101	123	124	
	006352	105	122	040	
	006355	122	105	123	
	006360	105	124	040	
	006363	102	111	124	
	006366	040	103	114	
	006371	105	101	122	
	006374	040	111	115	
	006377	115	105	104	
	006402	111	101	124	
	006405	105	114	131	
	006410	040	101	106	
	006413	124	105	122	
	006416	040	102	117	
	006421	101	122	104	
	006424	040	122	105	
	006427	123	105	124	
	006432	056	000		
1734	006434	040	040	040	.ASCIZ / BIT DEFECTIVE OR BOARD FIRMWARE ERROR./
	006437	040	102	111	
	006442	124	040	104	
	006445	105	106	105	
	006450	103	124	111	

	006453	126	105	040	
	006456	117	122	040	
	006461	102	117	101	
	006464	122	104	040	
	006467	106	111	122	
	006472	115	127	101	
	006475	122	105	040	
	006500	105	122	122	
	006503	117	122	056	
	006506	000			
1735	006507	040	040	115	EM0204:: .ASCIZ \ MR BIT WENT CLEAR WITHIN 1/2 SECOND OF BOARD RESET.
	006512	122	040	102	
	006515	111	124	040	
	006520	127	105	116	
	006523	124	040	103	
	006526	114	105	101	
	006531	122	040	127	
	006534	111	124	110	
	006537	111	116	040	
	006542	061	057	062	
	006545	040	123	105	
	006550	103	117	116	
	006553	104	040	117	
	006556	106	040	102	
	006561	117	101	122	
	006564	104	040	122	
	006567	105	123	105	
	006572	124	056	000	
1736	006575	040	040	040	.ASCIZ / BIT DEFECTIVE OR SELFTEST WAS (INCORRECTLY) SKIPPED./
	006600	040	102	111	
	006603	124	040	104	
	006606	105	106	105	
	006611	103	124	111	
	006614	126	105	040	
	006617	117	122	040	
	006622	123	105	114	
	006625	106	124	105	
	006630	123	124	040	
	006633	127	101	123	
	006636	040	050	111	
	006641	116	103	117	
	006644	122	122	105	
	006647	103	124	114	
	006652	131	051	040	
	006655	123	113	111	
	006660	120	120	105	
	006663	104	056	000	
1737	006666	115	101	123	EM0301:: .ASCIZ /MASTER RESET (SKIP SELFTEST) TEST /
	006671	124	105	122	
	006674	040	122	105	
	006677	123	105	124	
	006702	040	050	123	
	006705	113	111	120	
	006710	040	123	105	
	006713	114	106	124	
	006716	105	123	124	
	006721	051	040	124	

	006724	105	123	124	
	006727	040	000		
1738	006731	040	040	115	EM0302:: .ASCIZ / MR BIT CLR WITHIN 10 MILLISECOND AFTER BOARD RESET./
	006734	122	040	102	
	006737	111	124	040	
	006742	103	114	122	
	006745	040	127	111	
	006750	124	110	111	
	006753	116	040	061	
	006756	060	040	115	
	006761	111	114	111	
	006764	123	105	103	
	006767	117	116	104	
	006772	040	101	106	
	006775	124	105	122	
	007000	040	102	117	
	007003	101	122	104	
	007006	040	122	105	
	007011	123	105	124	
	007014	056	000		
1739	007016	040	040	040	.ASCIZ / BIT DEFECTIVE OR BOARD FIRMWARE ERROR./
	007021	040	102	111	
	007024	124	040	104	
	007027	105	106	105	
	007032	103	124	111	
	007035	126	105	040	
	007040	117	122	040	
	007043	102	117	101	
	007046	122	104	040	
	007051	106	111	122	
	007054	115	127	101	
	007057	122	105	040	
	007062	105	122	122	
	007065	117	122	056	
	007070	000			
1740	007071	040	040	115	EM0303:: .ASCIZ \ MR BIT WENT CLEAR 1/5 TO 5 SECONDS AFTER RESET.\
	007074	122	040	102	
	007077	111	124	040	
	007102	127	105	116	
	007105	124	040	103	
	007110	114	105	101	
	007113	122	040	061	
	007116	057	065	040	
	007121	124	117	040	
	007124	065	040	123	
	007127	105	103	117	
	007132	116	104	123	
	007135	040	101	106	
	007140	124	105	122	
	007143	040	122	105	
	007146	123	105	124	
	007151	056	000		
1741	007153	040	040	040	.ASCIZ / SELFTEST DID NOT GET SKIPPED (SHOULD HAVE BEEN SKIPPED).
	007156	040	123	105	
	007161	114	106	124	
	007164	105	123	124	
	007167	040	104	111	

	007172	104	040	116	
	007175	117	124	040	
	007200	107	105	124	
	007203	040	123	113	
	007206	111	120	120	
	007211	105	104	040	
	007214	050	123	110	
	007217	117	125	114	
	007222	104	040	110	
	007225	101	126	105	
	007230	040	102	105	
	007233	105	116	040	
	007236	123	113	111	
	007241	120	120	105	
	007244	104	051	056	
	007247	000			
1742	007250	122	102	125	EM0401:: .ASCIZ /RBUF REGISTER RX CHARACTER FIELD TEST /
	007253	106	040	122	
	007256	105	107	111	
	007261	123	124	105	
	007264	122	040	122	
	007267	130	040	103	
	007272	110	101	122	
	007275	101	103	124	
	007300	105	122	040	
	007303	106	111	105	
	007306	114	104	040	
	007311	124	105	123	
	007314	124	040	000	
1743	007317	040	040	111	EM0402:: .ASCIZ / IMPROPER CODE FOUND IN RX FIFO AFTER OUT RESET./
	007322	115	120	122	
	007325	117	120	105	
	007330	122	040	103	
	007333	117	104	105	
	007336	040	106	117	
	007341	125	116	104	
	007344	040	111	116	
	007347	040	122	130	
	007352	040	106	111	
	007355	106	117	040	
	007360	101	106	124	
	007363	105	122	040	
	007366	104	125	124	
	007371	040	122	105	
	007374	123	105	124	
	007377	056	000		
1744	007401	040	040	040	.ASCIZ / EXPECTED: SELFTEST CODE, ACTUAL: IMPROPER CODE./
	007404	040	105	130	
	007407	120	105	103	
	007412	124	105	104	
	007415	072	040	123	
	007420	105	114	106	
	007423	124	105	123	
	007426	124	040	103	
	007431	117	104	105	
	007434	054	040	040	
	007437	040	101	103	

	007442	124	125	101	
	007445	114	072	040	
	007450	111	115	120	
	007453	122	117	120	
	007456	105	122	040	
	007461	103	117	104	
	007464	105	056	000	
1745	007467	122	102	125	EM0501:: .ASCIIZ /RBUF REGISTER ERROR FLAGS FIELD TEST /
	007472	106	040	122	
	007475	105	107	111	
	007500	123	124	105	
	007503	122	040	105	
	007506	122	122	117	
	007511	122	040	106	
	007514	114	101	107	
	007517	123	040	106	
	007522	111	105	114	
	007525	104	040	124	
	007530	105	123	124	
	007533	040	000		
1746	007535	040	040	122	EM0502:: .ASCIIZ / RX ERROR FLAG(S) FOUND CLEAR ON SELFTEST CODE./
	007540	130	040	105	
	007543	122	122	117	
	007546	122	040	106	
	007551	114	101	107	
	007554	050	123	051	
	007557	040	106	117	
	007562	125	116	104	
	007565	040	103	114	
	007570	105	101	122	
	007573	040	117	116	
	007576	040	123	105	
	007601	114	106	124	
	007604	105	123	124	
	007607	040	103	117	
	007612	104	105	056	
	007615	000			
1747	007616	040	040	040	.ASCIIZ / EXPECTED: ALL ERROR FLAGS SET, ACTUAL: FLAG(S) CLEAR./
	007621	040	105	130	
	007624	120	105	103	
	007627	124	105	104	
	007632	072	040	101	
	007635	114	114	040	
	007640	105	122	122	
	007643	117	122	040	
	007646	106	114	101	
	007651	107	123	040	
	007654	123	105	124	
	007657	054	040	040	
	007662	101	103	124	
	007665	125	101	114	
	007670	072	040	106	
	007673	114	101	107	
	007676	050	123	051	
	007701	040	103	114	
	007704	105	101	122	
	007707	056	000		

1748	007711	040	040	122	EM0525:: .ASCIZ / RX INTERRUPT(S) RECEIVED WITH RX INTERRUPTS DISABLED./
	007714	130	040	111	
	007717	116	124	105	
	007722	122	122	125	
	007725	120	124	050	
	007730	123	051	040	
	007733	122	105	103	
	007736	105	111	126	
	007741	105	104	040	
	007744	127	111	124	
	007747	110	040	122	
	007752	130	040	111	
	007755	116	124	105	
	007760	122	122	125	
	007763	120	124	123	
	007766	040	104	111	
	007771	123	101	102	
	C 7774	114	105	104	
	007777	056	000		
1749	010001	040	040	124	EM0526:: .ASCIZ / TX INTERRUPT(S) RECEIVED WITH TX INTERRUPTS DISABLED./
	010004	130	040	111	
	010007	116	124	105	
	010012	122	122	125	
	010015	120	124	050	
	010020	123	051	040	
	010023	122	105	103	
	010026	105	111	126	
	010031	105	104	040	
	010034	127	111	124	
	010037	110	040	124	
	010042	130	040	111	
	010045	116	124	105	
	010050	122	122	125	
	010053	120	124	123	
	010056	040	104	111	
	010061	123	101	102	
	010064	114	105	104	
	010067	056	000		
1750	C10071	103	123	122	EM0601:: .ASCIZ /CSR RX.DATA.AVAIL BIT TEST /
	010074	040	122	130	
	010077	056	104	101	
	010102	124	101	056	
	010105	101	126	101	
	010110	111	114	040	
	010113	102	111	124	
	010116	040	124	105	
	010121	123	124	040	
	010124	000			
1751	010125	040	040	122	EM0602:: .ASCIZ / RX.DATA.AVAIL BIT FOUND CLEAR AFTER RESET COMPLETION./
	010130	130	056	104	
	010133	101	124	101	
	010136	056	101	126	
	C10141	101	111	114	
	010144	040	102	111	
	010147	124	040	106	
	010152	117	125	116	
	010155	104	040	103	

	010160	114	105	101	
	010163	122	040	101	
	010166	106	124	105	
	010171	122	040	122	
	010174	105	123	105	
	010177	124	040	103	
	010202	117	115	120	
	010205	114	105	124	
	010210	111	117	116	
	010213	056	000		
1752	010215	040	040	040	.ASCIIZ / EXPECTED BIT TO BE SET FROM SELFTEST CODES IN FIFO./
	010220	040	105	130	
	010223	120	105	103	
	010226	124	105	104	
	010231	040	102	111	
	010234	124	040	124	
	010237	117	040	102	
	010242	105	040	123	
	010245	105	124	040	
	010250	106	122	117	
	010253	115	040	123	
	010256	105	114	106	
	010261	124	105	123	
	010264	124	040	103	
	010267	117	104	105	
	010272	123	040	111	
	010275	116	040	106	
	010300	111	106	117	
	010303	056	000		
1753	010305	040	040	122	EM0603:: .ASCIIZ / RX.DATA.AVAIL BIT COULD NOT BE CLEARED BY PURGING FIFO./
	010310	130	056	104	
	010313	101	124	101	
	010316	056	101	126	
	010321	101	111	114	
	010324	040	102	111	
	010327	124	040	103	
	010332	117	125	114	
	010335	104	040	116	
	010340	117	124	040	
	010343	102	105	040	
	010346	103	114	105	
	010351	101	122	105	
	010354	104	040	102	
	010357	131	040	120	
	010362	125	122	107	
	010365	111	116	107	
	010370	040	106	111	
	010373	106	117	056	
	010376	000			
1754	010377	040	040	040	.ASCIIZ / 600 CHARS READ FROM FIFO WITHOUT R.D.A BIT CLEARING./
	010402	040	066	060	
	010405	060	040	103	
	010410	110	101	122	
	010413	123	040	122	
	010416	105	101	104	
	010421	040	106	122	
	010424	117	115	040	

C5

	010427	106	111	106	
	010432	117	040	127	
	010435	111	124	110	
	010440	117	125	124	
	010443	040	122	056	
	010446	104	056	101	
	010451	040	102	111	
	010454	124	040	103	
	010457	114	105	101	
	010462	122	111	116	
	010465	107	056	000	
1755	010470	122	102	125	EM0701:: .ASCIZ /RBUF RX.DATA.VALID BIT TEST /
	010473	106	040	122	
	010476	130	056	104	
	010501	101	124	101	
	010504	056	126	101	
	010507	114	111	104	
	010512	040	102	111	
	010515	124	040	124	
	010520	105	123	124	
	010523	040	000		
1756	010525	040	040	122	EM0702:: .ASCIZ / RX.DATA.VALID BIT FOUND CLEAR AFTER RESET COMPLETION. '
	010530	130	056	104	
	010533	101	124	101	
	010536	056	126	101	
	010541	114	111	104	
	010544	040	102	111	
	010547	124	040	106	
	010552	117	125	116	
	010555	104	040	103	
	010560	114	105	101	
	010563	122	040	101	
	010566	106	124	105	
	010571	122	040	122	
	010574	105	123	105	
	010577	124	040	103	
	010602	117	115	120	
	010605	114	105	124	
	010610	111	117	116	
	010613	056	000		
1757	010615	040	040	040	.ASCIZ / EXPECTED BIT TO BE SET FROM SELFTEST CODES IN FIFO. /
	010620	040	105	130	
	010623	120	105	103	
	010626	124	105	104	
	010631	040	102	111	
	010634	124	040	124	
	010637	117	040	102	
	010642	105	040	123	
	010645	105	124	040	
	010650	106	122	117	
	010653	115	040	123	
	010656	105	114	106	
	010661	124	105	123	
	010664	124	040	103	
	010667	117	104	105	
	010672	123	040	111	
	010675	116	040	106	

	010700	111	106	117	
	010703	056	000		
1758	010705	040	040	122	EM0703:: .ASCIIZ / RX.DATA.VAL ID BIT COULD NOT BE CLEARED BY PAR,IN, FIFO./
	010710	130	056	104	
	010713	101	124	101	
	010716	056	126	101	
	010721	114	111	104	
	010724	040	102	111	
	010727	124	040	103	
	010732	117	125	114	
	010735	104	040	116	
	010740	117	124	040	
	010743	102	105	040	
	010746	103	114	105	
	010751	101	122	105	
	010754	104	040	102	
	010757	131	040	120	
	010762	125	122	107	
	010765	111	116	107	
	010770	040	106	111	
	010773	106	117	056	
	010776	000			
1759	010777	040	040	040	.ASCIIZ / 600 CHARS READ FROM FIFO WITHOUT R.D.V BIT CLEARING./
	011002	040	066	060	
	011005	060	040	103	
	011010	110	101	122	
	011013	123	040	122	
	011016	105	101	104	
	011021	040	106	122	
	011024	117	115	040	
	011027	106	111	106	
	011032	117	040	127	
	011035	111	124	110	
	011040	117	125	124	
	011043	040	122	056	
	011046	104	056	126	
	011051	040	102	111	
	011054	124	040	103	
	011057	114	105	101	
	011062	122	111	116	
	011065	107	056	000	
1760	011070	122	102	125	EM0801:: .ASCIIZ /RBUF RX.LINE.NUMBER FIELD TEST /
	011073	106	040	122	
	011076	130	056	114	
	011101	111	116	105	
	011104	056	116	125	
	011107	115	102	105	
	011112	122	040	106	
	011115	111	105	114	
	011120	104	040	124	
	011123	105	123	124	
	011126	040	000		
1761	011130	040	040	114	EM0802:: .ASCIIZ / LINE NUMBER IS WRONG ON A SELFTEST CODE./
	011133	111	116	105	
	011136	040	116	125	
	011141	115	102	105	
	011144	122	040	111	

	011147	123	040	127	
	011152	122	117	116	
	011155	107	040	117	
	011160	116	040	101	
	011163	040	123	105	
	011166	114	106	124	
	011171	105	123	124	
	011174	040	103	117	
	011177	104	105	056	
	011202	000			
1762	011203	103	110	105	EM0901:: .ASCIIZ /CHECK FOR BMP CODES TEST/
	011206	103	113	040	
	011211	106	117	122	
	011214	040	102	115	
	011217	120	137	103	
	011222	117	104	105	
	011225	123	040	124	
	011230	105	123	124	
	011233	000			
1763	011234	125	116	105	EM0902:: .ASCIIZ /UNEXPECTED BMP CODES FOUND /
	011237	130	120	105	
	011242	103	124	105	
	011245	104	040	102	
	011250	115	120	040	
	011253	103	117	104	
	011256	105	123	040	
	011261	106	117	125	
	011264	116	104	056	
	011267	000			
1764	011270	104	111	101	EM1001:: .ASCIIZ /DIAGNOSTIC FAIL (SKP SELFTEST) TEST/
	011273	107	116	117	
	011276	123	124	111	
	011301	103	040	106	
	011304	101	111	114	
	011307	040	050	123	
	011312	113	120	040	
	011315	123	105	114	
	011320	106	124	105	
	011323	123	124	051	
	011326	040	124	105	
	011331	123	124	000	
1765	011334	040	040	123	EM1002:: .ASCIIZ / SKIP SELF TEST TOOK TOO LONG TO COMPLETE, > 50 MS./
	011337	113	111	120	
	011342	040	123	105	
	011345	114	106	055	
	011350	124	105	123	
	011353	124	040	124	
	011356	117	117	113	
	011361	040	124	117	
	011364	117	040	114	
	011367	117	116	107	
	011372	040	124	117	
	011375	040	103	117	
	011400	115	120	114	
	011403	105	124	105	
	011406	054	040	076	
	011411	040	065	060	

	011414	040	115	123	
	011417	056	000		
1766	011421	040	040	123	EM1003:: .ASCIZ / SKIP SELF TEST COMPLETED TOO SOON, < 10 MS./
	011424	113	111	120	
	011427	040	123	105	
	011432	114	106	055	
	011435	124	105	123	
	011440	124	040	103	
	011443	117	115	120	
	011446	114	105	124	
	011451	105	104	040	
	011454	124	117	117	
	011457	040	123	117	
	011462	117	116	054	
	011465	040	074	040	
	011470	061	060	040	
	011473	115	123	056	
	011476	000			
1767	011477	123	113	111	EM1101:: .ASCIZ /SKIP SELF TEST TEST/
	011502	120	040	123	
	011505	105	114	106	
	011510	055	124	105	
	011513	123	124	040	
	011516	124	105	123	
	011521	124	000		
1768	011523	123	105	114	EM1201:: .ASCIZ /SELF TEST TEST/
	011526	106	055	124	
	011531	105	123	124	
	011534	040	124	105	
	011537	123	124	000	
1769	011542	040	040	123	EM1202:: .ASCIZ / SELF-TEST TOOK TOO LONG TO COMPLETE, > 3 SECONDS./
	011545	105	114	106	
	011550	055	124	105	
	011553	123	124	040	
	011556	124	117	117	
	011561	113	040	124	
	011564	117	117	040	
	011567	114	117	116	
	011572	107	040	124	
	011575	117	040	103	
	011600	117	115	120	
	011603	114	105	124	
	011606	105	054	040	
	011611	076	040	063	
	011614	040	123	105	
	011617	103	117	116	
	011622	104	123	056	
	011625	000			
1770	011626	040	040	123	EM1203:: .ASCIZ \ SELF TEST COMPLETED TOO SOON, < 1/2 SECOND.\
	011631	105	114	106	
	011634	055	124	105	
	011637	123	124	040	
	011642	103	117	115	
	011645	120	114	105	
	011650	124	105	104	
	011653	040	124	117	
	011656	117	040	123	

	011661	117	117	116	
	011664	054	040	074	
	011667	040	061	057	
	011672	062	040	123	
	011675	105	103	117	
	011700	116	104	056	
	011703	000			
1771	011704	040	040	123	EM1204:: .ASCIZ / SELF TEST DID NOT EXECUTE/
	011707	105	114	106	
	011712	055	124	105	
	011715	123	124	040	
	011720	104	111	104	
	011723	040	116	117	
	011726	124	040	105	
	011731	130	105	103	
	011734	125	124	105	
	011737	000			
1772	011740	040	040	104	EM1205:: .ASCIZ / DIAG_FAIL BIT BAD/
	011743	111	101	107	
	011746	137	106	101	
	011751	111	114	040	
	011754	102	111	124	
	011757	040	102	101	
	011762	104	000		
1773	011764	106	101	111	EM1301:: .ASCIZ /FAIL SELF-TEST TEST/
	011767	114	040	123	
	011772	105	114	106	
	011775	055	124	105	
	012000	123	124	040	
	012003	124	105	123	
	012006	124	000		
1774	012010	040	123	105	EM1302:: .ASCIZ / SELF-TEST ERROR REPORTING BAD/
	012013	114	106	055	
	012016	124	105	123	
	012021	124	040	105	
	012024	122	122	117	
	012027	122	040	122	
	012032	105	120	117	
	012035	122	124	111	
	012040	116	107	040	
	012043	102	101	104	
	012046	000			
1775	012047	122	117	115	EM1401:: .ASCIZ /ROM VERSION NUMBER TEST/
	012052	040	126	105	
	012055	122	123	111	
	012060	117	116	137	
	012063	116	125	115	
	012066	102	105	122	
	012071	040	124	105	
	012074	123	124	000	
1776	012077	040	040	106	EM1402:: .ASCIZ / FIFO EMPTY, ONE OR MORE ROM VERSION NUMBERS MISSING/
	012102	111	106	117	
	012105	040	105	115	
	012110	120	124	131	
	012113	054	040	117	
	012116	116	105	040	
	012121	117	122	040	

	012124	115	117	122	
	012127	105	040	122	
	012132	117	115	040	
	012135	126	105	122	
	012140	123	111	117	
	012143	116	137	116	
	012146	125	115	102	
	012151	105	122	123	
	012154	040	115	111	
	012157	123	123	111	
	012162	116	107	000	
1777	012165	040	040	122	EM1403:: .ASCIZ / ROM VERSION NUMBER FOUND OUT OF SEQUENCE/
	012170	117	115	040	
	012173	126	105	122	
	012176	123	111	117	
	012201	116	137	116	
	012204	125	115	102	
	012207	105	122	040	
	012212	106	117	125	
	012215	116	104	040	
	012220	117	125	124	
	012223	040	117	106	
	012226	040	123	105	
	012231	121	125	105	
	012234	116	103	105	
	012237	000			
1778	012240	040	040	117	EM1404:: .ASCIZ / ONE OR MORE ROM VERSION NUMBERS MISSING/
	012243	116	105	040	
	012246	117	122	040	
	012251	115	117	122	
	012254	105	040	122	
	012257	117	115	040	
	012262	126	105	122	
	012265	123	111	117	
	012270	116	137	116	
	012273	125	115	102	
	012276	105	122	123	
	012301	040	115	111	
	012304	123	123	111	
	012307	116	107	000	
1779	012312	040	040	040	EM1405:: .ASCIZ / PROC_1/
	012315	040	120	122	
	012320	117	103	137	
	012323	061	000		
1780	012325	040	040	040	EM1406:: .ASCIZ / PROC 2/
	012330	040	120	122	
	012333	117	103	137	
	012336	062	000		
1781	012340	116	117	124	EM1407:: .ASCIZ /NOT FOUND/
	012343	040	106	117	
	012346	125	116	104	
	012351	000			
1782	012352	106	117	125	EM1408:: .ASCIZ /FOUND/
	012355	116	104	000	
1783	012360	124	111	115	EM1601:: .ASCIZ /TIMEOUT OCCURRED WAITING FOR MASTER RESET TO CLEAR/
	012363	105	117	125	
	012366	124	040	117	

	012371	103	103	125	
	012374	122	122	105	
	012377	104	040	127	
	012402	101	111	124	
	012405	111	116	107	
	012410	040	106	117	
	012413	122	040	115	
	012416	101	123	124	
	012421	105	122	040	
	012424	122	105	123	
	012427	105	124	040	
	012432	124	117	040	
	012435	103	114	105	
	012440	101	122	000	
1784	012443	104	105	126	EM1604:: .ASCIZ \DEVICE REGISTER WORD READ/WRITE TEST \
	012446	111	103	105	
	012451	040	122	105	
	012454	107	111	123	
	012457	124	105	122	
	012462	040	127	117	
	012465	122	104	040	
	012470	122	105	101	
	012473	104	057	127	
	012476	122	111	124	
	012501	105	040	124	
	012504	105	123	124	
	012507	040	000		
1785	012511	104	105	126	EM1701:: .ASCIZ \DEVICE REGISTER WORD READ/MODIFY/WRITE TEST \
	012514	111	103	105	
	012517	040	122	105	
	012522	107	111	123	
	012525	124	105	122	
	012530	040	127	117	
	012533	122	104	040	
	012536	122	105	101	
	012541	104	057	115	
	012544	117	104	111	
	012547	106	131	057	
	012552	127	122	111	
	012555	124	105	040	
	012560	124	105	123	
	012563	124	040	000	
1786	012566	104	105	126	EM1801:: .ASCIZ \DEVICE REGISTER BYTE READ/WRITE TEST \
	012571	111	103	105	
	012574	040	122	105	
	012577	107	111	123	
	012602	124	105	122	
	012605	040	102	131	
	012610	124	105	040	
	012613	122	105	101	
	012616	104	057	127	
	012621	122	111	124	
	012624	105	040	124	
	012627	105	123	124	
	012632	040	000		
1787	012634	104	105	126	EM1901:: .ASCIZ \DEVICE REGISTER BYTE READ/MODIFY/WRITE TEST \
	012637	111	103	105	

GLOBAL MESSAGE AREA

	012642	040	122	105	
	012645	107	111	123	
	012650	124	105	122	
	012653	040	102	131	
	012656	124	105	040	
	012661	122	105	101	
	012664	104	057	115	
	012667	117	104	111	
	012672	106	131	057	
	012675	127	122	111	
	012700	124	105	040	
	012703	124	105	123	
	012706	124	040	000	
1788	012711	104	105	126	EM2001:: .ASCIZ /DEVICE STAT REGISTER ID BIT TEST /
	012714	111	103	105	
	012717	040	123	124	
	012722	101	124	040	
	012725	122	105	107	
	012730	111	123	124	
	012733	105	122	040	
	012736	111	104	040	
	012741	102	111	124	
	012744	040	124	105	
	012747	123	124	040	
	012752	000			
1789	012753	111	104	040	EM2002:: .ASCIZ /ID BIT BAD. EXPECTED: CLEAR, ACTUAL: SET./
	012756	102	111	124	
	012761	040	102	101	
	012764	104	056	040	
	012767	040	105	130	
	012772	120	105	103	
	012775	124	105	104	
	013000	072	040	103	
	013003	114	105	101	
	013006	122	054	040	
	013011	101	103	124	
	013014	125	101	114	
	013017	072	040	123	
	013022	105	124	056	
	013025	000			
1790	013026	116	117	040	EM2101:: .ASCIZ \NO TX_DATA_VALID/NO TX_ACTION TEST\
	013031	124	130	137	
	013034	104	101	124	
	013037	101	137	126	
	013042	101	114	111	
	013045	104	057	116	
	013050	117	040	124	
	013053	130	137	101	
	013056	103	124	111	
	013061	117	116	040	
	013064	124	105	123	
	013067	124	000		
1791	013071	040	040	124	EM2102:: .ASCIZ / TX_ACTION FOUND AFTER INVALID DATA WORD WRITTEN TO LINE:
	013074	130	137	101	
	013077	103	124	111	
	013102	117	116	040	
	013105	106	117	125	

185

	013110	116	104	040	
	013113	101	106	124	
	013116	105	122	040	
	013121	111	116	126	
	013124	101	114	111	
	013127	104	040	104	
	013132	101	124	101	
	013135	040	127	117	
	013140	122	104	040	
	013143	127	122	111	
	013146	124	124	105	
	013151	116	040	124	
	013154	117	040	114	
	013157	111	116	105	
	013162	072	040	000	
1792	013165	124	130	137	EM2201:: .ASCIZ \TX_DATA_VALID/TX ACTION TEST\
	013170	104	101	124	
	013173	101	137	126	
	013176	101	114	111	
	013201	104	057	124	
	013204	130	137	101	
	013207	103	124	111	
	013212	117	116	040	
	013215	124	105	123	
	013220	124	000		
1793	013222	040	040	116	EM2202:: .ASCIZ / NO TX ACTION FOUND AFTER VALID DATA WORD TX'D ON LINE: /
	013225	117	040	124	
	013230	130	137	101	
	013233	103	124	111	
	013236	117	116	040	
	013241	106	117	125	
	013244	116	104	040	
	013247	101	106	124	
	013252	105	122	040	
	013255	126	101	114	
	013260	111	104	040	
	013263	104	101	124	
	013266	101	040	127	
	013271	117	122	104	
	013274	040	124	130	
	013277	047	104	040	
	013302	117	116	040	
	013305	114	111	116	
	013310	105	072	040	
	013313	000			
1794	013314	040	040	111	EM2203:: .ASCIZ / INCORRECT LINE NUMBER FOUND WITH TX ACT AFTER DATA TX D ON LINE :
	013317	116	103	117	
	013322	122	122	105	
	013325	103	124	040	
	013330	114	111	116	
	013333	105	040	116	
	013336	125	115	102	
	013341	105	122	040	
	013344	106	117	125	
	013347	116	104	040	
	013352	127	111	124	
	013355	110	040	124	

	013360	130	137	101	
	013363	103	124	040	
	013366	101	106	124	
	013371	105	122	040	
	013374	104	101	124	
	013377	101	040	124	
	013402	130	047	104	
	013405	040	117	116	
	013410	040	114	111	
	013413	116	105	040	
	013416	072	040	000	
1795	013421	124	130	137	EM2301:: .ASCIZ /TX ENABLE (INACTIVE) BIT TEST/
	013424	105	116	101	
	013427	102	114	105	
	013432	040	050	111	
	013435	116	101	103	
	013440	124	111	126	
	013443	105	051	040	
	013446	102	111	124	
	013451	040	124	105	
	013454	123	124	000	
1796	013457	040	040	124	EM2302:: .ASCIZ / TX ENABLE BIT BAD ON LINE: /
	013462	130	137	105	
	013465	116	101	102	
	013470	114	105	040	
	013473	102	111	124	
	013476	040	102	101	
	013501	104	040	117	
	013504	116	040	114	
	013507	111	116	105	
	013512	072	040	000	
1797	013515	124	130	137	EM2401:: .ASCIZ /TX ENABLE (ACTIVE) BIT TEST/
	013520	105	116	101	
	013523	102	114	105	
	013526	040	050	101	
	013531	103	124	111	
	013534	126	105	051	
	013537	040	102	111	
	013542	124	040	124	
	013545	105	123	124	
	013550	000			
1798	013551	122	105	103	EM2601:: .ASCIZ /RECEIVE INTERRUPT TEST /
	013554	105	111	126	
	013557	105	040	111	
	013562	116	124	105	
	013565	122	122	125	
	013570	120	124	040	
	013573	124	105	123	
	013576	124	040	000	
1799	013601	040	040	116	EM2602:: .ASCIZ / NO RX INT GENERATED (DATA_VALID SET, RX INTS ENABLED). /
	013604	117	040	122	
	013607	130	040	111	
	013612	116	124	040	
	013615	107	105	116	
	013620	105	122	101	
	013623	124	105	104	
	013626	040	050	104	

	013631	101	124	101	
	013634	137	126	101	
	013637	114	111	104	
	013642	040	123	105	
	013645	124	054	040	
	013650	122	130	040	
	013653	111	116	124	
	013656	123	040	105	
	013661	116	101	102	
	013664	114	105	104	
	013667	051	056	000	
1800	013672	040	040	116	EM2603:: .ASCIIZ / NO RX INT GENERATED (NO CODES IN FIFO AFTER RESET)./
	013675	117	040	122	
	013700	130	040	111	
	013703	116	124	040	
	013706	107	105	116	
	013711	105	122	101	
	013714	124	105	104	
	013717	040	050	116	
	013722	117	040	103	
	013725	117	104	105	
	013730	123	040	111	
	013733	116	040	106	
	013736	111	106	117	
	013741	040	101	106	
	013744	124	105	122	
	013747	040	122	105	
	013752	123	105	124	
	013755	051	056	000	
1801	013760	040	040	116	EM2604:: .ASCIIZ / NO RX INT GENERATED (RX DATA AVAIL CLR, RX INTS ENABLED)./
	013763	117	040	122	
	013766	130	040	111	
	013771	116	124	040	
	013774	107	105	116	
	013777	105	122	101	
	014002	124	105	104	
	014005	040	050	122	
	014010	130	137	104	
	014013	101	124	101	
	014016	137	101	126	
	014021	101	111	114	
	014024	040	103	114	
	014027	122	054	040	
	014032	122	130	040	
	014035	111	116	124	
	014040	123	040	105	
	014043	116	101	102	
	014046	114	105	104	
	014051	051	056	000	
1802	014054	040	040	122	EM2605:: .ASCIIZ / RX INTERRUPT GENERATED WITH RX_DATA AVAIL CLEAR./
	014057	130	040	111	
	014062	116	124	105	
	014065	122	122	125	
	014070	120	124	040	
	014073	107	105	116	
	014076	105	122	101	
	014101	124	105	104	

	014104	040	127	111	
	014107	124	110	040	
	014112	122	130	137	
	014115	104	101	124	
	014120	101	137	101	
	014123	126	101	111	
	014126	114	040	103	
	014131	114	105	101	
	014134	122	056	000	
1803	014137	124	122	101	EM2606:: .ASCIZ /TRANSMIT INTERRUPT TEST ERROR:/
	014142	116	123	115	
	014145	111	124	040	
	014150	111	116	124	
	014153	105	122	122	
	014156	125	120	124	
	014161	040	124	105	
	014164	123	124	040	
	014167	105	122	122	
	014172	117	122	072	
	014175	000			
1804	014176	040	040	124	EM2607:: .ASCIZ / TX_ACTION SET REPEATEDLY AFTER BOARD RESET, NO DATA SENT./
	014201	130	137	101	
	014204	103	124	111	
	014207	117	116	040	
	014212	123	105	124	
	014215	040	122	105	
	014220	120	105	101	
	014223	124	105	104	
	014226	114	131	040	
	014231	101	106	124	
	014234	105	122	040	
	014237	102	117	101	
	014242	122	104	040	
	014245	122	105	123	
	014250	105	124	054	
	014253	040	116	117	
	014256	040	104	101	
	014261	124	101	040	
	014264	123	105	116	
	014267	124	056	000	
1805	014272	040	040	124	EM2608:: .ASCIZ / TX_ACTION STUCK SET AFTER BOARD RESET./
	014275	130	137	101	
	014300	103	124	111	
	014303	117	116	040	
	014306	123	124	125	
	014311	103	113	040	
	014314	123	105	124	
	014317	040	101	106	
	014322	124	105	122	
	014325	040	102	117	
	014330	101	122	104	
	014333	040	122	105	
	014336	123	105	124	
	014341	056	000		
1806	014343	040	040	124	EM2609:: .ASCIZ / TX INTERRUPT GENERATED WITH TX_ACTION CLEAR./
	014346	130	040	111	
	014351	116	124	105	

	014354	122	122	125	
	014357	120	124	040	
	014362	107	105	116	
	014365	105	122	101	
	014370	124	105	104	
	014373	040	127	111	
	014376	124	110	040	
	014401	124	130	137	
	014404	101	103	124	
	014407	111	117	116	
	014412	040	103	114	
	014415	105	101	122	
	014420	056	000		
1807	014422	040	040	116	EM2610:: .ASCIZ / NO TX INTERRUPT WITH TX ACTION SET AND TX INTS ENABLED./
	014425	117	040	124	
	014430	130	040	111	
	014433	116	124	105	
	014436	122	122	125	
	014441	120	124	040	
	014444	127	111	124	
	014447	110	040	124	
	014452	130	137	101	
	014455	103	124	111	
	014460	117	116	040	
	014463	123	105	124	
	014466	040	101	116	
	014471	104	040	124	
	014474	130	040	111	
	014477	116	124	123	
	014502	040	105	116	
	014505	101	102	114	
	014510	105	104	056	
	014513	000			
1808	014514	040	040	124	EM2611:: .ASCIZ / TX ACTION NOT SET AFTER CHARS SENT ON ALL LINES./
	014517	130	137	101	
	014522	103	124	111	
	014525	117	116	040	
	014530	116	117	124	
	014533	040	123	105	
	014536	124	040	101	
	014541	106	124	105	
	014544	122	040	103	
	014547	110	101	122	
	014552	123	040	123	
	014555	105	116	124	
	014560	040	117	116	
	014563	040	101	114	
	014566	114	040	114	
	014571	111	116	105	
	014574	123	056	000	
1809	014577	111	116	124	EM3001:: .ASCIZ / INTERRUPT BR LEVEL TEST /
	014602	105	122	122	
	014605	125	120	124	
	014610	040	102	122	
	014613	040	114	105	
	014616	126	105	114	
	014621	040	124	105	

	014624	123	124	040	
	014627	000			
1810	014630	040	040	116	EM3002:: .ASCIZ / NO RX DATA AVAIL FROM SELF TEST CODES IN FIFO AFTER RESET./
	014633	117	040	122	
	014636	130	137	104	
	014641	101	124	101	
	014644	137	101	126	
	014647	101	111	114	
	014652	040	106	122	
	014655	117	115	040	
	014660	123	105	114	
	014663	106	124	105	
	014666	123	124	040	
	014671	103	117	104	
	014674	105	123	040	
	014677	111	116	040	
	014702	106	111	106	
	014705	117	040	101	
	014710	106	124	105	
	014713	122	040	122	
	014716	105	123	105	
	014721	124	056	000	
1811	014724	040	040	124	EM3003:: .ASCIZ / TX INTERRUPT GENERATED AT WRONG BR LEVEL:/
	014727	130	040	111	
	014732	116	124	105	
	014735	122	127	125	
	014740	120	124	040	
	014743	107	105	116	
	014746	105	122	101	
	014751	124	105	104	
	014754	040	101	124	
	014757	040	127	122	
	014762	117	116	107	
	014765	040	102	122	
	014770	040	114	105	
	014773	126	105	114	
	014776	072	000		
1812	015000	040	040	122	EM3004:: .ASCIZ / RX INTERRUPT GENERATED AT WRONG BR LEVEL:/
	015003	130	040	111	
	015006	116	124	105	
	015011	122	122	125	
	015014	120	124	040	
	015017	107	105	116	
	015022	105	122	101	
	015025	124	105	104	
	015030	040	101	124	
	015033	040	127	122	
	015036	117	116	107	
	015041	040	102	122	
	015044	040	114	105	
	015047	126	105	114	
	015052	072	000		
1813	015054	040	040	124	EM3005:: .ASCIZ / TX INTERRUPT GIVEN PRECEDENCE OVER SIMULTANEOUS RX INT./
	015057	130	040	111	
	015062	116	124	105	
	015065	122	122	125	
	015070	120	124	040	

106,

CVDHABO DHV 11 FUNC TST PART2 MACHU M1200 10 OCT 83 11:55 PAGE 35 22
GLOBAL MESSAGE AREA

3ED 6A

	015073	107	111	126	
	015076	105	116	040	
	015101	120	122	105	
	015104	103	105	104	
	015107	105	116	103	
	015112	105	040	117	
	015115	126	105	122	
	015120	040	123	111	
	015123	115	125	114	
	015126	124	101	116	
	015131	105	117	125	
	015134	123	040	122	
	015137	130	040	111	
	015142	116	124	056	
	015145	000			
1814	015146	104	111	101	EM3101:: .ASCIIZ /DIAGNOSTIC FIELD (BMP) TEST/
	015151	107	116	117	
	015154	123	124	111	
	015157	103	040	106	
	015162	111	105	114	
	015165	104	040	050	
	015170	102	115	120	
	015173	051	040	124	
	015176	105	123	124	
	015201	000			
1815	015202	040	040	104	EM3102:: .ASCIIZ / DIAGNOSTIC FIELD BAD ON LINE: /
	015205	111	101	107	
	015210	116	117	123	
	015213	124	111	103	
	015216	040	106	111	
	015221	105	114	104	
	015224	040	102	101	
	015227	104	040	117	
	015232	116	040	114	
	015235	111	116	105	
	015240	072	040	000	
1816	015243	105	130	120	EM9009:: .ASCIIZ /EXPECTED OR CORRECT/
	015246	105	103	124	
	015251	105	104	040	
	015254	117	122	040	
	015257	103	117	122	
	015262	122	105	103	
	015265	124	000		
1817	015267	101	103	124	EM9010:: .ASCIIZ /ACTUAL OR MEASURED /
	015272	125	101	114	
	015275	040	117	122	
	015300	040	115	105	
	015303	101	123	125	
	015306	122	105	104	
	015311	040	000		
1818	015313	123	125	115	EM9014:: .ASCIIZ /SUMMARY REPORTS FOR LINES WITH EXCESSIVE NUMBERS OF ERRORS:/
	015316	115	101	122	
	015321	131	040	122	
	015324	105	120	117	
	015327	122	124	123	
	015332	040	106	117	
	015335	122	040	114	

F6

	015340	111	116	105	
	015343	123	040	127	
	015346	111	124	110	
	015351	040	105	130	
	015354	103	105	123	
	015357	123	111	126	
	015362	105	040	116	
	015365	125	115	102	
	015370	105	122	123	
	015373	040	117	106	
	015376	040	105	122	
	015401	122	117	122	
	015404	123	072	000	
1819	015407	040	040	106	EM9017:: .ASCII / FIFO WILL NOT PURGE (DATA.VALID STUCK SET)./
	015412	111	106	117	
	015415	040	127	111	
	015420	114	114	040	
	015423	116	117	124	
	015426	040	120	125	
	015431	122	107	105	
	015434	040	050	104	
	015437	101	124	101	
	015442	056	126	101	
	015445	114	111	104	
	015450	040	123	124	
	015453	125	103	113	
	015456	040	123	105	
	015461	124	051	054	
1820	015464	040	122	105	.ASCIIZ / REMAINDER OF TEST SKIPPED./
	015467	115	101	111	
	015472	116	104	105	
	015475	122	040	117	
	015500	106	040	124	
	015503	105	123	124	
	015506	040	123	113	
	015511	111	120	120	
	015514	105	104	056	
	015517	000			
1821	015520	116	117	040	EM9018:: .ASCIIZ /NO CODE/
	015523	103	117	104	
	015526	105	000		
1822	015530	116	117	116	EM9019:: .ASCIIZ /NON SELFTEST/
	015533	055	123	105	
	015536	114	106	124	
	015541	105	123	124	
	015544	000			
1823	015545	123	105	114	EM9020:: .ASCIIZ /SELFTEST ERROR CODE/
	015550	106	124	105	
	015553	123	124	040	
	015556	105	122	122	
	015561	117	122	040	
	015564	103	117	104	
	015567	105	000		
1824	015571	104	101	124	EM9022:: .ASCIIZ /DATA CHARACTER/
	015574	101	040	103	
	015577	110	101	122	
	015602	101	103	124	

	015605	105	122	000	
1825	015610	115	117	104	EM9023:: .ASCIZ /MODEM STATUS CODE/
	015613	105	115	040	
	015616	123	124	101	
	015621	124	125	123	
	015624	040	103	117	
	015627	104	105	000	
1826	015632	123	105	114	EM9024:: .ASCIZ /SELFTEST CODE/
	015635	106	124	105	
	015640	123	124	040	
	015643	103	117	104	
	015646	105	000		
1827	015650	040	040	040	EM9026:: .ASCIZ / LPR CONTENTS: /
	015653	040	040	114	
	015656	120	122	040	
	015661	103	117	116	
	015664	124	105	116	
	015667	124	123	072	
	015672	040	000		
1828	015674	102	115	120	EM9301:: .ASCIZ /BMP CODE REPORT/
	015677	040	103	117	
	015702	104	105	040	
	015705	122	105	120	
	015710	117	122	124	
	015713	000			
1829	015714	102	115	120	EM9302:: .ASCIZ /BMP CODE FOUND IN TEST /
	015717	040	103	117	
	015722	104	105	040	
	015725	106	117	125	
	015730	116	104	040	
	015733	111	116	040	
	015736	124	105	123	
	015741	124	040	000	
1830	015744	124	110	105	EM9303:: .ASCIZ /THE LAST BMP CODE WAS FOUND IN TEST /
	015747	040	114	101	
	015752	123	124	040	
	015755	102	115	120	
	015760	040	103	117	
	015763	104	105	040	
	015766	127	101	123	
	015771	040	106	117	
	015774	125	116	104	
	015777	040	111	116	
	016002	040	124	105	
	016005	123	124	040	
	016010	000			
1831	016011	125	116	105	EM9304:: .ASCIZ /UNEXPECTED BMP CODES FOUND DURING THIS PASS/
	016014	130	120	105	
	016017	103	124	105	
	016022	104	040	102	
	016025	115	120	040	
	016030	103	117	104	
	016033	105	123	040	
	016036	106	117	125	
	016041	116	104	040	
	016044	104	125	122	
	016047	111	116	107	

CVDHABO DMV 11 FUNC 1ST PART2 MACRO M1200 10 OCT 83 11:55 PAGE 35 25
GLOBAL MESSAGE AREA

SEQ 11

016052	040	124	110
016055	111	123	040
016060	120	101	123
016063	123	000	

1432

.EVEN

H6

1834
1835
1836
1837
1838
1839
1840
1841
1842
1843
1844
1845
1846
1847
1848
1849
1850

```
;*****  
;  
;          FVTSKL2.P11  
;  
;*****  
  
.SBTTL  GLOBAL ERROR REPORT SECTION  
  
;**  
; THE GLOBAL ERROR REPORT SECTION CONTAINS MESSAGE PRINTING AREAS  
; USED BY MORE THAN ONE TEST TO OUTPUT ADDITIONAL ERROR INFORMATION. PRINTB  
; (BASIC) AND PRINTX (EXTENDED) CALLS ARE USED TO CALL PRINT SERVICES.  
; -
```

```
1852 .SBTTL GLOBAL ERROR REPORTING ROUTINE ER0101
1853 ;*****
1854 ;* THIS IS AN ERROR REPORTING SUBROUTINE WHICH PRINTS ADDITIONAL ERROR
1855 ;* INFORMATION IF AN ERROR IS DETECTED IN TEST 1 (REGISTER ADDRESS
1856 ;* ACCESS TEST). THIS SUBROUTINE REPORTS THE TYPE OF ACCESS (READ OR
1857 ;* WRITE OR BOTH) WHICH CAUSED A BUS TIME OUT TRAP (004 TRAP).
1858 ;* A MESSAGE INDICATING THAT THE DHV MAY BE AT THE WRONG Q BUS ADDRESS
1859 ;* IS ALSO PRINTED.
1860 ;*
1861 ;* INPUTS: R5 ERROR FLAG WORD.
1862 ;* IF BIT 0 IS SET, A READ ERROR OCCURED.
1863 ;* IF BIT 1 IS SET, A WRITE ERROR OCCURED.
1864 ;*
1865 ;* OUTPUTS: MESSAGES ARE PRINTED AT THE OPERATOR CONSOLE.
1866 ;*
1867 ;* CALLING SEQUENCE: INCLUDE THE LABEL "ER0101" AS THE MESSAGE POINTER
1868 ;* PARAMETER IN THE DRS ERROR REPORT MACRO CALL.
1869 ;*
1870 ;* COMMENTS:
1871 ;*
1872 ;* SUBORDINATE ROUTINES USED: NONE.
1873 ;*****
1874
1875 016066 BGNMSG ER0101
1876 016066 ER0101::
1877 016066 SAVE ;SAVE THE GPR CONTENTS.
1878 016066 JSR R5,PREG05 ;CALL REGISTER SAVE SUBRT.
1879 016072 032705 000001 BIT #BIT0,R5 ;TEST FOR READ ERROR.
1880 016076 001410 BEQ 2$ ;SKIP READ ERROR MSG IF NO READ ERROR.
1881 016100 012746 016172 PRINTB #MSG1 ;PRINT READ ERROR MESSAGE.
1882 016104 012746 000001 MOV #MSG1,-(SP)
1883 016110 010600 MOV #1,(SP)
1884 016112 104414 MOV SP,R0
1885 016114 062706 000004 TRAP C$PNTB
1886 016120 032705 000002 2$: BIT #BIT1,R5 ;TEST FOR WRITE ERROR.
1887 016124 001410 BEQ 4$ ;SKIP WRITE ERROR MSG IF NO WRITE ERROR.
1888 016126 012746 016250 PRINTB #MSG2 ;PRINT WRITE ERROR MESSAGE.
1889 016132 012746 000001 MOV #MSG2,-(SP)
1890 016136 010600 MOV #1,(SP)
1891 016140 104414 MOV SP,R0
1892 016142 062706 000004 TRAP C$PNTB
1893 016146 012746 016327 4$: PRINTX #MSG3 ;SUGGEST THAT DHV MAY BE AT WRONG ADDRESS.
1894 016152 012746 000001 MOV #MSG3,-(SP)
1895 016156 010600 MOV #1,-(SP)
1896 016160 104415 MOV SP,R0
1897 016162 062706 000004 TRAP C$PNTX
1898 016166 004736 PASS ;RESTORE THE GPR CONTENTS.
1899 016170 004736 JSR PC,@(SP)+ ;RETURN TO PREG05 SUBRT.
1900 016170 104423 L10002: TRAP C$MSG
1901 016172 045 101 102 MSG1:: .ASCIZ /#ABUS TIME OUT TRAP CAUSED BY READ ATTEMPT.#N/
```

J6

CVDHABO DHV 11 FUNC TST PART2
GLOBAL ERROR REPORTING ROUTINE

MACRO M1200 10 OCT 83 11:35 PAGE 37 1
ER0101

SEQ 74

	016175	125	123	040	
	016200	124	111	115	
	016203	105	055	117	
	016206	125	124	040	
	016211	124	122	101	
	016214	120	040	103	
	016217	101	125	123	
	016222	105	104	040	
	016225	102	131	040	
	016230	122	105	101	
	016233	104	040	101	
	016236	124	124	105	
	016241	115	120	124	
	016244	056	045	116	
	016247	000			
1889	016250	045	101	102	MSG2:: .ASCIZ /#ABUS TIME OUT TRAP CAUSED BY WRITE ATTEMPT.#N/
	016253	125	123	040	
	016256	124	111	115	
	016261	105	055	117	
	016264	125	124	040	
	016267	124	122	101	
	016272	120	040	103	
	016275	101	125	123	
	016300	105	104	040	
	016303	102	131	040	
	016306	127	122	111	
	016311	124	105	040	
	016314	101	124	124	
	016317	105	115	120	
	016322	124	056	045	
	016325	116	000		
1890	016327	045	101	10'	MSG3:: .ASCIZ /#ADHV MAY BE AT THE WRONG Q BUS ADDRESS.#N#N/
	016332	110	126	040	
	016335	115	101	131	
	016340	040	102	105	
	016343	040	101	124	
	016346	040	124	110	
	016351	105	040	127	
	016354	122	117	116	
	016357	107	040	121	
	016362	055	102	125	
	016365	123	040	101	
	016370	104	104	122	
	016373	105	123	123	
	016376	056	045	116	
	016401	045	116	000	
1891					
1892					.EVEN

```
1894 .SBTTL GLOBAL ERROR REPORTING ROUTINE - ERO201
1895 ;*****
1896 ;* THIS IS AN ERROR REPORTING SUBROUTINE WHICH PRINTS 2 CONTIGUOUS
1897 ;* ASCII ERROR MESSAGES. THE ADDRESS OF THE FIRST MESSAGE IS PASSED
1898 ;* AS AN INPUT PARAMETER AND THE ADDRESS OF THE SECOND IS FOUND BY
1899 ;* SEARCHING FOR THE END OF THE FIRST MESSAGE.
1900 ;*
1901 ;* INPUTS: R1 ADDRESS OF THE FIRST MESSAGE TO PRINT.
1902 ;*
1903 ;* OUTPUTS: A MESSAGES IS PRINTED AT THE OPERATOR CONSOLE.
1904 ;*
1905 ;* CALLING SEQUENCE: LOAD THE ADDRESS OF THE FIRST MESSAGE IN R1.
1906 ;* INCLUDE THE LABEL "ERO201" AS THE MESSAGE POINTER
1907 ;* PARAMETER IN THE DIAG SUPER ERROR REPORT MACRO CALL.
1908 ;*
1909 ;* COMMENTS: THE MESSAGE IS PRINTED AS BASIC ERROR INFORMATION.
1910 ;* THE SECOND MESSAGE SHOULD FOLLOW THE FIRST ONE IN THE PROGRAM
1911 ;* MEMORY. EACH MESSAGE SHOULD BE DEFINED USING .ASCIZ
1912 ;*
1913 ;* SUBORDINATE ROUTINES USED: NONE.
1914 ;*****
1915
1916 016404 BGNMSG ERO201
1917 016404 ERO201::
1918 016404 004567 165366 SAVE JSR ;SAVE THE GPR CONTENTS.
1919 016410 010102 ;R5,PREG05 ;CALL REGISTER SAVE SUBRT.
1920 016412 105722 2$: MOV R1,R2 ;CHECK FOR A ZERO BYTE (END OF MESSAGE).
1921 016414 001376 TSTB (R2); ;LOOP UNTIL NEXT MESSAGE IS FOUND.
1922 BNE 2$
1923 016416 PRINTB 0EF0503,R1 ;PRINT THE FIRST MESSAGE.
1924 016416 010146 MOV R1,(SP)
016420 012746 004132 MOV 0EF0503,-(SP)
016424 012746 000002 MOV 02,-(SP)
016430 010600 MOV SP,R0
016432 104414 TRAP C$PNTB
016434 062706 000006 ADD 06,SP
1924 016440 PRINTB 0EF0503,R2 ;PRINT THE SECOND MESSAGE.
1925 016440 010246 MOV R2,-(SP)
1926 016442 012746 004132 MOV 0EF0503,-(SP)
1927 016446 012746 000002 MOV 02,-(SP)
1928 016452 010600 MOV SP,R0
016454 104414 TRAP C$PNTB
016456 062706 000006 ADD 06,SP
1925
1926 016462 PASS JSR ;RESTORE THE GPR CONTENTS.
1927 016462 004736 JSR PC,0(SP); ;RETURN TO PREG05 SUBRT.
1928 016464 ENDMSG
016464 L10003: TRAP C$MSG
016464 104423
```

```

1930 .SBTTL GLOBAL ERROR REPORTING ROUTINE - ER0503
1931 ;*****
1932 ;* THIS IS AN ERROR REPORTING SUBROUTINE WHICH PRINTS AN ADDITIONAL ERROR
1933 ;* MESSAGE WHOSE ADDRESS IS PASSED AS AN INPUT PARAMETER.
1934 ;*
1935 ;* INPUTS: R1 - ADDRESS OF THE MESSAGE TO PRINT.
1936 ;*
1937 ;* OUTPUTS: A MESSAGE IS PRINTED AT THE OPERATOR CONSOLE.
1938 ;*
1939 ;* CALLING SEQUENCE: LOAD THE ADDRESS OF THE MESSAGE IN R1.
1940 ;* INCLUDE THE LABEL "ER0503" AS THE MESSAGE POINTER
1941 ;* PARAMETER IN THE DIAG SUPER ERROR REPORT MACRO CALL.
1942 ;*
1943 ;* COMMENTS: THE MESSAGE IS PRINTED AS BASIC ERROR INFORMATION.
1944 ;*
1945 ;* JL JINATE ROUTINES USED: NONE.
1946 ;*****
1947
1948 016466 BGNMSG ER0503
1949 016466 ER0503::
1950 016466 PRINTB #EF0503,R1 ;PR: THE MESSAGE.
1951 016466 010146 MOV R1,-(SP)
1952 016470 012746 004132 MOV #EF0503,-(SP)
1953 016474 012746 000002 MOV #2,-(SP)
1954 016500 010600 MOV SP,R0
1955 016502 104414 TRAP C$PNTB
1956 016504 062706 000006 ADD #6,SP
1957
1958 ENDMSG
1959
1960 L10004:
1961 TRAP C$MSG

```

```

1954 .SBTTL GLOBAL ERROR REPORTING ROUTINE - ER0504
1955 ;*****
1956 ;* THIS IS AN ERROR REPORTING SUBROUTINE WHICH PRINTS ADDITIONAL ERROR
1957 ;* MESSAGES WHEN ILLEGAL INTERRUPTS ARE RECEIVED.
1958 ;*
1959 ;* INPUTS: R1 ADDRESS OF THE MESSAGE TO PRINT.
1960 ;* R2 NUMBER OF ILLEGAL INTEPRUPTS RECEIVED.
1961 ;*
1962 ;* OUTPUTS: MESSAGESS ARE PRINTED AT THE OPERATOR CONSOLE.
1963 ;*
1964 ;* CALLING SEQUENCE: LOAD THE ADDRESS OF THE MESSAGE IN R1.
1965 ;* LOAD THE NUMBER OF ILLEGAL INTS IN R2.
1966 ;* INCLUDE THE LABEL 'ER0504' AS THE MESSAGE POINTER
1967 ;* PARAMETER IN THE DIAG SUPER ERROR REPORT MACRO CALL.
1968 ;*
1969 ;* COMMENTS:
1970 ;*
1971 ;* SUBORDINATE ROUTINES USED: NONE.
1972 ;*****
1973
1974 BGNMSG ER0504
1975
1976 ER0504::
1976 PRINTB #EF0503,R1 ;PRINT THE FIRST LINE OF THE MESSAGE.
1976 016512 010146 MOV R1,-(SP)
1976 016514 012746 004132 MOV #EF0503,-(SP)
1976 016520 012746 000002 MOV #2,-(SP)
1976 016524 010600 MOV SP,R0
1976 016526 104414 TRAP C$PNTB
1976 016530 062706 000006 ADD #6,SP
1977 PRINTX #EF0505,R2 ;PRINT THE NUMBER OF INTS RECEIVED.
1977 016534 010246 MOV R2,-(SP)
1977 016536 012746 004137 MOV #EF0505,-(SP)
1977 016542 012746 000002 MOV #2,(SP)
1977 016546 010600 MOV SP,R0
1977 016550 104415 TRAP C$PNTX
1977 016552 062706 000006 ADD #6,SP
1978
1979 ENDMSG
1979 016556
1979 016556 L10005:
1979 016556 104423 TRAP C$MSG

```

```

1981 .SBTTL GLOBAL ERROR REPORTING ROUTINE - ER1401
1982 ;*****
1983 ;* THIS IS AN ERROR REPORTING SUBROUTINE WHICH PRINTS ADDITIONAL ERROR
1984 ;* INFORMATION IF AN ERROR IS DETECTED IN THE ROM VERSION TEST.
1985 ;* THIS SUBROUTINE ANALYSES THE INPUT PARAMETERS WHICH CONTAIN THE
1986 ;* ROM VERSION NUMBERS FOR PROC_1 AND PROC_2 AND REPORTS THE APPROPRIATE
1987 ;* ERROR MESSAGE TO THE OPERATOR.
1988 ;*
1989 ;* INPUTS: R1 - CONTAINS THE ADDRESS OF THE FIRST MESSAGE TO BE REPORTED.
1990 ;* R3 - CONTAINS THE ROM VERSION NUMBER OF PROC_1.
1991 ;* R4 - CONTAINS THE ROM VERSION NUMBER OF PROC_2.
1992 ;*
1993 ;* OUTPUTS: BASIC AND EXTENDED ERROR MESSAGES ARE REPORTED, AT THE
1994 ;* OPERATORS CONSOLE.
1995 ;*
1996 ;* CALLING SEQUENCE: INCLUDE THE LABEL "ER1401" AS THE MESSAGE POINTER
1997 ;* PARAMETER IN THE DRS ERROR REPORT MACRO CALL.
1998 ;*
1999 ;* COMMENTS:
2000 ;*
2001 ;* SUBORDINATE ROUTINES USED: NONE.
2002 ;*****
2003
2004 016560 BGNMSG ER1401
2005 016560 ER1401::
2006 016560 PRINTB #EF0503,R1 ;REPORT THE ERROR MESSAGE PASSED IN.
016560 010146 MOV R1,-(SP)
016562 012746 004132 MOV #EF0503,-(SP)
016566 012746 000002 MOV #2,-(SP)
016572 010600 MOV SP,R0
016574 104414 TRAP C$PNTB
016576 062706 000006 ADD #6,SP

2007
2008 ;*
2009 ; DETERMINE WHICH ROM VERSION NUMBER(S) ARE MISSING.
2010 ; -
2011
2012 016602 012705 000143 MOV #99,R5 ;GET INVALID ROM NUMBER.
2013 016606 012701 012312 MOV #EM1405,R1 ;SELECT PROC_1 MESSAGE.
2014 016612 012702 012340 MOV #EM1407,R2 ;SELECT THE "NOT FOUND" MESSAGE.
2015 016616 120305 CMPB R3,R5 ;CHECK PROC_1 ROM VERSION NUMBER.
2016 016620 001402 BEQ 2$ ;GO REPORT PROC_1 CODE NOT FOUND.
2017 016622 012702 012352 MOV #EM1408,R2 ;SELECT "FOUND" MESSAGE.
2018 016626 004767 000026 2$: JSR PC,50$ ;GO REPORT MESSAGE.
2019
2020 016632 012701 012325 MOV #EM1406,R1 ;SELECT PROC_2 MESSAGE.
2021 016636 012702 012340 MOV #EM1407,R2 ;SELECT THE "NOT FOUND" MESSAGE.
2022 016642 120405 CMPB R4,R5 ;CHECK PROC_2 ROM VERSION NUMBER.
2023 016644 001402 BEQ 4$ ;GO REPORT PROC_2 CODE NOT FOUND.
2024 016646 012702 012352 MOV #EM1408,R2 ;SELECT "FOUND" MESSAGE.
2025 016652 004767 000002 4$: JSR PC,50$ ;GO REPORT THE MESSAGE.
2026 016656 000413 BR 60$ ;EXIT.
2027
2028 016660 50$: PRINTX #EF1402,R1,R2 ;REPORT THE MESSAGE.
016660 010246 MOV R2,(SP)
016662 010146 MOV R1,-(SP)

```

CVDHABO DHV 11 FUNC TST PART2
GLOBAL ERROR REPORTING ROUTINE

MACRO M1200 10 OCT 83 11:55 PAGE 41 1
ER1401

CE2 79

016664 012746 004314
016670 012746 000003
016674 010600
016676 104415
016700 062706 000010
2029 016704 000207
2030 016706
016706
016706 104423

601: RTS PC
ENDMSG

;RETURN.

MOV DEF1402, (SP)
MOV 03, (SP)
MOV SP, R0
TRAP C8PN7
ADD 010, SP

L10006: TRAP C8M55

```
2032 .SBTTL GLOBAL ERROR REPORTING ROUTINE ER1601
2033 .....
2034 ;* THIS AN ERROR REPORTING SUBROUTINE WHICH PRINTS ADDITIONAL ERROR
2035 ;* INFORMATION IF AN ERROR IS DETECTED IN ONE OF THE DEVICE REGISTER
2036 ;* ACCESS TESTS.
2037 ;* THIS SUBROUTINE REPORTS THE ACTUAL AND EXPECTED FROM THE DEVICE
2038 ;* REGISTER(S) WHICH IS(ARE) IN FAULTY
2039 ;*
2040 ;* INPUTS: R1 - ACTUAL DATA (UNUSED BITS SET TO 0).
2041 ;* R2 - EXPECTED DATA (UNUSED BITS SET TO 0).
2042 ;* R3 - OFFSET (IN BYTES) TO THE REGISTER BEING TESTED.
2043 ;* R5 - LINE NUMBER OF REGISTER BEING TESTED.
2044 ;* RMTBB - LABEL AT BASE OF REGISTER MESSAGE ADDRESS TABLE.
2045 ;*
2046 ;* OUTPUTS: MESSAGES ARE PRINTED AT THE OPERATORS CONSOLE.
2047 ;*
2048 ;* CALLING SEQUENCE: INCLUDE THE LABEL "ER1601" AS THE MESSAGE POINTER
2049 ;* PARAMETER IN THE DRS ERROR REPORT MACRO CALL.
2050 ;*
2051 ;* COMMENTS:
2052 ;*
2053 ;* SUBORDINATE ROUTINES CALLED: NONE
2054 .....
2055 016710 BGNMSG ER1601
2056 016710 ER1601::
2057 016710 016304 002304 MOV RMTBB(R3),R4 ;FETCH ADDRESS OF REGISTER NAME MESSAGE.
2058
2059 016714 PRINTB #EF1604,R4,R5 ;REPORT BASIC MESSAGE (REG NAME AND LINE #).
2060 016714 010546 MOV R5,-(SP)
2061 016716 010446 MOV R4,-(SP)
2062 016720 012746 004501 MOV #EF1604,(SP)
2063 016724 012746 000003 MOV #3,(SP)
2064 016730 010600 MOV SP,R0
2065 016732 104414 TRAP C$PNTB
2066 016734 062706 000010 ADD #10,SP
2067 016740 PRINTX #EF1602,R2 ;PRINT THE EXPECTED DATA.
2068 016740 010246 MOV R2,(SP)
2069 016742 012746 004375 MOV #EF1602,(SP)
2070 016746 012746 000002 MOV #2,-(SP)
2071 016752 010600 MOV SP,R0
2072 016754 104415 TRAP C$PNTX
2073 016756 062706 000006 ADD #6,SP
2074 016762 PRINTX #EF1603,R1 ;PRINT THE ACTUAL DATA.
2075 016762 010146 MOV R1,(SP)
2076 016764 012746 004437 MOV #EF1603,(SP)
2077 016770 012746 000002 MOV #2,(SP)
2078 016774 010600 MOV SP,R0
2079 016776 104415 TRAP C$PNTX
2080 017000 062706 000006 ADD #6,SP
2081 017004 ENDMSG
2082 017004
2083 017004 104423 L10007: TRAP C$MSG
```

```

2064 .SBTTL GLOBAL ERROR REPORTING ROUTINE - ER1603
2065 ;*****
2066 ;* THIS ERROR REPORTING ROUTINE IS USED TO PRINT OUT A BASIC ERROR
2067 ;* MESSAGE, ALONG WITH A MESSAGE INFORMING THE OPERATOR WHICH TEST IS
2068 ;* ABOUT TO BE ABORTED.
2069 ;*
2070 ;* INPUTS: R1 - CONTAINS THE ADDRESS OF THE MESSAGE TO BE PRINTED.
2071 ;* ERRMSG - CONTAINS THE ADDRESS OF THE MESSAGE THAT INDICATES
2072 ;* THE TEST THAT IS BEING PERFORMED, EG DMA, BREAK ETC.
2073 ;*
2074 ;* OUTPUTS: MESSAGES ARE PRINTED AT THE OPERATORS CONSOLE.
2075 ;* "TESTNAME TEST ABORTED"
2076 ;*
2077 ;* CALLING SEQUENCE: INCLUDE THE LABEL "ER1603" AS THE MESSAGE POINTER
2078 ;* PARAMETER IN THE DRS ERROR REPORT MACRO CALL.
2079 ;*
2080 ;* COMMENTS:
2081 ;*
2082 ;*
2083 ;* SUBORDINATE ROUTINES CALLED: NONE.
2084 ;*****
2085 017006 BGNMSG ER1603
2086 017006 ER1603::
2087 017006 004567 164764 SAVE JSR ;SAVE THE CONTENTS OF THE GPRS.
2088 017012 ;PRINT BASIC MESSAGE ON OPERATORS CONSOLE.
2089 017012 010146 MOV R1,-(SP)
2090 017014 012746 004132 MOV #EF0503,-(SP)
2091 017020 012746 000002 MOV #2,-(SP)
2092 017024 010600 MOV SP,R0
2093 017026 104414 TRAP C$PNTB
2094 017030 062706 000006 ADD #6,SP
2095 017034 016702 164732 MOV ERRMSG,R2 ;GET THE 'TEST MESSAGE'.
2096 017040 010246 PRINTB #EF1601,R2 ;PRINT 'TEST ABORTED' MESSAGE.
2097 017042 012746 004351 MOV R2,-(SP)
2098 017046 012746 000002 MOV #EF1601,(SP)
2099 017052 010600 MOV SP,R0
2100 017054 104414 TRAP C$PNTB
2101 017056 062706 000006 ADD #6,SP
2102 017062 004736 PASS ;RESTORE THE CONTENTS OF THE GPRS.
2103 017064 JSR PC,8(SP). ;RETURN TO PREG05 SUBRT.
2104 017064 ENDMMSG
2105 017064 104423 L10010: TRAP C$MSG

```

```
2096 .SBTTL GLOBAL ERROR REPORTING ROUTINE ER3001
2097 .....
2098 ;* THIS IS AN ERROR REPORTING SUBROUTINE WHICH IS INTENDED FOR USE IN THE
2099 ;* INTERRUPT BR LEVEL TEST. IT REPORTS ADDITIONAL INFORMATION WHEN AN
2100 ;* INTERRUPT HAS OCCURRED AT THE WRONG BR LEVEL.
2101 ;*
2102 ;* INPUTS: R1 ADDRESS OF MESSAGE TO PRINT FIRST.
2103 ;* R4 BR LEVEL AT WHICH THE INT REQUEST OCCURRED.
2104 ;* R5 - EXPECTED OR CORRECT BR LEVEL FOR THE DUT.
2105 ;*
2106 ;* OUTPUTS: A MESSAGE IS PRINTED AT THE OPERATOR CONSOLE.
2107 ;*
2108 ;* CALLING SEQUENCE: INCLUDE THE LABEL 'ER3001' AS THE MESSAGE POINTER
2109 ;* PARAMETER IN THE DIAG SUPER ERROR REPORT MACRO CALL.
2110 ;*
2111 ;* COMMENTS: THE MESSAGE IS PRINTED AS BASIC AND EXTENDED ERROR INFORMATION.
2112 ;*
2113 ;* SUBORDINATE ROUTINES USED: NONE.
2114 .....
2115
2116 BGNMSG ER3001
2117
2118 ER3001::
2119 PRINTB #EF0503,R1 ;PRINT THE FIRST LINE OF THE MESSAGE.
2120
2121 MOV R1,-(SP)
2122 MOV #FF0503,(SP)
2123 MOV #2,-(SP)
2124 MOV SP,R0
2125 TRAP C:PNTB
2126 ADD #6,SP
2127
2128 PRINTX #EF3001,R5 ;REPORT EXPECTED BR LEVEL.
2129
2130 MOV R5,(SP)
2131 MOV #EF3001,(SP)
2132 MOV #2,-(SP)
2133 MOV SP,R0
2134 TRAP C:PNTX
2135 ADD #6,SP
2136
2137 PRINTX #EF3002,R4 ;REPORT ACTUAL BR LEVEL.
2138
2139 MOV R4,-(SP)
2140 MOV #EF3002,(SP)
2141 MOV #2,-(SP)
2142 MOV SP,R0
2143 TRAP C:PNTX
2144 ADD #6,SP
2145
2146 ENDMSG
2147
2148 L10011:
2149 TRAP C:MSG
2150
2151
2152
2153
2154
2155
2156
2157
2158
2159
2160
2161
2162
2163
2164
2165
2166
2167
2168
2169
2170
2171
2172
2173
2174
2175
2176
2177
2178
2179
2180
2181
2182
2183
2184
2185
2186
2187
2188
2189
2190
2191
2192
2193
2194
2195
2196
2197
2198
2199
2200
```

```

2124 .SBTTL GLOBAL ERROR REPORTING ROUTINE ER9004
2125 ;*****
2126 ;* THIS IS AN ERROR REPORTING SUBROUTINE WHICH REPORTS ERROR SUMMARIES
2127 ;* FOR LINES WHICH HAVE EXCEEDED THE SPECIFIED MAXIMUM NUMBER OF
2128 ;* INDIVIDUAL RECEPTION ERRORS.
2129 ;*
2130 ;* INPUTS: R1 - ADDRESS OF MESSAGE TO PRINT FIRST.
2131 ;* ERCNTB - LABEL AT BASE OF LINE ERROR COUNTERS TABLE.
2132 ;* ERSMBF "REPORT ERROR SUMMARY FOR LINE" FLAGS.
2133 ;*
2134 ;* OUTPUTS: A MESSAGE IS PRINTED AT THE OPERATOR CONSOLE.
2135 ;*
2136 ;* CALLING SEQUENCE: INCLUDE THE LABEL "ER9004" AS THE MESSAGE POINTER
2137 ;* PARAMETER IN THE DIAG SUPER ERROR REPORT MACRO CALL.
2138 ;*
2139 ;* COMMENTS: THE MESSAGE IS PRINTED AS BASIC AND EXTENDED ERROR INFORMATION.
2140 ;* THE CONTENTS OF GPR'S R2, R3, R4, AND R5 ARE DESTROYED.
2141 ;*
2142 ;* SUBORDINATE ROUTINES USED: NONE.
2143 ;*****
2144
2145 017156 BGNMSG ER9004
2146 017156 ER9004::
2147 017156 PRINTB 0EF0503,0EM9014 ;REPORT THE SECONDARY ERROR MESSAGE.
2148 017156 012746 015313 MOV 0EM9014,-(SP)
2149 017162 012746 004132 MOV 0EF0503,(SP)
2150 017166 012746 000002 MOV 02,(SP)
2151 017172 010600 MOV SP,R0
2152 017174 104414 TRAP C$PNTB
2153 017176 062706 000006 ADD 06,SP
2154 017202 005002 CLR R2 ;CLEAR THE LINE COUNTER.
2155 017204 016703 163252 MOV ERSMBF,R3 ;GET THE ERROR SUMMARY FLAGS.
2156 017210 005004 CLR R4 ;CLEAR "LINE COUNTER TIMES 2" OFFSET.
2157 017212 000241 2$: CLC ;CLEAR THE CARRY FOR THE FOLLOWING ROTATE.
2158 017214 006003 ROR R3 ;SHIFT ANOTHER ERROR SUMMARY FLAG INTO CARRY.
2159 017216 103013 BCC 4$ ;SKIP PRINTING MESSAGE IF FLAG FOR LINE CLEAR.
2160 017220 PRINTX 0EF9010,R2,ERCNTB(R4)
2161 017220 016446 002464 MOV ERCNTB(R4),(SP)
2162 017224 010246 MOV R2,-(SP)
2163 017226 012746 005223 MOV 0EF9010,-(SP)
2164 017232 012746 000003 MOV 03,-(SP)
2165 017236 010600 MOV SP,R0
2166 017240 104415 TRAP C$PNTX
2167 017242 062706 000010 ADD 010,SP
2168 017246 012405 4$: MOV (R4)+,R5 ;INCREMENT THE LINE OFFSET BY 2.
2169 017250 005202 INC R2 ;INCREMENT THE LINE COUNTER.
2170 017252 005703 TST R3 ;CHECK THE ERROR SUMMARY FLAGS.
2171 017254 001356 BNE 2$ ;IF MORE FLAGS SET, LOOP TO DO OTHER LINES.
2172 017256 ENDMSG
2173 017256 L10012:
2174 017256 104423 TRAP C$MSG

```

```

2162 .SBTTL GLOBAL ERROR REPORTING ROUTINE ER9007
2163 ;*****
2164 ;* THIS IS AN ERROR REPORTING SUBROUTINE WHICH IS USED TO REPORT THAT
2165 ;* SOMETHING OTHER THAN A SELFTEST CODE WAS FOUND IN A SELFTEST CODE
2166 ;* FIFO SLOT DURING THE REMOVAL OF THE SELFTEST CODES FROM THE FIFO.
2167 ;* THIS ROUTINE IS USED BY THE RSTRPT ROUTINE.
2168 ;*
2169 ;* INPUTS: R1 ADDRESS OF ERROR MESSAGE QUALIFIER STRING.
2170 ;* R2 - INCORRECT CODE AS READ FROM THE SELFTEST CODE FIFO SLOT.
2171 ;* R3 - LINE NUMBER ASSOCIATED WITH THE SELFTEST FIFO SLOT.
2172 ;*
2173 ;* OUTPUTS: A MESSAGE IS PRINTED AT THE OPERATOR CONSOLE.
2174 ;*
2175 ;* CALLING SEQUENCE: INCLUDE THE LABEL "ER9007" AS THE MESSAGE POINTER
2176 ;* PARAMETER IN THE DIAG SUPER ERROR REPORT MACRO CALL.
2177 ;*
2178 ;* COMMENTS: THE MESSAGE IS PRINTED AS BASIC AND EXTENDED ERROR INFORMATION.
2179 ;*
2180 ;* SUBORDINATE ROUTINES USED: NONE.
2181 ;*****
2182
2183 BGNMSG ER9007
2184
2185 017260 042703 177760 BIC #177760,R3 ;REMOVE ALL BUT LINE # BITS FROM LINE # WORD.
2186 017264 010346 PRINTB #EF9018,R1,R3 ;REPORT SECONDARY ERROR MESSAGE.
2187 017264 010146 MOV R3,(SP)
2188 017270 012746 005473 MOV R1,(SP)
2189 017274 012746 000003 MOV #EF9018,-(SP)
2190 017300 010600 MOV #3,-(SP)
2191 017302 104414 MOV SP,R0
2192 017304 062706 000010 TRAP C$PNTB
2193 017310 PRINTX #EF9017,R1,R2 ;REPORT THE ACTUAL INCORRECT CODE.
2194 017310 010246 MOV R2,-(SP)
2195 017312 010146 MOV R1,-(SP)
2196 017314 012746 005417 MOV #EF9017,-(SP)
2197 017320 012746 000003 MOV #3,-(SP)
2198 017324 010600 MOV SP,R0
2199 017326 104415 TRAP C$PNTX
2200 017330 062706 000010 ADD #10,SP
2201
2202 ENDMSG
2203
2204 017334 104423 L10013: TRAP C$MSG

```

```

2191 .SBTTL GLOBAL ERROR REPORTING ROUTINE ER9008
2192 ;*****
2193 ;* THIS IS AN ERROR REPORTING SUBROUTINE WHICH IS USED TO REPORT THAT
2194 ;* AN UNEXPECTED CODE OR CHARACTER HAS BEEN FOUND IN THE DUT RECEIVE
2195 ;* CHARACTER FIFO.
2196 ;*
2197 ;* INPUTS: R1 - ADDRESS OF PARTIAL ERROR MESSAGE STRING.
2198 ;* R2 INCORRECT CODE AS READ FROM THE SELFTEST CODE FIFO SLOT.
2199 ;*
2200 ;* OUTPUTS: A MESSAGE IS PRINTED AT THE OPERATOR CONSOLE.
2201 ;*
2202 ;* CALLING SEQUENCE: INCLUDE THE LABEL "ER9008" AS THE MESSAGE POINTER
2203 ;* PARAMETER IN THE DIAG SUPER ERROR REPORT MACRO CALL.
2204 ;*
2205 ;* COMMENTS: THE MESSAGE IS PRINTED AS BASIC AND EXTENDED ERROR INFORMATION.
2206 ;*
2207 ;* SUBORDINATE ROUTINES USED: NONE.
2208 ;*****
2209
2210 017336 BGNMSG ER9008
2211 017336
2212
2213 ;*
2214 ;* EXTRACT THE LINE NUMBER FROM THE INCORRECT CODE OR CHARACTER WHICH WAS READ
2215 ;* FROM THE SELFTEST CODE FIFO SLOT.
2216 017336 010203
2217 017340 000303
2218 017342 042703 177760
2219 017346
2220 017346 010346
2221 017350 010146
2222 017352 012746 005322
2223 017356 012746 000003
2224 017362 010600
2225 017364 104414
2226 017366 062706 000010
2227 017372
2228 017372 010246
2229 017374 010146
2230 017376 012746 005417
2231 017402 012746 000003
2232 017406 010600
2233 017410 104415
2234 017412 062706 000010
2235
2236 017416
2237 017416
2238 017416 104423
2239
2240
2241
2242
2243
2244
2245
2246
2247
2248
2249
2250
2251
2252
2253
2254
2255
2256
2257
2258
2259
2260
2261
2262
2263
2264
2265
2266
2267
2268
2269
2270
2271
2272
2273
2274
2275
2276
2277
2278
2279
2280
2281
2282
2283
2284
2285
2286
2287
2288
2289
2290
2291
2292
2293
2294
2295
2296
2297
2298
2299
2300
2301
2302
2303
2304
2305
2306
2307
2308
2309
2310
2311
2312
2313
2314
2315
2316
2317
2318
2319
2320
2321
2322
2323
2324
2325
2326
2327
2328
2329
2330
2331
2332
2333
2334
2335
2336
2337
2338
2339
2340
2341
2342
2343
2344
2345
2346
2347
2348
2349
2350
2351
2352
2353
2354
2355
2356
2357
2358
2359
2360
2361
2362
2363
2364
2365
2366
2367
2368
2369
2370
2371
2372
2373
2374
2375
2376
2377
2378
2379
2380
2381
2382
2383
2384
2385
2386
2387
2388
2389
2390
2391
2392
2393
2394
2395
2396
2397
2398
2399
2400
2401
2402
2403
2404
2405
2406
2407
2408
2409
2410
2411
2412
2413
2414
2415
2416
2417
2418
2419
2420
2421
2422
2423
2424
2425
2426
2427
2428
2429
2430
2431
2432
2433
2434
2435
2436
2437
2438
2439
2440
2441
2442
2443
2444
2445
2446
2447
2448
2449
2450
2451
2452
2453
2454
2455
2456
2457
2458
2459
2460
2461
2462
2463
2464
2465
2466
2467
2468
2469
2470
2471
2472
2473
2474
2475
2476
2477
2478
2479
2480
2481
2482
2483
2484
2485
2486
2487
2488
2489
2490
2491
2492
2493
2494
2495
2496
2497
2498
2499
2500
2501
2502
2503
2504
2505
2506
2507
2508
2509
2510
2511
2512
2513
2514
2515
2516
2517
2518
2519
2520
2521
2522
2523
2524
2525
2526
2527
2528
2529
2530
2531
2532
2533
2534
2535
2536
2537
2538
2539
2540
2541
2542
2543
2544
2545
2546
2547
2548
2549
2550
2551
2552
2553
2554
2555
2556
2557
2558
2559
2560
2561
2562
2563
2564
2565
2566
2567
2568
2569
2570
2571
2572
2573
2574
2575
2576
2577
2578
2579
2580
2581
2582
2583
2584
2585
2586
2587
2588
2589
2590
2591
2592
2593
2594
2595
2596
2597
2598
2599
2600
2601
2602
2603
2604
2605
2606
2607
2608
2609
2610
2611
2612
2613
2614
2615
2616
2617
2618
2619
2620
2621
2622
2623
2624
2625
2626
2627
2628
2629
2630
2631
2632
2633
2634
2635
2636
2637
2638
2639
2640
2641
2642
2643
2644
2645
2646
2647
2648
2649
2650
2651
2652
2653
2654
2655
2656
2657
2658
2659
2660
2661
2662
2663
2664
2665
2666
2667
2668
2669
2670
2671
2672
2673
2674
2675
2676
2677
2678
2679
2680
2681
2682
2683
2684
2685
2686
2687
2688
2689
2690
2691
2692
2693
2694
2695
2696
2697
2698
2699
2700
2701
2702
2703
2704
2705
2706
2707
2708
2709
2710
2711
2712
2713
2714
2715
2716
2717
2718
2719
2720
2721
2722
2723
2724
2725
2726
2727
2728
2729
2730
2731
2732
2733
2734
2735
2736
2737
2738
2739
2740
2741
2742
2743
2744
2745
2746
2747
2748
2749
2750
2751
2752
2753
2754
2755
2756
2757
2758
2759
2760
2761
2762
2763
2764
2765
2766
2767
2768
2769
2770
2771
2772
2773
2774
2775
2776
2777
2778
2779
2780
2781
2782
2783
2784
2785
2786
2787
2788
2789
2790
2791
2792
2793
2794
2795
2796
2797
2798
2799
2800
2801
2802
2803
2804
2805
2806
2807
2808
2809
2810
2811
2812
2813
2814
2815
2816
2817
2818
2819
2820
2821
2822
2823
2824
2825
2826
2827
2828
2829
2830
2831
2832
2833
2834
2835
2836
2837
2838
2839
2840
2841
2842
2843
2844
2845
2846
2847
2848
2849
2850
2851
2852
2853
2854
2855
2856
2857
2858
2859
2860
2861
2862
2863
2864
2865
2866
2867
2868
2869
2870
2871
2872
2873
2874
2875
2876
2877
2878
2879
2880
2881
2882
2883
2884
2885
2886
2887
2888
2889
2890
2891
2892
2893
2894
2895
2896
2897
2898
2899
2900
2901
2902
2903
2904
2905
2906
2907
2908
2909
2910
2911
2912
2913
2914
2915
2916
2917
2918
2919
2920
2921
2922
2923
2924
2925
2926
2927
2928
2929
2930
2931
2932
2933
2934
2935
2936
2937
2938
2939
2940
2941
2942
2943
2944
2945
2946
2947
2948
2949
2950
2951
2952
2953
2954
2955
2956
2957
2958
2959
2960
2961
2962
2963
2964
2965
2966
2967
2968
2969
2970
2971
2972
2973
2974
2975
2976
2977
2978
2979
2980
2981
2982
2983
2984
2985
2986
2987
2988
2989
2990
2991
2992
2993
2994
2995
2996
2997
2998
2999
3000
3001
3002
3003
3004
3005
3006
3007
3008
3009
3010
3011
3012
3013
3014
3015
3016
3017
3018
3019
3020
3021
3022
3023
3024
3025
3026
3027
3028
3029
3030
3031
3032
3033
3034
3035
3036
3037
3038
3039
3040
3041
3042
3043
3044
3045
3046
3047
3048
3049
3050
3051
3052
3053
3054
3055
3056
3057
3058
3059
3060
3061
3062
3063
3064
3065
3066
3067
3068
3069
3070
3071
3072
3073
3074
3075
3076
3077
3078
3079
3080
3081
3082
3083
3084
3085
3086
3087
3088
3089
3090
3091
3092
3093
3094
3095
3096
3097
3098
3099
3100
3101
3102
3103
3104
3105
3106
3107
3108
3109
3110
3111
3112
3113
3114
3115
3116
3117
3118
3119
3120
3121
3122
3123
3124
3125
3126
3127
3128
3129
3130
3131
3132
3133
3134
3135
3136
3137
3138
3139
3140
3141
3142
3143
3144
3145
3146
3147
3148
3149
3150
3151
3152
3153
3154
3155
3156
3157
3158
3159
3160
3161
3162
3163
3164
3165
3166
3167
3168
3169
3170
3171
3172
3173
3174
3175
3176
3177
3178
3179
3180
3181
3182
3183
3184
3185
3186
3187
3188
3189
3190
3191
3192
3193
3194
3195
3196
3197
3198
3199
3200
3201
3202
3203
3204
3205
3206
3207
3208
3209
3210
3211
3212
3213
3214
3215
3216
3217
3218
3219
3220
3221
3222
3223
3224
3225
3226
3227
3228
3229
3230
3231
3232
3233
3234
3235
3236
3237
3238
3239
3240
3241
3242
3243
3244
3245
3246
3247
3248
3249
3250
3251
3252
3253
3254
3255
3256
3257
3258
3259
3260
3261
3262
3263
3264
3265
3266
3267
3268
3269
3270
3271
3272
3273
3274
3275
3276
3277
3278
3279
3280
3281
3282
3283
3284
3285
3286
3287
3288
3289
3290
3291
3292
3293
3294
3295
3296
3297
3298
3299
3300
3301
3302
3303
3304
3305
3306
3307
3308
3309
3310
3311
3312
3313
3314
3315
3316
3317
3318
3319
3320
3321
3322
3323
3324
3325
3326
3327
3328
3329
3330
3331
3332
3333
3334
3335
3336
3337
3338
3339
3340
3341
3342
3343
3344
3345
3346
3347
3348
3349
3350
3351
3352
3353
3354
3355
3356
3357
3358
3359
3360
3361
3362
3363
3364
3365
3366
3367
3368
3369
3370
3371
3372
3373
3374
3375
3376
3377
3378
3379
3380
3381
3382
3383
3384
3385
3386
3387
3388
3389
3390
3391
3392
3393
3394
3395
3396
3397
3398
3399
3400
3401
3402
3403
3404
3405
3406
3407
3408
3409
3410
3411
3412
3413
3414
3415
3416
3417
3418
3419
3420
3421
3422
3423
3424
3425
3426
3427
3428
3429
3430
3431
3432
3433
3434
3435
3436
3437
3438
3439
3440
3441
3442
3443
3444
3445
3446
3447
3448
3449
3450
3451
3452
3453
3454
3455
3456
3457
3458
3459
3460
3461
3462
3463
3464
3465
3466
3467
3468
3469
3470
3471
3472
3473
3474
3475
3476
3477
3478
3479
3480
3481
3482
3483
3484
3485
3486
3487
3488
3489
3490
3491
3492
3493
3494
3495
3496
3497
3498
3499
3500
3501
3502
3503
3504
3505
3506
3507
3508
3509
3510
3511
3512
3513
3514
3515
3516
3517
3518
3519
3520
3521
3522
3523
3524
3525
3526
3527
3528
3529
3530
3531
3532
3533
3534
3535
3536
3537
3538
3539
3540
3541
3542
3543
3544
3545
3546
3547
3548
3549
3550
3551
3552
3553
3554
3555
3556
3557
3558
3559
3560
3561
3562
3563
3564
3565
3566
3567
3568
3569
3570
3571
3572
3573
3574
3575
3576
3577
3578
3579
3580
3581
3582
3583
3584
3585
3586
3587
3588
3589
3590
3591
3592
3593
3594
3595
3596
3597
3598
3599
3600
3601
3602
3603
3604
3605
3606
3607
3608
3609
3610
3611
3612
3613
3614
3615
3616
3617
3618
3619
3620
3621
3622
3623
3624
3625
3626
3627
3628
3629
3630
3631
3632
3633
3634
3635
3636
3637
3638
3639
3640
3641
3642
3643
3644
3645
3646
3647
3648
3649
3650
3651
3652
3653
3654
3655
3656
3657
3658
3659
3660
3661
3662
3663
3664
3665
3666
3667
3668
3669
3670
3671
3672
3673
3674
3675
3676
3677
3678
3679
3680
3681
3682
3683
3684
3685
3686
3687
3688
3689
3690
3691
3692
3693
3694
3695
3696
3697
3698
3699
3700
3701
3702
3703
3704
3705
3706
3707
3708
3709
3710
3711
3712
3713
3714
3715
3716
3717
3718
3719
3720
3721
3722
3723
3724
3725
3726
3727
3728
3729
3730
3731
3732
3733
3734
3735
3736
3737
3738
3739
3740
3741
3742
3743
3744
3745
3746
3747
3748
3749
3750
3751
3752
3753
3754
3755
3756
3757
3758
3759
3760
3761
3762
3763
3764
3765
3766
3767
3768
3769
3770
3771
3772
3773
3774
3775
3776
3777
3778
3779
3780
3781
3782
3783
3784
3785
3786
3787
3788
3789
3790
3791
3792
3793
3794
3795
3796
3797
3798
3799
3800
3801
3802
3803
3804
3805
3806
3807
3808
3809
3810
3811
3812
3813
3814
3815
3816
3817
3818
3819
3820
3821
3822
3823
3824
3825
3826
3827
3828
3829
3830
3831
3832
3833
3834
3835
3836
3837
3838
3839
3840
3841
3842
3843
3844
3845
3846
3847
3848
3849
3850
3851
3852
3853
3854
3855
3856
3857
3858
3859
3860
3861
3862
3863
3864
3865
3866
3867
3868
3869
3870
3871
3872
3873
3874
3875
3876
3877
3878
3879
3880
3881
3882
3883
3884
3885
3886
3887
3888
3889
3890
3891
3892
3893
3894
3895
3896
3897
3898
3899
3900
3901
3902
3903
3904
3905
3906
3907
3908
3909
3910
3911
3912
3913
3914
3915
3916
3917
3918
3919
3920
3921
3922
3923
3924
3925
3926
3927
3928
3929
3930
3931
3932
3933
3934
3935
3936
3937
3938
3939
3940
3941
3942
3943
3944
3945
3946
3947
3948
3949
3950
3951
3952
3953
3954
3955
3956
3957
3958
3959
3960
3961
3962
3963
3964
3965
3966
3967
3968
3969
3970
3971
3972
3973
3974
3975
3976
3977
3978
3979
3980
3981
3982
3983
3984
3985
3986
3987
3988
3989
3990
3991
3992
3993
3994
3995
3996
3997
3998
3999
4000
4001
4002
4003
4004
4005
4006
4007
4008
4009
4010
4011
4012
4013
4014
4015
4016
4017
4018
4019
4020
4021
4022
4023
4024
4025
4026
4027
4028
4029
4030
4031
4032
4033
4034
4035
4036
4037
4038
4039
4040
4041
4042
4043
4044
4045
4046
4047
4048
4049
4050
4051
4052
4053
4054
4055
4056
4057
4058
4059
4060
4061
4062
4063
4064
4065
4066
4067
4068
4069
4070
4071
4072
4073
4074
4075
4076
4077
4078
4079
4080
4081
4082
4083
4084
4085
4086
4087
4088
4089
4090
4091
4092
4093
4094
4095
4096
4097
4098
4099
4100
4101
4102
4103
4104
4105
4106
4107
4108
4109
4110
4111
4112
4113
4114
4115
4116
4117
4118
4119
4120
4121
4122
4123
4124
4125
4126
4127
4128
4129
4130
4131
4132
4133
4134
4135
4136
4137
4138
4139
4140
4141
4142
4143
4144
4145
4146
4147
4148
4149
4150
4151
4152
4153
4154
4155
4156
4157
4158
4159
4160
4161
4162
4163
4164
4165
4166
4167
4168
4169
4170
4171
4172
4173
4174
4175
4176
4177
4178
4179
4180
4181
4182
4183
4184
4185
4186
4187
4188
4189
4190
4191
4192
4193
4194
4195
4196
4197
4198
4199
4200
4201
4202
4203
4204
4205
4206
4207
4208
4209
4210
4211
4212
4213
4214
4215
4216
4217
4218
4219
4220
4221
4222
4223
4224
4225
4226
4227
4228
4229
4230
4231
4232
4233
4234
4235
4236
4237
4238
4239
4240
4241
4242
4243
4244
4245
4246
4247
4248
4249
4250
4251
4252
4253
4254
4255
4256
4257
4258
4259
4260
4261
4262
4263
4264
4265
4266
4267
4268
4269
4270
4271
4272
4273
4274
4275
4276
4277
4278
4279
4280
4281
4282
4283
4284
4285
4286
4287
4288
4289
4290
4291
4292
4293
4294
4295
4296
4297
4298
4299
4300
4301
4302
4303
4304
4305
4306
4307
4308
4309
4310
4311
4312
4313
4314
4315
4316
4317
4318
4319
4320
4321
4322
4323
4324
4325
4326
4327
4328
4329
4330
4331
4332
4333
4334
4335
4336
4337
4338
4339
4340
4341
4342
4343
4344
4345
4346
4347
4348
4349
4350
4351
4352
4353
4354
4355
4356
4357
4358
4359
4360
4361
4362
4363
4364
4365
4366
4367
4368
4369
4370
4371
4372
4373
4374
4375
4376
4377
4378
4379
4380
4381
4382
4383
4384
4385
4386
4387
4388
4389
4390
4391
4392
4393
4394
4395
4396
4397
4398
4399
4400
4401
4402
4403
4404
4405
4406
4407
4408
4409
4410
4411
4412
4413
4414
4415
4416
4417
4418
4419
4420
4421
4422
4423
4424
4425
4426
4427
4428
4429
4430
4431
4432
4433
4434
4435
4436
4437
4438
4439
4440
4441
4442
4443
4444
4445
4446
4447
4448
4449
4450
4451
4452
4453
4454
4455
4456
4457
4458
4459
4460
4461
4462
4463
4464
4465
4466
4467
4468
4469
4470
4471
4472
4473
4474
4475
4476
4477
4478
4479
4480
4481
4482
4483
4484
4485
4486
4487
4488
4489
4490
4491
4492
4493
4494
4495
4496
4497
4498
4499
4500
4501
4502
4503
4504
4505
4506
4507
4508
4509
4510
4511
4512
4
```

```

2224 .SBTTL GLOBAL ERROR REPORTING ROUTINE ER9101
2225 :*****
2226 :* THIS IS A GENERAL ERROR REPORTING SUBROUTINE WHICH REPORTS A MESSAGE
2227 :* WHICH TAKES A SINGLE, 2 DIGIT DECIMAL ARGUMENT AFTER THE END OF AN
2228 :* ASCII MESSAGE.
2229 :*
2230 :* INPUTS: R1 VALUE TO BE PRINTED AFTER MSG AS 2 DECIMAL DIGITS.
2231 :* R2 ADDRESS OF MESSAGE TO PRINT FIRST.
2232 :*
2233 :* OUTPUTS: A MESSAGES IS PRINTED AT THE OPERATOR CONSOLE.
2234 :*
2235 :* CALLING SEQUENCE: INCLUDE THE LABEL "ER9101" AS THE MESSAGE POINTER
2236 :* PARAMETER IN THE DIAG SUPER ERROR REPORT MACRO CALL.
2237 :*
2238 :* COMMENTS: THE MESSAGE IS PRINTED AS BASIC ERROR INFORMATION.
2239 :*
2240 :* SUBORDINATE ROUTINES USED: NONE.
2241 :*****
2242
2243 017420 BGNMSG ER9101
2244 017420 ER9101::
2245 017420 PRINTB 0EF9006,R2,R1 ;REPORT THE STRING FOLLOWED BY THE NUMBER.
2246 017420 010146 MOV R1,-(SP)
2247 017422 010246 MOV R2,-(SP)
2248 017424 012746 005177 MOV 0EF9006,-(SP)
2249 017430 012746 000003 MOV 03,-(SP)
2250 017434 010600 MOV SP,R0
2251 017436 104414 TRAP C$PNTB
2252 017440 062706 000010 ADD 010,SP
2253
2254 017444 ENDMSG
2255 017444 L10015:
2256 017444 104423 TRAP C$MSG

```

```
2249 .SBTTL GLOBAL ERROR REPORTING ROUTINE ER9301
2250 :*****
2251 :* THIS IS AN ERROR REPORTING SUBROUTINE WHICH PRINTS ANY BMP CODES
2252 :* THAT ARE FOUND IN THE BMP CODE QUEUE, TOGETHER WITH THE THE NUMBER OF
2253 :* THE TEST THAT WAS EXECUTING AT THE TIME THE BMP CODE WAS LOGGED.
2254 :*
2255 :* INPUTS: R1 - THE ADDRESS OF THE FIRST MESSAGE TO BE REPORTED.
2256 :* R2 - THE ADDRESS OF THE NEXT EMPTY CELL IN THE QUEUE.
2257 :*
2258 :* OUTPUTS: THE TEST NUMBER FOLLOWED BY THE BMP CODE ARE PRINTED AT THE
2259 :* OPERATOR CONSOLE.
2260 :*
2261 :* CALLING SEQUENCE: INCLUDE THE LABEL "ER9301" AS THE MESSAGE POINTER
2262 :* PARAMETER IN THE DIAG SUPER ERROR REPORT MACRO CALL.
2263 :*
2264 :* COMMENTS: THE MESSAGE IS PRINTED AS BASIC ERROR INFORMATION.
2265 :*
2266 :* SUBORDINATE ROUTINES USED: NONE.
2267 :*****
2268
2269 017446 BGNMSG ER9301
2270 017446 ER9301::
2271 017446 SAVE ;SAVE THE GPRS ON THE STACK.
2272 017446 004567 164324 JSR R5,PREG05 ;CALL REGISTER SAVE SUBRT
2273 017452 PRINTB #EF0503,R1 ;REPORT UNEXPECTED BMP CODES FOUND.
2274 017452 010146 MOV R1,-(SP)
2275 017454 012746 004132 MOV #EF0503,(SP)
2276 017460 012746 000002 MOV #2,-(SP)
2277 017464 010600 MOV SP,R0
2278 017466 104414 TRAP C$PNTB
2279 017470 062706 000006 ADD #6,SP
2280 017474 012703 002526 MOV #BMPQ08,R3 ;GET THE START ADDRESS OF THE BMP CODE QUEUE.
2281 017500 012705 015714 MOV #EM9302,R5 ;GET THE MESSAGE TO BE REPORTED.
2282 2$: MOV (R3)+,R1 ;GET THE NUMBER OF THE TEST THAT WAS EXECUTING.
2283 017504 012301 MOV (R3)+,R4 ;GET BMP CODE THAT WAS REPORTED OFF THE QUEUE.
2284 017506 012304 JSR PC,50$ ;GO REPORT THE BMP CODE.
2285 017510 004767 000056 CMP R3,R2 ;CHECK IF ALL CODES HAVE BEEN REPORTED.
2286 017514 020302 BLO 2$ ;IF IT IS NOT THE LAST BMP CODE THEN LOOP.
2287 017516 103772
2288
2289 :* CHECK IF OVERFLOW HAS OCCURRED.
2290 :* THE CONDITIONS FOR OVERFLOW ARE: THE POINTER CONTAINS THE ADDRESS OF THE
2291 :* LAST CELL IN THE QUEUE, AND A BMP CODE HAS ALREADY BEEN WRITTEN INTO THAT
2292 :* CELL.
2293 :
2294 017520 020227 002722 CMP R2,#BMPQ0E-4 ;CHECK IF THE POINTER IS AT THE LAST LOCATION.
2295 017524 001036 BNE 60$ ;EXIT IF NOT AT THE LAST LOCATION.
2296 017526 005762 000002 TST 2(R2) ;CHECK FOR A BMP CODE IN THE LAST CELL
2297 017532 001433 BEQ 60$ ;EXIT IF NO OVERFLOW HAS OCCURED, CELL EMPTY.
2298 017534 012301 MOV (R3)+,R1 ;GET THE TEST NUMBER OFF THE QUEUE.
2299 017536 011304 MOV (R3),R4 ;GET THE BMP CODE OFF THE QUEUE.
2300 017540 012705 015744 MOV #EM9303,R5 ;SELECT THE MESSAGE TO BE REPORTED.
2301 017544 PRINTX #EF9302 ;REPORT OVERFLOW CONDITION.
2302 017544 012746 005702 MOV #EF9302,(SP)
2303 017550 012746 000001 MOV #1,(SP)
2304 017554 010600 MOV SP,R0
2305 017556 104415 TRAP C$PNTX
```

```

017560 062706 000004
2294 017564 004767 000002      JSR    PC,50$      ;REPORT THE LAST BMP CODE PLACED ON THE QUEUE.
2295 017570 000414              BR      60$      ;EXIT.
2296
2297 017572              50$: PRINTX 0EF9301,R5,R1,R4 ;PRINT THE MESSAGE.
017572 010446              MOV     R4,-(SP)
017574 010146              MOV     R1,(SP)
017576 010546              MOV     R5,(SP)
017600 012746 005624      MOV     0EF9301,(SP)
017604 012746 000004      MOV     64,(SP)
017610 010600              MOV     SP,R0
017612 104415              TRAP    C$PNTX
017614 062706 000012      ADD     12,SP
2298 017620 000207
2299 017622              60$: RTS     PC      ;RETURN.
017622 004736              PASS      ;RESTORE THE GPR CONTENTS.
2300              JSR      PC,2(SP).      ;RETURN TO PREGOS SUBRT.
2301 017624              ENDMSG
017624              L10016:
017624 104423              TRAP    C$MSG

```

```
2303 .SBTTL GLOBAL SUBROUTINES SECTION
2305 ;*****
2306 ;
2307 ; FVTSKL3.P11
2308 ;
2309 ;*****
2311
2312
2313 ;..
2314 ; THE GLOBAL SUBROUTINES SECTION CONTAINS THE SUBROUTINES
2315 ; THAT ARE USED IN MORE THAN ONE TEST.
2316 ;
```

```

2318 .SBTTL GLOBAL SUBROUTINE ALTFLD
2319 ;* *****
2320 ;* - ALTER DEVICE REGISTER FIELDS ROUTINE
2321 ;* THIS SUBROUTINE ALTERS THE SPECIFIED FIELD OF THE SPECIFIED DEVICE
2322 ;* REGISTER FOR THE SPECIFIED LINES. THIS ROUTINE CAN BE USED TO SET
2323 ;* OR CLEAR BITS WITHIN SELECTED FIELDS OF SELECTED REGISTERS.
2324 ;* USE EXAMPLES: SET RX.BAUD.RATE FIELDS ON LINES 3 AND 6.
2325 ;* CLEAR TX.DMA BITS ON ALL LINES.
2326 ;*
2327 ;* INPUTS: R1 ADDRESS OF THE REGISTERS TO ALTER.
2328 ;* R2 BIT FIELDS SET TO DESIRED STATES.
2329 ;* R3 BIT MAP OF LINES FOR WHICH TO ALTER REGISTER.
2330 ;* R4 MASK OF BITS TO ALTER (1 INDICATES CHANGE BIT).
2331 ;* CSRA CONTAINS THE ADDRESS OF THE DEVICE CSR.
2332 ;* IESTAT SAVED STATES OF THE INTERRUPT ENABLE BITS.
2333 ;*
2334 ;* OUTPUTS: DEVICE REGISTERS - SPECIFIED REGISTER FIELDS ALTERED.
2335 ;* CSR IND.ADR.REG FIELD - DESTROYED.
2336 ;*
2337 ;* CALLING SEQUENCE: JSR PC,ALTFLD
2338 ;*
2339 ;* COMMENTS: THIS ROUTINE READS THE SPECIFIED REGISTERS FOR ALL LINES
2340 ;* WITH NUMBERS LOWER THAN THE HIGHEST SPECIFIED LINE.
2341 ;* THIS ROUTINE DOES NOT READ THE CSR.
2342 ;*
2343 ;* SUBROUTINES CALLED: NONE.
2344 ;* *****
2345
2346 017626 ALTFLD:: SAVE ;SAVE CONTENTS OF GPRS R0 THRU R5.
017626 004567 164144 JSR R5,PREG05 ;CALL REGISTER SAVE SUBRT.
2347
2348 ;*
2349 ;* SET UP TO LOOP FOR EACH LINE:
2350 ;* PREPARE THE WORD TO BE ORED INTO THE REGISTER CONTENTS.
2351 ;* SET UP THE WORD TO WRITE INTO THE IND.ADR.REG FIELD OF THE CSR.
2352 ;*
2353 017632 010400 MOV R4,R0 ;CALCULATE THE NEW CONTENTS OF THE
2354 017634 005100 COM R0 ; REGISTER FIELDS WHICH ARE TO BE
2355 017636 040002 BIC R0,R2 ; ALTERED BY THIS ROUTINE.
2356 017640 016705 162464 MOV IESTAT,R5 ;SET UP TO WRITE IND.ADR.REG FIELD TO 0.
2357 ;*
2358 ;* LOOP ONCE FOR EACH LINE, ALTERING THE SPECIFIED FIELD IN THE SPECIFIED
2359 ;* REGISTER IF THE LINE HAS BEEN SELECTED FOR ALTERING.
2360 ;* EXIT THE LOOP IF NO MORE LINES TO ALTER, OR IF WE HAVE ALTERED THE MAX
2361 ;* ALLOWABLE NUMBER OF LINES (AS SPECIFIED BY NUMLNS).
2362 ;*
2363 017644 000241 CLC ;PREPARE FOR ROTATE, "TST R5" DOES THIS BELOW.
2364 017646 006003 ROR R3 ;GET THE LINE SELECT BIT FOR THIS LINE.
2365 017650 103006 BCC 4$ ;SKIP SETUP IF LINE IS NOT SELECTED.
2366 017652 010577 162366 MOV R5,@CSRA ;SET OUT CSR IND.ADR.REG FIELD TO THIS LINE.
2367 017656 011100 MOV (R1),R0 ;GET THE PRESENT CONTENTS OF THE REG TO ALTER.
2368 017660 040400 BIC R4,R0 ;CLEAR THE BIT FIELDS WE ARE TO ALTER.
2369 017662 050200 BIS R2,R0 ;OR IN THE NEW STATES OF THE FIELDS.
2370 017664 010011 MOV R0,(R1) ;WRITE THE NEW REGISTER CONTENTS TO THE REG.
2371 017666 005205 4$ INC R5 ;SET LINE NUMBER TO THE NEXT LINE.
2372 017670 005703 TST R3 ;CHECK FOR UNHANDLED LINES, CLEAR CARRY FLAG.
2373 017672 001365 BNE 2$ ;LOOP IF SELECTED LINE(S) IS NOT HANDLED.

```

N7

CVDHABO DHV 11 FUNC 1ST PART2 MACRO M1200 10 OCT 83 11:35 PAGE 51 1
GLOBAL SUBROUTINE - ALTFLD -

SEQ 91

2374

2375 017674

017674 004736

2376 017676 000207

604:

PASS

RTS

PC

JSR

;RESTORE GPRS.

PC,@(SP);

;RETURN TO CALLING ROUTINE.

;RETURN TO PREG05 SUBRT.

```

2378 .SBTTL GLOBAL SUBROUTINE CALMSL
2379 ;* *****
2380 ;* - CALIBRATE MILLI SECOND LOOP COUNT SUBROUTINE
2381 ;* THIS SUBROUTINE CALIBRATES THE TIMING LOOP WHICH IS USED IN THE MSLOOP
2382 ;* ROUTINE. THIS SUBROUTINE CALCULATES A VALUE FOR THE MSLCNT VARIABLE
2383 ;* WHICH IS THE NUMBER OF SOFTWARE LOOPS WHICH TAKES 1 MS TO EXECUTE IN
2384 ;* THE MSLOOP ROUTINE. THIS ROUTINE CALIBRATES THE COUNT BY USING THE
2385 ;* LINE TIME CLOCK (LTC), SO IF NO LTC IS AVAILABLE THE DEFAULT VALUE FOR
2386 ;* THE DELAY COUNT MUST BE USED.
2387 ;*
2388 ;*
2389 ;* INPUTS: MSLCNT DEFAULT 1 MS DELAY LOOP COUNT VALUE, OR
2390 ;* VALUE FROM PREVIOUS CALIBRATION.
2391 ;* MSTICK NUMBER OF MS PER LTC CLOCK TICK.
2392 ;* TIMER1 - TIMER COUNTER CHANGED BY LTC INTERRUPT SERVICE RTN.
2393 ;* CLKHRZ - NUMBER OF LTC CLICKS PER SECOND (50 OR 60).
2394 ;*
2395 ;* OUTPUTS: CARRY SET IF LTC IS AVAILABLE, AND NEW CALIBRATION PERFORMED.
2396 ;* MSLCNT NEW 1 MS DELAY LOOP COUNT VALUE IF LTC AVAILABLE, OR
2397 ;* UNCHANGED IF NO LTC IS AVAILABLE.
2398 ;*
2399 ;* CALLING SEQUENCE: JSR PC,CALMSL
2400 ;*
2401 ;* COMMENTS:
2402 ;*
2403 ;* SUBORDINATE ROUTINES CALLED: UNSDIV,OOPS.
2404 ;* - *****
2405
2406 017700 CALMSL:: SAVE ;SAVE CONTENTS OF GPRS R0 THRU R5.
2407 017700 004567 164072 JSR R5,PREG05 ;CALL REGISTER SAVE SUBRT.
2408 017704 005067 000210 CLR 62$ ;CLEAR THE 2ND TIME FLAG.
2409 ;*
2410 ;* SYNCHRONIZE WITH THE LTC.
2411 017710 012705 000001 2$: MOV #1,R5 ;SET OUTER LOOP COUNTER TO 1 LOOP.
2412 ;* ;INCREASE THE VALUE LOADED INTO THIS COUNTER IF THE ***
2413 ;* ;FOLLOWING LOOP FAILS ON FUTURE, FASTER PROCESSORS. ***
2414 017714 005000 CLR R0 ;CLEAR THE WAIT FOR CLOCK INT COUNTER.
2415 017716 012767 000001 162440 MOV #1,TIMER1 ;SET UP COUNT OF 1 TO SYNCH WITH LTC.
2416 017724 005767 162434 4$: TST TIMER1 ;CHECK FOR COUNTER HAVING GONE TO ZERO.
2417 017730 001410 BEQ 6$ ;JUMP OUT OF LOOP IF LTC HAS INTERRUPTED.
2418 017732 005200 INC R0 ;COUNT THIS ITERATION OF THE INNER LOOP.
2419 017734 001373 BNE 4$ ;LOOP IF COUNTER HAS NOT TURNED OVER.
2420 017736 005305 DEC R5 ;DECREMENT THE INNER LOOP COUNTER.
2421 017740 003371 BGT 4$ ;LOOP IF OUTER LOOP COUNT NOT UP.
2422 ;*
2423 ;* IF WE GOT NO LTC INTERRUPT, INDICATE THAT THERE IS NO LTC AVAILABLE.
2424 ;* LTC MUST BE FLAKEY, OR NOT REALLY AN LTC AT ALL.
2425 ;*
2426 017742 005067 162414 CLR CLKHRZ ;CLEAR LTC FREQUENCY WORD TO INDICATE NO LTC.
2427 017746 000241 CLC ;INDICATE FAILURE FOR RETURN.
2428 017750 000461 BR 60$ ;BYPASS THE FOLLOWING CALIBRATION PROCEDURES.
2429 ;*
2430 ;* WE ARE NOW SYNCHRONIZED WITH THE LTC.
2431 ;* SET UP FOR THE CALIBRATION LOOP.
2432 ;*
2433 017752 012704 002364 6$: MOV #TIMER1,R4 ;WILL TEST TIMER1 IN THE LOOP BELOW.

```

C2

```
2434 017756 005001          CLR      R1          ;CLEAR THE OUTER LOOP COUNTER.
2435 017760 005002          CLR      R2          ;INDICATE TO CHECK ALL BITS OF TIMER1.
2436 017762 005003          CLR      R3          ;INDICATE TO CHECK FOR TIMER1 CLEAR.
2437 017764 012714 000001    MOV      R1,(R4)    ;LOAD TIMER1 WITH COUNT OF 1.
2438
2439 017770 016705 162402    8$:      MOV      MSLCNT,R5    ;LOAD MS LOOP COUNT.
2440 017774 011400          10$:     MOV      (R4),R0    ;GET THE TIMER1 VALUE.
2441 017776 010067 000120    MOV      R0,64$    ;SAVE WORD (LIKE IN THE REAL LOOP).
2442 020002 040200          BIC      R2,R0    ;LEAVE ALL THE BITS.
2443 020004 020003          CMP      R0,R3    ;COMPARE AGAINST ZERO.
2444 020006 000261          SEC            ;SET CARRY IN CASE OF SUCCESS.
2445 020010 001406          BEQ      12$    ;EXIT LOOP IF TIMER1 HAS CLEARED.
2446 020012 005305          DEC      R5    ;COUNT DOWN THE INSIDE MS LOOP COUNT.
2447 020014 001367          BNE      10$    ;LOOP IF MS NOT UP.
2448 020016 005301          DEC      R1    ;DECREMENT THE MS TIME COUNT.
2449 020020 001363          BNE      8$    ;KEEP LOOPING.
2450 020022 004767 000440    JSR      PC,00PS    ;WE OVERFLOWED, SOMETHING IS WRONG, ABORT.
2451
2452          ;*
2453          ; WE HAVE NOW HAVE LOOP COUNT INFORMATION FOR ONE CLOCK TICK.
2454          ; WE HAVE NEGATIVE OF NUMBER OF OUTER LOOPS IN R1, EACH IS MSLCNT INNER LOOPS.
2455          ; WE HAVE THE PORTION OF THE LAST OUTER LOOP NOT EXECUTED, IN R5.
2456          ; NOW WE CALCULATE THE TOTAL NUMBER OF INNER LOOPS EXECUTED.
2457
2457 020026 005401          12$:     NEG      R1          ;GET NUMBER OF OUTER LOOPS.
2458 020030 016702 162342    MOV      MSLCNT R2    ;GET THE NUMBER OF INNER LOOPS PER OUTER LOOP.
2459 020034 010203          MOV      R2,R3    ;COPY NUMBER OF LOOPS FOR MULTIPLY.
2460 020036 160502          SUB      R5,R2    ;CALC # OF INNER LOOPS DONE IN LAST OUTER LOOP
2461 020040 010204          MOV      R2,R4    ; AND ADD TO ACCUMULATOR LSWORD.
2462 020042 005005          CLR      R5    ;CLEAR ACCUMULATOR MSWORD.
2463 020044 005301          14$:     DEC      R1    ;CHECK R1 FOR 0 CONDITION
2464 020046 100403          BMI      16$    ; SKIP MULTIPLICATION IF ZERO
2465 020050 060304          ADD      R3,R4    ;MULTIPLY NUMBER OF INNER
2466 020052 005505          ADC      R5    ; LOOPS PER OUTER LOOP BY
2467 020054 000773          BR       14$    ;NUMBER OF OUTER LOOPS PERFORMED.
2468
2469          ;*
2470          ; DIVIDE THE TOTAL NUMBER OF INNER LOOPS BY THE NUMBER OF MS PER LTC TICK.
2471
2471 020056 016701 162312    16$:     MOV      MSLCNT,R1    ;# OF MS PER LTC TICK IS DIVISOR.
2472 020062 010403          MOV      R4,R3    ;LSWORD OF LOOP COUNT IS LSWORD OF DIVIDEND.
2473 020064 010502          MOV      R5,R2    ;MSWORD OF LOOP COUNT IS MSWORD OF DIVIDEND.
2474 020066 004767 002756    JSR      PC,UNSDIV    ;DIVIDE NUMBER OF LOOPS BY MS PER LTC TICK.
2475 020072 103402          BCS      18$    ;BYPASS OOPS IF WE'RE OK.
2476 020074 004767 000366    JSR      PC,00PS    ;CLOCK ROUTINES ARE NOT LONG ENOUGH, OR BUG.
2477 020100 010167 162272    18$:     MOV      R1,MSLCNT    ;SET NEW VALUE FOR MS LOOP COUNT.
2478 020104 005167 000010    COM      62$    ;SET THE 2ND ITERATION FLAGS IF 1ST ITERATION.
2479 020110 001277          BNE      2$    ;BRANCH IF ONLY ONE ITERATION DONE.
2480 020112 000261          SEC            ;SET THE SUCCESS FLAG FOR EXIT.
2481
2482 020114          60$:     PASS          ;RESTORE GPRS.
2483 020116 001736          RTS      PC          JSR      PC,@(SP) ;RETURN TO PREG05 SUBRT.
2484          ; CARRY - SUCCESS FLAG. SET IF SUCCESS.
2485 020120 000000          62$:     .WORD    0    ;2ND CALIBRATION ITERATION FLAGS.
2486 020122 000000          64$:     .WORD    0    ;DUMMY WORD FOR STORAGE OF THE READ WORD.
```

```

2488 .SBTTL GLOBAL SUBROUTINE CKTRAP
2489 .....
2490 ;* CHECK TRAP ROUTINE
2491 ;* THIS SUBROUTINE IS USED TO CHECK FOR A BUS TIME OUT TRAP (004 TRAP)
2492 ;* WHICH IS CAUSED BY AN ACCESS TO A NON EXISTENT MEMORY OR I/ LOCATION.
2493 ;* IF THE TRAP DOES NOT OCCUR, THIS ROUTINE RETURNS A SUCCESS INDICATION.
2494 ;*
2495 ;* INPUTS: R0 SOURCE ADDRESS FOR MOVE.
2496 ;* R1 - DESTINATION ADDRESS FOR MOVE.
2497 ;* (R0) SOURCE FOR THE MOVE.
2498 ;*
2499 ;* OUTPUTS: (R1) WRITTEN TO THE CONTENTS OF (R0).
2500 ;* CARRY FLAG - SET ON RETURN IF NO 004 TRAP DETECTED.
2501 ;* TP4FLG NONZERO IF TRAP OCCURRED, CLEARED OTHERWISE.
2502 ;*
2503 ;* CALLING SEQUENCE: JSR PC,CKTRAP
2504 ;*
2505 ;* COMMENTS: IF THIS SUBROUTINE CAUSES A TRAP, EITHER THE ADDRESS WHICH
2506 ;* IS LABELED ADRPTR WILL BE THE TRAP PC ADDRESS ON THE STACK.
2507 ;*
2508 ;* SUBORDINATE ROUTINES CALLED: NONE.
2509 .....
2510
2511 020124 CKTRAP:: SAVE ;SAVE CONTENTS OF GPRS R0 THRU R5.
2512 020124 004567 163646 ;R5,PREG05 ;CALL REGISTER SAVE SUBRT.
2513 020130 005067 162212 ;CLR TP4FLG ;CLEAR THE 004 TRAP FLAGS.
2514 020134 011011 ;MOV (R0),(R1) ;PERFORM THE MOVE IN QUESTION.
2515 020136 005767 162204 ADRPTR:: TST TP4FLG ;CHECK FOR OCCURENCE OF TRAP.
2516 020144 001401 SEC ;INDICATE SUCCESS.
2517 020146 000241 BEQ 60$ ;EXIT WITH SUCCESS IF TRAP DID NOT OCCUR.
2518 020150 000241 CLC ;INDICATE FAILURE.
2519 020152 000207 60$: PASS ;RESTORE GPRS.
2519 020152 000207 RTS PC JSR PC,8(SP). ;RETURN TO PREG05 SUBRT.

```

```
2521 .SBTTL GLOBAL SUBROUTINE CLNRST
2522 ;*****
2523 ; CLEAN RESET OF THE DEVICE UNDER TEST
2524 ; THIS SUBROUTINE IS USED TO RESET THE DUT TO A KNOWN STATE.
2525 ; THE DUT'S SELF TEST IS SKIPPED, AND THE FIFO IS PURGED OF ANY ERROR
2526 ; CODES, ETC.
2527 ; IF THE RESET DOES NOT SUCCESSFULLY COMPLETE, THEN THE CARRY BIT IS
2528 ; PASSED BACK TO THE CALLING ROUTINE (CLEAR).
2529 ;
2530 ; INPUTS: CSRA CONTAINS THE ADDRESS OF THE CSR
2531 ; TXBFCA CONTAINS ADDRESS OF DUT DMA BUFFER COUNT REGISTER.
2532 ; ERRNBR - ERROR NUMBER FOR POSSIBLE ERROR REPORT.
2533 ; ERRBLK - ERRTP,ERNBR,AND ERRMSG SET UP CORRECTLY.
2534 ;
2535 ; OUTPUTS: THE DUT PERFORMS ITS RESET FUNCTION INTO A KNOWN STATE.
2536 ; CARRY CLEAR INDICATES THE TEST IS TO BE ABORTED.
2537 ; ERRBLK - VALUE MAY BE DESTROYED.
2538 ; IESTAT - TX AND RX INTERRUPT FLAGS ARE CLEARED.
2539 ; TX AND RX INTERRUPT ENABLE BITS IN THE DUT'S CSR ARE CLEARED.
2540 ;
2541 ; CALLING SEQUENCE: JSR PC,CLNRST
2542 ;
2543 ; COMMENTS: THIS SUBROUTINE CAN REPORT ERRORS WITH NUMBERS ERRNBR.
2544 ; THIS ROUTINE DOES NOT DESTROY THE VALUE OF ERRNBR.
2545 ;
2546 ; SUBORDINATE ROUTINES CALLED: DELAY,MSLGET,PUFIFO,RESETT.
2547 ;*****
2548
2549 020154 CLNRST:: SAVE ;SAVE CONTENTS OF GPRS R0 THRU R5.
2550 020154 004567 163616 JSR R5,PREG05 ;CALL REGISTER SAVE SUBRT.
2551 ;
2552 ; RESET THE DUT.
2553 ; THIS ROUTINE REPORTS ERRORS WITH NUMBERS FROM ERRNBR THRU ERRNBR+2.
2554 ;
2554 020160 004767 001270 JSR PC,RESETT ;RESET THE DUT TO A KNOWN STATE.
2555 020164 103002 BCC 60$ ;EXIT ROUTINE WITH ABORT TEST INDICATOR.
2556 ;
2557 ; PURGE THE FIFO OF ERROR CODES, SAVE ANY BMP CODES FOUND.
2558 ;
2559 020166 004767 000522 JSR PC,PUFIFO ;PURGE THE FIFO.
2560 ;
2561 020172 60$: ;EXIT THE TEST USING RESETT OR PUFIFO STATUS.
2562 020172 PASS ;RESTORE GPRS, PASS THE FOLLOWING INTACT:
2563 020172 004736 JSR PC,B(SP). ;RETURN TO PREG05 SUBRT.
2564 020174 000207 RTS PC ;CARRY BIT-IF CLEAR, THEN ABORT THE TEST.
```

```

2566 .SBTTL GLOBAL SUBROUTINE CLR16W -
2567 ;... *****
2568 ;* CLEAR SIXTEEN WORDS ROUTINE -
2569 ;* THIS SUBROUTINE CLEARS 16 WORDS STARTING WITH THE SPECIFIED WORD.
2570 ;*
2571 ;* INPUTS: R0 ADDRESS OF THE FIRST WORD TO CLEAR.
2572 ;*
2573 ;* OUTPUTS: (R0) TO (R0+15) 16 WORDS OF MEMORY ARE CLEARED TO 0.
2574 ;*
2575 ;* CALLING SEQUENCE: JSR PC,CLR16W
2576 ;*
2577 ;* COMMENTS:
2578 ;*
2579 ;* SUBORDINATE ROUTINES CALLED: NONE.
2580 ;-- *****
2581
2582 020176 CLR16W:: SAVE ;SAVE CONTENTS OF GPRS R0 THRU R5.
2583 020176 004567 163574 JSR R5,PREG05 ;CALL REGISTER SAVE SUBRT.
2584 020202 012701 000020 ;SET THE LOOP COUNTER TO 16.
2585 020206 005020 ;CLEAR A WORD OF MEMORY.
2586 020210 005301 ;COUNT THIS LOOP.
2587 020212 001375 ;LOOP IF NOT 16 WORD CLEARED.
2588 020214 004736 ;RESTORE GPRS.
2589 020214 000207 JSR PC,@(SP) ;RETURN TO PREG05 SUBRT.
2590 RTS PC

```

```

2590 .SBTTL GLOBAL SUBROUTINE CNTERR
2591 ;... *****
2592 ;* COUNT ERROR ROUTINE -
2593 ;* THIS SUBROUTINE IS USED TO COUNT A "DATA" ERROR ON THE SPECIFIED
2594 ;* LINE. IT CHECKS WHETHER ERROR SUMMARY REPORTING IS ACTIVE, OR SHOULD
2595 ;* BE MADE ACTIVE ON THIS LINE, AND ACTIVATES IT IF NECESSARY.
2596 ;*
2597 ;* INPUTS: R5 LINE NUMBER OF LINE UNDER CONSIDERATION.
2598 ;* ERCNTB LABEL AT BASE OF ERROR COUNTERS TABLE.
2599 ;* ERSMAF - ERROR SUMMARY FLAGS (BIT SET IF LINE IN SUMMARY MODE).
2600 ;* NDERPT NUMBER OF INDIVIDUAL DATA ERRORS TO REPORT ON A LINE.
2601 ;*
2602 ;* OUTPUTS: CARRY - SET IF LINE IS IN ERROR SUMMARY MODE.
2603 ;* ERCNT ERROR COUNTER INCREMENTED FOR SPECIFIED LINE.
2604 ;* ERSMAF - BIT SET IF LINE SHOULD BE IN SUMMARY MODE.
2605 ;*
2606 ;* CALLING SEQUENCE: JSR PC,CNTERR
2607 ;*
2608 ;* COMMENTS:
2609 ;*
2610 ;* SUBORDINATE ROUTINES CALLED: NONE.
2611 ;* - - - - -
2612
2613 020220 CNTERR:: SAVE ;SAVE CONTENTS OF GPRS R0 HRU R5.
020220 004567 163552 JSR R5,PREG05 ;CALL REGISTER SAVE SUBRT.
2614
2615 ;* COUNT THE ERROR ON THE COUNTER FOR THE SPECIFIED LINE.
2616 ;
2617 020224 006305 ; ASL R5 ;FORM WORD OFFSET FROM LINE NUMBER.
2618 020226 016501 002464 ; MOV ERCNTB(R5),R1 ;GET THE PRESENT ERROR COUNT FOR THIS LINE.
2619 020232 005201 ; INC R1 ;COUNT ERROR.
2620 020234 103402 ; BCS 2$ ;OVERFLOW? YES, DON'T UPDATE COUNTER IN TABLE.
2621 020236 010165 002464 ; MOV R1,ERCNTB(R5) ;UPDATE ERROR COUNTER TABLE ENTRY.
2622 020242 005767 161762 2$: ; TST NDERPT
2623 020246 001411 ; BEQ 60$ ;SUMMARYS DISABLED? YES, EXIT WITH CARRY 0.
2624 020250 020167 161754 ; CMP R1,NDERPT ;NO, CHECK FOR ENOUGH ERRORS FOR SUMMARY USE.
2625 020254 101002 ; BHI 4$ ;ENOUGH ERRORS TO USE SUMMARY? YES, GO HANDLE.
2626 020256 000241 ; CLC ;INDICATE NOT TO USE SUMMARY REPORT YET.
2627 020260 000404 ; BR 60$ ;EXIT WITH CARRY 0.
2628 020262 056567 002410 162172 4$: ; BIS BITTBL(R5),ERSMAF ;SET THE ERROR SUMMARY FLAG FOR LINE.
2629 020270 000261 ; SEC ;INDICATE TO USE SUMMARY REPORT.
2630 020272 004736 60$: ; PASS ;RESTORE GPRS.
020272 004736 JSR PC,@(SP) ;RETURN TO PREG05 SUBRT.
2631 020274 000207 ; RTS PC

```

```

2633 .SBTTL GLOBAL SUBROUTINE DELAY
2634 ;*****
2635 ;* DELAY SUBROUTINE -
2636 ;* THIS SUBROUTINE IS USED TO DELAY A VARIABLE NUMBER OF MILLI SECONDS.
2637 ;*
2638 ;* INPUTS: R4 CONTAINS THE NUMBER OF MS TO DELAY.
2639 ;* MSLCNT.
2640 ;*
2641 ;* OUTPUTS: NONE.
2642 ;*
2643 ;* CALLING SEQUENCE: JSR PC,DELAY
2644 ;*
2645 ;* COMMENTS: IF NO HARDWARE CLOCK INTERRUPTS ARE OCCURING, CONTROL CS WILL
2646 ;* NOT BE HONORED FOR THE DURATION OF THE DELAY.
2647 ;*
2648 ;* SUBORDINATE ROUTINES CALLED: NONE.
2649 ;*****
2650
2651 020276 DELAY:: SAVE ;SAVE CONTENTS OF GPRS RC THRU R5.
2652 020276 004567 163474 JSR R5,PREG05 ;CALL REGISTER SAVE SUBRT.
2653 020302 010401 MOV R4,R1 ;PASS NUMBER OF MS DELAY AS TIME-OUT VALUE.
2654 020310 005003 MOV #-1,R2 ;TELL MSLOOP ROUTINE TO CHECK ALL BITS.
2655 020312 012704 020334 CLR R3 ;TELL MSLOOP RTN TO CHECK FOR ALL BITS CLEAR.
2656 020316 004767 000130 MOV #62$,R4 ;TELL MSLOOP TO CHECK DUMMY NON-ZERO WORD.
2657 020322 103002 JSR PC,MSLOOP ;DELAY THE REQUESTED # OF MS.
2658 020324 004767 000136 BCC 60$ ;EXIT ROUTINE IF WE TIMED-OUT.]
2659 020330 020330 004736 JSR PC,OOPS ;IF NO TIME OUT, BAD PROGRAM OR HOST MACHINE.
2660 020332 000207 60$. PASS ;RESTORE GPRS.
2661 ; PC,@(SP). ;RETURN TO PREG05 SUBRT.
2662 020334 177777 62$: .WORD 1 ;DUMMY, NON ZERO WORD.

```

```
2664 .SBTTL GLOBAL SUBROUTINE MSLGET
2665 ;*****
2666 ;* - MILLI SECONDS LOOP WHICH RETURNS READ WORD AND REMAINING TIME -
2667 ;* THIS SUBROUTINE IS A GENERAL PURPOSE TEST LOOP SUBROUTINE. IT IS USED
2668 ;* TO VERIFY THAT A CERTAIN ACTION OCCURS BEFORE A TIME-OUT PERIOD. THE
2669 ;* CALLING ROUTINE PASSES IN WHICH BITS SHOULD BE SET AND CLEARED FOR THE
2670 ;* DESIRED CONDITION AND THE TIME-OUT VALUE IN MILLI-SECONDS.
2671 ;* THIS ROUTINE CHECKS FOR THE DESIRED CONDITION UPON ENTRANCE INTO THE
2672 ;* ROUTINE AND THEN ONCE EACH MILLI SECOND THERE AFTER.
2673 ;* UPON RETURN, THE LAST WORD WHICH WAS READ TO CHECK FOR THE CONDIT'ION
2674 ;* IS RETURNED BY THIS SUBROUTINE.
2675 ;*
2676 ;* INPUTS: R1 - TIME-OUT VALUE IN MILLI-SECONDS (UP TO 64K MS).
2677 ;* R2 - BIT MAP OF BITS TO TEST (1 INDICATES TO TEST THE BIT).
2678 ;* R3 - DESIRED STATES OF THE INDICATED FIELDS IN R2.
2679 ;* R4 - ADDRESS OF THE WORD TO TEST.
2680 ;* MSLCNT - MILLI SECOND SOFTWARE LOOP COUNT.
2681 ;*
2682 ;* OUTPUTS: R0 - THE LAST WORD WHICH WAS READ TO CHECK FOR THE CONDITION.
2683 ;* R1 - REMAINING NUMBER OF MS IN TIME-OUT TIME.
2684 ;* CARRY - SUCCESS FLAG (SET IF CONDITION IS MET BEFORE TIME-OUT).
2685 ;*
2686 ;* CALLING SEQUENCE: JSR PC,MSLGET
2687 ;*
2688 ;* COMMENTS: THIS ROUTINE WORKS WITH OR WITHOUT A HARDWARE CLOCK, BUT THE
2689 ;* CALIBRATION IS ONLY GUARENTEED WHEN A LINE CLOCK IS AVAILABLE
2690 ;* ON THE SYSTEM.
2691 ;* THIS ROUTINE CAN BE USED AS A DELAY ROUTINE, BY SPECIFYING THE
2692 ;* DESIRED DELAY AS THE TIME-OUT AND SPECIFYING A CONDITION TO
2693 ;* LOOK FOR WHICH WILL NOT BE MET DURING THE DELAY.
2694 ;* IF A TIME-OUT VALUE OF 0 IS SPECIFIED, THIS ROUTINE CHECKS FOR
2695 ;* THE DESIRED CONDITION BEFORE RETURNING. IT INDICATES SUCCESS
2696 ;* IF THE CONDITION IS MET, FAILURE OTHERWISE.
2697 ;*
2698 ;*
2699 ;* SUBORDINATE ROUTINES CALLED: NONE.
2700 ;*****
2701
2702 020336 MSLGET:: SAVE ;SAVE CONTENTS OF GPRS R0 THRU R5.
2703 020336 004567 163434 JSR R5,PREG05 ;CALL REGISTER SAVE SUBRT.
2704 ;*
2705 ;* SET UP MASK FOR REMOVING UNUSED BITS IN THE TEST WORD, AND CLEAR UNUSED
2706 ;* BITS IN THE DESIRED STATE WORD TO ALLOW DIRECT COMPARISON.
2707 ;*
2707 020342 005102 COM R2 ;GET MASK OF UNUSED BITS.
2708 020344 040203 BIC R2,R3 ;MASK OUT UNUSED BITS IN DESIRED STATE WORD.
2709 ;*
2710 ;* HANDLE THE TEST AND EXIT IF WE HAVE A 0 TIME-OUT VALUE.
2711 ;*
2712 020346 005701 TST R1 ;TEST THE TIME-OUT VALUE FOR ZERO.
2713 020350 001011 BNE 2$ ;IF NON-ZERO TIME-OUT, GO LOOP AND TEST.
2714 020352 011400 MOV (R4),R0 ;GET THE WORD TO TEST BEFORE EXITING.
2715 020354 010067 000070 MOV R0,R2$ ;SAVE VALUE SO WE CAN RETURN IT.
2716 020360 040200 BIC R2,R0 ;MASK OUT UNTESTED BITS OF WORD.
2717 020362 020003 CMP R0,R3 ;COMPARE AGAINST DESIRED STATE WORD.
2718 020364 000261 SEC ;INDICATE SUCCESS IN CASE WORDS ARE EQUAL.
2719 020366 001420 BEQ 6$ ;EXIT WITH SUCCESS IF WORDS ARE EQUAL.
```

GLOBAL SUBROUTINE

MSLGET

```

2720 020370 000241          CLC          ;INDICATE FAILURE (TIME OUT).
2721 020372 000416          BR          68 ;EXIT WITH FAILURE, WORDS AREN'T EQUAL.
2722                          ;*
2723                          ; NON ZERO TIME-OUT VALUE. LOOP, WAITING FOR CONDITION OR TIME OUT.
2724                          ;
2725 020374 016705 161776    28:      MOV      MSLCNT,R5      ;LOAD MS LOOP COUNT.
2726 020400 011400          48:      MOV      (R4),R0        ;GET THE WORD TO TEST.
2727 020402 010067 000042    48:      MOV      R0,628        ;SAVE WORD IN CASE THIS IS THE LAST.
2728 020406 040200          BIC      R2,R0        ;MASK OUT UNTESTED BITS OF WORD.
2729 020410 020003          CMP      R0,R3        ;COMPARE AGAINST DESIRED STATE WORD.
2730 020412 000261          SEC                ;SET CARRY IN CASE OF SUCCESS.
2731 020414 001405          BEQ      68          ;EXIT WITH SUCCESS IF WORDS ARE EQUAL.
2732 020416 005305          DEC      R5          ;COUNT DOWN THE INSIDE MS LOOP COUNT.
2733 020420 001367          BNE      48          ;LOOP IF MS NOT UP.
2734 020422 005301          DEC      R1          ;DECREMENT THE MS TIME COUNT.
2735 020424 001363          BNE      28          ;IF TIME NOT UP, LOOP TO COUNT ANOTHER MS.
2736 020426 000241          CLC                ;CLEAR CARRY, WE TIMED-OUT.
2737                          ;*
2738                          ; HAVE EITHER FOUND CONDITION, OR TIMED-OUT (POSSIBLY FROM 0 TIME OUT VALUE).
2739                          ; RESTORE THE LAST CONTENTS READ FROM THE TEST WORD. EXIT ROUTINE.
2740                          ;
2741 020430 016700 000014    68:      MOV      628,R0        ;PASS OUT THE LAST READ WORD.
2742 020434 010066 000002    608:     PASS      R0,R1        ;RESTORE GPRS, EXCEPT THE FOLLOWING:
                                MOV      R0,R0SLOT(SP)      ;PUT R0 IN STACK SLOT.
                                MOV      R1,R1SLOT(SP)      ;PUT R1 IN STACK SLOT.
                                JSR      PC,8(SP)           ;RETURN TO PREG05 SUBRT.
2743                          ;R0 LAST READ WORD CHECKED FOR CONDITION.
2744                          ;R1 REMAINING TIME (0 IF TIME OUT OCCURED).
2745 020446 000207          RTS      PC          ;CARRY SET IF SUCCESS, CLEAR IF TIME OUT.
2746                          ;*
2747                          ; LOCAL STORAGE.
2748                          ;
2749 020450 000000          628:      .WORD      0          ;STORAGE FOR THE LAST READ WORD.

```

GLOBAL SUBROUTINE.

MSLOOP -

```

2751 .SBTTL GLOBAL SUBROUTINE MSLOOP
2752 :*****
2753 :* TEST LOOP SUBROUTINE -
2754 :* THIS SUBROUTINE IS A GENERAL PURPOSE TEST LOOP SUBROUTINE. IT IS USED
2755 :* TO VERIFY THAT A CERTAIN ACTION OCCURS BEFORE A TIME-OUT PERIOD. THE
2756 :* CALLING ROUTINE PASSES IN WHICH BITS SHOULD BE SET AND CLEARED FOR THE
2757 :* DESIRED CONDITION AND THE TIME-OUT VALUE IN MILLI-SECONDS.
2758 :* THIS ROUTINE CHECKS FOR THE DESIRED CONDITION UPON ENTRANCE INTO THE
2759 :* ROUTINE AND THEN ONCE EACH MILLI-SECOND THEREAFTER.
2760 :*
2761 :* INPUTS: R1 TIME-OUT VALUE IN MILLI-SECONDS (UP TO 64K MS).
2762 :* R2 - BIT MAP OF BITS TO TEST (1 INDICATES TO TEST THE BIT).
2763 :* R3 - DESIRED STATES OF THE INDICATED FIELDS IN R2.
2764 :* R4 ADDRESS OF THE WORD TO TEST.
2765 :* MSLCNT - MILLI SECOND SOFTWARE LOOP COUNT.
2766 :*
2767 :* OUTPUTS: CARRY - SUCCESS FLAG (SET IF CONDITION IS MET BEFORE TIME OUT).
2768 :*
2769 :* CALLING SEQUENCE: JSR PC,MSLOOP
2770 :*
2771 :* COMMENTS: THIS ROUTINE WORKS WITH OR WITHOUT A HARDWARE CLOCK, BUT THE
2772 :* CALIBRATION IS ONLY GUARENTEED WHEN A LINE CLOCK IS AVAILABLE
2773 :* ON THE SYSTEM.
2774 :* THIS ROUTINE CAN BE USED AS A DELAY ROUTINE, BY SPECIFYING THE
2775 :* DESIRED DELAY AS THE TIME-OUT AND SPECIFYING A CONDITION TO
2776 :* LOOK FOR WHICH WILL NOT BE MET DURING THE DELAY.
2777 :* IF A TIME-OUT VALUE OF 0 IS SPECIFIED, THIS ROUTINE CHECKS FOR
2778 :* THE DESIRED CONDITION BEFORE RETURNING. IT INDICATES SUCCESS
2779 :* IF THE CONDITION IS MET, FAILURE OTHERWISE.
2780 :*
2781 :* SUBORDINATE ROUTINES CALLED: MSLGET.
2782 :*****
2783
2784 020452 MSLOOP:: SAVE ;SAVE CONTENTS OF GPRS R0 THRU R5.
2785 020452 004567 163320 JSR R5,PREG05 ;CALL REGISTER SAVE SUBRT.
2786
2787 ;
2788 ; CALLING THE MSLGET ROUTINE FROM THE MSLOOP ROUTINE ISOLATES THE CALLER OF
2789 ; MSLOOP FROM THE RETURNED TEST WORD AND REMAINING TIME-OUT VALUES.
2790 020456 004767 177654 JSR PC,MSLGET ;CALL THE MULTI PURPOSE MS LOOP AND SEARCH RTN.
2791
2792 020462 601: PASS ;RESTORE GPRS,
2793 020462 004736 JSR PC,B(SP). ;RETURN TO PREG05 SUBRT.
2793 020464 000207 RTS PC ;CARRY - SET IF SUCCESS, CLEAR IF TIME-OUT.

```

GLOBAL SUBROUTINE

OOPS -

```

2795 .SBTTL GLOBAL SUBROUTINE                                OOPS
2796 ;... *****
2797 ;* PROGRAM ABORT SUBROUTINE
2798 ;* THIS SUBROUTINE IS USED TO ABORT THE PROGRAM WHEN A FATAL ERROR IS
2799 ;* DETECTED IN THE PROGRAM OR THE HOST SYSTEM HARDWARE. AN ERROR MESSAGE
2800 ;* IS PRINTED GIVING SOME INFORMATION ABOUT THE NATURE OF THE ABORT.
2801 ;*
2802 ;* INPUTS:      R1  ERROR CODE GIVING REASON FOR ABORT.
2803 ;*
2804 ;* OUTPUTS:     AN ERROR MESSAGE IS PRINTED.
2805 ;*              A LIST OF RETURN PC VALUES FOR ALL SUBROUTINE CALLS IS PRINTED.
2806 ;*
2807 ;* CALLING SEQUENCE:  JSR  PC,OOPS
2808 ;*
2809 ;* COMMENTS:
2810 ;*
2811 ;* SUBORDINATE ROUTINES CALLED: NONE.
2812 ; - *****
2813
2814 020466 OOPS:: SAVE ;SAVE CONTENTS OF GPRS R0 THRU R5.
2815 020466 004567 163304 JSR R5,PREG05 ;CALL REGISTER SAVE SUBRT.
2816 ; REPORT "HOST COMPUTER HARDWARE OR SOFTWARE BUG ENCOUNTERED." ERROR.
2816 020472 ERRSF 101,EM0101
2816 020472 104454 TRAP C$ERSF
2816 020474 000145 .WORD 101
2816 020476 020532 .WORD EM0101
2816 020500 000000 .WORD 0
2817 ; REPORT 'PROGRAM HUNG. WAITING FOR A CONTROL C.'
2818 020502 PRINTF #EM0102
2818 020502 012746 020616 MOV #EM0102, (SP)
2818 020506 012746 000001 MOV #1, -(SP)
2818 020512 010600 MOV SP,R0
2818 020514 104417 TRAP C$PNTF
2818 020516 062706 000004 ADD #4,SP
2819 020522 2$: BREAK ;LOOK FOR OPERATOR CONTROL C INPUT.
2819 020522 104422 TRAP C$BRK
2820 020524 000776 BR 2$ ;INFINITE LOOP.
2821 020526 60$: PASS ;DON'T NEED THIS, BUT SOMEBODY MAY CHANGE THIS
2821 020526 004736 JSR PC,@(SP); ;RETURN TO PREG05 SUBRT.
2822 020530 000207 RTS PC ; ROUTINE IN THE FUTURE, SO BE CONSISTANT.
2823
2824 020532 110 117 123 EM0101:: .ASCIZ /HOST COMPUTER HARDWARE OR SOFTWARE BUG ENCOUNTERED./
2824 020535 124 040 103
2824 020540 117 115 120
2824 020543 125 124 105
2824 020546 122 040 110
2824 020551 101 122 104
2824 020554 127 101 122
2824 020557 105 040 117
2824 020562 122 040 123
2824 020565 117 106 124
2824 020570 127 101 122
2824 020573 105 040 102
2824 020576 125 107 040
2824 020601 105 116 103
2824 020604 117 125 116
2824 020607 124 105 122

```

GLOBAL SUBROUTINE

OOPS -

	020612	105	104	056	
	020615	000			
2825	020616	045	116	045	EM0102:: .ASCIZ /ENMAPROGRAM HUNG, WAITING FOR A CONTROL-C. <*****ENEN/
	020621	101	120	122	
	020624	117	107	122	
	020627	101	115	040	
	020632	110	125	116	
	020635	107	054	040	
	020640	127	101	111	
	020643	124	111	116	
	020646	107	040	106	
	020651	117	122	040	
	020654	101	040	103	
	020657	117	116	124	
	020662	122	117	114	
	020665	055	103	056	
	020670	040	074	052	
	020673	052	052	052	
	020676	052	052	052	
	020701	052	052	052	
	020704	052	052	052	
	020707	045	116	045	
2826	020712	116	000		

.EVEN

GLOBAL SUBROUTINE

PUFIFO -

```

2828 .SBTTL GLOBAL SUBROUTINE PUFIFO
2829 ;*****
2830 ;* PURGE THE FIFO
2831 ;* THIS ROUTINE TRIES TO REMOVE ALL THE CHARACTERS FROM THE FIFO.
2832 ;* ANY BMP CODES THAT ARE FOUND ARE SAVED ON THE BMP CODE QUEUE.
2833 ;*
2834 ;* INPUTS: RBUFA- CONTAINS THE ADDRESS OF THE RECEIVER.
2835 ;*
2836 ;*
2837 ;* OUTPUTS: CARRY BIT - INDICATES THE STATE OF THE FIFO, SET=* PURGED.
2838 ;* BMPCQ THE CONTENTS OF THE BMP CODE QUEUE MAY BE UPDATED.
2839 ;*
2840 ;* CALLING SEQUENCE: JSR PC,PUFIFO
2841 ;*
2842 ;* COMMENTS:
2843 ;*
2844 ;* SUBORDINATE ROUTINES CALLED: SAVBMP.
2845 ;*****
2846
2847 020714 PUFIFO::SAVE ;SAVE CONTENTS OF GPRS R0 THRU R5.
020714 004567 163056 JSR R5,PREG05 ;CALL REGISTER SAVE SUBRT.
2848 020720 012701 001000 MOV #512.,R1 ;SET MAXIMUM TRY COUNT OF 512.
2849 020724 016704 161316 MOV RBUFA,R4 ;GET ADDRESS OF THE RECEIVER BUFFER REGISTER.
2850
2851 020730 011402 2$: MOV (R4),R2 ;GET THE CONTENTS OF THE RECEIVER BUFFER REG.
2852 020732 100016 BPL 6$ ;EXIT IF THE FIFO IS EMPTY, DATA_VALID CLR.
2853
2854 ;*
2855 ; CHECK IF THE READ CHARACTER IS ACTUALLY A BMP CODE.
2856 ; IF IT IS, THEN SAVE IT ON THE BMP CODE QUEUE TO BE REPORTED LATER.
2857 020734 012700 070000 ;*
2858 020740 040200 MOV #70000,R0 ;GENERATE A BIT MAP OF CHAR ERROR BITS
2859 020742 001006 BIC R2,R0 ; WHICH ARE NOT SET FOR CHAR.
2860 BNE 4$ ;THROW CHAR AWAY IF NOT BMP OR SELFTEST CODE.
2861
2862 ;*
2863 ; CHECK IF THE READ DATA IS MODEM STATUS , BMP OR SELFTEST?.
2864 020744 012700 000300 ;*
2865 020750 040200 MOV #300,R0 ; CHECK IF BMP OR SELFTEST?.
2866 020752 001002 BIC R2,R0 ;TRY TO CLEAR BMP FLAGS IN THE READ DATA.
2867 020754 004767 001306 BNE 4$ ;IF IT IS MODEM OR SELFTEST CODE THROW IT AWAY.
2868 020760 005301 JSR PC,SAVBMP ;SAVE BMP CODE ON THE QUEUE.
2869 020762 001362 4$: DEC R1 ;DECREMENT THE TRY COUNT.
2870 020764 000241 BNE 2$ ;LOOP TO TRY AGAIN.
2871 020766 000401 CLC ;CLEAR CARRY,TO INDICATE FIFO NOT PURGED.
2872 020770 000261 BR 60$ ;EXIT WITH CARRY CLEAR.
2873 6$: SEC ;SET CARRY, TO INDICATE FIFO PURGED.
2874 020772 004736 60$: PASS ;RESTORE GPRS,
020772 PC,@(SP)+ JSR ;RETURN TO PREG05 SUBRT.
2875 ;CARRY BIT, SET INDICATES FIFO PURGED.
2876 020774 000207 RTS PC

```

GLOBAL SUBROUTINE

RDPDR

```

2878 .SBTTL GLOBAL SUBROUTINE RDPDR
2879 ;*****
2880 ;* READ AND VERIFY DATA PATTERN FROM DEVICE REGISTERS ROUTINE
2881 ;* THIS ROUTINE READS AND VERIFIES THE ROTATED DATA PATTERN WHICH HAS
2882 ;* BEEN WRITTEN BY THE WOPDR SUBROUTINE.
2883 ;* EACH ACTIVE LINE'S REGISTER'S CONTENTS IS READ AND COMPARED WITH THE
2884 ;* WRITTEN DATA.
2885 ;* AFTER THE UNUSED AND READ ONLY (RO) BITS ARE MASKED OUT, ANY ERRORS ARE
2886 ;* REPORTED FROM THIS ROUTINE.
2887 ;* THIS ROUTINE WILL TAKE INTO ACCOUNT THE TYPE OF WRITE OPERATION WHICH
2888 ;* WAS PERFORMED BY THE WOPDR SUBROUTINE.
2889 ;*
2890 ;* INPUTS: R2 - USED TO PASS IN THE DATA PATTERN TO BE ROTATED & VERIFIED.
2891 ;* R3 - BYTE INDICATOR (- => LO BYTE, . => HI BYTE, 0 => BOTH).
2892 ;* R4 - OPERATION TYPE INDICATOR (- => BIC, . => BIS, 0 => MOV).
2893 ;* ACTLNS - BIT MAP OF ACTIVE LINES ON THE DEVICE UNDER TEST.
2894 ;* CSRA - CONTAINS THE CSR ADDRESS OF THE DEVICE UNDER TEST.
2895 ;* DRADRT - BASE ADDRESS OF DEVICE REGISTER ADDRESS TABLE.
2896 ;* ERCNTB - LABEL AT BASE OF ERROR COUNTERS TABLE FOR LINES.
2897 ;* ERRMSG - SET UP WITH THE PROPER ERROR MESSAGE FOR THIS TEST.
2898 ;* ERRNBR - SET UP WITH THE PROPER ERROR NUMBER.
2899 ;* LPRO - EQUATED TO LPR REG OFFSET FROM DEVICE CSR ADDRESS.
2900 ;* NUMLNS - NUMBER OF LINES ON THE DEVICE UNDER TEST.
2901 ;* NDERPT - NUMBER OF INDIVIDUAL DATA ERRORS TO REPORT ON A LINE.
2902 ;* TXBFCO - EQUATED TO TBUFFT REG OFFSET FROM DEVICE CSR ADDRESS.
2903 ;* UNBITB - BASE ADDRESS OF THE UNUSED BIT TABLE.
2904 ;*
2905 ;* OUTPUTS: ERROR MESSAGES MAY BE PRINTED AT THE OPERATOR'S CONSOLE.
2906 ;* ERCNT - ERROR COUNTERS TABLE IS UPDATED FOR LINE UNDER TEST.
2907 ;* ERRBLK - CONTENTS DESTROYED.
2908 ;* ERSMRF - ERROR SUMMARY FLAGS BIT SET IF LINE IN SUMMARY MODE.
2909 ;* UUT CSR - ALL BITS CLEARED, EXCEPT IND.ADR.REG FIELD DESTROYED.
2910 ;*
2911 ;* CALLING SEQUENCE: JSR PC,RDPDR
2912 ;*
2913 ;* COMMENTS: FOR BYTE ACCESSES, ONLY THE SPECIFIED BYTE IS VERIFIED.
2914 ;*
2915 ;* SUBORDINATE ROUTINES CALLED: ER1601,ROLDAP.
2916 ;* *****
2917
2918 020776 RDPDR:: SAVE ;SAVE CONTENTS OF GPRS R0 THRU R5.
2919 020776 004567 162774 JSR R5,PREG05 ;CALL REGISTER SAVE SUBRT.
2920 021002 012767 016710 162764 MOV #ER1601,ERRBLK ;SET UP THE ADDRESS OF THE ERROR REPORT RTN.
2921 ;*
2922 ;* DETERMINE WHETHER REGISTER DATA SHOULD BE INVERTED FROM DATA PATTERN.
2923 ;*
2924 021010 005704 TST R4 ;CHECK THE OPERAND TYPE INDICATOR.
2925 021012 100001 BPL 21 ;BIC WRITE PERFORMED? NO, USE STANDARD DATA.
2926 021014 005102 COM R2 ;YES, INVERT THE DATA PATTERN.
2927 ;*
2928 ;* SET UP OUTER LOOP.
2929 021016 005005 21: CLR R5 ;CLEAR LINE COUNTER TO SELECT LINE 0.
2930 ;*
2931 ;* THE OUTER LOOP FOLLOWS. EACH PASS THROUGH THIS LOOP READS AND COMPARES DATA
2932 ;* FROM ALL OF THE DEVICE REGISTERS FOR A PARTICULAR LINE IF THE LINE IS ACTIVE.
2933 ;

```

GLOBAL SUBROUTINE

RDPDR

```

2934 021020 010267 000170      4:      MOV      R2,70:      ;SAVE THE OUTER LOOP DATA PATTERN.
2935 021024 010577 161214      MOV      R5,@CSRA      ;SET CSR IND.ADR.REG FIELD TO THIS LINE.
2936 021030 010500      MOV      R5,R0
2937 021032 006300      ASL      R0
2938 021034 036067 002410 161174      BIT      BIT,BL(R0),ACTLNS
2939 021042 001452      BEQ      16:      ;IS THE LINE ACTIVE? NO, SKIP THE LINE.
2940 021044 012703 000004      MOV      @LPRO,R3      ;YES, INITIALIZE REGISTER OFFSET FOR LPR.
2941      ;
2942      ; THE INNER LOOP FOLLOWS. EACH PASS THROUGH THIS LOOP READS AND COMPARES
2943      ; DATA FROM A DEVICE REGISTER.
2944      ;
2945 021050 010204      6:      MOV      R2,R4      ;SAVE THE INNER LOOP DATA PATTERN.
2946 021052 046302 002264      BIC      UNBITB(R3),R2      ;REMOVE UNUSED BITS FROM EXPECTED DATA.
2947 021056 016300 002244      MOV      DRADRT(R3),R0
2948 021062 005766 000010      TST      R3SLOT(SP)      ;CHECK THE ACCESS TYPE INDICATOR.
2949 021066 001002      BNE      8:      ;BYTE ACCESS? YES, GO PERFORM BYTE READ.
2950 021070 011001      MOV      (R0),R1      ;NO, PERFORM WORD READ OF DEVICE REGISTER.
2951 021072 007416      BR      12:
2952 021074 100410      8:      BMI      10:      ;LOW BYTE ACCESS? YES, GO DO LOW BYTE READ.
2953 021076 005200      INC      R0      ;HIGH BYTE ACCESS. FORM HIGH BYTE ADDRESS.
2954 021100 111001      MOV      (R0),R1      ;READ THE HI BYTE OF THE DUT REGISTER.
2955 021102 000301      SWAB      R1      ;PUT HI BYTE BACK INTO THE HI BYTE.
2956 021104 042701 000377      BIC      @377,R1      ;REMOVE THE UNUSED BYTE IN ACTUAL DATA.
2957 021110 042702 000377      BIC      @377,R2      ;REMOVE THE UNUSED BYTE IN EXPECTED DATA.
2958 021114 000405      BR      12:
2959 021116 111001      10:     MOV      (R0),R1      ;READ THE LOW BYTE OF THE DUT REGISTER.
2960 021120 042701 177400      BIC      @177400,R1      ;REMOVE THE UNUSED BYTE.
2961 021124 042702 177400      BIC      @177400,R2      ;FORM EXPECTED LOW BYTE FOR COMAPARISON.
2962
2963 021130 046301 002264      12:     BIC      UNBITB(R3),R1      ;REMOVE UNUSED BITS FROM ACTUAL DATA.
2964 021134 020102      CMP      R1,R2      ;COMPARE ACTUAL AND EXPECTED DATA.
2965 021136 001404      BEQ      14:      ;ACTUAL = EXPECTED? YES, SKIP ERROR.
2966 021140 004767 177054      JSR      PC,CTERR      ;NO, COUNT THE ERROR, CHECK FOR ERROR SUMMARY.
2967 021144 103401      BCS      14:      ;USE ERROR SUMMARY? YES, SKIP ERROR.
2968      ;NO, REPORT "BAD BIT(S) IN DEVICE XXXXX REGISTER FOR LINE NN (D)"
2969 021146      ERROR
2970 021150 010402      14:     MOV      R4,R2      ;RESTORE THE INNER LOOP DATA PATTERN.
2971 021152 004767 000410      JSR      PC,ROLDAP      ;ROTATE DATA PATTERN LEFT, NOT THROUGH CARRY.
2972 021156 062703 000002      ADD      @2,R3      ;SET REGISTER OFFSET TO THE NEXT REGISTER.
2973 021162 020327 000016      CMP      R3,@TXBFC0      ;COMPARE REG OFFSET WITH OFFSET OF LAST REG.
2974 021166 003730      BLE      6:      ;LOOP IF NOT ALL REG DONE FOR THIS LINE.
2975      ;
2976      ; BACK INTO THE OUTER LOOP. NOW SET UP FOR NEXT LINE. LOOP IF NOT DONE.
2977      ;
2978 021170 016702 000020      16:     MOV      70:,R2      ;SET UP TO ROTATE THE DATA PATTERN.
2979 021174 004767 000366      JSR      PC,ROLDAP      ;ROTATE THE DATA PATTERN.
2980 021200 005205      INC      R5      ;COUNT THIS LINE
2981 021202 020527 000010      CMP      R5,@NUMLNS      ;COMPARE LINE COUNT WITH NUMBER OF LINES.
2982 021206 002704      BLT      4:      ;LOOP IF SOME LINES NOT DONE.
2983
2984 021210      60:     PASS      ;RESTORE GPRS.
2985 021210 004736      JSR      PC,@(SP),      ;RETURN TO PREG05 SUBRT.
2986 021212 000207      RTS      PC
2987 021214 000000      70:     .WORD      0      ;STORAGE FOR DATA PATTERN OUTSIDE INNER LOOP.

```

GLOBAL SUBROUTINE

REGTST

```

2989 .SBTTL GLOBAL SUBROUTINE REGTS*
2990 ;*****
2991 ;* - REGISTERS TEST SUBROUTINE
2992 ;* SUBROUTINE TO TEST THE DEVICE UNDER TEST (DUT) REGISTERS. THE USED
2993 ;* BITS OF THE REGISTERS ARE EITHER ALL CLEARED OR ALL SET AND THEN THE
2994 ;* DATA PATTERN IS WRITTEN AND VERIFIED USING EITHER WORD OR BYTE
2995 ;* ACCESSES IN READ/WRITE OR READ/MODIFY/WRITE MODE.
2996 ;*
2997 ;* INPUTS: R3 BYTE INDICATOR ( - -> LOW, + -> HIGH, 0 -> BOTH BYTES ).
2998 ;* R4 ACCESS MODE ( 1 -> SET THEN PIC, 1 -> CLEAR THEN BIS,
2999 ;* ( 2 -> SET THEN MOV, +2 CLEAR THEN MOV ).
3000 ;* ERRNBR - SET UP WITH INITIAL ERROR NUMBER.
3001 ;*
3002 ;* OUTPUTS: GPRS0 GPR SAVE AREA 0 IS DESTROYED.
3003 ;* DEVICE UNDER TEST REGISTERS ARE WRITTEN.
3004 ;* ERROR MESSAGES MAY BE PRINTED AT THE OPERATORS CONSOLE.
3005 ;*
3006 ;* CALLING SEQUENCE: JSR PC,REGTST
3007 ;*
3008 ;* COMMENTS: THIS ROUTINE LOOP 16 TIMES WRITING THE SAME DATA PATTERN
3009 ;* ROTATED LEFT ONCE EACH ITERATION.
3010 ;* THIS ROUTINE CAN REPORT ERRORS INITIAL ERRNBR THRU INITIAL+2.
3011 ;*
3012 ;* SUBORDINATE ROUTINES CALLED: RDPDR,ROLDAP,SWAPO,WOPDR
3013 ;* *****
3014
3015 021216 REGTST:: SAVE ;SAVE CONTENTS OF GPRS R0 THRU R5.
021216 004567 162554 JSR R5,PREGOS ;CALL REGISTER SAVE SUBRT.
3016
3017 ; SET UP THE GPRS FOR THE WRITING OF THE DATA PATTERN.
3018 ;
3019 021222 012705 000020 MOV #16,R5 ;SET UP LOOP COUNTER TO COUNT 16 ITERATIONS.
3020 021226 012702 167410 MOV #167410,R2 ;INITIALIZE THE DATA PATTERN.
3021 021232 032704 000001 BIT #BIT0,R4 ;TEST FOR R/W ACCESS.
3022 021236 001001 BNE 2$ ;R/M/W ACCESS? YES, R4 IS ALL SET UP.
3023 021240 005004 CLR R4 ;NO. INDICATE R/W ACCESS.
3024 021242 2$:
3025 ;
3026 ; SET UP THE GPRS FOR THE CLEARING OR SETTING OF ALL THE USED BITS.
3027 ;
3028 021242 010400 MOV R4,R0 ;PASS OPERATION TYPE INDICATOR AROUND SWAPO.
3029 021244 004767 001142 JSR PC,SWAPO ;GET ALTERNATE GPR SET IN R1 THRU R5.
3030 021250 016701 162514 MOV ERRNBR,R1 ;SAVE THE INITIAL ERROR NUMBER.
3031 021254 010004 MOV R0,R4
3032 021256 005404 NEG R4 ;SET UP OP TYPE FOR CLEARING OR SETTING.
3033 021260 005002 CLR R2 ;SET UP CLEAR WRITE PATTERN.
3034 021262 026627 000012 000002 CMP R4,SLOT(SP),#2 ;TEST FOR CLEAR THEN MOV TEST SEQUENCE.
3035 021270 001401 BEQ 4$ ;CLEAR THEN MOV? YES, LEAVE WRITE PAT CLEAR.
3036 021272 005102 COM R2 ;NO, SET ALL BITS OF WRITE PATTERN.
3037 021274 005003 4$: CLR R3 ;INDICATE THAT WORD ACCESSES SHOULD BE USED.
3038 021276 005000 CLR R0 ;SET ALTERNATE BYTE EXPECTED DATA PAT TO CLEAR.
3039 021300 026627 000012 177776 CMP R4,SLOT(SP),#2 ;TEST FOR SET THEN MOV TEST SEQUENCE.
3040 021306 001001 BNE 6$ ;SET THEN MOV? YES, LEAVE ALT BYTE PAT CLEAR.
3041 021310 005100 COM R0 ;NO, SET ALT BYTE EXPECTED DATA PAT TO ALL 1 S.
3042 021312 004767 001074 6$: JSR PC,SWAPO ;RESTORE SWAPPED GPR VALUES TO R1 THRU R5.
3043 ;
3044 ; START OF DATA PATTERN LOOP.

```

GLOBAL SUBROUTINE

REGIST .

```

3045      ;
3046 021316      81:
3047      ;
3048      ; SET OR CLEAR ALL THE USED BITS OF THE DEVICE REGISTERS FOR ALL LINES.
3049      ; VERIFY THAT ALL THE BITS WERE SET OR CLEAPED CORRECTLY.
3050      ;
3051 021316 004767 001070      JSR      PC,SWAPO      ;GET ALTERNATE GPRS FOR SETTING INTIAL STATES.
3052 021322 004767 002006      JSR      PC,WDPDR      ;GO CLEAR ALL USED REGISTER BITS, ALL LINES.
3053 021326 010167 162436      MOV      R1,ERRNBR      ;SET UP ERROR NUMBER TO INITIAL ERRNBR.
3054 021332 004767 177440      JSR      PC,RDPDR      ;VERIFY ALL USED REGISTER BITS, ALL LINES.
3055 021336 004767 001050      JSR      PC,SWAPO      ;RESTORE MAIN GPRS CONTENTS.
3056      ;
3057      ; WRITE DATA PATTERNS, ALL LOWER BYTE USED BITS, ALL REGISTERS, ALL LINES.
3058      ; VERIFY THAT THE DATA PATTERN WAS WRITTEN CORRECTLY.
3059      ;
3060 021342 004767 001766      JSR      PC,WDPDR      ;WRITE DATA PATTERN TO DEVICE REGISTERS.
3061 021346 005267 162416      INC      ERRNBR      ;SET ERROR NUMBER TO INITIAL+1.
3062 021352 004767 177420      JSR      PC,RDPDR      ;VERIFY DATA PATTERN IN ALTERED BYTE(S).
3063 021356 005703      TST      R3      ;CHECK THE BYTE INDICATOR.
3064 021360 001411      BEQ      101      ;WORD ACCESS? YES, SKIP SECOND BYTE CHECK.
3065      ;
3066      ; CHECK THAT THE ALTERNATE (UNMODIFIED) BYTE IS CLEAR OR SET AS EXPECTED.
3067      ;
3068 021362 010201      MOV      R2,R1      ;SAVE THE DATA PATTERN.
3069 021364 010002      MOV      R0,R2      ;GET THE ALTERNATE BYTE EXPECTED DATA.
3070 021366 005403      NEG      R3      ;INDICATE THAT OTHER BYTE IS TO BE CHECKED.
3071 021370 005267 162374      INC      ERRNBR      ;SET ERROR NUMBER TO INITIAL+2.
3072 021374 004767 177376      JSR      PC,RDPDR      ;VERIFY DATA PATS IN OTHER BYTES OF REGISTERS.
3073 021400 005403      NEG      R3      ;RESTORE BYTE INDICATOR.
3074 021402 010102      MOV      R1,R2      ;RESTORE DATA PATTERN.
3075      ;
3076      ; PREPARE THE NEXT DATA PATTERN AND LOOP IF NOT DONE.
3077      ;
3078 021404 004767 000156      101: JSR      PC,ROLDAP      ;ROTATE DATA PATTERN LEFT, NOT THROUGH CARRY.
3079 021410 005305      DEC      R5      ;COUNT THIS ITERATION OF THE LOOP.
3080 021412 003341      BGT      81      ;ALL PATTERNS DONE? NO, LOOP.
3081      ;YES, RESTORE ERROR NUMBER AND EXIT.
3082 021414 016767 161030 162346 601: MOV      GPRS08,ERRNBR      ;GET THE ERROR NUMBR FROM GPR SWAP STORAGE.
3083 021422      PASS      ;RESTORE GPRS.
3084 021424 000207      RTS      PC      JSR      PC,@(SP)      ;RETURN TO PREG05 SUBRT.

```

GLOBAL SUBROUTINE

REPSMR

```

3086 .SBTTL GLOBAL SUBROUTINE REPSMR
3087 ;* .....
3088 ;* REPORT ERROR SUMMARY ROUTINE
3089 ;* THIS SUBROUTINE REPORTS AN ERROR SUMMARY FOR THOSE LINES WHICH HAVE
3090 ;* EXCEEDED THE NUMBER OF INDIVIDUAL ERRORS TO REPORT FOR A SINGLE LINE
3091 ;* IN A SINGLE TEST. THIS PARAMETER CAN BE SPECIFIED BY THE OPERATOR IF
3092 ;* HE/SHE ANSWERS THE SOFTWARE PARAMETER QUESTIONS.
3093 ;*
3094 ;* INPUTS: ERCNTB LABEL AT BASE OF LINE ERROR COUNTERS TABLE.
3095 ;* ERRMSG - ADDRESS OF PRIMARY ERROR MESSAGE.
3096 ;* ERRNBR ERROR NUMBER OF ERRORS IN THIS ROUTINE.
3097 ;* ERSMRF - "REPORT ERROR SUMMARY FOR LINE" FLAGS.
3098 ;*
3099 ;* OUTPUTS: ERRBLK ADDRESS OF ERROR REPORTING ROUTINE (DESTROYED).
3100 ;* SUMMARY MESSAGES MAY BE PRINTED AT THE OPERATOR CONSOLE.
3101 ;*
3102 ;* CALLING SEQUENCE: JSR PC,REPSMR
3103 ;*
3104 ;* COMMENTS: IF NO LINES HAVE EXCEEDED THE MAXIMUM NUMBER OF INDIVIDUAL
3105 ;* ERRORS TO REPORT, NO MESSAGES ARE PRINTED BY THIS ROUTINE.
3106 ;* ERROR SUMMARIES IN THIS ROUTINE ARE REPORTED AS ERRORS.
3107 ;* THE CONTENTS OF ERRBLK ARE DESTROYED.
3108 ;*
3109 ;* SUBORDINATE ROUTINES CALLED:
3110 ;* .....
3111
3112 021426 REPSMR:: SAVE ;SAVE CONTENTS OF GPRS R0 THRU R5.
3113 021426 004567 162344 ;R5,PREG05 ;CALL REGISTER SAVE SUBRT.
3114 021432 005767 161024 ;CHECK THE "PRINT LINE ERROR SUMMARY" FLAGS.
3115 021436 001404 ;EXIT WITHOUT ACTION IF NO SUMMARY FLAGS SET.
3116 ;*
3117 ;* WE HAVE SOME ERROR SUMMARIES TO REPORT.
3118 021440 012767 017156 162326 ; MOV 0ER9004,ERRBLK ;SELECT ERROR REPORTING ROUTINE.
3119 ;*
3120 ;* REPORT
3121 ;* "ERROR SUMMARY REPORT FOR LINES WITH EXCESSIVE NUMBERS OF ERRORS:"
3122 ;*
3123 021446 ERROR
3124 021446 104460 TRAP C$ERROR
3125 021450 60$: PASS ;RESTORE GPRS.
3126 021450 004736 ;PC,@(SP). ;RETURN TO PREG05 SUBRT.
3126 021452 000207 RTS PC

```

GLOBAL SUBROUTINE

RESETT -

```

3128 .SBTTL GLOBAL SUBROUTINE                      RESETT
3129 ;*****
3130 ;*          RESET DEVICE UNDER TEST
3131 ;*          THIS SUBROUTINE IS USED TO RESET THE DUT TO A KNOWN STATE.
3132 ;*          IF RESET DOES NOT SUCCESSFULLY COMPLETE, IE. TIME-OUT OCCURS, THEN
3133 ;*          AN ABORT TEST ERROR MESSAGE IS REPORTED.
3134 ;*
3135 ;* INPUTS:      CSRA   CONTAINS THE ADDRESS OF THE CSR
3136 ;*              TXBFCA - CONTAINS ADDRESS OF DUT DMA BUFFER COUNT REGISTER.
3137 ;*              ERRIBL- ERRITP,ERNBR,AND ERRMSG SET UP CORRECTLY.
3138 ;*
3139 ;* OUTPUTS:     THE DUT PERFORMS ITS RESET FUNCTION INTO A KNOWN STATE.
3140 ;*              CARRY - CLEAR INDICATES THE TEST IS TO BE ABORTED.
3141 ;*              ERRIK - VALUE MAY BE DESTROYED.
3142 ;*              IESTAT - TX AND RX INTERRUPT FLAGS ARE CLEARED.
3143 ;*              TX AND RX INTERRUPT ENABLE BITS IN THE DUT'S CSR ARE CLEARED.
3144 ;*
3145 ;* CALLING SEQUENCE:  JSR      PC,RESETT
3146 ;*
3147 ;* COMMENTS:      THIS SUBROUTINE CAN REPORT ERRORS WITH NUMBERS INITIAL ERNBR
3148 ;*                THIS ROUTINE DOES NOT DESTROY THE VALUE OF ERNBR.
3149 ;*
3150 ;* SUBORDINATE ROUTINES CALLED: DELAY,MSLGET.
3151 ;*****
3152
3153 021454 004567 162316  RESETT:: SAVE                ;SAVE CONTENTS OF GPRS R0 THRU R5.
3154 021454 012702 000040      MOV      #BIT05,R2      JSR      R5,PREGOS      ;CALL REGISTER SAVE SUBRT.
3155 ;*
3156 ;* TEST THE STATE OF THE MASTER RESET BIT IN THE CSR.
3157 ;* IF MR IS SET THEN WAIT FOR SELF-TEST TO COMPLETE.
3158 ;* IF TIME-OUT OCCURS, REPORT THE ERROR AND PASS-OUT ABORT TEST INDICATOR.
3159 ;
3160 021464 016704 160554      MOV      CSRA,R4          ;GET THE ADDRESS OF THE DUT'S CSR.
3161 021470 030214          BIT      R2,(R4)          ;CHECK STATE OF MASTER RESET BIT.
3162 021472 001406          BEQ      2$              ;DON'T DELAY IF MR IS ALREADY CLEAR.
3163 021474 005003          CLR      R3              ;SET UP DESIRED STATE OF MASTER RESET BIT.
3164 021476 012701 004704      MOV      #2500.,R1      ;PASS TIME-OUT VALUE OF 2.5 SECONDS.
3165 021502 004767 176630      JSR      PC,MSLGET      ;WAIT FOR SELF-TEST TO COMPLETE, MR CLEAR.
3166 021506 103012          BCC      4$              ;GO REPORT ERROR IF TIMEOUT OCCURRED.
3167
3168 ;*
3169 ;* SET MASTER RESET BIT IN CSR. CLEAR TX AND RX ENABLE BITS, ETC.
3170 ;* SKIP THE SELFTEST.
3171 ;* TIME-OUT OF 2.5 SECS. JUST IN CASE THE SELF-TEST EXECUTES.
3172 ;
3173 021510 010277 160530 2$:  MOV      R2,@CSRA          ;SET MASTER RESET BIT, DISABLE TX AND RX INTS.
3174 021514 004767 000614      JSR      PC,SKPSTS      ;TRY TO SKIP THE SELFTEST.
3175 ;*
3176 ;* SET SELF-TEST TIME-OUT OF 2.5 SECONDS, AND WAIT FOR M.R TO CLEAR.
3177 ;* IF TIME-OUT OCCURS, THEN REPORT THE FATAL ERROR AND PASS OUT THE ABORT
3178 ;* TEST INDICATOR.
3179 ;
3180 021520 005003          CLR      R3              ;SET UP DESIRED STATE OF MASTER RESET BIT.
3181 021522 012701 004704      MOV      #2500.,R1      ;PASS TIME-OUT VALUE OF 2.5 SECONDS.
3182 021526 004767 176604      JSR      PC,MSLGET      ;WAIT FOR SELF TEST TO COMPLETE, MR CLEAR.
3183 021532 103410          BCS      6$              ;SKIP ERROR REPORT IF MR CLEARED IN TIME.

```

GLOBAL SUBROUTINE

RESETT -

```

3184
3185 ;*
3186 ; SET UP ERROR MESSAGE TO REPORT "FATAL ERROR FOUND DURING RESET, TEST ABORTED".
3187 ; INDICATE TEST IS TO BE ABORTED BY CLEARING THE CARRY BIT.
3188 021534 012701 012360 4$: MOV #EM1601,R1 ;PASS ERROR MESSAGE TO REPORT.
3189 021540 012767 017006 162226 MOV #ER1603,ERRBLK ;PASS ADDRESS OF ERROR HANDLING ROUTINE.
3190 ;REPORT ERROR "TIME OUT OCCURRED WAITING FOR MASTER RESET TO CLEAR"
3191 ; "TEST ABORTED"
3192 021546 ERROR ; >>>>> ERROR <<<<<
3193 021550 000241 CLC ;INDICATE TEST IS TO BE ABORTED.
3194 021552 000403 BR 60$ ;EXIT THIS SUBROUTINE, ABORT TEST INDICATOR.
3195 ;*
3196 ; CLEAR TX AND RX INTERRUPT ENABLE STATUS FLAGS IN IESTAT.
3197 ; EXIT WITH CONTINUE TEST INDICATOR SET (IE,CARRY SET).
3198 ;
3199 021554 005067 160550 6$: CLR IESTAT ;CLEAR TX AND RX INTERRUPT STATUS FLAGS.
3200 021560 000261 SEC ;INDICATE SUCCESS, CONTINUE TEST.
3201
3202 021562 60$: PASS ;RESTORE GPRS, PASS THE FOLLOWING INTACT:
3203 021562 004736 JSR PC,@(SP); ;RETURN TO PREG05 SUBRT.
3204 021564 000207 RTS PC ;CARRY BIT: IF CLEAR, INDICATES ABORT TEST.
3205

```

GLOBAL SUBROUTINE.

ROLDAP

```
3207 .SBTTL GLOBAL SUBROUTINE ROLDAP
3208 ;.....
3209 ;* ROTATE LEFT DATA PATTERN
3210 ;* THIS ROUTINE ROTATES THE PASSED INPUT DATA PATTERN LEFT,WITHOUT GOING
3211 ;* THROUGH THE CARRY.THE CARRY IS INITIALLY SET OR CLEARED DEPENDING
3212 ;* UPON THE STATE OF THE MSB OF THE DATA PATTERN,BEFORE A ROL INSTRUCTION
3213 ;* IS EXECUTED.
3214 ;*
3215 ;* INPUTS: R2 CONTAINS THE DATA PATTERN TO BE ROTATED
3216 ;*
3217 ;* OUTPUTS: R2 CONTAINS THE ROTATED DATA PATTERN
3218 ;*
3219 ;* CALLING SEQUENCE: JSR PC,ROLDAP
3220 ;*
3221 ;* COMMENTS:
3222 ;*
3223 ;* SUBORDINATE ROUTINES CALLED: NONE
3224 ;.....
3225
3226 021566 ROLDAP::SAVE ;SAVE CONTENTS OF GPRS R0 THRU R5.
3227 021566 004567 162204 JSR R5,PREG05 ;CALL REGISTER SAVE SUBRT.
3228 021572 010202 ;SET PROCESSOR STATUS CODES
3229 021574 005702 ;CHECK MSB
3230 021576 100402 ;BRANCH IF SET
3231 021600 000241 ;CLEAR CARRY BIT IF MSB CLEAR
3232 021602 000401 ;
3233 021604 000261 2%: SEC ;SET CARRY IF MSB SET
3234 021606 006102 4%: ROL R2 ;ROTATE DATA PATTERN LEFT
3235 021610 010266 000006 60%: PASS R2 ;RESTORE GPRS,EXCEPT
3236 021614 004736 MOV R2,R2SLOT(SP) ;PUT R2 IN STACK SLOT.
3237 JSR PC,@(SP) ;RETURN TO PREG05 SUBRT.
3238 ;R2 CONTAINS THE ROTATED DATA PATTERN
3239 RTS PC
```

GLOBAL SUBROUTINE

RSTRPT -

```

3238 .SBTTL GLOBAL SUBROUTINE RSTRPT -
3239 ;... *****
3240 ;* - REPORT ANY RESET ERRORS ROUTINE -
3241 ;* THIS ROUTINE DETERMINES IF ANY ERROR CODES ARE AMONG THE DIAGNOSTIC
3242 ;* CODES REPORTED PLACED IN THE DUT RECEIVED CHARACTER FIFO BY THE
3243 ;* SELF TEST. IF ANY NON BMP ERROR CODES ARE FOUND, OR IF OTHER ERRORS
3244 ;* ARE ENCOUNTERED, APPROPRIATE ERRORS ARE REPORTED. ANY BMP CODES THAT
3245 ;* ARE FOUND, ARE PLACED ON THE BMP CODE QUEUE TO BE REPORTED LATER.
3246 ;* THIS ROUTINE ALSO PURGES THE DUT FIFO LOOKING FOR ANY CHARACTERS
3247 ;* OR MODEM STATUS CODES. IF ANY ARE FOUND, ERRORS ARE REPORTED.
3248 ;*
3249 ;* INPUTS: ERRMSG - ADDRESS OF THE PRIMARY ERROR MESSAGE.
3250 ;*          ERRNBR - ERROR NUMBER OF FIRST ERROR REPORTED BY THIS ROUTINE.
3251 ;*          NUMLNS - EQUATED TO THE NUMBER OF LINE ON THE DUT.
3252 ;*          RBUFA - CONTAINS ADDRESS OF THE DUT RECEIVER FIFO.
3253 ;*
3254 ;* OUTPUTS: CARRY - SUCCESS FLAG (SET IF FIFO CLEARED SUCCESSFULLY).
3255 ;*          ERRBLK - ADDRESS OF THE ERROR REPORT ROUTINE (DESTROYED).
3256 ;*          ERROR MESSAGES CAN BE PRINTED AT THE OPERATORS CONSOLE.
3257 ;*
3258 ;* CALLING SEQUENCE: JSR PC,RSTRPT
3259 ;*
3260 ;* COMMENTS: THIS SUBROUTINE CAN REPORT ERRORS WITH NUMBERS INITIAL ERRNBR
3261 ;*            THRU INITIAL ERRNBR+4.
3262 ;*            THIS ROUTINE DOES NOT DESTROY THE VALUE OF ERRNBR.
3263 ;*
3264 ;* SUBORDINATE ROUTINES CALLED: ER0503,ER9007,ER9008,SAVBMP.
3265 ;*-- *****
3266
3267 021620 RSTRPT:: SAVE ;SAVE CONTENTS OF GPRS R0 THRU R5.
3268 021620 004567 162152 JSR R5,PREG05 ;CALL REGISTER SAVE SUBRT.
3269 ;*
3270 ;* READ CORRECT NUMBER (NUMBER OF LINE ON DUT) OF CHARS FROM THE FIFO.
3271 ;* VERIFY THAT EACH CHAR IS A SELFTEST SUCCESS CODE.
3272 ;*
3273 021624 005003 CLR R3 ;CLEAR THE CODE COUNTER.
3274 021626 016705 162136 MOV ERRNBR,R5 ;SAVE ERRNBR FOR RESTORATION LATER.
3275 021632 017702 160410 MOV RBUFA,R2 ;READ A CHAR FROM THE DUT FIFO.
3276 021636 100412 BMI 4$ ;SKIP ERROR IF DATA.VALID SET FOR CHAR.
3277 ;*
3278 ;* WE EXPECT A SELFTEST CODE, BUT THIS FIFO SLOT IS EMPTY.
3279 ;*
3280 021640 010567 162124 MOV R5,ERRNBR ;RESTORE ERROR NUMBER TO INITIAL VALUE.
3281 021644 012701 015520 MOV #EM9018,R1 ;PASS ERROR MESSAGE INFO TO ER9007 ROUTINE.
3282 021650 012767 017260 162116 MOV #ER9007,ERRBLK ;SELECT PROPER ERROR REPORT ROUTINE.
3283 ;*
3284 ;* REPORT ERROR WITH NUMBER INITIAL ERRNBR.
3285 ;* "NO SELFTEST CODE IN SELFTEST CODE FIFO SLOT FOR LINE NN AFTER RESET."
3286 021656 ERROR ; >>>>> ERROR <<<<<.
3287 021656 104460 TRAP C$ERROR
3288 ;*
3289 ;* INDICATE "SUCCESS" (BECAUSE FIFO IS PURGED), AND EXIT THIS ROUTINE.
3290 ;*
3291 021660 000261 SEC ;SET SUCCESS FLAG.
3292 021662 000545 BR 60$ ;EXIT ROUTINE.

```

GLOBAL SUBROUTINE

RSTRPT

```

3293      ; DETERMINE IF THIS IS NOT A SELFTEST CODE.
3294      ;
3295 021664 012700 070001 4$: MOV #70001,R0 ;GENERATE BIT MAP OF ANY CLEAR ERROR BITS OR
3296 021670 040200 BIC R2,R0 ; BIT 0 WHICH ARE CLEAR.
3297 021672 001033 BNE 8$ ;GO TO REPORT ERROR IF THIS IS NOT A TEST CODE.
3298      ;
3299      ; WE HAVE A TEST CODE (EITHER BMP OR SELFTEST CODE).
3300      ; DETERMINE WHAT TYPE OF CODE WE HAVE.
3301      ;
3302 021674 032702 000200 BIT #BIT7,R2 ;TEST ROM VERSION CODE INDICATOR BIT.
3303 021700 001443 BEQ 10$ ;SKIP ERRORS IF SELFTEST ROM VERSION CODE.
3304 021702 120227 000203 CMPB R2,#203 ;CHECK IF SKIP SELF TEST CODE.
3305 021706 001440 BEQ 10$ ;SKIP ERROR REPORT IF SKIP SELF TEST CODE FOUND
3306 021710 120227 000201 CMPB R2,#201 ;CHECK IF NULL CODE PRESENT.
3307 021714 001435 BEQ 10$ ;SKIP ERROR REPORT IF SELF TEST NULL CODE.
3308 021716 012700 000300 MOV #300,R0 ;TEST CODE TYPE BITS FOR BOTH CODE
3309 021722 040200 BIC R2,R0 ; TYPE BITS SET (INDICATING BMP CODE).
3310 021724 001003 BNE 6$ ;IF IT IS NOT A BMP CODE GO REPORT ERROR.
3311 021726 004767 000334 JSR PC,SAVBMP ;SAVE THE BMP CODE ON THE QUEUE.
3312 021732 000426 BR 10$ ;GO GET THE NEXT CHARACTER FROM THE FIFO.
3313      ;
3314      ; WE HAVE A SELFTEST ERROR CODE.
3315      ;
3316 021734 010567 162030 6$: MOV R5,ERRNBR ;RESTORE ERROR NUMBER TO INITIAL VALUE.
3317 021740 005267 162024 INC ERRNBR ;CALCULATE INITIAL ERROR NUMBER PLUS 1.
3318 021744 012701 015545 MOV #EM9020,R1 ;PASS ERROR MESSAGE INFO TO ER9008 ROUTINE.
3319 021750 012767 017336 162016 MOV #ER9008,ERRBLK ;SELECT PROPER ERROR REPORT ROUTINE.
3320      ;
3321      ; REPORT ERROR WITH NUMBER INITIAL ERRNBR + 1.
3322      ; "UNEXPECTED SELFTEST ERROR CODE FOR LINE NN IN FIFO AFTER RESET:
3323      ;
3324 021756 104460 ERROR ; >>>>> ERROR <<<<<.
3325 021760 000413 BR 10$ ;GO TO END OF LOOP. TRAP C$ERROR
3326      ;
3327      ; WE HAVE A NON-SELFTEST CODE (EITHER BMP CODE OR DATA CHAR).
3328      ;
3329 021762 010567 162002 8$: MOV R5,ERRNBR ;RESTORE ERROR NUMBER TO INITIAL VALUE.
3330 021766 062767 000002 161774 ADD #2,ERRNBR ;CALCULATE INITIAL ERROR NUMBER PLUS 2.
3331 021774 012701 015530 MOV #EM9019,R1 ;PASS ERROR MESSAGE INFO TO ER9007 ROUTINE.
3332 022000 012767 017260 161766 MOV #ER9007,ERRBLK ;SELECT PROPER ERROR REPORT ROUTINE.
3333      ;
3334      ; REPORT ERROR WITH NUMBER INITIAL ERRNBR + 2.
3335      ; "NON-SELFTEST CODE IN SELFTEST CODE FIFO SLOT FOR LINE NN AFTER RESET.
3336      ;
3337 022006 104460 ERROR ; >>>>> ERROR <<<<<. TRAP C$ERROR
3338      ;
3339      ; END OF LOOP. LOOP IF NOT ALL CHARS HAVE BEEN READ FROM THE FIFO.
3340      ;
3341 022010 005203 10$: INC R3 ;SET CODE COUNTER FOR NEXT ITERATION OF LOOP.
3342 022012 020327 000010 CMP R3,#NUMLNS ;TEST FOR ALL CODES READ.
3343 022016 002705 BLT 2$ ;LOOP IF NOT CHARS READ FROM FIFO.
3344      ;
3345      ; PURGE THE FIFO UNTIL DATA.VALID IS CLEAR OR UNTIL TOO MANY CHARS ARE READ.
3346      ;
3347 022020 012704 000022 MOV #18.,R4 ;INITIALIZE THE CHARACTER COUNTER.

```

GLOBAL SUBROUTINE

RSTRPT .

```

3348 022024 010567 161740      MOV    R5,ERRNBR      ;GET INITIAL VALUE OF THE ERROR NUMBER.
3349 022030 062767 000003 161732  ADD    #3,ERRNBR      ;CALCULATE ERROR NUMBER OF NEXT ERROR.
3350 022036 012767 017336 161730  MOV    #ER9008,ERRBLK ;SELECT PROPER ERROR REPORT ROUTINE.
3351 022044 017702 160176      MOV    @RBUFA,R2      ;READ A CHARACTER FROM THE DUT FIFO.
3352 022050 000261      SEC                      ;INDICATE SUCCESS IN CASE DATA.VALID IS CLEAR.
3353 022052 100051      BPL     60$              ;EXIT ROUTINE WITH SUCCESS IF DATA.VALID CLEAR.
3354
3355      ;*
3356      ; WE HAVE A CHARACTER.
3357      ; DETERMINE IF CHARACTER IS A DATA CHARACTER.
3358 022054 012700 070000      MOV    #70000,R0      ;TEST BITS 12 THRU 14 OF THE
3359 022060 040200      BIC    R2,R0              ; CODE READ FROM THE DUT FIFO.
3360 022062 001403      BEQ    14$              ;SKIP THIS ERROR IF CODE IS NOT A DATA CHAR.
3361
3362      ;*
3363      ; WE HAVE AN UNEXPECTED DATA CHARACTER: SET UP AND GO TO REPORT ERROR.
3364 022064 012701 015571      MOV    #EM9022,R1      ;SELECT ERROR MSG INFO FOR ER0808 ROUTINE.
3365 022070 000423      BR     22$              ;GO TO REPORT THIS ERROR.
3366
3367      ;*
3368      ; WE HAVE AN UNEXPECTED CODE.
3369      ; DETERMINE IF THE CODE IS A MODEM STATUS CODE.
3370 022072 032702 000001 14$:  BIT    #BIT0,R2      ;TEST MODEM STATUS INDICATOR BIT OF CODE.
3371 022076 001003      BNE    16$              ;SKIP THIS ERROR IF NOT MODEM STATUS CODE.
3372
3373      ;*
3374      ; WE HAVE A MODEM STATUS CODE: SET UP AND GO TO REPORT ERROR.
3375 022100 012701 015610      MOV    #EM9023,R1      ;SELECT ERROR MSG INFO FOR ER0808 ROUTINE.
3376 022104 000415      BR     22$              ;GO TO REPORT THIS ERROR.
3377
3378      ;*
3379      ; WE HAVE AN ONBOARD TEST CODE.
3380      ; DETERMINE IF THIS CODE IS A BMP CODE.
3381 022106 032702 000200 16$:  BIT    #BIT7,R2      ;TEST THE ROM VERSION BIT OF THE CODE.
3382 022112 001404      BEQ    18$              ;GOTO SET UP FOR SELFTEST CODE IF ROM VERSION.
3383 022114 012700 000300      MOV    #300,R0              ;
3384 022120 040200      BIC    R2,R0              ;TEST THE ERROR TYPE BITS OF THE CODE.
3385 022122 001403      BEQ    20$              ;SKIP THIS ERROR IF BMP CODE.
3386
3387      ;*
3388      ; WE HAVE A SELFTEST CODE: SET UP AND GO TO REPORT ERROR.
3389 022124 012701 015632 18$:  MOV    #EM9024,R1      ;SELECT ERROR MSG INFO FOR ER0808 ROUTINE.
3390 022130 000403      BR     22$              ;GO TO REPORT THIS ERROR.
3391
3392      ;*
3393      ; WE HAVE A BMP CODE: SAVE IT ON THE QUEUE.
3394 022132 004767 000130 20$:  JSR    PC,SAVBMP      ;SAVE THE BMP CODE ON THE QUEUE.
3395 022136 000401      BR     24$              ;
3396
3397      ;*
3398      ; REPORT THE ERROR WITH ERROR NUMBER OF INITIAL ERRNBR + 3.
3399      ; 'UNEXPECTED XXX XXXX FOR LINE NN IN FIFO AFTER RESET;'
3400 022140      22$:  ERROR                      ;
3401 022140 104460      ;>>>> ERROR <<<<<.
3402      ; TRAP    C$ERROR
3403      ;*
3404      ; END OF LOOP.
3405      ; COUNT THE CHARACTER WE JUST RECEIVED, AND CHECK FOR TOO MANY RECEIVED.

```

GLOBAL SUBROUTINE

RSTRPT

```

3404
3405 022142 005304      24$: DEC R4      ;COUNT THIS CHARACTER.
3406 022144 001337      BNE 12$      ;LOOP IF NOT TOO MANY CHARACTERS PURGED.
3407
3408      ; WE READ TOO MANY VALID CHARACTERS WHILE TRYING TO PURGE THE FIFO.
3409      ; REPORT ERROR AND EXIT WITHOUT SUCCESS.
3410      ; "FIFO WILL NOT PURGE (DATA.VALID STUCK SET), REMAINDER OF TEST SKIPPED."
3411
3412 022146 012701 015407      MOV #EM9017,R1      ;SELECT PROPER ERROR MESSAGE.
3413 022152 010567 161612      MOV R5,ERRNBR      ;GET INITIAL ERROR NUMBER.
3414 022156 062767 000004 161604      ADD #4,ERRNBR      ;CALCULATE INITIAL ERRNBR + 4.
3415 022164 012767 016466 161602      MOV #ER0503,ERRBLK      ;SELECT PROPER ERROR REPORT ROUTINE.
3416      ;PRINT ERROR REPORT.
3417 022172      ERROR      ;
3418 022172 104460      ;>>>> ERROR <<<<<.
3419 022174 000241      TRAP C$ERROR
3420 022176      CLC      ;CLEAR THE SUCCESS FLAG.
3421 022200 004736      60$: PASS      ;RESTORE GPRS,
      PC      JSR      PC,@(SP)+      ;RETURN TO PREG05 SUBRT.
      RTS      ; CARRY - SUCCESS FLAG (SET IF FIFO IS PURGED).

```

GLOBAL SUBROUTINE

RXIEO

```

3423 .SBTTL GLOBAL SUBROUTINE - RXIEO
3424 ;* *****
3425 ;* - RECEIVER INTERRUPT DISABLE -
3426 ;* THIS ROUTINE IS USED TO DISABLE RECEIVER INTERRUPTS IN THE DHV11.
3427 ;*
3428 ;* INPUTS: NONE.
3429 ;*
3430 ;* OUTPUTS: THE RX.INT.ENBL BIT IS CLEARED IN THE DUT CSR.
3431 ;* IESTST CONTAINS THE UPDATED STATUS OF THE TX AND RX INTERRUPT
3432 ;* ENABLE BITS.
3433 ;*
3434 ;* CALLING SEQUENCE: JSR PC,RXIEO
3435 ;*
3436 ;* COMMENTS: THE CONTENTS OF THE INDIRECT ADDRESS REGISTER FIELD IN
3437 ;* THE DUT CSR ARE DESTROYED.
3438 ;*
3439 ;* SUBORDINATE ROUTINES CALLED: NONE.
3440 ;* *****
3441 022202 010046 RXIEO:: MOV RO,-(SP) ;SAVE CONTENTS OF RO ON THE STACK.
3442 022204 104440 GETPRI -(SP) ;SAVE PROCESSOR PRIORITY ON STACK.
3443 022204 104440 TRAP C$GPRI
3443 022206 010046 MOV RO,-(SP)
3443 022210 SETPRI #PRI07 ;IGNORE ANY INTERRUPT THAT MAY BE GENERATED.
3443 022210 012700 000340 MOV #PRI07,RO
3443 022214 104441 TRAP C$SPRI
3444 022216 042767 137777 160104 BIC #137777,IESTAT ;CLEAR RX.INT.ENBL BIT IN IESTAT.
3445 022224 016777 160100 160012 MOV IESTAT,@CSRA ;DISABLE RX INTERRUPTS.
3446 022232 SETPRI (SP)+ ;ENABLE INTERRUPTS TO THE PROCESSOR AGAIN.
3446 022232 012600 MOV (SP)+,RO
3446 022234 104441 TRAP C$SPRI
3447 022236 012600 MOV (SP)+,RO
3448 022240 000207 RTS PC ;RESTORE RO.

```

GLOBAL SUBROUTINE

RXIE1

```

3450      .SBTTL GLOBAL SUBROUTINE                      RXIE1
3451      ; .. *****
3452      ; *      . RECEIVER INTERRUPT ENABLE .
3453      ; *      THIS ROUTINE IS USED TO ENABLE RECEIVER INTERRUPTS IN THE DMV11.
3454      ; *
3455      ; * INPUTS:      NONE.
3456      ; *
3457      ; * OUTPUTS:     THE RX.INT.ENBL BIT IS SET IN THE DUT CSR.
3458      ; *               IESTAT -CONTAINS THE UPDATED STATUS OF THE TX AND RX INTERRUPT
3459      ; *               ENABLE BITS.
3460      ; *
3461      ; * CALLING SEQUENCE:  JSR      PC,RXIE1
3462      ; *
3463      ; * COMMENTS:     THE CONTENTS OF THE INDIRECT ADDRESS REGISTER FIELD IN
3464      ; *               THE DUT CSR ARE DESTROYED.
3465      ; *
3466      ; * SUBORDINATE ROUTINES CALLED: NONE.
3467      ; - *****
3468
3469 022242 052767 000100 160060 RXIE1:: BIS      @BIT06,IESTAT ;SET RX.INT.ENBL BIT IN IESTAT.
3470 022250 042767 137677 160052      BIC      @137677,IESTAT ;CLEAR ALL OTHER BITS, EXCEPT TX AND RX I.E.
3471 022256 016777 160046 157760      MOV      IESTAT,@CSRA  ;ENABLE RX INTERRUPTS.
3472 022264 000207      RTS      PC

```

GLOBAL SUBROUTINE

SAVBMP -

```

3474 .SBTTL GLOBAL SUBROUTINE SAVBMP
3475 :.. .....
3476 :* SAVE BMP CODES ROUTINE
3477 :* THIS ROUTINE SAVES THE PARAMETER PASSFD IN, ONTO THE BMP CODE QUEUE
3478 :* TOGETHER WITH THE NUMBER OF THE CURRENTLY EXECUTING TEST.
3479 :
3480 :* INPUTS: R2 - CONTAINS THE BMP CODE THAT IS TO BE PLACED ON THE QUEUE.
3481 :* BMPCQP - CONTAINS ADDRESS OF NEXT LOCATION IN THE BMP QUEUE.
3482 :* BMPCQB - LABEL AT BASE OF THE BMP CODE QUEUE.
3483 :* BMPCQE - LABEL OF NEXT LOCATION AFTER THE END OF THE BMP QUEUE.
3484 :* TSTNUM - CONTAINS THE NUMBER OF THE CURRENT TEST.
3485 :
3486 :* OUTPUTS: BMPCQP - INCREMENTED BY 4.
3487 :* THE CONTENTS OF THE BMP CODE QUEUE ARE UPDATED.
3488 :
3489 :* CALLING SEQUENCE: JSR PC,SAVBMP
3490 :
3491 :* COMMENTS: IF THE OVERFLOW OCCURS THEN THE LAST LOCATION WILL BE
3492 :* OVERWRITTEN BY ANY SUBSEQUENT ATTEMPTS TO UPDATE THE QUEUE.
3493 :
3494 :* SUBORDINATE ROUTINES CALLED: NONE.
3495 : .....
3496 :
3497 022266 SAVBMP:: SAVE ;SAVE CONTENTS OF GPRS R0 THRU R5.
3498 022266 004567 161504 ;R5,PREG05 ;CALL REGISTER SAVE SUBRT.
3499 022272 016704 160226 ;GET THE POINTER TO THE NEXT LOCATION IN QUEUE.
3500 022302 005204 160024 ;SAVE THE CURRENT TEST NUMBER ON THE QUEUE.
3501 022304 042702 177400 ;INCREMENT THE POINTER TO GIVE AN EVEN ADDRESS.
3502 022310 010224 ;CLEAR THE UNWANTED BITS FROM THE BMP CODE.
3503 022312 020427 002726 ;SAVE THE BMP CODE ON THE QUEUE.
3504 022316 103402 ;CHECK IF OVERFLOW WILL OCCUR THE NEXT TIME.
3505 022320 162704 000004 ;GO SAVE THE POINTER IF WE WILL NOT OVERFLOW.
3506 022324 010467 160174 ;RESET THE POINTER TO THE LAST LOCATION IN QJE.
3507 ;SAVE THE POINTER.
3508 022330 ;RESTORE GPRS.
3509 022330 004736 ;RETURN TO PREG05 SUBRT.
3509 022332 000207 RTS PC JSR PC,@(SP).

```

GLOBAL SUBROUTINE

SKPSTS

```

3511 .SBTTL GLOBAL SUBROUTINE SKPSTS
3512 ;* .....
3513 ;* SKIP SELFTEST ROUTINE
3514 ;* THIS SUBROUTINE IS USED TO SKIP THE SELFTEST AFTER A DUT RESET HAS BEEN
3515 ;* INITIATED. IT MUST BE ENTERED IMMEDIATELY AFTER SETTING THE DUT MASTER
3516 ;* RESET ROUTINE OR AFTER THE EXECUTION OF A BUS RESET (BECAUSE OF TIMING
3517 ;* CONSIDERATIONS).
3518 ;*
3519 ;* INPUTS: CSRA CONTAINS ADDRESS OF THE DUT CSR.
3520 ;* TXBFCA CONTAINS ADDRESS OF DUT DMA BUFFER COUNT REGISTER.
3521 ;*
3522 ;* OUTPUTS: SKIP SELFTEST CODES ARE WRITTEN TO THE DUT REGISTERS.
3523 ;*
3524 ;* CALLING SEQUENCE: JSR PC,SKPSTS
3525 ;*
3526 ;* COMMENTS:
3527 ;*
3528 ;* SUBORDINATE ROUTINES CALLED: DELAY.
3529 ;* .....
3530
3531 SKPSTS:: SAVE ;SAVE CONTENTS OF GPRS R0 THRU R5.
3532 022334 004567 161436 JSR R5,PREG05 ;CALL REGISTER SAVE SUBRT.
3533 022340 012704 000012 MOV #10,R4 ;PASS DELAY VALUE OF 10 MILLI-SECONDS.
3534 022344 004767 175726 JSR PC,DELAY ;DELAY FOR 10 MILLI-SECONDS.
3535 ;*
3536 ;* WRITE SKIP SELF-TEST CODE (52525) TO ALL THE INDEXED DUT REGISTERS.
3537 ;*
3538 ;* MOV #NUMLNS!BIT05,R1 ;FORM IND.ADR.REG FIELD (PLUS M.R. BIT) WORD.
3539 ;* THE ABOVE INCLUSION OF THE M.R. BIT IS NECESSARY BECAUSE OF THE
3540 ;* LACK OF A M.R. BIT WRITE LOCK-OUT ON THE DMV-11.
3541 022354 012703 052525 MOV #52525,R3 ;INITIALISE THE SKIP SELF-TEST CODE.
3542 022360 005301 4$: DEC R1 ;SELECT THE NEXT SET OF DEVICE REGISTERS.
3543 022362 016704 157656 MOV CSRA,R4 ;GET THE ADDRESS OF THE CSR OF THE DUT.
3544 022366 010124 MOV R1,(R4)+ ;SELECT A BANK OF DUT REGISTERS.
3545 022370 010324 6$: MOV R3,(R4)+ ;WRITE THE CODE TO A DUT REGISTER.
3546 022372 020467 157664 CMP R4,TXBFCA ;COMPARE POINTER WITH LAST REGISTER ADDRESS.
3547 022376 103774 BLO 6$ ;LOOP IF NOT ALL REGS DONE IN THIS BANK.
3548 022400 032701 000017 BIT #17,R1 ;TEST FOR IND.ADR.REG FIELD DECREMENTED TO 0.
3549 022404 001365 BNE 4$ ;LOOP UNTIL ALL REGISTERS CONTAIN THE CODE.
3550 022406 004736 60$: PASS ;RESTORE GPRS.
3551 022410 000207 JSR PC,@(SP)+ ;RETURN TO PREG05 SUBRT.

```

GLOBAL SUBROUTINE

SWAPO

```

3553 .SBTTL GLOBAL SUBROUTINE SWAPO
3554 ;* .....
3555 ;* SWAP GPRS WITH GPR SET 0 ROUTINE
3556 ;* THIS SUBROUTINE SWAPS THE PRESENT CONTENTS OF GPRS R1 THRU R5 WITH
3557 ;* THE CONTENTS OF THE NUMBER ZERO GPR SAVE AREA. THE CONTENTS OF R0
3558 ;* ARE NOT ALTERED BY THIS SUBROUTINE.
3559 ;*
3560 ;* INPUTS: GPR CONTENTS R1 THRU R5.
3561 ;* GPRS08 LABEL AT BASE OF GPR SAVE AREA NUMBER ZERO.
3562 ;*
3563 ;* OUTPUTS: R1 THRU R5 CONTAIN THE PREVIOUS CONTENTS OF GPR SAVE AREA
3564 ;* ZERO WORDS 1 THRU 5 RESPECTIVELY.
3565 ;* GPRS0 GPR SAVE AREA 0 WORDS 1 THRU 5, CONTAIN PREVIOUS
3566 ;* CONTENTS OF GPRS R1 THRU R5 RESPECTIVELY.
3567 ;*
3568 ;* CALLING SEQUENCE: JSR PC,SWAPO
3569 ;*
3570 ;* COMMENTS: THE STATE OF THE CARRY FLAG IS NOT ALTERED BY THIS ROUTINE.
3571 ;*
3572 ;* SUBORDINATE ROUTINES CALLED: NONE.
3573 ;* .....
3574
3575 022412 010046 SWAPO:: MOV R0, (SP) ;SAVE THE CONTENTS OF R0.
3576 ;*
3577 ; LOAD THE STACK FROM THE GPRS.
3578 ;*
3579 022414 010146 MOV R1, -(SP) ;SAVE THE CONTENTS OF R1.
3580 022416 010246 MOV R2, -(SP) ;SAVE THE CONTENTS OF R2.
3581 022420 010346 MOV R3, -(SP) ;SAVE THE CONTENTS OF R3.
3582 022422 010446 MOV R4, -(SP) ;SAVE THE CONTENTS OF R4.
3583 022424 010546 MOV R5, (SP) ;SAVE THE CONTENTS OF R5.
3584 ;*
3585 ; LOAD THE GPRS FROM THE GPR SAVE AREA 0.
3586 ;*
3587 022426 012700 002450 MOV #GPRS08,R0 ;GET THE BASE ADDRESS OF GPR SAVE AREA 0.
3588 022432 012001 MOV (R0)+,R1 ;LOAD R1 WITH GPR SAVE AREA 0 WORD 1.
3589 022434 012002 MOV (R0)+,R2 ;LOAD R1 WITH GPR SAVE AREA 0 WORD 2.
3590 022436 012003 MOV (R0)+,R3 ;LOAD R1 WITH GPR SAVE AREA 0 WORD 3.
3591 022440 012004 MOV (R0)+,R4 ;LOAD R1 WITH GPR SAVE AREA 0 WORD 4.
3592 022442 012005 MOV (R0)+,R5 ;LOAD R1 WITH GPR SAVE AREA 0 WORD 5.
3593 ;*
3594 ; LOAD THE GPR SAVE AREA 0 FROM THE STACK.
3595 ;*
3596 022444 012640 MOV (SP)+, -(R0) ;LOAD GPR SAVE AREA 0 WORD 5 WITH SAVED R5.
3597 022446 012640 MOV (SP)+, -(R0) ;LOAD GPR SAVE AREA 0 WORD 4 WITH SAVED R4.
3598 022450 012640 MOV (SP)+, -(R0) ;LOAD GPR SAVE AREA 0 WORD 3 WITH SAVED R3.
3599 022452 012640 MOV (SP)+, -(R0) ;LOAD GPR SAVE AREA 0 WORD 2 WITH SAVED R2.
3600 022454 012640 MOV (SP)+, -(R0) ;LOAD GPR SAVE AREA 0 WORD 1 WITH SAVED R1.
3601
3602 022456 012600 MOV (SP)+,R0 ;RESTORE THE INITIAL VALUE OF R0.
3603
3604 022460 000207 RTS PC

```

GLOBAL SUBROUTINE

TSABRT -

```

3606 .SBTTL GLOBAL SUBROUTINE TSABRT -
3607 ;... *****
3608 ;* - TEST ABORT ROUTINE
3609 ;* THIS SUBROUTINE IS USED WHEN A NON-TEST RELATED ERROR HAS BEEN FOUND
3610 ;* DURING THE EXECUTION OF THE CURRENT TEST.
3611 ;* IT IS USED TO INFORM THE OPERATOR THAT THE CURRENT TEST HAS BEEN
3612 ;* ABORTED.
3613 ;*
3614 ;* INPUTS: ERRMSG CONTAINS THE NAME OF THE CURRENT TEST.
3615 ;*          ERRNR - CONTAINS THE CORRECT ERROR NUMBER.
3616 ;*          THE REMAINDER OF THE ERRBL IS CORRECTLY INITIALISED.
3617 ;*
3618 ;* OUTPUTS: MESSAGES ARE REPORTED TO THE OPERATOR.
3619 ;*
3620 ;* CALLING SEQUENCE: JSR PC,TSABRT
3621 ;*
3622 ;* COMMENTS:
3623 ;*
3624 ;* SUBORDINATE ROUTINES CALLED: ER1603.
3625 ;... *****
3626
3627 022462 TSABRT:: SAVE ;SAVE CONTENTS OF GPRS R0 THRU R5.
3628 022462 004567 161310 JSR R5,PREG05 ;CALL REGISTER SAVE SUBRT.
3629 022466 012701 022504 MOV #2%,R1 ;PASS ADDRESS OF FIRST MESSAGE TO BE REPORTED.
3630 022472 012767 017006 161274 MOV #ER1603,ERRBLK ;SET UP THE ERROR REPORTING ROUTINE.
3631 022500 104460 ;>>>> ERROR <<<<.
3632 022502 000432 TRAP C$ERRCR
3633 022504 040 116 117 2%: BR 60% ;
3634 022507 116 055 122 .ASCIZ / NON RELATED TEST ERROR FOUND DURING TEST EXECUTION/
3635 022512 105 114 101
3636 022515 124 105 104
3637 022520 040 124 105
3638 022523 123 124 040
3639 022526 105 122 122
3640 022531 117 122 040
3641 022534 106 117 125
3642 022537 116 104 040
3643 022542 104 125 122
3644 022545 111 116 107
3645 022550 040 124 105
3646 022553 123 124 040
3647 022556 105 130 105
3648 022561 103 125 124
3649 022564 111 117 116
3650 022567 000
3651 .EVEN
3652 60%: PASS
3653 RTS PC JSR ;RESTORE GPRS.
3654 022570 004736 PC,0(SP). ;RETURN TO PREG05 SUBRT.
3655 022572 000207

```

GLOBAL SUBROUTINE

TXDSBL -

```

3637 .SBTTL GLOBAL SUBROUTINE - TXDSBL
3638 ;* *****
3639 ;* TRANSMITTER DISABLE -
3640 ;* THIS SUBROUTINE IS USED TO DISABLE TRANSMISSION ON SELECTED LINES BY,
3641 ;* CLEARING THE ASSOCIATED TX.ENABLE BIT ON THE OUT.
3642 ;*
3643 ;* INPUTS: R5 BIT'S SET CORRESPOND TO LINES ON WHICH TO CLEAR TX.ENABLE.
3644 ;* CSRA - CONTAINS THE ADDRESS OF THE OUT CSR.
3645 ;* IESTAT - CONTAINS THE STATE OF TXIE AND RXIE BITS IN THE CSR.
3646 ;* NUMLNS - EQUATED TO BE THE MAXIMUM NUMBER OF LINES AVAILABLE.
3647 ;* TXAD2A - CONTAINS THE ADDRESS OF THE TBUFFAD2 REGISTER.
3648 ;*
3649 ;* OUTPUTS: R5 - BIT'S SET INDICATE THE INITIAL STATES OF ALL TX.ENABLE BITS.
3650 ;* TBUFFAD2 - THE STATE OF THE TX.ENABLE BIT MAY BE ALTERED.
3651 ;* THE CONTENTS OF THE IND.ADD.REG FIELD IN THE CSR ARE DESTROYED.
3652 ;*
3653 ;* CALLING SEQUENCE: JSR PC,TXDSBL
3654 ;*
3655 ;* COMMENTS:
3656 ;*
3657 ;* SUBORDINATE ROUTINES CALLED: NONE.
3658 ;* *****
3659
3660 022574 TXDSBL:: SAVE ;SAVE CONTENTS OF GPRS R0 THRU R5.
022574 004567 161176 JSR R5,PREG05 ;CALL REGISTER SAVE SUBRT.
3661 022600 010500 MOV R5,R0 ;COPY BIT MAP OF LINES TO DISABLE TRANSMISSION.
3662 022602 012701 000001 MOV #BIT0,R1 ;INITIALIZE THE SELECTED LINE BIT MASK.
3663 022606 016702 157446 MOV TXAD2A,R2 ;GET THE ADDRESS OF THE TBUFFAD2 REGISTER.
3664 022612 005202 INC R2 ;GET THE ADDRESS OF THE MSBYTE OF TBUFFAD2 REG.
3665 022614 012703 000010 MOV #NUMLNS,R3 ;GET MAXIMUM LINE NUMBER PLUS ONE.
3666 022620 016704 157504 MOV IESTAT,R4 ;GET THE STATES OF THE INT ENABLE BITS.
3667 022624 005005 CLR R5 ;LOG POSSIBLE TX DISABLED ON ALL LINES.
3668 ;*
3669 ;* SELECT EVERY LINE IN TURN, AND LOG THE STATE OF EACH TX.ENABLE BIT.
3670 ;*
3671 022626 010477 157412 2$: MOV R4,@CSRA ;WRITE TO OUT CSR TO SELECT LINE REGISTERS.
3672 022632 105712 TSTB (R2) ;CHECK STATE OF TX.ENABLE BIT ON SELECTED LINE.
3673 022634 100001 BPL 4$ ;SKIP NEXT INSTRUCTION IF TX.ENABLE CLEAR.
3674 022636 050105 BIS R1,R5 ;LOG TX ENABLE BIT SET FOR SELECTED LINE.
3675 ;*
3676 ;* CLEAR TX.ENABLE ON LINES THAT HAVE A CORRESPONDING BIT SET IN THE TX DISABLE
3677 ;* LINE BIT MAP.
3678 ;*
3679 022640 030100 4$: BIT R1,R0 ;CHECK STATE OF DISABLE LINE BIT MAP.
3680 022642 001402 BEQ 6$ ;BRANCH IF THIS LINE TO REMAIN UNALTERED.
3681 022644 142712 000200 BICB #BIT7,(R2) ;CLEAR TX.ENABLE BIT ON SELECTED LINE.
3682 022650 005204 6$: INC R4 ;PREPARE TO SELECT REGISTERS FOR NEXT LINE.
3683 022652 006301 ASL R1 ;SHIFT BIT MAP FOR NEXT LINE.
3684 022654 005303 DEC R3 ;DECREMENT LINE NUMBER.
3685 022656 001363 BNE 2$ ;LOOP TO CHECK NEXT LINE.
3686 ;*
3687 022660 60$: PASS R5 ;RESTORE GPRS,EXCEPT
022660 010566 000014 MOV R5,R5SLOT(SP) ;PUT R5 IN STACK SLOT.
022664 004736 JSR PC,@(SP) ;RETURN TO PREG05 SUBRT.
3688 ;R5 PREVIOUS STATES OF ALL TX.ENABLE BITS.
3689 022666 000207 RTS PC

```

GLOBAL SUBROUTINE

TXENBL -

```

3691 .SBTTL GLOBAL SUBROUTINE TXENBL
3692 ;... *****
3693 ;* TRANSMITTER ENABLE
3694 ;* THIS SUBROUTINE IS USED TO ENABLE TRANSMISSION ON SELECTED LINES BY
3695 ;* SETTING THE ASSOCIATED TX.ENABLE BIT ON THE OUT.
3696 ;*
3697 ;* INPUTS: R5 BIT'S SET CORRESPOND TO LINES ON WHICH TO SET TX.ENABLE.
3698 ;* CSRA CONTAINS THE ADDRESS OF THE OUT CSR.
3699 ;* IESTAT - CONTAINS THE STATE OF TXIE AND RXIE BITS IN THE CSR.
3700 ;* NUMLNS - EQUATED TO BE THE MAXIMUM NUMBER OF LINES AVAILABLE.
3701 ;* TXAD2A - CONTAINS THE ADDRESS OF THE TBUFFAD2 REGISTER.
3702 ;*
3703 ;* OUTPUTS: R5 BIT'S SET INDICATE PREVIOUSLY DISABLED LINES.
3704 ;* TBUFFAD2 - THE STATE OF THE TX.ENABLE BIT MAY BE ALTERED.
3705 ;* THE CONTENTS OF THE IND.ADD.REG FIELD IN THE CSR ARE DESTROYED.
3706 ;*
3707 ;* CALLING SEQUENCE: JSR PC,TXENBL
3708 ;*
3709 ;* COMMENTS:
3710 ;*
3711 ;* SUBORDINATE ROUTINES CALLED: NONE.
3712 ; - *****
3713
3714 022670 TXENBL:: SAVE ;SAVE CONTENTS OF GPRS R0 THRU R5.
3715 022670 004567 161102 JSR R5,PREG05 ;CALL REGISTER SAVE SUBRT.
3716 022674 010500 ;COPY BIT MAP OF LINES TO ENABLE.
3717 022676 012701 000001 MOV #BIT0,R1 ;INITIALIZE THE SELECTED LINE BIT MASK.
3718 022702 016702 157352 MOV TXAD2A,R2 ;GET THE ADDRESS OF THE TBUFFAD2 REGISTER.
3719 022706 005202 INC R2 ;GET THE ADDRESS OF THE MSBYTE OF TBUFFAD2 REG.
3720 022710 012703 000010 MOV #NUMLNS,R3 ;GET MAXIMUM LINE NUMBER.
3721 022714 016704 157410 MOV IESTAT,R4 ;GET THE STATES OF THE INT ENABLE BITS.
3722 022720 005005 CLR R5 ;CLEAR TX.ENABLE BIT LOG OF DISABLED LINES.
3723 ;*
3724 ;* SELECT EVERY LINE IN TURN,AND LOG ANY TX.ENABLE BIT THAT IS CLEAR.
3725 022722 010477 157316 2$: MOV R4,BCSRA ;WRITE TO OUT CSR TO SELECT LINE REGISTERS.
3726 022726 105712 TSTB (R2) ;CHECK STATE OF TX.ENABLE BIT ON SELECTED LINE.
3727 022730 100401 BMI 4$ ;SKIP NEXT INSTRUCTION IF TX.ENABLE SET.
3728 022732 050105 BIS R1,R5 ;LOG TX ENABLE BIT CLEAR FOR SELECTED LINE.
3729 ;*
3730 ;* SET TX.ENABLE ON LINES THAT HAVE A CORRESPONDING BIT SET IN THE TX ENABLE
3731 ;* LINE BIT MAP.
3732 ;*
3733 022734 030100 4$: BIT R1,R0 ;CHECK STATE OF TX.ENABLE LINE BIT MAP.
3734 022736 001402 BEQ 6$ ;BRANCH IF THIS LINE TO REMAIN UNALTERED.
3735 022740 152712 000200 BISB #BIT7,(R2) ;ENABLE TRANSMISSION ON SELECTED LINE.
3736 022744 005204 6$: INC R4 ;PREPARE TO SELECT REGISTERS FOR NEXT LINE.
3737 022746 006301 ASL R1 ;SHIFT BIT MAP FOR NEXT LINE.
3738 022750 005303 DEC R3 ;DECREMENT LINE NUMBER.
3739 022752 001363 BNE 2$ ;LOOP TO CHECK NEXT LINE.
3740
3741 022754 60$: PASS R5 ;RESTORE GPRS,EXCEPT
3742 022754 010566 000014 MOV R5,R5SLOT(SP) ;PUT R5 IN STACK SLOT.
3743 022760 004736 JSR PC,B(SP) ;RETURN TO PREG05 SUBRT.
3744 022762 000207 RTS PC ;R5 LINE BIT MAP CORRESPONDING TO THE
; PREVIOUS LINES THAT WERE DISABLED.

```

GLOBAL SUBROUTINE

TXIEO

```

3746 .SBTTL GLOBAL SUBROUTINE TXIEO -
3747 :... *****
3748 :* TRANSMITTER INTERRUPT DISABLE -
3749 :* THIS ROUTINE IS USED TO DISABLE TRANSMITTER INTERRUPTS IN THE DHV11.
3750 :*
3751 :* INPUTS: NONE.
3752 :*
3753 :* OUTPUTS: THE TX.INT.ENBL BIT IS CLEARED IN THE DUT CSR.
3754 :* IESTAT CONTAINS THE UPDATED STATUS OF THE TX AND RX INTERRUPT
3755 :* ENABLE BITS.
3756 :*
3757 :* CALLING SEQUENCE: JSR PC,TXIEO
3758 :*
3759 :* COMMENTS: THE CONTENTS OF THE INDIRECT ADDRESS REGISTER FIELD IN
3760 :* THE DUT CSR ARE DESTROYED.
3761 :*
3762 :* SUBORDINATE ROUTINES CALLED: NONE.
3763 :*****
3764 022764 010046 TXIEO:: MOV RO,(SP) ;SAVE CONTENTS OF RO ON THE STACK.
3765 022766 104440 GETPRI (SP) ;SAVE CURRENT PROCESSOR PRIORITY ON THE STACK.
3766 022770 010046 TRAP C$GPRI
3766 022772 012700 000340 SETPRI $PRI07 ;IGNORE ANY INTERRUPTS THAT MAY BE GENERATED.
3766 022776 104441 MOV RO,-(SP)
3767 023000 042767 177677 157322 BIC $177677,IESTAT ;CLEAR TX.INT.ENBL BIT IN IESTAT.
3768 023006 016777 157316 157230 MOV IESTAT,$CSRA ;DISABLE TX INTERRUPTS.
3769 023014 012600 SETPRI (SP)+ ;ENABLE INTERRUPTS TO THE PROCESSOR AGAIN.
3769 023016 104441 TRAP C$SPRI
3770 023020 012600 MOV (SP)+,RO
3771 023022 000207 RTS PC ;RESTORE RO.

```

GLOBAL SUBROUTINE.

TXIE1

```

3773 .SBTTL GLOBAL SUBROUTINE TXIE1 -
3774 ;.. *****
3775 ;* - TRANSMITTER INTERRUPT ENABLE
3776 ;* THIS ROUTINE IS USED TO ENABLE TRANSMITTER INTERRUPTS IN THE DHV11.
3777 ;*
3778 ;* INPUTS: NONE.
3779 ;*
3780 ;* OUTPUTS: THE TX.INT.ENBL BIT IS SET IN THE DUT CSR.
3781 ;* IESTST -CONTAINS THE UPDATED STATUS OF THE TX AND RX INTERRUPT
3782 ;* ENABLE BITS.
3783 ;*
3784 ;* CALLING SEQUENCE: JSR PC,TXIE1
3785 ;*
3786 ;* COMMENTS: THE CONTENTS OF THE INDIRECT ADDRESS REGISTER FIELD IN
3787 ;* THE DUT CSR ARE DESTROYED.
3788 ;*
3789 ;* SUBORDINATE ROUTINES CALLED: NONE.
3790 ; - *****
3791
3792 023024 052767 040000 157276 TXIE1:: BIS #BIT14,IESTAT ;SET TX.INT.ENBL BIT IN IESTAT.
3793 023032 042767 137677 157270 BIC #137677,IESTAT ;CLEAR ALL BITS EXCEPT TX RX I.E BITS.
3794 023040 016777 157264 157176 MOV IESTAT,@CSRA ;DISABLE TX INTERRUPTS.
3795 023046 000207 RTS PC

```

GLOBAL SUBROUTINE

UNSDIV -

```

3797 .SBTTL GLOBAL SUBROUTINE UNSDIV
3798 ;... *****
3799 ;* - UNSIGNED DIVIDE ROUTINE -
3800 ;* THIS SUBROUTINE IS USED TO DIVIDE A 32 BIT UNSIGNED DIVIDEND BY A
3801 ;* 16 BIT UNSIGNED DIVISOR GIVING A 16 BIT QUOTIENT. ALL NUMBERS ARE
3802 ;* CONSIDERED TO BE UNSIGNED. A SUCCESS FLAG IS NOT SET ON RETURN IF
3803 ;* THE QUOTIENT WAS TOO BIG TO BE CONTAINED IN 16 BITS.
3804 ;*
3805 ;* INPUTS: R1 - THE DIVISOR, UNSIGNED, 16 BITS.
3806 ;* R2 MOST SIGNIFICANT WORD OF THE DIVIDEND, UNSIGNED, 16 BITS.
3807 ;* R3 LEAST SIGNIFICANT WORD OF THE DIVIDEND, UNSIGNED, 16 BITS.
3808 ;*
3809 ;* OUTPUTS: R1 - QUOTIENT, UNSIGNED, 16 BITS (177777 IF OVERFLOW).
3810 ;* CARRY - SUCCESS FLAG, SET IF COMPLETE QUOTIENT FITS IN 16 BITS.
3811 ;*
3812 ;* CALLING SEQUENCE: JSR PC,UNSDIV
3813 ;*
3814 ;* COMMENTS: IF THE DIVISOR IS 0 THE QUOTIENT IS RETURNED AS ALL ONES
3815 ;* (177777) AND THE CARRY IS CLEAR REGARDLESS OF THE DIVIDEND.
3816 ;*
3817 ;* SUBORDINATE ROUTINES CALLED: NONE.
3818 ;* - *****
3819
3820 023050 UNSDIV:: SAVE ;SAVE CONTENTS OF GPRS R0 THRU R5.
3821 023050 004567 160722 JSR R5,PREG05 ;CALL REGISTER SAVE SUBRT.
3822
3823 ; CHECK FOR QUOTIENT GREATER THAN 16 BITS CONDITION.
3824
3824 023054 010204 MOV R2,R4 ;GET MSW OF DIVIDEND FOR SUBTRACT.
3825 023056 160104 SUB R1,R4 ;SUBTRACT DIVISOR FROM MSW OF DIVIDEND.
3826 023060 103403 BCS 2$ ;IF IT DIDN'T GO, WE HAVE QUOTIENT < 16 BITS.
3827 023062 012701 177777 MOV #1,R1 ;SET QUOTIENT TO ALL ONES (177777).
3828 023066 000442 BR 60$ ;EXIT WITH CARRY CLEAR.
3829
3830 ; SET UP COUNTERS AND VARIOUS WORKING GPRS.
3831
3832 023070 005004 2$: CLR R4 ;CLEAR THE LSW OF THE DIVISOR.
3833 023072 000241 CLC ;CLEAR CARRY FOR THE SHIFT OF THE DIVISOR.
3834 023074 006001 ROR R1 ;DIVISOR BY
3835 023076 006004 ROR R4 ;2(UNSIGNED)
3836 023100 012700 000020 MOV #16,R0 ;SET UP INITIAL SHIFT COUNT TO 16.
3837
3838 ; THE SUBTRACT AND SHIFT LOOP.
3839
3840 023104 010246 4$: MOV R2,(SP) ;SAVE MSWORD OF DIVIDEND.
3841 023106 010346 MOV R3,-(SP) ;SAVE LSWORD OF DIVIDEND.
3842 023110 160403 SUB R4,R3 ;LSWORD DIVIDEND - LSWORD OF DIVISOR.
3843 023112 005602 SBC R2 ;MSWORD DIVIDEND - BORROW.
3844 023114 103402 BCS 6$ ;IF BORROW FROM BORROW SUBTRACT, IT DIDN'T GO.
3845 023116 160102 SUB R1,R2 ;MSWORD DIVIDEND - MSWORD OF DIVISOR.
3846 023120 103003 BCC 8$ ;IF NO BORROW, IT WENT, CARRY IS CLEAR.
3847
3848 ; IT DIDN'T GO, SO WE SHIFT A 1 INTO THE QUOTIENT (COMPLEMENTED LATER).
3849 ; CARRY IS SET.
3850
3851 023122 012603 6$: MOV (SP)+,R3 ;RESTORE LSWORD OF DIVIDEND.
3852 023124 012602 MOV (SP)+,R2 ;RESTORE MSWORD OF DIVIDEND.

```

GLOBAL SUBROUTINE

UNSDIV

```

3853 023126 000401      BR      10$      ;GOTO SHIFT 1 INTO THE QUOTIENT.
3854
3855      ;*
3856      ; IT WENT, SO WE RESTORE THE STACK AND SHIFT A 0 INTO QUOTIENT (WILL BE
3857      ; COMPLEMENTED LATER).  CARRY IS CLEAR.
3858 023130 012626      8$:  MOV      (SP)+,(SP)+      ;POP THE SAVED DIVIDEND OFF OF THE STACK.
3859
3860      ;*
3861      ; SHIFT THE RESULT OF THE SUBTRACT ATTEMPT INTO THE QUOTIENT SHIFT REG.
3862 023132 006105      10$:  ROL      R5      ;SHIFT NEXT BIT INTO THE INVERTED QUOTIENT.
3863 023134 000241      CLC      ;DIVIDE THE
3864 023136 006001      ROR      R1      ; DIVISOR BY
3865 023140 006004      ROR      R4      ; 2 (UNSIGNED).
3866 023142 005300      DEC      R0      ;COUNT THIS SHIFT AND SUBTRACT.
3867 023144 001357      BNE      4$      ;LOOP FOR ANOTHER SHIFT & SUB IF NOT DONE.
3868 023146 005105      COM      R5      ;GET QUOTIENT FROM INVERTED QUOTIENT.
3869
3870      ;*
3871      ; NOW WE EITHER ROUND UP OR LEAVE QUOTIENT ALONE.
3872 023150 000241      CLC      ;CLEAR THE CARRY FOR THE SHIFT OF THE DIVIDEND.
3873 023152 006103      ROL      R3      ;MULTIPLY LSWORD OF DIVIDEND BY 2, MSWORD IS 0.
3874 023154 103402      BCS      12$      ;IF CARRY FROM SHIFT, ROUND UP.
3875 023156 160403      SUB      R4,R3      ;SUBTRACT DIVISOR FROM DIVIDEND.
3876 023160 103403      BCS      14$      ;IF BORROW, DON'T ROUND UP.
3877
3878      ;*
3879      ; ROUND UP, EXTRA SUBTRACT WENT.
3880 023162 005205      12$:  INC      R5      ;INCREMENT THE QUOTIENT BY ONE.
3881 023164 001001      BNE      14$      ;IF NO OVERFLOW, WE LEAVE THE ROUND UP.
3882 023166 005305      DEC      R5      ;DON'T LET ROUNDING CAUSE OVERFLOW.
3883
3884      ;*
3885      ; ALL DONE, PASS QUOTIENT AND EXIT.
3886 023170 010501      14$:  MOV      R5,R1      ;PASS QUOTIENT BACK IN R1.
3887 023172 000261      SEC      ;INDICATE NO OVERFLOW.
3888
3889 023174      60$:  PASS      R1      ;RESTORE GPRS, LEAVE THE FOLLOWING INTACT:
      023174 010166 000004      MOV      R1,R1SLOT(SP)      ;PUT R1 IN STACK SLOT.
      023200 004736      JSR      PC,@(SP)+      ;RETURN TO PREGOS SUBRT.
3890
3891 023202 000207      RTS      PC      ;R1 - 16 BIT, UNSIGNED QUOTIENT.
      ;CARRY - SLT INDICATES NO OVERFLOW (SUCCESS).

```

GLOBAL SUBROUTINE

WAIBIC -

```

3893 .SBTTL GLOBAL SUBROUTINE - WAIBIC
3894 :.. *****
3895 :* - WAIT FOR BIT CLEAR ROUTINE -
3896 :* THIS SUBROUTINE WAITS FOR THE SPECIFIED BIT TO BECOME CLEAR. IF THE
3897 :* SPECIFIED BIT GOES TO A CLEAR STATE WITHIN THE SPECIFIED TIME-OUT
3898 :* PERIOD A SUCCESS INDICATION IS RETURNED BY THIS ROUTINE.
3899 :* THE LAST VALUE WHICH IS READ LOOKING FOR THE CONDITION IS RETURNED TO
3900 :* ALLOW THE USE OF THIS ROUTINE TO LOOK FOR DESTRUCTIVE READ CONDITIONS.
3901 :*
3902 :* INPUTS: R1 TIME-OUT VALUE AND BIT NUMBER INDICATION:
3903 :*          BITS 15 THRU 12 - NUMBER OF BIT TO TEST (RANGE 0 THRU 15).
3904 :*          BITS 11 THRU 0 - TIME-OUT VALUE IN MILLI SECONDS (4095 MAX).
3905 :*          R2 - ADDRESS OF WORD CONTAINING THE BIT TO TEST.
3906 :*          MSLCNT.
3907 :*
3908 :* OUTPUTS: R2 - THE LAST WORD WHICH WAS READ TO CHECK FOR THE CONDITION.
3909 :*          CARRY - SUCCESS FLAG (CARRY SET IF BIT CLR BEFORE TIME OUT).
3910 :*
3911 :* CALLING SEQUENCE: MOV #130040,R1 ;PASS BIT 11 (13 OCTAL) AND
3912 :*                   ; 32 (40 OCTAL) MS DELAY.
3913 :*                   MOV #LABEL,R2 ;TEST BIT IN WORD AT "LABEL".
3914 :*                   JSR PC,WAIBIC ;WAIT 32 MS FOR BIT 11 TO CLR.
3915 :*
3916 :* COMMENTS:
3917 :*
3918 :* SUBORDINATE ROUTINES CALLED: MSLGET.
3919 :.. *****
3920
3921 023204 WAIBIC:: SAVE ;SAVE CONTENTS OF GPRS R0 THRU R5.
3922 023204 004567 160566 JSR R5,PREG05 ;CALL REGISTER SAVE SUBRT.
3923 023210 010204 MOV R2,R4 ;SET UP THE ADDRESS PARAMETER FOR MSLGET.
3924 023212 010102 MOV R1,R2
3925 023214 042701 170000 BIC #170000,R1 ;SEPERATE DELAY COUNT OUT OF PASSED PARAMETER.
3926 023220 042702 007777 BIC #7777,R2 ;SEPERATE LINE NUMBER FIELD OF PASSED PARAM.
3927 023224 000302 SWAB R2 ;PUT LINE NUMBER FIELD IN LS BYTE.
3928 023226 006202 ASR R2 ;SHIFT THE LINE NUMBER FIELD INTO THE PROPER
3929 023232 006202 ASR R2 ; POSITION TO USE IT AS A WORD TABLE OFFSET
3930 023234 016202 002410 MOV BITTBL(R2),R2 ; FOR THE TABLE LOOKUP OF THE LINE BIT MAP.
3931 023240 005003 CLR R3 ;GET BIT MAP OF LINE TO TEST FROM TABLE.
3932 023242 004767 175070 JSR PC,MSLGET ;INDICATE THAT THE BIT SHOULD BE CLR.
3933 :* ;WAIT FOR THE BIT TO BE CLR WITHIN TIME-OUT.
3934 023246 010002 MOV R0,R2 ; CARRY IS CORRECT UPON MSLGET RETURN.
3935 023250 010266 000006 60$: PASS R2 ;PASS LAST VALUE READ AS OUTPUT PARAMETER.
3936 023254 004736 MOV R2,R2SLOT(SP) ;RESTORE GPRS, EXCEPT THE FOLLOWING:
3937 023256 000207 JSR PC, @ (SP)+ ;PUT R2 IN STACK SLOT.
; RETURN TO PREG05 SUBRT.
; R2 - LAST VALUE READ LOOKING FOR CONDITION.
; CARRY - SUCCESS FLAG (SET IF BIT FOUND CLR).

```

GLOBAL SUBROUTINE

WAIBIS -

```

3939 .SBTTL GLOBAL SUBROUTINE - WAIBIS -
3940 ;* *****
3941 ;* WAIT FOR BIT SET ROUTINE
3942 ;* THIS SUBROUTINE WAITS FOR THE SPECIFIED BIT TO BECOME SET. IF THE
3943 ;* SPECIFIED BIT GOES TO A SET STATE WITHIN THE SPECIFIED TIME OUT
3944 ;* PERIOD A SUCCESS INDICATION IS RETURNED BY THIS ROUTINE.
3945 ;* THE LAST VALUE WHICH IS READ LOOKING FOR THE CONDITION IS RETURNED TO
3946 ;* ALLOW THE USE OF THIS ROUTINE TO LOOK FOR DESTRUCTIVE READ CONDITIONS.
3947 ;*
3948 ;* INPUTS: R1 TIME-OUT VALUE AND BIT NUMBER INDICATION:
3949 ;* BITS 15 THRU 12 - NUMBER OF BIT TO TEST (RANGE 0 THRU 15).
3950 ;* BITS 11 THRU 0 - TIME OUT VALUE IN MILLI-SECONDS (4095 MAX).
3951 ;* R2 - ADDRESS OF WORD CONTAINING THE BIT TO TEST.
3952 ;* MSLCNT.
3953 ;*
3954 ;* OUTPUTS: R2 - THE LAST WORD WHICH WAS READ TO CHECK FOR THE CONDITION.
3955 ;* CARRY - SUCCESS FLAG (CARRY SET IF BIT SET BEFORE TIME OUT).
3956 ;*
3957 ;* CALLING SEQUENCE: MOV #130040,R1 ;PASS BIT 11 (13 OCTAL) AND
3958 ;* ; 32 (40 OCTAL) MS DELAY.
3959 ;* MOV #LABEL,R2 ;TEST BIT IN WORD AT "LABEL".
3960 ;* JSR PC,WAIBIS ;WAIT 32 MS FOR BIT 11 TO SET.
3961 ;*
3962 ;* COMMENTS:
3963 ;*
3964 ;* SUBORDINATE ROUTINES CALLED: MSLGET.
3965 ;* - - *****
3966
3967 WAIBIS:: SAVE ;SAVE CONTENTS OF GPRS R0 THRU R5.
3968 023260 004567 160512 JSR R5,PREG05 ;CALL REGISTER SAVE SUBRT.
3969 023264 010204 MOV R2,R4 ;SET UP THE ADDRESS PARAMETER FOR MSLGET.
3970 023266 010102 MOV R1,R2
3971 023270 042701 170000 BIC #170000,R1 ;SEPERATE DELAY COUNT OUT OF PASSED PARAMETER.
3972 023274 042702 007777 BIC #7777,R2 ;SEPERATE LINE NUMBER FIELD OF PASSED PARAM.
3973 023300 000302 SWAB R2 ;PUT LINE NUMBER FIELD IN LSBYTE.
3974 023302 006202 ASR R2 ;SHIFT THE LINE NUMBER FIELD INTO THE PROPER
3975 023304 006202 ASR R2 ; POSITION TO USE IT AS A WORD TABLE OFFSET
3976 023306 006202 ASR R2 ; FOR THE TABLE LOOKUP OF THE LINE BIT MAP.
3977 023310 016202 002410 MOV BITTBL(R2),R2 ;GET BIT MAP OF LINE TO TEST FROM TABLE.
3978 023314 010203 MOV R2,R3 ;INDICATE THAT THE BIT SHOULD BE SET.
3979 023316 004767 175014 JSR PC,MSLGET ;WAIT FOR THE BIT TO BE SET WITHIN TIME-OUT.
3980 023322 010002 MOV R0,R2 ; CARRY IS CORRECT UPON MSLGET RETURN.
3981 023324 010266 000006 60$: PASS R2 ;PASS LAST VALUE READ AS OUTPUT PARAMETER.
3982 023330 004736 MOV R2,R2SLOT(SP) ;RESTORE GPRS, EXCEPT THE FOLLOWING:
3983 023332 000207 JSR PC, @ (SP)+ ;FUT R2 IN STACK SLOT.
; RETURN TO PREG05 SUBRT.
; R2 - LAST VALUE READ LOOKING FOR CONDITION.
; CARRY - SUCCESS FLAG (SET IF BIT FOUND SET).

```

GLOBAL SUBROUTINE

WOPDR

```

3985 .SBTTL GLOBAL SUBROUTINE WOPDR
3986 ;* .....
3987 ;* - WRITE DATA PATTERN TO DEVICE REGISTERS
3988 ;* THIS ROUTINE WRITES A ROTATED DATA PATTERN TO EACH OF THE 6 DEVICE
3989 ;* REGISTERS OF EACH ACTIVE LINE OF THE DEVICE UNDER TEST.
3990 ;* THE DATA PATTERN IS ROTATED ONCE AFTER EACH WRITE TO A DEVICE REGISTER
3991 ;* ON A PARTICULAR LINE. THE STARTING DATA PATTERN FOR EACH LINE
3992 ;* IS ROTATED ONCE AFTER WRITING ALL THE REGISTERS ON A PARTICULAR
3993 ;* LINE. THIS LEADS TO THE FOLLOWING DATA PATTERN:
3994 ;* LINE 0, REGISTER 0 - SHIFTED 0 BIT POSITIONS
3995 ;* LINE 0, REGISTER 1 - SHIFTED 1 BIT POSITION
3996 ;*
3997 ;* LINE 1, REGISTER 0 - SHIFTED 1 BIT POSITION
3998 ;* LINE 2, REGISTER 1 - SHIFTED 2 BIT POSITIONS
3999 ;*
4000 ;* ANY BITS FIELDS IN THE DEVICE REGISTERS THAT CANNOT BE ALTERED
4001 ;* ARE MASKED OUT OF THE DATA PATTERN BEFORE IT IS WRITTEN.
4002 ;* THIS ROUTINE WILL USE EITHER MOV, MOVB, BIS, BISB, BIC, OR BICB
4003 ;* INSTRUCTIONS. THE UPPER OR LOWER BYTE CAN BE SPECIFIED FOR WRITING.
4004 ;*
4005 ;* INPUTS: R2 - USED TO PASS IN THE DATA PATTERN TO BE ROTATED & WRITTEN.
4006 ;* R3 - BYTE INDICATOR (- >> LO BYTE, . >> HI BYTE, 0 >> BOTH).
4007 ;* R4 - OPERATION TYPE INDICATOR (- >> BIC, . >> BIS, 0 >> MOV).
4008 ;* ACTLNS - BIT MAP OF THE ACTIVE LINES ON THE DEVICE UNDER TEST.
4009 ;* CSRA - CONTAINS THE CSR ADDRESS OF THE DEVICE UNDER TEST.
4010 ;* DRADRT - BASE ADDRESS OF DEVICE REGISTER ADDRESS TABLE.
4011 ;* LPRO - EQUATED TO LPR REG OFFSET FROM DEVICE CSR ADDRESS.
4012 ;* NUMLNS - NUMBER OF LINES ON THE DEVICE UNDER TEST.
4013 ;* TXBFCO - EQUATED TO TBUFFT REG OFFSET FROM DEVICE CSR ADDRESS.
4014 ;* UNBITB - BASE ADDRESS OF THE UNUSED BIT TABLE.
4015 ;*
4016 ;* OUTPUTS: DEVICE REGISTERS ON ALL ACTIVE DEVICE LINES ARE MODIFIED.
4017 ;*
4018 ;* CALLING SEQUENCE: JSR PC,WOPDR
4019 ;*
4020 ;* COMMENTS: THIS ROUTINE DOES NOT WRITE ANY DATA TO THE TX.CHAR REGISTERS.
4021 ;* THE CSR IS CLEARED EXCEPT FOR THE IND.ADR.REG FIELD.
4022 ;*
4023 ;* SUBORDINATE ROUTINES CALLED: ROLDAP.
4024 ;* .....
4025
4026 023334 WOPDR:: SAVE ;SAVE CONTENTS OF GPRS R0 THRU R5.
4027 023334 004567 160436 JSR R5,PREGOS ;CALL REGISTER SAVE SUBRT.
4028 ;*
4029 ;* SET UP OUTER LOOP WHICH WRITES THE DATA PATTERN TO EACH LINE'S REGISTERS
4030 023340 005005 CLR R5 ;CLEAR LINE COUNTER TO SELECT LINE 0.
4031 ;*
4032 ;* THE OUTER LOOP FOLLOWS. EACH PASS THROUGH THIS LOOP WRITES DATA TO ALL OF
4033 ;* THE DEVICE REGISTERS FOR A PARTICULAR LINE IF IT IS ACTIVE.
4034 ;*
4035 023342 010204 2$: MOV R2,R4 ;SAVE THE OUTER LOOP DATA PATTERN.
4036 023344 010577 156674 MOV R5,@CSRA ;SET CSR IND.ADR.REG FIELD TO THIS LINE.
4037 023350 006305 ASL R5 ;TURN LINE NUMBER INTO A WORD OFFSET.
4038 023352 036567 002410 156656 BIT BITTBL(R5),ACTLNS
4039 023360 001451 BEQ 20$ ;LINE ACTIVE? NO, SKIP THIS LINE.
4040 023362 012701 000004 MOV @LPRO,R1 ;YES, INITIALIZE THE REGISTER OFFSET.

```

GLOBAL SUBROUTINE

WDPDR

```

4041      ;*
4042      ; THE INNER LOOP FOLLOWS.  EACH PASS THROUGH THIS LOOP WRITES DATA TO A
4043      ; DEVICE REGISTER.
4044      ;
4045      4045 023366 010200      4045:  MOV    R2,R0
4046 023370 046100 002264      BIC    UNBITB(R1),R0      ;CLEAR BIT FIELDS FOR UNUSED REGISTER BITS.
4047 023374 016103 002244      MOV    DRADRT(R1),R3      ;GET THE ADDRESS OF THE DEVICE REGISTER.
4048 023400 005766 000010      TST    R3SLOT(SP)      ;CHECK THE OPERAND TYPE INDICATOR.
4049 023404 003402      BLE    60:      ;HIGH BYTE? NO, SKIP HIGH BYTE ADDRESS SET UP.
4050 023406 005203      INC    R3      ;YES, SET THE REG ADDRESS TO THE HIGH BYTE.
4051 023410 000300      SWAB   R0      ;MOVE HIGH BYTE DATA INTO THE LOW BYTE.
4052 023412 005766 000010      60:  TST    R3SLOT(SP)      ;CHECK THE OPERAND TYPE INDICATOR.
4053 023416 001412      BEQ    120:      ;WORD ACCESS? YES, GO PERFORM WORD ACCESS.
4054      ;*
4055      ;PERFORM BYTE ACCESS TO THE SPECIFIED BYTE OF THE SPECIFIED REGISTER.
4056      ;
4057 023420 005766 000012      TST    R4SLOT(SP)      ;NO, CHECK THE ACCESS TYPE INDICATOR.
4058 023424 100403      BMI    80:      ;USE BIC? YES, GO PERFORM BICB INSTRUCTION.
4059 023426 001404      BEQ    100:      ;USE MOV? YES, GO PERFORM MOVVB INSTRUCTION.
4060 023430 150013      BISB   R0,(R3)      ;NEITHER. PERFORM BISB ACCESS TO REGISTER.
4061 023432 000415      BR     180:
4062 023434 140013      80:  BICB   R0,(R3)      ;PERFORM BICB ACCESS TO REGISTER.
4063 023436 000413      BR     180:
4064 023440 110013      100:  MOVVB  R0,(R3)      ;PERFORM MOVVB ACCESS TO REGISTER.
4065 023442 000411      BR     180:
4066      ;*
4067      ;PERFORM WORD ACCESS TO THE SPECIFIED REGISTER.
4068      ;
4069 023444 005766 000012      120:  TST    R4SLOT(SP)      ;CHECK THE ACCESS TYPE INDICATOR.
4070 023450 100403      BMI    140:      ;USE BIC? YES, GO PERFORM BIC INSTRUCTION.
4071 023452 001404      BEQ    160:      ;USE MOV? YES, GO PERFORM MOV INSTRUCTION.
4072 023454 050013      BIS    R0,(R3)      ;NEITHER. PERFORM BIS ACCESS TO REGISTER.
4073 023456 000403      BR     180:
4074 023460 040013      140:  BIC    R0,(R3)      ;PERFORM BIC ACCESS TO REGISTER.
4075 023462 000401      BR     180:
4076 023464 010013      160:  MOV    R0,(R3)      ;PERFORM MOV ACCESS TO REGISTER.
4077      ;*
4078      ; PREPARE THE DATA PATTERN AND OFFSET FOR THE NEXT REGISTER ON THIS LINE.
4079      ;
4080 023466 004767 176074      180:  JSR    PC,ROLDAP      ;ROTATE DATA PATTERN LEFT, NOT THROUGH CARRY.
4081 023472 062701 000002      ADD    02,R1      ;INCREMENT OFFSET FOR NEXT REGISTER.
4082 023476 020127 000016      CMP    R1,0TXBFCO      ;COMPARE REG OFFSET WITH OFFSET OF LAST REG.
4083 023502 003731      BLE    40:      ;LOOP IF NOT ALL REG DONE FOR THIS LINE.
4084      ;*
4085      ; BACK INTO THE OUTER LOOP.  NOW SET UP FOR NEXT LINE.  LOOP IF NOT DONE.
4086      ;
4087 023504 010402      200:  MOV    R4,P2      ;SET UP TO ROTATE THE DATA PATTERN.
4088 023506 004767 176054      JSR    PC,ROLDAP      ;ROTATE THE DATA PATTERN.
4089 023512 006205      ASR    R5      ;CONVERT BACK TO LINE NUMBER FROM WORD OFFSET.
4090 023514 005205      INC    R5      ;COUNT THIS LINE.
4091 023516 020527 000010      CMP    R5,0NUMLNS      ;COMPARE LINE COUNT WITH NUMBER OF LINES.
4092 023522 002707      BLT    20:      ;LOOP IF SOME LINES NOT DONE.
4093      ;*
4094 023524      600:  PASS      ;RESTORE GPRS.
4095 023524 004736      JSR    PC,0(SP).      ;RETURN TO PREGC5 SUBR.
4095 023526 000207      RTS    PC

```

GLOBAL SUBROUTINE

WTWLNC

```

4097 .SBITL GLOBAL SUBROUTINE                                WTWLNC
4098 ;* *****
4099 ;* - LINE CONTROL REGISTER SETUP ROUTINE
4100 ;* THIS SUBROUTINE IS USED TO SET THE DEVICE UNDER TEST (DUT) LINE
4101 ;* CONTROL REGISTERS (LNCTRL) TO THE SPECIFIED STATE. ONLY THE LNCTRLS
4102 ;* FOR THE SPECIFIED LINES ARE ALTERED
4103 ;*
4104 ;* INPUTS:      R0 - NEW LINE PARAMETERS.
4105 ;*              R5 - BIT MAP OF LINES TO BE ALTERED.
4106 ;*              CSRA - CONTAINS ADDRESS OF THE DUT CSR.
4107 ;*              IESTAT - CONTAINS THE CURRENT STATE OF THE TX AND RX INTERRUPT
4108 ;*              ENABLE BITS IN THE CSR.
4109 ;*              LNCTRA - CONTAINS ADDRESS OF THE DUT LNCTRL REGISTERS.
4110 ;*
4111 ;* OUTPUTS:     LNCTRL - SPECIFIED DUT LINE CONTROL REGISTERS ARE ALTERED.
4112 ;*
4113 ;* CALLING SEQUENCE:  JSR      PC,WTWLNC
4114 ;*
4115 ;* COMMENTS:
4116 ;*
4117 ;* SUBORDINATE ROUTINES CALLED:  ALTFD.
4118 ;* *****
4119
4120 023530 004567 160242 WTWLNC.: SAVE                ;SAVE CONTENTS OF GPRS R0 THRU R5.
                                JSR      R5,PREG05      ;CALL REGISTER SAVE SUBRT.
4121
4122 ; SET UP THE PARAMETERS FOR THE CALL TO ALTFD.
4123 ;
4124 023534 016701 156514      MOV      LNCTRA,R1      ;SET UP THE REGISTER ADDRESS PARAMETER.
4125 023540 010002      MOV      R0,R2      ;SET UP THE DESIRED REGISTER CONTENTS.
4126 023542 010503      MOV      R5,R3      ;SET UP THE BIT MAP OF LINES TO ALTER.
4127 023544 012704 177777      MOV      #-1,R4      ;SELECT ALL REGISTER BITS TO BE ALTERED.
4128
4129 ; CALL THE SUBROUTINE WHICH ALTERS THE REGISTER CONTENTS.
4130 ;
4131 023550 004767 174052      JSR      PC,ALTFD      ;ALTER THE REGISTER CONTENTS.
4132
4133 023554 004736 604: PASS                ;RESTORE GPRS.
                                JSR      PC,0(SP).    ;RETURN TO PREG05 SUBRT.
4134 023556 000207      RTS      PC

```

GLOBAL SUBROUTINE

WTWLN5

```

4136 .SBTTL GLOBAL SUBROUTINE WTWLN5
4137 ;*****
4138 ;* - WRITE WORD TO ALL LINES ROUTINE -
4139 ;* THIS SUBROUTINE WRITES A SPECIFIED WORD TO THE SPECIFIED DMV DEVICE
4140 ;* REGISTER FOR EACH OF THE DMV LINES. IT COULD BE USED TO CLEAR ALL
4141 ;* OF THE LNCTRL REGISTERS OR TO INITIALIZE ALL OF THE LPR REGISTERS TO
4142 ;* THE SAME PARAMETERS.
4143 ;*
4144 ;* INPUTS: R1 - ADDRESS OF THE SPECIFIED REGISTERS.
4145 ;* R2 - WORD TO WRITE INTO THE SPECIFIED REGISTERS.
4146 ;* IESTAT - SAVED STATES OF THE TX.IE AND RX.IE BITS.
4147 ;* MAPLNS - EQUATED TO BIT MAP OF LINES ON DEVICE (8 FOR DMV11).
4148 ;* CSRA.
4149 ;*
4150 ;* OUTPUTS: DEVICE REGISTERS - SPECIFIED REGISTERS GIVEN NEW VALUE.
4151 ;* CSR IND.ADR.REG FIELD - DESTROYED.
4152 ;* CSR INTERRUPT ENABLE BITS - SET TO STATES IN IESTAT.
4153 ;*
4154 ;* CALLING SEQUENCE: JSR PC,WTWLN5
4155 ;*
4156 ;* COMMENTS: NOTE THAT THE SPECIFIED REGISTERS FOR ALL LINES ARE ALTERED
4157 ;* BY THIS ROUTINE. THIS ROUTINE SHOULD NOT BE USED TO ALTER
4158 ;* THE STATES OF PARTIAL REGISTER FIELDS OR TO ALTER A REGISTER
4159 ;* FOR FEWER THAN ALL OF THE LINES.
4160 ;* THE SPECIFIED REGISTERS ARE READ BEFORE BEING WRITTEN.
4161 ;*
4162 ;* SUBROUTINES CALLED: ALTFLD.
4163 ;*****
4164
4165 023560 WTWLN5:: SAVE ;SAVE CONTENTS OF GPRS R0 THRU R5
023560 004567 160212 JSR R5,PREG05 ;CALL REGISTER SAVE SUBRT.
4166
4167 ; SET UP THE BIT MAP OF LINES TO CHANGE AND MASK OF BITS TO ALTER PARAMETERS.
4168 ;
4169 023564 012703 000377 MOV #MAPLNS,R3 ;GET THE BIT MAP OF LINES TO CHANGE.
4170 023570 012704 177777 MOV #1,R4 ;INDICATE ALL 16 BITS TO BE CHANGED.
4171
4172 ; CALL THE SUBROUTINE TO WRITE THE SPECIFIED REGISTERS.
4173 ;
4174 023574 004767 174026 JSR PC,ALTFLD ;CHANGE THE REGISTERS.
4175
4176 023600 601: PASS ;RESTORE GPRS.
023600 004736 JSR PC,8(SP). ;RETURN TO PREG05 SUBRT.
4177 023602 000207 RTS PC

```

GLOBAL SUBROUTINE.

WTWLPR -

```

4179 .SBTTL GLOBAL SUBROUTINE - WTWLPR -
4180 ;* *****
4181 ;* - LINE PARAMETER REGISTER SETUP ROUTINE -
4182 ;* THIS SUBROUTINE IS USED TO SET THE DEVICE UNDER TEST (DUT) LINE
4183 ;* PARAMETER REGISTERS (LPR) TO THE SPECIFIED STATE. ONLY THE LPRS FOR
4184 ;* THE SPECIFIED LINES ARE ALTERED.
4185 ;*
4186 ;* INPUTS: RO - NEW LINE PARAMETERS.
4187 ;* RS - BIT MAP OF LINES TO BE ALTERED.
4188 ;* CSRA - CONTAINS ADDRESS OF THE DUT CSR.
4189 ;* IESTAT - CONTAINS THE CURRENT STATE OF THE TX AND RX INTERRUPT
4190 ;* ENABLE BITS IN THE CSR.
4191 ;* LPRA - CONTAINS ADDRESS OF THE DUT LPR.
4192 ;*
4193 ;* OUTPUTS: LPR SPECIFIED DUT LINE PARAMETER REGISTERS ARE ALTERED.
4194 ;*
4195 ;* CALLING SEQUENCE: JSR PC,WTWLPR
4196 ;*
4197 ;* COMMENTS:
4198 ;*
4199 ;* SUBORDINATE ROUTINES CALLED: ALTFLD.
4200 ;* - *****
4201
4202 023604 WTWLPR:: SAVE ;SAVE CONTENTS OF GPRS R0 THRU R5.
4203 023604 004567 160166 JSR R5,PREG05 ;CALL REGISTER SAVE SUBRT.
4204 ;*
4205 ;* SET UP THE PARAMETERS FOR THE CALL TO ALTFLD.
4206 ;*
4206 023610 016701 156434 MOV LPRA,R1 ;SET UP THE REGISTER ADDRESS PARAMETER.
4207 023614 010002 MOV R0,R2 ;SET UP THE DESIRED REGISTER CONTENTS.
4208 023616 010503 MOV R5,R3 ;SET UP THE BIT MAP OF LINES TO ALTER.
4209 023620 012704 177777 MOV #-1,R4 ;SELECT ALL REGISTER BITS TO BE ALTERED.
4210 ;*
4211 ;* CALL THE SUBROUTINE WHICH ALTERS THE REGISTER CONTENTS.
4212 ;*
4213 023624 0C4767 173776 JSR PC,ALTFLD ;ALTER THE REGISTER CONTENTS.
4214 ;*
4215 023630 603: PASS ;RESTORE GPRS.
4215 023630 004736 JSR PC,@(SP)+ ;RETURN TO PREG05 SUBRT.
4216 023632 000207 RTS PC

```

INTERRUPT SERVICE ROUTINE

CACHRX -

```

4218 .SBTTL  INTERRUPT SERVICE ROUTINE          CACHRX
4219 ;* .....
4220 ;*      CATCH RECEIVER INTERRUPT.
4221 ;*      THIS ROUTINE IS USED IN SEVERAL TESTS, TO LOG A COUNT OF THE
4222 ;*      NUMBER OF RECEIVER INTERRUPTS THAT OCCUR.
4223 ;*
4224 ;* INPUTS:      CSRA  CONTAINS THE ADDRESS OF THE CSR.
4225 ;*              RXINTC  HOLDS THE COUNT OF THE NUMBER OF RX INTERRUPTS
4226 ;*                      THAT OCCURRED.
4227 ;*
4228 ;* OUTPUTS:     RXINTC  CONTAINS THE UPDATED INTERRUPT COUNT.
4229 ;*
4230 ;*
4231 ;* CALLING SEQUENCE:  PUT THE ADDRESS OF THE LABEL CACHRX IN THE VECTOR
4232 ;*                   LOCATION.
4233 ;*
4234 ;* COMMENTS:
4235 ;*
4236 ;* SUBORDINATE ROUTINES CALLED: NONE
4237 ;* .....
4238
4239 023634 CACHRX::SAVE                                ;SAVE CONTENTS OF GPRS R0 THRU R5.
          023634 004567 160136                        R5,PREG05          ;CALL REGISTER SAVE SUBRT.
4240 023640 016701 156470                        MOV    RXINTC,R1      ;GET THE RECEIVER INTERRUPT COUNT
          023644 005201                                INC    R1          ;INCREMENT THE COUNT
4242 023646 102001                                BVC    2$          ;BRANCH IF NO OVERFLOW OCCURRED
4243 023650 005301                                DEC    R1          ;RESET THE COUNT TO 177777
4244 023652 010167 156456                        2$:  MOV    R1,RXINTC      ;SAVE NEW COUNT VALUE
4245 023656                                60$:  PASS          ;RESTORE GPRS.
          023656 004736                                JSR                    PC,@(SP)+ ;RETURN TO PREG05 SUBRT.
4246 023660 000002                                RTI

```

INTERRUPT SERVICE ROUTINE

CACHTX -

```

4248 .SBTTL  INTERRUPT SERVICE ROUTINE          CACHTX -
4249 ;* *****
4250 ;*      - CATCH TRANSMITTER INTERRUPT.
4251 ;*      THIS ROUTINE IS USED IN SEVERAL TESTS, TO LOG A COUNT OF THE
4252 ;*      NUMBER OF TRANSMISSION INTERRUPTS THAT OCCUR.
4253 ;*
4254 ;* INPUTS:      CSRA  CONTAINS THE ADDRESS OF THE CSR.
4255 ;*              TXINTC  HOLDS THE COUNT OF THE NUMBER OF TX INTERRUPTS
4256 ;*                      THAT OCCURRED.
4257 ;*
4258 ;* OUTPUTS:     TXINTC - CONTAINS THE UPDATED INTERRUPT COUNT.
4259 ;*
4260 ;*
4261 ;* CALLING SEQUENCE:  PUT THE ADDRESS OF THE LABEL CACHTX IN THE VECTOR
4262 ;*                      LOCATION.
4263 ;*
4264 ;* COMMENTS:
4265 ;*
4266 ;* SUBORDINATE ROUTINES CALLED: NONE
4267 ;* -- *****
4268
4269 023662 CACHTX:  SAVE                                ;SAVE CONTENTS OF GPRS R0 THRU R5.
023662 004567 160110                                R5,PREG05 ;CALL REGISTER SAVE SUBRT.
4270 023666 016701 156446                                MOV    TXINTC,R1 ;GET THE TRANSMISSION INTERRUPT COUNT
4271 023672 005201                                INC    R1 ;INCREMENT THE COUNT
4272 023674 102001                                BVC    2$ ;BRANCH IF NO OVERFLOW OCCURRED
4273 023676 005301                                DEC    R1 ;RESET THE COUNT TO 177777
4274 023700 010167 156434 2$: MOV    R1,TXINTC ;SAVE NEW COUNT VALUE
4275 023704 004736 60$:  PASS ;RESTORE GPRS.
023704 004736                                JSR    PC,@(SP). ;RETURN TO PREG05 SUBRT.
4276 023706 000002                                RTI

```

INTERRUPT SERVICE ROUTINE

CLKINT -

```

4278 .SBTTL INTERRUPT SERVICE ROUTINE CLKINT
4279 ;... *****
4280 ;* THIS ROUTINE IS EXECUTED CLKHRZ TIMES PER SECOND. IT DECREMENTS THE
4281 ;* TWO TIMER COUNTERS DOWN TO ZERO.
4282 ;*
4283 ;* INPUTS: TIMER1 TIMER COUNTER #1.
4284 ;* TIMER2 - TIMER COUNTER #2.
4285 ;* TIMER3 TIMER COUNTER FOR CALL OF BREAK MACRO.
4286 ;*
4287 ;* OUTPUTS: THE 2 TIMER COUNTERS ARE DECREMENTED IF THEY ARE NOT ZERO.
4288 ;*
4289 ;* CALLING SEQUENCE: PUT #CLKINT IN THE CLOCK INTERRUPT VECTOR SLOT.
4290 ;* PUT THE DESIRED TIME PERIOD (SECONDS TIMES CLKHRZ) IN
4291 ;* EITHER TIMER1 OR TIMER2 AND POLL THE RESPECTIVE TIMER
4292 ;* COUNTER TO DETECT ITS GOING TO 0 ON TIME-OUT.
4293 ;*
4294 ;* COMMENTS: THE 2 COUNTERS WILL NOT WRAPAROUND BUT WILL STOP AT 0. THIS
4295 ;* ALLOWS THE DETECTION OF A TIME-OUT ANY TIME AFTER THE TIME OUT
4296 ;* HAS OCCURRED UNTIL THE TIMER COUNTER IS SET TO ANOTHER VALUE.
4297 ;*
4298 ;* SUBORDINATE ROUTINES CALLED: NONE.
4299 ;- *****
4300
4301 023710 005767 156450 CLKINT:: TST TIMER1 ;CHECK FOR TIMER1 AT ZERO.
4302 023714 001402 BEQ 2$ ;BRANCH TO LEAVE IT AT ZERO IF IT IS ZERO.
4303 023716 005367 156442 DEC TIMER1 ;DECREMENT TIME COUNT.
4304 023722 005767 156440 2$: TST TIMER2 ;CHECK FOR TIMER2 AT ZERO.
4305 023726 001402 BEQ 4$ ;BRANCH TO LEAVE IT ALONE IF IT'S ALREADY ZERO.
4306 023730 005367 156432 DEC TIMER2 ;DECREMENT TIME COUNT.
4307 023734 005367 156430 4$: DEC TIMER3 ;DECREMENT THE BREAK COUNT.
4308 023740 001006 BNE 60$ ;EXIT IF NOT TIME TO CALL BREAK.
4309 023742 016767 156424 156420 MOV BCOUNT,TIMER3 ;SET UP TIME TILL NEXT BREAK.
4310 023750 010046 MOV RO,-(SP) ;SAVE CONTENTS OF RO FROM BREAK MACRO.
4311 023752 BREAK ;CHECK FOR OPERATOR CONTROL/C.
4312 023754 104422 TRAP C$BRK
4313 023756 012600 MOV (SP)+,RO ;RESTORE CONTENTS OF RO.
4313 023756 000002 60$: RTI

```

INTERUPT SERVICE ROUTINE

RXBRRT -

```

4315 .SBTTL  INTERRUPT SERVICE ROUTINE      . RXBRRT
4316 ;... *****
4317 ;*      - BR LEVEL TEST RECEIVE INTERRUPT SERVICE ROUTINE -
4318 ;*      THIS SERVICE ROUTINE HANDLES RECEIVE INTERRUPTS DURING THE INTERRUPT
4319 ;*      BR LEVEL TEST.  THIS ROUTINE COUNTS THE INTERRUPT AND SETS A FLAG
4320 ;*      TO INDICATE THAT THE INTERRUPT HAS OCCURRED.  IT ALSO CHECKS THE
4321 ;*      FLAG WHICH INDICATES THAT A TX INTERRUPT HAS OCCURRED.  IF THE TX
4322 ;*      INTERRUPT FLAG IS SET, THIS ROUTINE SETS AN INTERRUPT ORDER ERROR
4323 ;*      FLAG INDICATING THAT A TRANSMIT INTERRUPT WAS SERVICED BEFORE A
4324 ;*      SIMULTANEOUS RECEIVE INTERRUPT.
4325 ;*
4326 ;* INPUTS:      RXINTC  HOLDS THE COUNT OF THE NUMBER OF RX INTERRUPTS.
4327 ;*              RXINTF  RX INTERRUPT FLAGS.
4328 ;*
4329 ;* OUTPUTS:     RXINTC  CONTAINS THE UPDATED INTERRUPT COUNT.
4330 ;*              RXINTF  RX INT FLAGS:
4331 ;*                      (BIT 0 SET, BIT 14 SET IF TXINTF BIT 0 IS SET.)
4332 ;*
4333 ;* CALLING SEQUENCE:  PUT THE ADDRESS OF THE LABEL RXBRRT IN THE VECTOR
4334 ;*                      LOCATION.
4335 ;*
4336 ;* COMMENTS:      NOTE: THE FIFO IS PURGED BY THIS ROUTINE.
4337 ;*
4338 ;* SUBORDINATE ROUTINES CALLED: NONE.
4339 ; *****
4340
4341 RXBRRT:  SAVE                                ;SAVE CONTENTS OF GPRS R0 THRU R5.
4342          023760 004567 160012                R5,PREG05      ;CALL REGISTER SAVE SUBRT.
4343          023764 017700 156256                MOV    BRBUFA,R0  ;READ THE CHAR OUT OF THE FIFO.
4344          023770 016701 156340                MOV    RXINTC,R1  ;GET THE INTERRUPT COUNT.
4345          023774 005201                        INC    R1        ;INCREMENT THE COUNT.
4346          023776 001402                        BEQ    2$        ;BYPASS UPDATING COUNT IF OVERFLOW OCCURRED.
4347          024000 010167 156330                MOV    R1,RXINTC  ;SAVE NEW COUNT VALUE.
4348          024004 016701 156326                2$:  MOV    RXINTF,R1  ;GET THE RX INTERRUPT FLAGS.
4349          024010 052701 000001                BIS    #BIT0,R1  ;SET THE RX INTERRUPT HAS OCCURRED FLAG.
4350          024014 032767 000001 156320        BIT    #BIT0,TXINTF ;TEST THE "TX INT HAS OCCURRED" FLAG.
4351          024022 001402                        BEQ    4$        ;SKIP SETTING ERROR FLAG IF NO TX INT.
4352          024024 052701 040000                BIS    #BIT14,R1  ;SET THE INTERRUPT ORDER ERROR FLAG.
4353 ;*
4354 ;* 8 FIFO CODES WILL CAUSE 8 INTERRUPTS, AFTER THESE 8 CODES WE DON'T WANT
4355 ;*  TO CHECK THE INTERRUPT ORDER, BECAUSE PERHAPS A BMP CODE HAS COME IN
4356 ;*  BETWEEN THE SERVICING OF THE 8 FIFO CODE INTERRUPTS AND THE SERVICING
4357 ;*  OF ONE OF THE TX INTERRUPTS.
4358          024030 026767 156300 153752        4$:  CMP    RXINTC,NUMLNS ;TEST FOR ALL SELFTEST CODE INTS DONE.
4359          024036 003002                        BGT    60$        ;SKIP UPDATING RX INT FLAGS IF EXTRA RX INTS.
4360          024040 010167 156272                MOV    R1,RXINTF  ;UPDATE THE RX INTERRUPT FLAGS.
4361          024044 004736                        60$:  PASS      ;RESTORE GPRS.
4362          024046 000002                        RTI              ;RETURN TO PREG05 SUBRT.

```

INTERUPT SERVICE ROUTINE

RXINPT -

```

4364 .SBTTL  INTERUPT SERVICE ROUTINE          RXINPT
4365 :*****
4366 :      RECEIVE CHARACTER INPUT INTERRUPT SERVICE ROUTINE
4367 :      THIS SERVICE ROUTINE INPUTS A CHARACTER FROM THE DUT AND LOADS THE
4368 :      CHAR (COMPLETE WITH STATUS FLAGS) INTO A RECEIVE CHAR BUFFER IN
4369 :      MEMORY.  THE INTERRUPT IS ALSO COUNTED.  THE RECEIVE CHAR BUFFER IS
4370 :      MONITORED TO ENSURE THAT IT DOES NOT OVERFLOW.
4371 :
4372 :      INPUTS:      BUFEND  LABELS THE END OF THE HOST MEMORY BUFFER.
4373 :                   BUFPTR - CONTAINS ADDRESS OF NEXT FREE BUFFER LOCATION.
4374 :                   CSRA   CONTAINS THE ADDRESS OF THE DUT CSR.
4375 :                   RBUFA  CONTAINS THE ADDRESS OF THE RBUF DUT REGISTER.
4376 :                   RXINTC - HOLDS THE COUNT OF THE NUMBER OF RX INTERRUPTS.
4377 :                   RXINTF  RX INTERRUPT FLAGS.
4378 :
4379 :      OUTPUTS:     BUFPTR  CONTAINS UPDATED ADDRESS OF NEXT FREE BUFFER LOCATION.
4380 :                   RXINTC  CONTAINS THE UPDATED INTERRUPT COUNT.
4381 :                   RXINTF  RX INT FLAGS (BIT 15 SET IF RX.DATA.AVAIL IS CLEAR).
4382 :
4383 :      CALLING SEQUENCE:  PUT THE ADDRESS OF THE LABEL RXINPT IN THE VECTOR
4384 :                           LOCATION.
4385 :
4386 :      COMMENTS:      IN CASE OF OVERFLOW OF THE MEMORY BUFFER, BUFPTR WILL BE
4387 :                      MAINTAINED EQUAL TO BUFEND AND THE WORD AT BUFPTR WILL BE
4388 :                      THE LAST WORD READ FROM THE DUT FIFO.
4389 :                      NOTE: THIS ROUTINE CAN DESTROY TX.ACTIONS BY READING THE CSR.
4390 :
4391 :      SUBORDINATE ROUTINES CALLED:  NONE.
4392 :*****
4393
4394 024050 RXINPT:: SAVE                                ;SAVE CONTENTS OF GPRS R0 THRU R5.
4395 024050 004567 157722                                R5,PREG05 ;CALL REGISTER SAVE SUBRT.
4396 024054 032777 000200 156162                        BIT  #BIT7,@CSRA ;TEST RX.DATA.AVAIL BIT OF THE CSR (READS CSR).
4397 024062 001003                                BNE  2$ ;BRANCH AROUND SETTING FLAG IF BIT IS SET.
4398 024064 052767 100000 156244                        BIS  #BIT15,RXINTF ;SET THE RX.DATA.AVAIL CLEAR FLAG.
4399 024072 016701 156236 2$:                          MOV  RXINTC,R1 ;GET THE INTERRUPT COUNT.
4400 024100 001402                                INC  R1 ;INCREMENT THE COUNT.
4401 024102 010167 156226                                BEQ  4$ ;BYPASS UPDATING COUNT IF OVERFLOW OCCURRED.
4402 024106 016702 156212 4$:                          MOV  R1,RXINTC ;SAVE NEW COUNT VALUE.
4403 024112 017722 156130                                MOV  BUFPTR,R2 ;GET THE POINTER TO NEXT FREE BUFFER WORD.
4404 024116 020267 157604                                MOV  @RBUFA,(R2). ;READ A CHAR FROM THE FIFO INTO BUFFER.
4405 024122 103002                                CMP  R2,BUFEND ;TEST FOR POINTER BEYOND END OF BUFFER.
4406 024124 010267 156174                                BHS  60$ ;SKIP THE PTR UPDATE IF PTR OUT OF BOUNDS.
4407 024130 60$:  MOV  R2,BUFPTR ;UPDATE THE BUFFER POINTER.
4408 024130 004736                                PASS ;RESTORE GPRS.
4408 024132 000002                                JSR  PC,@(SP). ;RETURN TO PREG05 SUBRT
                                RTI

```

GLOBAL TRAP SERVICE ROUTINE TP4RTN

```

4410 .SBTTL GLOBAL TRAP SERVICE ROUTINE TP4RTN
4411 ;*****
4412 ;* BUS TIME OUT TRAP (004 TRAP) SERVICE ROUTINE
4413 ;* THIS ROUTINE IS USED DURING THE DEVICE REGISTER ADDRESS ACCESS TEST.
4414 ;* IT DETERMINES IF THE 004 TRAP WAS CAUSED BY AN "EXPECTED" ERROR OR
4415 ;* NOT BY EXAMINING THE RETURN PC VALUE ON THE STACK. IF THE TRAP IS
4416 ;* UNEXPECTED, THIS ROUTINE JUMPS TO THE NORMAL DIAGNOSTIC SUPERVISOR
4417 ;* 004 TRAP HANDLING ROUTINE.
4418 ;*
4419 ;* INPUTS: SP POINTS TO THE PC WHERE THE TRAP OCCURED.
4420 ;* ADRPTR - LABEL AT THE ADDRESS WHERE "EXPECTED" TRAPS OCCUR.
4421 ;* TP4FLG - 004 TRAP FLAGS.
4422 ;*
4423 ;* OUTPUTS: TP4FLG - BIT 15 IS SET IF "EXPECTED" TRAP OCCURED.
4424 ;*
4425 ;* CALLING SEQUENCE: PUT ADDRESS POINTED TO BY TP4RTN IN 004 VECTOR.
4426 ;* OCCURENCE OF 004 TRAP VECTORS TO THIS ROUTINE.
4427 ;*
4428 ;* COMMENTS: ANY 004 TRAP WHICH OCCURS AT AN ADDRESS OTHER THAN THAT LABELED
4429 ;* ADRPTR WILL BE HANDLED BY THE NORMAL 004 TRAP SERVICE ROUTINE.
4430 ;*
4431 ;* SUBORDINATE ROUTINES CALLED: NONE.
4432 ;*****
4433
4434 024134 021627 020136 TP4RTN:: CMP (SP),@ADRPTR ;COMPARE EXPECTED ADR AGAINST TRAP RET PC.
4435 024140 001402 BEQ 2$ ;IF THEY MATCH, CONTINUE THIS ROUTINE.
4436 024142 000177 156176 JMP @TP4VEC ;IF NOT, JUMP TO NORMAL 004 TRAP SERVICE RTN.
4437 024146 052767 100000 156172 2$: BIS @BIT15,TP4FLG ;SET THE 004 TRAP OCCURED FLAG.
4438 024154 000002 RTI ;ALL DONE, GO BACK TO THE TEST.

```

INTERUPT SERVICE ROUTINE

TXINTR -

```

4440 .SBTTL  INTERRUPT SERVICE ROUTINE          TXINTR
4441 ;.. *****
4442 ;*      TRANSMIT INTERRUPT SERVICE ROUTINE
4443 ;*      THIS ROUTINE HANDLES A TRANSMIT INTERRUPT FROM THE DEVICE UNDER TEST
4444 ;*      (DUT) BY COUNTING THE INTERRUPT AND READING THE DUT CSR TO CLEAR THE
4445 ;*      INTERRUPT REQUEST.  THIS ROUTINE ALSO SETS A FLAG TO INDICATE THAT
4446 ;*      A TX INTERRUPT HAS OCCURRED AND SETS A FLAG IF THE TX.ACTION BIT IS
4447 ;*      NOT SET IN THE READ CONTENTS OF THE DUT CSR.
4448 ;*
4449 ;* INPUTS:      CSRA - CONTAINS THE ADDRESS OF THE CSR.
4450 ;*              TXINTC - HOLDS THE COUNT OF THE NUMBER OF TX INTERRUPTS.
4451 ;*              TXINTF - TX INTERRUPT FLAGS.
4452 ;*
4453 ;* OUTPUTS:     TXINTC - CONTAINS THE UPDATED TX INTERRUPT COUNT.
4454 ;*              TXINTF - TX INT FLAGS (BIT 0 SET, BIT 15 SET IF TX.ACTION CLR).
4455 ;*
4456 ;* CALLING SEQUENCE:  PUT THE ADDRESS OF THE LABEL TXINTR IN THE VECTOR
4457 ;*                    LOCATION.
4458 ;*
4459 ;* COMMENTS:
4460 ;*
4461 ;* SUBORDINATE ROUTINES CALLED: NONE
4462 ;.. *****
4463
4464 024156 TXINTR:: SAVE                                ;SAVE CONTENTS OF GPRS R0 THRU R5.
4465 024156 004567 157614                                R5,PREG05 ;CALL REGISTER SAVE SUBRT.
4466 024162 016701 156152                                MOV    TXINTC,R1 ;GET THE TX INTERRUPT COUNT.
4467 024170 102001                                INC    R1 ;INCREMENT THE COUNT.
4468 024172 005301                                BVC    2$ ;BRANCH IF NO OVERFLOW OCCURRED.
4469 024174 010167 156140                                DEC    R1 ;RESET THE COUNT TO 177777.
4470 024200 016703 156136 2$: MOV    R1,TXINTC ;SAVE NEW COUNT VALUE.
4471 024204 017702 156034                                MOV    TXINTF,R3 ;GET THE TX INTERRUPT FLAGS.
4472 024210 100402                                MOV    @CSRA,R2 ;READ THE CSR.
4473 024212 052703 100000                                BMI    4$ ;SKIP SETTING OF FLAG IF TX.ACTION IS SET.
4474 024216 052703 000001                                BIS    @BIT15,R3 ;SET THE TX.ACTION CLEAR FLAG.
4475 024222 010367 156114 4$: BIS    @BIT0,R3 ;SET THE TX INT HAS OCCURRED FLAG.
4476 024226                                MOV    R3,TXINTF ;UPDATE THE TX INTERRUPT FLAGS.
4477 024230 000002                                PASS 60$: ;RESTORE GPRS.
                                JSR    PC,@(SP)+ ;RETURN TO PREG05 SUBRT.
                                RTI

```

INTERUPT SERVICE ROUTINE

1XINTR -

```

4479
4480 ;*****
4481 ;
4482 ;          FVTA.RPT
4483 ;
4484 ;*****
4485
4486
4487
4488 .SBTTL  REPORT CODING SECTION
4489
4490 ;**
4491 ; THE REPORT CODING SECTION CONTAINS THE
4492 ; 'PRINTS' CALLS THAT GENERATE STATISTICAL REPORTS.
4493 ;
4494
4495 024232      BGNRPT
4496 024232
4497 024232      EXIT      RPT
4498 024232      000167
4499 024234      000000
4500
4501 024236
4502 024236
4503 024236      104425
4504
4505
4506
4507
4508
4509
4510
4511
4512
4513
4514
4515
4516
4517
4518
4519
4520
4521
4522
4523
4524
4525
4526
4527
4528
4529
4530
4531
4532
4533
4534
4535
4536
4537
4538
4539
4540
4541
4542
4543
4544
4545
4546
4547
4548
4549
4550
4551
4552
4553
4554
4555
4556
4557
4558
4559
4560
4561
4562
4563
4564
4565
4566
4567
4568
4569
4570
4571
4572
4573
4574
4575
4576
4577
4578
4579
4580
4581
4582
4583
4584
4585
4586
4587
4588
4589
4590
4591
4592
4593
4594
4595
4596
4597
4598
4599
4600
4601
4602
4603
4604
4605
4606
4607
4608
4609
4610
4611
4612
4613
4614
4615
4616
4617
4618
4619
4620
4621
4622
4623
4624
4625
4626
4627
4628
4629
4630
4631
4632
4633
4634
4635
4636
4637
4638
4639
4640
4641
4642
4643
4644
4645
4646
4647
4648
4649
4650
4651
4652
4653
4654
4655
4656
4657
4658
4659
4660
4661
4662
4663
4664
4665
4666
4667
4668
4669
4670
4671
4672
4673
4674
4675
4676
4677
4678
4679
4680
4681
4682
4683
4684
4685
4686
4687
4688
4689
4690
4691
4692
4693
4694
4695
4696
4697
4698
4699
4700
4701
4702
4703
4704
4705
4706
4707
4708
4709
4710
4711
4712
4713
4714
4715
4716
4717
4718
4719
4720
4721
4722
4723
4724
4725
4726
4727
4728
4729
4730
4731
4732
4733
4734
4735
4736
4737
4738
4739
4740
4741
4742
4743
4744
4745
4746
4747
4748
4749
4750
4751
4752
4753
4754
4755
4756
4757
4758
4759
4760
4761
4762
4763
4764
4765
4766
4767
4768
4769
4770
4771
4772
4773
4774
4775
4776
4777
4778
4779
4780
4781
4782
4783
4784
4785
4786
4787
4788
4789
4790
4791
4792
4793
4794
4795
4796
4797
4798
4799
4800
4801
4802
4803
4804
4805
4806
4807
4808
4809
4810
4811
4812
4813
4814
4815
4816
4817
4818
4819
4820
4821
4822
4823
4824
4825
4826
4827
4828
4829
4830
4831
4832
4833
4834
4835
4836
4837
4838
4839
4840
4841
4842
4843
4844
4845
4846
4847
4848
4849
4850
4851
4852
4853
4854
4855
4856
4857
4858
4859
4860
4861
4862
4863
4864
4865
4866
4867
4868
4869
4870
4871
4872
4873
4874
4875
4876
4877
4878
4879
4880
4881
4882
4883
4884
4885
4886
4887
4888
4889
4890
4891
4892
4893
4894
4895
4896
4897
4898
4899
4900
4901
4902
4903
4904
4905
4906
4907
4908
4909
4910
4911
4912
4913
4914
4915
4916
4917
4918
4919
4920
4921
4922
4923
4924
4925
4926
4927
4928
4929
4930
4931
4932
4933
4934
4935
4936
4937
4938
4939
4940
4941
4942
4943
4944
4945
4946
4947
4948
4949
4950
4951
4952
4953
4954
4955
4956
4957
4958
4959
4960
4961
4962
4963
4964
4965
4966
4967
4968
4969
4970
4971
4972
4973
4974
4975
4976
4977
4978
4979
4980
4981
4982
4983
4984
4985
4986
4987
4988
4989
4990
4991
4992
4993
4994
4995
4996
4997
4998
4999
5000

```

PROTECTION TABLE

4503
4504
4505
4506
4507
4508
4509
4510
4511
4512
4513
4514
4515
4516
4517
4518 024240
4519 024240
4520 024240 177777
4521 024242 177777
4522 024244 177777
4523
4524 024246
4525

.SBTTL PROTECTION TABLE

;*****

;
; FVTSK14.P11
;

;*****

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

BGNPROT

L1PROT::

1 ;OFFSET INTO P TABLE FOR CSR ADDRESS
1 ;OFFSET INTO P TABLE FOR MASSBUS ADDRESS
1 ;OFFSET INTO P TABLE FOR DRIVE NUMBER

ENDPROT

```

4540 ;*****
4541 ;
4542 ;          FVTA.INI
4543 ;
4544 ;*****
4545 ;*****
4546 ;*****
4547 ;*****
4548 ;*****
4549 .SBTTL INITIALIZE SECTION
4550 ;**
4551 ;*****
4552 ;*      THIS SECTION CONTAINS THE CODE WHICH IS PERFORMED AT THE BEGINNING OF
4553 ;*      EACH PASS OR AFTER A CONTINUE COMMAND.
4554 ;*      THIS CODE PERFORMS THE FOLLOWING ACTIONS:
4555 ;*
4556 ;*      MOVES THE INFORMATION HELD IN THE HARDWARE P TABLE INTO THE GLOBAL
4557 ;*      DATA AREA.
4558 ;*
4559 ;*****
4560 ;--
4561 024246 BGNINIT
4562 024246 L$INIT::
4563 ;SEE IF PROGRAM JUST STARTED, BR IF YES
4564 024246 012700 000040 MOV     DEF.START,R0
4565 024252 104447 TRAP    C$REFG
4566 024254 BCOMPLETE NEWSTA
4567 024254 103416 BCS     NEWSTA
4568 ;SEE IF PROGRAM JUST RESTARTED, BR IF YES
4569 024256 012700 000037 MOV     DEF.RESTART,R0
4570 024262 104447 TRAP    C$REFG
4571 024264 BCOMPLETE NEWRES
4572 024264 103556 BCS     NEWRES
4573 ;SEE IF THIS IS A NEW PASS, BR IF YES
4574 024266 012700 0C0035 MOV     DEF.NEW,R0
4575 024272 104447 TRAP    C$REFG
4576 024274 BCOMPLETE NEWPAS
4577 024274 103555 BCS     NEWPAS
4578 ;SEE IF PROGRAM WAS JUST CONTINUED
4579 024276 012700 000036 MOV     DEF.CONTINUE,R0
4580 024302 104447 TRAP    C$REFG
4581 024304 BNCOMPLETE GETPRM
4582 024304 103161 BCC     GETPRM
4583 024306 000167 000552 JMP     ENDIT
4584 024312 NEWSTA: BRESET           ;RESET THE BUS TO PREVENT ILLEGAL INTERRUPTS.
4585 024312 104433 TRAP    C$RESET
4586 ;*
4587 ; SET UP FOR LINE TIME CLOCK INTERRUPTS.
4588 ;
4589 024314 CLOCK L,R1             ;GET THE CLOCK PARAMETERS.
4590 024314 012700 000114 MOV     L,R0
4591 024320 104462 TRAP    C$CLK

```

INITIALIZE SECTION

```

024322 010001
4581 024324 012167 156024      MOV      (R1),CLKCSR      ;STORE CLOCK CSR ADDRESS.
4582 024330 012167 156022      MOV      (R1),CLKBRL     ;STORE CLOCK BUS REQ INT LEVEL.
4583 024334 012167 156020      MOV      (R1),CLKVEC     ;STORE CLOCK INTERRUPT VECTOR.
4584 024340 012167 156016      MOV      (R1),CLKHRZ     ;STORE CLOCK FREQUENCY.
4585 024344 026727 156012 000062 CMP      CLKHRZ,#50.      ;TEST FOR 50HZ LINE FREQUENCY.
4586 024352 061004              BNE      2$              ;BRANCH IF CLOCK IS NOT 50HZ.
4587 024354 012767 000024 156012 MOV      #20.,MSTICK      ;INDICATE 20MS PER CLOCK TICK.
4588 024362 000403              BR      4$
4589 024364 012767 000021 156002 2$: MOV      #17.,MSTICK      ;INDICATE 17 MS PER CLOCK TICK.
4590 024372              4$: SETVEC  CLKVEC,#CLKINT,PRI06 ;INITIALIZE CLOCK INTERRUPT VECTOR.
                                MOV      PRI06,(SP)
                                MOV      #CLKINT, -(SP)
                                MOV      CLKVEC, -(SP)
                                MOV      #3, -(SP)
                                TRAP     C$SVEC
                                ADD      #10,SP
024372 016746 153702
024376 012746 023710
024402 016746 155752
024406 012746 000003
024412 104437
024414 062706 000010
4591 024420 016700 155736      MOV      CLKHRZ,RO      ;INITIALIZE THE BREAK COUNT
4592 024424 006300              ASL      RO              ; TO CAUSE A BREAK
4593 024426 010067 155740      MOV      RO,BCOUNT    ; EVERY 2 SECONDS.
4594 024432              SETPRI  #PRI05      ;ALLOW CLOCK INTERRUPTS DISABLE OTHERS.
                                MOV      #PRI05,RO
                                TRAP     C$SPRI
024432 012700 000240
024436 104441
4595
4596 ;*
4597 ; ENABLE THE LINE TIME CLOCK (LTC) CHECKING TO MAKE SURE THAT THE CSR
4598 ; IS ACCESSABLE.
4599 ; FIRST SET UP TO CATCH ANY 004 TRAPS WHICH OCCUR:
4600 024440 016767 153340 155676 ;*
4601 024446 012767 024134 153330 ;-
                                MOV      4,TP4VEC      ;SAVE THE EXISTING 004 TRAP VECTOR.
                                MOV      #TP4RTN,4      ;SET 004 TRAP VECTOR TO OUR SERVICE RTN ADR.
4602 ;*
4603 ; ENABLE LTC CHECKING FOR 004 TRAP IN CASE CSR IS NOT THERE.
4604 ;
4605 024454 005067 155666
4606 024460 012767 000100 155662
4607 024466 012700 002350
4608 024472 016701 155656
4609 024476 004767 173422
4610 024502 016767 155636 153274
4611 024510 103403
4612 024512 005067 155644
4613 024516 000402
4614 ;*
4615 ; CALIBRATE THE DELAY ROUTINE MILLI-SECOND DELAY COUNT VALUE.
4616 ;
4617 024520 004767 173154      6$: JSR      PC,CALMSL
4618 ;*
4619 ; CHECK FOR MEMORY MANAGEMENT PRESENT ON THIS MACHINE.
4620 ; IF MEM MGT IS PRESENT, DISABLE IT.
4621 ;
4622 024524 016767 153254 155612 ;*
4623 024532 012767 024134 153244 ;-
                                MOV      4,TP4VEC      ;SAVE THE EXISTING 004 TRAP VECTOR.
                                MOV      #TP4RTN,4      ;SET 004 TRAP VECTOR TO OUR SERVICE RTN ADR.
                                CLR      TP4FLG      ;CLEAR THE 004 TRAP FLAG.
                                CLR      WORD1      ;PREPARE TO CLEAR THE MEM MGT SRO REGISTER.
                                MOV      #WORD1,RO    ;SELECT CLEARED WORD AS CKTRAP RTN SOURCE.
                                MOV      #MSRO,R1     ;SELECT MEM MGT SRO REGISTER AS DESTINATION.
                                CLR      #MPRES      ;INDICATE NO MEM MGT PRESENT IN CASE IT ISN'T.
4624 024540 005067 155602
4625 024544 005067 155600
4626 024550 012700 002350
4627 024554 016701 155620
4628 024560 005067 155616

```

INITIALIZE SECTION

```

4629 024564 005067 155614          CLR    MMENAB          ;INDICATE MEM MGT IS NOT ENABLED.
4630 024570 004767 173330          JSR    PC,CKTRAP      ;CLEAR THE MEM MGT SRO REG AND CHECK FOR TRAP.
4631 024574 016767 155544 153202    MOV    TP4VEC,4        ;RESTORE THE NORMAL 004 TRAP VECTOR.
4632 024602 103003                   BCC    10$              ;SKIP INDICATING MEM MGT PRESENT IF IT ISN'T.
4633 024604 012767 000001 155570    MOV    #1,MMPRES        ;INDICATE THAT MEM MGT IS PRESENT.
4634 024612 005067 155514 10$:     CLR    PASCNT          ;CLR COUNTER USED IN REPORTING ROM VERSION #.
4635 024616 000167 000006          JMP    NEWPAS          ;SKIP AROUND THE BUS RESET, IT S BEEN DONE.
4636
4637 024622                   NEWRES: BRESET          ;RESET THE BUS TO PREVENT ILLEGAL INTERRUPTS.
                                TRAP    CSRESF
4638 024622 104433                   CLR    PASCNT          ;CLR COUNTER USED IN REPORTING ROM VERSION #.
4639 024630                   NEWPAS:
4640 024630 012767 177777 155402    MOV    #1,UNITN        ;RESET LOGICAL DEVICE TO 1
4641
4642
4643
4644
4645 024636 005267 155470          INC    PASCNT          ;INCREMENT THE PASS COUNTER.
4646 024642 001002                   BNE    GETPRM          ;BRANCH IF WE HAVE NOT YET OVERFLOWED.
4647 024644 005367 155462          DEC    PASCNT          ;SET PASS COUNT TO 177777 OCTAL..
4648
4649
4650 024650                   ; GET THE HARDWARE PARAMETERS FOR THIS UNIT.
4651 024650 005267 155364 155130    GETPRM: INC    UNITN          ;INCREMENT LOGICAL DEVICE NUMBER
4652 024654 026767 155360          CMP    UNITN,L$UNIT      ;SEE IF MAXIMUM UNIT NO. EXCEEDED
4653 024662 002362                   BGE    NEWPAS          ;BR IF YES
4654
4655 024664          GPHARD UNITN,R1          ;GET P TABLE POINTER INTO R1
                                MOV     UNITN,R0
                                TRAP    CS$GPHARD
                                MOV     R0,R1
4656 024674          BCOMPLETE 30$          ;BR IF DEVICE AVAILABLE
4657 024676 103401 000764          BR     GETPRM          ;SKIP THIS DEVICE
4658
4659
4660
4661 024700 012167 155340 30$:      ;***** HARDWARE PARAMETER MOVING CODE *****
4662 024704 012102          MOV    (R1)+,CSRA          ;STORE DHV-11 CSR ADDRESS IN DEV.REG.ADDRESS TABLE
4663 024706 010267 155320          MOV    (R1)+,R2          ;GET THE RX INTERRUPT VECTOR ADDRESS.
4664 024712 062702 000004          MOV    R2,RXVECA        ;STORE PX INT VECTOR ADDRESS.
4665 024716 010267 155312          ADD    #4,R2          ;CALCULATE TX INTERRUPT VECTOR ADDRESS.
4666 024722 012167 155310          MOV    R2,TXVECA        ;STORE TX INT VECTOR ADDRESS.
4667 024726 012702 000377          MOV    (R1)+,ACTLNS        ;STORE DHV-11 ACTIVE LINE BIT MAP
4668 024732 005102          MOV    #MAPLNS,R2          ;GET THE BIT MAP FOR ALL LINES.
4669 024734 040267 155276          COM    R2          ;GET A BIT MAP OF NON EXISTANT LINES.
4670 024740 112167 155276          BIC    R2,ACTLNS        ;CLEAR NON-EXISTANT LINES FROM ACTLNS.
4671
4672
4673
4674
4675 024744 016701 155274          MOV    CSRA,R1          ;COPY CSR ADDRESS
4676 024750 005201          INC    R1          ;INCREMENT CSR ADDRESS
4677 024752 005201          INC    R1          ; COPY BY 2.
4678 024754 012703 000007          MOV    #7,R3          ;SET UP REGISTER COUNT
4679 024760 012702 002246          MOV    #RBUFA,R2        ;GET LOCATION WHERE RBUF ADDRESS GOES IN TABLE
4680 024764 010122 12$:          MOV    R1,(R2)+          ;STORE REGISTER ADDRESS IN TABLE

```

INITIALIZE SECTION

```

4681 024766 005201      INC      R1      ; INCREMENT REGISTER ADDRESS
4682 024770 005201      INC      R1      ; BY 2, FOR THE NEXT DEVICE REGISTER.
4683 024772 005303      DEC      R3      ; DECREMENT REGISTER COUNT
4684 024774 001373      BNE      12$     ; LOOP IF NOT DONE
4685
4686
4687
4688      ;*
      ; INITIALISE THE BMP CODE QUEUE.
      ;
      MOV      @BMPQ8,R0      ; GET THE START ADDRESS OF THE QUEUE.
      MOV      @BMPQ8,R1      ; GET THE END ADDRESS OF THE QUEUE.
      MOV      R0,BMPQ8      ; SET THE POINTER TO THE START OF THE QUEUE.
14$:      CLR      (R0).      ; CLEAR OUT THE CONTENTS OF THE QUEUE.
      CMP      R0,R1      ; CHECK IF END OF QUEUE HAS BEEN REACHED.
      BLO      14$      ; LOOP IF NOT ALL DONE.
      ;*
      ; REPORT THE UNIT NUMBER IF THE SOFTWARE P-TABLE QUESTION WAS ANSWERED YES,
      ; AND THE MAXIMUM UNIT NUMBER IS GREATER THAN 1.
      ;
      BIT      @BIT4,OPTION      ; CHECK IF THE QUESTION WAS ANSWERED YES.
      BEQ      16$      ; SKIP REPORTING UNIT NUMBER IF IT IS DISABLED.
      CMP      L$UNIT,01      ; CHECK MAXIMUM NUMBER OF UNITS SELECTED.
      BLE      16$      ; DO NOT REPORT UNIT NUMBER IF MAX NUMBER < 1.
      PRINTF   @MFUNIT,UNITN      ; REPORT UNIT NUMBER.
      MOV      UNITN,(SP)
      MOV      @MFUNIT,-(SP)
      MOV      02,-(SP)
      MOV      SP,R0
      TRAP     C$PRINTF
      ADD      06,SP
4704 025064      16$:
4705
4706 025064 005067 155262      ENDIT: CLR      CTRLCF      ; CLR THE CTRL C TEST ABORT FLAG.
4707
4708      ;*
4709      ; SET THE PROCESSOR PRIORITY TO ALLOW LTC INTERRUPTS BUT NOT OTHERS.
      ;
      SETPRI   @PRI07      ; SET PROCESSOR PRIORITY TO 7.
      MOV      @PRI07,R0
      TRAP     C$SPRI
4710 025070
      025070 012700 000340
      025074 104441
      025076
      025076 104411
      025076
      L10021:      TRAP     C$INIT
4712
4713      TNUM == 0      ; INITIALIZE THE ASSEMBLER TEST NUMBER VARIABLE.

```

INITIALIZE SECTION

```

4716 ;*****
4717 ;
4718 ;          FVTA.ATD
4719 ;
4720 ;*****
4721
4722
4723
4724 .SBTTL  AUTODROP SECTION
4725
4726
4727 ;**
4728 ; THIS CODE IS EXECUTED IMMEDIATELY AFTER THE INITIALIZE CODE IF
4729 ; THE 'ADR' FLAG WAS SET.  THE UNIT(S) UNDER TEST ARE CHECKED TO
4730 ; SEE IF THEY WILL RESPOND.  THOSE THAT DON T ARE IMMEDIATELY
4731 ; DROPPED FROM TESTING.
4732 ;
4733
4734 025100          BGNAUTO
4735 025100
4736
4737
4738
4739 025100          ENDAUTO
4740 025100
4741 025100 104461
4742
4743
4744
4745
4746
4747
4748
4749
4750
4751
4752
4753
4754
4755
4756
4757
4758
4759
4760
4761
4762
4763
4764
4765
4766
4767
4768
4769
4770
4771
4772
4773
4774
4775
4776
4777
4778
4779
4780
4781
4782
4783
4784
4785
4786
4787
4788
4789
4790
4791
4792
4793
4794
4795
4796
4797
4798
4799
4800
4801
4802
4803
4804
4805
4806
4807
4808
4809
4810
4811
4812
4813
4814
4815
4816
4817
4818
4819
4820
4821
4822
4823
4824
4825
4826
4827
4828
4829
4830
4831
4832
4833
4834
4835
4836
4837
4838
4839
4840
4841
4842
4843
4844
4845
4846
4847
4848
4849
4850
4851
4852
4853
4854
4855
4856
4857
4858
4859
4860
4861
4862
4863
4864
4865
4866
4867
4868
4869
4870
4871
4872
4873
4874
4875
4876
4877
4878
4879
4880
4881
4882
4883
4884
4885
4886
4887
4888
4889
4890
4891
4892
4893
4894
4895
4896
4897
4898
4899
4900
4901
4902
4903
4904
4905
4906
4907
4908
4909
4910
4911
4912
4913
4914
4915
4916
4917
4918
4919
4920
4921
4922
4923
4924
4925
4926
4927
4928
4929
4930
4931
4932
4933
4934
4935
4936
4937
4938
4939
4940
4941
4942
4943
4944
4945
4946
4947
4948
4949
4950
4951
4952
4953
4954
4955
4956
4957
4958
4959
4960
4961
4962
4963
4964
4965
4966
4967
4968
4969
4970
4971
4972
4973
4974
4975
4976
4977
4978
4979
4980
4981
4982
4983
4984
4985
4986
4987
4988
4989
4990
4991
4992
4993
4994
4995
4996
4997
4998
4999
5000

```

L\$AUTO::

L10022: TRAP C\$AUTO

AUTODROP SECTION

```

4745
4746 ;*****
4747 ;
4748 ;           FVT.CUC
4749 ;
4750 ;*****
4751
4752
4753
4754 .SBTTL  CLEANUP CODING SECTION
4755
4756 ;**
4757 ; THE CLEANUP CODING SECTION CONTAINS THE CODING THAT IS PERFORMED
4758 ; AFTER THE HARDWARE TESTS HAVE BEEN PERFORMED.
4759 ;
4760
4761 025102      BGNCLN
4762 025102
4763
4771 025102 005767 155244      TST      CTRLCF      ;DID WE GET HERE BY CTRL-C FROM TEST?
4772 025106 001401            BEQ      2$          ;CTRL C FROM TEST? NO, SKIP BUS RESET.
4773 025110            BRESET          ;YES, CLR ANY DMAS OR OUTSTANDING INTERRUPTS.
4774 025110 104433            TRAP      C$RESET
4775 025112 2$:            EXIT      CLN
4776 025112 104432            TRAP      C$EXIT
4777 025114 000002            .WORD    L10023 .
4778
4788
4789
4790
4791 025116      ENDCLN
4792 025116
4793 025116 104412            L10023: TRAP      C$CLEAN

```

CLEANUP CODING SECTION

```

4793
4794
4795
4796
4797
4798
4799
4800
4801
4802
4803
4804
4805
4806
4807
4808
4809 025120      BGNDU
      025120
4810
4811
4812
4813
4814
4815
4816
4817
4818 025120      PRINTF #DROP,RO      ;REPORT UNIT THAT HAS BEEN DROPPED.
      025120      010046
      025122      012746      025144
      025126      012746      000002
      025132      010600
      025134      104417
      025136      062706      000006
4819 025142      000427
4820
4821 025144      045      101      040      DROP:  .ASCIZ/NA UNIT#D6NA DROPPED FROM FURTHER TESTING.#N/
      025147      125      116      111
      025152      124      045      104
      025155      066      045      101
      025160      040      104      122
      025163      117      120      120
      025166      105      104      040
      025171      106      122      117
      025174      115      040      106
      025177      125      122      124
      025202      110      105      122
      025205      040      124      105
      025210      123      124      111
      025213      116      107      056
      025216      045      116      000
4822
4823 025222      EDROP:  .EVEN
4824 025222      EXIT      DU
      025222      000167
      025224      000000
4825
4826

```

;=====
 ;
 ; FVTA.DRP
 ;
 ;=====

.SBTTL DROP UNIT SECTION

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

L\$DU::

;=====
 ; INSERT DROP CODE HERE. THIS CODE WILL BE EXECUTED AFTER
 ; A "DROP" COMMAND OR A "DODU" MACRO EXECUTION. THE PURPOSE
 ; OF THIS CODE IS TO DO ANY NECESSARY HOUSEKEEPING AFTER A
 ; UNIT HAS BEEN DROPPED. THIS SECTION IS OPTIONAL.
 ;=====

MOV RO, (SP)
 MOV #DROP, -(SP)
 MOV #2, (SP)
 MOV SP, RO
 TRAP C\$PNTF
 ADD #6, SP

BR EDROP ;BRANCH AROUND THE MESSAGE.

.WORD J\$JMP
 .WORD L10024 2

J12

CVDHABO DMV 11 FUNC TST PART2 MACRO M1200 10 OCT 83 11:55 PAGE 97 1

SEQ 152

DROP UNIT SECTION

4827

4828 025226

025226

025226 104453

E*DDU

L10024:

TRAP

(SD)

DROP UNIT SECTION

```

4830
4831 ;*****
4832 ;
4833 ;          FVTA.ADD
4834 ;
4835 ;*****
4836
4837
4838
4839 .SBTTL  ADD UNIT SECTION
4840
4841 ;**
4842 ; THE ADD UNIT SECTION CONTAINS ANY CODE THE PROGRAMMER WISHES
4843 ; TO BE EXECUTED IN CONJUNCTION WITH THE ADDING OF A UNIT BACK
4844 ; TO THE TEST CYCLE.
4845 ;
4846
4847 025230          BGNAU
4848                L$AU::
4849
4850 ;*****
4851 ;          INSERT ADD CODE HERE.  THIS CODE WILL BE EXECUTED AFTER
4852 ;          AN "ADD" COMMAND.  THE PURPOSE OF THIS CODE IS TO DO ANY
4853 ;          HOUSEKEEPING THAT MAY BE NECESSARY AFTER A UNIT HAS BEEN ADDED.
4854 ;          THIS SECTION IS OPTIONAL.
4855 ;*****
4856 025230          EXIT    AU
4857                .WORD    J$JMP
4858                .WORD    L10025 2 .
4859
4860 .EVEN
4861 025234          ENDAU
4862                L10025:  TRAP    C$AU

```

HARDWARE TEST

ADRA -

```

4864 .SBTTL  HARDWARE TEST          - ADRA -
4865 :
4866 :*****
4867 :                                REGISTER ADDRESS TEST -
4868 :
4869 :                                THIS TEST VERIFIES THAT THE Q-BUS CAN READ AND WRITE TO THE DMV11
4870 :                                DEVICE REGISTERS.  IF THE DMV11 DOES NOT RESPOND TO THE ACCESS
4871 :                                ATTEMPTS (IF THE DMV11 IS AT THE WRONG ADDRESS, FOR EXAMPLE) THE
4872 :                                004 BUS TIME OUT TRAP IS DETECTED BY THIS ROUTINE AND AN ERROR
4873 :                                IS REPORTED.
4874 :
4875 :*****
4876 :
4877 :
4878 025236      BGNTST
      025236
4879      000001      TNUM == TNUM + 1      ;INCREMENT THE ASSEMBLY TIME TEST COUNTER.
4880 025236 012767 000001 155062      MOV  @TNUM,TSTNUM      ;SET UP THE TEST NUMBER.      (1)
4881 025244 012767 177777 155100      MOV  @-1,CTRLCF      ;INDICATE THAT WE ARE IN A TEST.
4882 :
4883 : SET UP TO CATCH ANY 004 TRAPS WHICH OCCUR:
4884 :
4885 025252 016767 152526 155064      MOV  4,TP4VEC      ;SAVE THE EXISTING 004 TRAP VECTOR.
4886 025260 012767 024134 152516      MOV  @TP4RTN,4      ;SET 004 TRAP VECTOR TO OUR SERVICE RTN ADR.
4887 025266 005005      CLR  R5      ;CLEAR THE ERROR FLAGS.
4888 :
4889 :
4890 : SET UP FOR THE INITIAL ITERATION OF THE TEST LOOP:
4891 :
4892 025270 005004      CLR  R4      ;CLEAR THE LINE COUNTER.
4893 :
4894 :
4895 : HERE BEGINS THE LOOP TO TEST THE REGISTERS FOR A LINE.
4896 : FIRST TEST THE CSR AND SET THE INC.ADR.REG (I.A.R) FIELD.
4897 :
4898 025272 005067 155050      2$: CLR  TP4FLG      ;CLEAR THE 004 TRAP FLAG.
4899 025276 016700 154742      MOV  CSRA,R0      ;SET UP CSR AS THE CKTRAP MOVE SOURCE.
4900 025302 012701 025516      MOV  @52$,R1      ;SET UP DESTINATION LOCATION FOR CKTRAP MOVE.
4901 025306 004767 172612      JSR  PC,CKTRAP      ;MOVE AND CHECK FOR TRAP.
4902 025312 103402      BCS  4$      ;IF NO TRAP, BYPASS ERROR.
4903 025314 052705 100001      BIS  @100001,R5      ;SET FATAL READ ERROR FLAGS.
4904 025320 042767 000017 000170 4$: BIC  @17,52$      ;CLEAR THE I.A.R FIELD OF THE CSR DATA.
4905 025326 050467 000164      BIS  R4,52$      ;OR IN THE LINE COUNTER TO THE I.A.R FIELD.
4906 025332 010100      MOV  R1,R0      ;USE OLD DESTINATION FOR SOURCE OF CKTRAP MOVE.
4907 025334 016701 154704      MOV  CSRA,R1      ;SET UP CSR AS THE CKTRAP MOVE DESTINATION.
4908 025340 004767 172560      JSR  PC,CKTRAP      ;MOVE AND CHECK FOR TRAP.
4909 025344 103403      BCS  6$      ;IF NO TRAP, BYPASS ERROR.
4910 025346 052705 100002      BIS  @100002,R5      ;SET FATAL WRITE ERROR FLAGS.
4911 025352 000440      BR   40$      ;EXIT AND REPORT FATAL ERROR.
4912 :
4913 :
4914 : NOW, WE TEST EACH REGISTER FOR THIS LINE.
4915 025354 012702 000010      6$: MOV  @10,R2      ;INIT REGISTER COUNTER TO 8.
4916 025360 016767 154660 000126      MOV  CSRA,50$      ;INITIALIZE THE REGISTER POINTER.
4917 025366 012700 025514      8$: MOV  @50$,R0      ;SET UP REGISTER AS THE SOURCE FOR CKTRAP MOVE.
4918 025372 012701 025516      MOV  @52$,R1      ;SET UP LOCAL STORAGE AS THE DES FOR CKTRAP.
4919 025376 004767 172522      JSR  PC,CKTRAP      ;PERFORM THE MOVE, CHECK FOR TRAP.

```

HARDWARE TEST

- ADRA -

```

4920 025402 103402      BCS      10$      ;IF NO TRAP, BYPASS THE SETTING OF ERROR FLAGS.
4921 025404 052705 100001      BIS      $100001,R5      ;SET FATAL READ ERROR FLAGS.
4922 025410 010100      10$:      MOV      R1,R0      ;USE OLD DEST AS SRC FOR CKTRAP MOVE.
4923 025412 012701 025514      MOV      $50$,R1      ;SET UP REGISTER AS THE DEST FOR CKTRAP MOVE.
4924 025416 004767 172502      JSR      PC,CKTRAP      ;PERFORM THE MOVE, CHECK FOR TRAP.
4925 025422 103402      BCS      12$      ;IF NO TRAP BYPASS THE SETTING OF ERROR FI AGS.
4926 025424 052705 100002      BIS      $100002,R5      ;SET FATAL WRITE ERROR FLAGS.
4927 025430 005267 000060      12$:      INC      50$      ;INCREMENT THE REGISTER
4928 025434 005267 000054      INC      50$      ; POINTER BY 2.
4929 025440 005302      DEC      R2      ;COUNT THE REGISTER.
4930 025442 001351      BNE      8$      ;LOOP TO TEST THE NEXT REGISTER ADDRESS.
4931
4932
4933      ;*
4934      ; NOW WE SET UP TO TEST THE NEXT LINE, OR TO EXIT IF WE ARE DONE.
4935 025444 005204      INC      R4      ;INCREMENT THE LINE COUNTER.
4936 025446 020427 000010      CMP      R4,$NUMLNS      ;COMPARE LINE COUNTER AGAINST NUMBER OF LINES.
4937 025452 002707      BLT      2$      ;LOOP TO TEST THE NEXT LINE IF WE'RE NOT DONE.
4938
4939      ;*
4940      ; DONE CHECKING DEVICE REGISTER ADDRESSES.
4941      ; REPORT ANY ERRORS AND EXIT.
4942
4943 025454 016767 154664 152322 40$:      MOV      TP4VEC,4      ;RESTORE THE NORMAL 004 TRAP VECTOR.
4944 025462 005705      TST      R5      ;CHECK THE ERROR FLAGS.
4945 025464 100015      BPL      60$      ;EXIT ROUTINE IF NO ERRORS.
4946      ; REPORT 'DEVICE REGISTER ACCESS ERRORS"
4947 025466      ERRDF      101,EM0103,ER0101;      >>>>> ERROR #101 <<<<<.
4948      025466 104455      TRAP      C$ERDF
4949      025470 000145      .WORD      101
4950      025472 006065      .WORD      EM0103
4951      025474 016066      .WORD      ER0101
4952
4953      DODU      UNITN      ;DROP THIS UNIT FROM FUTHER TESTING.
4954      025476 016700 154536      MOV      UNITN,R0
4955      025502 104451      TRAP      C$DODU
4956 025504 005067 154642      CLR      CTRLCF      ;INDICATE NO CTRL-C ABORT FROM TEST.
4957 025510      DOCLN      ;ABORT THIS SUB PASS.
4958 025510 104444      TRAP      C$DCLN
4959 025512 000402      BR      60$      ;
4960
4961      ;*
4962      ; LOCAL STORAGE.
4963      ;-
4964 025514 000000      50$:      .WORD      0      ;STORAGE FOR THE SOURCE OR DEST OF THE CKTRAP MOVE.
4965 025516 000000      52$:      .WORD      0      ;STORAGE FOR THE SOURCE OR DEST OF THE CKTRAP MOVE.
4966 025520 005067 154626      60$:      CLR      CTRLCF      ;INDICATE THAT WE ARE NOT WITHIN A TEST.
4967 025524      ENDTST
4968
4969      L10026:      TRAP      C$ETST

```

HARDWARE TEST

MRSTA

```

4961 .SBTTL HARDWARE TEST MRSTA -
4962 ;* *****
4963 ;* MASTER RESET WITH SELFTEST TEST -
4964 ;* THIS TEST VERIFIES THAT THE MASTER RESET BIT WILL CLEAR AFTER A DEVICE
4965 ;* RESET AND THE PERFORMANCE OF THE DUT ROM BASED SELFTEST.
4966 ;*
4967 ; - *****
4968 025526 BGNST
4969 025526 TNUM == TNUM + 1 ;INCREMENT THE ASSEMBLY TIME TEST COUNTER.
4970 025526 000002 012767 000002 154572 MOV #TNUM,TSTNUM ;SET UP THE TEST NUMBER. (2)
4971 025534 012767 177777 154610 MOV #-1,CTRLCF ;INDICATE THAT WE ARE IN A TEST.
4972 025542 SETPRI #PRI05 ;ALLOW LTC INTERRUPTS.
4973 025542 012700 000240 MOV #PRI05,RO
4974 025546 104441 TRAP C$SPRI
4975 025550 012767 000001 156210 MOV #1,ERRTYP ;SET ERROR TYPE AS FATAL IN ERROR TABLE.
4976 025556 012767 006123 156206 MOV #EM0201,ERRMSG ;SET ERROR MESSAGE ADDRESS IN ERROR TABLE.
4977 025564 012767 016404 156202 MOV #ERO201,ERRBLK ;SET ERROR ROUTINE ADDRESS IN ERROR TABLE.
4978 ;*
4979 ; WAIT UP TO 5 SECONDS FOR THE DUT MASTER RESET BIT TO CLEAR.
4980 ; -
4981 025572 012701 011610 MOV #5000.,R1 ;TIME-OUT VALUE IS 5.0 SECONDS.
4982 025576 012702 000040 MOV #BIT05,R2 ;WAITING FOR MASTER RESET BIT.
4983 025602 005003 CLR R3 ;WAITING FOR BIT TO CLEAR.
4984 025604 016704 154434 MOV CSRA,R4 ;BIT IS IN THE DUT'S CSR.
4985 025610 004767 172522 JSR PC,MSLGET ;WAIT FOR DUT_CSR_MR BIT TO CLEAR.
4986 025614 103410 BCS 2$ ;SKIP TO RESET DUT IF MR CLEAR.
4987 ;*
4988 ; DUT MASTER RESET BIT DID NOT GO CLEAR. DEVICE MAY BE STUCK IN SOME
4989 ; ODD STATE. TRY TO RESET DEVICE WITH A BUS RESET.
4990 ; -
4991 025616 BRESET ;NO, TRY TO JOG DEVICE WITH BUS RESET.
4992 025616 104433 TRAP C$RESET
4993 025620 004767 174510 JSR PC,SKPSTS ;TRY TO SKIP THE SELFTEST.
4994 025624 012701 011610 MOV #5000.,R1 ;TIME OUT VALUE IS 5.0 SECONDS.
4995 025630 004767 172502 JSR PC,MSLGET ;WAIT FOR DUT_CSR_MR BIT TO CLEAR.
4996 025634 103016 BCC 4$ ;GO REPORT ERROR IF MR BIT DID NOT CLEAR.
4997 ;*
4998 ; SET THE MASTER RESET BIT AND VERIFY THAT IT CLEARS WITHIN THE PROPER TIME.
4999 ; -
5000 025636 012701 011610 2$: MOV #5000.,R1 ;TIME-OUT VALUE IS 5.0 SECONDS.
5001 025642 010214 MOV R2,(R4) ;SET THE DUT MASTER RESET BIT.
5002 025644 004767 172466 JSR PC,MSLGET ;WAIT FOR DUT_CSR_MR BIT TO CLEAR.
5003 025650 103010 BCC 4$ ;GO REPORT ERROR IF MR BIT DID NOT CLEAR.
5004 025652 012702 011610 MOV #5000.,R2 ;CALCULATE # OF MS FOR MR TO CLEAR.
5005 025656 160102 SUB R1,R2 ;GO REPORT ERROR IF MR CLEAR IMMEDIATELY.
5006 025660 001413 BEQ 6$
5007 025662 020227 000764 CMP R2,#500. ;GO REPORT ERROR IF MR CLEAR IN < 1/2 SECOND.
5008 025666 002417 BLT 8$ ;EXIT THE TEST WITHOUT ERROR.
5009 025670 000424 BR 60$
5010 ;*
5011 ; ERROR REPORTS:
5012 ;
5013 ; REPORT MR BIT WOULD NOT CLEAR AFTER A DUT RESET.
5014 025672 012767 000311 156070 4$: MOV #201.,ERRNBR ;SET THE ERROR NUMBER IN ERROR TABLE.
5015 025700 012701 006171 MOV #EM0202,R1 ;SELECT ERROR MESSAGE.
5016 025704 ERROR ;REPORT ERROR. >>>> ERROR #201 <<<<

```

HARDWARE TEST

MRSTA

```

025704 104460
5014 025706 000415      BR      60$      ;EXIT THE TEST.      TRAP      C$ERROR
5015
5016
5017 025710 012767 000312 156052 6$: ;REPORT MR BIT CLEAR IMMEDIATELY AFTER DUT RESET.
5018 025716 012701 006344      MOV      0202.,ERRNBR      ;SET THE ERROR NUMBER IN ERROR TABLE.
5019 025722      MOV      0EM0203,R1      ;SELECT ERROR MESSAGE.
025722 104460      ERROR      ;REPORT ERROR.      >>>>> ERROR 0202 <<<<<
5020 025724 000406      BR      60$      ;EXIT THE TEST.      TRAP      C$ERROR
5021
5022
5023 025726 012767 000313 156034 8$: ;REPORT MR CLEAR WITHIN 1/2 SECOND OF DUT RESET.
5024 025734 012701 006507      MOV      0203.,ERRNBR      ;SET THE ERROR NUMBER IN ERROR TABLE.
5025 025740      MOV      0EM0204,R1      ;SELECT ERROR MESSAGE.
025740 104460      ERROR      ;REPORT ERROR.      >>>>> ERROR 0203 <<<<<
5026      TRAP      C$ERROR
5027 025742      60$:      SETPRI      0PRI07      ;DISABLE ALL INTERRUPTS.
025742 012700 000340      MOV      0PRI07,R0
025746 104441      TRAP      C$SPRI
5028 025750 005067 154376      CLR      CTRLCF      ;INDICATE THAT WE COMPLETED THE TEST.
5029 025754      ENDTST
025754 104401      L10027:      TRAP      C$ETST

```

HARDWARE TEST

- MRSSTA

```

5031 .SBTTL HARDWARE TEST MRSSTA
5032 ;* .....
5033 ;* MASTER RESET WITH SKIP SELFTEST TEST
5034 ;* THIS TEST VERIFIES THAT THE MASTER RESET BIT WILL CLEAR AFTER A DEVICE
5035 ;* RESET AND THE SKIPPING OF THE DUT ROM BASED SELFTEST.
5036 ;* .....
5037 ;*
5038 025756 BGNSTST
5039 025756 000003 TNUM == TNUM + 1 ; INCREMENT THE ASSEMBLY TIME TEST COUNTER.
5040 025756 012767 000003 154342 MOV #TNUM,ISTNUM ; SET UP THE TEST NUMBER. (3)
5041 025764 012767 177777 154360 MOV #1,CIRLCF ; INDICATE THAT WE ARE IN A TEST.
5042 025772 SETPRI #PRI05 ; ALLOW LTC INTERRUPTS.
5043 025772 012700 000240 MOV #PRI05,R0
5044 025776 104441 TRAP C$PRI
5045 026000 012767 000001 155760 MOV #1,ERRTYP ; SET ERROR TYPE AS FATAL IN ERROR TABLE.
5046 026006 012767 006666 155756 MOV #EM0301,ERRMSG ; SET ERROR MESSAGE ADDRESS IN ERROR TABLE.
5047 026014 012767 016404 155752 MOV #ER0201,ERRBLK ; SET ERROR ROUTINE ADDRESS IN ERROR TABLE.
5048 ;*
5049 ; WAIT UP TO 5 SECONDS FOR THE DUT MASTER RESET BIT TO CLEAR.
5050 ;*
5051 MOV #5000.,R1 ; TIME-OUT VALUE IS 5.0 SECONDS.
5052 MOV #BIT05,R2 ; WAITING FOR MASTER RESET BIT.
5053 CLR R3 ; WAITING FOR BIT TO CLEAR.
5054 MOV CSRA,R4 ; BIT IS IN THE DUT'S CSR.
5055 JSR PC,MSLGET ; WAIT FOR DUT_CSR_MR BIT TO CLEAR.
5056 BCS 2$ ; SKIP TO RESET DUT IF MR CLEAR.
5057 ;*
5058 ; DUT MASTER RESET BIT DID NOT GO CLEAR. DEVICE MAY BE STUCK IN SOME
5059 ; ODD STATE. TRY TO RESET DEVICE WITH A BUS RESET.
5060 ;*
5061 BRESET ; NO, TRY TO JOG DEVICE WITH BUS RESET.
5062 JSR PC,SKPSTS ; TRY TO SKIP THE SELFTEST.
5063 MOV #5000.,R1 ; TIME OUT VALUE IS 5.0 SECONDS.
5064 JSR PC,MSLGET ; WAIT FOR DUT_CSR_MR BIT TO CLEAR.
5065 RCC 6$ ; GO REPORT ERROR IF MR BIT DID NOT CLEAR.
5066 ;*
5067 ; SET THE MASTER RESET BIT, TRY TO SKIP THE SELFTEST, AND VERIFY THAT THE
5068 ; MR BIT CLEARS WITHIN 1/5 SECOND.
5069 ;*
5070 2$: MOV #200.,R1 ; TIME-OUT VALUE IS 1/5 SECOND.
5071 MOV R2,(R4) ; SET THE DUT MASTER RESET BIT.
5072 JSR PC,SKPSTS ; TRY TO SKIP THE SELFTEST.
5073 JSR PC,MSLGET ; WAIT FOR DUT_CSR_MR BIT TO CLEAR.
5074 BCC 4$ ; GO FIND OUT WHAT IS WRONG IF MR NOT CLEAR.
5075 MOV #200.,R2
5076 SUB R1,R2 ; CALCULATE # OF MS FOR MR TO CLEAR.
5077 CMP R2,#10.
5078 BLT 8$ ; GO REPORT ERROR IF MR CLEAR IN < 10 MS.
5079 BR 60$ ; EXIT THE TEST WITHOUT ERROR.
5080 ;*
5081 ; MR DID NOT CLEAR WITHIN 1/5 SECOND. SEE IF IT CLEARS WITHIN 5 SECONDS.
5082 ;*
5083 4$: MOV #4800.,R1 ; TIME-OUT VALUE IS 5 SECONDS MINUS 1/5 SECOND.
5084 JSR PC,MSLGET ; WAIT FOR DUT_CSR_MR BIT TO CLEAR.
5085 BCS 10$ ; GO REPORT ERROR IF MR CLEARED FINALLY.

```

HARDWARE TEST

MRSSTA

```

5084
5085
5086
5087
5088 026136 012767 000455 155624 6$: ;REPORT MR BIT WOULD NOT CLEAR AFTER A DUT RESET.
5089 026144 012701 006171      MOV    #0301.,ERRNBR ;SET THE ERROR NUMBER IN ERROR TABLE.
5090 026150      MOV    #EM0202,R1 ;SELECT ERROR MESSAGE.
5091 026152 000415      ERROR    ;REPORT ERROR. >>>> ERROR #0301 <<<<
5092      BR      60$      TRAP    C$ERROR
5093      ;EXIT THE TEST.
5094 026154 012767 000456 155606 8$: ;REPORT MR BIT CLEAR WITHIN 10 MS AFTER DUT RESET.
5095 026162 012701 006731      MOV    #0302.,ERRNBR ;SET THE ERROR NUMBER IN ERROR TABLE.
5096 026166      MOV    #EM0302,R1 ;SELECT ERROR MESSAGE.
5097 026170 000406      ERROR    ;REPORT ERROR. >>>> ERROR #0302 <<<<
5098      BR      60$      TRAP    C$ERROR
5099      ;EXIT THE TEST.
5100 026172 012767 000457 155570 10$: ;REPORT MR CLEARED BETWEEN 1/5 SECOND AND 5 SECONDS OF DUT RESET.
5101 026200 012701 007071      MOV    #0303.,ERRNBR ;SET THE ERROR NUMBER IN ERROR TABLE.
5102 026204      MOV    #EM0303,R1 ;SELECT ERROR MESSAGE.
5103 026206      ERROR    ;REPORT ERROR. >>>> ERROR #0303 <<<<
5104 026206      BR      60$      TRAP    C$ERROR
5105 026206 012700 000340      SETPRI  #PRI07 ;DISABLE ALL INTERRUPTS.
5106 026212 104441      MOV    #PRI07,R0
5107 026214 005067 154132      TRAP    C$SPRI
5108 026220      CLR    CTRLCF ;INDICATE THAT WE COMPLETED THE TEST.
5109 026220      ENDTST
5110 026220 104401      L10030:
5111      TRAP    C$ETST

```

HARDWARE TEST

RXCHRA

```

5108 .SBTTL  HARDWARE TEST          RXCHRA
5109 ;* .....
5110 ;*          RBUF REGISTER RX CHARACTER FIELD TEST .
5111 ;*          THIS TEST VERIFIES THAT THE RX CHARACTER FIELD OF THE DUT RBUF REGISTER
5112 ;*          APPEARS TO BE FUNCTIONING CORRECTLY.  THIS TEST USES THE CODES WHICH
5113 ;*          SHOULD BE IN THE FIFO AFTER A BOARD RESET AND SKIP SELFTEST SEQUENCE.
5114 ;* .....
5115 ;
5116 026222      BGNTEST
5117 026222      SETPRI  @PRI05          ;ALLOW LTC INTERRUPTS.          T4::
5118 026222      012700  000240          MOV      @PRI05,R0
5119 026222      104441          TRAP      C$SPRI
5120 026230      012767  000004  154070      TNUM == TNUM + 1          ;INCREMENT THE ASSEMBLY TIME TEST COUNTER.
5121 026236      012767  177777  154106      MOV      @TNUM,TSTNUM          ;SET UP THE TEST NUMBER.          (4)
5122 026244      012767  000001  155514      MOV      @-1,CTRLCF          ;INDICATE THAT WE ARE WITHIN A TEST.
5123 026252      012767  007250  155512      MOV      @1,ERRPTYP          ;SET ERROR TYPE AS FATAL IN ERROR TABLE.
5124 026260      012767  016404  155506      MOV      @EM0401,ERRMSG          ;SET ERROR MESSAGE ADDRESS IN ERROR TABLE.
5125 ;*          ;SET THE DUT CSR MASTER RESET (MR) BIT, PERFORM THE SKIP SELFTEST SEQUENCE,
5126 ;*          ;AND WAIT UP TO 5 SECONDS FOR THE MR BIT TO CLEAR.
5127 ;
5128 026266      012701  011610          MOV      @5000.,R1          ;TIME-OUT VALUE IS 5.0 SECONDS.
5129 026272      012702  000040          MOV      @BIT05,R2          ;WAITING FOR MASTER RESET BIT.
5130 026276      005003          CLR      R3          ;WAITING FOR BIT TO CLEAR.
5131 026300      016704  153740          MOV      CSRA,R4          ;BIT IS IN THE DUT'S CSR.
5132 026304      010214          MOV      R2,(R4)          ;SET THE DUT MASTER RESET BIT.
5133 026306      004767  174022          JSR      PC,SKPSTS          ;SKIP THE SELFTEST.
5134 026312      004767  172020          JSR      PC,MSLGET          ;WAIT FOR DUT_CSR_MR BIT TO CLEAR.
5135 026316      103015          BCC      4$          ;GO REPORT ERROR IF MR DID NOT CLEAR.
5136 ;*
5137 ;* READ 6 CHARACTERS FROM THE DUT AND VERIFY THAT THEY ARE VALID SELFTEST
5138 ;* CODES.
5139 ;
5140 026320      012400          MOV      (R4)+,R0          ;INCREMENT POINTER TO POINT TO DUT RBUF REGSTR.
5141 026322      012701  000006          MOV      @6,R1          ;INITIALIZE THE LOOP COUNTER.
5142 026326      011402          2$: MOV      (R4),R2          ;READ A CHARACTER FROM THE DUT RBUF REGISTER.
5143 026330      010200          MOV      R2,R0
5144 026332      042700  177476          BIC      @177476,R0          ;REMOVE ALL BUT BITS SPECIFIC TO SELFTEST CODE.
5145 026336      020027  000201          CMP      R0,@201          ;CHECK THAT BITS 0,6, AND 7 ARE CORRECT.
5146 026342      001012          BNE      6$          ;GO REPORT ERROR IF CODE IS NOT SELFTEST CODE.
5147 026344      005301          DEC      R1          ;COUNT THIS LOOP ITERATION.
5148 026346      001367          BNE      2$          ;LOOP IF NOT ALL LINES DONE.
5149 026350      000415          BR       60$          ;EXIT TEST, NO ERROR FOUND.
5150 ;
5151 ;* ERROR REPORTS:
5152 ;
5153 ;
5154 ;* ;REPORT MR BIT WOULD NOT CLEAR AFTER 3 DUT RESET.
5155 026352      012767  000621  155410  4$: MOV      @0401.,ERRNBR          ;SET THE ERROR NUMBER IN ERROR TABLE
5156 026360      012701  006171          MOV      @EM0202,R1          ;SELECT ERROR MESSAGE.
5157 026364          104460          ERROR          ;REPORT ERROR.          >>>>> ERROR @0401 <<<<<
5158 026366      000406          BR       60$          ;EXIT THE TEST.          TRAP      C$ERROR
5159 ;
5160 ;REPORT IMPROPER CODE FOUND IN DUT RBUF AFTER RESET (SKIP SELFTEST).

```

HARDWARE TEST

- RXCHRA -

5161	026370	012767	000622	155372	64:	MOV	#0402.,ERRNBR	;SET THE ERROR NUMBER IN ERROR TABLE.
5162	026376	012701	007317			MOV	#FM0402,R1	;SELECT ERROR MESSAGE.
5163	026402					ERROR		;REPORT ERROR. >>>> ERROR #0402 <<<<
	026402	104460						TRAP C\$ERRPR
5164								
5165	026404				604:	SETPRI	#PRI07	;DISABLE ALL INTERRUPTS.
	026404	012700	000340					MOV #PRI07,RO
	026410	104441						TRAP C\$SPRI
5166	026412	005067	153734			CLR	CTRLCF	;INDICATE THAT WE COMPLETED THE TEST.
5167	026416					ENDTST		
	026416							L10031:
	026416	104401						TRAP C\$ETST

HARDWARE TEST

RXFFDA -

```

5169 .SBTTL  HARDWARE TEST          - RXFFDA -
5170 ;* *****
5171 ;*          RBUF REGISTER RX FLAG FIELD TEST
5172 ;*          THIS TEST VERIFIES THAT THE FIELD OF 3 FLAG BITS IN THE RBUF READS
5173 ;*          AS ALL ONES WHEN THE SELFTEST CODES ARE BEING READ FROM THE DUT
5174 ;*          AFTER A BOARD RESET AND SKIP SELFTEST SEQUENCE.
5175 ;*
5176 ;* *****
5177 026420 BGNTEST
5178 026420          SETPRI  #PRI05          ;ALLOW LTC INTERRUPTS.          TS::
026420          012700 000240          MOV          #PRI05,R0
026424          104441          TRAP          C$SPRI
5179          000005          TNUM == TNUM + 1          ;INCREMENT THE ASSEMBLY TIME TEST COUNTER.
5180 026426 012767 000005 153672          MOV          #TNUM,TSTNUM          ;SET UP THE TEST NUMBER.          (5)
5181 026434 012767 177777 153710          MOV          #-1,CTRLCF          ;INDICATE THAT WE ARE WITHIN A TEST.
5182 026442 012767 000001 155316          MOV          #1,ERRTYP          ;SET ERROR TYPE AS FATAL IN ERROR TABLE.
5183 026450 012767 007467 155314          MOV          #EM0501,ERRMSG          ;SET ERROR MESSAGE ADDRESS IN ERROR TABLE.
5184 026456 012767 016404 155310          MOV          #ER0201,ERRBLK          ;SET ERROR ROUTINE ADDRESS IN ERROR TABLE.
5185
5186 ;*
5187 ; SET THE DUT CSR MASTER RESET (MR) BIT, PERFORM THE SKIP SELFTEST SEQUENCE.
5188 ; AND WAIT UP TO 5 SECONDS FOR THE MR BIT TO CLEAR.
5189 026464 012701 011610          MOV          #5000.,R1          ;TIME-OUT VALUE IS 5.0 SECONDS.
5190 026470 012702 000040          MOV          #BIT05,R2          ;WAITING FOR MASTER RESET BIT.
5191 026474 005003          CLR          R3          ;WAITING FOR BIT TO CLEAR.
5192 026476 016704 153542          MOV          CSRA,R4          ;BIT IS IN THE DUT'S CSR.
5193 026502 010214          MOV          R2,(R4)          ;SET THE DUT MASTER RESET BIT.
5194 026504 004767 173624          JSR          PC,SKPSTS          ;SKIP THE SELFTEST.
5195 026510 004767 171622          JSR          PC,MSLGET          ;WAIT FOR DUT_CSR_MR BIT TO CLEAR.
5196 026514 103013          BCC          4$          ;GO REPORT ERROR IF MR DID NOT CLEAR.
5197
5198 ;*
5199 ; READ 8 CHARACTERS FROM THE DUT AND VERIFY THAT ALL 3 RX ERROR FLAGS ARE
5200 ; SET FOR EACH CHARACTERS.
5201 026516 012400          MOV          (R4)+,R0          ;INCREMENT POINTER TO POINT TO DUT RBUF REGSTR.
5202 026520 012701 000010          MOV          #8.,R1          ;INITIALIZE THE LOOP COUNTER.
5203 026524 011402          MOV          (R4),R2          ;READ A CHARACTER FROM THE DUT RBUF REGISTER.
5204 026526 012700 070000          MOV          #70000,R0
5205 026532 040200          BIC          R2,R0          ;CALCULATE BIT MAP OF CLEAR RX ERROR FLAGS.
5206 026534 001012          BNE          6$          ;GO REPORT ERROR IF NOT ALL RX ERROR FLAGS SET.
5207 026536 005301          DEC          R1          ;COUNT THIS LOOP ITERATION.
5208 026540 001371          BNE          2$          ;LOOP IF NOT ALL LINES DONE.
5209 026542 000415          BR          60$          ;EXIT TEST, NO ERROR FOUND.
5210
5211 ;*
5212 ; ERROR REPORTS:
5213 ;
5214 ;REPORT MR BIT WOULD NOT CLEAR AFTER A DUT RESET.
5215 026544 012767 000765 155216 4$:          MOV          #0501.,ERRNBR          ;SET THE ERROR NUMBER IN ERROR TABLE.
5216 026552 012701 006171          MOV          #EM0202,R1          ;SELECT ERROR MESSAGE.
5217 026556          ERROR          ;REPORT ERROR.          >>>>> ERROR #0501 <<<<<
026556          104460          TRAP          C$ERROR
5218 026560 000406          BR          60$          ;EXIT THE TEST.
5219
5220 ;REPORT ONE OR MORE RX ERROR FLAGS FOUND SET WITH SELFTEST CODE.
5221 026562 012767 000766 155200 6$:          MOV          #0502.,ERRNBR          ;SET THE ERROR NUMBER IN ERROR TABLE.

```

HARDWARE TEST

- RXFFDA -

```

5222 026570 012701 007535          MOV    #EM0502,R1      ;SELECT ERROR MESSAGE.
5223 026574          ERROR          ;REPORT ERROR.        >>>> ERROR #0502 <<<<
                                TRAP    C$ERROR
5224
5225 026576          604:  SETPRI  #PRI07          ;DISABLE ALL INTERRUPTS.
                                MOV      #PRI07,R0
                                TRAP    C$SPRI
                                026576 012700 000340
                                026602 104441
5226 026604 005067 153542          CLR    CTRLCF        ;INDICATE THAT WE COMPLETED THE TEST.
5227 026610          ENDTST
                                L10032:
                                TRAP    C$ETST
                                026610 104401

```

HARDWARE TEST

RDAA -

```
5229 .SBTTL HARDWARE TEST RDAA -
5230 ;* *****
5231 ;* - CSR RX DATA AVAILABLE BIT TEST
5232 ;* THIS TEST VERIFIES THAT THE DUT CSR RX DATA AVAILABLE BIT IS SET BY THE
5233 ;* INCLUSION OF THE SELFTEST CODES IN THE DUT FIFO AND THAT THE BIT CLEARS
5234 ;* AFTER THE FIFO HAS BEEN EMPTIED.
5235 ;*
5236 ;* *****
5237 026612 BGNTST
5238 026612 SETPRI #PRI05 ;ALLOW LTC INTERRUPTS. T6::
5239 026612 012700 000240 MOV #PRI05,R0
5240 026612 104441 TRAP C$SPRI
5241 026612 000006 TNUM == TNUM + 1 ;INCREMENT THE ASSEMBLY TIME TEST COUNTER.
5242 026620 012767 000006 153500 MOV #TNUM,T$TNUM ;SET UP THE TEST NUMBER. (6)
5243 026626 012767 177777 153516 MOV #-1,CTRLCF ;INDICATE THAT WE ARE WITHIN A TEST.
5244 026634 012767 000001 155124 MOV #1,ERRTYP ;SET ERROR TYPE AS FATAL IN ERROR TABLE.
5245 026642 012767 010071 155122 MOV #EM0601,ERRMSG ;SET ERROR MESSAGE ADDRESS IN ERROR TABLE.
5246 026650 012767 016404 155116 MOV #ER0201,ERRBLK ;SET ERROR ROUTINE ADDRESS IN ERROR TABLE.
5247 ;*
5248 ; SET THE DUT CSR MASTER RESET (MR) BIT, PERFORM THE SKIP SELFTEST SEQUENCE,
5249 ; AND WAIT UP TO 5 SECONDS FOR THE MR BIT TO CLEAR.
5250 026656 012701 011610 MOV #5000.,R1 ;TIME-OUT VALUE IS 5.0 SECONDS.
5251 026662 012702 000040 MOV #BIT05,R2 ;WAITING FOR MASTER RESET BIT.
5252 026666 005003 CLR R3 ;WAITING FOR BIT TO CLEAR.
5253 026670 016704 153350 MOV CSRA,R4 ;BIT IS IN THE DUT'S CSR.
5254 026674 010214 MOV R2,(R4) ;SET THE DUT MASTER RESET BIT.
5255 026676 004767 173432 JSR PC,SKPSTS ;SKIP THE SELFTEST.
5256 026702 004767 171430 JSR PC,MSLGET ;WAIT FOR DUT_CSR MR BIT TO CLEAR.
5257 026706 103016 BCC 4$ ;GO REPORT ERROR IF MR DID NOT CLEAR.
5258 ;*
5259 ; CHECK THAT THE RX DATA AVAILABLE BIT IS SET.
5260 026710 032714 000200 BIT #BIT7,(R4) ;TEST THE DUT RX.DATA.AVAIL BIT.
5261 026714 001422 BEQ 6$ ;GO REPORT ERROR IF BIT IS NOT SET.
5262 ;*
5263 ; READ CHARACTERS FROM THE DUT RX FIFO AND WAIT FOR RX.DATA.AVAIL TO GO CLEAR.
5264 ;*
5265 026716 012705 001130 MOV #600.,R5 ;ALLOW READING 600 CHARS BEFORE ERROR.
5266 026722 010403 MOV R4,R3
5267 026724 012300 MOV (R3),R0 ;CALCULATE THE RBUF ADDRESS.
5268 026726 011300 MOV (R3),R0 ;READ A CHARACTER FROM THE RX FIFO.
5269 026730 032714 000200 BIT #BIT7,(R4) ;TEST THE DUT RX.DATA.AVAIL BIT.
5270 026734 001427 BEQ 60$ ;EXIT TEST WITHOUT ERROR IF RX.DATA.AVAIL CLR.
5271 026736 005305 DEC R5 ;COUNT THE CHARACTER JUST READ.
5272 026740 001372 BNE 2$ ;LOOP IF NOT TOO MANY CHARS READ FROM FIFO.
5273 026742 000416 BR 8$ ;GO REPORT ERROR IF RX.DATA.AVAIL WOULDN'T CLR.
5274 ;*
5275 ; ERROR REPORTS:
5276 ;
5277 ;
5278 ;REPORT MR BIT WOULD NOT CLEAR AFTER A DUT RESET.
5279 026744 012767 001131 155016 4$: MOV #0601.,ERRNBR ;SET THE ERROR NUMBER IN ERROR TABLE.
5280 026752 012701 006171 MOV #EM0202,R1 ;SELECT ERROR MESSAGE.
5281 026756 104460 ERROR ;REPORT ERROR. >>>>> ERROR #0601 <<<<<
; TRAP C$ERROR
```

HARDWARE TEST

- RDAA

```

5282 026760 000415          BR      60$          ;EXIT THE TEST.
5283
5284
5285 026762 012767 001132 155000 6$: ;REPORT THAT RX.DATA.AVAIL BIT WAS NOT SET AFTER A RESET COMPLETION.
5286 026770 012701 010125          MOV      #0602.,ERRNBR ;SET THE ERROR NUMBER IN ERROR TABLE.
5287 026774          MOV      #EM0602,R1 ;SELECT ERROR MESSAGE.
          ERROR          ;REPORT ERROR.          >>>>> ERROR #0602 <<<<<
          026774 104460          TRAP      C$ERROR
5288 026776 000406          BR      60$          ;EXIT THE TEST.
5289
5290          ;REPORT THAT RX.DATA.AVAIL BIT COULD NOT BE CLEARED BY PURGING FIFO.
5291 027000 012767 001133 154762 8$: MOV      #0603.,ERRNBR ;SET THE ERROR NUMBER IN ERROR TABLE.
5292 027006 012701 010305          MOV      #EM0603,R1 ;SELECT ERROR MESSAGE.
5293 027012          ERROR          ;REPORT ERROR.          >>>>> ERROR #0603 <<<<<
          027012 104460          TRAP      C$ERROR
5294
5295 027014          60$: SETPRI  #PRI07          ;DISABLE ALL INTERRUPTS.
          027014 012700 000340          MOV      #PRI07,R0
          027020 104441          TRAP      C$SPRI
5296 027022 005067 153324          CLR      CTRLCF          ;INDICATE THAT WE COMPLETED THE TEST.
5297 027026          ENDTST
          027026          L10033:
          027026 104401          TRAP      C$ETS

```

HARDWARE TEST

- RDVA -

```

5299 .SBTTL HARDWARE TEST RDVA -
5300 ;* *****
5301 ;* RBUF RX DATA VALID BIT TEST
5302 ;* THIS TEST VERIFIES THAT THE DUT RBUF RX DATA VALID BIT IS SET BY THE
5303 ;* INCLUSION OF THE SELFTEST CODES IN THE DUT FIFO AND THAT THE BIT CLEARS
5304 ;* AFTER THE FIFO HAS BEEN EMPTIED.
5305 ;* *****
5306 ;
5307 027030 BGNTST
5308 027030 SETPRI @PRIOS ;ALLOW LTC INTERRUPTS. T7:
5309 027030 012700 000240 MOV @PRIOS,R0
5310 027034 104441 TRAP C$SPRI
5311 000007 TNUM == TNUM + 1 ;INCREMENT THE ASSEMBLY TIME TEST COUNTER.
5312 027036 012767 000007 153262 MOV @TNUM,T$TNUM ;SET UP THE TEST NUMBER. (7)
5313 027044 012767 177777 153300 MOV @-1,CTRLCF ;INDICATE THAT WE ARE WITHIN A TEST.
5314 027052 012767 000001 154706 MOV @1,ERRTYP ;SET ERROR TYPE AS FATAL IN ERROR TABLE.
5315 027060 012767 010470 154704 MOV @EM0701,ERRMSG ;SET ERROR MESSAGE ADDRESS IN ERROR TABLE.
5316 027066 012767 016404 154700 MOV @ER0201,ERRBLK ;SET ERROR ROUTINE ADDRESS IN ERROR TABLE.
5317 ;*
5318 ; SET THE DUT CSR MASTER RESET (MR) BIT, PERFORM THE SKIP SELFTEST SEQUENCE,
5319 ; AND WAIT UP TO 5 SECONDS FOR THE MR BIT TO CLEAR.
5320 027074 012701 011610 MOV @5000.,R1 ;TIME-OUT VALUE IS 5.0 SECONDS.
5321 027100 012702 000040 MOV @BIT05,R2 ;WAITING FOR MASTER RESET BIT.
5322 027104 005003 CLR R3 ;WAITING FOR BIT TO CLEAR.
5323 027106 016704 153132 MOV CSRA,R4 ;BIT IS IN THE DUT'S CSR.
5324 027112 010214 MOV R2,(R4) ;SET THE DUT MASTER RESET BIT.
5325 027114 004767 173214 JSR PC,SKPSTS ;SKIP THE SELFTEST.
5326 027120 004767 171212 JSR PC,MSLGET ;WAIT FOR DUT_CSR_MR BIT TO CLEAR.
5327 027124 103012 BCC 4$ ;GO REPORT ERROR IF MR DID NOT CLEAR.
5328 ;*
5329 ; CHECK THAT THE RX DATA VALID BIT IS SET.
5330 027126 012400 MOV (R4)+,R0 ;INCREMENT POINTER TO PNT TO DUT RBUF REG.
5331 027130 005714 TST (R4) ;TEST THE DUT RX.DATA.VALID BIT.
5332 027132 100016 BPL 6$ ;GO REPORT ERROR IF BIT IS NOT SET.
5333 ;*
5334 ; READ CHARACTERS FROM THE DUT RX FIFO AND WAIT FOR RX.DATA.VALID TO GO CLEAR.
5335 ;*
5336 027134 012705 001130 MOV @600.,R5 ;ALLOW READING 600 CHARS BEFORE ERROR.
5337 027140 011400 2$: MOV (R4),R0 ;READ A CHARACTER FROM THE RX FIFO.
5338 027142 100027 BPL 60$ ;EXIT TEST WITHOUT ERROR IF BIT IS CLEAR.
5339 027144 005305 DEC R5 ;COUNT THE CHARACTER JUST READ.
5340 027146 001374 BNE 2$ ;LOOP IF NOT TOO MANY CHARS READ FROM FIFO.
5341 027150 000416 BR 8$ ;GO REPORT ERROR IF RX.DATA.VALID WOULDN'T CLR.
5342 ;*
5343 ; ERROR REPORTS:
5344 ;
5345 ;
5346 ;REPORT MR BIT WOULD NOT CLEAR AFTER A DUT RESET.
5347 027152 012767 001275 154610 4$: MOV @0701.,ERRNBR ;SET THE ERROR NUMBER IN ERROR TABLE.
5348 027160 012701 006171 MOV @EM0202,R1 ;SELECT ERROR MESSAGE.
5349 027164 ERROR ;REPORT ERROR. >>>>> ERROR @0701 <<<<<
5350 027166 104460 TRAP C$ERROR
5351 000415 BR 60$ ;EXIT THE TEST.

```

- RDVA -

5352						;REPORT THAT RX.DATA.VALID BIT WAS NOT SET AFTER A RESET COMPLETION.			
5353	027170	012767	001276	154572	6\$:	MOV #0702.,ERRNBR		;SET THE ERROR NUMBER IN ERROR TABLE.	
5354	027176	012701	010525			MOV #EM0702,R1		;SELECT ERROR MESSAGE.	
5355	027202					ERROR		;REPORT ERROR.	>>>> ERROR #0702 <<<<
	027202	104460							TRAP C\$ERRCR
5356	027204	000406				BR 60\$		;EXIT THE TFST.	
5357									
5358						;REPORT THAT RX.DATA.VALID BIT COULD NOT BE CLEARED BY PURGING FIFO.			
5359	027206	012767	001277	154554	8\$:	MOV #0703.,ERRNBR		;SET THE ERROR NUMBER IN ERROR TABLE.	
5360	027214	012701	010705			MOV #EM0703,R1		;SELECT ERROR MESSAGE.	
5361	027220					ERROR		;REPORT ERROR.	>>>> ERROR #0703 <<<<
	027220	104460							TRAP C\$ERROR
5362									
5363	027222				60\$:	SETPRI #PRI07		;DISABLE ALL INTERRUPTS.	
	027222	012700	000340						MOV #PRI07,R0
	027226	104441							TRAP C\$SPRI
5364	027230	005067	153116			CLR CTRLCF		;INDICATE THAT WE COMPLETED THE TEST.	
5365	027234					ENDTST			L10034:
	027234	104401							TRAP C\$E*ST

HARDWARE TEST

- RLNA -

```

5367 .SBTTL HARDWARE TEST - RLNA -
5368 ;* *****
5369 ;*          RBUF RX LINE NUMBER FIELD TEST
5370 ;*          THIS TEST VERIFIES THAT THE DUT RBUF RX LINE NUMBER FIELD IS WORKING
5371 ;*          CORRECTLY BY UTILIZING THE SELFTEST CODES WHICH ARE PUT IN THE RX
5372 ;*          FIFO AFTER A BOARD RESET.
5373 ;*
5374 ;* *****
5375 ;*          BGNST
5376 027236          SETPRI  #PRI05          ;ALLOW LTC INTERRUPTS.          T8::
5377 027236          MOV      #PRI05,R0
5378 027236          TRAP    C$SPRI
5379 027242 012700 000240          TNUM == TNUM + 1          ;INCREMENT THE ASSEMBLY TIME TEST COUNTER.
5380 027242 012700 000010          MOV      #TNUM,TSTNUM      ;SET UP THE TEST NUMBER. (8)
5381 027242 012700 153054          MOV      #-1,CTRLCF        ;INDICATE THAT WE ARE WITHIN A TEST.
5382 027242 012700 153072          MOV      #1,ERRTYP         ;SET ERROR TYPE AS FATAL IN ERROR TABLE.
5383 027242 012700 154500          MOV      #EM0801,ERRMSG      ;SET ERROR MESSAGE ADDRESS IN ERROR TABLE.
5384 027242 012700 154476          ;*
5385 027242 012700 000010          ; SET THE DUT CSR MASTER RESET (MR) BIT, PERFORM THE SKIP SELFTEST SEQUENCE,
5386 027242 012700 177777          ; AND WAIT UP TO 5 SECONDS FOR THE MR BIT TO CLEAR.
5387 027242 012700 153054          ;*
5388 027242 012700 153072          MOV      #5000.,R1          ;TIME-OUT VALUE IS 5.0 SECONDS.
5389 027242 012700 153072          MOV      #BIT05,R2          ;WAITING FOR MASTER RESET BIT.
5390 027242 012700 153072          CLR      R3                ;WAITING FOR BIT TO CLEAR.
5391 027242 012700 153072          MOV      CSRA,R4            ;BIT IS IN THE DUT'S CSR.
5392 027242 012700 153072          MOV      R2,(R4)            ;SET THE DUT MASTER RESET BIT.
5393 027242 012700 153072          JSR      PC,SKPSTS           ;SKIP THE SELFTEST.
5394 027242 012700 153072          JSR      PC,MSLGET           ;WAIT FOR DUT_CSR_MR BIT TO CLEAR.
5395 027242 012700 153072          BCC      4$                 ;GO REPORT ERROR IF MR DID NOT CLEAR.
5396 027242 012700 153072          ;*
5397 027242 012700 153072          ; READ CHARACTERS FROM THE DUT RX FIFO AND VERIFY THAT THE LINE NUMBERS ARE
5398 027242 012700 153072          ; CORRECT.
5399 027242 012700 153072          ; ONE CHARACTER IS READ FROM THE FIFO FOR EACH POSSIBLE LINE ON THE DUT.
5400 027242 012700 153072          ;*
5401 027242 012700 153072          CLR      R1                ;CLEAR THE LINE COUNTER.
5402 027242 012700 153072          MOV      (R4)+,R0           ;INCREMENT POINTER TO PNT TO THE DUT RBUF REG.
5403 027242 012700 153072          MOV      (R4),R2            ;READ A CHARACTER FROM THE DUT RX FIFO.
5404 027242 012700 153072          MOV      R2,R3
5405 027242 012700 153072          SWAB    R3
5406 027242 012700 153072          BIC      #177760,R3         ;REMOVE ALL BUT LINE NUMBER BITS.
5407 027242 012700 153072          CMP      R3,R1              ;COMPARE WITH EXPECTED LINE NUMBER.
5408 027242 012700 153072          BNE      6$                 ;GO REPORT ERROR IF LINE NUMBERS DON'T MATCH.
5409 027242 012700 153072          INC      R1                ;INCREMENT THE EXPECTED LINE NUMBER.
5410 027242 012700 153072          CMP      R1,#NUMLNS         ;COMPARE WITH NUMBER OF LINES ON DUT.
5411 027242 012700 153072          BNE      2$                 ;LOOP UNTIL CODES FOR ALL LINES ARE READ.
5412 027242 012700 153072          BR       60$                ;EXIT TEST WITHOUT ERROR.
5413 027242 012700 153072          ;*
5414 027242 012700 153072          ; ERROR REPORTS:
5415 027242 012700 153072          ;*
5416 027242 012700 153072          ;REPORT MR BIT WOULD NOT CLEAR AFTER A DUT RESET.
5417 027242 012700 153072          4$: MOV      #0801.,ERRNBR    ;SET THE ERROR NUMBER IN ERROR TABLE.
5418 027242 012700 153072          MOV      #ER0503,ERRBLK     ;SET ERROR ROUTINE ADDRESS IN ERROR TABLE.
5419 027242 012700 153072          MOV      #EM0202,R1         ;SELECT ERROR MESSAGE.
5420 027242 012700 153072          ERROR    ;REPORT ERROR.          >>>> ERROR #0801 <<<<
5421 027242 012700 153072          TRAP    C$ERROR

```

HARDWARE TEST

- RLNA -

```

5420 027404 000411          BR      60$          ;EXIT THE TEST.
5421
5422          ;REPORT THAT RX LINE NUMBER FIELD IS WRONG FOR SELFTEST CODE.
5423 027406 012767 001442 154354 6$:      MOV      #0802.,ERRNBR      ;SET THE ERROR NUMBER IN ERROR TABLE.
5424 027414 012767 016404 154352          MOV      #ER0201,ERRBLK      ;SET ERROR ROUTINE ADDRESS IN ERROR TABLE.
5425 027422 012701 011130          MOV      #EM0802,R1      ;SELECT ERROR MESSAGE.
5426 027426          ERROR          ;REPORT ERROR.          >>>>> ERROR #0802 <<<<<
          027426 104460          TRAP      C$ERROR
5427
5428 027430          60$:      SETPRI   #PRI07          ;DISABLE ALL INTERRUPTS.
          027430 012700 000340          MOV      #PRI07,R0
          027434 104441          TRAP      C$SPRI
5429 027436 005067 152710          CLR      CTRLCF          ;INDICATE THAT WE COMPLETED THE TEST.
5430 027442          ENDTST
          027442          L10035:
          027442 104401          TRAP      C$ETST

```

HARDWARE TEST

BMPCHK

```

5432 .SBTTL HARDWARE TEST BMPCHK
5433 .....
5434 BMP CHECK TEST
5435 THIS TEST IS USED TO VERIFY THAT THE DUT DOES NOT IMMEDIATELY FAIL
5436 THE ON BOARD BACKGROUND MONITOR PROGRAM, AND HENCE INVALIDATE
5437 SUCCEEDING TESTS.
5438 THIS TEST LOOKS FOR BMP CODES IN THE FIFO FOR A SET PERIOD IMMEDIATELY
5439 AFTER THE SELF TEST IS SKIPPED.
5440 ANY BMP CODES THAT ARE FOUND ARE SAVED ON THE QUEUE AND ARE ALSO
5441 REPORTED IN THIS TEST.
5442 .....
5443 BGNTEST
5444 027444 SETPRI #PRIOS ;ALLOW LTC INTERRUPTS. T9::
5445 027444 MOV #PRIOS,R0
5446 027450 TRAP CSPTI
5447 000011 TNUM -- TNUM + 1 ;INCREMENT THE ASSEMBLY TIME TEST COUNTER.
5448 012767 MOV #TNUM,TSTNUM ;SET UP THE TEST NUMBER. (9)
5449 012767 MOV #1,CTRLCF ;INDICATE THAT WE ARE WITHIN A TEST.
5450 012767 MOV #1,ERRTYP ;SET ERROR TYPE AS FATAL IN ERROR TABLE.
5451 001605 MOV #0901,ERRNBR ;SET THE ERROR NUMBER.
5452 ;
5453 ; WAIT UP TO 3 SECONDS FOR THE DUT MASTER RESET BIT TO CLEAR.
5454 ; IF TIME-OUT OCCURS, THEN EXIT THIS TEST.
5455 027502 MOV #3000,R1 ;TIME-OUT VALUE IS 3.0 SECONDS.
5456 027506 MOV #BIT05,R2 ;WAITING FOR MASTER RESET BIT.
5457 005003 CLR R3 ;WAITING FOR BIT TO CLEAR.
5458 027514 MOV CSRA,R4 ;BIT IS IN THE DUT'S CSR.
5459 027520 JSR PC,MSLGET ;WAIT FOR DUT CSR MR BIT TO CLEAR.
5460 027524 BCC 508 ;ABORT THE TEST IF MR DID NOT CLEAR.
5461 ;
5462 ; RESET THE DUT, SKIP THE SELF TEST.
5463 ;
5464 027526 MOV R2,(R4) ;SET THE DUT MASTER RESET BIT.
5465 027530 JSR PC,SKPSTS ;WRITE THE SKIP SELFTEST CODES TO THE DUT.
5466 ;
5467 ; WAIT FOR MASTER RESET TO CLEAR. DELAY FOR 500 MILLI-SECS BEFORE PURGING
5468 ; THE FIFO.
5469 ;
5470 027534 MOV #500,R4 ;TIME OUT VALUE IS 500 MILLI SECONDS.
5471 027540 JSR PC,DELAY ;WAIT FOR BMP TO BEGIN EXECUTION.
5472 027544 JSR PC,PUFIFO ;PURGE THE FIFO, SAVING ANY BMP CODES.
5473 027550 BCC 508 ;ABORT THE TEST IF THE FIFO DID NOT CLEAR.
5474 ;
5475 ; REPORT THE ERROR IF ANY BMP CODES WERE FOUND.
5476 ;
5477 027552 MOV BMPQOP,R2 ;GET THE CONTENTS OF THE POINTER TO THE BMP Q.
5478 027556 MOV #BMPQOB,R3 ;GET THE START ADDRESS OF THE QUEUE.
5479 027562 CMP R2,R3 ;SEE IF THE POINTER HAS MOVED FROM THE BASE.
5480 027564 BEQ 608 ;EXIT NO CODES IN THE QUEUE.
5481 ;
5482 ; THERE IS AT LEAST ONE BMP CODE IN THE QUEUE. REPORT THE ERROR.
5483 ;
5484 ;REPORT ERROR BMP CODE FOUND IN TEST NN, BMP CODE:NNNNNN
5485 027566 MOV #EM0902,R1 ;PASS THE MESSAGE TO BE REPORTED.

```

HARDWARE TEST

- BMPCHK

```

5486 027572          ERROR 0001,FM0701,ER9501 ;          ..... ERROR #0901 .....
      027572 104455
      027574 001605
      027576 011203
      027600 017446
5487 027602 000405          BR          608
5488
5489 027604 012767 001606 154156 508:      MOV      @902.,FRRNBR      ;SET ..... ERROR #0902 .....
5490 027612 004767 172644          JSR      PC,TSABRT      ;REPORT NON-TEST RELATED ERROR.
5491
5492 027616          608:      SETPRI  @PRI07      ;DISABLE ALL INTERRUPTS.
      027616 012700 000340
      027622 104441
5493 027624 005067 152522          CLR      CTRLCF      ;INDICATE THAT WE COMPLETED THE TEST.
5494 027630          FNDTST
      027630 104401

```

TRAP C8F00F
 .WORD 901
 .WORD FM0901
 .WORD ER9501
 MOV @PRI07,PC
 TRAP C8F00F
 L10036: TRAP C8E757

HARDWARE TEST

- BMDCHK -

```

5496
5497
5498
5499
5500
5501
5502
5503
5504
5505
5506 027632
      027632
5507 027632
      027632 012700 000240
      027636 104441
5508      000012
5509 027640 012767 000012 152460
5510 027646 012767 177777 152476
5511 027654 012767 000001 154104
5512 027662 012767 011270 154102
5513 027670 012767 016466 154076
5514
5515
5516
5517
5518 027676 012701 005670
5519 027702 012702 000040
5520 027706 005003
5521 027710 016704 152330
5522 027714 004767 170416
5523 027720 103037
5524
5525
5526
5527
5528
5529 027722 012701 000062
5530 027726 010214
5531 027730 004767 172400
5532 027734 004767 170376
5533 027740 103011
5534 027742 020127 000050
5535 027746 003015
5536
5537
5538
5539
5540
5541
5542 027750 012767 001753 154012
5543 027756 004767 171636
5544 027762 000423
5545
5546
5547
5548
5549 027764 012767 001751 153776

```

```

.SBT'L HARDWARE TEST - SKSELF
;*****
; - SKIP SELF TEST TEST
; THIS TEST VERIFIES THAT THE DUT SKIPS THE SELF TEST WITHIN THE
; TIME ALLOWED, AND THAT THE FIFO CONTAINS THE CORRECT CODES AFTER ITS
; COMPLETION.
;*****
      BGMTST
;*****
      SETPRI #PRI05          ;ALLOW LTC INTERRUPTS.
;*****
      MOV #PRI05,R0
      TRAP C:SPRI
      TNUM == TNUM + 1      ;INCREMENT THE ASSEMBLY TIME TEST COUNTER.
      MOV #TNUM,TSTNUM      ;SET UP THE TEST NUMBER. (10)
      MOV #-1,CTRLCF        ;INDICATE THAT WE ARE WITHIN A TEST.
      MOV #1,ERRTYP         ;SET ERROR TYPE AS FATAL IN ERROR TABLE.
      MOV #EM1001,ERRMSG     ;SET ERROR MESSAGE ADDRESS IN ERROR TABLE.
      MOV #ER0505,ERRBLK    ;SET ERROR ROUTINE ADDRESS IN ERROR TABLE.
;*****
; WAIT UP TO 3 SECONDS FOR THE DUT MASTER RESET BIT TO CLEAR.
; IF TIME-OUT OCCURS, THEN EXIT THIS TEST.
;*****
      MOV #3000.,R1          ;TIME-OUT VALUE IS 3.0 SECONDS.
      MOV #BIT05,R2          ;WAITING FOR MASTER RESET BIT.
      CLR R3                 ;WAITING FOR BIT TO CLEAR.
      MOV CSRA,R4            ;BIT IS IN THE DUT'S CSR.
      JSR PC,MSLGET          ;WAIT FOR DUT_CSR_MR BIT TO CLEAR.
      BCC 501                ;ABORT THE TEST IF MR DID NOT CLEAR.
;*****
; DETERMINE IF THE DUT TAKES TOO SHORT OR TOO LONG A TIME TO SKIP THE SELF-TEST
; SET-UP A TIME-OUT OF 50 MILLI-SECONDS, IF MR IS CLEAR IN LESS THAN 10 MILLI
; -SECOND, OR GREATER THAN 50 MILLI SECONDS, REPORT THE ERROR.
;*****
      MOV #50.,R1            ;TIME OUT VALUE IS 50 MILLI SECONDS.
      MOV R2,(R4)            ;SET THE DUT MASTER RESET BIT.
      JSR PC,SKPSTS          ;WRITE THE SKIP SELFTEST CODES TO THE DUT.
      JSR PC,MSLGET          ;WAIT FOR DUT_CSR_MR BIT TO CLEAR.
      BCC 21                 ;GO REPORT ERR IF SKIPPING STEST TOOK TOO LONG.
      CMP R1,#40.            ;GO REP ERR IF SELFTEST COMPLETED IN < 10 MS.
      BGT 41
;*****
; SELF-TEST COMPLETED WITHIN 10 MILLI-SEC TO 50 MILLI-SECONDS.
; VERIFY THAT THE SELF-TEST CODES IN THE FIFO ARE "GOOD" CODES ,IE THE DUT
; SUCCESSFULLY COMPLETED THE SELF-TEST.
; THIS SUBROUTINE REPORTS ERRORS WITH NUMBERS >>>>> 1003 THRU 1007 <<<<<.
;*****
      MOV #1003.,ERRNBR      ;SET ERROR NUMBER TO 1003.
      JSR PC,RSTRT          ;CHECK SELF TEST CODES IN THE FIFO.
      BR 601                ;EXIT TEST.
;*****
; ERROR REPORTS:
;*****
; REPORT SKIP SELF TEST TOOK TOO LONG.
      MOV #1001.,ERRNBR      ;SET THE ERROR NUMBER IN THE ERROR TABLE.

```

HARDWARE TEST

- SKSELF -

```

5550 027772 012701 011334      MOV    #EM1002,R1      ;SELECT ERROR MESSAGE.
5551 027776      ERROR      ;REPORT ERROR.          ..... ERROR #1001 .....
                                TRAP    C1ERROR
5552 030000 000414      BR      601      ;EXIT THE TEST.
5553
5554      ;REPORT SKIP SELF TEST COMPLETED TOO SOON.
5555 030002 012767 001752 153760 41:      MOV    #1002,ERRNBR      ;SET THE ERROR NUMBER IN THE ERROR TABLE.
5556 030010 012701 011421      MOV    #EM1003,R1      ;SELECT ERROR MESSAGE.
5557 030014      ERROR      ;REPORT ERROR.          ..... ERROR #1002 .....
                                TRAP    C1ERROR
5558 030016 000405      BR      601      ;EXIT THE TEST.
5559
5560 030020 012767 001753 153742 501:      MOV    #1003,ERRNBR      ;SET ERROR NUMBER.
5561 030026 004767 172430      JSR     PC,TSABRT      ;REPORT NON TEST RELATED ERROR.
5562
5563 030032      601:      SETPRI    #PRI07      ;DISABLE ALL INTERRUPTS.
5564 030032 012700 000340      MOV    #PRI07,PC      MOV    #PRI07,PC
5565 030036 104441      TRAP    C1SPRI
5564 030040 005067 152306      CLR     CTRLCF      ;INDICATE THAT WE COMPLETED THE TEST.
5565 030044      ENDTST
                                L10037:
030044 104401      TRAP    C1ETST

```

HARDWARE TEST

- SKSELF -

```

5567
5568
5569
5570
5571
5572
5573
5574
5575
5576 030046
5577 030046
5578 030046
5579 030046
5580 030052
5581 030054
5582 030054
5583 030054
5584 030054
5585 030054
5586 030054
5587 030054
5588 030112
5589 030116
5590 030122
5591 030124
5592 030130
5593 030134
5594 030134
5595 030134
5596 030134
5597 030136
5598 030140
5599 030140
5600 030144
5601 030150
5602 030154
5603 030154
5604 030156
5605 030162
5606 030166
5607 030166
5608 030166
5609 030166
5610 030166
5611 030166
5612 030166
5613 030166
5614 030166
5615 030166
5616 030170
5617 030174
5618 030176
5619 030202
5620 030204

012700 000240
012700 000013
012767 000013
012767 177777
012767 000001
012767 011477
012767 016466
012701 005670
012702 000040
005003
016704 152114
004767 170202
103044
010214
004767 172170
012701 000005
012702 020000
010203
016704 152062
004767 170150
103020
012701 000017
005003
004767 170134
103012
010105

.SBTTL HARDWARE TEST - DFSKST
;*****
; - DIAGNOSTIC FAIL BIT, SKIP SELF TEST TEST
; THIS TEST VERIFIES THAT THE DIAGNOSTIC FAIL BIT OF THE DUT, CORRECTLY
; CHANGES STATE AS THE ON BOARDED SELFTEST IS SKIPPED.
;*****
; BGNTEST
;
; SETPRI @PRI05 ; ALLOW LTC INTERRUPTS.
;
; T11::
; MOV @PRI05,R0
; TRAP C1SPRI
;
; TNUM == TNUM + 1 ; INCREMENT THE ASSEMBLY TIME TEST COUNTER.
; MOV @TNUM,TSTNUM ; SET UP THE TEST NUMBER. (11)
; MOV @1,CTRLCF ; INDICATE THAT WE ARE WITHIN A TEST.
; MOV @1,ERRTYP ; SET ERROR TYPE AS FATAL IN ERROR TABLE.
; MOV @EM1101,ERRMSG ; SET ERROR MESSAGE ADDRESS IN ERROR TABLE.
; MOV @ER0503,ERRBLK ; SET ERROR ROUTINE ADDRESS IN ERROR TABLE.
;
; WAIT UP TO 3 SECONDS FOR THE DUT MASTER RESET BIT TO CLEAR.
; IF TIME-OUT OCCURS, THEN EXIT THIS TEST.
;
; MOV @3000.,R1 ; TIME-OUT VALUE IS 3.0 SECONDS.
; MOV @BIT05,R2 ; WAITING FOR MASTER RESET BIT.
; CLR R3 ; WAITING FOR BIT TO CLEAR.
; MOV CSRA,R4 ; BIT IS IN THE DUT'S CSR.
; JSR PC,MSLGET ; WAIT FOR DUT_CSR_MR BIT TO CLEAR.
; BCC 508 ; ABORT THE TEST IF MR DID NOT CLEAR.
;
; RESET THE DUT, SKIP THE SELF-TEST.
;
; MOV R2,(R4) ; SET THE DUT MASTER RESET BIT.
; JSR PC,SKPSTS ; WRITE THE SKIP SELFTEST CODES TO THE DUT.
;
; SET TIME OUT OF 5 MILLI SECONDS, WAIT FOR DIAG FAIL BIT TO SET.
; IF TIME OUT OCCURS GO REPORT THE ERROR.
;
; MOV @5.,R1 ; TIME-OUT VALUE IS 5 MILLI SECONDS.
; MOV @BIT13,R2 ; WAITING FOR DIAGNOSTIC FAIL BIT.
; MOV R2,R3 ; WAITING FOR BIT TO SET.
; MOV CSRA,R4 ; BIT IS IN THE DUT'S CSR.
; JSR PC,MSLGET ; WAIT FOR DUT_CSR_DF BIT TO CLEAR.
; BCC 41 ; IF DIAG_FAIL DID NOT SET, GO REPORT ERROR.
;
; SET TIME-OUT OF 15 MILLI-SECS, WAIT FOR DIAG_FAIL TO CLEAR.
; IF TIME OUT OCCURS GO REPORT THE ERROR.
; VERIFY THE DIAG FAIL BIT IS IN A STABLE STATE BEFORE CONTINUING. LOOP
; BACK IF THE STATE WAS TRANSITORY, USING THE REMAINDER OF THE 15 MS TIME OUT.
;
; MOV @15.,R1 ; TIME-OUT VALUE IS 15 MILLI SECONDS.
; CLR R3 ; WAITING FOR BIT TO CLEAR.
; JSR PC,MSLGET ; WAIT FOR DUT_CSR_DF BIT TO CLEAR.
; BCC 41 ; IF DIAG_FAIL DID NOT CLEAR, GO REPORT ERROR.
; MOV R1,R5 ; SAVE THE REMAINING TIME OUT VALUE.

```

HARDWARE TEST

DFSKST

```

5621 030206 012701 000001      MOV    #1,R1      ;SET TIME OUT OF 1 MILLI SECOND.
5622 030212 052703 020000      BIS    #BIT13,R3    ;WAIT FOR BIT TO SET.
5623 030216 004767 170114      JSR     PC,MSLGET    ;DOUBLE CHECK TO ELIMINATE NOISE PROBLEMS.
5624 030222 103016              BCC     60$      ;EXIT IF DIAG_FAIL BIT STILL CLEAR.
5625 030224 010501              MOV     R5,R1      ;PASS THE REMAINING TIME-OUT VALUE.
5626 030226 000762              BR      2$      ;LOOP TO CHECK AGAIN.
5627
5628      ;
5629      ; ERROR REPORTS:
5630      ;
5631 030230 012767 002115 153532 4$: ;REPORT DIAGNOSTIC FAIL BIT BAD.
5632 030236 012701 011740      MOV     #1101,ERRNBR ;SET THE ERROR NUMBER IN THE ERROR TABLE.
5633 030242 104460              MOV     #EM1205,R1 ;SELECT ERROR MESSAGE.
5634 030244 000405              ERROR    ;REPORT ERROR.
5635      ;
5636 030246 012767 002116 153514 50$: ;EXIT THE TEST.
5637 030254 004767 172202      BR      60$      ;
5638      ;
5639 030260 012700 000340 60$: ;SET THE ERROR NUMBER FOR TSABRT RTN.
5640 030266 005067 152060      MOV     #1102,ERRNBR ;REPORT NON-TEST RELATED ERROR.
5641 030272 104401              JSR     PC,TSABRT
5642 030272 104401              SETPRI  #PRI07    ;DISABLE ALL INTERRUPTS.
5643 030272 104401              CLRA      ;
5644 030272 104401              ENDTST   ;INDICATE THAT WE COMPLETED A TEST.
5645 030272 104401              TRAP     #PRI07,RC ;
5646 030272 104401              TRAP     C$SPRI
5647 030272 104401              TRAP     C$ETST
5648 030272 104401
5649 030272 104401
5650 030272 104401
5651 030272 104401
5652 030272 104401
5653 030272 104401
5654 030272 104401
5655 030272 104401
5656 030272 104401
5657 030272 104401
5658 030272 104401
5659 030272 104401
5660 030272 104401
5661 030272 104401
5662 030272 104401
5663 030272 104401
5664 030272 104401
5665 030272 104401
5666 030272 104401
5667 030272 104401
5668 030272 104401
5669 030272 104401
5670 030272 104401
5671 030272 104401
5672 030272 104401
5673 030272 104401
5674 030272 104401
5675 030272 104401
5676 030272 104401
5677 030272 104401
5678 030272 104401
5679 030272 104401
5680 030272 104401
5681 030272 104401
5682 030272 104401
5683 030272 104401
5684 030272 104401
5685 030272 104401
5686 030272 104401
5687 030272 104401
5688 030272 104401
5689 030272 104401
5690 030272 104401
5691 030272 104401
5692 030272 104401
5693 030272 104401
5694 030272 104401
5695 030272 104401
5696 030272 104401
5697 030272 104401
5698 030272 104401
5699 030272 104401
5700 030272 104401
5701 030272 104401
5702 030272 104401
5703 030272 104401
5704 030272 104401
5705 030272 104401
5706 030272 104401
5707 030272 104401
5708 030272 104401
5709 030272 104401
5710 030272 104401
5711 030272 104401
5712 030272 104401
5713 030272 104401
5714 030272 104401
5715 030272 104401
5716 030272 104401
5717 030272 104401
5718 030272 104401
5719 030272 104401
5720 030272 104401
5721 030272 104401
5722 030272 104401
5723 030272 104401
5724 030272 104401
5725 030272 104401
5726 030272 104401
5727 030272 104401
5728 030272 104401
5729 030272 104401
5730 030272 104401
5731 030272 104401
5732 030272 104401
5733 030272 104401
5734 030272 104401
5735 030272 104401
5736 030272 104401
5737 030272 104401
5738 030272 104401
5739 030272 104401
5740 030272 104401
5741 030272 104401
5742 030272 104401
5743 030272 104401
5744 030272 104401
5745 030272 104401
5746 030272 104401
5747 030272 104401
5748 030272 104401
5749 030272 104401
5750 030272 104401
5751 030272 104401
5752 030272 104401
5753 030272 104401
5754 030272 104401
5755 030272 104401
5756 030272 104401
5757 030272 104401
5758 030272 104401
5759 030272 104401
5760 030272 104401
5761 030272 104401
5762 030272 104401
5763 030272 104401
5764 030272 104401
5765 030272 104401
5766 030272 104401
5767 030272 104401
5768 030272 104401
5769 030272 104401
5770 030272 104401
5771 030272 104401
5772 030272 104401
5773 030272 104401
5774 030272 104401
5775 030272 104401
5776 030272 104401
5777 030272 104401
5778 030272 104401
5779 030272 104401
5780 030272 104401
5781 030272 104401
5782 030272 104401
5783 030272 104401
5784 030272 104401
5785 030272 104401
5786 030272 104401
5787 030272 104401
5788 030272 104401
5789 030272 104401
5790 030272 104401
5791 030272 104401
5792 030272 104401
5793 030272 104401
5794 030272 104401
5795 030272 104401
5796 030272 104401
5797 030272 104401
5798 030272 104401
5799 030272 104401
5800 030272 104401
5801 030272 104401
5802 030272 104401
5803 030272 104401
5804 030272 104401
5805 030272 104401
5806 030272 104401
5807 030272 104401
5808 030272 104401
5809 030272 104401
5810 030272 104401
5811 030272 104401
5812 030272 104401
5813 030272 104401
5814 030272 104401
5815 030272 104401
5816 030272 104401
5817 030272 104401
5818 030272 104401
5819 030272 104401
5820 030272 104401
5821 030272 104401
5822 030272 104401
5823 030272 104401
5824 030272 104401
5825 030272 104401
5826 030272 104401
5827 030272 104401
5828 030272 104401
5829 030272 104401
5830 030272 104401
5831 030272 104401
5832 030272 104401
5833 030272 104401
5834 030272 104401
5835 030272 104401
5836 030272 104401
5837 030272 104401
5838 030272 104401
5839 030272 104401
5840 030272 104401
5841 030272 104401
5842 030272 104401
5843 030272 104401
5844 030272 104401
5845 030272 104401
5846 030272 104401
5847 030272 104401
5848 030272 104401
5849 030272 104401
5850 030272 104401
5851 030272 104401
5852 030272 104401
5853 030272 104401
5854 030272 104401
5855 030272 104401
5856 030272 104401
5857 030272 104401
5858 030272 104401
5859 030272 104401
5860 030272 104401
5861 030272 104401
5862 030272 104401
5863 030272 104401
5864 030272 104401
5865 030272 104401
5866 030272 104401
5867 030272 104401
5868 030272 104401
5869 030272 104401
5870 030272 104401
5871 030272 104401
5872 030272 104401
5873 030272 104401
5874 030272 104401
5875 030272 104401
5876 030272 104401
5877 030272 104401
5878 030272 104401
5879 030272 104401
5880 030272 104401
5881 030272 104401
5882 030272 104401
5883 030272 104401
5884 030272 104401
5885 030272 104401
5886 030272 104401
5887 030272 104401
5888 030272 104401
5889 030272 104401
5890 030272 104401
5891 030272 104401
5892 030272 104401
5893 030272 104401
5894 030272 104401
5895 030272 104401
5896 030272 104401
5897 030272 104401
5898 030272 104401
5899 030272 104401
5900 030272 104401
5901 030272 104401
5902 030272 104401
5903 030272 104401
5904 030272 104401
5905 030272 104401
5906 030272 104401
5907 030272 104401
5908 030272 104401
5909 030272 104401
5910 030272 104401
5911 030272 104401
5912 030272 104401
5913 030272 104401
5914 030272 104401
5915 030272 104401
5916 030272 104401
5917 030272 104401
5918 030272 104401
5919 030272 104401
5920 030272 104401
5921 030272 104401
5922 030272 104401
5923 030272 104401
5924 030272 104401
5925 030272 104401
5926 030272 104401
5927 030272 104401
5928 030272 104401
5929 030272 104401
5930 030272 104401
5931 030272 104401
5932 030272 104401
5933 030272 104401
5934 030272 104401
5935 030272 104401
5936 030272 104401
5937 030272 104401
5938 030272 104401
5939 030272 104401
5940 030272 104401
5941 030272 104401
5942 030272 104401
5943 030272 104401
5944 030272 104401
5945 030272 104401
5946 030272 104401
5947 030272 104401
5948 030272 104401
5949 030272 104401
5950 030272 104401
5951 030272 104401
5952 030272 104401
5953 030272 104401
5954 030272 104401
5955 030272 104401
5956 030272 104401
5957 030272 104401
5958 030272 104401
5959 030272 104401
5960 030272 104401
5961 030272 104401
5962 030272 104401
5963 030272 104401
5964 030272 104401
5965 030272 104401
5966 030272 104401
5967 030272 104401
5968 030272 104401
5969 030272 104401
5970 030272 104401
5971 030272 104401
5972 030272 104401
5973 030272 104401
5974 030272 104401
5975 030272 104401
5976 030272 104401
5977 030272 104401
5978 030272 104401
5979 030272 104401
5980 030272 104401
5981 030272 104401
5982 030272 104401
5983 030272 104401
5984 030272 104401
5985 030272 104401
5986 030272 104401
5987 030272 104401
5988 030272 104401
5989 030272 104401
5990 030272 104401
5991 030272 104401
5992 030272 104401
5993 030272 104401
5994 030272 104401
5995 030272 104401
5996 030272 104401
5997 030272 104401
5998 030272 104401
5999 030272 104401
6000 030272 104401

```

HARDWARE TEST

SELFTEST

```

5643 .SBTTL HARDWARE TEST SELFTEST
5644 ;*****
5645 ;* SELF TEST TEST
5646 ;* THIS TEST VERIFIES THAT THE DUT'S SELF TEST EXECUTES WITHIN THE
5647 ;* TIME ALLOWED, AND THAT THE FIFO CONTAINS THE CORRECT CODES AFTER ITS
5648 ;* COMPLETION.
5649 ;*
5650 ;*****
5651 030274 BGNTEST
5652 030274 SETPRI @PRIOS ;ALLOW LTC INTERRUPTS. T12::
5653 030274 012700 000240 MOV @PRIOS,R0
5654 030300 104441 TRAP C$SPRI
5655 030302 000014 TNUM == TNUM + 1 ;INCREMENT THE ASSEMBLY TIME TEST COUNTER.
5656 030310 012767 000014 152016 MOV @TNUM,TSTNUM ;SET UP THE TEST NUMBER. (12)
5657 030316 012767 177777 152034 MOV @1,CTRLCF ;INDICATE THAT WE ARE WITHIN A TEST.
5658 030324 012767 000001 153442 MOV @1,ERRTYP ;SET ERROR TYPE AS FATAL IN ERROR TABLE.
5659 030332 012767 011523 153440 MOV @EM1201,ERRMSG ;SET ERROR MESSAGE ADDRESS IN ERROR TABLE.
5660 030332 012767 016466 153434 MOV @ER0503,ERRBLK ;SET ERROR ROUTINE ADDRESS IN ERROR TABLE.
5661 ;*
5662 ;* WAIT UP TO 3 SECONDS FOR THE DUT MASTER RESET BIT TO CLEAR.
5663 ;* IF TIME-OUT OCCURS, THEN EXIT THIS TEST.
5664 ;
5665 030340 012701 005670 MOV @3000.,R1 ;TIME-OUT VALUE IS 3.0 SECONDS.
5666 030344 012702 000040 MOV @BIT05,R2 ;WAITING FOR MASTER RESET BIT.
5667 030350 005003 CLR R3 ;WAITING FOR BIT TO CLEAR.
5668 030352 016704 151666 MOV CSRA,R4 ;BIT IS IN THE DUT'S CSR.
5669 030356 004767 167754 JSR PC,MSLGET ;WAIT FOR DUT_CSR_MR BIT TO CLEAR.
5670 030362 103062 BCC 50$ ;ABORT THE TEST IF MR DID NOT CLEAR.
5671 ;*
5672 ;* DETERMINE IF THE SELF TEST TAKES TOO SHORT OR TOO LONG A TIME TO COMPLETE.
5673 ;* SET-UP A TIME-OUT OF 3 SECONDS, IF MR IS CLEAR IN LESS THAN 1/2 SECOND, OR
5674 ;* GREATER THAN 3 SECONDS, REPORT THE ERROR.
5675 ;
5676 030364 012701 005670 MOV @3000.,R1 ;TIME OUT VALUE IS 3.0 SECONDS.
5677 030370 010214 MOV R2,(R4) ;SET THE DUT MASTER RESET BIT.
5678 030372 004767 167740 JSR PC,MSLGET ;WAIT FOR DUT_CSR_MR BIT TO CLEAR.
5679 030376 103030 BCC 4$ ;GO REPORT ERROR SELFTEST TOOK TOO LONG.
5680 030400 012702 005670 MOV @3000.,R2
5681 030404 160102 SUB R1,R2 ;CALCULATE # OF MS SELFTEST TO COMPLETE.
5682 030406 020227 000062 CMP R2,#50.
5683 030412 002431 BLT 6$ ;SELFTEST SKIPPED? YES, GO REPORT ERROR.
5684 030414 020227 000764 CMP R2,#500.
5685 030420 002434 BLT 8$ ;GO REP ERR IF SELFTEST COMPLETED IN < 1/2 SEC.
5686 ;*
5687 ;* SELF TEST COMPLETED WITHIN 1SEC TO 3 SECONDS.
5688 ;* CHECK THE STATE OF THE DIAGNOSTIC FAIL BIT, REPORT ERROR IF IT IS SET.
5689 ;
5690 030422 032714 020000 BIT @BIT13,(R4) ;DETERMINE IF THE DIAG_FAIL BIT IS CLEAR.
5691 030426 001406 BEQ 2$ ;SKIP ERROR REPORT IF BIT IS CLEAR.
5692 ;REPORT DIAGNOSTIC FAIL BIT BAD.
5693 030430 012767 002264 153332 MOV @1204.,ERRNBR ;SET ERROR NUMBER TO IN ERROR TABLE.
5694 030436 012701 011740 MOV @EM1205,R1 ;SELECT THE ERROR MESSAGE.
5695 030442 030442 104460 ERROR ;***** ERROR #1204 *****
5696 TRAP C$ERROR
5697 ;*
5698 ;* VERIFY THAT THE SELF TEST CODES IN THE FIFO ARE 'GOOD' CODES ,IF THE DUT

```

HARDWARE TEST - SELFIS

```

5696 ; SUCCESSFULLY COMPLETED THE SELF TEST.
5697 ; THIS SUBROUTINE REPORTS ERRORS WITH NUMBERS >>>> 1205 THRU 1209 <<<<.
5698 ;
5699 030444 012767 002265 153316 2: MOV #1205.,ERRNBR ;SET ERROR NUMBER TO 1205.
5700 030452 004767 171142 JSR PC,WSRPT ;CHECK SELF TEST CODES IN THE FIFO.
5701 030456 000431 BR 60: ;EXIT TEST.
5702 ;
5703 ; ERROR REPORTS:
5704 ;
5705 ;REPORT SELF TEST TOOK TOO LONG TO COMPLETE.
5706 030460 012767 002261 153302 4: MOV #1201.,ERRNBR ;SET THE ERROR NUMBER IN THE ERROR TABLE.
5707 030466 012701 011542 MOV #EM1202,R1 ;SELECT ERROR MESSAGE.
5708 030472 104460 ERROR ;REPORT ERROR. >>>> ERROR #1201 <<<<
5709 030474 000422 BR 60: ;EXIT THE TEST. TRAP C$ERROR
5710 ;
5711 ;REPORT SELF-TEST DID NOT EXECUTE AFTER DUT RESET.
5712 030476 012767 002262 153264 6: MOV #1202.,ERRNBR ;SET THE ERROR NUMBER IN ERROR TABLE.
5713 030504 012701 011704 MOV #EM1204,R1 ;SELECT ERROR MESSAGE.
5714 030510 104460 ERROR ;REPORT ERROR. >>>> ERROR #1202 <<<<
5715 TRAP C$ERROR
5716 ;REPORT SELF-TEST COMPETED TOO SOON.
5717 030512 012767 002263 153250 8: MOV #1203.,ERRNBR ;SET THE ERROR NUMBER IN THE ERROR TABLE.
5718 030520 012701 011626 MOV #EM1203,R1 ;SELECT ERROR MESSAGE.
5719 030524 104460 ERROR ;REPORT ERROR. >>>> ERROR #1203 <<<<
5720 030526 000405 BR 60: ;EXIT THE TEST. TRAP C$ERROR
5721 ;
5722 030530 012767 002272 153232 50: MOV #1210.,ERRNBR ;SET THE ERROR NUMBER FOR TSABRT RTN.
5723 030536 004767 171720 JSR PC,TSABRT ;REPORT NON TEST RELATED ERROR.
5724 ;
5725 030542 012700 000340 60: SETPRI #PRI07 ;DISABLE ALL INTERRUPTS.
5726 030546 104441 MOV #PRI07,R0
5727 030554 005067 151576 CLR CTRLCF ;INDICATE THAT WE COMPLETED THE TEST.
5728 030554 104401 TRAP C$SPRI
5729 030554 104401 L10041: TRAP C$ETST

```

HARDWARE TEST

SELFTEST

5729
5730
5731
5732
5733
5734
5735
5736
5737
5738
5739 030556
030556
5740 030556
030556 012700 000240
030562 104441
5741 000015
5742 030564 012767 000015 151534
5743 030572 012767 177777 151552
5744 030600 012767 000001 153160
5745 030606 012767 011764 153156
5746 030614 012767 016466 153152
5747 030622 012767 002425 153140
5748
5749
5750
5751
5752 030630 012701 005670
5753 030634 012702 000040
5754 030640 005003
5755 030642 016704 151376
5756 030646 004767 167464
5757 030652 103120
5758
5759
5760
5761
5762 030654 010214
5763 030656 004767 171452
5764 030662 012701 000764
5765 030666 004767 167444
5766 030672 103110
5767 030674 005267 153070
5768 030700 012700 000010
5769 030704 005003
5770 030706 016704 151334
5771 030712 011402
5772 030714 100077
5773 030716 042702 007402
5774 030722 020227 170001
5775 030726 001002
5776 030730 012703 177777
5777 030734 005300
5778 030736 001365
5779 030740 005703
5780 030742 100466
5781
5782

```
.SBTTL HARDWARE TEST          SIFAIL
;*****
; SELF TEST FAIL TEST
; THIS TEST VERIFIES THAT THE DUT WILL REPORT SELFTEST ERRORS VIA THE
; FIFO. AND THAT THE DIAGNOSTIC FAIL BIT WILL INDICATE THE ERROR.
; THIS IS ACCOMPLISHED VIA A SOFTWARE 'HOOK' IN THE SELF TEST, WHICH
; FORCES A 'PROC1 TO RAM ERROR' TO BE PLACED IN THE FIFO.
;*****
BGNTST
;*****
; T13::
SETPRI @PRI05          ;ALLOW LTC INTERRUPTS.
MOV @PRI05,R0
TRAP C:SPRI
TNUM == TNUM + 1      ;INCREMENT THE ASSEMBLY TIME TEST COUNTER.
MOV @TNUM,TSTNUM      ;SET UP THE TEST NUMBER. (13)
MOV @-1,CTRLCF        ;INDICATE THAT WE ARE WITHIN A TEST.
MOV @1,ERRTYP         ;SET ERROR TYPE AS FATAL IN ERROR TABLE.
MOV @EM1301,ERRMSG    ;SET ERROR MESSAGE ADDRESS IN ERROR TABLE.
MOV @ER0503,ERRBLK    ;SET ERROR ROUTINE ADDRESS IN ERROR TABLE.
MOV @1301,ERRNBR      ;SET ERROR NUMBER TO 1301.
;
; WAIT UP TO 3 SECONDS FOR THE DUT MASTER RESET BIT TO CLEAR.
; IF TIME-OUT OCCURS, THEN EXIT THIS TEST.
;
MOV @3000,R1          ;TIME-OUT VALUE IS 3.0 SECONDS.
MOV @BI105,R2         ;WAITING FOR MASTER RESF1 BIT.
CLR R3                ;WAITING FOR BIT TO CLEAR.
MOV CSRA,R4           ;BIT IS IN THE DUT'S CSR.
JSR PC,MSLGET         ;WAIT FOR DUT_CSR_MR BIT TO CLEAR.
BCC 50$               ;GO REPORT ERROR IF MR DID NOT CLEAR.
;
; RESET THE DUT, CHECK FOR ROM VERSION 0.
; IF VERSION 0 IS FOUND THEN EXIT THIS TEST.
;
MOV R2,(R4)           ;RESET THE DUT.
JSR PC,SKPSTS         ;SKIP THE SELFTEST.
MOV @500,R1           ;PASS TIME OUT VALUE OF 500 MILLISECS.
JSR PC,MSLGET         ;WAIT FOR MR BIT TO CLEAR.
BCC 50$               ;GO REPORT ERROR IF TIME-OUT OCCURRED.
INC ERRNBR            ;SET ERROR NUMBER TO 1302.
MOV @8,R0             ;SET MAXIMUM READ COUNT.
CLR R3                ;CLEAR THE ROM VERSION 0 FLAG.
MOV @BUFA,R4          ;GET ADDRESS OF THE RECEIVER BUFFER REGISTER.
2$: MOV (R4),R2         ;READ A CODE FROM THE FIFO.
BPL 50$               ;GO REPORT ERROR IF THE FIFO IS EMPTY.
BIC @7402,R2          ;REMOVE THE LINE NUMBER AND PROC INDICATOR.
CMP R2,@170001        ;COMPARE WITH ROM VERSION #0 CODE.
BNE 4$                ;ROM VERSION #0? NO, SKIP SETTING FLAG.
MOV @1,R3             ;YES, SET THE ROM VERSION #0 FLAGS.
4$: DEC R0            ;DECREMENT MAX READ COUNTER.
BNE 2$                ;LOOP IF 8 CODES HAVE NOT BEEN READ.
TST R3                ;CHECK THE ROM VERSION #1 INDICATOR.
BMI 60$               ;ROM VERSION 0 IN EITHER PROCESSOR? YES, EXIT.
;*****
```

HARDWARE TEST

- STFAIL

```

5783 ; RESET THE DUT, DELAY FOR 25 MILLI SECONDS BEFORE WRITING THE FAIL SELF TEST
5784 ; CODE TO TBUFFT REGISTER ON CHANNEL 0.
5785 ;
5786 030744 012777 000040 151272      MOV    #BIT05,@CSRA    ;SET DUT MASTER RESET BIT, SELECT CHANNEL 0.
5787 030752 012704 000031             MOV    #25.,R4          ;PASS DELAY PERIOD OF 25 MILLI SECS.
5788 030756 004767 167314             JSR    PC,DELAY          ;WAIT FOR SLFTTEST TO INITIALISE.
5789 030762 012777 146314 151272      MOV    #146314,@TXBFCA ;WRITE THE FAIL SELF TEST CODE TO TBUFFT REG.
5790 ;
5791 ; WAIT UP TO 2 SECONDS FOR THE SELF-TEST TO COMPLETE.
5792 ; IF TIME OUT OCCURS, THEN EXIT THIS TEST.
5793 ;
5794 030770 005267 152774             INC     ERRNBR          ;SET ERROR NUMBER TO 1303.
5795 030774 012701 003720             MOV    #2000.,R1         ;TIME-OUT VALUE IS 2.0 SECONDS.
5796 031000 012702 000040             MOV    #BIT05,R2         ;PASS THE BIT MAP OF THE BIT TO TEST.
5797 031004 005003                    CLR     R3              ;SET UP THE EXPECTED STATE.
5798 031006 016704 151232             MOV    CSRA,R4          ;BIT IS IN THE DUT'S CSR.
5799 031012 004767 167320             JSR    PC,MSLGET         ;WAIT FOR DUT CSR MR BIT TO CLEAR.
5800 031016 103036                    BCC     50$             ;GO REPORT ERROR IF MR DID NOT CLEAR.
5801 ;
5802 ; VERIFY THE DIAGNOSTIC FAIL BIT IS SET, INDICATING THE ERROR.
5803 ; REPORT ERROR IF DIAGNOSTIC FAIL BIT IS CLEAR.
5804 ;
5805 031020 005267 152744             INC     ERRNBR          ;SET ERROR NUMBER TO 1304.
5806 031024 032714 020000             BIT     #BIT13,(R4)        ;CHECK THE STATE OF THE DIAG_FAIL BIT.
5807 031030 001425                    BEQ     10$             ;GO REPORT ERROR IF DIAG FAIL BIT CLEAR.
5808 ;
5809 ; REMOVE THE 8 SELF TEST CODES FORM THE FIFO, AND VER. Y THAT AT LEAST
5810 ; ONE IS A PROC1 TO RAM ERROR CODE (231).
5811 ;
5812 031032 005267 152732             INC     ERRNBR          ;SET ERROR NUMBER TO 1305.
5813 031036 012700 000010             MOV    #8.,R0           ;SET MAXIMUM READ COUNT.
5814 031042 005001                    CLR     R1              ;CLEAR THE CORRECT CODE COUNTER.
5815 031044 016704 151176             MOV    RBUFA,R4         ;GET ADDRESS OF THE RECEIVER BUFFER REGISTER.
5816 031050 011402                    MOV    (R4),R2          ;READ A CODE FROM THE FIFO.
5817 031052 100020                    BPL     50$             ;GO REPORT ERROR IF THE FIFO IS EMPTY.
5818 031054 042702 007400             BIC     #7400,R2         ;REMOVE THE LINE NUMBER FROM THE CODE.
5819 031060 120227 170231             CMPB   R2,#170231        ;IS IT THE CORRECT ERROR CODE?.
5820 031064 001001                    BNE     8$             ;SKIP NEXT INSTRUCTION, IF NOT A 231 CODE.
5821 031066 005201                    INC     R1              ;INCREMENT COUNTER.
5822 031070 005300                    DEC     R0              ;DECREMENT MAX READ COUNTER.
5823 031072 001366                    BNE     6$             ;LOOP IF 8 CODES HAVE NOT BEEN READ.
5824 031074 005701                    TST     R1              ;WERE ANY 231 CODES FOUND?.
5825 031076 001010                    BNE     60$            ;YES, THEN EXIT.
5826 031100 005267 152664             INC     ERRNBR          ;NO, SET ERROR NUMBER TO 1306 AND REPORT ERROR.
5827 ;REPORT SELF-TEST ERROR REPORTING BAD.
5828 031104 012701 012010            10$: MOV    #EM1302,R1        ;SELECT ERROR MESSAGE.
5829 031110                                ERROR              ;REPORT ERROR.
5830 031112 000402                    BR      60$             ;EXIT THE TEST.
5831 ;
5832 031114 004767 171342            50$: JSR    PC,TSABRT        ;REPORT NON RELATED TEST ERROR.
5833 ;
5834 031120                                60$: SETPRI  #PRI07        ;DISABLE ALL INTERRUPTS.
5835 031126 005067 151220                                MOV    #PRI07,R0
5836 031132                                CLR     CTRLCF        ;INDICATE THAT WE COMPLETED THE TEST.
                                TRAP    C$SPRI

```

L 14

CVDHABO DMV 11 FUNC TST PART2 MACRO M1200 10 OCT 83 11:55 PAGE 111 2

SEQ 190

HARDWARE TEST - SFAIL -

031132
031132 104401

110042: TRAP C8F1C1

HARDWARE TEST

STFAIL -

```

5838
5839
5840
5841
5842
5843
5844
5845
5846
5847
5848
5849 031134
      031134
5850 031134
      031134 012700 000240
      031140 104441
5851      000016
5852 031142 012767 000016 151156
5853 031150 012767 177777 151174
5854 031156 012767 000001 152602
5855 031164 012767 012047 152600
5856 031172 012767 016466 152574
5857
5858
5859
5860
5861 031200 012701 005670
5862 031204 012702 000040
5863 031210 005003
5864 031212 016704 151026
5865 031216 004767 167114
5866 031222 103131
5867
5868
5869
5870 031224 010214
5871 031226 004767 171102
5872 031232 012701 005670
5873 031236 004767 167074
5874 031242 103121
5875
5876
5877
5878
5879
5880
5881
5882 031244 012705 000040
5883 031250 012703 000143
5884 031254 010304
5885 031256 012767 002571 152504
5886 031264 012701 012077
5887
5888 031270 017702 150752
5889 031274 100077
5890
5891

.SBTTL HARDWARE TEST - ROMVER
; *****
; - ROM VERSION TEST -
; THIS TEST VERIFIES THAT THE DUT'S SELF TEST PLACES VALID ROM VERSION
; NUMBERS IN THE FIFO AFTER IT HAS BEEN SKIPPED. THE ROM VERSION NUMBERS
; WILL BE REPORTED (ON THE FIRST PASS ONLY), IF AN AFFIRMATIVE ANSWER
; WAS GIVEN TO THE SOFTWARE P TABLE QUESTION.
; *****
; BGNSTST
;
; T14::
; SETPRI #PRI05 ;ALLOW LTC INTERRUPTS.
;
; MOV #PRI05,R0
; TRAP C:SPRI
;
; TNUM == TNUM + 1 ;INCREMENT THE ASSEMBLY TIME TEST COUNTER.
; MOV #TNUM,TSTNUM ;SET UP THE TEST NUMBER (14)
; MOV #-1,CTRLCF ;INDICATE THAT WE ARE WITHIN A TEST.
; MOV #1,ERRTYP ;SET ERROR TYPE AS FATAL IN ERROR TABLE.
; MOV #EM1401,ERRMSG ;SET ERROR MESSAGE ADDRESS IN ERROR TABLE.
; MOV #ER0503,ERRBLK ;SET ERROR ROUTINE ADDRESS IN ERROR TABLE.
;
; ; WAIT UP TO 3 SECONDS FOR THE DUT MASTER RESET BIT TO CLEAR.
; ; IF TIME-OUT OCCURS, THEN EXIT THIS TEST.
; -
; MOV #3000.,R1 ;TIME-OUT VALUE IS 3.0 SECONDS.
; MOV #BIT05,R2 ;WAITING FOR MASTER RESET BIT.
; CLR R3 ;WAITING FOR BIT TO CLEAR.
; MOV CSRA,R4 ;BIT IS IN THE DUT'S CSR.
; JSR PC,MSLGET ;WAIT FOR DUT_CSR_MR BIT TO CLEAR.
; BCC 50$ ;ABORT THE TEST IF MR DID NOT CLEAR.
;
; ; SET THE MASTER RESET BIT, AND SKIP THE SELF TEST.
; -
; MOV R2,(R4) ;SET THE MASTER RESET BIT.
; JSR PC,SKPSTS ;SKIP THE SELF TEST.
; MOV #3000.,R1 ;TIME-OUT VALUE IS 3.0 SECONDS.
; JSR PC,MSLGET ;WAIT FOR DUT_CSR_MR BIT TO CLEAR.
; BCC 50$ ;ABORT THE TEST IF MR DID NOT CLEAR.
;
; ; REMOVE CHARACTERS FROM THE FIFO UNTIL EITHER:
; ; (A) THE FIFO IS PURGED, GO REPORT THE ERROR.
; ; (B) THE MAXIMUM TRY COUNTER IS ZERO, GO REPORT THE ERROR.
; ; (C) PROC_1'S ROM VERSION NUMBER WAS FOUND BEFORE PROC_2'S, GO REPORT ERROR.
; ; (D) BOTH ROM VERSION NUMBERS HAVE BEEN FOUND.
; -
; MOV #4*NUMLNS,R5 ;SET MAXIMUM TRY COUNTER.
; MOV #99.,R3 ;SET AN INVALID ROM VERSION NUMBER FOR PROC_1.
; MOV R3,R4 ;SET AN INVALID ROM VERSION NUMBER FOR PROC_2.
; MOV #1401.,ERRNBR ;SET THE ERROR NUMBER TO 1401.
; MOV #EM1402,R1 ;SELECT MESSAGE TO BE REPORTED IF FIFO EMPTY.
;
; 2$: MOV #RBUFA,R2 ;READ THE NEXT CHAR FROM THE FIFO.
; BPL 12$ ;GO REPORT ERROR IF FIFO EMPTY.
;
; ; CHECK IF THE READ DATA IS A BMP CODE.

```

HARDWARE TEST

ROMVER -

```

5892
5893 031276 012700 000301      ;
5894 031302 040200      MOV    #301,R0      ;SET UP A BIT MASK OF A BMP CODE.
5895 031304 001003      BIC    R2,R0      ;TRY TO CLEAR THE BIT MASK WITH THE READ DATA.
5896 031306 004767 170754      BNE    4$      ;BRANCH IF NOT A BMP CODE.
5897 031312 000435      JSR    PC,SAVBMP    ;SAVE THE BMP CODE ON THE QUEUE.
5898      BR    8$      ;
5899      ;*
5900      ; CHECK IF THE READ DATA IS A SELF TEST CODE.
5901 031314 012700 000201 4$:    MOV    #201,R0      ;SET-UP A BIT MASK OF A SELFTEST CODE.
5902 031320 040200      BIC    R2,R0      ;TRY TO CLEAR THE BIT MASK WITH THE READ DATA.
5903 031322 001431      BEQ    8$      ;BRANCH IF IT IS A SELFTEST CODE.
5904
5905      ;*
5906      ; THE READ DATA IS A ROM VERSION NUMBER, DETERMINE WHICH ONE IT IS.
5907      ; -
5908 031324 032702 000002      BIT    #BIT1,R2      ;CHECK THE PROCESSOR NUMBER BIT IN THE CODE.
5909 031330 001407      BEQ    6$      ;BRANCH IF IT IS PROC 1 ROM VERSION NUMBER.
5910 031332 010204      MOV    R2,R4      ;SAVE PROC_2 ROM VERSION NUMBER.
5911 031334 042704 177603      BIC    #177603,R4      ;CLEAR ANY UNWANTED BITS.
5912 031340 000241      CLC      ;CLEAR THE CARRY BIT.
5913 031342 006004      ROR    R4      ;SHIFT THE CODES ALONG TO GET THE ROM
5914 031344 006004      ROR    R4      ; VERSION NUMBER IN THE LOW 5 BITS.
5915 031346 000417      BR    8$      ;
5916 031350 010203 6$:    MOV    R2,R3      ;SAVE PROC_1 ROM VERSION NUMBER.
5917 031352 042703 177603      BIC    #177603,R3      ;CLEAR ANY UNWANTED BITS.
5918 031356 000241      CLC      ;CLEAR THE CARRY BIT.
5919 031360 006003      ROR    R3      ;SHIFT THE CODE ALONG TO GET THE ROM
5920 031362 006003      ROR    R3      ; VERSION NUMBER IN THE LOW 5 BITS.
5921 031364 020427 000143      CMP    R4,#99.      ;CHECK IF WE HAVE RECEIVE PROC 2 ROM CODE.
5922 031370 001016      BNE    10$      ;GO REPORT BOTH ROM VERSION NUMBERS.
5923
5924      ;*
5925      ; RECEIVED ROM VERSION NUMBERS OUT OF SEQUENCE.
5926      ; IE, PROC_1'S ROM VERSION NUMBER FOUND IN THE FIFO BEFORE PROC_2'S.
5927 031372 012701 012165      ; -
5928 031376 012767 002572 152364      MOV    #EM1403,R1      ;SELECT THE ERROR MESSAGE TO BE REPORTED.
5929 031404 000433      MOV    #1402.,ERRNBR      ;SET THE ERROR NUMBER.
5930      BR    12$      ;GO REPORT ERROR.
5931
5932 031406 005305 8$:    DEC    R5      ;DECREMENT THE MAX TRY COUNTER.
5933 031410 001327      BNE    2$      ;LOOP TO GET THE NEXT CHAR FROM THE FIFO.
5934 031412 012701 012240      MOV    #EM1404,R1      ;SELECT THE ERROR MESSAGE TO BE REPORTED.
5935 031416 012767 002573 152344      MOV    #1403.,ERRNBR      ;SET THE ERROR NUMBER.
5936 031424 000423      BR    12$      ;GIVE UP, GO REPORT ERROR.
5937
5938      ;*
5939      ; IF THIS IS THE FIRST PASS, AND SOFTWARE P-TABLE QUESTION WAS ANSWERED YES,
5940      ; THEN REPORT THE ROM VERSION NUMBERS TO HE OPERATOR.
5941 031426 032767 000001 150572 10$:    BIT    #BIT0,OPTION      ;CHECK ON THE STATE OF THE SOFTWARE SWITCH.
5942 031434 001431      BEQ    60$      ;EXIT IF NO ROM VERSION PRINTOUT WAS REQUESTED.
5943 031436 026727 150670 000001      CMP    PASCNT,#1      ;CHECK IF THIS IS THE FIRST PASS.
5944 031444 003025      BGT    60$      ;EXIT IF ROM VERS HAVE ALREADY BEEN REPORTED.
5945      PRINTB #EF1401,R3,R4      ;PRINT THE ROM VERSION NUMBERS.
5946      MOV    R4,-(SP)
5947      MOV    R3,-(SP)
5948      MOV    #EF1401,(SP)
5949      MOV    #3,-(SP)

```

[illegible]

HARDWARE TEST

REGWRW

```

5960 .SBTTL HARDWARE TEST                      REGWRW
5961 ; .....
5962 ; DEVICE REGISTER WORD ACCESS READ AND WRITE TEST
5963 ;
5964 ; THIS TEST VERIFIES THAT THE DEVICE REGISTERS CAN BE READ AND WRITTEN
5965 ; CORRECTLY USING WORD ACCESSES.
5966 ; .....
5967
5968
5969 031534      BGNTST
031534
5970 031534      SETPRI @PRI05                ;ALLOW THE LTC TO INTERRUPT.
031534 012700 000240      MOV @PRI05,R0
031540 104441      TRAP C@SPRI
000017
5971 031542 012767 000017 150556      TNUM == TNUM + 1      ;INCREMENT THE ASSEMBLY TIME TEST COUNTER.
5972 031542 012767 000017 150556      MOV @TNUM,TSTNUM      ;SET UP THE TEST NUMBER. (16)
5973 031550 012767 177777 150574      MOV @-1,CTRLCF        ;INDICATE THAT WE ARE WITHIN A TEST.
5974 031556 012767 000001 152202      MOV @1,ERRTYP          ;SET UP DEVICE FATAL INDICATOR IN ERROR TYPE.
5975 031564 012767 003101 152176      MOV @1601,ERRNBR        ;SET UP ERROR NUMBER IN THE ERROR TABLE.
5976 031572 012767 012443 152172      MOV @EM1604,ERRMSG      ;SET UP ERROR MESSAGE FOR TEST IN ERROR TABLE.
5977 031600 005067 150656      CLR ERSNRF                    ;CLEAR THE ERROR SUMMARY FLAGS.
5978 031604 012700 002464      MOV @ERCNTB,R0
5979 031610 004767 166362      JSR PC,CLR16W                ;CLEAR THE ERROR COUNTER TABLE.
5980
5981 ; RESET THE DUT TO A KNOWN STATE, DO NOT REMOVE THE STATUS CODES FROM THE FIFO.
5982 ; CLEAR TX AND RX INTERRUPT ENABLE BITS IN THE CSR.
5983 ; THIS SUBROUTINE REPORTS ERRORS >>>> 1601 <<<<.
5984
5985 031614 004767 167634      JSR PC,RESETT                ;RESET THE DMV-11, REPORT ANY ERRORS FOUND.
5986 031620 103402      BCS .+6                            ;FATAL RESET ERROR? NO, CONTINUE WITH TEST.
5987 031622 000167 000116      JMP 608                      ;YES, EXIT THE TEST.
5988
5989 ;
5990 ; VERIFY READ/WRITE CAPABILITY TO INDIRECT ADDRESS FIELD OF CSR
5991
5991 031626 005267 152136      INC ERRNBR                    ;SET THE ERROR REPORT NUMBER TO 1602.
5992 031632 012702 000017      MOV @17,R2                  ;SET LOOP COUNT.
5993 031636 016704 150402      MOV CSRA,R4                  ;GET CSR ADDRESS.
5994 031642 010214      2$: MOV R2,(R4)                    ;WRITE COUNT TO CSR.
5995 031644 011401      MOV (R4),R1                        ;READ BACK THE CONTENTS OF THE CSR
5996 031646 042761 177760      BIC @177760,R1              ;MASK OUT ALL BUT THE IND.ADR.REG FIELD.
5997 031652 020102      CMP R1,R2                          ;CHECK FOR CORRECT DATA WRITTEN/READ.
5998 031654 001406      BEQ 4$                              ;IS EXPECTED DATA BAD? NO, SKIP ERROR REPORT.
5999 ;REPORT "BAD BIT(S) IN DEVICE CSR REGISTER FOR LINE 0 (D)."
6000 031656 012767 016710 152110      MOV @ER1601,ERRBLK      ;SELECT THE PROPER ERROR REPORT ROUTINE.
6001 031664 005003      CLR R3                              ;SET OFFSET TO 0 TO CAUSE REPORT OF CSR REG.
6002 031666 005005      CLR R5                              ;CAUSE REPORT OF LINE 0.
6003 031670      ERROR                                     ; >>>> ERROR # 1602 <<<<
031670 104460      TRAP C$ERROR
6004 031672 005302      4$: DEC R2                          ;DECREMENT LOOP COUNT/IND.ADD.REG ADDRESS.
6005 031674 002362      BGE 2$                              ;LOOP BACK TO TEST NEXT ADDRESS IF NOT DONE.
6006
6007 ; WRITE AND VERIFY 16 DATA PATTERNS IN ALL USED BITS OF ALL REGISTERS ON ALL
6008 ; ACTIVE LINES. BEFORE WRITING EACH PATTERN, CLEAR ALL THE BITS.
6009 ; REGTST ROUTINE REPORTS ERRORS WITH NUMBERS >>>> ERROR 1603 1605 <<<<.
6010
6011 031676 005267 152066      INC ERRNBR                    ;SET THE ERROR NUMBER TO 1603.
6012 031702 005003      CLR R3                              ;INDICATE THAT WORD ACCESSES ARE TO BE USED.

```

HARDWARE TEST

- REGWRW

```

6013 031704 012704 000002      MOV    #2,R4      ;INDICATE R/W ACCESS, CLEAR FIRST.
6014 031710 004767 167302      JSR    PC,REGTST ;WRITE AND VERIFY DATA PATTERNS.
6015                               ;.
6016       ; WRITE AND VERIFY 16 DATA PATTERNS IN ALL USED BITS OF ALL REGISTERS ON ALL
6017       ; ACTIVE LINES. BEFORE WRITING EACH PATTERN, SET ALL THE BITS.
6018       ; REGTST ROUTINE REPORTS ERRORS WITH NUMBERS >>>>> ERROR 1606 1608 <<<<<.
6019       ;
6020 031714 012767 003106 152046      MOV    #1606.,ERRNBR ;SET UP ERROR NUMBER FOR REGTST ROUTINE.
6021 031722 005003              CLR    R3      ;INDICATE THAT WORD ACCESSES ARE TO BE USED.
6022 031724 005404              NEG    R4      ;INDICATE R/W ACCESS, SET FIRST.
6023 031726 004767 167264      JSR    PC,REGTST ;WRITE AND VERIFY DATA PATTERNS.
6024                               ;.
6025       ; PRINT ERROR SUMMARY REPORTS IF NECESSARY.
6026       ; THE FOLLOWING ROUTINE REPORTS ERRORS WITH NUMBER >>>>> ERROR # 1609 <<<<<
6027       ;
6028 031732 012767 003111 152030      MOV    #1609.,ERRNBR ;SET UP ERROR NUMBER FOR NEXT RTN.
6029 031740 004767 167462      JSR    PC,REPSMR  ;REPORT ERROR SUMMARY IF NECESSARY.
6030 031744 005067 150402      603:   CLR    CTRLCF  ;INDICATE THAT WE COMPLETED THE TEST.
6031 031750              ENDTST
      031750
      031750 104401

```

L10044: TRAP C8ETST

HARDWARE TEST

REGWRM

```

6037 .SBTTL HARDWARE TEST REGWRM
6038 ;* .....
6039 ;* DEVICE REGISTER WORD ACCESS READ/MODIFY/WRITE TEST
6040 ;*
6041 ;* THIS TEST VERIFIES THAT THE DEVICE REGISTERS CAN BE WRITTEN CORRECTLY
6042 ;* USING WORD READ/MODIFY/WRITE ACCESSES.
6043 ;* .....
6044 BGNTEST
6045
6046 T16::
6047 SETPRI @PRI05 ;ALLOW THE LTC TO INTERRUPT.
6048 MOV @PRI05,R0
6049 TRAP C:SPRI
6050
6051 TNUM -- TNUM + 1 ;INCREMENT THE ASSEMBLY TIME TEST COUNTER.
6052 MOV @TNUM,TSTNUM ;SET UP THE TEST NUMBER. (17)
6053 MOV @1,CIRLCF ;INDICATE THAT WE ARE WITHIN A TEST.
6054 MOV @1,ERRTYP ;SET UP DEVICE FATAL INDICATOR IN ERROR TYPE.
6055 MOV @1701,ERRNBR ;SET UP ERROR NUMBER IN THE ERROR TABLE.
6056 MOV @EM1701,ERRMSG ;SET UP ERROR MESSAGE FOR TEST IN ERROR TABLE.
6057 CLR ERSUMF ;CLEAR THE ERROR SUMMARY FLAGS.
6058 MOV @ERCNTB,R0
6059 JSR PC.CLR16W ;CLEAR THE ERROR COUNTER TABLE.
6060
6061 ;* RESET THE DUT TO A KNOWN STATE. DO NOT REMOVE THE STATUS CODES FROM THE FIFO.
6062 ;* CLEAR TX AND RX INTERRUPT ENABLE BITS IN THE CSR.
6063 ;* THIS SUBROUTINE REPORTS ERRORS >>>> 1701 <<<<.
6064
6065 JSR PC.RESETT ;RESET THE DMV 11. REPORT ANY ERRORS FOUND.
6066 BCS .+6 ;FATAL RESET ERROR? NO, CONTINUE WITH TEST.
6067 JMP 608 ;YES, EXIT THE TEST.
6068
6069 ;* VERIFY READ/MODIFY/WRITE CAPABILITY TO INDIRECT ADDRESS FIELD OF CSR
6070 ;*
6071 INC ERRNBR ;SET THE ERROR REPORT NUMBER TO 1702.
6072 MOV @17,R2 ;SET LOOP COUNT.
6073 MOV CSRA,R4 ;GET CSR ADDRESS.
6074 28: BIC @17,(R4) ;CLEAR THE DUT CSR USING READ/MODIFY/WRITE.
6075 BIS R2,(R4) ;WRITE COUNT TO CSR USING READ/MODIFY/WRITE.
6076 MOV (R4),R1 ;READ BACK THE CONTENTS OF THE CSR
6077 BIC @177760,R1 ;MASK OUT ALL BUT THE IND.ADR.REG FIELD.
6078 CMP R1,R2 ;CHECK FOR CORRECT DATA WRITTEN/READ.
6079 BEQ 48 ;IS EXPECTED DATA BAD? NO, SKIP ERROR REPORT.
6080 ;REPORT "BAD BIT(S) IN DEVICE CSR REGISTER FOR LINE 0 (D)."
6081 MOV @ER1601,ERRBLK ;SELECT THE PROPER ERROR REPORT ROUTINE.
6082 CLR R3 ;SET OFFSET TO 0 TO CAUSE REPORT OF CSR REG.
6083 CLR R5 ;CAUSE REPORT OF LINE 0.
6084 ERROR ;>>>> ERROR @ 1702 <<<<
6085
6086 48: DEC R2 ;DECREMENT LOOP COUNT/IND.ADD.REG ADDRESS.
6087 BGE 28 ;LOOP BACK TO TEST NEXT ADDRESS IF NOT DONE.
6088
6089 ;* WRITE AND VERIFY 16 DATA PATTERNS IN ALL USED BITS OF ALL REGISTERS ON ALL
6090 ;* ACTIVE LINES USING R/M/W. BEFORE WRITING EACH PATTERN, CLEAR ALL THE BITS.
6091 ;* REGTST ROUTINE REPORTS ERRORS WITH NUMBERS >>>> ERROR 1703 - 1705 <<<<.
6092 ;
6093 INC ERRNBR ;SET THE ERROR NUMBER TO 1703.

```

HARDWARE TEST

- REGWRM -

```

6086 032124 005003          CLR    R3          ;INDICATE THAT WORD ACCESSES ARE TO BE USED.
6087 032126 012704 000001    MOV    #1,R4      ;INDICATE R/M/W ACCESS, CLEAR FIRST.
6088 032132 004767 167060    JSR     PC,REGTST  ;WRITE AND VERIFY DATA PATTERNS.
6089
6090      ; WRITE AND VERIFY 16 DATA PATTERNS IN ALL USED BITS OF ALL REGISTERS ON ALL
6091      ; ACTIVE LINES USING R/M/W. BEFORE WRITING EACH PATTERN, SET ALL THE BITS.
6092      ; REGTST ROUTINE REPORTS ERRORS WITH NUMBERS >>>>> ERROR 1706 1708 <<<<<.
6093
6094 032136 012767 003252 151624    MOV    #1706.,ERRNBR ;SET UP ERROR NUMBER FOR REGTST ROUTINE.
6095 032144 005003          CLR    R3          ;INDICATE THAT WORD ACCESSES ARE TO BE USED.
6096 032146 005404          NEG    R4          ;INDICATE R/M/W ACCESS, SET FIRST.
6097 032150 004767 167042    JSR     PC,REGTST  ;WRITE AND VERIFY DATA PATTERNS.
6098
6099      ; PRINT ERROR SUMMARY REPORTS IF NECESSARY.
6100      ; THE FOLLOWING ROUTINE REPORTS ERRORS WITH NUMBER >>>>> ERROR # 1709 <<<<<
6101
6102 032154 012767 003255 151606    MOV    #1709.,ERRNBR ;SET UP ERROR NUMBER FOR NEXT RTN.
6103 032162 004767 167240    JSR     PC,REPSMR  ;REPORT ERROR SUMMARY IF NECESSARY.
6104 032166 005067 150160    604:   CLR    CTRLCF  ;INDICATE THAT WE COMPLETED THE TEST.
6105 032172          ENDTST
      032172
      032172 104401

```

L10045: TRAP C8E757

HARDWARE TEST

REGPRW

```

6107 .SBTTL HARDWARE TEST - REGPRW
6108 ;... *****
6109 ;* DEVICE REGISTER BYTE ACCESS READ AND WRITE TEST
6110 ;*
6111 ;* THIS TEST VERIFIES THAT THE DEVICE REGISTERS CAN BE READ AND WRITTEN
6112 ;* CORRECTLY USING BYTE ACCESSES.
6113 ;*
6114 ; *****
6115
6116 BGNTEST
6117 032174 012700 000240 SETPRI @PRI05 ;ALLOW THE LTC TO INTERRUPT.
6118 032174 104441 MOV @PRI05,R0
6119 032174 000021 TRAP C$SPRI
6120 032202 012767 000021 TNUM ** TNUM + 1 ;INCREMENT THE ASSEMBLY TIME TEST COUNTER.
6121 032210 012767 177777 MOV @TNUM,T$TNUM ;SET UP THE TEST NUMBER. (18)
6122 032216 012767 000001 MOV @1,CTRLCF ;INDICATE THAT WE ARE WITHIN A TEST.
6123 032224 012767 003411 MOV @1,ERRTYP ;SET UP DEVICE FATAL INDICATOR IN ERROR TYPE.
6124 032232 012767 012566 MOV @1801,ERRNBR ;SET UP ERROR NUMBER IN THE ERROR TABLE.
6125 032240 005067 150216 MOV @EM1801,ERRMSG ;SET UP ERROR MESSAGE FOR TEST IN ERROR TABLE.
6126 032244 012700 002464 CLR ERSRWF ;CLEAR THE ERROR SUMMARY FLAGS.
6127 032250 004767 165722 MOV @ERCNTB,R0
6128 JSR PC,CLR16W ;CLEAR THE ERROR COUNTER TABLE.
6129 ;*
6130 ; RESET THE DUT TO A KNOWN STATE. DO NOT REMOVE THE STATUS CODES FROM THE FIFO.
6131 ; CLEAR TX AND RX INTERRUPT ENABLE BITS IN THE CSR.
6132 ; THIS SUBROUTINE REPORTS ERRORS >>>> 1801 <<<<.
6133 JSR PC,RESET ;RESET THE DMV 11. REPORT ANY ERRORS FOUND.
6134 BCS .+6 ;FATAL RESET ERROR? NO, CONTINUE WITH TEST.
6135 JMP 608 ;YES, EXIT THE TEST.
6136 MOV @1802,ERRNBR ;SET THE ERROR REPORT NUMBER TO 1802.
6137 ;*
6138 ; VERIFY READ/WRITE CAPABILITY TO INDIRECT ADDRESS FIELD OF CSR.
6139 ; USE BYTE ACCESSES.
6140 MOV @17,R2 ;SET LOOP COUNT.
6141 MOV CSRA,R4 ;GET CSR ADDRESS.
6142 MOV R2,(R4) ;WRITE COUNT TO CSR.
6143 MOV R4,R1 ;READ BACK THE CONTENTS OF THE CSR
6144 BIC @177760,R1 ;MASK OUT ALL BUT THE IND.ADR.REG FIELD.
6145 CMP R1,R2 ;CHECK FOR CORRECT DATA WRITTEN/READ.
6146 BEQ 48 ;IS EXPECTED DATA BAD? NO, SKIP ERROR REPORT.
6147 ;REPORT "BAD BIT(S) IN DEVICE CSR REGISTER FOR LINE 0 (D)."
6148 MOV @ER1601,ERRBLK ;SELECT THE PROPER ERROR REPORT ROUTINE.
6149 CLR R3 ;SET OFFSET TO 0 TO CAUSE REPORT OF CSR REG.
6150 CLR R5 ;CAUSE REPORT OF LINE 0.
6151 ERROR ; >>>> ERROR # 1802 <<<<
6152 032332 104460 TRAP C$ERROR
6153 032334 005302 48: DEC R2 ;DECREMENT LOOP COUNT/IND.ADD.REG ADDRESS.
6154 032336 002362 BGE 28 ;LOOP BACK TO TEST NEXT ADDRESS IF NOT DONE.
6155 ;*
6156 ; WRITE AND VERIFY 16 DATA PATTERNS IN ALL USED BITS OF ALL LOWER BYTES OF ALL
6157 ; REGISTERS ON ALL ACTIVE LINES. USE READ/WRITE ACCESSES. BEFORE WRITING
6158 ; EACH PATTERN, CLEAR ALL THE USED BITS OF ALL ACTIVE REGISTERS.
6159 ; REGIST ROUTINE REPORTS ERRORS WITH NUMBERS >>>> ERROR 1803 1805 <<<<.

```

HARDWARE TEST

- RESRW -

```

6160 032340 005267 151424      INC  ERRNBR      ;SET THE ERROR NUMBER TO 1803.
6161 032344 012703 177777      MOV  #1,R3      ;INDICATE THAT LO BYTE ACCESSES ARE TO BE USED.
6162 032350 012704 000002      MOV  #2,R4      ;INDICATE R/W ACCESS, CLEAR FIRST.
6163 032354 004767 166636      JSR   PC,REGTST  ;WRITE AND VERIFY DATA PATTERNS.
6164
6165      ;*
6166      ; WRITE AND VERIFY 16 DATA PATTERNS IN ALL USED BITS OF ALL HIGH BYTES OF ALL
6167      ; REGISTERS ON ALL ACTIVE LINES. USE READ/WRITE ACCESSES. BEFORE WRITING
6168      ; EACH PATTERN, CLEAR ALL THE USED BITS OF ALL ACTIVE REGISTERS.
6169      ; REGTST ROUTINE REPORTS ERRORS WITH NUMBERS >>>> ERROR 1806 - 1808 <<<<.
6170 032360 012767 003416 151402      MOV  #1806.,ERRNBR  ;SET UP ERROR NUMBER FOR REGTST ROUTINE.
6171 032366 005403      NEG  R3      ;INDICATE THAT HI BYTE ACCESSES ARE TO BE USED.
6172 032370 004767 166622      JSR   PC,REGTST  ;WRITE AND VERIFY DATA PATTERNS.
6173
6174      ;*
6175      ; WRITE AND VERIFY 16 DATA PATTERNS IN ALL USED BITS OF ALL LOWER BYTES OF ALL
6176      ; REGISTERS ON ALL ACTIVE LINES. USE READ/WRITE ACCESSES. BEFORE WRITING
6177      ; EACH PATTERN, SET ALL THE USED BITS OF ALL ACTIVE REGISTERS.
6178      ; REGTST ROUTINE REPORTS ERRORS WITH NUMBERS >>>> ERROR 1809 - 1811 <<<<.
6179 032374 012767 003421 151366      MOV  #1809.,ERRNBR  ;SET UP ERROR NUMBER FOR REGTST ROUTINE.
6180 032402 005403      NEG  R3      ;INDICATE THAT LO BYTE ACCESSES ARE TO BE USED.
6181 032404 005404      NEG  R4      ;INDICATE R/W ACCESS, SET FIRST.
6182 032406 004767 166604      JSR   PC,REGTST  ;WRITE AND VERIFY DATA PATTERNS.
6183
6184      ;*
6185      ; WRITE AND VERIFY 16 DATA PATTERNS IN ALL USED BITS OF ALL HIGH BYTES OF ALL
6186      ; REGISTERS ON ALL ACTIVE LINES. USE READ/WRITE ACCESSES. BEFORE WRITING
6187      ; EACH PATTERN, SET ALL THE USED BITS OF ALL ACTIVE REGISTERS.
6188      ; REGTST ROUTINE REPORTS ERRORS WITH NUMBERS >>>> ERROR 1812 - 1814 <<<<.
6189 032412 012767 003424 151350      MOV  #1812.,ERRNBR  ;SET UP ERROR NUMBER FOR REGTST ROUTINE.
6190 032420 005403      NEG  R3      ;INDICATE THAT HI BYTE ACCESSES ARE TO BE USED.
6191 032422 004767 166570      JSR   PC,REGTST  ;WRITE AND VERIFY DATA PATTERNS.
6192
6193      ;*
6194      ; PRINT ERROR SUMMARY REPORTS IF NECESSARY.
6195      ; THE FOLLOWING ROUTINE REPORTS ERRORS WITH NUMBER >>>> ERROR # 1815 <<<<
6196 032426 012767 003427 151334      MOV  #1815.,ERRNBR  ;SET UP ERROR NUMBER FOR NEXT RTN.
6197 032434 004767 166766      JSR   PC,REPSMR  ;REPORT EPROR SUMMARY IF NECESSARY.
6198 032440 005067 147706      CLR  CTRLCF  ;INDICATE THAT WE COMPLETED THE TEST.
6199 032444
        032444
        032444 104401

```

L10046:

TRAP C\$ETST

HARDWARE TEST

REGBRM -

```

6201 .SBTTL HARDWARE TEST                      REGBRM
6202 ; .. *****
6203 ; *                               DEVICE REGISTER BYTE ACCESS READ/MODIFY/WRITE TEST
6204 ; *
6205 ; * THIS TEST VERIFIES THAT THE DEVICE REGISTERS CAN BE READ AND WRITTEN
6206 ; * CORRECTLY USING BYTE ACCESSES IN READ/MODIFY/WRITE MODE.
6207 ; *
6208 ; .. *****
6209
6210 032446      BGNTEST
6211 032446      SETPRI @PRI05                  ;ALLOW THE LTC TO INTERRUPT.
6212 032446 012700 000240                      ;
6213 032446 104441                                MOV @PRI05,R0
6214 032452 000022                                TRAP C$SPRI
6215 032454 012767 000022 147644                TNUM ** TNUM + 1 ;INCREMENT THE ASSEMBLY TIME TEST COUNTER.
6216 032462 012767 177777 147662                MOV @TNUM,TSTNUM ;SET UP THE TEST NUMBER. (19)
6217 032470 012767 000001 151270                MOV @1,CTRLCF ;INDICATE THAT WE ARE WITHIN A TEST.
6218 032476 012767 003555 151264                MOV @1,ERRTYP ;SET UP DEVICE FATAL INDICATOR IN ERROR TYPE.
6219 032504 012767 012634 151260                MOV @1901,ERRNBR ;SET UP ERROR NUMBER IN THE ERROR TABLE.
6220 032512 005067 147744                        MOV @EM1901,ERRMSG ;SET UP ERROR MESSAGE FOR TEST IN ERROR TABLE.
6221 032516 012700 002464                        CLR ERSRFF ;CLEAR THE ERROR SUMMARY FLAGS.
6222 032522 004767 165450                        MOV @ERCNTB,R0
6223 ; ..
6224 ; RESET THE DUT TO A KNOWN STATE, DO NOT REMOVE THE STATUS CODES FROM THE FIFO.
6225 ; CLEAR TX AND RX INTERRUPT ENABLE BITS IN THE CSR.
6226 ; THIS SUBROUTINE REPORTS ERRORS >>>> 1901 <<<<.
6227 ;
6228 032526 004767 166722                        JSR PC,RESETT ;RESET THE DHV-11, REPORT ANY ERRORS FOUND.
6229 032532 103402                                BCS .+6 ;FATAL RESET ERROR? NO, CONTINUE WITH TEST.
6230 032534 000167 000156                        JMP 608 ;YES, EXIT THE TEST.
6231 032540 012767 003556 151222                MOV @1902,ERRNBR ;SET THE ERROR REPORT NUMBER TO 1902.
6232 ; ..
6233 ; VERIFY READ/WRITE CAPABILITY TO INDIRECT ADDRESS FIELD OF CSR.
6234 ; USE BYTE ACCESSES.
6235 ;
6236 032546 012702 000017                        MOV @17,R2 ;SET LOOP COUNT.
6237 032552 016704 147466                        MOV CSRA,R4 ;GET CSR ADDRESS.
6238 032556 142714 000017 28: BICB @17,(R4) ;CLEAR THE DUT CSR USING READ/MODIFY/WRITE.
6239 032562 150214                                BISB R2,(R4) ;WRITE COUNT TO CSR USING READ/MODIFY/WRITE.
6240 032564 111401                                MOVB (R4),R1 ;READ BACK THE CONTENTS OF THE CSR
6241 032566 042701 177760                        BIC @177760,R1 ;MASK OUT ALL BUT THE IND.ADR.REG FIELD.
6242 032572 020102                                CMP R1,R2 ;CHECK FOR CORRECT DATA WRITTEN/READ.
6243 032574 001406                                BEQ 48 ;IS EXPECTED DATA BAD? NO, SKIP ERROR REPORT.
6244 032576 012767 016710 151170                ;REPORT "BAD BIT(S) IN DEVICE CSR REGISTER FOR LINE 0 (D)."
6245 032604 005003                                MOV @ER1601,ERRBLK ;SELECT THE PROPER ERROR REPORT ROUTINE.
6246 032606 005005                                CLR R3 ;SET OFFSET TO 0 TO CAUSE REPORT OF CSR REG.
6247 032610                                CLR R5 ;CAUSE REPORT OF LINE 0.
6248 032612                                ERROR ; >>>> ERROR # 1902 <<<<
6249 032610 104460                                TRAP C$ERROR
6250 032612 005302 48: DEC R2 ;DECREMENT LOOP COUNT/IND.ADD.REG ADDRESS.
6251 032614 002360                                BGE 28 ;LOOP BACK TO TEST NEXT ADDRESS IF NOT DONE.
6252 ; ..
6253 ; WRITE AND VERIFY 16 DATA PATTERNS IN ALL USED BITS OF ALL LOWER BYTES OF ALL
; REGISTERS ON ALL ACTIVE LINES. USE READ/MODIFY/WRITE ACCESSES. BEFORE
; WRITING EACH PATTERN, CLEAR ALL THE USED BITS OF ALL ACTIVE REGISTERS.
; REGIST ROUTINE REPORTS ERRORS WITH NUMBERS >>>> ERROR 1903 1905 <<<<.

```

HARDWARE TEST

- REGPRM -

```

6254
6255 032616 005267 151146      ;      INC      ERRNBR      ;SET THE ERROR NUMBER TO 1903.
6256 032622 012703 177777      ;      MOV      @ 1,R3      ;INDICATE THAT LO BYTE ACCESSSES ARE TO BE USED.
6257 032626 012704 000001      ;      MOV      @1,R4      ;INDICATE R/M/W ACCESS, CLEAR FIRST.
6258 032632 004767 166360      ;      JSR      PC,REGTST    ;WRITE AND VERIFY DATA PATTERNS.
6259
6260      ;*
6261      ; WRITE AND VERIFY 16 DATA PATTERNS IN ALL USED BITS OF ALL HIGH BYTES OF ALL
6262      ; REGISTERS ON ALL ACTIVE LINES. USE READ/MODIFY/WRITE ACCESSSES. BEFORE
6263      ; WRITING EACH PATTERN, CLEAR ALL THE USED BITS OF ALL ACTIVE REGISTERS.
6264      ; REGTST ROUTINE REPORTS ERRORS WITH NUMBERS >>>> ERROR 1906 - 1908 <<<<.
6265 032636 012767 003562 151124      ;      MOV      @1906.,ERRNBR ;SET UP ERROR NUMBER FOR REGTST ROUTINE.
6266 032644 005403              ;      NEG      R3      ;INDICATE THAT HI BYTE ACCESSSES ARE TO BE USED.
6267 032646 004767 166344      ;      JSR      PC,REGTST    ;WRITE AND VERIFY DATA PATTERNS.
6268
6269      ;*
6270      ; WRITE AND VERIFY 16 DATA PATTERNS IN ALL USED BITS OF ALL LOWER BYTES OF ALL
6271      ; REGISTERS ON ALL ACTIVE LINES. USE READ/MODIFY/WRITE ACCESSSES. BEFORE
6272      ; WRITING EACH PATTERN, SET ALL THE USED BITS OF ALL ACTIVE REGISTERS.
6273      ; REGTST ROUTINE REPORTS ERRORS WITH NUMBERS >>>> ERROR 1909 - 1911 <<<<.
6274 032652 012767 003565 151110      ;      MOV      @1909.,ERRNBR ;SET UP ERROR NUMBER FOR REGTST ROUTINE.
6275 032660 005403              ;      NEG      R3      ;INDICATE THAT LO BYTE ACCESSSES ARE TO BE USED.
6276 032662 005404              ;      NEG      R4      ;INDICATE R/M/W ACCESS, SET FIRST.
6277 032664 004767 166326      ;      JSR      PC,REGTST    ;WRITE AND VERIFY DATA PATTERNS.
6278
6279      ;*
6280      ; WRITE AND VERIFY 16 DATA PATTERNS IN ALL USED BITS OF ALL HIGH BYTES OF ALL
6281      ; REGISTERS ON ALL ACTIVE LINES. USE READ/MODIFY/WRITE ACCESSSES. BEFORE
6282      ; WRITING EACH PATTERN, SET ALL THE USED BITS OF ALL ACTIVE REGISTERS.
6283      ; REGTST ROUTINE REPORTS ERRORS WITH NUMBERS >>>> ERROR 1912 - 1914 <<<<.
6284 032670 012767 003570 151072      ;      MOV      @1912.,ERRNBR ;SET UP ERROR NUMBER FOR REGTST ROUTINE.
6285 032676 005403              ;      NEG      R3      ;INDICATE THAT HI BYTE ACCESSSES ARE TO BE USED.
6286 032700 004767 166312      ;      JSR      PC,REGTST    ;WRITE AND VERIFY DATA PATTERNS.
6287
6288      ;*
6289      ; PRINT ERROR SUMMARY REPORTS IF NECESSARY.
6290      ; THE FOLLOWING ROUTINE REPORTS ERRORS WITH NUMBER >>>> ERROR @ 1915 <<<<
6291 032704 012767 003573 151056      ;      MOV      @1915.,ERRNBR ;SET UP ERROR NUMBER FOR NEXT RTN.
6292 032712 004767 166510      ;      JSR      PC,REPSMR    ;REPORT ERROR SUMMARY IF NECESSARY.
6293 032716 005067 147430      601:  ;      CLR      CTRLCF      ;INDICATE THAT WE COMPLETED THE TEST.
6294 032722
        032722
        032722 104401

```

L10047:

TRAP C0ETST

HARDWARE TEST

IDBIT

```

6296 .SBTTL HARDWARE TEST IDBIT
6297 ;... *****
6298 ;* DEVICE REGISTER ID BIT TEST
6299 ;*
6300 ;* THIS TEST VERIFIES THAT THE DUT STAT REGISTER ID BIT READS AS CLEAR.
6301 ;*
6302 ;- *****
6303
6304 BGNTST
6305 032724 012700 000240 SETPRI #PRI05 ;ALLOW THE LTC TO INTERRUPT. T19::
032724 104441 MOV #PRI05,R0
032730 000023 TRAP C$SPR1
6306 TNUM == TNUM + 1 ;INCREMENT THE ASSEMBLY TIME TEST COUNTER.
6307 032732 012767 000023 147366 MOV #TNUM,TSTNUM ;SET UP THE TEST NUMBER. (20)
6308 032740 012767 177777 147404 MOV #-1,CTRLCF ;INDICATE THAT WE ARE IN A TEST.
6309 032746 012767 000001 151012 MOV #1,ERRTYP ;SET UP DEVICE FATAL INDICATOR IN ERROR TYPE.
6310 032754 012767 003721 151006 MOV #2001,ERRNBR ;SET UP ERROR NUMBER IN THE ERROR TABLE.
6311 032762 012767 012711 151002 MOV #EM2001,ERRMSG ;SET UP ERROR MESSAGE FOR TEST IN ERROR TABLE.
6312 ;*
6313 ; RESET THE DUT TO A KNOWN STATE, DO NOT REMOVE THE STATUS CODES FROM THE FIFO.
6314 ; CLEAR TX AND RX INTERRUPT ENABLE BITS IN THE CSR.
6315 ; THIS SUBROUTINE REPORTS ERRORS >>>> 2001 <<<<.
6316 ;-
6317 032770 004767 166460 JSR PC,RESETT ;RESET THE DMV 11, REPORT ANY ERRORS FOUND.
6318 032774 103016 BCC 60$ ;FATAL RESET ERROR? YES, EXIT THE TEST.
6319 ;*
6320 ; READ THE STAT REGISTER ID BIT AND VERIFY THAT IT IS CLEAR.
6321 ;
6322 032776 017701 147250 MOV #STAT0,R1 ;READ THE STAT REGISTER CONTENTS.
6323 033002 032701 000400 BIT #BIT8,R1 ;CHECK THE ID BIT.
6324 033006 001411 BEQ 60$ ;ID BIT CLEAR? YES, EXIT THE TEST.
6325 033010 012767 003722 150752 MOV #2002,ERRNBR ;NO, SET THE ERROR REPORT NUMBER TO 2002.
6326 033016 012701 012753 MOV #EM2002,R1 ;GET THE PROPER ERROR MESSAGE.
6327 033022 012767 016466 150744 MOV #ER0503,ERRBLK ;SELECT THE PROPER ERROR REPORT ROUTINE.
6328 033030 ERROR ;ERROR NUMBER >>>> 2002 <<<<
033030 104460 TRAP C$ERROR
6329 033032 005067 147314 60$: CLR CTRLCF ;INDICATE THAT WE COMPLETED THE TEST.
6330 033036 ENDTST
033036 104401 L10050: TRAP C$ETST

```

HARDWARE TEST

- NOTXDV -

```

6332 .SBTTL  HARDWARE TEST          - NOTXDV
6333 ;* *****
6334 ;*          NO TX DATA VALID/NO TX ACTION TEST
6335 ;*          THIS TEST VERIFIES THAT IF A DATA WORD IS WRITTEN WITHOUT THE
6336 ;*          TX DATA_VALID BIT SET, NO TX ACTION WILL BE GENERATED.
6337 ;*          TO ENSURE DATA IS NOT ACCIDENTALLY TRANSMITTED, THE TEST IS PERFORMED
6338 ;*          IN INTERNAL LOOPBACK, AND ON ALL ACTIVE LINES.
6339 ;* *****
6340 ; - *****
6341 033040          BGNTST
6342 033040          SETPRI  @PRI05          ;ALLOW LTC INTERRUPTS.          T20::
        033040          012700  000240          MOV          @PRI05,R0
        033044          104441          TRAP          C:SPRI
6343          000024          TNUM == TNUM + 1          ;INCREMENT THE ASSEMBLY TIME TEST COUNTER.
6344 033046          012767  000024  147252          MOV          @TNUM,TSTNUM          ;SET UP THE TEST NUMBER.          (21)
6345 033054          012767  177777  147270          MOV          @ 1,CTRLCF          ;INDICATE THAT WE ARE IN A TEST.
6346 033062          012767  000001  150676          MOV          @1,ERRTYP          ;SET ERROR TYPE AS FATAL IN ERROR TABLE.
6347 033070          012767  004065  150672          MOV          @2101,ERRNBR          ;SET THE FIRST ERROR NUMBER IN ERROR TABLE.
6348 033076          012767  013026  150666          MOV          @EM2101,ERRMSG          ;SET ERROR MESSAGE ADDRESS IN ERRTABL.
6349 033104          012767  017420  150662          MOV          @ER9101,ERRBLK          ;SELECT THE CORRECT ERROR REPORTING ROUTINE.
6350 ;*
6351 ; RESET THE DUT TO A KNOWN STATE, REMOVE THE STATUS CODES FROM THE FIFO.
6352 ; CLEAR TX AND RX INTERRUPT ENABLE BITS IN THE CSR.
6353 ; THIS SUBROUTINE REPORTS ERROR >>>> 2101 <<<<.
6354 ; -
6355 033112          004767  165036          JSR          PC,CLNRST          ;RESET THE DMV-11, REPORT ANY ERRORS FOUND.
6356 033116          103054          BCC          60$          ;RESET FAILURE?, ABORT THIS TEST.
6357 033120          005267  150644          INC          ERRNBR          ;SET THE ERROR NUMBER TO 2102.
6358 ;*
6359 ; SET INTERNAL LOOPBACK ON ALL ACTIVE LINES.
6360 ; SET LPR ON ALL LINES TO 38.4K BAUD, 8 BITS PER CHARACTER, ODD PARITY,
6361 ; 2 STOP BITS.
6362 ; DISABLE TRANSMITTERS ON ALL ACTIVE LINES.
6363 ;
6364 033124          016705  147106          MOV          ACTLNS,R5          ;PASS THE ACTIVE LINE BIT MAP.
6365 033130          012700  000200          MOV          @200,R0          ;PASS THE LNCTRL CONTENTS.
6366 033134          004767  170370          JSR          PC,WTWLNLC          ;INITIALISE THE LNCTRL REGISTERS.
6367 033140          012700  177670          MOV          @177670,R0          ;PASS THE LPR CONTENTS.
6368 033144          004767  170434          JSR          PC,WTWLPR          ;INITIALISE THE LPR REGISTERS ON ALL LINES.
6369 033150          012704  000012          MOV          @10,R4          ;PASS DELAY TIME OF 10 MILLI SFCS.
6370 033154          004767  165116          JSR          PC,DELAY          ;WAIT FOR LNCTRL AND LPR REGS TO BE UPDATED.
6371 033160          004767  167410          JSR          PC,TXDSBL          ;DISABLE TRANSMITTERS ON ALL ACTIVE LINES.
6372 ;*
6373 ; TEST ALL ACTIVE LINES INDIVIDUALLY.
6374 ; WRITE A DATA WORD TO THE TXCHAR REGISTER WITH TX DATA_VALID CLEAR.
6375 ; VERIFY NO TX ACTION IS GENERATED.
6376 ;
6377 033164          016705  147046          MOV          ACTLNS,R5          ;GET THE ACTIVE LINE BIT MAP.
6378 033170          005004          CLR          R4          ;CLEAR THE LINE NUMBER COUNTER.
6379 033172          000241          CLC          ;CLEAR THE CARRY BIT PRIOR TO SHIFTING BIT MAP.
6380 033174          006005          ROR          R5          ;SHIFT THE BIT MAP INTO THE CARRY BIT.
6381 033176          103020          BCC          4$          ;DO NOT TEST THE LINE IF IT IS INACTIVE.
6382 ;*
6383 ; SELECT THE LINE UNDER TEST.
6384 ; WRITE DATA WORD (ASCII <LF>) TO TXCHR REGISTER WITH THE MOST SIGNIFICANT
6385 ; BIT (TX_DATA_VALID) CLEAR.

```

HARDWARE TEST

NOTXDV

```

6386
6387 033200 010477 147040      ;
6388 033204 012777 000012 147034      MOV    R4,@CSRA      ;SELECT THE LINE CURRENTLY UNDER TEST.
6389                                MOV    @12,@TXCHA      ;WRITE THE DATA WORD TO THE DUT'S TXCHAN REG.
6390                                ;
6391                                ; WAIT FOR A TX ACTION TO BE RETURNED, REPORT ERROR IF TX ACTION FOUND
6392                                ; BEFORE TIME OUT OCCURS.
6393 033212 012701 170002      MOV    @170002,R1      ;TEST BIT 15, TIMEOUT OF 2 MILLI SECS.
6394 033216 016702 147022      MOV    CSRA,R2      ;PASS THE ADDRESS OF THE REGISTER TO TEST.
6395 033222 004767 170032      JSR    PC,WAIBIS      ;WAIT FOR TX ACTION TO COME BACK.
6396 033226 103004      RCC    4$      ;SKIP ERROR REPORT IF TX-ACTION NOT FOUND.
6397
6398 033230 010401      MOV    R4,R1      ;PASS THE NUMBER OF CURRENT LINE UNDER TEST.
6399 033232 012702 C13071      MOV    @EM2102,R2      ;PASS THE ERROR MESSAGE TO BE REPORTED.
6400                                ; 'TX_ACT FOUND AFTER INVALID DATA WORD WRITTEN'
6401 033236      ERROR      ; >>>> ERROR #4102 <<<<<.
6402                                TRAP    C$ERROR
6403                                ;
6404                                ; VERIFY ALL ACTIVE LINES HAVE BEEN TESTED.
6405 033240 005204      4$:    INC    R4      ;INCREMENT THE LINE NUMBER COUNTER.
6406 033242 005705      TST    R5      ;ARE THERE ANY MORE ACTIVE LINES TO TEST?.
6407 033244 001352      BNE    2$      ;YES; BRANCH TO TEST THE NEXT LINE.
6408 033246 000400      BR     60$      ;NO; EXIT THIS TEST.
6409
6410 033250 005067 147076      60$:    CLR    CTRLCF      ;INDICATE THAT WE ARE NOT WITHIN A TEST.
6411 033254      ENDTST
6412 033254      L10051:
6413 033254 104401      TRAP    C$ETST

```

HARDWARE TEST

TXDVAL

```

6413 .SBTTL HARDWARE TEST - TXDVAL
6414 ; .. *****
6415 ; * - TX_DATA_VALID/TX ACTION TEST
6416 ; * THIS TEST VERIFIES THAT IF A DATA WORD IS WRITTEN TO THE TXCHAR REGISTER
6417 ; * WITH THE TX_DATA_VALID BIT SET, A CORRESPONDING TX_ACTION WILL BE
6418 ; * GENERATED.
6419 ; * TO ENSURE DATA IS NOT ACCIDENTALLY TRANSMITTED, THE TEST IS PERFORMED
6420 ; * IN INTERNAL LOOPBACK, AND ON ALL ACTIVE LINES.
6421 ; *
6422 ; - *****
6423
6424 033256 BGN1ST
6425 033256 SETPRI @PRI05 ;ALLOW LTC INTERRUPTS. T21::
6426 033256 012700 000240 MOV @PRI05,RC
6427 033262 104441 TRAP C$SPRI
6428 033262 000025 TNUM == TNUM + 1 ;INCREMENT THE ASSEMBLY TIME TEST COUNTER.
6429 033264 012767 000025 147034 MOV @TNUM,TSTNUM ;SET UP THE TEST NUMBER. (22)
6430 033272 012767 177777 147052 MOV @-1,CTRLCF ;INDICATE THAT WE ARE IN A TEST.
6431 033300 012767 000001 150460 MOV @1,ERRTYP ;SET ERROR TYPE AS FATAL IN ERROR TABLE.
6432 033306 012767 004231 150454 MOV @2201.,ERRNBR ;SET THE FIRST ERROR NUMBER IN ERROR TABLE.
6433 033314 012767 013165 150450 MOV @EM2201,ERRMSG ;SET ERROR MESSAGE ADDRESS IN ERTBL.
6434 033322 012767 017420 150444 MOV @ER9101,ERRBLK ;SELECT THE CORRECT ERROR REPORTING ROUTINE.
6435 ; *
6436 ; * RESET THE DUT TO A KNOWN STATE, REMOVE THE STATUS CODES FROM THE FIFO.
6437 ; * CLEAR TX AND RX INTERRUPT ENABLE BITS IN THE CSR.
6438 ; * THIS SUBROUTINE REPORTS ERROR >>>> 2201 <<<<.
6439 033330 004767 164620 JSR PC,CLNRST ;RESET THE DMV-11, REPORT ANY ERRORS FOUND.
6440 033334 103066 BCC 60$ ;RESET FAILURE?, ABORT THIS TEST.
6441 ; *
6442 ; * SET INTERNAL LOOPBACK ON ALL ACTIVE LINES.
6443 ; * SET LPR ON ALL LINES TO 38.4K BAUD, 8 BITS PER CHARACTER, ODD PARITY,
6444 ; * 2 STOP BITS.
6445 ; * DISABLE TRANSMITTERS ON ALL ACTIVE LINES.
6446 033336 016705 146674 MOV ACTLNS,R5 ;PASS THE ACTIVE LINE BIT MAP.
6447 033342 012700 000200 MOV @200,RC ;PASS THE LNCTRL CONTENTS.
6448 033346 004767 170156 JSR PC,WTWLNCR ;INITIALISE THE LNCTRL REGISTERS.
6449 033352 012700 177670 MOV @177670,RC ;PASS THE LPR CONTENTS.
6450 033356 004767 170222 JSR PC,WTWLPR ;INITIALISE THE LPR REGISTERS ON ALL LINES.
6451 033362 012704 000012 MOV @10.,R4 ;PASS DELAY TIME OF 10 MILLI-SECONDS.
6452 033366 004767 164704 JSR PC,DELAY ;WAIT FOR LNCTR AND LPR REGS TO BE UPDATED.
6453 033372 004767 167176 JSR PC,TXDSBL ;DISABLE TRANSMITTERS ON ALL ACTIVE LINES.
6454 ; *
6455 ; * TEST ALL ACTIVE LINES INDIVIDUALLY.
6456 ; * WRITE A DATA WORD TO THE TXCHAR REGISTER WITH TX_DATA_VALID SET.
6457 ; * VERIFY THAT A CORRESPONDING TX_ACTION IS GENERATED.
6458 ; *
6459 033376 016705 146634 MOV ACTLNS,R5 ;GET THE ACTIVE LINE BIT MAP.
6460 033402 005004 CLR R4 ;CLEAR THE LINE NUMBER COUNTER.
6461 033404 012767 004232 150356 2$: MOV @2202.,ERRNBR ;SET THE ERROR NUMBER TO 2202.
6462 033412 000241 CLC ;CLEAR THE CARRY BIT PRIOR TO SHIFTING BIT MAP.
6463 033414 006005 ROR R5 ;SHIFT THE BIT MAP INTO THE CARRY BIT.
6464 033416 103032 BCC 8$ ;DO NOT TEST THE LINE IF IT IS INACTIVE.
6465 ; *
6466 ; * SELECT THE LINE UNDER TEST.

```

HARDWARE TEST

TXDVAL

```

6467      ; WRITE DATA WORD (ASCII 115) TO TXCHAR REGISTER WITH THE MOST SIGNIFICANT
6468      ; BIT (TX DATA VALID) SET.
6469      ;
6470 033420 010477 146620      MOV    R4,0CSRA      ;SELECT THE LINE CURRENTLY UNDER TEST.
6471 033424 012777 100012 146614  MOV    #100012,0TXCHA ;WRITE THE DATA WORD TO THE DUT'S TXCHAR REG.
6472      ;
6473      ; WAIT FOR A TX ACTION TO BE RETURNED, REPORT ERROR IF NO TX ACTION
6474      ; FOUND BEFORE TIME OUT OCCURS.
6475      ;
6476 033432 012701 170002      MOV    #170002,R1      ;TEST BIT 15, TIMEOUT OF 2 MILLI SECS.
6477 033436 016702 146602      MOV    CSRA,R2      ;PASS THE ADDRESS OF THE REGISTER TO TEST.
6478 033442 004767 167612      JSR    PC,WAIBIS    ;WAIT FOR TX ACTION TO COME BACK.
6479 033446 103403      BCS    48      ;SKIP ERROR REPORT IF TX ACTION FOUND.
6480 033450 012702 013222      MOV    #0EM2202,R2    ;PASS THE ERROR MESSAGE TO BE REPORTED.
6481      ; NO TX ACT FOUND AFTER VALID DATA WORD TX D .
6482 033454 000411      BR      68      ;GO REPORT THE ERROR.
6483      ;
6484      ; VERIFY TX ACTION RETURNED FROM CORRECT LINE.
6485      ;
6486 033456 005267 150306      48:    INC    ERRNBR      ;INCREMENT ERROR NUMBER TO 2103.
6487 033462 000302      SWAB    R2      ;GET THE LINE NUMBER IN THE LOW BYTE.
6488 033464 042702 177760      BIC    #177760,R2    ;CLEAR THE UNWANTED BITS.
6489 033470 020204      CMP     R2,R4      ;IS IT THE CORRECT LINE NUMBER?.
6490 033472 001404      BEQ     88      ;YES; SKIP THE ERROR REPORT.
6491 033474 012702 013314      MOV    #0EM2203,R2    ;PASS THE ERROR MESSAGE TO BE REPORTED.
6492      ; INCORRECT LINE # RETURNED WITH TX ACT
6493 033500 010401      68:    MOV    R4,R1      ;PASS THE NUMBER OF CURRENT LINE UNDER TEST.
6494 033502      ERROR      ;
6495      ; >>>> ERROR <<<<.
6496      ; TRAP    C$ERROR
6497      ;
6498      ; VERIFY ALL ACTIVE LINES HAVE BEEN TESTED.
6499      ;
6500 033504 005204      88:    INC    R4      ;INCREMENT THE LINE NUMBER COUNTER.
6501 033506 005705      TST     R5      ;ARE THERE ANY MORE ACTIVE LINES TO TEST?.
6502 033510 001335      BNE     28      ;YES; BRANCH TO TEST THE NEXT LINE.
6503 033512 005067 146634      60:    CLR    CTRLCF    ;INDICATE THAT WE ARE NOT WITHIN A TEST.
6504 033516      ENDTST
6505 033516 104401      L10052:
6506      ; TRAP    C$ETST

```

HARDWARE TEST

TXENBI

```

0508 033520 012700 000240
0509 033520 012700 000026
0510 033520 012767 000026 146572
0511 033534 012767 177777 146610
0512 033547 012767 000001 150216
0513 033550 012767 004375 150212
0514 033556 012767 013421 150206
0515 033564 012767 017420 150202
0516 033572 004767 164356
0517 033576 103110
0518 033600 016705 146432
0519 033604 012700 000200
0520 033610 004767 167714
0521 033614 012700 177670
0522 033620 004767 167760
0523 033624 012704 000012
0524 033630 004767 164442
0525 033634 012705 000377
0526 033640 004767 167024
0527 033644 012703 000001
0528 033650 005004
0529 033652 012767 004376 150110 24:
0530 033660 030367 146352
0531 033664 001447
0532
0533
0534
0535
0536
0537
0538
0539
0540
0541
0542
0543
0544
0545
0546
0547
0548
0549
0550
0551
0552
0553
0554
0555
0556
0557
0558
0559

```

```

;SBTTL HARDWARE TEST TXENBI
;*****
; TX ENABLE (INACTIVE) TEST
; THIS TEST VERIFIES THAT WHEN THE LINE UNDER TEST'S TX_ENABLE BIT IS
; CLEAR, TRANSMISSION WILL NOT TAKE PLACE ON THAT LINE.
; THIS TEST IS PERFORMED IN INTERNAL LOOPBACK, AND ON ALL ACTIVE LINES.
;*****
BGNTST
SETPRI: #PRIOS ;ALLOW LTC INTERRUPTS. T22::
MOV #PRIOS,PC
TRAP C$PRI
TNUM == TNUM + 1 ;INCREMENT THE ASSEMBLY TIME TEST COUNTER.
MOV #TNUM,TSTNUM ;SET UP THE TEST NUMBER. (23)
MOV #-1,CTRLCF ;INDICATE THAT WE ARE IN A TEST.
MOV #1,ERRTYP ;SET ERROR TYPE AS FATAL IN ERROR TABLE.
MOV #2301,ERRNBR ;SET THE FIRST ERROR NUMBER IN ERROR TABLE.
MOV #EM2301,ERRMSG ;SET ERROR MESSAGE ADDRESS IN ERRTABL.
MOV #ER9101,ERRBLK ;SELECT THE CORRECT ERROR REPORTING ROUTINE.
;
; RESET THE DUT TO A KNOWN STATE, REMOVE THE STATUS CODES FROM THE FIFO.
; CLEAR TX AND RX INTERRUPT ENABLE BITS IN THE CSR.
; THIS SUBROUTINE REPORTS ERROR >>>> 2301 <<<<.
;
JSR PC,CLRST ;RESET THE DMV-11, REPORT ANY ERRORS FOUND.
BCC 60$ ;RESET FAILURE?, ABORT THIS TEST.
;
; SET INTERNAL LOOPBACK ON ALL ACTIVE LINES.
; SET LPR ON ALL LINES TO 38.4K BAUD, 8 BITS PER CHARACTER, ODD PARITY,
; 2 STOP BITS.
; ENABLE TRANSMITTERS ON ALL LINES.
;
MOV ACTLNS,R5 ;PASS THE ACTIVE LINE BIT MAP.
MOV #200,R0 ;PASS THE LNCTRL CONTENTS.
JSR PC,WTWLNLC ;INITIALISE THE LNCTRL REGISTERS.
MOV #177670,R0 ;PASS THE LPR CONTENTS.
JSR PC,WTWLPR ;INITIALISE THE LPR REGISTERS ON ALL LINES.
MOV #10,R4 ;PASS DELAY TIME OF 10 MILLI-SECONDS.
JSR PC,DELAY ;WAIT FOR LNCTR AND LPR REGS TO BE UPDATED.
MOV #MAPLNS,R5 ;PASS THE BIT MAP CORRESPONDING TO ALL LINES.
JSR PC,TXENBL ;ENABLE TRANSMITTERS ON ALL LINES.
;
; TEST ALL ACTIVE LINES INDIVIDUALLY.
; DISABLE TRANSMISSION ON EACH ACTIVE LINE.
;
MOV #1,R3 ;SET UP THE LINE BIT MAP FOR CHANNEL 0.
CLR R4 ;CLEAR THE LINE NUMBER COUNTER.
MOV #2302,ERRNBR ;SET THE ERROR NUMBER TO 2302.
BIT R3,ACTLNS ;CHECK IF THE LINE IS ACTIVE.
BEQ 6$ ;SKIP TESTING THIS LINE IF IT IS INACTIVE.
;
; CLEAR THE TX_ENABLE BIT IN TBUFFAD2 REGISTER.
; SELECT THE LINE UNDER TEST.
; VERIFY IT IS CLEAR, REPORT ERROR IF SET.
;

```

HARDWARE TEST

- TXENBI

```

6560 033666 010305      MOV      R3,R5      ;PASS THE BIT MAP OF THE LINE UNDER TEST.
6561 033670 004767 166700 JSR      PC,TXDSBL    ;DISABLE TRANSMISSION ON THE LINE UNDER TEST.
6562 033674 010477 146344 MOV      R4,@CSRA      ;SELECT THE LINE CURRENTLY UNDER TEST.
6563 033700 005777 146354 TST      @TXAD2A        ;VERIFY THE TX_ENABLE BIT IS SET.
6564 033704 100433      BMI      4$      ;GO REPORT ERROR IF TX_ENABLE BIT SET.
6565
6566      ;*
6567      ; WRITE DATA WORD (ASCII <LF>) TO TXCHR REGISTER.
6568      ; WAIT FOR A TX_ACTION TO BE RETURNED, REPORT ERROR IF A TX ACTION
6569      ; IS FOUND BEFORE TIME-OUT OCCURS.
6570 033706 012767 004377 150054 MOV      @2303,ERRNBR    ;SET ERROR NUMBER TO 2303.
6571 033714 012777 100012 146324 MOV      @100012,@TXCHA ;WRITE THE DATA WORD TO THE DUT'S TXCHAR REG.
6572 033722 012701 170002      MOV      @170002,R1    ;TEST BIT 15, TIMEOUT OF 2 MILLI SECS.
6573 033726 016702 146312      MOV      CSRA,R2      ;PASS THE ADDRESS OF THE REGISTER TO TEST.
6574 033732 004767 167322 JSR      PC,WAIBIS      ;WAIT FOR TX_ACTION TO COME BACK.
6575 033736 103016      BCC      4$      ;GO REPORT ERROR IF NO TX-ACTION FOUND.
6576
6577      ;*
6578      ; WAIT FOR THE DATA TO APPEAR IN THE FIFO, REPORT ERROR IF DATA FOUND.
6579 033740 005267 150024      INC      ERRNBR      ;SET ERROR NUMBER TO 2304.
6580 033744 012701 070012      MOV      @70012,R1    ;TEST BIT 7, TIMEOUT OF 10 MILLI SECS.
6581 033750 016702 146270      MOV      CSRA,R2      ;PASS THE ADDRESS OF THE REGISTER TO TEST.
6582 033754 004767 167300 JSR      PC,WAIBIS      ;WAIT FOR RX_DATA_AVAILABLE TO SET.
6583 033760 103405      BCS      4$      ;REPORT ERROR IF DATA RECEIVED IN THE FIFO.
6584 033762 005267 150002      INC      ERRNBR      ;SET ERROR NUMBER TO 2305.
6585 033766 017702 146254      MOV      @RBUFA,R2    ;READ THE DATA FROM THE FIFO.
6586 033772 100004      BPL      6$      ;SKIP ERROR REPORT IF DATA IS THERE.
6587
6588 033774 010401      4$:      MOV      R4,R1      ;PASS THE NUMBER OF CURRENT LINE UNDER TEST.
6589 033776 012702 013457      MOV      @EM2302,R2    ;PASS THE MESSAGE TO BE REPORTED.
6590
6591 034002      ERROR      ; 'TX_ENABLE BIT BAD ON LINE: NN'.
6592 034002 104460      ;          >>>>> ERROR <<<<<.
6593
6594      ;*
6595 034004 000241      ;*
6596 034006 006103      ; VERIFY ALL ACTIVE LINES HAVE BEEN TESTED.
6597 034010 005204      ;*
6598 034012 020427 000010 6$:      CLC      ;CLEAR THE CARRY BIT PRIOR TO ROTATION.
6599 034016 002715      ROL      R3      ;SHIFT THE BIT MAP FOR THE NEXT LINE.
6600
6601 034020 005067 146326      INC      R4      ;INCREMENT THE LINE NUMBER COUNTER.
6602 034024      CMP      R4,@NUMLNS    ;HAVE ALL THE LINES BEEN TESTED?.
        034024      BLT      2$      ;NO; BRANCH TO TEST THE NEXT LINE.
        034024 104401      60$:      CLR      CTRLCF    ;INDICATE THAT WE ARE NOT WITHIN A TEST.
                                L10053:
                                TRAP      C$ETST

```

HARDWARE TEST

TXENBA

```

6604 .SBTTL HARDWARE TEST TXENBA
6605 ;* *****
6606 ;* TX_ENABLE (ACTIVE) TEST
6607 ;* THIS TEST VERIFIES THAT WHEN THE TX_ENABLE BIT IS SET IN THE APPROPRIATE
6608 ;* LINE REGISTER, TRANSMISSION WILL TAKE PLACE ON THAT LINE.
6609 ;* THIS TEST IS PERFORMED IN INTERNAL LOOPBACK, AND ON ALL ACTIVE LINES.
6610 ;*
6611 ;* *****
6612
6613 034026 BGNTEST
6614 034026 SETPRI @PRI05 ;ALLOW LTC INTERRUPTS. T23::
6615 034026 012700 000240 MOV @PRI05,R0
6616 034032 104441 TRAP C$SPRI
6617 000027 TNUM == TNUM + 1 ;INCREMENT THE ASSEMBLY TIME TEST COUNTER.
6618 034034 012767 000027 146264 MOV @TNUM,T$TNUM ;SET UP THE TEST NUMBER. (24)
6619 034042 012767 177777 146302 MOV @-1,CTRLCF ;INDICATE THAT WE ARE IN A TEST.
6620 034050 012767 000001 147710 MOV @1,ERRTYP ;SET ERROR TYPE AS FATAL IN ERROR TABLE.
6621 034056 012767 004541 147704 MOV @2401,ERRNBR ;SET THE FIRST ERROR NUMBER IN ERROR TABLE.
6622 034064 012767 013515 147700 MOV @EM2401,ERRMSG ;SET ERROR MESSAGE ADDRESS IN ERTBL.
6623 034072 012767 017420 147674 MOV @ER9101,ERRBLK ;SELECT THE CORRECT ERROR REPORTING ROUTINE.
6624 ;*
6625 ;* RESET THE DUT TO A KNOWN STATE, REMOVE THE STATUS CODES FROM THE FIFO.
6626 ;* CLEAR TX AND RX INTERRUPT ENABLE BITS IN THE CSR.
6627 ;* THIS SUBROUTINE REPORTS ERROR >>>> 2401 <<<<.
6628 034100 004767 164050 JSR PC,CLRST ;RESET THE DMV 11, REPORT ANY ERRORS FOUND.
6629 034104 103127 BCC 60$ ;RESET FAILURE?, ABORT THIS TEST.
6630 ;*
6631 ;* SET INTERNAL LOOPBACK ON ALL ACTIVE LINES.
6632 ;* SET LPR ON ALL LINES TO 38.4K BAUD, 8 BITS PER CHARACTER, ODD PARITY,
6633 ;* 2 STOP BITS.
6634 ;* DISABLE TRANSMITTERS ON ALL LINES.
6635 ;*
6636 034106 016705 146124 MOV ACTLNS,R5 ;PASS THE ACTIVE LINE BIT MAP.
6637 034112 012700 000200 MOV @200,R0 ;PASS THE LNCTRL CONTENTS.
6638 034116 004767 167406 JSR PC,WTWLCN ;INITIALISE THE LNCTRL REGISTERS.
6639 034122 012700 177670 MOV @177670,R0 ;PASS THE LPR CONTENTS.
6640 034126 004767 167452 JSR PC,WTWLPR ;INITIALISE THE LPR REGISTERS ON ALL LINES.
6641 034132 012704 000012 MOV @10,R4 ;PASS DELAY TIME OF 10 MILLI-SECONDS.
6642 034136 004767 164134 JSR PC,DELAY ;WAIT FOR LNCTR AND LPR REGS TO BE UPDATED.
6643 034142 012705 000377 MOV @MAPLNS,R5 ;PASS THE BIT MAP CORRESPONDING TO ALL LINES.
6644 034146 004767 166422 JSR PC,TXDSBL ;DISABLE TRANSMITTERS ON ALL LINES.
6645 ;*
6646 ;* TEST ALL ACTIVE LINES INDIVIDUALLY.
6647 ;* ENABLE TRANSMISSION ON EACH ACTIVE LINE.
6648 ;*
6649 034152 012703 000001 MOV @1,R3 ;SET UP THE LINE BIT MAP FOR CHANNEL 0.
6650 034156 005004 CLR R4 ;CLEAR THE LINE NUMBER COUNTER.
6651 034160 012767 004542 147602 2$: MOV @2402,ERRNBR ;SET THE ERROR NUMBER TO 2402.
6652 034166 030367 146044 BIT R3,ACTLNS ;CHECK IF THE LINE IS ACTIVE.
6653 034172 001463 BEQ 8$ ;SKIP TESTING THIS LINE IF IT IS INACTIVE.
6654 ;*
6655 ;* SELECT THE LINE UNDER TEST.
6656 ;* SET THE TX_ENABLE BIT IN TBUFAD2 REGISTER.
6657 ;* VERIFY IT IS SET, REPORT ERROR IF CLEAR.

```

HARDWARE TEST

TXENBA

```

6658 034174 010305      MOV      R3,R5      ;PASS THE BIT MAP OF THE LINE UNDER TEST.
6659 034176 004767 166466 JSR      PC,TXENBL    ;ENABLE TRANSMISSION ON THE LINE UNDER TEST.
6660 034202 012705 000012 MOV      #10.,R5      ;SET TXCHAR/LOOP COUNT TO 10.
6661 034206 010477 146032 MOV      R4,@CSRA     ;SELECT THE LINE CURRENTLY UNDER TEST.
6662 034212 005777 146042 TST      @TXAD2A      ;VERIFY THE TX_ENABLE BIT IS SET.
6663 034216 100045      BPL      6$      ;GO REPORT ERROR IF TX_ENABLE BIT CLEAR.
6664      ;
6665      ; WRITE DATA WORD (ASCII <LF>) TO TXCHR REGISTER.
6666      ; WAIT FOR A TX ACTION TO BE RETURNED, REPORT ERROR IF NO TX_ACTION
6667      ; FOUND BEFORE TIME-OUT OCCURS.
6668      ;
6669 034220 012767 004543 147542 4$: MOV      #2403.,ERRNBR ;SET ERROR NUMBER TO 2403.
6670 034226 012777 100012 146012 MOV      #100012,@TXCHA ;WRITE THE DATA WORD TO THE DUT'S TXCHAR REG.
6671 034234 012701 170002 MOV      #170002,R1    ;TEST BIT 15, TIMEOUT OF 2 MILLI SECS.
6672 034240 016702 146000 MOV      CSRA,R2      ;PASS THE ADDRESS OF THE REGISTER TO TEST.
6673 034244 004767 167010 JSR      PC,WAIBIS    ;WAIT FOR TX ACTION TO COME BACK.
6674 034250 103030      BCC      6$      ;GO REPORT ERROR IF NO TX-ACTION FOUND.
6675      ;
6676      ; WAIT FOR THE DATA TO APPEAR IN THE FIFO, REPORT ERROR IF TIME OUT.
6677      ;
6678 034252 005267 147512      INC      ERRNBR    ;SET ERROR NUMBER TO 2404.
6679 034256 012701 070012 MOV      #70012,R1    ;TEST BIT 7, TIMEOUT OF 10 MILLI SECS.
6680 034262 016702 145756 MOV      CSRA,R2      ;PASS THE ADDRESS OF THE REGISTER TO TEST.
6681 034266 004767 166766 JSR      PC,WAIBIS    ;WAIT FOR RX DATA AVAILABLE TO SET.
6682 034272 103017      BCC      6$      ;REPORT ERROR IF NO DATA RECEIVED IN THE FIFO.
6683 034274 005267 147470      INC      ERRNBR    ;SET ERROR NUMBER TO 2405.
6684 034300 017702 145742 MOV      @RBUFA,R2    ;READ THE DATA FROM THE FIFO.
6685 034304 100012      BPL      6$      ;GO REPORT ERROR IF THER IS'NT ANY DATA THERE.
6686 034306 005267 147456      INC      ERRNBR    ;SET ERROR NUMBER TO 2406.
6687 034312 000302      SWAB     R2      ;PUT THE LINE NUMBER IN THE LOW BYTE.
6688 034314 042702 177760      BIC      #177760,R2    ;CLEAR THE UNWANTED BITS.
6689 034320 020204      CMP      R2,R4      ;DID THE DATA COME FROM THE CORRECT LINE?.
6690 034322 001003      BNE      6$      ;NO; GO REPORT THE ERROR.
6691 034324 005305      DEC      R5      ;DECREMENT THE TXCHAR/LOOP COUNTER.
6692 034326 001334      BNE      4$      ;LOOP TO TX THE NEXT CHAR.
6693 034330 000404      BR      8$      ;GO TEST THE NEXT LINE.
6694      ;
6695 034332 010401      6$: MOV      R4,R1      ;PASS THE NUMBER OF CURRENT LINE UNDER TEST.
6696 034334 012702 013457      MOV      #EM2302,R2    ;PASS THE MESSAGE TO BE REPORTED.
6697      ;
6698 034340      ERROR      ; "TX_ENABLE BIT BAD ON LINE: NN".
6699 034340 104460      ; >>>> ERROR <<<<<.
6700      ; TRAP      C$ERROR
6701      ;
6702      ; VERIFY ALL ACTIVE LINES HAVE BEEN TESTED.
6703      ;
6704 034342 010305 166224 8$: MOV      R3,R5      ;PASS THE BIT MAP OF THE LINE UNDER TEST.
6705 034344 004767      JSR      PC,TXDSBL    ;CLEAR THE TX_ENABLE BIT ON THIS LINE.
6706 034350 000241      CLC      ;CLEAR THE CARRY BIT PRIOR TO ROTATION.
6707 034352 006103      ROL      R3      ;SHIFT THE BIT MAP FOR THE NEXT LINE.
6708 034354 005204      INC      R4      ;INCREMENT THE LINE NUMBER COUNTER.
6709 034356 020427 000010      CMP      R4,#NUMLNS    ;HAVE ALL THE LINES BEEN TESTED?.
6710 034362 002676      BLT      2$      ;NO; BRANCH TO TEST THE NEXT LINE.
6711      ;
6712 034364 005067 145762 60$: CLR      CTRLCF    ;INDICATE THAT WE ARE NOT WITHIN A TEST.
6713 034370      ENDTST
6714 034370      ;
6715 034370 104401      ;

```

L10054:

TRAP C\$ETST

HARDWARE TEST

INTA -

```

6713 .SBTTL HARDWARE TEST INTA
6714 ;* *****
6715 ;* INTERRUPT TEST
6716 ;* THIS TEST VERIFIES THAT THE DEVICE UNDER TEST (DUT) WILL GENERATE
6717 ;* RECEPTION AND TRANSMISSION INTERRUPTS CORRECTLY. THIS TEST DOES
6718 ;* NOT DEPEND ON THE USE OF THE SERIAL LINE TRANSMISSION OR RECEPTION
6719 ;* CAPABILITIES OF THE DUT. THE LINES ARE PUT IN INTERNAL LOOPBACK
6720 ;* TO MINIMIZE ANY EXTERNAL EFFECTS THAT COULD BE CAUSED ON DEVICES
6721 ;* ATTACHED TO THE SERIAL LINES.
6722 ;*
6723 ;* *****
6724
6725 034372 BGNTEST
6726 034372 T24::
034372 SETPRI #PRI05 ;ALLOW THE LTC TO INTERRUPT.
034376 104441 MOV #PRI05,R0
034376 000030 TRAP C$SPRI
6727 000030 TNUM == TNUM + 1 ;INCREMENT THE ASSEMBLY TIME TEST COUNTER.
6728 034400 012767 000030 145720 MOV #TNUM,TSTNUM ;SET UP THE TEST NUMBER. (26)
6729 034406 012767 177777 145736 MOV #1,CTRLCF ;INDICATE THAT WE ARE IN A TEST.
6730 034414 012767 000001 147344 MOV #1,ERRTYP ;SET ERROR FATAL ERROR TYPE IN ERROR TABLE.
6731 034422 012767 003101 147340 MOV #1601.,ERRNBR ;SET FIRST ERROR REPORT NUMBER IN ERROR TABLE.
6732 034430 012767 013551 147334 MOV #EM2601.,ERRMSG ;SET TEST ERROR MESSAGE IN ERROR TABLE.
6733 ;*
6734 ;* RESET THE DUT TO A KNOWN STATE. DO NOT REMOVE THE STATUS CODES FROM THE FIFO.
6735 ;* CLEAR TX AND RX INTERRUPT ENABLE BITS IN THE CSR.
6736 ;* THIS SUBROUTINE REPORTS ERRORS FROM >>>> 2601 THRU 2602 <<<<.
6737 ;*
6738 034436 004767 165012 JSR PC,RESETT ;RESET THE DHV-11, REPORT ANY ERRORS FOUND.
6739 034442 103402 BCS 2$ ;SKIP AROUND ABORTING TEST IF NO ERROR FOUND.
6740 034444 000167 000740 JMP 60$ ;ABORT TEST IF FATAL ERROR FOUND DURING RESET.
6741 034450 012767 005053 147312 2$: MOV #2603.,ERRNBR ;SET THE ERROR REPORT NUMBER TO 2603.
6742 ;*
6743 ;* ENABLE TRANSMITTERS ON ALL LINES.
6744 ;*
6745 034456 012705 000377 4$: MOV #MAPLNS,R5 ;PASS ACTIVE LINE BIT MAP.
6746 034462 004767 166202 JSR PC,TXENBL ;ENABLE TRANSMISSION ON ALL LINES.
6747 ;*
6748 ;* TEST RECEPTION INTERRUPTS.
6749 ;* SET UP FOR RX AND TX INTERRUPTS:
6750 ;* RX INTERRUPT SERVICE ROUTINE INPUTS A CHAR AND COUNTS THE INTERRUPT.
6751 ;* TX INTERRUPT SERVICE ROUTINE COUNTS TX INTERRUPTS.
6752 ;*
6753 034466 005067 145642 CLR RXINTC ;CLEAR THE RX INTERRUPT COUNTER.
6754 034472 005067 145640 CLR RXINTF ;CLEAR THE RX INTERRUPT FLAGS.
6755 034476 005067 145636 CLR TXINTC ;CLEAR THE TX INTERRUPT COUNTER.
6756 034502 012767 002726 145614 MOV #BUFBAS,BUFPTR ;LOAD THE BUFFER PTR WITH THE BUFFER BASE ADR.
6757 034510 SETVEC RXVECA,#RXINPT,#PRI05 ;SET UP INTERRUPT VECTOR TO CATCH RX INT.
034510 012746 000240 MOV #PRI05,(SP)
034514 012746 024050 MOV #RXINPT,(SP)
034520 016746 145506 MOV RXVECA,(SP)
034524 012746 000003 MOV #3,-(SP)
034530 104437 TRAP C$SVEC
034532 062706 000010 ADD #10,SP
6758 034536 SETVEC TXVECA,#CACHTX,#PRI05 ;SET UP INTERRUPT VECTOR TO CATCH TX INT.
034536 012746 000240 MOV #PRI05,(SP)
034542 012746 023662 MOV #CACHTX,-(SP)

```

HARDWARE TEST - INTA -

```

034546 016746 145462      MOV    T/VECA, -(SP)
034552 012746 000003      MOV    #3, (SP)
034556 104437              TRAP    C$SVEC
034560 062706 000010      ADD     #10, SP
6759 034564              SETPRI  #PRI03      ;ALLOW DEVICE INTERRUPTS.
034564 012700 000140      M,      #PRI03, R0
034570 104441              TRAP    C$SPRI

6760      ;*
6761      ;ENABLE RECEPTION INTERRUPTS.
6762      ;DELAY 4 MS TO ALLOW TIME FOR THE INTERRUPTS TO TAKE PLACE.
6763      ;DISABLE RECEPTION INTERRUPTS.
6764      ;
6765 034572 004767 165444      JSR     PC, RXIE1      ;ENABLE THE RECEPTION INTERRUPTS.
6766 034576 012704 000004      MOV     #4, R4      ;PASS 4 MS COUNT TO THE DELAY ROUTINE.
6767 034602 004767 163470      JSR     PC, DELAY      ;DELAY 4 MILLI SECONDS.
6768 034606 004767 165370      JSR     PC, RXIE0      ;DISABLE RECEPTION INTERRUPTS.
6769      ;*
6770      ; VERIFY THAT THE CORRECT INTERRUPTS TOOK PLACE.
6771      ; TEST THE INT COUNTER TO VERIFY THAT INTERRUPTS TOOK PLACE.
6772      ;
6773 034612 005767 145516      TST     RXINTC      ;CHECK THE RX INTERRUPT COUNT.
6774 034616 001017              BNE     #0      ;SKIP THE FOLLOWING ERRORS IF COUNT <> 0.
6775      ;*
6776      ; DETERMINE REASON FOR NO RX INTERRUPTS AND PRINT PROPER ERROR MESSAGE.
6777      ;
6778 034620 012701 013760      MOV     #EM2604, R1      ;SET UP MSG IN CASE 'RX.DATA.AVAIL IS CLR'.
6779 034624 032777 000200 145412      BIT     #BIT7, @CSRA      ;TEST THE RX.DATA.AVAIL BIT OF THE CSR.
6780 034632 001416              BEQ     #0      ;GO REPORT ERROR IF RX.DATA.AVAIL IS CLR.
6781 034634 012701 013672      MOV     #EM2603, R1      ;SET UP MSG IN CASE "DATA.VALID IS CLEAR".
6782 034640 032777 100000 145400      BIT     #BIT15, @RBUFA      ;TEST THE DATA.VALID BIT OF THE FIFO.
6783 034646 001410              BEQ     #0      ;GO REPORT ERROR IF DATA.VALID IS CLEAR.
6784 034650 012701 013601      MOV     #EM2602, R1      ;SET UP MSG, "DATA.VALID IS SET".
6785 034654 000405              BR      #0      ;GO REPORT THE ERROR.
6786      ;*
6787      ; IF RX INTS OCCURRED WITH RX.DATA.AVAIL CLEAR, REPORT THE ERROR.
6788      ;
6789 034656 005767 145454      6$:    TST     RXINTF      ;CHECK THE RX INTERRUPT FLAGS.
6790 034662 100006              BPL     #0      ;SKIP THE ERROR IF FLAG IS CLEAR.
6791 034664 012701 014054      MOV     #EM2605, R1      ;SET UP THE PROPER MESSAGE.
6792      ;*
6793      ; REPORT THE ERROR WHICH HAS BEEN FOUND.
6794      ;
6795 034670      8$:    ERRDF 2603, EM2601, ER0503;      >>>> ERROR #2603 <<<<.
034670 104455              TRAP    C$ERDF
034672 005053              .WORD   2603
034674 013551              .WORD   EM2601
034676 016466              .WORD   ER0503

6796      ;*
6797      ; VERIFY THAT NO TX INTERRUPTS HAVE BEEN GENERATED SO FAR IN THIS TEST.
6798      ;
6799 034700 016702 145434      10$:   MOV     TXINTC, R2      ;LOAD # OF TX INTERRUPTS FOR ER0504 RTN.
6800 034704 001406              BEQ     #0      ;SKIP ERROR IF NO TX INTERRUPTS.
6801      ;REPORT "TX INTERRUPTS(S) RECEIVED WITH TX INTERRUPTS DISABLED."
6802 034706 012701 010001      MOV     #EM0526, R1      ;SET UP MESSAGE ADR FOR INDIRECT PRINT.
6803 034712      ERRDF 2604, EM2601, ER0504;      >>>> ERROR #2604 <<<<.
034712 104455              TRAP    C$ERDF
034714 005054              .WORD   2604

```

HARDWARE TEST

. INTA .

```

034716 013551
034720 016512
6804
6805
6806
6807 034722
034722 012700 000240
034726 104441
6808 034730
034730 016700 145276
034734 104436
6809 034736
034736 016700 145272
034742 104436
6810
6811
6812
6813
6814
6815
6816 034744 005067 145364
6817 034750 005067 145364
6818 034754 005067 145362
6819 034760
034760 012746 000240
034764 012746 023634
034770 016746 145236
034774 012746 000003
035000 104437
035002 062706 000010
6820 035006
035006 012746 000240
035012 012746 024156
035016 016746 145212
035022 012746 000003
035026 104437
035030 062706 000010
6821 035034
035034 012700 000140
035040 104441
6822
6823
6824
6825 035042 012705 000022
6826 035046 012701 000144
6827 035052 012702 100000
6828 035056 016704 145162
6829 035062 012703 100000
6830 035066 004767 163360
6831 035072 103020
6832 035074 005003
6833 035076 004767 163350
6834 035102 103005
6835 035104 005305
6836 035106 001365
6837
6838 035110 012701 014176

```

```

; CLEAN OUT THE INTERRUPT VECTORS USED IN THIS TEST.
12$: SETPRI @PRI05 ;DISABLE DEVICE INTERRUPTS.
CLRVEC RXVECA ;RETURN RX INT VECTOR TO UNUSED POOL.
CLRVEC TXVECA ;RETURN TX INT VECTOR TO UNUSED POOL.
; TEST TRANSMISSION INTERRUPTS.
; SET UP FOR RX AND TX INTERRUPTS:
; RX INTERRUPT SERVICE ROUTINE COUNTS RX INTERRUPTS.
; TX INTERRUPT SERVICE ROUTINE COUNTS THE INTERRUPT AND SFTS FLAGS.
CLR RXINTC ;CLEAR THE RX INTERRUPT COUNTER.
CLR TXINTC ;CLEAR THE TX INTERRUPT COUNTER.
CLR TXINTF ;CLEAR THE RX INTERRUPT FLAGS.
SETVEC RXVECA,@CACHRX,@PRI05 ;SET UP INTERRUPT VECTOR TO CATCH RX INT.
MOV @PRI05,-(SP)
MOV @CACHRX,-(SP)
MOV RXVECA,-(SP)
MOV @3,-(SP)
TRAP C$SVEC
ADD @10,SP
SETVEC TXVECA,@TXINTR,@PRI05 ;SET UP INT VECTOR TO TX INT ROUTINE.
MOV @PRI05,-(SP)
MOV @TXINTR,-(SP)
MOV TXVECA,(SP)
MOV @3,-(SP)
TRAP C$SVEC
ADD @10,SP
SETPRI @PRI03 ;ALLOW DEVICE INTERRUPTS.
MOV @PRI03,R0
TRAP C$SPRI
; VERIFY THAT THE TX_ACTION BIT IS CLEAR.
;
MOV @18,R5 ;INITIALIZE THE LOOP COUNTER.
MOV @100,R1 ;SET 100 MS TIME-OUT.
MOV @BIT15,R2 ;SELECT TX_ACTION BIT TO TEST.
MOV CSRA,R4 ;PASS OUT CSR AS THE WORD TO TEST.
14$: MOV @BIT15,R3 ;WAIT FOR TX_ACTION TO BE SET.
JSR PC,MSLOOP ;WAIT UP TO 100 MS FOR TX_ACTION SET.
BCC 20$ ;IF TIME-OUT, CONSIDER TX_ACTION CLEAR.
CLR R3 ;NOW, WAIT FOR TX_ACTION CLEAR.
JSR PC,MSLOOP ;WAIT UP TO 100 MS FOR TX_ACTION CLEAR.
BCC 16$ ;IF TIME-OUT, REPORT TX_ACTION WCN'T CLEAR.
DEC R5 ;DECREMENT THE TX_ACTION SET COUNTER.
BNE 14$ ;LOOP IF NOT TOO MANY TX_ACTIONS FOUND.
;REPORT "TX_ACTION SET REPEATEDLY AFTER RESET, NO DATA SENT."
MOV @EM2607,R1 ;SELECT ERROR MESSAGE.

```

.WORD EM2601
.WORD ER0504

HARDWARE TEST

INTA .

```

6839 035114 000402
6840 035116 012701 014272
6841 035122
      035122 104455
      035124 005055
      035126 014137
      035130 016466
6842 035132 000424
6843
6844
6845
6846 035134 004767 165664
6847 035140 012704 000062
6848 035144 004767 163126
6849 035150 005767 145164
6850 035154 001413
6851 035156 012701 014176
6852 035162 005767 145154
6853 035166 100002
6854 035170 012701 014343
6855
6856 035174
      035174 104455
      035176 005056
      035200 014137
      035202 016466
6857
6858
6859
6860 035204 005067 145130
6861 035210 005067 145126
6862
6863
6864
6865 035214 012705 000377
6866 035220 012700 000200
6867 035224 004767 166300
6868 035230 012700 156430
6869 035234 004767 166344
6870
6871
6872
6873 035240 016701 145002
6874 035244 012702 100000
6875 035250 004767 166304
6876
6877
6878
6879 035254 012704 000372
6880 035260 004767 163012
6881
6882
6883
6884 035264 005767 145050
6885 035270 001007
6886
6887

      BR      18:      ;GO TO REPORT THE ERROR.
      MOV     18:      ;SELECT TX ACTION STICK SET MSG.
      ERDF    2605,EM2606,ER0503;      >>>> ERROR #2605 <<<<.
                                      TRAP    C$ERDF
                                      .WORD    2605
                                      .WORD    EM2606
                                      .WORD    ER0503

      BR      24:      ;GO TO TEST WITH TX_ACTION SET.
;
; VERIFY THAT NO INTERRUPTS OCCUR WITH TX_ACTION CLEAR.
20:
      JSR     PC, TXIE1      ;ENABLE TX INTERRUPTS.
      MOV     #50.,R4      ;PASS 50 MS TIME TO THE DELAY ROUTINE.
      JSR     PC, DELAY      ;DELAY 50 MILLI-SECONDS TO ALLOW INTS TO OCCUR.
      TST     TXINTC      ;TEST THE TX INTERRUPT COUNT.
      BEQ     24:      ;SKIP THE ERROR IF NO TX INTERRUPTS.
      MOV     #EM2607,R1      ;SELECT MESSAGE IN CASE TX INT FLAG CLEAR.
      TST     TXINTF      ;TEST THE TX INTERRUPT FLAGS.
      BPL     22:      ;GO REPORT ERROR IF TX FLAG IS CLEAR.
      MOV     #EM2609,R1      ;TX FLAG IS SET, SELECT PROPER ERROR MESSAGE.
;REPORT "TRANSMIT INTERRUPT TEST ERROR:..."
22:
      ERDF    2606,EM2606,ER0503;      >>>> ERROR #2606 <<<<.
                                      TRAP    C$ERDF
                                      .WORD    2606
                                      .WORD    EM2606
                                      .WORD    ER0503

;
; PREPARE TX INTERRUPT COUNTER AND FLAGS.
24:
      CLR     TXINTC      ;CLEAR THE TX INTERRUPT COUNT.
      CLR     TXINTF      ;CLEAR THE TX INTERRUPT FLAGS.
;
; SET UP LINE PARAMETERS FOR TRANSMISSION.
;
      MOV     #MAPLNS,R5      ;PASS ACTIVE LINES BIT MAP.
      MOV     #200,R0      ;PASS INERT STATE, INTERNAL LOOPBACK.
      JSR     PC, WTWLNC      ;DISABLE RECEPTION AND DMA, ETC. ON OUT.
      MOV     #156430,R0      ;SPECIFY 9600BPS, 1STOP, NO PARITY, 8BITS/CHAR.
      JSR     PC, WTWLPR      ;WRITE TO ALL LPR REGISTERS.
;
; SEND A NULL CHAR TO EACH LINE.
;
      MOV     TXCHA,R1      ;SET UP TXCHAR REGISTER ADDRESS.
      MOV     #100000,R2      ;SET CHARACTER TO BE TRANSMITTED = NULL.
      JSR     PC, WTWLNS      ;SEND NULL CHAR TO EACH LINE.
;
; DELAY 250 MILLI-SECONDS TO ALLOW INTERRUPTS TO OCCUR.
;
      MOV     #250.,R4      ;SET UP FOR 250 MS DELAY.
      JSR     PC, DELAY      ;WAIT 250 MS.
;
; VERIFY THAT TX INTERRUPTS OCCURRED.
;
      TST     TXINTC      ;CHECK THE TX INTERRUPT COUNTER.
      BNE     26:      ;SKIP THE FOLLOWING ERROR IF WE GOT TX INTS.
;
; DETERMINE THE REASON THAT WE RECEIVED NO INTERRUPTS.

```

HARDWARE TEST

INTA -

```

6888
6889 035272 012701 014422      ; MOV    #EM2610,R1      ;SET UP MSG IN CASE TX ACTION IS SET.
6890 035276 005777 144742      ; TST    @CSRA          ;CHECK THE DUT CSR.
6891 035302 100407              ; BMI    28$           ;GO TO REPORT ERROR IF TX ACTION NOT SET.
6892 035304 012701 014514      ; MOV    #EM2611,R1      ;SET UP TX_ACTION NOT MESSAGE.
6893
6894      ; CHECK TO VERIFY THAT TX_ACTION WAS SET FOR EACH INTERRUPT.
6895
6896 035310 005767 145026      26$:  TST    TXINTF          ;CHECK THE TX INTERRUPT FLAGS.
6897 035314 100006              ; BPL    30$           ;SKIP ERROR IF TX ACTION CLR FLAG IS CLEAR.
6898 035316 012701 014343      ; MOV    #EM2609,R1      ;SET UP TX INT WITH TX_ACTION CLR MSG.
6899
6900      ; REPORT TRANSMIT INTERRUPT TEST ERROR:....'
6901
6902 035322              28$:  ERDF    2607,EM2606,ER0503;      >>>>> ERROR #2607 <<<<<.
        035322 104455              TRAP    C$ERDF
        035324 005057              .WORD    2607
        035326 014137              .WORD    EM2606
        035330 016466              .WORD    ER0503
6903
6904      ; VERIFY THAT NO TX INTERRUPTS HAVE BEEN GENERATED SO FAR IN THIS TEST.
6905
6906 035332 016702 144776      30$:  MOV    RXINTC,R2          ;LOAD # OF RX INTERRUPTS FOR ER0504 RTN.
6907 035336 001406              ; BEQ    32$           ;SKIP ERROR IF NO RX INTERRUPTS.
6908 035340 012701 007711      ; MOV    #EM0525,R1      ;SET UP MESSAGE ADR FOR INDIRECT PRINT.
6909      ;REPORT "RX INTERRUPTS(S) RECEIVED WITH RX INTERRUPTS DISABLED."
6910 035344              ; ERDF    2608,EM2606,ER0504;      >>>>> ERROR #2608 <<<<<.
        035344 104455              TRAP    C$ERDF
        035346 005060              .WORD    2608
        035350 014137              .WORD    EM2606
        035352 016512              .WORD    ER0504
6911
6912      ; DISABLE INTERRUPTS AND CLEAN OUT THE INTERRUPT VECTORS USED IN THIS TEST.
6913
6914 035354 005001              32$:  CLR    R1              ;CLEAR BOTH TRANSMITTER
6915 035356 004767 165402      ; JSR    PC,TXIE0        ; INTERRUPT ENABLE AND RECEIVER
6916 035362 004767 164614      ; JSR    PC,RXIE0        ; INTERRUPT ENABLE BITS IN THE DUT CSR.
6917 035366              ; SETPRI  #PRI05          ;DISABLE DEVICE INTERRUPTS.
        035366 012700 000240              MOV    #PRI05,R0
        035372 104441              TRAP    C$SPRI
6918 035374              CLRVEC  RXVECA          ;RETURN RX INT VECTOR TO UNUSED POOL.
        035374 016700 144632              MOV    RXVECA,R0
        035400 104436              TRAP    C$CVEC
6919 035402              CLRVEC  TXVECA          ;RETURN TX INT VECTOR TO UNUSED POOL.
        035402 016700 144626              MOV    TXVECA,R0
        035406 104436              TRAP    C$CVEC
6920
6921 035410 005067 144736      60$:  CLR    CTRLCF          ;INDICATE THAT WE ARE NOT WITHIN A TEST.
6922 035414              ; SETPRI  #PRI07          ;DISABLE ALL INTERRUPTS.
        035414 012700 000340              MOV    #PRI07,R0
        035420 104441              TRAP    C$SPRI
6923 035422              ENDTST
        035422              L10055:  TRAP    C$ETST
6924 035422 104401

```

HARDWARE TEST

BR.EVA

```

6924 .SBTTL HARDWARE TEST BRLEVA
6925 .....
6926
6927
6928
6929
6930
6931
6932
6933
6934
6935
6936
6937
6938 035424 BGNTST
6939 035424
6939 035424 012700 000240
6939 035424 104441
6939 035430 000031
6940
6941 035432 012767 000031 144666
6942 035440 012767 177777 144704
6943 035446 012767 000001 146312
6944 035454 012767 005671 146306
6945 035462 012767 014577 146302
6946 035470 005067 144766
6947
6948
6949
6950
6951
6952 035474 004767 163754
6953 035500 103402
6954 035502 000167 000572
6955 035506 012767 005673 146254
6956
6957
6958
6959 035514 012705 000377
6960 035520 004767 165144
6961
6962
6963
6964
6965 035524
6965 035524 012700 000340
6965 035530 104441
6966 035532
6966 035532 012746 000340
6966 035536 012746 024156
6966 035542 016746 144466
6966 035546 012746 000003
6966 035552 104437
6966 035554 062706 000010
6967
6968
6969
6970
6971

; BR LEVEL TEST B
; THIS TEST VERIFIES THAT THE DEVICE UNDER TEST (DUT) WILL GENERATE
; RECEPTION AND TRANSMISSION INTERRUPTS AT THE CORRECT BR LEVEL.
; THIS TEST DOES NOT DEPEND ON THE USE OF THE SERIAL LINE TRANSMISSION
; OR RECEPTION CAPABILITIES OF THE DUT. THE LINES ARE PUT IN INTERNAL
; LOOPBACK TO MINIMIZE ANY EXTERNAL EFFECTS THAT COULD BE CAUSED ON
; DEVICES ATTACHED TO THE SERIAL LINES.
; .....
;
; T25::
; SETPRI @PRI05 ;ALLOW LTC INTERRUPTS.
;
; MOV @PRI05,R0
; TRAP C$SPRI
;
; INUM ** INUM + 1 ;INCREMENT THE ASSEMBLY TIME TEST COUNTER.
; MOV @INUM,ISTNUM ;SET UP THE TEST NUMBER. (30)
; MOV @1,CTRLCF ;INDICATE THAT WE ARE IN A TEST.
; MOV @1,ERRTYP ;SET ERROR TYPE AS FATAL IN ERROR TABLE.
; MOV @3001,ERRNBR ;SET THE FIRST ERROR NUMBER IN ERROR TABLE.
; MOV @EM3001,ERRMSG ;SET ERROR MESSAGE ADDRESS IN ERRTABL.
; CLR ERSMHF ;INITIALIZE THE "REPORT ERROR SUMMARY" FLAGS.
;
; RESET THE DUT TO A KNOWN STATE, DO NOT REMOVE THE STATUS CODES FROM THE FIFO.
; CLEAR TX AND RX INTERRUPT ENABLE BITS IN THE CSR.
; THIS SUBROUTINE REPORTS ERRORS FROM >>>> 3001 THRU 3002 <<<<.
;
; JSR PC,RESETT ;RESET THE DMV 11, REPORT ANY ERRORS FOUND.
; BCS 2$ ;SKIP AROUND ABORTING TEST IF NO ERROR FOUND.
; JMP 60$ ;ABORT TEST IF FATAL ERROR FOUND DURING RESET.
; MOV @3003,ERRNBR ;SET THE ERROR REPORT NUMBER TO 3003.
;
; ENABLE TRANSMITTERS ON ALL LINES.
;
; 4$: MOV @MAPLNS,R5 ;PASS ACTIVE LINE BIT MAP.
; JSR PC,TXENBL ;ENABLE TRANSMISSION ON ALL LINES.
;
; GENERATE A TRANSMISSION INTERRUPT REQUEST.
; PROCESSOR PRIORITY SHOULD BE AT 7 DISABLING INTS.
;
; SETPRI @PRI07 ;DISABLE ALL INTERRUPTS.
;
; MOV @PRI07,R0
; TRAP C$SPRI
; TX INT.
; MOV @PRI07,(SP)
; MOV @TXINTR,(SP)
; MOV TXVECA,(SP)
; MOV @3,(SP)
; TRAP C$SVEC
; ADD @10,SP
;
; SET UP DUT FOR TRANSMISSION INTERRUPTS:
; SET UP INTERNAL LOOPBACK.
; SET UP LINE PARAMETERS FOR TRANSMISSION.
;

```

HARDWARE TEST

BRLEVA -

```

6972 035560 012705 000377      MOV    #MAPLNS,R5      ;PASS ACTIVE LINES BIT MASK.
6973 035564 012700 000200      MOV    #200,R0        ;PASS INERT STATE, INTERNAL LOOPBACK.
6974 035570 004767 165734      JSR     PC,WTWLNLC    ;DISABLE RECEPTION AND DMA, ETC. ON DUT.
6975 035574 012700 156430      MOV    #156430,R0      ;SPECIFY 9600BPS,1STOP,NO PARITY,8BITS/CHAR.
6976 035600 004767 166000      JSR     PC,WTWLPR     ;WRITE INTO ALL LPR REGISTERS.
6977
6978      ; SEND A NULL CHAR TO EACH LINE.
6979
6980 035604 016701 144436      MOV    TXCHAR,R1      ;SET UP TXCHAR REGISTER ADDRESS.
6981 035610 012702 100000      MOV    #100000,R2     ;SET CHARACTER TO BE TRANSMITTED = NULL.
6982 035614 004767 165740      JSR     PC,WTWLNLS    ;SEND NULL CHAR TO EACH LINE.
6983
6984      ; DELAY 50 MS TO ALLOW TIME FOR THE INTERRUPT TO BE GENERATED.
6985
6986 035620 012704 000062      MOV    #50.,R4        ;PASS 50 MS TIME TO THE DELAY ROUTINE.
6987 035624 004767 162446      JSR     PC,DELAY      ;DELAY 50 MILLI-SECONDS.
6988
6989      ; GENERATE A RECEPTION INTERRUPT REQUEST.
6990
6991 035630      SETVEC  RXVECA,#RXBRRT,#PRI07 ;SET UP INTERRUPT VECTOR TO CATCH RX INT.
        035630 012746 000340      MOV    #PRI07,(SP)
        035634 012746 023760      MOV    #RXBRRT,(SP)
        035640 016746 144366      MOV    RXVECA,(SP)
        035644 012746 000003      MOV    #3,(SP)
        035650 104437      TRAP    C$SVEC
        035652 062706 000010      ADD     #10,SP
6992
6993      ; SET UP FOR THE LOOP WHICH TESTS THE INTERRUPT BR LEVELS.
6994
6995 035656 012705 000340      MOV    #340,R5        ;SET UP THE PRIORITY LEVEL TO 7.
6996 035662 005003      CLR     R3            ;CLEAR THE RX PRIORITY STORE AND FLAGS.
6997 035664 005002      CLR     R2            ;CLEAR THE TX PRIORITY STORE AND FLAGS.
6998
6999      ; ENABLE TX AND RX INTERRUPTS.
7000      ; PROCESSOR PRIORITY SHOULD BE AT 7 DISABLING THE INTERRUPTS.
7001
7002 035666 004767 164350      JSR     PC,RXIE1      ;ENABLE RECEIVER INTERRUPTS.
7003 035672 004767 165126      JSR     PC,TXIE1      ;ENABLE TRANSMITTER INTERRUPTS.
7004
7005      ; LOOP, LOWERING THE PROCESSOR PRIORITY UNTIL THE DUT INTERRUPTS ON RX AND TX.
7006
7007 035676 005067 144436      6$:    CLR     TXINTC      ;CLEAR THE TX INTERRUPT COUNTER.
7008 035702 005067 144434      CLR     TXINTF      ;CLEAR THE TX INTERRUPT FLAGS.
7009 035706 005067 144422      CLR     RXINTC      ;CLEAR THE RX INTERRUPT COUNTER.
7010 035712 005067 144420      CLR     RXINTF      ;CLEAR THE RX INTERRUPT FLAGS.
7011 035716      SETPRI  R5        ;SET PROCESSOR PRIORITY TO THE SELECTED VALUE.
        035716 010500      MOV    R5,R0
        035720 104441      TRAP    C$SPRI
7012 035722 012704 000001      MOV    #1,R4        ;PASS 1 MS COUNT TO THE DELAY ROUTINE.
7013 035726 004767 162344      JSR     PC,DELAY      ;DELAY 1 MS TO ALLOW INTERRUPTS TO OCCUR.
7014
7015      ; DETERMINE IF ANY RX DUT INTERRUPTS OCCURRED.
7016      ; LOG THE PROCESSOR PRIORITY FOR THE RX INTERRUPT IF FIRST RX INT.
7017
7018 035732 005767 144376      TST     RXINTC      ;CHECK THE RECEIVE INTERRUPT COUNTER.
7019 035736 001412      BEQ     B$        ;SKIP THE PRIORITY LOG IF NO RX INT OCCURRED.
7020

```

HARDWARE TEST

BR. EVA

```

7021      ; IF THIS IS THE FIRST RX INTERRUPT, LOG THE PRIORITY.
7022      ;
7023 035740 005703      TST      R3      ;CHECK THE RX PRIORITY STORE AND FLAGS.
7024 035742 001010      BNE      B1      ;GOTO TEST FOR TX INTS IF NOT THE FIRST RX INT.
7025 035744 010503      MOV      R5,R3      ;LOG THE PRESENT PRIORITY IN THE RX PRIO STORE.
7026 035746 052703 100000      BIS      @BIT15,R3      ;SET THE RX INT HAS OCCURRED FLAG.
7027 035752 016700 144360      MOV      RXINTF,R0      ;GET THE RX INTERRUPT ROUTINE FLAGS.
7028 035756 042700 137777      RLC      @137777,R0      ;CLEAR ALL BUT THE TX INT ERROR FLAG.
7029 035762 050003      BIS      R0,R3      ;IF TX INT ERROR, SET BIT 14 OF THE PRIO FLAGS.
7030      ;
7031      ; DETERMINE IF ANY TX DUT INTERRUPTS HAVE OCCURRED.
7032      ; LOG THE PRESENT PROCESSOR PRIORITY IF THIS IS THE FIRST TX INTERRUPT.
7033      ;
7034 035764 005767 144350      B1:      TST      TXINTC      ;CHECK THE TRANSMIT INTERRUPT COUNTER.
7035 035770 001405      BEQ      10$      ;SKIP THE PRIORITY LOG IF NO TX INT OCCURRED.
7036      ;
7037      ; IF THIS IS THE FIRST TX INTERRUPT, LOG THE PRIORITY.
7038      ;
7039 035772 005702      TST      R2      ;CHECK THE TX PRIORITY STORE AND FLAGS.
7040 035774 100403      BMI      10$      ;SKIP THE LOGGING IF NOT FIRST TX INTERRUPT.
7041 035776 010502      MOV      R5,R2      ;LOG THE PRESENT PRIORITY IN THE TX PRIO STORE.
7042 036000 052702 100000      BIS      @BIT15,R2      ;SET THE TX INT HAS OCCURRED FLAG.
7043      ;
7044      ; SELECT NEXT PROCESSOR PRIORITY.
7045      ; TEST FOR BOTH RX AND TX INTERRUPTS HAVING OCCURRED, LOOP IF NOT.
7046      ;
7047 036004 162705 000040      10$:      SUB      @40,R5      ;DECREMENT PRIORITY LEVEL BY ONE.
7048 036010 002402      BLT      12$      ;GOTO CHECK FOR ERRORS IF BELOW PRIORITY ZERO.
7049 036012 030203      BIT      R2,R3      ;AND PRIO FLAGS TOGETHER, ALTER NONE OF THEM.
7050 036014 100330      BPL      6$      ;LOOP IF RX AND TX INTS HAVEN'T BOTH OCCURRED.
7051      ;
7052      ; DISABLE INTERRUPTS AND CLEAR INTERRUPT VECTORS.
7053      ;
7054 036016      12$:      SETPRI      @PRI07      ;DISABLE ALL INTERRUPTS.
7055 036022 104441      MOV      @PRI07,R0      ;RETURN RX INT VECTOR TO UNUSED POOL.
7056 036024      CLRVEC      RXVECA      ;RETURN TX INT VECTOR TO UNUSED POOL.
7057 036032      CLRVEC      TXVECA      ;RETURN TX INT VECTOR TO UNUSED POOL.
7058 036036 104436      MOV      C$SPRI      ;
7059      ;
7060      ; VERIFY THAT RX AND TX INTERRUPTS OCCURRED.
7061      ; AT THE PROPER BR LEVEL, AND
7062      ; IN THE PROPER ORDER.
7063 036040 005702      TST      R2      ;DETERMINE WHETHER TX INT OCCURRED OR NOT.
7064 036042 100414      BMI      16$      ;SKIP THESE ERRORS IF TX INT OCCURRED.
7065      ;
7066      ; DETERMINE REASON THAT NO TX INT OCCURRED.
7067      ;
7068 036044 012701 014422      MOV      @EM2610,R1      ;SELECT "NO TX INT FROM TX.ACTION" MESSAGE.
7069 036050 005777 144170      TST      @CSRA      ;CHECK THE TX.ACTION BIT OF THE DUT CSR.
7070 036054 100402      BMI      14$      ;SKIP TX.ACTION CLR MSG SELECTION IF IT IS SET.
7071 036056 012701 014514      MOV      @EM2611,R1      ;SELECT "TX.ACTION CLEAR AFTER CHARS SENT" MSG.

```

HARDWARE TEST

BRLEVA -

```

7072 ;REPORT 'INTERRUPT BR LEVEL TEST ERROR:'
7073 144: ERRDF 3003,EM3001,ER0503; >>>> ERROR #3003 <<<<.
      036062 104455 TRAP C$ERRDF
      036062 005673 .WORD 3003
      036064 014577 .WORD EM3001
      036066 016466 .WORD ER0503
7074 036072 000423 BR 184 ;SKIP THE BR LEVEL CHECK, NO TX INT OCCURRED.
7075 ;
7076 ; VERIFY THAT THE TX INTERRUPT WAS AT THE PROPER BR LEVEL.
7077 ;
7078 036074 010204 164: MOV R2,R4 ;CALCULATE THE BR LEVEL
7079 036076 042704 177400 BIC #177400,R4 ; THAT THE TRANSMIT
7080 036102 006204 ASR R4 ; INTERRUPT WAS
7081 036104 006204 ASR R4 ; REQUESTED AT, WHICH
7082 036106 006204 ASR R4 ; IS ONE GREATER THAN
7083 036110 006204 ASR R4 ; THE PROCESSOR PRIORITY
7084 036112 006204 ASR R4 ; LEVEL AT WHICH THE
7085 036114 005204 INC R4 ; TRANSMIT INTERRUPT OCCURRED.
7086 036116 116705 144120 MOVB BRLEVL,R5 ;GET THE EXPECTED INTERRUPT BR LEVEL.
7087 036122 120405 CMPB R4,R5 ;COMPARE THE INTERRUPT BR LEVEL WITH EXPECTED.
7088 036124 001406 BEQ 184 ;SKIP THE ERROR IF BR LEVEL IS CORRECT.
7089 ;REPORT 'TX INTERRUPT GENERATED AT WRONG BR LEVEL: ...'
7090 036126 012701 014724 MOV #EM3003,R1 ;SELECT THE ERROR MESSAGE FOR THE ERROR CALL.
7091 036132 036132 104455 ERRDF 3004,EM3001,ER3001; >>>> ERROR #3004 <<<<.
      036134 005674 TRAP C$ERRDF
      036136 014577 .WORD 3004
      036140 017066 .WORD EM3001
      .WORD ER3001
7092 ;
7093 ; DETERMINE IF RX INTERRUPT OCCURRED.
7094 ;
7095 036142 005703 184: TST R3 ;CHECK THE RX INT OCCURRED FLAG.
7096 036144 100415 BMI 224 ;SKIP THESE ERRORS IF RX INT OCCURRED.
7097 ;
7098 ; DETERMINE REASON THAT NO RX INT OCCURRED.
7099 ;
7100 036146 012701 013601 MOV #EM2602,R1 ;SELECT "NO RX INT FROM TX ACTION" MSG.
7101 036152 032777 700200 144064 BIT #BIT7,DCSRA ;CHECK THE RX.DATA.AVAIL BIT OF THE DUT CSR.
7102 036160 001002 BNE 204 ;SKIP RX.DATA.AVAIL CLR MSG IF BIT IS SET.
7103 036162 012701 714630 MOV #EM3002,R1 ;SELECT "NO RX.DATA.AVAIL AFTER RESET" MSG.
7104 ;REPORT 'INTERRUPT BR LEVEL TEST ERROR:'
7105 204: ERRDF 3005,EM3001,ER0503; >>>> ERROR #3005 <<<<.
      036166 104455 TRAP C$ERRDF
      036170 005675 .WORD 3005
      036172 014577 .WORD EM3001
      036174 016466 .WORD ER0503
7106 036176 000423 BR 244 ;SKIP THE BR CHECK IF NO RX INT OCCURRED.
7107 ;
7108 ; VERIFY THAT THE RX INTERRUPT WAS AT THE PROPER BR LEVEL.
7109 ;
7110 036200 010304 224: MOV R3,R4 ;CALCULATE THE BR LEVEL
7111 036202 042704 177400 BIC #177400,R4 ; THAT THE RECEIVE
7112 036206 006204 ASR R4 ; INTERRUPT WAS
7113 036210 006204 ASR R4 ; REQUESTED AT, WHICH
7114 036212 006204 ASR R4 ; IS ONE GREATER THAN
7115 036214 006204 ASR R4 ; THE PROCESSOR PRIORITY
7116 036216 006204 ASR R4 ; LEVEL AT WHICH THE

```

HARDWARE TEST

- BRLEVA -

```

7117 036220 005204          INC      R4          ; RECEIVE INTERRUPT OCCURRED.
7118 036222 116705 144014    MOV8     BRLEVL,R5    ; GET THE EXPECTED INTERRUPT BR LEVEL.
7119 036226 120405          CMP8     R4,R5         ; COMPARE THE INTERRUPT BR LEVEL WITH EXPECTED.
7120 036230 001406          BEQ      24$          ; SKIP THE ERROR IF BR LEVEL IS CORRECT.
7121                                ; REPORT "RX INTERRUPT GENERATED AT WRONG BR LEVEL: ..."
7122 036232 012701 015000    MOV      @EM3004,R1    ; SELECT ERROR MESSAGE FOR THE ERROR CALL.
7123 036236          ERRDF     3006,EM3001,ER3001;    >>>>> ERROR #3006 <<<<<.
                                TRAP      C$ERDF
                                .WORD     3006
                                .WORD     EM3001
                                .WORD     ER3001
                                036236 104455
                                036240 005676
                                036242 014577
                                036244 017066

7124
7125                                ;*
7126                                ; TEST FOR INTERRUPTS OCCURRING IN THE PROPER ORDER.
7127                                ;*
7128 036246 032703 040000    24$: BIT     @BIT14,R3    ; CHECK THE IMPROPER INT ORDER ERROR FLAG.
7129 036252 001406          BEQ      26$          ; SKIP ERROR REPORT IF ERROR DID NOT OCCUR.
7130                                ; REPORT "TX INTERRUPT GIVEN PRECEDENCE OVER SIMULTANEOUS RX INT."
7131 036254 012701 015054    MOV      @EM3005,R1    ; SELECT THE ERROR MESSAGE FOR INDIRECT PRINT.
7132 036260          ERRDF     3007,EM3001,ER0503;    >>>>> ERROR #3007 <<<<<.
                                TRAP      C$ERDF
                                .WORD     3007
                                .WORD     EM3001
                                .WORD     ER0503
                                036260 104455
                                036262 005677
                                036264 014577
                                036266 016466

7133                                ;*
7134                                ; CLEAN UP, EXIT THE TEST.
7135                                ;*
7136 036270 004767 164470    26$: JSR      PC,TXIE0    ; CLEAR TRANSMITTER INTERRUPTS.
7137 036274 004767 163702    JSR      PC,RXIE0    ; CLEAR RECEIVER INTERRUPTS.
7138 036300 005067 144046    60$: CLR      CTRLCF    ; INDICATE THAT WE ARE NOT WITHIN A TEST.
7139 036304          SETPRI    @PRI07              ; DISABLE ALL INTERRUPTS.
                                MOV      @PRI07,R0
                                TRAP      C$SPRI
                                036304 012700 000340
                                036310 104441

7140 036312          ENDTST
                                L10056:
                                036312 104401          TRAP      C$ETST
7141

```

HARDWARE TEST

- DIABMP -

```

7143 .SBTTL HARDWARE TEST DIABMP
7144 ;* .....
7145 ;* DIAGNOSTIC FIELD (BMP) TEST
7146 ;* THIS TEST VERIFIES THAT A REQUEST TO THE DUT TO REPORT BMP STATUS
7147 ;* CODES IS COMPLIED WITH, WITHIN THE SPECIFIED TIME.
7148 ;* ALL ACTIVE LINES ARE TESTED.
7149 ;* .....
7150 ;
7151 036314 BGNTEST
7152 036314 SETPRI #PRI05 ;ALLOW LTC INTERRUPTS. T26::
7153 036314 012700 000240 MOV #PRI05,R0
7154 036320 104441 TRAP C$SPRI
7155 000032 TNUM -- TNUM + 1 ;INCREMENT THE ASSEMBLY TIME TEST COUNTER.
7156 036322 012767 000032 143776 MOV #TNUM,TSTNUM ;SET UP THE TEST NUMBER. (31)
7157 036330 012767 177777 144014 MOV #1,CTRLCF ;INDICATE THAT WE ARE IN A TEST.
7158 036336 012767 000001 145422 MOV #1,ERRTYP ;SET ERROR TYPE AS FATAL IN ERROR TABLE.
7159 036344 012767 006035 145416 MOV #3101,ERRNBR ;SET THE FIRST ERROR NUMBER IN ERROR TABLE.
7160 036352 012767 015146 145412 MOV #EM3101,ERRMSG ;SET ERROR MESSAGE ADDRESS IN ERRTABL.
7161 036360 012767 017420 145406 MOV #ER9101,ERRBLK ;SELECT THE CORRECT ERROR REPORTING ROUTINE.
7162 ;*
7163 ; RESET THE DUT TO A KNOWN STATE, REMOVE THE STATUS CODES FROM THE FIFO.
7164 ; CLEAR TX AND RX INTERRUPT ENABLE BITS IN THE CSR.
7165 ; THIS SUBROUTINE REPORTS ERROR >>>> 3101 <<<<.
7166 ;
7167 JSR PC,CLNRST ;RESET THE DMV 11, REPORT ANY ERRORS FOUND.
7168 BCC 60$ ;RESET FAILURE?, ABORT THIS TEST.
7169 ;*
7170 ; TEST ALL ACTIVE LINES INDIVIDUALLY.
7171 ; WRITE THE REQUEST CODE TO THE DIAGNOSTIC FIELD IN THE LPR REGISTER.
7172 ; VERIFY THAT A BMP CODE IS RETURNED WITHIN THE CORRECT TIME.
7173 ;
7174 036374 016705 143636 MOV ACTLNS,R5 ;GET THE ACTIVE LINE BIT MAP.
7175 036400 005004 CLR R4 ;CLEAR THE LINE NUMBER COUNTER.
7176 036402 016703 143636 MOV CSRA,R3 ;GET THE ADDRESS OF THE DUT'S CSR.
7177 036406 000241 2$: CLC ;CLEAR THE CARRY BIT PRIOR TO SHIFTING BIT MAP.
7178 036410 006005 ROR R5 ;SHIFT THE BIT MAP INTO THE CARRY BIT.
7179 036412 103064 BCC 8$ ;DO NOT TEST THE LINE IF IT IS INACTIVE.
7180 ;*
7181 ; SELECT THE LINE UNDER TEST.
7182 ; WRITE THE BMP REQUEST CODE TO THE DIAG FIELD IN THE LPR REGISTER.
7183 ;
7184 036414 012767 006036 145346 MOV #3102,ERRNBR ;SET THE ERROR NUMBER TO 3102.
7185 036422 010413 MOV R4,(R3) ;SELECT THE LINE CURRENTLY UNDER TEST.
7186 036424 052777 000002 143616 BIS #2,ALPRA ;WRITE THE BMP REQUEST CODE TO THE LPR.
7187 ;*
7188 ; WAIT FOR BMP REQUEST CODE TO BE CLEARED, REPORT ERROR IF TIME-OUT
7189 ; OCCURS.
7190 ;
7191 036432 012701 010764 MOV #10764,R1 ;TEST BIT 1, TIMEOUT OF 500 MILLI SECS.
7192 036436 016702 143606 MOV LPRA,R2 ;PASS THE ADDRESS OF THE REGISTER TO TEST.
7193 036442 004767 164536 JSR PC,WAIBIC ;WAIT FOR REQUEST CODE TO CLEAR.
7194 036446 103042 BCC 6$ ;GO REPORT ERROR IF CODE DID NOT CLEAR IN TIME.
7195 ;*
7196 ; WAIT FOR BMP CODE TO APPEAR IN THE FIFO, REPORT ERROR IF TIME OUT
; OCCURS.
;

```

HARDWARE TEST

- DIABMP -

```

7197 036450 005267 145314      INC      ERRNBR      ;SET ERROR NUMBER TO 3103.
7198 036454 012701 070012      MOV      #70012,R1      ;TEST BIT 7, TIMEOUT OF 10 MILLI SECS.
7199 036460 016702 143560      MOV      CSRA,R2      ;PASS THE ADDRESS OF THE REGISTER TO TEST.
7200 036464 004767 164570      JSR      PC,WAIBIS      ;WAIT FOR RX DATA AVAILABLE TO SET.
7201 036470 103031              BCC      6$      ;GO REPORT ERROR IF CODE DID NOT CLEAR IN TIME.
7202                          ;
7203                          ; READ THE BMP CODE (IF IT IS THERE) FROM THE RBUF REGISTER.
7204                          ; DETERMINE IF IT IS A VALID BMP CODE.
7205                          ; VERIFY THE BMP CODE WAS RECEIVED FROM THE CORRECT CHANNEL.
7206                          ; IF THE BMP CODE DOES NOT INDICATE DUT RUNNING OK, THEN SAVE IT ON
7207                          ; THE QUEUE TO BE REPORTED IN A LATER TEST.
7208                          ;
7209 036472 005267 145272      INC      ERRNBR      ;SET ERROR NUMBER TO 3104.
7210 036476 017702 143544      MOV      @RBUFA,R2      ;GET THE BMP CODE FROM THE FIFO.
7211 036502 100024              BPL      6$      ;GO REPORT ERROR IF NO BMP CODE FOUND.
7212 036504 005267 145260      INC      ERRNBR      ;SET ERROR NUMBER TO 3105.
7213 036510 012700 170301      MOV      #170301,R0      ;SET-UP A BMP CODE MASK.
7214 036514 040200              BIC      R2,R0      ;TRY TO CLEAR THE BMP MASK.
7215 036516 001016              BNE      6$      ;GO REPORT ERROR IF IT IS NOT A VALID BMP CODE.
7216 036520 005267 145244      INC      ERRNBR      ;SET THE ERROR NUMBER TO 3106.
7217 036524 010200              MOV      R2,R0      ;COPY THE BMP CODE.
7218 036526 000300              SWAB      R0      ;PUT THE LINE NUMBER IN THE LOW BYTE.
7219 036530 042700 177760      BIC      #177760,R0      ;CLEAR THE UNWANTED BITS.
7220 036534 120400              CMPB      R4,R0      ;DID THE BMP CODE COME FROM THE CORRECT LINE?.
7221 036536 001006              BNE      6$      ;NO; GO REPORT ERROR.
7222 036540 120227 000305      CMPB      R2,#305      ;IS THE BMP CODE A "GOOD ONE"?.
7223 036544 001407              BEQ      8$      ;YES; SKIP SAVING THE BMP CODE ON THE QUEUE.
7224 036546 004767 163514      JSR      PC,SAVBMP      ;SAVE THE BMP CODE ON THE QUEUE.
7225 036552 000404              BR      8$      ;GO SEE IF THERE ARE ANY MORE LINE TO TEST.
7226                          ;
7227 036554 010401              6$:      MOV      R4,R1      ;PASS THE LINE NUMBER TO BE REPORTED.
7228 036556 012702 015202      MOV      #EM3102,R2      ;PASS THE ERROR MESSAGE TO BE REPORTED.
7229                          ;
7230 036562              ERROR      ;'BMP REQUEST BIT BAD ON LINE:"
7231 036562 104460              ;          >>>>> ERROR <<<<<.
7232                          ;          TRAP      C$ERROR
7233                          ;
7234                          ; VERIFY ALL ACTIVE LINES HAVE BEEN TESTED.
7235                          ;
7236 036564 005204              8$:      INC      R4      ;INCREMENT THE LINE NUMBER COUNTER.
7237 036566 005705              TST      R5      ;ARE THERE ANY MORE ACTIVE LINES TO TEST?.
7238 036570 001306              BNE      2$      ;YES; BRANCH TO TEST THE NEXT LINE.
7239 036572 005067 143554      60$:      CLR      CTRLCF      ;INDICATE THAT WE ARE NOT WITHIN A TEST.
7240 036576              ENDTST
7241 036576              L10057:
7242 036576 104401              TRAP      C$ETST

```

HARDWARE TEST

DIABMP -

```

7240
7241
7242
7243
7244
7245
7246
7247
7248
7249
7250
7251 036600
      036600
7252      000033
7253 036600 012767 000033 143520
7254 036606 012767 177777 143536
7255 036614 016702 143704
7256 036620 012703 002526
7257 036624 020203
7258 036626 001411
7259
7260
7261
7262
7263
7264 036630 012701 016011
7265 036634
      036634 104455
      036636 022125
      036640 015674
      036642 017446
7266
7267 036644 012767 002526 143652
7268
7269 036652 005067 143474
7270 036656
      036656
      036656 104401

      .SBTTL  HARDWARE TEST                      REPBMP
      ;... *****
      ;*      REPORT ANY BMP CODES IN THE QUEUE
      ;*      THIS IS A PSEUDO-TEST USED TO REPORT ANY BMP CODES THAT WERE FOUND
      ;*      IN THE DUT'S FIFO DURING PREVIOUS TEST, AND LOGGED IN THE BMP CODE
      ;*      QUEUE.
      ;*      IT IS UNLIKELY THAT RUNNING THIS PSEUDO-TEST ALONE WILL PRODUCE ANY
      ;*      ERROR REPORTS.
      ;*      *****
      ;*      BGNTEST
      ;*
      ;*      T27::
      ;*      TNUM = TNUM + 1      ;INCREMENT THE ASSEMBLY TIME TEST COUNTER.
      ;*      MOV  #TNUM,TSTNUM    ;SET UP THE TEST NUMBER. (93)
      ;*      MOV  #1,CTRLCF      ;INDICATE THAT WE ARE IN A TEST.
      ;*      MOV  BMPCQP,R2      ;GET THE CONTENTS OF THE POINTER.
      ;*      MOV  #BMPCQB,R3     ;GET THE START ADDRESS OF THE QUEUE.
      ;*      CMP   R2,R3         ;SEE IF THE POINTER HAS MOVED FROM THE BASE.
      ;*      BEQ   60$           ;EXIT NO CODES IN THE QUEUE.
      ;*
      ;*      ; THERE IS AT LEAST ONE BMP CODE IN THE QUEUE. REPORT THE ERROR.
      ;*
      ;*      ;REPORT ERROR BMP CODE FOUND IN TEST NN, BMP CODE:NNNNNN"
      ;*
      ;*      MOV  #EM9304,R1      ;PASS THE FIRST MESSAGE TO BE REORTED.
      ;*      ERDF  9301,EM9301,ER9301 ; >>>>> ERROR #9301 <<<<<.
      ;*
      ;*      TRAP  C$ERDF
      ;*      .WORD 9301
      ;*      .WORD EM9301
      ;*      .WORD ER9301
      ;*
      ;*      MOV  #BMPCQB,BMPCQP ;SET POINTER BACK TO THE BEGINING OF THE QUE.
      ;*
      ;*      60$: CLR  CTRLCF      ;INDICATE THAT WE ARE NOT WITHIN A TEST.
      ;*      ENDTST
      ;*
      ;*      L10060:
      ;*      TRAP  C$ETST

```

HARDWARE TEST

- REOPMP -

```

7273 ;*****
7274 ;
7275 ;           FVTB.HWQ
7276 ;
7277 ;*****
7278
7279
7280
7281 .SBTTL  HARDWARE PARAMETER CODING SECTION
7282
7283
7284
7285 ;**
7286 ; THE HARDWARE PARAMETER CODING SECTION CONTAINS MACROS
7287 ; THAT ARE USED BY THE SUPERVISOR TO BUILD P-TABLES.  THE
7288 ; MACROS ARE NOT EXECUTED AS MACHINE INSTRUCTIONS BUT ARE
7289 ; INTERPRETED BY THE SUPERVISOR AS DATA STRUCTURES.  THE
7290 ; MACROS ALLOW THE SUPERVISOR TO ESTABLISH COMMUNICATIONS
7291 ; WITH THE OPERATOR.
7292 ;
7293
7294 036660          BGNHRD
7295 036660          000022
7296 036662
7297
7298
7299
7300
7301
7302
7303
7304
7305 ;DEVICE CSR ADDRESS QUESTION:
7306 036662          GPRMA  HWPTQ1,0,0,160000,177776,YES
7307 036662          000031
7308 036664          036726
7309 036666          160000
7310 036670          177776
7311
7312
7313
7314
7315 ;DEVICE INTERRUPT VECTOR QUESTION:
7316 036672          GPRMA  HWPTQ2,2,0,40,776,YES
7317 036672          001031
7318 036674          036744
7319 036676          000040
7320 036700          000776
7321
7322
7323
7324
7325 ;ACTIVE LINES BIT MAP QUESTION:
7326 036702          GPRMD  HWPTQ3,4,0,MAPLNS,0,177777,YES
7327 036702          002032
7328 036704          036777
7329 036706          000377
7330 036710          000000
7331 036712          177777
7332
7333
7334
7335 ;INTERRUPT BR LEVEL QUESTION:
7336 036714          GPRMD  HWPTQ5,6,0,377,0,6,YES
7337 036714          003032
7338 036716          037025
7339 036720          000377
7340 036722          000000
7341 036724          000006
7342
7343
7344
7345
7346
7347
7348
7349
7350
7351
7352
7353
7354
7355
7356
7357
7358
7359
7360
7361
7362
7363
7364
7365
7366
7367
7368
7369
7370
7371
7372
7373
7374
7375
7376
7377
7378
7379
7380
7381
7382
7383
7384
7385
7386
7387
7388
7389
7390
7391
7392
7393
7394
7395
7396
7397
7398
7399
7400
7401
7402
7403
7404
7405
7406
7407
7408
7409
7410
7411
7412
7413
7414
7415
7416
7417
7418
7419
7420
7421
7422
7423
7424
7425
7426
7427
7428
7429
7430
7431
7432
7433
7434
7435
7436
7437
7438
7439
7440
7441
7442
7443
7444
7445
7446
7447
7448
7449
7450
7451
7452
7453
7454
7455
7456
7457
7458
7459
7460
7461
7462
7463
7464
7465
7466
7467
7468
7469
7470
7471
7472
7473
7474
7475
7476
7477
7478
7479
7480
7481
7482
7483
7484
7485
7486
7487
7488
7489
7490
7491
7492
7493
7494
7495
7496
7497
7498
7499
7500
7501
7502
7503
7504
7505
7506
7507
7508
7509
7510
7511
7512
7513
7514
7515
7516
7517
7518
7519
7520
7521
7522
7523
7524
7525
7526
7527
7528
7529
7530
7531
7532
7533
7534
7535
7536
7537
7538
7539
7540
7541
7542
7543
7544
7545
7546
7547
7548
7549
7550
7551
7552
7553
7554
7555
7556
7557
7558
7559
7560
7561
7562
7563
7564
7565
7566
7567
7568
7569
7570
7571
7572
7573
7574
7575
7576
7577
7578
7579
7580
7581
7582
7583
7584
7585
7586
7587
7588
7589
7590
7591
7592
7593
7594
7595
7596
7597
7598
7599
7600
7601
7602
7603
7604
7605
7606
7607
7608
7609
7610
7611
7612
7613
7614
7615
7616
7617
7618
7619
7620
7621
7622
7623
7624
7625
7626
7627
7628
7629
7630
7631
7632
7633
7634
7635
7636
7637
7638
7639
7640
7641
7642
7643
7644
7645
7646
7647
7648
7649
7650
7651
7652
7653
7654
7655
7656
7657
7658
7659
7660
7661
7662
7663
7664
7665
7666
7667
7668
7669
7670
7671
7672
7673
7674
7675
7676
7677
7678
7679
7680
7681
7682
7683
7684
7685
7686
7687
7688
7689
7690
7691
7692
7693
7694
7695
7696
7697
7698
7699
7700
7701
7702
7703
7704
7705
7706
7707
7708
7709
7710
7711
7712
7713
7714
7715
7716
7717
7718
7719
7720
7721
7722
7723
7724
7725
7726
7727
7728
7729
7730
7731
7732
7733
7734
7735
7736
7737
7738
7739
7740
7741
7742
7743
7744
7745
7746
7747
7748
7749
7750
7751
7752
7753
7754
7755
7756
7757
7758
7759
7760
7761
7762
7763
7764
7765
7766
7767
7768
7769
7770
7771
7772
7773
7774
7775
7776
7777
7778
7779
7780
7781
7782
7783
7784
7785
7786
7787
7788
7789
7790
7791
7792
7793
7794
7795
7796
7797
7798
7799
7800
7801
7802
7803
7804
7805
7806
7807
7808
7809
7810
7811
7812
7813
7814
7815
7816
7817
7818
7819
7820
7821
7822
7823
7824
7825
7826
7827
7828
7829
7830
7831
7832
7833
7834
7835
7836
7837
7838
7839
7840
7841
7842
7843
7844
7845
7846
7847
7848
7849
7850
7851
7852
7853
7854
7855
7856
7857
7858
7859
7860
7861
7862
7863
7864
7865
7866
7867
7868
7869
7870
7871
7872
7873
7874
7875
7876
7877
7878
7879
7880
7881
7882
7883
7884
7885
7886
7887
7888
7889
7890
7891
7892
7893
7894
7895
7896
7897
7898
7899
7900
7901
7902
7903
7904
7905
7906
7907
7908
7909
7910
7911
7912
7913
7914
7915
7916
7917
7918
7919
7920
7921
7922
7923
7924
7925
7926
7927
7928
7929
7930
7931
7932
7933
7934
7935
7936
7937
7938
7939
7940
7941
7942
7943
7944
7945
7946
7947
7948
7949
7950
7951
7952
7953
7954
7955
7956
7957
7958
7959
7960
7961
7962
7963
7964
7965
7966
7967
7968
7969
7970
7971
7972
7973
7974
7975
7976
7977
7978
7979
7980
7981
7982
7983
7984
7985
7986
7987
7988
7989
7990
7991
7992
7993
7994
7995
7996
7997
7998
7999
8000

```

HARDWARE PARAMETER CODING SECTION

7324	036726	103	123	122	HWPTQ1: .ASCII? /CSR ADDRESS: /
	036731	040	101	104	
	036734	104	122	105	
	036737	123	123	072	
	036742	040	000		
7325	036744	111	116	124	HWPTQ2: .ASCII? /INTERRUPT VECTOR ADDRESS: /
	036747	105	122	122	
	036752	125	120	124	
	036755	040	126	105	
	036760	103	124	117	
	036763	122	040	101	
	036766	104	104	122	
	036771	105	123	123	
	036774	072	040	000	
7326	036777	101	103	124	HWPTQ3: .ASCII? /ACTIVE LINE BIT MAP: /
	037002	111	126	105	
	037005	040	114	111	
	037010	116	105	040	
	037013	102	111	124	
	037016	040	115	101	
	037021	120	072	040	
	037024	000			
7327	037025	111	116	124	HWPTQ5: .ASCII? /INTERRUPT BR LEVEL: /
	037030	105	122	122	
	037033	125	120	124	
	037036	040	102	122	
	037041	040	114	105	
	037044	126	105	114	
	037047	072	040	000	
7328					
7329					.EVEN

HARDWARE PARAMETER CODING SECTION

```

7332 ;*****
7333 ;
7334 ;           FVTA.SWQ
7335 ;
7336 ;*****

7338
7339
7340 .SBTTL  SOFTWARE PARAMETER CODING SECTION
7341
7342 ;**
7343 ; THE SOFTWARE PARAMETER CODING SECTION CONTAINS MACROS
7344 ; THAT ARE USED BY THE SUPERVISOR TO BUILD P-TABLES.  THE
7345 ; MACROS ARE NOT EXECUTED AS MACHINE INSTRUCTIONS BUT ARE
7346 ; INTERPRETED BY THE SUPERVISOR AS DATA STRUCTURES.  THE
7347 ; MACROS ALLOW THE SUPERVISOR TO ESTABLISH COMMUNICATION
7348 ; WITH THE OPERATOR.
7349 ;--
7350
7351 037052      BGNSFT
7352 037052      000013
7353 037054
7354                                     .WORD  L10062 L$SOFT/2
7355                                     L$SOFT::
7356
7357 ;UNIT NUMBER PRINTOUT QUESTION:
7358 GPRML  SWPTQ1,0,20,YES
7359
7360                                     .WORD  T$CODE
7361                                     .WORD  SWPTQ1
7362                                     .WORD  20
7363
7364 ;NUMBER OF INDIVIDUAL DATA ERRORS TO REPORT ON A LINE QUESTION:
7365 GPRMD  SWPTQ2,2,0,177777,0,177777,YES
7366
7367                                     .WORD  T$CODE
7368                                     .WORD  SWPTQ2
7369                                     .WORD  177777
7370                                     .WORD  T$LLOLIM
7371                                     .WORD  T$HILIM
7372
7373 ;ROM VERSION PRINTOUT ON FIRST PASS QUESTION:
7374 GPRML  SWPTQ3,0,1,YES
7375
7376                                     .WORD  T$CODE
7377                                     .WORD  SWPTQ3
7378                                     .WORD  1
7379
7380 .EVEN
7381
7382 ENDSFT
7383
7384                                     .EVEN
7385                                     L10062:
7386
7387 SWPTQ1: .ASCIZ  /REPORT UNIT NUMBER AS EACH UNIT IS TESTED: /
7388
7389 037102      122      105      120
7390 037105      117      122      124
7391 037110      040      125      116
7392 037113      111      124      040
7393 037116      116      125      115
7394 037121      102      105      122
7395 037124      040      101      123
7396 037127      040      105      101
7397 037132      103      110      040
7398 037135      125      116      111

```

SOFTWARE PARAMETER CODING SECTION

	037140	124	040	111
	037143	123	040	124
	037146	105	123	124
	037151	105	104	072
	037154	040	000	
7380	037156	116	125	115
	037161	102	105	122
	037164	040	117	106
	037167	040	111	116
	037172	104	111	126
	037175	111	104	125
	037200	101	114	040
	037203	104	101	124
	037206	101	040	105
	037211	122	122	117
	037214	122	123	040
	037217	124	117	040
	037222	122	105	120
	037225	117	122	124
	037230	040	117	116
	037233	040	101	040
	037236	114	111	116
	037241	105	072	040
	037244	000		
7381	037245	122	117	115
	037250	040	126	105
	037253	122	123	111
	037256	117	116	040
	037261	120	122	111
	037264	116	124	117
	037267	125	124	040
	037272	117	116	040
	037275	124	110	105
	037300	040	106	111
	037303	122	123	124
	037306	040	120	101
	037311	123	123	072
	037314	040	000	

SWPTQ2: .ASCIIZ /NUMBER OF INDIVIDUAL DATA ERRORS TO REPORT ON A LINE: /

SWPTQ3: .ASCIIZ /ROM VERSION PRINTOUT ON THE FIRST PASS: /

7382 .EVEN

SOFTWARE PARAMETER CODING SECTION

```

7384
7385
7386      :*****
7387      :
7388      :          FVTSKL6.P11
7389      :
7390      :*****
7391
7392
7393 037316      $PATCH:
7394 037316      .BLKW    24
7395
7402
7403
7404
7405
7406 037366      LASTAD
                                .EVEN
                                .WORD    0
                                .WORD    0
                                037366 000000
                                037370 000000
                                037372
7407 037372      L$LAST:
                                ENDMOD
7408
7409
7410
7411
7412
7413
7414
7415      000001      .END

```

SYMBOL TABLE

ACTLNC	002236	G	CTRLCF	002352	G	DRADRT	002244	G	EM0603	010305	G	EM3005	015054	G
ADR	= 000020	G	C\$AU	= 000052		DROP	025144		EM0701	010470	G	EM3101	015146	G
ADRPTR	020136	G	C\$AUTO	= 000061		DR00MG	006002	G	EM0702	010525	G	EM3102	015202	G
ALTFLD	017626	G	C\$BRK	= 000022		DR02MG	006006	G	EM0703	010705	G	EM9009	015243	G
ASSEMB	= 000010		C\$BSEG	= 000004		DR04MG	006013	G	EM0801	011070	G	EM9010	015267	G
BCOUNT	002372	G	C\$BSUB	= 000002		DR06MG	006017	G	EM0802	011130	G	EM9014	015313	G
BITTBL	002410	G	C\$CEFG	= 000045		DR10MG	006024	G	EM0901	011203	G	EM9017	015407	G
BIT0	= 000001	G	C\$CLCK	= 000062		DR12MG	006033	G	EM0902	011234	G	EM9018	015520	G
BIT00	= 000001	G	C\$CLEA	= 000012		DR14MG	006044	G	EM1001	011270	G	EM9019	015530	G
BIT01	= 000002	G	C\$CLOS	= 000035		DR16MG	006055	G	EM1002	011334	G	EM9020	015545	G
BIT02	= 000004	G	C\$CLP1	= 000006		EDROP	025222		EM1003	011421	G	EM9022	015571	G
BIT03	= 000010	G	C\$CVEC	= 000036		EF.CON	= 000036	G	EM1101	011477	G	EM9023	015610	G
BIT04	= 000020	G	C\$DCLN	= 000044		EF.NEW	= 000035	G	EM1201	011523	G	EM9024	015632	G
BIT05	= 000040	G	C\$DODU	= 000051		EF.PWR	= 000034	G	EM1202	011542	G	EM9026	015650	G
BIT06	= 000100	G	C\$DRPT	= 000024		EF.RES	= 000037	G	EM1203	011626	G	EM9301	015674	G
BIT07	= 000200	G	C\$DU	= 000053		EF.STA	= 000040	G	EM1204	011704	G	EM9302	015714	G
BIT08	= 000400	G	C\$EDIT	= 000003		EF0503	004132	G	EM1205	011740	G	EM9303	015744	G
BIT09	= 001000	G	C\$ERDF	= 000055		EF0505	004137	G	EM1301	011764	G	EM9304	016011	G
BIT1	= 000002	G	C\$ERHR	= 000056		EF1401	004212	G	EM1302	012010	G	ENDETB	003726	G
BIT10	= 002000	G	C\$ERRO	= 000060		EF1402	004314	G	EM1401	012047	G	ENDIT	025064	
BIT11	= 004000	G	C\$ERSF	= 000054		EF1601	004351	G	EM1402	012077	G	ERCNTB	002464	G
BIT12	= 010000	G	C\$ERSO	= 000057		EF1602	004375	G	EM1403	012165	G	ERTBL	002726	G
BIT13	= 020000	G	C\$ESCA	= 000010		EF1603	004437	G	EM1404	012240	G	ERRBLK	003774	G
BIT14	= 040000	G	C\$ESEG	= 000005		EF1604	004501	G	EM1405	012312	G	ERRMSG	003772	G
BIT15	= 100000	G	C\$ESUB	= 000003		EF3001	004576	G	EM1406	012325	G	ERRNBR	003770	G
BIT2	= 000004	G	C\$ETST	= 000001		EF3002	004645	G	EM1407	012340	G	ERRTYP	003766	G
BIT3	= 000010	G	C\$EXIT	= 000032		EF9001	004714	G	EM1408	012352	G	ERSMRF	002462	G
BIT4	= 000020	G	C\$GETB	= 000020		EF9002	004776	G	EM1601	012360	G	ER0101	016066	G
BIT5	= 000040	G	C\$GETW	= 000027		EF9003	005055	G	EM1604	012443	G	ER0201	016404	G
BIT6	= 000100	G	C\$GMAN	= 000043		EF9004	005111	G	EM1701	012511	G	ER0503	016466	G
BIT7	= 000200	G	C\$GPHR	= 000042		EF9005	005146	G	EM1801	012566	G	ER0504	016512	G
BIT8	= 000400	G	C\$GPLO	= 000030		EF9006	005177	G	EM1901	012634	G	ER1401	016560	G
BIT9	= 001000	G	C\$GPRI	= 000040		EF9010	005223	G	EM2001	012711	G	ER1601	016710	G
BMPCQB	002526	G	C\$INIT	= 000011		EF9016	005322	G	EM2002	012753	G	ER1603	017006	G
BMPCQE	002726	G	C\$INLP	= 000020		EF9017	005417	G	EM2101	013026	G	ER3001	017066	G
BMPCQP	002524	G	C\$MANI	= 000050		EF9018	005473	G	EM2102	013071	G	ER9004	017156	G
BOE	= 000400	G	C\$MEM	= 000031		EF9019	005600	G	EM2201	013165	G	ER9007	017260	G
BRLEVL	002242	G	C\$MSG	= 000023		EF9301	005624	G	EM2202	013222	G	ER9008	017336	G
BUFBAS	002726	G	C\$OPEN	= 000034		EF9302	005702	G	EM2203	013314	G	ER9101	017420	G
BUFEND	003726	G	C\$PNTB	= 000014		EM0101	020532	G	EM2301	013421	G	ER9301	017446	G
BUFMID	003326	G	C\$PNTF	= 000017		EM0102	020616	G	EM2302	013457	G	EVL	= 000004	G
BUFPTR	002324	G	C\$PNTS	= 000016		EM0103	006065	G	EM2401	013515	G	E\$END	= 002100	
BUF3QT	003526	G	C\$PNTX	= 000015		EM0201	006123	G	EM2601	013551	G	E\$LOAD	= 000035	
CACHRX	023634	G	C\$QIO	= 000377		EM0202	006171	G	EM2602	013601	G	F\$AU	= 000015	
CACHTX	023662	G	C\$RDBU	= 000007		EM0203	006344	G	EM2603	013672	G	F\$AUTO	= 000020	
CALMSL	017700	G	C\$REFG	= 000047		EM0204	006507	G	EM2604	013760	G	F\$BGN	= 000040	
CKTRAP	020124	G	C\$RESE	= 000033		EM0301	006666	G	EM2605	014054	G	F\$CLEA	= 000007	
CLKBRL	002356	G	C\$REVI	= 000003		EM0302	006731	G	EM2606	014137	G	F\$DU	= 000016	
CLKCSR	002354	G	C\$RFLA	= 000021		EM0303	007071	G	EM2607	014176	G	F\$END	= 000041	
CLKHRZ	002362	G	C\$RPT	= 000025		EM0401	007250	G	EM2608	014272	G	F\$HARD	= 000004	
CLKINT	023710	G	C\$SEFG	= 000046		EM0402	007317	G	EM2609	014343	G	F\$HW	= 000013	
CLKVEC	002360	G	C\$SPRI	= 000041		EM0501	007467	G	EM2610	014422	G	F\$INIT	= 000006	
CLNRST	020154	G	C\$SVEC	= 000037		EM0502	007535	G	EM2611	014514	G	F\$JMP	= 000050	
CLR16W	020176	G	C\$TPRI	= 000013		EM0525	007711	G	EM3001	014577	G	F\$MOD	= 000000	
CNTERR	020220	G	DELAY	020276	G	EM0526	010001	G	EM3002	014630	G	F\$MSG	= 000011	
CSRA	002244	G	DFPTBL	002214	G	EM0601	010071	G	EM3003	014724	G	F\$PROT	= 000021	
CSRO	= 000000	G	DIAGMC	= 000000		EM0602	010125	G	EM3004	015000	G	F\$PWR	= 000017	

SYMBOL TABLE

F\$RPT = 000012	I\$TST = 000041	L\$SPTP 002024 G	MFUNIT 004074 G	RXIE1 022242 G
F\$SEG = 000003	J\$JMP = 000167	L\$STA 002030 G	MMENAB 002404 G	RXINPT 024050 G
F\$SOFT= 000005	LNCTRA 002254 G	L\$SW 002226 G	MMPRES 002402 G	RXINTC 002334 G
F\$SRV = 000010	LNCTRO= 000010 G	L\$TEST 002114 G	MMSRO 002400 G	RXINTF 002336 G
F\$SUB = 000002	LOE = 040000 G	L\$TIML 002014 G	MSG1 016172 G	RXVECA 002232 G
F\$SW = 000014	LOT = 000010 G	L\$UNIT 002012 G	MSG2 016250 G	ROSL0T= 000002 G
F\$TEST= 000001	LPCSLT= 000036 G	L10000 002224	MSG3 016327 G	R1SL0T= 000004 G
GETPRM 024650	LPRA 002250 G	L10001 002232	MSLCNT 002376 G	R2SL0T= 000006 G
GPRSOB 002450 G	LPRO = 000004 G	L10002 016170	MSLGET 020336 G	R3SL0T= 000010 G
G\$CNT0= 000200	L\$ACP 002110 G	L10003 016464	MSLOOP 020452 G	R4SL0T= 000012 G
G\$DELM= 000372	L\$APT 002036 G	L10004 016510	MSTICK 002374 G	R5SL0T= 000014 G
G\$DISP= 000003	L\$AU 025230 G	L10005 016556	NDERPT 002230 G	SAVBMP 022266 G
G\$EXCP= 000400	L\$AUT 002070 G	L10006 016706	NEWPAS 024630	SFPTBL 002226 G
G\$HILI= 000002	L\$AUTO 025100 G	L10007 017004	NEWRES 024622	SKPSTS 022334 G
G\$LOLI= 000001	L\$CCP 002106 G	L10010 017064	NEWSTA 024312	STATA 002252 G
G\$NO = 000000	L\$CLEA 025102 G	L10011 017154	NUMLNS= 000010 G	STAT0 = 000006 G
G\$OFFS= 000400	L\$CO 002032 G	L10012 017256	OOPS 020466 G	SVCGBL= 000000
G\$OFSI= 000376	L\$DEPO 002011 G	L10013 017334	OPTION 002226 G	SVCINS= 000001
G\$PRMA= 000001	L\$DESC 004046 G	L10014 017416	O\$APTS= 000000	SVCSUB= 000001
G\$PRMD= 000002	L\$DESP 002076 G	L10015 017444	O\$AU = 000000	SVCTAG= 000001
G\$PRML= 000000	L\$DEVP 002060 G	L10016 017624	O\$BGNR= 000001	SVCTST= 000001
G\$RADA= 000140	L\$DISP 002124 G	L10017 024236	O\$BGNS= 000001	SWAPO 022412 G
G\$RADB= 000000	L\$DLY 002116 G	L10021 025076	O\$DU = 000001	SWPTQ1 037102
G\$RADD= 000040	L\$DTP 002040 G	L10022 025100	O\$ERRT= 000001	SWPTQ2 037156
G\$RADL= 000120	L\$DTYP 002034 G	L10023 025116	O\$GNSW= 000001	SWPTQ3 037245
G\$RADO= 000020	L\$DU 025120 G	L10024 025226	O\$POIN= 000001	S\$LSYM= 010000
G\$XFER= 000004	L\$DUT 002072 G	L10025 025234	O\$SETU= 000000	TIMER1 002364 G
G\$YES = 000010	L\$DVTY 004036 G	L10026 025524	PAROA 002406 G	TIMER2 002366 G
HELP = 000000	L\$EF 002052 G	L10027 025754	PASCNT 002332 G	TIMER3 002370 G
HOE = 100000 G	L\$ENVI 002044 G	L10030 026220	PCSL0T= 000016 G	TNUM = 000033 G
HWPTQ1 036726	L\$ERRT 003766 G	L10031 026416	PNT = 001000 G	TP4FLG 002346 G
HWPTQ2 036744	L\$ETP 002102 G	L10032 026610	PREGRT 004020 G	TP4RTN 024134 G
HWPTQ3 036777	L\$EXP1 002046 G	L10033 027026	PREG05 003776	TP4VEC 002344 G
HWPTQ5 037025	L\$EXP4 002064 G	L10034 027234	PRI = 002000 G	TSABRT 022462 G
IBE = 010000 G	L\$EXP5 002066 G	L10035 027442	PRI00 = 000000 G	TSTNUM 002326 G
IDU = 000040 G	L\$HARD 036662 G	L10036 027630	PRI01 = 000040 G	TXAD1A 002256 G
IER = 020000 G	L\$HIME 002120 G	L10037 030044	PRI02 = 000100 G	TXAD10= 000012 G
IESTAT 002330 G	L\$HPCP 002016 G	L10040 030272	PRI03 = 000140 G	TXAD2A 002260 G
ISR = 000100 G	L\$HPTP 002022 G	L10041 030554	PRI04 = 000200 G	TXAD20= 000014 G
IXE = 004000 G	L\$HW 002214 G	L10042 031132	PRI05 = 000240 G	TXBFCA 002262 G
I\$AU = 000041	L\$ICP 002104 G	L10043 031532	PRI06 = 000300 G	TXBFCA= 000016 G
I\$AUTO= 000041	L\$INIT 024246 G	L10044 031750	PRI07 = 000340 G	TXCHA 002246 G
I\$CLN = 000041	L\$LADP 002026 G	L10045 032172	PUFIFO 020714 G	TXCHRO= 000002 G
I\$DU = 000041	L\$LAST 037372 G	L10046 032444	RBUFA 002246 G	TXDSBL 022574 G
I\$HRD = 000041	L\$LOAD 002100 G	L10047 032722	RBUFO = 000002 G	TXENBL 022670 G
I\$INIT= 000041	L\$LUN 002074 G	L10050 033036	RDPDR 020776 G	TXIE0 022764 G
I\$MOD = 000041	L\$MREV 002050 G	L10051 033254	REGTST 021216 G	TXIE1 023024 G
I\$MSG = 000041	L\$NAME 002000 G	L10052 033516	REPSMR 021426 G	TXINTC 002340 G
I\$PROT= 000040	L\$PRIO 002042 G	L10053 034024	RESETT 021454 G	TXINTF 002342 G
I\$PTAB= 000041	L\$PROT 024240 G	L10054 034370	RMATBB 002304 G	TXINTR 024156 G
I\$PWR = 000041	L\$PRT 002112 G	L10055 035422	ROLDAP 021566 G	TXVECA 002234 G
I\$RPT = 000041	L\$REPP 002062 G	L10056 036312	RSTRPT 021620 G	T\$ARGC= 000003
I\$SEG = 000041	L\$REV 002010 G	L10057 036576	RXBDTX= 000030 G	T\$CODE= 000130
I\$SETU= 000041	L\$RPT 024232 G	L10060 036656	RXBETX= 000020 G	T\$ERRN= 022125
I\$SFT = 000041	L\$SOFT 037054 G	L10061 036726	RXBFUL= 000100 G	T\$EXCP= 000000
I\$SRV = 000041	L\$SPC 002056 G	L10062 037102	RXBRRT 023760 G	T\$FLAG= 000050
I\$SJB = 000041	L\$SPCP 002020 G	MAPLNS= 000377 G	RXIE0 022202 G	T\$GMAN= 000000

SYMBOL TABLE

T\$HILI= 177777	T\$TEST= 000033	T\$TES= 010060	T22 033520 G	UNITN 002240 G
T\$LAST= 000001	T\$TSTM= 177777	T1 025236 G	T23 034026 G	UNSDIV 023050 G
T\$LOI= 000000	T\$TSTS= 000001	T10 027632 G	T24 034372 G	WAIBIC 023204 G
T\$LSYM= 010000	T\$AU= 010025	T11 030046 G	T25 035424 G	WAIBIS 023260 G
T\$LTNO= 000033	T\$AUT= 010022	T12 030274 G	T26 036314 G	WOPDR 023334 G
T\$NEST= 177777	T\$CLE= 010023	T13 030556 G	T27 036400 G	WORD1 002350 G
T\$NSO= 000000	T\$DU= 010024	T14 031134 G	T3 025756 G	WTWLC 023530 G
T\$NS1= 000005	T\$HAR= 010061	T15 031534 G	T4 026222 G	WTWLS 023560 G
T\$PTNU= 000000	T\$HW= 010000	T16 031752 G	T5 026420 G	WTWLP 023604 G
T\$SAVL= 177777	T\$INI= 010021	T17 032174 G	T6 026612 G	X\$ALWA= 000000
T\$SEGL= 177777	T\$MSG= 010016	T18 032446 G	T7 027030 G	X\$FALS= 000040
T\$SUBN= 000000	T\$PRO= 010020	T19 032724 G	T8 027236 G	X\$OFFS= 000400
T\$TAGL= 177777	T\$RPT= 010017	T2 025526 G	T9 027444 G	X\$TRUE= 000020
T\$TAGN= 010063	T\$SOF= 010062	T20 033040 G	UAM = 000200 G	\$PATCH 037316 G
T\$TEMP= 000000	T\$SW= 010001	T21 033256 G	UNBTB 002264 G	

. ABS. 037372 000
000000 001

ERRORS DETECTED: 0

VIRTUAL MEMORY USED: 28671 WORDS (112 PAGES)
DYNAMIC MEMORY: 20060 WORDS (77 PAGES)
ELAPSED TIME: 00:04:36
VAAOE0.OBJ,VAAOE0.LST/ SP=SVC34R/ML,VAAOE0.P11